1999

Real time fuzzy logic and PID implemetation and control in labview

Peter Bejoum Eyabi
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Eyabi, Peter Bejoum, "Real time fuzzy logic and PID implemetation and control in labview" (1999). *Master's Theses*. 1805.

DOI: <https://doi.org/10.31979/etd.gdx2-gdst>
https://scholarworks.sjsu.edu/etd_theses/1805

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

REAL TIME FUZZY LOGIC AND PID IMPLEMENTATION AND
CONTROL IN LABVIEW

A Thesis

Presented to

The Faculty of the Department of Aerospace and Mechanical Engineering
San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Peter Bejoum Eyabi

May 1999

UMI Number: 1394517

UMI Microform 1394517

Copyright 1999, by UMI Company. All rights reserved.

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI

**300 North Zeeb Road
Ann Arbor, MI 48103**

© 1998

Peter Bejoum Eyabi

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF AEROSPACE
AND MECHANICAL ENGINEERING

Prof. Hank Pernicka

Prof. Ji Wang

Prof. Burford Furman

APPROVED FOR THE UNIVERSITY

William Fish

ABSTRACT
REAL TIME FUZZY LOGIC AND PID IMPLEMENTATION
AND CONTROL IN LABVIEW

by Peter B. Eyabi

The scope of this thesis will be limited to the investigation and comparison of the characteristics and performances of Fuzzy Logic Control to those of Proportional, Integral and Derivative Control (PID), in different control applications. The controller performance and characteristics will be analyzed through computer simulation, and real time implementation

Fuzzy Logic controllers with different characteristics will be designed in Matlab (control systems design software). The process to be controlled is that of a paper-stock flow box (paper manufacturing machine flowbox).

Implementation of fuzzy logic control will be by a data acquisition board, to control the temperature of an aluminum plate, subjected to external disturbances. A fuzzy logic controller will be designed and implemented in Labview™ (data acquisition software).

Each controller will be subjected to a large disturbance input. The process to be controlled does not have an accurate transfer function.

Simulation results demonstrate that fuzzy logic controllers are faster in response, compared to the PID. The number of rules, as well as the degree of overlap of

membership functions affect controller output. There is need for more extensive research in the area of fuzzy logic implementation.

Acknowledgments

I am particularly grateful to my thesis advisor, Prof. Ji Wang, for his patience, and for teaching me much of what I know.

This work is dedicated to my one, and only son, Jorgen Cortez Eyabi, and my mother, Comfort Eyabi, for their love and support.

	Table of Contents	Page Numbers
1.0	Introduction	2
2.0	Overview of fuzzy logic	3
2.1	Limitations of conventional controllers	3
2.2	Benefits of fuzzy controllers	4
3.0	The structure and operation of a fuzzy controller	5
3.1	Linguistic variables and terms	6
3.2	Rule-based systems	7
4.0	Fuzzy logic controller structure	8
4.1	Defuzzification using linguistic variables	10
5.0	Non-linear control of paper-stock flowbox	12
5.1	Description of process	12
5.2	Process and models	13
6.0	Closed loop control	17
6.1	Dual non-interacting control	18
7.0	Simulation	19
8.0	Design of fuzzy controller for paper stock flow	21
8.1	Fuzzification	22
8.1.1	Pressure	22
8.1.2	Level	24

	Table of Contents	Page Numbers
8.2	Rules Notation	25
9.0	Simulation	29
9.1	Results	33
10.0	Temperature control of aluminum plate using Labview	36
10.1	Introduction	36
10.1.1	Electronic circuit	36
11.0	Design of controller	38
11.1	PID controller	39
11.2	Fuzzy logic controller	42
12.0	Results	50
13.0	Discussion	58
14.0	Conclusion	62
15.0	Appendix	
A.	Input membership function plot for " pressure error," of forty nine rule controller	64
	Output membership function plot for "pressure," of forty nine rule controller	64

APPENDIX	DESCRIPTION	PAGE NUMBER
B	Input membership function plot for "level rate," of forty nine rule controller	65
	Input membership function plot for "level error," of forty nine rule controller	65
C	Output membership function plot for "level," of forty nine rule controller	66
	Input membership function plot for "rate," of five rule controller	66
D	Input membership function plot for "error," of five rule controller	67
	Output membership function plot for "du," of forty nine rule controller	67
E	PID Virtual instrument or VI	68
F	Input membership function "temp rate" for PET1.fc	69
G	Output membership function "voltage" for PET1.fc	70
H	Input membership function "temp error" for PET5.fc	71
I	Output membership function "voltage" for PET5fc	72
16.0	Bibliography	73

LIST OF TABLES

TABLE NO.	DESCRIPTION	PAGE NO.
1	Controller parameters for non interacting control	19
2	Rules Table for "five rule" Fuzzy controller	25
3	Rules Table for "forty nine" Fuzzy controller	26
4	Rules Table for "PET1.fc" Fuzzy controller	45
5	Rules Table for "PET5.fc" Fuzzy controller	46

LIST OF FIGURES

FIGURE NO	DESCRIPTION	PAGE NUMBER
1.0	Modeling Uncertainty by fuzzy set membership	7
2.0	Diagram of the structure of a Fuzzy logic controller	9
3.0	Defuzzification by center of Area method	11
4.0	Cross sectional view of Flow box	13
5.0	Block diagram of non-interacting control	19
6.0	Subsystem block diagram of paperstock process dynamics	21
7.0	Input membership function plot for "pressure rate," of forty nine rule controller	24
8.0	Block diagram of paperstock process controlled by fuzzy logic/PID	31
9.0	Three dimensional control surface for "forty nine rule" pressure controller	32
10.0	Three dimensional control surface for "forty nine rule" level controller	32
11.0	Three dimensional control surface for "five rule" controller	33
12.0	Pressure/Level response to step input for 49 rule controlled process	34

FIGURE NO	DESCRIPTION	PAGE NUMBER
13.0	Pressure/Level response to step input for five rule controlled process	35
14.0	Schematic of control system layout	37
15.0	Wiring diagrams for sensor, heating and cooling circuits	39
16.0	PID Control block diagram	41
17.0	PID front panel	42
18.0	Input membership function "temp error" for PET1.fc	44
19.0	Input membership function "temp rate" for PET5.fc	48
20.0	Fuzzy logic control block diagram	49
21.0	Transient response of PID controlled process to large disturbance	52
22.0	Transient response of fuzzy logic (PET.fc1) controlled process to large disturbance	53
23.0	Transient response of fuzzy logic controller(PET.fc5)	54
24.0	Transient response of fuzzy logic (PET.fc5) controlled process to large disturbance	55
25.0	Transient response of fuzzy logic (PET.fc5) controlled process to small disturbance	56

FIGURE NO	DESCRIPTION	PAGE NUMBER
26	(Top) Transient Response of PID versus Fuzzy Logic (Middle) Temperature error of PID versus Fuzzy Logic (Bottom) Control Input of PID versus Fuzzy Logic	57
27.0	Control surface of PET5.fc	59
28.0	Control surface of PET1.fc	60

The purpose of this thesis is to investigate and compare the characteristic and performance of Fuzzy Logic Control to that of Proportional, Integral, and Differential Control (PID), in different control applications. A study of the theory of fuzzy logic controller design will be presented, and an attempt will be made to examine the effects of design criteria on controller characteristics and performance. The controller performance and characteristics will be analyzed through computer simulation and real time implementation. In each case, the fuzzy logic controller performance will be compared to that of a conventional PID controller. The intention is not to present fuzzy logic control as a better alternative to PID control, but as an available tool that could be effective in certain control applications, where traditional control methods have been ineffective.

There will be three sections to this thesis:

A) **Overview of Fuzzy Logic:** The theory of fuzzy logic and the fuzzy set theory will be presented. This will be used to explain the design and structure of a fuzzy logic controller.

