THE UNIVERSITY OF ALBERTA

FUZZY PROCESS IDENTIFICATION AND CONTROL

BY

CHEE HENG WONG ©

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

PROCESS CONTROL

DEPARTMENT OF CHEMICAL ENGINEERING

EDMONTON, ALBERTA

FALL, 1996

National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-18334-3

Canada

UNIVERSITY OF ALBERTA

LIBRARY RELEASE FORM

NAME OF AUTHOR	:	Chee Heng Wong
TITLE OF THESIS	:	Fuzzy Process Identification and Control
DEGREE	:	Master of Science
YEAR THIS DEGREE GRANTED	:	1996

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

ADDRESS :

107 Lebuh Sungai Senam
Taman Ipoh
31400 Ipoh Perak
Malaysia

Date : July 15, 1996

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled FUZZY PROCESS IDENTIFICATION AND CONTROL submitted by CHEE HENG WONG in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in PROCESS CONTROL.

....
Dr. S.L. Shah
Supervisor

....
Dr. D.G. Fisher

....
Dr. P.G. van Beek

Date : July 10, 1996

Dedicated to my parents

Abstract

Two new fuzzy relational identification algorithms were formulated the Neuron-Based Predictive Identification (NBPI) algorithm and the Fuzzy Relational Predictive Identification (FRPI) algorithm. These algorithms minimize the fuzzy prediction error over a user-specified prediction horizon and maintain the predictive capability of fuzzy relational models. These proposed schemes were found to give better results than Shaw's fuzzy relational identification technique. The FRPI algorithm was also found to be practical for on-line applications.

The servo and regulatory performance of two fuzzy relational controllers namely the Self-Learning Predictive Fuzzy Controller (SLPFC) and the Fuzzy Relational Long Range Predictive Controller (FRLRPC) were evaluated experimentally on a highly non-linear, interacting level process. Based on the integral sum of errors, the SLPFC gave a slighly better performance. The impact of on-line fuzzy relational identification algorithms on controller performance was also verified. It was found that the FRPI algorithm results in an improvement in controller performance compared to Shaw's fuzzy relational identification algorithm.

Acknowledgement

I would like to express my gratitude to Dr. Sirish Shah for his guidance and supervision. My research would not have been successful without his help and motivation. I also like to thank Dr. D.G. Fisher for his critical feedback and suggestions.

All the members in the Process Control group have help d make graduate studies a very positive experience. I like to thank Lanre Badmus, Mary Bourke, Pranob Banerje [illegible] Huang and Laksh Lakshminarayanan for their assistance; Albert Chiu, Ricky Leu [illegible] Yiu and Rohit Patwardhan have all helped make the days interesting.

[illegible] cial support from the Department of Chemical Engineering and the Natural Sciences a d Engineering Research Council of Canada is appreciated.

All this would not have been possible without my parents. Their love and sacrifice is deeply acknowledged. Finally, I would like to thank the Lord Jesus Christ for all that He has done.

Contents

List of Tables

List of Figures

LIST OF SYMBOLS

Chapter 2

e	error
K_c	proportional gain
$N_{(\cdot)}$	number of membership functions in the variable $(\cdot)$

Greek Symbols

δ	differencing operator, $1 - q^{-1}$
η	performance index
μ	degree of membership

Superscripts

$\tilde{}$	defuzzified value
$\bar{}$	complement of

Other Symbols

$\cap$	intersection operator
$\cup$	union operator
$\times$	Cartesian product

Chapter 3

R	relational matrix
$N_{(\cdot)}$	number of membership functions in the variable $(\cdot)$

Greek Symbols

λ	possibility measure

Other Symbols

$\circ$	max triangular norm operator
$*$	algebraic product

Chapter 4

d	time-delay
e	error
$N_{(\cdot)}$	number of membership functions in the variable $(\cdot)$
p	possibility vector
R	relational matrix
t	triangular norm operator
$U(k)$	input possibility vector
$Y(k)$	output possibility vector

Greek Symbols

β_s	relative contribution
μ	degree of membership

Superscripts

$\hat{}$	estimated value

Other Symbols

$\bigcup$	union operator
$\circ$	max triangular norm operator
$\equiv$	equality index
$\times$	Cartesian product
$\wedge$	minimum
$\vee$	maximum

Chapter 5

d	time-delay
e	error
N_1	minimum output prediction horizon
N_2	maximum output prediction horizon
N_p	number of predictions in the prediction horizon $(N_2 - N_1 + 1)$
R	relational matrix
$u(k)$	process input
$y(k)$	process output

Greek Symbols

α	momentum term
η	learning rate
λ	control move suppression factor
ϕ	fuzzification operator

Superscripts

$\hat{}$	estimated value

Subscripts

FPC	fuzzy predictive controller
$FPID$	fuzzy predictive identification
$FRPI$	fuzzy relational predictive identification
ID	identification
$LRPI$	long range predictive identification
$NBPI$	neuron-based predictive identification
sp	set-point

Other Symbols

$\equiv$	equality index
$\rightarrow$	implication
$\vee$	maximum
$\wedge$	minimum

Chapter 6

d	time-delay
e	error
f_u	filter constant of process input filter
f_e	filter constant of feedback filter
N_1	minimum output prediction horizon
N_2	maximum output prediction horizon
N_p	number of predictions in the prediction horizon $(N_2 - N_1 + 1)$
R	relational matrix
$u(k)$	process input
$y(k)$	process output
$U(k)$	fuzzified process input
$Y(k)$	fuzzified process output

Greek Symbols

α	tuning parameter which determines aggresiveness of self-learning predictive fuzzy controller
δ	differencing operator
λ	control move suppression factor

Subscripts

$FRLRPC$	fuzzy relational long range predictive controller
$FRPI$	fuzzy relational predictive identification
$LRPC$	long range predictive control
$SLPFC$	self-learning predictive fuzzy controller
sp	set-point

Chapter 7

d	time-delay

Subscripts

$FRLRPC$	fuzzy relational long range predictive controller
$FRPI$	fuzzy relational predictive identification
$SLPFC$	self-learning predictive fuzzy controller

Chapter 1

Introduction

So far as the laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality.

Albert Einstein

Introduced in 1965 by Zadeh, fuzzy logic is an attempt to capture the essence of the human decision-making and thought process. The human mind works in a fuzzy manner, it uses linguistic concepts rather than precise numerical values in the decision-making process. This fuzziness allows humans to perform complicated tasks which computers and machines do not do well. From a process control standpoint, the use of fuzzy logic has usually met with a high degree of success (see Mamdani (1974), Ostergaad (1977), Tong *et al.* (1980), Ono *et al.* (1989), Czogala & Rawlik (1989), Roffel & Chan (1991) and Oishi *et al.* (1991)). Some of the features found in successful applications of fuzzy logic controllers are

- Fuzzy, inprecise or uncertain process measurements

- Complex and poorly understood processes

- Process as previously controlled by a human operator

Therefore, fuzzy control can be a good candidate for certain control problems but can not be expected to replace conventional control strategies.

Previous applications of fuzzy logic in process control have concentrated on the use of a rule-base which consists of a set of "*IF-THEN*" rules. The linguistic nature of the rule-base makes rule-based fuzzy systems simple to construct and implement. However, the fine tuning of fuzzy logic controllers (FLC) may be time-consuming as it is an iterative procedure. Relational-based fuzzy logic has been used to introduce a more systematic controller design procedure. Relational-based fuzzy logic focuses on the use of fuzzy relational equations. A systematic process identification procedure exists for relational-based fuzzy systems (see Pedrycz (1983, 1984) and Shaw & Krüger (1992)). Recently, fuzzy relational models have been used in the development of model-based control schemes (see Graham & Newell (1988, 1989), Postlethwaite (1994), Bourke (1995), Bourke & Fisher (1996) and Valente de Oliveira & Lemos (1995)).

1.1 Scope and Objectives of Study

One of the most comprehensive studies of fuzzy relational identification and control was conducted by Bourke (1995). The author surveyed fuzzy relational identification algorithms and presented the development of a self-learning predictive controller. One of the shortcomings of fuzzy relational models raised by Bourke was their poor predictive nature. This drawback limits the effectiveness of fuzzy relational controllers when dealing with systems with lengthy time-delays. In chemical processes where time-delays are extremely common, this drawback of fuzzy relational models is a serious problem and must be addressed. Therefore, in this study, efforts will be [illegible] to the improvement of the predictive capability of fuzzy relational models.

Advances in fuzzy relational model-based control strategies has been made by Graham & Newell (1988), Postlethwaite (1994), Bourke (1995) and Valente de Oliveira (1995). Postlethwaite (1994) has shown that the performance of his controller is superior to that by Graham *et al.* (1988). The controller by Postlethwaite (1994) and Valente de Oliveira & Lemos (1995) are similar to conventional Long Range Predictive Controller (LRPC) in that they minimize the sum of prediction errors over a prediction horizon. The difference between Postlethwaite's controller to that of Valente de Oliveira & Lemos (1995) lies in the fact that the prediction horizon for Postlethwaite's controller is restricted to two. Prior to this work, no efforts has been made to compare the performance of Postlethwaite's controller to that by Bourke & Fisher (1996). In addition, limited experimental results are available for these two controllers. It is an objective of this study to compare the servo and regulatory performance of these controllers on a lab-scale process.

Hence, the objectives of this thesis are summarized as follows :

- To improve the predictive capability of fuzzy relational models

- To gain a better understanding of fuzzy relational controllers
- To experimentally evaluate the performance of fuzzy relational controllers

1.2 Thesis Organization

The structure of this thesis is outlined below but a more detailed description can be found at the beginning of each chapter.

The thesis is written assuming that the reader has no previous knowledge of fuzzy logic. Therefore, Chapter 2 begins by introducing the fundamentals of fuzzy logic. The intuitiveness of rule-based systems is a good starting point for the eventual understanding of fuzzy relational systems. The development of a fuzzy tuner for PID controllers is also presented which illustrates the key ideas in rule-based reasoning.

Chapter 3 introduces fuzzy relational equations or models. Interpretations of fuzzy relational equations are first given which are then followed by some issues pertaining to their role in process identification.

An in-depth description of the procedure involved in fuzzy relational identification is presented in Chapter 4. A review of key ideas in existing fuzzy relational identification algortihms is also given. The advantages and disadvantages of current schemes are also identified.

The issue of the predictive capability of fuzzy relational models is addressed in Chapter 5. Specifically, two new fuzzy relational algorithms are formulated, namely the Neuron-Based Predictive Identification (NBPI) algorithm and the Fuzzy Relational Predictive Identification (FRPI) algorithm. The aim of these new identification techniques is to provide fuzzy relational models with a better predictive capability. A comparison between the proposed schemes with previous fuzzy relational identification algorithms is also performed.

Chapter 6 begins by reviewing the recent developments in fuzzy relational controllers. The effectiveness of two fuzzy relational controllers namely those by Postlethwaite (1994) and Bourke & Fisher (1996) are evaluated experimentally on a laboratory scale process. The impact of on-line fuzzy relational identification algorithms on controller performance is also investigated.

Chapter 7 summarizes the main results and contributions of this thesis. It also lists some of the areas in fuzzy relational logic which deserve some further research and study.

Bibliography

[1] M. Bourke. *"Self-Learning Predictive Control Using Relational-Based Fuzzy Logic"*. PhD thesis, University of A[illegible]erta, 1995.

[2] M.M. Bourke and D.G. Fisher. A Predictive Fuzzy Relational Controller. To be presented at IEEE International Conference of Fuzzy Systems, New Orleans, 1996.

[3] E. Czogala and T. Rawlik. Modeling of Fuzzy Controller with Application to the Control of Biological Processes. *Fuzzy Sets and Systems*, 31:13 22, 1989.

[4] J. Valente de Oliveira and J.M. Lemos. "Long-range predictive adaptive fuzzy relational control". *Fuzzy Sets and Systems*, 70:337 357, 1995.

[5] B.P. Graham and R.B. Newell. "Fuzzy Identification and Control Of A Liquid Level Rig". *Fuzzy Sets and Systems*, 26:255 273, 1988.

[6] B.P. Graham and R.B. Newell. Fuzzy Adaptive Control of a First Order Process. *Fuzzy Sets and Systems*, 31:47 65, 1989.

[7] I.S. Shaw and J.J. Krüger. "New Fuzzy Learning Model with Recursive Estimation for Dynamic System". *Fuzzy Sets and Systems*, 48:217 229, 1992.

[8] C.C. Lee. Fuzzy logic in control systems : fuzzy logic controller, Parts I and II. *IEEE Trans. Systems Man and Cybernetics*, 20(2):404–435, 1990.

[9] E.H. Mamdani. Applications of fuzzy algortihms for simple dynamic plant. *Proc. IEE*, 121(12):1585–1588, 1974.

[10] K. Oishi, M. Tominaga, A. Kawato, Y. Abe, S. Imayasu, and A. Namba. Application of fuzzy control theory to the Sake brewing process. *Journal of Fermentation and Bioengineering*, 72(2):115–121, 1991.

[11] H. Ono, T. Ohnishi, and Y. Terada. Combustion Control of Refuse Incineration Plant by Fuzzy Logic. *Fuzzy Sets and Systems*, 32:193–206, 1989.

[12] J.J. Ostergaad. "Fuzzy logic control of a heat exchange process". In M.M. Gupta, G.N. Saridis, and B.R. Gaines, editors, *Fuzzy Automata and Decision Processes*, pages 285–320. North Holland, Amsterdam, 1977.

[13] W. Pedrycz. "Numerical and Applicational Aspects of Fuzzy Relational Equations". *Fuzzy Sets and Systems*, 11:1–18, 1983.

[14] W. Pedrycz. "An Identification Algorithm in Fuzzy Relational Systems". *Fuzzy Sets and Systems*, 13:153–167, 1984.

[15] B. Postlethwaite. "A Model-Based Fuzzy Controller". *Chemical Engg. Research and Design, Part A : Trans IChemE*, 72:38–46, 1994.

[16] B. Roffel and P. Chan. Fuzzy Control of a Polymerization Reactor. *Hydrocarbon Processing*, pages 47–49, June 1991.

[17] J.S. Roger-Jang and N. Gulley. *Fuzzy Logic Toolbox User's Guide*. The Math Works Inc., 1995.

[18] R.M. Tong, M.B. Beck, and A. Latten. Fuzzy control of the activated sludge wastewater treatment process. *Automatica*, 16(6):695–701, 1980.

Chapter 2

Rule-Based Fuzzy Systems

2.1 Introduction

Fuzzy logic was introduced in 1965 by Zadeh. His purpose was to provide a tool to aid in the modeling of complex phenomena especially, but not restricted to those involving human agents. Conventional mathematical techniques which involve exact quantitative values do not fulfil this role. Zadeh's (1973) principle of incompatibility states that "as the complexity of a system increases, our ability to make precise and yet significant statements about its behaviour diminishes until a threshold is reached beyond which precision and significance, or relevance, become almost mutually exclusive characteristics". This principle best sums up the role fuzzy logic is to address.

In this chapter, the essentials of fuzzy set theory will be presented. This will be followed by a description of rule-based fuzzy models and fuzzy logic controllers. The reasoning mechanism involved in fuzzy logic will also be presented. Finally, the decision-making capabilities of fuzzy logic will be applied to develop a tuner for conventional PID controllers.

2.2 Fundamentals of Fuzzy Set Theory

Ordinary or classical set theory is a particular case of fuzzy set theory. Therefore, starting with ordinary sets, it is possible to generalize and obtain a helpful understanding of fuzzy set theory.

In ordinary set theory, a subset must be defined with respect to some universe of discourse (domain). If A is a subset in the universe $\mathbf{U}$, $A \subset \mathbf{U}$, an element x in $\mathbf{U}$ which is a member of A is written as $x \in A$. The idea of an element, x, being a member of a subset A can also be indicated by its characteristic function, $\mu_A(x)$. The concept of membership is then summed up as

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \tag{2.1}$$

In other words, ordinary set theory draws its degree of membership or membership grade from two values: 0 or 1.

Like ordinary subsets, fuzzy subsets are also defined on some universe of discourse. However, the fuzzy subset extends the idea of an ordinary subset by allowing the characteristic function to lie in the interval [0,1] instead of in the finite set $\{0,1\}$. Therefore, there are now infinite degrees of belonging. Formally,

$$\mu_A(x) = [0, 1]$$

In the framework of fuzzy set theory, the characteristic function is called the membership function. The membership function assigns the degree for which x is a member of A.

Fuzzy sets are usually used to convey the meaning in the natural language of some variable. For example, one can define a fuzzy set to represent *hot* which is one linguistic value for the variable temperature. To illustrate, consider a variable such as ambient temperature. One possible definition of the concept of *hot* is shown in Figure (2.1). A temperature of 40 is considered to be trully *hot* (having a membership grade of 1) and as the temperature decreases its degree of belonging to *hot* is also reduced. At a temperature of less than 5, the fuzzy set *hot* does not describe the condition of the ambient temperature. Clearly more fuzzy sets can be defined to fully describe the ambient temperature. Figure (2.2) uses three fuzzy sets name *cold*, *tepid* and *hot* to convey the status of the ambient temperature. In this illustration, the temperature is known as a **linguistic variable** since its status is expressed by words in the natural language. *Cold*, *tepid* and *hot* are called **labels** of fuzzy sets.

For the following definitions, the fuzzy set Y is defined in the universe $\mathbf{Y}$. The elements in $\mathbf{Y}$ are denoted as $\{y\}$.

Definition 2.1 *A fuzzy set Y is* **normal** *if there exists at least one element $y \in \mathbf{Y}$*

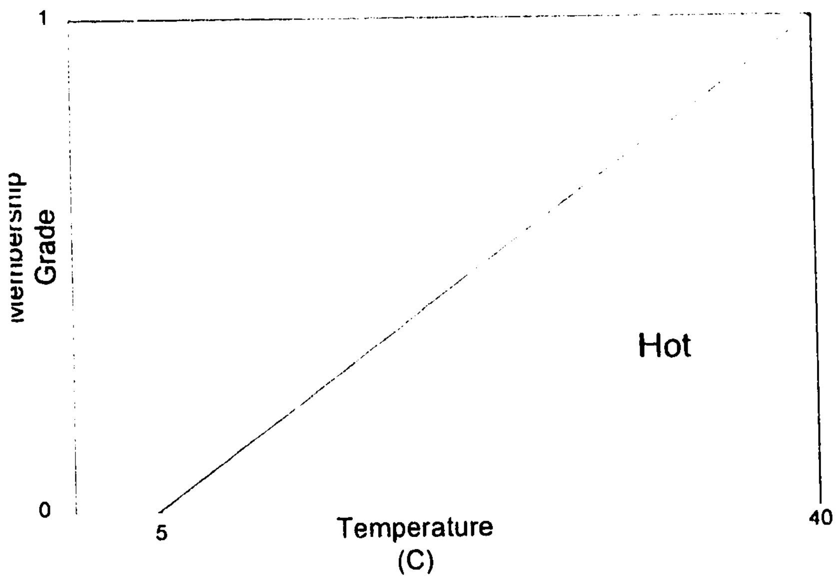

Figure 2.1: A Membership Function for Temperature

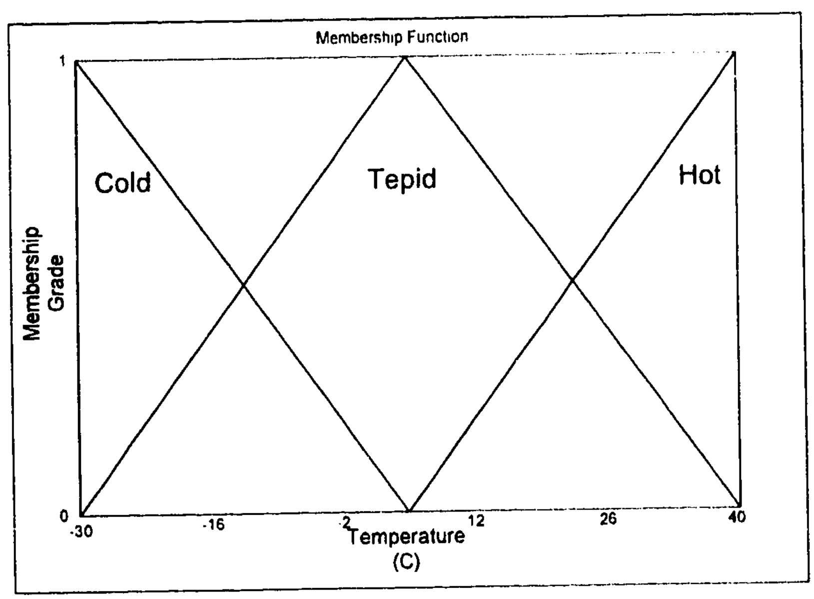

Figure 2.2: Membership Functions for Temperature

such that its membership grade in Y is 1.

Definition 2.2 *The* **support** *of a fuzzy set Y is the set of all elements y in* **Y** *such that the membership grade is greater than 0 i.e.* $\mu_Y(y) > 0$.

Definition 2.3 *The* **height** *of a fuzzy set Y is the largest membership grade of all elements in Y.*

Definition 2.4 *A* **fuzzy singleton** *is a fuzzy set whose support is a single point in* **U** *with a membership grade of 1.*

2.3 Set Theoretic Operations

For the following definitions, assume A and B to be fuzzy sets in **U**. $\mu_A(\cdot)$ indicates the membership grade of $(\cdot)$ in A.

Definition 2.5 *The* **union** *of A with B results in a fuzzy set with membership function given by*

$$\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u))$$

for all $u \in \mathbf{U}$. *The* max *operation is often denoted by* $\vee$.

Definition 2.6 *The* **intersection** *of A with B gives a fuzzy set with the membership function given by*

$$\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u))$$

for all $u \in \mathbf{U}$. *The* min *operation is often denoted by* $\wedge$.

Definition 2.7 *The* **negation** *or complement of A, denoted $\bar{A}$, gives a membership function given by*

$$\mu_{\bar{A}} = 1 - \mu_A(u)$$

for all $u \in \mathbf{U}$.

Definition 2.8 *Let $A_1, A_2, \ldots, A_n$ be fuzzy sets defined in $\mathbf{U}_1, \mathbf{U}_2, \ldots \mathbf{U}_n$ respectively. The* **Cartesian Product** *of $A_1, A_2, \ldots, A_n$ denoted $A_1 \times A_2 \times \ldots \times A_n$ is a fuzzy set in $\mathbf{U}_1, \mathbf{U}_2, \ldots \mathbf{U}_n$ with membership function given by*

$$\mu_{A_1 \times A_2 \times \ldots \times A_n}(u_1, u_2, \ldots, u_n) = \min\{\mu_{A_1}(u_1), \mu_{A_2}(u_2), \ldots, \mu_{A_n}(u_n)\}$$

or

$$\mu_{A_1 \times A_2 \times \ldots \times A_n}(u_1, u_2, \ldots, u_n) = \mu_{A_1}(u_1) \cdot \mu_{A_2}(u_2) \cdot \ldots, \mu_{A_n}(u_n)$$

for $u_i \in \mathbf{U_i}$ where $i = 1, 2, \ldots, n$.

Definition 2.9 *An n-ary* **fuzzy relationship** *is a fuzzy set in $\mathbf{U}_1 \times \mathbf{U}_2 \times \ldots \times \mathbf{U}_n$ and is expressed as*

$$\begin{aligned} R_{\mathbf{U}_1 \times \mathbf{U}_2 \times \ldots \times \mathbf{U}_n} &= \{((u_1, u_2, \ldots, u_n), \mu_R(u_1, u_2, \ldots, u_n)) \\ &\quad |(u_1, u_2, \ldots, u_n) \in \mathbf{U}_1 \times \mathbf{U}_2 \times \ldots \times \mathbf{U}_n\} \end{aligned}$$

2.4 Rule-Based Fuzzy Models

Zadeh (1973) proposed the use of *fuzzy conditional statements* in the modeling of systems which are too complex or ill-defined to be modelled by conventional mathematical techniques. The *fuzzy conditional statements* are rules of the form IF A THEN B where A and B have some meaning, e.g., IF T is *cold* THEN U is *large*. From a process control perspective, a set of rules may be derived and implemented as a control strategy. Roffel *et al.*(1991) and Oishi (1991) are some examples of the use of rule-based fuzzy logic in process control. Much interest is focused on controllers with a PID structure known as Fuzzy Logic Controllers (FLC). A typical rule in a FLC has the following form

$$\begin{aligned} &\textbf{IF } \text{error } e(k) \text{ is } ZERO \textbf{ AND } \text{change in error, } \Delta e(k) \text{ is } POSITIVE \\ &\textbf{AND } \text{sum of errors } \textstyle\sum e(k) \text{ is POSITIVE} \\ &\textbf{THEN } \text{the control } u(k) \text{ is } SMALLPOSITIVE \end{aligned} \tag{2.2}$$

The above fuzzy rule / model has a consequence which attaches a linguistic value, $SMALLPOSITIVE$ to the consequent variable, $u(k)$. Takagi & Sugeno (1985) proposed a fuzzy rule model where the consequent variable is some explicit mathematical function of the premise variables. In a FLC using the Takagi type model, a rule would have the following form :

$$\begin{aligned}&\textbf{IF} \text{ error } e(k) \text{ is } ZERO \textbf{ AND } \text{change in error, } \Delta e(k) \text{ is } POSITIVE\\&\textbf{AND} \text{ sum of errors } \textstyle\sum e(k) \text{ is POSITIVE}\\&\textbf{THEN} \text{ the control } u(k) = b_{1i}e(k) + b_{2i}\Delta e(k) + b_{3i}\textstyle\sum e(k)\end{aligned} \tag{2.3}$$

Often times a fuzzy model is required to process precise or crisp measurements. and produce a crisp output. Since a fuzzy model processes and produces fuzzy values, an interface before and after the fuzzy model is required. The configuration of a fuzzy model with the fuzzification and defuzzification interface is shown in Figure (2.3).

