

AUTOMATING KNOWLEDGE ACQUISITION AND REPRESENTATION WITH FUZZY LOGIC

By

Lester Arlington Jordon

M.Sc. Reading University, Reading, England, 1984

B.Sc. Imperial College, London, England, 1983

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
DEPARTMENT OF MINING AND MINERAL PROCESS ENGINEERING

We accept this thesis as conforming to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

September 1991

copyright © Lester Arlington Jordon, 1991

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

Department of Mining & Mineral Process Engineering
The University of British Columbia
Vancouver, Canada

Date 28/10/91

ABSTRACT

An interactive computer program has been developed to automate the construction of knowledge bases in expert system development. The program incorporates many user friendly features which allow users who may not understand the concepts of fuzzy logic or the complexity of the knowledge acquisition process to develop useful and complete Expert Systems.

Expert systems formalism are being successfully used in the industry. A major bottleneck in development of these systems lies in the knowledge acquisition phase. This study presents an induction system, based on fuzzy logic, which successfully automates the knowledge acquisitions process when the domain knowledge is embodied within discrete numerical data.

A tabloid database representing some desired input/output relationship is input to the system. Fuzzy expressions such as "high", "low" or "large" are either created automatically by the system or described explicitly by the user. These fuzzy expressions are automatically associated into an accurate and minimum rule set which models the desired input/output relationship.

The system was tested on the development of a knowledge base for a secondary crusher controller and a second more complex non-monotonic input/output relationship. In the controller example the knowledge bases developed were simpler, more accurate and built much faster than using a traditional interview approach. The non-monotonic relationship was also accurately described by a simple knowledge base.

CONTENTS

ABSTRACT	ii
CONTENTS	iii
LIST OF FIGURES.....	iv
LIST OF TABLES	ix
NOTATION	xii
ACKNOWLEDGEMENTS	xiv
1.0 INTRODUCTION	1
2.0 EXPERT SYSTEM AND FUZZY LOGIC FORMALISMS	3
2.1 Why Fuzzy Logic Formalisms.....	3
2.2 The Concept of Fuzziness	3
2.2.1 Notation and Terminology	4
2.2.2 Operations of Fuzzy Sets	6
2.2.3 Function Representation of Fuzzy Sets	8
2.2.4 If... Then Rules	10
2.2.5 Inferencing Ambiguities	11
2.2.6 Creation of Parent Fuzzy Sets	12
2.3 Traditional Construction Of Expert Systems	15
2.4 Verification of Knowledge Base	17
3.0 AUTOMATED ACQUISITION AND REPRESENTATION	19
3.1 Rule Induction	20
3.1.1 Deduction	20
3.1.2 Induction	20
3.1.3 Requirements for Induction	21
3.2 Personal Construct Theory	23
3.2.1 Repertory Grid Analysis	23

3.2.3	Grid Elicitation	25
3.2.4	Grid Analysis	26
4.0	ACQUISITION AND REPRESENTATION METHODS	28
4.1	Knowledge Acquisition Technique	28
4.1.1	Reference Data	29
4.1.2	Definition of Fuzzy Subsets	29
4.2	Locating Fuzzy Subsets on Reference Data	31
4.2.1	Location of X and Y Subsets	31
4.2.2	Location of Z Subsets	32
4.3	Selecting Dominant Subsets	32
4.4	Minimizing the Number of Rule Descriptions	33
4.4.1	Defining parent fuzzy sets	33
4.4.2	Reduction by "not" qualifier	34
4.4.3	Final Rule Description	34
4.5	Defuzzification	35
4.5.1	Weighted Average	36
4.5.2	Confidence Level	37
4.5.3	Saddle Point	37
4.5.4	Maximum Value	38
4.5.5	Selection of Defuzzification Method	38
4.6.	Error Determination	39
4.7	Error Reduction Strategies	40
4.7.1	Adding a Single Set to Each Variable	41
4.7.2	Relocating Subsets	44
4.7.3	Adding Sets to Area of Greatest Error	43
5.0	DESCRIPTION OF FLIX	46
5.1	Program Specifications	46
5.2	Main Routines	46

5.2.1	Input/Output Files	47
5.3	Knowledge from Data	47
5.3.1	Manual Rule Generation	49
5.3.1.1	Prerequisite Inputs	49
5.3.1.2	Generated Reports	49
5.3.1.3	Graphical Output	50
5.3.2	Automatic Rule Generation	50
5.3.2.1	Increasing Subset Definitions	50
5.4	Data from Knowledge	51
5.4.1	Prerequisite Inputs	51
5.5	Combining Modes for Knowledge Base Development	51
6.0	MODELLING A SECONDARY CRUSHER CONTROLLER	53
6.1	The Process	53
6.2.1	Original Knowledge Acquisition Methods	54
6.2.1.1	Observing and Interviewing the Operator ...	55
6.2.1.2	Standardizing Linguistics	55
6.2.1.3	Definition of Membership Functions	55
6.3.1	Original Control Rules.....	55
6.4	Knowledge Acquisition with FLIX	60
6.4.1	Reference Controller Profile	60
6.4.2	Generation of Knowledge Base	60
6.4.2.1	Automatic Acquisition.....	60
6.4.2.2	Manual Acquisition	64
6.5	Test on Alternative Non-monotonic Surface.....	66
7.0	FUTURE DEVELOPMENTS	67
7.1	Limitation of Baysian Logic	68
7.2	Neural Networks.....	68
7.3	Extension to Multiple Dimensions	70

7.4	Rule Reduction Methods	70
7.5	Error Reduction Strategies - Automated	71
8.0	CONCLUSIONS	72
	REFERENCES	73
APPENDIX 1	Fuzzy Membership Functions	75
APPENDIX 2	Error Functions	79
APPENDIX 3	FLIX Menu Options	81
APPENDIX 4	Sample Default Files of Non-monotonic Data Surface	91
APPENDIX 5	Data For Controller Tests	94
APPENDIX 6	Table of C Source Code FLIX Files	106

LIST OF FIGURES

2.1.1	Fuzzy Set "Bin Level High"	5
2.1.2	Non-Fuzzy "Bin Level High"	5
2.2	Operations of Fuzzy Sets	7
2.3	Operators Subjective Definition of Bin Level "High"	9
2.4.1	Fuzzy Subset Descriptions of Bin Levels	12
2.4.2	Creation of Parent Set "between (low & medium)"	14
2.5	Information System Compared to Expert System	15
3.1	Construction of Repertory Grid (after Hart 1989).....	25
3.2	Focused Grid of Operators Skill	27
4.1	Membership Functions Generated for X, Y and Z Subsets ..	30
4.2	Three Stages of Single Set Additions to X Variable.....	42
5.1.1	Knowledge from Data Mode	48
5.1.2	Data from Knowledge Mode	48
5.3	Integration of FLIX Modes for Knowledge Based Development	52
6.1	Secondary Crusher Control System	53
6.2.1	Original Set Definitions for Controller after Meech and Jordon (1990)	57
6.2.2	Automatic FLIX Generated Definitions for Controller	58
6.2.3	Manual FLIX Generated Set Definitions for Controller ...	59
6.3.1	Original Rule Generated Profile	61
6.3.2	Desired Controller Profile	61
6.3.3	Automatic FLIX Controller Profile	62
6.3.4	Manual FLIX Controller Profile	62
6.4.1	Reference Data for Non-monotonic Data Surface	67

6.4.2	Rule Generated Data for Non-monotonic Data Surface	67
7.1	Neural Network Representation of Knowledge.....	69
A1.1	General Definition of Flat Bell Membership Function	75

LIST OF TABLES

2.1	Function Representation of Fuzzy Subsets	10
2.2	Qualifiers and Description	14
2.3	Parent Sets in Bin Level Domain	14
2.4	Stages and Tasks for Development of Expert Systems after (Hayes-Roth, 1983)	16
2.5	Issues for Resolution in Expert System Building after (Hart, 1989)	17
3.1	Operators Average Daily Production	21
3.2	Types of Attributes with Examples	22
3.3	Repertory Grid Comparing Operators in a Plant	26
4.1	Reference Database of Z Values	28
4.2	Membership Functions for X,Y and Z Subsets	29
4.3	Mapping of X and Y Subsets in Fuzzy Plane with Cells Showing Discrete z values	31
4.4	Mapping of Subset in Fuzzy Plane, Cells Show Grades of Membership for Z Subsets	32
4.5	Dominant Fuzzy Subsets	33
4.6	Final Rules	35
4.7	Generated Data by Weighted Average Method	39
4.8	Error Report with Difference Values	40
4.9	Stages for Adding Single Subsets to X Variable	41
4.10	Third Stage Subset Definitions for X,Y and Z variables	44
4.11	Relocating X,Y and Z Subsets	45
4.12	Additional Subsets for Error Reduction	45
5.1	Manual Knowledge From Data Sub-Menus	49
5.2	Data From Knowledge Sub-Menus	51

6.1	Original Control Rules	56
6.2	Correlation Coefficient for Automatic Subset Additions.....	63
6.3	Automatic Generated Rules	64
6.4	Correlation Coefficients for Manual Subsets Additions.....	65
6.5	Manual Generated Controller Rules	65
6.6	Final Automatic Rules for Non-Monotonic Surface.....	66
A4.1	<i>zdata</i> File (same as <i>zin</i>)	91
A4.2	<i>zsets</i> File	91
A4.3	<i>zmap</i> File	92
A4.4	<i>zdom</i> File.....	92
A4.5	<i>zrules</i> File (same as <i>zknow</i>)	92
A4.6	<i>zout</i> File.....	93
A4.7	<i>zerr</i> File.....	93
A5.1	Reference Database of Feed Rate Changes at Screen Bin Level and Current Draw Values	94
A5.2	Automatic Membership Functions for Controller.....	95
A5.3	Automatic Dominant Subsets for Feed Rate Changes at Screen Bin Levels and Current Draw Values	96
A5.4	Automatic Final Rules for Controller	97
A5.5	Automatic Generated Database of Feed Rate Changes at Screen Bin Level and Current Draw Values	94
A5.6	Automatic Error Report with Difference Values for Controller	99
A5.7	Stage 1- Manual Membership Functions for Controller	100
A5.8	Stage 1- Manual Error Report for Controller.....	101
A5.9	Stage 2- Manual Membership Functions	102
A5.10	Stage 2- Manual Error Report for Controller.....	103

A5.11	Stage 6- Manual Membership Functions	104
A5.12	Stage 6- Dominant Subsets for Manual Tests	105
A5.13	Stage 6- Manual Error Report for Controller.....	105

NOTATION

Fuzzy Notation

$M_F(u)$	-	Membership value or degree of belief of fuzzy set
\in	-	An element of
\cup	-	Union of distinct subsets
\cap	-	Intersection
\subset	-	Subset
$'$	-	Complement
\wedge	-	Minimum
\vee	-	Maximum
$+$	-	Union of membership and discrete value $M_F(u) / u$ pairs

General Notation

C	-	Confidence Limit
i	-	i th location from a total of n rows
j	-	j th location from a total of m columns
k	-	k th fuzzy subset in the Z domain
M_k	-	Grade of membership of the k th subset.
$M_{\hat{k}}$	-	Selected k th subset with maximum grade of membership.
$M_k(u) / u$	-	Point on fuzzy membership function.
$[n \times m]$	-	Table with n rows and m columns
p	-	Total number of subsets in the X or Y domain
Sup_k	-	Supremum value of the k th fuzzy subset.

Sup_k	-	Supremum value of the maximum k th fuzzy subset.
S	-	Saddle Point
t	-	Total number of fuzzy subsets in the Z domain
w_i	-	Link weight of i th connection in a neural network
x_i	-	A discrete member of X at i th location
x	-	Discrete member of X
X	-	Abcissa variable
y	-	Discrete member of Y
Y	-	Ordinate variable
Z	-	Z variable in Cartesian coordinate system
z	-	Discrete member of Z
z_{ij}	-	(i,j) element of reference data
\hat{z}_{ij}	-	(i,j) element of rule generated data
\tilde{z}_{ij}	-	(i,j) element of error report or ($z_{ij} - \hat{z}_{ij}$)

ACKNOWLEDGEMENTS

Dedicated to - Jacqueline C. M. Jordon and Sheila Y. Jordon

The author wishes to thank Dr. J. A. Meech for his supervision. Financial support from NSERC is gratefully acknowledged. Special thanks are extended to Dave Gagne who probably regrets introducing me to *main {}* which committed him to teaching me more than I ever wanted to know about C. Appreciation is also expressed to E. Cifuentes and C. A. Harris for the millions of parts they played which must add up to something.

1.0 INTRODUCTION

In a plant, many unit processes or systems can be adequately described using inexact relationships, heuristics or rules-of-thumb. Very often, information representing the behavior of the process exists in a numerical format. Experienced personnel can interpret this data, to explain both the process behavior and how specific control actions affect the process. These explanations are often stated using linguistic expressions that are easy to understand by colleagues or new personnel.

Expert System formalism is especially attractive because of the ability to process symbolic or heuristic information. However since the formative years of development, it has been stated frequently that knowledge acquisition is "the critical bottleneck" (Feigenbaum & McCorduck, 1984; Hart, 1989).

Generally the knowledge acquisition phase involves close interaction between a knowledge engineer and expert personnel. The expert understands the data (numbers) and its heuristic interpretation while the knowledge engineer knows how to efficiently elicit and organize the heuristics for processing by the expert system. These tasks are both tedious and complex in their methodologies, although some knowledge domains are exceptions to this general rule. If the methodologies can be deciphered, then computer tools could be implemented that would automate the acquisition and representation processes.

The objective of this research is focused on methodologies which can simplify and automate knowledge acquisition and representation from historical discrete data. An induction system is presented where the principals of fuzzy logic are applied to interpret data describing a particular relationship in numerical format into heuristic or symbolic descriptions. The outcome of this process is a rule-based model, that accurately represents the desired relationship.

The system has been applied to model a control system used for a secondary crusher. Numerical data, representing the major dependent and manipulated variables in the circuit are input. A rule-based model is then induced to represent the existing control strategy. The effectiveness of the procedure to acquire and represent knowledge is compared with conventional procedures.

2.0 EXPERT SYSTEM AND FUZZY LOGIC FORMALISMS

The precise method of storage and manipulation of concepts and information in the mind is a subject of considerable research in the Artificial Intelligence community. Several popular techniques have emerged which include probability theory, personal construct theory, neural networks and fuzzy logic, each with its own merits and successes (Hart, 1989; Michie & Johnston, 1985).

2.1 Why Fuzzy Logic Formalisms?

We communicate our conscious processes to other people with words rather than numbers. This may be why humans find fuzzy logic highly-intuitive and very easy to understand. Language is used to convey the many shades of meaning e.g. "very", "slightly", "definitely". We are then able to make logical deductions without the precise rigor of mathematical theory, logic or even computer languages (Hart, 1989).

2.2 The Concept of Fuzziness

It is widely recognized that much human reasoning is not conducted using classical Aristotelian logic. According to Zadeh, the strength of human reasoning is the ability to grasp inexact concepts directly, rather than approximating some precisely defined process by using a mathematical simulation based on true/false or on/off logic (Zadeh, 1984).

For example a shift operator might explain his control actions as:

'Whenever the screen-bin level is high, I make a large decrease in the feed rate'.

The perception of a "high" bin level and the inference of a "large" decrease to the feed are characteristically human. Reproduction of this "fuzzy thinking" through the use of logical processes based on precise knowledge is limited by the requirement of a precise

definition of the terms "high" and "large". Fuzzy Set Theory and Approximate Reasoning deal with the simulation of the human thought process, by introducing the concepts of vagueness and imprecise measure into the interpretation of the information (Harris & Meech, 1987).

A fuzzy set consists of particular objects, or ranges of numbers and their respective grades of membership in the set. The *grade of membership* or *degree of belief* (dob) of an object is given by a subjectively defined membership function. The value of the grade of membership ranges from 0 to 1. A value of 1 denotes full membership while the closer the value is to 0, the weaker the membership in the fuzzy set.

2.2.1. Notation and Terminology

Consider material in bins, at the two levels, 50% and 80% respectively. We can express these levels using the fuzzy concept "High" for bin levels. Grades of membership then provide a natural way of expressing difference in levels. By subjectively assigning grades of membership for each level we can express the fact that 80% is higher than 50% by $M_{\text{high}}(50\%) = 0.5$ and $M_{\text{high}}(80\%) = 0.9$ respectively. Figure 2.1 illustrates this definition together with some frequently used terms.

The *support* of the subset "high" is the set of elements of U for which $M_F(u)$ is positive so that we have any degree of belief in high.

The *supremum* of "high" is the set of elements of U for which $M_F(u)$ is maximum.

Fuzzy sets are said to be *normal* if their height is unity, or $\sup_{u \in U} M_F(u) = 1$.

Fig. 2.1.1 Fuzzy Set "Bin Level High"

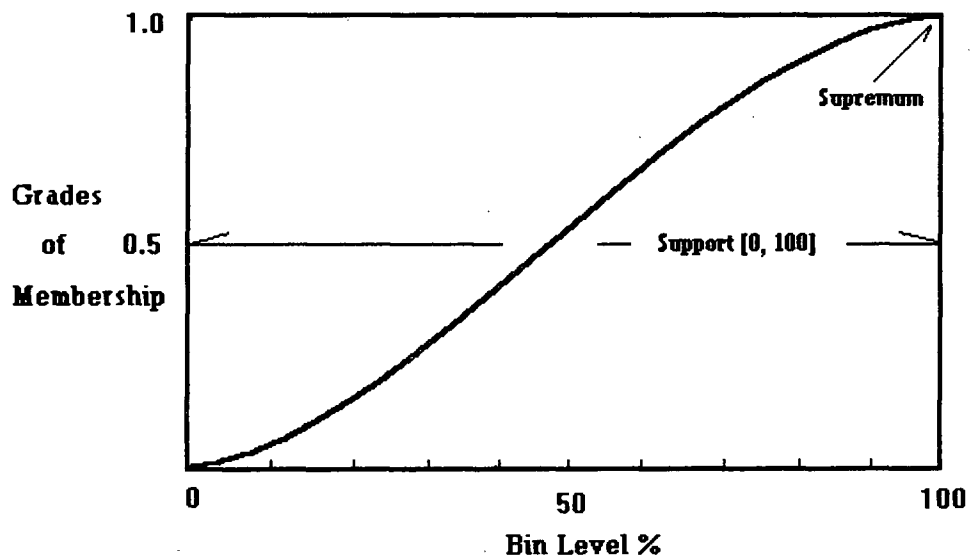
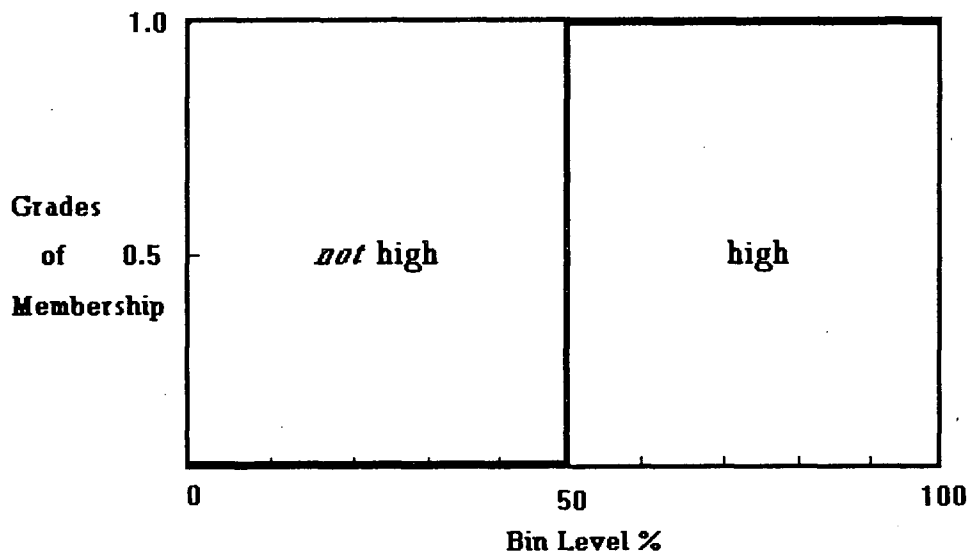


Fig. 2.1.2 Non-Fuzzy "Bin Level High"



In set theory a *universe of discourse (domain)* F is defined as all elements or concepts which can be grouped as identifiable labelled units known as sets or subsets:

$$F = F_1 + F_2 + \dots + F_n$$

Which for our example of a bin level could be:

Bin Levels = empty + low + medium ++ high + full

So F is the set of n objects F_1, F_2, \dots, F_n which are considered. A *fuzzy set* is described by its members and grades of membership (membership function) to that set. For example let U be the universe of discourse with the generic element of U denoted by u . Then *fuzzy set* F of U is characterized by the membership function M_F , which associates with each element u of U , a number $M_F(u)$ representing the grade of membership of u in F . Then:

$$F = \{(u, M_F(u)) \mid u \in U\}$$

or

$$F = M_1/u_1 + M_2/u_2 + \dots + M_n/u_n$$

u_1, u_2, \dots, u_n are members with respective grades of membership M_1, M_2, \dots, M_n . In other words, this equation is a way to list the various members together with their degrees of membership. Using this description, the high subset in Figure 2.1 could be defined as:

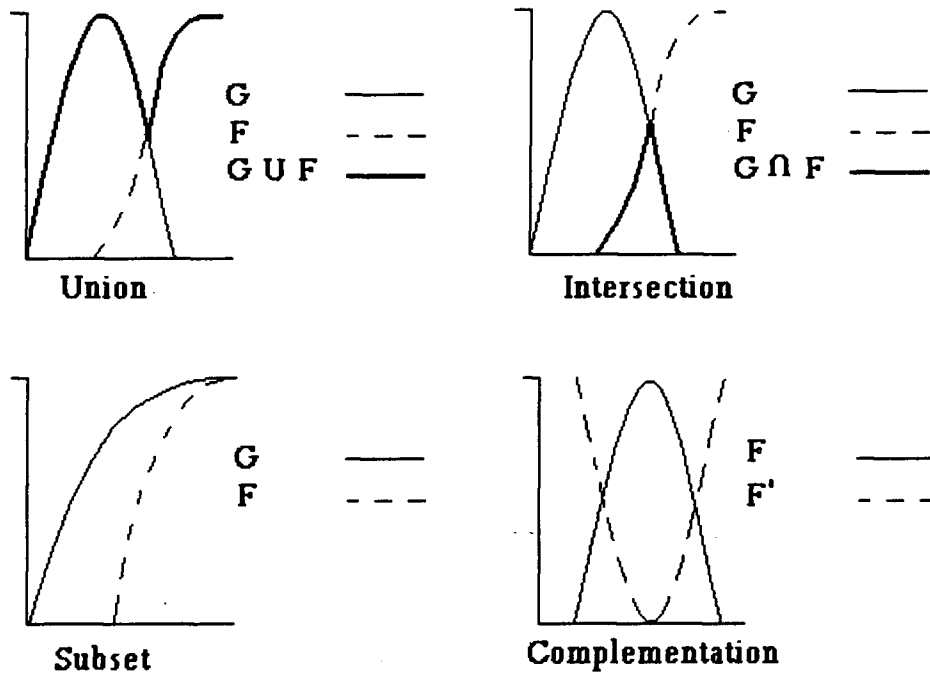
$$\text{high} = 0/0 + 0.5/50 + 0.8/90 + 1.0/100$$

The number of elements used to define any particular subset is arbitrary. There are many ways to draw the set through the points selected. One common method is to use a straight line between adjacent points.

2.2.2. Operations of Fuzzy Sets

F is a *subset* of G or is *contained* in G if $M_F(u) \leq M_G(u)$ for all elements u of U . The notation is listed below and illustrated in Figure 2.2.

Fig. 2.2 Operations of Fuzzy Sets



$$F \subset G = M_F(u) \leq M_G(u), \quad u \in U.$$

The *union* of fuzzy subsets F and G is a fuzzy subset, $F \cup G$ (F OR G), which defines a grade of membership of an element u by its maximum membership in each of the two sets. So:

$$F \cup G = \sum M_F(U) \vee M_G(U) / U$$

The *intersection* of fuzzy subsets F and G is a fuzzy subset, $F \cap G$ (F AND G), which defines a grade of membership, of an element u by its minimum membership in each of the two sets. So:

$$F \cap G = \sum M_F(U) \wedge M_G(U) / U$$

The *compliment* of a set F is F' (NOT F) which is a reflection of F across the 0.5 membership line and is defined by:

$$F' = 1 - \sum M_F(U) / U$$

2.2.3. Function Representation of Fuzzy Sets

Let the universe of discourse U for a bin level be the interval [0, 100] with u interpreted as level in percentage (%). To define a fuzzy subset H of U labeled "High", it is convenient to express the high membership function M_H as a standard function that maps a fuzzy subset of a real line onto the interval [0, 1]. One very useful definition is the S-function, a piecewise quadratic expression defined as follows:

$$\begin{aligned} S_{2pt_bell_up}(u; a, c) &= 0 && \text{for } u \leq a \\ &= 2 \left(\frac{u-a}{c-a} \right)^2 && \text{for } a \leq u \leq b \\ &= 1 - 2 \left(\frac{u-c}{c-a} \right)^2 && \text{for } b \leq u \leq c \\ &= 1 && \text{for } u \geq c \end{aligned}$$

where the parameters a and c are the lower and upper fuzzy subset interval bounds (critical points), respectively, and the parameter $b = (a+c)/2$ is the *crossover point*, i.e. the value of u at which $S_{2pt_bell_up}(u; a, c) = 0.5$ (Zemankova-Leech, 1984).

Using the S-function definition, the subset "High" could be characterized subjectively, by two different operators as the following membership functions:

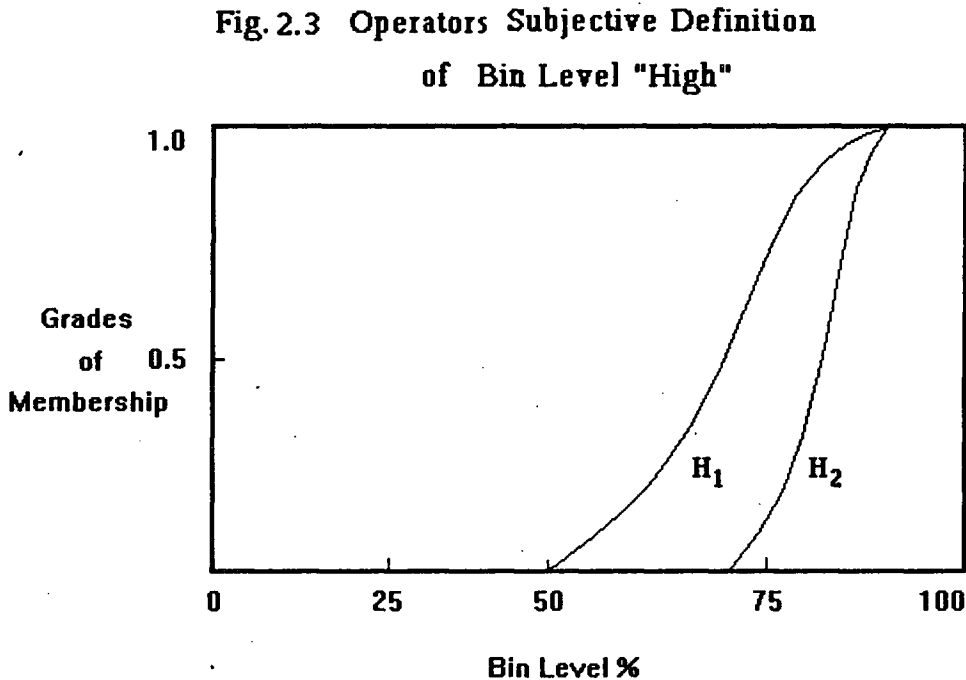
$$M_{H_1}(u) = S_{2pt_bell_up}(u; 50, 90)$$

or

$$M_{H_2}(u) = S_{2pt_bell_up}(u; 70, 90)$$

where $M_{H_1}(u)$ is given by one operator who expressed a wider range of belief in the concept "High" than the second operator who preferred a narrower definition $M_{H_2}(u)$.

Figure 2.3 shows a mapping of these membership functions.



Notice that $M_{H_1}(80) = 0.88$ whereas $M_{H_2}(80) = 0.5$. This has the interpretation that the first operator has 0.88 belief in the statement "80% bin level is high" versus 0.5 belief for the second. The first operator had a lower crossover point of 70 and wider support [50, 100] than the second with a crossover of 80 and support of [70, 100]. In general since, $M_{H_2}(u) \leq M_{H_1}(u)$ for all u in U , then definition H_2 is a subset of H_1 . Operator 1 might be viewed as a more experienced person than operator 2. On the other hand, operator 1 might be considered as less safe than operator 2.

Other shapes with their respective function representations are listed in Table 2.1. The critical points (a to f) in each function description are parameters which specify the lower and upper fuzzy subset interval boundaries. Details for these representations are shown in Appendix 1.

Table 2.1 Function Representations of Fuzzy Subsets

Shape Description	Function Description	Critical Points
Four Point Flat Bell	$S_{4pt_flat_bell}(u;a,c,d,f)$	a, c, d, f
Two Point Bell Up	$S_{2pt_bell_up}(u;a,c)$	a, c
Two Point Bell Down	$S_{2pt_bell_down}(u;d,f)$	d, f
Trapezium	$T_{trap}(u;a,c,d,f)$	a, c, d, f
Three Point Bell	$S_{3pt_bell}(u;a,c,f)$	a, c, f
Half Bell Down	$S_{half_bell_down}(u;d)$	d
Half Bell Up	$S_{half_bell_up}(u;c)$	c
Bell	$S_{bell}(u;c)$	c
Flat Bell	$S_{bell}(u;c,d)$	c, d

2.2.4. If...Then Rules

Fuzzy conditional statements or rules are constructs of the form:

If A then B.

Where A and B are fuzzy subsets. This states that if A is true then B is necessarily true also. It does not say anything about A, and B does NOT imply A, since B might be true without A by using some other rule in the system.

Using the previous control statement as an example:

'Whenever the bin level is "high" then make a "large" decrease in the feed rate'.

If we determined that the operator was making a large decrease in the feed rate, we cannot then deduce that the bin level was high. Other factors such as an over-loaded

conveyor may influence the decision to make a large decrease in the feed, even when the bin level is low.

2.2.5 Inferencing Ambiguities

Statements based on logic can be made even more complex by the use of the conjunctions AND and OR. AND is easier to understand, since OR is ambiguous, as it can have an inclusive or exclusive interpretation. From the statement:

'You can work the day or night shift'

OR has an exclusive interpretation when both choices are **not** allowed. If working a double shift is permitted then the OR becomes inclusive.

Further ambiguities arise when both AND and OR are used in the same statement. For example:

'if you are an experienced operator and a trainee was on the incoming shift or production was low then you must work overtime.'

There are at least two inferences possible from this statement, the first being:

'if you are an experienced operator and a trainee was on the incoming shift or if you are an experienced operator and production was low then you must work overtime.'

or alternatively:

'if you are an experienced operator and a trainee was on the incoming shift or production was low then you must work overtime.'

Only the person who formulated the policy can identify the correct interpretation. It has an ambiguous interpretation for an inexperienced operator working on a low production shift.

Fuzzy Logic allows us to overcome the constraints of crisp logic. From the rule

If (W AND X) OR Y then Z

we may not be able to make exact statements about the truth of W, X or Y. If we have subjective degrees of belief w, x and y for W, X and Y respectively, then we should be able to access z , the degree of belief of Z.

Using the respective degree of belief for each set we would have:

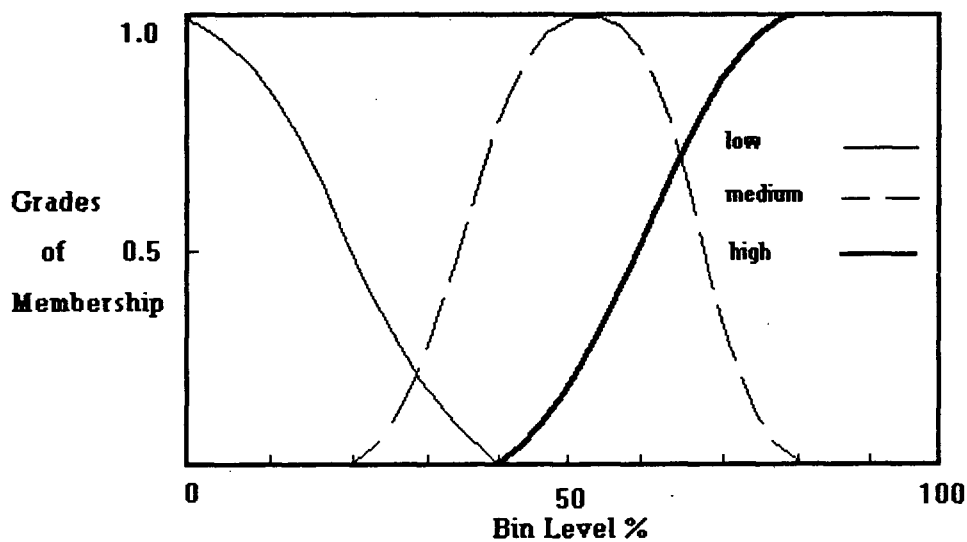
$$z = (w \wedge x) \vee y$$

The strength of belief in z depends on the weakest link in the logic and is independent of any possible inter-relationships between W, X and Y.

2.2.6 Creation of Parent Fuzzy Sets

Fuzzy set membership can represent linguistic values for a very simple grammar. For example let the domain of possible membership values for the screen bin be as illustrated in Figure 2.4.1:

Fig. 2.4.1 Fuzzy Subset Descriptions of Bin Levels



So:

$$\text{bin levels} = \text{low} + \text{medium} + \text{high}$$

and we can subjectively define the subsets low, medium and high as:

$$\begin{aligned}\text{low} &= 1.0/0 + 0.5/20 + 0/40 \\ \text{medium} &= 0/20 + 0.5/45 + 1.0/50 + 0.5/65 + 0/80 \\ \text{high} &= 0/40 + 0.5/60 + 1.0/80 + 1.0/100\end{aligned}$$

These definitions allows any membership within the universe of discourse to have a degree of belief in each subset. Hence a bin level of 60% implies zero belief in low, 0.95 in medium and 0.5 in high. These primary fuzzy set definitions can be extended to infer terms like "*not* high", "high or medium", etc. from discrete values.

The grammar can be extended by defining parent fuzzy sets to characterize a special case of set union. Qualifiers, defined in Table 2.2 are combined with subset arguments to create a parent set, *between* (low & medium), shown in Figure 2.4.2. The parent has a maximum grade of membership, of 1.0, from the supremum position of low to medium. This modification resolves the problem of membership grades, of say 30%, having unrealistic values of 0.6, had a normal union been defined. It therefore ensures that belief in the parent is maintained at a maximum between the supremum position of the child sets low and medium.

Fig.2.4.2 Creation of Parent Set *between* (low &medium)

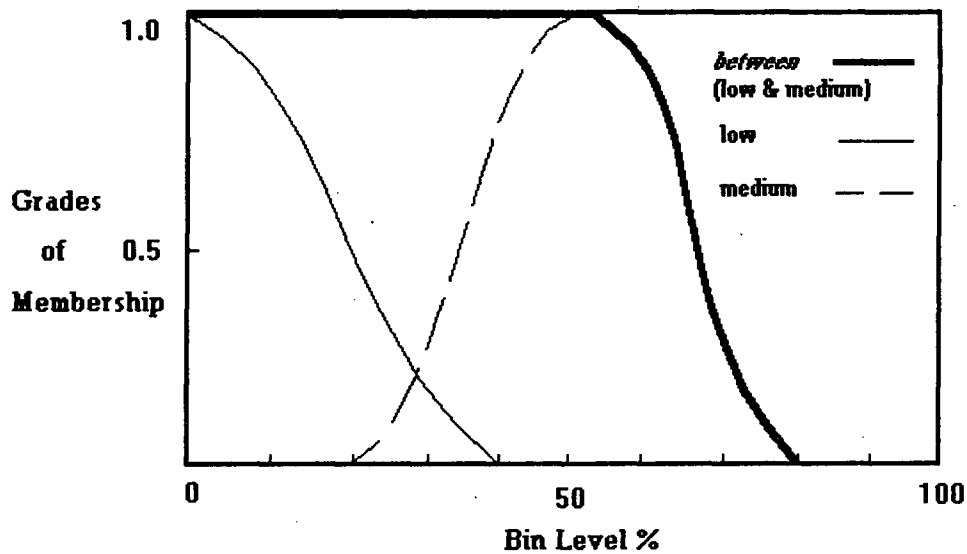


Table 2.2 Qualifiers and Description

Qualifier	Description
<i>any</i>	parent set from all subsets in domain
<i>not</i>	compliment of qualifying subset
<i>between</i>	parent set including subsets with supremum values greater than the first qualifying subset but less than the second subset inclusive.
<i>only</i>	only the specified subset is valid

The *any* qualifier requires no arguments since it pertains to all subsets of the domain. Following in Table 2.3 is a list of parent subsets with their expanded analogies in the bin level domain.

Table 2.3 Parent Sets in Bin level Domain

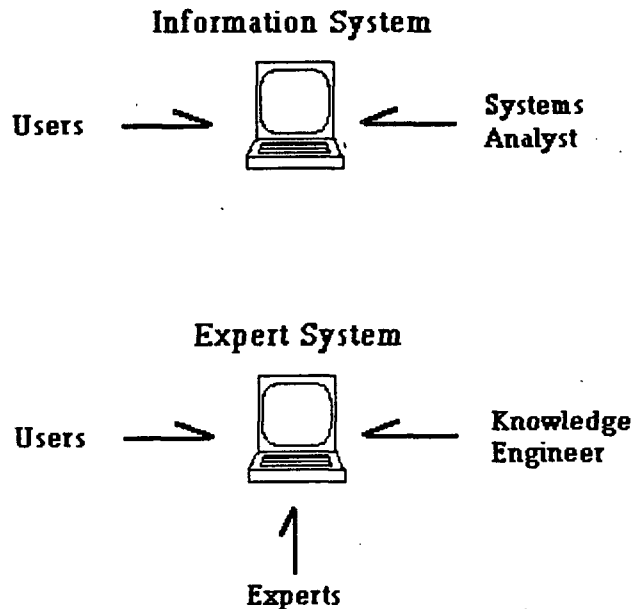
Parent Subset	Child Sets Included
<i>any</i>	low, medium and high.
<i>not low</i>	low'.
<i>between</i> (low & medium)	low and medium
<i>between</i> (medium & high)	medium and high

2.3. Traditional Construction Of Expert Systems

Greenwell (1988) refers to expert systems as "the limousines of the computer industry". This analogy is very appropriate since they are scarce, expensive, require excessive time for custom development, and on completion are often only suited to a "special taste". Several factors make the construction of these systems very different from conventional data processing or systems analysis.

Figure 2.5 shows the similarity and differences between traditional information systems and expert system development. The knowledge engineer has a similar role to the system analyst, and reconciles the needs of the users with the knowledge of the expert. Addressing who the users are and when and how they are involved in the development is critical. They must participate in both a proactive and reactive role.

Fig. 2.5 Information System Compared to Expert System



The common methods used in systems analysis of observing, sampling from existing records, interviewing, conducting group discussion, using questionnaires or forms are also common to knowledge acquisition. In systems analysis, it is fairly clear what information is required, but in knowledge acquisition, this is often not the case. Experts are not merely people who know lots of facts and procedures. They have built-up a body of knowledge through years of experience which they use to make informed and wise decisions. They often make judgements based on intuition without the need to explicitly state how they arrive at conclusions. Hayes-Roth et al (1983) formulated the stages in Table 2.4, for the development of expert systems. There is a strong similarity between these tasks and those used in systems analysis.

**Table 2.4 Stages and Tasks for Development of Expert Systems after
(Hayes-Roth, 1983)**

Stages	Tasks
Identification	Objectives of the expert system Who are the experts Who are the users Domain of knowledge
Knowledge Acquisition	Conceptualization of problem Knowledge extracted and represented in conceptual model
Design	Implementation of model Choice of structures representing Knowledge base
Development and testing	Design details implemented Cyclic implementing, evaluation and modification
Use	Initial cautious use. Continuous review and evaluation. Modification

Most authors agree that the major bottle-neck in the development cycle is knowledge acquisition. Hart (1989) lists some critical questions, shown in Table 2.5, that must be answered by the development team. Resolution of these issues cannot be classified by the common methods of fact-finding as in systems analysis. They are more related to the

domain of psychology, where information is often subjective, experiential, intuitive and perhaps social.

**Table 2.5 Issues for Resolution in Expert System
Building after (Hart, 1989)**

Conceptualization Issues	Detailed Breakdown of Issues
What are the inputs or problems? What are the outputs or solutions? Which types of inputs cause difficulties? How are the problems characterized? How are the solutions characterized? What sort of knowledge is used? How are problems or methods broken down into smaller units?	What data are input; and in what order? What are the relationships between data items? How important and accurate are the data items? What data might be missing? What assumptions can be made? What constraints exist? What sort of inferences can be made? How are the concepts and hypothesis formed? How do these relate to each other? How does the expert move from one state of belief to another? What evidence suggest particular goals or concepts? What are the causal relationships? Which problems are easy, common, hard, interesting, etc.?

2.4. Verification of Knowledge Base

Nazareth (1988) claims that there are inherent difficulties in demonstrating correct performance of expert systems and outlines some of the drawbacks. These include ill-defined or missing specifications, inapplicability of conventional verification and validation techniques. There is lack of agreement on evaluation methodologies, and low priority is often associated with the evaluation of system performance. Errors normally encountered in the development of rule-based systems take the form of redundancy, conflict, circularity and incompleteness of coverage. These are summarized as follows:

Redundancy occurs when combinations of premises and conclusions are adequately represented in other rules. It is likely to occur when such sets of rules are deleted,

modified or enhanced. It can also arise in chained inferencing where intermediate deduction can not be ascertained outside this chain of inference.

Conflict occurs when contradictory facts are deduced from a given set of information, or when facts deduced are in disagreement with the original information set. This can be introduced through the expansion of system knowledge to cover specific cases which may be in disagreement with more general rules.

Circular logic occurs when rules create some form of self reference, either directly or through chained inference. This can arise through an inability to distinguish between premise and conclusion statements among facts.

Incompleteness or gaps in the rule set, may be intentional. They represent restrictions in problem scope, and can also arise through oversight. In any case, the gaps translate into a system with pertinent facts about the domain omitted from the analysis, or an inadequate set of rules to manipulate relevant facts. They then show up as dead-ends in the chain of inferencing, unreachable goals or separable or unconnected segments of a rule set.

3.0 AUTOMATED ACQUISITION AND REPRESENTATION

Greenwell (1988) gives a definition of knowledge acquisition as the collection and analysis of information from one or more domain experts and any other sources leading to the production of a number of documents which form the basis of a functioning knowledge-base. He further states that a substantial subset of knowledge acquisition is knowledge elicitation. Knowledge elicitation, he defines, as that area of knowledge acquisition which deals with obtaining information directly from domain experts.

Most authorities acknowledge that knowledge elicitation is the major bottleneck, but there is debate on the usefulness or effectiveness of automated acquisition and representation techniques (Hart, 1989; Greenwell, 1988; Hall et al, 1990).

Interactive knowledge acquisition aids have been designed to automate the knowledge acquisition process. Hall et al (1990) suggest that these aids are only used by knowledge engineers to produce useful results in cases where the relationships (rules) are straightforward. They further suggest that knowledge acquisition aids present a restricted environment in which to work and this creates considerable duress for the user who has to express or use knowledge in a confined atmosphere. They conclude that since knowledge acquisition (engineering) is in its infancy, this poses a fundamental problem with attempts that are been made to automate the acquisition process for which very little is understood.

There are three main thrusts predominating automatic acquisition: induction, repertory grids and documentation systems. Greenwell (1988) claims that all three approaches have produced little in the way of useful tools for two main reasons. Firstly, the various knowledge representation languages and environments are usually incompatible with one another, so that a knowledge acquisition tool is usually dependent on a particular environment which restricts the market for such a product. Secondly the inferential

power of most acquisition systems is particular suspect and lacks the "Taken-for-granted" common sense which is inherent in intelligent humans as they come to understand the expert's tasks and domain.

3.1 Rule Induction

Ford et al (1991) defines induction, as the production of universal rules by generalizations based on a finite number of pieces of evidence. If the premises are true, it is merely probable that the conclusion is also true.

3.1.1 Deduction

In deduction, if we are given a set of rules or facts, we can deduce others. For example we might be told that:

Only experienced operators can achieve the daily target of 5,000 tpd.

So if we know that:

Joe's average production was 7,000 tpd for the last week

we can deduce that:

Joe is an experienced operator.

This reasoning from general to specific is a top-down approach where once the rule is true and we have made no false assumptions then the deduction will be true.

3.1.2 Induction

Induction works in the reverse, bottom-up mode where if we had several examples (instances) and did not know the rules, then we can make sensible guesses to induce rules which describe the examples. If the rules are good, they will apply not only to the

specific examples but also in general. Suppose we were given the following information in Table 3.1 on operators' tones per day (tpd) production:

Table 3.1. Operators Average Daily Production

Operator	Production tpd	Experienced?
Joe	7000	yes
Bob	6000	yes
Simon	2000	no
Chris	3000	no

Using the VP-Expert (1987) approach which induces rules which conclude on the left-most field. We would induce the following rule;

If an operator can achieve 6,000 tpd then he is experienced.

The critical production target could be 4,000 or 5,000 tpd. The uncertainty in the induced rule is caused by the incompleteness of the example set. Since we had no examples at the 5,000 tpd production level then the correct cut off point which differentiated inexperienced from experienced operators could not be determined. The utility of computers are demonstrated, when large examples sets are given. They can easily deal with logic, arithmetic and pattern matching routines to induce rules.

3.1.3 Requirements for Induction

Hart (1989) lists the prerequisites for induction as the following:

Examples - the examples or training set form the basis of the induction process. An incomplete or inadequate set of examples is likely to result in poor rules. In the problem discussed above, the training set is the set of data for each operator.

Premise - The examples have sets of characteristics which describe them, and enable comparisons to be made between different examples. Some of these characteristics form the factors which influence the rules. Examples of these characteristics are tabulated as follows:

Table 3.2 Types of Premise with Examples

Characteristics	Examples
descriptive categories	happy, sad, merry
descriptive fuzzy sets	low, medium, high
real measurements	height in meters
integer values	age in years
logical descriptions	yes, no, true, false
special numbers	dates, serial numbers

In the operator production case above, the only premise is the integer values of daily production (tph).

Conclusions - The conclusions represent the decision or classification by the expert, e.g. the operator is experienced.

Inductive Algorithm - The algorithm is the method which the computer program uses to induce rules from the training set. It will depend on the form of the examples and the required output.

The quality of the induced rules depends on all of these components and the training set, which must be rich enough to contain enough examples with sufficient descriptors to enable quality rules to be induced.

Hart surmises that inductive algorithms designed to cope with specific types of domain knowledge, have been more successful than general ones which tend to be less efficient and produce results which are less spectacular. She argues that computers are objective, without preconceived ideas. Experts on the other hand cannot always see a simple pattern or principle and suggest overly complex rules, when they describe what they ought to do, rather than what they actually do. She contends that research in rule devising has shown that a computer program can induce neater and simpler rules than people can.

3.2 Personal Construct Theory

Personal construct theory, as formulated by the psychologist Kelly (1955) assumes that people typically use cognitive dimensions termed "*constructs*" to evaluate their experience. Each construct: "hot-cold", "happy-sad" by definition represents a single bipolar distinction. This method implies that no construct could be understood without considering the meaning of both its poles. Hence the construct then has a specific *range of convenience* which constitutes all the things to which the user would find the application useful. The model matches our view of experts at work since any particular construct may have a somewhat different context for each person who uses it.

Kelly then postulates that a necessary condition for organized thought is some degree of overlap between constructs in terms of their respective ranges of convenience. This overlap between the ranges of convenience then enables an event to be anticipated.

3.2.1. Repertory Grid Analysis

The repertory grid is derived from personal construct theory and provides a tabular representation of a person's view of a particular problem. The grid is composed of constructs and elements which are similar to the premises and examples described in induction. Construct, as explained above, are bipolar characteristic which can be represented on a linear scale. Hence while "up-down", "light-heavy" are good examples of constructs, "shape" is not, since there are many, perhaps too many, ways to define it, which makes the construction of bipolar characteristics difficult for curves, lines, inflections, etc. The expert must understand what makes a valid construct and how it is used. Rating could be described by names rather than numbers, although subsequent analysis normally uses numerical values. For example on a scale of 1-5 with a construct low-high suitable descriptions could be:

- | | |
|---|-----------|
| 1 | very low |
| 2 | low |
| 3 | average |
| 4 | high |
| 5 | very high |

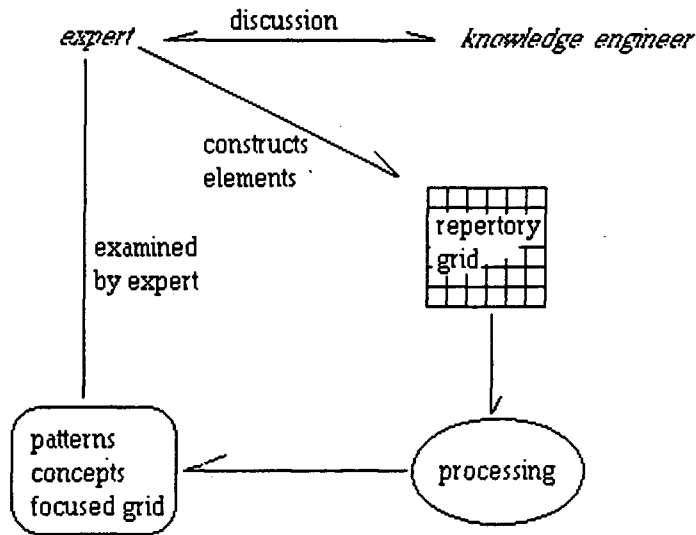
Each construct must have the same scale throughout, although terms may be used to describe particular ratings which could vary from construct to construct.

Elements are key examples, produced by the expert, and described by the values of the constructs. Each element is rated according to each construct, using a subjective rating, similar to the fuzzy logic, supplied by the expert. For example, suppose the construct is production level "very high - very low" on a scale of 1-5. If the day shift is rated 5 and the night shift is 4 and the weekend shift is 2, this implies that the production level for the day shift is higher than the night shift which is in turn higher than the weekend shift. However it does not imply that the night shift production is twice the day shift.

The development cycle for a grid is illustrated in Figure 3.1. It provides a cross referencing system between the elements and constructs similar to the training set used in induction. It is therefore useful to the knowledge engineer for two reasons:

- 1) The elicitation of a grid makes the expert think in a more structured way about a problem, and helps to clarify issues in the mind.
- 2) Standard analytical techniques are available to analyse the grid to find patterns or associations for further investigation. This is especially useful at the earlier overview or conceptual level in elicitation. Larger grids describing greater detail quickly become unmanageable.

Fig. 3.1 Construction of Repertory Grid after (Hart, 1989)



3.2.3. Grid Elicitation

Before drawing up a grid it is important to define the problem under investigation. Here the expert or elicitor must state clearly what is the objective of the analysis.

Having defined the objective the expert sets about the task of drawing up the grid. In the simplest cases the elements and constructs are obvious hence the grid can be analyzed immediately. However in the majority of cases this elicitation is non-trivial and involves repeated comparison, changing rating scales, or introduction or removal of elements and constructs until the expert is satisfied that the grid describes his view of the problem under investigation.

An example grid is elicited in Table 3.3 to classify operators by their skills. Possible constructs are shown for "experienced - inexperienced", "high - low" ratings for average tonnage and "long service - recent" for years of service, "slow - fast", "conservative - liberal" etc.

Table 3.3 **Repertory Grid Comparing Operators in A Plant**
rating scale (1-3)

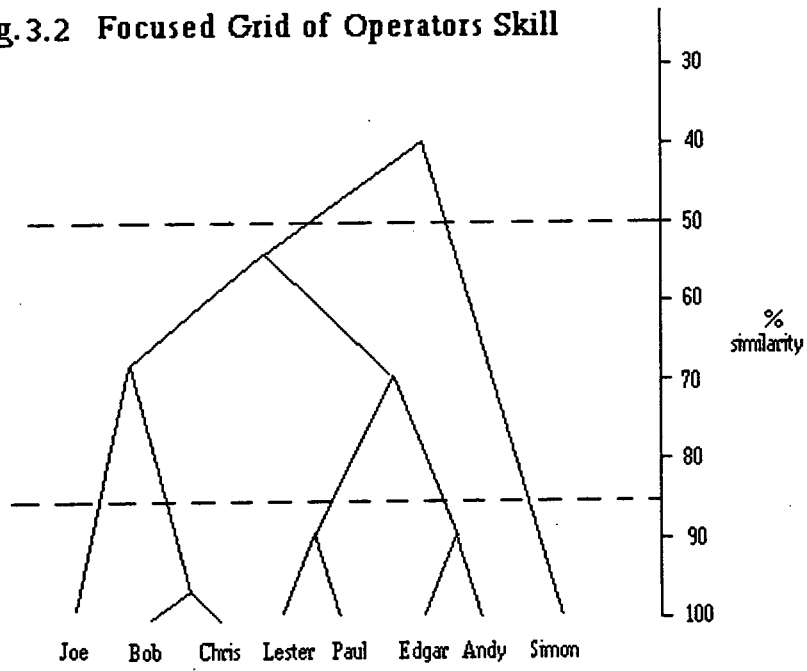
Constructs	Elements								Constructs
	Andy	Bob	Chris	Edgar	Joe	Lester	Paul	Simon	
experienced	2	1	1	2	1	1	1	3	inexperienced
high	3	1	2	2	1	2	2	3	low
long service	2	2	2	2	1	3	3	3	recent
slow	2	3	3	2	3	2	1	1	fast
conservative	2	3	2	3	3	3	2	1	liberal

3.2.4. **Grid Analysis**

Once the grid has been elicited, as shown in the example of Table 3.3, it can be analyzed. The analysis is intended to exhibit structure and pattern in the grid and therefore provide feedback to the expert. The analysis is a tool to help the expert identify structures which are recognized as useful. Grid analysis is based on methods used in statistical theory, which include factor analysis, principal component analysis or cluster analysis. The results of the analysis enable attributes and elements to be reordered so that similar objects appear near to each other in a focused grid.

An example of a focused grid which group the operators is shown in Figure 3.2. Each operator is an element. Clustering will show the grouping of operators depending on the level of similarity. Therefore Bob and Chris, are most similar while Simon who may have recently joined the crew and achieved low production totals will show on the grid as being very unlike any of the others. At a level of 85% similarity there would be five groups; at a level of 50% there would be two groups.

Fig.3.2 Focused Grid of Operators Skill



4.0 ACQUISITION AND REPRESENTATION METHODS

The acquisition methodologies investigated, in this study, borrows from the induction and repertory grid approaches. The knowledge domain is comprised of:

- 1) a reference database and
- 2) fuzzy expressions commonly used in that domain.

The reference database represents some desired relationship or historical trend similar to examples sets in induction. Fuzzy expressions give meaning to concepts when used in rules. This allows discrete information to be interpreted as a quantifiable degrees of belief in concepts.

4.1 Knowledge Acquisition Technique

To demonstrate acquisition and representation techniques, a reference database in Table 4.1, along with fuzzy subset descriptions in Table 4.2 were chosen. Rules are induced from this discrete information from the subset descriptions. Examples which follow in the remainder of this chapter illustrate the technique.

Table 4.1 Reference Database of Z Values

		X Variable										
		0	10	20	30	40	50	60	70	80	90	100
Y Variable	0	-100	-66.7	-40	-20	-6.7	0	6.7	20	40	66.7	100
	10	-90	-60	-36	-18	-6	0	6	18	36	60	90
	20	-60	-40	-24	-12	-4	0	4	12	24	40	60
	30	60	40	24	12	4	0	-4	-12	-24	-40	-60
	40	90	60	36	18	6	0	-6	-18	-36	-60	-90
	50	100	66.7	40	20	6.7	0	-6.7	-20	-40	-66.7	-100
	60	90	60	36	18	6	0	-6	-18	-36	-60	-90
	70	60	40	24	12	4	0	-4	-12	-24	-40	-60
	80	-60	-40	-24	-12	-4	0	4	12	24	40	60
	90	-90	-60	-36	-18	-6	0	6	18	36	60	90
	100	-100	-66.7	-40	-20	-6.7	0	6.7	20	40	66.7	100

Table 4.2 Membership Functions for X,Y and Z Subsets

Variable	Subset Name	Shape	Critical Point
X	low_x	half_bell_down	0
	mid_x	bell	50
	high_x	half_bell_up	100
Y	low_y	half_bell_down	0
	mid_y	bell	50
	high_y	half_bell_up	100
Z	low_z	half_bell_down	-100
	mid_z	bell	0
	high_z	half_bell_up	100

4.1.1. Reference Data

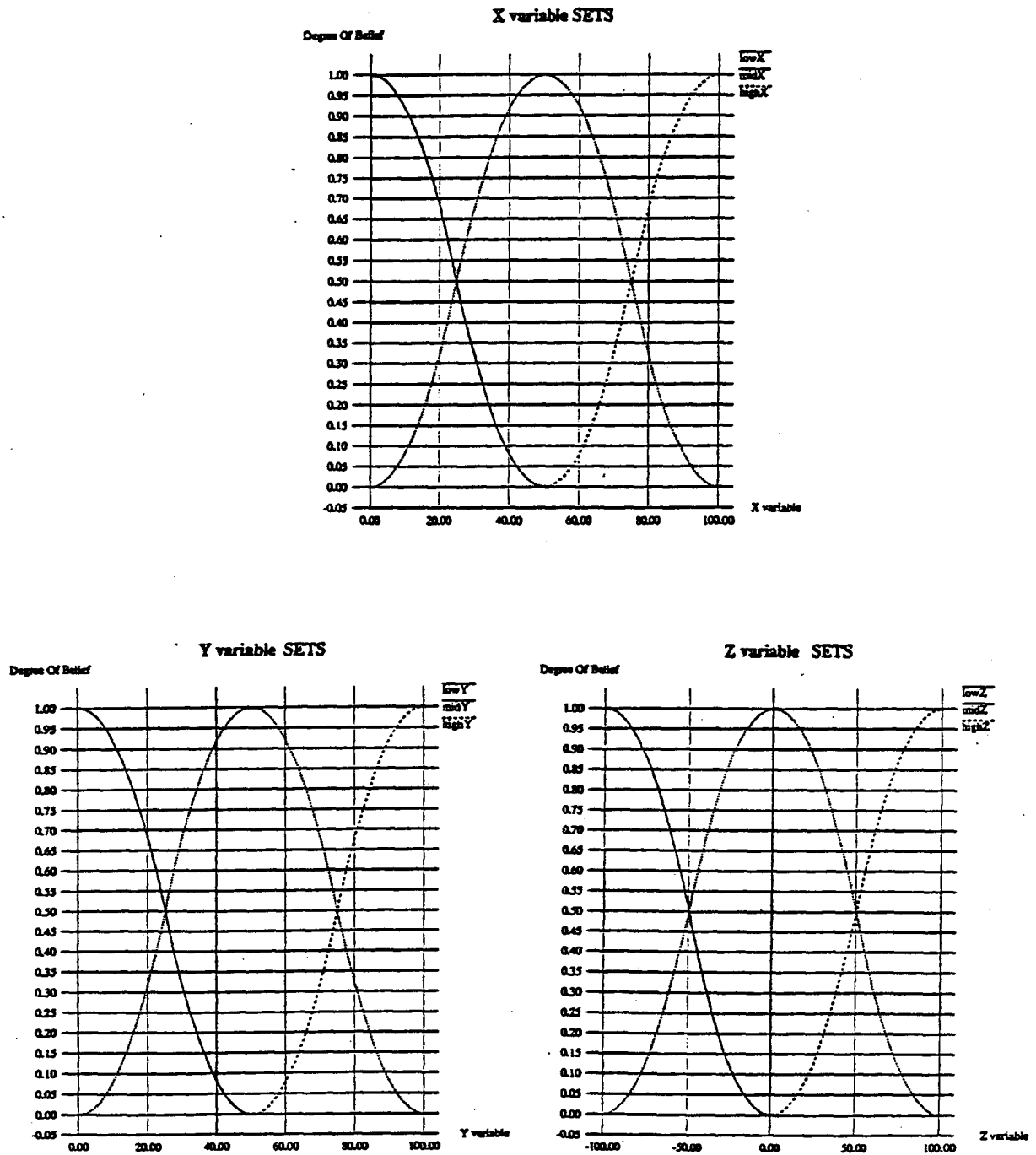
The reference database is symmetrical, therefore the X and Y variables have an identical domain of [0, 100]. The body of the table is comprised of the Z variable in an [11 x 11] array, within the domain [-100, 100].

4.1.2. Definition of Fuzzy Subsets

Three fuzzy expressions; high, medium, and low describe each variable. Each expression has a membership function description, characterized by a shape and by critical points. Membership functions are computed from standard definitions, previously described in Section 2.2.3. The functions listed in Table 4.2 are illustrated graphically in Figure 4.1. where the subsets align themselves in a sequence which guarantees;

- 1) completeness in the domain
- 2) a smooth transition of belief in one concept or subset to another as discrete values change incrementally.
- 3) non-conflict since it is not possible, for any discrete x to have greater membership in either of the end sets, **low_x** and **high_x**, than the intermediate **mid_x**. This is achieved by assigning the supremum position of an adjacent set to the end point of the set being defined.

Fig. 4.1 **Membership Functions Generated for X, Y and Z Subsets**



4.2. Locating Fuzzy Subsets on Reference Data

The X, Y and Z subsets are mapped, at their critical points, onto the referenced data. This is analogous to a transformation of discrete values, in a Cartesian system, to a fuzzy plane or description shown in Table 4.3.

Table 4.3 Mapping of X and Y Subsets in Fuzzy Plane with Cells Showing Discrete z Values

Set Name Discrete Value Discrete Member		X Domain		
		low_x	mid_x	high_x
		0 x_0	50 x_5	100 x_{10}
Y Domain	low_y 0 y_0 }	$z_{0,0}$ -100	$z_{5,0}$ 0	$z_{10,0}$ 100
	mid_y 50 y_5 }	$z_{0,5}$ 100	$z_{5,5}$ 0	$z_{10,5}$ -100
	high_y 100 y_{10} }	$z_{0,10}$ -100	$z_{5,10}$ 0	$z_{10,10}$ 100

4.2.1 Location of X and Y Subsets

The defined X and Y subsets are located by positioning the critical points where they coincide with the respective x and y values of the reference data. For example, the **low_x** subset with critical point at 0, is located at position 0 ($x_0=0$) in the ordinate field of the reference data. In a similar way, the **mid_y** bell is located at position 5 ($y_5=50$), within the abscissa field.

4.2.2. Location of Z Subsets

The location of X and Y subsets at their respective (i,j) positions are used to extract discrete z_{ij} values, shown in Table 4.3, from the reference data. These values could be interpreted by the example;

When we are most certain that x is low (**low_x** at $x_0=0$) and y is medium (**mid_y** at $y_5=50$) then $z_{0,5}=100$.

z values are transformed to the fuzzy plane, for the universe of Z subsets. These are displayed in Table 4.4 where for the example above at ($x_0=0$, $y_5=50$), **high_z** had full membership while **mid_z** and **low_z** had zero membership.

Table 4.4 Mapping of Subset in Fuzzy Plane, Cells Show Grades of Membership for Z Subsets

		X Domain			
		low_x	mid_x	high_x	
		0 x_0	50 x_5	100 x_{10}	
Y Domain	low_y 0 y_0 }	1	0	0	low_z
		0	1	0	mid_z
		0	0	1	high_z
	mid_y 50 y_5 }	0	0	1	low_z
		0	1	0	mid_z
		1	0	0	high_z
	high_y 100 y_{10} }	1	0	0	low_z
		0	1	0	mid_z
		0	0	1	high_z

4.3 Selecting Dominant Subsets

Each cell of Table 4.4, comprising of a universe of Z subsets, is considered separately. The subset with the maximum membership is selected. The result is a full symbolic representation in Table 4.5 of the initial reference data. As can be seen, the reference data, consisting of 121 elements, is reduced to a fuzzy mapping of 9 rules.

Table 4.5 Dominant Fuzzy Subsets

		X		
		low_x	mid_x	high_x
Y	low_y	low_z	mid_z	high_z
	mid_y	high_z	mid_z	low_z
	high_y	low_z	mid_z	high_z

Examples of rules deduced by this process are;

IF low_x AND low_y
THEN low_z

or

IF high_x AND low_y
THEN high_z

4.4. Minimizing The Number Of Rule Descriptions

4.4.1 Defining parent fuzzy sets

Additional reduction of the symbolic representations in Table 4.5 is achieved by defining parent subsets. For example three rules conclude on a **mid_z** subset. The *any* qualifier replaces the three premise statements;

IF low_y AND mid_x
OR
IF mid_y AND mid_x
OR
IF high_y AND mid_x
THEN mid_z

with the single statement;

IF any Y AND mid_x
THEN mid_z

A parent subset (*any Y*) is created from the universe of Y subsets. This is analogous to specifying independence of Y, since any discrete y value has full membership. Stated explicitly by;

$$M_{anyY}(y)=1 \quad \text{for } (y \in Y)$$

4.4.2 Reduction by *not* Qualifier

A second pattern in Table 4.5 can be described by the *not* qualifier, which creates a complementary set. The **not** qualifier, by specifying exclusion, reduces the description from two premise statements:

```
IF      low_y      AND  low_x
OR
IF      high_y     AND  low_x
THEN   low_z
```

to a single premise:

```
IF      not mid_y   AND  low_x
THEN   low_z
```

4.4.3. Final Rule Description

The induction system imposes a consistent format, in Table 4.6, of the final rules. A rule is therefore composed of a qualifier and a subset combined to form a condition. The X and Y conditions are 'ANDed' for a unique premise statement. Premise statements are 'ORed' to conclude on a specified Z subset.

Table 4.6 Final Rules

Y		X	Z
<i>not</i>	mid_y	low_x	low_z
	mid_y	high_x	
<i>any</i>		mid_x	mid_z
<i>not</i>	mid_y	high_x	high_z
	mid_y	low_x	

The 3 final rules (5 premise statements) of Table 4.6 then represent the 9 rules of Table 4.5 which in turn represent the 121 data points of Table 4.1. The accuracy of this representation can be determined, first by generation of discrete data from the rules, followed by a comparison with the reference data.

4.5. Defuzzification

The process of generating data from rules composed of fuzzy sets is referred to as defuzzification. First the respective grades of membership are determined for concluding Z subsets. This is illustrated using rules from Table 4.6 and given the conditions:

$$x_2 = 20 \quad \text{and} \quad y_1 = 10$$

The respective grade of membership for the X and Y subsets can be determined from the membership functions in Figure 4.1 where;

$$\begin{array}{lll} M_{\text{low}_x}(20)=0.68 & M_{\text{mid}_x}(20)=0.32 & M_{\text{high}_x}(20)=0.0 \\ M_{\text{low}_y}(10)=0.92 & M_{\text{mid}_y}(10)=0.08 & M_{\text{high}_y}(10)=0.0 \end{array}$$

We can deduce the grades of membership for each Z subset from each rule where;

From rule 1:

$$\begin{aligned} M_{\text{low}_z}(z_{2,1}) &= (\text{mid}_y' \wedge \text{low}_x) \vee (\text{mid}_y \wedge \text{high}_x) \\ &= (0.92 \wedge 0.68) \vee (0.08 \wedge 0.0) \\ &= 0.68 \quad \vee \quad 0.0 \\ &= 0.68 \end{aligned}$$

Similarly from rule 2:

$$\begin{aligned} M_{\text{mid}_z}(z_{2,1}) &= (\text{any } Y) \wedge (\text{mid}_x) \\ &= 1.0 \wedge 0.32 \\ &= 0.32 \end{aligned}$$

Finally from rule3:

$$\begin{aligned} M_{\text{high}_z}(z_{2,1}) &= (\text{mid}_y' \wedge \text{high}_x) \vee (\text{mid}_y \wedge \text{low}_x) \\ &= (0.92 \wedge 0.0) \vee (0.08 \wedge 0.68) \\ &= 0.0 \vee 0.08 \\ &= 0.08 \end{aligned}$$

Several methods were examined for the final computation of discrete z output namely:

- 1) Weighted Average
- 2) Use of Confidence Level
- 3) Use of Saddle Point
- 3) Maximum Value

4.5.1 Weighted Average

The weighted average method requires only the grades of membership and the supremum position of each concluding Z subset. Discrete $z_{2,1}$ values are computed from the weighted average relation;

$$z = \frac{\sum_{k=1}^{k=t} M_k \text{Sup}_k}{\sum_{k=1}^{k=t} M_k}$$

Substituting membership and supremum values from the example above gives:

$$z = \frac{0.68x - 100 + 0.32x0 + 0.08x100}{0.68 + 0.32 + 0.08} = -55.6$$

4.5.2 Confidence Level

This method introduces an additional scaling factor or confidence level below which we ignore any evidence. The confidence level is therefore some limiting grade of membership above which we are prepared to consider some proposition and below which it is ignored.

Discrete $z_{2,1}$ values are computed from the weighted average relation;

$$z = \frac{\sum_{k=1}^{k=t} M_k Sup_k}{\sum_{k=1}^{k=t} M_k} \quad \text{for all } M_k \geq C$$

If we assume a 0.1 Confidence Level and substitute membership and supremum values from the example above, the output value will be:

$$z = \frac{0.68 \times -100 + 0.32 \times 0}{0.68 + 0.32} = -68.0$$

4.5.3 Saddle Point

This method improves on the Confidence Level method by including the influence of disbelief in a concept into the computation of discrete z . The saddle point is therefore a limiting grade of membership about which we measure both belief and disbelief to compute the z values from the relation;

$$z = \frac{\sum_{k=1}^{k=t} (M_k - S) Sup_k}{\sum_{k=1}^{k=t} |(M_k - S)|}$$

By definition the Saddle Point is 0.5, so by substituting membership and supremum values from the example above, the discrete output value will be:

$$z = \frac{(0.68 - 0.5) \times -100 + (0.32 - 0.5) \times 0 + (0.08 - 0.5) \times 100}{(0.68 - 0.5) + (0.5 - 0.32) + (0.5 - 0.08)} = -76.9$$

4.5.4 Maximum Value

The method of maximum membership selects the output subsets with the highest grade of membership. The supremum value of the selected membership function gives discrete z values as in the example;

$$z = \text{Sup}_k$$

Substituting supremum value for the example above gives:

$$z = -100.0$$

4.5.5 Selection of Defuzzification Method

The Maximum Membership method is the least complex in its computation. It produced "spiky" output for incremental changes in X and Y variables. This is especially evident as variables change between the boundaries of membership function definitions. Special consideration is required when two or more fuzzy subsets have equal and maximum grades of membership.

The Weighted Average method was selected because of its simplicity and effectiveness. It resolved some of the problems found in the Maximum Membership method. Smoother output is generated because the averaging technique considers relative weights of all the subsets of the Z domain. No special consideration is necessary for membership functions

with equal grades of membership. This method is a special case of the Confidence Level method where a zero confidence level is implied. It is used to generate the final defuzzification data in Table 4.7 from the final rules in Table 4.6.

The Confidence Level approach is also very effective. It produced similar results to the Weighted Average method. The confidence level parameter provides the flexibility to bias the defuzzification output towards more conservative values at lower confidence level settings. The method requires special consideration only when grades of membership for all subsets lie below the confidence level selected. When these grades fall below the specified confidence level, this creates an incomplete condition or "vacuum", where no conclusions can be reached. Meech and Jordon (1990) used this method effectively with the maximum method when the confidence level was too high to prevent "vacuums" in the knowledge.

Table 4.7 Generated Data by Weighted Average Method

		X Variable										
		0	10	20	30	40	50	60	70	80	90	100
Y Variable	0	-100	-92	-68	-32	-8	0	8	32	68	92	100
	10	-84	-77.8	-55.6	-22.2	0	0	0	22.2	55.6	77.8	84
	20	-36	-33.3	-27.3	0	0	0	0	0	27.3	33.3	36
	30	36	33.3	27.3	0	0	0	0	0	-27.3	-33.3	-36
	40	84	77.8	55.6	22.2	0	0	0	-22.2	-55.6	-77.8	-84
	50	100	92	68	32	8	0	-8	-32	-68	-92	-100
	60	84	77.8	55.6	22.2	0	0	0	-22.2	-55.6	-77.8	-84
	70	36	33.3	27.3	0	0	0	0	0	-27.3	-33.3	-36
	80	-36	-33.3	-27.3	0	0	0	0	0	27.3	33.3	36
	90	-84	-77.8	-55.6	-22.2	0	0	0	22.2	55.6	77.8	84
	100	-100	-92	-68	-32	-8	0	8	32	68	92	100

4.6. Error Determination

The rule generated output in Table 4.7, is compared to the reference data in Table 4.1 by indicators which are listed in an error report of Table 4.8. Four statistical indicators are computed in Appendix 2 and listed below namely;

- 1) Difference between elements
- 2) Sum of the squares difference
- 3) Distribution of sum of squares error
- 4) The correlation coefficient

Table 4.8 Error Report with Difference Values

		X Variable												Sum of	Error
		0	10	20	30	40	50	60	70	80	90	100	Squares	Dist	
Y Variable	0	0	25.3	28	12	1.3	0	-1.3	-12	-28	-25.3	0	3140	14	
	10	-6	17.8	19.6	4.2	-6	0	6	-4.2	-19.6	-17.8	6	1577	7	
	20	-24	-6.7	3.3	-12	-4	0	4	12	-3.3	6.7	24	1582	7	
	30	24	6.7	-3.3	12	4	0	-4	-12	3.3	-6.7	-24	1582	7	
	40	6	-17.8	-19.6	-4.2	6	0	-6	4.2	19.6	17.8	-6	1577	7	
	50	0	-25.3	-28	-12	-1.3	0	1.3	12	28	25.3	0	3140	14	
	60	6	-17.8	-19.6	-4.2	6	0	-6	4.2	19.6	17.8	-6	1577	7	
	70	24	6.7	-3.3	12	4	0	-4	-12	3.3	-6.7	-24	1582	7	
	80	-24	-6.7	3.3	-12	-4	0	4	12	-3.3	6.7	24	1582	7	
	90	-6	17.8	19.6	4.2	-6	0	6	-4.2	-19.6	-17.8	6	1577	7	
	100	0	25.3	28	12	1.3	0	-1.3	-12	-28	-25.3	0	3140	14	
Sum of Squares		2448	3362	3925	1079	213	0	213	1079	3925	3362	2448	22054		
Error Dist.		11	15	18	5	1	0	1	5	18	15	11	100		
Correlation Coefficient = 0.964															

4.7. Error Reduction Strategies

Several strategies were studied to improve the closeness of fit between the rule generated data and the reference data. Increasing the number of subsets describing each domain generally had the effect of achieving higher correlation coefficients. This result was expected, since a larger number of subsets constitute greater precision in defining variables, hence more exact rules should be created. Two strategies were examined with:

- 1) single subset additions to each variable
- 2) relocation of subsets
- 3) subset additions to areas of greatest error

4.7.1. Adding a Single Set to Each Variable

This strategy begins with two subsets defining each variable. If these produce unacceptable correlations on rule generation, each domain is discretized more finely by increasing the number of subsets. Membership functions, for the X variable subset as defined in Table 4.9 at each stage, are shown in Figure 4.2. (a), (b) and (c).

Table 4.9 Stages for Adding Single Subsets to the X Variable

Stage	Subset Name	Shape	Critical Points	Function Definitions
1	low_x	half_bell_down	0	$S_{\text{half_bell_down}}(x;0)$
	high_x	half_bell_up	100	$S_{\text{half_bell_up}}(x;100)$
2	low_x	half_bell_down	0	$S_{\text{half_bell_down}}(x;0)$
	mid_x	bell	50	$S_{\text{bell}}(x;50)$
	high_x	half_bell_up	100	$S_{\text{half_bell_up}}(x;100)$
3	low_x	half_bell_down	0	$S_{\text{half_bell_down}}(x;0)$
	mid_low_x	bell	33.3	$S_{\text{bell}}(x;33.3)$
	mid_high_x	bell	66.7	$S_{\text{bell}}(x;66.7)$
	high_x	half_bell_up	100	$S_{\text{half_bell_up}}(x;100)$

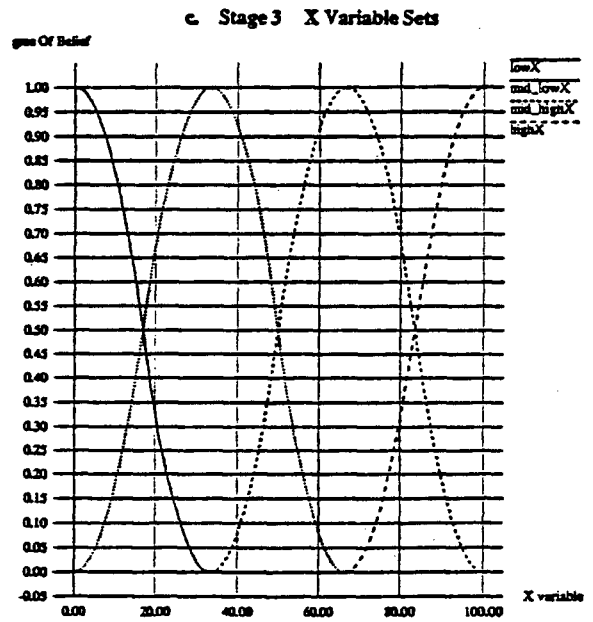
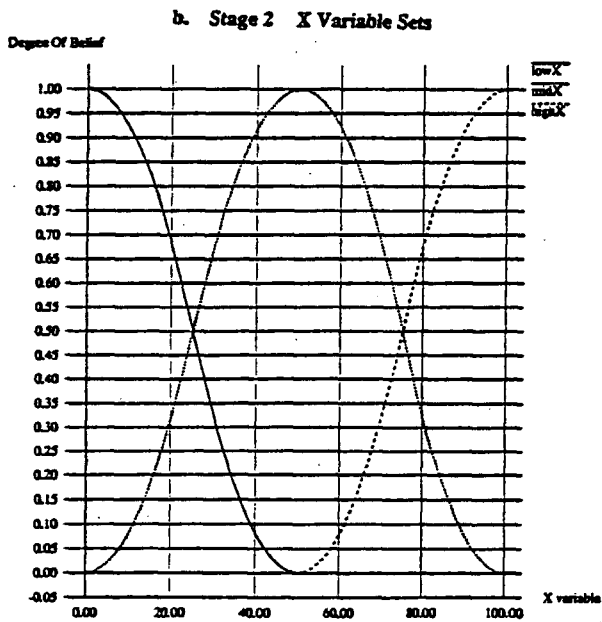
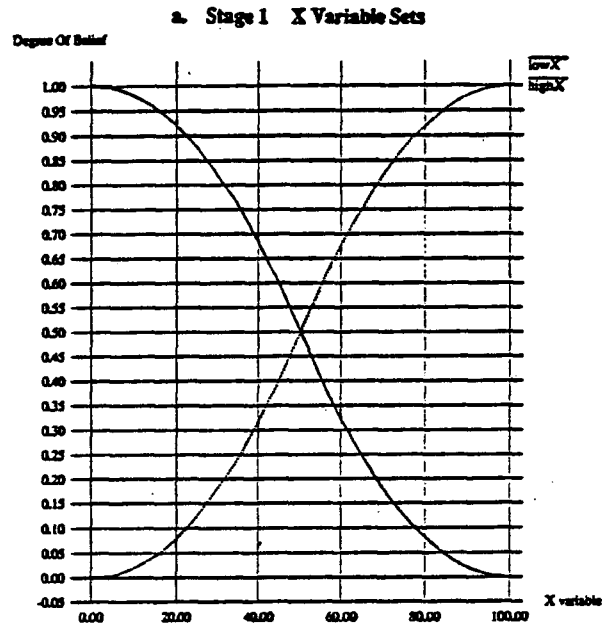
The process begins with two initial subsets for each of the X, Y and Z variables. The lowest set member is a half_bell_down shape with a critical point at the minimum value of the domain. The highest subset, is a half_bell_up with critical point at the maximum of the domain. Subsequent additions are bell-shaped subsets between the lowest and highest half_bell shapes with the critical points defined by the relation;

$$\text{Critical points} = \frac{\text{Width of Domain}}{p-1} \quad \text{for } p=2 \text{ to } n$$

where; n = number of subsets in domain

Rules are deduced along with an error report. If the error is unacceptable, then the number of sets for each variable is incremented by one. No change is made to the input definition of the lowest half_bell_down or highest half_bell_up subset definitions, although membership functions are altered to those shown in Figure 4.2 (b) and (c) for the X variable. Similar changes are implemented for the Y and Z variable at each stage.

Fig 4.2 Three Stages of Single Set Additions to X Variable



4.7.2 Relocating Subsets

This method of relocating existing subsets was especially useful in conjunction with the strategy of adding a single subset to each variable. The third stage automatic additions for the X variable shown in Table 4.10 could be applied to all three X,Y and Z variables as shown in Table 4.11. Error reduction may be improved by redefining the X and Y subsets so that their supremums (critical points) are precisely located at existing x_i and y_j values in the reference data. The Z subset can also be redefined so that the critical point of the new subsets lie on the corresponding z_{ij} values. However, the exact redefinition of the new z subsets may not be as simple as just described and each user may select different locations based on an examination of the neighboring discrete values.

This strategy could be implemented for example, to improve the correlation coefficient for subsets definitions, at the third stage shown in Table 4.10. Here the new subset definitions shown in Table 4.11, could be established by selecting exact values from the reference data in Table 4.1 that are close to the current automatically generated critical points. The corresponding Z subsets could be positioned at -12 and 12 as can be seen from Table 4.1. Alternatively a user might prefer to try critical points for X and Y of 20 and 80 respectively which would give Z subset critical points of -24 and 24. Even further examination of the data might lead to the relocation of the critical points to be somewhere between -30 to -40 and 30 to 40, since there are a number of neighboring points close to these ranges in the same X,Y vicinity.

A detailed study was not conducted on this technique. Further work is necessary to gain a better understanding of its application.

Table 4.10 Third Stage Subset Definitions for X, Y and Z Variables

Variable	Subset Name	Shape	Critical Points
X	low_x	half_bell_down	0
	mid_low_x	bell	33.3
	mid_high_x	bell	66.7
	high_x	half_bell_up	100
Y	low_y	half_bell_down	0
	mid_low_y	bell	33.3
	mid_high_y	bell	66.7
	high_y	half_bell_up	100
Z	low_z	half_bell_down	-100
	mid_low_z	bell	-33.3
	mid_high_z	bell	33.3
	high_z	half_bell_up	100

Table 4.11 Relocating X,Y and Z Subsets

Variable	Subset Name	Shape	Critical Point
X	mid_low_x	bell	30
	mid_high_x	bell	70
Y	mid_low_y	bell	30
	mid_high_y	bell	70
Z	mid_low_z	bell	-12
	mid_high_z	bell	12

4.7.3 Adding Sets to Area of Greatest Error

The strategy of adding subsets to areas of greatest error is more complex than the former method where a single subset is added to each variable. This method involves a more conservative and selective subset addition. Additions are strategically placed to eliminate error at the point of insertion. This invariably results in an increased correlation coefficient. The strength of this technique is based on the principal that any \tilde{z}_{ij} value of the error report can be reduced to zero by explicitly defining three subsets (for X, Y and Z) with critical points defined from the reference data, at $(x_i, y_j$ and $z_{ij})$. The rule

induced will always result in perfect correlation at the z_{ij} location. Addition of subsets therefore involves a careful analysis of the error report.

If the correlation coefficient is unacceptable (too low) then the distribution of the sum of squares error is examined for the (i,j) location of greatest error contribution. This error could be eliminated (reduced to zero) by three subset definitions (one for each variable) at this point. For example, from the error report shown in Table 4.8 the distribution of the maximum error for the Y variable is 14% at the y_0 , y_5 and y_{10} location. Similarly for the X variable the maximum error of 18% occurs at x_2 and x_8 . This example is complicated by the fact that there are multiple areas, for each variable, with equal and maximum error. The process, of determining the optimal sequence of subset additions which achieves the maximum error reduction is complex and needs further work. If we choose to eliminate the error at the $z_{2,5}$ location, this requires the addition of the following subsets shown in Table 4.12 to the original subsets definitions in Table 4.2. A third subset for the Y variable is not necessary since the **mid_y** subset at the y_5 location already exists.

Table 4.12 Additional Subsets for Error Reduction

Variable	Subset Name	Shape	Critical Point
X	mid_low_x	bell	20
Z	mid_low_z	bell	40

5.0 DESCRIPTION OF FLIX

The program developed for this study is called FLIX (Fuzzy Logic Interpreted eXpert system). FLIX is an interactive tool that automates the construction of knowledge bases. It interprets linguistic fuzzy expressions such as "high", "large" or "hot" into an accurate and minimum rule set which models an input/output discrete data relationship.

5.1 Program Specifications

FLIX was coded in C on a UNIX workstation (Sun Sparc 1+). Approximately 5000 lines of source code were written, and compiled to a 150 Kbytes executable file. The program was designed to be interactive and portable. A text interface controls program flow and prompts for input/output. Graphical utilities plot discrete data from files external to the main program.

Keyboard input selects menu options. ASCII Data files, are used to input and output data, samples are shown in Appendix 4. By typing *flix* at the command line, the user is presented with the main menu options. Detailed descriptions of the menu options are in Appendix 3.

5.2. Main Routines

The principal operating modes are *Knowledge from Data* in Figure 5.1.1 and *Data from Knowledge* in Figure 5.1.2. They share common *Defuzzification* and *Plotting* routines. In the *Data from Knowledge* mode, the user inputs a knowledge base and subset descriptions. Alternatively, in the *Knowledge from Data* mode, the user inputs a reference database and subset descriptions. The *Position Subsets*, *Select Dominant Rule Sets* and *Minimize Rules* routines then induce a knowledge base. The knowledge base

generates data on defuzzification. In the *Knowledge from Data* mode generated data is compared to the reference data by the *Compute Error* routine.

5.2.1 Input/Output Files

By default, FLIX reserves specific names for input/output files. The default file names are indicated in bold italicized type in Figures 5.1. Default names can be over-written, by stating alternative file names during program execution. A Main Menu option allows the user to display selected system default files during program execution.

5.3 Knowledge from Data

The *Knowledge from Data* mode, shown in Figure 5.1.1, generates rules from discrete data and fuzzy subset descriptions. The accuracy of the generated knowledge base is verified by examination of the error report. These correlates the rule generated data with the original reference data. Set additions are made using strategies which minimize the number of rules required to describe the reference data. The *Knowledge from Data* mode has two sub-options where automatic and manual strategies are implemented to generate rules.

Fig. 5.1.1 Knowledge from Data Mode

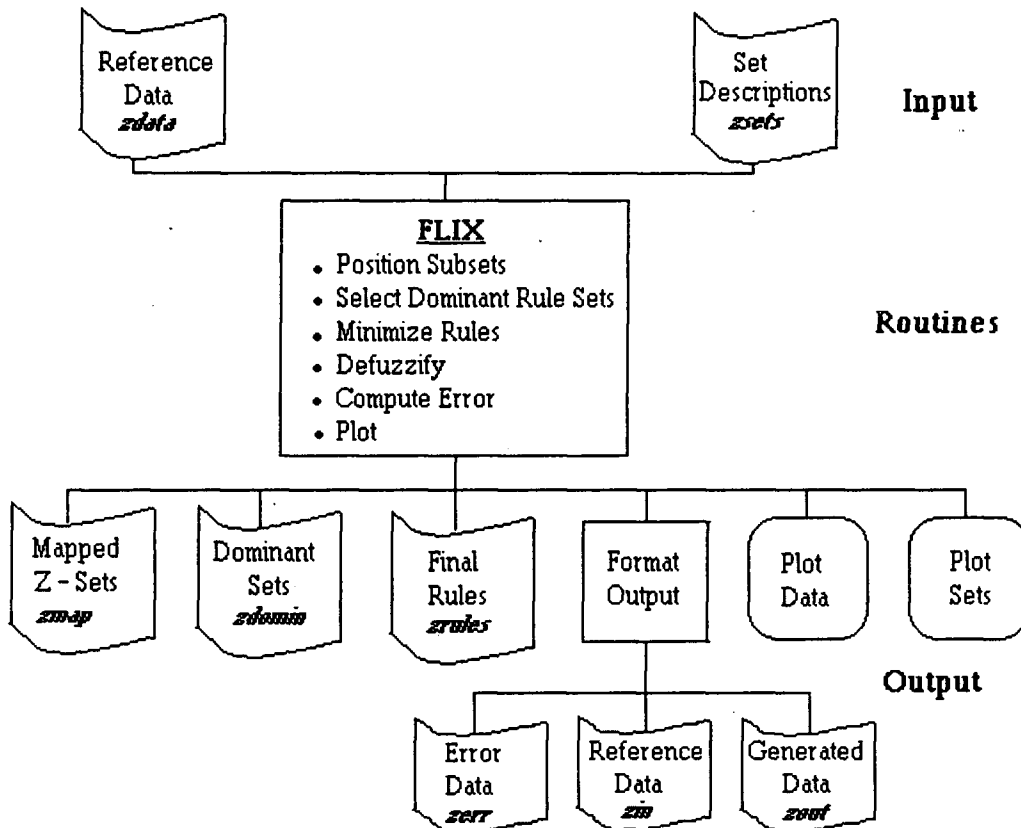
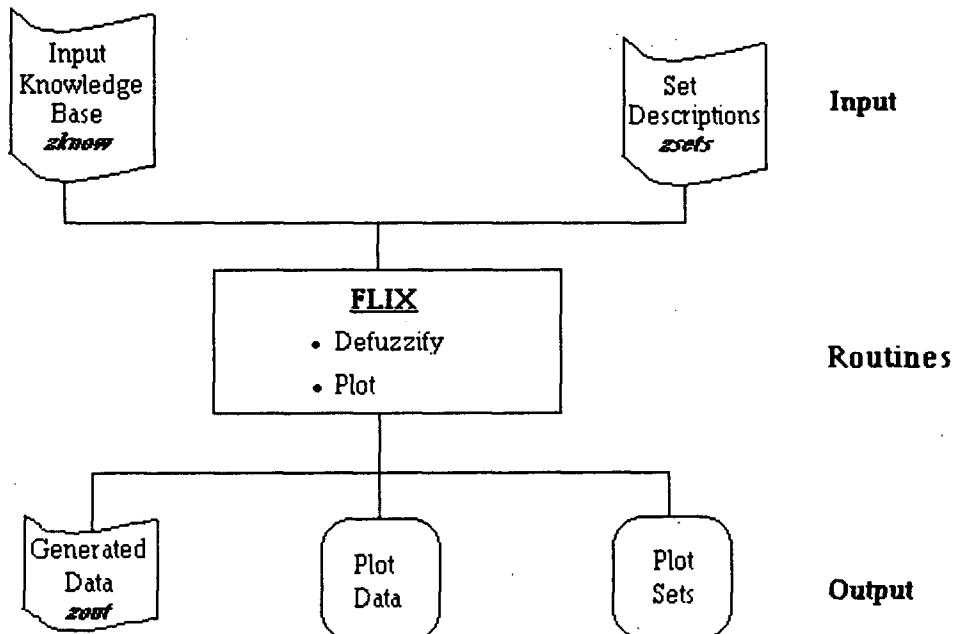


Fig. 5.1.2 Data From Knowledge Mode



5.3.1 Manual Rule Generation

This option allows a user to define input subsets, redirect input/output to console or specified files and selectively execute sub-menus which are tabulated and described as follows:

Table 5.1 Manual Knowledge From Data Sub-Menus

Menu Options	Procedure	Default Files
<i>Read Reference Database</i>	Input	<i>zdata</i>
<i>Read Input Sets</i>		<i>zsets</i>
<i>Write Mapped Sets</i>	Output	<i>zmap</i>
<i>Write Dominant Sets</i>		<i>zdom</i>
<i>Write Final Rules</i>		<i>zrules</i>
<i>Write Output Data</i>		<i>zin</i> <i>zout</i> <i>zerr</i>
<i>Plot Sets</i>		<i>console</i>
<i>Plot Data</i>		<i>console</i>

5.3.1.1 Prerequisite Inputs

The two input options of *Read Reference Database* and *Read Input Sets* are prerequisites for subsequent processing and generation of rules. The *Read Reference Database* option reads the reference data from the *zdata* file. The *Read Input Sets* option, allows flexibility in the definition of input subsets, which are read either interactively from the keyboard or directly from the *zsets* file.

5.3.1.2 Generated Reports

Write Mapped Set option executes the *Position Subset* routine which maps the input fuzzy subsets at their critical points unto the reference data to generate the *zmap* file.

Write Dominant Sets option executes the *Select Dominant Rule Sets* routine which selects dominant subsets, from the **zmap** file to generated dominant subset descriptions in the **zdom** file.

Write Final Rules executes the *Minimize Rules* routine which assigns qualifiers when describing dominant subsets in file **zdom**. Final rules are output to the **zrule** file

Write Output Data executes the *Defuzzify* routine on the rules file **zrule**. to generate discrete data in file **zout**. The *Compute Error* routine compares rule generated data to reference data to produce error indicators in file **zerr**.

5.3.1.3 Graphical Output

Plot Sets graphically displays the current fuzzy subset membership functions while **Plot Data** graphically displays the reference data, generated data and error data.

5.3.2 Automatic Rule Generation

Automatic rule generation implements the strategy of adding a new fuzzy subset to each variable as was discussed in Section 4.7.1. Default system files are used for input and output. All routines listed in Figure 5.1.1 are executed. The plotting routines graphically presents the current subset membership functions as well as the reference data, generated data, and error data.

5.3.2.1 Increasing Subset Definitions

In the automatic mode, the *Increase set definition to next level* adds a subset to each variable. The output graphs along with the reported correlation coefficient, allows the user to decide when to stop subset additions.

5.4 Data from Knowledge

The *Data from Knowledge* mode shown in Figure 5.1 shares the common *Read Input Sets*, *Defuzzification* and *Plot* routines with the *Knowledge from Data* module. The main difference is that *Knowledge from Data* requires additional routines to deduce a knowledge base, while in the *Data from Knowledge* mode, the user defined knowledge base is read and defuzzified.

Table 5.2 Data from Knowledge Sub-Menus

Menu Options	Procedure	Default Files
<i>Read Reference Knowledge Base</i>	Input	<i>zknow</i>
<i>Read Input Sets</i>		<i>zsets</i>
<i>Write Output Data</i>	Output	<i>zout</i>
<i>Plot Sets</i>		<i>console</i>
<i>Plot Data</i>		<i>console</i>

5.4.1 Prerequisite Inputs

The two input routines of *Read Reference Knowledge Base* and *Read Input Sets*, shown in Table 5.2, are prerequisites for defuzzification. The *Read Reference Knowledge Base* routine reads the Knowledge base from a *zknow* file. The *Read Input Sets* option, is the same as described previously for the manual Knowledge from Data option in Section 5.3.1.1

5.5. Combining Modes for Knowledge Base Development

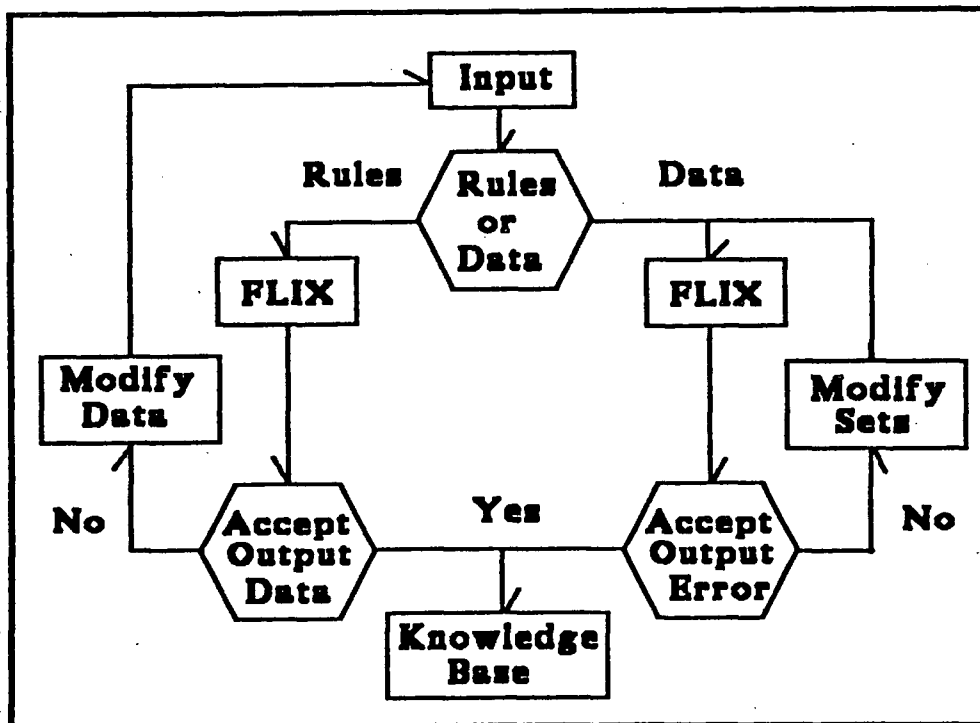
The two principal operating modes; *Data from Knowledge* and *Knowledge from Data* can be combined effectively in knowledge engineering. Figure 5.3 shows this integration, to produce final rules which accurately describe an input/output relationship.

If the user believed that a particular knowledge base represented some desired input/output relationship, this knowledge base along with the relevant subset

descriptions, could be read by FLIX operating in the Data from Knowledge mode. On defuzzification the user can see the generated profile. If the profile is acceptable, then the knowledge base with the subset descriptions are taken as conceptual representations.

If the profiles are unacceptable, then undesirable regions are modified by changing discrete values within the database. The modified database then serves as the reference data for the Knowledge from Data mode. Here either automatic or manual rule generation modes or combinations of both could be implemented to generate an alternative knowledge base which accurately represents the desired data.

Fig. 5.3 Integration of FLIX Modes for Knowledge Base Development

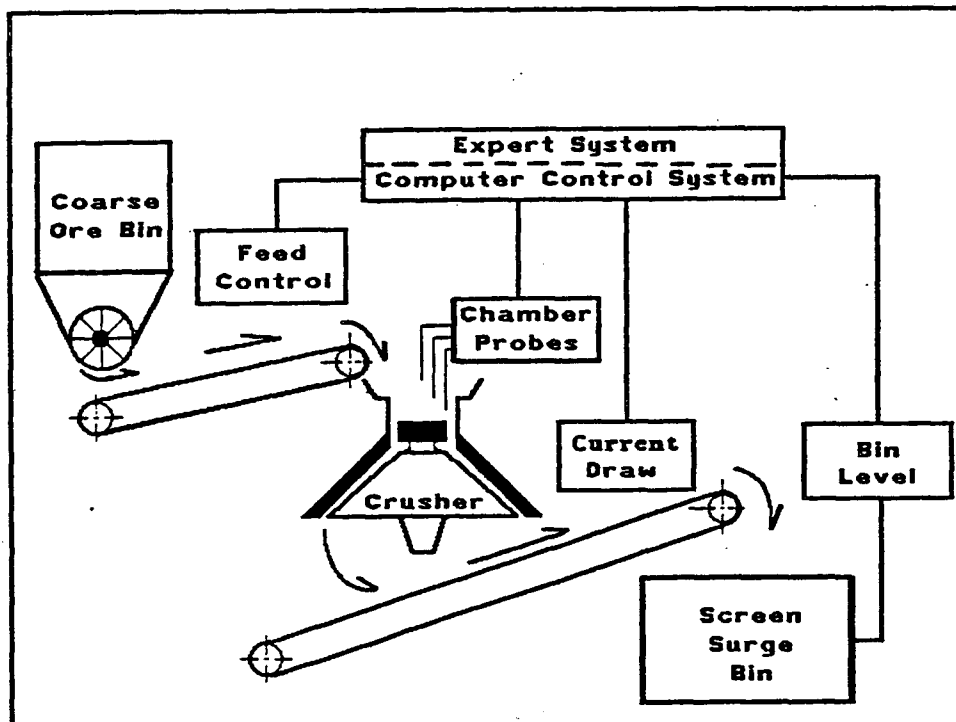


6.0 MODELLING A SECONDARY CRUSHER CONTROLLER

6.1 The Process

The FLIX acquisition system was tested to build a controller for a simulated secondary crusher. The original simulator was designed at Queens University by Tucker and Meech (1984). It utilized conventional PID control techniques. This simulator was later modified by Harris and Meech (1986) where manual and expert operating modes were incorporated. In this study, FLIX has been used to automate the acquisition process thereby producing rules for the expert control system.

Fig. 6.1 Secondary Crusher Control System



The process investigated, as shown in Figure 6.1, consists of a 2.1m standard cone crusher operating with a 350 h.p. motor. The plant supplies production to a milling operation of approximately 10,000 to 15,000 tonnes per day. The coarse ore bin delivers material via a feeder to the crusher chamber. Chamber Probes and an Ammeter monitor chamber levels and current consumption respectively for the crusher. The screen surge bin is used to store and feed ore to other down-stream crushing and screening operations. Level Sensors monitor bin levels.

The Computer Control System had a knowledge base within its expert module. The control system received three discrete controller inputs ie,.

- 1) Chamber Level from the Crusher Chamber Probes
- 2) Current draw from the Ammeter
- 3) Level from Screen Surge Bin

The discrete signals of Current draw and Screen Surge Bin levels after processing by the rules in the expert system. Subsequent defuzzification generate a controller output of desired feed rate change setpoint for the local control loop of the coarse ore feeder.

6.2.1 Original Knowledge Acquisition Methods.

Harris (1986) followed the approach listed below to acquire the knowledge for the expert system. The steps used are as follows:

Observing and interviewing the operator

Listing the rules of thumb

Identifying and standardizing the linguistic measures of each process state

Defining membership functions for subsets representing the linguistic measures

Assembling fuzzy rules of strategy.

6.2.1.1 Observing and Interviewing the Operator

The initial stage involved questioning and observing the operators to determine their strategy. Scenarios representing proposed and possible reactions were noted. Crude rules of thumb were derived for this process such as:

"Whenever the power is low, I increase the feedrate a lot, providing that the screen bin is not too high."

6.2.1.2 Standardizing Linguistics

Rules of thumb had to be organized and listed with standardized terminology for the various process states. Identification of input to and output from the controller was structured so that inputs formed the premise of rules and output formed the conclusion.

6.2.1.3 Definition of Membership functions

Two key issues considered here were:

- 1) The universe of variable definition
- 2) The resolution of fuzzy sets

For completeness, the domain of sets had to be carefully defined so that any discrete x from the universe of a specific variable X had non-zero membership in at least one set. Incompleteness would cause control inaction, because of nonexistent definitions of controller action.

6.3.1 Original Control Rules

Harris (1986) evolved a lengthy process over several days with trial and error methods of tuning. Later Meech and Jordon (1990) employed similar techniques which made further improvements to the desired output profile. The tuning techniques included:

- 1) modification of control rules

- 2) changing ranges over which fuzzy sets were defined
- 3) redefinition of membership functions dynamically based on ore conditions.

The original rules are listed in Table 6.1 with their fuzzy subset definitions shown in Figure 6.2.1

Table 6.1 Original Controller Rules

CURRENT DRAW	SCREEN BIN LEVEL	FEED RATE CHANGE
<i>Any</i>	High	Negative Big
High	<i>Any</i>	
Medium High	<i>Any</i>	
Okay	Fine	
Medium Low	Fine	Negative Small
Okay	<i>not</i> Fine	
Low	Fine	No Change
Medium Low	<i>not</i> Fine	Positive Small
Low	<i>not</i> Fine	Positive Big

Fig. 6.2.1 Original Set Definitions for Controller after Meech and Jordon (1990)

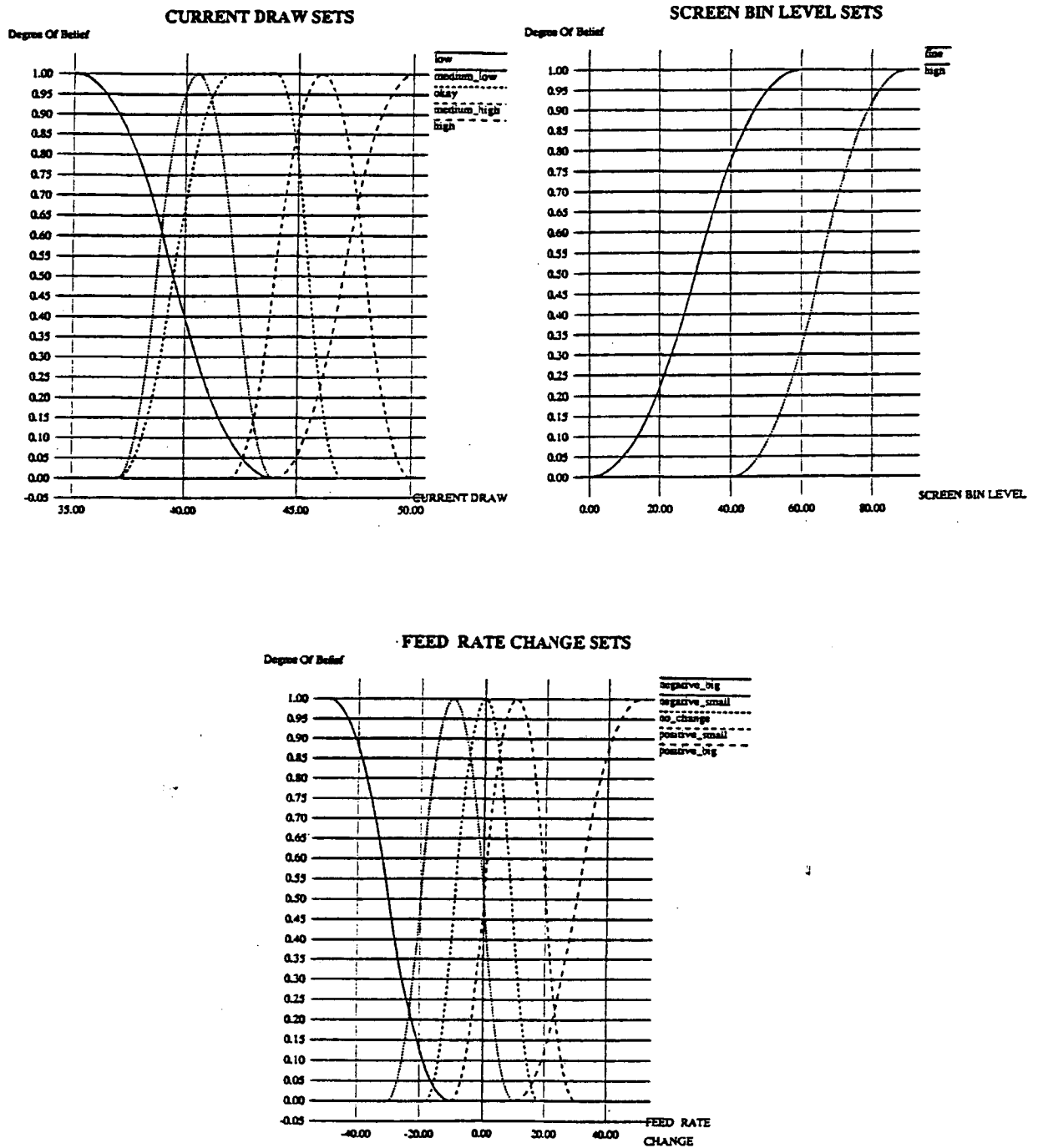


Fig. 6.2.2 Automatic FLIX Generated Set Definitions for Controller

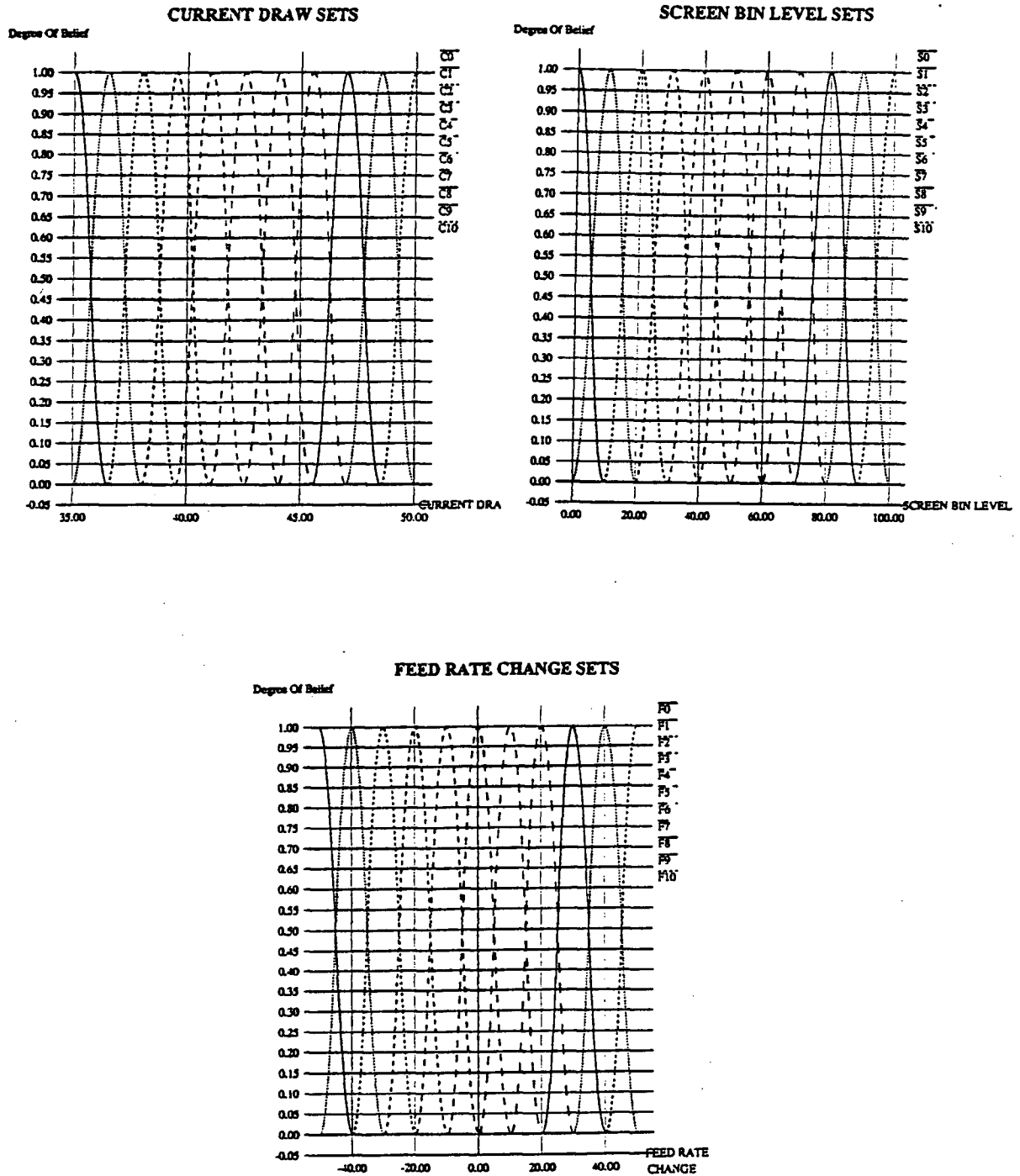
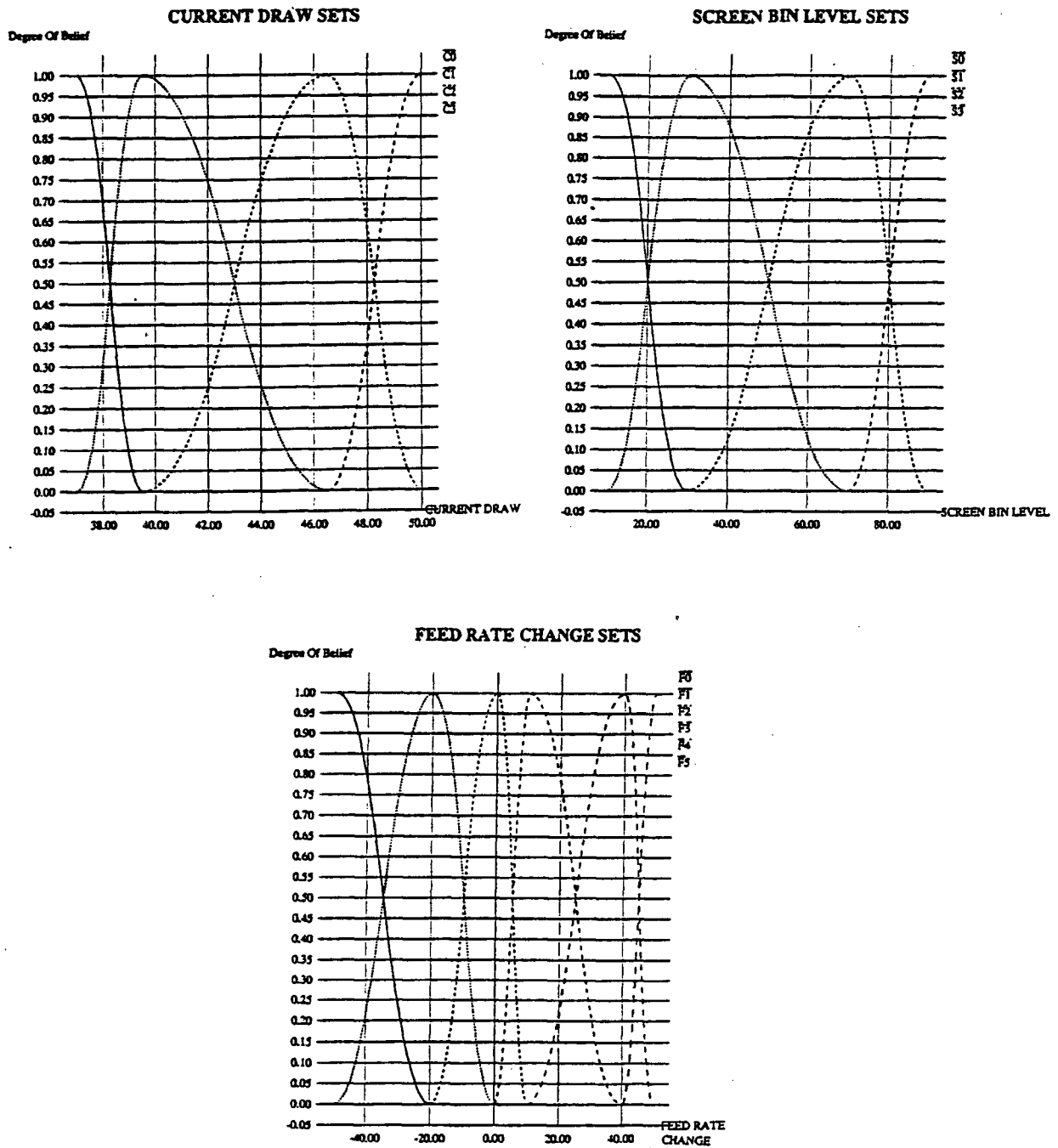


Fig. 6.2.3 Manual FLIX Generated Set Definitions for Controller



6.4. Knowledge Acquisition With FLIX

FLIX was used in the Knowledge to Data mode, on the 1990 rule sets, to generate data for the performance of the controller. The original rule-generated profile of Figure 6.3.1. was found to be undesirable, in the 60-100% screen bin levels and 35-40 amp current draw region. The undesirable rule-generated profile was caused by an inadvertent contradiction in the knowledge base shown in Table 6.1. At high screen bin levels (100%) and low current draws (35 amps) the first and seventh premise statements in the knowledge base are true and contributes belief in the Negative Big and No Change conclusions. Belief in both Negative Big and No Change without belief in Negative Small is contradictory. Belief in the No Change is clearly undesirable since only large reductions (Negative Big) to the feed rate are appropriate to avoid an overflow.

6.4.1 Reference Controller Profile

The profile was corrected by changing unacceptable discrete values to those in Figure 6.3.2. This corrected profile was used as reference data in FLIX to regenerate an accurate rule set which describes the corrected profile.

6.4.2 Generation of Knowledge Base

Both automatic and manual rule generation modes were investigated using FLIX with the modified reference data of Figure 6.3.2. These methods are described below and data for this work is shown in Appendix 5.

6.4.2.1 Automatic Acquisition

Automatic rule generation implemented the strategy of adding a single subset to each variable outlined in Section 4.7.1. The correlation coefficients from successive subsets additions are listed in Table 6.2. The coefficient generally decreased with increasing subset definitions. The highest coefficient of 0.998 was obtained at 11 subset additions for each variable.

Fig. 6.3.1 Original Rule Generated Profile

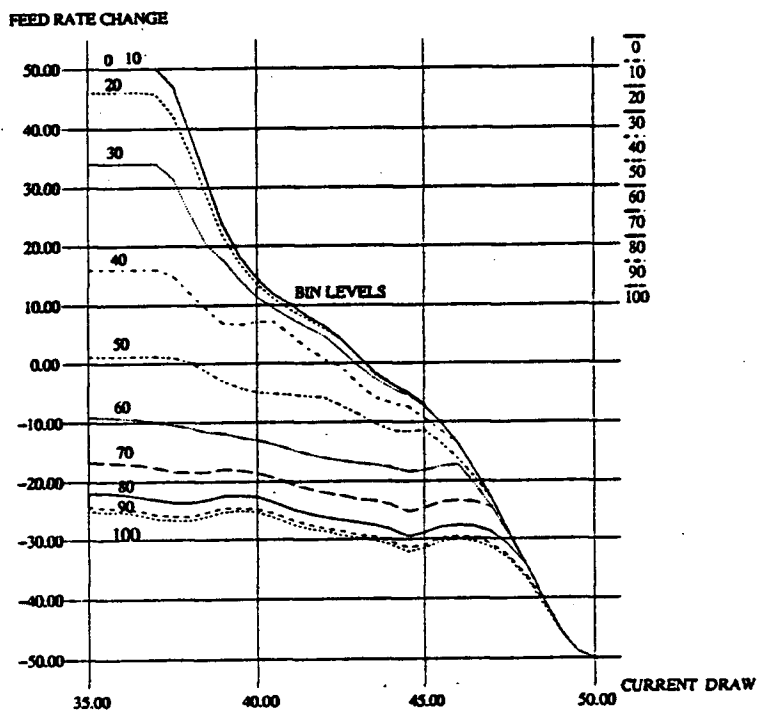


Fig. 6.3.2 Desired Controller Profile

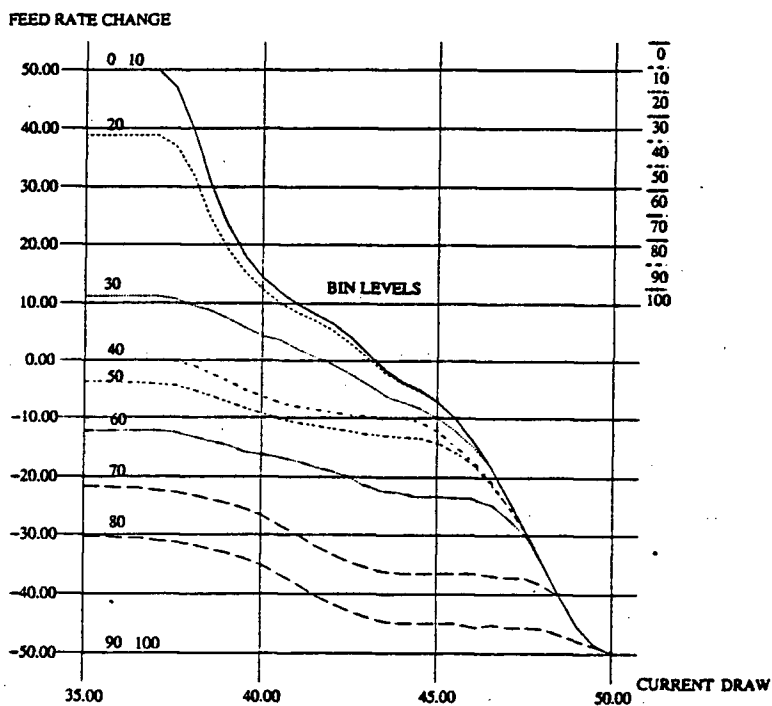


Fig. 6.3.3 Automatic FLIX Controller Profile

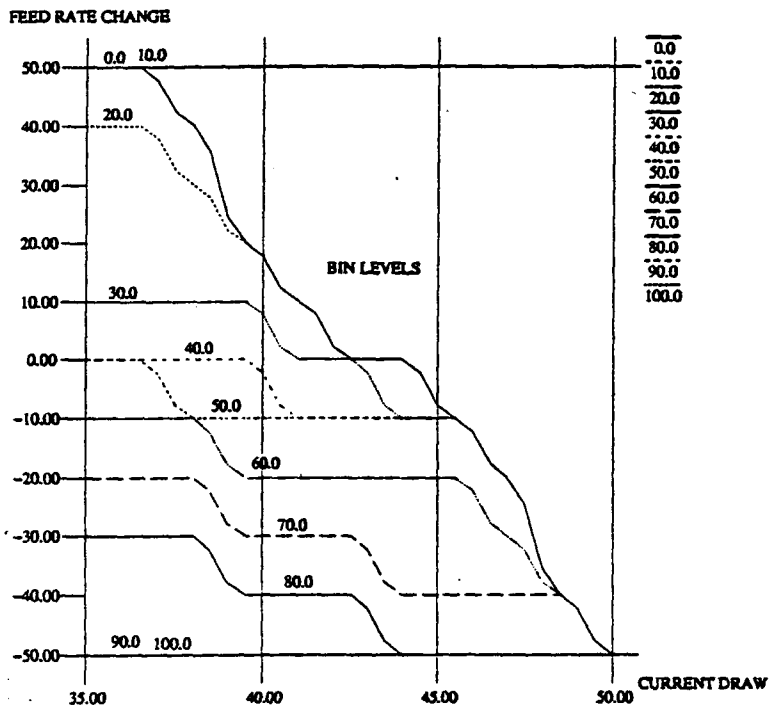


Fig. 6.3.4 Manual FLIX Controller Profile

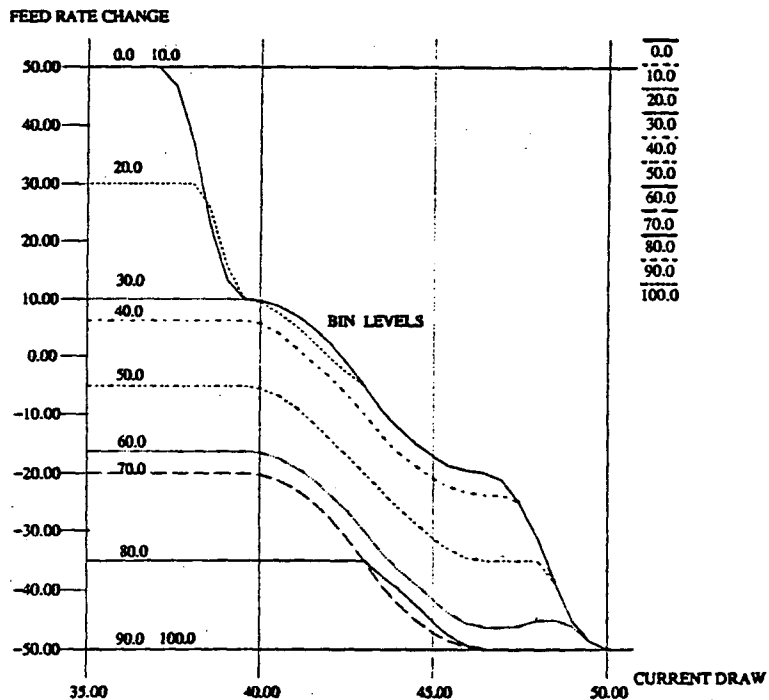


Table 6.2 Correlation Coefficients for Automatic Subset Additions

Number of Subsets	Correlation Coefficient
2	0.961
3	0.974
4	0.972
5	0.965
6	0.989
7	0.987
8	0.986
9	0.985
10	0.987
11	0.998

Although the final 0.998 coefficient was acceptable, the output profiles in Figure 6.3.3 are clearly unacceptable because of the irregular and sharp transients in the automatic generated output. The final output rules, because of fine discretization of the subsets, shown in Figure 6.2.2, resulted in a very complex rule description as shown in Table 6.3

Table 6.3 Automatic Generated Rules

CURRENT DRAW	SCREEN BIN LEVEL	FEED RATE CHANGE
<i>Any</i>	S9	F0
<i>Any</i>	S10	
C10	<i>Any</i>	
<i>between</i> C6 C10	S8	
<i>between</i> C6 C9	S7	F1
<i>between</i> C3 C5	S8	
C9	<i>between</i> S0 S7	
<i>between</i> C3 C5	S7	
<i>between</i> C0 C2	S8	F2
C8	S6	
<i>between</i> C3 C7	S6	
<i>between</i> C0 C2	S7	
C8	<i>between</i> S0 S5	F3
<i>between</i> C6 C7	S3	F4
<i>between</i> C4 C7	S4	
<i>between</i> C2 C7	S5	
<i>between</i> C0 C2	S6	
C7	<i>between</i> S0 S5	F5
<i>between</i> C5 C6	S0	
<i>between</i> C5 C6	S1	
<i>between</i> C5 C6	S2	
<i>between</i> C4 C5	S3	F6
<i>between</i> C0 C3	S4	
<i>between</i> C0 C1	S5	
<i>between</i> C0 C3	S3	
C4	<i>between</i> S0 S2	F7
C3	<i>between</i> S0 S2	F8
C2	S2	F9
<i>between</i> C0 C1	S2	
C2	<i>between</i> S0 S1	
<i>between</i> C0 C1	S0	
<i>between</i> C0 C1	S1	F10

6.4.2.2 Manual Acquisition

Manual acquisition methods achieved better results which are shown in Table 6.4 and Table 6.5. The strategy of set additions to the areas of greatest error, outlined in Section 4.7.3, was implemented to increased the correlation coefficient for the first four stages.

For the final fifth and sixth stages a second strategy of relocating subsets, discussed in Section 4.7.2, produced smoother profiles.

Table 6.4 Correlation Coefficients for Manual Subset Additions

Stages	Number of Subsets			Correlation Coefficient
	Current Draw	Bin Level	Feed Rate Change	
1	2	2	2	0.953
2	3	3	3	0.965
3	4	3	4	0.974
4	4	3	5	0.988
5	4	4	5	0.985
6	4	4	6	0.985

Final correlation coefficients of 0.985 were achieved from the set definitions in Figure 6.2.3. The rules generated manually in Table 6.5, were less complex than those automatically generated in Table 6.3. The manual profiles in Figure 6.3.4 were also smoother.

Table 6.5 Manual Generated Controller Rules

CURRENT DRAW	SCREEN BIN LEVEL	FEED RATE CHANGE
<i>Any</i>	S3	F0
C3	<i>Any</i>	
C2	S2	
<i>between</i> C0 C1	S2	F1
C2	<i>between</i> S0 S1	
<i>between</i> C0 C1	S1	F2
C1	S0	
C0	S0	
		F5

6.5 Test on Alternative Non-monotonic Surface

A second test was conducted with a more complex non-monotonic surface, previously referred to as reference data in Section 4. This surface, shown graphically in Figure 6.4.1 had greater variability than the control example. After three levels of subsets additions in the automatic mode, FLIX induced the knowledge base shown in Table 6.6. The rule generated data shown in Figure 6.4.2, produced a correlation coefficient of 0.963. Data for this approach are shown in Appendix 4.

Table 6.6 Final Automatic Rules for Non-Monotonic Surface

Y		X	Z
<i>not</i>	mid_y	low_x	low_z
	mid_y	high_x	
<i>any</i>		mid_x	mid_z
<i>not</i>	mid_y	high_x	high_z
	mid_y	low_x	

Fig. 6.4.1 **Reference Data for Non-monotonic Data Surface**

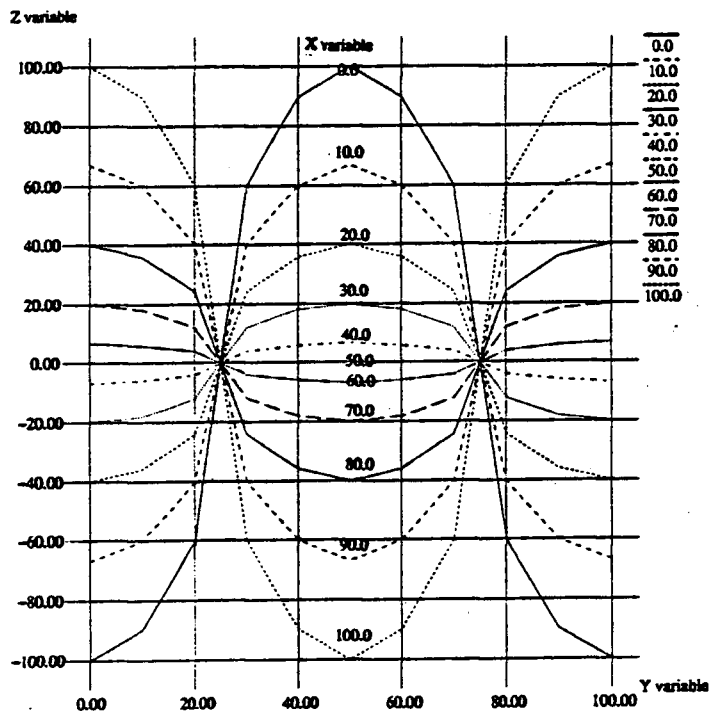
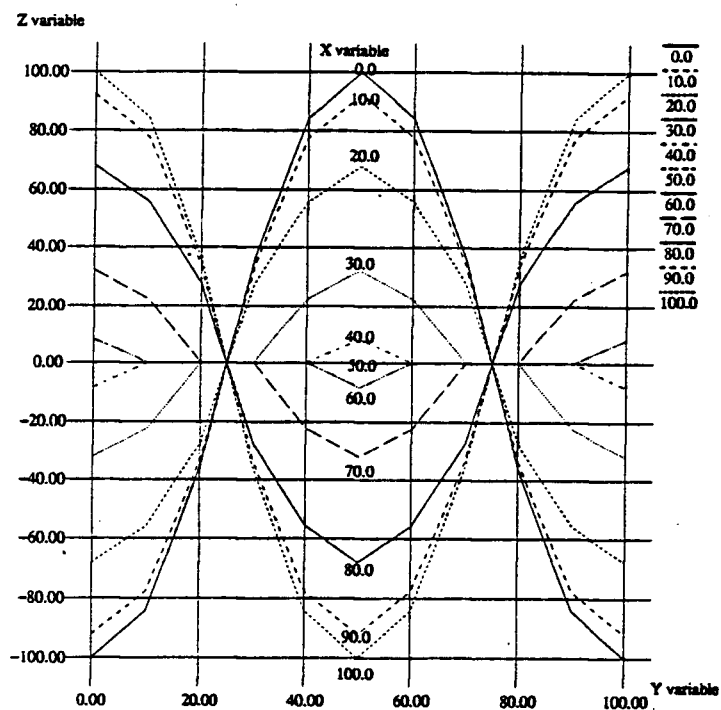


Fig. 6.4.2 **Rule Generated Data for Non-monotonic Data Surface**



7.0 FUTURE DEVELOPMENTS

7.1 Limitation of Bayesian Logic

The generation of rules to describe smooth gentle data surfaces from large numbers of fuzzy descriptions proved unsatisfactory. This problem was seen in the example using the Automated mode for the controller surface in Section 6. The result was unsatisfactory for two reasons:

Complex rules - The complexity of the rules results from the large number of subsets (11 levels) which were necessary for precision. The rules are difficult to interpret since large numbers of subsets are required in very complex associations. There is considerable redundancy included in the rule descriptions.

Irregular surfaces - The larger number of subset descriptions resulted in steeper gradients for each membership function. This created significant transients across rule boundaries because of the If-then association. Belief in premise conditions and rules then varied sharply for small incremental changes in variables.

7.2 Neural Networks

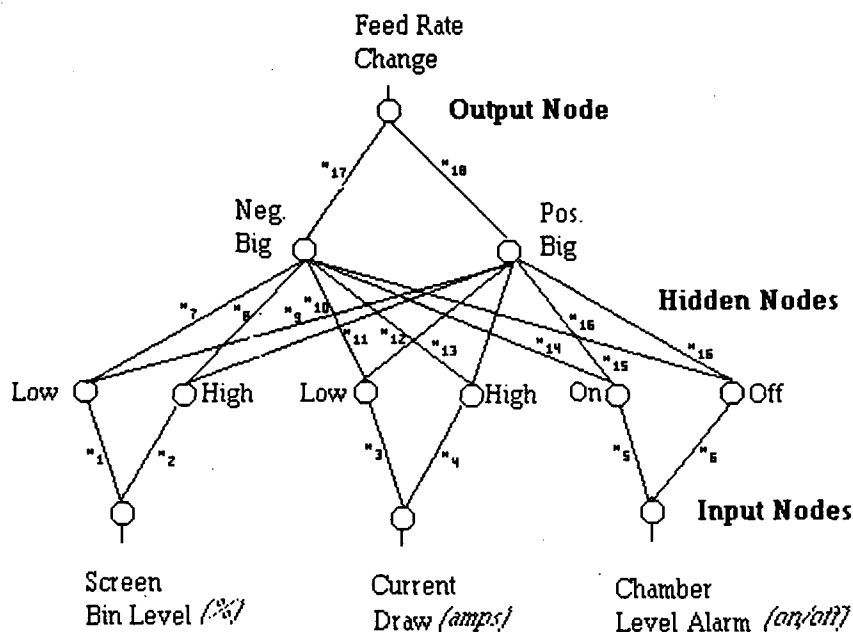
Recently there has been an explosion of interest and research into neural networks. These systems mimic some of the structure and function of the brain. There is potential for a marriage of fuzzy logic and expert systems formalisms with neural networks to resolve some of the limitations of pure fuzzy logic systems or rule based systems (Flament et al, 1990; Hart, 1989).

A neural network representation of the knowledge or generated rules may resolve both issues of surface irregularity and rule complexity. Knowledge stored explicitly in the form of rules could instead be stored implicitly in the strength of connecting nodes of a network as shown in Figure 7.1. A network to resolve the controller example in Section 6 would operate as follows: Discrete data for the screen bin and current draw as well as a

third logical value for a chamber level variable are fed in at the bottom input layer. The input to each node in the network is normally the sum of each signal times the link weight (w_i). The link weights are calculated using a back-propagation pattern matching analysis. The iterations on this link weight matrix continue until the error between output and desired output is acceptable. The output from each node is determined by a specific nodal function relationship with each input signal to the node.

The power of this technique is demonstrated in the fact that large rule sets and complex relationships could be described by a simpler neural net. The problem of sharp transients resulting from If-Then associations within rules may be resolved when replaced with weighting functions within the net. For neural networks which suffer from an inability to provide satisfactory explanations, the labelling of nodes as fuzzy set descriptions could allow the system to provide justification for decisions made which would be understandable to operating personnel.

Fig. 7.1 Neural Network Representation of Knowledge Using 2 Fuzzy Descriptions for Each Variable



7.3 Extension to Multiple Dimensions

FLIX is capable of resolving data which comprise two premises and a single conclusion. Further research is required for extension of the system to resolve multi-dimensional data. Extension to resolve, multiple premises and conclusions are separate issues.

Multiple Conclusions - extensions for multiple conclusions is the simpler of the two. Conclusion statements in the rules are independent of inter-relationships between premise statements. Therefore multiple conclusions could be stated independently in the formation of rules.

Multiple Premises - Multiple premise statements in rules require multi-dimensional databases with inputs for all possible coordinate positions. Up to three premise variables could be visually represented in a Cartesian system after which premise conditions can be displayed in a spreadsheet format. Fuzzy subsets associations could then be described by more elaborate inductive algorithms within the rule reduction routines.

7.4 Rule Reduction Methods

Further rule reductions or more succinct descriptions could be achieved by a comparison and selection algorithms within the rule reduction routine. This routine would compare rule reduction patterns and thereby select the most efficient description. An example of this is shown with the following premise statements:

<i>between C5 C6</i>	<i>S0</i>
<i>between C5 C6</i>	<i>S1</i>
<i>between C5 C6</i>	<i>S2</i>

which could be more succinctly described as follows;

<i>between C5 C6</i>	<i>between S0 S2</i>
----------------------	----------------------

7.5 Error Reduction Strategies - Automated

Several issues related to the error reduction strategies could be studied in greater depth.

These include:

Adding Sets to Areas of Greatest Error - This strategy needs further study before it can be automated. Issues to be resolved include: the order of set additions, adding single versus multiple sets at a time and mapping of subsets when multiple positions have equal error.

Adjusting Existing Subsets - The methods of relocating and redefining existing fuzzy subset has not been examined thoroughly, although better correlation coefficients were observed from trial and error redefinitions for bell shapes and critical points for the Z subsets to reflect average z values from the reference data.

Using Alternative Subsets Shapes - The influence of alternative shapes such as flat_bell and trapezium should be studied. In some instances, the flat_bell shape achieved significantly better correlation coefficients than the bell. The optimal shape and position of subsets will depend on the shape and features of the original data. Points of inflection and changes in slope must be used to select shape positions and numbers of subsets required.

8.0 CONCLUSIONS

An interactive system has been developed in the Mining and Mineral Processing Department at the University of British Columbia, to automated knowledge acquisition, the critical bottleneck area in expert system development. Accuracy of the knowledge base selected by the user becomes a trade off between efficiency (small number of subsets in rules) and effectiveness (large number of subsets definitions for increased precision).

The system incorporates many user friendly menu driven features which make it useful to users who may not understand the concepts of fuzzy logic. Rules are generated automatically from desired data. Therefore data could be viewed as a knowledge base generated from desired rules or visa versa. Rules can be automatically modified to better represent data. This allows knowledge bases to be built much faster than using a traditional interview approach.

The system was successfully tested to generate a minimum rule set to describe an expert system controller for a secondary crusher. The error of the output map compared to the desired output was minimized. The success of this system therefore makes knowledge acquisition for expert systems easier to implement in the industry.

The power of fuzzy logic to represent complex relationships with simple and elegant rule structures has been demonstrated by this work. Future efforts should be expended to automate further the error reduction process. Expansion to handle multi-variable input/output also needs to be examined.

REFERENCES

- Feigenbaum, E.A. McCorduck, P. (1984). "The fifth generation", Pan Books London.
- Flament, F. Thibault, J. Hodouin, D. (1990). "Potential Applications of Neural Networks for the Control of Mineral Processing Plants" Proc 29th Annual Conference of Metallurgist, Aug 90, Hamilton Ont. p.10.
- Ford, K.M. Petry, F.E. Adams-Webber, J.R. Chang, P.J. (1991). "An Approach to Knowledge Acquisition Based on the Structure of Personal Construct Systems", IEEE Transactions On Knowledge and Data Engineering, Vol. 3, No. 1, p. 78-88
- Greenwell, M (1988). "Knowledge Engineering for Expert Systems" Ellis Harwood Series in Expert Systems. Addison Wiley, London
- Hall, M.B Harris, C.A. Miller, R.S. Woo, E. (1990). "Knowledge Acquisition Techniques for Expert Systems" Proc, 29th Annual Conference of Metallurgist. Hamilton Ontario p. 8
- Harris, C.A. (1986) "Fuzzy Logic: A Potential Control Technique For Mineral Processes", MSc Thesis. Queen's University. p. 77
- Harris C.A. Meech J.A. (1987) "Fuzzy Logic: A Potential Control Technique for Mineral Processes" CIM Bulletin, Sept 87, p. 51-59.
- Hart, A. (1989). "Knowledge Acquisition for Expert Systems", Kogan Page Ltd. London p. 196
- Hayes Roth, F. Waterman, D.A. Lenat, D.B (1983). "Building Expert Systems" Addison Wiley. London
- Kelly, G.A. (1955) "The Psychology of Personal Constructs" Norton, New York
- Kennedy, J.B. Neville A.M. (1976) "Basic Statistical Methods for Engineers & Scientists" Dun Donnelley, New York New York.
- Meech J.A Jordon L.A. (1990) "Development of A Self-Tuning Fuzzy Logic Controller" Proc 29th Annual Conference of Metallurgist, Aug 90, Hamilton Ont.
- Michie, D. Johnson, R. (1985) "The Creative Computer" Pelican England.
- Nazareth, D.L. (1988) "Issues in the verification of knowledge in rule-based systems;" The Foundation of Knowledge acquisition vol 4, p. 161-175.

Tucker, R.J. Meech J. A. (1984) "A Dynamic Computer Model of Crushing Plants for teaching Purposes and Operator Training" Proc 1984 Summer Simulation Conference, Boston, Mass., Vol. 2, July 84 , pp. 1107-1113.

VP-Expert (1987) "Rule based Expert System Development Tool." Manual, Paperback Software Inc.

Zemankova-Leech, M. Kandel, A. (1984). "Fuzzy Relational Databases: A Key to Expert Systems" , Köln: Verlag TÜV Rheinland. p. 177

Zadeh, L.A. (1984). "Making Computers Think Like People", IEEE Spectrum, August, 1984, pp. 26-32.

APPENDIX 1 Fuzzy Membership Functions

Functions are described by the format:

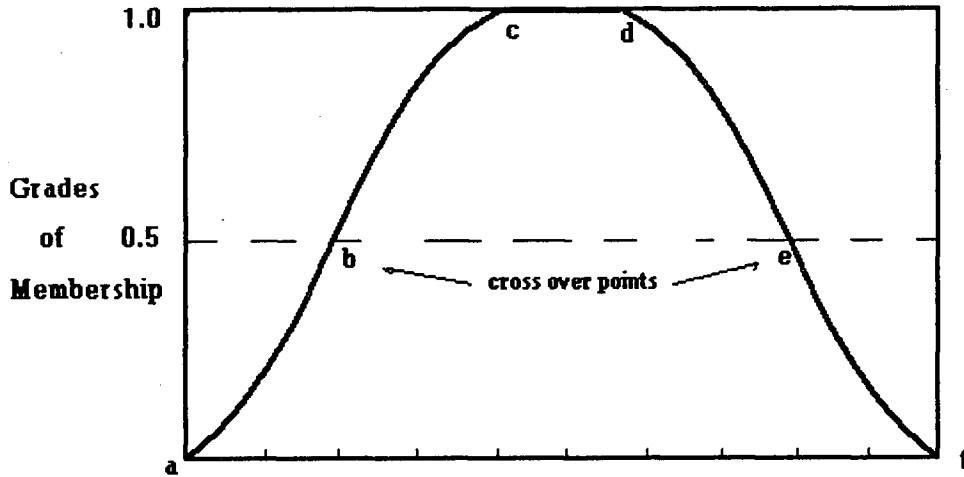
$$M_{\text{name}}(u; a, b, c, \dots)$$

where;

- M - function type ie. S for quadratic and T for triangular
- name - description of function type
- u - $u \in U$ over specified interval or domain of U
- a, b, c... - user defined critical points which specify turning points of function

An example of the Four Point Flat Bell is shown in Figure A1.1:

Fig. A1.1 General Definition of Flat Bell Membership Function



1.1 Four Point Flat Bell $S_{4\text{pt_flat_bell}}(u; a, c, d, f)$

$$0 \quad \text{for} \quad u \leq a$$

$$2 \left(\frac{u-a}{c-a} \right)^2 \quad \text{for} \quad a \leq u \leq b \quad \text{and} \quad b = \left(\frac{a+c}{2} \right)$$

$$\begin{array}{lll}
 1 - 2\left(\frac{u-c}{c-a}\right)^2 & \text{for} & b \leq u \leq c \\
 1 & \text{for} & c \leq u \leq d \\
 1 - 2\left(\frac{u-d}{f-d}\right)^2 & \text{for} & d \leq u \leq e \quad \text{and} \quad e = \left(\frac{d+f}{2}\right) \\
 2\left(\frac{u-f}{f-d}\right)^2 & \text{for} & e \leq u \leq f \\
 0 & \text{for} & u \geq f
 \end{array}$$

1.2 Two Point Bell Up $S_{2pt_bell_up}(u;a,c)$

$$\begin{array}{lll}
 0 & \text{for} & u \leq a \\
 2\left(\frac{u-a}{c-a}\right)^2 & \text{for} & a \leq u \leq b \quad \text{and} \quad b = \left(\frac{a+c}{2}\right) \\
 1 - 2\left(\frac{u-c}{c-a}\right)^2 & \text{for} & b \leq u \leq c \\
 1 & \text{for} & c \leq u
 \end{array}$$

1.3 Two Point Bell Down $S_{2pt_bell_down}(u;d,f)$

$$\begin{array}{lll}
 1 & \text{for} & u \leq d \\
 1 - 2\left(\frac{u-d}{f-d}\right)^2 & \text{for} & d \leq u \leq e \quad \text{and} \quad e = \left(\frac{d+f}{2}\right) \\
 2\left(\frac{u-f}{f-d}\right)^2 & \text{for} & e \leq u \leq f \\
 0 & \text{for} & u \geq f
 \end{array}$$

1.4 Trapezium $T_{\text{trap}}(u;a,c,d,f)$

$$\begin{aligned} 0 & \quad \text{for} \quad u \leq a \\ \left(\frac{u-a}{c-a} \right) & \quad \text{for} \quad a \leq u \leq c \\ 1 & \quad \text{for} \quad c \leq u \leq d \\ 1 - \left(\frac{u-f}{f-d} \right) & \quad \text{for} \quad e \leq u \leq f \\ 0 & \quad \text{for} \quad u \geq f \end{aligned}$$

1.5 Three Point Bell $S_{\text{3pt_bell}}(u;a,c,f)$

Special case of Four Point Flat Bell $S_{\text{4pt_flat_bell}}(u;a,c,d,f)$ where $c=d$.

1.6 Half Bell Down $S_{\text{half_bell_down}}(u;d)$

Special case of Two Point Bell Down $S_{\text{2pt_bell_down}}(u;d,f)$ where d is the user defined critical point and f is the zero membership point which coincides with the supremum position of the adjacent subset to the right.

1.7 Half Bell Up $S_{\text{half_bell_up}}(u;c)$

Special case of Two Point Bell Up $S_{\text{2pt_bell_up}}(u;a,c)$ where c is the user defined critical point and a is the zero membership point which coincides with the supremum position of the adjacent subset to the left.

1.8 Bell $S_{\text{bell}}(u;c)$

Special case of Three Point Bell $S_{\text{3pt_bell}}(u;a,c,f)$ where c is the user defined critical point, f is the first zero membership point which coincides with the supremum position of the adjacent subset to the right and a is the second zero membership point which coincides with the supremum position of the adjacent subset to the left.

1.9 Flat Bell $S_{\text{bell}}(u;c,d)$

Special case of Four Point Flat Bell $S_{\text{4pt_flat_bell}}(u;a,c,d,f)$ where c and d are user defined critical points between which the Flat Bell has full membership. f is the first zero membership point which coincides with the supremum position of the adjacent subset to the right and a is the second zero membership point which coincides with the supremum position of the adjacent subset to the left.

APPENDIX 2 Error Functions

2.1. Difference Between Elements

The body of the error table lists the difference \bar{z}_{ij} for each element of the reference data z_{ij} to its corresponding rule generated output \hat{z}_{ij} . Where:

$$\bar{z}_{ij} = z_{ij} - \hat{z}_{ij}$$

The difference indicates the magnitude of the error at point locations between the reference and rule generated surfaces.

2.2. Sum of Squares Difference

The sum of the squares difference $\sum Z^2$ is calculated for each row, column and the combined table. It is computed by the following equations;

The sum of squares for the jth row is given by:

$$\sum_{i=1,j}^n (z_{ij} - \hat{z}_{ij})^2$$

Similarly for the ith column where;

$$\sum_{i,j=1}^m (z_{ij} - \hat{z}_{ij})^2$$

A total sum of squares for the table where;

$$\sum_{i=1,j=1}^{nm} (z_{ij} - \hat{z}_{ij})^2$$

Each sum of squares, indicates the magnitude of error by row, by column or for the overall table.

2.3. Distribution of Sum of Squares Error

The distribution of the sum of squares error is computed respectively for each column and row by the following relations;

$$\frac{\sum_{i=1}^n (z_{ij} - \hat{z}_{ij})^2}{\sum_{i=1, j=1}^{nm} (z_{ij} - \hat{z}_{ij})^2} \quad \text{or} \quad \frac{\sum_{j=1}^m (z_{ij} - \hat{z}_{ij})^2}{\sum_{i=1, j=1}^{nm} (z_{ij} - \hat{z}_{ij})^2}$$

It indicates the relative error contribution for each row or column in the error table.

2.4. The Correlation Coefficient

The correlation coefficient r is given by the equation;

$$r = \frac{\sum Z \hat{Z}}{\sqrt{\sum Z^2 \sum \hat{Z}^2}}$$

where:

\hat{Z} - element of the reference data

Z - element of the output data

The r coefficient is a standard statistical measure for the closeness of fit between data points Kennedy, (1976). It provides a convenient normalised correlation between rule generated profiles to a common reference data surface. This r coefficient lies in the range $0 \leq |r| \leq 1$: when the value is zero there is no correlation; when it is unity, the correlation is perfect.

APPENDIX 3

MENU OPTIONS

3.1 Program Description

FLIX, by default, reserves specific names for input/output files. Default file names are shown in bold small type in Figure 5.1. Default names can be over-written, by stating alternative file names during program execution. By typing *flix* at the command line the user is presented with the main menu options. Italicized word within the system menus are imbedded text which change as different options are executed. Keyboard input selects menu options. ASCII Data files, with blank spaces and Tab separators, is used for input/output.

3.2 Main Menu Options

The main menu options are:

- | | |
|-----------------------------------|-------------------------------------------------|
| 1) | Knowledge from Database |
| 2) | Data from Knowledgeable |
| 3) | Display Default System Files |
| 4) | Interactive Switch Present Mode: <i>ON</i> |
| 5) | Exit Program |
| Select a <i>Main Menu</i> Option: | |

Options from the list are selected by entering the appropriate number at the Select a *Main Menu* Option prompt.

3.3 Knowledge from Database

Program flow is directed to generate rules from an existing database and fuzzy subset descriptions. Selection of this mode generates two sub-modes as follows

- 0) Manual Rule Generation
 - 1) Automatic Rule Generation
- Select a *Rule Generation* Option:

3.4 Manual Rule Generation

This option allows a user to define; fuzzy subsets descriptions, redirection of input/output to console or files and execute a number of the Manual Rule Generation sub-menus. The sub-menus are as follows;

- 0) Read Reference Database
 - 1) Read Input Sets
 - 2) Write Mapped Sets
 - 3) Write Dominant Sets
 - 4) Write Final Rules
 - 5) Write Output Data
 - 6) Plot Sets
 - 7) Plot Data
 - 8) Do All
 - 9) Main Menu
- Select a *Manual Rule Generation* Option:

3.5 Read Reference Database

This routine along with the Read Input Sets option are prerequisites for subsequent processing and generation of output routines. The reference database file is limited to a maximum size of [31 x 31] discrete values. The first row and column contains the discrete elements for the X and Y variable respectively. The body of the table contains the Z variable elements.

A sub-menu is available for accepting files with non-default names:

- 1) Read *Reference Data* from Default File; **zdata**
 - 2) Read New *Reference Data* File
- Select a *Reference Data* File Option:

3.6 Read *Reference Data* from Default File

The reference database is read from a file with default name zdata.

3.7 Read New *Reference Data* File

The user is prompted with:

Enter New Data Filename:

The file name for the reference data is typed in from the keyboard (maximum length 50 characters).

3.8 Read Input Sets

This routine along with the Read Reference Database option are prerequisites for any subsequent processing and output routines in the Manual Rule Generation module. The fuzzy subset descriptions can be read from the console or a file. A maximum of 15 subsets can be defined for each variable. The following sub-options are available for subset input;

- 1) Read Subsets from Default File: **zsets**
 - 2) Read New Subsets File
 - 3) Read Subsets from Console
 - 4) Save Subsets to File
- Select a Subset File Option:

3.9 Read *Subsets* from Default File

The format of the subset file *zsets* is shown in figure 2. Information on the three variables is entered in blocks which are separated by a line of * symbols. Each block defines a separate variable so there are a total of three blocks. Each subsequent row in a block defines a fuzzy set description for the variable. The first and subsequent elements of each of these rows contain fuzzy subset descriptions; The description is comprised of the subset Name (max 15 characters), Shape (10 possible shapes) and Critical Points (number depends on shape) . A line of * symbols mark the end of the block.

3.10 Read Subsets from Console

This option allows the user to input subsets directly from the console in an interactive mode. The user is prompted to;

Enter the Y *variable* name; *Temperature*

A description of the Y variable is entered (maximum of 15 characters).

The system requests the number of subsets describing the Y variable.

How many fuzzy sets characterize the *Temperature*:

This number is limited to a maximum of 15.

A list of possible subset shapes is presented:

Which shape best describes the *first* fuzzy set for the *Temperature*

- 0) trapezium
- 1) bell
- 2) flat_bell
- 3) half_bell
- 4) half_bell_down
- 5) half_bell_up
- 6) 2pt_half_bell_down
- 7) 2pt_half_bell_up
- 8) 3pt_bell
- 9) 4pt_bell

Select a *Shape* Option:

The critical points are requested for the previously selected shape by;

Enter the critical point(s) for the *half_bell_down Temperature*:

The required number of critical points are entered separated by blank spaces.

The system continues to request subset information (shapes and critical points) for the specified number of subset for each variable.

3.12 Save Subsets to File

Fuzzy subset descriptions are entered interactively from the Read Subset From Console module are saved as a file of similar format to Figure 2 . The sub-menus of this option include:

- 1) Write *output subsets* to file; zsets
- 2) Write *output subsets* to new file

Select an Output Subset Option:

3.13 Write Mapped Sets

This option generates a table, where the X, Y and Z fuzzy subsets are mapped at their critical points onto the reference data. The sub options available are:

- 1) Write *map sets* to default file; zmap
 - 2) Write *map sets* to a new file
 - 3) Write *map sets* to the console
- Select a *map sets* option;

Output for the mapped fuzzy subsets is directed to the selected option.

3.14 Write Dominant Sets

Next a table is generated where the dominant X, Y and Z fuzzy subsets are shown as symbolic representations for rules. The sub options listed are:

- 1) Write *dominant sets* to default file; zdom
 - 2) Write *dominant sets* to a new file
 - 3) Write *dominant sets* to the console
- Select a *dominant set* option;

3.15 Write Final Rules

A final table is generated in which the final rules are directed to specified devices/files with the following sub-options:

- 1) Write *final rules* to default file; zrules
 - 2) Write *final rules* to a new file
 - 3) Write *final rules* to the console
- Select a *final rules* option;

3.16 Write Output Data

Discrete tabloid data for the reference, generated and error databases are generated.

Options include;

- | | |
|--------------------------------------|----------------------|
| 1) | Write Reference Data |
| 2) | Write Generated Data |
| 3) | Write Error Data |
| 4) | Do All |
| Select an <i>Output Data</i> Option: | |

3.17 Write Reference Data

This option directs output from the Write Output Data menu to the following file;

- | | |
|----------------------------------------|--------------------------------------|
| 1) | Write <i>data to file</i> <i>zin</i> |
| 2) | Write <i>data to new file</i> |
| 3) | Write <i>data to console</i> |
| Select a <i>reference data</i> option: | |

3.18 Do All

Executes all of the numbered options sequentially.

3.19 Plot Sets

Presents graphs of the current fuzzy subset membership functions.

3.20 Plot Data

Presents graphs of the reference data, generated data and error data.

3.21 Main Menu

Return to main menu display

3.22 Automatic Rule Generation

This option automatically defines new fuzzy subsets by using the strategy of adding a single subset to each variable, as was discussed in Section 4 . The module uses default system files for input/output. All output routines in are activated and graphs of the current subset membership functions, reference, generated and error data are displayed. The current correlation coefficient is reported in the list of menu options as follows;

Current correlation coefficient = <i>0.101</i>
0) Write Output Data
1) Reset subset definitions to two levels
2) Increase set definitions to a <i>third</i> level
3) Main Menu
Select an Automatic Rule Generation Option

3.23 Reset Subset Definitions to Two levels

The Automatic Rule Generation mode is reset to two subsets defining each variable.

3.24 Increase Set Definitions to *Next* Level

Selection causes the current fuzzy subset descriptions for each variable to increase by one and the Automatic Rule Generation mode executed.

3.25 Display Default System Files

This option allows the user to display any of the current system default files on the console. These files are selected from the following lists;

- | | |
|------------------------------|---------------------------|
| 0) | Original Reference Data |
| 1) | Original Set Descriptions |
| 3) | Output Fuzzy Set Mapping |
| 4) | Defuzzified data |
| 5) | Formatted Reference Data |
| 6) | Formatted Generated Data |
| 7) | Formatted Error Data |
| 9) | Generated Rules |
| 10) | Dominant Rules |
| 11) | Input Knowledge Base |
| 12) | Current Set Definitions |
| 13) | Main Menu |
| Select a system file option: | |

3.26 **Interactive Switch Mode: ON**

This is an on/off switch which displays its current mode.

- | | | |
|--------------------------------|--------------------|-----|
| 0) | Interactive switch | ON |
| 1) | Interactive switch | OFF |
| Select a <i>Switch</i> option; | | |

It directs data generated, in the Manual Data to Knowledge mode, to and from default system files. When the switch is off, the user is not prompted for input/output for the Manual Data to Knowledge menu options.

3.27 **Data from Knowledge Base**

This module shares many of the features of the Knowledge from Data module. The main difference is that the Knowledge from Data module induces its knowledge base while the Data from Knowledge module reads a user-defined Knowledge Base. The menu list modules, shared with the manual Knowledge from Data module as follows:

- | | |
|-----------------------------------------|---------------------------|
| 0) | Read Input Knowledge Base |
| 1) | Read Input Sets |
| 2) | Write Output Data |
| 3) | Plot Sets |
| 4) | Plot Data |
| 5) | Do All |
| 6) | Main Menu |
| Select a Manual Rule Generation Option: | |

3.28 Read Input Knowledge Base

The input Knowledge Base is read from a file similar to Figure . The file has a similar format to the generated rules shown in table , the difference being that the *only* qualifier must be explicitly stated in the Input Knowledge Base. Similar output options are available as in the Manual Rule Generation mode as follows:

- | | | |
|---------------------------------|-------------------------------------|-------|
| 1) | Read knowledge base from file | zknow |
| 2) | Read knowledge base from a new file | |
| Select a knowledge base option: | | |

APPENDIX 4

Sample Default Files of Non-monotonic Data Surface

Table A4.1 *zdata* File (same as *zin*)

		X Variable										
		0	10	20	30	40	50	60	70	80	90	100
Y Variable	0	-100	-66.7	-40	-20	-6.7	0	6.7	20	40	66.7	100
	10	-90	-60	-36	-18	-6	0	6	18	36	60	90
	20	-60	-40	-24	-12	-4	0	4	12	24	40	60
	30	60	40	24	12	4	0	-4	-12	-24	-40	-60
	40	90	60	36	18	6	0	-6	-18	-36	-60	-90
	50	100	66.7	40	20	6.7	0	-6.7	-20	-40	-66.7	-100
	60	90	60	36	18	6	0	-6	-18	-36	-60	-90
	70	60	40	24	12	4	0	-4	-12	-24	-40	-60
	80	-60	-40	-24	-12	-4	0	4	12	24	40	60
	90	-90	-60	-36	-18	-6	0	6	18	36	60	90
	100	-100	-66.7	-40	-20	-6.7	0	6.7	20	40	66.7	100

Table A4.2 *zsets* File

Variable	Subset Name	Shape	Critical Point
X	low_x	half_bell_down	0
	mid_x	bell	50
	high_x	half_bell_up	100
Y	low_y	half_bell_down	0
	mid_y	bell	50
	high_y	half_bell_up	100
Z	low_z	half_bell_down	-100
	mid_z	bell	0
	high_z	half_bell_up	100

Table A4.3 *zmap* File

Set Name Discrete Value Discrete Member		X Domain			
		low_x	mid_x	high_x	
		0 x_0	50 x_5	100 x_{10}	
Y Domain	low_y	1	0	0	low_z
	0	0	1	0	mid_z
	y_0	0	0	1	high_z
	mid_y	0	0	1	low_z
	50	0	1	0	mid_z
	y_5	1	0	0	high_z
	high_y	1	0	0	low_z
	100	0	1	0	mid_z
	y_{10}	0	0	1	high_z

Table A.4.4 *zdom* File

		X		
		low_x	mid_x	high_x
Y	low_y	low_z	mid_z	high_z
	mid_y	high_z	mid_z	low_z
	high_y	low_z	mid_z	high_z

Table A4.5 *zrules* File (same as *zknow*)

Y		X	Z
<i>not</i>	mid_y	low_x	low_z
	mid_y	high_x	
<i>any</i>		mid_x	mid_z
<i>not</i>	mid_y	high_x	high_z
	mid_y	low_x	

Table A4.6 *zout* File

[illegible]

Table A5.2 Automatic Membership Functions for Controller

Variable	Subset Name	Shape	Critical Point
Screen Bin Level	S0	half_bell_down	0
	S1	bell	10
	S2	bell	20
	S3	bell	30
	S4	bell	40
	S5	bell	50
	S6	bell	60
	S7	bell	70
	S8	bell	80
	S9	bell	90
	S10	half_bell_up	100
Current Draw	C0	half_bell_down	35.0
	C1	bell	36.5
	C2	bell	38.0
	C3	bell	39.5
	C4	bell	41.0
	C5	bell	42.5
	C6	bell	44.0
	C7	bell	45.5
	C8	bell	47.0
	C9	bell	48.5
	C10	half_bell_up	50.0
Feed Rate Change	F0	half_bell_down	-50
	F1	bell	-40
	F2	bell	-30
	F3	bell	-20
	F4	bell	-10
	F5	bell	0
	F6	bell	10
	F7	bell	20
	F8	bell	30
	F9	bell	40
	F10	half_bell_up	50

Table A5.3 Automatic Dominant Subsets for Feed Rate Changes at Screen Bin Levels and Current Draw Values

[illegible]

Table A5.4 Automatic Final Rules for Controller

CURRENT DRAW	SCREEN BIN LEVEL	FEED RATE CHANGE
Any	S9	
Any	S10	
C10	Any	
between C6 C10	S8	F0
between C6 C9	S7	
between C3 C5	S8	
C9	between S0 S7	F1
between C3 C5	S7	
between C0 C2	S8	
C8	S6	F2
between C3 C7	S6	
between C0 C2	S7	
C8	between S0 S5	F3
between C6 C7	S3	
between C4 C7	S4	
between C2 C7	S5	
between C0 C2	S6	
C7	between S0 S5	F4
between C5 C6	S0	
between C5 C6	S1	
between C5 C6	S2	
between C4 C5	S3	
between C0 C3	S4	
between C0 C1	S5	F5
between C0 C3	S3	
C4	between S0 S2	F6
C3	between S0 S2	F7
C2	S2	F8
between C0 C1	S2	
C2	between S0 S1	F9
between C0 C1	S0	
between C0 C1	S1	F10

Table A5.5 Automaic Generated Database of Feed Rate Changes at Screen Bin Levels and Current Draw Values

[illegible]

		Screen Bin Level												Sum of	Error
		0.0	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0	Squares	Dist.	
Current Draw (amps)	35.0	0.0	0.0	-1.1	1.1	0.0	-3.7	-2.4	-1.7	0.0	0.0	0.0	25	1	
	35.5	0.0	0.0	-1.1	1.1	0.0	-3.7	-2.4	-1.7	0.0	0.0	0.0	25	1	
	36.0	0.0	0.0	-1.1	1.1	0.0	-3.8	-2.4	-1.9	0.0	0.0	0.0	26	1	
	36.5	0.0	0.0	-1.1	1.1	0.0	-3.9	-2.4	-2.0	0.0	0.0	0.0	28	2	
	37.0	2.2	2.2	1.1	1.1	0.0	-1.9	-2.4	-2.3	-1.0	0.0	0.0	28	2	
	37.5	4.9	4.9	4.6	0.7	-0.2	3.3	-2.6	-2.7	-2.0	0.0	0.0	99	6	
	38.0	-0.3	-0.3	2.0	-0.4	-0.9	4.9	-3.1	-3.1	-3.0	0.0	0.0	58	3	
	38.5	-4.6	-4.6	-2.7	-1.3	-2.0	3.9	-1.6	-1.5	-1.8	0.0	0.0	79	4	
	39.0	-1.0	-1.0	-2.9	-2.6	-3.5	2.8	3.0	3.2	2.8	0.0	0.0	65	4	
	39.5	-1.8	-1.8	-4.7	-4.2	-5.0	1.7	4.1	4.5	4.0	0.0	0.0	127	7	
	40.0	-3.3	-3.3	-5.6	-3.5	-4.1	0.6	3.7	3.3	2.4	0.0	0.0	113	6	
	40.5	-0.4	-0.4	-2.3	1.4	0.5	-0.1	3.2	1.8	0.3	0.0	0.0	22	1	
	41.0	-0.1	-0.1	-1.7	2.0	1.9	-0.8	2.5	0.0	-1.7	0.0	0.0	21	1	
	41.5	0.4	0.4	-0.9	1.6	1.4	-1.4	1.7	-1.7	-2.0	0.0	0.0	17	1	
	42.0	4.3	4.3	3.0	-0.5	0.9	-1.9	0.9	-3.3	-3.0	0.0	0.0	71	4	
	42.5	4.2	4.2	3.0	-2.2	0.5	-2.3	0.2	-4.6	-5.0	0.0	0.0	101	6	
	43.0	1.2	1.2	0.5	-1.9	0.2	-2.7	-0.3	-3.4	-3.9	0.0	0.0	41	2	
	43.5	-1.6	-1.6	-1.9	1.0	0.1	-2.9	-0.6	1.5	1.3	0.0	0.0	23	1	
	44.0	-3.5	-3.5	-3.9	2.2	0.0	-3.2	-0.7	3.5	3.2	0.0	0.0	78	4	
	44.5	-2.9	-2.9	-3.0	1.6	-0.7	-3.6	-1.2	3.5	3.2	0.0	0.0	65	4	
45.0	0.6	0.6	0.6	0.0	-2.4	-3.4	-2.8	3.5	3.2	0.0	0.0	49	3		
45.5	0.0	0.0	0.0	-2.1	-4.6	-4.8	-2.8	3.5	3.2	0.0	0.0	79	4		
46.0	-1.5	-1.5	-1.5	-2.4	-5.0	-5.6	-0.6	2.8	3.2	0.0	0.0	87	5		
46.5	-0.3	-0.3	-0.3	-0.3	-2.8	-3.2	4.3	2.8	3.2	0.0	0.0	54	3		
47.0	-3.2	-3.2	-3.2	-3.2	-4.5	-4.5	3.0	2.8	3.2	0.0	0.0	110	6		
47.5	-4.2	-4.2	-4.2	-4.2	-4.6	-4.6	2.6	2.8	3.2	0.0	0.0	137	8		
48.0	1.2	1.2	1.2	1.2	1.2	1.2	3.4	1.7	3.2	0.0	0.0	33	2		
48.5	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	-0.3	3.2	0.0	0.0	11	1		
49.0	-3.1	-3.1	-3.1	-3.1	-3.1	-3.1	-3.1	-3.1	3.1	0.0	0.0	87	5		
49.5	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0	1.2	0.0	0.0	9	1		
50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0		
Sum of Squares		169	169	198	121	180	301	183	226	218	0	0	1766		
Error Dist.		10	10	11	7	10	17	10	13	12	0	0	100		
Correlation Coefficient = 0.998															

Table A5.7 Stage 1 - Manual Membership Functions for Controller

Variable	Subset Name	Shape	Critical Point
Bin Level	S0	half_bell_down	10
	S1	half_bell_up	90
Current Draw	C0	half_bell_down	37.0
	C1	half_bell_up	50.0
Feed Rate Change	F0	half_bell_down	-50
	F1	half_bell_up	50

Table A5.8

Stage 1- Manual Error Report for Controller

		Screen Bin Level												Sum of	Error
		0.0	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0	Squares	Dist.	
Current Draw (amps)	35.0	0.0	0.0	-8.0	-26.4	-21.9	-3.7	9.5	15.8	16.9	0.0	0.0	1877	4	
	35.5	0.0	0.0	-8.0	-26.4	-21.9	-3.7	9.5	15.8	16.9	0.0	0.0	1876	4	
	36.0	0.0	0.0	-8.0	-26.4	-21.9	-3.8	9.5	15.6	16.9	0.0	0.0	1873	4	
	36.5	0.0	0.0	-8.0	-26.4	-21.9	-3.9	9.5	15.5	16.9	0.0	0.0	1868	4	
	37.0	0.0	0.0	-8.0	-26.4	-21.9	-4.1	9.5	15.2	15.9	0.0	0.0	1828	3	
	37.5	-2.6	-2.6	-10.0	-26.8	-22.1	-4.4	9.3	14.8	14.9	0.0	0.0	1869	4	
	38.0	-9.1	-9.1	-14.9	-27.9	-22.8	-5.1	8.7	14.4	13.9	0.0	0.0	2188	4	
	38.5	-16.4	-16.4	-21.8	-28.8	-23.9	-6.1	8.0	13.7	12.9	0.0	0.0	2869	5	
	39.0	-21.8	-21.8	-26.0	-30.1	-25.3	-7.2	7.1	13.0	11.9	0.0	0.0	3583	7	
	39.5	-24.4	-24.4	-27.3	-31.7	-26.9	-8.3	6.0	12.0	10.9	0.0	0.0	4036	8	
	40.0	-24.9	-24.9	-27.1	-33.2	-28.2	-9.4	5.5	10.8	9.3	0.0	0.0	4198	8	
	40.5	-23.7	-23.7	-25.6	-31.9	-29.2	-10.1	5.1	9.3	7.2	0.0	0.0	3908	7	
	41.0	-21.2	-21.2	-22.7	-29.1	-29.9	-10.8	4.4	7.5	5.2	0.0	0.0	3377	6	
	41.5	-17.9	-17.9	-19.2	-24.5	-30.5	-11.4	3.5	5.8	4.9	0.0	0.0	2735	5	
	42.0	-13.9	-13.9	-15.2	-20.9	-29.5	-11.9	2.7	4.2	3.9	0.0	0.0	2106	4	
	42.5	-10.0	-10.0	-11.2	-16.4	-23.7	-12.3	2.1	2.9	1.9	0.0	0.0	1322	3	
	43.0	-6.2	-6.2	-6.9	-11.5	-17.2	-12.7	1.6	1.9	0.7	0.0	0.0	717	1	
	43.5	-1.6	-1.6	-1.9	-6.7	-9.9	-12.9	1.3	1.2	0.4	0.0	0.0	324	1	
	44.0	3.9	3.9	3.5	-0.4	-2.6	-5.8	1.2	1.0	0.1	0.0	0.0	85	0	
	44.5	9.1	9.1	9.0	5.8	3.5	0.6	0.6	1.0	0.1	0.0	0.0	296	1	
45.0	13.3	13.3	13.3	10.4	8.0	7.1	-0.9	1.0	0.1	0.0	0.0	752	1		
45.5	16.0	16.0	16.0	13.9	11.5	11.3	3.2	1.0	0.1	0.0	0.0	1235	2		
46.0	17.4	17.4	17.4	16.4	13.8	13.3	8.3	0.3	0.1	0.0	0.0	1611	3		
46.5	17.4	17.4	17.4	17.4	15.0	14.5	12.0	0.3	0.1	0.0	0.0	1793	3		
47.0	16.1	16.1	16.1	16.1	14.8	14.8	12.4	2.1	0.1	0.0	0.0	1640	3		
47.5	14.0	14.0	14.0	14.0	13.6	13.6	13.0	5.4	0.1	0.0	0.0	1349	3		
48.0	10.9	10.9	10.9	10.9	10.9	10.9	10.9	6.9	0.1	0.0	0.0	882	2		
48.5	7.1	7.1	7.1	7.1	7.1	7.1	7.1	7.1	0.5	0.0	0.0	400	1		
49.0	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	1.9	0.0	0.0	102	0		
49.5	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.0	0.0	8	0		
50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0		
Sum of Squares		5489	5489	6991	13713	11851	2518	1598	2613	2443	0	0	52704		
Error Dist.		10	10	13	26	22	5	3	5	5	0	0	100		
Correlation Coefficient = 0.953															

Table A5.9 Stage 2 Manual Membership Functions

Variable	Subset Name	Shape	Critical Point
Bin Level	S0	half_bell_down	10
	S1	bell	30
	S2	half_bell_up	90
Current Draw	C0	half_bell_down	37.0
	C1	bell	39.5
	C2	half_bell_up	50.0
Feed Rate Change	F0	half_bell_down	-50
	F1	bell	0
	F2	half_bell_up	50

Table A5.10

Stage 2 Manual Error Report for Controller

		Screen Bin Level											Sum of	Error
		0.0	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100.0	Squares	Dist.
Current Draw (amps)	35.0	0.0	0.0	13.9	11.1	2.8	7.4	12.6	17.2	17.2	0.0	0.0	1130	3
	35.5	0.0	0.0	13.9	11.1	2.8	7.4	12.6	17.1	17.2	0.0	0.0	1128	3
	36.0	0.0	0.0	13.9	11.1	2.8	7.3	12.6	17.0	17.2	0.0	0.0	1124	3
	36.5	0.0	0.0	13.9	11.1	2.8	7.2	12.6	16.9	17.2	0.0	0.0	1116	3
	37.0	0.0	0.0	13.9	11.1	2.8	7.0	12.6	16.6	16.2	0.0	0.0	1071	3
	37.5	1.1	1.1	11.9	10.7	2.5	6.7	12.4	16.2	15.2	0.0	0.0	956	3
	38.0	5.7	5.7	7.0	9.6	1.8	6.0	11.9	15.7	14.2	0.0	0.0	835	2
	38.5	15.0	15.0	5.5	8.7	0.7	5.1	11.2	15.1	13.2	0.0	0.0	1109	3
	39.0	19.5	19.5	12.4	7.4	-0.7	4.0	10.3	14.3	12.2	0.0	0.0	1444	4
	39.5	18.2	18.2	15.3	5.8	-2.2	2.8	9.1	13.3	11.2	0.0	0.0	1327	4
	40.0	14.7	14.7	12.6	4.5	-3.6	1.8	8.7	12.2	9.6	0.0	0.0	943	3
	40.5	12.7	12.7	11.7	4.5	-4.5	1.0	8.2	10.7	7.6	0.0	0.0	743	2
	41.0	11.9	11.9	12.1	4.0	-5.3	0.3	7.5	8.9	5.5	0.0	0.0	640	2
	41.5	11.8	11.8	13.2	5.2	-5.0	-0.2	6.7	7.2	5.2	0.0	0.0	628	2
	42.0	12.2	12.2	14.5	5.2	-3.4	-0.8	5.9	5.6	4.2	0.0	0.0	629	2
	42.5	12.4	12.4	15.3	6.0	-1.3	-1.2	5.2	4.3	2.2	0.0	0.0	630	2
	43.0	12.3	12.3	15.9	7.0	1.4	-1.6	4.7	3.3	1.1	0.0	0.0	645	2
	43.5	12.9	12.9	16.5	7.8	4.6	1.6	4.4	2.6	0.7	0.0	0.0	715	2
	44.0	14.8	14.8	17.3	10.6	8.4	5.2	4.3	2.4	0.4	0.0	0.0	972	3
	44.5	17.6	17.6	18.6	14.3	12.0	9.1	3.8	2.4	0.4	0.0	0.0	1416	4
	45.0	20.2	20.2	20.2	17.3	14.9	14.0	4.5	2.4	0.4	0.0	0.0	1965	6
	45.5	21.6	21.6	21.6	19.5	17.1	16.9	8.8	2.4	0.4	0.0	0.0	2444	7
	46.0	21.8	21.8	21.8	20.9	18.2	17.7	12.7	1.7	0.4	0.0	0.0	2670	8
	46.5	20.8	20.8	20.8	20.8	18.3	17.9	15.4	1.7	0.4	0.0	0.0	2629	8
	47.0	18.6	18.6	18.6	18.6	17.3	17.3	14.9	4.6	0.4	0.0	0.0	2230	6
47.5	15.7	15.7	15.7	15.7	15.3	15.3	14.7	7.1	0.4	0.0	0.0	1723	5	
48.0	12.0	12.0	12.0	12.0	12.0	12.0	12.0	8.0	0.4	0.0	0.0	1076	3	
48.5	7.7	7.7	7.7	7.7	7.7	7.7	7.7	7.7	1.2	0.0	0.0	474	1	
49.0	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8	2.2	0.0	0.0	119	0	
49.5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	9	0	
50.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0	
Sum of Squares		5371	5371	6395	3924	2325	2381	2935	3265	2575	0	0	34542	
Error Dist.		16	16	19	11	7	7	8	9	7	0	0	100	
Correlation Coefficient = 0.965														

Table A5.11 Stage 6 Manual Membership Functions

Variable	Subset Name	Shape	Critical Point
Bin Level	S0	half_bell_down	10
	S1	bell	30
	S2	bell	70
	S3	half_bell_up	90
Current Draw	C0	half_bell_down	37.0
	C1	bell	39.5
	C2	bell	46.5
	C3	half_bell_up	50.0
Feed Rate Change	F0	half_bell_down	-50
	F1	bell	-20
	F2	bell	0
	F3	bell	10
	F4	bell	40
	F5	half_bell_up	50

Stage 6 Dominant Subsets for Manual Tests

APPENDIX 6 Table of C Source Code FLIX Files

File Name	Size (bytes)	File Description
config.h	2463	Definitions for Constants and Strings
defs.h	3405	Definitions for structures and enum types
extern.h	734	Definitions for global variables
addfuz.c	2626	Automatic fuzzy set additions
calcout.c	1170	Determine weighted average output
compact.c	2267	Minimize rules
console.c	5385	Write output to console
decision.c	6706	Determines appropriate qualifier
display.c	2839	Reads an ASCII file displayed on screen
dominant.c	1301	Select dominant rules
error.c	4191	Determines error indicators
function.c	8052	Defines fuzzy membership functions
globals.c	4150	Defines global variables and strings
inferenc.c	5936	Defuzzify conclusion rule set
input.c	7959	Read reference database
inputrul.c	6209	Read input knowledge base
memory.c	1721	Manage memory allocation
moredeci.c	4254	Determines appropriate qualifier
option.c	12130	Controls program flow and menu options
output.c	9776	Writes output database
plotdata.c	4092	Plot routines
proj.c	3363	Main program subroutines
prompt.c	7275	Prompts for menu options
read.c	6047	Read input fuzzy sets
select.c	10263	Returns degree of belief in a fuzzy set