B) **Design and Simulation:** Fuzzy Logic controllers with different characteristics will be designed in Matlab (control systems design software). A simulation study will be performed to investigate and compare the performances of these controllers to those of a conventional PID controller. The process to be controlled is that of a paper-stock flow box (paper manufacturing machine flowbox). This process demonstrates a simple, but practical example of an interacting multivariable-control system, with a nonlinear

dynamics. The complexity of the system dynamic equations, and its nonlinear characteristics, present a practical example of a system, on which PID control (linear control) could be ineffective, and fuzzy logic control (nonlinear control) could be effective and perhaps, perform better.

C) Design and Implementation: Due to the difficulty and high cost of obtaining a paper-stock flowbox, an alternative process will be used to demonstrate and implement fuzzy logic control. This will be by a data acquisition board, to control the temperature of an aluminum plate, subjected to external disturbances. A fuzzy logic controller will be designed and implemented in Labview (data acquisition software), to regulate the set point of a temperature control system. Each controller will be subjected to a large disturbance input, in the form of air flow from a fan. Turning on the fan adjacent to the heated plate, has the effect of reducing the temperature of the plate. Again, its performance will be compared to that of a conventional PID controller. The process to be controlled does not have an exact dynamic equation and thus, no accurate transfer function.

2.0 A) OVERVIEW OF FUZZY LOGIC

Fuzzy Logic is a method of rule-based decision making used for expert systems and process control that emulates the rule-of-thumb thought process used by human beings (National Instruments Corporation, 1997, page 2-1). It is based on the Fuzzy set theory, which is defined quantitatively by linguistic terms called membership functions. The membership function specifically defines degrees of membership, which might be any value between 0 and 1.

With membership functions defined for system inputs and outputs, a rule base of IF-THEN type conditional rules, can be formulated. Such a rule base and the corresponding membership functions, are employed to analyze controller inputs and determine controller outputs by the process of fuzzy logic inference. This method, known as the *Mamdani-type* fuzzy processing (Roger, page 2-53) will be the method used to design the fuzzy controllers.

There are certain limitations of conventional controllers, which make them inappropriate to be used in certain applications.

2.1 Limitations of conventional controllers:

- ◇ *Plant non-linearity*: The object under control may have a non-linear dynamics, and this presents stability problems when controlled with a linear controller (conventional PID controller). This is because the dynamic equation is usually linearized about an operating point. The controller is then designed to work in this region. If the dynamics

of the system changes away from this operating point, the controller performance diminishes.

- ◇ *Plant uncertainty*: Certain plants may have an inaccurate mathematical model, or may be too complex and non-linear to be modeled.
- ◇ *Temporal behavior*: Plants may have properties and constraints that vary with time, and the conventional controller parameters become invalid when these changes occur.
- ◇ *Uncertainty in measurements*: Certain uncertain plant parameters may not have stochastic noise models, so it becomes difficult to apply conventional statistical methods to design a conventional controller for that plant.
- ◇ *Multivariables*: Plants with multi-inputs and multi-outputs usually have a lot of interactions and dependencies, rendering conventional controllers inappropriate.

Fuzzy control, however, presents certain characteristics that make it applicable in certain cases where conventional control may not perform satisfactorily.

2.2 Benefits of fuzzy controllers:

- ◇ *Easy to understand*: The mathematical concepts behind fuzzy logic controller design are very simple and fundamental. Fuzzy logic itself is based on natural language.
- ◇ *Robustness*: Fuzzy controllers are more robust than PID controllers, because they can cover a much wider range of operating conditions than a PID, and can operate with a wide variety of noise and disturbances. This characteristic is important in applications

where there is imprecise data, uncertain facts, or where human experience is relevant in operating a plant efficiently.

- ◇ *Cheap*: It is cheaper to develop and implement fuzzy controllers than conventional controllers to perform identical functions.
- ◇ *Customizable*: Fuzzy controller characteristics could easily be changed, by changing the rules. It can also be used to regulate and optimize the performance of a PID controller.

3.0 THE STRUCTURE AND OPERATION OF A FUZZY CONTROLLER

One of the basic concepts in fuzzy logic is the mathematical description of the linguistic uncertainty of a fuzzy set. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. The elements of a fuzzy set are related by a *membership function*. This is a curve that defines how each element in the input space, is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the *universe of discourse*.

If X is the universe of discourse, and its elements are denoted by x , then a fuzzy set A , in X , is defined as a set of ordered pairs:

$$A = \{x, \mu_A(x) \mid x \in X\}$$

$\mu_A(x)$ is called the membership function (or MF) of x in A . The function maps each element of X to a membership value between 0 and 1.

For example, consider the degree of dirtiness in a set of dirty laundry (DL). Let the degree of dirtiness be assigned an arbitrary value of one to ten, one being least dirty and ten being most dirty. It is very difficult to model an uncertain fact, like degree of dirtiness, with a dual membership value of TRUE or FALSE. It allows for intermediate membership degrees, since dirtiness is a "more-or-less" relative concept, i.e., it is fuzzy. The various degrees of dirtiness belong to fuzzy set. A graphical representation of such a set is shown below in Figure 5. Each degree of dirtiness is associated with a certain degree of membership ($\mu(L)$), to the *dirty laundry* set. The function $\mu(L)$ is called degree of membership of the element ($D \in DL$) to the fuzzy set *dirty laundry*. The degree of dirtiness is called the characteristic quantity or base variable L to the universe DL . It should be noted that μ ranges from 0 to 1, representing absolutely no membership to the set, and complete membership, respectively. The only condition a membership function must really satisfy is that it must vary between 0 and 1. There are several membership function types, one of the simplest, being the *triangular* membership function, will be referred to most often.

3.1 Linguistic variables and terms: The primary building block of fuzzy logic systems is the linguistic variable. This variable is one with a value, that is a language expression, referring to some quantity of interest, related to the system of interest. It translates real values into linguistic values. Linguistic terms represent possible values of a

linguistic variable. Each linguistic term is represented by a fuzzy set defined by a membership function.

$\mu(L)$ Membership(dirty laundry)

Figure 1.0 Modeling Uncertainty by Fuzzy Set Membership
(Mapping a physical quantity, degree of dirtiness to a fuzzy set through a membership function).

3.2 Rule-based systems: The rule-based decision-making process is another fundamental fuzzy logic concept. Usually, certain complex plants are being managed by human operators without having knowledge of the dynamics of the plant. Whenever it becomes difficult for an accurate system mathematical model to be formed, a fuzzy logic linguistic representation of engineering knowledge is used to implement a control strategy. A control strategy can be defined, by using IF-THEN rules such as the following:

IF <situation> **THEN** <action>

The above rule format describes the necessary reaction or conclusion to a certain situation or condition.

4.0 FUZZY LOGIC CONTROLLER STRUCTURE

Firstly all sensor signals must be translated into linguistic variables. This is called *Fuzzification* because it uses fuzzy sets for translating real variables into linguistic variables (see Figure 2.0).