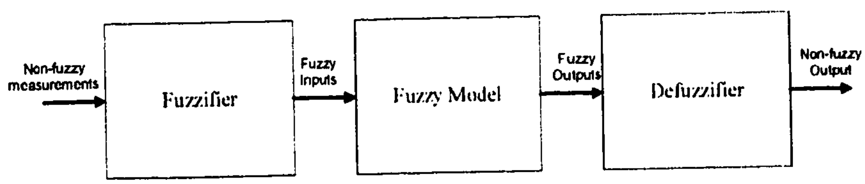

Figure 2.3: A Block Diagram of a Fuzzy Model Processing Crisp Data

The fuzzification interface transforms crisp data into membership grades in fuzzy sets defined in an universe of discourse. Given $e = e_o$, the membership grade of e_o in the fuzzy set E_i is calculated by :

$$\mu_{E_i}(e_o) = E_i(e_o) \tag{2.4}$$

where E_i is the membership function. Figure (2.4) illustrates the meaning of ([illegible])

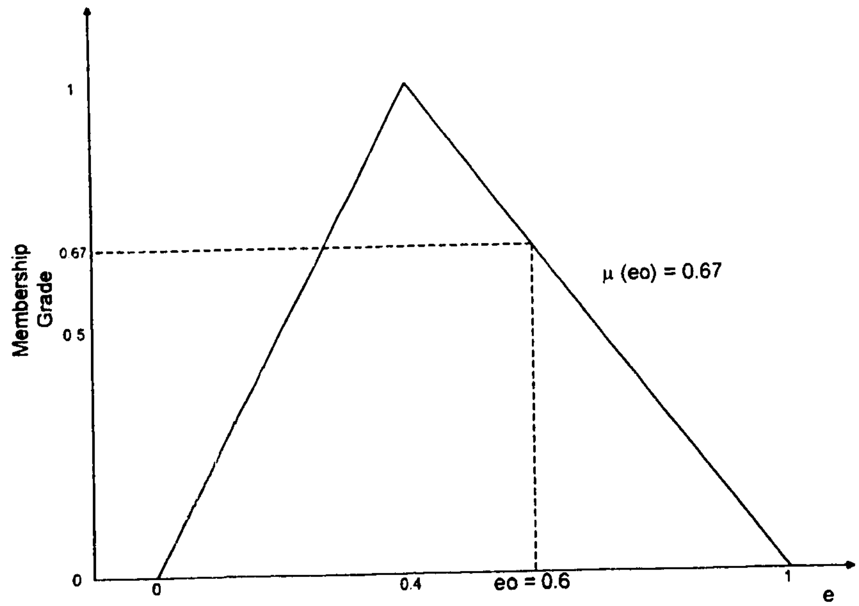

Figure 2.4: Fuzzification of Crisp Value

The defuzzification interface converts fuzzy sets into a crisp value. Hence, defuzzification performs the opposite task as fuzzification. There is no fixed method to perform defuzzification. The commonly used strategies are

- Average of Maxima
- Median
- Centre of Gravity
- Area

The most widely used strategy is the centre of gravity approach which is given by :

$$\tilde{e} = \frac{\sum_{j=1}^{N_e} \mu_{e_j} \cdot e_j^*}{\sum_{j=1}^{N_e} \mu_{e_j}}$$

where N_e denotes number of membership functions, μ_{e_j} is the membership grade in the j^{th} membership function and e_j^* is a representative value for the j^{th} membership function.

2.5 Reasoning In Rule-Based Systems

Lee (1990) and Yager & Filev (1994) provide a detailed description of the reasoning mechanism involved in rule-based systems. In this section, a graphical interpretation of the reasoning mechanism is given. For simplicity, two rules in a fuzzy logic controller with a PI structure will be considered. Consider the following rules :

$$\begin{aligned}
&\textbf{IF } e(k) = POSITIVE \quad \textbf{AND } \Delta e(k) = ZERO \\
&\qquad\qquad\qquad\qquad\qquad \textbf{THEN } \Delta u(k) = POSITIVE \\
&\textbf{IF } e(k) = ZERO \quad \textbf{AND } \Delta e(k) = NEGATIVE \\
&\qquad\qquad\qquad\qquad\qquad \textbf{THEN } \Delta u(k) = NEGATIVE
\end{aligned} \tag{2.5}$$

For the above rules, *Negative, Zero and Positive* are fuzzy sets. Before the reasoning process can be carried out, some mathematical operators must be specified for fuzzy connectives (**AND, OR**), fuzzy implication and aggregation. For

Fuzzy implication is represented by *minimum* and the aggregation is performed by *maximum*. The role of these operators in the reasoning process is best shown graphically. Figure (2.5) shows the details involved in rule-based reasoning with the rules as given in (2.5).

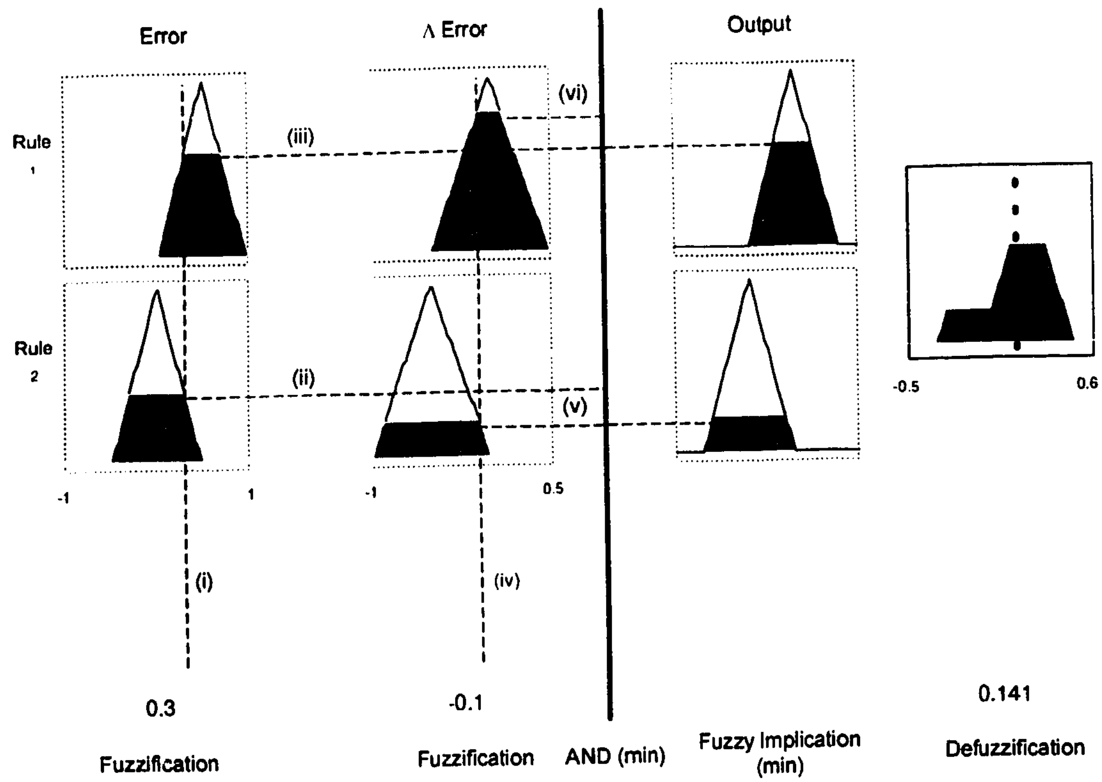

Figure 2.5: Graphical Illustration of Reasoning in Rule-Based System

The first step in the reasoning process as shown in Figure (2.5) involves the fuzzification of the crisp values for $e(k)$ and $\Delta e(k)$ via equation (2.4). The fuzzification of $e(k)$ yields two membership grades as indicated by lines (*ii*) and implemented by the *minimum* operator, the minimum of the two truth values in each rule (*i.e.* (*iii*) and (*vi*) for rule 1, and (*ii*) and (*v*) for rule 2) is found. For rule 1 and rule 2 the above mentioned operation gives (*iii*) and (*v*) respectively. A fuzzy set results from each rule after fuzzy implication. For rule 1, this fuzzy set is found by taking the minimum between (*iii*) and the output fuzzy set. This is repeated for the rule 2. To aggregate the two resultant fuzzy sets, the *maximum* operator is used which gives the fuzzy set as seen above "Defuzzification" in Figure (2.5). A suitable defuzzification is then used to compute a crisp value.

2.6 Fuzzy Logic Based Tuner for PID Controllers

Since close to 95% of feedback control loops still use conventional PID controllers, it is not surprising that there is a great deal of interest in improving the tuning of PID controllers. In recent years the development of PID tuners especially those that use fuzzy logic has increased (see Chand (1991), He *et al.* (1993) and Lee (1994)). In this section, the ability of fuzzy logic to capture the human decision-making process is demonstrated with the development of a tuner for PID controllers. Most tuners for PID controllers consists of two major components :

- A Metric for Performance
- Decision-making Logic

errors (ISE) to measure the performance of the controller. For the fuzzy tuner described here, the performance is measured by the control loop performance monitoring scheme of Huang *et al.* (1995). The performance monitoring scheme by Huang *et al.* (1995) is named "FCOR" and uses the minimum variance controller as its benchmark. A performance measure of 0 to 1 corresponds to the range from the worst to the best achievable control performance respectively. This performance measure is then used by the fuzzy-logic tuner which only modifies the proportional term in the PID controller. The task of the fuzzy-logic tuner is then to adjust the proportional gain so that the performance measure approaches 1. More specifically if increasing or decreasing the proportional action results in an increase in the performance index then the proportional action will be continually increased or decreased. However, the adjustment of the proportional gain is more conservative as the performance index approaches 1.

The update mechanism is as follows :

$$K_c(t) = K_c(t-1) + \Delta K_c(t) \tag{2.6}$$

$$\Delta K_c(t) = sign(\Delta K_c(t-1)) \cdot \delta(t) \tag{2.7}$$

$$\delta(t) = \alpha \cdot \Delta\eta(t) \tag{2.8}$$

$$\Delta\eta(t) = \eta(t) - \eta(t-1) \tag{2.9}$$

function respectively. t denotes the instant where adjustment of the proportional term is to take place. Clearly, tuning does not have to correspond to each and every sampling interval. The fuzzy tuner produces α given the current performance index and α determines the magnitude of ΔK_c. The rule-base in the fuzzy tuner may be as follows :

$$\begin{aligned} &\textbf{IF } \eta = SMALL \quad &\textbf{THEN } \alpha_1 = HIGH \\ &\textbf{IF } \eta = MEDIUM \quad &\textbf{THEN } \alpha_2 = MEDIUM \\ &\textbf{IF } \eta = LARGE \quad &= LOW \end{aligned} \tag{2.10}$$

The essense of these rules is that as ormance index approaches 1, the change made to the controller decreases. The meaning of the fuzzy sets $HIGH$, $MEDIUM$, LOW will determine the magnitude of the changes in K_c. The fuzzy-logic tuner was implemented on a simulated process with the following transfer function :

$$G_p(z) = \frac{0.39}{z^7 - 0.61z^6}$$

The following disturbance was assumed :

$$G_d(z) = \frac{z(z + 0.80)}{(z - 0.61)(z - 1)}$$

A block diagram of the process is shown in Figure (2.6) where $\epsilon(t)$ is a zero-mean white noise sequence. A change in K_c was made at every 600 sampling intervals. The initial tuning parameters were obtained using the Ziegler-Nichols

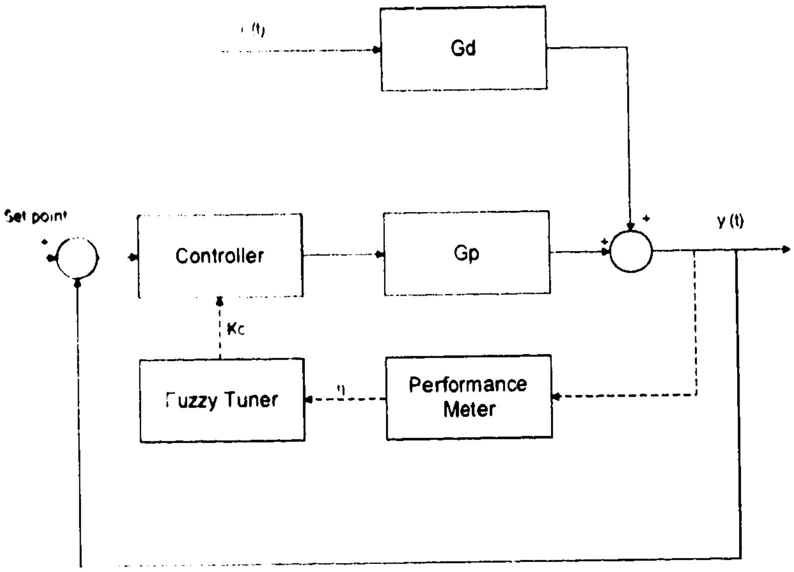

Figure 2.6: Block Diagram of Process with a Fuzzy-Logic Tuner for PID controllers

tuning method. The fuzzy model of Takagi & Sugeno (1985) was used where $HIGH$, $MEDIUM$ and LOW for α have the following meaning :

$$LOW = a_{10} + a_{11} \cdot \eta$$

$$MEDIUM = a_{20} + a_{21} \cdot \eta$$

$$HIGH = a_{30} + a_{31} \cdot \eta$$

For the following simulation, $a_{10} = 0$, $a_{11} = 0.2$, $a_{20} = 0.2$, $a_{21} = 0.4$, $a_{30} = 0.8$ and $a_{31} = -0.01$. Denoting

$$\alpha_i = a_{i0} + a_{i1} \cdot \eta$$

then α is calculated by

$$\alpha = \frac{\sum_{j=1}^{N_\alpha} \mu_j(\eta) \cdot \alpha_j}{\sum_{j=1}^{N_\alpha} \mu_j(\eta)}$$

where N_α corresponds to the number of rules. After a period of tuning the averaged performance index, $\bar{\eta}$, over 9000 sampling intervals was calculated together with the integral sum of error (ISE) and are tabulated in Table (2.1).

Table 2.1: Performance Measures for PID Controller after 9000 Sampling Intervals

	$\bar{\eta}$	ISE
Initial ZN tuning	0.41	187.7
With fuzzy tuner	0.43	180.5

Figure (2.7) shows the performance index η for both the above cases and it is worthwhile noting that the fluctuation in η is not due to the tuner since the η for the untuned case exhibits some fluctuation as well. It is also important to note that this fluctuation in η makes controller tuning difficult. Overall, the tuner appears to function well.

2.7 Conclusions

An introductory tutorial on rule-based fuzzy logic was presented in this chapter. The objective was to provide the reader with an appreciation of this area and the

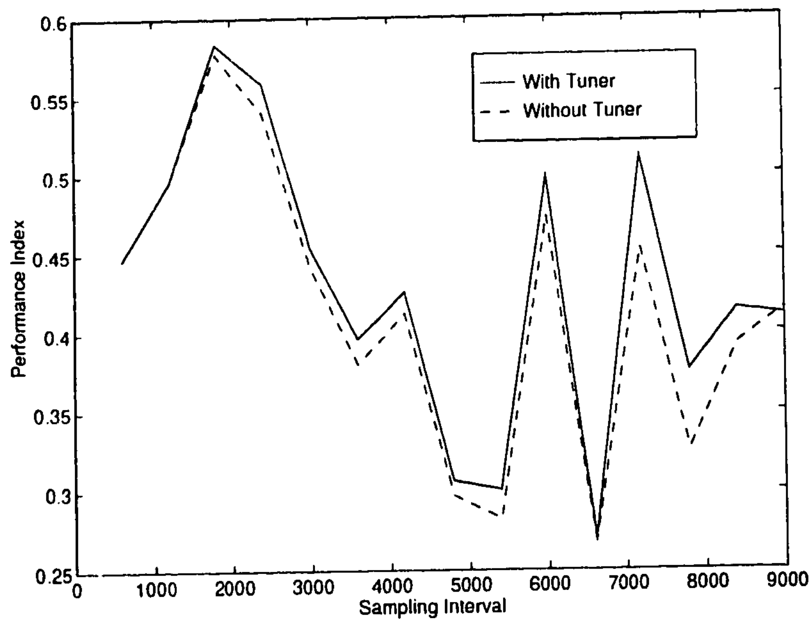

Figure 2.7: Performance Index, η, for the Controller with and without the Fuzzy Tuner

essential theory involved in rule-based fuzzy logic. An illustration of the reasoning mechanism was given which was followed by the application of fuzzy-logic in the development of a fuzzy-tuner for the proportional mode of PID controllers. The fuzzy-tuner developed gave better controller performance compared to a controller without the fuzzy-tuner.

Bibliography

[1] S. Chand. On-line self-monitoring tuner for proportional integral derivative controllers. In *Proceedings of the 30th IEEE Conference on Decision and Control*, number 2, pages 1905 1906, Brighton, England, 1991.

[2] S.Z. He, S.H. Tan, and P.Z. Wang. PID self-tuning control using a fuzzy adaptive mechanism. In *Second IEEE International Conference on Fuzzy Systems*, pages 708 713, San Francisco, CA, USA, 1993.

[3] B. Huang, S. Shah, and K. Kwok. On-line Control Performance Monitoring of MIMO Processes. In *Proceedings of American Control Conference*, pages 1250-1254, Seattle, Washington, USA, 1995. American Control Conference.

[4] A. Kaufmann. *Introduction to the Theory of Fuzzy Subsets*, volume 1. Academic Press, New York, 1975.

[5] C.C. Lee. Fuzzy logic in control systems : fuzzy logic controller, Parts I and II. *IEEE Trans. Systems Man and Cybernetics*, 20(2):404 435, 1990.

[6] K.B. Lee. Application of a fuzzy logic based self-tuning PID controller to the control of drum water level of a boiler for a power generating plant.

In *Proceedings of the 2nd International Conference on Advances in Power System Control, Operation and Management*, volume 2, pages 849–854, Hong Kong, 1994.

[7] K. Oishi, M. Tominaga, A. Kawato, Y. Abe, S. Imayasu, and A. Namba. Application of fuzzy control theory to the Sake brewing process. *Journal of Fermentation and Bioengineering*, 72(2):115–121, 1991.

[8] B. Roffel and P. Chan. Fuzzy Control of a Polymerization Reactor. *Hydrocarbon Processing*, pages 47–49, June 1991.

[9] T. Tagaki and M. Sugeno. Fuzzy identification of systems and its applications to modelling and control. *IEEE Trans. Systems Man and Cybernetics*, 15(1):116–132, 1985.

[10] R.R. Yager and D.P. Filev. *Essentials of Fuzzy Modeling and Control*. John Wiley and Sons Inc., 1994.

[11] L.A. Zadeh. Fuzzy Sets. *Inform. and Control*, 8:338–353, 1965.

[12] L.A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Systems Man and Cybernetics*, 3(1):28–44, 1973.

[13] Hans J. Zimmermann. *Fuzzy Set Theory - and Its Applications*. Kluwer Nijhoff Publishing, 1985.

Chapter 3

Fuzzy Relational Systems

The importance of relations is almost self evident. Science is in a sense, the discovery of relations between observables ...

Goguen (1967)

3.1 Introduction

Engineers are frequently interested in quantifying the behaviour of a system in terms of some known variables. In highly complex systems, it may be difficult to express such a relationship in a useful mathematical form. In addition, there are also processes where information cannot be represented in a deterministic or crisp fashion. Fuzzy relational equations provide a platform to represent such complexities and fuzziness. Consequently, fuzzy relational equations have been applied to a broad class of problems such as medical diagnosis (Sanchez(1977)), fuzzy controllers (Mamdani(1976) , Kickert & Mamdani (1978)), technical diagnostics

(Tsukamoto(1978)) and decision-making (Bellman & Zadeh (1970),

The main purpose of this chapter is to provide a foundation for the development of fuzzy relational equations for a system identification and control application. The next section will give a description of the terminology and notation which is then followed by some interpretations of fuzzy relational equations. In addition, some results for the analytical resolution of fuzzy relational equations will be presented. Finally, an explanation of the justification for an alternative to the analytical solution of fuzzy relational equations will be given.

3.2 Notation and Terminology

A static single input single output fuzzy relational equation can be expressed as

$$Y = X \circ R \tag{3.1}$$

where

$X \in \mathrm{F}(\mathbf{X})$ and $Y \in \mathrm{F}(\mathbf{Y})$

$R \in \mathrm{F}(\mathbf{X} \times \mathbf{Y})$

$\circ$ denotes the max triangular norm operator

Here $\mathrm{F}(\cdot)$ denotes a family of fuzzy sets (or relations) defined in a relevant universe of discourse (or a Cartesian product space) . For example, $X \in \mathrm{F}(\mathbf{X})$ means that X is defined in a universe of discourse $\mathbf{X}$, and $R \in \mathrm{F}(\mathbf{X} \times \mathbf{Y})$ denotes a fuzzy relation defined in the Cartesian product of $\mathbf{X}$ and $\mathbf{Y}$. R is also known as the relational matrix. The notation introduced above will be adhered to for the remainder of this chapter.

Examples of triangular norm operators are minimum, algebraic product, bounded product and drastic product. If the algebraic product is chosen as the triangular norm operator then the membership grade of Y' is found by:

$$Y'(y) = \max_{x \in \mathbf{X}} \; [X'(x) * R(x, y)] \tag{3.2}$$

where

$*$ is the algebraic product

Relationships involving more than one input can be similarly expressed as

$$Y' = X_1 \circ X_2 \circ \ldots \circ X_n \circ R \tag{3.3}$$

where $R \in \mathrm{F}(\mathbf{X_1} \times \mathbf{X_2} \times \cdots \times \mathbf{X_n} \times \mathbf{Y})$.

And if the triangular norm is substituted by the algebraic product, the degree of membership is given by :

$$Y'(y) = \max_{\substack{x \in \mathbf{X_1} \\ x \in \mathbf{X_2} \\ \vdots \\ x \in \mathbf{X_n}}} [X_1(x_1) * X_2(x_2) * \ldots * X_n(x_n) * R(x_1, x_2, \ldots, x_n, y)] \tag{3.4}$$

3.3 Interpretation of Fuzzy Relational Equations

An interpretation of the fuzzy relational equation is provided in Pedrycz (1984). Consider a fuzzy relational equation of the form :

$$X_{k+1} = U_{k-\tau} \circ X_k \circ R \tag{3.5}$$

where τ is the time-delay of the system. U and X are possibility vectors representing the input and state respectively. The relational matrix R has $N_{U_{k-\tau}}$ N_{X_k} $N_{X_{k+1}}$ implication statements where $N_{(\cdot)}$ denotes the number of term sets (fuzzy subsets in the membership function) for the variable $(\cdot)$. The implication statements are of the following form :

$$\begin{aligned} &\text{if input is } U_i \text{ and state (k}^{th} \text{ time moment) is } X_j \\ &\quad \text{then state ((k + 1)}^{th} \text{ time moment) is } X_l \end{aligned} \tag{3.6}$$

i=1,2,..., $N_{U_{k-\tau}}$, j=1,2,...,N_{X_k}, l=1,2,...,$N_{X_{k+1}}$ with a possibility measure λ_{ijl} assigned to each implication statement. The possibility measure, λ_{ijl}, corresponds to the $(ijl)^{th}$ entry of the matrix R. When $U_{k-\tau}$ and X_k are known and transformed according to (3.5), we obtain $N_{X_{k+1}}$ possibility measures which correspond to the following linguistic interpretation :

Possibility that the state X_{k+1} is X_1 $= \lambda_1$
Possibility that the state X_{k+1} is X_2 $= \lambda_2$
⋮

Possibility that the state X_{k+1} is $X_{X_{X_{k+1}}} = \lambda_{X_{X_{k+1}}}$

An alternative interpretation of the relational equation is that it is simply a mapping between the fuzzy sets of the variables present in the equation which for (3.5) are $U_{k-\tau}$, X_k and X_{k+1}. Consider the following example.

Example

Let X = { x_1, x_2, x_3 }, Y = {y_1, y_2, y_3 }, and

$$Y = X \circ R$$

Let

$$R = \begin{bmatrix} (x_1, y_1) & (x_1, y_2) & (x_1, y_3) \\ (x_2, y_1) & (x_2, y_2) & (x_2, y_3) \\ (x_3, y_1) & (x_3, y_2) & (x_3, y_3) \end{bmatrix}$$

$$= \begin{bmatrix} 0.1 & 0.2 & 0.9 \\ 0.0 & 1.0 & 0.2 \\ 0.8 & 0.2 & 0.0 \end{bmatrix}$$

Cell (x_i, y_j) corresponds to the likelihood or possibility that Y is y_j given X is x_i. For the above, x_i and y_j may have linguistic values such as the following:

$x_1 = \textit{low}$, $x_2 = \textit{medium}$, $x_3 = \textit{high}$
$y_1 = \textit{negative}$, $y_2 = \textit{zero}$, $y_3 = \textit{positive}$

Therefore, if X is arbitrarily defined as [0.8 0.2 0], Y is obtained as follows:

$$\begin{aligned} Y &= X \circ R \\ &= [0.80\ 0.20\ 0.00] \circ \begin{bmatrix} 0.1 & 0.2 & 0.9 \\ 0.0 & 1.0 & 0.2 \\ 0.8 & 0.2 & 0.0 \end{bmatrix} \end{aligned}$$

If $\circ$ = max-product, applying (3.2) gives

$$= [0.08\ 0.20\ 0.72]$$

Y can then be interpretated as follows :

Membership grade of Y in the linguistic value *negative* is 0.08

Membership grade of Y in the linguistic value *zero* is 0.20

Membership grade of Y in the linguistic value *positive* is 0.72

□

The two interpretations of the fuzzy relational equations can be easily extended to relational equations of higher order.

3.4 Resolution of Fuzzy Relational Equations

Taking (3.1) into account, the problem of resolution of fuzzy relational equations can be stated as follows :

1. X, Y are given, determine R

2. Y, R are given, determine X

(1) is the identification problem, that is, determine the model parameters given a set of input-output data. (2) is the control problem which involves finding the input which achieves some objective given the model parameters. The resolution of these equations was first performed by Sanchez (1976) . But before stating the results, the following definitions are required.

Definition 3.1

$$a \varphi b = sup\{c \in [0,1] \mid a \ t \ c \leq b\}$$

If the triangular norm operator specified is the *minimum*, then the α operator results.

Definition 3.2

$$\begin{aligned} a \alpha b &= sup\{c \in [0,1] \mid a \wedge c \leq b\} \\ &= \begin{cases} 1 & \text{if } a \leq b \\ b & \text{if } a > b \end{cases} \end{aligned}$$

Considering equation (3.1) for problems (1) and (2), we first assume that the corresponding families of solutions of $\mathcal{R}$ and $\mathcal{X}$ are non-empty, namely

$$\mathcal{R} = \{R \mid X \circ R = Y\} \neq \phi$$

$$\mathcal{X} = \{X \mid X \circ R = Y\} \neq \phi$$

Theorem 3.1 *For problem (1), if $\mathcal{R} \neq \phi$, then the greatest element of $\mathcal{R}$ denoted by $\hat{\mathcal{R}}$, is given by :*

$$\hat{\mathcal{R}} = X \varphi Y$$

Theorem 3.2 *For problem (2), if $\mathcal{X} \neq \phi$, then the greatest element of $\mathcal{X}$, denoted by $\hat{\mathcal{X}}$ is given by :*

$$\hat{\mathcal{X}} = R \varphi Y$$

The proof of the above propositions can be found in Sanchez (1976).