Once all input variable values are translated into corresponding linguistic variable values, the *Fuzzy Inference* step is executed to derive a conclusion from the rule base that represents the control strategy. The result of this step is a linguistic value for the output variable. The fuzzy inference step consists of two components:

- Aggregation: Evaluation of the IF part (condition) of each rule
- Composition: Evaluation of the THEN part (conclusion) of each rule

For example:

IF laundry is <i>less dirty</i> (degree of truth = 0.2)	AND minimum	<i>color dark,</i> (degree of truth = 0.5)
---	-----------------------	---

THEN set water temperature *low*

2. IF laundry is <i>average dirty</i> (degree of truth = 0.3)	AND	<i>color dull,</i> (degree of truth = 0.7)
---	------------	---

THEN set water temperature *medium*

3. IF laundry is <i>very dirty</i> (degree of truth = 0.5)	AND	<i>color bright,</i> (degree of truth = 0.8)
--	------------	---

THEN set water temperature *high*

Figure 2.0 Diagram of the structure of a Fuzzy Logic Controller
(The procedure for a fuzzy logic controller design usually starts from identifying the measured quantities and then follows the structure shown above)

In the above set of rules, *laundry* and *color* are *input membership functions*, while *temperature* is an *output membership function*. Also, *less* and *dark* are *input terms*, while *low* is an *output term*. In the example, the IF part of each rule combines two linguistic terms from different linguistic variables with the word AND. Because the linguistic terms represent conditions that are partially true, non logic operators are used to model the words AND, OR, and NOT.

$$\text{AND: } \mu_A * B = \min(\mu_A, \mu_B)$$

$$\text{OR: } \mu_A + B = \max(\mu_A, \mu_B)$$

$$\text{NOT: } \mu_{\neg A} = 1 - \mu_A$$

The minimum operator represents the word AND. It is applied in the aggregation step to calculate a degree of truth for the IF condition of each rule in the rule base, indicating how adequately each rule describes the current situation. For example, in rule indicated by (1), if *less dirty* has a degree of truth of 0.2, and *color dark* has a degree of truth of 0.5, the minimum operator is applied in the *aggregation* step since they are linked by the word "and". The resulting action is to set the temperature with a degree of 0.2. The rule indicated by (2), leads to a conclusion with a degree of truth of 0.3.

In the above example, if only the first rule is a valid description of a current situation it is called an *active rule*, while the others are called *inactive*.

If two or more rules are activated for a current situation, they are logically linked by the word OR. Because the resulting conclusion of each rule is partially true, the maximum operator, OR, is applied. For example, the first two rules assert different degrees of truth, the first asserting a linguistic term *low*, with a degree of truth of 0.2, while the second asserting a linguistic term *medium*, with a degree of truth of 0.3. Because the two rules are related by the OR operator, the output of the fuzzy inference for the linguistic term is the maximum value for 0.7. This type of fuzzy inference is called *Max-Min inference*. Individual weights are associated to each rule as an optimization procedure in the fuzzy system.

4.1 Defuzzification Using Linguistic Variables

The *Defuzzification* step translates the linguistic result back into a real value representing the current value of the control variable. The fuzzy inference process resulted in a linguistic value for the output of 0.2 , 0.3 and 0.5. To use this linguistic value to adjust the temperature, it must be translated into a real (physical) value. Thus, the *defuzzification* step is initiated to form a crisp, real value. This value must be the best final conclusion received from the outputs of all the active rules generated by the inference engine. One of the most common methods for calculating the best compromise is the *Center-of-Area* method (CoA), also called the *Center-of-gravity* (Cog), as shown on Figure 3.0 below. (See National Instruments Corporation, March 1997 page 2-18).

Figure 3.0 Defuzzification by Center of Area Method
(Linguistic values with different outputs or degree of membership are defuzzified, or translated into real values. The final output being the centroid of the superimposed areas).

defuzzified result: $T = 50^{\circ} \text{F}$ Water Temperature

Following the Center-of-Area method, all membership functions representing the conclusion terms are truncated at the degree of validity of the rule to which the conclusion term belongs. The areas under the resulting function of all truncated terms are superimposed. Numeric integration is applied to find the geometric center of the resulting area to determine the crisp compromise value. Other defuzzification methods include: Center-of-maximum, modified Center-of-Area, Mean-of-Maximum, etc. (See National Instruments Corporation, March 1997 page 2-16).

5.0 B) **NONLINEAR CONTROL OF A PAPER-STOCK FLOWBOX**

5.1 **Description of Process:**

Paper stock enters the box at a controlled flowrate and slurry from the flow box flows through a slit orifice, and it is distributed over the width of a moving wire mesh belt. The box is closed at the top and air pressure is applied above the slurry (see Figure 4.0). The reasons being:

- ◇ Slurry flow should be constant and evenly distributed over the belt.
- ◇ A predetermined rate of slurry flow from the orifice should be maintained. Pressure and slurry level should be controlled, since a small change in pressure will cause a large change in slurry flowrate from the orifice. Without level control, the slurry delivery rate and outflow will cause the level to rise and fall with time. (Gunn, and Sinha, IEE. 1983).

5.2 Process and Models:

a = cross-sectional area of air bleed

a_s = cross-sectional area of slurry slit

A = horizontal cross-sectional area of flow box

Figure 4.0 Cross sectional view of flowbox

(The two inputs into the flowbox are air and slurry flowrate. These are used to control the slurry height and air pressure to give a desired slurry flowrate and uniform distribution out of the slit)

C_o = orifice constant for air

C_{os} = orifice constant for slurry

g = acceleration due to gravity

h = level of slurry above slit

h_r = reference value of slurry level inside box
 H = total height of box above slit
 J_p = integral square error of pressure
 J_h = integral square error of level
 K_{cp} = gain of PID controller in pressure loop
 K_{ch} = gain of PID controller in level loop
 M = mass of air inside box
 p = air pressure inside box
 p_a = ambient pressure
 p_r = reference value of air pressure inside box
 q = flow of air into box
 q_o = flow of air bleed
 q_i = flow of slurry into box
 q_s = flow of slurry out of box
 R = gas constant
 T = temperature
 V = fluid velocity
 ε_p = deviation of pressure from reference value
 ε_h = deviation of level from reference value
 ρ = air density
 ρ_s = fluid density

τ_{ip} = integral time of PID controller in pressure loop

τ_{ir} = integral time of PID controller in level loop

τ_{dp} = derivative time of PID controller in pressure loop

τ_{dh} = derivative time of PID controller in level loop

The flow of stock is dependent on the level, h , and the pressure, p . Flow is analyzed using Bernoulli's equation of fluid flow:

$$\frac{p}{\rho_s g} + \frac{V^2}{2g} + h = C \text{ (constant)} \quad \text{-----} \quad 1$$

On the surface of the box, $V=0$, hence:

$$\frac{p}{\rho_s g} + h = C \text{ (constant)} \quad \text{-----} \quad 2$$

At slit, $h=0$, hence:

$$\frac{p}{\rho_s g} + \frac{V^2}{2g} = C \text{ (constant)} \quad \text{-----} \quad 3$$

since, $V=q/a$:

$$\frac{p}{\rho_s g} + \frac{V^2}{2g} = \frac{p}{\rho_s g} + \frac{(q_s/a)^2}{2g} = C \text{ (constant)} \quad \text{-----} \quad 4$$

Flow of slurry out of box:

$$q_s = a_s \cos[2g\{(p - p_a)/\rho_s g + h\}]^{1/2} \quad \text{-----} \quad 5$$

From continuity equation:

$$A dh/dt = q_i - q_s \quad \text{-----} \quad 6$$

From equation 4:

$$dh/dt = q_i / A - a_s / A \cos[2g\{(p - p_a) / \rho_s g + h\}]^{1/2} \quad \text{---} \quad 7$$

Equation 7 shows that the rate of change of slurry level (a state) is dependent on the height, inflow, and pressure.