3.5 Why Alternative Solutions?

The analytical resolution of a fuzzy relational equation relies on the assumption that the family of solutions is a non-empty one. However, it is difficult to guarantee that the family of solutions is indeed non-empty. When more than one fuzzy relational equation has to be solved, the likelihood that there exists a non-empty solution is reduced. This is often encountered in system identification whereby a set of data is available. To illustrate, given a set of fuzzy input-output data namely (X_1, Y_1), (X_2, Y_2), $\ldots$, (X_j, Y_j) , $\ldots$, (X_K, Y_K) where X_j and Y_j $i, j =$ 1,2, $\ldots$, K are fuzzy sets defined in universes of discourse **X** and **Y**. Then we have the following equations,

$$
\begin{aligned}
X_1 \circ R &= Y_1 \\
X_2 \circ R &= Y_2 \\
&\vdots \\
X_j \circ R &= Y_j \\
&\vdots \\
X_K \circ R &= Y_K
\end{aligned} \tag{3.7}
$$

Denoting $\mathcal{R}_j$ as a family of fuzzy relations satisfying the j^{th} equation of (3.7) as follows

$$\mathcal{R}_j = \{R \in \mathbf{F}(\mathbf{X} \times \mathbf{Y}) \mid X_j \circ R = Y_j\}$$

and assume $\mathcal{R}_j \neq \phi$. If the intersection of the families of equations forms a non-empty set, that is,

$$\mathcal{R} = \bigcap_{j=1}^{K} \mathcal{R}_j \neq \phi$$

then its greatest element, denoted $\hat{\mathcal{R}}$ is given by

$$\hat{\mathcal{R}} = \bigcap_{k=1}^{K} (X_k \; \alpha \; Y_k)$$

It is easy to see that if one pair of (X_{jo}, Y_{jo}) has an empty solution set,

$\mathcal{R}_{jo}$, the intersection with all the remaining families of solutions $\mathcal{R}_j$ where $j \neq jo$, namely

$$\mathcal{R}' = \bigcap_{j \neq jo}^{K} \mathcal{R}_k \neq \phi$$

and

$$\mathcal{R}_{jo} \bigcap \mathcal{R}' = \phi$$

This implies that the entire system of equations has no solution. In short, if *at least one* data pair of (X_j, Y_j), $j = 1, 2, \ldots, K$ results in an empty solution, the analytical solution will not exist for the entire system of equations. In a real plant environment, an analytical solution of fuzzy relational equations may not exist due to noise, disturbance and model plant mismatch. Therefore, alternative solutions to the identification and control problem are needed. They are discussed in Chapters 4 and 6.

3.6 Summary

This chapter has reviewed the basics of fuzzy relational equations. In addition, interpretations of the fuzzy relational equation were given. It has also been pointed out why an exact analytical solution to the identification and control problem may not exist.

Bibliography

[1] R.E. Bellman and L.A. Zadeh. "Decision-making in a fuzzy environment". *Management Sci.*, 17:141–154, 1970.

[2] W. J. M. Kickert and E. H. Mamdani. "Analysis of a fuzzy logic controller". *Fuzzy Sets and Systems*, 1:169–180, 1978.

[3] E. H. Mamdani. "Advances in the linguistic synthesis of fuzzy controllers". *Int. J. Man-Machine Studies*, 8:669–678, 1976.

[4] W. Pedrycz. "An Identification Algorithm in Fuzzy Relational Systems". *Fuzzy Sets and Systems*, 13:153–167, 1984.

[5] E. Sanchez. "Resolution of composite fuzzy relation equations". *Inform. and Control*, 30:38–48, 1976.

[6] E. Sanchez. "Solutions in composite fuzzy relation equations : Applications to medical diagnosis in Brouwerian logic". In M.M. Gupta, G.N. Saridis, and B.R. Gaines, editors, *Fuzzy Automata and Decision Processes*, pages 221–234. North Holland, New York, 1977.

[7] Y. Tsukamoto and T. Terano. "Diagnosis of engine trouble by fuzzy logic". In *Summary of Papers on General Fuzzy Problems*, volume 4, pages 16–19. 1978.

Chapter 4

Fuzzy Relational Identification

4.1 Introduction

The use of fuzzy models has [illegible] gaining popularity in recent years because they have the ability to approximate processes which are not well modelled by conventional identification techniques. The fuzzy identification goal involves the derivation of a set of rules or fuzzy implications. Fuzzy models can be divided into rule-based or relational based. The identification procedure for rule-based systems usually involves the translation of knowledge from an expert usually a process operator. On the other hand, fuzzy relational identification accepts input-output data and offers a systematic procedure for model development.

This chapter will focus on fuzzy relational identification. After an introductory tutorial to fuzzy relational identification, a survey of existing identification algorithms will be presented. This survey will touch on some key ideas which are essential to the understanding of fuzzy relational identification.

4.2 Fuzzy Identification Procedure

Consider the following single-input single-output fuzzy relational equation of order n:

$$Y(k) = Y(k-1) \circ Y(k-2) \circ \ldots \circ Y(k-n) \circ U(k-d-1) \circ U(k-d-2) \circ \ldots \circ U(k-d-p) \circ R \tag{4.1}$$

The objective of fuzzy relational identification is to determine the parameters in the relational matrix, R. The number of parameters in R is given by the product of the following terms:

$$N_{Y(k-1)}\ N_{Y(k-2)} \ldots\ N_{U(k-d-1)}\ N_{U(k-d-2)} \ldots N_{U(k-d-p)}\ N_{Y(k)}$$

where $N_{(\cdot)}$ denotes the number of fuzzy sets associated with $(\cdot)$. The identification procedure can be summarized as follows :

- Define the universes of discourse for input-output variables.
- Select the number and shape of referential membership functions for each variable.
- Fuzzifiy data in terms of membership functions defined previously.
- Select model order and choice of compositional operator.
- Determine model parameters in relational matrix.
- Model Validation

4.2.1 Universe of Discourse

The universe of discourse for a given variable is the range which covers all possible values for that variable. For example, the universe of discourse for an input which corresponds to a valve position is from 0-100 %.

4.2.2 Selection of Membership Function

For each variable present in the fuzzy model, membership functions must be selected. The membership functions are made up of fuzzy subsets which may have linguistic values attached to them. A variable such as temperature may have membership functions as shown in Figure (4.1) which consists of five fuzzy sets. The fuzzy sets defined are also called referential fuzzy sets. The selected referential fuzzy sets will be used in fuzzification and defuzzification. The choice of membership functions involves the selection of the number, shape and distribution of the referential fuzzy sets. The number of referential membership functions selected impacts on the resolution or accuracy of the model. However, a larger number of referential fuzzy sets results in an exponential increase in the number of parameters in the relational matrix. For example, for (4.1) if $n = 1$ and $p = 1$ and if both input and output variables have 5 referential fuzzy sets, then the relational matrix will have $5^3 = 125$ elements. If the number of referential fuzzy sets were increased to 7, then there will be $7^3 = 343$ elements in the relational matrix. If n and p are both increased to 2 with 7 referential fuzzy sets then the relational matrix will have $7^5 = 16807$ elements. In short, an increase in referential fuzzy

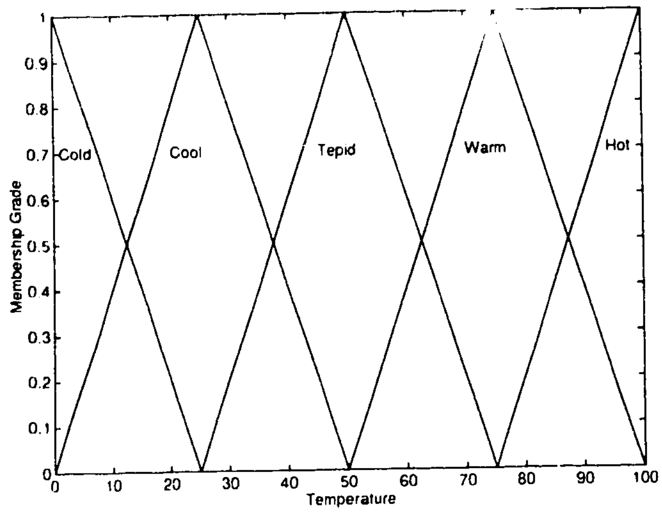

Figure 4.1: An Example of Membership Functions

sets increases storage and computational requirements. Therefore, one should be parsimonious on the choice of the number of referential fuzzy sets.

Before proceeding any further, some definitions must be given.

Definition 4.1 *A fuzzy subset Y defined in the universe* $\mathbf{Y}$ *is* **normal** *if there exists at least one element, $y \in \mathbf{Y}$ such that its membership grade in Y is 1.*

Definition 4.2 *A fuzzy subset Y is* **convex** *if*

$$\mu_Y(\ \lambda y_1 + (1-\lambda)y_2\ )\ \geq\ min(\mu_Y(y_1)\ ,\ \mu_Y(y_2))$$

for $y_1, y_2 \in \mathbf{Y}$, $\lambda \in [0,1]$. $\mu_Y(\cdot)$ denotes the membership grade of $(\cdot)$ in the fuzzy subset Y. A graphical interpretation of convexity and non-convexity can be found in Figure (4.2) and Figure (4.3).

To ensure the performance of the fuzzy model, the referential fuzzy sets should be normal, convex and satisfy the completeness condition as outlined below.

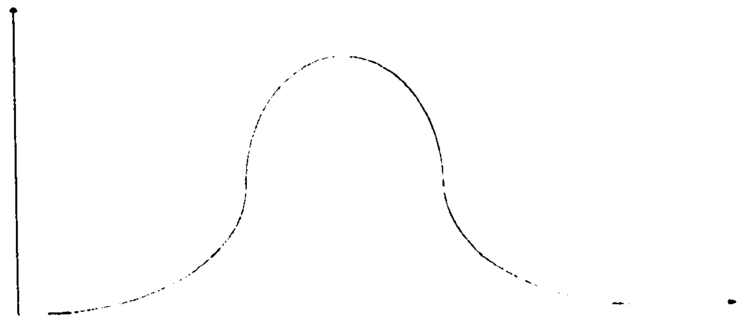

Figure 4.2: A Convex Fuzzy Set

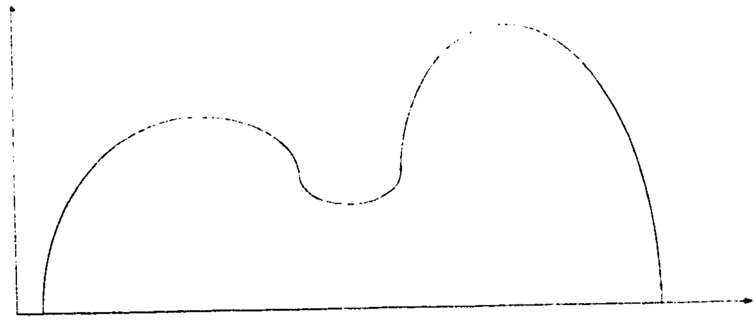

Figure 4.3: A Non-Convex Fuzzy Set

Definition 4.3 *If $Y_1, Y_2, \ldots, Y_c$ are the referential fuzzy sets defined in the universe of discourse* **Y**, *then*

$$\forall\ y(k) \text{ in } \mathbf{Y}\ \exists\ 1 \leq i \leq c \text{ such that } \mu_{Y_i}(y(k)) > 0 \tag{4.2}$$

for $k = 1, 2, \ldots, N$ where N is the number of data points.

This means that for any given data point, its membership grade must be non-zero in at least one referential fuzzy set. Although the conditions given above are necessary, there still remains much choice in specifying the shape and distribution of the referential membership functions. Previously, the selection was done subjectively or by clustering techniques. Xu & Lu (1987) states that there is no evidence to support that any analytical or computational method yields better results than a subjective decision. Recently, de Oliveira (1993) has formulated the problem of membership function selection into a contrained optimization problem. Based on the Box-Jenkins gas furnace data set, the author showed that the selection of membership functions via the contrained optimization method yields superior results compared to fuzzy clustering. Pedrycz (1994) provided a theroretical motivation behind the common use of triangular membership functions. More specifically, if triangular membership functions with a specific overlap of $\frac{1}{2}$ (namely the height of intersection of each two successive fuzzy sets equals $\frac{1}{2}$), the objective used in the constrained optimization method will be satisfied. Based on these developments, there is now a systematic approach in membership function selection.

4.2.3 Fuzzification

The fuzzification step converts either a fuzzy or non-fuzzy value into a possibility vector based on the previously defined referential fuzzy sets. The possibility vector has N elements where N is the number of referential fuzzy sets in a particular universe of discourse. The i^{th} element in a possibility vector is the degree of membership of the original value (fuzzy or non-fuzzy) in the i^{th} referential fuzzy set. The process of fuzzification is described below.

Given a fuzzy set $\mathcal{Y}$ defined in $\mathbf{Y}$ with reference fuzzy sets $Y_1, Y_2, \ldots, Y_{N_y}$, the possibility vector $p = [p_1\ p_2 \cdots\ p_{N_y}]$ can be found via

$$p_i = Possibility(\mathcal{Y} \mid Y_i) \triangleq \sup_{y \in \mathbf{Y}} [\mathcal{Y} \; t \; Y_i] \tag{4.3}$$

where $p_i \in [0, 1]$ and t is a triangular-norm operator. If given a crisp or non-fuzzy value y_o, then (4.3) reduces to

$$p_i = \mu_{Y_i}(y_o) \tag{4.4}$$

where $\mu_{Y_i}(\cdot)$ represents the membership grade of $(\cdot)$ in Y_i.

4.2.4 Model Structure

For the input-output fuzzy relational equation defined in (4.1), the model order, n and p, the time-delay, d, and the composition operator $\circ$ must be selected.

Once again, the selection of n and p has an impact on storage and computational requirements. The number of parameters grows exponentially with increased model order. It is important to strike a balance between model performance and computational demands when choosing the model order. The time-delay, d, can be estimated based on some *a priori* knowledge about the plant. This estimated value can be further refined by defining an objective function such as the mean prediction error and picking the time delay which minimizes the objective function.

There is a family of compositional operators that can be applied to (4.1). Any max triangular-norm operator can be used but the most widely used ones are the max-min and max-product. Bourke (1995) explored the use of these two operators when applied to identification and control.

4.2.5 Parameter Estimation

In recent years, much effort has been devoted to the development of better schemes to estimate the parameters in the relational matrix. There exist many algorithms but they can be divided into two groups namely

- Linguistic Algorithms
- Numerical Algorithms

Some of the more notable algorithms which are classified as linguistic are Pedrycz (1984) , Xu & Lu (1987) , Shaw & Krüger , Chen *et al.* (1994) and Bourke (1995) . Pedrycz (1983, 1991) , Ikoma *et al.* (1993) and Valente de Oliveira (1993a) presented numerical identification algorithms. A more in-depth coverage of these identification algorithms will be the subject of the next section.

4.2.6 Model Validation

As in conventional identification techniques, the model validation stage verifies the accuracy of the identified model. The predicted model output is compared with the actual process output. If the actual output is deterministic or crisp then the fuzzy values obtained via the fuzzy model must be defuzzified.

Defuzzification refers to the process of converting a fuzzy possibility vector into a crisp or deterministic value. There are various defuzzification techniques namely :

- Average of Maxima
- Median
- Centre of Gravity
- Area

Of the above methods, the centre of gravity method is the most popular. However, there are results (see Mizumoto (1989)) which suggest that it does not necessarily give the best performance. It must be pointed out that a fair comparison of defuzzification techniques is rather difficult due to the fact that the performance of a particular defuzzification techniques may rely on the identification algorithm used and the data set which was used in the identification exercise.

4.3 Survey of Existing Identification Methods

In order to present a complete foundation for fuzzy relational identification, the key idea behind some of the popular identification techniques will be investigated. Pedrycz's identification algorithm (1984) and its many variants will be presented first, followed by Shaw's (1992) algorithm together with some of its modifications. Identification methods which solve fuzzy relational equations numerically will also be dealt with.

4.3.1 Pedrycz's Identification Algorithm (1984)

The identification algorithm by Pedrycz (1984) is considered to be a linguistic algorithm. Consider a first order SISO fuzzy relational equation such as the following :

$$Y(k) = Y(k-1) \circ U(k-d-1) \circ R \tag{4.5}$$

where $U(k-d-1), Y(k-1)$ and $Y(k)$ are possibility vectors. Czogala *et.al.* (1981) has shown that there exists a family of relational matrices satisfying (4.5), bounded by a least upper bound matrix, $\hat{R}(k)$ and a greatest lower bound matrix, $\check{R}(k)$. Any solution $R(k)$, satisfies

$$\check{R}(k) \;\leq\; R(k) \;\leq\; \hat{R}(k)$$

One such solution is given by

$$R'(k) \;=\; U(k-d-1) \times Y(k-1) \times Y(k) \tag{4.6}$$

where $\times$ is the Cartesian product. Then the final relational matrix, $\tilde{R}$, derived from an entire data set, is calculated by

$$R = \bigcup_{k=d+2}^{N} R'(k) \tag{4.7}$$

where $\bigcup$ stands for union. The cartesian product, $\times$, is calculated with respect to the composition operator which is applied as [illegible]

operator then $\times$ is the minimum. If $\circ$ represents the max-product operator then $\times$ is the product. To estimate the time delay for a system with crisp measurements, the optimum time-delay would be one which minimizes the sum of squares of the prediction errors :

$$J = \sum_{k=d_{opt}+2}^{N} (y(k) - \hat{y}(k))^2 \tag{4.8}$$

Similarly, a suitable performance index for fuzzy data can be defined as

$$J = \sum_{k=d_{opt}+2}^{N} \sum_{i=1}^{N_{Y_2}} (Y_i(k) - \hat{Y}_i(k))^2 \tag{4.9}$$

where N_{Y_2} is the number of referential fuzzy sets in Y_k. This identification algorithm maximizes the relational matrix over all data points. In a real plant environment, disturbances usually corrupt the data set. Since relational models such as 4.5 do not explicitly contain a disturbance variable, the identified model will be corrupted with the disturbance. Therefore, if a maximization of the relational matrix is performed over all data points, the fuzzy model will not have the ability to forget corrupted data or forget *old* data of a slowly time-varying process. Xu & Lu (1987) and later Chen (1994) attempted to remedy the drawback of this algorithm.

Identification Algorithm of Xu (1987)

The essence of this on-line algorithm is the modification of the relational matrix via a tuning parameter, a. The new estimate of the relational matrix at the k^{th} instant, $\hat{R}(k)$, is updated by

$$\hat{R}(k) = a \cdot R'(k) + (1 - a) \cdot \hat{R}(k-1) \tag{4.10}$$

where $R'(k)$ is given by (4.6). When $a = 0$, no update is performed while if $a = 1$, the previous relational matrix is totally replaced with information from the latest data which is incorporated into $R'(k)$. The magnitude of this tuning parameter is influenced by

1. The amplitude of the prediction error, $e(k)$
2. The relative contribution of the rules, β_s, which give the predicted output, β_s.

The algorithm can be summarized as follows :

- Use $\hat{R}(k-1)$ and data $U(k-d-1)$, $Y(k-1)$ to produce a prediction, $\hat{Y}(k)$.
- Calculate $R'(k)$ by (4.6).
- Determine prediction error, $e(k) = y(k) - \hat{y}(k)$
- The relative contribution $\beta_s \triangleq [\hat{Y}_s]^2$ where $s = 1,2, \ldots, N_{Y_2}$.
- The tuning parameter is found by $a = h \cdot \beta_s \cdot |e(k)|$ where s=1,2, …, N_{Y_k}. h is a constant used to control the range of a_s.

- If $e_k > \epsilon$ where ϵ is some tolerance level, update the relational matrix via (4.10).

Identification Algorithm by Chen *et. al.*(1994)

Chen *et. al.* (1994) proposed to improve on Pedrycz's (1984) algorithm by introducing a forgetting factor which enables the elimination of erroneous or old data. The update is performed by :

$$\tilde{R}(k) = [\tilde{R}(k-1) \wedge \overline{\beta \cdot R''(k)}] \vee \alpha \cdot R'(k) \tag{4.11}$$

where

$$R''(k) = U(k-d-1) \times Y(k-1) \times \hat{Y}(k)$$

$$\bar{R}''(k) = 1 - R''(k)$$

$$R'(k) = U(k-d-1) \times Y(k-1) \times Y(k)$$

β is the forgetting factor which is some function of the prediction error. β is directly proportional to the rate of forgetting. The first part of (4.11) *ie.* $(R(k-1) \wedge \overline{\beta \cdot R''(k)})$ forgets data while the second part *ie.* $(\alpha \cdot R'(k))$ incorporates the latest information into the relational matrix.

Since $R''(k)$ contains all parameters which give the current prediction, and if the current prediction is poor then these parameters must be updated. It is important to remember that $\hat{Y}(k)$ is taken over a maximum. Therefore, the first part of (4.11) reduces the relevant elements in R so that its effect on $\hat{Y}(k)$ will be diminished. α is defined as

$$\alpha = \begin{cases} 1 & \text{if } |e| > \epsilon \\ 0 & \text{if } |e| \leq \epsilon \end{cases}$$

Therefore, if the error is less than some specified tolerance, the latest data is not incorporated into the relational matrix.

4.3.2 Weighted Average Identification Algorithms

Shaw & Krüger (1992) proposed a probablistic fuzzy relation building method. The key feature of this approach is to treat each entry in the relational matrix as the possibility of obtaining an output referential fuzzy set given a referential fuzzy set for each state and input variable in the relational equation. If the fuzzy relational equation has the form as given in (4.5), then the $(ijl)^{th}$ entry of R, R_{ijl}, is the possibility of obtaining $Y_i(k)$ from $Y_j(k-1)$ and $U_l(k-d-1)$ where i, j and l indicates the i^{th}, j^{th} and l^{th} referential fuzzy set respectively.

The relational matrix is found via

$$R = \frac{\sum_{k=d+2}^{N} \prod_{l=1}^{N_u} \prod_{j=1}^{N_x} \prod_{i=1}^{N_y} (U_l(k-d-1), Y_j(k-1), Y_i(k))}{\sum_{k=d+2}^{N} \prod_{l=1}^{N_u} \prod_{j=1}^{N_x} (U_l(k-d-1), Y_j(k-1))} \qquad (4.12)$$

Bourke (1995) modified the above equation to

$$R = \frac{\sum_{k=d+2}^{N} [\prod_{l=1}^{N_u} \prod_{j=1}^{N_r} (U_l(k-d-1), Y_j(k-1) \bigodot_{i=1}^{N_y} Y_i(k))]}{\sum_{k=d+2}^{N} \prod_{l=1}^{N_u} \prod_{j=1}^{N_r} (U_l(k-d-1), Y_j(k-1))} \tag{4.13}$$

where $\odot$ is the minimum inverse operator for the max-product composition defined below.

Definition 4.4 *For a and $b \in [0,1]$,*

$$a \bigodot b = \begin{cases} 0 & \textit{if } a < b \textit{ or } a = b = 0 \\ \frac{b}{a} & \textit{if } a \geq b \end{cases}$$

Bourke (1995) pointed out that if the weighted average identification algorithms were implemented on-line it would lose its adaptation ability or "fall-asleep" after a huge number of data points have been processed. This is analogous to the determination of the mean of a series of points $x_1, x_2, \ldots, x_N$. As N increases the effect of the latest data x_N on the mean is greatly diminished. Bourke proposed a method to overcome this problem by resetting or normalizing the relational matrix.

4.3.3 Numerical Resolution of Identification Problem

Pedrycz (198?) provided a numerical resolution of fuzzy relational equations for static fuzzy relational equations. It can easily be extended to dynamic fuzzy relational equations. Consider,

$$Y(k) = Y(k-1) \circ U(k-d-1) \circ R \tag{4.14}$$

where $\circ$ denotes the max triangular-norm operator. The triangular-norm operator will be assumed to be the product operator from here on. In terms of membership functions (4.14) can be written as

$$Y_l(k) = \bigvee_{i=1}^{N_{Y1}} \bigvee_{j=1}^{N_U} (Y_i(k-1) \cdot U_j(k-d-1) \cdot R_{ijl}) \tag{4.15}$$

for $l = 1, 2, \ldots, N_{Y2}$ and where Y_i and U_j denotes the i^{th} and j^{th} element of the possibility vector respectively. Then the identification problem is :

$$\min_{R \in F(\mathbf{Y}(k-1) \times \mathbf{U}(k-d-1) \times \mathbf{Y}(k))} J = \sum_{l=1}^{N_{Y2}} [\bigvee_{i=1}^{N_{Y1}} \bigvee_{j=1}^{N_U} (Y_i(k-1) \cdot U_j(k-d-1) \cdot R_{ijl}) - Y_l(k)]^2 \tag{4.16}$$

A necessary condition for a minimum is that $\frac{\partial J}{\partial R} = \mathbf{0}$ where $\mathbf{0}$ is a null matrix of dimension $(N_{Y1} \times N_U \times N_{Y2})$. The Newton method can be used to solve this systems of equations but due to the computational load required, the following update for the relational matrix elements may be preferable.

$$R_{ijl}^{(n+1)} = R_{ijl}^{(n)} - \alpha_{ijl} \cdot \frac{\partial J}{\partial R_{ijl}}$$

where n is the n^{th} iteration step. The derivative $\frac{\partial J}{\partial R_{ijl}}$, $i = 1, 2, \ldots N_{Y1}$, $j = 1, 2, \ldots N_U$, $l = 1, 2, \ldots, N_{Y2}$ is calculated by

$$
\begin{aligned}
\frac{\partial J}{\partial R_{stv}} &= \frac{\partial}{\partial R_{stv}}\{ \sum_{l=1}^{N_{Y2}} [\bigvee_{i=1}^{N_{Y1}} \bigvee_{j=1}^{N_U} (Y_i(k-1)\cdot U_j(k-d-1)\cdot R_{ijl}) - Y_l(k)]^2 \} \\
&= 2\sum_{l=1}^{N_{Y2}} [\bigvee_{i=1}^{N_{Y1}} \bigvee_{j=1}^{N_U} (Y_i(k-1)\cdot U_j(k-d-1)\cdot R_{ijl}) - Y_l(k)]\, P_{stv}
\end{aligned}
$$

where

$$
\begin{aligned}
P_{stv} &= \frac{\partial}{\partial R_{stv}}[\bigvee_{i=1}^{N_{Y1}} \bigvee_{j=1}^{N_U} (Y_i(k-1)\cdot U_j(k-d-1)\cdot R_{ijv})] \\
&= \frac{\partial}{\partial R_{stv}}\{ \bigvee_{i\neq s} \bigvee_{j\neq t} (Y_i(k-1)\cdot U_j(k-d-1)\cdot R_{ijv}) \\
&\qquad \bigvee (Y_s(k-1)\cdot U_t(k-d-1) R_{stv})\} \\
&= \begin{cases} Y_s(k-1)\cdot U_t(k-d-1) & \text{if } \Phi_v \text{ is true} \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

where Φ_v is defined as

$$
(Y_s(k-1)\cdot U_t(k-d-1)\cdot R_{stv}) \geq (\bigvee_{i\neq s} \bigvee_{j\neq t} (Y_i(k-1)\cdot U_j(k-d-1)\cdot R_{ijv})
$$

Although the above analysis was for a first order fuzzy relational equation under max-product composition, it can easily be extended to higher order equations under max triangular-norm composition. Ikoma *et.al.* (1993) proposed a probablistic descent minimization which is suppose to overcome problems arising from convergence to a local minima.