Also, applying Bernoulli's equation to the air flow out of the orifice, with change in pressure from p to p_a :

$$\frac{p}{\rho_a g} = \frac{p_a}{\rho_a g} + \frac{V^2}{2g} = \frac{p_a}{\rho_a g} + \frac{(q_o/a)^2}{2g}$$

Also:

$$q_o = a Co [2g\{(p - p_a) / \rho_a g + h\}]^{1/2} \quad \text{-----} \quad 8$$

But from the Ideal Gas law:

$$pV = MRT \quad \text{-----} \quad 9$$

$$V = A(H-h), \text{ and } M = 28.8 \text{ grams}$$

From continuity equation:

mass flowrate into box = mass flowrate out of box

$$q\rho - q_o\rho_a = dM/dt \quad \text{-----} \quad 10$$

But $\rho = 28.8p/RT$, $\rho_a = 28.8p_a/RT$

$$q28.8p/RT - q_o28.8p_a/RT = d/dt [28.8p(H-h)A/RT] \quad \text{-----} \quad 11$$

substituting for q_o in (8):

$$\frac{dp}{dt} = \frac{p}{H-h} \frac{dh}{dt} - \frac{p_a a}{A(H-h)} C_o \left[\{2RT(p - p_a) \times 28.8\} / RT \right]^{1/2} + \frac{qp}{A(H-h)} \quad \text{---12}$$

Error, ε , of the rate of slurry flow out of the slit from the desired value q_{set} :

$$\varepsilon = q_s - q_{set} = a_s [2g\{(p - p_a)/\rho g + h\}]^{1/2} - q_{set} \quad \text{---13}$$

From the above analysis, it can be inferred that this is a two input, two output control problem.

Controlled variables or Outputs:

h = liquid level

p = air pressure

Desired inputs:

p_r = reference value of air pressure inside box (lbf/in² abs)

h_r = reference value of slurry level inside box (in)

Manipulated variables:

q = air inflow

q_i = slurry inflow

6.0 CLOSED LOOP CONTROL

Various methods of control were investigated, namely, interacting control, pressure compensated slurry flow control and dual non-interacting control (see Gunn, and Sinha IEE 1983). The intention is to determine which method of control will give the best decoupling effect between the two outputs.

The dual non-interacting control was determined to be the best (see Gunn, IEE 1983), and the control system was designed using this method, in this thesis.

6.1 Dual non-interacting control

From equation 6 and 12 above, a change in q , for example, to control pressure, p , will also force a change in level, h ; whereas, a change in q_i , will cause a change in p . The intention is to decouple the control system, so that a change in level h , will have no effect on the value of p , and vice versa. From equation 12, if a change in level, dh/dt , is registered, the pressure, p , may be maintained at a constant value, by setting the flow rate, q , to be:

$$q = p_a a C_o / p \left[\{2RT(p - p_a) \times 28.8\} / RT \right]^{1/2} - A dh/dt \quad \text{-----} \quad 14$$

From equation 7, replacing dh/dt in the above equation:

$$q = p_a a C_o / p \left[\{2RT(p - p_a) \times 28.8\} / RT \right]^{1/2} - q_i - a_s / \cos[2g\{(p - p_a) / \rho_s g + h\}]^{1/2} \quad \text{-----} \quad 15$$

Thus, a deviation in either of the controlled variables, p , or h , is compensated by a direct decoupling control action, that is calculated as described above.

A change in pressure, p , will not cause a change in level, h , if according to equation 6, q_i is set by the following condition:

$$q_i = a_s \cos[2g\{(p - p_a) / \rho_s g + h\}]^{1/2} \quad \text{-----} \quad 16$$

Thus, a change in pressure is corrected by a change in flow rate, q , given by equation 17, and a compensating q_i , given by equation 18, so that h is maintained constant.

$$q = K_{cp} \left\{ \epsilon_p + 1 / \tau_{ip} \sum^t \epsilon_p dt + \tau_{dp} d\epsilon_p / dt \right\} \quad \text{-----} \quad 17$$

$$q_i = Kch \left\{ \varepsilon_h + \frac{1}{\tau_{ih}} \sum_0^t \varepsilon_h dt + \tau_{dh} d\varepsilon_h/dt \right\} \quad \text{-----} \quad 18$$

Figure 5.0 Block diagram of non- interacting control
(The flowbox is a two input, two output problem with a separate controller for each output, with an intention to decouple the states)

7.0 SIMULATION

	Controller gain (Kc)	Integral time (τ_{ip})	Derivative time (τ_{dp})
Pressure	80.0	0.125	0.10
Level	17.0	0.2	.001

Table 1. Controller parameter for non-interacting control

Parameter values:

$$a = 0.002 \text{ in}^2$$

$$a_s = 0.48 \text{ in}^2$$

$$A = 28 \text{ in}^2$$

$$C_o = 0.60$$

$$C_{os} = 0.60$$

$$g = 386.4 \text{ in/s}^2$$

$$R = 1.1513\text{E}03$$

$$T = 298 \text{ K}$$

$$\rho = 80 \text{ lb/ft}^3$$

$$h_o = 12 \text{ in}$$

$$p_o = 16.1 \text{ lbf/sq.in}$$

$$H = 24 \text{ in}$$

The parameters from Table 1 were used to design a PID controller, and the process dynamics was simulated using Simulink™ (Control systems simulation software). The process block diagram is shown in Figures 6.0. The process was simulated as a non-linear dynamic system to preserve its true system dynamics.

Figure 6.0 Subsystem block diagram of paperstock process dynamics
(The block diagram of the paperstock system dynamics is very non-linear and was simulated as such, to maintain true system characteristics)

8.0 DESIGN OF FUZZY CONTROLLER FOR PAPER-STOCK FLOW

The intention is to design a fuzzy controller to replace the PID controller. The controller is going to be a PI controller. The controller will decouple the control of

Pressure and Level into two separate controllers for each output. This is termed *non-interacting control*.

8.1 Fuzzification

8.1.1 Pressure:

The method of fuzzification will be the Min-Max method. Two control inputs will be *pressure error* and *rate of change pressure error*, which will be the process state variables. Error is defined as the difference between the desired output of the process (setpoint) and the actual output.

$$e(k) = y_{sp} - y(k)$$

where,

$$e(k) = \text{error}$$

$$y_{sp} = \text{setpoint}$$

$$y(k) = \text{actual output}$$

Rate of change of pressure error (error rate) is the difference between the previous and current error.

$$\Delta e = e(k-1) - e(k)$$

where,

$$\Delta e : \text{error rate}$$

$$e(k-1) : \text{previous error}$$

$$e(k) : \text{current error}$$

k: current time

The range of error is set at $[-16 \ 16]$, and error rate $[-16 \ 16]$. The controller output signal (process input), has a range of $[-16 \ 16]$. These values were chosen to reflect expected limits within which the pressure of air in the flowbox may vary (see Figure 8 and Appendix A).

The equation for a conventional PI-controller is:

$$u(k) = K_p * e(k) + K_i [I(k-1) + e(k) \Delta T]$$

where,

K_p = proportional gain constant

K_i = integral gain constant

$I(k)$ = present integral value($I(k) = I(k-1) + e(k) \Delta T$)

ΔT = sampling time

Taking derivatives of both sides,

$$\Delta u(k) = k_p * \Delta e(k) + K_i * e(k)$$

where,

$\Delta u(k)$: change in output control signal

This implies the output from the fuzzy controller must be integrated, taking the initial value (initial condition for Pressure, $P_o = 16 \text{ lb/in}^2 \text{ abs}$) for the output as a lower limit.

Figure 7.0 Input membership function (pressure rate) plot for forty nine rule controller

(The triangular input membership function has seven terms, with full term overlap. Each input membership function describes a measurable form the system input can take)

8.1.2 Level:

The two control inputs into the Level controller shall be *level error* and *rate of change level error*, which will be the process state variables. The definitions are similar to those of pressure, of which are stated above.

The range of error is set at $[-16 \ 16]$, and error rate $[-16 \ 16]$. The controller output signal (process input), has a range of $[-16 \ 16]$. These values were chosen to reflect expected limits within which the level of slurry is expected to vary (see Appendix B and C).

The initial condition for Level, $lo = 13$ in.

8.2 Rules Notation

Two sets of rules will be written each with a different number of rules to investigate how the number of rules will affect the response of the system.

The first set of rules will contain only five rules, as shown in Table 2

1. If (error is en) and (rate is rn) then (du is on) (1)
2. If (error is en) and (rate is rp) then (du is oz) (1)
3. If (error is ep) and (rate is rn) then (du is oz) (1)
4. If (error is ep) and (rate is rp) then (du is op) (1)
5. If (error is ez) then (du is oz) (1)

Table 2 Rules Table for "Five Rule Fuzzy Controller"

where,

du = change of output

en, rn, on = error negative, rate negative, output negative

ep, rp, op = error positive, rate positive, output positive

ez, rz, oz = error zero, rate negative, output negative

The "IF" part of each rule contains both inputs, while the "THEN" part of each rule contains the output of the controller. Each rule has a weighting of one.

The second set of rules will contain 49 rules. This takes into account small changes in linguistic terms, as shown in Table 3.0

1. If (error is PB) and (rate is PB) then (PRESSURE is PB) (1)
2. If (error is PB) and (rate is PM) then (PRESSURE is PB) (1)
3. If (error is PB) and (rate is PS) then (PRESSURE is PB) (1)
4. If (error is PB) and (rate is Z) then (PRESSURE is PB) (1)
5. If (error is PB) and (rate is NS) then (PRESSURE is PM) (1)
6. If (error is PB) and (rate is NM) then (PRESSURE is PS) (1)
7. If (error is PB) and (rate is NB) then (PRESSURE is Z) (1)
8. If (error is PM) and (rate is PB) then (PRESSURE is PB) (1)
9. If (error is PM) and (rate is PM) then (PRESSURE is PB) (1)
10. If (error is PM) and (rate is PS) then (PRESSURE is PB) (1)
11. If (error is PM) and (rate is Z) then (PRESSURE is PM) (1)
12. If (error is PM) and (rate is NS) then (PRESSURE is PS) (1)
13. If (error is PM) and (rate is NM) then (PRESSURE is Z) (1)
14. If (error is PM) and (rate is NB) then (PRESSURE is NS) (1)

15. If (error is PS) and (rate is PB) then (PRESSURE is PB) (1)
16. If (error is PS) and (rate is PM) then (PRESSURE is PB) (1)
17. If (error is PS) and (rate is PS) then (PRESSURE is PM) (1)
18. If (error is PS) and (rate is Z) then (PRESSURE is PS) (1)
19. If (error is PS) and (rate is NS) then (PRESSURE is Z) (1)
20. If (error is PS) and (rate is NM) then (PRESSURE is NS) (1)
21. If (error is PS) and (rate is NB) then (PRESSURE is NM) (1)
22. If (error is Z) and (rate is PB) then (PRESSURE is PB) (1)
23. If (error is Z) and (rate is PM) then (PRESSURE is PM) (1)
24. If (error is Z) and (rate is PS) then (PRESSURE is PS) (1)
25. If (error is Z) and (rate is Z) then (PRESSURE is Z) (1)
26. If (error is Z) and (rate is NS) then (PRESSURE is NS) (1)
27. If (error is Z) and (rate is NM) then (PRESSURE is NM) (1)
28. If (error is Z) and (rate is NB) then (PRESSURE is NB) (1)
29. If (error is NS) and (rate is PB) then (PRESSURE is PM) (1)
30. If (error is NS) and (rate is PM) then (PRESSURE is PS) (1)
31. If (error is NS) and (rate is PS) then (PRESSURE is Z) (1)
32. If (error is NS) and (rate is Z) then (PRESSURE is NS) (1)
33. If (error is NS) and (rate is NS) then (PRESSURE is NM) (1)
34. If (error is NS) and (rate is NM) then (PRESSURE is NB) (1)
35. If (error is NS) and (rate is NB) then (PRESSURE is NB) (1)

36. If (error is NM) and (rate is PB) then (PRESSURE is PS) (1)
37. If (error is NM) and (rate is PM) then (PRESSURE is Z) (1)
38. If (error is NM) and (rate is PS) then (PRESSURE is NS) (1)
39. If (error is NM) and (rate is Z) then (PRESSURE is NM) (1)
40. If (error is NM) and (rate is NS) then (PRESSURE is NB) (1)
41. If (error is NM) and (rate is NM) then (PRESSURE is NB) (1)
42. If (error is NM) and (rate is NB) then (PRESSURE is NB) (1)
43. If (error is NB) and (rate is PB) then (PRESSURE is Z) (1)
44. If (error is NB) and (rate is PM) then (PRESSURE is NS) (1)
45. If (error is NB) and (rate is PS) then (PRESSURE is NM) (1)
46. If (error is NB) and (rate is Z) then (PRESSURE is NB) (1)
47. If (error is NB) and (rate is NS) then (PRESSURE is NB) (1)
48. If (error is NB) and (rate is NM) then (PRESSURE is NB) (1)
49. If (error is NB) and (rate is NB) then (PRESSURE is NB) (1)

Table 3.0 Rules Table for "Forty Nine Rule" Fuzzy Controller

where,

Z = zero

NS = negative small

PS = positive small

PM = positive medium

NB = negative big

etc.

There will also be a set of 49 rules for the level output variable. These rules are by no means rigid rules that should be implemented. The number of rules, as well as the linguistic term could be changed, depending on the intended output of the controller.

The *defuzzification* method used was the *centroid* method. The shape of membership function chosen for both inputs and output was the triangular membership function with full term overlap (see Appendix A-C). The full overlap of the antecedence terms means that at any time two rules are active simultaneously. The full overlap of the consequent part results in nonlinear controller characteristics, which is desired.

9.0 SIMULATION

The block diagram of the dynamics of paperstock flow process was built in Matlab™ (Vers. 5.0), as shown in Figure 6.0. The control of the system was decoupled, each output, level and pressure having a separate controller. Control could be switched from PID to fuzzy logic by changing the value in the switch block (see Figure 8.0).

Separate simulations were done for the fuzzy controller with five rules, and that with forty nine rules. The system response was continuously tuned by changing the input and output scaling factors, varying the degree of membership function overlap, and the range of each membership function. These changes will be reflected on the control surfaces of each output. The three dimensional control surfaces of the five rule and forty nine rule

controller are shown in Figure 9,10 and 11 respectively. This is a "picture" of the characteristics of the controller. The membership functions could be dynamically tuned on this diagram also, to give a desired output, and this will be reflected by a change in the shape of the control surface.

The surface of the controller with forty nine rules is seen to be curved, demonstrating much desired nonlinear characteristics. Its shape is steepest where the values of both inputs come close to zero, demonstrating a high sensitivity about the set point. Alternatively, the shape of the control surface for the controller with five rules is level demonstrating a linear characteristic. It is less sensitive to small changes about the set point, i.e. when the values of error and rate are zero (flatness). It is more sensitive when the values of error and rate are large (steep shape).

Figure 8.0 Block Diagram of Paperstock Process Controlled by Fuzzy Logic/PID

(The system could be controlled by a PID or fuzzy logic controller through the switch; the subsystem block contains the process dynamics)

Figure 9.0 Three dimensional control surface for (forty nine rules) pressure controller

Figure 10 Three dimensional control surface for (forty nine rules) level controller

Figure 11 Three dimensional control surface for (five rules) level controller
(The flatness of the control surface shows a linear and less sensitive characteristic. The surface defines the relationship between the input and output membership functions)

9.1 Results

Generally, the response of pressure is faster than that for level due to the longer time constant of slurry flow. The response of the process controlled by fuzzy logic is much faster than that controlled by PID, i.e. faster rise time and settling time (see Figure 12 and 13). This is more evident in the response of the slurry level, where the rise time for the fuzzy logic controlled process is about 25% less than that of the PID controlled process. Therefore, the speed of response of the fuzzy logic controller is more significant in systems with large time constants. It pushes the pressure to attain steady state much faster.