4.3.4 Neuron Inspired Identification Algorithms

Focusing on static fuzzy relational equations Pedrycz (1991) gave a neural network representation and provided a solution under max-min composition. Valente de Oliveira (1993) extended the algorithm to dynamic fuzzy relational equations under max triangular-norm compositional operator. The neural network representation for one basic unit of (4.14) with three referential fuzzy sets in $U(k-d-1)$ and $Y(k-1)$ is shown in Figure (4.4). In Figure (4.4), the output, Y_i denotes the i^{th} element of the output possibility vector. If there are N_{Y2} referential fuzzy sets in $Y(k)$ then the complete neural network representation will consist of N_{Y2} outputs.

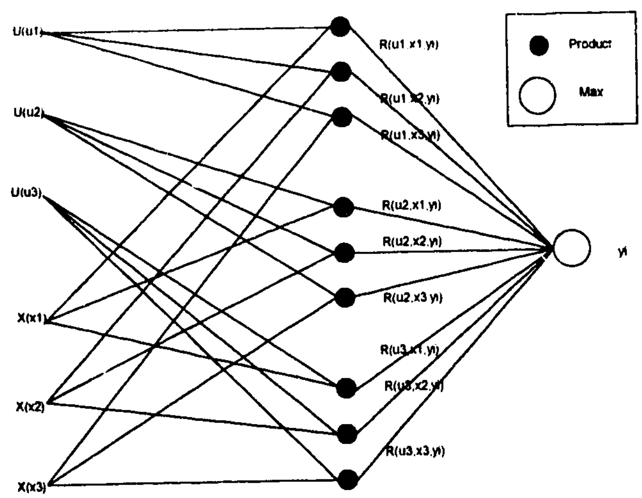

Figure 4.4: Basic Unit of First-Order Fuzzy Relational Equation

A performance index typically used in neural networks is defined as

$$J_i = \sum_{k=1}^{N} (Y_i(k) \equiv \hat{Y}_i(k)) \tag{4.17}$$

where $\equiv$ is the equality index and i denotes the i^{th} referential fuzzy set. Therefore, if c referential fuzzy sets were selected for the output variable then c performance indices will be optimized. The equality index is given the following interpretation:

$$Y_i \equiv \hat{Y}_i = \frac{1}{2}\{ \; [\; (Y_i \to \hat{Y}_i) \wedge (\hat{Y}_i \to Y_i) \;] \; + \; [\; (\bar{Y}_i \to \bar{\hat{Y}}_i \wedge (\bar{\hat{Y}}_i \to \bar{Y}_i) \;] \; \} \qquad (4.18)$$

where $\to$ denotes implication specified between two membership grades. The first part of (4.18) describes the degree of equality between Y_i and $\hat{Y}_i$: it gives the minimum degree to which Y_i implies $\hat{Y}_i$ and vice versa. The second part of (4.18) is similar to the first except that it uses the negation of Y_i and $\hat{Y}_i$. With this definition, the purpose of identification will be to maximize the equality index for each output referential fuzzy set. By choosing a Lukasiewicz implication for (4.18), a relatively simple realization for the equality index is

$$Y_i \equiv \hat{Y}_i = \begin{cases} 1 + Y_i - \hat{Y}_i & \text{if } \hat{Y}_i > Y_i \\ 1 + \hat{Y}_i - Y_i & \text{if } \hat{Y}_i < Y_i \\ 1 & \text{if } Y_i = \hat{Y}_i \end{cases} \qquad (4.19)$$

Applying (4.19) to (4.17) gives

$$\begin{aligned} J_i \;\; = \;\; & \sum_{k:\; \hat{Y}_i(k)=Y_i(k)}^{N} (1) \; + \sum_{k:\; \hat{Y}_i(k)<Y_i(k)}^{N} (1 + \hat{Y}_i(k) - Y_i(k)) \\ & + \sum_{k:\; \hat{Y}_i(k)>Y_i(k)}^{N} (1 + Y_i(k) - \hat{Y}_i(k)) \end{aligned}$$

Now taking the derivative with respect to a generic element of $R(\cdot)$ results in

$$\frac{\partial J_i}{\partial R(\cdot)} = \sum_{k:\ \hat{Y}_i(k) < Y_i(k)}^{N} \frac{\partial \hat{Y}_i(k)}{R(\cdot)} - \sum_{k:\ \hat{Y}_i(k) > Y_i(k)}^{N} \frac{\partial \hat{Y}_i(k)}{R(\cdot)} \tag{4.20}$$

With this expression, a gradient based learning method can then be used in learning the parameters in the relational matrix. Valente de Oliveira (1993) extended this algorithm to dynamic fuzzy relational equations under max triangular-norm compositional operator.

4.4 Summary of Identification Algorithms

Identification algorithms as presented in the previous section can be divided into two major groups.

- Those based on the Linguistic Approach
- Those based on the Numerical Approach

The Numerical approaches minimize or maximize an objective function to calculate the parameters in the relational matrix. On the other hand, the linguistic approach builds the relational matrix by applying one possible solution derived from the analytical resolution of fuzzy relational equation. Therefore, numerical schemes offer the flexibility of optimizing the relational matrix based on any desired criteria.

Bourke & Fisher (1995) compared the performance of various fuzzy relational identification algorithms. It was observed that based on the sum of squares of the prediction errors, the numerical schemes gave slightly better results. However, the drawback of numerical schemes appear to be the difficulties in selecting the tuning parameters. Most importantly, these types of algorithms may not be practical for on-line implementation since they process information in a batch fashion. Therefore, if a new numerical on-line fuzzy relational identification algorithm is proposed, the following issues must be addressed :

- Practicality as an on-line implementation
- Convergence

Bibliography

[1] M. Bourke. *"Self-Learning Predictive Control Using Relational-Based Fuzzy Logic"*. PhD thesis, University of Alberta, 1995.

[2] M.M. Bourke and D.G. Fisher. A Comparative Analysis of Identification Algorithms for Fuzzy Relational Matrices. *Fuzzy Sets and Systems*, submitted 1995, 1995.

[3] J.Q. Chen, J.H. Lu, and L.J. Chen. "An On-Line Identification Algorithm for Fuzzy Systems". *Fuzzy Sets and Systems*, 64:63–72, 1994.

[4] E. Czogala and W. Pedrycz. "On Identification in Fuzzy Systems and its Applications in Control Problems". *Fuzzy Sets and Systems*, 6:73–83, 1981.

[5] J. Valente de Oliveira. "Neuron Inspired Learning Rules for Fuzzy Relational Structures". *Fuzzy Sets and Systems*, 57:41–53, 1993.

[6] J. Valente de Oliveira. "On Optimal Fuzzy Systems I/O Interfaces". In *Proc. 2nd. Int. Conf. on Fuzzy Systems*, volume 2, pages 851–856, 1993.

[7] N. Ikoma, W. Pedrycz, and K. Hirota. "Estimation of Fuzzy Relational Matrix by Using Probablistic Descent Method". *Fuzzy Sets and Systems*, 57:335–349, 1993.

[8] I.S. Shaw and J.J. Krüger. "New Fuzzy Learning Model with Recursive Estimation for Dynamic System". *Fuzzy Sets and Systems*, 48:21[illegible]–229, 1992.

[9] M. Mizumoto. "Improvement Methods of Fuzzy Controls". In *Proceedings of 3rd. IFSA*, pages 60–62, Seattle, WA, Aug 1989.

[10] W. Pedrycz. "Numerical and Applicational Aspects of Fuzzy Relational Equations". *Fuzzy Sets and Systems*, 11:1–18, 1983.

[11] W. Pedrycz. "An Identification Algorithm in Fuzzy Relational Systems". *Fuzzy Sets and Systems*, 13:153–167, 1984.

[12] W. Pedrycz. "Neurocomputations in Relational Systems". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(3):289–296, March 1991.

[13] W. Pedrycz. "Why Triangular Membership Functions?". *Fuzzy Sets and Systems*, 64:21–30, 1994.

[14] C.W. Xu and Y.Z. Lu. "Fuzzy Model Identification and Self-Learning for Dynamic Systems". *IEEE Trans. of Systems, Man and Cybernetics*, 17:683–689, 1987.

[15] Hans J. Zimmermann. *Fuzzy Set Theory - and Its Applications*. Kluwer Nijhoff Publishing, 1985.

Chapter 5

Fuzzy Predictive Identification

5.1 Introduction

The availability of a systematic fuzzy relational identification procedure has resulted in the development of predictive controllers which utilize the fuzzy relational equation. Graham & Newell (1988), Postlethwaite (1994), Valente de Oliveira & Lemos (1995) and Bourke & Fisher (1996) have proposed the use of fuzzy predictive controllers. Obviously, the performance of such controllers depends on the identification algorithm used. Bourke (1995) compared the performance of various identification algorithms based on the Box-Jenkins gas furnace data set. Clearly, the fuzzy identification algorithm has an impact on the quality of the model.

In this chapter, two fuzzy relational identification algorithms namely the Neuron-Based Predicitive Identification (NBPI) algorithm and the Fuzzy Relational Predictive Identification (FRPI) algorithm will be presented. The distinguishing feature of the proposed algorithms from previous fuzzy relational identification algorithms is the optimization of the model parameters so that the prediction error is minimized over a user-defined prediction horizon. The minimization of the prediction error over a prediction horizon is an attempt at improving the predictive capability of fuzzy relational models. Finally, the effectiveness of these new algorithms will be demonstrated by a comparison with a popular identification scheme.

5.2 Why Fuzzy Predictive Identification?

In conventional control, the concept of control-relevant identification was proposed by Shook *et al.* (1991, 1992). The authors proposed an on-line identification strategy for the Generalized Predictive Controller (GPC). The same idea will be used but this time from the point of view of fuzzy predictive controllers. The following discussion will concentrate on fuzzy predictive controllers as proposed by Postlethwaite (1994) and Bourke & Fisher (1996). An important point to note about fuzzy predictive controllers is that they utilize some form of numerical search to obtain the control action that will minimize the predicted future error(s). For this numerical search to work, the fuzzy relational equation must provide a prediction of the output. Therefore, the predictive capability of the relational model is of key importance.

Postlethwaite (1994) proposed a controller which has the same objective function as conventional Long Range Predictive Controllers (LRPC). The objective function of the controller is then

$$J = \sum_{j=N_1}^{N_2} (y_{sp}(k+j) - \hat{y}(k+j))^2 + \lambda u(k)^2 \tag{5.1}$$

However, Postlethwaite restricted N_2 to be equal to $(N_1 + 1)$.

The controller by Bourke & Fisher (1996) uses two different models - a dynamic model and a steady state model. The dynamic model is used to obtain a dead beat control action. Therefore, the objective is simply the first term of (5.1) where $N_2 = d + 1$. For these controllers a $(d + 1)$ step ahead prediction must be made where $(d + 1)$ corresponds to the total time-delay of the process. For a process described by a first order relational equation which is expressed as :

$$Y(k) = Y(k-1) \circ U(k-d-1) \circ R \tag{5.2}$$

where $\circ$ represents the max triangular norm operator. The one-step ahead prediction can be found via :

$$Y(k+1) = Y(k) \circ U(k-d) \circ R \tag{5.3}$$

To obtain a multi-step ahead prediction, $Y(k+1)$ must be defuzzified and fuzzified again before being fed back into (5.2). This is necessary to alleviate leakage effects which results in a distortion of the states which are used for the next prediction. Leakage effects occur due to the fact that the fuzzification and defuzzification operations are not exact inverses.

Definition 5.1 *Define $\wp$ as the fuzzification operator which is the* 1 *to N_{Y2} mapping whereby a crisp value is converted into a possibility vector with N_{Y2} elements.*

Definition 5.2 *Define $\wp^{-1}$ as the defuzzification operator which is the N_{Y2} to* 1 *mapping whereby a possibility vector is coverted into a crisp value.*

Then the next prediction is given by

$$\hat{Y}(k+2) = \tilde{\hat{Y}}(k+1) \circ U(k-d-1) \circ R \tag{5.4}$$

where $\tilde{\hat{Y}}(k+1) = \phi\phi^{-1}(\hat{Y}(k+1))$. By applying the same procedure more predictions can be made. It is obvious that even for the minimization of (5.1) a multi-step prediction may be required.

All existing identification algorithms attempt to obtain a good single step prediction. For example, linguistic identification methods such as those by Pedrycz (1984), Xu (1987), Shaw & Krüger (1992), Chen (1994) and Bourke (1995) would learn a first order fuzzy relational equation by processing fuzzy data $Y(k), Y(k-1)$ and $U(k-d-1)$ for $k = d+2,\ d+3,\ \ldots, N$. No effort is made to optimize the relational matrix over some prediction horizon. The same can be said about numerical identification algorithms such as those by Pedrycz (1983, 1991) , Valente de Oliveira (1993) , and Ikoma *et.al.* (1993) which estimates the relational matrix based on some objective function. The objective function used by Pedrycz (1983) and Ikoma *et.al.* (1993) has the following form.

$$J = \frac{1}{2} \sum_{k=d+1}^{N} [\ Y_i(k+1) - \hat{Y}_i(k+1|k)\ ]^2 \tag{5.5}$$

for $i = 1, 2, \ldots, N_{Y2}$ where N_{Y2} is the number of referential fuzzy sets in Y. The neuron-based schemes by Pedrycz (1991) and Valente de Oliveira (1993) maximizes an equality index and the objective function is expressed as

$$J_i = \sum_{k=d+1}^{N} [\ Y_i(k+1) \equiv \hat{Y}_i(k+1|k)\ ] \tag{5.6}$$

for $i = 1, 2, \ldots, N_{Y2}$ and $\equiv$ denotes the equality index. A common feature in equations (5.5) and (5.6) is that the model parameters are that which gives a good one-step ahead prediction. A good single-step prediction does not guarantee that the multi-step prediction is good. Hence, none of the above mentioned schemes is really designed for use with a fuzzy predictive controller. For algorithms which maximize or minimize an objective function, a new objective function which considers a multi-step ahead prediction must be included.

In general, all fuzzy relational predictive controllers to date have a control objective which is a subset of the following objective function :

$$J_{FPC} = \sum_{j=N_1}^{N_2} [\ y_{sp}(k+j) - y(k+j)\ ]^2 \tag{5.7}$$

For Bourke (1995) and Postlethwaite (1994), N_2 is equal to $d+1$ and $d+2$ respectively. For the Fuzzy Relational Long Range Predictive Controller (FRLRPC), N_1 and N_2 would correspond to the prediction horizon. Following the development of the LRPI for conventional Generalized Predictive Controllers (Shook, 1991), (5.7) can be written as :

$$J_{FPC} = \sum_{j=N_1}^{N_2} [\ y_{sp}(k+j) - \hat{y}(k+j|k)\ ]^2 + \sum_{j=N_1}^{N_2} [\ y(k+j) - \hat{y}(k+j|k)\ ]^2$$

$$- 2 \sum_{j=N_1}^{N_2} [\, y_{sp}(k+j) - \hat{y}(k+j|k) \,] \, [y(k+j) - \hat{y}(k+j|k)] \quad (5.8)$$

The first and second term in (5.8) corresponds to the objective function for the control and identification algorithm respectively. So the optimal identification method for the control objective must provide the model which predicts best over N_2 steps. The identification objective is then given by

$$J_{ID} = \sum_{j=N_1}^{N_2} [\, y(k+j) - \hat{y}(k+j|k) \,]^2 \quad (5.9)$$

Since the model in a fuzzy predictive controller deals with fuzzy values in tead of crisp measurements, (5.9) must be modified to

$$J_{FPID,i} = \sum_{j=N_1}^{N_2} [\, Y_i(k+j) - \hat{Y}_i(k+j|k) \,]^2 \quad (5.10)$$

for $i = 1, 2, \ldots, N_{Y2}$ where N_{Y2} is the number of referential fuzzy sets in $Y(k)$.

5.3 A Neuron Based Predictive Identification Algorithm (NBPI)

The neuron based identification algorithm as presented by Pedrycz (1991) and Valente de Oliveira (1993) involves the maximization of the following performance index

$$J_i = \sum_{k=d+1}^{N} [Y_i(k+1) \equiv \hat{Y}_i(k+1|k) \,] \quad (5.11)$$

for $i = 1, 2, \ldots, N_{Y2}$ where i dentoes the i^{th} referential fuzzy set and $\equiv$ indicates the equality index. The interpretation for the equality index is given by

$$(Y \equiv \hat{Y}) = \frac{1}{2}\left\{[(Y \rightarrow \hat{Y}) \wedge (\hat{Y} \rightarrow Y)] + [(\bar{Y} \rightarrow \bar{\hat{Y}}) \wedge (\bar{\hat{Y}} \rightarrow \bar{Y}]\right\} \tag{5.12}$$

where $\rightarrow$ is the implication specified between two membership grades and $^{-}$ represents the negation. This equality index is a measure of the similarity between two membership grades and hence, in fuzzy relational identification, it is desired that the equality index be as high as possible at any time for all elements in the output possibility vector. By selecting the implication to be represented by the Lukasiewitcz implication, (5.12) is reduced to

$$(Y \equiv \hat{Y}) = \begin{cases} 1 + Y - \hat{Y} & \text{if } \hat{Y} > Y \\ 1 + \hat{Y} - Y & \text{if } \hat{Y} < Y \\ 1 & \text{if } Y = \hat{Y} \end{cases} \tag{5.13}$$

To incorporate a long range predictive capability into this algorithm, 5.11 can be reformulated to

$$J_i = \sum_{k=d+1}^{N-N_2} \sum_{j=1}^{N_2} [Y_i(k+j) \equiv \hat{Y}_i(k+j|k)] \tag{5.14}$$

Rewrite (5.14) as

$$J_i = \sum_{k,j} \sum_{: \hat{Y}_i < Y_i} [\, Y_i(k+j) \equiv \hat{Y}_i(k+j|k) \,] + \sum_{k,j} \sum_{: \hat{Y}_i > Y_i} [\, Y_i(k+j) \equiv \hat{Y}_i(k+j|k) \,]$$
$$+ \sum_{k,j} \sum_{: \hat{Y}_i = Y_i} [\, Y_i(k+j) \equiv \hat{Y}_i(k+j|k) \,] \qquad (5.15)$$

Then applying (5.13) to (5.15) and taking the derivative with respect to a generic element in R, denoted by $R(\cdot)$ yields

$$\frac{\partial J_i}{\partial R(\cdot)} = \sum_{k,j} \sum_{: \hat{Y}_i < Y_i} \frac{\partial \hat{Y}_i(k+j|k)}{\partial R(\cdot)} - \sum_{k,j} \sum_{: \hat{Y}_i > Y_i} \frac{\partial \hat{Y}_i(k+j|k)}{\partial R(\cdot)} \qquad (5.16)$$

where $\hat{Y}_i(k+j|k)$ for a first-order relational equation is found by :

$$\hat{Y}_i(k+j|k) = [\, \bigvee_{l_1=1}^{N_u} \bigvee_{l_2=1}^{N_Y} (U_{l_1}(k+j-d-1) \; t \; \hat{Y}_{l_2}(k+j-1|k) \; t \; R_{l_1 l_2 i}) \,] \vee \nu_i$$

where ν_i is the bias term and t is a triangular-norm operator. The bias term is incorporated to ensure that each $Y_i(k)$ produces significant output. Details of calculating the predicted output were discussed in the previous section. Therefore,

$$\frac{\partial \hat{Y}_i(k+j|k)}{\partial R(\cdot)} = \frac{\partial \hat{Y}_i(k+j|k)}{\partial R_{j_1 j_2 j}}$$
$$= \begin{cases} \frac{\partial}{\partial R_{j_1 j_2 j}} [U_{j_1}(k+j-d-1) \; t \; \hat{Y}_{j_2}(k+j-1|k) \; t \; R_{j_1 j_2 j}] \vee \nu_i & \text{if } \Phi_i \text{ is true} \\ 0 & \text{Otherwise} \end{cases}$$

where

$$\Phi_i = \bigvee_{l_1=1}^{N_U} \bigvee_{l_2=1}^{N_{Y1}} [\, U_{l_1}(k+j-d-1) \; t \; \hat{Y}_{l_2}(k+j-1) \; t \; R_{l_1 l_2 i} \,] \geq \nu_i$$

and

$$\bigvee_{l_1 \neq j_1} \bigvee_{l_2 \neq j_2} [\, U_{l_1}(k+j-d-1) \; t \; \hat{Y}_{l_2}(k+j-1) \; t \; R_{l_1 l_2 i} \,] \leq$$
$$[\, U_{j_1}(k+j-d-1) \; t \; \hat{Y}_{j_2}(k+j-1) \; t \; R_{j_1 j_2 i} \,]$$

The gradient of J_i with respect to the bias terms, ν_i must also be calculated. Taking the derivative with respect to ν_i in (5.15) yields :

$$\frac{\partial J_i}{\partial \nu_i} = \sum_{k,j \,:\, \hat{Y}_i < Y_i} \sum \frac{\partial \hat{Y}_i(k+j|k)}{\partial \nu_i} - \sum_{k,j \,:\, \hat{Y}_i > Y_i} \sum \frac{\partial \hat{Y}_i(k+j|k)}{\partial \nu_i}$$

where

$$\frac{\partial \hat{Y}_i(k+j|k)}{\partial \nu_i} = \begin{cases} 1 & \text{if } \Gamma_i \text{ is true} \\ 0 & \text{Otherwise} \end{cases}$$

where Γ_i is defined as :

$$\bigvee_{l_1=1}^{N_U} \bigvee_{l_2=1}^{N_{Y1}} [U_{l_1}(k+j-d-1) \; t \; \hat{Y}_{l_2}(k+j-d-1) \; t \; R_{l_1 l_2 i} \leq \nu_i$$

Model parameters are updated by

$$R^{p+1}_{j_1 j_2 j} = R^{p}_{j_1 j_2 j} + \eta_{j_1 j_2 j} \cdot \; [\, \frac{\partial J^{p+1}}{\partial R_{j_1, j_2, j}} + \alpha_{j_1 j_2 j} \frac{\partial J^{p}}{\partial R_{j_1 j_2 j}} \,] \qquad (5.17)$$

and,

$$\nu_i^{p+1} = \nu_i^p + \eta_i \cdot \left[\frac{\partial J^{p+1}}{\partial \nu_i} + \alpha_i \frac{\partial J^p}{\partial \nu_i} \right] \tag{5.18}$$

where $R_{j_1 j_2 j}$ indicates the $j_1 j_2 j^{th}$ entry of the relational matrix and p is the iteration number, η is the learning rate and α is the momentum term.

Although convergence difficulties have been citied as a drawback of numerical identification algorithms such as this one, this problem can be quite easily remedied. A linguistic fuzzy identification algorithm can be used to obtain an initial estimate of the relational matrix and conservative tuning parameters can be used. This usually results in significantly faster convergence. The simple yet effective linguistic algorithm by Pedrycz (1984) is recommended.

However, the computational requirements for this algorithm are still quite high and may not be a good candidate in an on-line implementation. A computationally cheaper algorithm is sought.

5.4 Fuzzy Relational Predictive Identification (FRPI)

With the objective of identifying a model which gives optimal predictions over some prediction horizon for GPC, Shook *et al.* (1991, 1992) proposed the following objective function :

$$J_{LRPI} = \frac{1}{N - N_2} \sum_{k=1}^{N-N_2} \frac{1}{N_p} \sum_{j=N_1}^{N_2} [\, y(k+j) - \hat{y}(k+j|k)\,]^2 \tag{5.19}$$

where $N_p = N_2 - N_1 + 1$. N_1 and N_2 is the prediction horizon. In the fuzzy domain, $y(k)$ is represented by its possibility vector. Hence, (5.19) becomes

$$J_{FRPI} = \frac{1}{N - N_2} \sum_{k=1}^{N-N_2} \frac{1}{N_p} \sum_{j=N_1}^{N_2} \sum_{i=1}^{N_{Y2}} [\, Y_i(k+j) - \hat{Y}_i(k+j|k) \,]^2 \tag{5.20}$$

Here $\hat{Y}_i$ is the predicted membership grade for the i^{th} element of the possibility vector. Rearranging (5.20) yields,

$$J_{FRPI} = \sum_{i=1}^{N_{Y2}} \frac{1}{N_p \cdot (N - N_2)} \sum_{k=1}^{N-N_2} \sum_{j=N_1}^{N_2} [\, Y_i(k+j) - \hat{Y}_i(k+j|k) \,]^2 \tag{5.21}$$

Then,

$$J_{FRPI} = \sum_{i=1}^{N_{Y2}} J_{FRPI,i}$$

where

$$J_{FRPI,i} = \frac{1}{N_p \cdot (N - N_2)} \sum_{k=1}^{N-N_2} \sum_{j=N_1}^{N_2} [\, Y_i(k+j) - \hat{Y}_i(k+j|k) \,]^2 \tag{5.22}$$

Minimization of J_{FRPI} involves the minimization of each $J_{FRPI,i}$ for $i = 1, 2, \ldots, N_{Y2}$.