Though the pressure has a tendency to oscillate about the set point, the fuzzy control signal pushes it to attain a steady state value just after two oscillations

It can also be inferred from Figures 12 and 13, that the fuzzy controller with forty nine rules is faster than that with five rules, and more sensitive to small changes about its set point. But the speed of response obtained by increasing the number of rules may be outweighed by the time consuming process of forming the rules and membership functions.

Figure 12 Pressure/level response to step input for forty nine rule controlled process
[Pressure (psi)/Level (in) vs. Time (secs.)]
(The response of the closed loop system is faster when controlled by a fuzzy logic controller than when controlled by a PID controller).

Figure 13 Response to step input for five rule controlled process

[Pressure (psi)/Level (in) vs. Time (secs.)]

(The response of the closed loop system is faster when controlled by a fuzzy logic controller than when controlled by a PID controller, but less sensitive than the case with forty nine rules)

10.0 C) **TEMPERATURE CONTROL OF ALUMINUM PLATE USING LABVIEW**

10.1 Introduction: The setup consists of two heating resistors, and a semiconductor temperature sensor mounted on an aluminum plate. The output of the sensor is connected to the analog input of a Data Acquisition card (DAQ). (See Figure 14). The analog output of the DAQ card is connected to a non inverting amplifier which supplies current to the heating resistors. The digital output from the DAQ card is used to switch a set of Darlington transistors that drive the fan.

10.1.1 Electronic Circuit: This consists of the temperature sensing circuit, and the heating and cooling circuit (Wang).

The temperature is being sensed by a National Semiconductor LM 35 in a TO-46 metal can package (Figure 15). The sensor range is -55 to 150°C , and it outputs a voltage, which is proportional to the temperature. The sensor will output 0 Volts, at 0°C . The temperature signal is then amplified by a non inverting amplifier, a National Semiconductor LM 748. It has a gain of 5, and is frequency compensated with a 33 pF capacitor. The temperature of the board is not expected to rise above 100°C , or 1 V sensor output. Hence, the maximum input into the data acquisition board will not exceed 5 V, which is the maximum voltage supported by most data acquisition boards (Wang).

The cooling circuit is consists of a Darlington transistor and a 12 V fan (see Figure 15).

Figure 14 Schematic of Control system layout

The fan draws a current of 160 mA, more than what the data acquisition board can source out. Therefore the Darlington transistor is used as an intermediate circuit to drive the fan. It sinks a current through the fan when it receives a 5V signal from the digital output port, A0, of the Data Acquisition card (DAQ). The transistor is a Motorola MC1411B, which is a chip containing an array of seven Darlington transistors.

The heating circuit consists of a voltage follower circuit, two transistors and two heating resistors (see Figure 15). The heating resistors can draw up to 2 Amps of current, more than what the DAQ card can support, hence an intermediate circuit is needed to amplify the current. The two heating resistors are 4.7 Ohm power resistors. They are wired in parallel to produce an equivalent resistance of 2.35 Ohms. The TIP31A

and 2N3055 transistors are arranged in totem pole configuration (common collector). Any current entering the base of the TIP31A transistor will cause a current to flow from the collector to the emitter of the 2N3055 transistor. This results in a current amplification of over 1000 (combined $h_{fe} > 1000$). The amount of current flowing through the base of the TIP31A transistor is controlled by an operational amplifier. The non inverting input of the operational amplifier is attached to the analog output port 0, of the DAQ card and the inverting input is attached to the emitter of the 2N3055 transistor. The op-amp will supply a voltage or current in order to keep the voltage or current at the emitter the same as the analog output voltage.

The emitter of the 2N3055 transistor is connected to the two resistors, therefore any current passing out of the transistors will go through the resistors.

The power supply is regulated to ± 6 VDC to power the field effect transistors and ± 12 VDC to power the fan and operational amplifiers .

11.0 DESIGN OF CONTROLLER

The function of the controller is set point regulation. The desired temperature of the plate is set at 80°C . The design intention is to design a controller that will drive the temperature to the set point as fast as possible, minimize overshoot, and attain a steady state temperature of within $\pm 0.5^{\circ}\text{C}$. A PID controller and fuzzy logic controller will be designed and incorporated into the Labview block diagram, and the performances of the two designed systems will be compared. (See Figure 16).

Wiring Diagrams.

Temperature Sensor Circuit

Heating and Cooling Circuit

Figure 15 Wiring diagrams for sensor, heating and cooling circuits
(The wiring diagram shows how each component is internally connected on the board, the linked to the DAQ card)

11.1 PID Controller : The PID control action will be performed by the PID Virtual Instrument, or VI. (See Appendix E). The VI will perform the calculations for the PID control algorithm, and determine the output based on the three gains, K_p (gain constant for proportional control action), K_i (gain constant for integral control action), and K_d (gain constant for derivative control action), the set point value, and the input value. The inputs to the VI are the control gains, the actual system output, the iteration number and the set point. The outputs from the VI are, the actual time (time between consecutive samples) and the control output.

The control algorithm is:

$$\text{Control output} = K_p * \text{Error}(n) + K_d * [\text{Error}(n) - \text{Error}(n-1)] / [t(n) - t(n-1)] + (K_i * \text{Sum Error}(n))$$

n = current iteration number

$$\text{Error}(n) = \text{Desired Output}(n) - \text{Actual Output}(n)$$

$$\text{Sum Error}(n) = \text{Sum Error}(n-1) + \{\text{Error}(n) * [t(n) - t(n-1)]\}$$

$t(n)$ = current time

The proportional, derivative, and integral control gains can be changed, to obtain an optimized value for each parameter. The front panel for the PID calculation is shown in Figure 17. The block diagram of the PID calculation VI consists of a Case statement nested inside of a While loop. Since this VI is non-recursive and is called more than once, a While loop is necessary to hold the values of the total error, the previous error and the time of the previous call. Three shift registers are used to hold these values from one call

of the VI to the next. The Case statement is needed to initialize the shift registers and outputs on the first call to this sub-VI. The VI returns the control Voltage and the actual time between the current and previous iteration as an output. The control voltage, which varies between 0V - 5V, is the control voltage for the heating resistors.

Figure 16 PID Control block diagram

(The control block diagram shows the flow of control signal in the feedback loop. Control voltage enters the D/A converter via the device 1 port on the right, and error signal enters the control loop via A/D converter through the device 1 port on the right)

Figure 17 PID front panel
 (The front panel displays the actual temperature, the control gains, etc. The values of the control gains could be changed from here)

11.2 Fuzzy Logic Controller: Two fuzzy logic controllers will be designed, with nine, and forty nine rules, respectively. The intention is to investigate the effect of the number of rules on controller characteristics. The controller with nine rules, "PET1.fc", has two linguistic input variables, temperature error, and temperature rate, with ranges between -15 and +15, and -5 and +5, respectively (see Figure18 and Appendix F). The linguistic output variable is voltage, ranging between -5.0V and +5.0V. (See Appendix G). If an output with a negative voltage is generated, a conditional statement has been set in the while loop to generate a default positive voltage of 0.80V. This is because realistically, there is no negative DC voltage. A negative voltage simply means that it is desired for the voltage to be reduced. These values were chosen from experience derived from running

the plant. This is because the physical meaning of a negative voltage is for the controller to output a small voltage to avoid overshoot.

Each linguistic variable has three linguistic terms: negative, zero, and positive. The ranges of each linguistic term were chosen from experience to actually represent how the linguistic variables of the process to be controlled would actually vary. This means running the process, collecting data, and learning through experience. There are nine rules, each with a weight factor, or Degree of Support (DoS) of one, as shown in Table 4. The DoS has the function of enhancing or reducing the influence of a rule on the controller characteristics. The DoS ranges between 0 and 1. The inference and defuzzification methods chosen are the Max-Min, and Center of Gravity methods, respectively.

The controller with forty nine rules, "PET5.fc", also has two linguistic input variables, temperature error, and temperature rate, with ranges between -15 and +15, and -5 and +5, respectively (see Figure 19 and Appendix H). The linguistic output variable is voltage, ranging between -5V and +5V. (See Appendix I). Most of the other aspects are similar to that of PET1.fc, apart from the fact that each linguistic variable has seven linguistic terms, each term with a different range, and there are forty nine rules, as shown in Table 5. This takes into account small changes about the set point, hence making it more sensitive. The controller was originally designed with full term overlap, and the degree of overlap was later altered to tune the controller output. It was not practical to use scaling factors to tune the controller parameters, since this method of tuning causes oscillation of the system response about the set point.

Fuzzy Logic Toolkit :

Linguistic Antecedence Variable

Date: 8/22/98

Controller Name : PET1.fc

Time: 10:56 AM

setpoints of ling. variable : TEMP ERROR

	left bottom	left top	right top	right bottom
NEG	-15.000	-15.000	-15.000	0.000
ZERO	-12.000	0.000	0.000	12.000
POS	0.000	15.000	15.000	15.000

Figure 18 Input membership function "temp error" for PET1.fc

Before running the fuzzy controller, the input/output characteristics were tested within Labview. Optimization and necessary modifications were carried out.

Fuzzy Logic Toolkit : Rules					Date: 10:57 AM
Controller Name : PET1.fc					Time: 8/22/98
Rule No.	IF TEMP ERROR	AND TEMP RATE	THEN VOLTAGE	DoS	
1	NEG	NEG	NEG	1.00	
2	NEG	ZERO	NEG	1.00	
3	NEG	POS	ZERO	1.00	
4	ZERO	NEG	NEG	1.00	
5	ZERO	ZERO	ZERO	1.00	
6	ZERO	POS	POS	1.00	
7	POS	NEG	ZERO	1.00	
8	POS	ZERO	POS	1.00	
9	POS	POS	POS	1.00	

Table 4 Rules table for "PET1.fc" fuzzy controller

Fuzzy Logic Toolkit : Rules

Date 12:32

Controller Name : PET5.fc

Time 3/4/98

Rule No.	IF TEMP ERROR	AND TEMP RATE	THEN VOLTAGE	DoS		
1	NB	NB	Z	1.00		
2	NB	NM	Z	1.00		
3	NB	NS	Z	1.00		
4	NB	Z	Z	1.00		
5	NB	PS	Z	1.00		
6	NB	PM	Z	1.00		
7	NB	PB	Z	1.00		
8	NM	NB	Z	1.00		
9	NM	NM	Z	1.00		
10	NM	NS	Z	1.00		
11	NM	Z	Z	1.00		
12	NM	PS	Z	1.00		
13	NM	PM	Z	1.00		
14	NM	PB	NS	1.00		
15	NS	NB	NM	1.00		
16	NS	NM	NS	1.00		
17	NS	NS	NS	1.00		
18	NS	Z	NS	1.00		
19	NS	PS	Z	1.00		
20	NS	PM	PS	1.00		
21	NS	PB	PM	1.00		
22	Z	NB	NB	1.00		
23	Z	NM	NM	1.00		
24	Z	NS	NS	1.00		
25	Z	Z	Z	1.00		
26	Z	PS	PM	1.00		
27	Z	PM	PM	1.00		
28	Z	PB	PB	1.00		
29	PS	NB	NS	1.00		
30	PS	NM	NS	1.00		
31	PS	NS	Z	1.00		
32	PS	Z	PS	1.00		
33	PS	PS	PS	1.00		
34	PS	PM	PS	1.00		
35	PS	PB	PS	1.00		
36	PM	NB	NS	1.00		
37	PM	NM	Z	1.00		
38	PM	NS	PS	1.00		
39	PM	Z	PM	1.00		
40	PM	PS	PB	1.00		

Table 5 Rules table for "PET5.fc" fuzzy controller (continued)

Fuzzy Logic Toolkit: Rules					
Controller Name: PET5.fc				Date: 12:32 PM	
				Time: 8/4/98	
Rule No.	IF TEMP ERROR	AND TEMP RATE	THEN VOLTAGE	DeS	
41	PM	PM	PB	1.00	
42	PM	PB	PB	1.00	
43	PB	NB	Z	1.00	
44	PB	NM	PS	1.00	
45	PB	NS	PM	1.00	
46	PB	Z	PB	1.00	
47	PB	PS	PB	1.00	
48	PB	PM	PB	1.00	
49	PB	PB	PB	1.00	

Table 5 Rules table for "PET5.fc" fuzzy controller (continued)

Figure 19 Input membership function "temp rate" for PET5.fc
 (The input membership function has a range which is chosen from actual plant data and the range of its terms and their degree of overlap is largely defined by their estimated operating ranges)

Before the fuzzy controller is loaded into the application block diagram, it needs to be programmed. The VI has many inputs but the ones used were the *controller data* from the Load Fuzzy Controller VI, *name 1* and *input 1*, for error, *name 2* and *input 2*, for error rate. (See Figure 20).

Figure 20 Fuzzy logic control block diagram
(The data for membership function and processed rules are loaded on to the diagram to effect control)

The output signal is the *analog output*. The Fuzzy Controller VI is programmed by the Load Fuzzy control VI. This loads the required data into the Fuzzy Controller VI, i.e. the parameters already defined in the rules table, membership functions, and linguistic variables. The outputs of the Load Fuzzy Controller VI that are connected to the Fuzzy Controller VI are the *controller out*, *input name 1*, and *input name 2*.

12.0 RESULTS

The PID controlled response reached steady state with a steady state error of 0.675%. After reaching steady, the fan was turned on to test the controller's ability to reject a step disturbance input. This response to disturbance input caused an oscillation about the set point, with a maximum error of $\pm 1\%$, as shown in Figure 21.

The fuzzy controller, PET1.fc, has a much faster response than the PID controller (see Figure 22). The error stayed at 0.375% overshoot, and the control signal is very uniform. When the fan was turned on, the control voltage increased, but oscillated. The system response oscillated about the set point with a maximum error of $\pm 1.075\%$.

The controller with forty nine rules, PET5.fc, is faster in response than the PID and PET1.fc controller. (See Figure 23). It attained steady state quickly with a maximum error of $\pm 0.375\%$. The control voltage was uniform at steady state. When subjected to a large disturbance input, the control voltage increased, but the control power (analog voltage to heating resistors) was not enough for large disturbance rejection. The system response then reached steady state with a constant undershoot of 1.075%, revealing a sensitive, but stable characteristic (see Figure 24). The

magnitude of the disturbance was then reduced by half by blocking off half of the intake, thus halving the air flow rate (see Figure 25). The response shows a total rejection of the disturbance. The average control voltage increased, but the error stayed at $\pm 0.375\%$.

A more revealing comparison of the fuzzy logic versus the PID controlled process is shown in Figure 26. The top plot shows a faster response and less overshoot for the fuzzy logic controlled process, and a more stable response after rejecting an external disturbance. The middle plot also reveals a faster error reduction for the fuzzy logic controlled process, revealing a better integral control characteristic. The bottom plot shows that the fuzzy logic controller is quicker in generating the control signal to effect a faster control action. The amplitude of the control signal is more uniform, with less transients, compared to the PID control signal. A transient is evident only when there was an unusual increase in error due to a disturbance input. This action is very much desired to increase the voltage to the heating resistors so as to drive the plate temperature back to the desired level. The PID controller signal, oscillates more and is less sensitive to a disturbance input, but performs satisfactorily.