Let

$$e_i(k+j) = Y_i(k+j) - \hat{Y}_i(k+j|k) \tag{5.23}$$

for $j = N_1, N_1 + 1, \ldots, N_2$. And further define

$$\varepsilon_i(k) = \begin{bmatrix} e_i(k + N_1) \\ e_i(k + N_1 + 1) \\ \vdots \\ e_i(k + N_2) \end{bmatrix} \tag{5.24}$$

Then (5.22) can be rewritten as

$$J_{FRPI,i} = \frac{1}{N_p \cdot (N - N_2)} \sum_{k=1}^{N-N_2} \varepsilon_i^T(k)\, \varepsilon_i(k) \tag{5.25}$$

A recursive prediction error identification method can then be applied on (5.25) to obtain the model parameters. Details of such a method can be found in the literature, for example, see Söderström & Stoica (1989). Further denoting

$$\begin{aligned} \psi_i(k) &= -\left[\frac{\partial \varepsilon_i(k)}{\partial \underline{\theta_i}}\right]^T \\ &= -\begin{bmatrix} \frac{\partial e_i(k+N_1)}{\partial \theta_{1,i}} & \frac{\partial e_i(k+N_1+1)}{\partial \theta_{1,i}} & \cdots & \frac{\partial e_i(k+N_2)}{\partial \theta_{1,i}} \\ \frac{\partial e_i(k+N_1)}{\partial \theta_{2,i}} & \frac{\partial e_i(k+N_1+1)}{\partial \theta_{2,i}} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_i(k+N_1)}{\partial \theta_{n,i}} & \frac{\partial e_i(k+N_1+1)}{\partial \theta_{2,i}} & \cdots & \frac{\partial e_i(k+N_2)}{\partial \theta_{n,i}} \end{bmatrix} \end{aligned} \tag{5.26}$$

$\theta_{1,i}, \theta_{2,i}, \ldots, \theta_{n,i}$ are the relational matrix elements which gives the i^{th} membership grade for $\hat{Y}$ *ie.* $\hat{Y}_i$. For example, in a first order fuzzy relational equation as given by

$$Y(k) = Y(k-1) \circ U(k-d-1) \circ R$$

with N_U, N_{Y1}, N_{Y2} referential fuzzy sets in $U(k-d-1), Y(k-1)$ and $Y(k)$, respectively, R would be a $N_U \times N_{Y1} \times N_{Y2}$ matrix. The possibility vector for $Y(k)$ is found via

$$Y_i(k) = \bigvee_{i_1=1}^{N_U} \bigvee_{i_2=1}^{N_{Y1}} [\, U_{i_1}(k-d-1) \; t \; Y_{i_2}(k-1) \; t \; R_{i_1 i_2 i} \,] \tag{5.27}$$

for $i = 1, 2, \ldots, N_{Y2}$. Then $R_{i_1 i_2 i}$ is now a $N_U \times N_{Y1}$ matrix in the following form :

$$R = \begin{bmatrix} R_{11,i} & R_{12,i} & \cdots & R_{iN_{Y1},i} \\ R_{21,i} & R_{22,i} & \cdots & R_{2N_{Y1},i} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N_U 1,i} & R_{N_U 2,i} & \cdots & R_{N_U N_{Y1},i} \end{bmatrix}$$

$$= \begin{bmatrix} \theta_{1,i} & \theta_{2,i} & \cdots & \theta_{N_{Y1},i} \\ \theta_{N_{Y1}+1,i} & \theta_{N_{Y1}+2,i} & \cdots & \theta_{2N_{Y1},i} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{N_{Y1}(N_U-1)+1,i} & \theta_{N_{Y1}(N_U-1)+2,i} & \cdots & \theta_{N_U * N_{Y1},i} \end{bmatrix}$$

n as defined in (5.26) is $N_U \times N_{Y1}$. The elements in (5.26) are found as follows :

$$\frac{\partial e_i(k+j)}{\partial \theta_i(\cdot)} = \frac{\partial e_i}{\partial R_{pqi}}$$

$$
\begin{aligned}
&= -\frac{\partial \hat{Y}_i(k+j|k)}{\partial R_{pqi}} \\
&= -\frac{\partial}{\partial R_{pqi}}\Big[\big(\bigvee_{i_1 \neq p} \bigvee_{i_2 \neq q} U_{i_1}(k-d-1+j) \circ Y_{i_2}(k-1) \circ R_{i_1 i_2 i}\big) \\
&\quad \bigvee \big(U_p(k-d-1+j) \circ Y_q(k-1) \circ R_{pqi}\big) \Big]
\end{aligned}
$$

If the triangular-norm operator in (5.27) is the product, then

$$
\frac{\partial e_i(k+j)}{\partial \theta_i(\cdot)} = \begin{cases} U_p(k-d-1+j) \cdot Y_q(k-1) & \text{if } \Omega_i \text{ is true.} \\ 0 & \text{if } \Upsilon_i \text{ is true} \end{cases} \tag{5.28}
$$

where Ω_i is defined as

$$
(U_p(k-d-1+j) \cdot Y_q(k-1) \cdot R_{pqi}) > (\bigvee_{i_1 \neq p} \bigvee_{i_2 \neq q} U_{i_1}(k-d-1+j) \cdot Y_{i_2}(k-1+j) \cdot R_{i_1 i_2 i})
$$

and Υ_i is defined as

$$
(U_p(k-d-1+j) \cdot Y_q(k-1) \cdot R_{pqi}) < (\bigvee_{i_1 \neq p} \bigvee_{i_2 \neq q} U_{i_1}(k-d-1+j) \cdot Y_{i_2}(k-1+j) \cdot R_{i_1 i_2 i})
$$

The above analysis applies to other triangular-norm operators and to higher order fuzzy relational equations. It is important to note that for $\frac{\partial e_i(k+j)}{\partial \theta_i(\cdot)}$ there is usually only one non-zero value. Equation (5.28) is undefined if

$$
(U_p(k-d-1+j) \cdot Y_q(k-1+j) \cdot R_{pqi}) = (\bigvee_{i_1 \neq p} \bigvee_{i_2 \neq q} U_{i_1}(k-d-1+j) \cdot Y_{i_2}(k-1+j) \cdot R_{i_1 i_2 i}).
$$

To overcome this problem, one approach is to approximate the maximum function

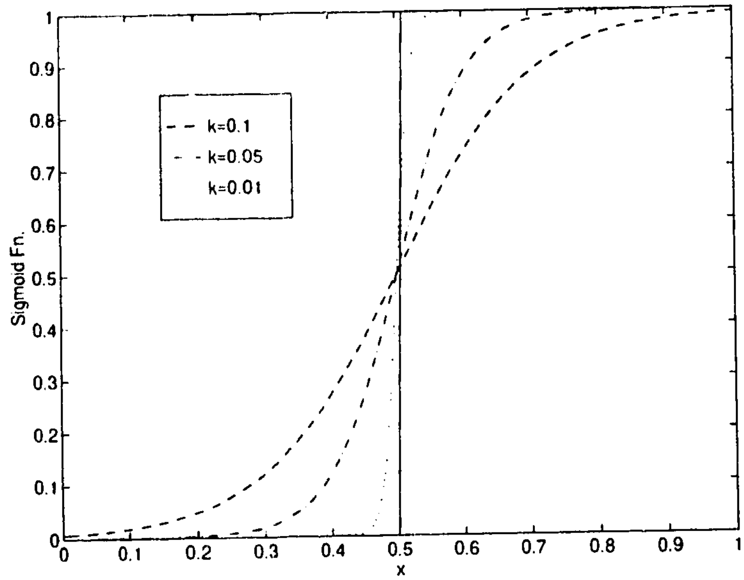

Figure 5.1: Effect of k on Sigmoid Function

by some smooth function. Following in the approach by Ikoma *et al.* (1993), the max function is replaced by the sigmoid function. Equation (5.28) becomes

$$\frac{\partial e_i(k+j)}{\partial \theta_i(\cdot)} = \begin{cases} U_p(k-d-1+j)\cdot Y_q(k-1)\cdot \dfrac{1}{1+e^{-\frac{(x-a)}{k}}} & \text{if } \Omega_i \text{ is true.} \\ 0 & \text{if } \Upsilon_i \text{ is true} \end{cases} \tag{5.29}$$

where

$$x = U_p(k-d-1+j)\cdot Y_q(k-1+j)\cdot R_{pqi}$$

$$a = \bigvee_{i_1\neq p}\ \bigvee_{i_2\neq q} U_{i_1}(k-d-1+j)\cdot Y_{i_2}(k-1+j)\cdot R_{i_1 i_2 i}$$

k is used to control the shape of the function. The effect of k on the shape of the sigmoid function is shown in Figure (5.1).

The update equations for the recursive prediction error method can then be used to obtain new estimates of the relational matrix elements. The update equations as given by Söderström & Stoica (1989) are

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K_i(k) \cdot \varepsilon_i(k)$$

$$K_i(k) = P_i(k-1) \cdot \psi_i(k) \left[\mathbf{I} + \psi_i^T(k) \cdot P_i(k-1) \cdot \psi_i(k) \right]^{-1}$$

$$P_i(k) = P_i(k-1) - K_i(k) \cdot \psi_i^T(k) \cdot P_i(k-1) \qquad (5.30)$$

Summary of Algorithm :

For the i^{th} element in the output possibility vector

1. Calculate $e_i(k+j)$ according to (5.23).

2. Determine $\dfrac{\partial e_i(k+j)}{\partial R_{i_1 i_2 i}}$ via (5.29) for $i_1 = 1, 2, \ldots, N_U$ and $i_2 = 1, 2, \ldots, N_{Y1}$.

3. Form ψ_i which is defined by (5.26).

4. Update elements in R according to (5.30)

5.5 Results

To evaluate the proposed identification algorithms namely the NBPI and FRPI algorithms, two data sets were used which are the Box-Jenkins gas furnace data set and the simulated pH neutralization data set of Hall & Seborg (1989).

5.5.1 Box-Jenkins Gas Furnace Data Set

The Box-Jenkins gas furnace data set is often used as a benchmark in fuzzy relational identification. This data set consists of one input and one output. The

Table 5.1: Mean Prediction Error For Box-Jenkins Data

No. of Prediction Steps	$J_{2,NBPI}$	$J_{2,FRPI}$	$J_{2,Shaw}$
1	0.2720	0.2607	0.4528
3	0.4953	0.4960	0.7867

input is the gas flowrate into the furnace while the output is the CO_2 concentration of the exit gas. Single step and three steps ahead predictions were made. A measure of the quality of the model is determined by computing the mean squared prediction error which is given by :

$$J_2 = \frac{1}{N-d-1} \sum_{k=d+2}^{N} [\ y(k) - \hat{y}(k)\ ]^2$$

As a comparison basis, the algorithm of Shaw & Krüger (1992) was also used to build the relational matrix, and the one and three steps ahead predictions are made. Figure (5.2) and Figure (5.3) shows the single step prediction of the NBPI and FRPI respectively. The three steps ahead prediction using NBPI and FRPI are shown in Figure (5.4) and Figure (5.5) respectively. Also included in all the figures is the output predicted using Shaw's scheme. The computed mean squared prediction error for all three methods are summarized in Table (5.1). The two proposed algorithms give better results than Shaw's algorithm even for a single step prediction. As expected when the number of prediction steps is increased, the accuracy of the prediction deteriorates (Pedrycz, 1993) but the proposed algorithms continue to outperform Shaw's algorithm.

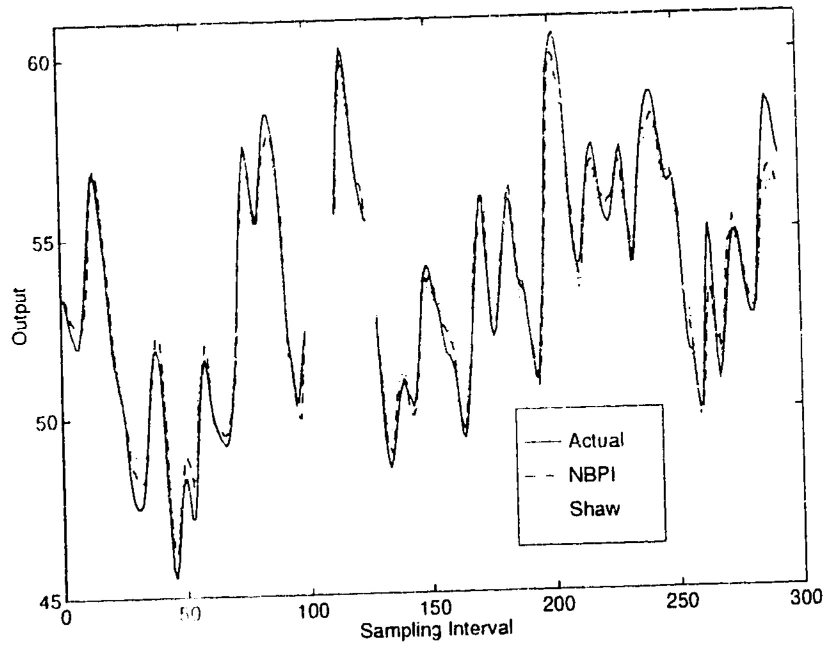

Figure 5.2: Single Step Prediction of NBPI for Box-Jenkins Data Set

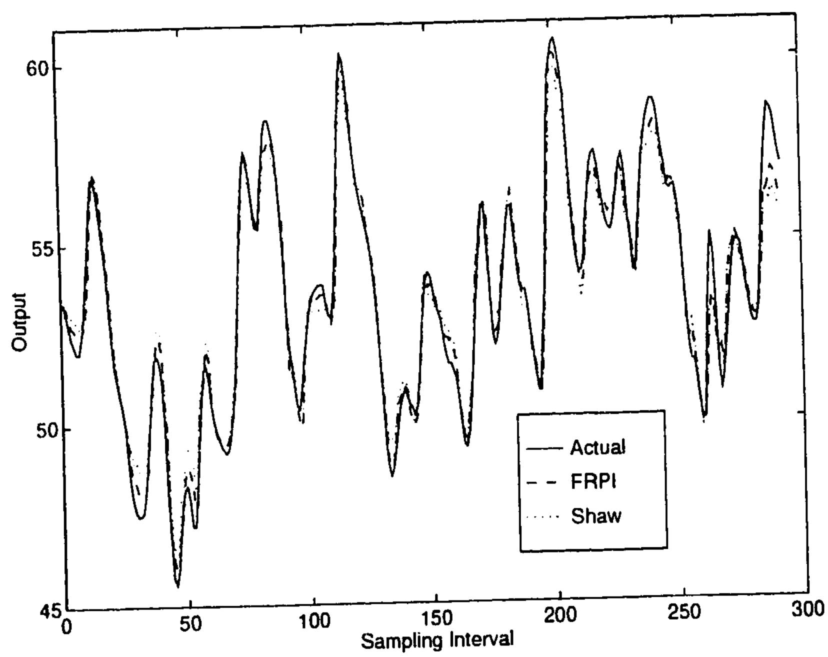

Figure 5.3: Single Step Prediction of FRPI for Box-Jenkins Data Set

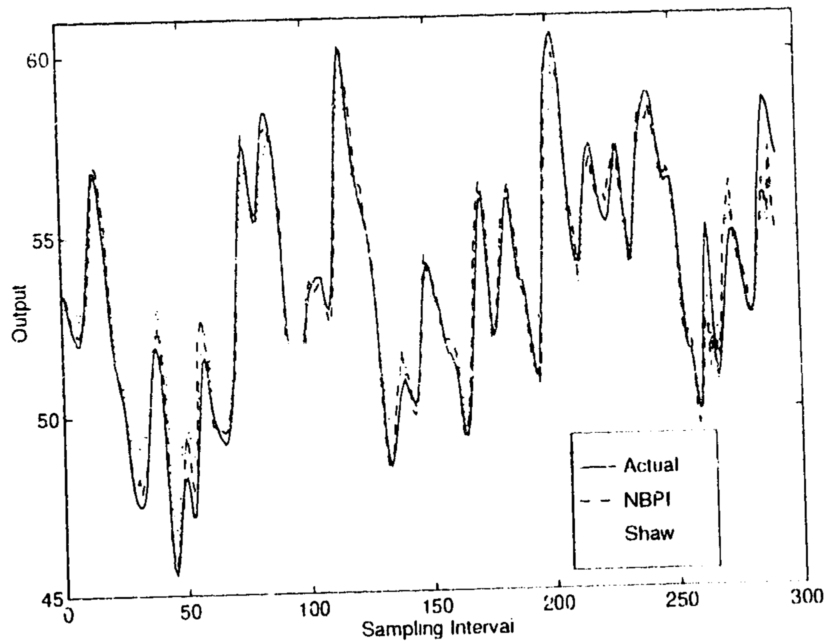

Figure 5.4: Three Steps Prediction of NBPI for Box-Jenkins Data Set

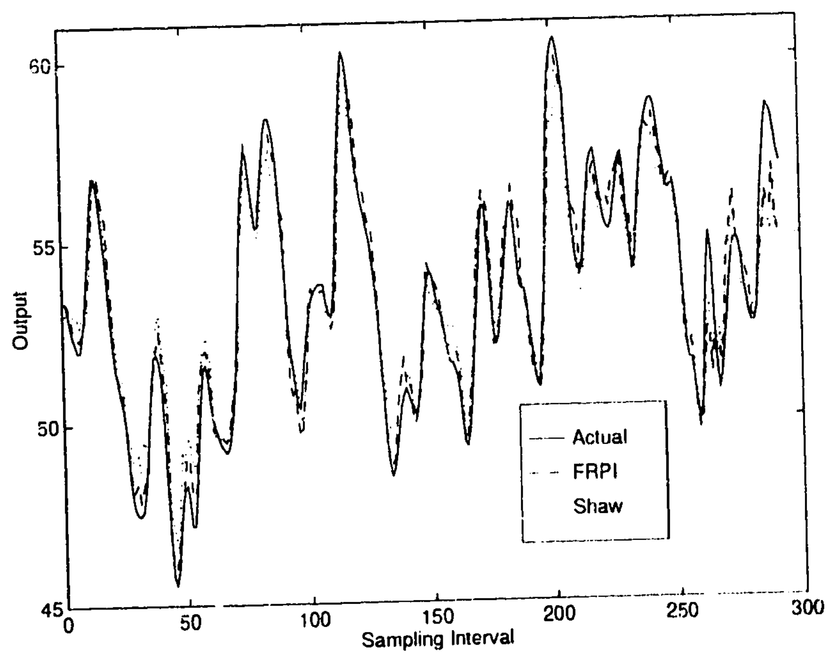

Figure 5.5: Three Steps Prediction of FRPI for Box-Jenkins Data Set

Table 5.2: Mean Prediction Error for Simulated pH Process

No. of Prediction Steps	$J_{2,NBPI}$	$J_{2,FRPI}$	$J_{2,Shaw}$
4	0.0055	0.0028	0.0049

5.5.2 pH Neutralization Data Set

The pH neutralization process offers a challenge due to its highly non-linear behaviour. The control of pH is of great interest as it is encountered in many chemical processes. The simulated neutralization process is depicted in Figure (5.6). The base and buffer flowrates were kept constant which reduces the identification problem to the SISO case. The output is the pH and the input is the acid flowrate into the stirred vessel. A four steps ahead prediction was made and the results are shown in Figure (5.7). Table (5.2) shows the mean squared prediction error for the three methods.

5.6 Convergence Properties

Convergence difficulties arise due to a poor choice of tuning parameters and/or a poor initial estimate of the relational matrix. Therefore, a good identification algorithm should give good predictions and at the same time be relatively robust. Figure (5.8) and Figure (5.9) shows the mean prediction error versus the iteration number of the NBPI and FRPI method, optimized over a prediction horizon of one with different initial guesses for R. The FRPI approach converges much quicker than the NBPI method. For batch identification especially for a very lengthy data

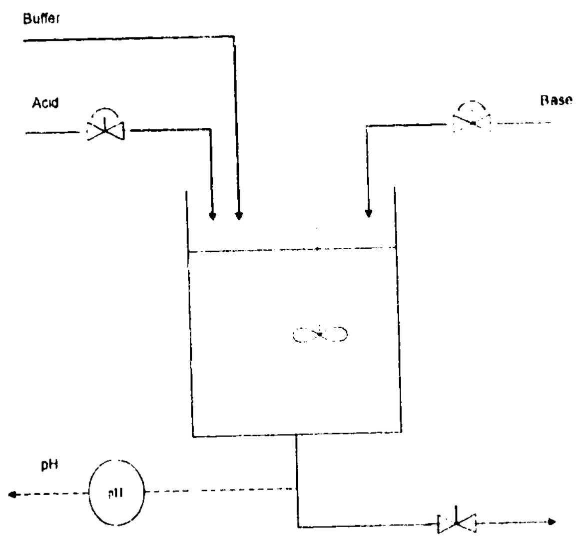

Figure 5.6: Schematic Diagram of Simulated pH Neutralization Process

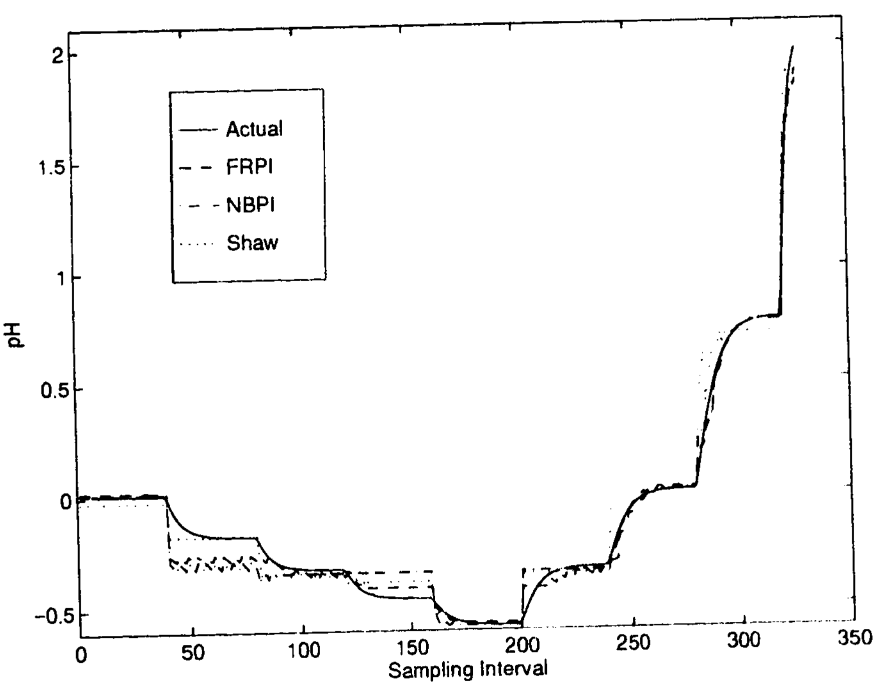

Figure 5.7: Four Steps Prediction of pH Neutralization Process

set, the ease of convergence can result in a substantial reduction in computational time.

5.7 Conclusions

Based on the identification results from the previous sections, it can be concluded that the FRPI approach is indeed an effective fuzzy relational identification technique. Its advantages include

- The prediction error is minimized over a prediction horizon
- It is a practical on-line scheme due to its recursive formulation
- It offers rapid convergence and is relatively insensitive to initial guesses of the relational matrix.

Although the NBPI technique gives a comparable performance to the FRPI, its utility is somewhat limited as it processes data in a batch sense. It is also more sensitive to the initial guess of the relational matrix and convergence is slower.

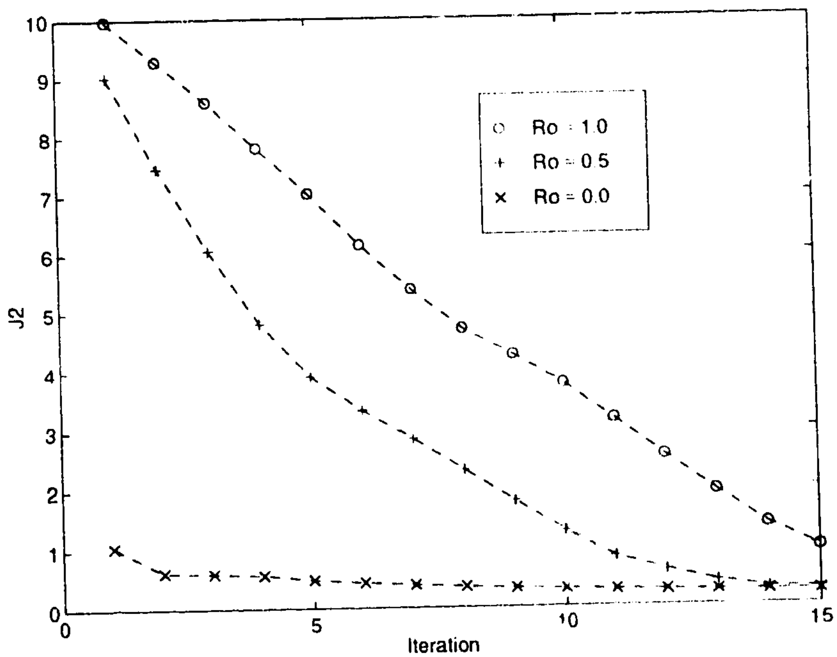

Figure 5.8: Mean Prediction Error vs. Iteration Number of NBPI

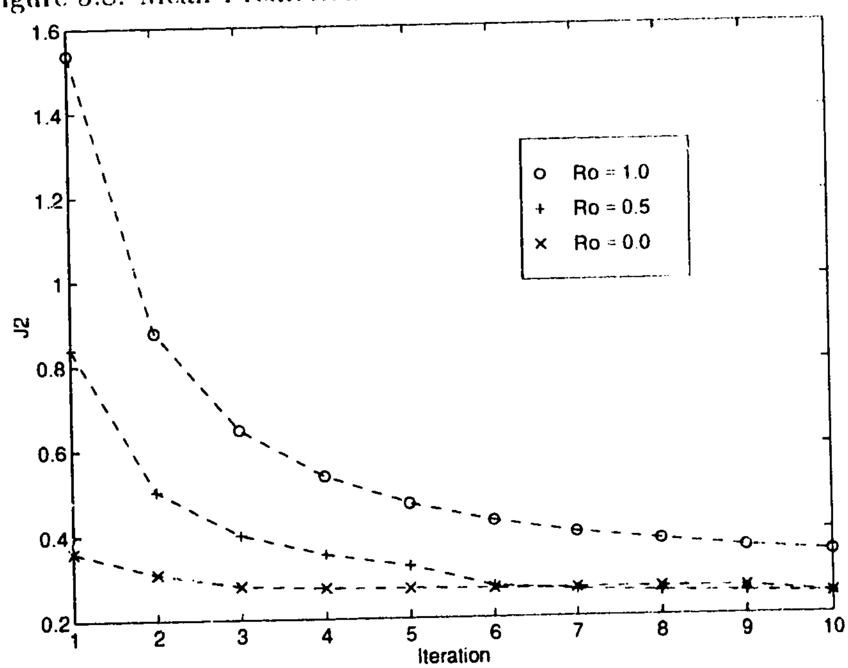

Figure 5.9: Mean Prediction Error vs Iteration Number of FRPI

Bibliography

[1] M. Bourke. "*Self-Learning Predictive Control Using Relational-Based Fuzzy Logic*". PhD thesis, University of Alberta, 1995.

[2] M.M. Bourke and D.G. Fisher. A Predictive Fuzzy Relational Controller. To be presented at IEEE International Conference of Fuzzy Systems, New Orleans, 1996.

[3] J.E. Box and G.M. Jenkins. *Time Series Analysis , Forecasting and Control.* Holden Day, San Francisco, CA, 1970.

[4] J.Q. Chen, J.H. Lu, and L.J. Chen. "An On-Line Identification Algorithm for Fuzzy Systems". *Fuzzy Sets and Systems*, 64:63–72, 1994.

[5] J. Valente de Oliveira. "Neuron Inspired Learning Rules for Fuzzy Relational Structures". *Fuzzy Sets and Systems*, 57:41–53, 1993.

[6] B.P. Graham and R.B. Newell. "Fuzzy Identification and Control Of A Liquid Level Rig". *Fuzzy Sets and Systems*, 26:255–273, 1988.

[7] B.P. Graham and R.B. Newell. Fuzzy Adaptive Control of a First Order Process. *Fuzzy Sets and Systems*, 31:47–65, 1989.

[8] R.C. Hall and D.E. Seborg. Modelling and Self-Tuning Control of a Multivariable pH Process. Part I : Modelling and Multiloop Control. In *Proceedings of the American Control Conference*, pages 1822 1827, Pittsburgh, 1989. American Control Conference.

[9] N. Ikoma, W. Pedrycz, and K. Hirota. "Estimation of Fuzzy Relational Matrix by Using Probablistic Descent Method". *Fuzzy Sets and Systems*, 57:335 349, 1993.

[10] I.S. Shaw and J.J. Krüger. "New Fuzzy Learning Model with Recursive Estimation for Dynamic System". *Fuzzy Sets and Systems*, 48:217 229, 1992.

[11] W. Pedrycz. "Numerical and Applicational Aspects of Fuzzy Relational Equations". *Fuzzy Sets and Systems*, 11:1 18, 1983.

[12] W. Pedrycz. "An Identification Algorithm in Fuzzy Relational Systems". *Fuzzy Sets and Systems*, 13:153 167, 1984.

[13] W. Pedrycz. "Neurocomputations in Relational Systems". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(3):289 296, March 1991.

[14] W. Pedrycz. *Fuzzy Control and Fuzzy Systems*. John Wiley & Sons, Inc., New York, N.Y., second extended edition, 1993.

[15] B. Postlethwaite. "A Model-Based Fuzzy Controller". *Chemical Engg. Research and Design, Part A : Trans IChemE*, 72:38 46, 1994.

[16] David S. Shook, Coorous Mohtadi, and Sirish L. Sh[illegible] A Control-Relevant Identification Strategy f[illegible] GPC. *IEEE Trans. Automatic Control*, 37(7):975–980, 1992.

[17] D.S. Shook, C. Mohtadi, and S.L. Shah. Identification for long range predictive control. *IEE-Proc.-D*, 138(1):75–84, 1991.

[18] T. Söderström and P. Stoica. *System Id[illegible]* [illegible]ries in Systems and Control Engineering. Prentice [illegible]

[19] C.W. Xu and Y.Z. Lu. "Fu[illegible] Mod[illegible] Identificati[illegible] and Self-Learning for Dynamic Systems". *IEEE Trans. of S[illegible] Man and Cyber[illegible]etics*, 17:683–689, 1987.

Chapter 6

Adaptive Fuzzy Relational Predictive Control

6.1 Introduction

There are certain control problems in which conventional control strategies have proven to be inadequate. These problems have the following characteristics :

- The process has fuzzy or imprecise measurements
- The process is poorly understood and highly complex
- The process is operated manually

Fuzzy logic based controllers have been used to control processes with the above characterics. There are two types of fuzzy controllers rule-based and relational-based. This chapter dwells into the issues of identification and design of fuzzy relational controllers for processes with the above mentioned characteristics.

Rule-based fuzzy controllers have been more widely used and in many cases have met with a high degree of success (e.g. Roffel & Chan (1991) and Oishi *et al.* (1991)) . These controllers utilize a set of "IF-THEN" rules involving linguistic values as its control strategy. However, a systematic controller design procedure does not exist. Controller design is intuitive and iterative. When a large number of variables are involved, this controller design procedure may be time-consuming.

A systematic identification procedure exists for fuzzy relational model. This leads to an orderly design methodology for model based controllers. This aspect of fuzzy relational models have led to the development of fuzzy relational controllers. Some of the more notable fuzzy relational controllers are those of Graham & Newell (1988), Postlethwaite (1994), Valente de Oliveira & Lemos (1995) and Bourke & Fisher (1996).

The main contribution of this chapter is a systematic theoretical and experimental evaluation of adaptive fuzzy relational predictive controllers especially those by Postlethwaite (1994) and Bourke & Fisher (1996). The performance of the above mentioned controllers are evaluated experimentally on a highly non-linear process. In addition, the effectiveness of different on-line fuzzy relational identification schemes namely those by Shaw & Krüger (1992) and the Fuzzy Relational Predictive Identification (FRPI) algorithm will be investigated when implemented with the fuzzy controllers.

6.2 Fuzzy Relational Identification

A fuzzy relational model is required for all fuzzy relational predictive controllers. The fuzzy relational identification procedure is summarized as follows :

- Determine an operating region and select the universe of discourse of each variable.
- Select the shape and size of membership function for both input and output variables.
- Collect input-output data over range of interest.
- Fuzzify input-output data.
- Select time-delay and model order.
- Estimate parameters of relational model using an appropriate fuzzy relational identification technique.
- Validate model.

Some of the identification algorithms in the literature include Pedrycz (1983, 1984, 1991), Shaw & Krüger (1992) and Ikoma *et al.* (1993). The identification algorithm by Shaw & Krüger (1992) estimates the parameters in the relational matrix by treating each entry of the relational matrix as the possibility of obtaining an output referential fuzzy set given a referential fuzzy set for each state and input variable in the relational equation. For a first order fuzzy relational equation such as the following

$$Y(k) = Y(k-1) \circ U(k-d-1) \circ R$$

where $Y(\cdot)$, $U(\cdot)$ are possibility vectors. R is the relational matrix and d is the time-delay. R is computed from

$$R = \frac{\sum_{k=d+2}^{N} \prod_{l=1}^{N_U} \prod_{j=1}^{N_{Y1}} \prod_{i=1}^{N_{Y2}} (U_l(k-d-1), Y_j(k-1), Y_i(k))}{\sum_{k=d+2}^{N} \prod_{l=1}^{N_u} \prod_{j=1}^{N_{Y1}} (U_l(k-d-1), Y_j(k-1))}$$

The FRPI algorithm is a numerical fuzzy relational identification algorithm which optimizes the relational matrix, R, such that the prediction error is minimized over some prediction horizon. The objection function for FRPI is then expressed as :

$$J_{FRPI} = \frac{1}{N-N_2} \sum_{k=1}^{N-N_2} \frac{1}{N_p} \sum_{j=N_1}^{N_2} \sum_{i=1}^{N_{Y2}} [\ Y_i(k+j) - \hat{Y}_i(k+j|k)\ ]^2$$

where $N_p = N_2 - N_1 + 1$. N_1 and N_2 is the prediction horizon. $\hat{Y}_i$ is the predicted membership grade for the i^{th} element of the possibility vector and N_{Y2} is the number of elements in the output possibility vector. Using a recursive prediction error method, the parameters in R which minimizes J_{FRPI} can be found.

Upon successful completion of the identification procedure, a fuzzy relational model will be available. The same membership functions chosen for the identification procedure must also be used for the controller since the fuzzy model was defined with respect to the membership functions used in the identification step.

There is an important difference between on-line fuzzy relational identification and conventional identification schemes. Conventional identification schemes assume a model and as the adaptation proceeds, the parameters of the model are refined to become increasingly accurate. Although the model may be inaccurate, it is complete, and can always be used to predict. Fuzzy identification starts with an empty model and no predictions can be made. When some data are available, only the part of the input-output space where the data is found will be updated. Therefore a fuzzy model may give good predictions at some parts of the operating regime and yet give meaningless predictions in others. A model which exhibits this deficency is known as an incomplete model. No analysis tools exist to guarantee that a given model is complete. Therefore it is desirable to collect open loop data which covers the entire operating range and obtain a model before implementing the controller. In addition, it may be wise to incorporate an on-line fuzzy relational identification scheme with the objective of maintaining a complete and accurate model.

6.3 Self-Learning Predictive Fuzzy Controller of Bourke (1995)

The Self-Learning Predictive Fuzzy Controller (SLPFC) is a SISO $(d+1)$ step ahead predictive controller where $(d+1)$ is the time-delay of the process (zero-order hold included). This controller utilizes two fuzzy relational models a dynamic model and a steady state model. A control action is found from each model and a combination of the two is selected based on the magnitude of a user-specified tuning parameter, α. The control action to be implemented at the k^{th} instant, $u(k)$ is calculated via,

$$u(k) = \alpha \cdot u_{gain}(k) + (1 - \alpha) \cdot u_{dync}(k)$$

where $u_{gain}(k)$ and $u_{dync}(k)$ are the control action as given by the steady-state and dynamic model respectively. $u_{gain}(k)$ which corresponds to the mean-level control action is calculated as follows :

$$U_{gain}(k) = Y_{sp}(k + d + 1) \circ G \tag{6.1}$$

where U_{gain} and $Y_{sp}(k + d + 1)$ are the fuzzy values of u_{gain} and $y_{sp}(k + d + 1)$ respectively. G represents the fuzzy mapping between steady-state values of the output and manipulated values.

$u_{dync}(k)$ is the deadbeat control action and is calculated by minimizing the following objective function :

$$J_{SLPFC} = |y_{sp}(k + d + 1) - \hat{y}(k + d + 1)| \tag{6.2}$$

A numerical search procedure must be used to compute $u_{dync}(k)$, and the heuristic search technique which was used involves the manipulation of the fuzzy vector $U_{dync}(k)$ to minimize (6.2). Details of the controller can be found in Bourke (1995). Figure (6.1) shows a block diagram of the SLPFC.

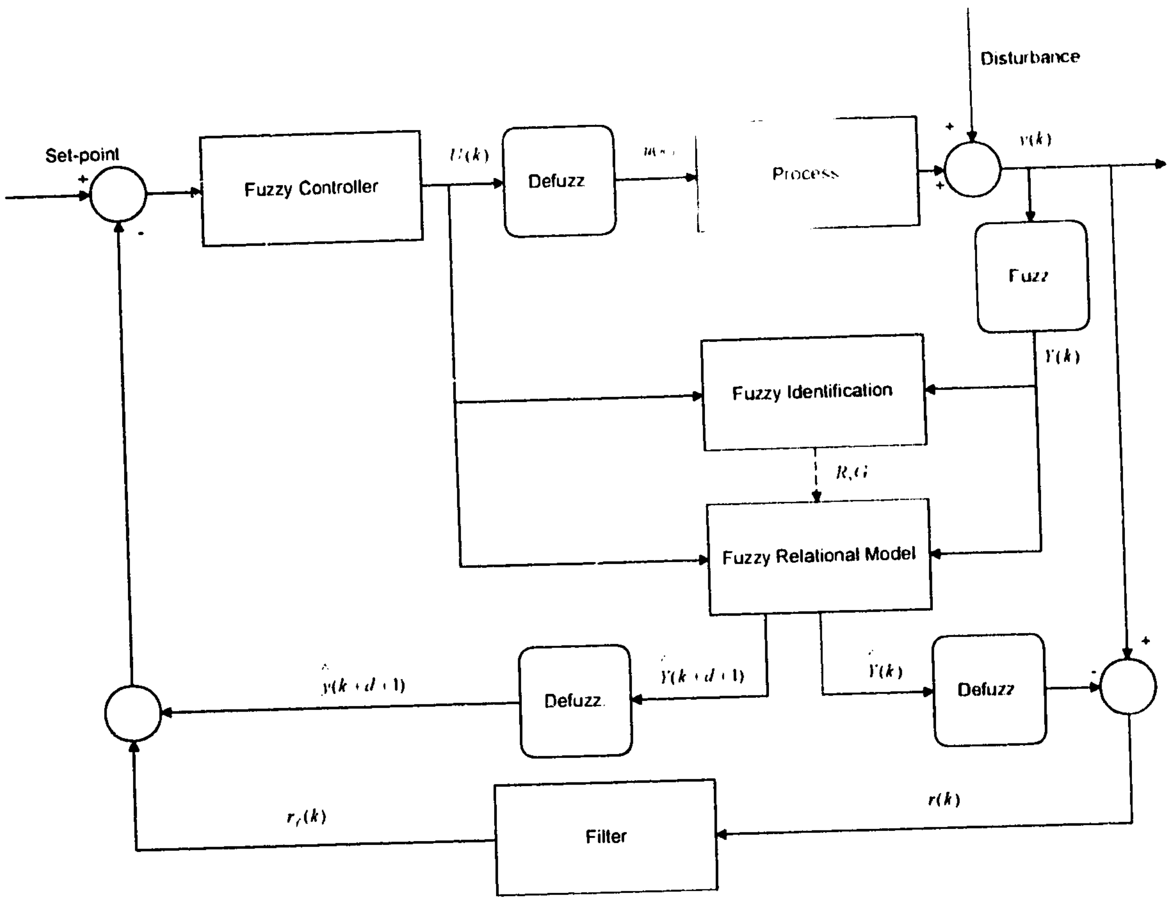

Figure 6.1: Block Diagram of Self-Learning Predictive Fuzzy Controller (SLPFC), Bourke (1995)

6.4 Fuzzy Relational Long Range Predictive Controller

The Fuzzy Relational Long Range Predictive Controller (FRLRPC) is designed to minimize the error over a multi-step prediction horizon. This is similar to the objective of conventional Long Range Predictive Controllers (LRPC) which is expressed as

$$J_{LRPC} = \sum_{j=N_1}^{N_2} (y_{sp}(k+j) - \hat{y}(k+j))^2 + \sum_{i=1}^{N_u} \lambda \, \Delta u(k+i-1)^2 \tag{6.3}$$

where N_1 and N_2 is the prediction horizon and N_u is the control horizon. The FRLRPC described here was proposed by Postlethwaite (1994). Postlethwaite (1994) used the fuzzy relational model instead of a discrete linear input-output model and minimized the following objective function :

$$J_{FRLRPC} = \sum_{j=N_1}^{N_2} (y_{sp}(k+j) - \hat{y}(k+j))^2 + \lambda u(k)^2 \tag{6.4}$$

However, Postlethwaite (1994) restricted N_2 to be $N_1 + 1$. To minimize (6.4) a numerical search technique must be used. Valente de Oliveira & Lemos (1995) provided a method to determine the gradient of J_{FRLRPC} with respect to $u(k)$ and presented simulation results for N_2 greater than $N_1 + 1$. Computing the optimal control action via a gradient-based search method is computationally expensive. To circumvent this problem, search methods which use function evaluations only are considered. Postlethwaite (1994) used the Fibonacci search method. Fibonacci's search is a highly efficient search technique which arrives at the optimum in a fixed number of function evaluations. To achieve a resolution of 1% of the manipulated variable range, eleven function evaluations are required.

The block diagram of the FRLRPC is shown in Figure (6.2).

$$\epsilon(k) = y(k) - \hat{y}(k)$$

$$E(k) = f_e \cdot E(k-1) + (1 - f_e) \cdot \epsilon(k)$$

and

$$u(k) = f_u \cdot u(k-1) + (1 - f_u) \cdot u_o(k)$$

6.5 Remarks about Fuzzy Relational Controllers

6.5.1 Offset

From a philosophical perspective, "zero-offset" should not be an issue in fuzzy control. It is unreasonable to measure the performance of fuzzy controllers using the same indicators as conventional control strategies since fuzzy controllers are supposed to complement conventional control strategies in areas where conventional control strategies have proven to be inadequate. Therefore, it does not seem reasonable to apply the same performance criteria to fuzzy controllers and expect to get results which will be classified as *good* when viewed from a conventional control strategy standpoint. The fuzzy equivalent of an offset free response

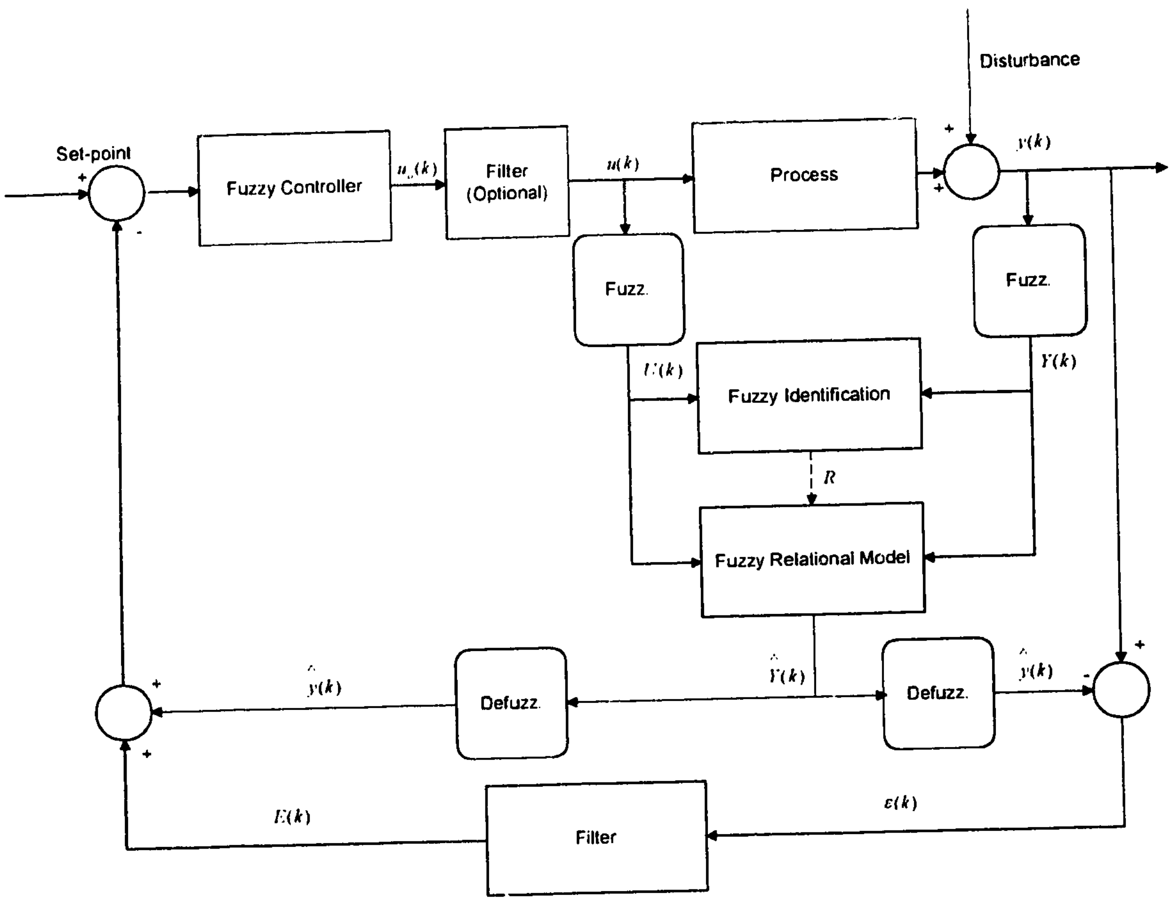

Figure 6.2: Block Diagram of Long Range Fuzzy Predictive Controller (FRLRPC)

should be when the output tracks the setpoint in the same membership function. Tracking within the same membership function does not mean that the output and setpoint are equal.

From a conventional control point of view one can see that for fuzzy relational controllers no explicit integrator exists for offset elimination. This was evident from the results of the SLPFC of Bourke (1995) when offset was observed in some of the closed-loop experiments. This can be attributed to the fact that in fuzzy relational models, the estimated process output, $\hat{Y}(k)$ is a function of the value of the manipulated variable, *i.e.* $U(k)$. A numerical search procedure is used to calculate the control action, $u(k)$, not the change in control action, $\Delta u(k)$. If $\Delta u(k)$ were determined, then an integrator would be present in the controller *i.e.*

$$u(k) = \frac{1}{\Delta} \cdot f$$

where f is some function and the offset for step inputs would be eliminated. For the FRLRPC, the minimization of (6.4) does not yield the increment in control action. An alternative approach is to adjust the process model so that the calculated control action, $u(k)$, will cause the process output to approach the desired value, *i.e.* drive the offset towards zero. This can be achieved by the incorporation of an on-line fuzzy relational identification scheme into the controller. The model has to be adapted such that offset is eliminated, *i.e.* a biassed gain may be estimated to remove or reduce offset.

6.5.2 Stability

Another concern regarding fuzzy controllers is the difficulty in proving their stability. P hwaite (1994) argues that since conventional controllers use linear proofs which cannot really be applied to non-linear systems, and more importantly, it is impossible to prove that a particular process model really represents the actual process then the issue of stability should not be considered as a major problem for fuzzy controller

6.6 Simulation Results

Bourke (1995) conducted extensive simulation studies on the SLPFC, hence, no further simulation ru s will be performed. Although Postlethwaite (1994) and Valente de Oliveira & Lemos (1995) have both conducted some closed-loop experiments using the FRLRPC, the ability of the FRLRPC in rejecting unmodelled and unmeasured disturbances has not been studied. In the following, both servo and regulatory performance of the FRLRPC will be investigated on processes which might be encountered in the chemical process industries.

6.6.1 pH Neutralization Process (Hall and Seborg (1989))

pH neutralization processes are often found in chemical processes. The highly non-linear behaviour makes it a very challenging control problem. A schematic diagram of the pH neutralization process of Hall & Seborg (1989) is shown in Figure (6.3). For simplicity, a SISO control problem was considered for this study.

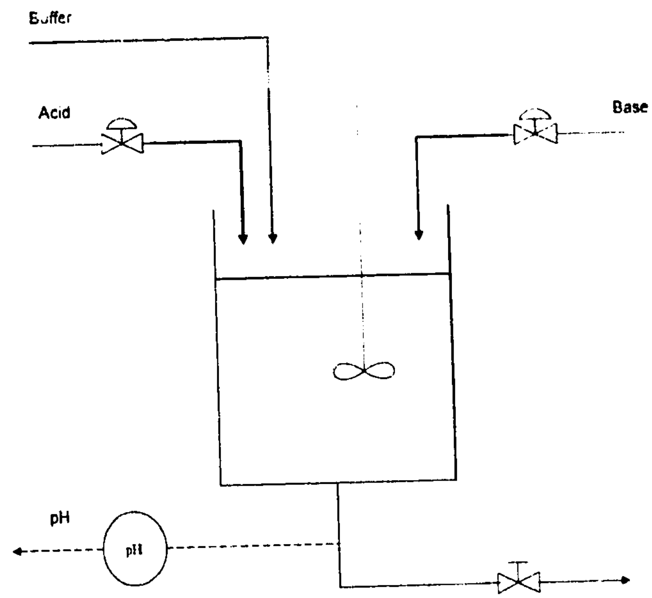

Figure 6.3: Schematic Diagram of pH Neutralization Process

The controlled and manipulated variables were effluent pH and acid flowrate respectively. The base and buffer flowrate were kept constant, and the level in the tank is not controlled. An unmodelled disturbance in the form of a variation in the base flowrate is introduced into the process to evaluate the regulatory performance at $t = 2700$ s (180^{th} sampling interval). A process model was identified off-line via the FRPI algorithm prior to the start of the closed-loop runs.

Figure (6.4) shows the servo and regulatory performance of the controller with FRPI as the on-line identification algorithm. The controller parameters as implemented in Figure (6.4) are $N_2 = 3$, $\lambda = 0$, $f_e = 0.9$ and $f_u = 0.1$. Overall, the controller is able to produce a good servo and regulatory performance.

Figure (6.5) shows the servo performance of the FRLRPC with FRPI im-

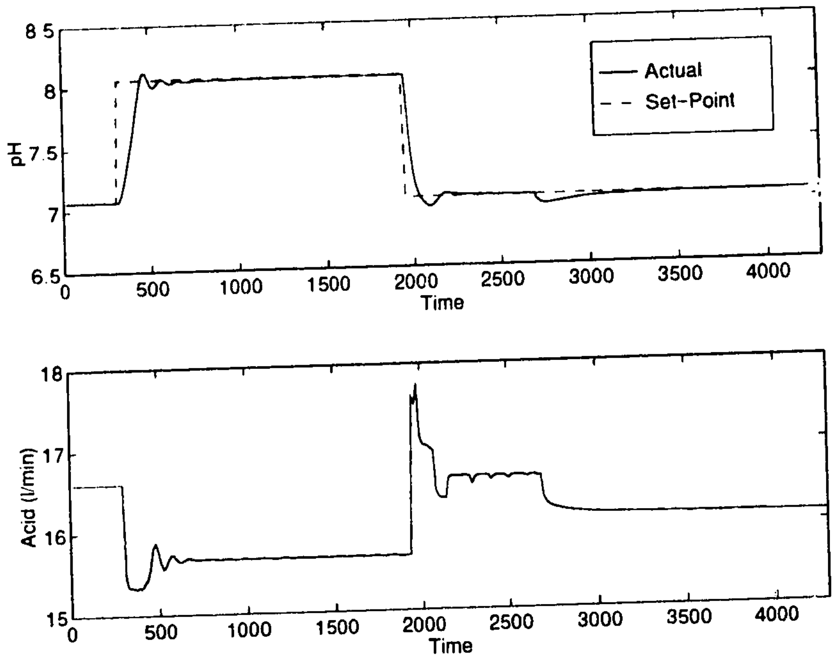

Figure 6.4: Servo & Regulatory Performance of FRLRPC in pH process

plemented while Figure (6.6) shows the controller performance without an on-line identification algorithm. The controller parameters for both these cases are $N_2 = 3$, $\lambda = 0$, $f_r = 0.9$ and $f_u = 0.1$. In Figure (6.5) the offset was finally eliminated with time but for Figure (6.6) the offset remained constant. Therefore, it is clear that the on-line identification algorithm eliminates offset. The on-line identification algorithm improves the model accuracy around the set-point such that any departure from the set-point can be detected. However, there is no guarantee that the incorporation of an on-line identification scheme will always eliminate offset.

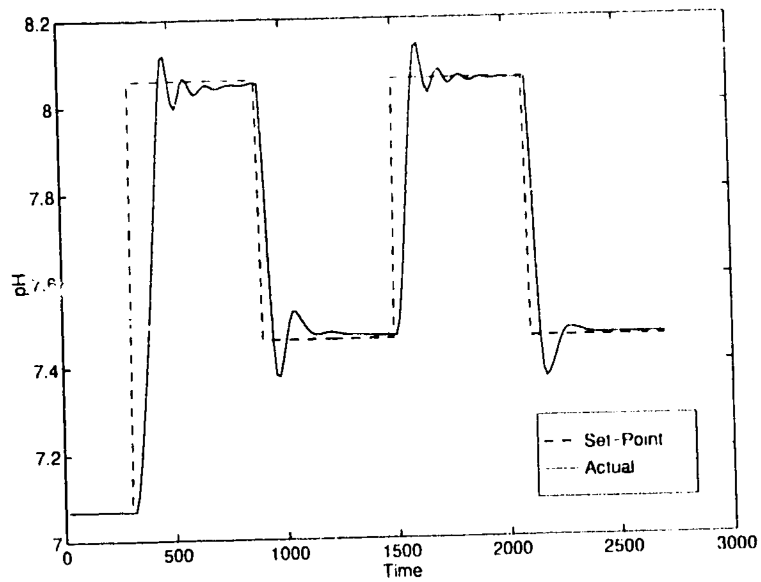

Figure 6.5: Servo Performance of FRLRPC with FRPI

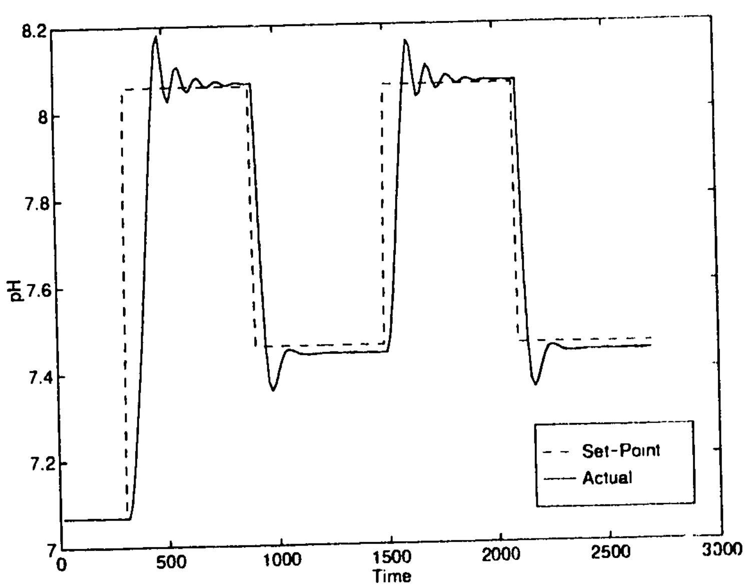

Figure 6.6: Servo Performance of FRLRPC without On-line Identification

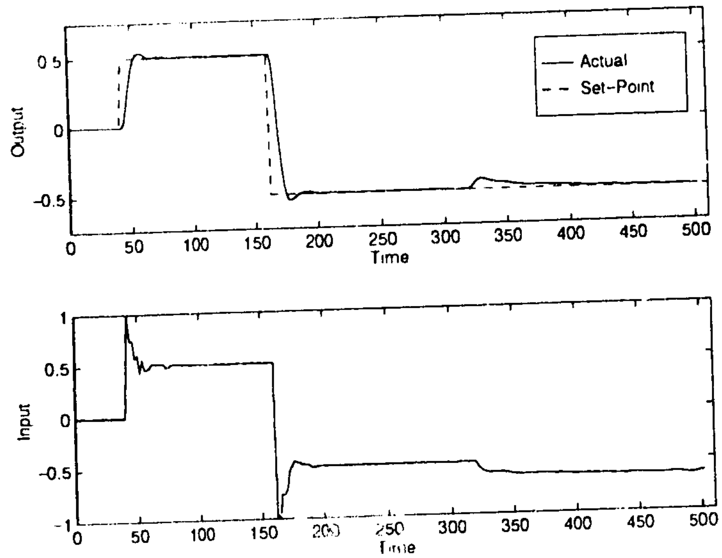

Figure 6.7: Performance of FRLRPC in McIntosh's Process ($N_2 = 3$)

6.6.2 High Order Process (McIntosh (1987))

The third-order linear process of McIntosh (1987) has the following transfer function :

$$G(s) = \frac{1}{(s+1)(3s+1)(5s+1)}$$

A first-order fuzzy relational model was used to approximate this process which was identified off-line with the FRPI algorithm prior to the start of the closed-loop run. A disturbance was introduced at $t = 320$s. The servo and regulatory response of the FRLRPC with FRPI as the on-line identification algorithm is shown in Figure (6.7) where $N_2 = 3$, $\lambda = 0$, $f_c = 0.9$ and $f_u = 0$. The controller performs well even in the presence of model plant mismatch.

6.7 Experimental Evaluation of Non-linear Interacting Tanks

The interacting tank apparatus consists of a pair of glass vessels connected by a manually adjustable valve (resistance), as shown schematically in Figure (6.8). The objective is to control the level at the conical section of the first tank by manipulating the inlet water flowrate to the firsttank. This results in a highly non-linear behaviour. Figure (6.9) shows step changes on the process; approximate process gains are listed as functions of level in Table (6.1). A parametric change in the process can be introduced by changing the positions of the adjustable valves at the outlet of either tank. This can be viewed as a disturbance in the process because in open-loop, an immediate change in liquid level will be observed. Nominal operating conditions for the interacting tanks are tabulated in Table (6.2).

Table 6.1: Approximate Gains of Experiment

Level (cm)	Gain (cm/%)
12.1 - 13.5	0.39
13.5 - 16.7	1.11

As mentioned earlier, the purpose of the experiment was also to assess the impact of different on-line fuzzy relational identification techniques on controller performance. The on-line algorithms which are to be evaluated include the FRPI algorithm and the algorithm by Shaw & Krüger (1992). The model was first identified by the algorithm to be evaluated on-line before starting the closed-loop

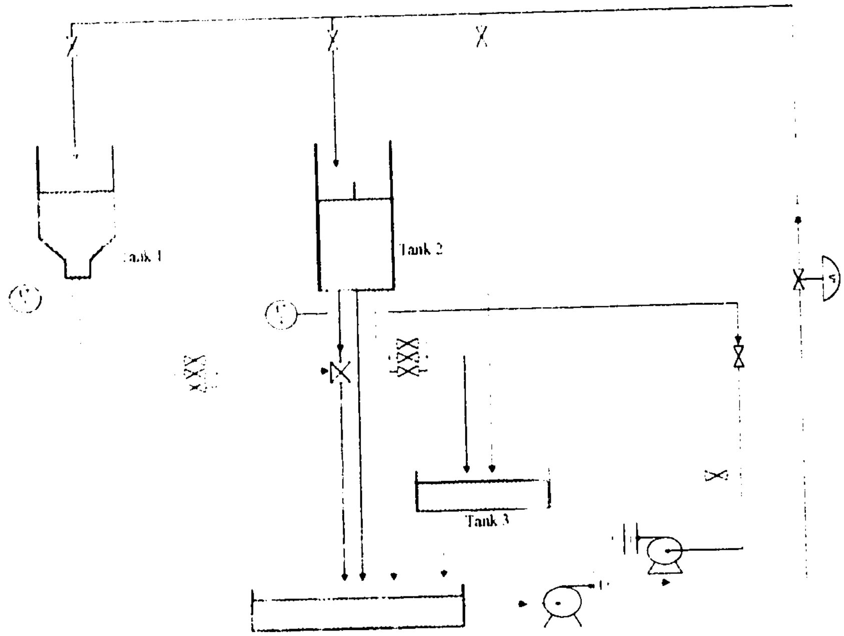

Figure 6.8: Schematic Diagram on Non-linear Interacting Tanks

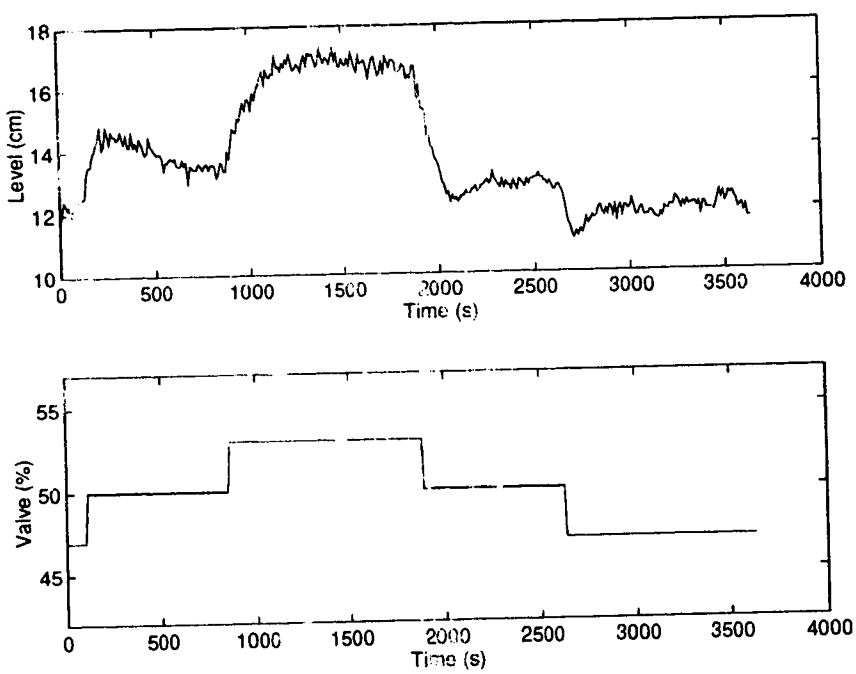

Figure 6.9: Open Loop Test of Interacting Tanks

Table 6.2: Nominal Operating Conditions for the Interacting Tanks

Pump Speed	50%
Valve Position	50%
Level	14.5 cm

runs.

A number of set-point changes were made and then a "disturbance" was introduced at the 160^{th} sampling interval (t =1600 s). Figure (6.10) and (6.11) shows the performance of the SLPFC implemented with Shaw's identification algorithm and FRPI algorithm respectively. Using the FRPI algorithm, the performance of the FRLRPC with a prediction horizon of 2 and 3 are shown in Figure (6.12) and (6.13) respectively. For the FRLRPC with a prediction horizon of 3, Shaw's identification algorithm was implemented and shown in Figure (6.14). Except for N_2 the rest of the controller parameters for all of the above three runs are identical, *i.e.* $\lambda = 0$, $f_c = 0.9$, $f_u = 0$. Table (6.3) summarizes the mean squared error (MSE) for all the experimental runs. Each run seen in Table (6.3) was repeated at least three times and the MSE varied by less than 0.01. The MSE tabulated are the least of all the runs made.

From Table (6.3), it is evident that the FRPI does offer some improvement in control performance. The improvement is most pronounced when the prediction horizon is longer *i.e.* when using the FRLRPC with $N_2 = 3$. An offset was observed at certain periods of the experimental run when using the FRLRPC with Shaw's scheme. As mentioned earlier, the on-line identification algorithm should

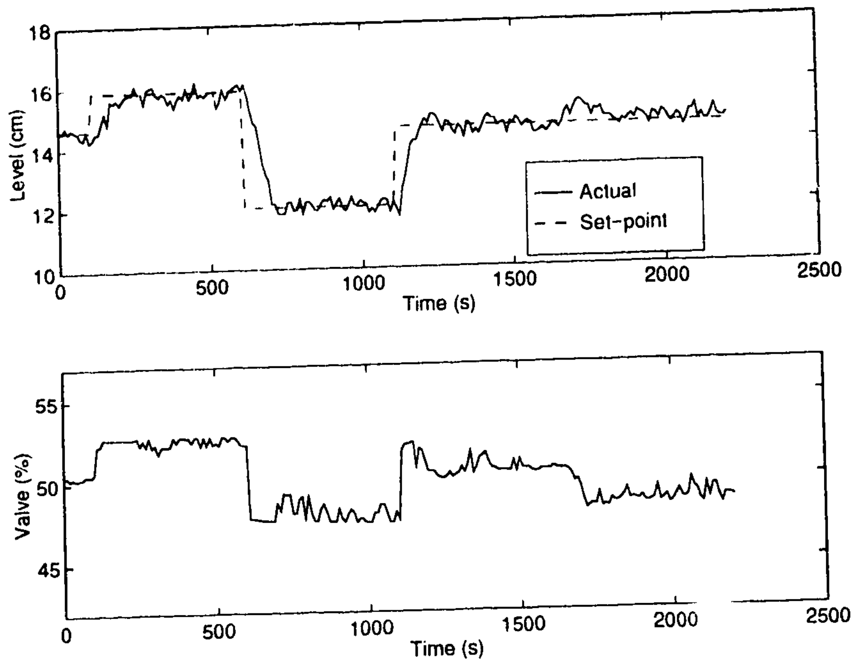

Figure 6.10: Performance of SLPFC with Shaw's Identification Scheme ($\alpha = 0.3$)

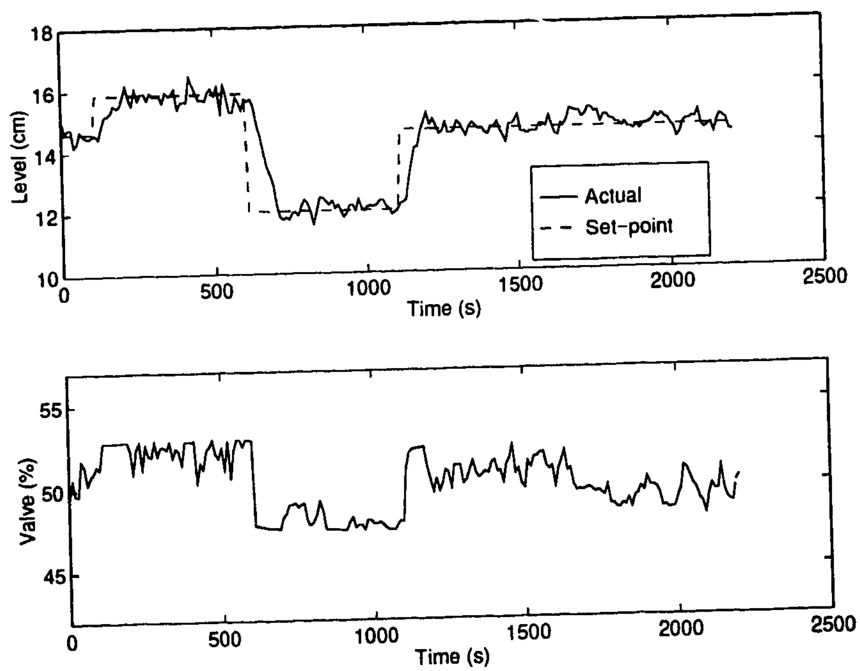

Figure 6.11: Performance of SLPFC with FRPI ($\alpha = 0.3$)

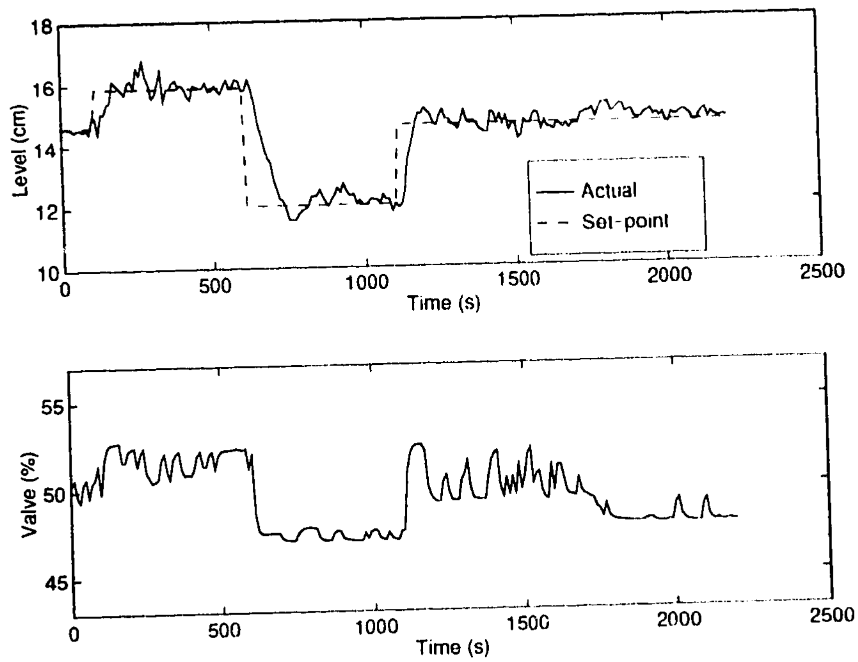

Figure 6.12: Performance of FRLRPC with FRPI Scheme ($N_2 = 2$)

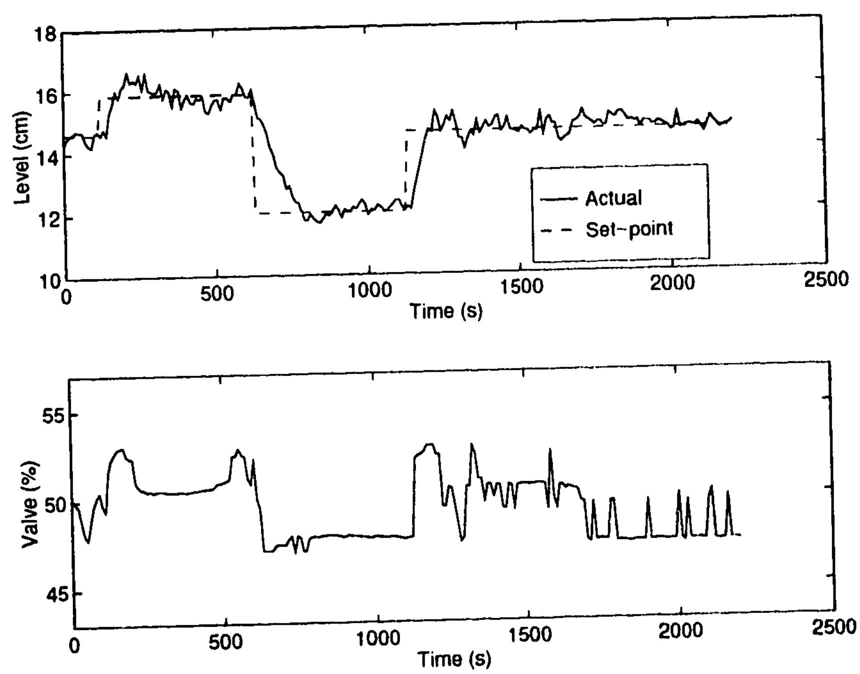

Figure 6.13: Performance of FRLRPC with FRPI Scheme ($N_2 = 3$)

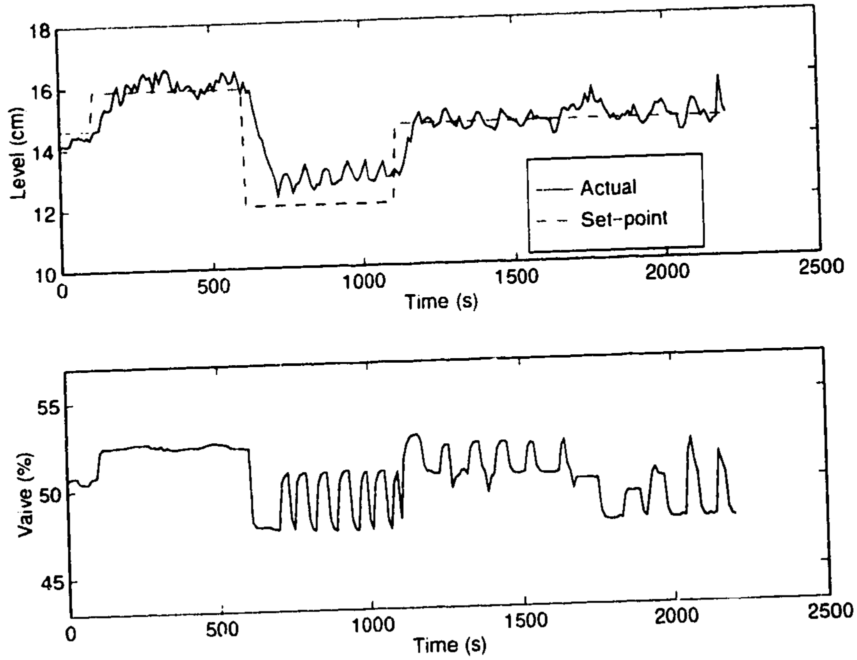

Figure 6.14: Performance of FRLRPC with Shaw's Identification Scheme ($N_2 = 3$)

improve model accuracy and eliminate offset. So it is shown here (Figure (6.14) that the conflict in identification objective (1-step prediction) and control objective (3-steps prediction) can sometimes lead to a significant decrease in fuzzy relational controller performance. In other words, the model obtained using Shaw's scheme gave good single step predictions but very poor long-range predictions which resulted in the poor control performance. It must be pointed out that the deterioration in controller performance when using a single step identification objective may not always be as severe as seen here. The highly non-linear behaviour of the process may have magnified the shortcomings of single step identification algorithms.

The aggresiveness of the controller output for both controllers can be further reduced if desired. For the SLFPC, this is achieved by increasing α. Increas-

Table 6.3: Mean Square of Errors for Experimental Runs

Controller	MSE_{Show}	MSE_{FRPI}
FRLRPC ($N_2 = 2$)		0.5281
FRLRPC ($N_2 = 3$)	0.6456	0.4501
SLFPC	0.4850	0.4408

ing α increases the weighting on the output given by the steady-state model and detunes the controller. For the FRLRPC, a detuned controller performance can be achieved by increasing the prediction horizon or filtering the control action. According to Pedrycz (1993) the predictive capability of the fuzzy relational models deteriorate rapidly with an increasing prediction horizon. Therefore, increasing the prediction horizon may not be a good approach to detune the controller as it may result in poor control performance. The effect of filtering the controller output is presented in Figure (6.15) where except for $f_u = 0.3$ the rest of the tuning parameters are identical to that used in Figure (6.12) . Based on the formulation of FRLRPC, increasing λ should also reduce the variance of the control action. In conventional LRPC, a good choice for λ is dependent on the process gain (McIntosh (1988)). For a non-linear process especially when the process gain varies greatly, a good choice of λ is not obvious. Further research must be directed at utilizing the λ-weighting in the FRLRPC.

6.8 Conclusions

Experimental evaluation of servo and regulatory tracking of two fuzzy relational predictive controllers namely the SLPFC (Bourke & Fisher, 1996) and FRLRPC

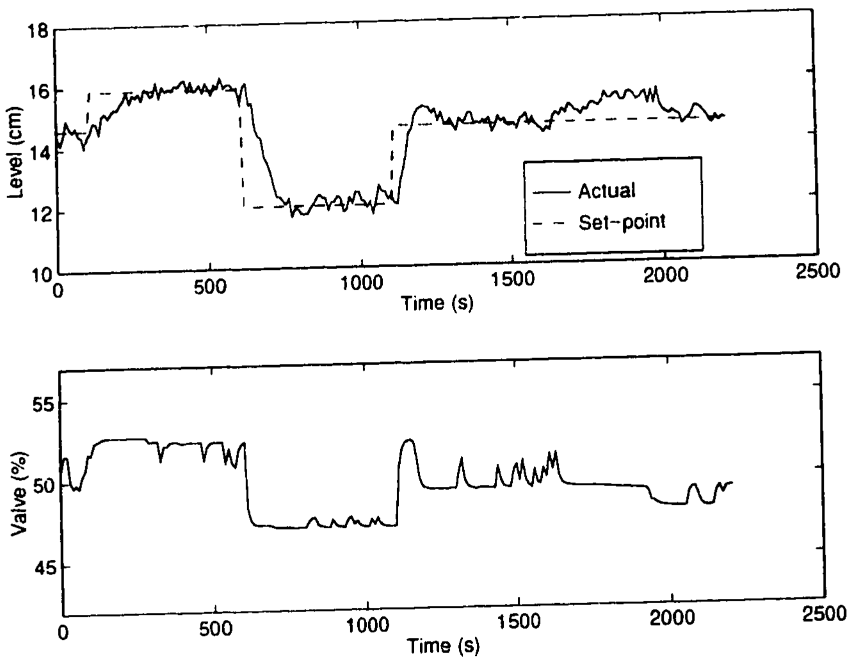

Figure 6.15: Performance of FRLRPC with filtered output and FRPI implemented.

(Postlethwaite, 1994) were performed. In addition, the effectiveness of on-line fuzzy relational identification algorithms such as those by Shaw & Krüger (1992) and the FRPI algorithm were investigated.

The two fuzzy relational controllers were found to give good performance. Based on the MSE, the SLPFC was slightly better than the FRLRPC. For on-line fuzzy relational identification algorithms, the FRPI algorithm as proposed in this thesis gave better control performances than the algorithm by Shaw & Krüger (1992). It was shown that the inconsistency between control objective and on-line identification objective may lead to a significant deterioration of control performance.

Bibliography

[1] M.M. Bourke and D.G. Fisher. A Predictive Fuzzy Relational Controller. To be presented at IEEE International Conference of Fuzzy Systems, New Orleans, 1996.

[2] Leon Cooper and David Steinberg. *Introduction to Methods of Optimization*. W.B. Saunders Company, 1970.

[3] J. Valente de Oliveira and J.M. Lemos. "Long-range predictive adaptive fuzzy relational control". *Fuzzy Sets and Systems*, 70:337 357, 1995.

[4] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, Inc., 1981.

[5] B.P. Graham and R.B. Newell. "Fuzzy Identification and Control Of A Liquid Level Rig". *Fuzzy Sets and Systems*, 26:255 273, 1988.

[6] B.P. Graham and R.B. Newell. Fuzzy Adaptive Control of a First Order Process. *Fuzzy Sets and Systems*, 31:47 65, 1989.

[7] R.C. Hall and D.E. Seborg. Modelling and Self-Tuning Control of a Multivariable pH Process. Part I : Modelling and Multiloop Control. In *Proceedings of*

the American Control Conference, pages 1822 1827, Pittsburgh, 1989. American Control Conference.

[8] N. Ikoma, W. Pedrycz, and K. Hirota. "Estimation of Fuzzy Relational Matrix by Using Probablistic Descent Method". *Fuzzy Sets and Systems*, 57:335 349, 1993.

[9] I.S. Shaw and J.J. Krüger. "New Fuzzy Learning Model with Recursive Estimation for Dynamic System". *Fuzzy Sets and Systems*, 48:217 229, 1992.

[10] A.R. McIntosh. Performance and Tuning of Adaptive GPC. Master's thesis, University of Alberta, 1988.

[11] K. Oishi, M. Tominaga, A. Kawato, Y. Abe, S. Imayasu, and A. Namba. Application of fuzzy control theory to the Sake brewing process. *Journal of Fermentation and Bioengineering*, 72(2):115 121, 1991.

[12] W. Pedrycz. "Numerical and Applicational Aspects of Fuzzy Relational Equations". *Fuzzy Sets and Systems*, 11:1-18, 1983.

[13] W. Pedrycz. "An Identification Algorithm in Fuzzy Relational Systems". *Fuzzy Sets and Systems*, 13:153-167, 1984.

[14] W. Pedrycz. "Neurocomputations in Relational Systems". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(3):289-296, March 1991.

[15] W. Pedrycz. *Fuzzy Control and Fuzzy Systems*. John Wiley & Sons, Inc., New York, N.Y., second extended edition, 1993.

[16] B. Postlethwaite. "A Model-Based Fuzzy Controller". *Chemical Engg. Research and Design, Part A : Trans IChemE*, 72:38–46, 1994.

[17] B. Roffel and P. Chan. Fuzzy Control of a Polymerization Reactor. *Hydrocarbon Processing*, pages 47–49, June 1991.

Chapter 7

Conclusions

7.1 Contributions and Conclusions

Although rule-based fuzzy logic has been used to solve very difficult problems in the area of process control, it is still difficult to synthesize fuzzy logic controllers. The difficulty is due to the number of parameters that must be decided, and the lack of tools to facilitate this decision. As a result, the task of controller design is often iterative and can often be time-consuming. The ability of relational-based fuzzy logic to derive process models directly from input-output data leads to a systematic controller design procedure. The trial-and-error inherent in fuzzy rule-based systems is eliminated. Therefore, the relational-based fuzzy system is indeed an attractive alternative to rule-based fuzzy logic. However, the study of fuzzy relational models in the context of process identification and control has not been very well-studied. Pedrycz (1983, 1984, 1991, 1994), Graham *et al.* (1988), Shaw *et al.* (1992), Valente de Oliveira (1993), Postlethwaite (1994) and Bourke (1995) have made some contributions in this area. The purpose of this thesis is to further improve the effectiveness of fuzzy relational logic as applied to process control and identification. Some of the contributions made toward this end include:

7.1.1 Identification

Two fuzzy relational identification algorithms aimed at giving accurate predictions over a practical prediction horizon were developed namely the Neuron-Based Predictive Identification (NBPI) algorithm and the Fuzzy Relational Predictive Identification (FRPI) algorithm. Based on the Box-Jenkins data set, both these algorithms gave better predictions (single and multi-step) compared to the effective fuzzy relational identification by Shaw *et al.* (1992). The FRPI algorithm is better than the NBPI because

- FRPI is formulated in a recursive fashion and is therefore, better suited for on-line implementations.
- FRPI gives faster convergence rates.
- FRPI is relatively insensitive to the initial guess of the relational matrix.

As a comfirmation of the utility of FRPI, the algorithm was implemented on-line to provide the model for fuzzy relational controllers. The fuzzy relational controllers were used to control a laboratory scale process. The controllers with FRPI gave significantly better performance in terms of sum of squares of error compared to those using the algorithm by Shaw *et al.* (1992).

7.1.2 Control

Two controllers namely those by Postlethwaite (1994) and Bourke (1995) were evaluated experimentally on a laboratory scale interating tanks process. Postlethwaite used a prediction horizon of $(d+2)$ where $(d+1)$ is the total time-delay of the process. In this work, the prediction horizon of the controller was extended and the controller was named as "Fuzzy Relational Long Range Predictive Controller" (FRLRPC). The performance of the FRLRPC was compared to Bourke's Self-Learning Predictive Fuzzy Controller (SLPFC). Both the servo and regulatory (unmeasured disturbance rejection) properties of this controller were investigated. The two controllers performed well under experimental servo and regulatory tracking. The performance of these two controllers was compared and the conclusion was the SLPFC gave a slightly better performance.

7.2 Recommendations

There still remains some aspects of fuzzy relational logic which deserve further research before these types of controllers can be implemented in a real plant environment.

7.2.1 Identification

Although the FRPI algorithm gives good results, it is conceivable that the prediction error achieved may not be the global minimum. Therefore global minimization techniques can be applied to solve for the relational matrix. However, global optimization schemes are known to be computationally intensive and may not be a practical on-line identification method. It is important to note that even if an alternative method which gives a global minimum can be found, the FRPI algorithm can still be used as an on-line fuzzy relational algorithm. The underlying assumption used in the recursive prediction error in FRPI is that the initial guess be relatively close to the minimum. So if a better initial guess is supplied, the FRPI approach should eventually converge or stay at the global minimum.

The *who* [illegible] e of fuzzy relational models must be excited to ensure that the model is c [illegible] incomplete model is one that gives poor predictions at certain parts of the [illegible] t space. However, there still remains no tools to aid in the design of the input sequence excitation during open loop test to ensure that the model is complete.

7.2.2 Control

A key part of all fuzzy relational controllers is the optimization which calculates the optimum process input. Again an optimization approach which guarantees a global solution is sought. An additional demand on this optimization approach is that it be computationally efficient since a control action must be calculated in a finite amount of time.

One of the key differences between the SLPFC and the FRLRPC is the presence of a steady-state model in the SLPFC. Since the SLPFC was found to perform slightly better the FRLRPC, perhaps a steady-state model can be incorporated into the FRLRPC. In addition, there is no explicit tuning parameter (λ-weighting in LRPC) to reduce the aggresiveness of the FRLRPC. Further research should be done so that the λ-weighting in FRLRPC can be fully utilized.

Bibliography

[1] J.E. Box and G.M. Jenkins. *Time Series Analysis , Forecasting and Control.* Holden Day, San Francisco, CA, 1970.

[2] J. Valente de Oliveira. "On Optimal Fuzzy Systems I/O Interfaces". In *Proc. 2nd. Int. Conf. on Fuzzy Systems*, volume 2, pages 851–856, 1993.

[3] I.S. Shaw and J.J. Krüger. "New Fuzzy Learning Model with Recursive Estimation for Dynamic System". *Fuzzy Sets and Systems*, 48:217–229, 1992.

[4] W. Pedrycz. "Numerical and Applicational Aspects of Fuzzy Relational Equations". *Fuzzy Sets and Systems*, 11:1–18, 1983.

[5] W. Pedrycz. "An Identification Algorithm in Fuzzy Relational Systems". *Fuzzy Sets and Systems*, 13:153–167, 1984.

[6] W. Pedrycz. "Neurocomputations in Relational Systems". *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(3):289–296, March 1991.

[7] W. Pedrycz. "Why Triangular Membership Functions?". *Fuzzy Sets and Systems*, 64:21–30, 1994.

[8] B. Postlethwaite. "A Model-Based Fuzzy Controller". *Chemical Engg. Research and Design, Part A : Trans IChemE*, 72:38 46, 1994.

Appendix A

Real-time Implementations

The real-time experimental evaluation of the Self-Learning Predictive Fuzzy Controller (SLPFC) and the Fuzzy Relational Long Range Predictive Controller (FRLRPC) was implemented on a 486-33 computer. The SLPFC and FRLRPC were implemented in LabVIEW and Real-time Matlab-Simulink respectively. The implementation of the two controllers is described in the next section.

A.1 Self-Learning Predictive Fuzzy Controller

LabVIEW is a graphical programming language with a front panel and a block diagram. The front panel is the interface between man and machine (program). Hence, it contains various knobs and dials to enable different options to be entered and also charts to show the state of the process under control. The block diagram in LabVIEW is where the source code of the program is written. The simplified block diagram of the SLPFC is shown in Figure (A.1).

Each block in Figure(A.1) is a sub-VI (sub virtual-instrument) which is equivalent to a subroutine in text-based programming language. The function of each of those blocks are :

- *G ID* : Allows for the on-line identification of the relational gain matrix, G.
- *FRPI* : Identifies the relational matrix, R.
- *Read Y* : Contains the necessary driver files to read the process outputs.
- *MS Pred.* : Performs a multi-step prediction.
- *Iterate* : Contains a heuristic numerical search technique that computes the optimum deadbead control action.
- *U AVG* : Computes the control action to be implemented.
- *Write* : Implements the control action on the process.
- *Ctrl Log* : Stores the data of the experiment in a file.

In Figure (A.1), P is the covariance matrix in the FRPI algorithm, F_Y is the fuzzified measured value of the process, $Prev$ contains all the previous process measurements and implemented control actions and New_V is the updated values of $Prev$.

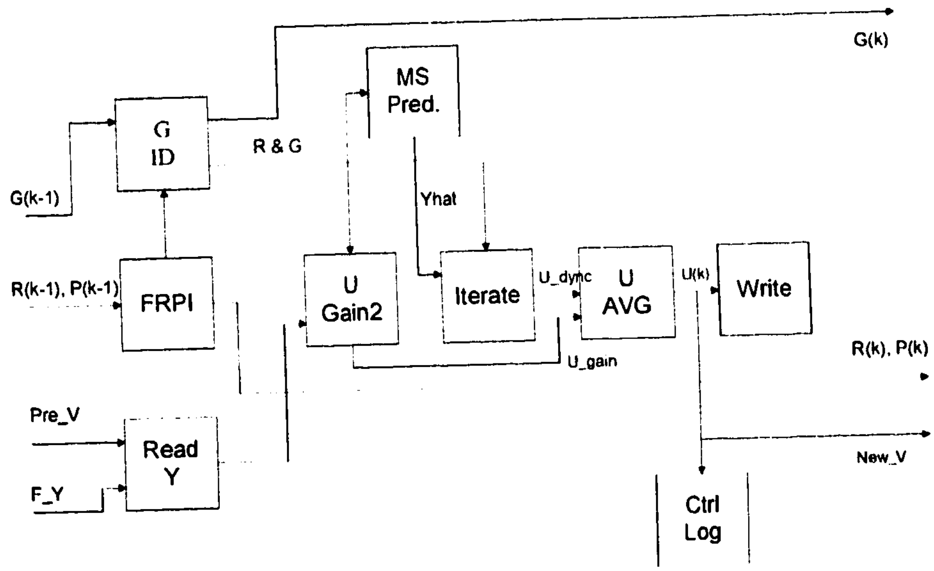

Figure A.1: LabVIEW Block Diagram of SLPFC

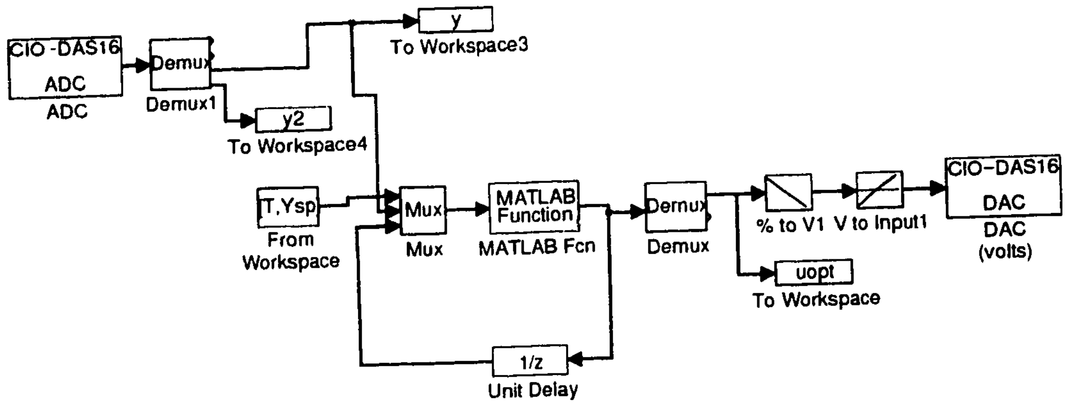

Figure A.2: Matlab-Simulink Block Diagram of FRLRPC

A.2 Fuzzy Relational Long Range Predictive Controller

The Real-time Matlab Simulink block diagram implementation of the FRL-RPC is shown in Figure (A.2).

The *Analog-Digital Converter (ADC)* and *Digital-Analog Converter (DAC)* blocks perform the convertion from analog signals to digital signals and vice versa. The actual control algorithm is written in a Matlab script file and is called from the *MATLAB Fcn* block. The *Unit Delay* block feeds back previous measured values and implemented control moves into the control algorithm. The *% to V1* and *V to Input1* blocks are calibration blocks which perform the necessary convertions.

Appendix B

Matlab Script Files

The Matlab script files required to run the Fuzzy Relational Long Range Predictive Controller (FRLRPC) are listed below. The files include mcontra.m and fibon3a.m . The adap.m and calxsi.m files are required to run the Fuzzy Relational Predictive Identification (FRPI) algorithm.

```
function [Out]=m_contra(In);
%**********************************************************************
% The input vector In contains the following information in order :
% ysp, y(t), uopt, U_f, Y_f, Yhat_f, Yhat_fd, Idx_hat, yhat, FB
% Sizes are :
% ysp = 1    x   1
% y(t) = 1    x  1
% uopt = n2 x   1
% U_f = n2   x   nref
% Y_f = n2   x   nref
% Yhat_f = n2^2 x nref
% Yhat_fd = n2^2 x nref
% Idx_hat = n2^2 x 2*nref
% yhat = n1 x 1
% FB = 1 x 1
% t = 1  x 1
% R = nref^2 x nref
```

```
% P = nref^3 x nref^2
%
% The output vector Out contains the following information in order :
% uopt, U_f, Y_f, Yhat_f, Yhat_fd, Idx_hat, yhat, FB
% Sizes are :
% uopt = n2 x   1
% U_f = n2   x   nref
% Y_f = n2   x   nref
% Yhat_f = n2^2 x nref
% Yhat_fd = n2^2 x nref
% Idx_hat = n2^2 x 2*nref
% yhat = n1 x 1
% FB = 1 x 1
% t = 1 x 1
% R = nref^2 x nref
% P = nref^3 x nref^2

tic;
global imfmid imfend omfmid omfend n2 n1 Ki lambda  gamma alpha up down;
global imfmid_ss imfend_ss omfmid_ss omfend_ss epsi dT

load mfn_4
nref=5;

% Extracting
[ysp,y,uopt,U_f,Y_f,Yhat_f,Yhat_fd,Idx_hat,yhat,FB,t,R,P]=extr1(In,
n1,n2,nref);
dT=t;

% Fuzzify measured y value
Y_f(2:n2,:)=Y_f(1:n2-1,:);
Y_f(1,:)=trifuzz(y,omfmid,omfend);

% Update R and P
if t> n2
remain=rem(t,n2);
if remain == 0
start = (n2-1)*n2 +1;
endd  = n2^2;
else
start = (remain-1)*n2 + 1;
endd  = remain*n2;
end
% more(20)
% Yhat_f
```

```
Y_id=Yhat_f(start:endd,:);
Index=Idx_hat(start:endd,:);
if t>15
[R,P]=adap(Y_id,Index,Y_f(1:n2,:),R,P,nref);
else
[R1,P]=adap(Y_id,Index,Y_f(1:n2,:),R,P,nref);
end
end

% Controller
remain=rem(t,n1);
if remain == 0
pix=n1;
else
pix=remain;
end

MPM= y - yhat(pix);
FB = filt*FB + (1-filt)*MPM;

disp(t);
[utemp,Utemp_f,yhatn,yss,Yhat_fdn,Idx_hatn,Yhat_fn]=fibon3a(fneval,
uopt,U_f,Y_f(1,:),R,ysp,FB,sum_error,rate_c,G,0,filt2);
%utemp
yhat(pix)=yhatn;

remain = rem(t,n2);
if remain == 0
start = (n2-1)*n2 +1;
endd  = n2^2;
else
start = (remain-1)*n2 + 1;
endd  = remain*n2;
end

Yhat_f(start:endd,:)=Yhat_fn;
Yhat_fd(start:endd,:)=Yhat_fdn;
Idx_hat(start:endd,:)=Idx_hatn;
U_f(2:n2,:)=U_f(1:n2-1,:);
U_f(1,:)=Utemp_f;
uopt(2:n2)=uopt(1:n2-1);
uopt(1)=utemp;

% Increment time
t = t+1;
```

```
% Restore output
[Out]=restr1(uopt,U_f,Y_f,Yhat_f,Yhat_fd,Idx_hat,yhat,FB,t,R,P);

% Save R,P if required
% if t==t_end   % Define t_end
% save Rspace1 R P Out
% end

% Clock and pause
time = toc;
ptime=Sam_T-time;
pause on
pause(ptime);
end



function [ut,Unew_f,yy,yss,Yhat_fd,Idx_hat,Yhat_f]=fibon3a(feval,uimp,
Uimp_f,Y_f,R,ysp,MPM,serr,rate_c,G,MPM2,filt2);
%*********************************************************************
% function [ut,Unew_f,yy,yss,Yhat_fd,Idx_hat,Yhat_f]=fibon3(feval,uimp,
Uimp_f,Y_f,R,ysp,MPM,serr,rate_c,G);
% Numerical search for optimum u in the LRPFC using Fibonacci's method
% ut : optimum u
% Unew_f : optimum u - fuzzified
% Yhat_fd : predicted output conditioned on time=t (fuzzy)
% uimp : implemented u sequences
% Uimp_f : implemented fuzzy u sequences
% Y_f  : fuzzy y from time = 1:t
% R    : relational matrix
% ysp    : set-point
% MPM    : Model Plant Mismatch
% yy    : estimated output n1 steps ahead
% serr    : sum of error
% rate_c   : rate constraint on input
% G    : steady state relational matrix
%*********************************************************************
global  omfmid omfend imfmid imfend n2 n1 Ki lambda gamma gain
global  omfmid_ss omfend_ss imfmid_ss imfend_ss epsi
global  dT neval
% Fill in implemented U's

if n1==1
```

```
U=[];
else
U(1:n1-1,:)=Uimp_f(n1-1:-1:1,:);
end

% Set search region
if nargin>=9
h_bound = rate_c + uimp(1);
l_bound = -rate_c + uimp(1);
else
h_bound=1e12;
l_bound=-1e12;
end

if size(imfmid,2)==3  % Triangular %
u_low=max(imfend(1,1),l_bound); % Low %
u_high=min(imfend(2,2),h_bound);% High%
else
u_low=max(imfend(1,1),l_bound);  % Low %
u_high=min(imfend(2,1),h_bound); % High%
end
ysp_hat=ysp+Ki*serr;

%*******
% [illegible] = total number of fn evaluations across search region
[illegible](1)-u_low;
n_[illegible](epsi*(uimp(1)-u_low)/(u_high-u_low));

if n_low<=0
n_low=1;
end
n_high=epsi-n_low;
if n_high<=0
n_high=1;
n_low=epsi-1;
end

d_low=(uimp(1)-u_low)/(n_low+1);
d_high=(u_high-uimp(1))/(n_high+1);

for i=1:n_low
u(i)=u_low+i*d_low;
end
i=i+1;
u(i)=uimp(1);
```

```
for j=1:n_high
i=i+1;
u(i)=uimp(1)+j*d_high;
end

for ii=1:epsi+1
ZZ=U;
tvec=trifuzz(u(ii),imfmid,imfend);
ZZ=[ZZ; tvec];
for jj=1:n2-n1
ZZ=[ZZ; tvec];
end
JZ(ii)=djdu_ss2(R,Y_f,ZZ,MPM,ysp_hat,uimp,G,MPM2);
end

[Jz_min,Jz_idx]=min(JZ);
if Jz_idx==1
u_high=u(Jz_idx+1);
elseif Jz_idx==epsi+1
u_low=u(Jz_idx-1);
else
u_low=u(Jz_idx-1);
u_high=u(Jz_idx+1);
end

% Fibonacci Search Follows %

U_lft=U;
U_rgt=U;
if (u_high-u_low)>1e-8
FN=[1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765];
% Set up region for Fibonacci search %
TOP=feval-1; BOT=feval+1;
delta=(u_high-u_low)*FN(TOP)/FN(BOT);
x_lft=u_low+delta;
x_rgt=u_high-delta;
fprintf('x_lft = %g ; x_rgt = %g \n',x_lft,x_rgt);
% fprintf('u_low = %g ; u_high = %g \n',u_low,u_high);



tvec_lft=trifuzz(x_lft,imfmid,imfend);
tvec_rgt=trifuzz(x_rgt,imfmid,imfend);
% pause;
```

```
U_lft=[U_lft; tvec_lft];
U_rgt=[U_rgt; tvec_rgt];
for j=1:n2-n1
U_lft=[U_lft; tvec_lft];
U_rgt=[U_rgt; tvec_rgt];
end
[J_lft,y_l    ss_lft,Yid_lft,Idx_lft,Yud_lft]=djdu_ss2(R,Y_f,
U_lft,MPM,y    hat,uimp,G,MPM2);
[J_rgt,y_rgt,yss_rgt,Yid_rgt,Idx_rgt,Yud_rgt]=djdu_ss2(R,Y_f,
U_rgt,MPM,ysp_hat,uimp,G,MPM2);
TOP=TOP-1; BOT=BOT-1;
% disp(J_lft); disp(J_rgt);
% pause;
if J_lft<J_rgt
u_high=x_rgt;
delta=(u_high-u_low)*FN(TOP)/FN(BOT);
xnext=u_low+delta;
J_rgt=J_lft; x_rgt=x_lft; y_rgt=y_lft; yss_rgt=yss_lft;
Yid_rgt=Yid_lft; Idx_rgt=Idx_lft; Yud_rgt=Yud_lft;
J_lft=[]; x_lft=[]; y_lft=[];
yss_lft=[]; Yid_lft=[]; Idx_lft=[]; Yud_lft=[];
elseif J_lft>J_rgt
u_low=x_lft;
delta=(u_high-u_low)*FN(TOP)/FN(BOT);
xnext=u_high-delta;
J_lft=J_rgt; x_lft=x_rgt; y_lft=y_rgt; yss_lft=yss_rgt;
Yid_lft=Yid_rgt; Idx_lft=Idx_rgt; Yud_lft=Yud_rgt;
J_rgt=[]; x_rgt=[]; y_rgt=[];
yss_rgt=[]; Yid_rgt=[]; Idx_rgt=[]; Yud_rgt=[];
end

for neval=3:feval-1
tvec=trifuzz(xnext,imfmid,imfend);
UZ=[U; tvec];
for j=1:n2-n1
UZ=[UZ; tvec];
end
[J,ytemp,ytemp2,ytemp3,ytemp4,ytemp5]=djdu_ss2(R,Y_f,UZ,MPM,
ysp_hat,uimp,G,MPM2);
if J_lft==[]
J_lft=J;
y_lft=ytemp;
yss_lft=ytemp2;
Yid_lft=ytemp3;
Idx_lft=ytemp4;
```

```
x_lft=xnext;
Yud_lft=ytemp5;
elseif J_rgt==[]
J_rgt=J;
y_rgt=ytemp;
yss_rgt=ytemp2;
Yid_rgt=ytemp3;
Idx_rgt=ytemp4;
x_rgt=xnext;
Yud_rgt=ytemp5;
end
if neval==feval-1
% fprintf('Final Evaluation at time %g \n',t);
fprintf('J_lft = %g , J_rgt = %g \n ',J_lft,J_rgt);
% pause;
if J_lft<=J_rgt
% fprintf('Path 1 \n');
ut1=x_lft;
yy=y_lft;
yss=yss_lft;
Yhat_fd=Yid_lft;
Yhat_f=Yud_lft;
Idx_hat=Idx_lft;
Unew_f=trifuzz(ut1,imfmid,imfend);
elseif J_lft>J_rgt
% fprintf('Path 2 \n');
ut1=x_rgt;
yy=y_rgt;
yss=yss_rgt;
Yhat_fd=Yid_rgt;
Yhat_f=Yud_rgt;
Idx_hat=Idx_rgt;
Unew_f=trifuzz(ut1,imfmid,imfend);
end
% ut1
ut=filt2*uimp(1) + (1-filt2)*ut1;
% ut
tvec=trifuzz(ut,imfmid,imfend);
UZ=[U; tvec];
for j=1:n2-n1
UZ=[UZ; tvec];
end
[J,yy,yss,Yhat_fd,Idx_hat,Yhat_f]=djdu_ss2(R,Y_f,
UZ,MPM,ysp_hat,uimp,G,MPM2);
Unew_f=tvec;
```

```
break;
end
TOP=TOP-1; BOT=BOT-1;
if J_lft<=J_rgt
u_high=x_rgt;
delta=(u_high-u_low)*FN(TOP)/FN(BOT);
xnext=u_low+delta;
J_rgt=J_lft; x_rgt=x_lft; y_rgt=y_lft; yss_rgt=yss_lft;
J_lft=[]; x_lft=[]; y_lft=[]; yss_lft=[];
elseif J_lft>J_rgt
u_low=x_lft;
delta=(u_high-u_low)*FN(TOP)/FN(BOT);
xnext=u_high-delta;
J_lft=J_rgt; x_lft=x_rgt; y_lft=y_rgt; yss_lft=yss_rgt;
J_rgt=[]; x_rgt=[]; y_rgt=[]; yss_rgt=[];
end

end
else
Ueq=U;
ut1=u_high;
ut=filt2*uimp(1) + (1-filt2)*ut1;
Unew_f=trifuzz(ut,imfmid,imfend);
Ueq=[Ueq;Unew_f];
for j=1:n2-n1
Ueq=[Ueq;Unew_f];
end
[J,yy,yss,Yhat_fd,Idx_hat,Yhat_f]=djdu_ss2(R,Y_f,Ueq,MPM,
ysp_hat,uimp,G,MPM2);
end
fprintf('Optimum u = %g \n',ut);



function [R,P]=adap(Y_id,Id_idx,Y_f,R,P,nref);

dim_R=[ 5 5 5];
for i=1:nref
Xsi=cal_xsi(Y_id,Id_idx,Y_f,i,dim_R,R);
E=Y_f(:,i)-Y_id(:,i);
[R,P]=update(P,E,Xsi,R,i,dim_R);
end
```

```
function [Xsi]=cal_Xsi(Yhat,Index,Y,i,dim_R,R);
%*********************************************************************%
% function [Xsi]=cal_Xsi(Yhat,Index,Y,i,dim_R,R); %
% Xsi is a npar X np matrix instead of a npar X 1      %
% vector. Xsi is calculated wrt the ith possibility vector element in Y
%
% Inputs : %
% Yhat = fuzzy estimated output and has the form %
% Yhat=[Yhat(n1) ;...;Yhat(n2)]; %
% Index= contains pairs of numbers corresponding to  %
% location of maxima as returned by ms_pred.m %
% the function ms_pred.m %
% Y    = fuzzy actual output. %
% dim_R= dimensions of R matrix u X x X y %
%*********************************************************************%
nref_y = size(Y,2);
np = size(Yhat,1);
bhat = Yhat(:,i);
b = Y(:,i);

% Extract relevant indices for ith mf of Yhat %
indx_i = Index(:,2*(i-1)+1:2*i);

% Convert 2D indices into a single number %
n_tita  = (indx_i(:,1)-1)*dim_R(1)+indx_i(:,2);

% Extract relevant layer for ith mf %
R_i = R((i-1)*dim_R(1)+1:i*dim_R(1),:);
Xsi = zeros(dim_R(1)*dim_R(2),np);
B = Yhat;

for j=1:np
% if R_i(indx_i(j,1),indx_i(j,2))==0
% error('Error in cal_Xsi 1');
% end
% if sum(B(j,1:size(B,2)))==0
% error('Error in cal_Xsi 2');
% end
Xsi(n_tita(j),j)=bhat(j)/max(R_i(indx_i(j,1),indx_i(j,2)),1e-6);
% S(j) =sum(B(j,1:size(B,2)));
% Xsi(n_tita(j),j)=dy/S(j);
end
```