Figure 21 Transient response of PID controlled process to large disturbance

Figure 22 Transient response of fuzzy logic controller (PET1.fc) to large disturbance

FUZZY
CONTROLLER

Figure 23 Transient response of fuzzy logic controller (PET5.fc)

Figure 24 Transient response of fuzzy logic controller (PET5fc) to large disturbance

Figure 25 Transient response of fuzzy logic controller (PET5fc) to small disturbance

Figure 26 (Top) Transient Response of PID versus Fuzzy Logic
 (Middle) Temperature error of PID versus Fuzzy Logic
 (Bottom) Control Input of PID versus Fuzzy Logic

13.0 DISCUSSION

This section is based on the personal observations, findings and inferences of the author. It can be deduced from the results that the design of a fuzzy logic controller for simulation purposes does not require any extensive mathematical calculations on the theory of fuzzy logic sets. This is because the use of mathematical theory or implications does not come into play when designing the controller and it does not help in achieving any control objective, since this is an actual practical exercise. What is relevant is an understanding of the dynamics and behavior of the plant in question. The knowledge of fuzzy logic theory is only useful as a background knowledge in appreciating the methods the operating software uses in generating the required data, or desired response. Obtaining a stable, simulated response may require the adherence to certain design procedures, since simulation is largely a theoretical exercise. These will be the methods of rules formation, the type of membership functions and the degree of overlap of the membership functions. More rules may be required for a more sensitive controller, but may also cause a reduced ability to reject disturbances. This is because increasing the number of rules fine tunes the controller output and makes it more responsive to small changes in control input. There is no limit to the number of rules to be formed, because this is purely an objective procedure. It is a decision the designer has to make depending on his control application and intentions. Generally, fewer rules are better for a controller where disturbance rejection is the prime objective. Many rules reduce the ability to reject large disturbances, but are relevant where set point tracking (less steady state error), increased speed of response and a very nonlinear control output are the ultimate intentions. Increasing the

number of rules could also be used as a tuning method. The weighting of each rule would normally be one, but different applications may require different weightings. Full membership overlap is necessary for a nonlinear control output. This makes the controller more fuzzy. Tuning is achieved by continuously changing the values of the input and output scaling factors, until a desired response is achieved. This is also reflected in the three dimensional control surface of each controller (see Figures 27 and 28).

Figure 27 Control surface of PET5.fc

Figure 28 Control surface of PET1.fc

The flat surfaces demonstrate less sensitivity to changes in input and output variables in those areas, while the steep slopes demonstrate high sensitivity to changes in input/output variables in these areas.

Unfortunately, controller design for real time implementation does not follow any rigid procedure. This is because it requires several trials to obtain a desired objective. Computer simulation results should be used as a guide to the actual solution. These results cannot be directly implemented on a hard ware to be controlled, because the model is only an approximation of the actual plant. Firstly, one needs to make an attempt to have a good understanding of the plant to be controlled, mostly the dynamics of the parameters

to be controlled, e.g. the magnitude of variance between the desired output and the actual output under different operating conditions. This means actually collecting physical data on the possible values the control inputs/outputs could take under different operating conditions. The rules are then formed through familiarity with how the plant behaves. Several trials may be necessary to achieve a desired set of rules. The idea is to form rules that are effectively "telling" the controller how to bring the plant to the desired state, as quickly as possible. For example, the IF part of the rules should reflect the different values that the control inputs have been observed to take, and the THEN part of the rules should reflect the actions required to bring these control inputs to their most desired state at that time. The membership functions are generally the variables to be controlled, and the linguistic terms should be a description of how the control variables vary. Their ranges should be obtained from raw data obtained from the plant. The degree of overlap of membership functions should be determined by the response desired. Less degree of overlap is desired when the operating conditions are far from the set point (large positive or negative errors), to drive the plant closer to the setpoint as quickly as possible, i.e. fast rise time. This implies a more certain (less fuzzy) objective. Full-term overlap is required around the set point, to reduce transient oscillations about the set point and to achieve a much reduced steady state error. Tuning of the response is better done by changing the rules until the actual response is close to the desired response; fine-tuning is achieved by changing the degree of overlap of the membership functions, since scaling factors are not applicable, because they cause oscillation of the output. Several runs are made after each tuning action and areas of undesired response are noted. Data on the input/output values is

then collected about the unstable regions. At this point the range of each membership function and linguistic term is then readjusted, together with the rules, if necessary. Usually, it is better to set all scaling factors to unity. This method is seen to be less tolerant of external disturbances. Design for disturbance rejection requires measuring the magnitude of the errors caused by the anticipated disturbance and reducing the degree of overlap of output membership functions where these values of error may occur. This destabilizes steady state response, hence, there may be trade-offs when designing for fast settling time, or for large disturbance rejection. Normally, fuzzy logic controllers, by their very fuzzy nature, are tolerant to disturbances.

14.0 CONCLUSIONS

Fuzzy logic control is mostly based on human intuition, and so its success and method of control varies with application and control intentions. Extensive trials should be carried out on practical implementation of fuzzy logic based control in different control applications. Most notably, uncoupled control and other higher levels of control which may require updating control output in response to changes in plant dynamics, e.g. adaptive control. It may also be necessary to investigate methods of integrating fuzzy logic and conventional control methods, so that a designated controller could be used to scale or tune the response of the other, e.g. supervisory control (Chiu, 1998) and neuro-fuzzy control systems (Reznik, 1997). Also, there is a need to implement fuzzy logic control as an embedded control system. This may require investigating methods implementing fuzzy logic algorithms in C++, or assembly language, and executing it within a micro-controller

or a standalone microprocessor. These may help develop fuzzy logic control as a realistic and practical method of control, which could some day be used in applications which may require a higher degree of reliability (Wang, 1998).

Appendix A

Input membership function (pressure error) plot for forty nine rule controller (Top)

Output membership function (pressure) plot for forty nine rule controller (Bottom)

Appendix B

Input membership function (level rate) plot for forty nine rule controller (Top)

Input membership function (level error) plot for forty nine rule controller (Bottom)

Appendix C

Output membership function (level) plot for forty nine rule controller (Top)

Input membership function (rate) plot for five rule controller (Bottom)

Appendix D

Input membership function (error) plot for five rule controller (Top)

Output membership function (du) plot for five rule controller (Bottom)

Appendix E

Eval
pid.vi
D:\labview\Project\pid.vi
Last modified on 8/29/98 at 11:18 PM
Printed on 10/29/98 at 9:55 PM

Page 1

PID Virtual Instrument, or VI

Appendix F

Input membership function "temp rate" for PET1.fc

Appendix G

Output membership function "voltage" for PET1.fc

Appendix H

Input membership function "temp error" for PET5.fc

Appendix I

Output membership function "voltage" for PET5.fc

Bibliography

- Avalone A.Eugene and Baumeister Eugene, "Mark's Standard Handbook For Mechanical Engineers," 9th edition, McGraw-Hill Inc.
- Chiu Stephen, "Using Fuzzy Logic in Control Applications: Beyond Fuzzy PID Control," IEEE Control Systems, Vol 18, No. 5, pp 100-104. October 1998
- Gunn, D.J. and. Sinha, A.K., "Nonlinear Control of Paper stock Flow," IEE Proc Vol 130, Pt. D. No 3, May 1983.
- Gulley, N, Jang Roger, J.S., "Fuzzy Logic Toolbox for Use with Matlab." (The Mathworks Inc.)
- National Instruments Corporation, "Fuzzy Logic for G Toolkit Reference Manual," (March 1997 edition.)
- National Instruments Corporation, "Labview Data Acquisition VI Reference Manual for Windows."
- Reznik Leonid, "Fuzzy Controllers, First edition," 1997. Newnes, Oxford.
- Wang, Ji C., "Analog Control Systems Analysis and Design Using Matlab," Mechanical Engineering Department, San Jose State University, 1998.
- Wang, Ji C., "Implementation of Digital Data Acquisition and Control Systems (Hardware and Software)," Mechanical Engineering Department, San Jose State University, 1998.

IMAGE EVALUATION TEST TARGET (QA-3)

APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved