NEURO FUZZY REASONING FOR PATTERN CLASSIFICATION AND OBJECT RECOGNITION

by

Jayati Ghoshal

Electronics and Communication Sciences Unit
Indian Statistical Institute
203,B.T.Road,Calcutta-700 035.
INDIA

Thesis submitted to Indian Statistical Institute for partial fulfilment of the requirements for the degree of Doctor of Philosophy.

March 1997

To my mother

Acknowledgements

As with my document that is a long time in the writing, there are probably more people to thank than there are pages. So, rather than trying to thank everyone I mention a few who are very close to me.

The greatest debt of gratitude to my supervisor Prof. Kumar Sankar Ray for showing me the light at the end of the tunnel i.e. my thesis. He needs special thanks for keeping track of my progress as well as mental sanity.

I would like to express my sincere gratitude to Prof. Bidyut Baran Chaudhuri for allowing me to use the computing facilities at Computer Vision and Pattern Recognition Unit at Indian Statistical Institute.

Special thanks to Mr. Tapan Kumar Dinda, Mr. Abul Houssain and Mr. Swapan Raha for their constructive criticism on several chapters (especially chapter 2) of the thesis.

I wish to thank all of my friends at Indian Statistical Institute (especially Mr. Ujjwal Bhattacharya and Mr. Tamaltaru Pal) for their help during the course of the work.

I am grateful to Mr. Sunil Chakraborty for dedicating many hours for the preparation of the diagrams without which the thesis is incomplete. I must thank Prof. N. R. Ganguli for helping me with the speech data.

Vote of thanks must be given to all the members of Electronics and Communication Sciences Unit. I thank each of them for their individual gift in making this department unique.

I express my indebtedness to my elder brother Mr. Arunabha Ghoshal for letting me complete this thesis without worrying much about the financial condition of our family.

Finally, I express my sincerest thanks to the authorities of Indian Statistical Institute for providing administrative and financial support and also allowing me to complete my thesis in an wonderful environment which is so full of good students and renowned scientists.

Contents

Chapter 1

Introduction

In real world, pattern classification and object recognition problems are faced with fuzziness that is connected with diverse facets of cognitive activity of the human being. An origin of sources of fuzziness is related to labels expressed in feature space as well as to labels of classes taken into account in classification and /or recognition procedures. Though a lot of scientific efforts have already been dedicated to pattern recognition problems, especially to classification procedures, still pattern recognition is confronted with a continuous challenge coming from a human being who can perform lot of extremely complex classification tasks by some sort of mental reasoning which can not be represented, in straightforward way, through computer algorithm. But, fuzzy set provides a plausible tool for modeling and mimicking cognitive processes of the human reasoning, especially those concerning recognition aspects. Fuzzy reasoning, proposed by Zadeh is one such tool. So far we have seen many successful applications of fuzzy reasoning [54,61,62,108] to the design of fuzzy logic controllers [8,70].

But, fuzzy reasoning approach can also be applied very successfully to pattern classification [33,34,35,36,115] and occluded object recognition problems where multiple classifications of a pattern and/or object is desired. In conventional approaches to pattern classification/ object recognition [55,56,57,58,85,86,87,97] such multiple classification, which is essentially needed due to the overlap of features of the patterns/ objects, is not considered.

There are two main approaches to pattern classification, namely the decision theoretic and the syntactic. Since, the fuzzy reasoning approach to pattern classification is similar to the decision theoretic method of pattern classification, we will first briefly describe the basic concept of the decision theoretic approach to pattern classification and then subsequently try to establish the similarity between the decision theoretic approach and the fuzzy reasoning approach to pattern classification.

Under decision theoretic approach, each pattern is represented by a vector of features. The pattern space is divided into a number of regions, each of which represents a prototype pattern or a cluster of patterns. A decision function maps the given patterns to previously determined regions.

In the fuzzy reasoning approach to pattern classification each element of the feature vector / pattern vector is represented by the fuzzy linguistic variable instead of a real number. For instance, suppose we have a (2×1) feature vector / pattern vector $F = (F_1, F_2)^T$, T is transpose where F_1 is the first formant frequency of a speech signal and F_2 is the second formant frequency. In the conventional decision theoretic approach to pattern classification, F_1 and F_2 are two features and are represented by, say, 800 Hz and 550 Hz. Whereas in the fuzzy reasoning approach to pattern classification F_1 and F_2 are represented by the fuzzy linguistic variables, e.g., F_1 is small and F_2 is medium. The elements of the feature vector / pattern vector constitute the antecedent part of the fuzzy implication. The consequent part of the fuzzy implication is a fuzzy set which represents the possibility of occurrence of different classes of patterns on the pattern space. Thus, fuzzy **If Then** rules, which map the given patterns to previously determined regions, may be used for pattern classification / object recognition problems.

The process of classification is usually divided into two steps, learning and recognition which sometimes overlap. The main stages in the learning process are feature extraction, feature selection, clustering and determination of the appropriate fuzzy **If Then** rules which will constitute a decision function.

The main stages in recognition are : extraction of a selected set of features, application of **If Then** rules and decision making based on the results of the application of the **If Then** rules. In the classification process, first an unknown pattern is presented to the system, then a set of predetermined features are extracted from the pattern. Finally, a set of **If Then** rules determines the possibility of occurrence of different classes of patterns on the pattern space.

The object recognition scheme, proposed in this thesis is a model based system in which recognition involves matching of the input image with a set of predefined models of objects. In such a system the known objects are precompiled, creating a model database and this database is used to recognize objects in an image scene.

Existing object recognition methods [4,55,56,57,58,85,86,87] can be categorized as either global or local in nature. Global methods are based on global features of the boundary or of an equivalent representation. Such techniques are the Fourier descriptors, the Moments and methods based on Autoregressive models. Local methods use local features such as critical points or holes and corners. They perform very well in the presence of noise, distortion or partial occlusion since such effects on an isolated region of the contour alter only the local features associated with that region, leaving all other local features unaffected. However, the choice of representative local features is not trivial and the recognition process based on local features is computationally more intensive and time consuming. On the other hand, global methods have disadvantage that a small distortion in a portion of a boundary of an object will result in changes to all global features. The occluded object recognition scheme (based on the local features of the model and scene) presented in this thesis is similar to the decision theoretic approach to pattern classification.

The basic motivation behind the use of fuzzy reasoning approach to pattern classification and / or occluded object recognition is to mimic the cognitive process of human reasoning for classification.

In this thesis first we study the potentiality of the existing fuzzy reasoning approach [61,62,108,109] to pattern classification problems. Subsequently we consider multidimensional fuzzy reasoning (MFR) [91] for pattern classification and / or occluded object recognition problems and introduce a new interpretation to multidimensional fuzzy implication (MFI). We demonstrate that a particular type of interpretation of MFI reduces to the existing fuzzy reasoning approaches [61,62,108,109] which essentially deal with one dimensional fuzzy implication. The alternative interpretation, which has been modified in the thesis to deal with the pattern classification and / or occluded object recognition problems in a more meaningful way, essentially preserves the notion of multidimensionality of fuzzy reasoning. Both the interpretations are implemented on backpropagation type neural network. In doing this we state the following problems [28,92,93] of the existing fuzzy reasoning approaches [61,62,91,108,109].

1. the lack of a definite method to determine the membership function
2. the lack of a definite method to determine the appropriate translating rule to translate a fuzzy implication to a relational matrix
3. the lack of a definite method to determine the appropriate compositional rule of inference and
4. the lack of a learning function or adaptability.

Instead of questioning the lack of a definite method to determine the membership function of a fuzzy set as mentioned by few researchers [28,92,93]; in this context we provide the following argument.

According to Zadeh [108] membership is the quantification of human perception about the situation at hand. And from psychological evidence perceptions about a particular situation achieved by different individuals are not unique; but they are close to each other provided the situation is perceived by reasonably experienced / expert individuals. Hence, it is quite natural to represent such perception through membership assignment of fuzzy set which captures the vagueness of the situation and the corresponding variations in perceptions. For instance, if we have a finite universe of discourse, say $X = \{1, 2, 3, 4\}$ and if we state X is "big" then the "bigness" can be perceived by the following assignment of membership to the fuzzy set by

$$\text{membership function, } \mu_x = \{.2/1 + .5/2 + .7/3 + 1/4\}.$$

Now alternatively we may have,

$$\text{membership function, } \mu_x = \{.1/1 + .4/2 + .6/3 + .9/4\} \text{ etc.}$$

which is different from and close to the earlier assignment of membership function to the fuzzy set "big". But we will never represent "big" X by the following membership function,

$$\mu_x = \{1/1 + .7/2 + .5/3 + .2/4\}$$

which is a representation of wrong perception about the particular situation. Sometimes to fit the situation more accurately we may need to tune the membership function further through some heuristic means by modifying shapes of triangular or trapezoidal function. Thus, whatever may be the method of determining the membership function we can successfully utilize them in many application domains of fuzzy reasoning, specially the fuzzy logic controller [8].

But sometimes the lack of an appropriate choice for translating rule which translates a fuzzy implication to fuzzy relational matrix and compositional rule for inference

Table 1.1: Features to be fused

Difference	Fuzzy Logic: Logic based approach Neural Network (NN): Learning function based approach
Similarity	(1) Output values of NN and membership function (2) Multiplication and addition operation in NN and MAX-MIN operation in fuzzy logic

affect the success of fuzzy reasoning to a great extent. Though there are some thumb rules [8] to select them (translating and compositional rules), they are not well supported by some theories. Hence, in the present work we try to avoid these two rules for fuzzy reasoning and try to introduce the feature of adaptability (learning) in fuzzy reasoning. And we achieve these goals through the implementation of the method of fuzzy reasoning on backpropagation neural network.

Both neural network and fuzzy theories attempt to model the human like behavior. Such attempts began spontaneously and rapidly at the same time. The similarities and mutual compensations between these, therefore, are quite useful. Hence, the methodology of the thesis is based on the individual merits and the similarities between these two (See Table 1.1)

Fuzzy logic can express logic explicitly taking a form of rule. Neural network is effective for pattern classification because of its learning function. From these advantages and the similarities stated in Table 1.1 we can generate fusion technologies of the following types, (i) learning function to fuzzy logic, (ii) incorporate logics in neural network structure. The fusion of the first kind is achieved by learning the relational matrix (R) (R represents relation between the antecedent clause(s) and the consequent clause of a fuzzy **If Then** rule) using the generalized delta rule of backpropagation neural network [69]; instead of realizing the said relational matrix (R) using Zadeh's arithmetic rule, Zadeh's maximum rule, Gödelian logic etc. The fusion of the second kind is achieved by exploiting the similarity that the MIN operation of input and fuzzy variables conducted at each proposition of IF parts of fuzzy inference rule corresponds to a product of input to the neuron and synaptic weights, and the MAX operation to obtain a final inference value from the THEN part of these plural inference rules corresponds to the input sum within neuron (see Figure 1.1(a,b)). In both the fusion methodologies we further superpose the first similarity (1) (see Table 1.1) to give a membership function to neural network outputs without causing a crisp boundary between classes formed by a pattern classification type neural network [69]. The neural network's learning scheme which is built in both the fusion methodologies produces less ambiguous class boundary than the existing logic based fuzzy reasoning for pattern classification (see Figure 1.1(c,d)) [14].

Recent interest in a combination of the neural networks and genetic algorithm paradigms arises from the wish to make use of the combinatorial power of genetic algorithm to enhance the flexibility of adaptation of the networks. Behind this motivation lie, of course, the biological roots both approaches have in common [119,120,121].

In this thesis the main application of genetic algorithms (GA) in neural networks (NN) has been the usage of the genetic algorithm as a learning rule instead of back-propagation algorithm to train the weights of feed forward networks. Thus the overall aim of the thesis is to present some results of the investigations that strengthen the effectiveness of fusion technology for handling uncertainty in the field of pattern classification and occluded object recognition problems. We propose fusion technologies for this purpose and study the merits and demerits of the said fusion technology.

Now, in the following sections, we give the chapterwise details of the thesis.

In Chapter 2, we first review the method of fuzzy reasoning and use it to design a pattern classifier. The performance of the proposed classifier is compared with con-

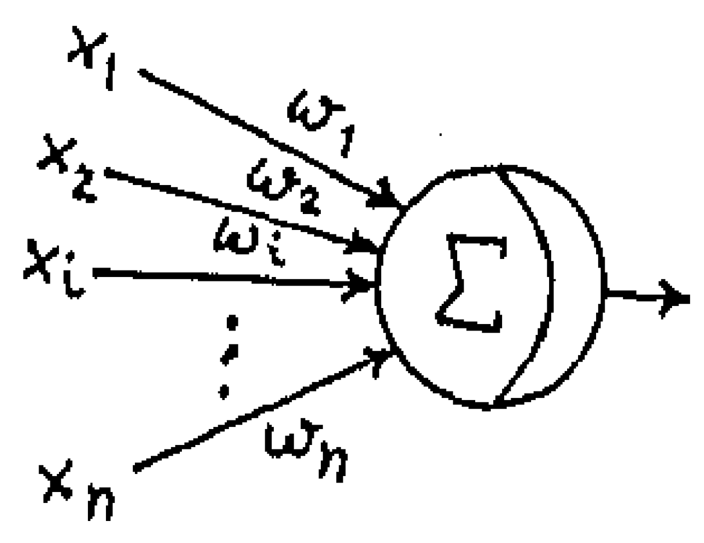


Fig. 1.1(a): Sum of Products of Neuron

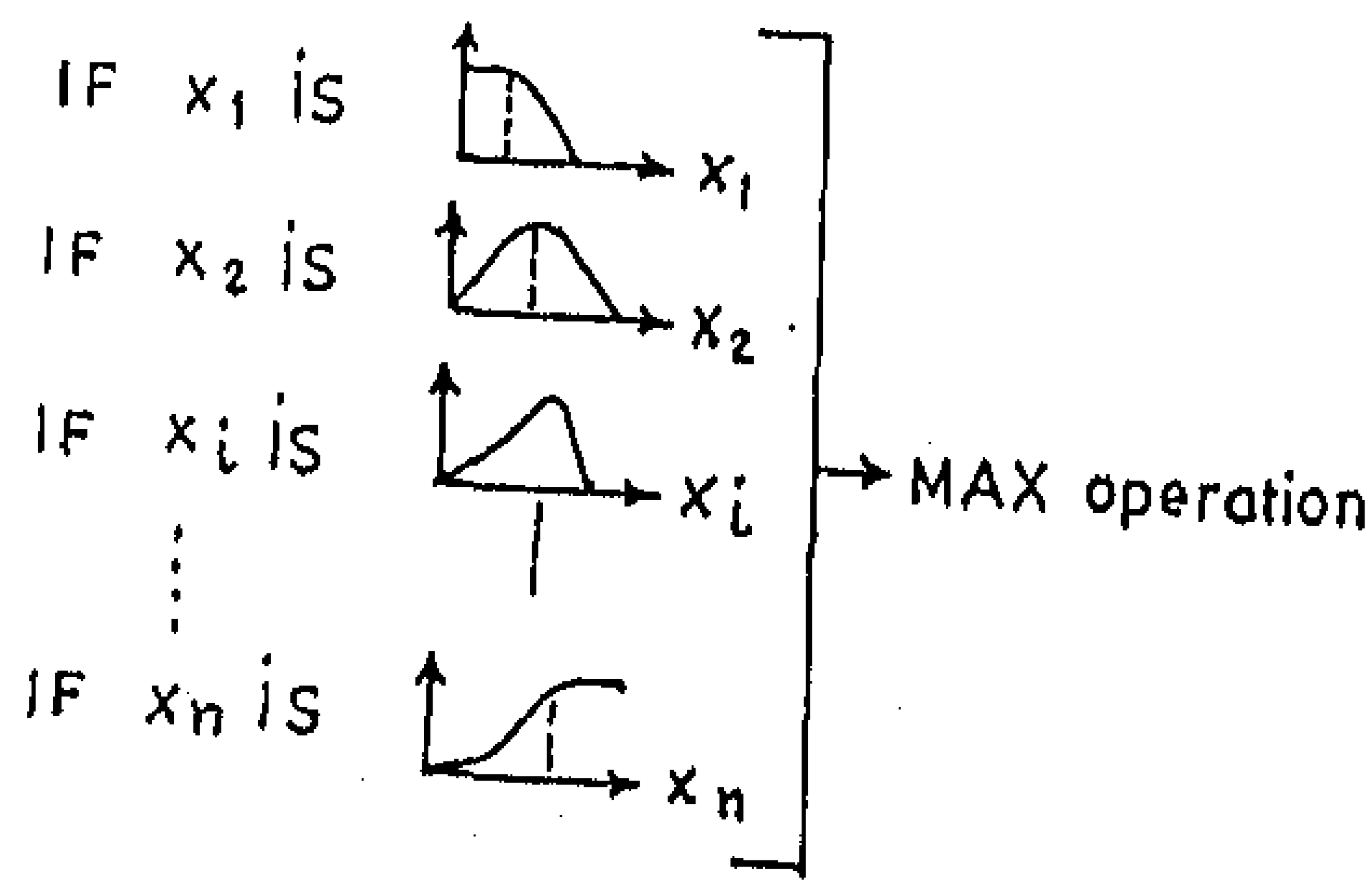


Fig. 1.1(b): MAX - MIN operation of Fuzzy Reasoning

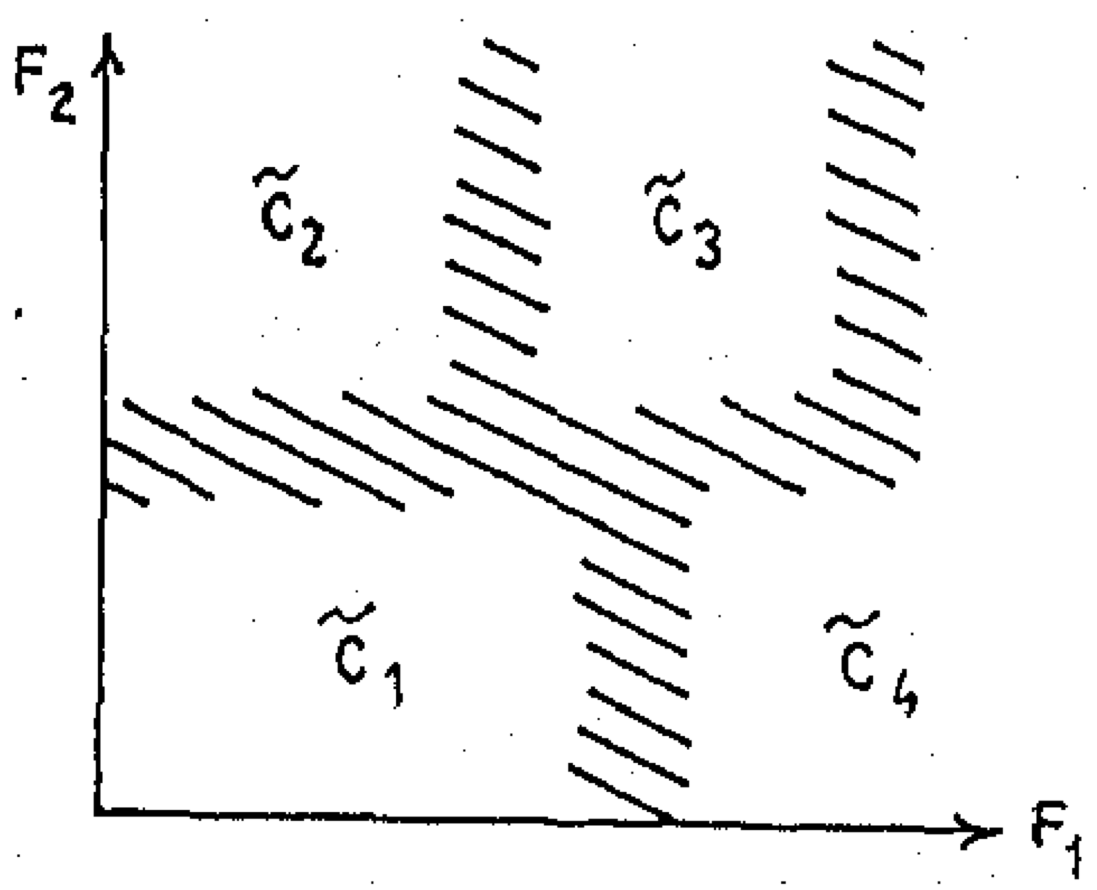


Fig. 1.1(c): Partition of the pattern space by ordinary fuzzy rules where F_1 and F_2 are two features of a pattern vector

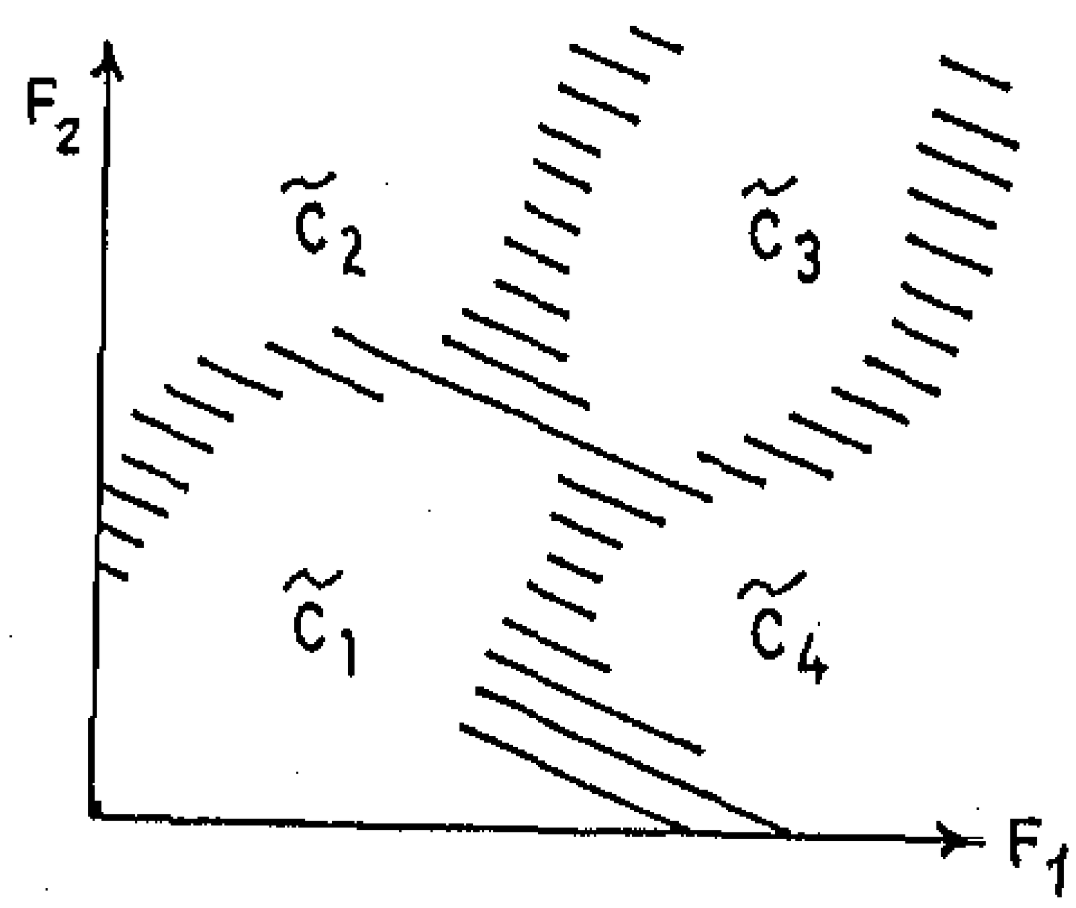


Fig. 1.1(d): Partition of the pattern space by neural network where F_1 and F_2 are two features of a pattern vector

ventional Bayesian approach. Subsequently, we discuss the two possible interpretations of multidimensional fuzzy implication (MFI). We demonstrate that one conventional interpretation of MFI reduces to the existing approach to fuzzy reasoning which is proposed by Zadeh and Mizumoto and based on which, in the earlier part of this chapter, we design one pattern classifier. The second interpretation of MFI has been modified in this chapter and a notion of fuzzy feature vector / pattern vector has been introduced. In the subsequent chapters we demonstrate the effectiveness of the newly proposed (i.e. the MFI) interpretation of MFI and the concept of fuzzy feature vector / pattern vector for pattern classification and object recognition problems. At the end of this chapter, we propose a new approach to fuzzy reasoning where some of the antecedent clauses of the given premise may not be available at the time of inferring or reasoning. Under such circumstances, we assume some approximate linguistic values of those antecedent clauses by default. Detail classifier design based on the default values of the features is yet to be developed.

In Chapter 3, we generate our first fusion methodology based on the conventional interpretation of MFI. Such conventional interpretation of MFI is similar to fuzzy reasoning and/or extended fuzzy reasoning as proposed by Zadeh [108] and Mizumoto [61,62]. We implement this conventional interpretation of MFI on backpropagation neural network, with an ambition to overcome some of the existing difficulties of fuzzy reasoning as stated earlier. We have tested our proposed scheme on several synthetic data and compared the performance of our classification with the Bayesian classifier.

In Chapter 4, we consider the new interpretation of MFI and implement that on backpropagation type neural network. Here, the scheme is tested on some synthetic data and some real vowel classification data. The performance of the proposed scheme is compared with some of the existing techniques of classification. At last, we consider the aspect of the management of uncertainty in pattern classification. One kind of uncertainty in pattern classification which is tackled by fuzzy reasoning method as stated earlier, occurs due to the variations of feature values caused by external noise, errors in the sensors etc. In addition to this stated uncertainty, we have realized another kind of uncertainty which occurs when we consider classification of same vowels (of some Indian language) uttered by different speakers and which can not be handled by fuzzy reasoning method alone. To tackle such uncertainties, which essentially occurs due to the variations of the vocal tract dynamics and tongue position of individuals, we use the concept of fuzzy masking which is experimentally designed in the present thesis and for which some mathematical basis is yet to be developed.

In Chapter 5, we consider recognition of occluded 2-dimensional objects using the neuro fuzzy fusion methodology which is already stated in Chapter 4. The results of Chapter 4 and Chapter 5 demonstrate that a unified framework based on the new interpretation of MFI, which is realized through backpropagation type neural network, is possible for pattern classification and object recognition problems. The performance of the proposed scheme is tested through recognition of unknown scene consisting of different two-dimensional occluded objects.

In Chapter 6, we develop fusion among fuzzy concept, neural network concept and genetic concept [119,120,121]. The fusion which we have already stated in the earlier chapters (i.e. Chapter 4 and Chapter 5) has further been enriched by genetic concepts in the present chapter. To tackle the pattern classification problems, first we consider the new interpretation of MFI and then realize the new interpretation through multilayer perceptron. The learning scheme of the network is based on the genetic algorithm. We apply the enriched concept of fusion to occluded object recognition problem. Subsequently, a weight 'smoothing' scheme is proposed to increase the neural network generalization capability. The 'smoothing' constraint is incorporated into the objective function of the network to reflect the neighborhood correlation and to seek those solutions which have 'smooth' connection weights. The performance of the proposed scheme is tested through synthetic data and real life vowel classification data of one

Indian language.

In Chapter 3 to 6, we essentially consider fusion of the first kind by introducing learning function to fuzzy logic. But in Chapter 3 we use the conventional interpretation of MFI. Whereas in Chapter 4 to Chapter 6, we use the modified interpretation of MFI. In Chapter 4 and 5, we use backpropagation algorithm for learning the weights of the neural network. But in Chapter 6 we replace the backpropagation concept by the genetic learning algorithm.

In Chapter 7, we consider fusion of the second kind, that is, we try to incorporate logic in the neural network structure. The fusion of this chapter is also based on the modified interpretation of MFI which has been already considered in Chapter 4 to 6. The success and failure stories of the fusion technology of the second kind is demonstrated through experimental studies.

In Chapter 8, we draw the conclusion and propose some areas for future work.

Chapter 2

Fuzzy Reasoning: A textbook example of how theory becomes practice in pattern classification

Fuzzy reasoning approach to pattern classification consists of linguistic rules tied together by means of two concepts: fuzzy implication and a compositional rule of inference. In this chapter [80], first we study the applicability of different laws of fuzzy implication to pattern classification problem and compare their performances over a set of synthetic data. We use the most applicable (according to our study) laws of implication for vowel classification of three different Indian languages and obtain very promising results. Subsequently, in this chapter we review the present state of art of multidimensional fuzzy reasoning (MFR) based on multidimensional fuzzy implication (MFI) and discuss two possible interpretations of MFI [82,83]. We demonstrate that the existing approach to fuzzy reasoning, which is discussed from section 2.1 to 2.3 and based on which we design one pattern classifier is derived from the conventional interpretation of MFI. The other interpretation of MFI, which is different from the conventional interpretation, (i.e. models of section 2.1 to 2.3) is further modified. We also introduce a notion of fuzzy pattern vector / feature vector [82,83]. In the later chapters we demonstrate that the notion of fuzzy pattern / feature vector and the modified MFR based on the newly proposed interpretation of MFI are very effective for pattern classification and / or occluded object recognition problems [82,83]. At the end of this chapter, we develop a new model for fuzzy reasoning which deals with some default values of the features in the antecedent part of the fuzzy rule and which is proposed to be useful for classification of pattern vector / feature vector having some missing components due to channel failure etc. We illustrate the effectiveness of the newly proposed model for fuzzy reasoning with the help of a synthetic example. Finally we apply the newly proposed model for fuzzy reasoning for classification of Telugu vowels whose second formant frequency (F_2) is intentionally suppressed. We compare the performance (in terms of recognition score) of the newly proposed model with those of existing methods and state the scope for future work. Incidentally, note that the said newly proposed model for fuzzy reasoning is based on the conventional interpretation of MFI.

2.1 Passage between conventional approach to pattern classification and fuzzy reasoning approach to pattern classification

For simplicity of discussion and / or demonstration, let us restrict ourselves to the problem of pattern classification on $\Re^2$. Without lack of any generality, all the discussions and / or demonstrations will be valid for the problem of pattern classification on $\Re^n$.

In conventional approach to pattern classification on $\Re^2$, we usually have two feature axes (say F_1 and F_2). Depending upon the limit of the operating range of features, we obtain a finite range of pattern space formed by the finite length of each feature axis (see Figure 2.1).

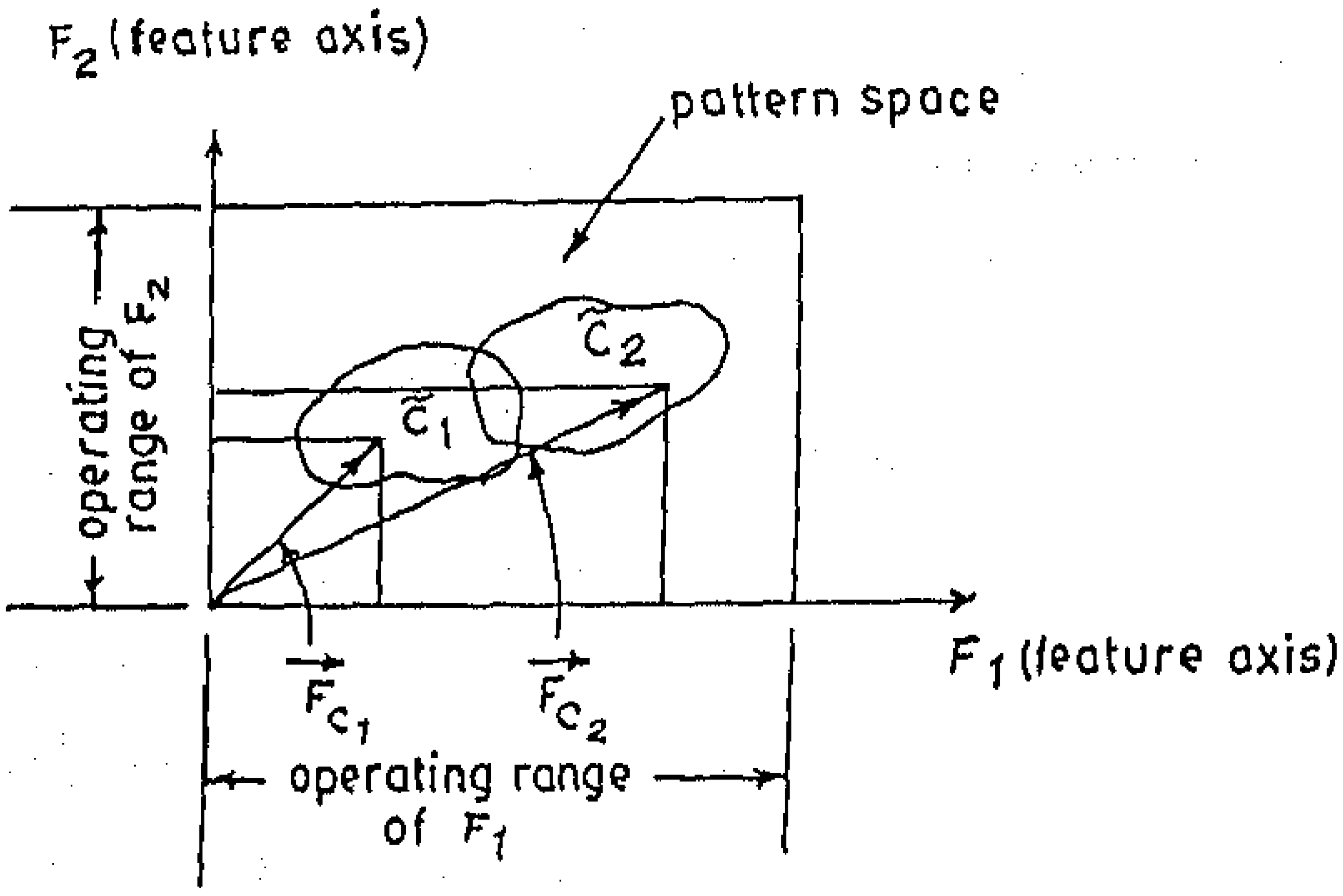


Fig. 2.1: Representation of pattern classes on $\Re^2$

Now, looking at the scatter diagram of the given data set we get an idea of how the data are grouped together and accordingly each group of data is labeled by a particular class say $\tilde{C}_i$, i = 1,2, ... Each data (pattern) of each class is represented by a pattern vector / feature vector. For instance the data (pattern) P on the pattern space is represented by the following pattern vector / feature vector (see Figure 2.26(a)).

$$\vec{F_{cj}} = \begin{bmatrix} F_{11} \\ F_{22} \end{bmatrix}, j = 1, 2, \ldots.$$

The task of the conventional approach to pattern classification is to classify each vector $\vec{F_{cj}}$ to one of the classes $\tilde{C}_i$ on $\Re^2$ [97].

As the primary concern of fuzzy reasoning approach to pattern classification is to mimic the cognitive process of human reasoning for pattern classification, we try to imitate the way human being perceive different classes of objects (patterns) based on some rough (inexact) informations of certain parameters (features). For instance, human being can easily distinguish between a poor person and a rich person just by looking at the individual's standard of living which can not be measured explicitly by any specific scale but can be indirectly estimated by considering the area where the person lives, the kind of food he / she takes, the kind of education his / her family takes, the kind of clothes he / she wears, the kind of commodities he / she uses etc. And such information (knowledge) can be represented by a rule (law of implication) as given below.

R_1 : if standard of living of a person is high then the person is rich,

R_2 : if standard of living of a person is low then the person is poor ;

where standard of living high, standard of living low etc. are linguistic terms and rich, *poor are different classes.* Each *primary* linguistic *term* (i.e. high / low etc.) is associated with a term set which is finite and where each primary term in the term set is defined on the same universe of discourse. The said universe of discourse is partitioned (in an overlapped manner) by the finite elements of the term set. For instance, if we range the standard of living between 0 to 10 then the Figure 2.2 explains how the

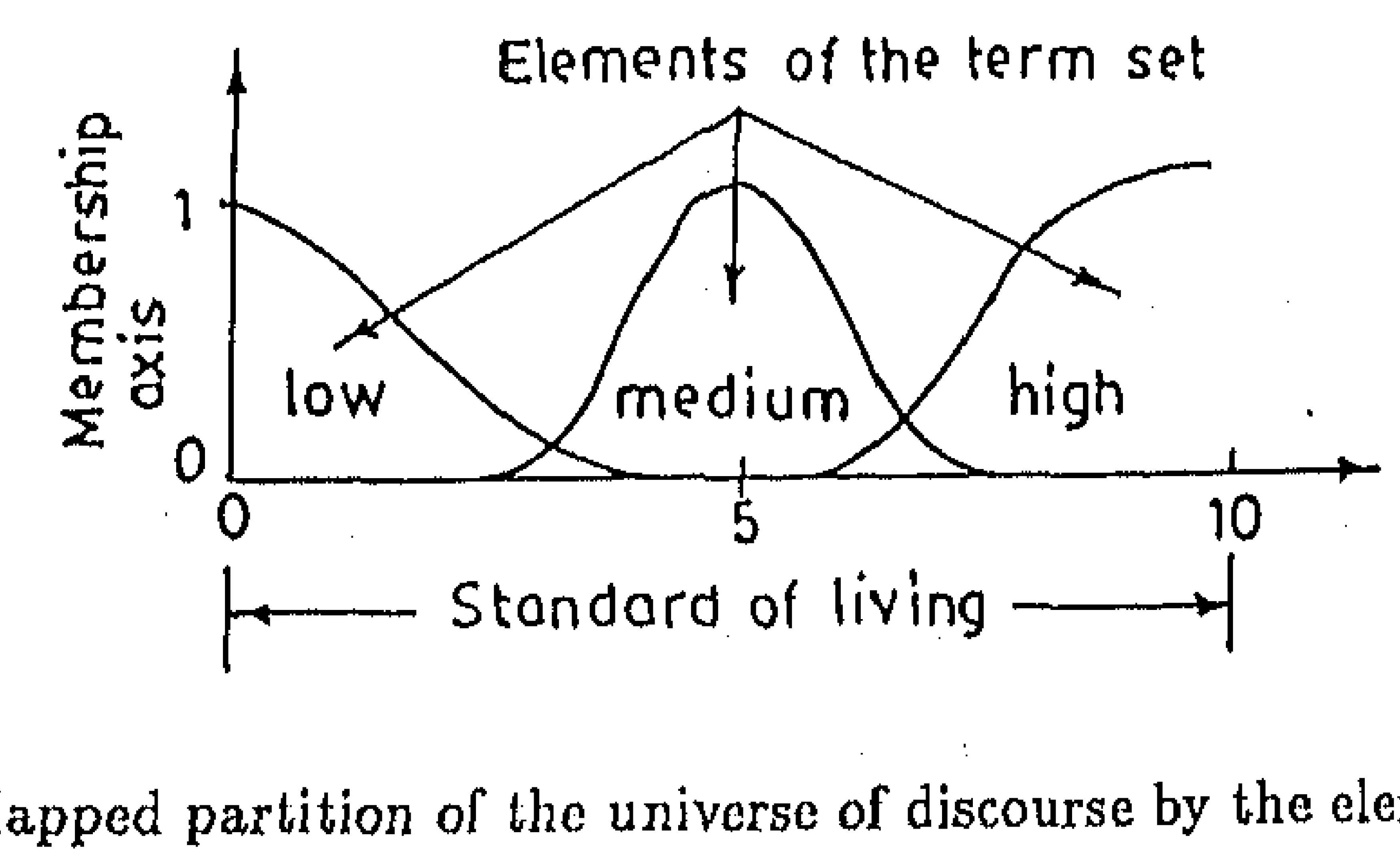


Fig. 2.2: Overlapped partition of the universe of discourse by the elements of the term set

elements of the term set partition the universe on which each element of the term set is defined.

Each primary term of the term set is represented by a fuzzy set. Between partition there is an overlap which indicates the degree of uncertainty of the elements of the said universe to become member of either of the fuzzy set (fuzzy set of low / fuzzy set of medium / fuzzy set of high).

Now, instead of having one dimensional implications (i.e. R_1 and R_2) we can have multidimensional implication for representing our knowledge (information). For instance,

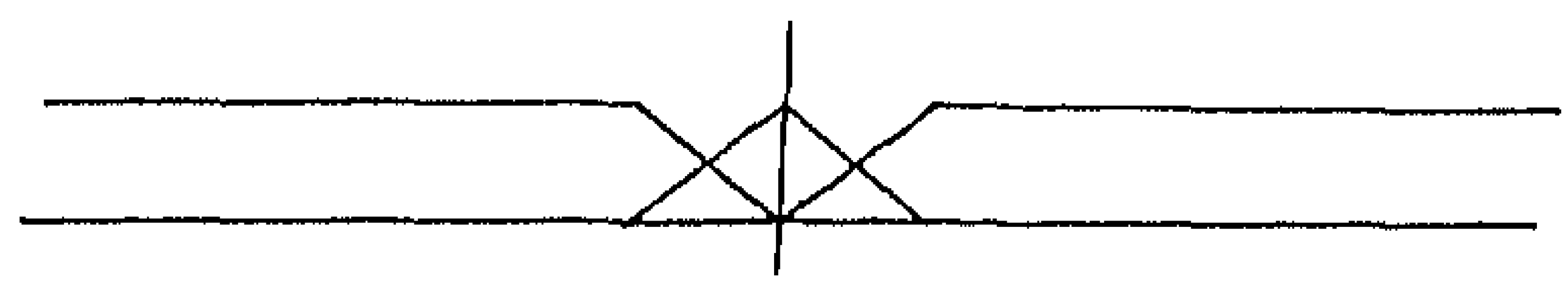
R_3 : if (behavior of a person is smart, appearance of a person is beautiful) then he / she becomes a candidate for interview of a personal assistant of a firm.

Usually, the above type of rule is read as,

R_4 : if behavior of a person is smart and appearance of a person is beautiful then he / she becomes a candidate for interview of a personal assistant of a firm.

Such one dimensional implication (i.e. R_4) is a kind of interpretation (see Equation (2.14(a))) of the said multidimensional form (i.e. R_3) of an implication.

Here, according to the one dimensional form of an implication, we have two antecedent clauses (e.g. smart behavior and beautiful appearance) which can be represented by two fuzzy sets defined over two different universe of discourses. In the consequent part, we always have single clause which can be represented by fuzzy set defined over a finite universe of different classes. The cardinality of the term set of each antecedent clause of an implication determines the number of rules that can be generated. For instance, if we have two antecedent clauses in an implication each having the cardinality 3 (say, low, medium and high), then we will have total $3 \times 3 = 9$ rules (see Figure 2.13). Note, that the cardinality of a term set defined over a given universe is not unique. Depending upon the need of the problem it is determined. It simply indicates the granularity (see Figure. 2.3) by which we want to partition the given universe to facilitate our representation of perception about grouping of objects (patterns).


(a)

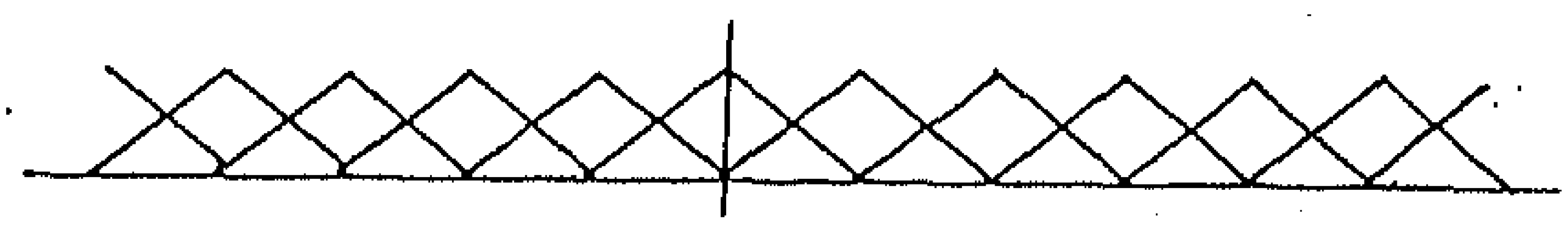

(b)

Fig. 2.3: Granularity of partition of universe

Depending on whether the universe of discourse is continuous or discrete, we can define the fuzzy sets of the antecedent clause of an implication by two ways. In case universe is continuous, we may go by functional definition, e.g. a bell shaped function (see Figure 2.2), triangle shaped function (see Figure 2.3), trapezoid shaped function or any arbitrary shaped function (see Figure 2.13 to 2.17). In case of discrete universe, we may go by numerical definition. In this case, the grade of membership function of a fuzzy set is represented by an one dimensional array of numbers. The length of the array depends on the degree of discretization. For example, see Table 4.2, 4.4 etc. of Chapter 4.

Discretization of a universe of discourse is frequently referred to as quantization. In effect quantization discretizes a universe into a certain number of segments (quantization levels). Each segment is labeled as a generic element and forms a discrete universe. A fuzzy set is then defined by assigning grade of membership values to each generic element of the new discrete universe (see Table 4.2,4.4 etc. of Chapter 4).

The consequent clause of an implication basically represents different classes of objects (patterns) existing over the finite range of the pattern space as stated earlier. The possibility of occurrence of different classes of patterns in the pattern space under a particular observation (it may be imprecise observation, like feature F_1 is high and feature F_2 is low, etc.) may be represented by a fuzzy set defined over the pattern space which is treated as universe.

Now, we try to give a more meaningful discussion on the correspondence between conventional approach to pattern classification and fuzzy reasoning approach to pattern classification.

In conventional approach, position of each pattern (say, P) on the finite range of the pattern space is represented by a pattern vector / feature vector $\vec{F_{cj}}$ and we always try to discriminate among patterns by classifying the pattern vector / feature vector $\vec{F_{cj}}$ [97].

Therefore, from the given data set (i.e. the training data set), we always know where the patterns are located and then try to separate them by some appropriate decision function. Subsequently, we use the said decision function to classify the test pattern vector / feature vector.

But, if we go by mimicking the cognitive process of human reasoning for pattern classification then the problem is, from a given set of imprecise observations stated in terms of fuzzy **If Then** rules, how to represent the patterns on the pattern space for developing suitable inferencing technique for classification. As the observations are given in terms of fuzzy **If Then** rules, it is clear that at every observation (rule) features are given in the form of fuzzy set defined over the feature universe. Therefore, in such case we can not represent / locate individual pattern by a vector. Instead, from a given observation (rule) we can represent / locate a collection of population of patterns in an area (in case of $\Re^2$; but region in general) on the pattern space by fuzzy pattern vector / feature vector (see Figure 2.26(a)) which is a very natural representation of the antecedent part of a multidimensional implication (i.e. R_3) represented by a fuzzy set, i.e. multidimensional fuzzy implication (MFI) (also see Section 2.9).

Now, we stipulate the following definition.

Definition 2.1: Let $\vec{F_{fj}}$ be a fuzzy vector having n components each of which is a fuzzy set A_i defined over the universe F_i. The fuzzy vector $\vec{F_{fj}}$ is a fuzzy set in the Cartesian product $F_1 \times F_2 \times ... \times F_n$. Each element of the fuzzy set is a vector having same initial point, but different terminal points. Each terminal point of each vector in the set carries one membership value indicating its (vector's) possibility to belong to the set $\vec{F_{fj}}$. A fuzzy vector $\vec{F_{fj}}$ is represented as,

$$\vec{F_{fj}} = \{(\mu_{\vec{F_{fj}}}(\vec{F}), \vec{F}) \forall \vec{F} \in F_1 \times F_2 \times ... \times F_n\}$$

where $\mu_{\vec{F_{fj}}}$: $F_1 \times F_2 \times ... \times F_n \rightarrow [0,1]$ is the membership function of $\vec{F_{fj}}$ and $\mu_{\vec{F_{fj}}}(\vec{F})$ is the grade of membership of $\vec{F} \in F_1 \times F_2 \times ... \times F_n$ in $\vec{F_{fj}}$ which is a set of vectors.

The process of defuzzification of the fuzzy vector $\vec{F_{fj}}$ is performed based on selecting the elements of the fuzzy vector $\vec{F_{fj}}$ (which is a fuzzy set) having highest membership values. The defuzzified version of the fuzzy vector $\vec{F_{fj}}$ is a fuzzy vector which is a fuzzy set. In case the defuzzified version of the fuzzy vector $\vec{F_{fj}}$ represents a fuzzy set which is a fuzzy singleton then the defuzzified version of $\vec{F_{fj}}$ becomes the crisp version as stated in Section 2.1 (see Example 2.1). The fuzzy set A_i as mentioned in the definition 2.1 is an element of the term set as discussed in section 2.1 (see Figure 2.2 and discussion of the last part of the fourth paragraph of section 2.1). The universe of the components of $\vec{F_{fj}}$, i.e. the fuzzy set A_i, may be continuous / discrete and the universe of $\vec{F_{fj}}$ may be continuous / discrete. In case universes are discrete we should follow the numerical definition of membership functions; otherwise, we should follow the functional definition. In case the defuzzified version of $\vec{F_{fj}}$ reduces to the crisp vector as stated earlier the membership value at the terminal point of the vector $\vec{F}$ of $\vec{F_{fj}}$ can alternatively be interpreted as the highest possibility of $\vec{F}$ to hold the property of the fuzzy vector $\vec{F_{fj}}$. By the term property we want to mean a particular combination of the elements of different term sets. For instance, with respect to Figure 2.26(a) the property assumed by fuzzy vector $\vec{F_{fj}}$ is medium $\times$ medium. Like this we can have property medium $\times$ small, medium $\times$ big etc (also see Figure 2.4). The crisp vector $\vec{F_{72}}$ of Figure 2.26(a) obtained by the process of defuzzification of the fuzzy vector $\vec{F_{fj}}$ has the highest possibility to hold the above said property (medium $= M_1 \times$ medium $= M_2$). In case the defuzzified version of $\vec{F_{fj}}$ is a fuzzy vector as stated earlier, all the elements of the defuzzified version of $\vec{F_{fj}}$ have equal possibility which is the highest possibility to hold the property of the element of the term set cylindrical extension of which produces such defuzzified version of fuzzy vector. For instance, in Figure 2.26(b), the vector $\vec{F_{126}}$ to $\vec{F_{150}}$ all have equal possibility which is the highest possibility to hold the property of the fuzzy set M_1 which is an element of the term set $\{S_1, M_1, B_1\}$ defined over the universe of F_1. The relation of the generic element represented by line segment $(9 \leq F_2 < 10)$ (defined over the universe of F_2) with all the elements of the fuzzy set M_1 is written by an array $\{.01/\vec{F_{10}}, .2/\vec{F_{35}}, .4/\vec{F_{60}}, .6/\vec{F_{85}}, .8/\vec{F_{110}}, 1/\vec{F_{135}}, .8/\vec{F_{160}}, .6/\vec{F_{185}}, .4/\vec{F_{210}}, .2/\vec{F_{235}}, .01/\vec{F_{260}}\}$. This array itself is a fuzzy set where the vectors $\vec{F_{10}}, \vec{F_{35}}, \vec{F_{60}}, \vec{F_{85}}, \vec{F_{110}}, \vec{F_{135}}, \vec{F_{160}}, \vec{F_{185}}, \vec{F_{210}}, \vec{F_{235}}, \vec{F_{260}}$ are its elements (see Figure 2.26(b)). The possibility of the vector $\vec{F_{10}}$ to hold the property of the vector $\vec{F_{135}}$ is .01.

Thus, we introduce the notion of fuzzy pattern vector / feature vector i.e. $\vec{F_{fj}}$ which is an analogous representation of $\vec{F_{cj}}$ on $\Re^2$ in case we write fuzzy **If Then** rules to represent patterns on $\Re^2$. Tip of the fuzzy pattern vector / feature vector no longer represents a single pattern on $\Re^2$; rather it represents a population of patterns.

Example 2.1: Let us consider the following fuzzy feature vector / pattern vector (see Figure 2.26(a)).

$$\begin{bmatrix} F_1 \text{ is } M_1 \\ F_2 \text{ is } M_2 \end{bmatrix} = \vec{F_{fj}} = \{(\mu_{\vec{F_{fj}}}(\vec{F}), \vec{F})\} = \{\mu_{\vec{F_{fj}}}(\vec{F})/\vec{F}\} = \mu_{\vec{F_{fj}}}(\vec{F_1})/\vec{F_1} + \dots +$$

$\mu_{\vec{F_{fj}}}(\vec{F_{143}})/\vec{F_{143}} = \Sigma_{i=1}^{143}(\mu_{\vec{F_{fj}}}(\vec{F_i})/\vec{F_i} = \{ (.01 / \vec{F_1})+....+(.01 / \vec{F_{11}})+....+(.01 / \vec{F_{13}})+ ....+(.8 / \vec{F_{58}})+...+(1 / \vec{F_{72}})+....+(.01 / \vec{F_{143}}) \}$ where $+$ and Σ are in the set theoretic sense and M_1 is a fuzzy set $\{ .01/(9 \leq F_1 < 10), .2/(10 \leq F_1 < 11), .4/(11 \leq F_1 < 12), .6/(12 \leq F_1 < 13), .8/(13 \leq F_1 < 14), 1/(14 \leq F_1 < 15), .8/(15 \leq F_1 < 16), .6/(16 \leq F_1 < 17), .4/(17 \leq F_1 < 18), .2/(18 \leq F_1 < 19), .01/(19 \leq F_1 < 20) \}$ on the feature axis F_1, M_2 is a fuzzy set $\{ .01/(6 \leq F_2 < 7), .1/(7 \leq F_2 < 8), .2/(8 \leq F_2 < 9), .4/(9 \leq F_2 < 10), .6/(10 \leq F_2 < 11), .8/(11 \leq F_2 < 12), 1/(12 \leq F_2 < 13), .8/(13 \leq F_2 < 14), .6/(14 \leq F_2 < 15), .4/(15 \leq F_2 < 16), .2/(16 \leq F_2 < 17), .1/(17 \leq F_2 < 18), .01/(18 \leq F_2 < 19) \}$ on the feature axis F_2 and each vector $\vec{F_j}$ which is an element of the fuzzy set $\vec{F_{fj}}$ represents one point on the pattern space. Thus instead of a point pattern a population of patterns is represented by $\vec{F_{fj}}$. Pattern p_7 of Figure 2.26(a) is represented by an element $\vec{F_{72}}$ of the fuzzy set represented by $\vec{F_{fj}}$ and pattern p_4 is represented by $\vec{F_{64}}$ of $\vec{F_{fj}}$. Here the fuzzy set $\vec{F_{fj}}$ is defined over the universe $F_1 \times F_2$, i.e. the pattern space. In this case, also note that the defuzzified version of the fuzzy vector $\vec{F_{fj}}$ is a fuzzy singleton represented by vector $\vec{F_{72}}$. ♡

Depending upon the area occupied by the tip of the $\vec{F_{fj}}$ on each class of patterns (see Figure 2.26(a)) we determine the possibility of occurrence of different classes of patterns under that $\vec{F_{fj}}$. Such possibility of occurrence is represented by a fuzzy set which is the consequent part of a MFI and which is defined in the pattern space which is the universe of all pattern classes. Thus in the same pattern space, i.e. in the Cartesian product $F_1 \times F_2$ (if $n = 2$) we define two types of fuzzy sets; one fuzzy set is represented by $\vec{F_{fj}}$ and the other fuzzy set is the consequent part of a MFI. The consequent part of a MFI, which is a fuzzy set, simply indicates the relative position of a fuzzy feature vector / pattern vector with respect to the different classes of patterns in the pattern space. Once the antecedent part and the consequent part of a MFI are represented by two types of fuzzy sets as stated above our next job is to attach a meaningful interpretation to the said representations. One possible interpretation is given by equation (2.14(a)) which is similar to R_4 represented by a fuzzy set. The other interpretation is given by equation (2.16). A particular interpretation simply provides the information how the antecedent part of a MFI is related to its consequent part. Depending upon different types of interpretations, we are having different types of model for fuzzy reasoning [61,91,108].

Incidentally, note that by fuzzy feature vector / pattern vector we basically represent the pattern space (see Figure 2.1) by few quantized zones where neighboring zones overlap each other (see Figure 2.4). The overlapped zones are called fuzzy zones of our description of the pattern space.

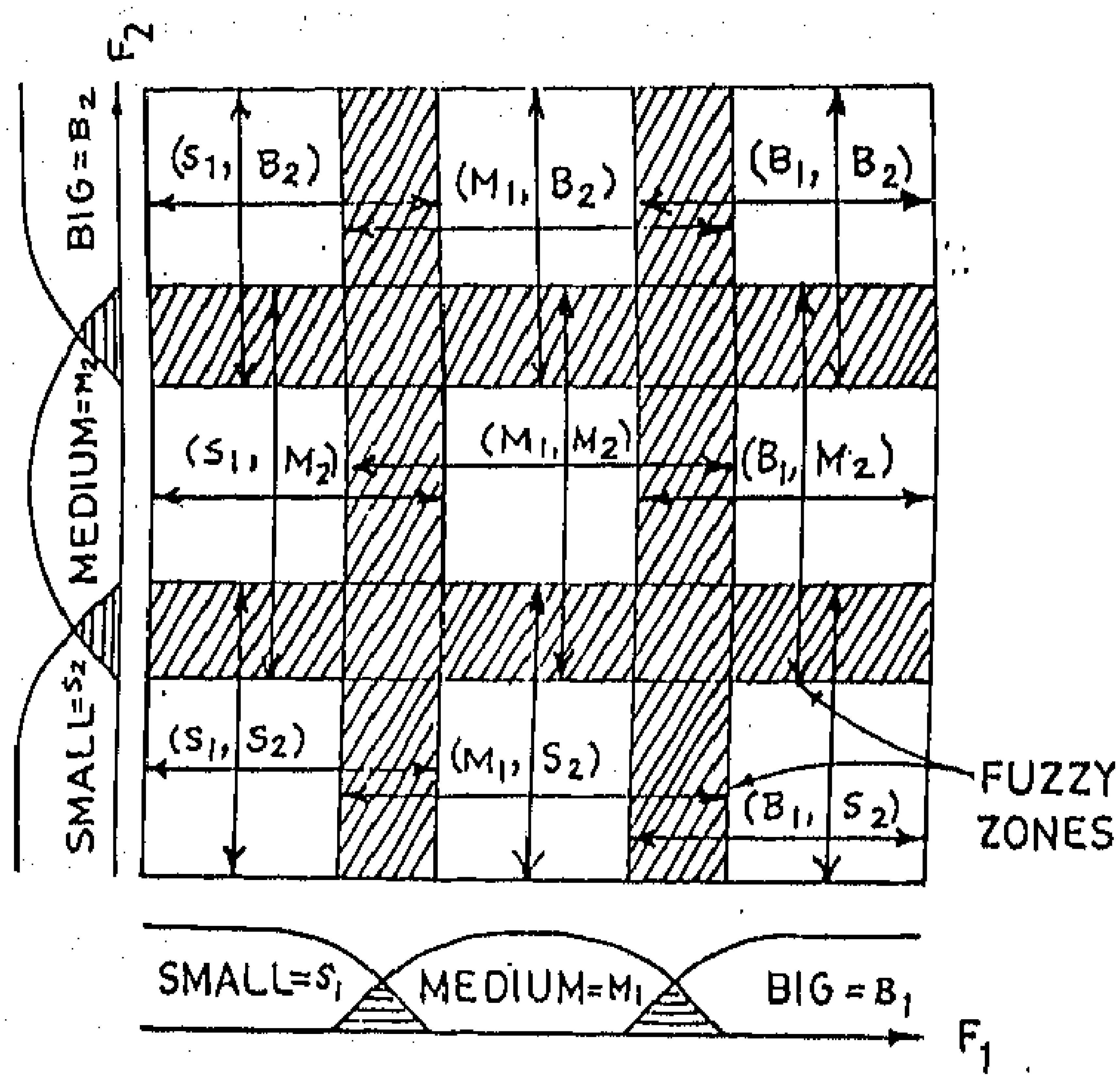


Fig. 2.4: Quantized pattern space

So far we have discussed fuzzy If Then rules where features are not directly measurable by some sensors, e,g, standard of living, behavior, appearance etc. Such features are fuzzified (see Figure 2.2) for the purpose of representing our perception of objects (patterns). But, apart from these, we can have features which can be sensed by standard transducers or which are measurable, e.g. formant frequency of a vowel, curvature and internal angle at a *dominant* point of a 2-D closed curve etc. Under such circumstances, given features are not fuzzy. But we can fuzzify them (measurable features) as we have done in case of unmeasurable features (See Figure 2.2) to represent our *perception* of classification of different objects (patterns). For instance,

if first formant frequency is small and second formant frequency is medium

R_5 : then $\{ m_1/\tilde{c}_1, m_2/\tilde{c}_2, \ldots, m_k/\tilde{c}_k \}$ where $\tilde{c}_i, i = 1,...,k$ and $m_j, j = 1,...,k$ are classes and membership values respectively (see Figure 2.26(a) and Example 2.2).

Let us now consider Figure 2.5 to provide further clarity about the process of *fuzzification* of the feature universe and the pattern space.

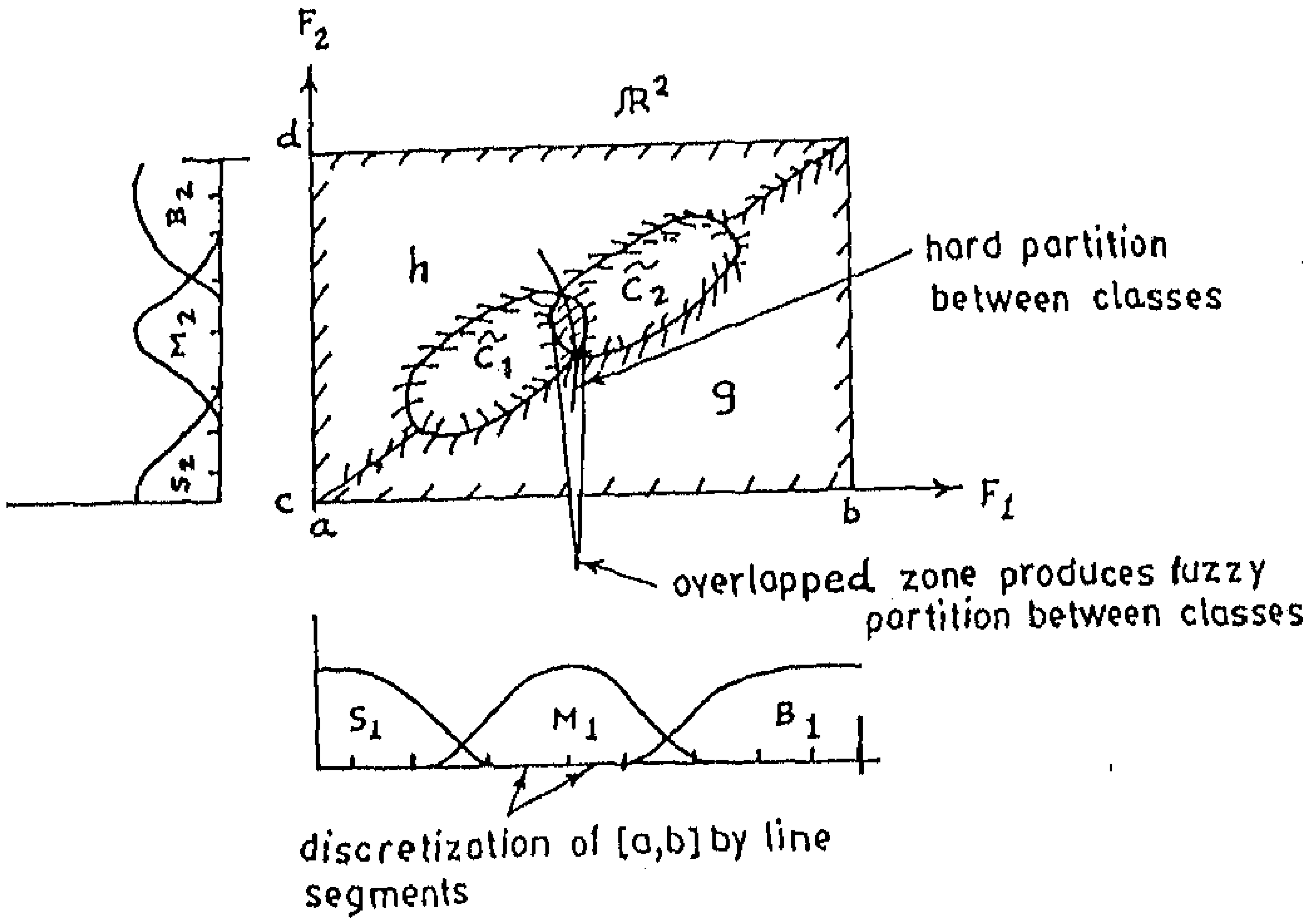


Fig. 2.5: Fuzzy sets represented over the discrete universe of features and pattern space

All the values in [a,b] $\subset$ R - real line is the universe of feature F_1. All measurable (i.e. sensed by transducers) values in [a,b] are the elements of the universe. We discretize [a,b] by line segments to have finite number of generic elements of [a,b]. The primary terms S_1, M_1, B_1 are represented by fuzzy sets defined over the discrete universe of [a,b]. Same things are true for feature F_2.

Let us now define the pattern space as the cover of the classes $\tilde{C_1}, \tilde{C_2}$, g and h and we consider them to be the generic elements of the pattern space which is the universe. From Figure 2.5 it is obvious that $\tilde{C_1}$ and $\tilde{C_2}$ are the classes of patterns. Whereas classes like g and h represent the empty portions of the pattern space as shown in Figure 2.5. Classes like these are considered to have a total cover of the pattern space. In all pattern classification problems we have considered such classes like g and h. In object recognition problem we have considered them implicitly. The possibility of occurrence of different classes of patterns under a particular $\vec{F_{fj}}$ is represented by a fuzzy set which is defined over the pattern space and which is the consequent part of a MFI.

Let,

$$\begin{aligned} \underline{F_1} &= \{S_1, M_1, B_1\} \\ \underline{F_2} &= \{S_2, M_2, B_2\} \\ C &= \{\tilde{C_1}, \tilde{C_2}, g, h\} \end{aligned}$$

I be the closed unit interval [0,1]

I^c be the collection of all mappings of C into I.

According to the usual interpretation of MFI (i.e. equation (2.14(a))), we have to estimate the relation $\Re$ such that,

$$\Re : \underline{F_1} \times \underline{F_2} \rightarrow I^c ;$$

and according to the new interpretation of MFI (i.e. equation (2.16)) we have to estimate two relations $\Re_1$ and $\Re_2$ as follows:

$$\Re_1 : \underline{F_1} \rightarrow I^c$$
$$\Re_2 : \underline{F_2} \rightarrow I^c$$

From the given **If Then** rules once the fuzzy relation / relations (i.e. *either* $\Re$ or $\Re_1$ and $\Re_2$) is / are estimated, we can use it / them further for inferencing for classification of patterns as discussed in the following sections and subsequent chapters.

In Figure 2.6, we depict the passage discussed above and then highlight the advantages we obtain.

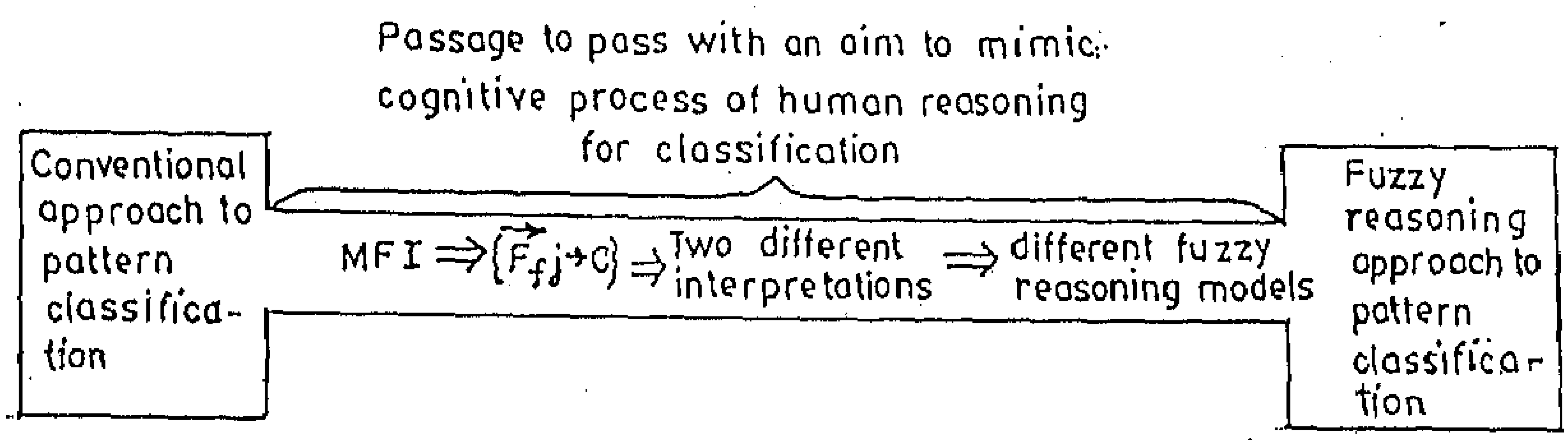


Fig. 2.6: Passage between conventional approach to pattern classification and fuzzy reasoning approach to pattern classification

The advantages which we obtain for fuzzy reasoning approach to pattern classification are as follows:

1. We obtain the local descriptions of the pattern space in terms of few quantized zones (see Figure 2.4). Depending upon the need of the problem we may increase / decrease the granularity of our description of the pattern space by smaller / bigger quantized zones.

2. At the training stage of a classifier (see Chapter 2,3,4,5,6 and 7) we do not need to select the representative data set for learning from the given set of data (patterns). Instead, we use the gross property of few populations of given data (patterns) spreaded over the pattern space by using few fuzzy pattern vectors / feature vectors which describe the overall distribution of patterns in pattern space. For

instance, we use nine fuzzy pattern vectors / feature vectors (see Figure 2.13)to describe the overall distribution of patterns of Figure 2.8.

3. We obtain multiple classification which is very natural in case of overlapped classes of patterns.

4. If we try to classify patterns on $\Re^n$, n > 2 our new interpretation (2.16) for MFI does not suffer from the curse of large feature dimension. For instance, according to conventional interpretation of MFI i.e. equation (2.14(a)), the antecedent clauses of fuzzy **If Then** rules will produce a large dimension of relation when n becomes large say 3,4,5 etc. Whereas, according to the new interpretation i.e. (2.16) n may assume any large value which does not affect the above said problem of dimensionality.

5. Classifier based on fuzzy **If Then** rules learns faster than crisp classifier systems. The reason for this is that fuzzy rules are high level languages [Chapter 3 of [15]] which keep the cardinality of the term set smaller than crisp rules. Thus, we can accurately describe the decision strategy over the pattern space by few number of rules (e.g. we use nine rules to describe the pattern space of Figure 2.8).

2.2 Fuzzy reasoning proposed by Zadeh

The aim of fuzzy set theory is to build up a quantitative framework that represents the vagueness of human knowledge as it is expressed through natural languages. The gap, which separates the conventional mathematical models of physical systems from the imprecise mental representation of them, essentially motivated Zadeh to study how the mental representation could be translated into computable entities so as to overcome some limitations of the conventional models.

In 1970, Zadeh [108] introduced the notion of quantitative fuzzy semantics and illustrated how linguistic terms can be expressed as fuzzy sets on universes of discourse and how a logical combination of linguistic terms can have a numerical counterpart in terms of fuzzy set theoretic operations. The universe of discourse, as presented by Zadeh are basically numerical sets for attributes like "age", "length", "size" etc. Thus, the "quantitative fuzzy semantics" provides a tool for interfacing numbers and symbols. Such a tool respects the continuity of the underlying universe of discourse and the gradeness of concepts when manipulating the symbols.

In 1973, Zadeh [109] generalized the concept of Modus Ponens for modeling the deductive process with fuzzy categories. Noticeably the concept of generalized modus ponens had a significant impact on engineering research because the concept of fuzzy logic controller first developed by Mamdani [8,54,70] was based on this.

Zadeh's fuzzy reasoning methodology was developed outside the school of thought of "Artificial Intelligence". But, there is a close relationship between the two approaches. The artificial intelligence approach ignores the idea of "number crunching" and essentially deals with symbolic manipulation. It is heavily based on logic and automates deduction using syntactic tools. Whereas, Zadeh's method of fuzzy reasoning addresses the interface between numbers and symbols. The stress on symbol manipulation, lead researchers to view the problem of uncertainty propagation in reasoning systems as one of attaching numbers (certainty factors) to symbols rather than one of interpreting symbols in terms of fuzzy sets defined on numerical scales (as done by Zadeh). Hence, we have two different methodologies for fuzzy reasoning

i) the syntactic methods that refer to certainty factors and

ii) the semantic approach that is based on the combination and projection of fuzzy relations (often numerical) universes of discourse.

Here we deal with the latter methodology. Further, we limit ourselves to non-probability aspects of fuzzy reasoning. The presentation of fuzzy quantities, fuzzily-quantified statements, statements involving hidden quantifiers (named 'dispositions') and usually involve a mixture of fuzzy and probabilistic tools which are beyond the scope of this thesis.

2.2.1 Basic approach to Zadeh's Fuzzy Reasoning

We consider the following form of inference [108]:

$$\left.\begin{array}{ll}\text{premise 1:} & \text{If X is A then y is B} \\ \text{premise 2:} & \text{X is } A'\end{array}\right\} \quad (2.1)$$

$$\text{Consequence:} \quad \text{Y is } B'$$

where A, A' are fuzzy sets in U and B, B' are fuzzy sets in V. The consequence B' is deduced from premise 1 and premise 2 by taking the max-min composition $\circ$ of the fuzzy set A' and the fuzzy relation A $\rightarrow$ B obtained from the fuzzy implication "if A then B". That means, we get,

$$\begin{aligned} B' &= A' \circ (A \rightarrow B), \\ \mu_{B'}(v) &= \bigvee_u \{\mu_{A'}(u) \wedge \mu_{A\rightarrow B}(u,v).\} \end{aligned}$$

If the fuzzy set A' is a singleton u_0, that is, $\mu_{A'}(u_0) = 1$ and $\mu_{A'}(u) = 0$ for $u \neq u_0$, the consequence B' is simplified as,

$$\left.\begin{aligned} \mu_{B'}(v) &= \bigvee_u \{\mu_{A'}(u) \wedge \mu_{A\rightarrow B}(u,v)\} \\ &= \bigvee_{u(\neq u_0)} \{0 \wedge \mu_{A\rightarrow B}(u,v)\} \vee \{1 \wedge \mu_{A\rightarrow B}(u_0,v)\} \\ &= \mu_{A\rightarrow B}(u_0,v) \quad . \end{aligned}\right\} \quad (2.2)$$

If the fuzzy implication A$\rightarrow$B is represented by the direct product A$\times$B of fuzzy sets A and B as in the case of Mamdani's method [61], B' is given as,

$$\mu_{B'}(v) = \mu_A(u_0) \wedge \mu_B(v) \text{ at } A \rightarrow B = A \times B.$$

Table 2.1: Some interpretations of fuzzy implications $\mu_{A\to}(u_0, v) = \mu_A(u_0) \to \mu_B(v)$

R_c :	$\mu_A(u_0) \wedge \mu_B(v)$	**Mamdani**
R_p :	$\mu_A(u_0) \bullet \mu_B(v)$	**Larsen**
R_{bp} :	$0 \vee [\mu_A(u_0) + \mu_B(v) - 1]$	**bounded product**
R_{dp} :	$\begin{cases} \mu_A(u_0), & \mu_B(v) = 1 \\ \mu_B(v), & \mu_A(u_0) = 1 \\ 0, & \mu_A(u_0), \mu_B(v) < 1 \end{cases}$	**drastic product**
R_a:	$1 \wedge [1 - \mu_A(u_0) + \mu_B(v)]$	**Zadeh's arithmetic rule**
R_m :	$[\mu_A(u_0) \wedge \mu_B(v)] \vee [1 - \mu_A(u_0)]$	**Zadeh's maximum rule**
R_b :	$[1 - \mu_A(u_0)] \vee \mu_B(v)$	**Boolean implication**
R_s :	$\begin{cases} 1, & \mu_A(u_0) \preceq \mu_B(v) \\ 0, & \mu_A(u_0) > \mu_B(v) \end{cases}$	**Standard sequence**
R_g :	$\begin{cases} 1, & \mu_A(u_0) \preceq \mu_B(v) \\ \mu_B(v), & \mu_A(u_0) > \mu_B(v) \end{cases}$	**Gödelian logic**
R_Δ :	$\begin{cases} 1, & \mu_A(u_0) \preceq \mu_B(v) \\ \mu_B(v)/\mu_A(u_0), & \mu_A(u_0) > \mu_B(v) \end{cases}$	**Gougen logic**
R^* :	$1 - \mu_A(u_0) + \mu_A(u_0) \bullet \mu_B(v)$	**Bandler logic**
$R_\#$:	$[1 - \mu_A(u_0) \vee \mu_B(v)] \wedge [\mu_A(u_0) \vee (1 - \mu_A(u_0))] \wedge [\mu_B(v) \vee (1 - \mu_B(v))]$	**Bandler logic**

In Table 2.1 [62], we list several fuzzy implications A→B which will be used in the fuzzy reasoning approach to pattern classification.

It should be noted that the composition operator (e.g. max-min) is uniquely related with the way in which the individual rules are combined. That means, the max-t composition is linked with the implication operator induced by the t-norm [70]. In this sense, the list of interpretations of Table 2.1 should be infinite. But, our major objective is to establish the effectiveness of the method of fuzzy reasoning in the field of pattern classification. Hence, we just suitably picked up few interpretations of fuzzy implications in Table 2.1.

2.3 Extended Fuzzy reasoning

Now, we consider the following form of inference [61] in which a fuzzy conditional proposition "if ... then ..." contains two fuzzy propositions "X is A" and "Y is B" combined using the connective "and".

$$\left.\begin{array}{ll}\text{Premise 1:} & \text{if X is A and Y is B then Z is C} \\ \text{premise 2:} & \text{X is } A' \text{ and Y is } B'\end{array}\right\} \tag{2.3}$$

Consequence: Z is C'

where A, A' are fuzzy sets in U; B, B' are fuzzy sets in V and C, C' are fuzzy sets in W.

The consequence C' can be deduced from Premise 1 and Premise 2 by taking the max-min composition $\circ$ of a fuzzy set (A' and B') in U $\times$ V and a fuzzy relation (A and B) $\rightarrow$ C in U $\times$ V $\times$ W. That means, we get,

$$\left.\begin{array}{rcl} C' & = & (A' \text{ and } B') \circ [(A \text{ and } B) \rightarrow C] \\ \mu_{C'}(w) & = & \bigvee\limits_{u,v} \{[\mu_{A'}(u) \wedge \mu_{B'}(v)] \wedge [(\mu_A(u) \wedge \mu_B(v)) \rightarrow \mu_C(w)]\} \end{array}\right\} \tag{2.4}$$

In the case of Mamdani's method R_c in Table 2.1, the fuzzy implication [(A and B) $\rightarrow$ C] is translated into $\mu_A(u) \wedge \mu_B(v) \wedge \mu_C(w)$ by virtue of a $\rightarrow$ b = a $\wedge$ b.Thus, the consequence C' is given as,

$$\left.\mu_{C'}(w) = \bigvee_{u,v} \{[\mu_{A'}(u) \wedge \mu_{B'}(v)] \wedge [\mu_A(u) \wedge \mu_B(v) \wedge \mu_C(w)]\}\right\} \tag{2.5}$$

Let $R_c(A, B; C)$ = (A and B) $\rightarrow$ C, $R_c(A; C) = A \rightarrow$ C and $R_c(B; C) = B \rightarrow$ C be fuzzy implications by Mamdani's method R_c. Then, the consequence C' of equation (2.5) is reduced to,

$$\begin{aligned}\mu_{C'}(w) &= \bigvee_u \{\mu_{A'}(u) \wedge \mu_A(u) \wedge \mu_C(w) \wedge \bigvee_v [\mu_{B'}(v) \wedge \mu_B(v) \wedge \mu_C(w)]\} \\ &= \bigvee_u \{\mu_{A'}(u) \wedge \mu_A(u) \wedge \mu_C(w) \wedge \mu_{B' \circ R_c(B;C)}(w)\} \\ &= \mu_{A' \circ R_c(A;C)}(w) \wedge \mu_{B' \circ R_c(B;C)}(w)\end{aligned}$$

Therefore, the consequence $C' = (A' and B') \circ R_c(A, B; C)$ can be obtained as the intersection of $A' \circ R_c(A; C)$ and $B' \circ R_c(B; C)$ for Mamdani's implication R_c. That means, we get,

$$\begin{aligned}C' &= (A' \text{ and } B') \circ R_c(A, B; C) \\ &= [A' \circ R_c(A; C)] \cap [B' \circ R_c(B; C)]\end{aligned}$$

Similarly, we can have,

$$(A' \text{ and } B') \circ [(A \text{ and } B) \to C] = [A' \circ (A \to C)] \cap [B' \circ (B \to C)]\} \tag{2.6}$$

for the fuzzy implications R_p, R_{pp} and R_{dp} in Table 2.1.

Note that R_a,R_b,R^*,R_s,R_g and R_Δ in Table 2.1, for which the equality $(a \wedge b) \to c = (a \to c) \wedge (b \to c)$ holds, satisfy the following [61],

$$(A' \text{ and } B') \circ [(A \text{ and } B) \to C] = [A' \circ (A \to C)] \cup [B' \circ (B \to C)] \tag{2.7}$$

If the fuzzy sets A' and B' are singletons in (2.2), i.e., $A' = u_0$ and $B' = v_0$, the consequence C' of (2.3) is represented as,

$$\left.\begin{aligned}\mu_{C'}(w) &= \bigvee_{\substack{u(\neq u_0) \\ or(v \neq v_0)}} \{0 \wedge [(\mu_A(u) \wedge \mu_B(v) \to \mu_C(w))]\} \\ &\qquad \vee \{1 \wedge [(\mu_A(u_0) \wedge \mu_B(v_0)) \to \mu_C(w)]\} \\ &= [\mu_A(u_0) \wedge \mu_B(v_0)] \to \mu_C(w)\end{aligned}\right\} \tag{2.8}$$

For example, in the case of R_c and R_a, we have consequences C' at $A' = u_0$ and $B' = v_0$ as follows,

$$R_c : [\mu_A(u_0) \wedge \mu_B(v_0)] \wedge \mu_C(w), \tag{2.9}$$

$$R_a : 1 \wedge [1 - (\mu_A(u_0) \wedge \mu_B(v_0)) + \mu_C(w)]. \tag{2.10}$$

Similar results can be obtained from other fuzzy implications in Table 2.1.

In the above discussion, the operation $\wedge$ (=min) is used as the meaning of "and". It is possible to introduce other operations, say, algebraic product $\bullet$ and more generally t-norms as "and".

2.4 Applicable form of fuzzy reasoning

As a generalized form of fuzzy reasoning we shall consider fuzzy reasoning with several fuzzy conditional propositions combined with "else".

$$\left.\begin{array}{llll} \text{Premise 1:} & \text{If X is } A_1 \text{ and} & \text{Y is } B_1 & \text{then Z is } C_1 \text{ else} \\ \text{Premise 2:} & \text{If X is } A_2 \text{ and} & \text{Y is } B_2 & \text{then Z is } C_2 \text{ else} \\ \vdots & & & \\ \text{Premise n:} & \text{If X is } A_n \text{ and} & \text{Y is } B_n & \text{then Z is } C_n \\ \text{Premise n+1:} & \text{If X is } A' \text{ and} & \text{Y is } B' & \end{array}\right\} \tag{2.11}$$

Consequence: Z is C'

If we interpret "else" as union (U) which is valid for the fuzzy implications R_c, R_p, R_{bp} and R_{dp} in Table 2.1, we can deduce the consequences C' (refer equation(2.6)) as,

$$
\left.\begin{aligned}
C' &= (A' \text{ and } B') \circ [((A_1 \text{ and } B_1) \to C_1) \cup \cdots \cup ((A_n \text{ and } B_n) \to C_n)] \\
&= [(A' \circ A_1 \to C_1) \cap (B' \circ B_1 \to C_1)] \cup \cdots \cup [(A' \circ A_n \to C_n) \cap (B' \circ B_n \to C_n)] \\
&= C_1' \cup C_2' \cup \cdots \cup C_n'
\end{aligned}\right\} \tag{2.12}
$$

whereas, for the fuzzy implications $R_a, R_m, R_b, R^*, R_\#$ and $R_\triangle$ in Table 2.1, "else" in (2.12) is interpreted as intersection ($\cap$).Thus, the consequences C' for these fuzzy implications are defined as,

$$
\begin{aligned}
C' &= (A' \text{ and } B') \circ [((A_1 \text{ and } B_1) \to C_1) \cap \cdots \cap ((A_n \text{ and } B_n) \to C_n)] \\
&\subseteq [(A' \circ A_1 \to C_1) \cup (B' \circ B_1 \to C_1)] \cap \cdots \cap [(A' \circ A_n \to C_n) \cup (B' \circ B_n \to C_n)]
\end{aligned}
$$

It is noted that the consequence C' is not equal to but contained in the intersection of fuzzy inference results $[(A' \circ A_i \to C_i) \cup (B' \circ B_i \to C_i)]$ $\forall i$. However, for simplicity of calculation C' will be represented as,

$$
C' = C_1' \cap C_2' \cap \cdots \cap C_n' \tag{2.13}
$$

Further, note that each one dimensional fuzzy implication of equation (2.11) is basically the conventional interpretation (i.e. equation (2.14(a)) of this chapter) of multidimensional fuzzy implication (MFI).

We have already mentioned in Chapter 1, that fuzzy reasoning approach to pattern classification is similar to the decision theoretic approach to pattern classification. In Fig. 2.7, we schematically represent the fuzzy reasoning approach to pattern classification [80].

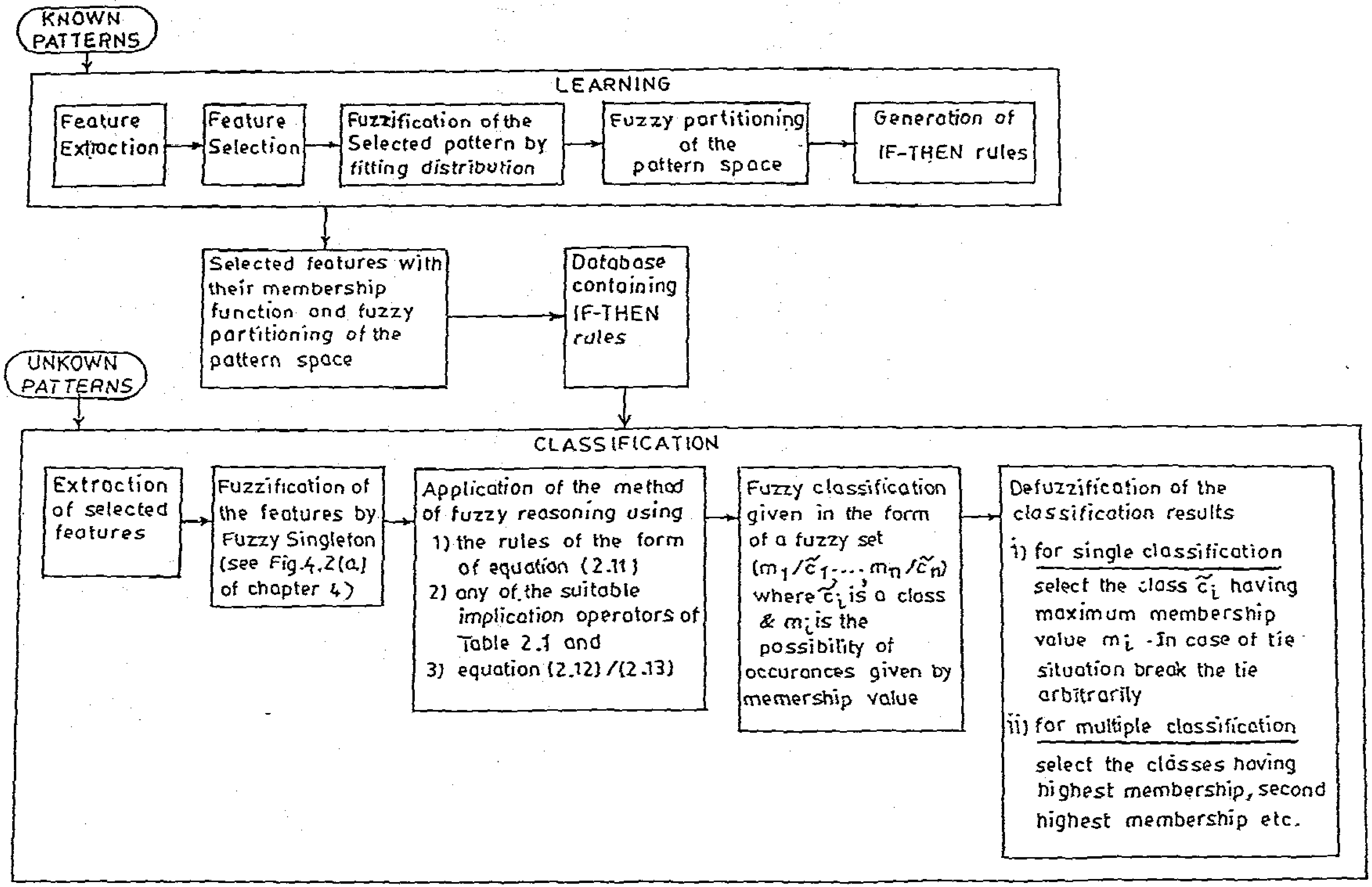


Fig.2.7: Schematic representation of fuzzy reasoning approach to pattern classification.

2.5 Formulation of the problem

First of all, at the learning stage, we partition the individual feature axis by the elements of the term set (see Figure 2.2 of section 2.1) each of which are defined over the universe of individual feature axis. We represent individual element of the term set by nonstandard membership functions (i.e. neither triangular nor trapezoidal nor bell shaped function) as shown in Figure 2.13 to 2.17. Also note that, we have not considered the numerical definition of membership function as discussed in Section 2.1. After defining the term sets we represent the fuzzy set of the consequent part of each one dimensional fuzzy implication using the interpretation of Equation (2.14(a)) of Section 2.9. That means, we use the *notion of fuzzy* feature vector / pattern vector (see Figure 2.26(a) and the discussion of paragraph 4 of Section 2.9) and consider the well defined cover of the pattern space as stated in Section 2.1. Note that, we use the most likely estimate of the membership value of each class of patterns to represent the fuzzy set of the consequent part of each one dimensional fuzzy implication (see Example 2.2). Thus, at the learning stage, we generate fuzzy **If Then** rules to classify the given patterns. After the initial generation of fuzzy **If Then** rules, we test the validity of the rules by classifying some known patterns. If we get satisfactory results, we proceed further; otherwise, we tune the rules by changing the shape of the distribution or by modifying the most likely estimate of the class membership.

If we write a fuzzy **If Then** rule as,
"if F_1 is small and F_2 is small then 0.8/a + 0.2/b + 0.1/c + 0.01/d", it means that the possibility of occurrence of class "a" is the highest, then class "b" and so on.

At the classification stage, the selected features are fuzzified using the concept of fuzzy singleton (see Figure 4.2(a) of Chapter 4) as discussed in section 2.2.1. To perform fuzzy classification the general rule is represented either by equation (2.12) or by equation (2.13). The classification results produce the possibility of occurrence of each pattern on the pattern space. At the time of taking nonfuzzy decision (i.e. defuzzification) out of this fuzzy classification we can go by selecting class having highest possibility value (i.e. membership value). In case of tie situation, which normally occurs for the patterns lying in the overlapped zones of different classes (see Figure 2.5 of Chapter 2), we have to state the equal possibility of a pattern to belong to both the classes. And such a conclusion is quite natural which normally does not exist in conventional classification approach. In some cases, patterns in the overlapped zones of different classes are classified with "almost equal" possibility of occurrence at more than one class. If such situation is treated as tie situation mentioned above, we have to select an appropriate threshold which entirely depends on the need of the problem. In case we select a single class having highest membership value we implicitly assume hard partitioning between classes. In case of multiple classification we consider the fuzzy partitioning between classes having overlapped zones (see Figure 2.5 of Chapter 2).

We have tested our proposed scheme on a set of synthetic data and some real life vowel classification problems which are discussed in the following sections [80].

2.6 Numerical Examples

To test the effectiveness of the proposed scheme as shown in Figure 2.7, simulations are run for Figures 2.8 to 2.12. The rules and the distribution patterns (of membership functions) are shown in Figure 2.13 to 2.17. The recognition scores are found to be quite satisfactory in all the cases. In Table 2.2 and Table 2.3, which are self explanatory in nature, we have produced a comparative study on the performance of the proposed scheme under different laws of implications stated in Table 2.1. At the time of calculating the recognition scores, we have considered exhaustive data for each Figure and we ignore multiple classification of patterns. If the recognition score is calculated considering the multiple classification of patterns then the recognition score will be further improved.

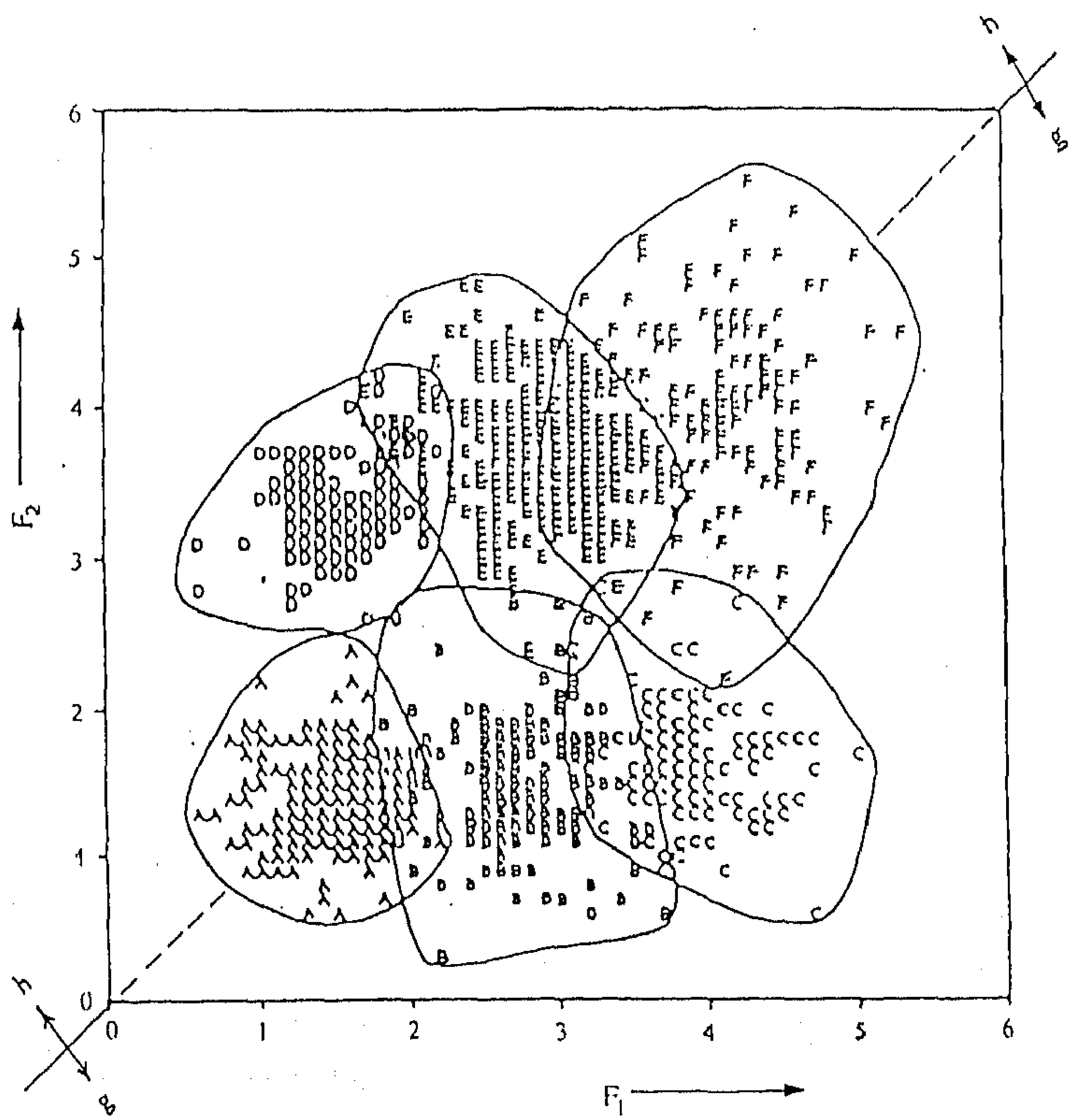


Fig. 2.8: The first synthetic data

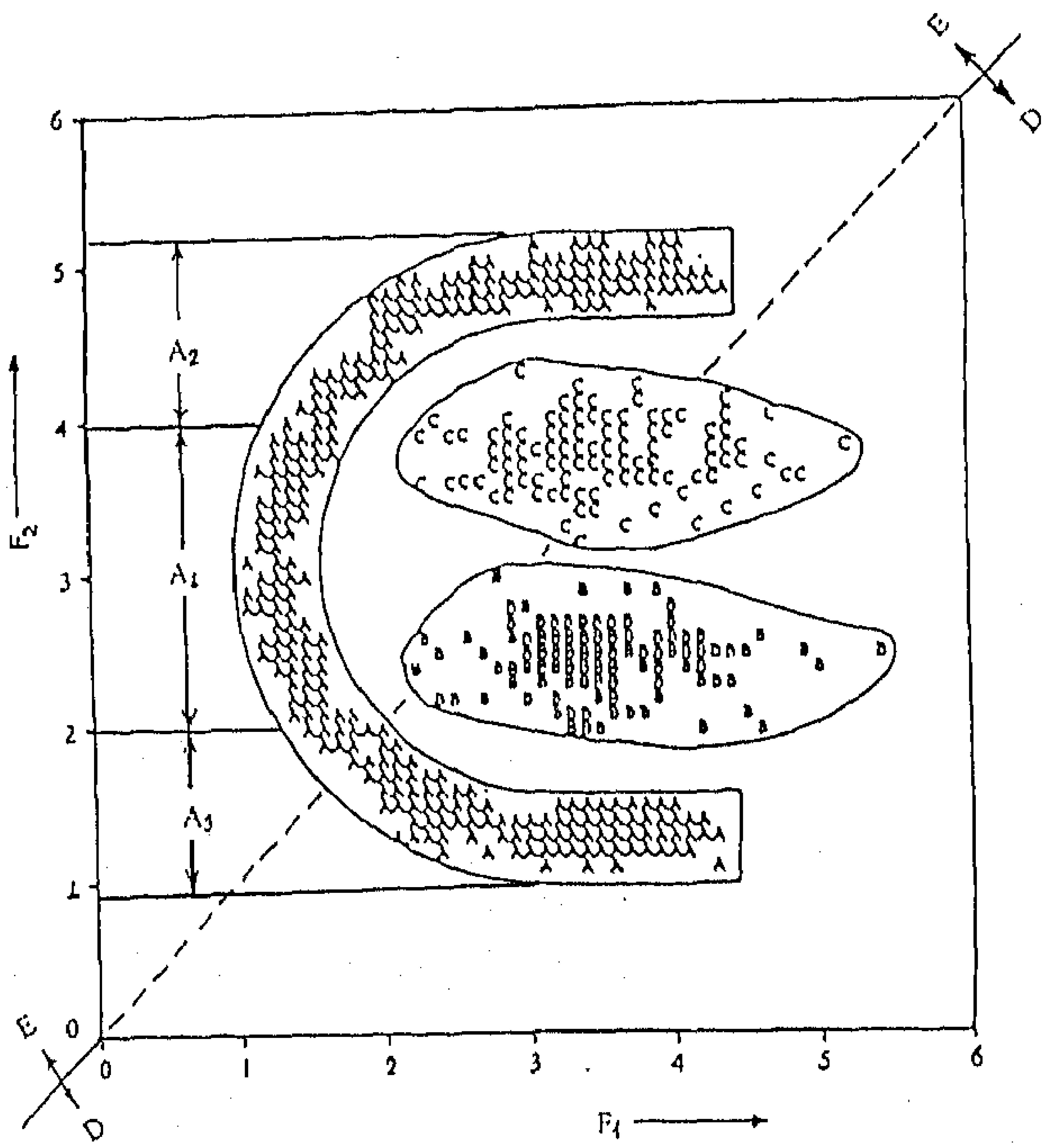


Fig. 2.9: The second synthetic data

Keys: To facilitate the generation of rules the elements of the cluster A are further subdivided into the segments A_1, A_2 and A_3 as shown in the Figure above. These subdivisions of the clusters are used from Chapter 4 onwards where we use the new interpretation of MFI (i.e. Equation (2.16)). But for Chapter 2 and 3, which are based on the conventional interpretation of MFI (i.e. Equation (2.14(a))) we do not consider such subdivisions of clusters for writing the rules at the learning stage of the classifier.

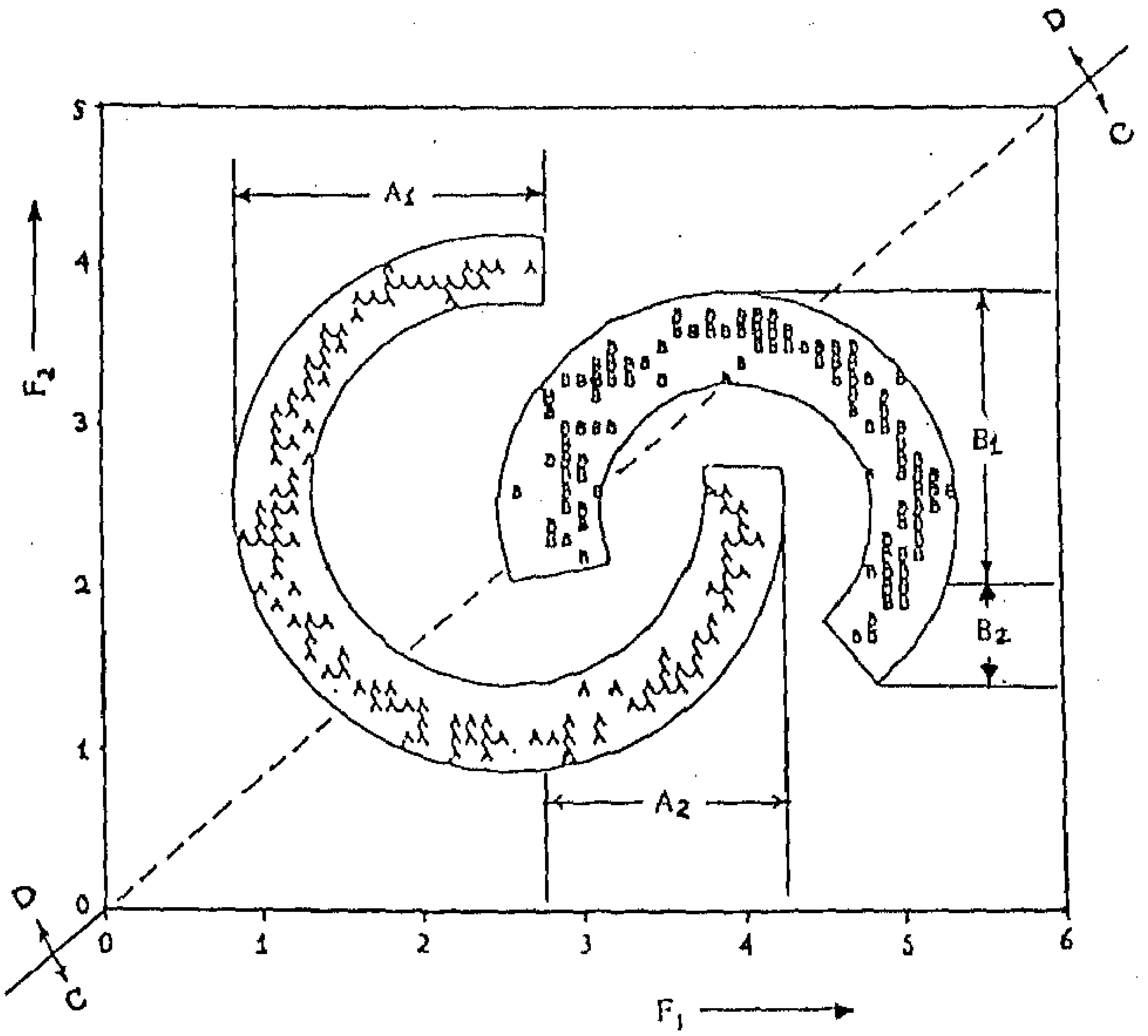


Fig. 2.10: The third synthetic data

Keys: To facilitate the generation of rules the elements of the cluster A are further subdivided into the segments A_1 and A_2 and the elements of the cluster B are further subdivided into the segments B_1 and B_2 as shown in the Figure above. These subdivisions of the clusters are used from Chapter 4 onwards where we use the new interpretation of MFI (i.e. Equation (2.16)). But for Chapter 2 and 3, which are based on the conventional interpretation of MFI (i.e. Equation (2.14(a))) we do not consider such subdivisions of clusters for writing the rules at the learning stage of the classifier.

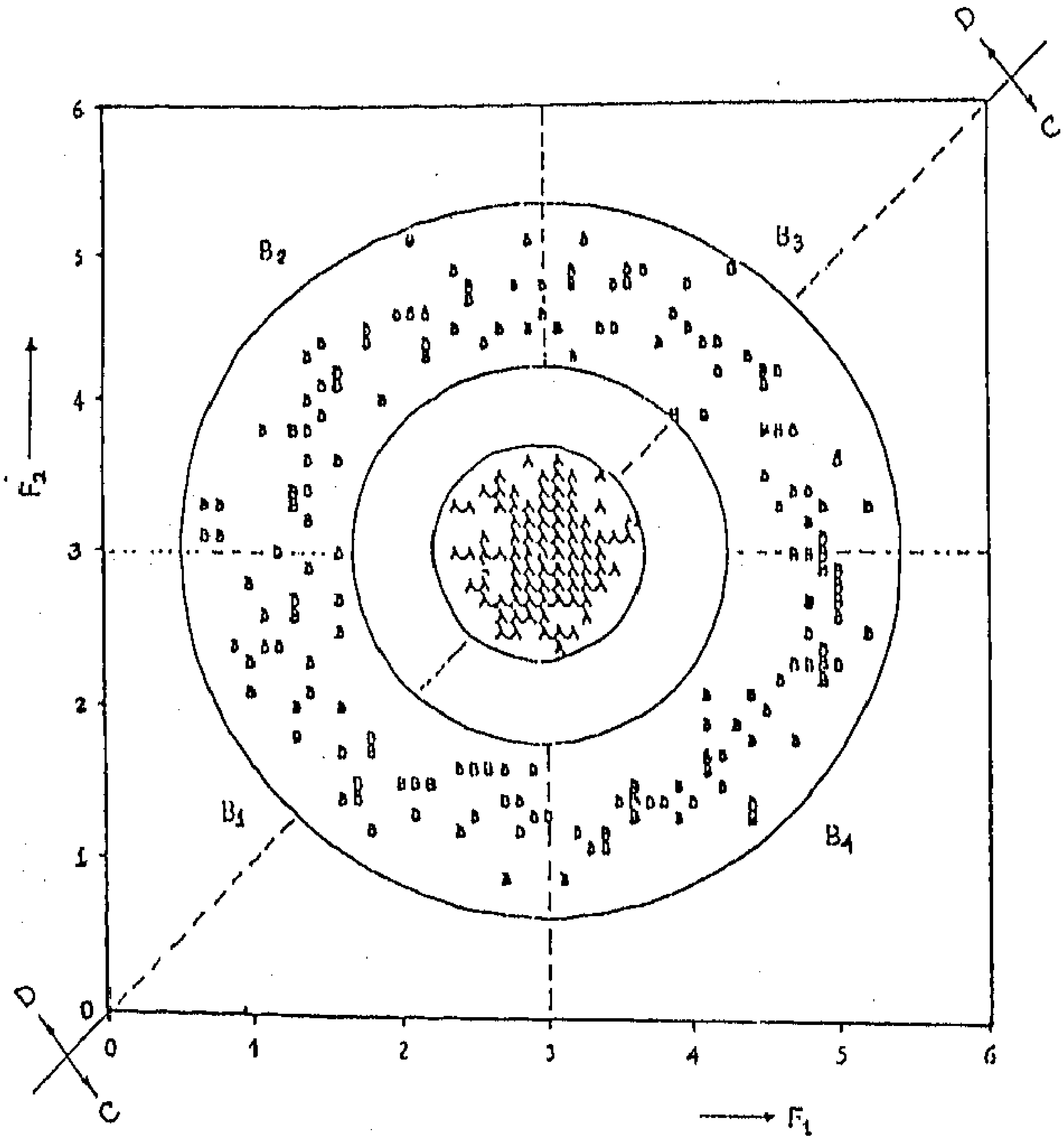


Fig. 2.11: The fourth synthetic data

Keys: To facilitate the generation of rules the elements of the cluster B are further subdivided into the segments B_1, B_2, B_3 and B_4 as shown in the Figure above. These subdivisions of the clusters are used from Chapter 4 onwards where we use the new interpretation of MFI (i.e. Equation (2.16)). But for Chapter 2 and 3, which are based on the conventional interpretation of MFI (i.e. Equation (2.14(a))) we do not consider such subdivisions of clusters for writing the rules at the learning stage of the classifier.

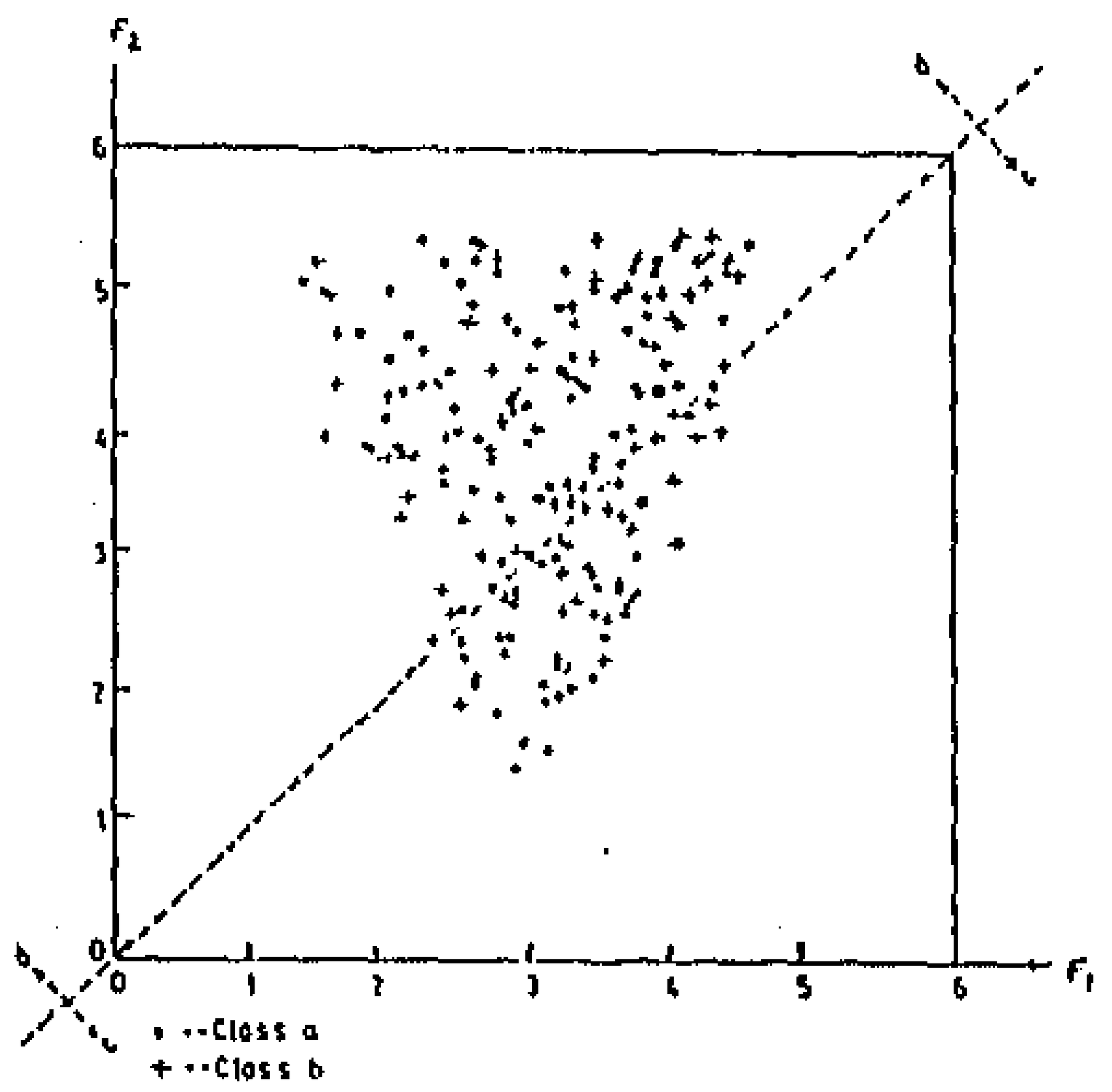


Fig. 2.12: The fifth synthetic data

Table 2.2:Recognition scores (in %) where "and" of the antecedent part of the rule is interpreted as "min"

type of operator	fig2.8						fig2.9			overall score	
	a	b	c	d	e	f	a	b	c	fig2.8	fig2.9
R_c	89.6	89.7	35.4	78.6	99.3	58.2	99.6	100	92	78.5	98.1
R_p	93.9	86.3	36.7	87.6	98.6	61.7	99.6	100	92	80.5	98.1
R_{bp}	93.9	81.2	46.8	87.6	98.6	71.3	100	94.8	79	82.4	94.6
R_{dp}	89.6	89.7	35.4	78.6	99.3	58.2	99.6	100	92	78.5	98.1
R_a	89.6	88.8	35.4	80.8	98.6	62.6	99.6	100	92	79.3	98.1
R_m	95.6	65.8	22.7	49.4	78.8	33.9	100	82.4	73	61.2	90.9
R_b	89.6	78.6	35.4	78.6	99.3	58.2	99.6	100	92	76.6	98.1
R_s	100	27.4	34.1	66.3	35.1	52.2	100	93.8	79	52	94.4
R_g	99.1	83.7	35.4	92.1	94.7	75.6	100	100	92	82.9	98.3
R_Δ	89.6	88.8	35.4	80.9	97.4	70.4	100	100	92	80.4	98.3
R^*	89.6	89.7	35.4	79.7	98.6	60	99.6	100	92	78.8	98.1
$R_\#$	95.6	65.8	22.8	49.4	78.8	33.9	100	82.5	73	61.2	90.9

(Contd.) Table 2.2:Recognition scores (in %) where "and" of the antecedent part of the rule is interpreted as "min"

type of operator	fig2.10		fig2.11		fig2.12		overall score		
	a	b	a	b	a	b	fig2.10	fig2.11	fig2.12
R_c	79.8	100	100	98.1	53.1	69.1	89.3	98.8	60.6
R_p	79.8	100	100	98.1	47.8	39.2	89.3	98.8	43.8
R_{bp}	82.4	100	100	98.1	55.3	64.3	90.7	98.8	59.5
R_{dp}	79.8	100	100	98.1	50	72.6	89.3	98.8	60.6
R_a	82.4	100	100	98.1	55.3	64.2	90.7	98.8	59.5
R_m	85.9	86.1	100	71.1	57.4	54.8	86.1	81.2	56.1
R_b	79.8	100	100	98.1	53.2	64.3	89.3	98.8	60.6
R_s	82.4	71.3	100	91.4	89.4	14.3	77.2	94.4	53.9
R_g	80.7	100	100	98.1	55.3	64.3	89.8	98.8	59.6
R_Δ	80.7	100	100	98.1	55.3	64.3	89.8	98.8	59.6
R^*	79.8	100	100	98.1	52.1	67.8	89.3	98.8	59.5
$R_\#$	86	86.1	100	71.2	36.2	54.8	86.1	81.2	45

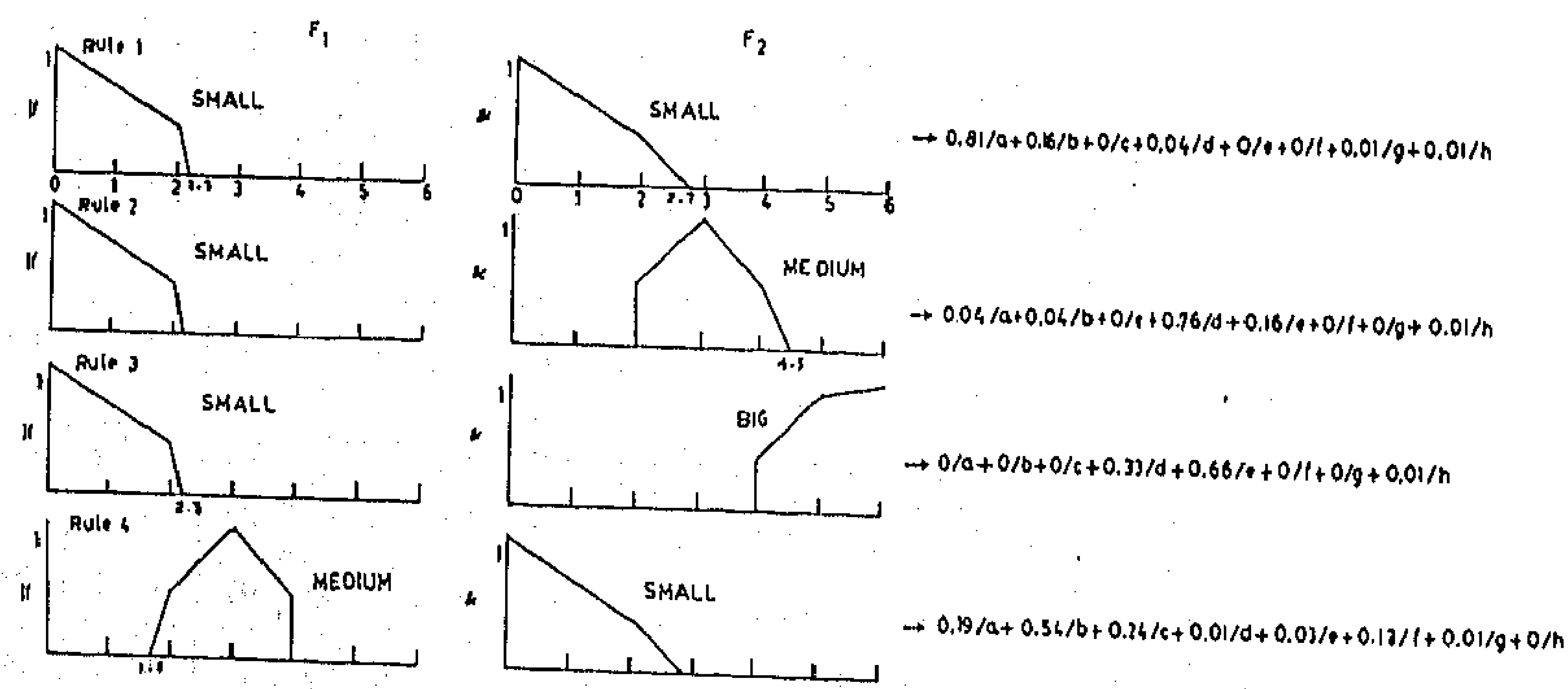


Fig. 2.13: Fuzzy If - Then rules for Fig. 2.8

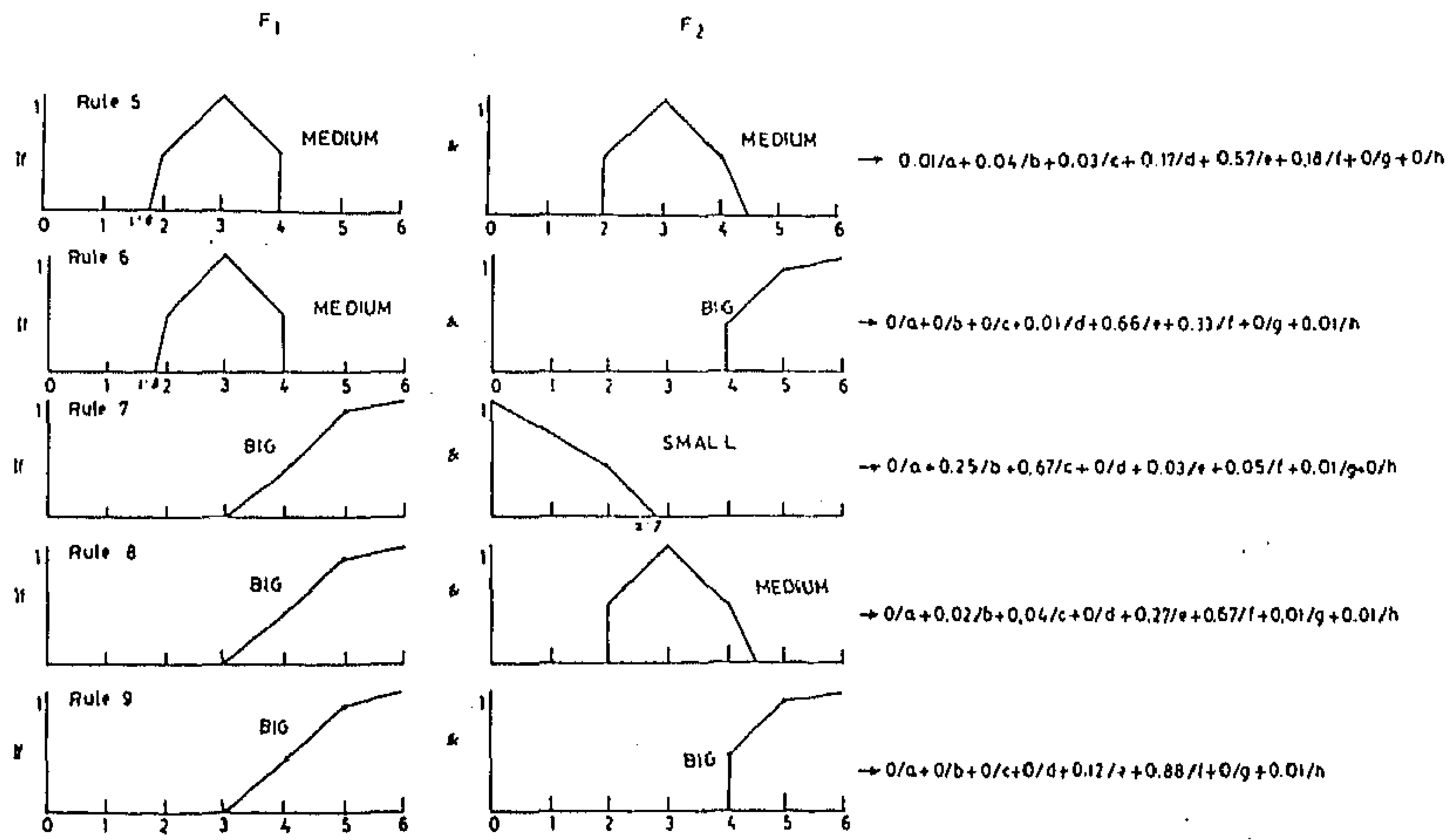


Fig. 2.13: Fuzzy If - Then rules for Fig. 2.8 (Contd.)

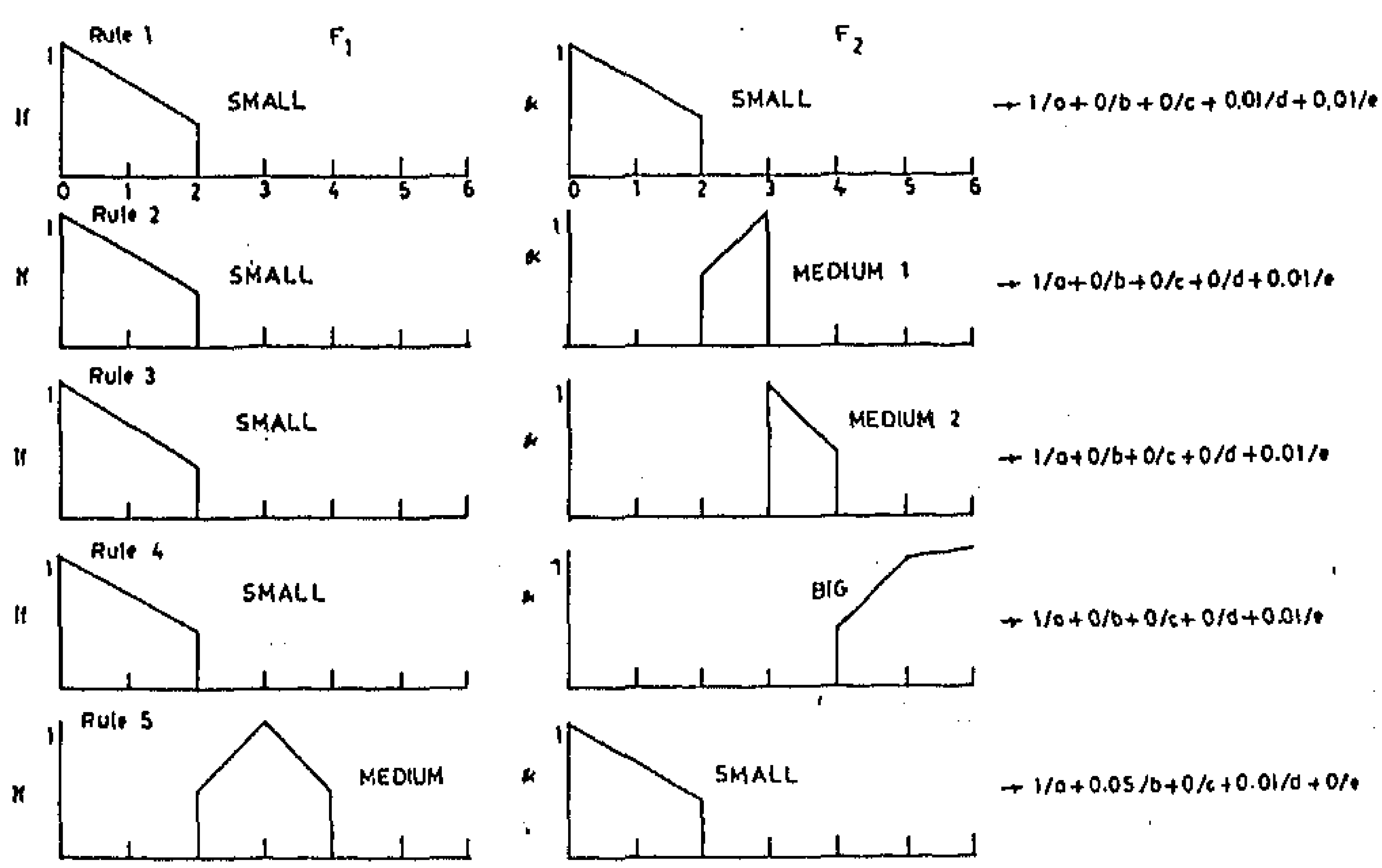


Fig. 2.14: Fuzzy If - Then rules for Fig. 2.9

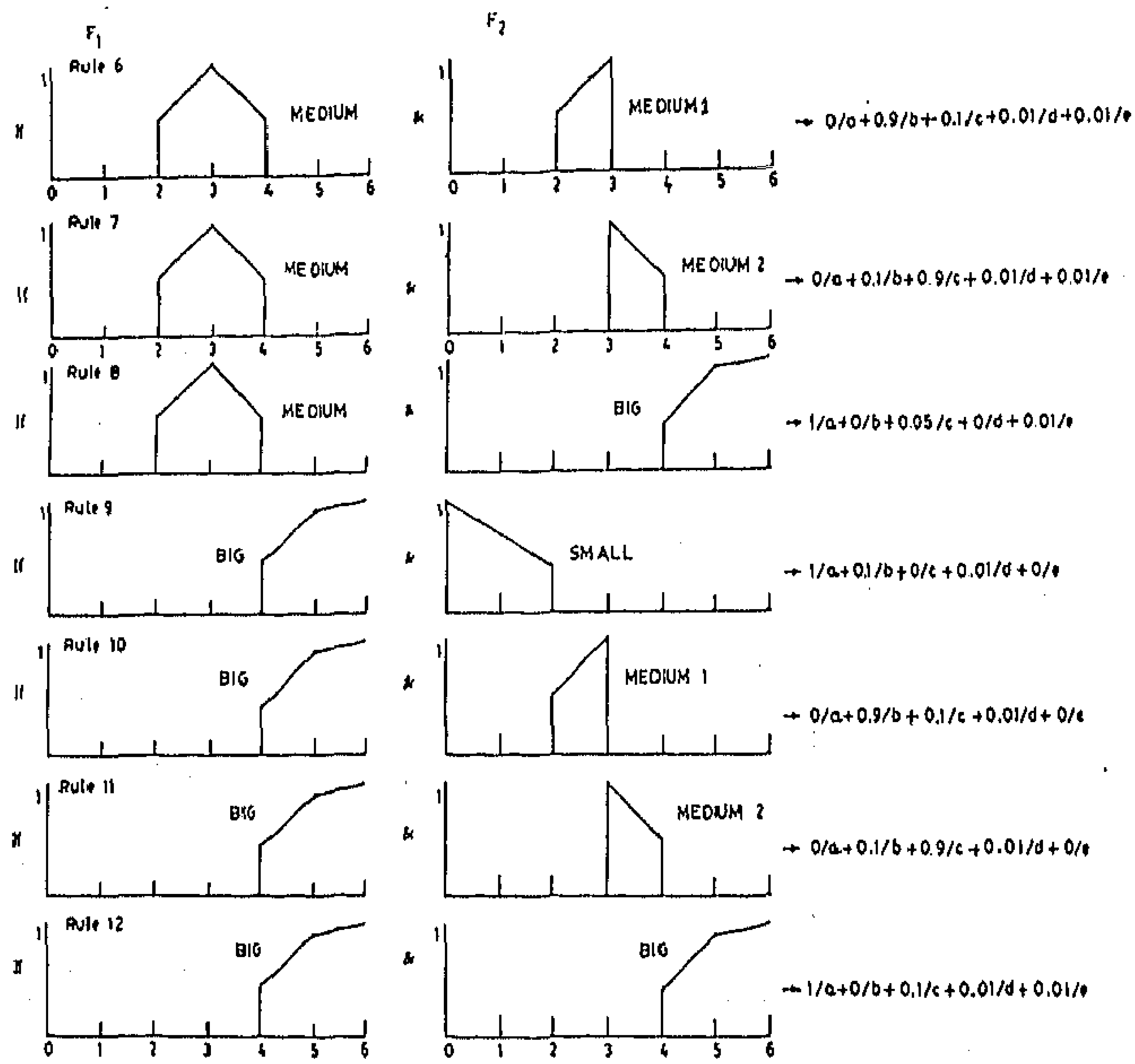


Fig. 2.14: Fuzzy If - Then rules for Fig. 2.9 (Contd.)

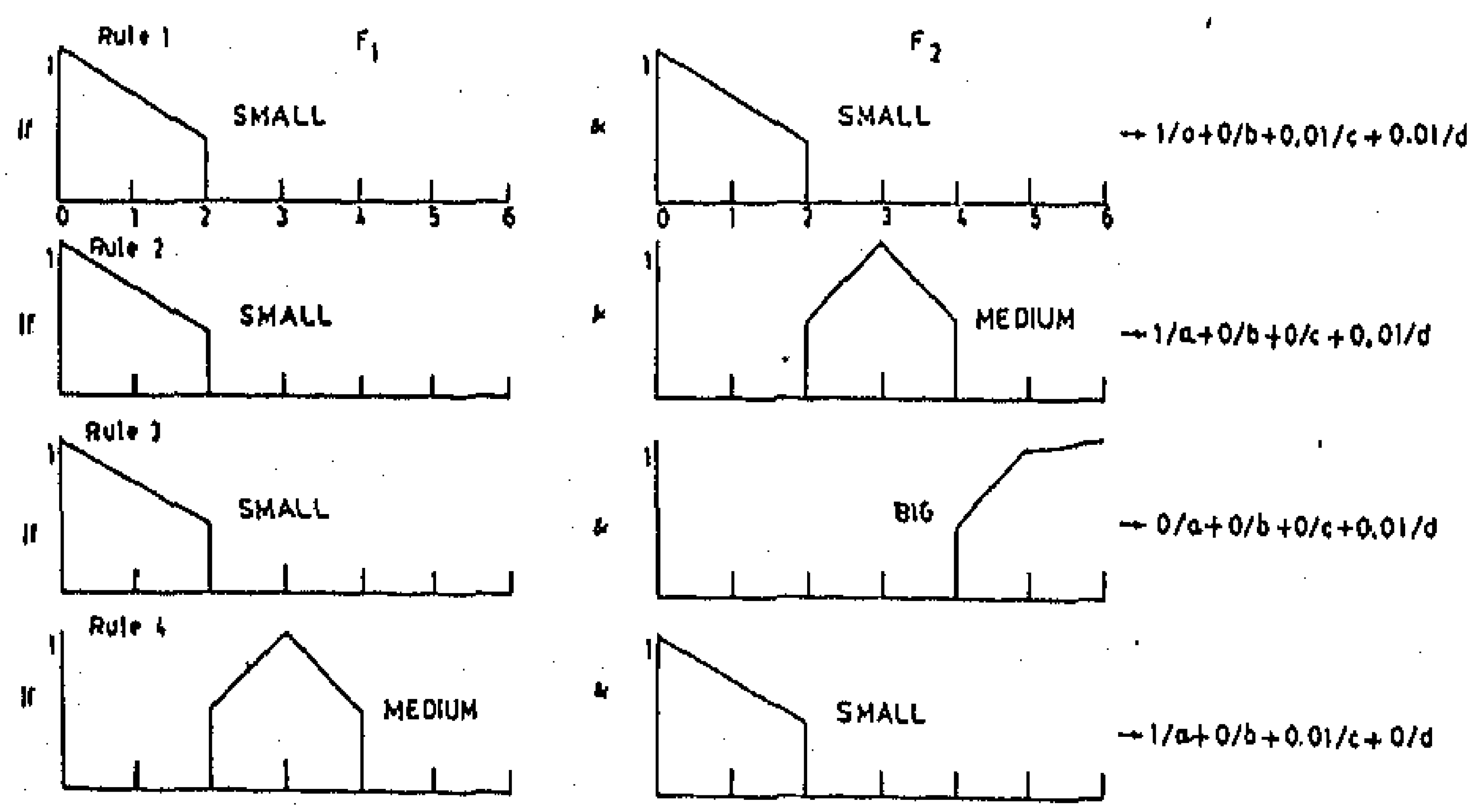


Fig. 2.15: Fuzzy If - Then rules for Fig. 2.10

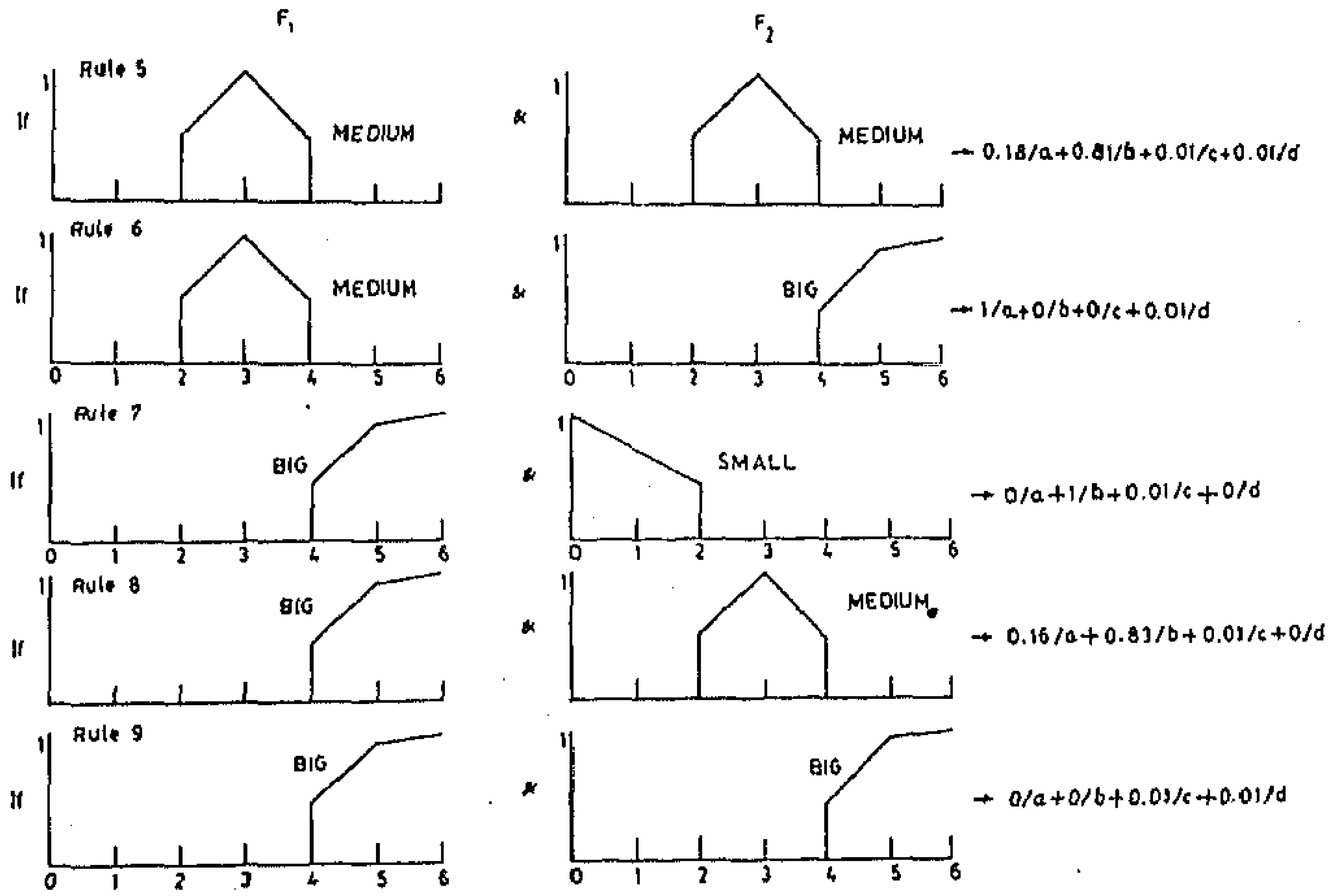


Fig. 2.15: Fuzzy If - Then rules for Fig. 2.10 (Contd.)

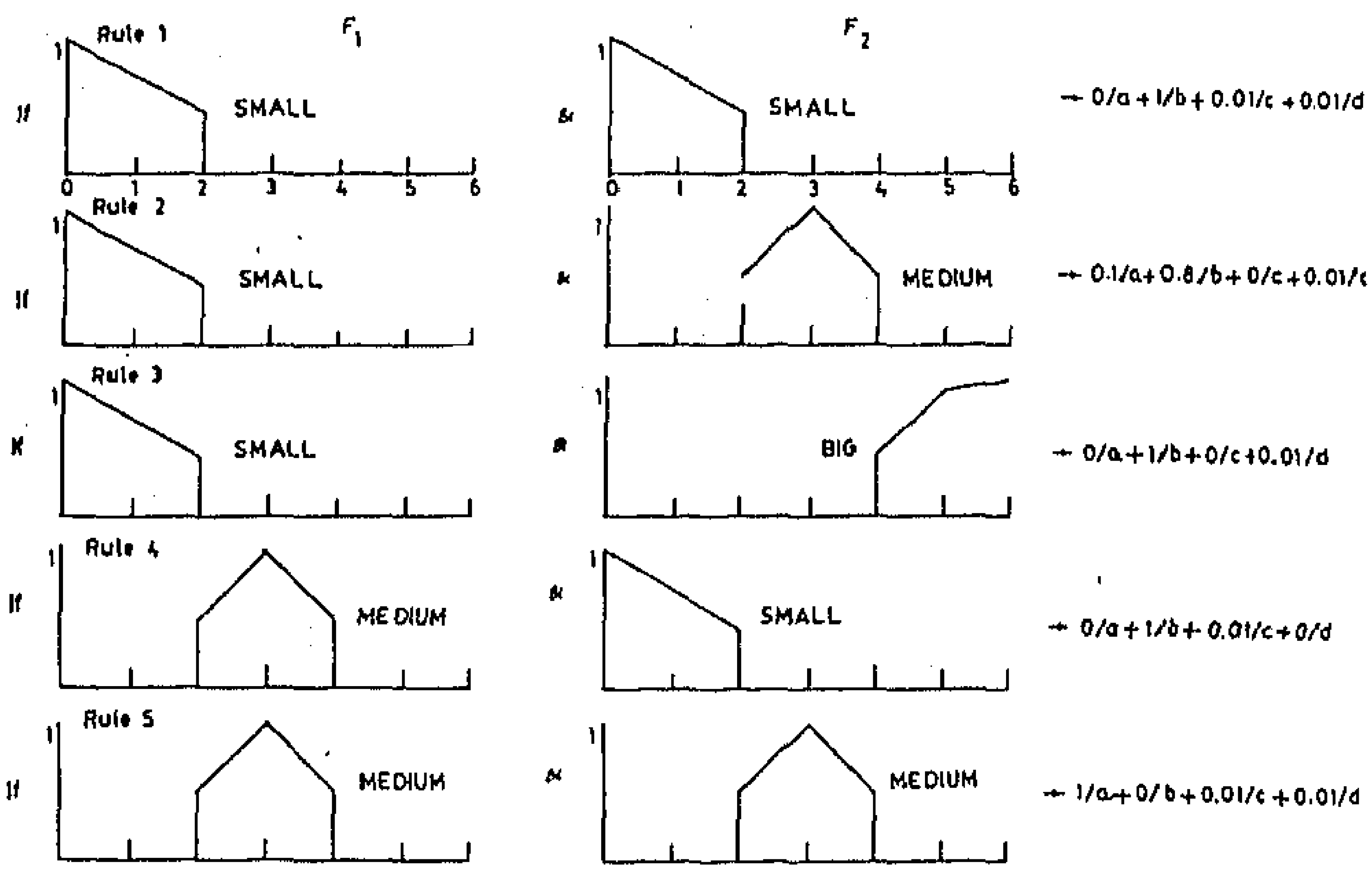


Fig. 2.16: Fuzzy If - Then rules for Fig. 2.11

Table 2.3:Recognition scores (in %) where "and" of the antecedent part of the rule is interpreted as "algebraic product"

type of operator	fig2.8						fig2.9			overall score	
	a	b	c	d	e	f	a	b	c	fig2.8	fig2.9
R_c	89.7	89.7	32.9	80.9	98.6	55.6	99.7	100	92	77.9	98.1
R_p	93.9	86.3	35.4	87.6	98.6	60	99.7	100	92	80	98.1
R_{bp}	95.7	54.7	21.5	84.3	91.4	66.1	100	94.8	79	72.1	68.6
R_{dp}	89.7	89.8	35.4	78.7	99.3	58.3	99.7	100	92	78.6	98.1
R_a	89.7	89.8	32.9	80.9	98.7	64.3	99.7	100	92	79.5	98.1
R_m	100	39.3	3.7	21.3	61.6	15.6	100	52.6	38	44.2	77.6
R_b	95.7	53.8	20.3	78.7	90.8	51.3	99.7	100	92	68.4	98.1
R_s	98.3	54.7	43.1	74.2	57.6	56.5	100	93.8	79	63.3	94.4
R_g	98.3	86.3	32.9	91.1	97.4	72.2	100	100	92	82.7	98.4
R_Δ	89.7	89.8	32.9	80.9	98.1	66.1	100	100	92	79.7	98.4
R^*	89.7	89.8	35.4	79.8	98.7	60	99.7	100	92	78.9	89.3
$R_\#$	100	39.3	3.8	21.4	61.6	15.7	100	52.6	38	44.3	77.7

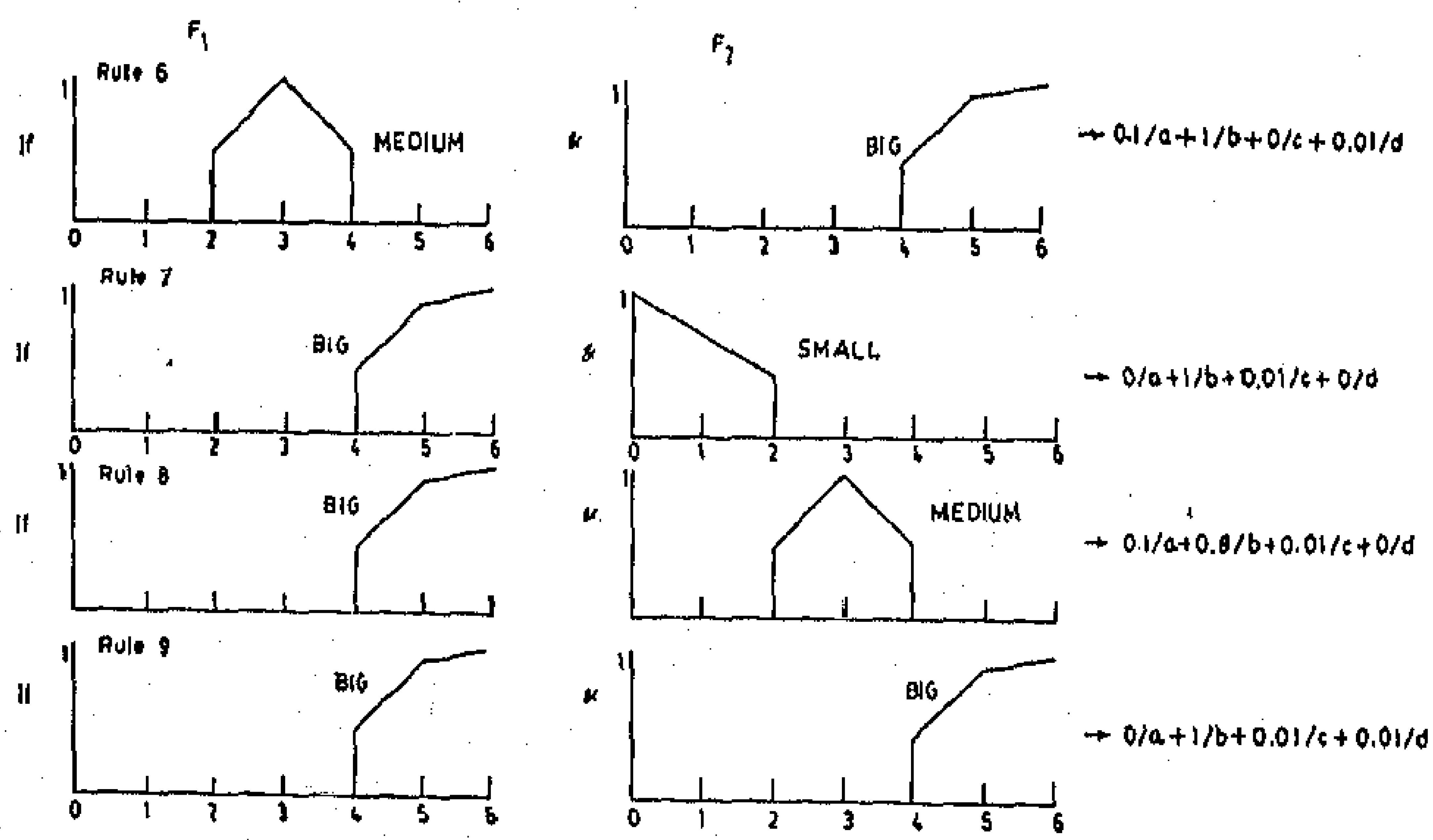


Fig. 2.16: Fuzzy If - Then rules for Fig. 2.11 (Contd.)

(Contd.) Table 2.3:Recognition scores (in %) where "and" of the antecedent part of the rule is interpreted as "algebraic product"

type of operator	fig2.10		fig2.11		fig2.12		overall score		
	a	b	a	b	a	b	fig2.10	fig2.11	fig2.12
R_c	79.8	100	100	98.1	65.9	54.7	89.3	98.8	60.7
R_p	79.8	100	100	98.1	52.1	67.9	89.3	98.8	59.6
R_{bp}	82.5	100	100	98.1	66	52.4	90.7	98.8	59.6
R_{dp}	79.8	100	100	98.1	50	72.7	89.3	98.8	60.7
R_a	82.5	100	100	98.1	66	51.2	90.7	98.8	59
R_m	100	71.3	100	36.2	69.2	41.6	86.5	58.4	56.2
R_b	79.8	100	100	98.1	66	53.6	89.3	98.8	60.2
R_s	82.5	85.2	100	94.5	87.2	19.1	83.7	96.4	55.1
R_g	80.7	100	100	98.2	66	52.4	89.8	98.8	59.6
R_Δ	80.7	100	100	98.2	66	52.4	89.8	98.8	59.6
R^*	79.8	100	100	98.2	52.2	67.9	89.3	98.8	59.6
$R_\#$	100	71.3	100	36.2	19.2	41.7	86.6	58.4	29.8

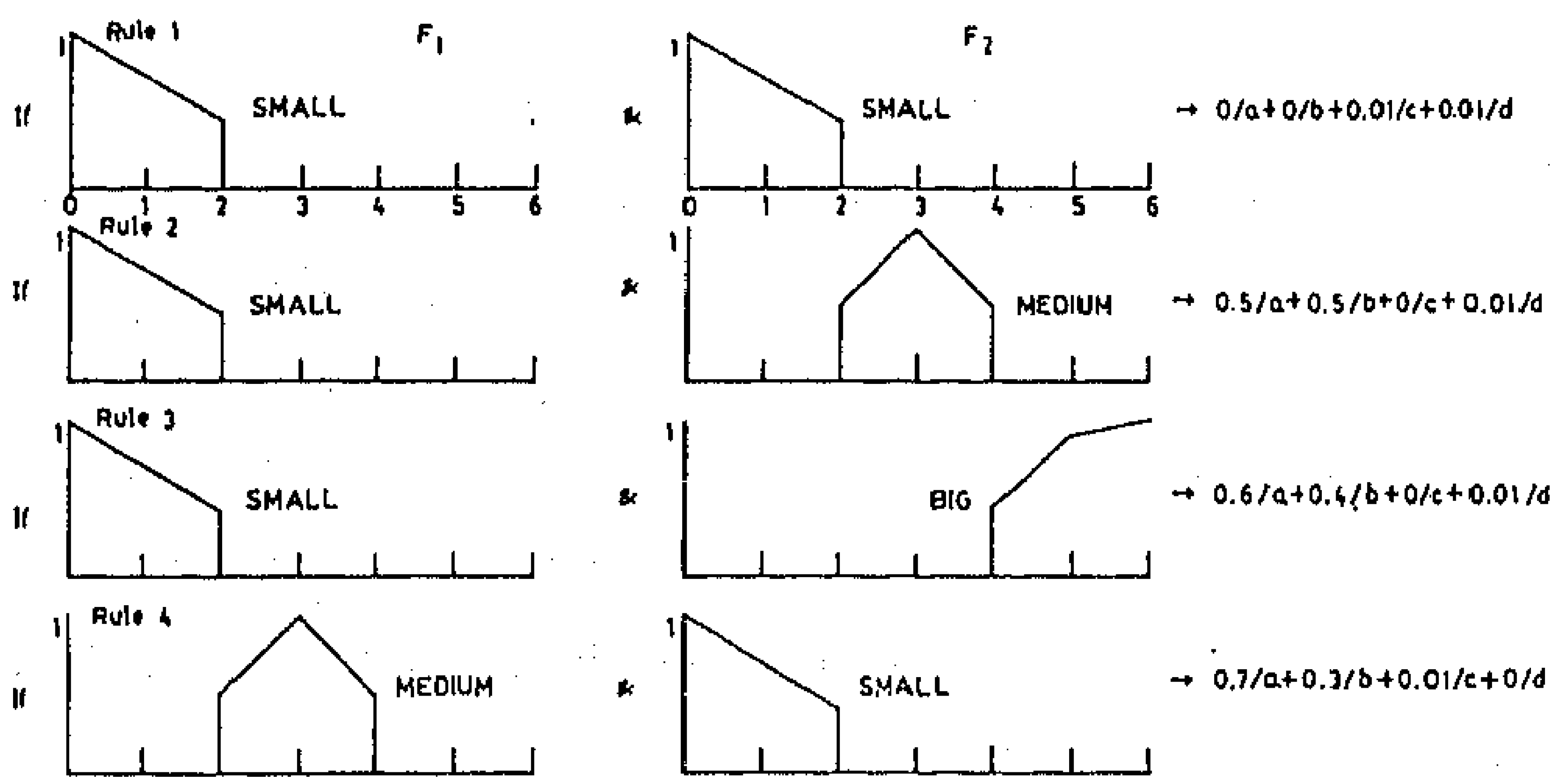


Fig. 2.17: Fuzzy If - Then rules for Fig. 2.12

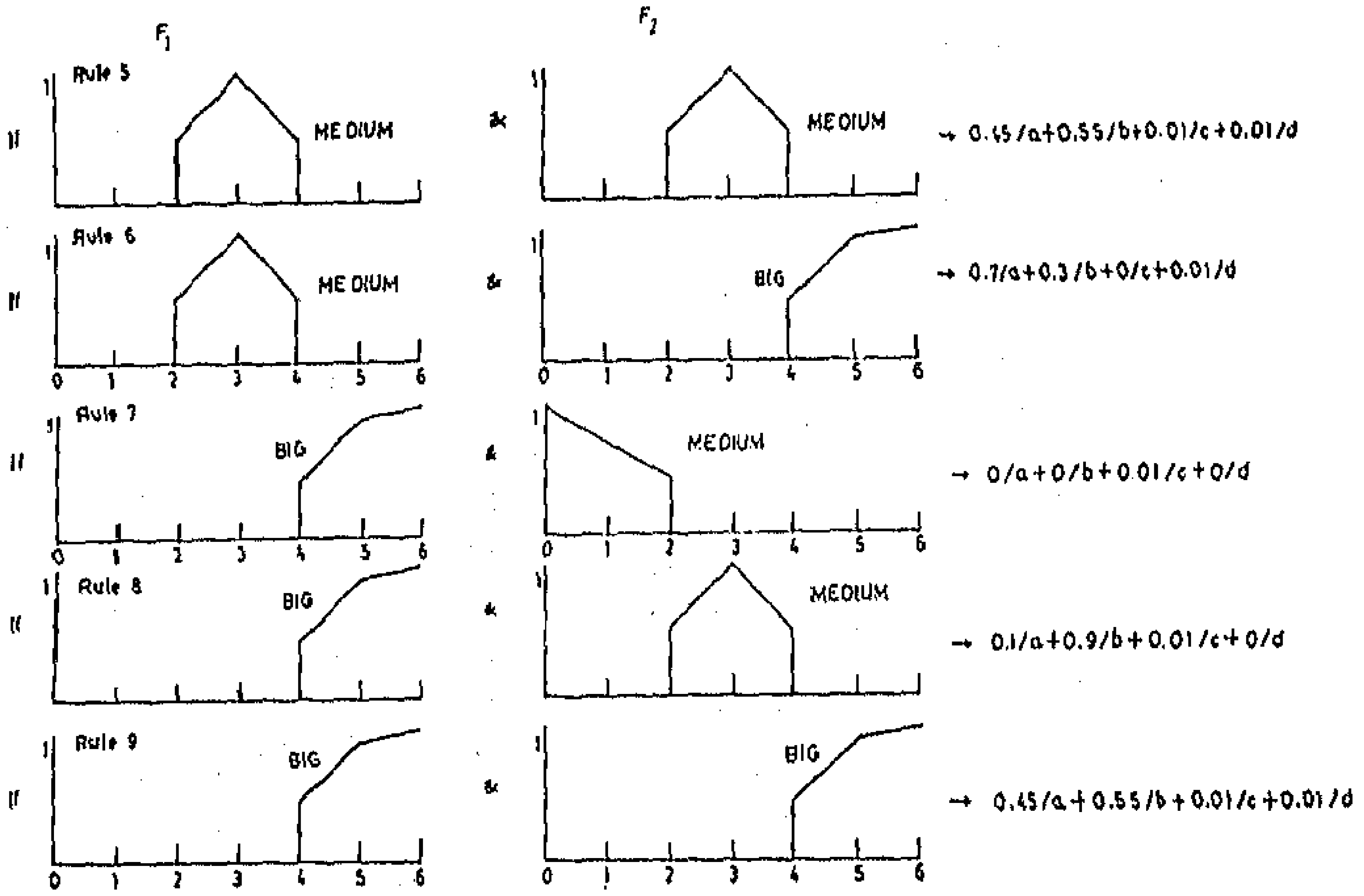


Fig. 2.17: Fuzzy If - Then rules for Fig. 2.12 (Contd.)

2.7 Applications

After achieving satisfactory results on synthetic set of data, we picked up Mamdani's law of implication, implication using Gödelian logic and implication using Gougen logic for vowel recognition problems of three Indian languages namely Telugu, Assamese and Bengali [80]. In the following subsections, we briefly discuss the experimental procedures for extraction of vowels of each language and the recognition procedure which is based on the scheme of Figure 2.7 and which is essentially pivoted on the method of fuzzy reasoning given in equation (2.12) and (2.13).

2.7.1 Experiment I with Telugu vowel

A number of discrete phonetically balanced speech samples for the Telugu vowels in CNC(**Consonant - Vowel nucleus-Consonant**) form were selected.

CNC combination is taken because the form of consonants connected to a vowel is responsible for influencing the role and quality of vowels. These speech units were recorded by five informants on an AKAI type recorder.The spectrographic analysis has been done on Kay Sonagraph Model 7029-A which is a very standard audio frequency spectrum analyzer that produces a permanent record of the spectra of any complex waveform in the range of 5 Hz to 16 KHz. For the present study of vowel, the spectrographic display of frequency Vs. time (See Figure 2.18) has been done for 800 Telugu words uttered by three male informants in the age group of 25 to 30 years chosen from 15 educated persons. The total bandwidth of the system is 80 Hz to 8 KHz with a resolution of 300 Hz.

The experiment deals with the formant frequencies at the steady state of the Telugu vowels and their variations in different consonantal context and for different speakers. The average positions (see Table 2.4) of different Telugu vowels with respect to cardinal vowels and their distribution in $F_1 - F_2$ frequency planes are considered (see Figure 2.19).

Segmentation Procedures

The purpose of segmentation is to determine the vowel boundaries and the boundary of the steady state. For the determination of vowel boundaries the segmentation procedure should satisfactorily solve the problems of determination of vowel boundary in relation to stops, fricatives, affricates, laterals in voiced/ unvoiced, aspirated/ unaspirated as well as their combined manners. The boundaries in these three different situations need

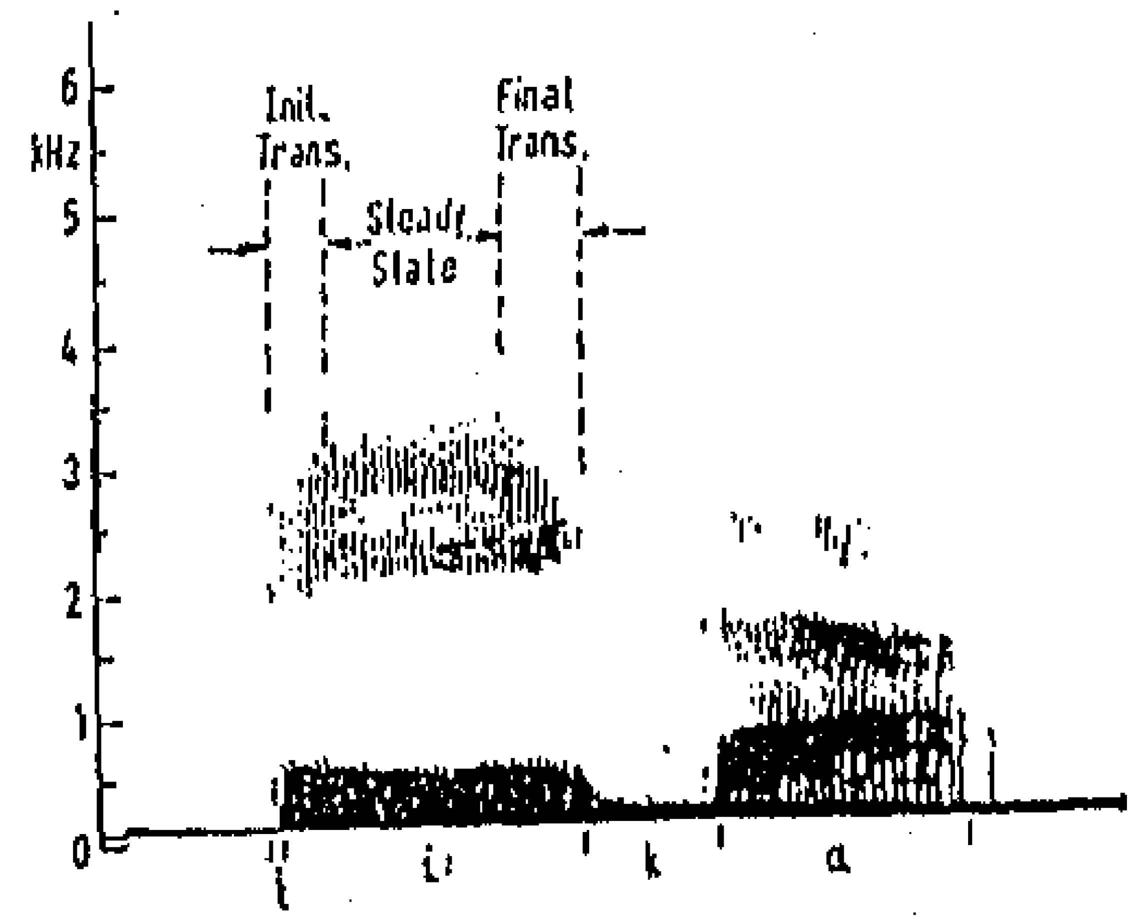


Fig 2.18: Spectrogram of the Telugu Word[ṭi:ka]

to be defined first.

1. **Vowel in combination with stop consonants**

   The unaspired stop in initial position appears as spike in the spectrogram and this energy of plosion will clearly demarcate stop consonant and the adjacent vowel formants (Figure 2.18). For aspirated stop in the initial position the onset of voicing is taken to be the boundary.

   For the stop consonant in the final position, the absence of all vowel formants with the start of occlusion period of the consonant will indicate the termination of the vowel (Figure 2.18).

2. **Vowel in combination with fricatives and affricates**

   The fricatives with a wide-band continuous energy spectrum of low intensity can be easily segmented from the narrow band of much stronger intensity of vowel formants. The line separating these two distinctly separate spectral distributions is the vowel boundary for fricatives in the initial and also in the final position. The line of demarcation for affricates in initial and final position is same as in fricative and stop consonants respectively.

3. **Vowel in combination with liquids**

The segmentation problem becomes more complex for the liquid and vowel combination. The liquids possess a formant like structure very similar to vowel formants and thus create real confusion. But, careful observation will reveal that the formant structure of liquids is less intense with a much lesser degree of transition. These characteristic differences are used for determining the vowel boundary in this case.

Measurement Procedure

The steady state of the vowel is that part on the record in which all formants lie parallel to the time axis (Figure 2.18). The transition is depicted by the inclined formant patterns (Figure 2.18).

The exact point of inflection is difficult to locate in the records. This can be done very satisfactorily by tracing the central line for each formant band. Once these points are located for all available formants, the steady state of the vowel is taken to be the shortest horizontal span for all the formants.

The formant frequencies are measured from the baseline (i.e. zero-line) of the spectrogram to the central line of formant bands where the formant is in a steady state. The scale used for this measurement is derived from the calibrated tune of 500 Hz recorded on each and every spectrogram. One small division of the scale is equal to 20 Hz. A rechecking of 5 % samples reveals that formant frequencies have been recorded within an accuracy of 10 Hz. For every 50 spectrograms two fine marker recordings, one at the beginning and one at the end are taken. The scale for duration is done by taking an average of these two recordings. However, throughout the whole recording non-cognizable differences between these two recordings are observed. In a few cases, for particularly fast informants, it has been noted that the vowel hardly reaches a stable state. In such cases, the congruence of on-glide and off-glide has been taken as the steady state.

Recognition of Telugu vowels

The present investigation has been carried out with the Telugu vowels (listed in Table 2.4) both short and long. It is well known that the first three formant frequencies carry most of the information regarding the vowel quality. But for all practical purposes of vowel recognition, we can use the first two formant frequencies, i.e. F_1 and F_2.

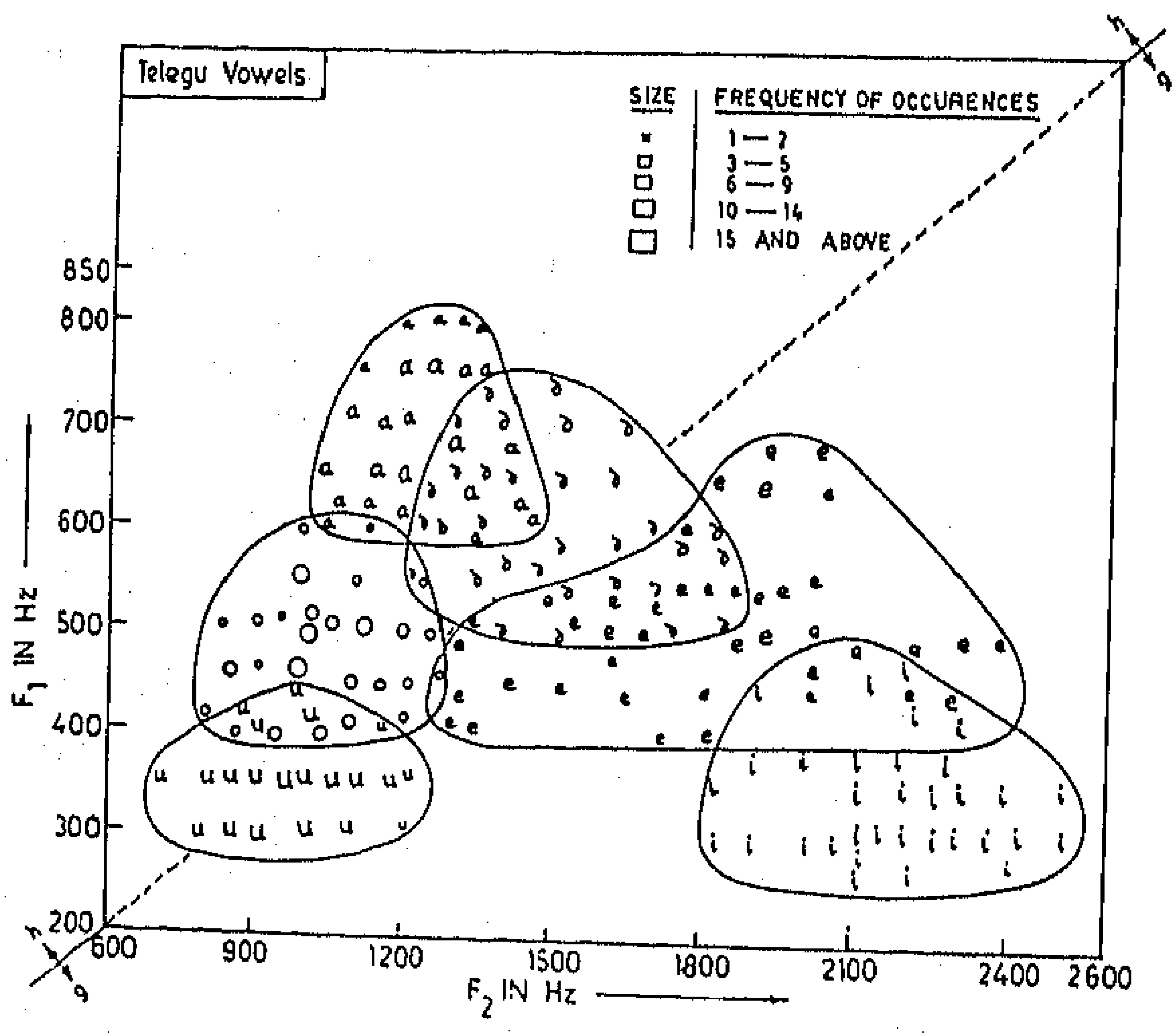


Fig. 2.19: Telugu Vowels in the F_1 - F_2 plane

For recognition of Telugu vowels the $F_1 - F_2$ distribution of Figure 2.19 is considered. The rules and the distribution patterns are shown in Figure 2.20. The recognition score with Mamdani's law of implication (where the connective "and" of the antecedent part of the rule is interpreted as "min") is given in Table 2.5. It is obvious from the Table 2.5 that when we consider the multiple classification of vowels the recognition score has been increased. The recognition score using other operators are also similar to the one mentioned in Table 2.5 and hence omitted here.

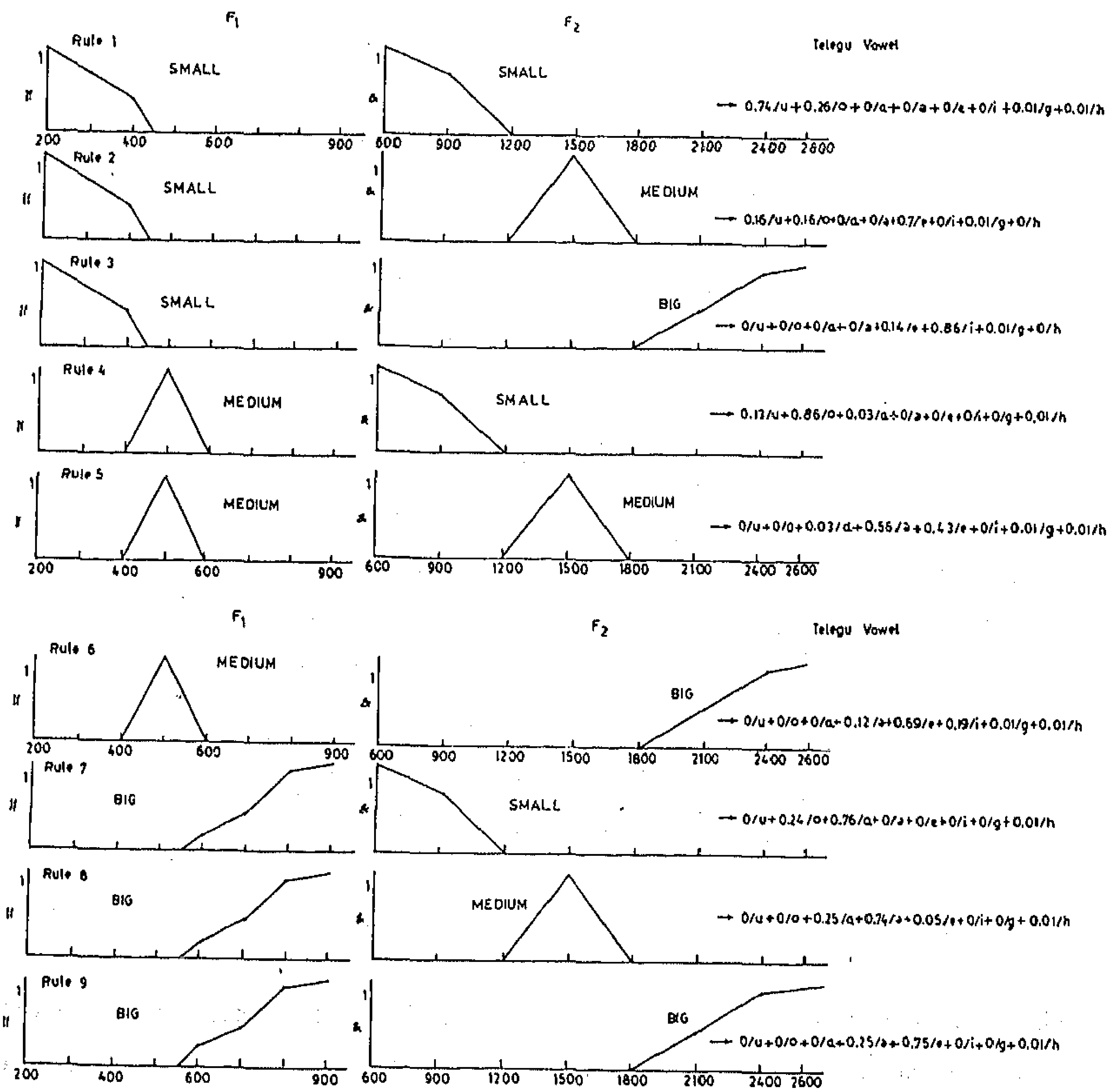


Fig. 2.20: Fuzzy If - Then rules for Telugu Vowels

Table 2.4: Average formant frequencies of Telugu vowels

PHONETIC SYMBOL	F_1 Hz	F_2 Hz	F_3 Hz
/ ə /	606	1473	2420
/ a : /	710	1240	2400
/ i ⊢ /	365	2116	2757
/ i : /	325	2260	2836
/ u ⊣ /	370	1066	2500
/ u : /	348	923	2543
/ e ⊢ /	517	1796	2633
/ e : /	470	1883	2657
/ o ⊣ /	476	1133	2630
/ o : /	486	1000	2540
/ ae /	575	1744	2700

Table 2.5: Recognition scores (in %) of the Telugu vowels where "and" of the antecedent part of the rule is interpreted as "min"

hard partitioning (i.e. classification by single choice)	fuzzy partitioning (i.e. multiple classification)
60.2	78

2.7.2 Experiment II with Bengali vowels

This experiment has been conducted on a sample of carefully selected 350 commonly spoken Bengali words so that the vowels can be studied in all possible **Consonant-Nucleus-Consonant** contexts. These words are uttered by 10 male and 10 female educated and phonetically conscious informants drawn from linguists, professors of literature and dons of performing arts. Of all these records, data for 3 male informants has been chosen on the basis of good and clear spectrographs. The age of the informants varied between 30 to 55 years. The spectrographic analysis has been done on Kay Sonagraph model 7029-A in the bandwidth of 80 Hz to 8 KHz using a resolution of 300 Hz. The acoustic data, namely the first four formant frequencies and the duration are derived from the spectrographs. As the vowels are embedded in various consonantal contexts in the multisyllabic words, appropriate segmentation procedures are adopted to accurately fix both the transitions and the steady state of vowels. The details of the segmentation procedure and measurement procedure are same as stated earlier. Table 2.6 represents the average formant frequencies for first three formants.

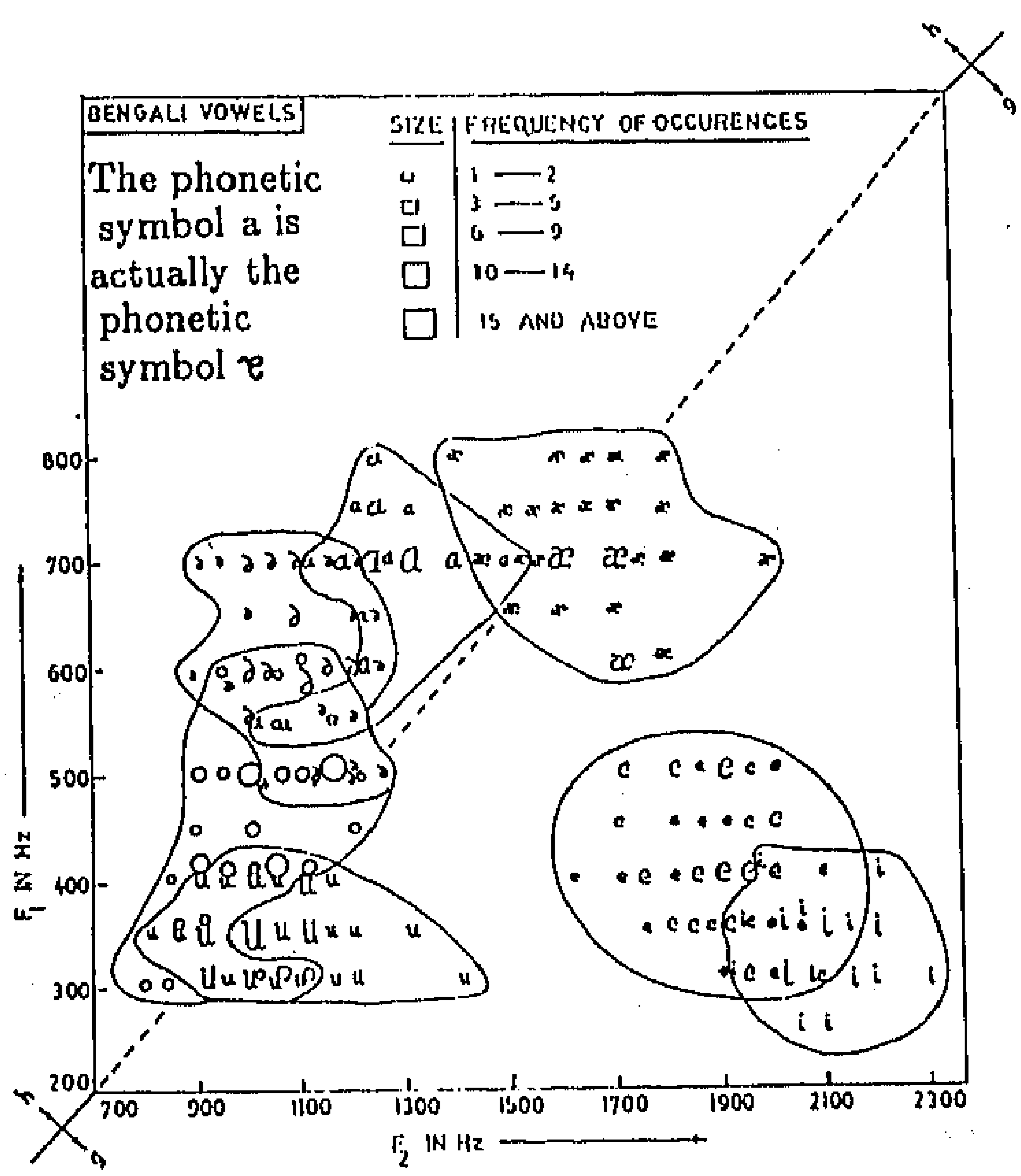


Fig. 2.21: Bengali vowels in the F_1 - F_2 plane

For recognition of Bengali vowels the $F_1 - F_2$ distribution of Figure 2.21 is considered. The rules and the distribution patterns are shown in Figure 2.22. The recognition score with Mamdani's law of implication (where the connective "and" of the antecedent part of the rule is interpreted as "min") is given in Table 2.7. It is obvious from Table 2.7 that when we consider the multiple classification of vowels, the recognition score has significantly improved. The recognition score using other operators are also similar to the one mentioned in Table 2.7 and hence omitted here.

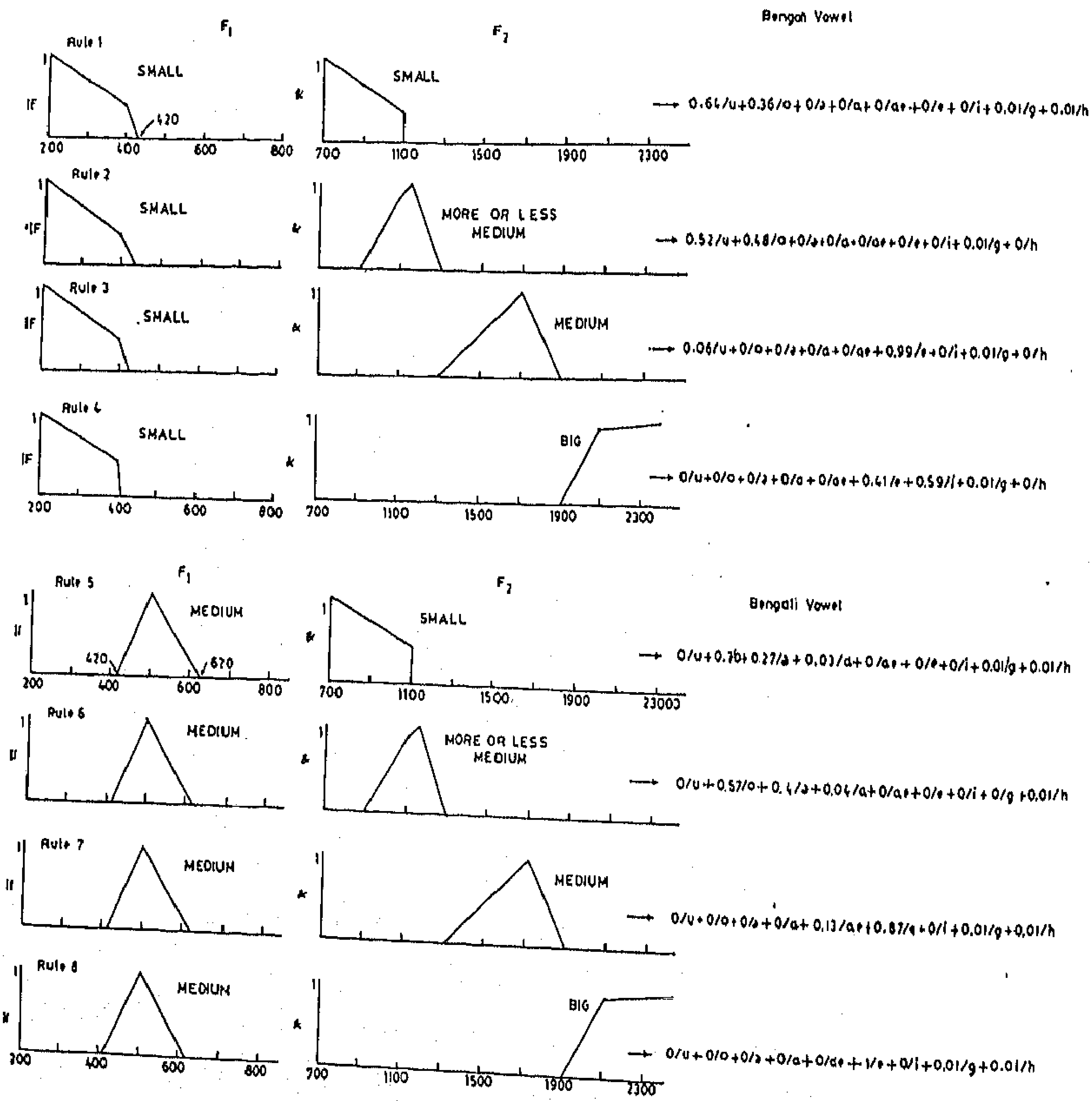


Fig. 2.22: Fuzzy If - Then rules for Bengali vowels

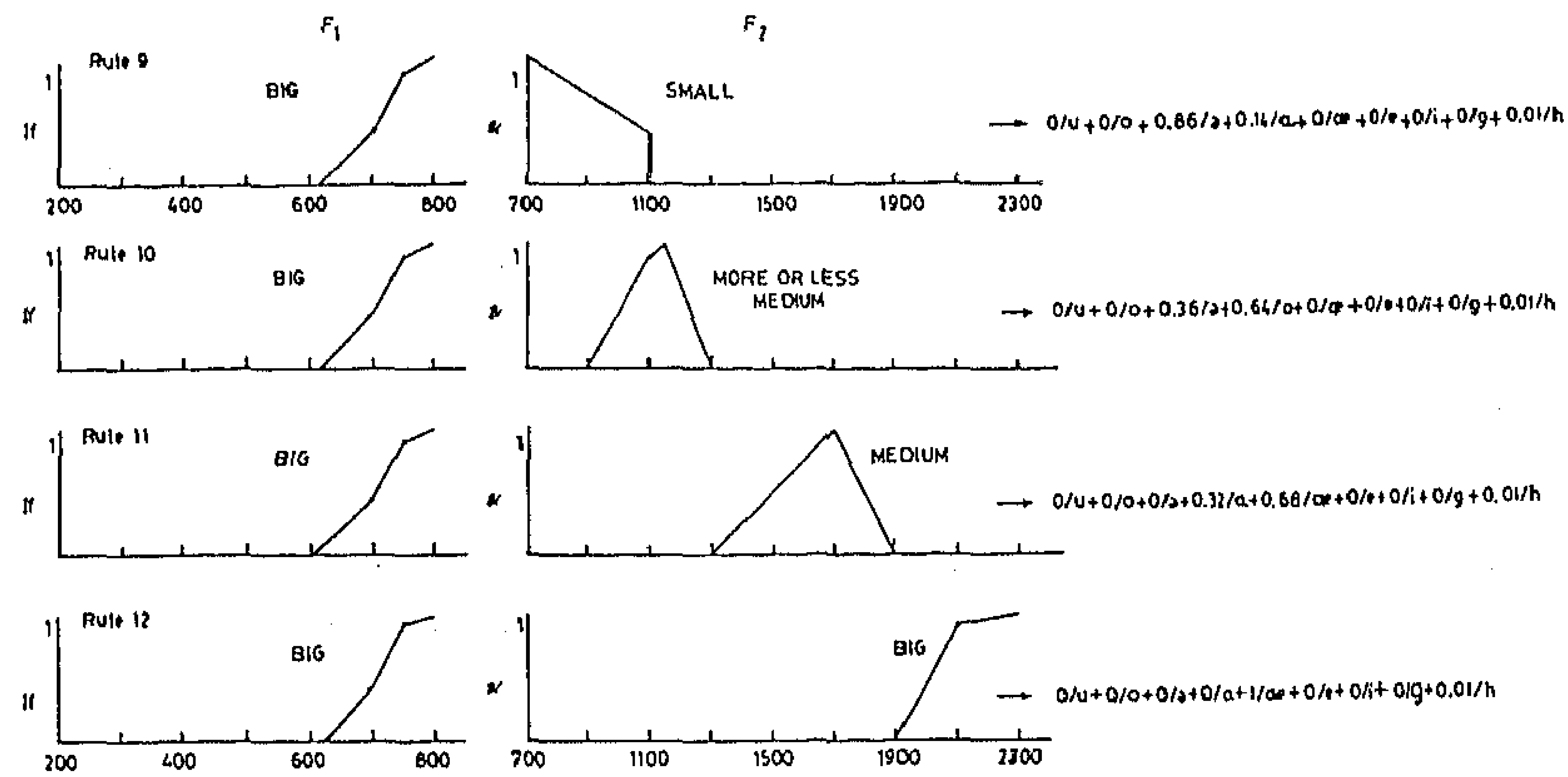


Fig. 2.22: Fuzzy If - Then rules for Bengali vowels (Contd.)

Table 2.6: Average formant frequencies of Bengali vowels

PHONETIC SYMBOL	F_1 Hz	F_2 Hz	F_3 Hz
/ u/	327	935	2198
/ o/	438	1015	2308
/ ə/	626	1095	2391
/ a/	695	1326	2424
/ ae/	681	1663	2320
/ e/	374	1935	2410
/ i/	304	2095	2565

Table 2.7: Recognition scores (in %) of the Bengali vowels where "and" of the antecedent part of the rule is represented as "min"

hard partitioning	fuzzy partitioning
62	96.4

2.7.3 Experiment III with Assamese vowels

This experiment has been conducted on a sample of carefully selected 300 commonly spoken Assamese words so that the vowels can be studied in all possible Consonant-Vowel-Consonant (CVC) contexts. These words are uttered by one male highly educated and phonetically conscious informant. The age of informant is about 55 years. The spectrographic analysis has been done on digital sonagraph model ASP 5500 in the bandwidth of 8/0 Hz to 8 KHz using a resolution of 300 Hz. The first three formant frequencies of the vowels are measured at the section taken at central portion of the steady state. The details of the segmentation and measurement procedures are same as stated earlier. Table 2.8 represents the average frequencies for the first three formants of Assamese vowels.

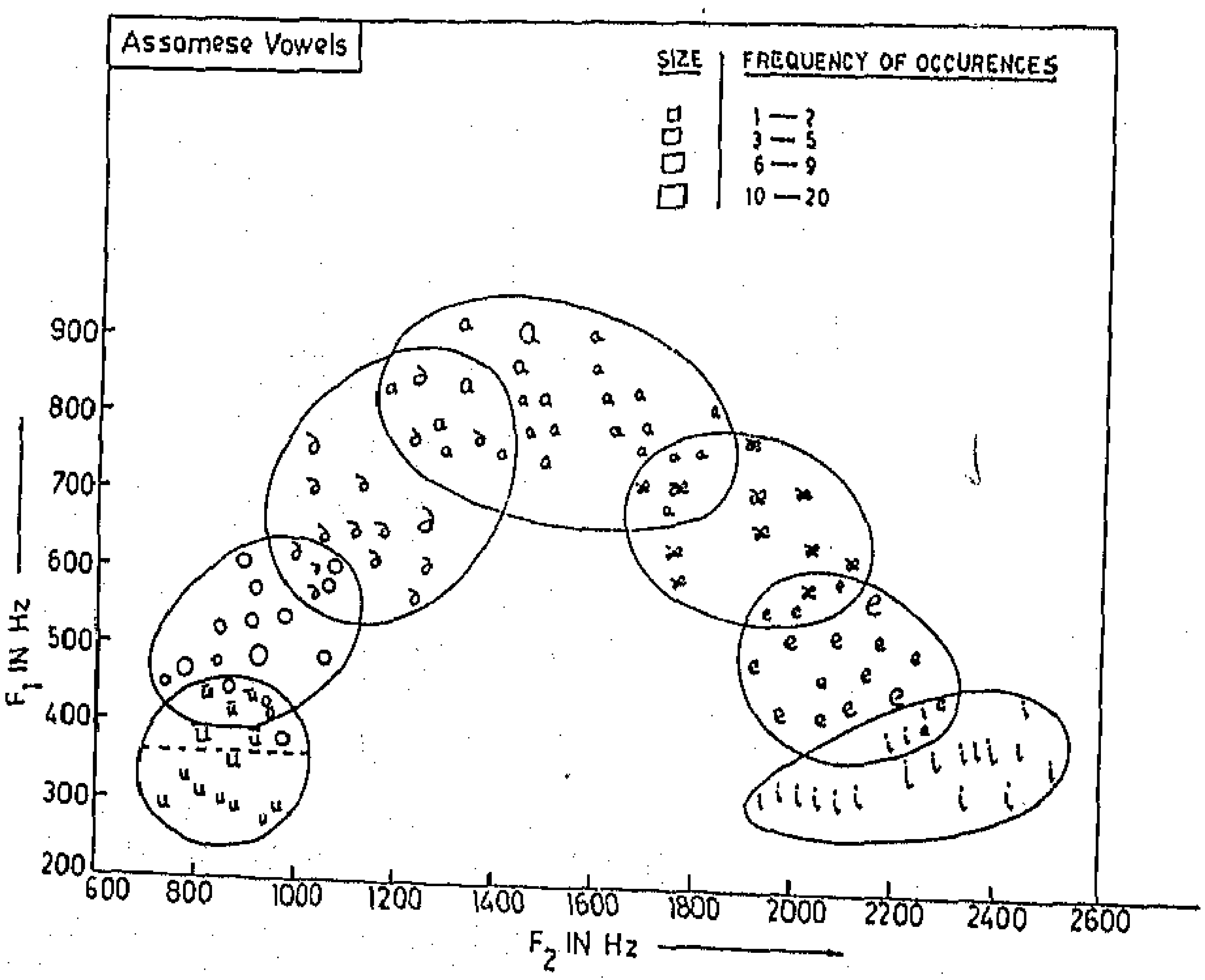


Fig. 2.23: Assamese vowels in the F_1 - F_2 plane

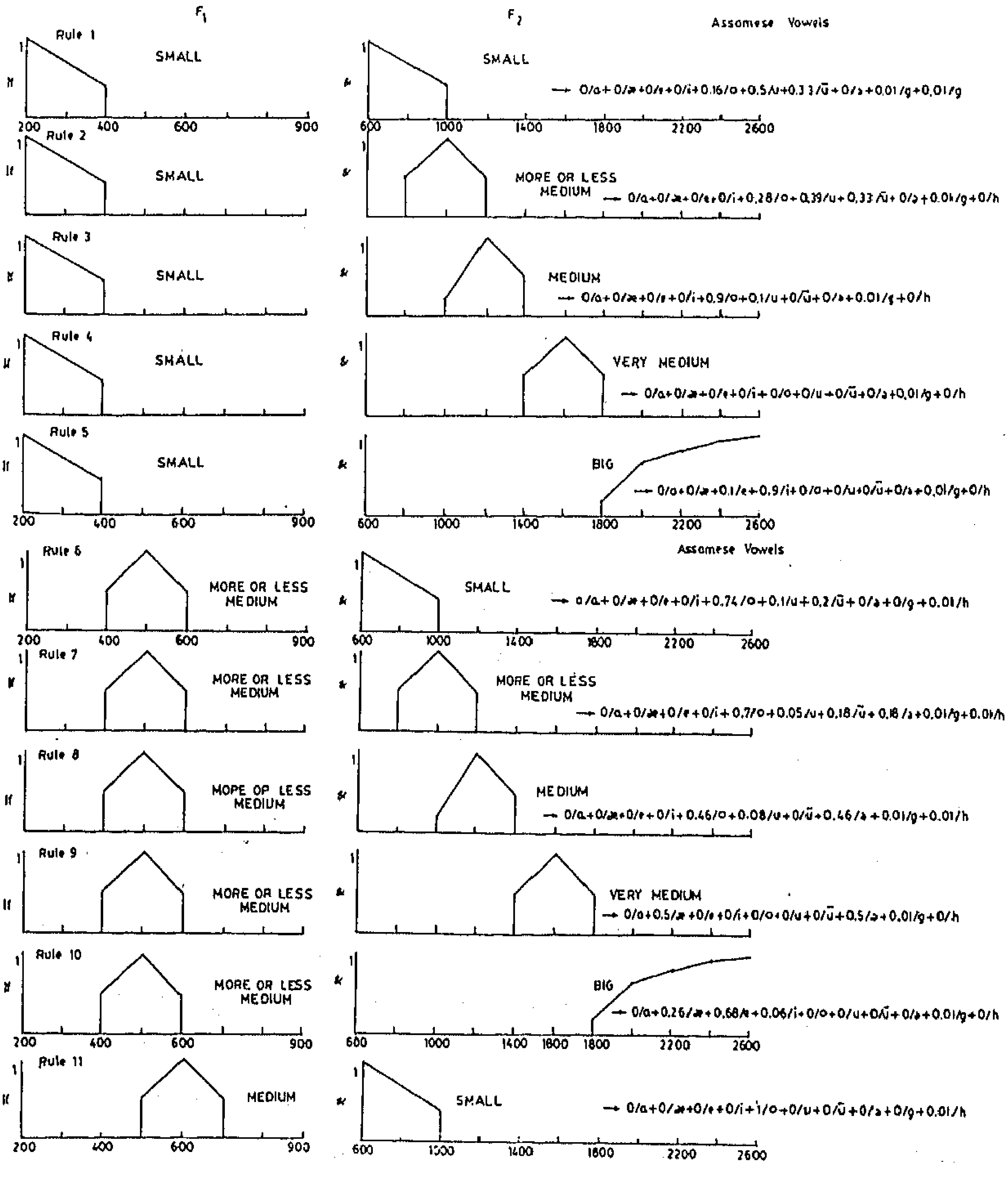


Fig. 2.24: Fuzzy If - Then rules for Assamse Vowels

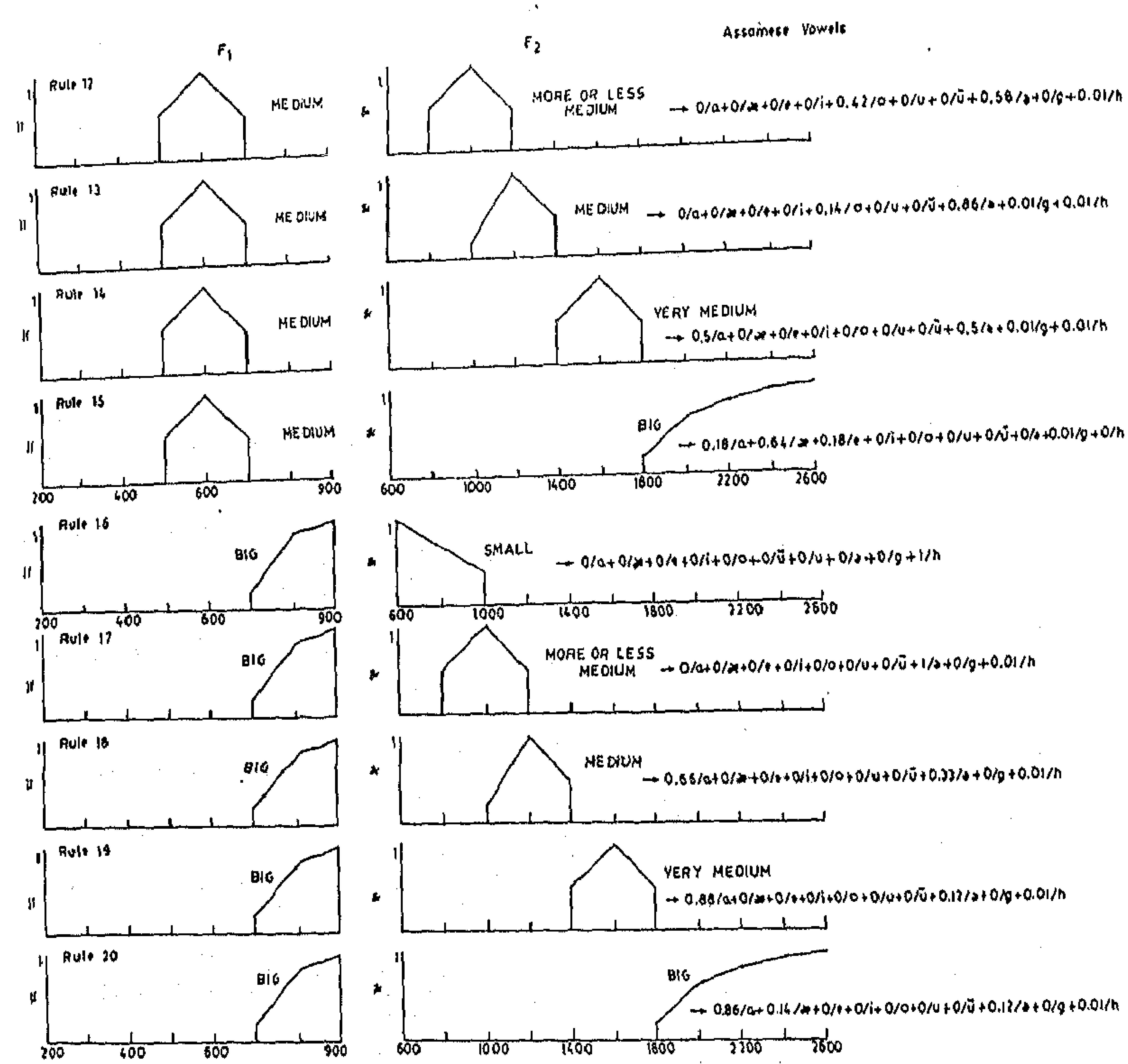


Fig. 2.24: Fuzzy If - Then rules for Assamse Vowels (Contd.)

For recognition of Assamese vowels the $F_1 - F_2$ distribution of Figure 2.22 is considered. The rules and the distribution patterns are shown in Figure 2.23. The recognition scores under different laws of implication of Table 2.1 and under two different interpretations of the connective "and" of the antecedent part of the rules are listed in Table 2.9. From Table 2.10, it is obvious that when we consider multiple classification the recognition scores have improved. In Table 2.11, we have compared the results obtained by our method and those obtained by Bayesian method. Thus, we further establish the suitability of the present method for pattern classification.

Table 2.8: Average formant frequencies of Assamese vowels

PHONETIC SYMBOL	F_1 Hz	F_2 Hz	F_3 Hz
/ *u*/	320	820	2880
/ ū/	380	800	2800
/ *o*/	430	840	3020
/ ə /	620	1290	2830
/ *a*/	820	1530	2780
/æ/	680	1860	2760
/ *e*/	460	2140	2720
/ *i*/	320	2380	2770

Table 2.9: Recognition scores (in %) of the Assamese vowels where "and" of the antecedent part of the rule is represented as "min" and then "algebraic product"

Type of operator	min	algebraic product
	hard partitioning	hard partitioning
R_c	78.4	76.7
R_p	81.8	81.8
R_{bp}	81.6	75.8
R_{dp}	76.7	76.6
R_a	75.8	75.8
R_m	63.8	50.8
R_b	75.8	69.8
R_s	45.6	59.5

Table 2.9 (Contd.):

Type of operator	min	algebraic product
	hard partitioning	hard partitioning
R_g	73.3	75
R_Δ	74.13	74.13
R^*	75.8	77.6
$R_\sharp$	61.2	45.6

Table 2.10: Recognition scores (in %) of the Assamese vowels where "and" of the antecedent part of the rule is represented as "min" and then "algebraic product"

Type of operator	Min	Algebraic product
	fuzzy partitioning	fuzzy partitioning
R_c	97.4	97.4
R_p	97.4	96
R_{bp}	97.4	95

Table 2.11: Results of Bayesian classification (in %)

fig 2.8	fig 2.9	fig 2.10	fig 2.11	fig 2.12	fig 2.19	fig 2.21	fig 2.23
78	86.9	88.4	60.8	59.3	80.3	86.1	82.7

2.8 Remarks

In this chapter, we have considered pattern classification on $\Re^2$. This approach can easily be extended for pattern classification on $\Re^n$. So far, people have experienced the success of the method of fuzzy reasoning for the design of the fuzzy logic controller. This work further establishes the potentiality of the method of fuzzy reasoning for rule based pattern classification. In this work, we have considered several examples using synthetic data and real life application to vowel recognition. In all these cases very satisfactory results have been achieved under different laws of fuzzy implications and the max - min composition. Instead of considering max - min composition, depending upon the need of the problem, we may also use other types of composition.

2.9 Multidimensional Fuzzy Reasoning

In the earlier sections, we design pattern classifiers using one dimensional fuzzy implication (see Equation (2.11)). But, we know that for pattern classification problem, we usually deal with pattern vector / feature vector. That means, essentially, we have to deal with multidimensional fuzzy implication(MFI).

Since a multidimensional fuzzy implication (MFI) such as "if(x is A, y is B) then z is C" where A, B and C are fuzzy sets is not merely a collection of one-dimensional implications, a conventional interpretation is usually taken in the multidimensional case. For Example, according to the conventional interpretation, the above two-dimensional implication is translated into,

$$\left.\begin{array}{llll} \text{1) if x is A} & \text{and y is B} & \text{then z is C,} & \text{(a)} \\ \text{or 2) if x is A} & \text{then if y is B} & \text{then z is C.} & \text{(b)} \end{array}\right\} \quad (2.14)$$

Thus, using the interpretation of (2.14(a)), we have designed the pattern classifier in section 2.3. In section 3.3 of Chapter 3, we implement this interpretation on back-propagation neural network to generate neuro fuzzy reasoning (fusion) technology for pattern classification.

Note that the interpretation of Equation (2.14(a)) of a MFI means a fuzzy vector shown in Figure 2.26(a) and a fuzzy set C formed by the relative position of the fuzzy vector and the well defined cover of the pattern space. The relation formed by the antecedent clauses of Equation (2.14(a)) is represented at the tip of the fuzzy vector i.e. the area ABCD of Figure 2.26(a). In Figure 2.26(a) the relation contained in the area ABCD is obtained by considering 'min' operation between the corresponding elements of the two fuzzy sets namely M_1 and M_2 defined over two feature axes F_1 and F_2 respectively. That means, the relation formed by the antecedent clauses of Equation (2.14(a)) is obtained by considering 'min' operator for the linguistic connective 'and' between two antecedent clauses. Instead of a 'min' operator we may consider other kind of operators like algebraic product etc. to replace the connective 'and' between the antecedent clauses of Equation (2.14(a)). In that case, the tip of the fuzzy vector i.e. the area ABCD of Figure 2.26(a) represents a relation of different nature. Precisely speaking to deal with the relation formed by the antecedent clauses of Equation (2.14(a)) means to deal with a fuzzy vector of the type shown in Figure 2.26(a). The relation which is contained in the area ABCD of Figure 2.26(a) implies the consequence C as indicated in Equation (2.14(a)). The fuzzy set C of Equation (2.14(a)) is obtained as discussed in Example 2.2.

According to Tsukamoto [100], a multidimensional fuzzy implication (MFI) can be interpreted as,

$$\left.\begin{array}{ll} \text{1) if x is A} & \text{then z is C} \\ \text{and 2) if y is B} & \text{then z is C} \end{array}\right\} \qquad (2.15)$$

and the intersection $C' \bigcap C''$, where C' is the inferred value from the first implication and C'' that from the second implication, is taken for the consequence of reasoning. In the multidimensional case, we have some difficulties in applying the compositional rule of inference, because we have to deal with multidimensional fuzzy relations.

Hence, for multidimensional fuzzy reasoning (MFR) using multidimensional fuzzy implication, the usual approach (based on Tsukamoto's interpretation) adopts a method similar to linear interpolation [91]. Let us consider a two dimensional case. We can start with four implications such that

$$\begin{aligned}
(\text{x is } A_1, \text{y is } B_1) &\to \text{z is } C_{11},\\
(\text{x is } A_1, \text{y is } B_2) &\to \text{z is } C_{12},\\
(\text{x is } A_2, \text{y is } B_1) &\to \text{z is } C_{21},\\
(\text{x is } A_2, \text{y is } B_2) &\to \text{z is } C_{22},
\end{aligned}$$

The important point we want to stress is that we can not translate (x is A, Y is B) into, for example, "(x , y) is $A \times B$" where $A \times B$ is Cartesian product, since generally a pair of two propositions (x is A, y is B) could be any relation. This means that we should deal with (x is A, y is B) directly without modification.
The situation of four implications is shown in Fig. 2.24(a).
Our problem is to infer "z is C" from a given premise (x is A, y is B) where A is assumed to be between A_1 and A_2 and also B between B_1 and B_2.
First let us infer the value of z at the point (A_1, B) from (A_1, B_1) as indicated by an arrow in Fig. 2.24(b). Modus ponens is written as,

$$\frac{(x \text{ is } A_1, y \text{ is } B)\ ,\ (x \text{ is } A_1, y \text{ is } B_1) \to z \text{ is } C_{11}}{z \text{ is } C'_{11}}$$

Here C'_{11} is easily obtained by one-dimensional reasoning.
Let us next infer z at the same point(A_1, B) this time from (A_1, B_2). That is

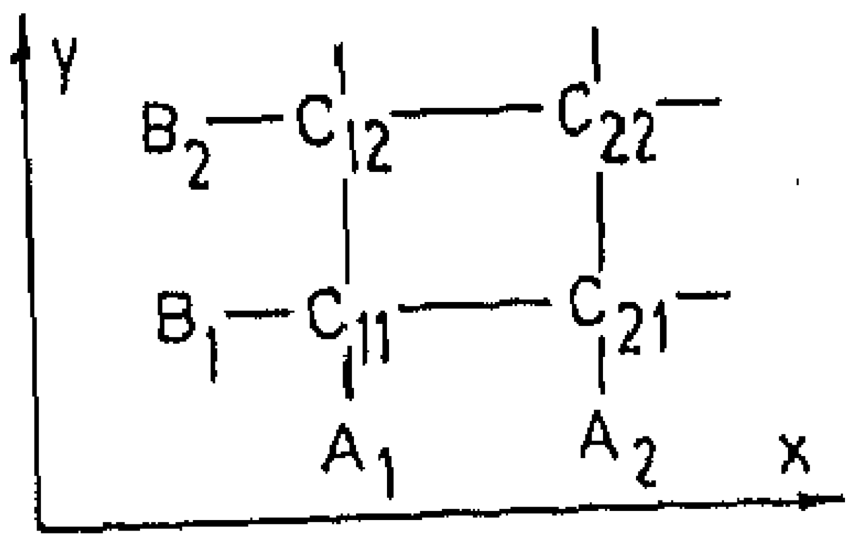
$$\frac{(x \text{ is } A_1, B), (x \text{ is } A_1, B_2) \to z \text{ is } C_{12}}{z \text{ is } C'_{12}}$$

Now, let

$$D_1 = \frac{(C'_{11} + C'_{12})}{2}.$$

Then, we obtain a new implication with respect to the point (A_1, B)such that

$$(x \text{ is } A_1, y \text{ is B }) \to z \text{ is } D_1.$$



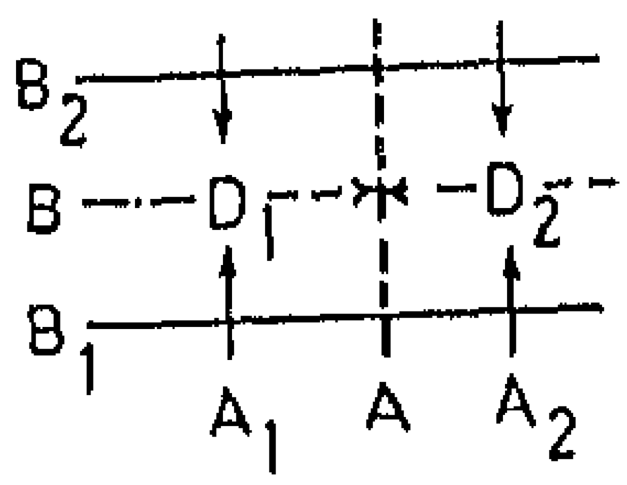


Fig. 2.25(a): Four implications [91] Fig. 2.25(b): Outline of algorithm [91]

Here D_1 may be obtained in another way: $D_1 = C11' \cap C'_{12}$. For the point (A_2, B) we can also obtain an implication such that,

$$(x \text{ is } A_2, y \text{ is B }) \rightarrow z \text{ is } D_2.$$

Finally, if we follow the same procedure along the dotted arrows in Fig. 2.24, we can infer the value of z at (A, B).

To tackle the pattern classification and / or object recognition problem using MFR we provide the following new interpretation of the multidimensional fuzzy implication(MFI) [82,83].

$$\left.\begin{array}{lll} & \text{if x is A} & \text{then z is } C_1 \\ \text{and} & \text{if y is B} & \text{then z is } C_2 \end{array}\right\} \qquad (2.16)$$

and the intersection $C'_1 \cap C'_2$, where C'_1 is the inferred value from the first implication and C'_2 is the inferred value from the second implication, is taken for the consequence of reasoning. Note that the fuzzy set $\hat{C} = C_1 \cap C_2$ and C will carry the same defuzzy information which is ultimately needed for classification of patterns / objects (see Example 2.3).

To deal with the individual DFI of Equation (2.16) means to deal with a single antecedent clause along with its cylindrical extension over the appropriate universe of the feature axis and the consequence C_i, where i varies from 1 to 2 (in case we consider pattern classification on $\Re^2$). The necessity of considering the cylindrical extension of the fuzzy set which represents the antecedent clause of a DFI is to induce a relation in the pattern space. The induced relation due to the cylindrical extension of the said fuzzy set is itself a fuzzy set (see Figure 2.26(b)) which according to Definition 2.1 is a fuzzy vector. Thus we can locate patterns on the pattern space by a fuzzy vector $\vec{F_{fj}}$ induced by the cylindrical extension of the fuzzy set of the antecedent clause of a DFI. The consequence C_i (i varies from 1 to 2) of a DFI is formed by the relative position of the induced fuzzy vector as stated above and the well defined cover of the pattern space (see Example 2.2). Therefore, when we consider the antecedent part of a DFI along with its consequence we implicitly consider an induced fuzzy vector as stated above *which implies the consequence* C_i, *i varies from 1 to 2.*

The essential difference between Tsukamoto model and the newly proposed model occurs at the interpretation of the consequent part of each of the decomposed fuzzy implication (DFI) of the multidimensional fuzzy implication (MFI). According to Tsukamoto, the consequent parts of the DFIs (see Equation (2.15)) of a MFI are same as the consequent part of the said MFI. Whereas, in the newly proposed model, the consequent parts of the DFIs (see Equation (2.16)) are different from the consequent part of the MFI. The linguistic connective "and" of equation(2.16) has a logical interpretation "$\bigcap$".

Based on the conventional interpretation of MFI, i.e. Equation (2.14(a)), we introduce the notion of fuzzy pattern vector / feature vector ($\vec{F_{fj}}$) (see Figure 2.26(a)) whose elements are the antecedent parts of DFIs (see Equation (2.16)) which (i.e. the antecedent parts of DFIs) are basically the linguistic features like F_1 is **small** and F_2 is **medium** etc. (see Figure 2.26(a),(b),(c)) where **small**, **medium** etc. are represented by fuzzy sets. The tip (ABCD) (see Figure 2.26(a)) of the fuzzy pattern vector / feature vector $\vec{F_{fj}}$ represents a population of patterns instead of a single pattern (see Example 2.1). In the context of pattern classification, the consequent part of a MFI which is a fuzzy set in the pattern space represents the possibility of occurrence of different classes of patterns which is determined by the relative position of the fuzzy feature vector / pattern vector $\vec{F_{fj}}$ with respect to the defined cover of the pattern space. According to a particular MFI, a particular class $\tilde{C}_i$ may have higher possibility of occurrence in pattern space $\Re^2$ (for simplicity of discussion we consider $\Re^2$. Such concept holds good for $\Re^n$) than that of a class $\tilde{C}_j$, i $\neq$ j, provided a larger area (we are talking in terms of "area" because for simplicity we restrict our discussion within $\Re^2$) of the class $\tilde{C}_i$ is occupied by the tip (ABCD) of $\vec{F_{fj}}$ (see Figure 2.26(a)) than that of $\tilde{C}_j$. Whereas, according to the same MFI, the class $\tilde{C}_k$, k $\neq$ i $\neq$ j, whose area is not covered up by the tip (ABCD) of $\vec{F_{fj}}$ (see Figure 2.26(a)) is having zero possibility of occurrence. This kind of judgement of possibility for different classes of patterns in the pattern space is basically an outcome of subjective quantification of human perception as mentioned by Zadeh [108]. The fuzzy sets C_1 and C_2 of equation (2.16) are not directly obtainable only from the fuzzy set C which is the consequent part of a MFI. The fuzzy set C is formed depending upon the relative position of the fuzzy feature vector / pattern vector $\vec{F_{fj}}$ with respect to the defined cover of the pattern space. On the other hand when we consider the elements of $\vec{F_{fj}}$ individually, which are also fuzzy sets on the respective feature axis the cylindrical extension of them (the said fuzzy sets) and the well defined cover of the pattern space provide the information about the fuzzy sets C_1 and C_2 (see Example 2.3).

Example 2.2: Let us consider the fuzzy feature vector / pattern vector $\vec{F_{fj}}$ of Figure 2.26(a). The position of $\vec{F_{fj}}$ on pattern space means the area ABCD. The position ABCD of $\vec{F_{fj}}$ is obtained when

$$\vec{F_{fj}} = \begin{bmatrix} F_1 \text{ is } M_1 \\ F_2 \text{ is } M_2 \end{bmatrix}.$$

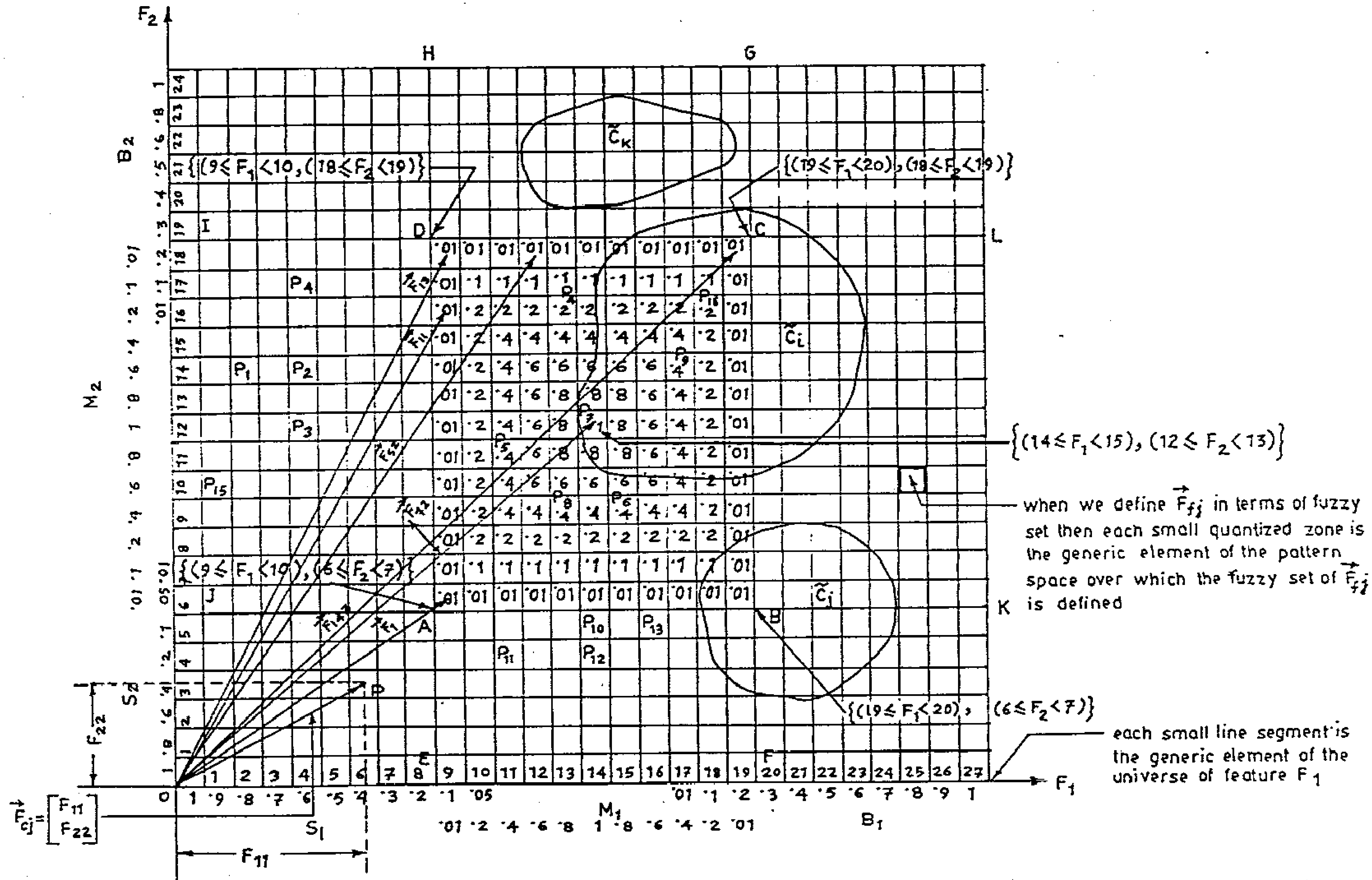


Fig-2.26(a): REPRESENTATION OF THE FUZZY FEATURE VECTOR/PATTERN VECTOR

KEY: The entries in the small quantized zones of the area ABCD are obtained by considering "min" operator between the corresponding elements of the fuzzy sets M_1 and M_2. For instance, the possibility value 0.01 of $\vec{F}_1$ at $\{(9 \leq F_1 < 10), (6 \leq F_2 < 7)\}$ is obtained by considering min $\{01/(9 \leq F_1 < 10), \cdot 01/(6 \leq F_2 < 7)\}$ of the fuzzy sets M_1 and M_2 respectively. Instead of "min" operator we may use algebraic product etc. depending upon the way we want to write the relation formed by the antecedent clauses of an one dimensional fuzzy implication.

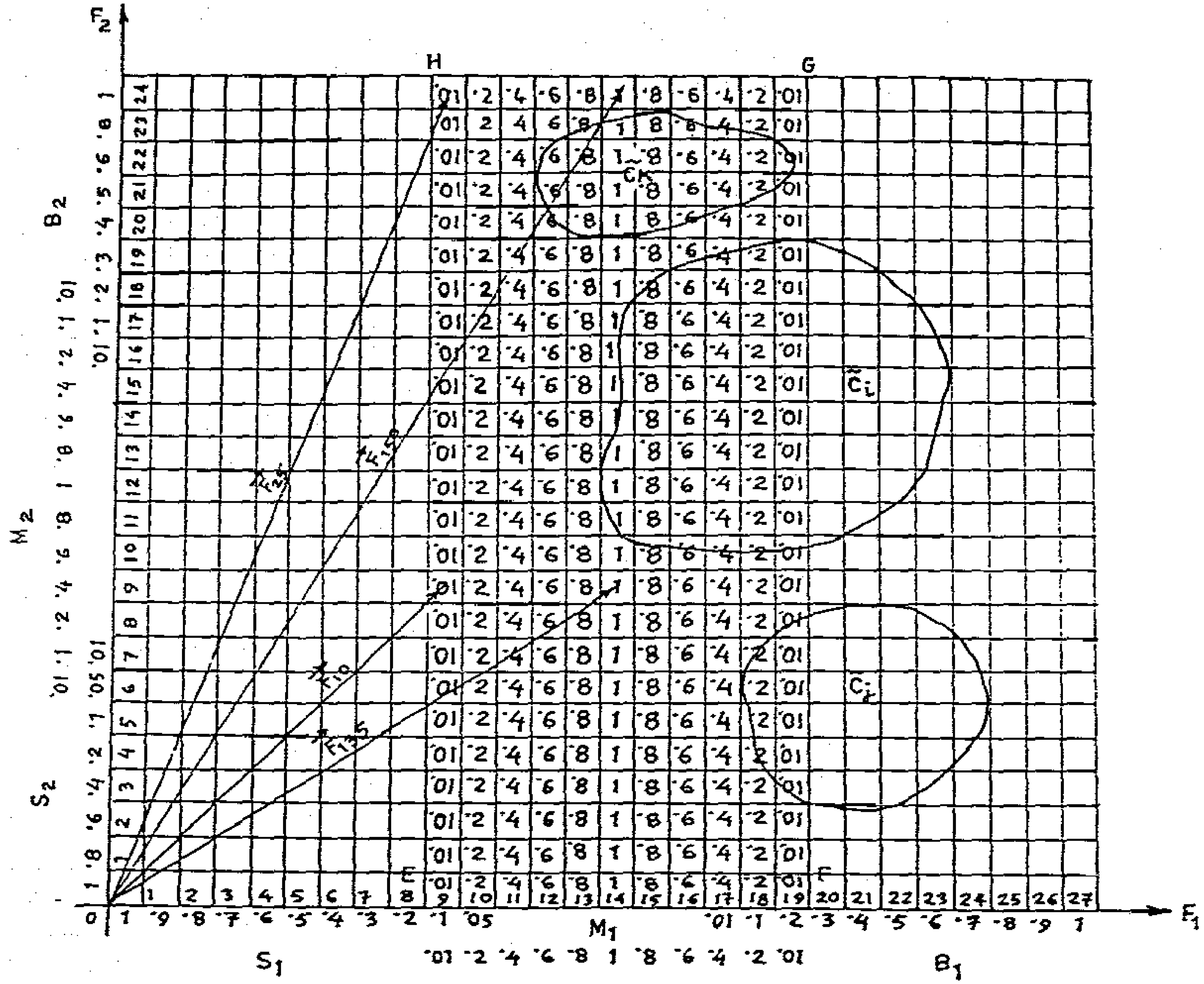


Fig.2.26(b): CYLINDRICAL EXTENSION OF M_1

KEY: The entries in the small quantized zones of the area EFGH is obtained by considering the cylindrical extension of the fuzzy set M_1. The vectors starting from $\vec{F}_{126}$ to $\vec{F}_{150}$ are the elements of the fuzzy set which are obtained by defuzzification of the fuzzy vector $\vec{F}_{fj}$ as mentioned in Example 2.2.

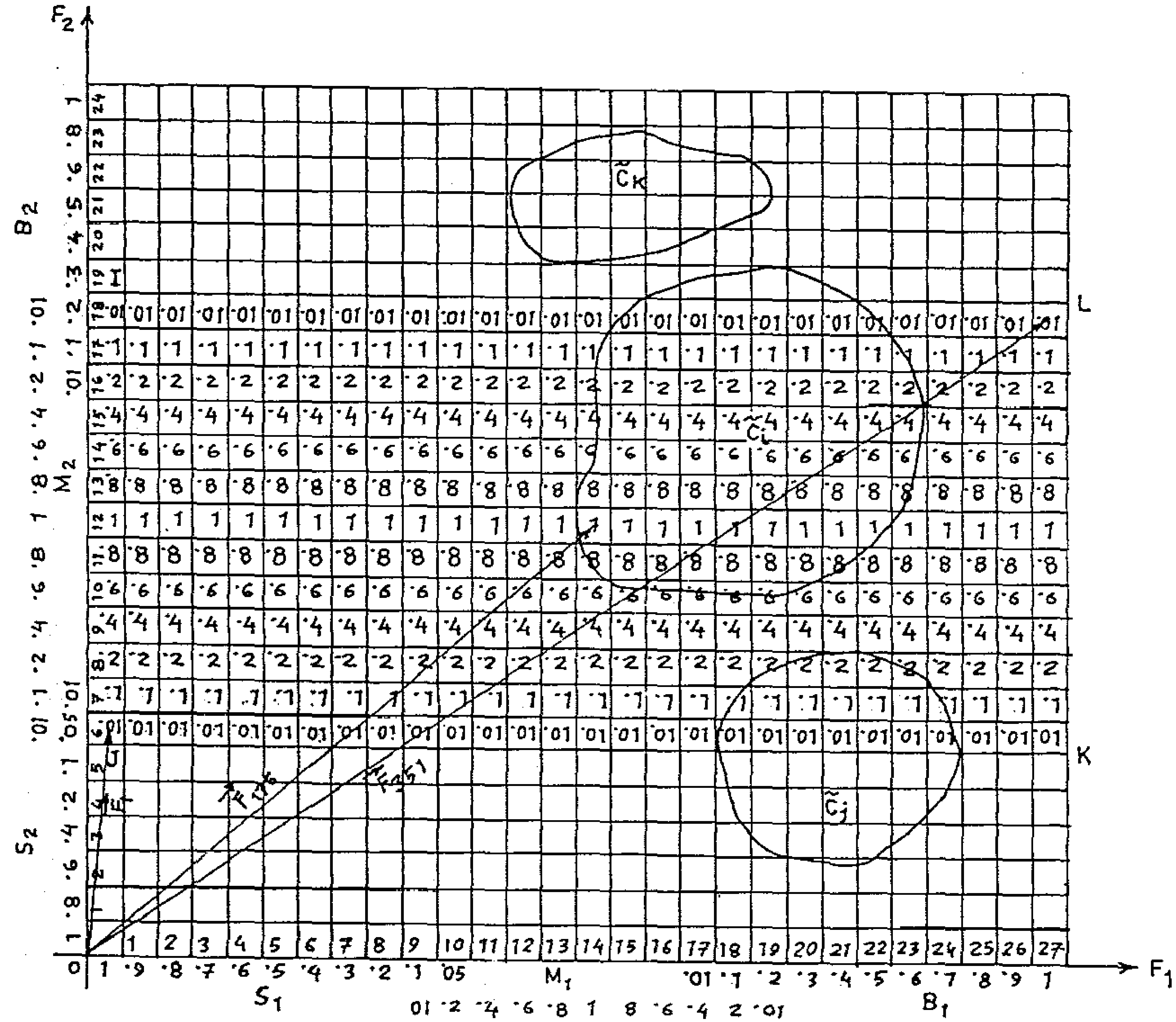


Fig.2.26(c): CYLINDRICAL EXTENSION OF M_2

KEY: The entries in the small quantized zones of the area IJKL is obtained by considering the cylindrical extension of the fuzzy set M_2.

And the position of $\vec{F_{fj}}$ is changed when

$$\vec{F_{fj}} = \begin{bmatrix} F_1 \text{ is } M_1 \\ F_2 \text{ is } S_2 \end{bmatrix} \text{etc.}$$

Now, if we try to compute the fuzzy set C which is the consequent part of the following MFI,

$$\text{if} \quad \begin{pmatrix} F_1 \text{ is } M_1 \\ F_2 \text{ is } M_2 \end{pmatrix} \rightarrow C$$

we have to consider the relative position of $\vec{F_{fj}}$, i.e. the area ABCD with respect to the defined cover $\tilde{C}_i, \tilde{C}_j$ and $\tilde{C}_k$. For simplicity of demonstration we consider partial cover, i.e. we ignore classes like g and h as indicated in Figure 2.5.

From Figure 2.26(a) it is obvious that the area ABCD has substantially occupied the class $\tilde{C}_i$. Looking at the possibility value of each small quantized zone of ABCD which are covered by the classes $\tilde{C}_i$, $\tilde{C}_j$ and $\tilde{C}_k$ we can have following four types of estimate of class membership for the class $\tilde{C}_i, \tilde{C}_j$ and $\tilde{C}_k$.

<u>For class $\tilde{C}_i$</u>

1. <u>optimistic estimate</u>: the highest possibility value of the zone covered by the class $\tilde{C}_i$. For instance, 1 indicated by $\vec{F_{72}}$ of Figure 2.26(a).

2. <u>pessimistic estimate</u>: the lowest possibility value of the zone covered by the class $\tilde{C}_i$. For instance .01.

3. <u>expected estimate</u>: average of all possibility values of all the zones covered by the class $\tilde{C}_i$. For instance .358.

4. most likely estimate: it is based on the psychological scaling [53,98] (see Section 4.3 of Chapter 4). It comes from the subjective quantification of human perception as mentioned in [108]. For instance .6.

For class $\tilde{C}_j$

1. optimistic estimate is = .2
2. pessimistic estimate is = .01
3. expected estimate is = $\frac{.2+.01+.1+.01+.01+.01}{6} = \frac{.34}{6}$ =.0566
4. most likely estimate is = .1

For class $\tilde{C}_k$

All the estimates are zero.

Thus we get four fuzzy sets for C.

$$C_{opt} = \{1/\tilde{C}_i, .2/\tilde{C}_j, 0/\tilde{C}_k\}$$
$$C_{pess} = \{.01/\tilde{C}_i, .01/\tilde{C}_j, 0/\tilde{C}_k\}$$
$$C_{averg} = \{.358/\tilde{C}_i, .0566/\tilde{C}_j, 0/\tilde{C}_k\}$$
$$C_{most} = \{.6/\tilde{C}_i, .1/\tilde{C}_j, 0/\tilde{C}_k\}$$

In all subsequent discussions when we write the fuzzy set C without mentioning anything like opt/pess/averg/most that means we go by most likely estimate (see Table 4.1 of Chapter 4). Throughout the thesis for all kind of estimates of the membership value we follow the notion of subjective quantification of human perception as stated by Zadeh [108].

In this context, we like to mention that the relation represented in the area EFGH of Figure 2.26(b) which is obtained by the cylindrical extension of the fuzzy set M_1 over the universe of the feature axis F_2 is a fuzzy set which is a fuzzy vector as per our earlier definition 2.1. Note that, in this case if we defuzzify the fuzzy vector as stated above we get a fuzzy vector which is a fuzzy set and which is not fuzzy singleton. The fuzzy set obtained after the defuzzification of the stated fuzzy vector will be having elements of equal membership values as shown in Figure 2.26(b). The meaning of this defuzzified version of a fuzzy vector which is a fuzzy set is that all the elements of the fuzzy set, as shown in Figure 2.26(b), are equally possible to occur under the defuzzified information of the fuzzy set M_1.

Next, let us consider the computation of the fuzzy set C_1 with respect to Figure 2.26(b),

$$
\begin{aligned}
C_{1_{opt}} &= \{1/\tilde{C}_i, .2/\tilde{C}_j, 1/\tilde{C}_k\} \\
C_{1_{pess}} &= \{.01/\tilde{C}_i, .01/\tilde{C}_j, .01/\tilde{C}_k\} \\
C_{1_{averg}} &= \{.469/\tilde{C}_i, .105/\tilde{C}_j, .543/\tilde{C}_k\} \\
C_{1_{most}} &= \{.7/\tilde{C}_i, .2/\tilde{C}_j, 1/\tilde{C}_k\}
\end{aligned}
$$

Computation of the fuzzy set C_2 with respect to Figure 2.26(c) is as follows;

$$
\begin{aligned}
C_{2_{opt}} &= \{1/\tilde{C}_i, .2/\tilde{C}_j, 0/\tilde{C}_k\} \\
C_{2_{pess}} &= \{.01/\tilde{C}_i, .01/\tilde{C}_j, 0/\tilde{C}_k\} \\
C_{2_{averg}} &= \{.508/\tilde{C}_i, .086/\tilde{C}_j, 0/\tilde{C}_k\} \\
C_{2_{most}} &= \{.9/\tilde{C}_i, .2/\tilde{C}_j, 0/\tilde{C}_k\}
\end{aligned}
$$

Note that, from now onwards whenever we will mention the fuzzy sets C_1 and C_2, we will assume the fuzzy sets C_1 and C_2 are obtained from the most likely estimation of the membership values of different classes of patterns. ♡

Example 2.3: Let us consider Figure 2.26(a). For simplicity of demonstration, we do not consider the total cover of the pattern space, that means we ignore classes like 'g' and 'h' indicated earlier in Figure 2.5.

Now,

$$\text{if} \begin{pmatrix} F_1 \text{ is } M_1 \\ F_2 \text{ is } M_2 \end{pmatrix} \rightarrow C = \{.6/\tilde{C}_i, .1/\tilde{C}_j, 0/\tilde{C}_k;\}$$

where the fuzzy set C is obtained from the position of the fuzzy set M_1 and M_2 on F_1 and F_2 respectively; that means from the relative position of the fuzzy feature vector / pattern vector $\vec{F_{fj}}$ (i.e. the relative position of the area ABCD) with respect to the defined cover of the pattern space (here we consider partial cover by considering classes $\tilde{C}_i, \tilde{C}_j$ and $\tilde{C}_k$ and ignoring g and h). Whereas,

$$\text{if } F_1 \text{ is } M_1 \rightarrow C_1 = \{.7/\tilde{C}_i, .2/\tilde{C}_j, 1/\tilde{C}_k\}$$

where the fuzzy set C_1 is obtained from the cylindrical extension of M_1 over the universe of the feature axis F_2 (i.e. from the position of the area EFGH on the pattern space of Figure 2.26(b)) and the well defined cover of the pattern space. And

$$\text{if } F_2 \text{ is } M_2 \rightarrow C_2 = \{.9/\tilde{C}_i, .2/\tilde{C}_j, 0/\tilde{C}_k\}$$

where the fuzzy set C_2 is obtained from the cylindrical extension of M_2 over the universe of the feature axis F_1 (i.e. from the position of the area IJKL on the pattern space of Figure 2.26(c)) and the well defined cover of the pattern space.

Therefore,

$$\hat{C} = C_1 \bigcap C_2 = \{.7/\tilde{C}_i, .2/\tilde{C}_j, 0/\tilde{C}_k\}$$

which is different from C but the defuzzy information provided by $\hat{C}$ and C is same, i.e. the class $\tilde{C}_i$ (we go by selecting class having highest membership value; in case of tie situation, we break the tie by taking arbitrary decision). ♡

Example 2.4:

Part I

With respect to the data of Figure 2.26(a), let us consider a MFI of the following form:

$$\text{if} \begin{pmatrix} F_1 \text{ is } M_1 \\ F_2 \text{ is } M_2 \end{pmatrix} \rightarrow C.$$

If we go by interpretation 2.14(a) then we can write the above MFI in the following one dimensional **If Then** form:

$$\text{if } F_1 \text{ is } M_1 \text{ and } F_2 \text{ is } M_2 \rightarrow C.$$

Now, the antecedent clauses of the above **If Then** rule form the following relation which is a fuzzy set (see Definition 2.1)

$$\{.01/\vec{F}_1 + ... + 1/\vec{F}_{72} + ... + .01/\vec{F}_{143}\}.$$

Whenever we consider interpretation (2.14(a)) for reasoning (fuzzy reasoning / neuro fuzzy reasoning) for classification the above mentioned fuzzy set acts as an input to the system at the learning stage. Thus, we inject information of a set of vectors $\{\vec{F}_1, ..., \vec{F}_{72}, ..., \vec{F}_{143}\}$ in terms of its possibilities to the system. At the learning stage the outputs of the system are equal to the possibility of each class of the above mentioned fuzzy set C.

Part II

With respect to the data of Figure 2.26(b) and Figure 2.26(c), let us consider DFIs of the following form:

$$\begin{array}{lll} \text{if } F_1 & \text{is } M_1 & \rightarrow C_1 \\ & \text{and} & \\ \text{if } F_2 & \text{is } M_2 & \rightarrow C_2 \end{array}$$

If we go by interpretation (2.16) and consider reasoning (fuzzy reasoning / neuro fuzzy reasoning) for classification then inputs of the system would be of the following form:

(a) the first input $\{.01/\vec{F_1} + ... + 1/\vec{F_{136}} + ... + .01/\vec{F_{275}}\}$ which implies the consequence C_1

(b) the second input $\{.01/\vec{F_1} + ... + 1/\vec{F_{176}} + ... + .01/\vec{F_{351}}\}$ which implies the consequence C_2.

The above inputs of (a) and (b) are obtained by considering the cylindrical extension of M_1 and M_2 over the universes of F_2 and F_1 respectively. Similarly the consequences C_1 and C_2 are obtained by considering the cylindrical extension of M_1 and the well defined cover of the pattern space and the cylindrical extension of M_2 and the well defined cover of the pattern space respectively. As the impact of the cylindrical extensions of M_1 and M_2 are reflected on the fuzzy set C_1 and C_2 respectively we may consider the fuzzy set M_1 instead of the input (a) and the fuzzy set M_2 instead of the input (b) as the two independent inputs to the system at the learning stage. Note that, the fuzzy set M_1 is nothing but the projection of the relation of Figure 2.26(b) over the universe of the feature axis F_1 and the fuzzy set M_2 is the projection of the relation of Figure

2.26(c) over the universe of the feature axis F_2. Thus, we get a reduced form of inputs i.e. M_1 and M_2 which imply the fuzzy sets C_1 and C_2 respectively. The fuzzy sets C_1 and C_2 are two independent outputs of the system (at the learning stage) which will be combined by an intersection operator to obtain the fuzzy set $\hat{C} = C_1 \bigcap C_2$ (see Figure 4.1 of Chapter 4). ♡

Thus, according to the newly proposed model (see Equation(2.16)) a suitable representation of MFI by DFIs and the notion of fuzzy feature vector / pattern vector $\vec{F_{fj}}$ are possible. In the context of pattern classification such meaningful representation of MFI is not possible by the conventional model (see Equation(2.15)) of Tsukamoto [100]. In the later chapters, we demonstrate that the notion of fuzzy pattern vector / feature vector and the newly modified MFR (i.e. Equation (2.16)) are very effective for pattern classification and / or occluded object recognition.

2.10 Extension of Fuzzy Reasoning for classification of pattern vector with missing component

Pattern classification systems are designed, as a rule, for the idealistic situation where all the components of pattern vectors / feature vectors are assumed to be available. In practical applications, however, we are often faced with the problem of classifying incomplete pattern vectors / feature vectors. This problem may arise, for instance, as a direct consequence of a fault in the sensor or preprocessing stages of the pattern classification system, failure of communication channels, temporary failure of some communication channels and subsequent recovery of part of the failured channels etc. Since conventional classifiers are unable to handle such situations, a pattern with missing components must be rejected although the information conveyed by it is often sufficient to make an acceptable decision.

The problem of classification of incomplete pattern vectors received considerable attention in the past. Urbakh [101] considered the effect of missing information on the reliability of the discriminant function classifier. He showed that each combination of missing components defines an unsafe region around the class separating surface and suggested that the complete space discriminant function be used only if an incomplete pattern vector lies outside the appropriate threshold region. Batchelor and Hand [2] proposed the use of a special type of the orthogonal function probability density function (p.d.f.) estimator which allows for simple integration of the estimate into any subspace of the pattern space. Hand [26] proposed another approach to incomplete data classification based on kernel approximation of p.d.f.

But, in the fuzzy reasoning set up, no such treatment is existing. Hence, in this section, we propose a new model for fuzzy reasoning which is different from the models stated in section 2.1, 2.2 and 2.3 and which is based on the conventional interpretation (i.e. the interpretation indicated by equation (2.14(a))) of MFI of section 2.9 for classification of fuzzy feature vector / pattern vector (see Figure 2.26(a)) where some of the components of the fuzzy feature vector / pattern vector $\vec{F_{fj}}$ are not available at the time of generating fuzzy **If Then** rules, but are partly recovered / available at the time of classification (i.e. at any subsequent stage). This is a very complex situation which we may have to face at any on-line classification problem. To tackle such problem of pattern classification, we may, by default, assume an average value (see Table 2.4, Table 2.6, Table 2.8) of each missing component (i.e. each missing feature) of the fuzzy feature vector / pattern vector (see Figure 2.26(a)) based on our tentative belief acquired from the past records of the features. Such average value can be linguistically classified on the feature axis (i.e. small, big etc.) and can be quantified by appropriate membership function. Therefore, for classification under missing component, we may write default fuzzy rule as given by equation (2.17). At the time of classification of test feature vector / pattern vector, some of the earlier missing component may be recovered / available.

Under such test situation, we may write down a premise as given by equation (2.18). Thus we get, a default rule

$$\left.\begin{array}{l} \text{Premise p: if } X_1 \text{ is } A_1 \text{ and } X_2 \text{ is } A_2 \text{ and } \ldots\ X_n \text{ is } A_n \\ \text{and it is consistent to believe that} \\ \\ \overbrace{Y_1 \text{ is } B_1 \text{ and } Y_2 \text{ is } B_2 \text{ and } \ldots Y_m \text{ is } B_m}^{\text{default values based on tentative belief}} \\ \text{then } Z \text{ is } C \end{array}\right\} \qquad (2.17)$$

where the variables X_i takes their values from the universes U_i; i=1,2, ...,n; Y_i takes their values from the universes V_i; i=1,2,...,m; and the variable Z takes its values from the universe of discourse W respectively. Also A_i is a fuzzy subset of U_i; i=1,2,...,n; B_i is a fuzzy subset of V_i; i=1,2,...,m; and C is a fuzzy subset of W. And the set of facts

$$\left.\begin{array}{rl} \text{Premise q:} & X_1 \text{ is } A'_1 \\ & X_2 \text{ is } A'_2 \\ & \vdots \\ & X_n \text{ is } A'_n \\ & \left.\begin{array}{l} Y_1 \text{ is } B'_1 \\ Y_2 \text{ is } B'_2 \\ \vdots \\ Y_k \text{ is } B'_k \end{array}\right\} \text{partly recovered components} \\ & k \leq m. \end{array}\right\} \qquad (2.18)$$

Here A'_i is a fuzzy subset of U_i; i=1,2,...,n and B'_i is a fuzzy subset of V_i; i=1,2,...,k, $k \leq m$.

The translation of equation (2.17) may be expressed as

$p \rightarrow \pi_{(X_1,X_2,\ldots,X_n,Y_1,Y_2,\ldots,Y_m,Z)} = R$ (say)

which is an (n+m+1) - dimensional relational matrix defined over the universe $U_1 \times$

$U_2 \times ... \times U_n \times V_1 \times V_2 \times ... \times V_m \times W$

where,

$\mu_R(u_1, u_2, ..., u_n, v_1, v_2, ..., v_m, w)$

$= \min \{ \mu_{A_1}(u_1), \mu_{A_2}(u_2), ..., \mu_{A_n}(u_n), \mu_{B_1}(v_1), \mu_{B_2}(v_2), ..., \mu_{B_m}(v_m), \mu_C(w) \}$

And the translation of equation (2.18) may be expressed as

$q \rightarrow \pi_{(X_1, X_2, ..., X_n, Y_1, Y_2, ..., Y_k)} = S$ (say)

which is an (n+k) - dimensional relational matrix defined over the universe $U_1 \times U_2 \times .. \times U_n \times V_1 \times V_2 \times .. \times V_k$ where

$\mu_S(u_1, u_2, .., u_n, v_1, v_2, .., v_k)$

$= \min \{ \mu_{A'}(u_1), \mu_{A'}(u_2), ..., \mu_{A'}(u_n), \mu_{B'}(v_1), \mu_{B'}(v_2), ..., \mu_{B'}(v_k) \}$

Now, keeping in mind the most optimistic values for the default variables about which the expert has no information we form the cylindrical extension of S over $V_{k+1} \times V_{k+2} \times ... \times V_m \times W$.

Let,

$\bar{S} = S \times V_{k+1} \times V_{k+2} \times .. \times V_m \times W$ then

$\mu_{\bar{S}}(u_1, u_2, ..., u_n, v_1, v_2, ..., v_m, w) = \mu_S(u_1, u_2, ..., u_n, v_1, v_2, ..., v_k)$

And the particularization of R by S, denoted by T, will be given by $T = R \bigcap \bar{S}$ and is such that

$\mu_T(u_1, u_2, ..., u_n, v_1, v_2, ..., v_m, w) = \{\mu_R(u_1, u_2, ..., u_n, v_1, v_2, ..., v_m, w)$ $\wedge \mu_S(u_1, u_2, ..., u_n, v_1, v_2, ..., v_k)\}$ where $\wedge$ stands for the well-known "min" operator. The required default inference can then be given by projecting T on W. It, therefore, follows that the desired inference will be

r $\leftarrow$ Z is $C' = Proj_W T$

where

$\mu_{C'}(w) = Sup_{(u_1, u_2, ..., u_n, v_1, v_2, ..., v_m)} \mu_T(u_1, ..., u_n, v_1, ..., v_m, w)$

$= \text{Sup} \{ \mu_R \wedge \mu_{\bar{S}} \}$

$= Sup_{(u_1, ..., u_n, v_1, ..., v_m)} \{\mu_R \wedge \mu_S(u_1, ..., u_n, v_1, ..., v_k)\}.$

Thus in this section, we develop a new fuzzy reasoning model for pattern classification under the condition that some of the components / elements of the fuzzy feature vector / pattern vector are missing. This new proposition simply gives a theoretical basis to tackle such kind of problem. And this theoretical basis can be used in future to develop a complete nonmonotonic reasoning system where revision of the tentative belief incorporated in the default fuzzy rule (see Equation (2.17)) will occur if we come across any contrary evidences [88].

2.10.1 Numerical Examples

To demonstrate the effectiveness of the newly proposed model, we consider a simple example, where the variables involved in the premises range over finite sets or can be approximated by variables ranging over such sets [108]. Let us consider a default rule
p: if X is A and it is consistent to believe that Y is B then Z is C
together with the fact that
q: X is A'.
The translation of p may be expressed as
p $\rightarrow \pi_{(X,Y,Z)} = R$ (say) ,
where

$$R = A \bigcap B \bigcap C$$

such that

$$\mu_R(u,v,w) = min\{\mu_A(u), \mu_B(v), \mu_C(w)\}.$$

Let, X,Y,Z range respectively over U,V,W given by

$$U = u_1 + u_2 + u_3 + u_4$$
$$V = v_1 + v_2 + v_3 + v_4$$
$$W = w_1 + w_2 + w_3$$

Now, in p and q, A and A' are fuzzy subsets of the universe U. B and C are fuzzy subsets of V and W respectively. Let them be

$$A = 1 / u_1 + .7 / u_2 + .4 / u_3 + .1 / u_4$$
$$B = .5 / v_1 + .8 / v_2 + 1 / v_3 + .6 / v_4$$
$$C = .25 / w_1 + .65 / w_2 + 1 / w_3$$

and $A' =$ very $A = 1 / u_1 + .49 / u_2 + .16 / u_3 + .01 / u_4$.

Then R= $.1/[u_4v_1w_1 + u_4v_1w_2 + u_4v_1w_3 + u_4v_2w_1 + u_4v_2w_2 + u_4v_2w_3 + u_4v_3w_1 + u_4v_3w_2 + u_4v_3w_3 + u_4v_4w_1 + u_4v_4w_2 + u_4v_4w_3] + .25/[u_1v_1w_1 + u_1v_2w_1 + u_1v_3w_1 + u_1v_4w_1 + u_2v_1w_1 + u_2v_2w_1 + u_2v_3w_1 + u_2v_4w_1 + u_3v_1w_1 + u_3v_2w_1 + u_3v_3w_1 + u_3v_4w_1] + .4/[u_3v_1w_2 + u_3v_1w_3 + u_3v_2w_2 + u_3v_2w_3 + u_3v_3w_2 + u_3v_3w_3 + u_3v_4w_2 + u_3v_4w_3] + .5/[u_1v_1w_2 + u_1v_1w_3 + u_2v_1w_2 + u_2v_1w_3] + .6/[u_1v_4w_2 + u_1v_4w_3 + u_2v_4w_2 + u_2v_4w_3] + .65/[u_1v_2w_2 + u_1v_3w_2 + u_2v_2w_2 + u_2v_3w_2] + .7/[u_2v_2w_3 + u_2v_3w_3] + .8/u_1v_2w_3 + 1/u_1v_3w_3$.

And

$$\bar{S} = S \times V \times W$$

will be given by

$\bar{S} = .01/[u_4v_1w_1 + u_4v_1w_2 + u_4v_1w_3 + u_4v_2w_1 + u_4v_2w_2 + u_4v_2w_3 + u_4v_3w_1 + u_4v_3w_2 + u_4v_3w_3 + u_4v_4w_1 + u_4v_4w_2 + u_4v_4w_3] + .16/[u_3v_1w_1 + u_3v_1w_2 + u_3v_1w_3 + u_3v_2w_1 + u_3v_2w_2 +$

$u_3v_2w_3 + u_3v_3w_1 + u_3v_2w_2 + u_3v_3w_3 + u_3v_4w_1 + u_3v_4w_2 + u_3v_4w_3] + .49/[u_2v_1w_1 + u_2v_1w_2 + u_2v_1w_3 + u_2v_2w_1 + u_2v_2w_2 + u_2v_2w_3 + u_2v_3w_1 + u_2v_3w_2 + u_2v_3w_3 + u_2v_4w_1 + u_2v_4w_2 + u_2v_4w_3] + 1/[u_1v_1w_1 + u_1v_1w_2 + u_1v_1w_3 + u_1v_2w_1 + u_1v_2w_2 + u_1v_2w_3 + u_1v_3w_1 + u_1v_3w_2 + u_1v_3w_3 + u_1v_4w_1 + u_1v_4w_2 + u_1v_4w_3]$.

And then $T = R \bigcup \bar{S}$

will be given by

$$T = .01/[u_4v_1w_1 + u_4v_1w_2 + u_4v_1w_3 + u_4v_2w_1 + u_4v_2w_2 + u_4v_2w_3 + u_4v_3w_1 + u_4v_3w_2 + u_4v_3w_3 + u_4v_4w_1 + u_4v_4w_2 + u_4v_4w_3] + .16/[u_3v_1w_1 + u_3v_1w_2 + u_3v_1w_3 + u_3v_2w_1 + u_3v_2w_2 + u_3v_2w_3 + u_3v_3w_1 + u_3v_3w_2 + u_3v_3w_3 + u_3v_4w_1 + u_3v_4w_2 + u_3v_4w_3] + .25/[u_1v_1w_1 + u_1v_2w_1 + u_1v_3w_1 + u_1v_4w_1 + u_2v_1w_1 + u_2v_2w_1 + u_2v_3w_1 + u_2v_4w_1] + .49/[u_2v_1w_2 + u_2v_1w_3 + u_2v_2w_2 + u_2v_2w_3 + u_2v_3w_2 + u_2v_3w_3 + u_2v_4w_2 + u_2v_4w_3] + .5/[u_1v_1w_2 + u_1v_1w_3] + .6/[u_1v_4w_2 + u_1v_4w_3] + .65/[u_1v_2w_2 + u_1v_3w_2] + .8/u_1v_2w_3 + 1/u_1v_3w_3 \ .$$

Now, let, $Proj_W T = C'$ (say). Then

$$\mu_{C'}(w) = Sup_{(u,v)}\mu_T(u, v, w)$$

which at once gives,

$$C' = .25/w_1 + .65/w_2 + 1/w_3 \ .$$

And the required inference will be

Z is C'. ♡

2.10.2 Application of the new model

Now, we apply the newly proposed model for pattern classification on the Telugu data of Figure 2.19. Here, we intentionally suppress the feature values of the second formant frequency (i.e. F_2) and generate a set of fuzzy default rules (i.e. like equation (2.17)) based on the average values of F_2 as given in Table 2.4. For testing, we first pick up the nonfuzzy feature F_1 from Figure 2.19 and fuzzify it as stated earlier. The fuzzy feature F_1 becomes a premise like equation (2.18). Then using the model of section 2.10, we classify the test features. The results of classification in terms of recognition score and a brief comparative study is given in Table 2.12.

Table 2.12: Recognition score on Telugu data of Figure 2.18 where feature values of F_2 are suppressed

	Urbakh method	Batchelor and Hand method	Hand method	present method	
				Hard Partitioning	Fuzzy Partitioning
Recognition score (in %)	62	60.1	65	60	77.5

The primary results obtained under the missing information are quite acceptable. Test results on Assamese vowels and Bengali vowels are similar and hence are not listed here. Detail implementation of the nonmonotonic reasoning based on default fuzzy rules and revision of tentative belief under the presence of contrary evidences are the part of a future research. We need to apply our newly proposed scheme to several other practical pattern classification problems where such missing informations are existing. But such detail applications are beyond the scope of the present thesis.

Chapter 3

Fusion technology based on the conventional interpretation of MFI

In the recent years fuzzy inference and neural networks have independently received a tremendous attention from the field of applied sciences and engineering. They share the common property that they promise to provide solutions for flexible information processing. Hence, attempt has begun to generate fusion technology which will compensate the inherent demerits of one field by the merits of another. Recent results on such fusion technology are reported in [28,92,93]. But these results show an implementation of the Sugeno and Kang model [90] of fuzzy reasoning [28,92,93] on backpropagation neural network. The aim of this present chapter is to generate a fusion technology [79] based on the conventional interpretation of MFI (reasoning based on Equation (2.14(a)) of Chapter 2).

3.1 Neural Network Driven Fuzzy Reasoning (NDF)

In the existing approach to NDF [28,92,93] the fuzzy inference rules are represented as follows:

$$\left.\begin{array}{ll} R_s: & \text{IF } X = (x_1, x_2, x_3, \ldots, x_n) \text{ is } A_s \\ & \text{THEN } y_s = NN_s(x_1, x_2, x_3, \ldots, x_m), \\ & s = 1, 2, 3, \ldots, r; \quad m \leq n \end{array}\right\} \quad (3.1)$$

where r represents the number of inference rules, A_s represents a fuzzy set of the antecedent part of each inference rule and $NN_s(x_1, x_2, x_3, \ldots, x_m)$ denotes a structure of model function that is characterized by an M-layer backpropagation neural network for a given input $X = (x_1, x_2, x_3, \ldots, x_n)$ and output y. The degree of attribution of input $X = (x_1, x_2, x_3, \ldots, x_n)$ to the antecedent part of the s^{th} inference rule is derived from the membership value of fuzzy set A_s to the input X. The membership values of the fuzzy set A_s is determined by the neural network NN_{mem}. Fig. 3.1 depicts the schematic representation of NDF which is essentially an implementation of Sugeno and Kang model [90] for fuzzy reasoning in neural network. But the consequent part of the Sugeno and Kang model is not suitable for representing the possibility of occurrence of different classes of patterns in the pattern space.

Hence, in the following sections we will consider the conventional interpretation of MFI (reasoning based on Equation (2.14(a)) of Chapter 2) which reduces to fuzzy reasoning [61,62,108] stated in Section 2.1, 2.2 and 2.3 of Chapter 2 (see the discussion of paragraph 3 of Section 2.9) and try to implement it in a backpropagation neural network. Also note that to deal with interpretation (2.14(a)) of a MFI means to deal with a fuzzy vector as shown in Figure 2.26(a) and a consequence C (see the discussion of paragraph 4 of Section 2.9). In the fuzzy reasoning model of Section 2.3 (i.e. Equation (2.11) of Chapter 2) we translate each rule i into a relational matrix $R_i(u, v, w)$

where,

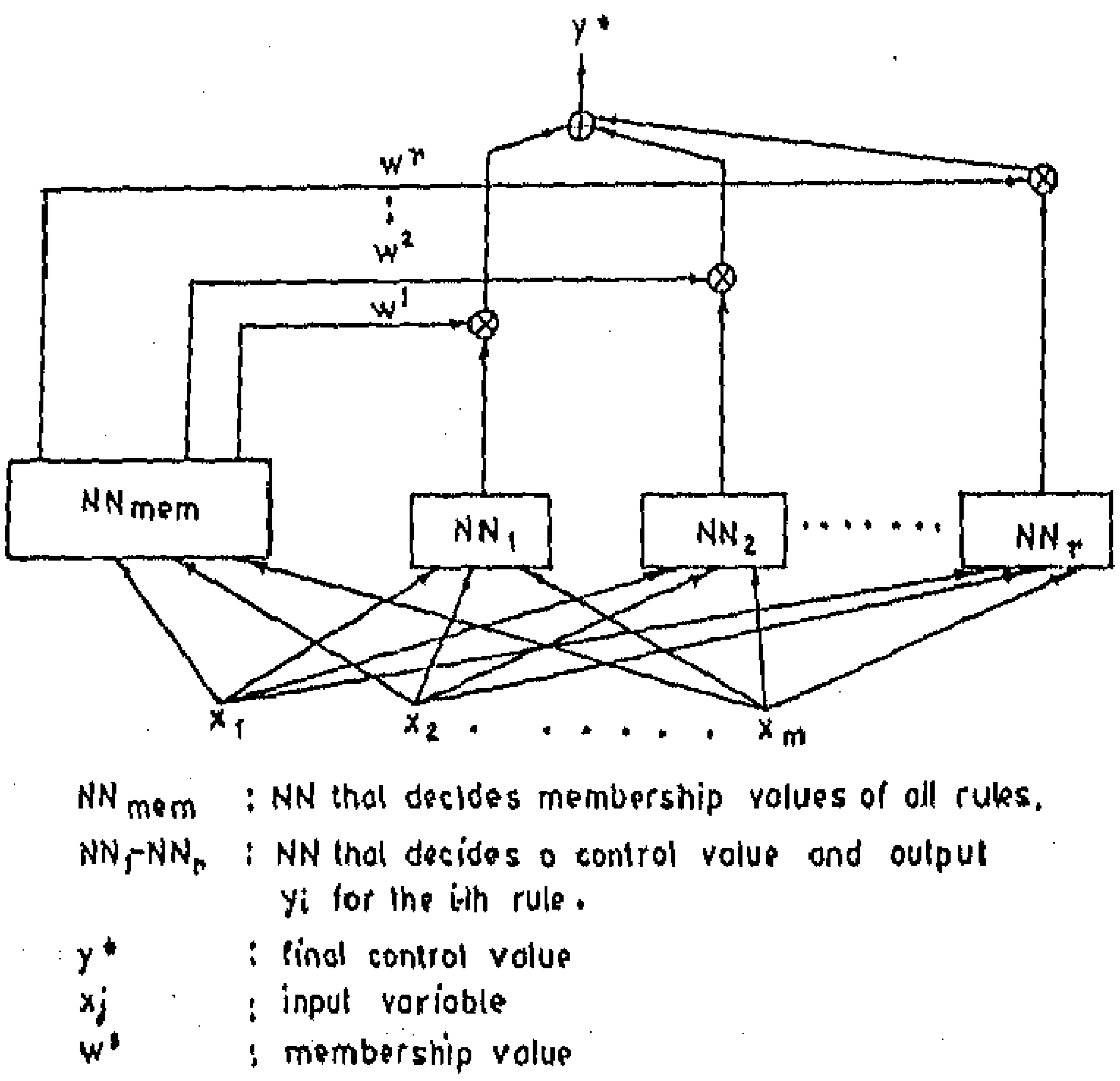


Fig. 3.1: Block Diagram of NN - driven Fuzzy Reasoning

$$\left.\begin{aligned} R_i(u,v,w) &= \mu_{A_i}(u) \wedge \mu_{B_i}(v) \wedge \mu_{C_i}(w) && \text{(a)} \\ \text{or } R_i(u,v,w) &= \mu_{A_i}(u) \bullet \mu_{B_i}(v) \wedge \mu_{C_i}(w) && \text{(b)} \end{aligned}\right\} \quad (3.2)$$

(considering Mamdani's min. operator).

The expression $\mu_{A_i}(u) \wedge \mu_{B_i}(v)$ of equation (3.2(a)) represents a fuzzy relation of the antecedent part of each rule i. In this particular situation we have shown this relation using "min" operator. But depending upon the need of the problem we may use any other suitable operators like algebraic product (equation (3.2(b))). In our process of implementation of the conventional interpretation of MFI into a backpropagation neural network we essentially replace the above said relational matrix $R_i(u,v,w)$ of equation (3.2) by the weights of the links of the neural network where the weights are learned by generalized delta rule. Thus we replace the logical approach of reasoning by learning scheme of neural network and generate fusion of the first kind as stated in Table 1.1

of Chapter 1. But before we give detail discussion on the fusion technology based on the conventional interpretation to MFI, we briefly discuss the essential concepts of backpropagation neural network.

3.2 Brief Review on Backpropagation Neural Network

One of the most interesting developments during the early days of pattern recognition was the perceptron, the idea that a network of elemental processors arrayed in a manner reminiscent of biological neural nets might be able to learn how to recognize and classify patterns. Correspondingly, one of the severe setbacks of early pattern recognition was the realization that simple linear networks were inadequate for that purpose, and that nonlinear nets lacked suitable learning algorithms. Nevertheless, the α - perceptron and the layered machine provided a solid conceptual base for further work. However, the generalized delta rule [69] constituted one practicable way of implementing a Perceptron like system and other generalization of that approach are being explored. This section is devoted to a brief discussion of the approach mentioned above to learning nonlinear discriminants using backpropagation network.

Let us now consider the network given in Figure 3.2. In general, such a net is made up of sets of nodes arranged in layers. The outputs of nodes in one layer are transmitted to nodes in another layer through links that amplify or attenuate such outputs through weighting factors. Except for the input layer nodes, the net input to each node is the sum of the weighted outputs of the nodes in the prior layer. Each node is activated in accordance with the input to the node, the activation function of the node and the bias of the node. The net input to a node in layer j is[69]

$$net_j = \sum w_{ji} o_i.$$

The output of node j is

$$o_j = f(net_j)$$

where f is the activation function.

Now there are several activation functions like threshold signal functions, hyperbolic tangent signal function, threshold linear signal function, linear signal function, exponential signal function and so on. Among them, the most popular is the sigmoidal signal function. For a sigmoidal activation function, we have

$$o_j = \frac{1}{1 + e^{-(net_j + \theta_j)/\theta_0}}$$

where θ_j is the bias. The effect of θ_0 is to modify the shape of the sigmoid. In our implementation software we treat "θ_0" as a parameter and we may vary "θ_0" when the decision boundaries are not well separated.

Continuing our description of the computational processes, we have for the nodes in layer k, the input

$$net_k = \sum w_{kj} o_j$$

and the corresponding outputs

$$o_k = f(net_k).$$

In the training phase, we present the patterns one by one as input and ask the net to adjust the set of weights as well as the biases of the nodes so that the desired output is obtained at the output nodes. Once this type of adjustment has been accomplished for a particular pair of input and output, we present the next pair and so on. In general, the outputs $\{o_{pk}\}$ will not be the same as the target values $\{t_{pk}\}$. For each pattern, the error to be minimized is

$$E_p = \frac{1}{2} \sum (t_{pk} - o_{pk})^2$$

and the average system error is

$$E = \frac{1}{2P} \sum \sum (t_{pk} - o_{pk})^2$$

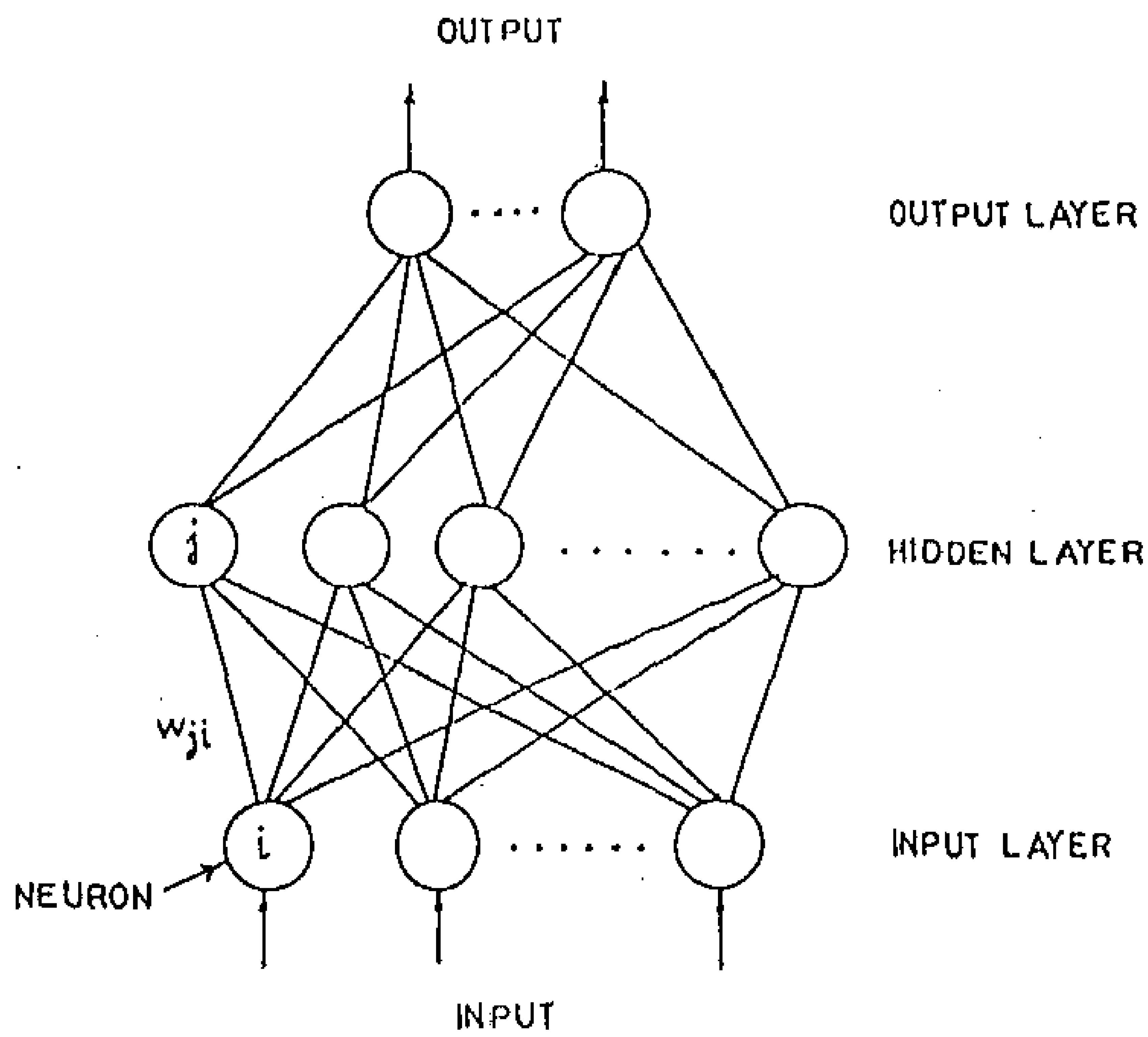


Fig. 3.2: A schematic depiction of a nonlinear feedforward neural network with one hidden layer

where P is the number of patterns used for training the net. A gradient search for minimum system error should be based on the minimization of the above expression.

In the generalized delta rule, the weights are updated as,

$$\Delta w_{kj} = -\eta \partial E / \partial w_{kj}$$

where η is the learning coefficient.

$$\frac{\partial net_k}{\partial w_{kj}} = o_j$$

Therefore,

$$\Delta w_{kj} = \eta \delta_k o_j$$

$$\text{where } \delta_k = -\frac{\partial E}{\partial net_k}.$$

Proceeding this way, we can find out,

$$\Delta w_{ji} = \eta \delta_j o_i$$

where $\delta_j = -\frac{\delta E}{\delta net_j}$. That is, starting at the highest layer - the output layer - we can propagate the errors backward to lower layers. Note that, θ_j s are learned in the same manner as are the other weights. It may be mentioned at this point, that the generalized delta rule used in the learning includes a momentum term and is stated as follows :

$$\Delta w_{ij}(n+1) = \eta(\delta_j o_i) + \alpha \Delta w_{ij}(n)$$

where the quantity $(n+1)$ is used to indicate the $(n+1)^{th}$ step, and α is a proportionality constant. The second term in the above expression is used to specify that the change in w_{ij} at $(n+1)^{th}$ step should be somewhat similar to the change undertaken at the n^{th} step.

Now, during training each pattern of the training set is used in succession to clamp the input and output layers of the network. A sequence of forward and backward passes constitutes a cycle and such a cycle through the entire training set is called a sweep. After a number of sweeps it is supposed that the network has learnt the relationship between the input and output vectors of the training samples.

In the testing phase, the neural net is expected to utilize the information stored to classify the input patterns properly. It should be noted that, the number of input nodes in the proposed scheme equals the number of elements of the relation generated by the "min" operation/algebraic product operation between the antecedent clauses of the **If Then** rules (see Figure 3.3(b)). The selection of optimal number of hidden layers and units in each hidden layer is still a research problem. However, in the present cases,what we have chosen will be discussed in the next section. The number of units in the output layer corresponds to the number of output classes to be classified.

MLP models using backpropagation (BKP) have been applied to the XOR problem, recognizing written text, predicting sunspots, identifying sonar targets and several other fields. In each field it provides a satisfactory performance. And hence we are motivated to apply it to realize the conventional interpretation of MFI (i.e. Equation

(2.14(a))) through backpropagation neural network.

3.3 Implementation of the conventional interpretation of MFI on neural network

Let us consider Equation (2.11) of Chapter 2 where each rule represents a law of implication. As stated earlier, each one dimensional fuzzy implication of Equation (2.11) is basically the conventional interpretation (i.e. Equation (2.14(a))) of multidimensional fuzzy implication (MFI). The input to the neural network is the fuzzy relation formed by the antecedent clauses of each rule of Equation (2.11) of Chapter 2. To deal with the relation formed by the antecedent clauses of an one dimensional implication means we are dealing with a fuzzy vector as shown in Figure 2.26(a) (see the discussion of paragraph 4 of Section 2.9). Therefore, the input to the neural network is basically a fuzzy vector. In conventional approach to pattern classification using backpropagation neural network the input to the network is a pattern vector / feature vector. By considering the interpretation of Equation (2.14(a)), we are replacing that pattern vector / feature vector by fuzzy pattern vector / feature vector. The relation formed by the antecedent clauses are represented in the form of one dimensional array whose elements are the input to neural network (see Example 3.1 and Figure 3.3(b)). The reference output of the network is the consequent clause of each rule of Equation (2.11) of Chapter 2. The consequent clause is represented by a fuzzy set which basically represents the possibility of occurrence of different classes of patterns in the pattern space. The said fuzzy set is obtained using the interpretation (2.14(a)) of Section 2.9 of Chapter 2 and the notion of fuzzy vector (see Figure 2.26(a) of Chapter 2). That means, we consider the well defined cover of the pattern space as stated in Section 2.1 of Chapter 2 and consider the fuzzy relation formed by the antecedent clause of an one dimensional fuzzy implication i.e. Equation (2.11). Such relation formed by the antecedent clause of an one dimensional fuzzy implication is located at the tip of the fuzzy vector (see area ABCD of Figure 2.26(a) of Chapter 2). The relative position of the fuzzy vector i.e. the tip of the fuzzy vector and the well defined cover of the pattern space represent the possibility of occurrence of different classes of patterns in the pattern space. This said possibility of occurrence is represented by a fuzzy set which is the consequent clause of an one dimensional fuzzy implication (see Example 2.2 of Chapter 2) and which forms the reference output of the neural network as stated above. Other features of the network are same as the conventional backpropagation neural network. Fig. 3.3(a,b) schematically represents the implementable architecture for fuzzy reasoning (Equation (2.14(a)) of MFI of Section 2.9 of Chapter 2) on back propagation neural network. The network is trained by a set of fuzzy **If Then** rules (see Equation (2.11)) using generalized delta rule [69]. Thus, depending upon the individual pattern error and the cumulative error neural net learns a weight matrix for the total set of training rules of **If Then** type. This weight matrix may be viewed as the relational matrix R of equation (3.2). Then for any given fact of the type shown in Equation (2.11) (i.e. premise n+1) the neural net produces a consequence C' of the type shown in Equation (2.11).

Example 3.1: Let, X = { x } = { 1,2,3 }, Y = { y } = { 1,2,3,4 }, Z = { z } = { 1,2, 3 }.

The fuzzy conditional statement expressing the dependence among the linguistic variables L, K and M is,

IF (L is "low") and (K is "high") THEN (M is "medium")

where "low" = 1/1 + 0.6/2 + 0.2/3 (considering X as universe of discourse of L)

"high" = 0.1/1 + 0.4/2 + 0.7/3 + 1/4 (considering Y as universe of discourse of K)

and "medium" = .5/1 + 1/2 + .5/3 (considering Z as universe of discourse of M)

Now, if we interpret the connective "and" of the said conditional statement as "min" operator (we may attach any other interpretation to the connective "and"; e.g. the algebraic product), then we may translate, using Mamdani's law of implication, the conditional statement as follows;

The antecedent clause of the conditional statement can be represented by the following relation,

Relation formed by antecedent clauses(RAC) $\equiv$

$$\left\{ \frac{.1}{(1,1)}, \frac{.4}{(1,2)}, \frac{.7}{(1,3)}, \frac{1}{(1,4)}, \frac{.1}{(2,1)}, \frac{.4}{(2,2)}, \frac{.6}{(2,3)}, \frac{.6}{(2,4)}, \frac{.1}{(3,1)}, \frac{.2}{(3,2)}, \frac{.2}{(3,3)}, \frac{.2}{(3,4)} \right\}$$

Finally, the conditional statement can be represented by the following relation;

$$\left\{ \frac{.1}{(1,1,1)}, \frac{.1}{(1,1,2)}, \frac{.1}{(1,1,3)}, \frac{.4}{(1,2,1)}, \frac{.4}{(1,2,2)}, \frac{.4}{(1,2,3)}, \frac{.5}{(1,3,1)}, \frac{.7}{(1,3,2)}, \frac{.5}{(1,3,3)}, \frac{.5}{(1,4,1)}, \frac{1}{(1,4,2)}, \frac{.5}{(1,4,3)}, \frac{.1}{(2,1,1)}, \frac{.1}{(2,1,2)}, \right.$$
$$\frac{.1}{(2,1,3)}, \frac{.4}{(2,2,1)}, \frac{.4}{(2,2,2)}, \frac{.4}{(2,2,3)}, \frac{.5}{(2,3,1)}, \frac{.6}{(2,3,2)}, \frac{.5}{(2,3,3)}, \frac{.5}{(2,4,1)}, \frac{.6}{(2,4,2)}, \frac{.5}{(2,4,3)}, \frac{.1}{(3,1,1)}, \frac{.1}{(3,1,2)}, \frac{.1}{(3,1,3)}, \frac{.2}{(3,2,1)},$$
$$\left. \frac{.2}{(3,2,2)}, \frac{.2}{(3,2,3)}, \frac{.2}{(3,3,1)}, \frac{.2}{(3,3,2)}, \frac{.2}{(3,3,3)}, \frac{.2}{(3,4,1)}, \frac{.2}{(3,4,2)}, \frac{.2}{(3,4,3)} \right\}$$

Now, if L is "medium" $= .4/1 + 1/2 + .4/3$

and K is "low" $= 1/1 + .7/2 + .4/3 + .1/4$,

we get the following relation;

$$\left\{ \frac{.4}{(1,1)}, \frac{.4}{(1,2)}, \frac{.4}{(1,3)}, \frac{.1}{(1,4)}, \frac{1}{(2,1)}, \frac{.7}{(2,2)}, \frac{.4}{(2,3)}, \frac{.1}{(2,4)}, \frac{.4}{(3,1)}, \frac{.4}{(3,2)}, \frac{.4}{(3,3)}, \frac{.1}{(3,4)} \right\}.$$

Corresponding to the above relation the induced M, using max-min composition, is given by,

$$(.4/1 + .4/2 + .4/3\ )\ \heartsuit$$

3.4 Discussion on the proposed method

To demonstrate the effectiveness of the proposed scheme we consider several classification problems. At the learning stage of classification, we partition the universe of individual feature axis by the elements of the term set as stated in Section 2.1 of Chapter 2. These elements will ultimately form the antecedent clause of an one dimensional fuzzy implication. The consequent clause of an one dimensional fuzzy implication will be a fuzzy set which is discussed in Section 3.3. Thus, we generate fuzzy **If Then** rules to train the neural network shown in Figure 3.3(a,b). Depending upon the results of cross validation (when the error of the objective function comes to a minimum value the net is tested with some known data. Depending upon the percentage of correct classification the net may be relearned with a modified set fuzzy **If Then** rules) we may add one or two additional rules for good classification scores. Once the weights and biases have been stabilized, the network will learn the relationship between the inputs (formed by the antecedent clauses of each rule) and outputs (formed by the consequent clause of each rule) and can provide a generalization mechanism. In the logical approach to fuzzy reasoning (see Section 2.1 of Chapter 2) we establish the relation between input (formed by the antecedent clauses of each rule) and output (formed by the consequent clause of each rule) by giving appropriate interpretation of implication as given in Table 2.1. Though there are many such interpretations available (see Table 2.1) but there is no guideline to select one appropriate interpretation for a specific problem of pattern classification. To overcome such difficulty, we adopt the method of learning using generalized delta rule. The outputs of the neural net represent a membership function which indicates the possibility of occurrence of different classes of patterns in the pattern space. For instance, if we have two outputs of the neural network, then, $m_1/\tilde{c}_1, m_2/\tilde{c}_2$ represents the possibility of occurrence of a pattern to class $\tilde{c}_i$ is m_i. Thus, we have the mechanism for multiple classification which is not available in conventional approach to pattern classification [97]. To determine the classification score we go by selecting the class having the highest membership function. In case of tie situation, we break the tie by taking arbitrary decision.

Once learned, for checking the classification score, we may fuzzify the nonfuzzy features of the unknown patterns (test patterns) in the following way.

Consider Figure 3.4(a,b). If any feature pair (F_1 and F_2) occurs (at any particular instance) from the range a_1^i and a_2^i then we assume that each feature is the generic

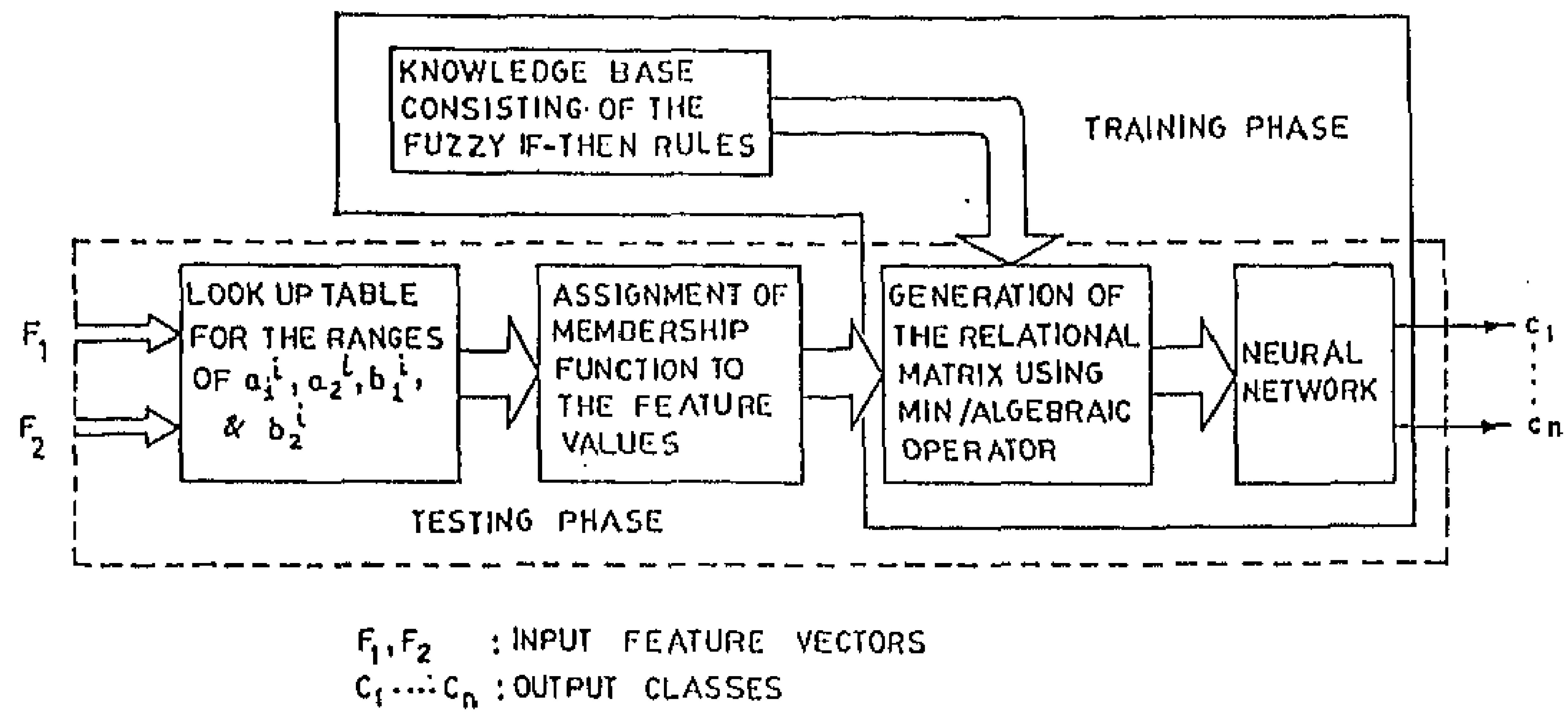


Fig. 3.3(a): Schematic representation of the proposed model

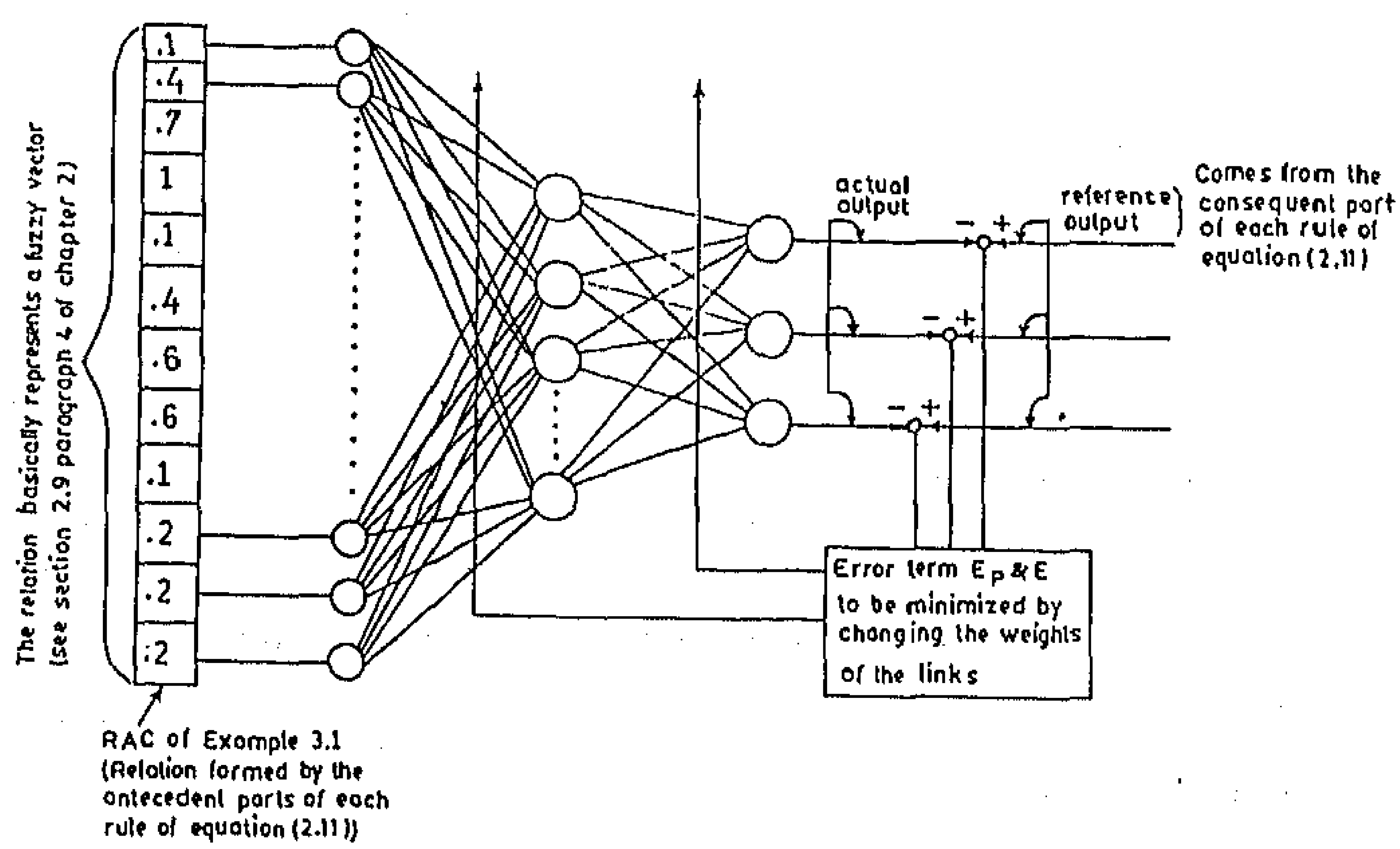


Fig. 3.3(b): Input - Output structure of the proposed Neural Network

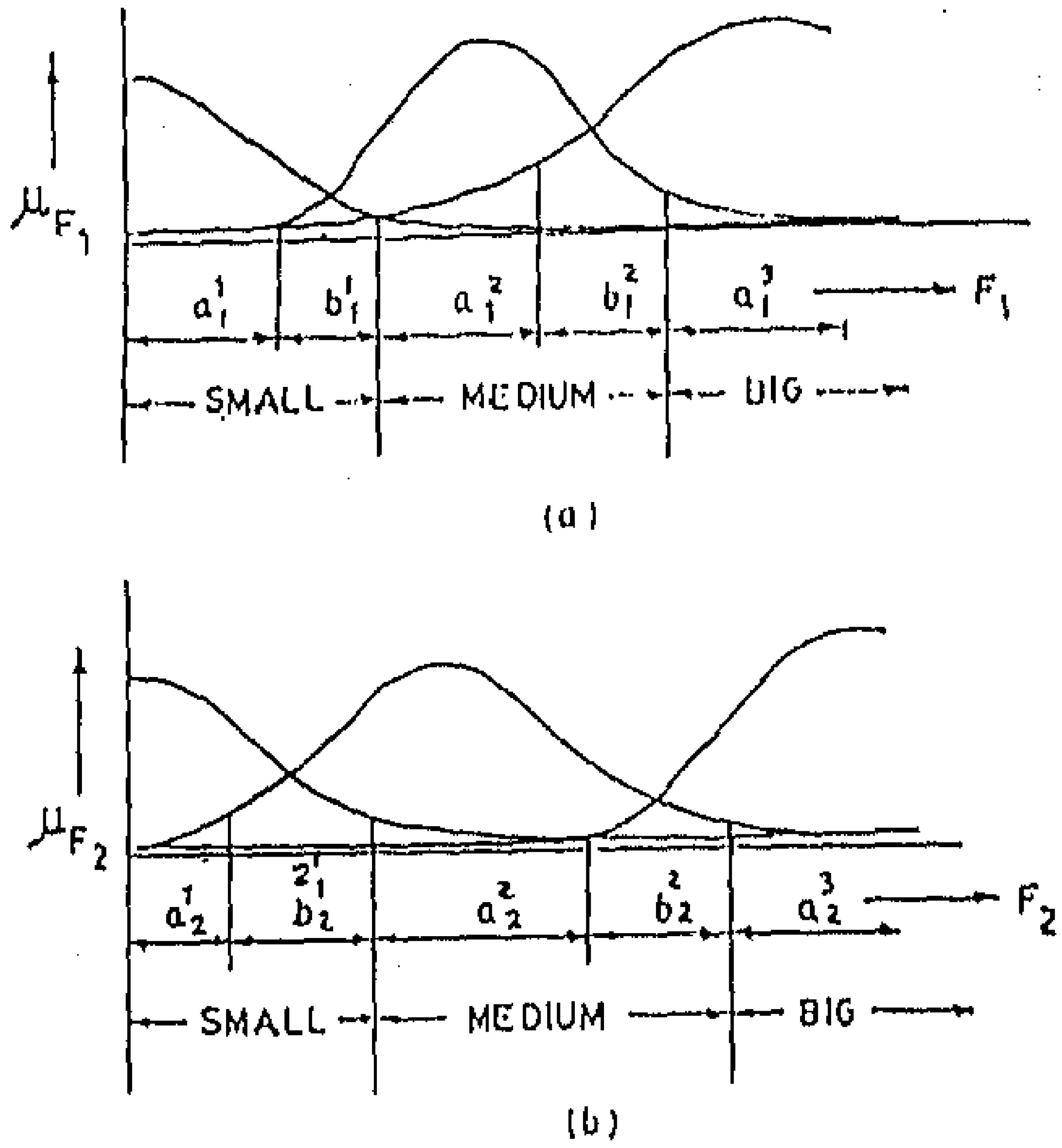


Fig. 3.4: Member functions of fuzzy sets SMALL, MEDIUM and BIG $b_1^1, b_2^1, b_1^2, b_2^2$ represent fuzziest zones

element of the original fuzzy set low / **medium** / **high**. Hence, for classification of pattern, we fuzzify them by the membership function of the corresponding fuzzy set low / **medium** / **high**. But, if the nonfuzzy feature values occur at the ranges b_1^i and b_2^i then we assume that they are the generic elements of the fuzzy sets (which are different from low / medium / high) represented by the membership function obtained through the weighted average of the membership function of the existing sets like low / **medium** / **high**. Here in Figure 3.4(a,b) for simplicity of explanation we have shown the functional definition of membership function. But, in practice, we consider the numerical definition.

Example 3.2: Suppose, any feature F_2 occurs at point 2 at any instance of time. We have to check the membership values of the fuzzy sets low / **medium** / **high** for that feature value. For our case, 0.4 is the value of membership corresponding to the fuzzy set low, 0.5 is the value corresponding to the fuzzy set **medium** and 0.01 is the value corresponding to the fuzzy set **high**. Therefore, according to our proposed fuzzification method, the fuzzy set which is different from the existing class (**low** / **medium** / **high**) corresponding to this feature value will be :

$$(0.4*1+0.5*0.01+0.01*0.01)/0+(0.4*0.7+0.5*0.01+0.01*0.01)/1+(0.4*0.4+0.5*0.5+0.01*0.01)/2+(0.4*0.01+0.5*1+0.01*0.01)/3+(0.4*0.01+0.5*0.5+0.01*0.4)/4+(0.4*0.01+0.5*0.01+0.01*0.9)/5+(0.4*0.01+0.5*0.01+0.01*1)/6=$$
$$0.4501/0+0.3301/1+0.4101/2+0.5041/3+0.258/4+0.018/5+0.019/6$$

i.e., we consider the weighted sum of membership *functions corresponding* to the linguistic labels. Thus we imagine a new fuzzy set in that fuzzy zone which is represented as follows;
(0.4501/0+0.3301/1+0.4101/2+0.5041/3+0.258/4+0.018/5+0.019/6) ♡

There are many other ways to fuzzify a nonfuzzy feature, e.g. fuzzy singleton (see Section 2.1.1 of Chapter 2), fuzzy masking (see Figure 4.2 of Chapter 4) etc. In this thesis, we have more frequently used the latter two methods of fuzzification instead of the current method used in this chapter. Though for a given nonfuzzy feature the three processes of fuzzification produce three different types of membership function, but the defuzzified values represent the similar nonfuzzy feature. Only different processes of fuzzification represent different degree of vagueness which we like to take care at the time of classification.

3.5 Experimental Results

To test the effectiveness of the proposed method, 5 simulations were run with multiple antecedent clause rules. The recognition scores are found to be quite satisfactory in all the cases.

Now, Figs. 2.8-2.12 of Chapter 2 which are typical pattern sets in two dimensional pattern space have been considered for design study. To implement the proposed algorithm, each fuzzy set (corresponding to the label (low, medium or high) is sampled at 7 discrete points of their corresponding domains (see Table 3.1). Then we generate the relationship in between the antecedent fuzzy sets in the first run by "min" operator and in the second run by algebraic product. We apply the relation as input to the neural net (see Figure 3.3(b)). We have used the concept of exemplification [108] to assign membership function to the linguistic labels and the pattern classes. Depending upon the need of the problem and the choice of the designer we may adopt any other schemes.

Table 3.1 displays the meaning of linguistic terms used during the proposed method. Tables 3.2 – 3.6 display the rules as well as the classification scores for pattern sets mentioned above. After each training activity, the network was tested with several random linguistic as well as real inputs (see Table 3.2(b)). It is to be mentioned that in all the cases we considered networks with single hidden layer. We have noticed through simulation studies that 35 nodes in the hidden layer produce so far the best set of results in most of the cases. However, depending on the complexity of the data set we have to increase the number of hidden neurons.e.g. for Figure 2.11 we used 40 neurons for good results.

Table 3.1

The meaning of linguistic terms defined on the domain [0,6] and sampled at integer points

Label	Membership						
Domain	$0 \leq F_1/F_2 < 1$	$1 \leq F_1/F_2 < 2$	$2 \leq F_1/F_2 < 3$	$3 \leq F_1/F_2 < 4$	$4 \leq F_1/F_2 < 5$	$5 \leq F_1/F_2 < 6$	$6 \leq F_1/F_2 < 7$
low	1.0	0.7	0.4/0.2[1]	0.01	0.01	0.01	0.01
very low	1.0	0.49	0.16	0.0001	0.0001	0.0001	0.0001
morl[2] low	0.1	0.5	1	0.5	0.1	0.1	0.1
noisy low	1.0	0.67	0.33	0.01	0.01	0.01	0.01
medium	0.01	0.01	0.5/0.2	1	0.5/0.2	0.01	0.01
very medium	0.0001	0.0001	0.25	1	0.25	0.0001	0.0001
morl medium	0.1	0.1	0.707	1	0.707	0.1	0.1
shifted medium1	0.01	0.5	1	0.5	0.01	0.01	0.01
shifted medium2	0.01	0.01	0.01	0.5	1	0.6	0.01
noisy medium	0.01	0.01	0.3	0.9	0.5	0.01	0.01
high	0.01	0.01	0.01	0.01	0.4/0.2	0.9	1
very high	0.0001	0.0001	0.0001	0.0001	0.16	0.81	1
morl high	0.01	0.1	0.4	0.7	1	0.7	0.3
noisy high	0.01	0.01	0.01	0.01	0.6	0.8	0.9

1 For convex data sets we have considered membership functions which are flat in nature. Hence, we have considered membership functions for **low** like

$$1/0 + 0.7/1 + 0.4/2 + 0.01/3 + 0.01/4 + 0.01/5 + 0.01/6$$

for fig. 2.8. But, for non-convex data sets (e.g., for fig. 2.10 we have considered membership functions like

$$1/0 + 0.7/1 + 0.2/2 + 0.01/3 + 0.01/4 + 0.01/5 + 0.01/6.$$

This representation remains same for **medium** and **high**.

2 morl is the abbreviated form of more or less

3 for the last segment, i.e. $6 \leq F_1/F_2 < 7$ we only consider F_1/F_2 equal to 6.

Table 3.2

Performance of the recognition scheme (Fig. 3.3(a,b)) for Fig 2.8 with 35 hidden neurons

a) Training rules[1]

Antecedent	Consequent[2]
1)If F_1 is low & F_2 is low Then	1/a+.03/b+0/c+0/d+0/e+0/f+.01/g+.01/h
2)If F_1 is low & F_2 is medium Then	.05/a+.01/b+0/c+0.9/d +.01/e+0/f+.01/g+.01/h
3)If F_1 is low & F_2 is high Then	0/a+0/b+0/c+.9/d+.2/e+0/f+.01/g+.01/h
4)If F_1 is medium & F_2 is low Then	.03/a+.8/b+.3/c+0/d+0/e+0/f+.01/g+.01/h
5) If F_1 is medium & F_2 is medium Then	0/a+.2/b+.1/c+.1/d+.8/e+.05/f+.01/g+.01/h
6)If F_1 is medium & F_2 is high Then	0/a+0/b+0/c+.01/d+.7/e+.05/f+.01/g+.01/h
7)If F_1 is high & F_2 is low Then	0/a+0/b+1/c+0/d+0/e+0/f+.01/g+.01/h
8)If F_1 is high & F_2 is medium Then	0/a+0/b+0/c+0/d+0/e+.7/f+.01/g+.01/h
9)If F_1 is high & F_2 is high Then	0/a+0/b+0/c+0/d+0/e+.7/f+.01/g+.01/h
10) If F_1 is more or less low & F_2 is more or less low Then	.8/a+.3/b+0/c+0/d+0/e+0/f+.1/g+.1/h
11) If F_1 is more or less high & F_2 is more or less low Then	0/a+.3/b+1/c+0/d+.1/e+.1/f+.1/g+0/h
12) If F_1 is more or less low & F_2 is more or less medium Then	.2/a+.1/b+0/c+.8/d+.2/e+0/f+0/g+.1/h
13) If F_1 is more or less high & F_2 is more or less medium Then	0/a+.1/b+.3/c+0/d+.8/e+.2/f+.1/g+0/h

1 Training terminated when the total sum of squared errors dropped below e = 0.01.

2 The Training rules 10 to 13 are added for achieving better classification results.

b)Testing results

Linguistic Test Input		Expected/Actual class
Feature1	Feature2	
low	low	a / a
low	medium	d / d
low	high	d / d
medium	low	b / b
medium	medium	e / e
medium	high	e / e
high	low	c / c
high	medium	f / f
high	high	f / f
very low	very low	a / a
very low	very medium	d / d
very low	very high	d / d
very medium	very low	b / b
very medium	very medium	e / e
very medium	very high	e / e
very high	very low	c / c
very high	very medium	f / f
very high	very high	f / f
morl low	morl low	a / a
morl low	morl medium	d / d
morl low	morl high	d / d
morl medium	morl low	c / c
morl medium	morl medium	e / e
morl medium	morl high	e / e
morl high	morl low	c / c
morl high	morl medium	f / f
morl high	morl high	f / f
shifted medium1	shifted medium1	a/b/d / a
shifted medium1	shifted medium2	d/e / e
shifted medium2	shifted medium1	c / c
shifted medium2	shifted medium2	f / f
low	noisy medium	d / d
noisy low	high	d / d

Testing Results (Contd.)

Linguistic Test Input		Expected/Actual class
Feature1	Feature2	
low	noisy high	d / d
noisy low	medium	d / d
medium	noisy medium	e / e
medium	noisy high	e / e
noisy low	noisy medium	d / d
noisy medium	high	e / e
medium	noisy low	e / e

For linguistic test inputs: Score : 100%

For Exhaustive search with real test inputs: Score : 83.6%

Note that by real test inputs we want to mean the nonfuzzy values of the features F_1 and F_2.

iii) Score of Bayesian Classification : 78%

Table 3.3

Performance of the recognition scheme (Fig. 3.3(a,b)) for Fig 2.9

with 35 hidden neurons

a) Training rules[1]

Antecedent	Consequent[2]
1) If F_1 is low & F_2 is low Then	1/a+0/b+0/c+.01/d+.01/e
2) If F_1 is low & F_2 is medium Then	1/a+0/b+0/c+0/d+.01/e
3) If F_1 is low & F_2 is high Then	1/a+0/b+0/c+0/d+.01/e
4) If F_1 is medium & F_2 is low Then	1/a+.1/b+0/c+.01/d+0/e
5) If F_1 is medium & F_2 is medium Then	.1/a+.9/b+.9/c+.01/d+.01/e
6) If F_1 is medium & F_2 is high Then	1/a+0/b+.1/c+0/d+.01/e
7) If F_1 is high & F_2 is low Then	1/a+.1/b+0/c+.01/d+0/e
8) If F_1 is high & F_2 is medium Then	0/a+.9/b+.9/c+.01/d+0/e
9) If F_1 is high & F_2 is high Then	1/a+0/b+.1/c+.01/d+.01/e
10) If F_1 is more or less high & F_2 is more or less low Then	.9/a+.1/b+0/c+.01/d+0/e

1 The 10^{th} rule has been added for better performance.

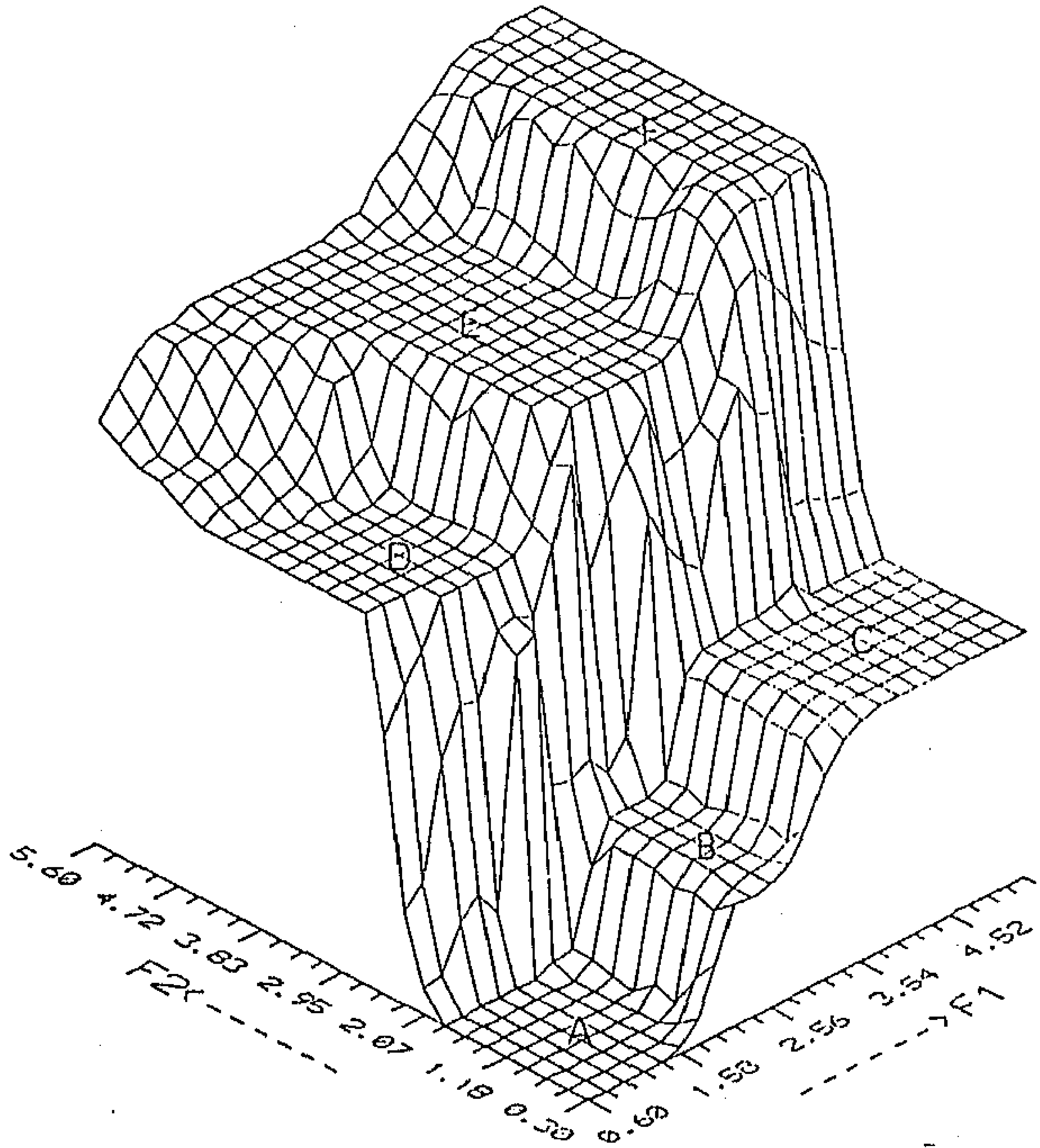


Fig. 3.5: Simulation of the recognition score for Fig. 2.8

b)Testing results

Linguistic Test Input		Expected/Actual class
Feature1	Feature2	
low	low	a / a
low	medium	a / a
low	high	a / a
medium	low	a / a
medium	medium	b, c^2 / b
medium	high	a / a

Testing Results (Contd.)

linguistic Test Input		Expected/Actual class
Feature1	Feature2	
high	low	a / a
high	medium	b, c / b, c
high	high	a / a
very low	very low	a / a
very low	very medium	a / a
very low	very high	a / a
very medium	very low	a / a
very medium	very medium	b,c / b
very medium	very high	a / a
very high	very low	a / a
very high	very medium	b,c / b
very high	very high	a / a
morl low	morl low	a / a
morl low	morl medium	a / a
morl low	morl high	a / a
morl medium	morl low	a / a
morl medium	morl medium	b,c / b,c
morl medium	morl high	a / a
morl high	morl low	a / a
morl high	morl medium	b,c / b,c
morl high	morl high	a / a
shifted medium1	shifted medium1	a / a
shifted medium1	shifted medium2	a / a
shifted medium2	shifted medium1	a,b / a
shifted medium2	shifted medium2	c / a
low	noisy medium	a / a
noisy low	high	a / a
low	noisy high	a / a
noisy low	medium	a / a
medium	noisy medium	b,c / b,c
medium	noisy high	a / a
noisy low	noisy medium	a / a
noisy medium	high	a / a
medium	noisy low	a / a

2 membership values for class b and class c differ only after the $3^{rd}/4^{th}$ decimal point. Same is the case for any other sample that belongs to more than one class for all the data sets.

For liguistic test inputs: Score : 97.5%

For Exhaustive search with real test inputs: Score : 87.6%

iii) Score of Bayesian Classification : 86.9%

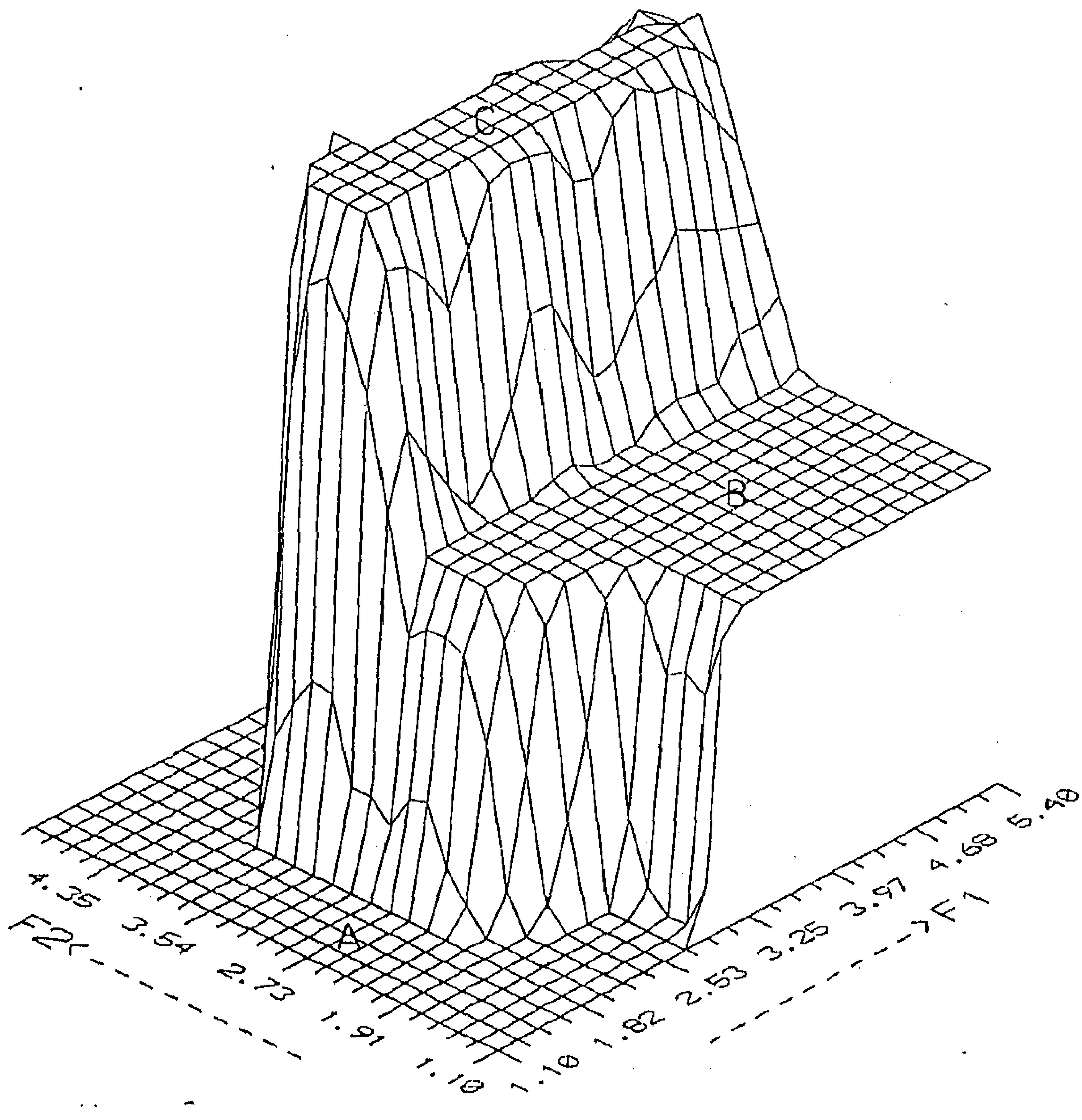


Fig. 3.6: Simulation of the recognition score for Fig. 2.9

Table 3.4

Performance of the recognition scheme (Fig. 3.4) for Fig 2.10 with 35 hidden neurons

a) Training rules[1]

Antecedent	Consequent[2]
1) If F_1 is low & F_2 is low Then	1/a+0/b+.01/c+.01/d
2) If F_1 is low & F_2 is medium Then	1/a+0/b+0/c+.01/d
3) If F_1 is low & F_2 is high Then	0/a+0/b+0/c+.01/d
4) If F_1 is medium & F_2 is low Then	1/a+0/b+.01/c+0/d
5) If F_1 is medium & F_2 is medium Then	.18/a+.81/b+.01/c+.01/d
6) If F_1 is medium & F_2 is high Then	1/a+0/b+0/c+.01/d
7) If F_1 is high & F_2 is low Then	0/a+1/b+.01/c+0/d
8) If F_1 is high & F_2 is medium Then	.16/a+.83/b+.01/c+0/d
9) If F_1 is high & F_2 is high Then	0/a+0/b+.01/c+.01/d
10) If F_1 is more or less high & F_2 is more or less medium Then	.5/a+.5/b+.01/c+.01/d
11) If F_1 is very high & F_2 is low Then	0/a+1/b+.01/c+.01/d
12) If F_1 is very high & F_2 is medium Then	0/a+1/b+.01/c+0/d

1 Rules 10 to 12 have been added for improving the recognition scores

b)Testing results

Linguistic Test Input		Expected/ Actual class
Feature1	Feature2	
low	low	a / a
low	medium	a / a
low	high	d / b
medium	low	a / a
medium	medium	b / b
medium	high	a / a
high	low	b / b
high	medium	b / b
high	high	c,d / b
very low	very low	c,d / a

Testing Results (Contd.)

linguistic Test Input		Expected/Actual class
Feature1	Feature2	
very low	very medium	a / a
very low	very high	d / b
very medium	very low	a / a
very medium	very medium	b / b
very medium	very high	a / a
very high	very low	b / b
very high	very medium	b / b
very high	very high	c,d / b
morl low	morl low	c,d / a
morl low	morl medium	a / a
morl low	morl high	d / a
morl medium	morl low	a / a
morl medium	morl medium	b / a
morl medium	morl high	a / a
morl high	morl low	b / b
morl high	morl medium	b / b
morl high	morl high	c,d / a
shifted medium1	shifted medium1	a / a
shifted medium1	shifted medium2	a / a
shifted medium2	shifted medium1	a,b / b
shifted medium2	shifted medium2	b / b
low	noisy medium	a / a
noisy low	high	d / b
low	noisy high	d / a
noisy low	medium	a / a
medium	noisy medium	b / b
medium	noisy high	a / a
noisy low	noisy medium	a / a
noisy medium	high	a / a
medium	noisy low	a / a

For linguistic test inputs: Score : 70%

For Exhaustive search with real test inputs: Score : 83.7%

iii) Score of Bayesian Classification : 88.4%

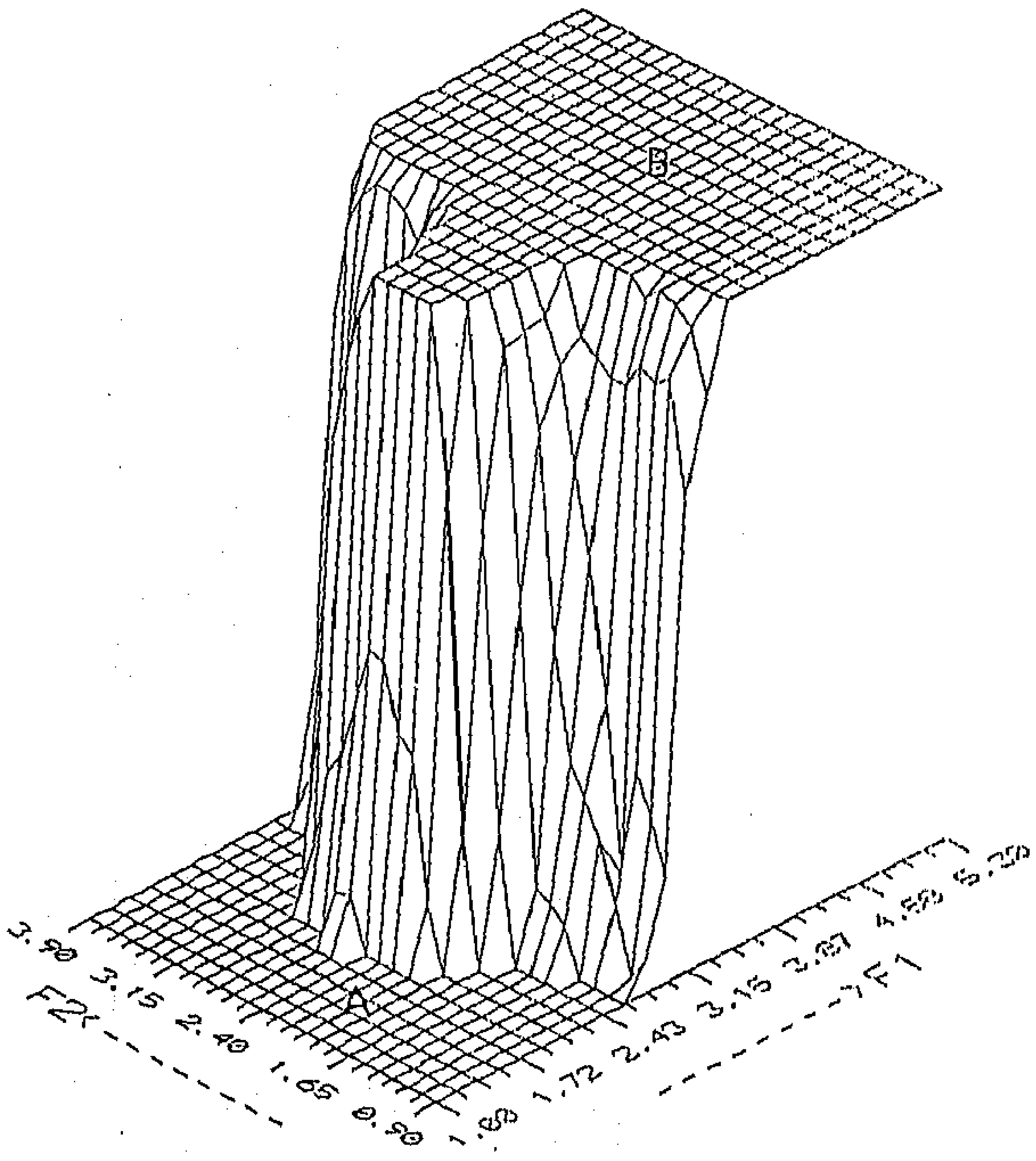


Fig. 3.7: Simulation of the recognition score for Fig. 2.10

Table 3.5

Performance of the recognition scheme (Fig. 3.3(a,b)) for Fig 2.11 with 40 hidden neurons

a) Training rules[1]

Antecedent	Consequent
1) If F_1 is low & F_2 is low Then	.0/a+1/b+.01/c+.01/d
2) If F_1 is low & F_2 is medium Then	.1/a+.8/b+0/c+.01/d
3) If F_1 is low & F_2 is high Then	0/a+1/b+0/c+.01/d
4) If F_1 is medium & F_2 is low Then	0/a+1/b+.01/c+0/d
5) If F_1 is medium & F_2 is medium Then	1/a+0/b+.01/c+.01/d

Training rules(Contd.)

Antecedent	Consequent
6) If F_1 is medium & F_2 is high Then	.1/a+1/b+.01/c+0/d
7) If F_1 is high & F_2 is low Then	0/a+1/b+.01/c+0/d
8) If F_1 is high & F_2 is medium Then	.1/a+.8/b+.01/c+0/d
9) If F_1 is high & F_2 is high Then	0/a+1/b+.01/c+.01/d
10) If F_1 is more or less low & F_2 is medium Then	.7/a+.3/b+.01/c+.01/d
11) If F_1 is medium & F_2 is more or less low Then	.7/a+.3/b+0/c+.01/d
12) If F_1 is medium & F_2 is more or less high Then	.7/a+.3/b+.01/c+0/d
13) If F_1 is more or less high & F_2 is medium Then	.7/a+.3/b+.01/c+.01/d

1 Rules 10 to 13 have been added for improving the recognition score

b)Testing results

Linguistic Test Input		Expected / Actual class
Feature1	Feature2	
low	low	b / b
low	medium	b / b
low	high	b / b
medium	low	b / b
medium	medium	a / a
medium	high	b / b
high	low	b / b
high	medium	b / b
high	high	b / b
very low	very low	b / b
very low	very medium	b / b
very low	very high	b / b

Testing Results (Contd.)

Linguistic Test Input		Expected / Actual class
Feature1	Feature2	
very medium	very low	b / b
very medium	very medium	b / b
very medium	very high	b / b
very high	very low	b / b
very high	very medium	b / b
very high	very high	b / b
morl low	morl low	b / b
morl low	morl medium	b / b
morl low	morl high	b / b
morl medium	morl low	b / b
morl medium	morl medium	a / a
morl medium	morl high	b / b
morl high	morl low	b / b
morl high	morl medium	b / b
morl high	morl high	b / b
shifted medium1	shifted medium1	b / a
shifted medium1	shifted medium2	b / b
shifted medium2	shifted medium1	b / b
shifted medium2	shifted medium2	b / b
low	noisy medium	b / b
noisy low	high	b / b
low	noisy high	b / b
noisy low	medium	b / b
medium	noisy medium	a / a
medium	noisy high	b / b
noisy low	noisy medium	b / b
noisy medium	high	b / b
medium	noisy low	b / b

For linguistic test inputs: Score : 97.5%

For Exhaustive search with real test inputs: Score : 74%

iii) Score of Bayesian Classification : 90.8%

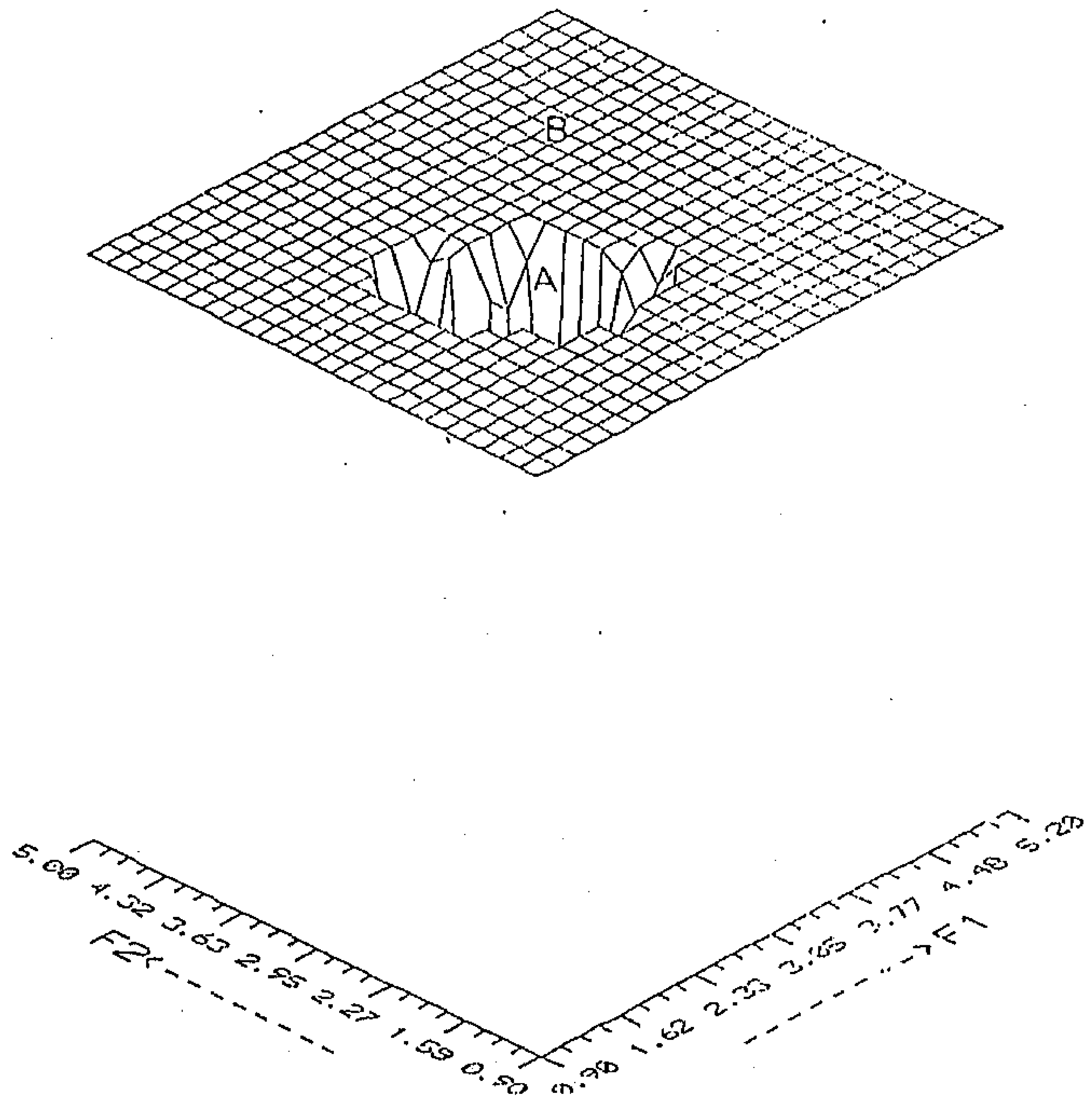


Fig. 3.8: Simulation of the recognition score for Fig. 2.11

Table 3.6

Performance of the recognition scheme (Fig. 3.3(a,b)) for Fig 2.12 with 35 hidden neurons

a) Training rules

Antecedent	Consequent
1) If F_1 is low & F_2 is low Then	0/a+0/b+.01/c+.01/d
2) If F_1 is low & F_2 is medium Then	.5/a+.5/b+0/c+.01/d
3) If F_1 is low & F_2 is high Then	.6/a+.4/b+0/c+.01/d
4) If F_1 is medium & F_2 is low Then	.7/a+.3/b+.01/c+0/d

Training rules(Contd.)

Antecedent	Consequent
5) If F_1 is medium & F_2 is medium Then	.45/a+.55/b+.01/c+.01/d
6) If F_1 is medium & F_2 is high Then	.7/a+.3/b+0/c+.01/d
7) If F_1 is high & F_2 is low Then	0/a+0/b+.01/c+0/d
8) If F_1 is high & F_2 is medium Then	.1/a+.9/b+.01/c+0/d
9) If F_1 is high & F_2 is high Then	.45/a+.55/b+.01/c+.01/d

b)Testing results

Linguistic Test Input		Expected / Actual class
Feature1	Feature2	
low	low	b / b
low	medium	a,b / a
low	high	a / a
medium	low	a / a
medium	medium	b / b
medium	high	a / a
high	low	c / b
high	medium	b / b
high	high	b / b
very low	very low	b / b
very low	very medium	a,b / a
very low	very high	a / a
very medium	very low	a / a
very medium	very medium	b / b
very medium	very high	a / a
very high	very low	c / b
very high	very medium	b / b
very high	very high	b / b
morl low	morl low	b / a
morl low	morl medium	a,b / a
morl low	morl high	a / a
morl medium	morl low	a / a
morl medium	morl medium	b / a
morl medium	morl high	a / a
morl high	morl low	c / a
morl high	morl medium	b / b
morl high	morl high	b / b
low	noisy medium	a,b / a
noisy low	high	a / a

Testing results (Contd.)

Linguistic Test Input		Expected / Actual class
Feature1	Feature2	
low	noisy high	a / a
noisy low	medium	a,b / a
medium	noisy medium	b / b
medium	noisy high	a / a
noisy low	noisy medium	a,b / a
noisy medium	high	a / a
medium	noisy low	a / a
shifted medium1	shifted medium1	b / a
shifted medium1	shifted medium2	a / b
shifted medium2	shifted medium1	a,b / b
shifted medium2	shifted medium2	b / b

For linguistic test inputs: Score : 82.5%

For Exhaustive search with real test inputs: Score : 53.6%

iii) Score of Bayesian Classification : 59.3%

3.6 Discussion on the Experimental Procedures

The classification results shown in Table 3.2-3.6 are based on concept of single choice. The overall classification results are comparable with the Bayesian classifications mentioned in Table 3.2-3.6. Now, if we consider the classification of patterns in the overlapped zone (see Fig. 2.8) we will get, in general, the following two situations.

a)Tie Situation; that means more than one class have maximum membership value (see item 1 of this section). Under such situation, we have to talk about multiple classification of a given pattern. However, because of numerical computation, occurrence of tie situation is rare.

b)Almost equal situation; that means patterns in the overlapped zones of classes are classified with almost equal possibility which is indicated by almost equal membership values representing more than one class. If such situation is treated as tie situation mentioned above, we have to select an appropriate threshold which entirely depends on the need of the problem.

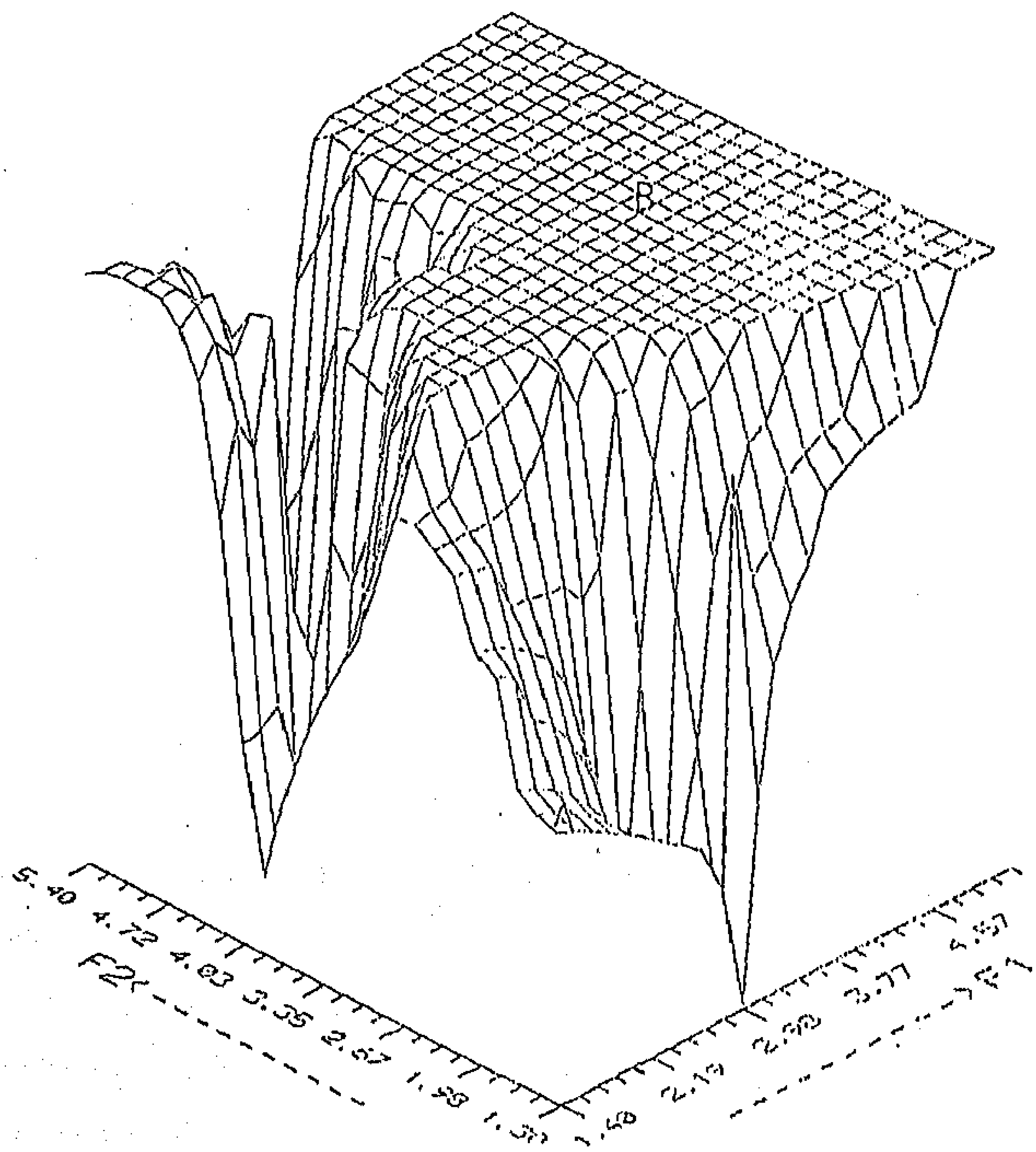


Fig. 3.9: Simulation of the recognition score for Fig. 2.12

Such a decision about multiple classification is quite natural which does not exist in conventional classification approach. The inclusion of multiple classification enhances the recognition score to a great extent. For instance, let us consider Fig. 2.8. Initially, we go by the concept of distinct classification (see Table 3.2,ii) ignoring the almost equal possibility of occurrence of patterns at the overlapped zone. Subsequently, when we consider the concept of multiple classification at the overlapped zone we get the recognition score (97.6 %) which is significantly high.

The exhaustive evaluation of the performance (using distinct classification) of the present classification scheme has been depicted in Figures 3.5 - 3.9 which clearly indicates the class segregation for each set of data. The recognition scores are also calculated based on exhaustive verification of the data and indicated in the tables 3.2 - 3.6 (under Testing Results (ii)).

Using algebraic product relation on the antecedent clause of the **If Then** rules we have got almost similar results. Hence, those results are not reported here.

3.7 Conclusion

We have used backpropagation neural network for fuzzy linguistic pattern classification. In our approach, we have precisely implemented the interpretation (2.14(a)) of MFI of Section 2.9 of Chapter 2 for fuzzy reasoning, which is nothing but the fuzzy reasoning proposed by Mizumoto [61], Zimmermann [62] and Zadeh [108] (see Section 2.2, 2.3 and 2.4) on the backpropagation neural network architecture. We have considered several examples and have obtained very promising results. But we have realized that when number of features of a pattern is more than two, in that case interpretation (2.14(a)) of Section 2.9 of Chapter 2 will be computationally heavy in the sense that the fuzzy relation formed by the antecedent clause of an one dimensional implication will be represented by a very large array of numbers which actually dictates the large number of inputs to the neural network (see Figure 3.3(b)). Thus, the network complexity increases. To overcome such problem in the following chapters we will discuss the interpretation (2.16) of Section 2.9 of Chapter 2.

Chapter 4

Fusion Technology based on the new interpretation of MFI

To tackle the pattern classification problems, in this chapter [82] first we consider new interpretation of MFI (i.e. Equation (2.16) of Chapter 2) and then realize that new interpretation through backpropagation type neural network. At the learning stage of the neural network, fuzzy linguistic statements have been used. Once learned, the nonfuzzy features of a pattern can be classified. At the time of classification of the nonfuzzy features of a pattern, we use the concept of fuzzy singleton. The performance of the proposed scheme is tested through synthetic data. Finally, we use the proposed scheme to the vowel classification problem of three Indian languages.

4.1 MFR and statement of the problem

In the multidimensional fuzzy reasoning (MFR) approach to pattern classification each element of the fuzzy feature vector / pattern vector $\vec{F_{fj}}$ (see Figure 2.26(a) of Chapter 2), as discussed earlier, is represented by the fuzzy linguistic variable instead of a real number. For instance, suppose, we have a (2×1) feature vector $F = (F_1, F_2)^T$, T is transpose where F_1 is the first formant frequency of a speech signal and F_2 is the second formant frequency. In the conventional approach to pattern recognition [97], F_1 and F_2 are two features and are represented by, say, 800 Hz and 550 Hz. Whereas in the MFR approach to pattern classification, F_1 and F_2 are represented by the fuzzy linguistic variables, e.g. F_1 is **small** and F_2 is **medium**. The elements of the feature vector $\vec{F_{fj}}$ which are represented by fuzzy linguistic variables (see Figure 2.26(a)) are characterized by their membership functions which quantize the pattern space as shown in Figure 2.4 of Chapter 2. These elements of the feature vector $\vec{F_{fj}}$ actually constitute the antecedent parts of the DFIs (see Equation (2.16)). The consequent part of the MFI, as discussed in Section 2.9, represents the possibility of occurrence of different classes of patterns in the pattern space. Thus, when we decompose the multidimensional fuzzy implication into the form shown in Equation (2.16), the first consequent C_1 represents a fuzzy set. The fuzzy set C_1 is obtained by considering the relative position of the relation achieved by the cylindrical extension of the fuzzy set like M_1 and the well defined cover of the pattern space as mentioned earlier in Section 2.1 of Chapter 2 (see Figure 2.26(b) and Example 2.2). Similarly we obtain the fuzzy set C_2 which is the consequent part of the second implication of Equation (2.16) of Chapter 2. The resultant consequent $\hat{C} = C_1 \bigcap C_2$ which represents a fuzzy set similar to the fuzzy Set C (see Example 2.2 of Chapter 2). Both the fuzzy sets $\hat{C}$ and C have same number of elements but having different membership values. This happens simply because the fuzzy set C and $\hat{C}$ are achieved through two different processes of quantification of perception. Though the basic aim is to perceive the same situation (i.e. relative position of the tip of the fuzzy vector ABCD and the well defined cover of the pattern space) but one is achieved (i.e. fuzzy set C) directly from the relative position of the area ABCD and the well defined cover of the pattern space whereas the other one (i.e. the fuzzy set $\hat{C} = C_1 \bigcap C_2$) is achieved looking at the relative position of two different relations obtained from the cylindrical extensions of two fuzzy sets like M_1 and M_2 and their (the relations formed by cylindrical extensions of M_1 and M_2) relative positions with the well defined cover of the pattern space. The multidimensional fuzzy **If Then** rules are decomposed in the form of Equation (2.16) which maps the given patterns to previously determined regions.

4.2 Implementation of the new interpretation of MFI on backpropagation type neural network

Let us consider Equation (2.16). The first expression of Equation (2.16) is one law of implication and the second expression of Equation (2.16) is another law of implication. Both of these laws of implication can be independently realized through two backpropagation neural networks which are basically the conventional three layered perceptrons (see Figure 4.1). The input of each neural network is the antecedent part of a DFI. The antecedent part of a DFI is represented by a fuzzy set (e.g. the fuzzy set M_1 of Figure 2.26(b)). The reference output of the network is the consequent part of the said DFI (e.g. the fuzzy set C_1 as mentioned earlier in Section 4.1 and Example 2.4 of Chapter 2). Like this, depending upon the number of DFIs, we have number of neural networks which are stacked one after another (see Figure 4.1). Features of each network are same as the conventional backpropagation neural network. Each network is trained independently by a set of one dimensional fuzzy **If Then** statements using generalized delta rule [69]. Each set of one dimensional fuzzy **If Then** rules is formed by considering different linguistic status of a particular feature of a given pattern. For instance, out of feature F_1 and F_2 if we consider one DFI which deals with different linguistic status of the feature F_1 (F_1 is small / F_1 is medium / F_1 is big) then we will be getting a collection of one dimensional fuzzy implications which forms the set stated above to train a backpropagation neural network of Figure 4.1. Like this, we can train other networks which are stacked one after another as shown in Figure 4.1. Once all the networks in the stack are trained independently, we can combine the output of each network by an intersection ($\bigcap$) operator (see Figure 4.1). Thus, if we have a two-dimensional law of fuzzy implication, we can realize MFR through a network configuration as shown in Figure 4.1. If we have n-dimensional law of fuzzy implication, we can realize MFR through a network which consists of n-number of independent 3-layered perceptrons which are trained using the principle of backpropagation neural networks. The outputs of n-networks are combined through intersection ($\bigcap$) operator. In this chapter, as we have considered pattern classification on $\Re^2$, we will always follow the network configuration of Figure 4.1. As we stated earlier (see the discussion of Section 2.9 after Equation (2.16)) to deal with an interpretation like (2.16) means we will be considering single antecedent clause of a DFI and its consequence along with the cylindrical extension of the said single antecedent clause of the DFI over the appropriate universe of the feature axis. Though the cylindrical extension is not explicitly shown in a network configuration like Figure 4.1, its effect is reflected when we represent the reference output of each backpropagation neural network (see Example 2.2 and 2.4 of Chapter 2 and Figure 2.26(b)) by a fuzzy set like C_i, i varies from 1 to 2, which is formed considering the relative position of the relation obtained by the cylindrical extension of the single antecedent clause of a DFI and the well defined cover of the pattern space.

4.3 Formulation of the problem

At the learning stage, first of all we discretize the universe of discourses of the features F_1 and F_2. Discretization is often referred to as quantization which discretizes a universe into a certain number of segments (quantization levels). Each segment is labeled as a generic element and forms a discrete universe. A fuzzy set is then defined by assigning grade of membership values to each generic element of the new discrete universe (see

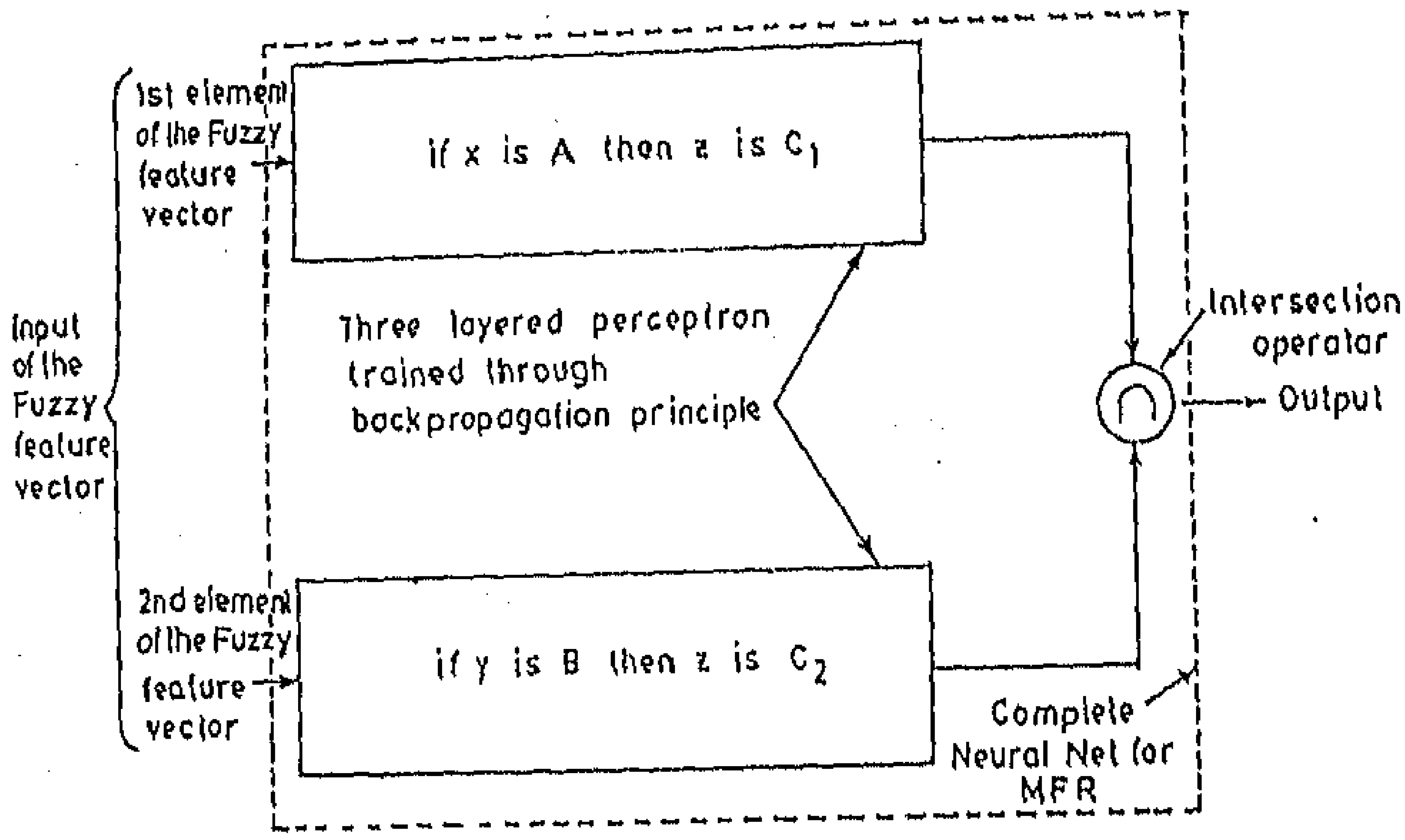


Fig. 4.1: Realization of two-dimensional MFR through backpropagation type neural network

Tables 4.2, 4.4, 4.5, 4.7, 4.10, 4.11, 4.13, 4.14, 4.16, 4.17).

The membership function of the consequent part of the **If Then** rule represents the possibility of occurrence of different classes of patterns in the pattern space. The detail procedure for representing the consequent part of a DFI is given below.

At the learning stage of the classifier, we depend on the expert's experience which is captured through fuzzy **If Then** rules. For this purpose, we form a Query Table(QT). We explain this table with the help of an example. Let F_1 and F_2 be the names of two linguistic variables and **High**, **Medium** and **Low** are three primary terms. Assume that F_1 and F_2 are the input features to the classifier. Let there be two classes, $\tilde{C}_1$ and $\tilde{C}_2$. Suppose the output linguistic variables are formed with the help of two primary terms **Low** and **High**. Now, we construct the QT as shown in Table 4.1.

If we consider Table 4.1 and read any of the rows first and then any of the columns of the QT, a question will be formed. The answer to this question has to be put in the

Table 4.1: QT MATRIX

IF	WILL $\tilde{C}_1$ BE		WILL $\tilde{C}_2$ BE	
(ANTECEDENT PART OF A DFI)	LOW	HIGH	LOW	HIGH
F_1 IS LOW	~	~	~	~
F_2 IS HIGH	-	+	+	-

corresponding elemental position of the matrix. If the answer is 'yes', we put "+", if 'no', a "-" and if not answerable a " ~ ". As for illustration, if we read the first row and then the fourth column, we will have the question: "**If** F_1 **is Low will** $\tilde{C}_2$ **be High ?** " If the answer is 'yes', we put a "+" in (1,4) position of the QT. For further clarity, see Tables 4.3, 4.6, 4.8, 4.9. Now, we set "**very high**" = 1, "**high**" = .9, "**more or less high**" = .7, "**medium**" = .6, "very low" = .01, "low" = .1 and "**more or less** low" = .2 to assign membership values to the consequent part of the **If Then** rules.

Such assignment of membership function to the consequent part of the fuzzy **If Then** rules does not follow any conventional approach to fuzzification. Rather we use the concept of scaling [98] and the exponential scales proposed by Lootsma [53]. This approach of possibility assignment is basically a subjective quantification of human (expert) perception [108] based on the aforesaid scaling [53,98]. For any practical application, there is no single scale which can outperform other scales [98]. Therefore, depending upon the application, we have to select an appropriate scale. In [53], there are some discussions on scales which are similar to the one used in the present thesis. There are basically two critical problems raised in scaling. One is how to quantify (numerically) the linguistic choices selected by the decision maker and the second is how to process the numerical values. Both the problems have nicely been tackled in the present scaling process using fuzzy set.

After the initial generation of the **If Then** rules, we test the validity of the rules by classifying some known patterns. If we get satisfactory classification results, we proceed further; otherwise, we tune the rules by changing the grades of the membership function of the antecedent part and the consequent part.

As off-line generation of fuzzy **If Then** rules for pattern classification basically deals with a static situation, the tuning of the grade of the membership functions of the antecedent part and the consequent part does not take a very long time which is a very common phenomenon for tuning fuzzy control rules of the dynamical system. In the present situation, tuning of the consequent part of the rule is primarily guided by the seed points of the clusters of the patterns and that of the antecedent part of the rule is guided by the error generated in the classification. After tuning, the net (Figure 4.1) is relearned by the refined set of rules.

At the classification stage, the selected features are fuzzified using the concept of fuzzy singleton. The classification results produce the possibility of occurrence of different classes of patterns in the pattern space. At the time of taking nonfuzzy decision out of this fuzzy classification (i.e. defuzzification) we can go by selecting the class having highest possibility value. In case of tie situation, which normally occurs for the patterns lying in the overlapped zones of different classes (see Figure 2.5 of Chapter 2), we have to state the equal possibility of a pattern to belong to different classes. And such a conclusion is quite natural which normally does not exist in conventional classification approach. In some cases, patterns in the fuzzy zones are classified with "**almost equal**" possibility of occurrence for more than one class. If such situation is treated as tie situation mentioned above, we have to select an appropriate threshold which entirely depends on the need of the problem. Thus, from the graded consequence, when we select a single class having the highest membership value, we consider the hard partitioning of the feature space. Whereas, from the graded consequence, when we consider the multiple classes occurring in the overlapped zone of different classes, we consider the fuzzy partitioning of the pattern space.

Table 4.2:Quantization of the feature space of Figure 2.9

	Small	Medium	Big
$0 \leq F_1, F_2 \leq 0.5$	1	0	0
$0.5 < F_1, F_2 \leq 1$	0.8	0	0
$1 < F_1, F_2 \leq 1.5$	0.6	0	0
$1.5 < F_1, F_2 \leq 2$	0.4	0.2	0
$2 < F_1, F_2 \leq 2.5$	0.2	0.7	0
$2.5 < F_1, F_2 \leq 3$	0	1	0
$3 < F_1, F_2 \leq 3.5$	0	0.7	0
$3.5 < F_1, F_2 \leq 4$	0	0.2	0.2
$4 < F_1, F_2 \leq 4.5$	0	0	0.4
$4.5 < F_1, F_2 \leq 5$	0	0	0.6
$5 < F_1, F_2 \leq 5.5$	0	0	0.8
$5.5 < F_1, F_2 \leq 6$	0	0	1

We have tested our proposed scheme on a set of synthetic data and some real life vowel classification problems which are discussed in the following sections.

4.4 Numerical Examples

To test the effectiveness of the proposed scheme as shown in Figure 4.1, simulations were run for Figures 2.8 to 2.11. The rules, distribution pattern, number of hidden nodes and the classification scores are shown in Table 4.3, 4.6, 4.8 and 4.9. The quantizations of feature space are as described in Table 4.2, 4.4, 4.5, 4.7. The recognition scores are found to be quite satisfactory in all the cases. At the time of calculating the recognition score, we consider the exhaustive data for each Figure.

Table 4.3: Results of Figure 2.8

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent							
		A	B	C	D	E	F	G	H
NN_1^1	If F_1 is Small Then	Very High	Low	Nil	More or Less High	Very Low	Nil	Very Low	Very Low
	If F_1 is Medium Then	Low	Very High	Medium	Very Low	High	Medium	Very Low	Very Low
	If F_1 is Big Then	Nil	Nil	High	Nil	Nil	High	Very Low	Very Low
NN_2^2	If F_2 is Small Then	Very High	Very High	Very High	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is Medium Then	Very Low	Very Low	Very Low	High	High	More or Less High	Very Low	Very Low
	If F_2 is Big Then	Nil	Nil	Nil	Low	More or Less High	High	Very Low	Very Low

1 learned in 90 iterations

2 learned in 115 iterations

Recognition Score

Hard Partitioning

class A	class B	class C	class D	class E	class F	overall score
93.1 %	85.4 %	86 %	41 %	68 %	94.7 %	78.4 %

Fuzzy Partitioning

class A	class B	class C	class D	class E	class F	overall score
93.1 %	93.1 %	93.6 %	41 %	100 %	97.3 %	88.4 %

Table 4.4: Quantization of the feature space for feature1 of Figure 2.9

	Small	Medium	Big
$0 \leq F_1 \leq 0.5$	1	0	0
$0.5 < F_1 \leq 1$	0.7	0	0
$1 < F_1 \leq 1.5$	0.3	0	0
$1.5 < F_1 \leq 2$	0.1	0.1	0
$2 < F_1 \leq 2.5$	0	0.5	0
$2.5 < F_1 \leq 3$	0	1	0
$3 < F_1 \leq 3.5$	0	0.5	0
$3.5 < F_1 \leq 4$	0	0.1	0
$4 < F_1 \leq 4.5$	0	0	0.1
$4.5 < F_1 \leq 5$	0	0	0.3
$5 < F_1 \leq 5.5$	0	0	0.7
$5.5 < F_1 \leq 6$	0	0	1

Table 4.5: Quantization of the feature space for feature2 of Figure 2.9

	Small	Medium1	medium2	Big
$0 \leq F_2 \leq 0.5$	1	0	0	0
$0.5 < F_2 \leq 1$	0.7	0	0	0
$1 < F_2 \leq 1.5$	0.3	0.1	0	0
$1.5 < F_2 \leq 2$	0.1	0.5	0	0
$2 < F_2 \leq 2.5$	0	0.7	0	0
$2.5 < F_2 \leq 3$	0	1	1	0
$3 < F_2 \leq 3.5$	0	0	0.7	0
$3.5 < F_2 \leq 4$	0	0	0.5	0
$4 < F_2 \leq 4.5$	0	0	0.1	0.1
$4.5 < F_2 \leq 5$	0	0	0	0.3
$5 < F_2 \leq 5.5$	0	0	0	0.7
$5.5 < F_2 \leq 6$	0	0	0	1

Table 4.6: Results of Figure 2.9

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent						
		A_1	A_2	A_3	B	C	D	E
NN_1^1	If F_1 is Small Then	Very High	Nil	Nil	Nil	Nil	Very Low	Very Low
	If F_1 is Medium Then	Nil	Very High	Very High	High	High	Very Low	Very Low
	If F_1 is Big Then	Nil	Very High	Very Low	More or Less High	More or Less High	Very Low	Very Low
NN_2^2	If F_2 is Small Then	Very High	Very High	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is Medium1 Then	More or Less High	Nil	Nil	Very High	Nil	Very Low	Very Low
	If F_2 is Medium2 Then	More or Less High	Nil	Nil	Nil	Very High	Very Low	Very Low
	If F_2 is Big Then	Very High	Very High	Nil	Nil	Low	Very Low	Very Low

1 learned in 101 iterations

2 learned in 102 iterations

Recognition Score

class A	class B	class C	overall score
94.4 %	61.8 %	92 %	87.4 %

Table 4.7: Quantization of the feature space of Figure 2.10 and 2.11

	Small	Medium	Big
$0 \leq F_1, F_2 \leq 0.5$	1	0	0
$0.5 < F_1, F_2 \leq 1$	0.7	0	0
$1 < F_1, F_2 \leq 1.5$	0.3	0	0
$1.5 < F_1, F_2 \leq 2$	0.1	0.1	0
$2 < F_1, F_2 \leq 2.5$	0	0.5	0
$2.5 < F_1, F_2 \leq 3$	0	1	0
$3 < F_1, F_2 \leq 3.5$	0	0.5	0
$3.5 < F_1, F_2 \leq 4$	0	0.1	0
$4 < F_1, F_2 \leq 4.5$	0	0	0.1
$4.5 < F_1, F_2 \leq 5$	0	0	0.3
$5 < F_1, F_2 \leq 5.5$	0	0	0.7
$5.5 < F_1, F_2 \leq 6$	0	0	1

4.5 Applications

After achieving satisfactory results on synthetic set of data, we first apply the proposed scheme for the vowel classification problems of three Indian languages, namely Telugu, Bengali and Assamese. Subsequently, the same vowels, uttered by different speakers are classified by the same neuro fuzzy classifier establishing its capacity to manage the uncertainties associated with the features of the vowel classification problem. The quantizations of Telugu vowels (Fig. 2.19) are described in Table 4.10 and 4.11. In Table 4.12 the rules, distribution pattern, number of hidden nodes and the classification scores are described. The quantizations of Bengali vowels (Fig. 2.21) are described in Table 4.13 and 4.14. In Table 4.15 the rules, distribution pattern, number of hidden nodes and the classification scores are described. The quantizations of Assamese vowels (Fig. 2.23) are described in Table 4.16 and 4.17. In Table 4.18 the rules, distribution pattern, number of hidden nodes and the classification scores are described.

Table 4.8: Results of Figure 2.10

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent					
		A_1	A_2	B_1	B_2	C	D
NN_1^1	If F_1 is Small Then	Very High	Nil	Nil	Nil	Very Low	Very Low
	If F_1 is Medium Then	Low	High	More or Less High	Nil	Very Low	Very Low
	If F_1 is Big Then	Nil	Very Low	More or Less High	Very High	Very Low	Very Low
NN_2^2	If F_2 is Small Then	Very High	Very High	Nil	Very High	Very Low	Very Low
	If F_2 is Medium Then	Very High	Low	Very High	Nil	Very Low	Very Low
	If F_2 is Big Then	Low	Very High	Nil	Nil	Very Low	Very Low

1 learned in 122 iterations

2 learned in 82 iterations

Recognition Score

class A	class B	overall score
98.2 %	72.2 %	86 %

Table 4.9: Results of Figure 2.11

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent						
		A	B_1	B_2	B_3	B_4	C	D
NN_1^1	If F_1 is Small Then	Nil	Very High	Very High	Nil	Nil	Very Low	Very Low
	If F_1 is Medium Then	Very High	More or Less High	More or Less High	More or Less High	More or Less High	Very Low	Very Low
	If F_1 is Big Then	Nil	Nil	Nil	Very High	Very High	Very Low	Very Low
NN_2^2	If F_2 is Small Then	Nil	Very High	Nil	Nil	Very High	Very Low	Very Low
	If F_2 is Medium Then	Very High	More or Less High	More or Less High	More or Less High	More or Less High	Very Low	very Low
	If F_2 is Big Then	Nil	Nil	Very High	Very High	Nil	Very Low	Very Low

1 learned in 100 iterations

2 learned in 100 iterations

Recognition Score

class A	class B	overall score
49 %	100 %	82 %

4.5.1 Experimental Results for Telugu Vowels

Table 4.10: Quantization of the feature space for feature1 of Telugu Vowels

	Small	Medium	Big
$200 \leq F_1 \leq 300$	1	0	0
$300 < F_1 \leq 400$	0.7	0.5	0
$400 < F_1 \leq 500$	0.4	1	0
$500 < F_1 \leq 600$	0.1	0.5	0.1
$600 < F_1 \leq 700$	0	0	0.4
$700 < F_1 \leq 800$	0	0	0.9
$800 < F_1 \leq 900$	0	0	1

Table 4.11: Quantization of the feature space for feature2 of Telugu Vowels

	Small	Medium	Big
$600 \leq F_2 \leq 750$	1	0	0
$750 < F_2 \leq 900$	0.7	0	0
$900 < F_2 \leq 1050$	0.4	0	0
$1050 < F_2 \leq 1200$	0.1	0.1	0
$1200 < F_2 \leq 1350$	0	0.5	0
$1350 < F_2 \leq 1500$	0	1	0
$1500 < F_2 \leq 1650$	0	0.5	0
$1650 < F_2 \leq 1800$	0	0.1	0.1
$1800 < F_2 \leq 1950$	0	0	0.2
$1950 < F_2 \leq 2100$	0	0	0.4
$2100 < F_2 \leq 2250$	0	0	0.6
$2250 < F_2 \leq 2400$	0	0	0.8
$2400 < F_2 \leq 2550$	0	0	0.9
$2550 < F_2 \leq 2600$	0	0	1

Table 4.12: Results of Telugu Vowels

Number of Hidden nodes = 3 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent							
		u	o	a	ə	e	i	g	h
NN_1^1	If F_1 is Small Then	Very High	Low	Nil	Nil	Low	Very High	Very Low	Very Low
	If F_1 is Medium Then	Very Low	Very High	Nil	High	Very High	Medium	Very Low	Very Low
	If F_1 is Big Then	Nil	Low	Very High	Very High	Medium	Nil	Very Low	Very Low
NN_2^2	If F_2 is Small Then	Very High	Very High	More or Less High	Very Low	Nil	Nil	Very Low	Very Low
	If F_2 is Medium Then	Very Low	Low	High	Very High	Very High	Nil	Very Low	Very Low
	If F_2 is Big Then	Nil	Nil	Nil	Very Low	Very High	Very High	Very Low	Very Low

1 learned in 218 iterations

2 learned in 471 iterations

Recognition Score

Hard Partitioning

class u	class o	class a	class ə	class e	class i	overall score
66.6 %	44.4 %	44 %	50 %	97.4 %	68.8 %	64.2 %

Fuzzy Partitioning

class u	class o	class a	class ə	class e	class i	overall score
66.6 %	55.5 %	92 %	100 %	100 %	90.6 %	86 %

4.5.2 Experimental Results for Bengali Vowels

Table 4.13: Quantization of the feature space for feature1 of Bengali Vowels

	Small	Medium	Big
$200 \leq F_1 \leq 250$	1	0	0
$250 < F_1 \leq 300$	0.7	0	0
$300 < F_1 \leq 350$	0.5	0	0
$350 < F_1 \leq 400$	0.3	0	0
$400 < F_1 \leq 450$	0.1	0.1	0
$450 < F_1 \leq 500$	0	0.5	0
$500 < F_1 \leq 550$	0	1	0
$550 < F_1 \leq 600$	0	0.5	0.1
$600 < F_1 \leq 650$	0	0.1	0.3
$650 < F_1 \leq 700$	0	0	0.5
$700 < F_1 \leq 750$	0	0	0.7
$750 < F_1 \leq 800$	0	0	1

Table 4.14: Quantization of the feature space for feature2 of Bengali Vowels

	Small	More or Less Medium	Medium	Big
$700 \leq F_2 \leq 800$	1	0	0	0
$800 < F_2 \leq 900$	0.7	0	0	0
$900 < F_2 \leq 1000$	0.5	0.1	0	0
$1000 < F_2 \leq 1100$	0.3	0.5	0	0
$1100 < F_2 \leq 1200$	0.1	1	0	0
$1200 < F_2 \leq 1300$	0	0.5	0	0
$1300 < F_2 \leq 1400$	0	0.1	0.1	0
$1400 < F_2 \leq 1500$	0	0	0.3	0
$1500 < F_2 \leq 1600$	0	0	0.5	0
$1600 < F_2 \leq 1700$	0	0	0.7	0
$1700 < F_2 \leq 1800$	0	0	1	0
$1800 < F_2 \leq 1900$	0	0	0.5	0.1
$1900 < F_2 \leq 2000$	0	0	0.3	0.3
$2000 < F_2 \leq 2100$	0	0	0.1	0.5
$2100 < F_2 \leq 2200$	0	0	0	0.7
$2200 < F_2 \leq 2300$	0	0	0	1

Table 4.15: Results of Bengali Vowels

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent								
		u	o	ə	a	ae	e	i	g	h
NN_1^1	If F_1 is Small Then	Very High	Medium	Nil	Nil	Nil	Medium	Very High	Very Low	Very Low
	If F_1 is Medium Then	Nil	Very High	Medium	Very Low	Very Low	More or Less High	Nil	Very Low	Very Low
	If F_1 is Big Then	Nil	Very Low	High	Very High	Very High	Nil	Nil	Very Low	Very Low
NN_2^2	If F_2 is Small Then	High	Very High	Very High	Very Low	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is More or Less Medium Then	Very High	Very High	Very High	High	Very Low	Nil	Nil	Very Low	Very Low
	If F_2 is Medium Then	Very Low	Nil	Nil	Very Low	Very High	High	Nil	Very Low	Very Low
	If F_2 is Big Then	Nil	Nil	Nil	Nil	Very Low	More or Less High	Very High	Very Low	Very Low

1 learned in 168 iterations

2 learned in 164 iterations

Recognition Score

Hard Partitioning

class u	class o	class ə	class a	class ae	class e	class i	overall score
74.6 %	92.9 %	32.8 %	42.8 %	76 %	96.3 %	63.9 %	70.2 %

Fuzzy Partitioning

class u	class o	class ə	class a	class ae	class e	class i	overall score
100 %	100 %	94.3 %	85.7 %	76 %	100 %	91.8 %	91.8 %

4.5.3 Experimental Results for Assamese vowels

Table 4.16: Quantization of the feature space for feature1 of Assamese Vowels

	Small	More or less Medium	Medium	Big
$200 \leq F_1 \leq 250$	1	0	0	0
$250 < F_1 \leq 300$	0.7	0	0	0
$300 < F_1 \leq 350$	0.5	0	0	0
$350 < F_1 \leq 400$	0.3	0.1	0	0
$400 < F_1 \leq 450$	0.1	0.5	0	0
$450 < F_1 \leq 500$	0	1	0.1	0
$500 < F_1 \leq 550$	0	0.5	0.5	0
$550 < F_1 \leq 600$	0	0.1	1	0
$600 < F_1 \leq 650$	0	0	0.5	0
$650 < F_1 \leq 700$	0	0	0.1	0.1
$700 < F_1 \leq 750$	0	0	0	0.3
$750 < F_1 \leq 800$	0	0	0	0.5
$800 < F_1 \leq 850$	0	0	0	0.7
$850 < F_1 \leq 900$	0	0	0	1

Table 4.17 : Quantization of the feature space for feature2 of Assamese Vowels

	Small	More or less Medium	Medium	Very Medium	Big
$600 \leq F_2 \leq 700$	1	0	0	0	0
$700 < F_2 \leq 800$	0.7	0.1	0	0	0
$800 < F_2 \leq 900$	0.5	0.5	0	0	0
$900 < F_2 \leq 1000$	0.3	1	0.1	0	0
$1000 < F_2 \leq 1100$	0.1	0.5	0.5	0	0
$1100 < F_2 \leq 1200$	0	0.1	1	0	0
$1200 < F_2 \leq 1300$	0	0	0.5	0	0
$1300 < F_2 \leq 1400$	0	0	0.1	0.1	0
$1400 < F_2 \leq 1500$	0	0	0	0.5	0
$1500 < F_2 \leq 1600$	0	0	0	1	0
$1600 < F_2 \leq 1700$	0	0	0	0.5	0
$1700 < F_2 \leq 1800$	0	0	0	0.1	0.1
$1800 < F_2 \leq 1900$	0	0	0	0	0.2
$1900 < F_2 \leq 2000$	0	0	0	0	0.4
$2000 < F_2 \leq 2100$	0	0	0	0	0.5
$2100 < F_2 \leq 2200$	0	0	0	0	0.6
$2200 < F_2 \leq 2300$	0	0	0	0	0.7
$2300 < F_2 \leq 2400$	0	0	0	0	0.8
$2400 < F_2 \leq 2500$	0	0	0	0	0.9
$2500 < F_2 \leq 2600$	0	0	0	0	1

Table 4.18: Results of Assamese Vowels

Number of Hidden nodes = 9 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent									
		u	ū	o	ə	a	ae	e	i	g	h
NN_1^1	If F_1 is Small Then	Very High	More or Less Low	Low	Nil	Nil	Nil	Very Low	Very High	Very Low	Very Low
	If F_1 is More or Less Medium Then	Nil	More or Less Low	Very High	More or Less Low	Nil	More or Less Low	Very High	Low	Very Low	Very Low
	If F_1 is Medium Then	Nil	Nil	High	High	Low	High	Medium	Nil	Very Low	Very Low
	If F_1 is Big Then	Nil	Nil	Nil	Medium	Very High	Medium	Nil	Nil	Very Low	Very Low
NN_2^2	If F_2 is Small Then	Very High	Very High	High	Very Low	Nil	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is More or Less Medium Then	High	Very High	Medium	High	Very Low	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is Medium Then	Nil	Nil	Low	Very High	Medium	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is Very Medium Then	Nil	Nil	Nil	Nil	High	Medium	Nil	Nil	Very Low	Very Low
	If F_2 is Big Then	Nil	Nil	Nil	Nil	Low	Medium	Very High	Very Low	Very Low	Very Low

1 learned in 128 iterations

2 learned in 142 iterations

Recognition Score

Hard Partitioning

class u	class ū	class o	class ə	class a	class ae	class e	class i	overall score
87.5 %	0 %	73.3 %	75 %	83.3 %	45.4 %	47 %	63.2 %	64.6 %

Recognition Score

Fuzzy Partitioning

class u	class ū	class o	class ə	class a	class ae	class e	class i	overall score
100 %	50 %	100 %	100 %	91.6 %	90.9 %	70.5 %	89.4 %	88.7 %

4.6 Comparison of the performance of the proposed method with that of some existing methods

To judge the effectiveness of the proposed method for vowel classification with respect to some of the existing methods, we provide a comparative study in Tables 4.19 - 4.25. In these Tables, when we mention recognition score of the proposed method, we mean to say scores under fuzzy partitioning which is the essential feature of the present approach. In case fuzzy partitioning is not required depending on the overlap in the feature space (see Table 4.6, 4.8 & 4.9), we denote the scores with an asterisk. For comparison, we consider conventional backpropagation, Bayesian classifier and McClelland's method [11].

Table 4.19: Comparative study for Figure 2.8

	Recognition Scores in %						
Methods	class A	class B	class C	class D	class E	class F	Overall
Conventional MLP	59.4	92.3	87.3	86.5	84.1	98.2	84.4
Bayesian Classifier	93.8	94.7	49.4	55.1	79.4	79.2	78
McClelland's method	84.4	95.7	86	88.7	77.4	95.6	87.5
NeuroFuzzy	93.1	93.1	93.6	41	100	97.3	88.4

Table 4.20: Comparative study for Figure 2.9

	Recognition Scores in %			
Methods	class A	class B	class C	Overall
Conventional MLP	54.7	81.4	98	69.1
Bayesian Classifier	100	71.1	72	86.9
McClelland's method	42.8	89.6	100	64.1
*NeuroFuzzy**	94.4	61.8	92	87.4

Table 4.21: Comparative study for Figure 2.10

	Recognition Scores in %		
Methods	class A	class B	Overall
Conventional MLP	67.5	98	81.8
Bayesian Classifier	90.3	83.1	88.4
McClelland's method	69.3	99	83.2
*NeuroFuzzy**	98.2	72.2	86

Table 4.22: Comparative study for Figure 2.11

Methods	Recognition Scores in %		
	class A	class B	Overall
Conventional MLP	100	78.5	84.1
Bayesian Classifier	48.2	67.4	60.8
McClelland's method	100	78.7	84.7
*NeuroFuzzy**	49	100	82

Table 4.23: Comparative study for Figure 2.19

Methods	Recognition Scores in %						
	class u	class o	class a	class ə	class e	class i	Overall
Conventional MLP	86.7	79	70.6	57.7	81.3	56	73.1
Bayesian Classifier	84.3	79	85.7	75	93.8	54.5	80.3
McClelland's method	77.1	91.6	75.2	79.3	74.1	60.6	79.2
NeuroFuzzy	66.6	55.5	92	100	100	90.6	86

Table 4.24: Comparative study for Figure 2.21

Methods	Recognition Scores in %							
	class u	class o	class ə	class a	class	class e	class i	Overall
Conventional MLP	66.6	83.3	72.3	88.5	91.6	91.4	85.7	82.2
Bayesian Classifier	84.1	75	83.1	88.5	97.2	90.1	91.1	86.1
McClelland's method	66.6	83.3	96.9	85.2	100	97.5	78.6	86
NeuroFuzzy	100	100	94.3	85.7	76	100	91.8	91.8

Table 4.25: Comparative study for Figure 2.23

Methods	Recognition Scores in %								
	class u	class ū	class o	class ə	class a	class e	class e	class i	Overall
Conventional MLP	50	100	42.9	100	100	45.4	58.8	40	68.1
Bayesian Classifier	87.5	100	92.8	100	95.8	27.3	82.3	70	82.7
McClelland's method	75	100	64.2	100	95.8	36.4	94.1	10	74.5
NeuroFuzzy	100	50	100	100	91.6	90.9	70.5	80.4	88.7

4.7 Management of Uncertainty in Pattern Classification: an ad-hoc approach

At last we consider the aspect of the management of uncertainty in pattern classification. One kind of uncertainty in pattern classification which is tackled by fuzzy reasoning method as stated earlier, occurs due to the variations of feature values caused by external noise, errors in the sensors etc. In addition to this stated uncertainty, we have realized another kind of uncertainty which occurs when we consider classification of same vowels (of some Indian language) uttered by different speakers (see the discussion of Appendix A) and which can not be handled by fuzzy reasoning alone. To tackle such uncertainties which essentially occurs due to the variations of the vocal tract dynamics and tongue position of individuals we use the concept of fuzzy masking (see Figure 4.2) which is experimentally designed in the present thesis and for which some mathematical basis is yet to be developed.

The concept of fuzzy masking essentially replaces the concept of fuzzy singleton (see section 4.0) for ultimate classification of nonfuzzy features of the uttered vowels. Generally, the fuzzy **If Then** rules for the classifier are generated looking at the approximate location of the seed points of each cluster for each vowel and the approximate spread of vowels in each cluster. When the neural network is trained (see section 4.2) by the rules so generated, we can use fuzzy masking as shown in Figure 4.2(a) for classification of vowels based upon their nonfuzzy features. As shown in Figure 4.2(a,b), there is no difference in the defuzzified value obtained either from fuzzy singleton or from fuzzy masking of a feature to be classified. Only when we use triangular masking, we take into consideration the possible spread of the said feature due to variation of vocal tract dynamics and tongue position of individual. In particular application of vowel classification, we make vocal tract dynamics and tongue position of individual responsible for total shift of the same pattern (vowel) class produced by different individuals (mechanism). Similar situation may occur in general when same pattern classes are generated by nonidentical mechanisms. Study of such uncertainties in pattern classification based on the statistical data given in Appendix B is a part of the speech recognition research of Electronics and Communication Sciences Unit of Indian Statistical Institute. Here, in this section, we simply report our primary findings which we obtain through some

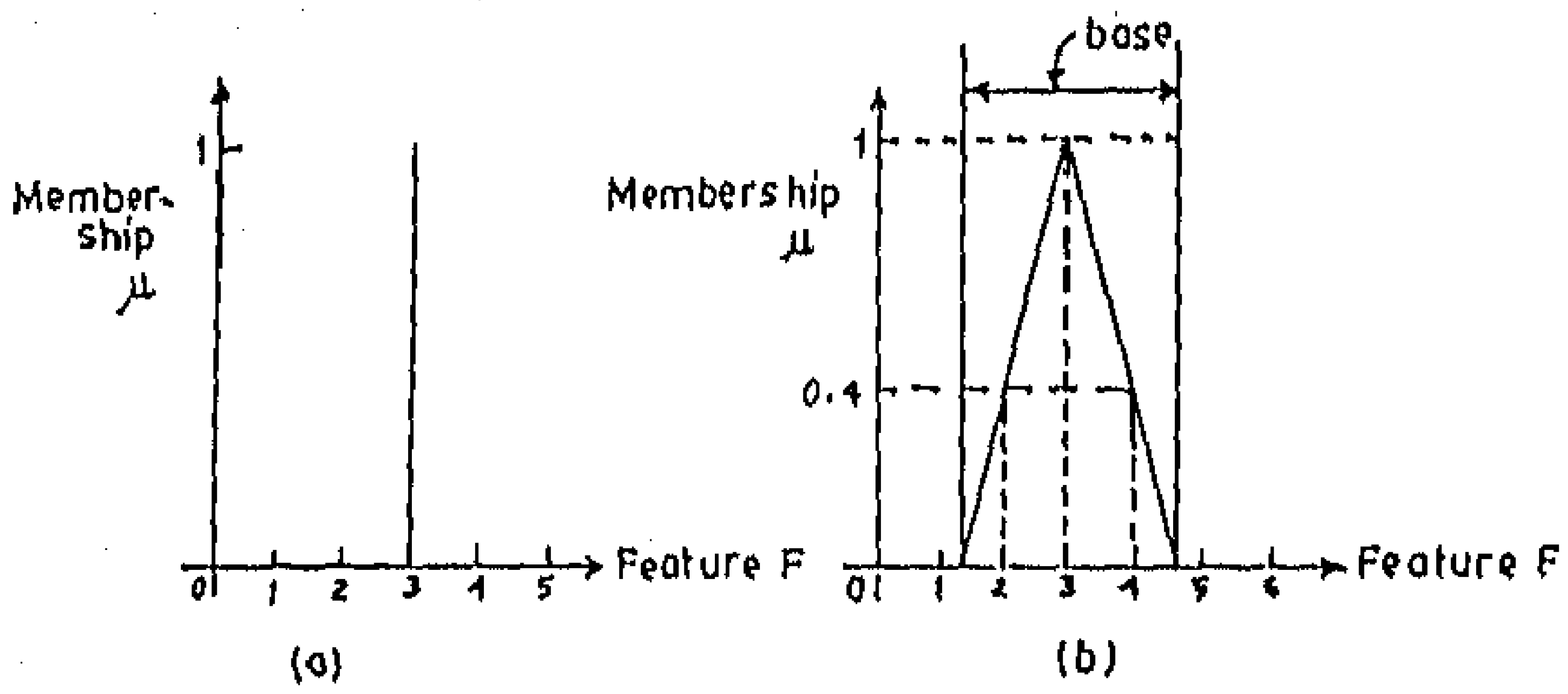


Fig. 4.2: Fuzzy Masking

Note: The possibility of the sensed signal to become 3 is 1. Whereas, the possibility of the sensed signal to become 2 is 0.4 etc. This variation in possibility of the sensed signal is to consider the uncertainty in the measurement. Thus, a particular sensed signal, say 3, is fuzzified by fuzzy masking. Depending upon data, other types of masking is possible. In the present case, for a class of data, we have considered a particular type of masking. This is basically a design heuristic.

experimental studies.

4.7.1 Management of uncertainty in vowel classification

In this experiment, first we consider the distribution of Bengali vowels in F_1 - F_2 plane for three different speakers. Such distributions are shown in Figure 4.3, Figure 4.4 and Figure 4.5. Table B.1 in Appendix B represents the informantwise variances of the first two formant frequencies for different vowels. In Figure 4.3, 4.4 and 4.5, the main concentration of each class of vowel is enclosed in a solid boundary with an aim to give a rough estimate/ impression about the degree of overlap among vowel classes, the shift of the center point (seed point) and the boundary of each class. Now, it is obvious from Figure 4.3, 4.4 and 4.5 that each vowel class has been shifted. Under such situation, to generate a classifier, our basic design steps remain same as stated in section 4.3 except for the concept of fuzzy masking which is used instead of fuzzy singleton, for classification of nonfuzzy test features of the vowels uttered by the same speaker (see Figure 4.3). The obtained results are reported in Table 4.26. Now, to test the design with the concept of fuzzy masking, we apply the classifier, which was originally trained and tested for the data of Figure 4.3, on the test data of Figure 4.4 and 4.5. The results obtained are reported in Table 4.27 and 4.28. From the obtained results, we understand that the neurofuzzy classifier associated with fuzzy masking is speaker independent. The uncertainty of the first kind, that is variation of features due to the error of the sensors, noise etc. can be well taken care of by the inherent property of fuzzy reasoning (i.e. Generalized Modus Ponens) and the uncertainty of the second kind that is the total shift of the individual pattern class at different observations can be well taken care of by fuzzy masking whose spread of the base should be guided by the variations of the feature as stated in Appendix B. In this thesis, we have selected the spread of the base experimentally and the theoretical basis of such situation is yet to be developed.

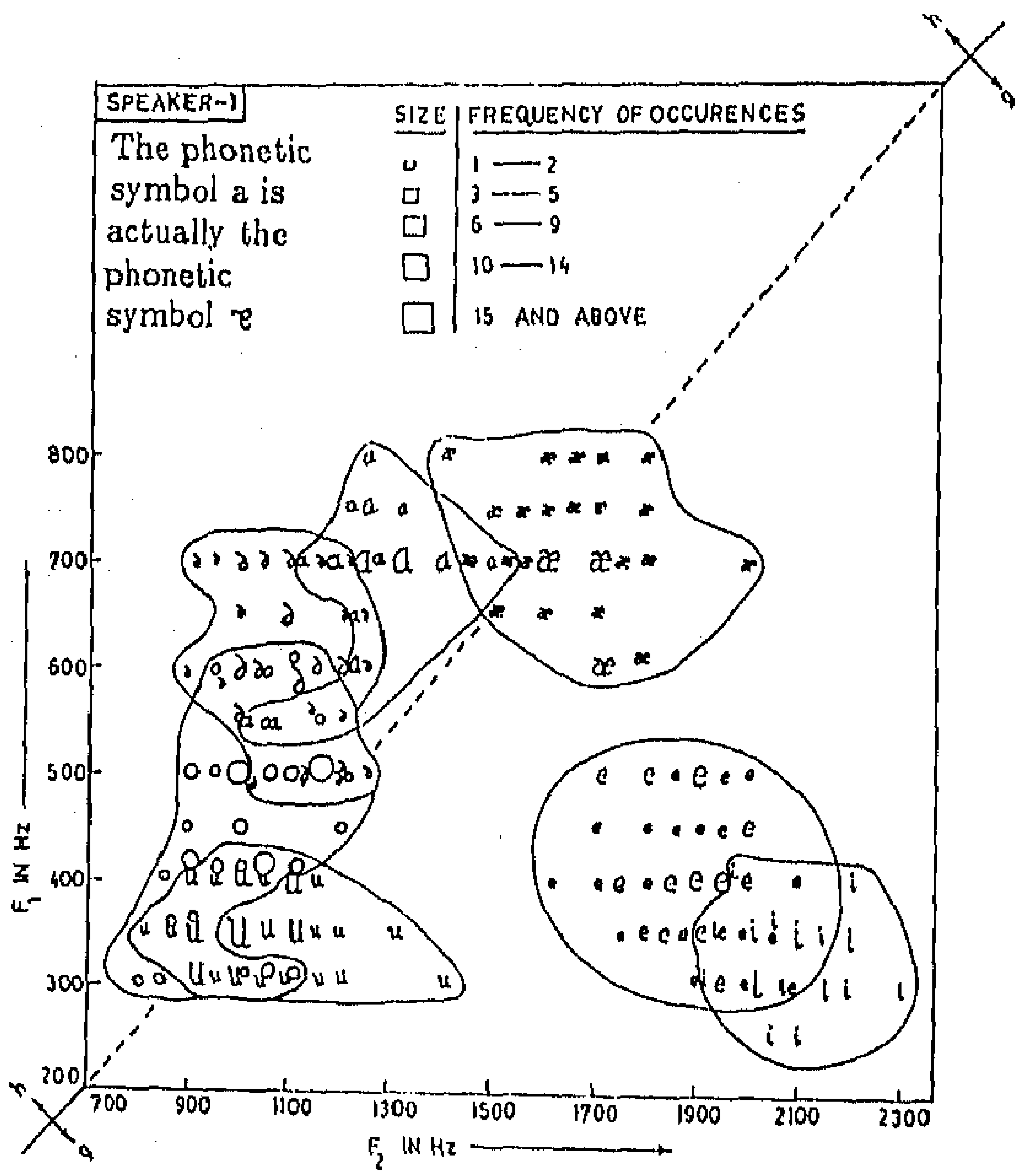


Fig. 4.3: Bengali vowels in the F_1 - F_2 plane

Table 4.26: Results of Bengali vowels in the $F_1 - F_2$ plane (speaker1)

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent								
		u	o	ə	a	ae	e	i	g	h
NN_1	If F_1 is Small Then	Very High	Medium	Nil	Nil	Nil	Medium	Very High	Very Low	Very Low
	If F_1 is Medium Then	Nil	Very High	Medium	Very Low	Very Low	More or Less High	Nil	Very Low	Very Low
	If F_1 is Big Then	Nil	Very Low	High	Very High	Very High	Nil	Nil	Very Low	Very Low
NN_2	If F_2 is Small Then	High	Very High	Very High	Very Low	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is More or Less Medium Then	Very High	Very High	Very High	High	Very Low	Nil	Nil	Very Low	Very Low
	If F_2 is Medium Then	Very Low	Nil	Nil	Very Low	Very High	High	Nil	Very Low	Very Low
	If F_2 is Big Then	Nil	Nil	Nil	Nil	Very Low	More or Less High	Very High	Very Low	Very Low

Fuzzy masking used over the nonfuzzy domain of the features with maximum membership function at a particular instance { .2 .5 .7 1 .7 .5 .2 }

Recognition Score

Hard Partitioning

class u	class o	classə	class a	class ae	class e	class i	overall score
77.3 %	91.6 %	36 %	0 %	88.8 %	93.8 %	69.6 %	65.3 %

Fuzzy Partitioning

class u	class o	classə	class a	class ae	class e	class i	overall score
100 %	100 %	100 %	93.4 %	100 %	100 %	100 %	98.8 %

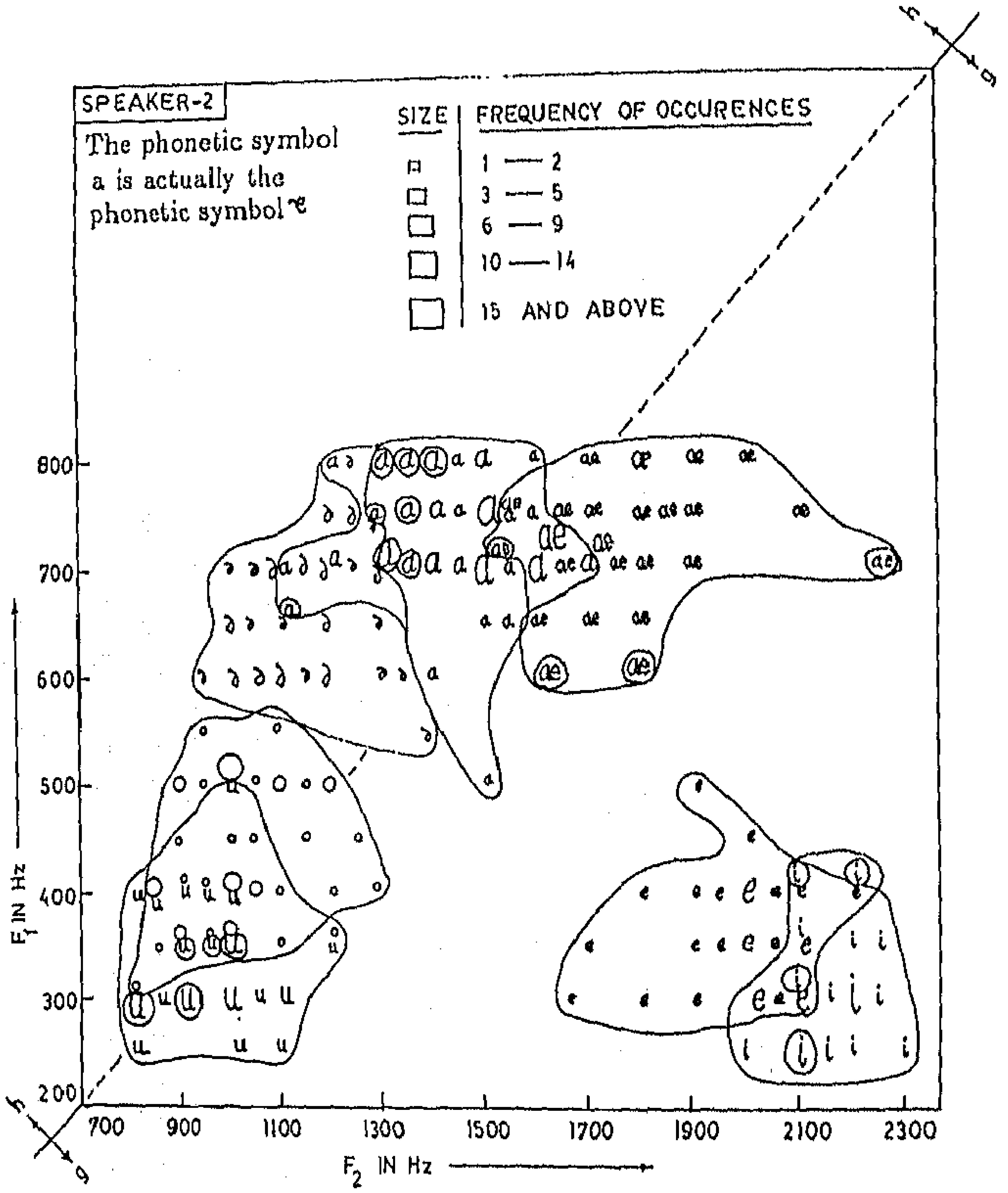


Fig. 4.4: Bengali vowels in the F_1 - F_2 plane

Key: The vowels in the distorted zones of the pattern space as mentioned in Appendix A are correctly classified using fuzzy masking and indicated by the circles

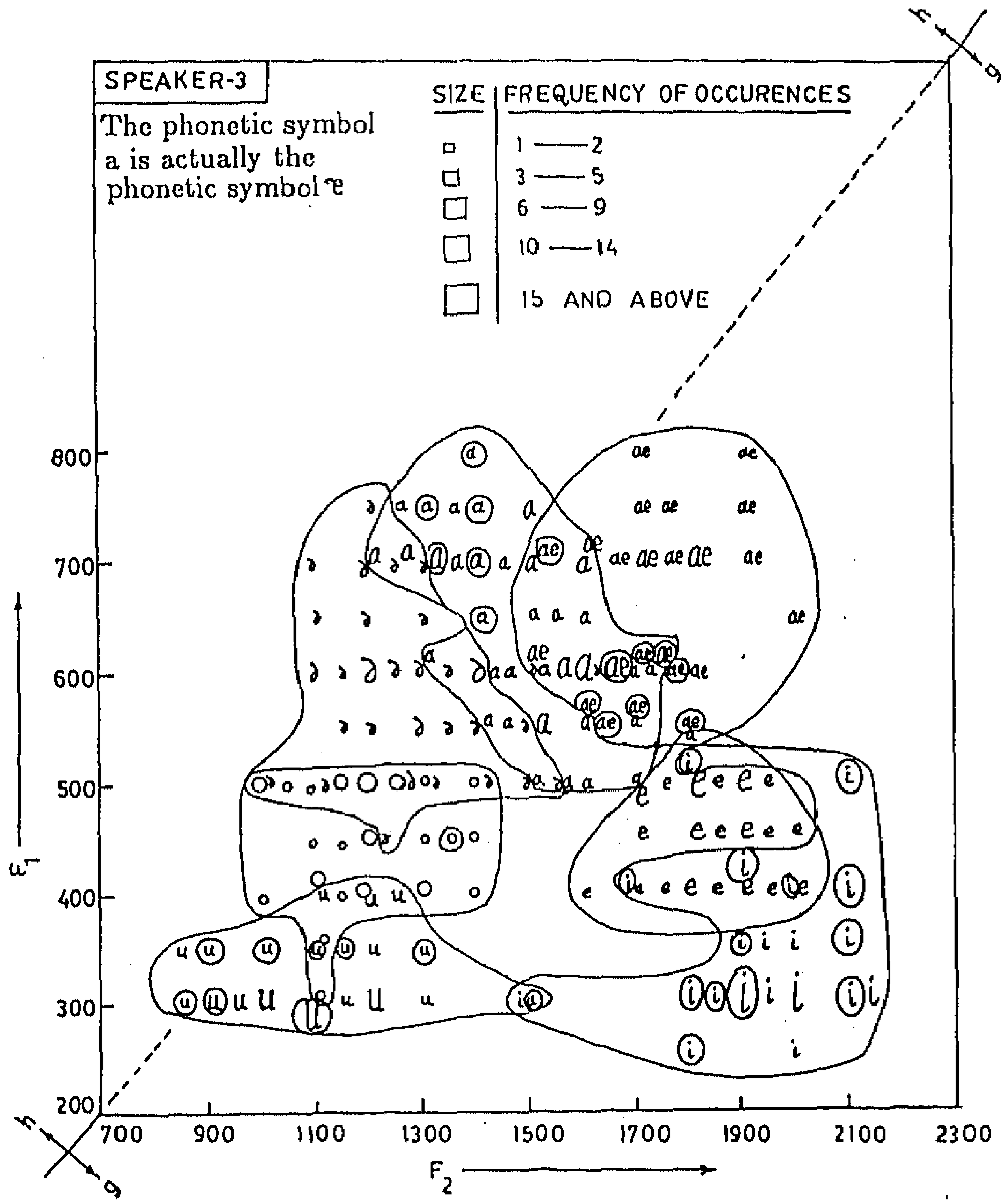


Fig. 4.5: Bengali vowels in the F_1 - $F2$ plane

Key: Same as Figure 4.4

Table 4.27: Results of Bengali vowels in the $F_1 - F_2$ plane (speaker2)

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

The rules used are the set of rules used for the classification of the data set of Fig. 2.15

Fuzzy masking used over the nonfuzzy domain of the features with maximum membership function at a particular instance

$\{ .2\ .5\ .7\ 1\ .7\ .5\ .2 \}$

Recognition Score

Hard Partitioning

class u	class o	class ɔ	class a	class ae	class e	class i	overall score
85.5 %	82.2 %	64 %	0 %	83.3 %	62 %	98.2 %	57.4 %

Fuzzy Partitioning

class u	class o	class ɔ	class a	class ae	class e	class i	overall score
100 %	100 %	100 %	41 %	100 %	100 %	100 %	83.9 %

Table 4.28: Results of Bengali vowels in the $F_1 - F_2$ plane (speaker3)

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

The rules used are the set of rules used for the classification of the data set of Fig. 2.15

Fuzzy masking used over the nonfuzzy domain of the features with maximum membership function at a particular instance

$\{ .2\ .5\ .7\ 1\ .7\ .5\ .2 \}$

Recognition Score

Hard Partitioning

class u	class o	classə	class a	class ae	class e	class i	overall score
96.4 %	95.9 %	15.9 %	0 %	57.4 %	100 %	16.2 %	43.3 %

Fuzzy Partitioning

class u	class o	classə	class a	class ae	class e	class i	overall score
100 %	100 %	98.5 %	23.7 %	100 %	100 %	100 %	82.3 %

4.8 Conclusion

To tackle pattern classification problem on $\Re^2$ we consider the new interpretation of MFI (i.e. Equation (2.16) of Chapter 2) and realize that new interpretation through backpropagation type neural network. This approach can easily be extended for pattern classification on $\Re^n$, n > 2. To test the validity of the proposed scheme we first consider a set of synthetic data and then apply the newly proposed scheme for the vowel classification problems of three Indian languages. In all the cases, we have obtained very good results. We also compare the performance of the newly proposed method with that of the existing methods like conventional backpropagation, Bayesian classifier and McClelland's method. The comparative results are also promising. For management of uncertainty we propose a simple scheme, which is experimentally developed and for which some theoretical basis is yet to be generated. We propose two types of fuzzification (fuzzy singleton and fuzzy masking) of the nonfuzzy test features for classification / recognition of patterns. From now onwards, we may use either fuzzy singleton or fuzzy masking depending upon the need of the problem. Also note that the MFR based on the new interpretation of MFI (i.e. Equation (2.16) of Chapter 2) does not suffer from the curse of high dimension (i.e. a pattern having more than 2 features) as we have seen in Chapter 3 which is based on the conventional interpretation of MFI (i.e. (2.14(a))). Thus we have developed two different methodologies based on two different interpretations of MFI (i.e. Equation (2.14(a)), (2.16) of Chapter 2) for fusion of the first kind for the classification of the patterns. Instead of doing any competition between these two approaches of fusion, we may use either of these two depending upon the complexity of the problem. That means if we have patterns having two features we may consider either of the methods for classification. Whereas if we have patterns of having features more than two, we prefer interpretation (2.16) of a MFI. In the next chapter, we will demonstrate that the basic principle of fusion technology which is developed in this chapter, is equally applicable for occluded object (two dimensional objects) recognition problem.

Chapter 5

Fusion technology for occluded object (2-dimensional object) recognition

To tackle the problem of occluded object recognition in this chapter [83], we consider Equation (2.16) of a MFI as discussed in Section 2.9 of Chapter 2. That means we consider the fusion technology which we have developed in the last chapter (see Section 4.2) and applied for pattern classification problems. Thus a unified framework has been developed for *pattern classification* and occluded object recognition. At the learning stage of the neural network fuzzy linguistic statements are used. Once learned, the nonfuzzy features of an occluded object can be classified. At the time of classification of the nonfuzzy features of an occluded object we use the concept of fuzzy singleton. An effective approach to recognize an unknown scene which consists of a set of occluded objects is to detect a number of significant (local) features on the boundary of the unknown scene. Thus *the major problems* fall into the selection of the appropriate set of features (local) for representing the object in the training stage, as well as in the detection of these features in the recognition process. The features should be invariant to scale, orientation and minor distortions in boundary shape. The performance of the proposed scheme is tested through several examples.

The object recognition scheme (using MFR), proposed in this chapter [83] is a model based system in which recognition involves matching the input image with a set of predefined models of objects. In such a system the known objects are precompiled, creating a model database and this database is used to recognize objects in an image scene.

Existing object recognition methods can be categorized as either global or local in nature. Global methods are based on global features of the boundary or of an equivalent representation. Such techniques are the Fourier descriptors, the Moments and methods based on Autoregressive models. Local methods use local features such as critical points or holes and corners. They perform very well in the presence of noise, distortion or partial occlusion since such effects on an isolated region of the contour alter only the local features associated with that region, leaving all other local features unaffected. However, the choice of representative local features is not trivial and the recognition process based on local features is more computationally intensive and time consuming. On the other hand, global methods have disadvantage that a small distortion in a portion of a boundary of an object will result in changes to all global features. In the present chapter, we use internal angles and curvatures of the significant points on the boundary of an object as local features for model based recognition. In our experimental study we have seen the said features (i.e. the internal angle and curvature of the significant points) provide reasonable separation of different classes of objects at the learning stage of the classifier. Depending upon the need of the problem we may choose any other suitable pair of features.

The basic motivation behind the use of the method of MFR is to take care of the uncertainties in the local features of the objects under different orientations and noisy environment. Thus invariance property of the local features of the object holds good under the assumption that significant changes in internal angles and curvatures are invariant under rotation and translation. In our model based object recognition scheme, object models are represented into a model database, using fuzzy **If Then** rules. Generally, increasing the number of object models in the model database greatly increases the computational complexity and the time requirements of the system. However, implementation of a model based object recognition scheme using backpropagation type neural network seems to be very attractive. First of all, neural network provides its own way to represent the knowledge (in terms of data which are basically the local features) that it stores [14]. In addition, the complexity and the computational burden increase very slowly as the number of data models increases. In general, the performance of the neural network is very good but it takes lot of time for learning. However, in our case, learning time is comparatively less because at the time of learning, instead of feeding individual data, we feed a cluster of data occupied by the antecedent part of the fuzzy **If Then** rules (see item number 5 on advantages discussed in the last paragraph of Section 2.1 of Chapter 2). Thus, the task of learning of the neural network about the total data set is completed much earlier.

5.1 Statement of the problem

Neural networks are generally robust static pattern classifiers. But, they are not effective in classifying patterns with inherent variations. In order to compensate for variations resulting from occlusion, a multilayer neural network system which realizes the new interpretation of MFI is proposed. The occluded object recognition approach is basically similar to the decision theoretic approach to pattern recognition.

In the MFR approach to occluded object recognition, each element of the (local) pattern vector / feature vector is represented by the fuzzy linguistic variable instead of a real number. For instance, suppose we have a (2×1) pattern vector /feature vector $F = (F_1, F_2)^T$, T is transpose where F_1 is the internal angle at a significant point on the boundary of an object and F_2 is the curvature at that significant point. In the decision theoretic approach to pattern recognition, F_1 and F_2 are represented by real numbers. Whereas in the MFR approach to occluded object recognition F_1 and F_2 are local features represented by the fuzzy linguistic variables, e.g. F_1 is **small** and F_2 is **medium**. The elements of the pattern vector / feature vector which are represented by fuzzy linguistic variables are represented by fuzzy sets. Thus, in the present approach, instead of a single pattern, a population of patterns is being represented by the fuzzy pattern vector / feature vector $\vec{F_{fj}}$ (see Figure 2.26(a) of Chapter 2). These elements of the pattern vector / feature vector actually constitute the antecedent part of the multidimensional fuzzy implication (MFI). The consequent part of the MFI represents the possibility of occurrence of different classes of patterns in the pattern space (see Equation (2.16) of Chapter 2).

The process of recognition is usually divided into two steps, learning and classification. The main stages in the learning process are feature extraction, feature selection, clustering and determination of the appropriate fuzzy **If Then** rules which constitutes a decision function.

The main stages in classification are : extraction of a selected set of features, application of **If Then** rules and decision making based on the results of the application of the **If Then** rules. In the classification process, first an unknown scene is presented to the system, then a set of predetermined features are extracted from the scene. Finally, a set of **If Then** rules determines the possibility of occurrence of each object in the scene.

The neural network configuration for occluded object recognition is same as Figure 4.1 of Chapter 4.

5.2 Formulation of the problem

As our basic aim is to recognize occluded objects from a given scene, we simply demonstrate clean object models and scene images; but all these model objects and image scenes are directly sensed by camera and necessary preprocessing are done to achieve such clean representation of model objects and scene images. As we used standard techniques for image processing we are not restating these methods in this thesis.

5.2.1 Local feature extraction

One class of techniques dealing with the problem of object recognition uses the global object features, such as area and perimeter of boundary curve, centroid and shape moments. However, this type of approach is not suited for the recognition of partially occluded objects whose global features are totally destroyed. Some of the local object attributes may be subjected to change, such as position and slope, but some of them will remain invariant, such as curvature and internal angle of a curved object. The invariance property of such object features is used to generate the $F_1 - F_2$ pattern space. In our case, we have considered the internal angle of significant points on a curve as the feature F_1 and its corresponding curvature as the feature F_2. Now, curvature is a measure of the rate of change of orientation per unit arc length. The geometric interpretation for the curvature is depicted in Figure 5.1. Let P be a point on a curve, T be the tangent at that point and A be a neighboring point on the curve. Let α denote the angle between the line AP and T, and $|\hat{AB}|$ the arc length between A and B. The curvature κ at P is the ratio $\alpha / |\hat{AB}|$.

If we deal with digital curves, it is not immediately clear how to define a discrete curvature and internal angle. Suppose, a digital curve is defined as a sequence of integer coordinate points $p_1, \cdots, p_n$ where p_{i+1} is a neighbor of p_i, $1 \leq i \leq n$. Now, on a digital curve, successive points can only differ in slope by a multiple of 45 degree. Hence, small changes in slope are impossible to define . This difficulty can be reduced using a smoothed slope measurement e.g. defining the slope at p_i as $(y_{i+k} - y_i) \ / \ (x_{i+k} - x_i)$ for some $k \succ 1$. It is not obvious which value of k to choose in a given application. This expression becomes unstable as $(x_{i+k} - x_i)$ approaches zero, therefore direct differences in this discrete analog to slope can not be used as estimators of point curvature.

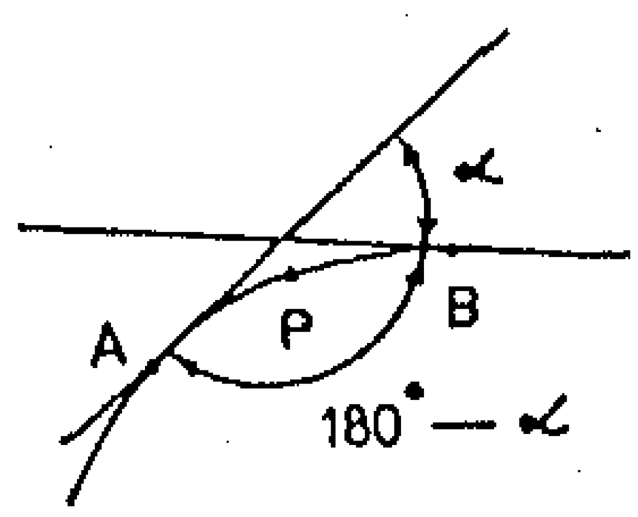


Fig. 5.1: The internal angle at point P, with respect to its neighborhood, is 180° - α

An early model [89] for points of inflection on a digital curve is as follows. The k-vectors at p_i are defined as

$$\underline{a_{ik}} = \ (x_i - x_{i+k}, y_i - y_{i+k}) \tag{5.1}$$

$$\underline{b_{ik}} = \ (x_i - x_{i-k}, y_i - y_{i-k}) \tag{5.2}$$

and the k-cosine at p_i as

$$C_{ik} = \ (\underline{a_{ik}} \bullet \underline{b_{ik}}) / \mid \underline{a_{ik}} \mid\mid \underline{b_{ik}} \mid \tag{5.3}$$

This C_{ik} is the cosine of the angle between $\underline{a_{ik}}$ and $\underline{b_{ik}}$. Thus, $-1 \leq C_{ik} \leq 1$, where C_{ik} is close to 1 if $\underline{a_{ik}}$ and $\underline{b_{ik}}$ make an angle near 0 degree , and C_{ik} is close to -1 if $\underline{a_{ik}}$

and $\underline{b_{ik}}$ make an angle near 180 degree ; in other words, C_{ik} is larger when the curve is turning rapidly and smaller when the curve is relatively straight.

The smoothing factor k at each point p_i of the curve can be selected as follows. At each point p_i, compute C_{ik} for $1 \leq k \leq m$, for some fixed m. In our case, we have considered m to be 8. A size h and smoothed k-cosine C_{ih} is assigned to each point such that,

$$C_{im} \prec C_{im-1} \prec \cdots \prec C_{ih} \not\prec C_{ih-1} \tag{5.4}$$

C_{ih} is considered as the cosine at p_i and p_i is an 'angle' point if C_{ih} is a local maxima in the sense that $| i - j | \leq h/2$ implies $C_{ih} \geq C_{jh}$ with C_{i0} = -1 to insure that there always is such an h between m and 1. Now, C_{ih} is the cosine at p_i is denoted by c_i which is a measure of the internal angle at p_i. Hence, the curvature at p_i is defined as $(\pi - cos^{-1}(c_i))/2h$ (see Figure 5.1).

5.2.2 Process of fuzzification

At the learning stage, first of all, we discretize the universe of discourse of the features F_1 (internal angle) and F_2 (curvature). Discretization is often referred to as quantization which discretizes a universe into a certain number of segments (quantization levels). Each segment is labeled as a generic element and forms a discrete universe. A fuzzy set is then defined by assigning grade of membership values to each generic element of the new discrete universe (see Table 5.4(a,b)). At the classification stage, the selected features are fuzzified using the concept of fuzzy singleton.

5.2.3 Assignment of the membership function to the consequent part of the If - Then rules

The membership function of the consequent part of the **If Then** rule represents the possibility of occurrence of different classes of patterns in the pattern space (see Figure 5.4). The membership function of the consequent part of the **If Then** rules are calculated as we have done in Section 4.3 of Chapter 4. We essentially take help of a Query Table (QT) which is exactly similar to Table 4.1 of Chapter 4.

5.2.4 Process of defuzzification

At the time of taking nonfuzzy decision out of the fuzzy classification (i.e. defuzzification) we can go by selecting the class having the highest membership value. In case of tie situation, which normally occurs for the features lying in the overlapped zone (see Figure 5.17), we have to state the equal possibility of a pattern to belong to both the classes. And such a conclusion is quite natural which normally does not exist in conventional classification approach. In some cases, patterns in the overlapped zones are classified with **"almost equal"** possibility of occurrence for more than one class. If such situation is treated as tie situation mentioned above, we have to select an appropriate threshold which entirely depends on the need of the problem.

Thus, from the graded consequence, when we select a single class having the highest membership value, we consider the hard partitioning of the pattern space. Whereas, from the graded consequence, when we consider multiple classifications occurring in the overlapped zone, we consider the fuzzy partitioning of the pattern space.

5.2.5 Generation of the model based object recognition scheme

The local features, i.e. the internal angle and the curvature of the model objects are calculated at different significant points. Subsequently, such values are plotted on F_1 - F_2 plane where F_1 axis represents the internal angle and F_2 axis represents the curvature (see Figure 5.4). Universes of feature axes F_1 and F_2 are partitioned by the elements of the term set and several fuzzy **IF Then** rules (see Table 5.5(a,b)) are constructed as we have done in Section 4.1 and 4.2 of Chapter 4 to capture the informations about the local features of the model objects. Thus, the fuzzy **If Then** rules form the knowledge base of the model based object recognition. As, this knowledge base is formed by the fuzzy **If Then** rules, it has the inherent capacity for management of uncertainties in local features due to the presence of noise and other disturbances. Each rule (a DFI of Equation (2.16)) of the knowledge base has two components, one is the antecedent part and the other is the consequent part. The antecedent part of the rule is fed as input to the neural net and the consequent part represents the target value with which the individual output of the network is compared and using the generalized delta rule the network (see Figure 4.1 of Chapter 4) is trained. After the training period, the network acquires the knowledge stored in the knowledge base and becomes intelligent enough to deal with new situations. Thus, after the training period, if the local features (i.e. the internal angle and the curvature) of an unknown scene, which may consist of the model objects, are calculated and injected as fuzzy singleton to the input of the network, then the network can classify which particular feature belongs to which particular model class.

The number of inputs of the network depends on the fuzzification process discussed in section 5.2.2 and the number of outputs equals the number of model objects to be classified.

As the scene consists of the model objects which are placed in different orientations and which are partially occluded by each other, it is quite obvious that the number of local features visible in each partially occluded object in the scene is less than the number of local features visible in the corresponding model object. In our recognition scheme, we assume that under occlusion, at least "50 % of the local features $\pm\sigma$" of model objects are visible in the scene. That means, maximum 50 % (approximately) occlusion of the model object is tolerable to the recognition scheme. At maximum occlusion, under different kind of uncertainties, σ is the spread around the expected number of features.

Note that the term "expected number" is not in any statistical sense; but in the following sense.

Let a model object has total 10 features. Under maximum occlusion (i.e. under 50 % occlusion) 5 features are expected to be visible in a scene consisting of that model and other model objects. During recognition process out of the 5 features of the said model either

i) some of the features (say, one or two) may not be visible due to some error in computation or noise or

ii) some additional features (say one or two) which are newly generated features due to overlap of two objects etc. may be recognized as features of the said model.

Thus, the concept of the spread σ around the expected number of features is generated.

After the recognition of the local features of the scene if the recognition scheme produces the number of model features visible in the scene (by checking the highest output of the network; in case of tie situation break the tie by taking arbitrary decision) for each object, then the following decision rule of Table 5.1 [25] will help us state the degree of possibility of each model object to be present in the scene.

Now, for each object 'i' in the scene, a rough estimate of the number of model features not visible in the scene is represented by the difference between the number of features (for occluded object 'i') correctly recognized (by checking the highest output of the neural net) minus k*TNMF (TNMF is considered for model object 'i'). In the occluded environment, under hard partitioning, this said difference is nonpositive. And for asserting the possibility of presence of an object in the scene we safely go by the rules of Table 5.1. But under hard partitioning, if the said difference is positive then it becomes an alarming situation indicating that some of the features of the model object 'j' are recognized as the features of the model object 'i' where $i \neq j$. In that case the vision system should be intervened in an interactive mode. In occluded environment, under hard partitioning, the value of k is experimentally chosen as 1.5. Thus, the vision system is given a freedom to recognize the features of some newly generated points which are either formed due to overlap of different objects in a scene or created due to some environmental uncertainties. As because, the newly generated points were not present at the time the net was trained, it is quite obvious that it is only the strength of fuzzy reasoning which can approximately recognize these additional points.

In this context, we like to state the basic design philosophy of the vision system which says, 'We can recognize the objects which we have seen earlier'. That means, in case of machine vision, the machine can recognize the objects which it has seen (or

Table 5.1: Decision Table

Number of model features visible in the scene	possibility of presence	Degree of presence
EQ 0	Nil	0
GT 0 but LT (50 % of TNMF - σ)	Poor	.3
EQ (50 % of TNMF $\pm\sigma$)	Fair	.7
GT (50 % of TNMF + σ) but LT (k*TNMF)	Good	1

Key: GT: Greater than, LT: Less than, EQ: Equal to, TNMF: Total number of features of a model object, σ represents the spread around the expected number of features under maximum allowable occlusion. The exact value of σ varies depending upon the dynamic ranges of the features. It (σ) is basically a design heuristic. k represents the factor of safety of the entire design scheme. Its values are different under hard partitioning and fuzzy partitioning. The degree of presence (right most column of Table 5.1) is basically a scaling process [98]

about which it has learned) earlier. But, under occlusion, it is quite obvious, that some of the model features which were seen earlier, may not be visible in the scene due to significant changes in the feature values and hence, may not be detectable (visible) by the recognition scheme. Conversely, some of the newly generated points (as mentioned in the previous paragraph), which were not seen earlier by the machine and which have feature values closed to the seen (by the machine) points of the model objects, are recognized correctly by the machine. Thus, the proposed vision system has reasonable tolerance around the basic design philosophy.

In case of fuzzy partitioning, which is very essential under uncertain environment, the value of k is experimentally chosen as 2. Thus, we have given some added flexibility in the design of the vision system at the cost of some risks. But the achievements (in terms of recognition, see Table 5.14 and 5.22) due to this additional flexibility is very significant in comparison to the risks involved and hence very much acceptable for practical use. Multiple classifications under fuzzy partitioning is very effective when similar features of two different objects overlap each other (see Figure 5.17).

5.2.6 Control Scheme of the vision process

There are several existing control schemes for the vision process [55,56,57,58,85,86,87]. For instance, in the following we provide representative control schemes for the vision process.

1. Bottom - Up control
2. Top - Down control
3. Feedback control
4. Heterarchical control

In the present chapter, for the recognition of two dimensional occluded objects, we follow the bottom up control scheme. Figure 5.2 shows the scheme stated above.

By adapting the above scheme for the vision process we can have the following advantages:

(i) Knowledge (model) of object is used only for matching scene descriptions

(ii) In Bottom - Up control, raw data are gradually converted into more organized and useful information

(iii) In Bottom - Up control objects may easily be changed simply by changing the models of the objects.

But the Bottom - Up scheme provides the following disadvantages also:

(i) Lower processes are not well adjusted to a given particular scene because they are domain independent. Errors may be inherited.

(ii) Inefficient because it is always executed in the same manner regardless of the scene.

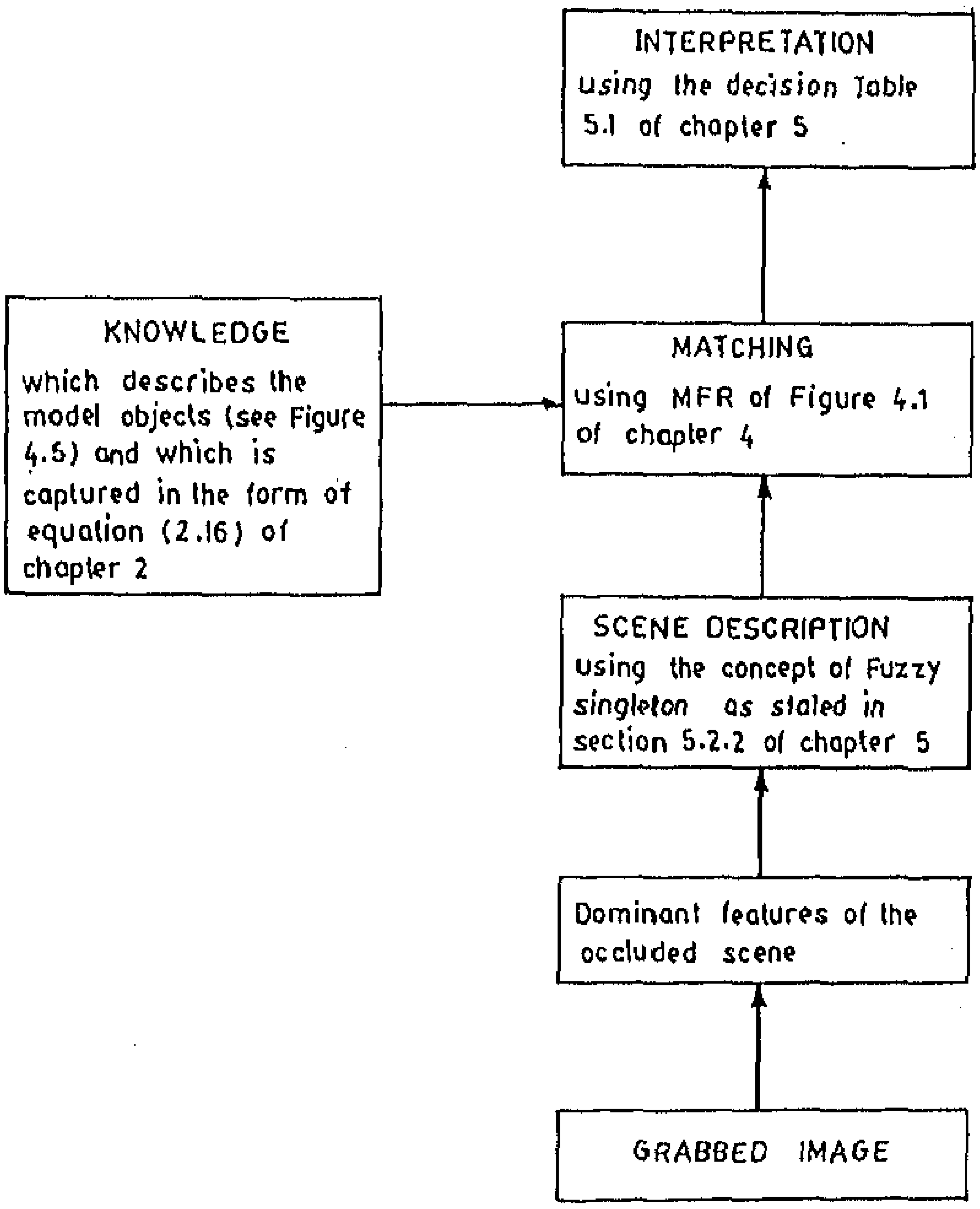


Fig. 5.2: Bottom Up control for the vision process

The present approach to the recognition of occluded object is guided by a decision table (Table 5.1) which basically follows a voting scheme [55,56,57,58]. As we do not consider the positioning problem and do not follow the hypothesis generation and verification paradigm, we simply stick to the Bottom - Up scheme. But depending upon the need of the problem we may follow any other scheme as stated earlier.

5.3 Experimental Results

We performed the experiment under three different case studies. In the first case, we consider recognition of scenes which consist of two objects. In the second case, we consider recognition of scenes which consist of three objects and in the third case, we consider recognition of scenes which consist of five objects. In all the three cases, we obtain similar type of results. Also, we notice that increment in the number of objects in a scene does not seriously affect the performance of the recognition scheme.

Case study 1

We consider two model objects, namely pliers and wrench (see Figure 5.3(a,b)). The internal angle θ (feature F_1) and curvature κ (feature F_2) are extracted at some significant points of the model objects (see Table 5.2 & 5.3). These local features are plotted in Figure 5.4. Figure 5.4 shows the clusters of features of model objects. The feature F_1 and the feature F_2 are fuzzified as discussed in section 5.2.2 (see Table 5.4(a) and Table 5.4(b)). Next we generate the fuzzy **If Then** rules (see Table 5.5(a) and Table

Table 5.2: Internal Angle Vs. Curvature of model object of Figure 5.3(a)

Model point	Internal Angle	Curvature
15	56.07	7.74
25	54.44	8.06
40	63.06	7.25
64	39.07	10.06
65	46.80	9.50
66	32.64	9.20

Table 5.3: Internal Angle Vs. Curvature of model object of Figure 5.3(b)

Model point	Internal Angle	Curvature
13	100.39	4.41
14	123.64	3.52
15	118.36	3.85
24	112.78	4.20
26	60.77	9.93
27	101.40	4.91
28	107.05	6.07
29	57.97	7.62
30	110.00	4.47

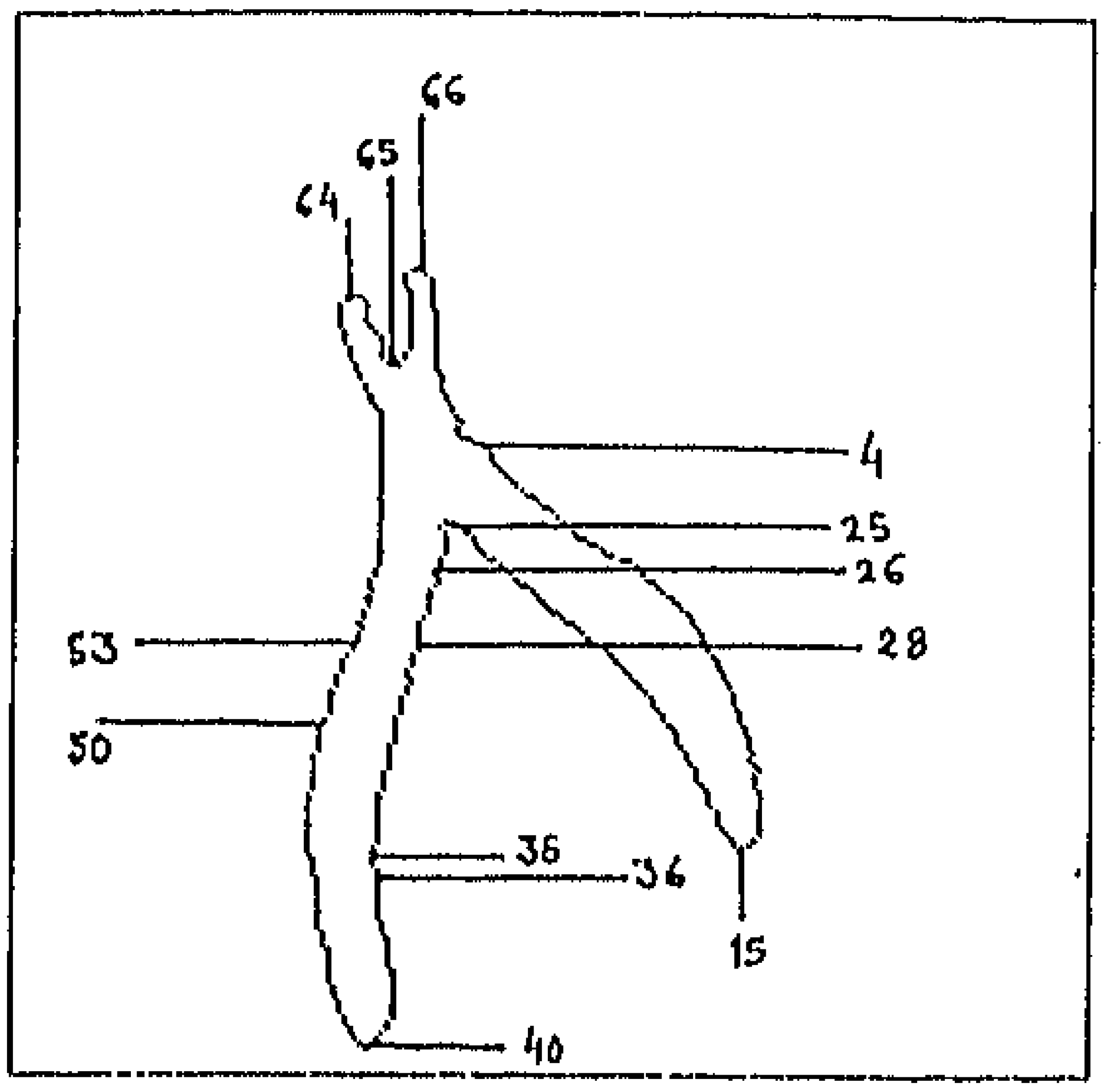


Fig. 5.3(a): model1

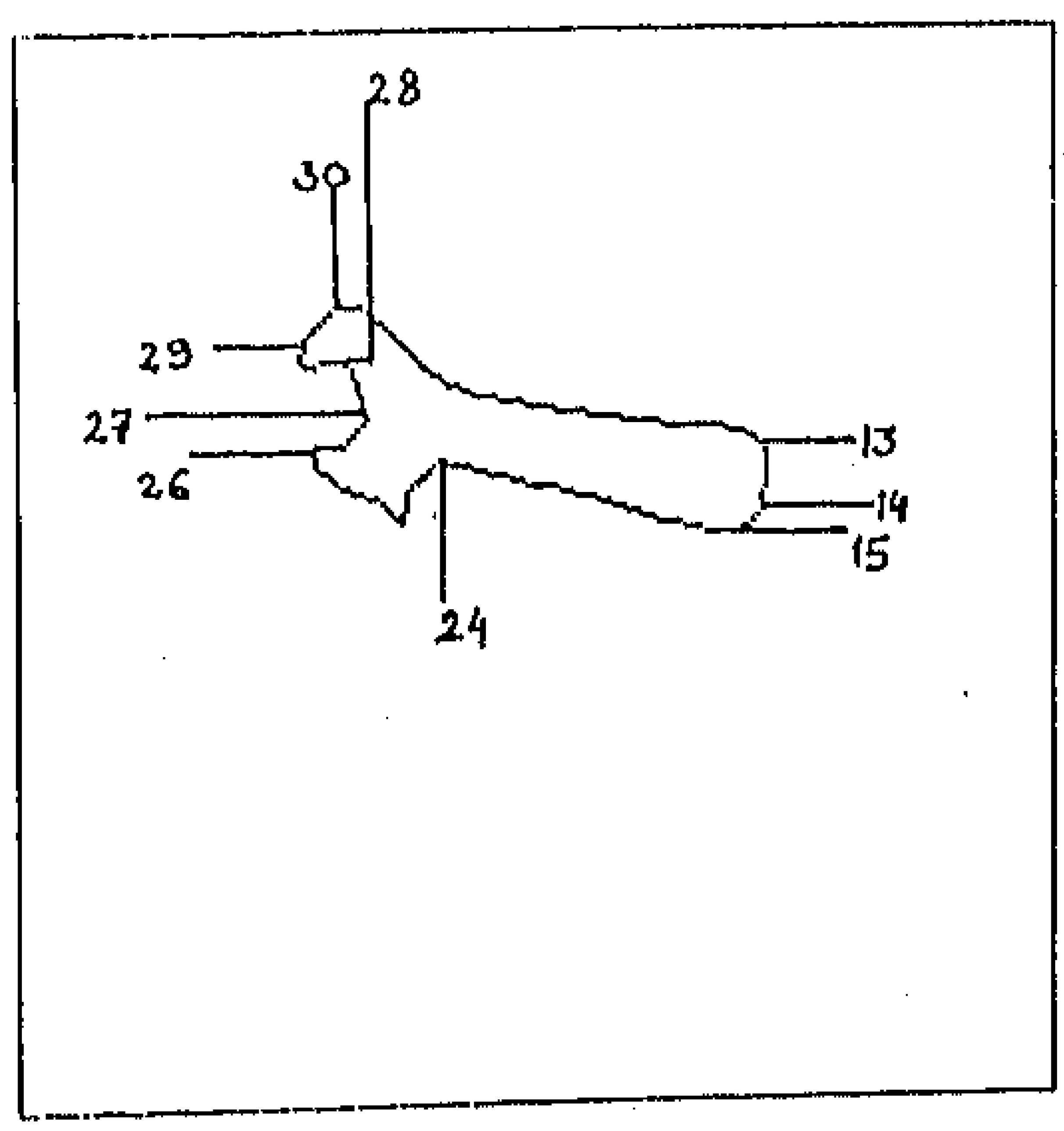
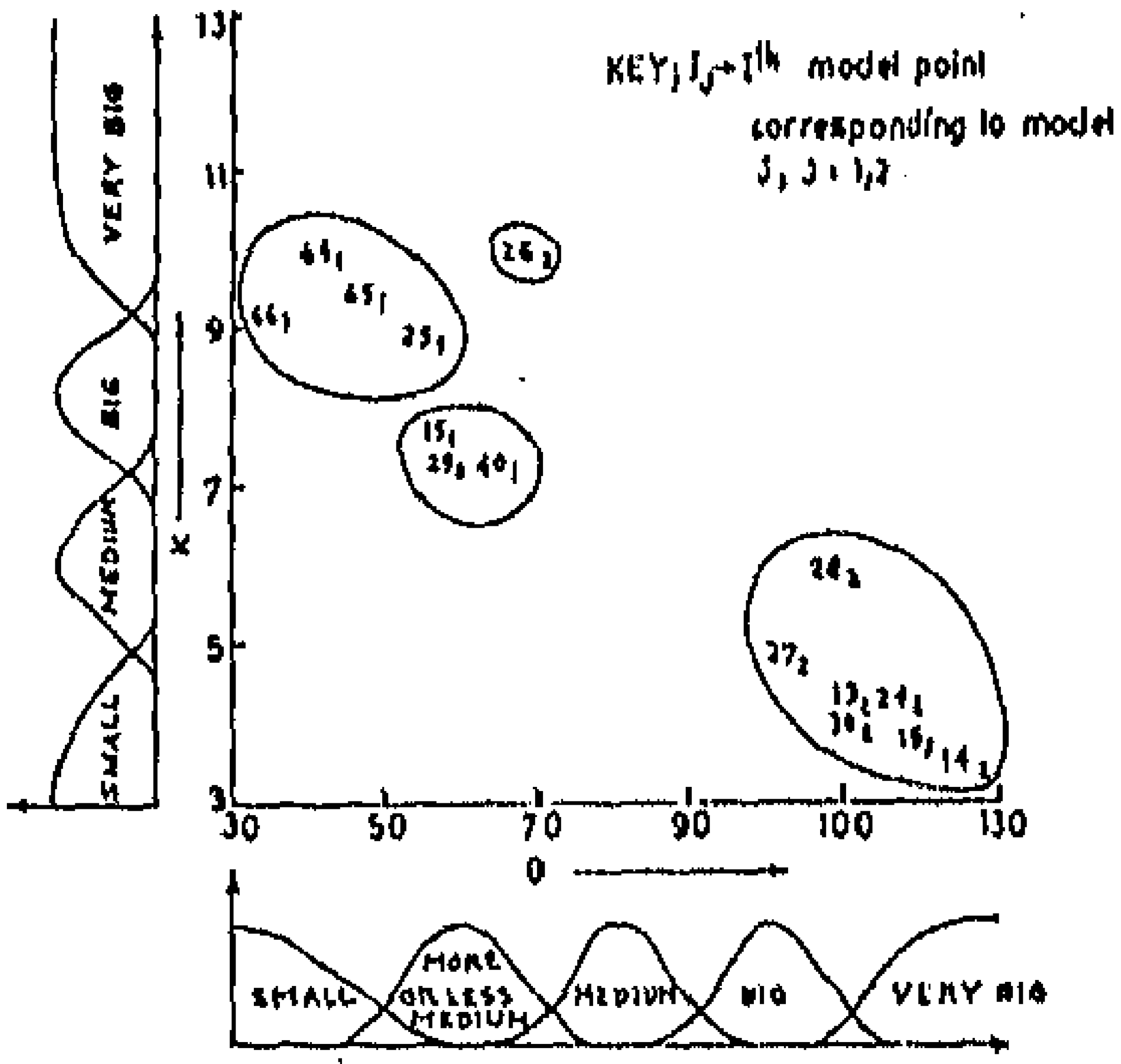


Fig. 5.3(b): model2



The membership functions for different linguistic labels e.g. S (Small), M (Medium) etc. are discrete as shown in Table 5.4 (a,b). But in the figure it is shown in a continuous form tomake it illustrative.

Fig. 5.4: Significant features of Pliers and Wrench

5.5(b)) based on the different locations of the clusters of the features of the model objects. The network of Figure 4.1 of Chapter 4 is trained by the said rules (a set of

Table 5.4(a):Quantization of the first feature of Figure 5.4

	Small	More or Less Medium	Medium	Big	Very big
$30 \le \theta < 35$	1	0	0	0	0
$35 \le \theta < 40$	.8	0	0	0	0
$40 \le \theta < 45$	.6	.2	0	0	0
$45 \le \theta < 50$	.4	.4	0	0	0
$50 \le \theta < 55$	.2	.6	0	0	0
$55 \le \theta < 60$	0	.8	0	0	0
$60 \le \theta < 65$	0	1	.2	0	0
$65 \le \theta < 70$	0	.8	.4	0	0
$70 \le \theta < 75$	0	.6	.6	0	0
$75 \le \theta < 80$	0	.4	.8	0	0
$80 \le \theta < 85$	0	.2	1	.2	0
$85 \le \theta < 90$	0	0	.8	.4	0
$90 \le \theta < 95$	0	0	.6	.6	0
$95 \le \theta < 100$	0	0	.4	.8	0
$100 \le \theta < 105$	0	0	.2	1	0
$105 \le \theta < 110$	0	0	0	.8	.2
$110 \le \theta < 115$	0	0	0	.6	.4
$115 \le \theta < 120$	0	0	0	.4	.6
$120 \le \theta < 125$	0	0	0	.2	.8
$125 \le \theta \le 130$	0	0	0	0	1

DFIs for each network of Figure 4.1) and is made intelligent to recognize the scenes, (shown in Figure 5.5 to 5.8) under different sorts of uncertainties in the feature values. Uncertainties in feature values are mainly caused by noise and different orientations of objects in the scene. At the time of classifying the scene features which are listed in Table C.1 to C.4 , we fuzzify them using the method of fuzzy singleton. Results obtained are shown in Table 5.6 & 5.7.

Table 5.4(b):Quantization of the second feature of Figure 5.4

	Small	Medium	Big	Very Big
$3 \leq \kappa \prec 4$	1	0	0	0
$4 \leq \kappa \prec 5$	.7	.1	0	0
$5 \leq \kappa \prec 6$	.4	.6	0	0
$6 \leq \kappa \prec 7$	.1	1	.1	0
$7 \leq \kappa \prec 8$	0	.6	.6	0
$8 \leq \kappa \prec 9$	0	.1	1	.2
$9 \leq \kappa \prec 10$	0	0	.6	.4
$10 \leq \kappa \prec 11$	0	0	.1	.6
$11 \leq \kappa \prec 12$	0	0	0	.8
$12 \leq \kappa \leq 13$	0	0	0	1

Table 5.5(a): Training Rules (a set of DFIs) for the first feature of Figure 5.4

Antecedent	Consequent (Possibility of occurrence)	
	Model1	Model2
If θ is Small Then	1	0
If θ is More or Less Medium Then	.9	.1
If θ is Medium Then	0	1
If θ is Big Then	0	1
If θ is Very Big Then	.1	.9

Table 5.5(b): Training Rules (a set of DFIs) for the second feature of Figure 5.4

Antecedent	Consequent (Possibility of occurrence)	
	Model1	Model2
If κ is Small Then	.1	.9
If κ is Medium Then	.5	.5
If κ is Big Then	.9	.1
If κ is Very Big Then	.5	.5

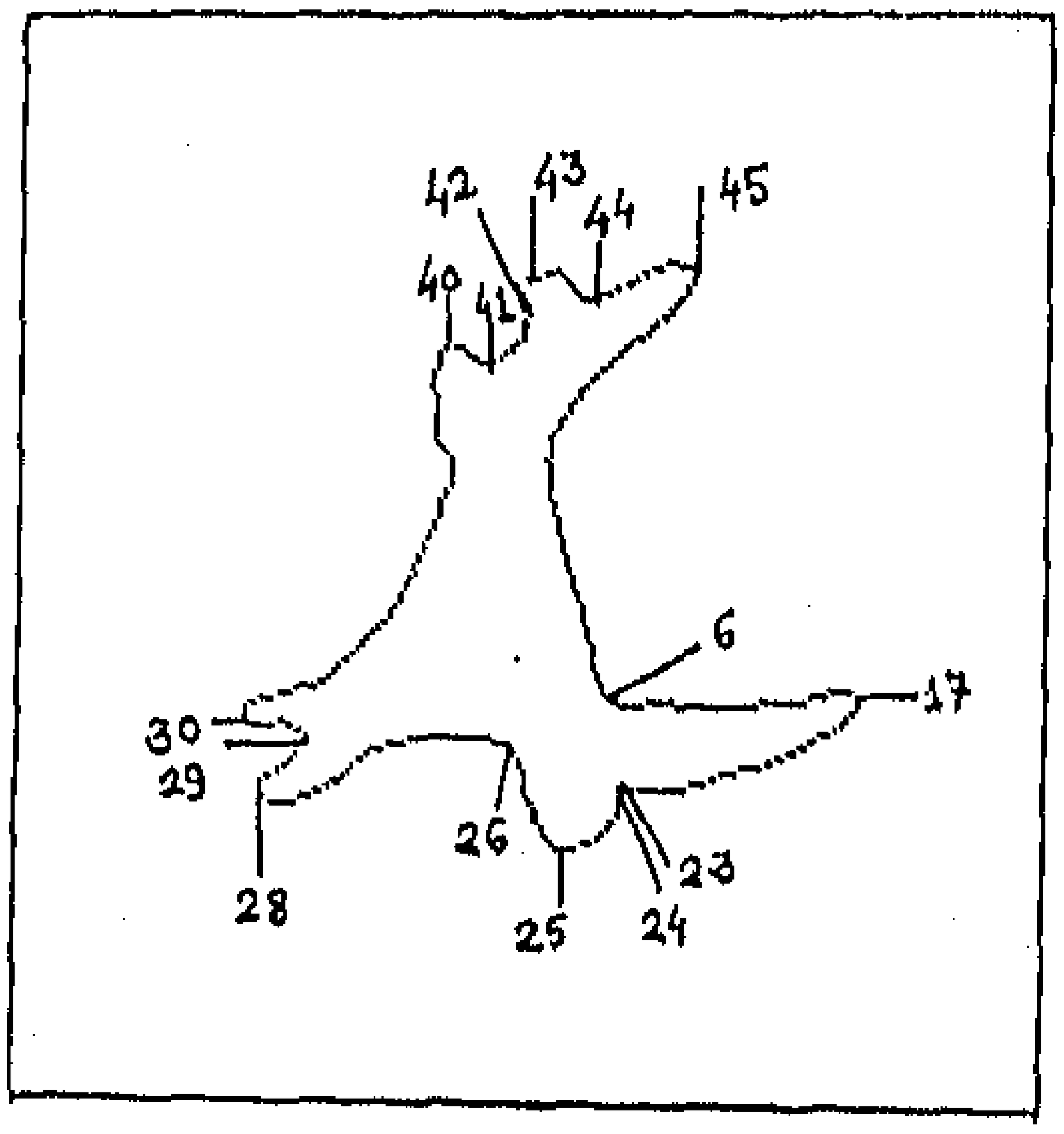


Fig. 5.5: occlusion1

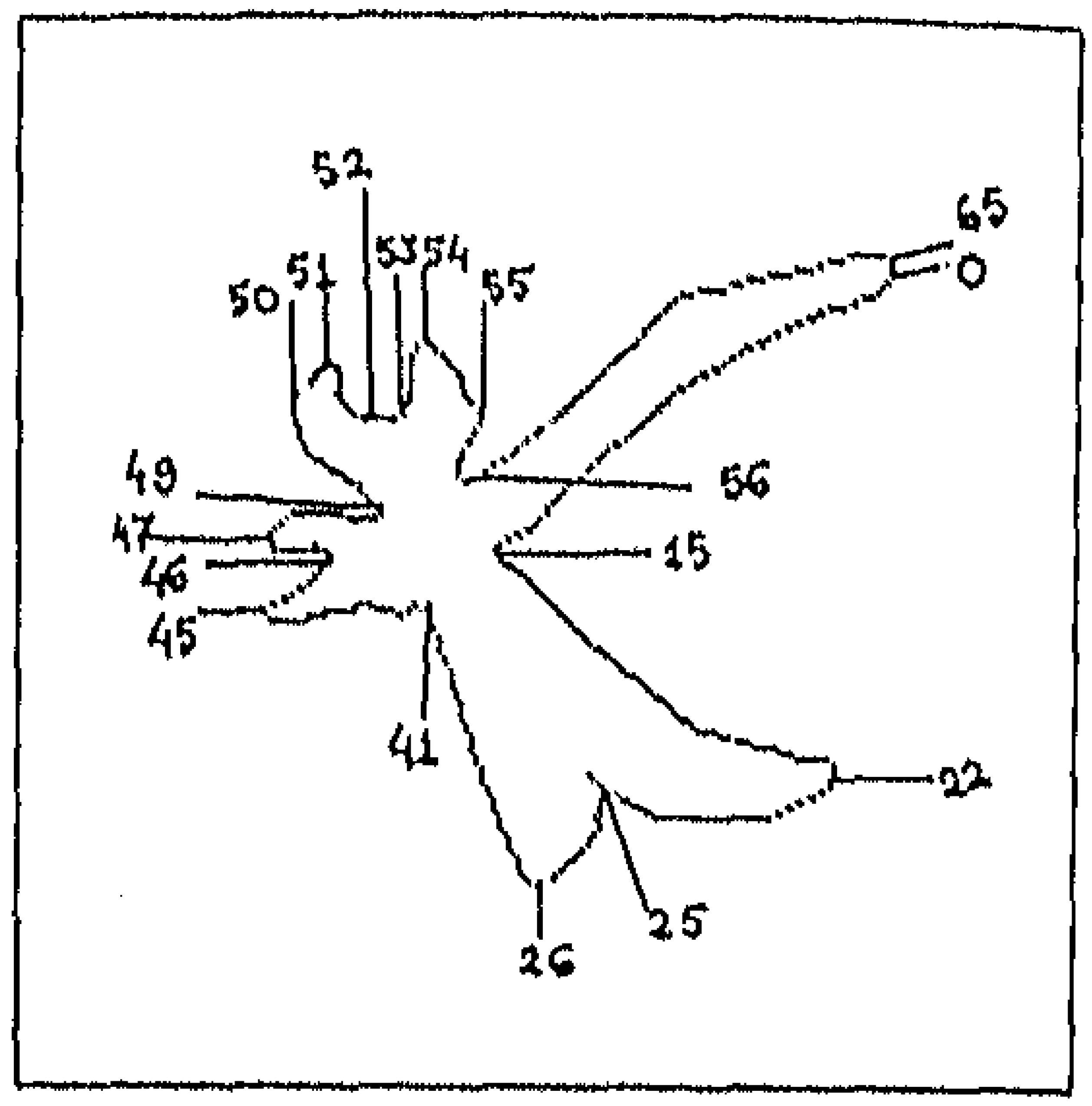


Fig. 5.6: occlusion2

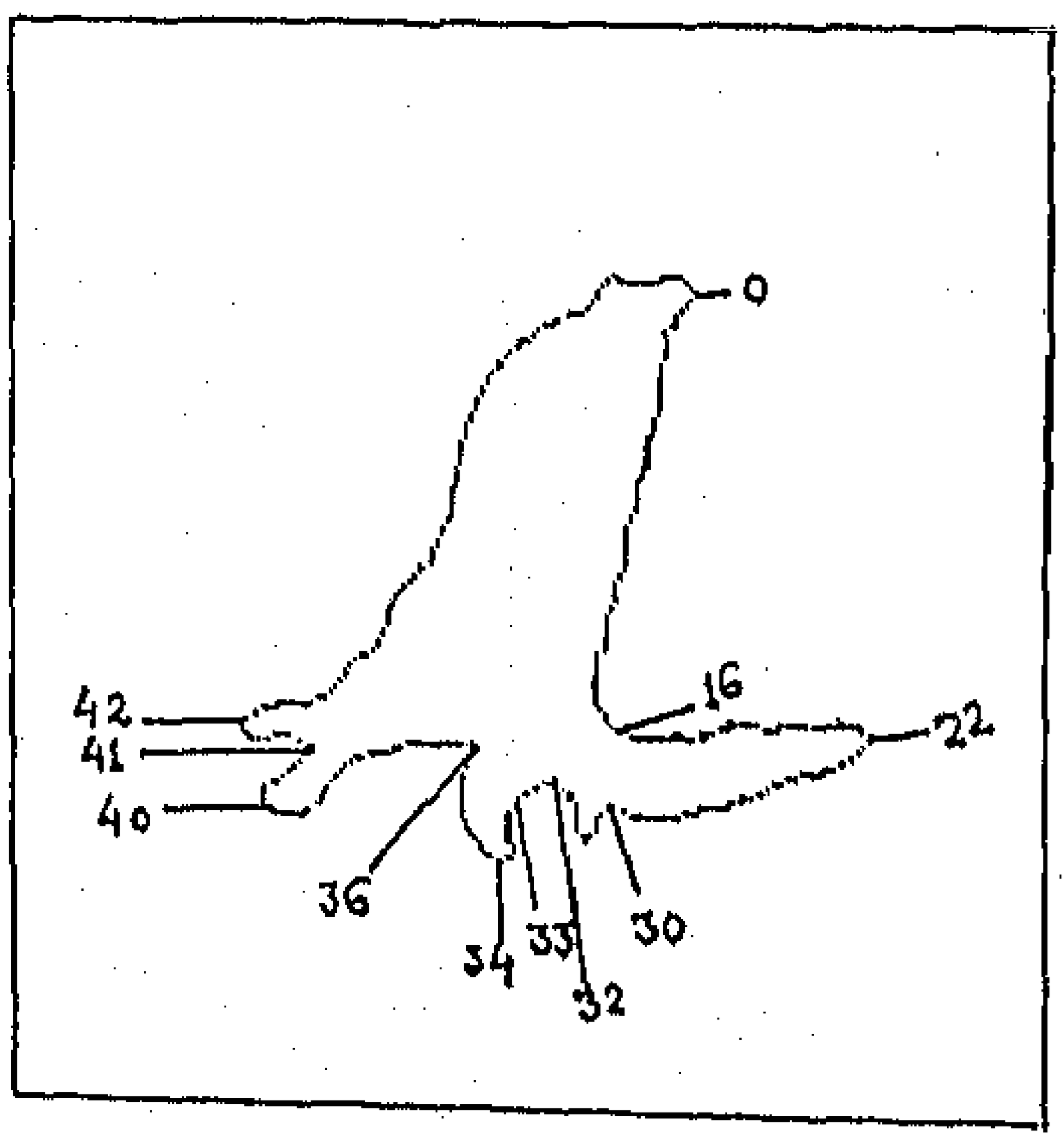


Fig. 5.7: occlusion3

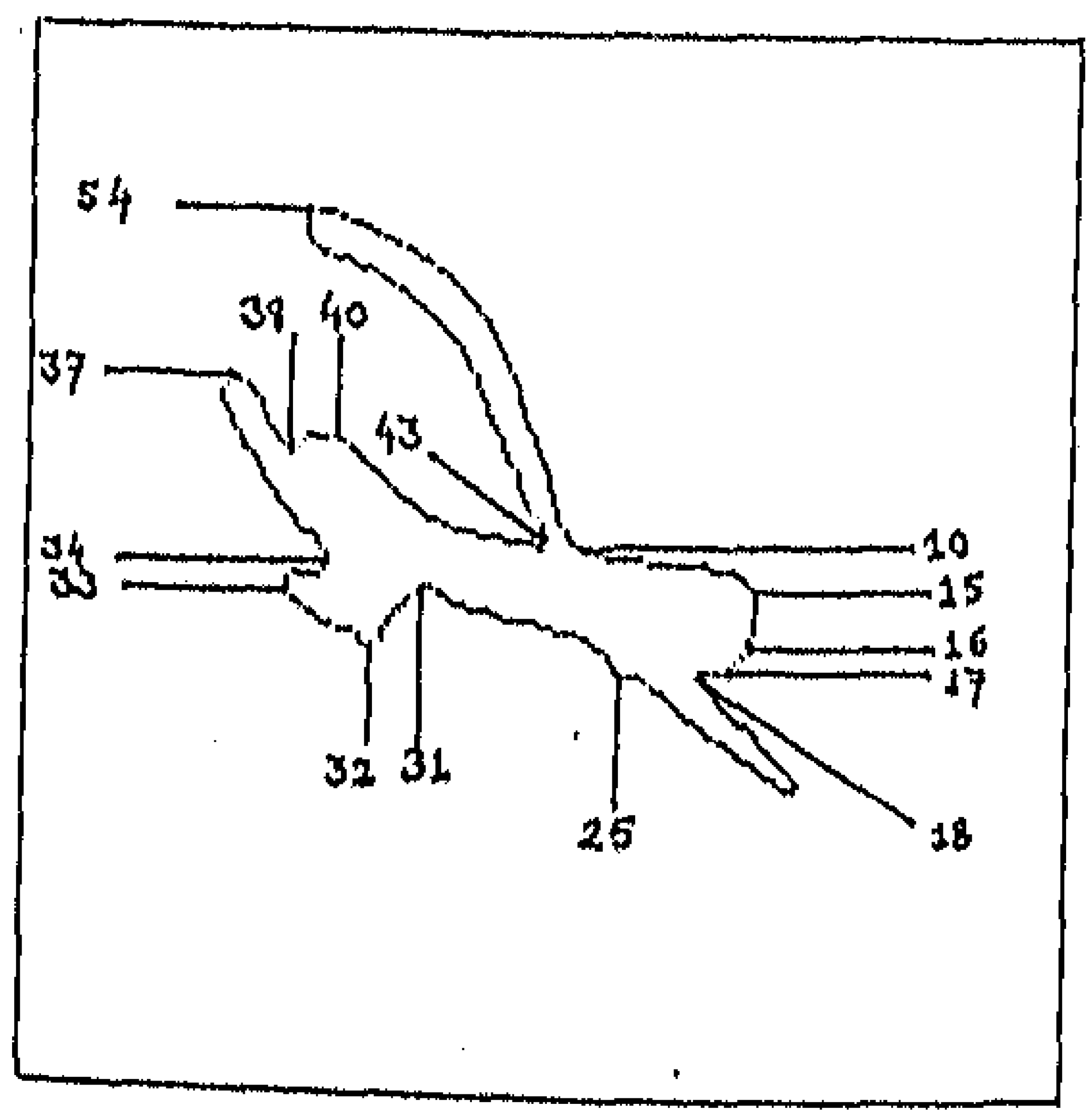


Fig. 5.8: occlusion4

Table 5.6:Recognition scores for the scene occlusion1 (i.e. Figure 5.5)

Scene points	Membership Values		Results	Remarks		Possibility of presence as per Table 2			
	Model No.			Model No.		Model No.			
	1	2		1	2	1		2	
						linguistic value	possibility degree	linguistic value	possibility degree
6	.26	.02	jn. point[1]						
17	.61	.20	correct	5 features out of 6 model features have been correctly classified	7 features out of 9 model features have been correctly classified	good	1	good	1
23	.22	.34	wrong						
24	.22	.40	jn. point						
25	.24	.62	correct						
26	.24	.72	correct						
28	.56	.34	correct						
29	.45	.34	correct						
30	.71	.27	correct						
40	.40	.49	correct						
41	.20	.68	correct						
42	.25	.29	correct						
43	.24	.49	correct						
44	.26	.62	jn. point						
46	.54	.34	correct						

1 jn. point indicates the junction points created by the overlapping of the objects. At this point, any one of the outputs can have higher possibility values.

In Table 5.6, we have given detailed recognition score for Figure 5.5. For brevity, the detailed recognition scores are not listed for each and every scene. Instead, the overall recognition scores are listed in Table 5.7.

Table 5.7:Overall Recognition scores for the occluded scenes (Figures 5.6 to 5.8)

Name of the scene	Percentage of features correctly classified		Possibility of presence as per Table 2			
	Model1	Model2	Model1		Model2	
			linguistic value	possibility degree	linguistic value	possibility degree
Figure 5.6:occlusion2	7 out of 6	5 out of 9	good	1	good	1
Figure 5.7:occlusion3	5 out of 6	2 out of 9	good	1	poor	.8
Figure 5.8:occlusion4	2 out of 6	4 out of 9	fair	.7	fair	.7

According to the results listed in Table 5.6 and 5.7, in cases of Figure 5.5 and 5.6 the performance of the vision system is very much satisfactory. In case of Figure 5.6, we see that 7 points of the scene are recognized as the points of model1 which has basically

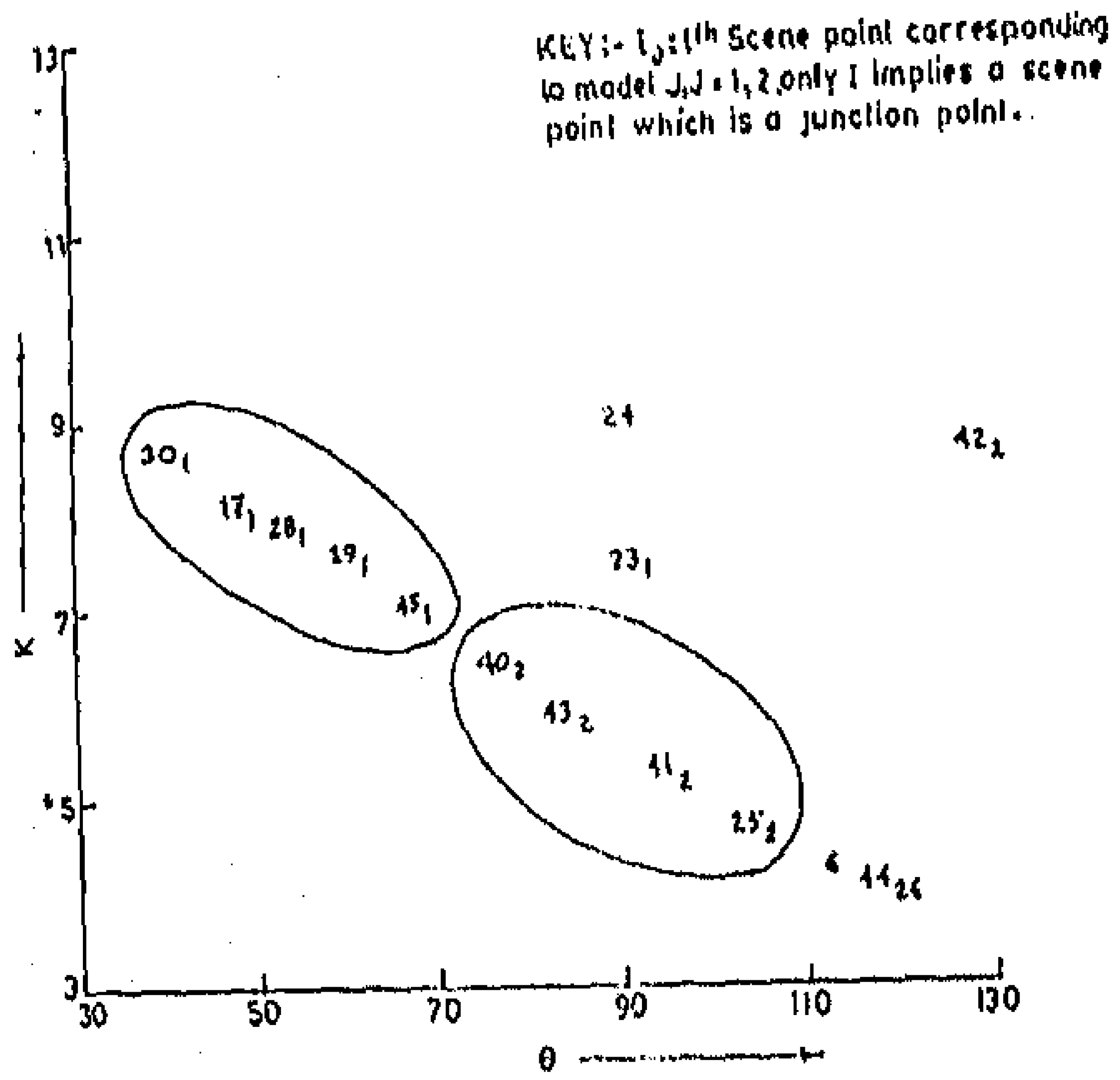


Fig. 5.9: Internal Angle Vs. Curvature plot of occlusion1 (i.e. Figure 5.5)

6 significant points. Here one newly generated point of the scene is recognized as a point of model object 1 (see the discussion of section 5.2.5). In the scene of Figure 5.7, model2, which has 9 significant points, is occluded more than 50 %, because the scene points 32, 33, 34 are the only representative points of model object 2. Hence, in this particular case, the ultimate performance of the vision system is not satisfactory (see the discussion of section 5.2.5). In all the scenes, shown in Figures 5.5,5.6 and 5.7, the arms of the pliers are stretched whereas in Figure 5.8, the arms of the pliers are kept closed. As a result, some significant features, i.e., the features of the significant points 64, 65 and 66 of model1 (see Figure 5.3(a)) are totally absent in the scene (i.e. Figure 5.8). Even then, the recognition scores based on the available features of the scenes are quite satisfactory, simply because in these cases, the problem has appeared to vision system as a problem of maximum occlusion (i.e. 50 % occlusion of model1; see the discussion of section 5.2.5) which is the maximum tolerable limit of occlusion to the vision mechanism.

In case study 1, the dynamic ranges of features for both models and scenes are 30 degree - 130 degree (internal angle) and 3 - 13 (curvature). It is to be noted that for the experiments of case study 1, we have considered the six significant points (as shown in Table 5.2) of pliers and nine significant points (as shown in Table 5.3) of wrench. But the total number of significant points in each scene not necessarily be equal to fifteen (six plus nine). For instance, in case of Figure 5.5 (occlusion1), we have total fifteen significant points out of which six points come directly from model1, five points come directly from model2 and the rest of the points are newly generated points due to the overlapping of two objects. Thus, total number of significant points of each scene may be equal to, less than and greater than the total number of significant points of all the models under consideration. As the newly generated points are formed due to the overlapping of different objects, at the time of recognition, they may be classified in any of the model classes or may be classified (having equal or almost equal membership) in more than one model class. Further, if we compare the clusters of Figure 5.4 with those of Figure 5.9, we will see the model features, whose significant changes are assumed to be invariant, are not located in exactly the same places of scene features in the $F_1 - F_2$ plane. These inherent uncertainties in the feature values, in addition to the uncertainties due to noise, may create lot of difficulties in recognition process provided we train the neural net by individual feature vector representing individual feature in the $F_1 - F_2$ plane. In the present experiment, we overcome such difficulties by giving training through fuzzy **If Then** rules which train the net according to the nature of population of pattern points. In the present approach, a population of pattern points is represented by a fuzzy pattern vector / feature vector as shown in Figure 2.26(a) of Chapter 2. This is the main motivation to introduce fuzzy rules to train the neural net. As a result, we can train the net, using the backpropagation principle, in quicker fashion (see the discussion of the last paragraph of Section 2.2 of Chapter 2). In case study 1, features of the scene are mostly linearly separable and there is no serious overlap of features of different objects. Hence, in the performance study of Table 5.6 & 5.7, we do

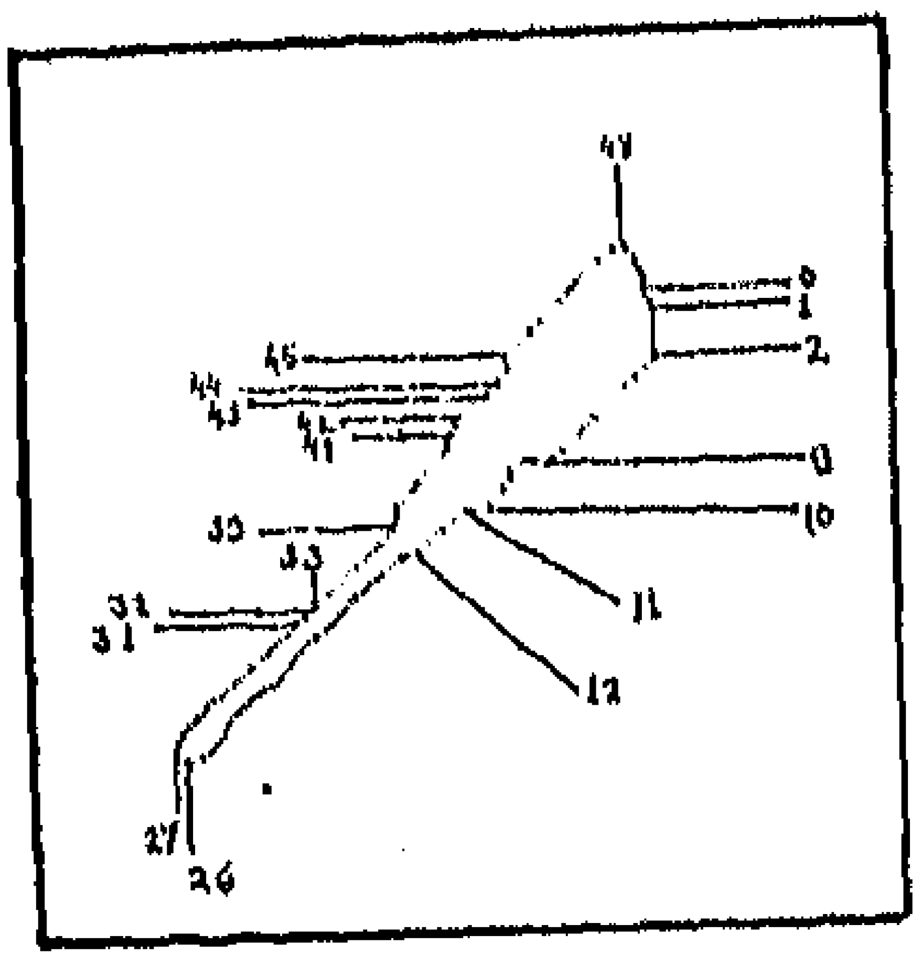


Fig. 5.10: model3

not consider multiple classifications in the overlapped zone.

The newly generated points which are basically the junction points in the scenes and the significant points whose feature values are not exactly the same as the feature values of the model objects are satisfactorily classified by the strength of approximate reasoning which is basically realized by the individual network of Figure 4.1 of Chapter 4. By the term "strength of approximate reasoning" we want to mean the generalized modus ponens property which is preserved in our cases also. That means, even under inexact situations, in contrast to exact matching, satisfactory reasoning for recognition is possible. After the initial recognition of the local features of the scene, the final decision about the existence of the model objects in the scene is taken in terms of the degree of possibility using the rules of Table 5.1 whose performance is experimentally verified. In the present case study, as the dynamic range of features varies from 30 degree - 130 degree and 3 - 13, we have taken the value of 'σ' equal to one. In the subsequent case studies, the dynamic ranges of the features have been extended for better recognition in a complex situation and hence the value of 'σ' in those cases are different. Of course, the choice of the value of σ is subjective and problem dependent.

Case study 2

We consider three model objects, namely pliers, wrench and screwdriver (see Figures 5.3(a), 5.3(b) and 5.10). The internal angle θ (feature F_1) and curvature κ (feature F_2) are extracted at some significant points of the model objects (see Table 5.8,5.9,5.10).

Table 5.8: Internal Angle Vs. Curvature of model object of Figure 5.3(a)

Model point	Internal Angle	Curvature
4	142.06	6.32
15	56.07	7.74
25	54.44	8.96
20	146.25	5.62
28	146.25	5.62
35	143.07	6.15
30	143.07	6.15
40	63.98	7.25
60	146.25	5.62
53	146.25	5.62
64	39.07	10.06
65	46.89	9.50
66	32.64	9.20

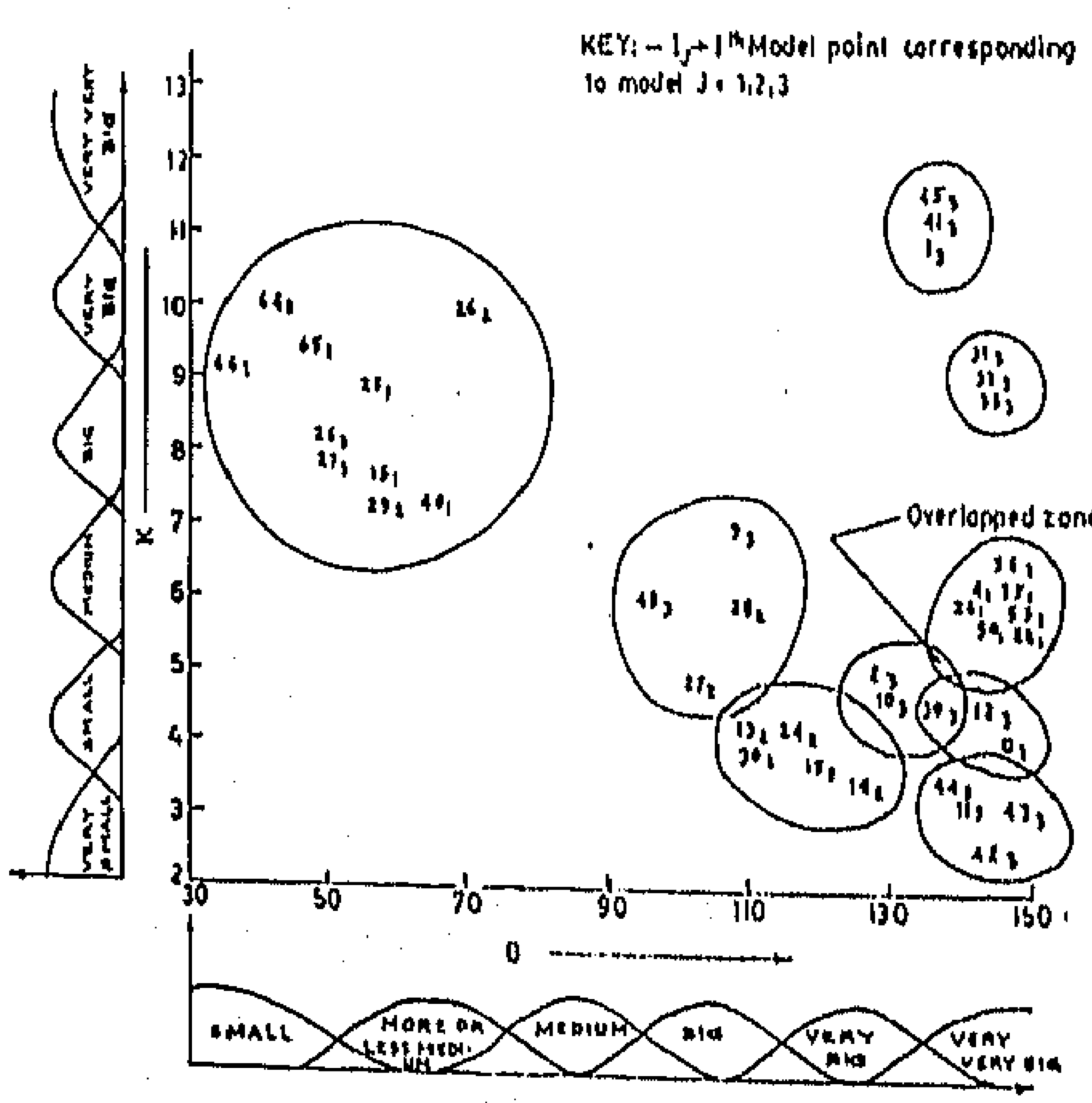
Table 5.9: Internal Angle Vs. Curvature of model object of Figure 5.3(b)

Model point	Internal Angle	Curvature
13	109.39	4.41
14	123.64	3.52
15	118.36	3.85
24	112.78	4.20
26	60.77	9.93
27	101.40	4.91
28	107.05	6.07
29	67.97	7.62
30	110.00	4.47

Table 5.10: Internal Angle Vs. Curvature of model object of Figure 5.10

Model point	Internal Angle	Curvature
0	140.25	4.21
1	134.04	11.20
2	128.60	5.13
9	100.60	7.03
10	128.60	5.13
11	140.13	3.32
12	143.07	4.61
26	40.37	8.16
27	40.37	8.16
31	143.07	0.23
32	143.07	0.23
33	143.07	0.23
39	134.04	4.60
41	134.04	11.26
42	142.68	2.05
43	148.07	3.10
44	138.31	3.47
45	134.04	11.20
48	94.72	0.09

These local features are plotted in Figure 5.11. Figure 5.11 shows the clusters of features of model objects. The feature F_1 and the feature F_2 are fuzzified as discussed



The membership functions for different linguistic labels e.g. S (Small), M (Medium) etc. are discrete as shown in Table 5.11 (a,b). But in the figure it is shown in a continuous form to make it illustrative.

Fig. 5.11: Significant features of Pliers, Wrench and Screw-driver

in section 5.2.2 (see Table 5.11(a) and Table 5.11(b)).

Table 5.11(a):Quantization of the first feature of Figure 5.11

	Small	More or Less Medium	Medium	Big	Very big	Very Very Big
$30 \leq \theta \prec 35$	1	0	0	0	0	0
$35 \leq \theta \prec 40$	.8	0	0	0	0	0
$40 \leq \theta \prec 45$	.6	.2	0	0	0	0
$45 \leq \theta \prec 50$	.4	.4	0	0	0	0
$50 \leq \theta \prec 55$	.2	.6	0	0	0	0
$55 \leq \theta \prec 60$	0	.8	0	0	0	0
$60 \leq \theta \prec 65$	0	1	.2	0	0	0
$65 \leq \theta \prec 70$	0	.8	.4	0	0	0
$70 \leq \theta \prec 75$	0	.6	.6	0	0	0
$75 \leq \theta \prec 80$	0	.4	.8	0	0	0
$80 \leq \theta \prec 85$	0	.2	1	.2	0	0
$85 \leq \theta \prec 90$	0	0	.8	.4	0	0
$90 \leq \theta \prec 95$	0	0	.6	.6	0	0
$95 \leq \theta \prec 100$	0	0	.4	.8	0	0
$100 \leq \theta \prec 105$	0	0	.2	1	.2	0
$105 \leq \theta \prec 110$	0	0	0	.8	.4	0
$110 \leq \theta \prec 115$	0	0	0	.6	.6	0
$115 \leq \theta \prec 120$	0	0	0	.4	.8	0
$120 \leq \theta \prec 125$	0	0	0	.2	1	0
$125 \leq \theta \prec 130$	0	0	0	0	.8	.2
$130 \leq \theta \prec 135$	0	0	0	0	.6	.4
$135 \leq \theta \prec 140$	0	0	0	0	.4	.6
$140 \leq \theta \prec 145$	0	0	0	0	.2	.8
$145 \leq \theta \leq 150$	0	0	0	0	0	1

Table 5.11(b):Quantization of the second feature of Figure 5.11

	Very Small	Small	Medium	Big	Very Big	Very Very Big
$2 \leq \kappa \prec 2.5$	1	0	0	0	0	0
$2.5 \leq \kappa \prec 3$	.7	0	0	0	0	0
$3 \leq \kappa \prec 3.5$	.4	.1	0	0	0	0
$3.5 \leq \kappa \prec 4$	.1	.6	0	0	0	0
$4 \leq \kappa \prec 4.5$	0	1	0	0	0	0
$4.5 \leq \kappa \prec 5$	0	.6	0	0	0	0
$5 \leq \kappa \prec 5.5$	0	.1	.1	0	0	0
$5.5 \leq \kappa \prec 6$	0	0	.6	0	0	0
$6 \leq \kappa \prec 6.5$	0	0	1	0	0	0
$6.5 \leq \kappa \prec 7$	0	0	.6	0	0	0
$7 \leq \kappa \prec 7.5$	0	0	.1	.1	0	0
$7.5 \leq \kappa \prec 8$	0	0	0	.6	0	0
$8 \leq \kappa \prec 8.5$	0	0	0	1	0	0
$8.5 \leq \kappa \prec 9$	0	0	0	.6	0	0
$9 \leq \kappa \prec 9.5$	0	0	0	.1	.1	0
$9.5 \leq \kappa \prec 10$	0	0	0	0	.6	0
$10 \leq \kappa \prec 10.5$	0	0	0	0	1	0
$10.5 \leq \kappa \prec 11$	0	0	0	0	.6	.2
$11 \leq \kappa \prec 11.5$	0	0	0	0	.1	.4
$11.5 \leq \kappa \prec 12$	0	0	0	0	0	.6
$12 \leq \kappa \prec 12.5$	0	0	0	0	0	.8
$12.5 \leq \kappa \leq 13$	0	0	0	0	0	1

Next we generate the fuzzy If Then rules (see Table 5.12(a) and Table 5.12(b)) based on the different locations of the clusters of the features of the model objects.

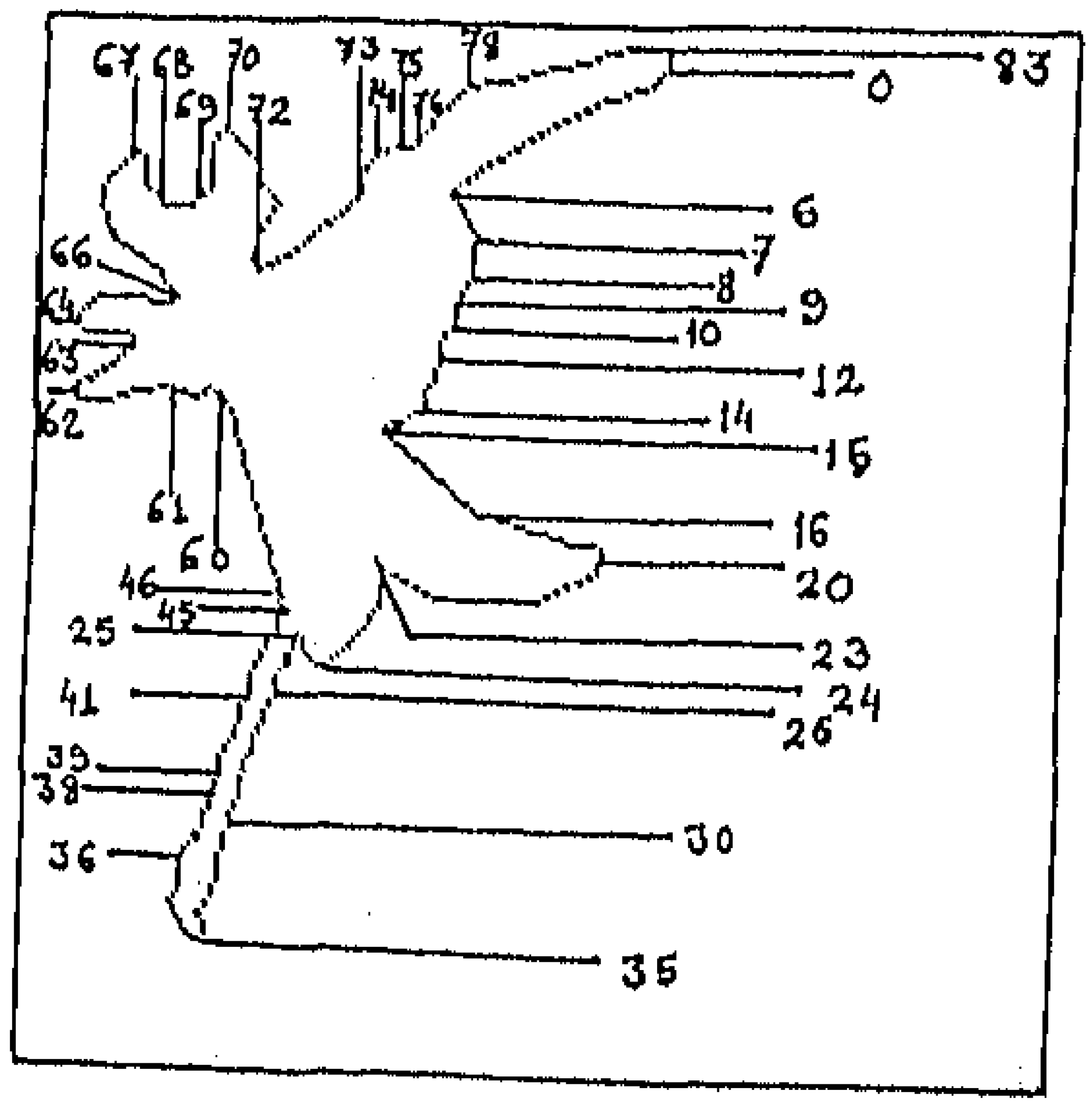


Fig. 5.12: occlusion5

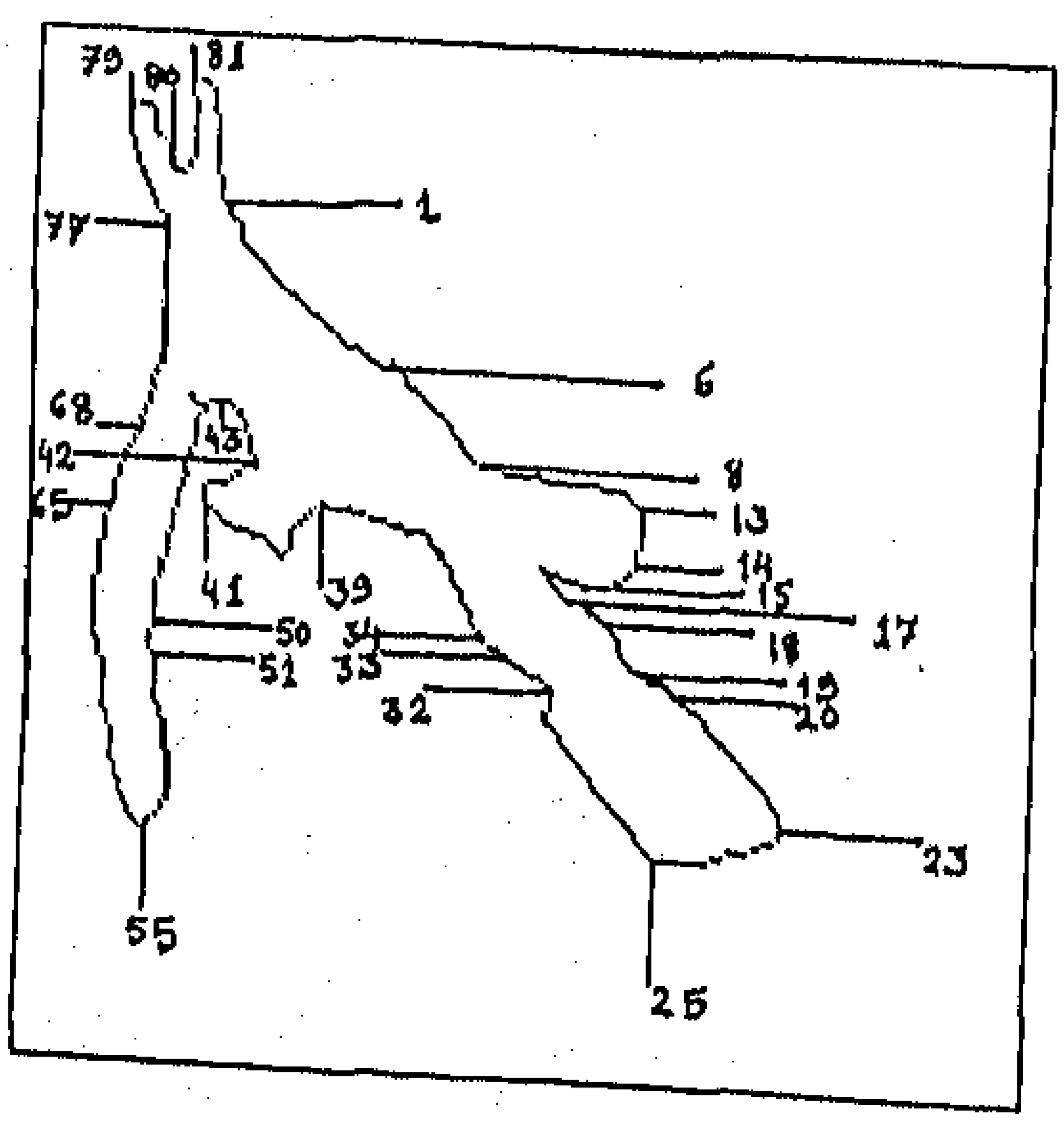


Fig. 5.13: occlusion6

The network of Figure 4.1 of Chapter 4 is trained by the said rules and is made intelligent to recognize the scenes (shown in Figure 5.12 and 5.13), under different sorts of uncertainties in the feature values. Uncertainties in feature values are mainly caused by noise and different orientations of objects in the scene. At the time of classifying the scene features which are listed in Table C.5 to C.6, we fuzzify them using the method of fuzzy singleton. Results obtained are shown in Table 5.13 and 5.14.

In case study 2, all other experimental details are same as case study 1, except that here the dynamic ranges of the features (see Figure 5.11) are extended and hence the value of 'σ' in Table 5.1 is selected as 3. In case study 1, we did not consider multiple classification for calculating the initial recognition scores given in Table 5.6 and 5.7; but in this case study, for further improvement in the recognition score we consider multiple classifications based on the selection of classes upto second highest membership value. The revised results are shown in Table 5.15. From Table 5.13, we understand that, under hard partitioning, the performance of the vision system is quite satisfactory.

Table 5.12(a): Training Rules for the first feature of Figure 5.11

Antecedent	Consequent (Possibility of occurrence)		
	Model1	Model2	Model3
If θ is Small Then	1	0	0
If θ is More or Less Medium Then	.9	.2	.1
If θ is Medium Then	0	1	0
If θ is Big Then	0	1	.1
If θ is Very Big Then	.1	.9	.1
If θ is Very Very Big Then	.1	0	1

Table 5.12(b): Training Rules for the second feature of Figure 5.11

Antecedent	Consequent (Possibility of occurrence)		
	Model1	Model2	Model3
If κ is Very Small Then	.1	.8	.6
If κ is Small Then	0	.9	.8
If κ is Medium Then	.8	.5	.2
If κ is Big Then	.9	.3	.1
If κ is Very Big Then	.9	.5	.1
If κ is Very Very Big Then	.1	.1	.8

Table 5.13:Recognition scores for the scene occlusion5 (Figure 5.12)

Scene points	Membership Values			Results	Remarks			Possibility of presence as per Table 2					
	Model No.				Model No.			Model No.					
	1	2	3		1	2	3	1		2		3	
								lingu -istic value	possi -bility degree	lingu -istic value	possi -bility degree	lingu -istic value	possi -bility degree
0	.64	.27	.17	correct	6	3	10	fair	.7	fair	.7	good	1
6	.21	.53	.20	jn. point	features	features	features						
7	.23	.32	.33	correct	out of	out of	out of						
8	.26	.08	.65	correct	13 model	9 model	10 model						
9	.25	.08	.27	correct	features	features	features						
10	.27	.11	.65	correct	have	have	have						
12	.27	.11	.66	correct	been	been	been						
14	.28	.23	.33	correct	correctly	correctly	correctly						
15	.32	41	.10	jn. point	classified	classified	classified						
16	.24	.08	.65	wrong									
20	.64	.27	.17	correct									
23	.48	.27	.23	jn. point									
24	.21	.41	.20	correct									
26	.60	.12	.21	jn. point									
6	.25	.08	.27	correct									
30	.26	.06	.27	correct									
35	.64	.14	.16	wrong									
36	.24	.08	.65	correct									
38	.25	.08	.27	correct									
39	.25	.08	.27	correct									
41	.28	.23	.40	correct									
45	.23	.32	.33	correct									
46	.23	.32	.33	jn. point									
60	.20	.54	.14	jn. point									
61	.25	.08	.55	wrong									
62	.64	.14	.16	correct									
63	.61	.12	.21	correct									
64	.67	.12	.21	correct									
66	.67	.14	.16	jn. point									
67	.56	.16	.23	wrong									
68	.20	.52	.19	correct									
69	.26	.56	.16	correct									
70	.54	.27	.17	wrong									
72	.68	.21	.12	jn. point									
73	.25	.08	.55	wrong									
74	.23	.23	.40	correct									
75	.23	.23	.40	correct									
76	.14	.32	.33	correct									
78	.23	.32	.33	wrong									
83	.48	.27	.23	correct									

Table 5.14:Detailed Recognition scores for the occluded scenes

Name of the scene	Hard Partitioning						Fuzzy Partitioning		
	scene points learned & correctly classified			scene points misclassified			extra scene points correctly classified		
	Model No.			Model No.			Model No.		
	1	2	3	1	2	3	1	2	3
Figure 5.12:occlusion5	0, 20, 62, 63, 64	24, 68, 69	7, 8, 9, 10, 12, 14, 26, 30, 36, 38, 39, 41, 45, 74, 75, 76	16, 61, 73, 78	67*, 70*	35*	16, 61, 73	70,	35
Figure 5.13:occlusion6	55, 79, 80, 81	13, 14, 15, 30, 41, 42, 43	17, 18, 19, 20, 33, 34	1, 6, 50*, 51*, 65*, 68, 77	-	23*, 25*, 32*	1, 6, 50, 51, 65, 68, 77	-	-

* indicates points those were learned

From the records listed in Table 5.14, we see that in cases of Figure 5.13, model3 is occluded more than 50 % which is beyond the tolerable limit of the vision system (see section 5.2.5) and hence the performance (in terms of the degree of possibility of presence of an object in the scene) of the vision system, even under fuzzy partitioning, does not improve for model 3. Whereas, detection of model1 (under fuzzy partitioning), which is not occluded more than 50 % in Figure 5.13, has significantly improved under fuzzy partitioning (see Table 5.15). Though the performance of the vision system for ultimate detection of model3 in the scenes of Figure 5.13 is not satisfactory, the available representative points of model3 in the scenes of Figure 5.12 and 5.13 are very satisfactorily detected (see Table 5.14; under hard partitioning, out of 18 representative points of model3, 1 is misclassified in Figure 5.12 and out of 9 representative points of model3, 3 are misclassified in Figure 5.13.

Table 5.15:Possibility of presence of the models in the occluded scenes as per Table 5.1

Name of of the scene	Hard Partitioning						Fuzzy Partitioning					
	Model No.						Model No.					
	1		2		3		1		2		3	
	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.
Figure 5.12:occlusion5	fair	.7	fair	.7	fair	.7	good	1	good	1	good	1
Figure 5.13:occlusion6	fair	.7	fair	.7	poor	.3	good	1	fair	.7	poor	.3

1 : l.v. stands for linguistic value

2 : p.d. stands for possibility degree

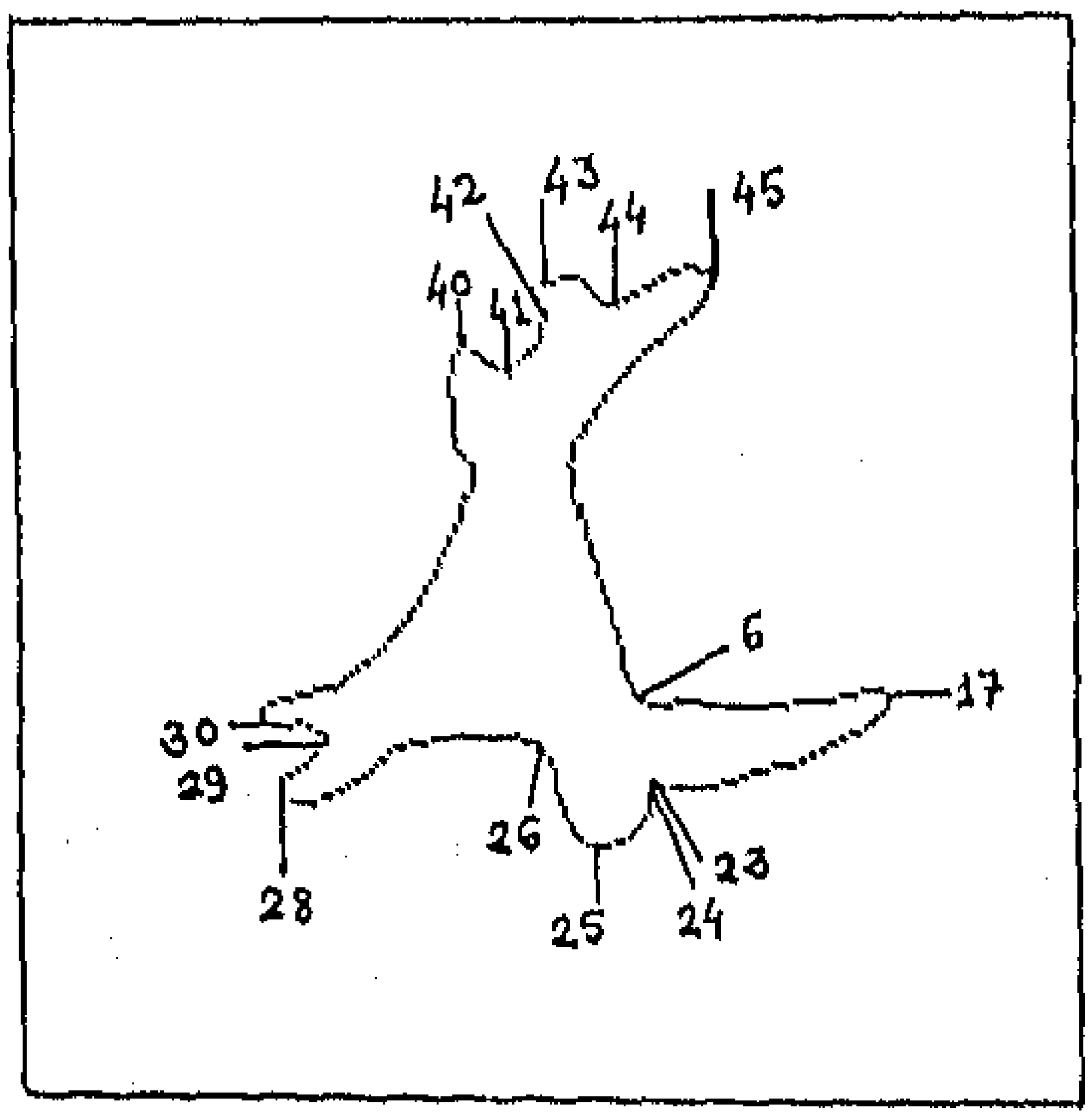


Fig. 5.14: occlusion7

Now, we consider another interesting situation. The net of Figure 4.1 of Chapter 4 is trained with the above said three model objects and then a scene consists of two model objects (one model object is absent), as shown in Figure 5.14, is given for recognition. After conducting a similar kind of experiment, we see that only the points 31 and 39, as shown in Figure 5.14, are misclassified. Therefore, in this situation, it is obvious that recognition score (according to scheme of Table 5.1) is good.

Thus, the failure and success story of case study 2 says that the proposed vision system works as per specification of design and if more than 50 % informations are not lost under occlusion then, the vision system can satisfactorily detect the model objects in the occluded scenes.

Case study 3

We consider five model objects, namely pliers, wrench, screwdriver, scissors and hammer (see Figure 5.3(a,b),5.10,5.15 and 5.16). The internal angle θ (feature F_1) and curvature κ (feature F_2) are extracted at some significant points of the model objects (see Table 5.16,5.17).

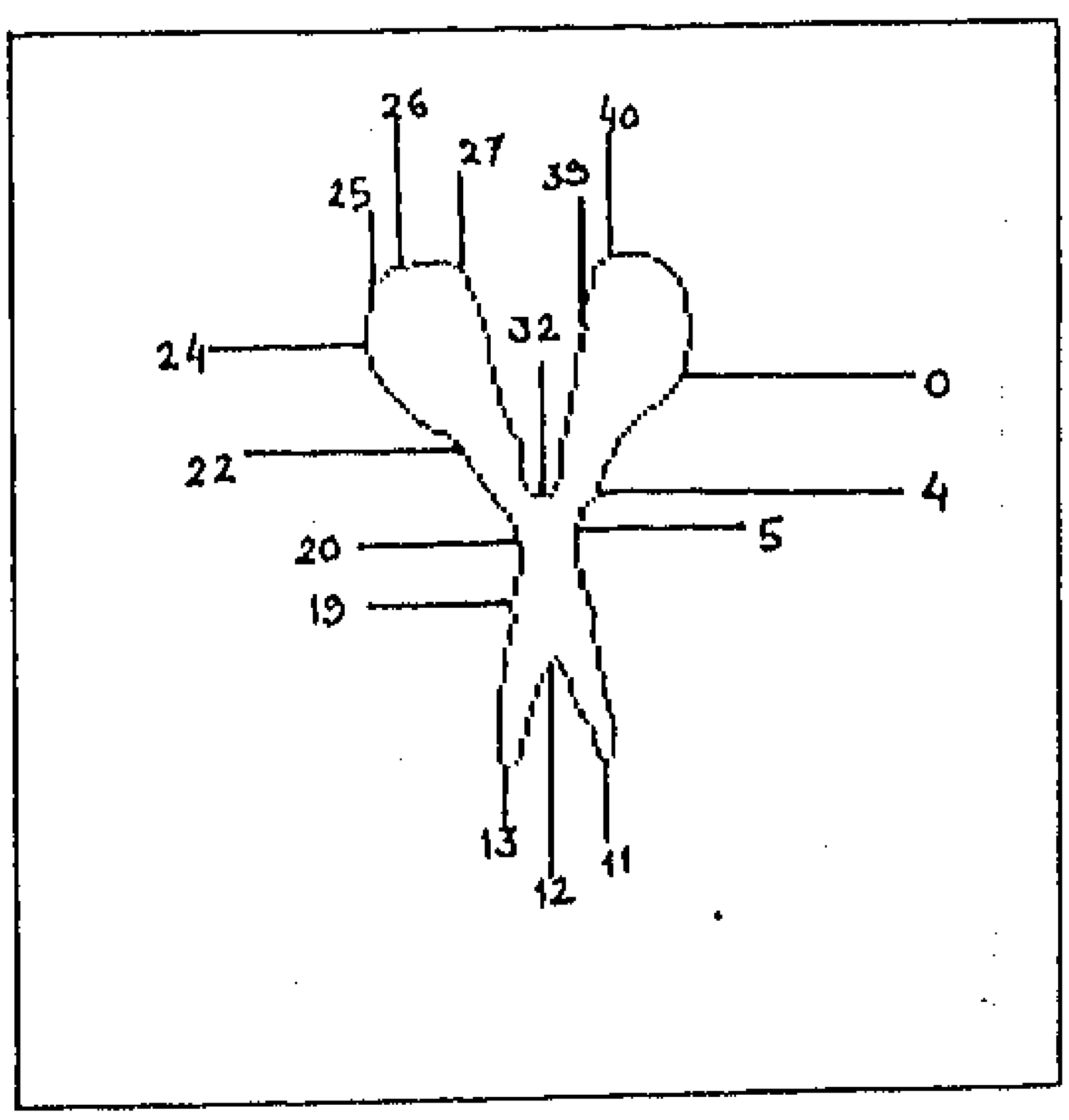


Fig. 5.15: model4

Table 5.16: Internal Angle Vs. Curvature of model object of Figure 5.15

Model point	Internal Angle	Curvature
0	142.06	2.37
4	134.04	11.20
5	140.25	2.81
11	31.62	9.27
12	52.02	9.07
13	36.48	8.06
10	143.07	6.15
20	140.81	2.44
22	149.42	2.18
24	140.81	2.44
25	127.62	3.72
26	123.04	3.52
27	114.83	4.07
32	67.35	7.04
30	130.34	5.08
40	112.12	4.24

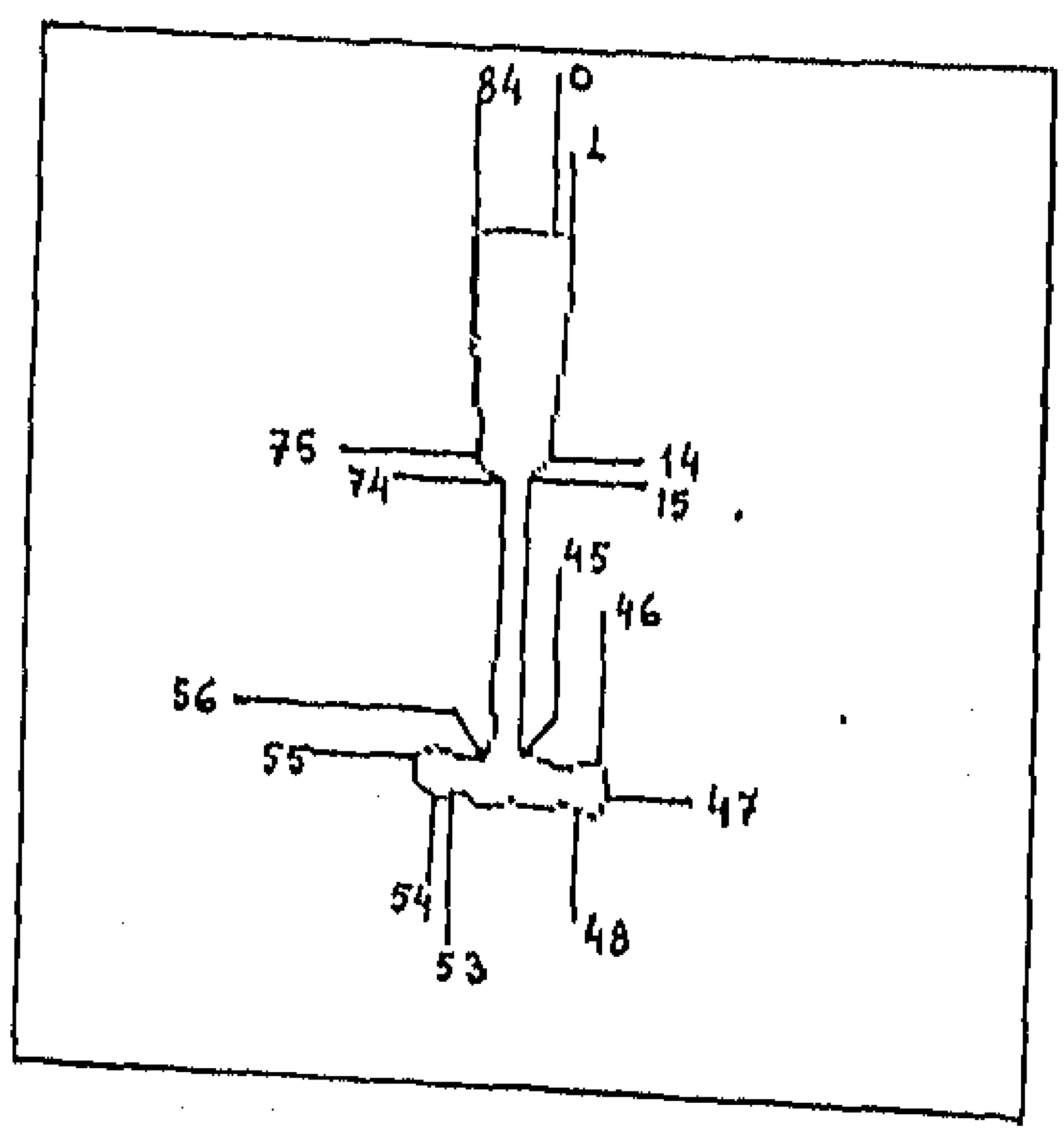


Fig. 5.16: model5

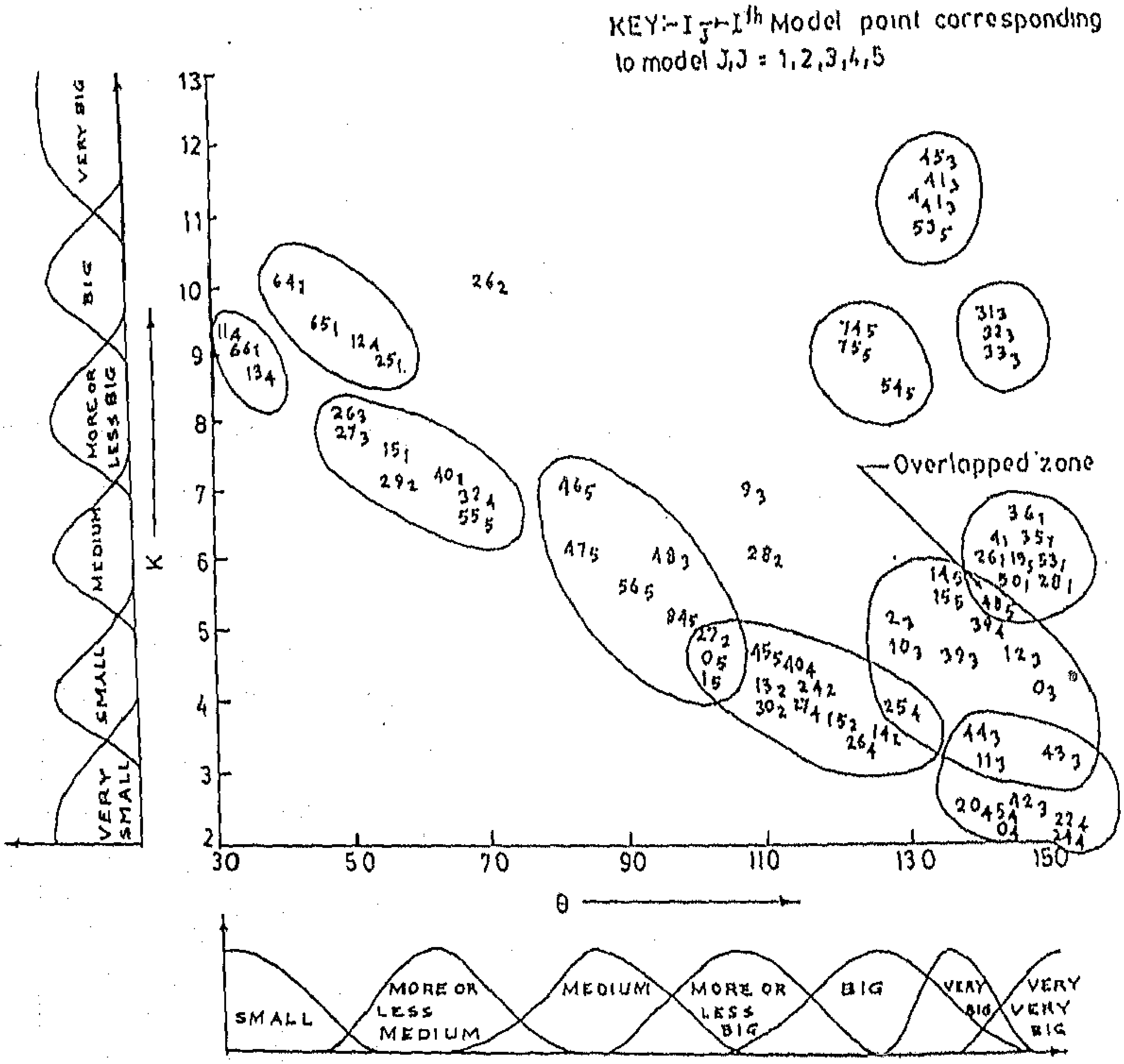
Table 5.17: Internal Angle Vs. Curvature of model object of Figure 5.16

Model point	Internal Angle	Curvature
0	103.99	4.75
1	103.99	4.75
14	134.94	5.63
15	134.94	5.63
45	110.51	4.34
46	81.83	7.01
47	82.15	6.11
48	139.34	5.08
53	134.94	11.26
54	127.82	8.69
55	67.35	7.04
56	89.96	5.62
74	123.64	9.39
75	123.64	9.39
84	97.08	5.18

These local features are plotted in Figure 5.17. Figure 5.17 shows the clusters of features of model objects. The feature F_1 and the feature F_2 are fuzzified as discussed in section 5.2.2 (see Table 5.18(a,b)).

Table 5.18(a):Quantization of the first feature of Figure 5.17

	Small	More or Less Medium	Medium	More or Less Big	Big	Very big	Very Very Big
$30 \leq \theta \prec 35$	1	0	0	0	0	0	0
$35 \leq \theta \prec 40$	.8	0	0	0	0	0	0
$40 \leq \theta \prec 45$	.6	.2	0	0	0	0	0
$45 \leq \theta \prec 50$	.4	.4	0	0	0	0	0
$50 \leq \theta \prec 55$	.2	.6	0	0	0	0	0
$55 \leq \theta \prec 60$	0	.8	0	0	0	0	0
$60 \leq \theta \prec 65$	0	1	.2	0	0	0	0
$65 \leq \theta \prec 70$	0	.8	.4	0	0	0	0
$70 \leq \theta \prec 75$	0	.6	.6	0	0	0	0
$75 \leq \theta \prec 80$	0	.4	.8	0	0	0	0
$80 \leq \theta \prec 85$	0	.2	1	.2	0	0	0
$85 \leq \theta \prec 90$	0	0	.8	.4	0	0	0
$90 \leq \theta \prec 95$	0	0	.6	.6	0	0	0
$95 \leq \theta \prec 100$	0	0	.4	.8	0	0	0
$100 \leq \theta \prec 105$	0	0	.2	1	.2	0	0
$105 \leq \theta \prec 110$	0	0	0	.8	.4	0	0
$110 \leq \theta \prec 115$	0	0	0	.6	.6	0	0
$115 \leq \theta \prec 120$	0	0	0	.4	.8	0	0
$120 \leq \theta \prec 125$	0	0	0	.2	1	.1	0
$125 \leq \theta \prec 130$	0	0	0	0	.8	.6	0
$130 \leq \theta \prec 135$	0	0	0	0	.6	1	0
$135 \leq \theta \prec 140$	0	0	0	0	.4	.6	.1
$140 \leq \theta \prec 145$	0	0	0	0	.2	.1	.6
$145 \leq \theta \leq 150$	0	0	0	0	0	0	1



The membership functions for different linguistic labels e.g. S (Small), M (Medium) etc. are discrete as shown in Table 5.18 (a,b). But in the figure it is shown in a continuous form to make it illustrative

Fig. 5.17: Significant features of Pliers, Wrench, Screw-driver, Scissors and Hammer

Table 5.18(b):Quantization of the second feature of Figure 5.17

	Very Small	Small	Medium	More or Less Big	Big	Very Big
$2 \leq \kappa \prec 2.5$	1	0	0	0	0	0
$2.5 \leq \kappa \prec 3$	.7	0	0	0	0	0
$3 \leq \kappa \prec 3.5$	.4	.1	0	0	0	0
$3.5 \leq \kappa \prec 4$	.1	.6	0	0	0	0
$4 \leq \kappa \prec 4.5$	0	1	0	0	0	0
$4.5 \leq \kappa \prec 5$	0	.6	0	0	0	0
$5 \leq \kappa \prec 5.5$	0	.1	.1	0	0	0
$5.5 \leq \kappa \prec 6$	0	0	.6	0	0	0
$6 \leq \kappa \prec 6.5$	0	0	1	0	0	0
$6.5 \leq \kappa \prec 7$	0	0	.6	0	0	0
$7 \leq \kappa \prec 7.5$	0	0	.1	.1	0	0
$7.5 \leq \kappa \prec 8$	0	0	0	.6	0	0
$8 \leq \kappa \prec 8.5$	0	0	0	1	0	0
$8.5 \leq \kappa \prec 9$	0	0	0	.6	0	0
$9 \leq \kappa \prec 9.5$	0	0	0	.1	.1	0
$9.5 \leq \kappa \prec 10$	0	0	0	0	.6	0
$10 \leq \kappa \prec 10.5$	0	0	0	0	1	0
$10.5 \leq \kappa \prec 11$	0	0	0	0	.6	.2
$11 \leq \kappa \prec 11.5$	0	0	0	0	.1	.4
$11.5 \leq \kappa \prec 12$	0	0	0	0	0	.6
$12 \leq \kappa \prec 12.5$	0	0	0	0	0	.8
$12.5 \leq \kappa \leq 13$	0	0	0	0	0	1

Next we generate the fuzzy **If Then** rules (see Table 5.19(a,b)) based on the different locations of the clusters of the features of the model objects.

Table 5.19(a): Training Rules for the first feature of Figure 5.17

Antecedent	Consequent (Possibility of occurrence)				
	Model1	Model2	Model3	Model4	Model5
If θ is Small Then	.9	0	0	.9	0
If θ is More or Less Medium Then	.8	.2	.1	.1	.1
If θ is Medium Then	0	.8	0	.1	.9
If θ is More or Less Big Then	0	.9	.1	.3	.9
If θ is Big Then	0	.8	0	.7	.5
If θ is Very Big Then	.9	.1	.8	.2	.2
If θ is Very Very Big Then	.9	0	.9	.9	.4

Table 5.19(b): Training Rules for the second feature of Figure 5.17

Antecedent	Consequent (Possibility of occurrence)				
	Model1	Model2	Model3	Model4	Model5
If κ is Very Small Then	.1	.5	.7	.9	0
If κ is Small Then	0	.9	.7	.4	.1
If κ is Medium Then	.7	.5	.5	.1	.8
If κ is More or Less Big Then	.8	.2	.5	.1	.9
If κ is Big Then	.7	.3	.1	.5	.5
If κ is Very Big Then	.1	.1	.5	.1	.5

The network of Figure 4.1 of Chapter 4 is trained by the said rules and is made intelligent to recognize the scenes (see Figure 5.18) under different sorts of uncertainties in the feature values. Uncertainties in feature values are mainly caused by noise and different orientations of objects in the scene. At the time of classifying the scene features which are listed in Table 5.20, we fuzzify them using the method of fuzzy singleton. Results obtained are shown in Table 5.21.

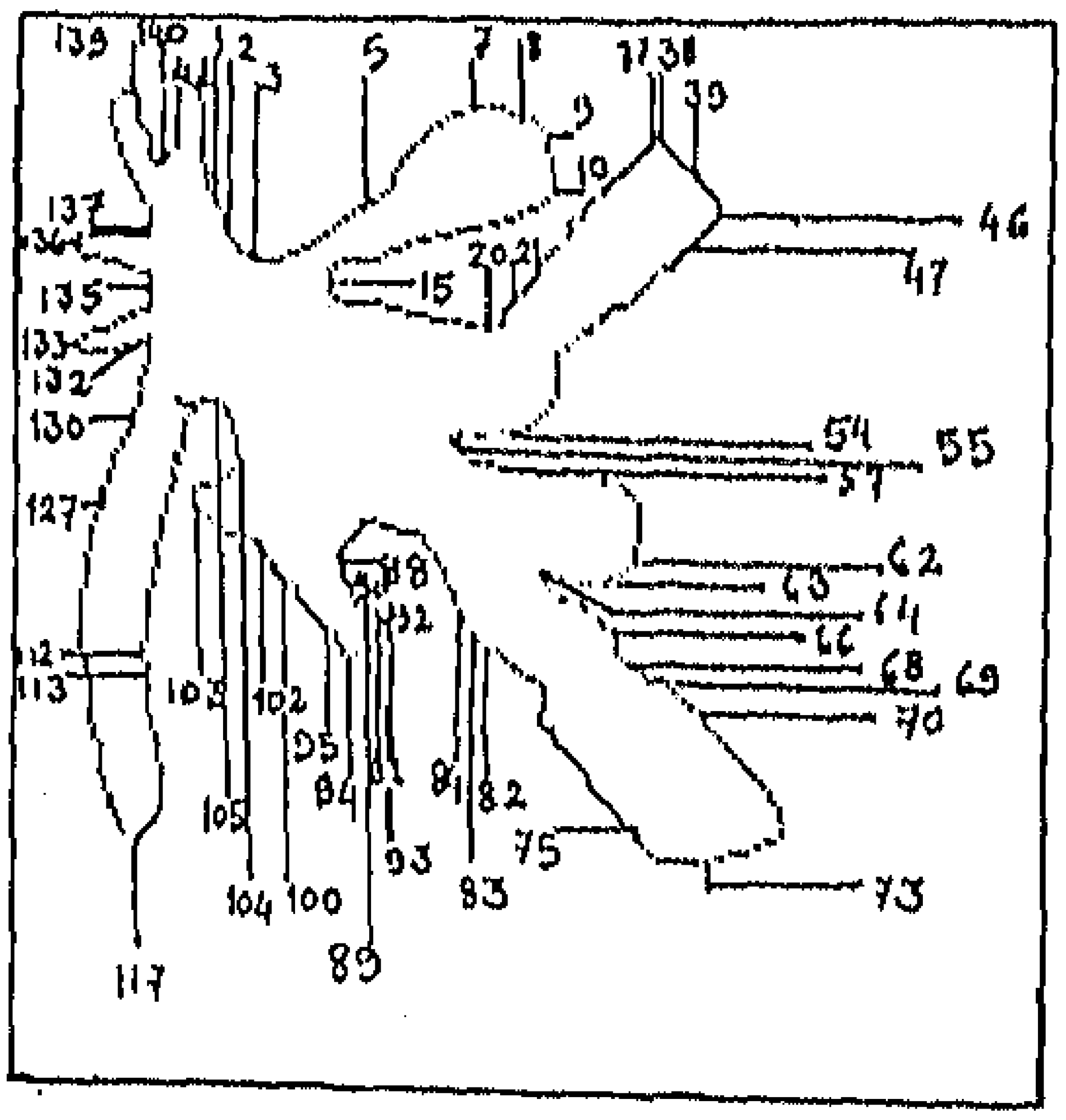


Fig. 5.18: occlusion8

Table 5.20: Internal Angle Vs. Curvature of occlusion8 (Figure 5.18)

Scene point	Internal Angle	Curvature
1	143.07	4.01
2	134.04	11.20
3	122.01	4.09
5	140.42	2.18
7	140.81	2.44
8	127.82	3.72
9	123.64	3.52
10	114.63	4.07
15	67.35	7.04
20	111.75	4.87
21	137.07	4.23
37	89.06	6.43
38	80.06	5.02
39	89.06	6.43

Table 5.20(Contd.)

Scene point	Internal Angle	Curvature
46	103.99	4.75
47	103.99	4.75
54	119.69	4.30
55	129.07	3.18
57	34.68	9.08
62	109.39	4.41
63	123.64	3.52
64	118.36	3.85
66	134.94	11.26
68	134.94	3.75
69	134.94	3.75
70	134.94	11.26
73	93.49	5.40
75	128.60	5.13
82	109.60	7.03
83	149.42	2.18
84	149.42	2.18
88	110.87	4.93
89	103.99	4.75
90	108.39	11.93
91	89.96	5.62
92	89.96	5.62
93	89.96	6.43
94	84.99	5.93
95	130.54	4.94
100	138.31	3.47
102	130.54	3.09
103	70.63	6.83
104	101.40	4.91
105	79.57	6.27
112	143.07	6.15
113	143.07	6.15
117	63.98	7.25
127	146.25	5.62
130	146.25	5.62
132	58.08	8.70
133	31.52	9.27
135	89.96	5.62
136	36.48	8.96
137	74.02	7.56
139	39.07	10.06
140	46.89	9.50
141	33.67	9.14

Table 5.21: Overall Recognition scores for the scene occlusion8 (Figure 5.18)

Scene points	Membership Values					Results	Remarks				
	Model No.						Model No.				
	1	2	3	4	5		1	2	3	4	5
1	.21	.07	.51	.36	.27	wrong	6	3	7	4	8
2	.23	.06	.50	.31	.23	wrong	feat	feat	feat	feat	feat
3	.09	.40	.11	.36	.10	wrong	-ures	-ures	-ures	-ures	-ures
5	.12	.06	.59	.65	.13	correct	out	out	out	out	out
7	.12	.07	.59	.65	.13	correct	of	of	of	of	of
8	.16	.20	.37	.33	.26	wrong	13	9	19	16	15
9	.16	.40	.11	.33	.26	wrong	have	have	have	have	have
10	.09	.33	.11	.36	.10	correct	been	been	been	been	been
15	.42	.16	.33	.35	.28	jn. point[1]	corre	corre	corre	corre	corre
20	.21	.33	.11	.38	.27	correct	-ctly	-ctly	-ctly	-ctly	-ctly
21	.09	.14	.43	.36	.10	jn. point	class	class	class	class	class
37	.39	.30	.27	.18	.59	correct	-ified	-ified	-ified	-ified	-ified
38	.39	.30	.27	.17	.60	correct					
39	.39	.30	.27	.18	.59	correct					
46	.21	.32	.17	.38	.27	wrong					
47	.21	.32	.17	.38	.27	wrong					
54	.09	.43	.13	.36	.10	wrong					
55	.20	.20	.37	.50	.24	jn. point					
57	.42	.10	.22	.30	.12	wrong					
62	.09	.39	.12	.36	.10	correct					
63	.16	.40	.11	.33	.26	correct					
64	.16	.43	.13	.33	.26	correct					
66	.23	.06	.50	.31	.23	correct					
68	.16	.06	.60	.31	.23	correct					
69	.16	.06	.60	.31	.23	correct					
70	.23	.06	.50	.31	.23	correct					
73	.35	.30	.21	.28	.42	wrong					
75	.36	.20	.37	.28	.36	correct					
82	.25	.39	.12	.35	.42	wrong					
83	.12	.06	.59	.66	.13	correct					
84	.12	.06	.59	.65	.13	correct					
88	.21	.33	.11	.38	.27	wrong					
89	.21	.32	.17	.38	.27	wrong					
90	.24	.26	.12	.17	.60	correct					
91	.39	.30	.27	.17	.59	correct					
92	.39	.30	.27	.17	.59	correct					
93	.39	.30	.27	.18	.59	correct					
94	.29	.39	.13	.17	.55	correct					
95	.21	.06	.51	.31	.23	wrong					
100	.20	.14	.43	.36	.24	wrong					
102	.20	.06	.50	.31	.23	jn. point					
103	.45	.27	.09	.22	.35	wrong					
104	.21	.32	.17	.35	.27	wrong					
105	.45	.24	.24	.18	.55	wrong					
112	.50	.07	.52	.18	.36	wrong					
113	.50	.07	.52	.18	.36	wrong					
117	.42	.16	.28	.35	.24	correct					
127	.53	.06	.48	.17	.38	correct					

Table 5.21(Contd.)

Scene points	Membership Values Model No.					Results	Remarks Model No.				
	1	2	3	4	5		1	2	3	4	5
130	.53	.06	.48	.17	.38	correct					
132	.52	.10	.30	.24	.22	jn. point					
133	.42	.10	.22	.30	.12	wrong					
135	.39	.30	.27	.17	.59	wrong					
136	.52	.11	.13	.24	.21	wrong					
137	.46	.27	.09	.14	.35	jn. point					
139	.61	.11	.13	.42	.21	correct					
140	.50	.11	.18	.34	.19	correct					
141	.42	.10	.22	.30	.12	correct					

1 jn. point stands for junction point

In case study 3, all other experimental details are same as case study 1 and case study 2. In this case, the recognition scores considering single classification and multiple classification are shown in Table 5.22 and Table 5.23 respectively.

From the records of Table 5.21, 5.22 and 5.23, we see that in Figure 5.18, the detection (under hard partitioning) of model4 in the scene is not satisfactory.

Table 5.22:Detailed Recognition scores for *occlusion8* (Figure 5.18)

Hard Partitioning										Fuzzy Partitioning				
scene points learned & correctly classified					scene points miscla-ssified					extra scene points correctly classified				
Model No.					Model No.					Model No.				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
117, 127, 130, 139, 140, 141	62, 63, 64	66, 68, 69, 70, 75, 83, 84	6, 7, 10, 20	37, 38, 39, 90, 91, 92, 93, 94	1, 2, 3, 112*, 113*	57, 103, 104*, 105*	73*, 82*	8*, 9*, 54, 133*, 135*, 136*	46*, 47, 88*, 89, 95*, 100	112, 113	104	-	8, 9, 54, 133, 136	-

* indicates points those were learned

Table 5.23(a):Possibility of presence of the models in the occluded scene occlusion8 as per Table 5.1 (Hard Partitioning)

Model No.									
1		2		3		4		5	
l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.
fair	.7	fair	.7	fair	.7	poor	.3	fair	.7

1 : l.v. stands for linguistic value

2 : p.d. stands for possibility degree

Table 5.23(b):Possibility of presence of the models in the occluded scene occlusion8 as per Table 5.1 (Fuzzy Partitioning)

Model No.									
1		2		3		4		5	
l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.
fair	.7	fair	.7	fair	.7	fair	.7	fair	.7

1 : l.v. stands for linguistic value

2 : p.d. stands for possibility degree

But when we consider fuzzy partitioning, the detection of model4 in the scene is significantly improved and the overall performance of the vision system is very much satisfactory.

5.4 Conclusion

Here we have successfully applied a neuro fuzzy reasoning to recognize partially occluded objects. We have tested our proposed scheme with several complex scenes and we have obtained very promising results. Throughout the case studies, we have seen that our proposed vision system has not generated any alarming signal which says that features of model object j are detected as features of model object i where $i \neq j$. Thus, we claim that our proposed vision system works as per the specifications of design study. The present scheme will be very suitable for industrial robot vision.

Chapter 6

Further Fusion of the first kind via Gene

In the earlier chapters (i.e. Chapter 3 [79], Chapter 4 [82] and Chapter 5 [83]) we have discussed the fusion between neural network and fuzzy reasoning technique. The marriage of genetic algorithm and fuzzy rule is still in its beginning. But the number of papers is beginning to grow as indicated by the number of contributions in [119,120,121]. In this chapter [81,84] we further try to improve our fusion methodology as discussed in the previous chapters (Chapter 4 and 5) by genetic algorithm. Neural nets provide proper internal representation among the connections. A genetic algorithm (GA) is best viewed as a function optimizer. To optimize neural net connection weights by GA, first we assume that the connectivity of the net has already been defined and that each connection weight can take a predefined finite range of settings. Next we encode each weight by a population of binary strings or some form that will allow us to manipulate among strings for achieving one optimal value for each weight. To tackle the occluded object recognition / pattern classification problems, we realize the new interpretation of MFI (i.e. Equation (2.16)) through multilayer perceptron. But, the learning scheme of the network is based on genetic algorithm which avoids the conventional backpropagation technique which depends on the sequence of training data and the choice of the learning rate. In the conventional backpropagation algorithm the objective function is taken as a sum of squared errors (MSE criterion). Whereas, for learning of MLP using genetic algorithm the fitness function is defined as E minus the MSE, where E is the maximum possible value of the error function. Such Gene based fusion methodologies is first applied on occluded object recognition problems which are already discussed in Chapter 5. Subsequently, we use the same for pattern classification problems which are already discussed in Chapter 4. For further improvement of MLP performance, we modify the said fitness function by subtracting an additional error term that measures the 'unsmoothness' of the connection weights of the neural net. In fact this technique of adding an extra error term is regularization which is commonly used to stabilize solutions to ill-posed problems [96].

The 'smoothing' constraint is incorporated into the fitness function of the network to reflect the neighborhood correlation and to seek those solutions which have smooth connection weights. At the learning stage of the neural network, fuzzy linguistic statements have been used as we have done earlier. Once learned, the nonfuzzy features can be fuzzified using fuzzy masking whose essential task is to manage uncertainty in ultimate classification / recognition task (see Section 4.8 of Chapter 4).

6.1 Brief Review on Genetic Algorithms on Neural Networks

One cycle of genetic algorithm proceeds as follows:

1) The genetic algorithm randomly generates a population of binary strings, each of which represents one set of connection weights for an entire net. Thus, each connection

weight is viewed as a separate parameter and each binary string is simply the concatenated connection weight parameters for a single net.

2) An interpreter takes each binary string and uses it to set the connection weights in the network.

3) The network is then run in a feed-forward fashion for each training pattern. The sum of the squared error is then accumulated; this value represents the 'fitness value' of the binary genotype. This performance measure is used to determine how to allocate reproductive opportunities. The smaller the error, the more reproductive opportunities a genotype will receive. Since, each genotype is competing with all the other genotypes in the population for reproductive opportunities, its actual chance of reproducing will be affected by the average performance of the population as a whole.

4) After the initial random population is generated, new 'genotypes' are created by recombination and then evaluated. The 'better' genotypes displace those with poorer performance evaluations and thus gain the chance to reproduce.

The genetic algorithm and the neural net are maintained as separate components. An interpreter simply translates the binary string into a neural network and the network is run using the input data it is given. Also, one genetic evaluation takes less computation time than one backpropagation cycle since only the feed forward computations need to be done.

6.2 Implementation of the new interpretation of MFI on MLP type neural network

Let us consider Equation (2.16). The first expression of Equation (2.16) is one law of implication and the second expression of Equation (2.16) is another law of implication. Both these laws of implication can be independently realized through two neural net-

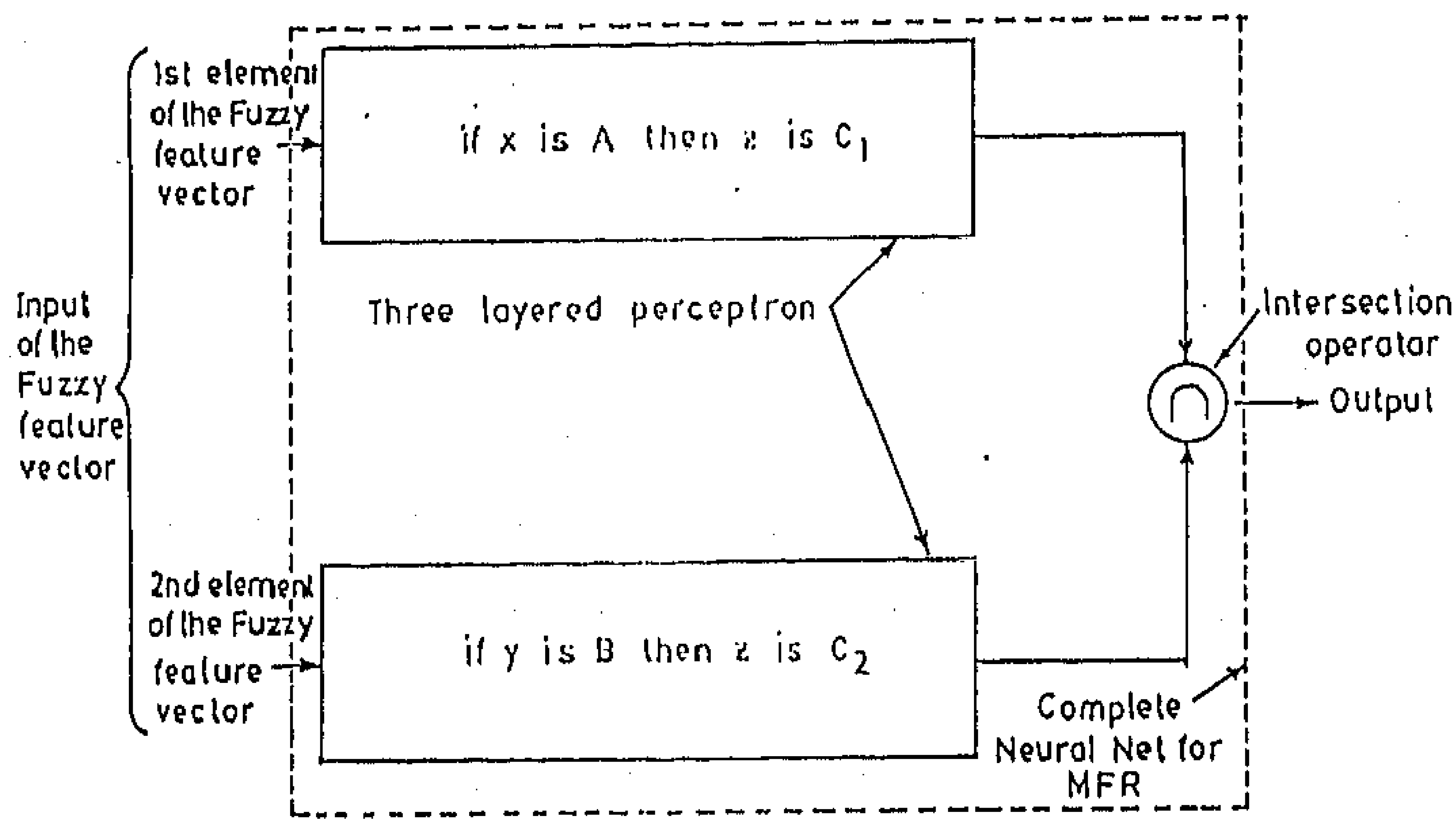


Fig. 6.1: Realization of two-dimensional MFR through MLP type neural network

works which are basically the conventional three layered perceptrons (see Figure 6.1). The input of each neural network is the antecedent part of each **If Then** clause of Equation (2.16). The antecedent part is represented by a fuzzy set. The reference output of each network is the consequent part of each clause of Equation (2.16). The consequent part is also represented by a fuzzy set which basically represents the possibility of occurrence of different classes of patterns in the pattern space. Note that the input and output of the network of Figure 6.1 are same as Figure 4.1. The only difference between the approach proposed in the present chapter and the approaches proposed in Chapter 4 and 5 occur at the stage of learning the weights of the neural network. That means the backpropagation learning algorithm of Chapter 4 and 5 is replaced by the GA learning algorithm in the present chapter. Other features of the networks (as shown in Figure 6.1) are same as the conventional MLP network. Each network is trained independently using a set of fuzzy **If Then** statements and genetic learning scheme [16]. Once the networks are trained we can combine the output of each network by an intersection ($\bigcap$)

operator (see Figure 6.1). Thus, if we consider Equation (2.16) we can realize MFR through a network configuration shown in Figure 6.1. If we have n-dimensional law of fuzzy implication, we can realize MFR through a network which consists of n number of independent 3 - layered perceptrons which are trained using a set of fuzzy **If Then** statements and the genetic algorithms.

The outputs of n networks are combined through intersection ($\bigcap$) operator. In this chapter, as we have considered object recognition / pattern classification on $\Re^2$, we always follow the network configuration of Figure 6.1.

6.3 Genetic algorithm (GA) based learning environment

In this section we proceed with a concise discussion of the paradigm of genetic computations.

6.3.1 Genetic algorithms for global optimization

The GA is a well known technique [16]. It is aimed at finding a global maximum of a function of many variables by performing a genetic-inclined search of the space. The variables are usually represented as strings of bits. The search is performed by modifying the binary strings with an aim to increase their fitness. The basic steps of GA include reproduction, crossover and mutation. At the first step the strings are reproduced contributing to the next generation according to their fitness. The probability of reproduction of the individual string is activated through the roulette mechanism. During crossover, the binary strings are mated by exchanging their substrings. Finally, the mutation mechanism changes (complements) randomly some of the bits of the strings.

The modification of this form produces a random walk across the search space. The string with the highest fitness encountered within all the populations is considered to become a solution to the optimization problem. GA explores the search space in a global fashion by studying the entire population of the strings.

6.3.2 Backpropagation Vs. GA

In the following section, we propose a methodology, based on genetic algorithms, which is capable of searching the global optimum solution. The proposed algorithm selects the appropriate weights randomly from a set of potential solutions and there is no need of the backpropagation technique. As a result, the algorithm is computationally less expensive, and the selection of learning rate η and the derivability of objective function are no longer required. Moreover, it considers only the global effect of the training set in updating weights; therefore it does not depend on the sequence of the training data set.

6.4 Four basic features of GA

6.4.1 Fitness function

In GA, the value of the objective/fitness function dictates the survival/ suitability of a string. A highly fitted string should have higher fitness value and it should result in low classification error. As shown below, a decreasing function F(E) of overall error E is considered as the fitness function.

$$F(E) = E_{max} - E \tag{6.1}$$

where E_{max} is a very large value which is much larger than the error value we have experienced during the experiments at the learning stage of Chapter 4 and Chapter 5 and E is the error term used as the objective function in multilayer perceptron and is defined as,

$$E = \frac{1}{2P}\Sigma_i\Sigma_j(t_{ij} - v_{ij})^2 \tag{6.2}$$

where t_{ij} is the target value at the output node j for the i^{th} training pattern and v_{ij} is the actual value at the output node j for the i^{th} input pattern, P is the no. of training patterns. To convert a minimization problem to an equivalent maximization problem, in GA approach we usually adopt the above mentioned technique [16].

6.4.2 Reproduction/ Selection

According to the fitness function values produced by the strings, the reproduction scheme copies the individual strings into the mating pool for the purpose of crossover and mutation operations. Let f_i be the fitness value obtained for a training set T corresponding to the i^{th} string, i=1,2,...,N. Then the mating pool will consist of n_i copies of S_i, where,

$$n_i = \frac{f_i}{\Sigma f_i} * N.$$

Note that n_i may not be an integer. In that case, we round it off such that $\Sigma_i n_i$ = N.

Let us define the raw fitness f and the scaled fitness f'. To control the number of offspring to the population member with maximum fitness, we choose a scaling relationship to obtain a scaled fitness $f' = C_{mult} * f$. We have chosen C_{mult} to be 2. C_{mult} is the number of expected copies desired for the best population member. At the end of a run, this C_{mult} stretches the fitnesses of each member. There is one problem with this type of stretching. Any individual of low fitness gets also stretched which is not desirable. To avoid this problem, after every run, we scale up those individuals which have a significant contribution in the next generation by multiplying their fitness values with the C_{mult} and scale down those individuals which do not have significant contributions by dividing their fitness values by C_{mult}. We have chosen C_{mult} to be 2.

6.4.3 Crossover

Since the size of the parameter set and consequently the length of the chromosomes is not small, it is intuitive that the single point crossover operation may not be useful for fast convergence. Therefore, instead of applying a crossover operation at a single point over the entire string, we apply this operation on each substring (chromosomal representation of an individual parameter). The proposed multiple point crossover operation [16] is demonstrated below for the substring of length q=10. The length of the substring may be more than 10 depending upon the need of the problem. Let,

a=1100010101 0100011010 ... 0111110001,
b=1000101110 1110110001 ... 0011010100

be two strings (parents) selected for crossover. Let the random number generated by the crossover operation be 7,5,...,4. Then the newly produced offspring will be

a'=1100010110 0100010001 ... 0111010100,
b'=1000101101 1110111010 ... 0011110001.

In this work, we have encoded each weight value of the neural network by a substring (q) of length 10 which is the chromosomal representation of each weight and the string which is used for genetic operations is made up of such substrings which are arranged in the string according to their connections in the neural network. The weight values are initialized as random floating point numbers; for instance, it may vary between -0.5 and +0.5 [69]. Hence, while encoding the weight values as strings, we have to encode the real as well as the floating point portion of the weight values. We left the first place of the substring for the sign of the weight. If there is a 0 in the first place of the substring then the corresponding weight value is positive. On the other hand, if there is a 1 in the first place of the substring, then the corresponding parameter has negative value. The next two places of the substring are kept for the real part of each parameter. As we have kept two places the real value of each weight can range upto 3. This choice of representing the real value of each weight depends on the need of the problem and absolutely a heuristic judgement. Instead of 2, we may keep 3, 4 or any higher places for the real portion of each weight. The rest of the substring (for our case, it is 7) is kept for the fractional portion of the weight values. Hence, for the present class of problem the neural network can learn weights in the range of -3.99 to +3.99. For different class of problems this range may be changed. The crossover is performed with a high probability (0.8 in our case).

6.4.4 Mutation

The mutation operation is performed with very low probability (P_{mut}), but it is difficult to determine the probability of performing this operation in order to produce good result. We have chosen P_{mut} in the range of 0.001 to 0.0001.

6.5 Algorithm

The block diagram of the proposed algorithm is presented in Figure 6.2. Given the pattern classes, a training data set T is selected for learning the neural network parameters. Initially, a binary string, generated randomly of length pq (substrings of length q are taken for each parameter and there are p number of parameters (weights of the neural net)) is considered as the chromosomal (string) representation for the parameter set associated with the network. A set of N such strings is considered as the initial population. In our experiment, we have assumed N=30. The fitness value corresponding to each string is calculated for the entire training set. A mating pool is created using the reproduction operation with highly fitted strings and a new population of size N for the next generation is then generated by the crossover and the mutation operations on the strings selected randomly from the new mating pool. The learning process of the network is then repeated with the parameter sets corresponding to this new population for the same training set T of patterns, and, consequently, a new population is generated further. The process terminates when the minimum value of the error over a population becomes less than some small preassigned value e (for the present case, it is 0.01). The decoded version of the string having minimum error value then represents the optimum parameter set for the network.

Note that the basic formulation of the problem is same as Section 4.3 of Chapter 4 (for pattern classification problems) and Section 5.2 of Chapter 5 (for occluded object recognition problems).

6.6 Recognition of occluded two-dimensional objects

Keeping all the procedures and experimental setup same as discussed in Section 5.2 and 5.3 of Chapter 5, except for the learning scheme which presently is modified by GA, we perform our experiments on Figure 6.3 and 6.4. The results obtained are listed in Table 6.1 to 6.6. From the results listed in Table 6.1 to 6.6, it is obvious that the overall performance (in terms of recognition) of the vision scheme is quite satisfactory in all cases. For preciseness of presentation, we do not consider other cases as discussed in Chapter 5, but in all other cases which are not explicitly discussed in this chapter we obtain satisfactory results (in terms of recognition of the objects in a scene). Here, for recognition instead of using only fuzzy singleton, we use the concept of fuzzy *singleton* as well as *fuzzy masking* (see Figure 4.2). The basic purpose of using fuzzy masking is already stated in Section 4.7. We realize through experiments that under ideal condition (i.e. features of the same pattern are not significantly shifted as stated in Section 4.7 of Chapter 4) fuzzy singleton and fuzzy masking are equally good. Table 6.1 depicts the results of the recognition scheme where the method of fuzzification is fuzzy singleton. Table 6.2 shows the results where the process of *fuzzification* is *done* with one type of fuzzy masking. Table 6.3 shows the results with a different type of fuzzy masking. The recognition scores considering single classification and multiple classification are shown in Table 6.4, 6.5 and 6.6.

Table 6.1:Detailed Recognition scores for the occluded scenes (fuzzy singleton)

Name of the scene	Hard Partitioning						Fuzzy Partitioning		
	scene points learned & correctly classified			scene points misclassified			extra scene points correctly classified		
	Model No.			Model No.			Model No.		
	1	2	3	1	2	3	1	2	3
Figure 6.3:occlusion1	16, 63, 65, 66, 80	70	9, 10, 12, 26, 32, 37, 40, 41, 43, 47, 76, 77	0*, 20*, 64, 85	24*, 69*, 71 *, 72*	7*, 8*, 14 *, 32*, 38*	64	71,	7, 8, 14, 32, 38
Figure 6.4:occlusion2	10, 12, 14, 37, 41, 42, 55, 64, 66	5, 6, 9, 58	3, 34, 75, 77, 84	13, 15*, 20, 35*, 36*, 39*, 45, 46, 53, 54*, 62, 65*	4*, 7*, 8*, 67*, 69*, 72, 73, 74	22*, 23*, 33*, 76, 78	20, 35, 36, 53, 54, 62, 65, 72	72	22, 23, 33, 76

* indicates points those were learned

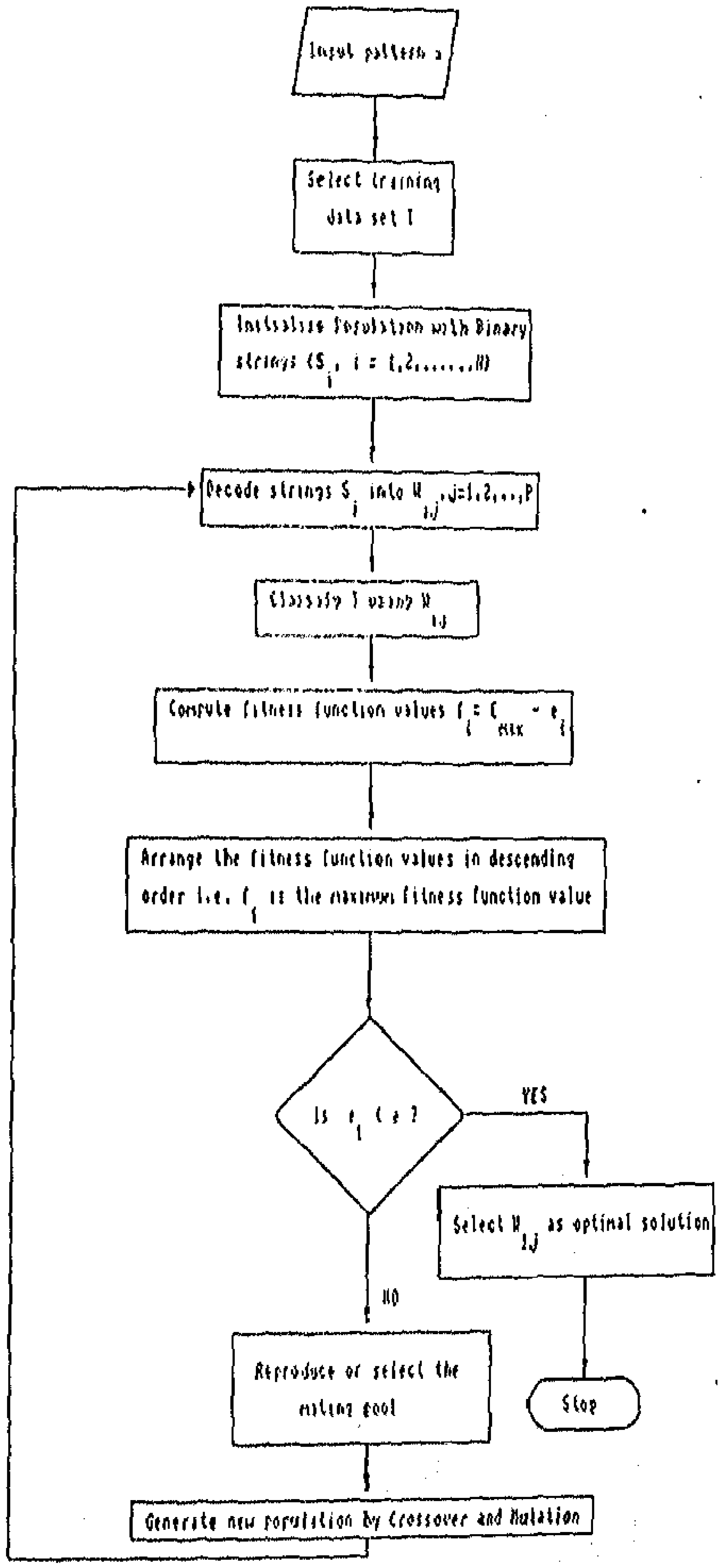


Fig. 6.2: Flow Chart for the learning scheme

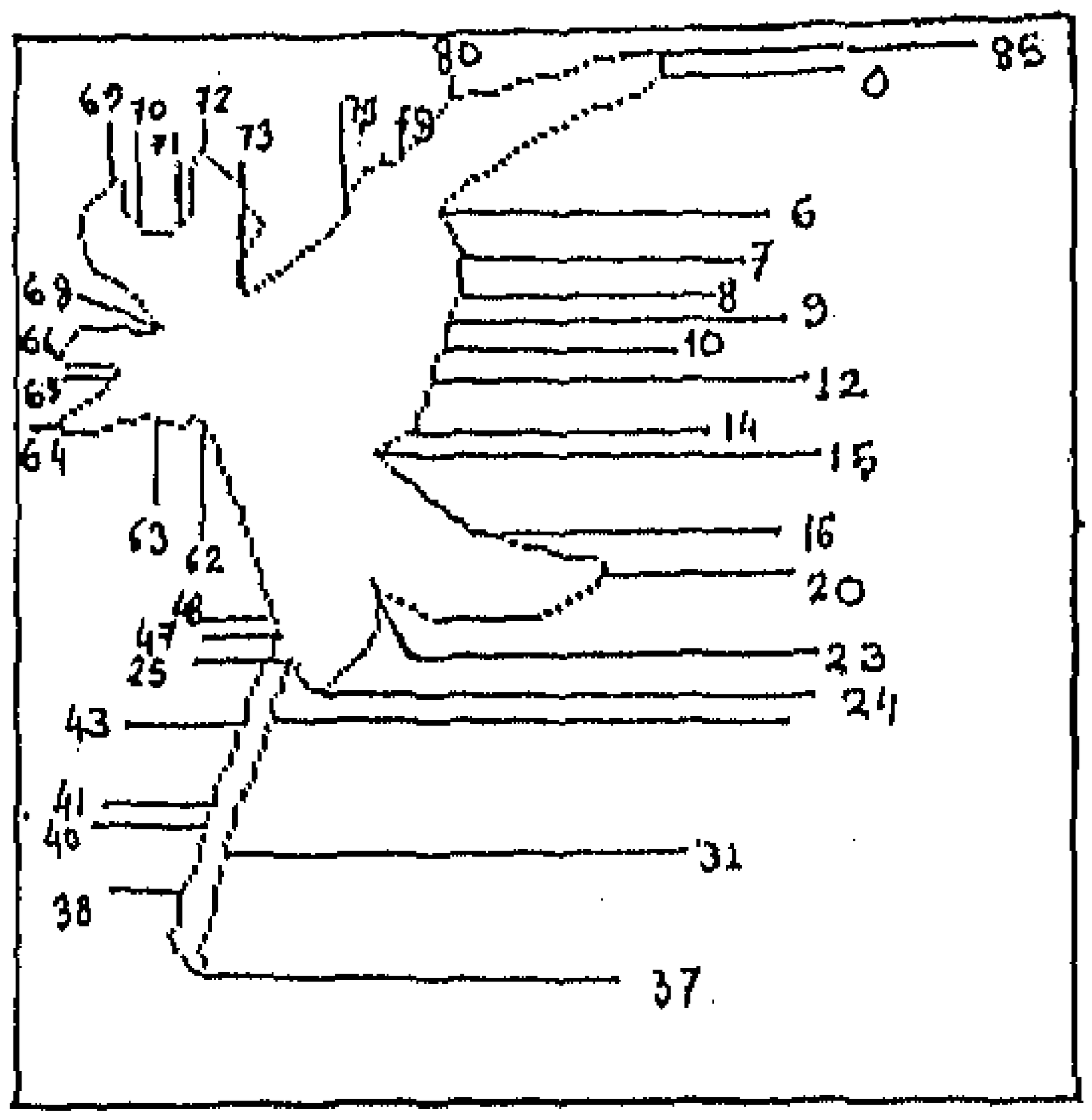


Fig. 6.3: Occlusion1

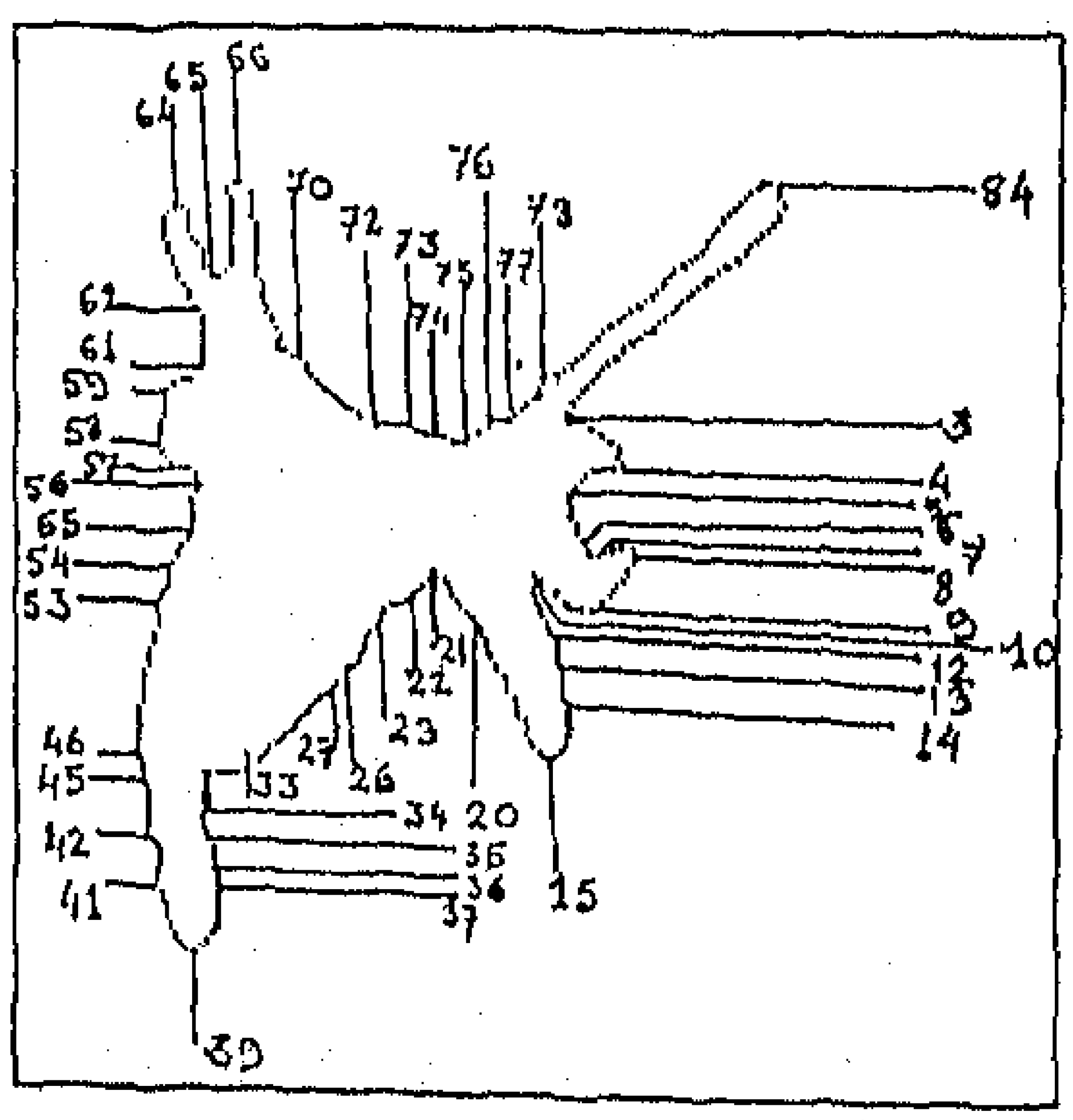


Fig. 6.4: Occlusion2

Table 6.2:Detailed Recognition scores for the occluded scenes

(fuzzy masking .2 .5 .7 1 .7 .5 .2)

Name of the scene	Hard Partitioning						Fuzzy Partitioning		
	scene points learned & correctly classified			scene points misclassified			extra scene points correctly classified		
	Model No.			Model No.			Model No.		
	1	2	3	1	2	3	1	2	3
Figure 6.3:occlusion1	0, 20, 64, 65, 66	24	7, 8, 9, 10, 12, 26, 32, 38, 40, 41, 47	16*, 63*, 80*, 85*	69*, 72*	14*, 37*, 43*, 76*, 77*	65	69, 72	37, 43, 76, 77
Figure 6.4:occlusion2	10, 64, 65, 66	5, 6, 7, 9, 57, 58	26, 27, 34	12, 13, 14, 15*, 20, 35*, 36*, 37, 39*, 41, 42, 45, 46, 53, 54, 55*, 62*	4*, 8*, 59*, 72, 73, 74	3*, 22*, 23*, 33*, 75*, 76, 77*, 78, 84*	15, 39, 41, 42,	4, 8, 59, 72, 73, 74	5, 22, 33, 34, 75, 77

* indicates points those were learned

Table 6.3:Detailed Recognition scores for the occluded scenes (fuzzy masking .1 .3 .5 .7 .9 1 .9 .7 .5 .3 .1)

Name of the scene	Hard Partitioning						Fuzzy Partitioning		
	scene points learned & correctly classified			scene points misclassified			extra scene points correctly classified		
	Model No.			Model No.			Model No.		
	1	2	3	1	2	3	1	2	3
Figure 6.3:occlusion1	64, 65, 66	24, 70, 71, 72	7, 8, 9, 10, 12, 26, 32, 38, 40, 41, 47, 76, 77	0*, 16*, 20*, 63*, 80*, 85	69*	14*, 37*, 43*	0, 20	69	14, 37, 43
Figure 6.4:occlusion2	10, 62, 64, 65, 66	6, 7, 57, 58, 59	26, 27, 33, 75	12, 13, 14, 15*, 20, 35*, 36*, 37, 39*, 41, 42, 45, 46, 53, 54, 55*	4*, 5*, 8*, 9*, 72, 73, 74	3*, 22*, 23*, 34*, 76, 77*, 78, 84*	12, 13, 15, 39, 41, 42, 53	4, 6, 8, 9, 72, 73, 74	22, 76, 77, 78

Table 6.4:Possibility of presence of the models in the occluded scenes as per Table 5.1 (Fuzzy singleton)

Name of of the scene	Hard Partitioning						Fuzzy Partitioning					
	Model No.						Model No.					
	1		2		3		1		2		3	
	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.
Figure 6.3:occlusion1	fair	.7	fair	.7	fair	.7	fair	.7	fair	.7	good	1
Figure 6.4:occlusion2	good	1	fair	.7	poor	.3	good	1	fair	.7	fair	.7

1 : l.v. stands for linguistic value

2 : p.d. stands for possibility degree

Table 6.5:Possibility of presence of the models in the occluded scenes as per Table 5.1 (Fuzzy masking .2 .5 .7 1 .7 .5 .2)

Name of of the scene	Hard Partitioning						Fuzzy Partitioning					
	Model No.						Model No.					
	1		2		3		1		2		3	
	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.
Figure 6.3:occlusion1	fair	.7	poor	.3	fair	.7	fair	.7	fair	.7	good	1
Figure 6.4:occlusion2	fair	.7	fair	.7	poor	.3	fair	.7	good	1	fair	.7

Table 6.6:Possibility of presence of the models in the occluded scenes as per Table 5.1 (Fuzzy masking .1 .3 .5 .7 .9 1 .9 .7 .5 .3 .1)

Name of of the scene	Hard Partitioning						Fuzzy Partitioning					
	Model No.						Model No.					
	1		2		3		1		2		3	
	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.	l.v.	p.d.
Figure 6.3:occlusion1	poor	.3	fair	.7	good	1	fair	.7	fair	.7	good	1
Figure 6.4:occlusion2	fair	.7	fair	.7	poor	.3	good	1	good	1	fair	.7

6.7 Improvement of the network performance using regularization

A neural network for pattern classification usually contains many connection weights and the training of a network is to search for a "good" solution in the weight space. Since the weight space of a neural network is usually huge, many solutions should exist for any training data set of practical size. Among these solutions, those with 'smoother' connection weights should be able to capture more reliable features and thus generalize better. It is thus desirable to choose a 'smoother' solution during the training process.

To locate a 'smoother' solution, the GA can be applied to minimize a new error function which include not only the normal square error term but an extra error term that measures the 'unsmoothness' of weight vectors. In fact this technique of adding an extra error term is regularization which is commonly used to stabilize solutions to ill-posed problems [96].

The measure of the 'unsmoothness' of the weight vectors can be written as,

$$\Sigma_{i=2}^{I}[W_{ji} - W_{j(i+1)} - W_{j(i-1)}]^2$$

And the new objective function becomes,

$$E_{new} = E + \lambda * \Sigma_{j=1}^{H}\Sigma_{i=2}^{I}[W_{ji} - W_{j(i+1)} - W_{j(i-1)}]^2$$

where λ is a small number and E is defined earlier by equation (6.2) of section 6.4.1. I is the number of nodes in a particular layer and H is the number of nodes in the next layer of the neural net.

Now, the modified fitness function can be written as, $F(E)_{new} = E_{max} - E_{new}$ Thus, by maximizing the fitness function in a particular generation , we are basically minimizing the 'unsmoothness' of the weights.

6.8 Pattern Classification Problems

To test the effectiveness of the neuro genetic approach to MFR for pattern classification, we have considered two classes of problems, viz., classification of synthetic data, classification of vowels. We report the results obtained in the said two domains in the following subsections. During our experiments on pattern classification problems, we have experienced many unsmooth connection weights which we want to replace by smooth connection weight. The basic approach to obtain smooth connection weight as stated in section 6.7 is general; but for demonstration we have considered a class of problems.

6.8.1 Classification of synthetic data without regularization

Let us consider the synthetic data shown in Figure 2.8 to 2.11. The rules, distribution pattern, number of hidden nodes and the classification scores are shown in Table 6.7 to 6.10. From Figure 2.8, it is obvious that data are very much overlapped. Hence we have considered the classification results under hard partitioning and fuzzy partitioning. Whereas, in Figure 2.9, 2.10 and 2.11 the data are not overlapped. Hence, in such cases, we have considered classification results under hard partitioning only. From Table 6.7 and 6.10, it is obvious that classification results for Figure 2.8 and Figure 2.11 are very satisfactory. But from Table 6.8 and 6.9, we understand that classification results of Figure 2.9 and 2.10 are not satisfactory. To improve the classification results of Figure 2.9 and 2.10 in section 6.8.3 we report the results of classification using the concept of regularization. At the time of classification, we consider exhaustive data for each Figure.

Table 6.7: Results of Fig. 2.8

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent							
		A	B	C	D	E	F	G	H
NN_1	If F_1 is Small Then	Very High	Low	Nil	More or Less High	Very Low	Nil	Very Low	Very Low
	If F_1 is Medium Then	Low	Very High	Medium	Very Low	High	Medium	Very Low	Very Low
	If F_1 is Big Then	Nil	Nil	High	Nil	Nil	High	Very Low	Very Low
NN_2	If F_2 is Small Then	Very High	Very High	Very High	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is Medium Then	Very Low	Very Low	Very Low	High	High	More or Less High	Very Low	Very Low
	If F_2 is Big Then	Nil	Nil	Nil	Low	More or Less High	High	Very Low	Very Low

Linguistic values learned

	Antecedent	Consequent							
		A	B	C	D	E	F	G	H
NN_1	If F_1 is Small Then	.98	.02	.054	.672	.064	.017	.003	.015
	If F_1 is Medium Then	.097	.934	.614	.000	.765	.603	.000	.008
	If F_1 is Big Then	.019	.021	.904	.005	.041	.891	.014	.000
NN_2	If F_2 is Small Then	.94	.95	.98	.00	.007	.079	.001	.017
	If F_2 is Medium Then	.007	.001	.107	.886	.862	.719	.010	.020
	If F_2 is Big Then	.038	.029	.088	.164	.76	.80	.001	.050

Fuzzy masking used over the nonfuzzy domain of the features with maximum membership function at a particular instance

{ .2 .5 .7 1 .7 .5 .2 }

Table 6.7(Contd.)

Recognition Score

Hard Partitioning

class A	class B	class C	class D	class E	class F	overall score
94.8 %	76.9 %	56.9 %	40.4 %	75.4 %	95.7 %	75.7 %

Fuzzy Partitioning

class A	class B	class C	class D	class E	class F	overall score
97.4 %	93.1 %	97.4 %	87.6 %	100 %	100 %	96.4 %

Table 6.8: Results of Fig. 2.9

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent						
		A_1	A_2	A_3	B	C	D	E
NN_1	If F_1 is Small Then	Very High	Nil	Nil	Nil	Nil	Very Low	Very Low
	If F_1 is Medium Then	Nil	Very High	Very High	High	High	Very Low	Very Low
	If F_1 is Big Then	Nil	Very High	Very Low	More or Less High	More or Less High	Very Low	Very Low
NN_2	If F_2 is Small Then	Very High	Very High	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is Medium1 Then	More or Less High	Nil	Nil	Very High	Nil	Very Low	Very Low
	If F_2 is Medium2 Then	More or Less High	Nil	Nil	Nil	Very High	Very Low	Very Low
	If F_2 is Big Then	Very High	Very High	Nil	Nil	Low	Very Low	Very Low

Linguistic values learned

	Antecedent	Consequent						
		A_1	A_2	A_3	B	C	D	E
NN_1	If F_1 is Small Then	.9	.023	.046	.070	.04	.00	.00
	If F_1 is Medium Then	.03	.99	.95	.90	.89	.01	.00
	If F_1 is Big Then	.04	.95	.03	.71	.72	.00	.00
NN_2	If F_2 is Small Then	.95	.95	.00	.02	.00	.00	.00
	If F_2 is Medium1 Then	.81	.09	.00	.96	.06	.02	.00
	If F_2 is Medium2 Then	.71	.04	.00	.02	.95	.00	.00
	If F_2 is Big Then	.98	.95	.00	.01	.10	.00	.00

Table 6.8(Contd.)

Fuzzy masking used over the nonfuzzy domain of the features with maximum membership function at a particular instance

$\{ .2\ .5\ .7\ 1\ .7\ .5\ .2 \}$

Recognition Score (Hard Partitioning)

class A	class B	class C	overall score
86.7 %	1 %	17 %	54.9 %

Table 6.9: Results of Fig. 2.10

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent					
		A_1	A_2	B_1	B_2	C	D
NN_1	If F_1 is Small Then	Very High	Nil	Nil	Nil	Very Low	Very Low
	If F_1 is Medium Then	Low	High	More or Less High	Nil	Very Low	Very Low
	If F_1 is Big Then	Nil	Very Low	More or Less High	Very High	Very Low	Very Low
NN_2	If F_2 is Small Then	Very High	Very High	Nil	Very High	Very Low	Very Low
	If F_2 is Medium Then	Very High	Low	Very High	Nil	Very Low	Very Low
	If F_2 is Big Then	Low	Very High	Nil	Nil	Very Low	Very Low

Linguistic values learned

	Antecedent	Consequent					
		A_1	A_2	B_1	B_2	C	D
NN_1	If F_1 is Small Then	.99	.01	.01	.00	.00	.00
	If F_1 is Medium Then	.06	.94	.66	.06	.02	.00
	If F_1 is Big Then	.015	.012	.72	.96	.03	.03
NN_2	If F_2 is Small Then	.96	.98	.00	.95	.01	.00
	If F_2 is Medium Then	.99	.10	.97	.03	.00	.00
	If F_2 is Big Then	.06	.98	.01	.04	.01	.00

Fuzzy masking used over the nonfuzzy domain of the features with maximum membership function at a particular instance

$$\{ .2 \ .5 \ .7 \ 1 \ .7 \ .5 \ .2 \}$$

Recognition Score (Hard Partitioning)

class A	class B	overall score
85.08 %	65 %	75.8 %

Table 6.10: Results of Fig. 2.11

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent						
		A	B_1	B_2	B_3	B_4	C	D
NN_1	If F_1 is Small Then	Nil	Very High	Very High	Nil	Nil	Very Low	Very Low
	If F_1 is Medium Then	Very High	More or Less High	More or Less High	More or Less High	More or Less High	Very Low	Very Low
	If F_1 is Big Then	Nil	Nil	Nil	Very High	Very High	Very Low	Very Low
NN_2	If F_2 is Small Then	Nil	Very High	Nil	Nil	Very High	Very Low	Very Low
	If F_2 is Medium Then	Very High	More or Less High	More or Less High	More or Less High	More or Less High	Very Low	very Low
	If F_2 is Big Then	Nil	Nil	Very High	Very High	Nil	Very Low	Very Low

Linguistic values learned

	Antecedent	Consequent						
		A	B_1	B_2	B_3	B_4	C	D
NN_1	If F_1 is Small Then	.04	.94	.94	.07	.05	.01	.00
	If F_1 is Medium Then	.95	.70	.68	.70	.70	.00	.00
	If F_1 is Big Then	.06	.02	.04	.97	.95	.00	.01
NN_2	If F_2 is Small Then	.02	.98	.04	.00	.96	.00	.00
	If F_2 is Medium Then	.92	.75	.69	.73	.66	.00	.00
	If F_2 is Big Then	.03	.10	.97	.97	.01	.00	.00

Fuzzy masking used over the nonfuzzy domain of the features with maximum membership function at a particular instance

$$\{ .2\ .5\ .7\ 1\ .7\ .5\ .2 \}$$

Recognition Score (Hard Partitioning)

class A	class B	overall score
97 %	100 %	99.3 %

6.8.2 Classification of vowels without regularization

After achieving satisfactory results on synthetic set of data, we apply the proposed scheme for the vowel recognition problems of an Indian language, namely Bengali (see Figure 2.21) which is already discussed in Section 2.7.2 of Chapter 2. Table 6.11 shows the classification results of Bengali data.

6.8.3 Classification of synthetic data with regularization

Here we have considered the weight smoothing scheme discussed in section 6.7. In Figure 6.5(a,b) we have shown the weight plot (before and after smoothing) for Figure 2.9. Table 6.12 shows the classification result of Figure 2.9. Classification result of Figure 2.9 has significantly improved using the concept of regularization. Similarly, in Table 6.13 we have shown the improved version of classification of Figure 2.10. In both the cases, we have used a decreasing staircase function for mutation and a decreasing staircase function for the multiplier (λ) (see Figure 6.6 and 6.7). The intention behind the selection of such decreasing function is to have better convergence. The decreasing functions are basically the design heuristics.

6.8.4 Classification of Bengali vowel with regularization

Table 6.14 shows the improved classification results of the vowel data of Figure 2.21. In this case study also we have used a decreasing staircase function for mutation and a decreasing staircase function for the multiplier (λ) (see Figure 6.8) as we have done in Section 6.8.3.

Table 6.11: Results of Bengali Vowels of Fig. 2.21

Number of Hidden nodes = 6 (for each 3 - layered neural network)

Rules used for learning the net

	Antecedent	Consequent								
		u	o	ə	a	ae	e	i	g	h
NN_1	If F_1 is Small Then	Very High	Medium	Nil	Nil	Nil	Medium	Very High	Very Low	Very Low
	If F_1 is Medium Then	Nil	Very High	Medium	Very Low	Very Low	More or Less High	Nil	Very Low	Very Low
	If F_1 is Big Then	Nil	Very Low	High	Very High	Very High	Nil	Nil	Very Low	Very Low
NN_2	If F_2 is Small Then	High	Very High	Very High	Very Low	Nil	Nil	Nil	Very Low	Very Low
	If F_2 is More or Less Medium Then	Very High	Very High	Very High	High	Very Low	Nil	Nil	Very Low	Very Low
	If F_2 is Medium Then	Very Low	Nil	Nil	Very Low	Very High	High	Nil	Very Low	Very Low
	If F_2 is Big Then	Nil	Nil	Nil	Nil	Very Low	More or Less High	Very High	Very Low	Very Low

Linguistic values learned

	Antecedent	Consequent								
		u	o	ə	a	ae	e	i	g	h
NN_1	If F_1 is Small Then	.94	.60	.03	.00	.00	.60	.98	.01	.00
	If F_1 is Medium Then	.12	.91	.58	.01	.02	.68	.01	.01	.00
	If F_1 is Big Then	.00	.03	.97	.96	.97	.01	.01	.00	.00
NN_2	If F_2 is Small Then	.89	.94	.97	.06	.02	.02	.03	.00	.00
	If F_2 is More or Less Medium Then	.99	.97	.98.	.88	.06	.01	.00	.00	.00
	If F_2 is Medium Then	.00	.05	.00	.00	.92	.84	.03	.00	.00
	If F_2 is Big Then	.00	.03	.00	.00	.07	.81	.91	.01	.00

Table 6.11(Contd.)

Fuzzy masking used over the nonfuzzy domain of the features with maximum membership function at a particular instance

{ .2 .5 .7 1 .7 .5 .2 }

Recognition Score

Hard Partitioning

class u	class o	classə	class a	class ae	class e	class i	overall score
98.5 %	54.2 %	6 %	46 %	88 %	85.2 %	2 %	54.6 %

Fuzzy Partitioning

class u	class o	classə	class a	class ae	class e	class i	overall score
100 %	100 %	100 %	87 %	100 %	92.6 %	69.7 %	92.9 %

Feature1 (F_1) of Figure 2.9

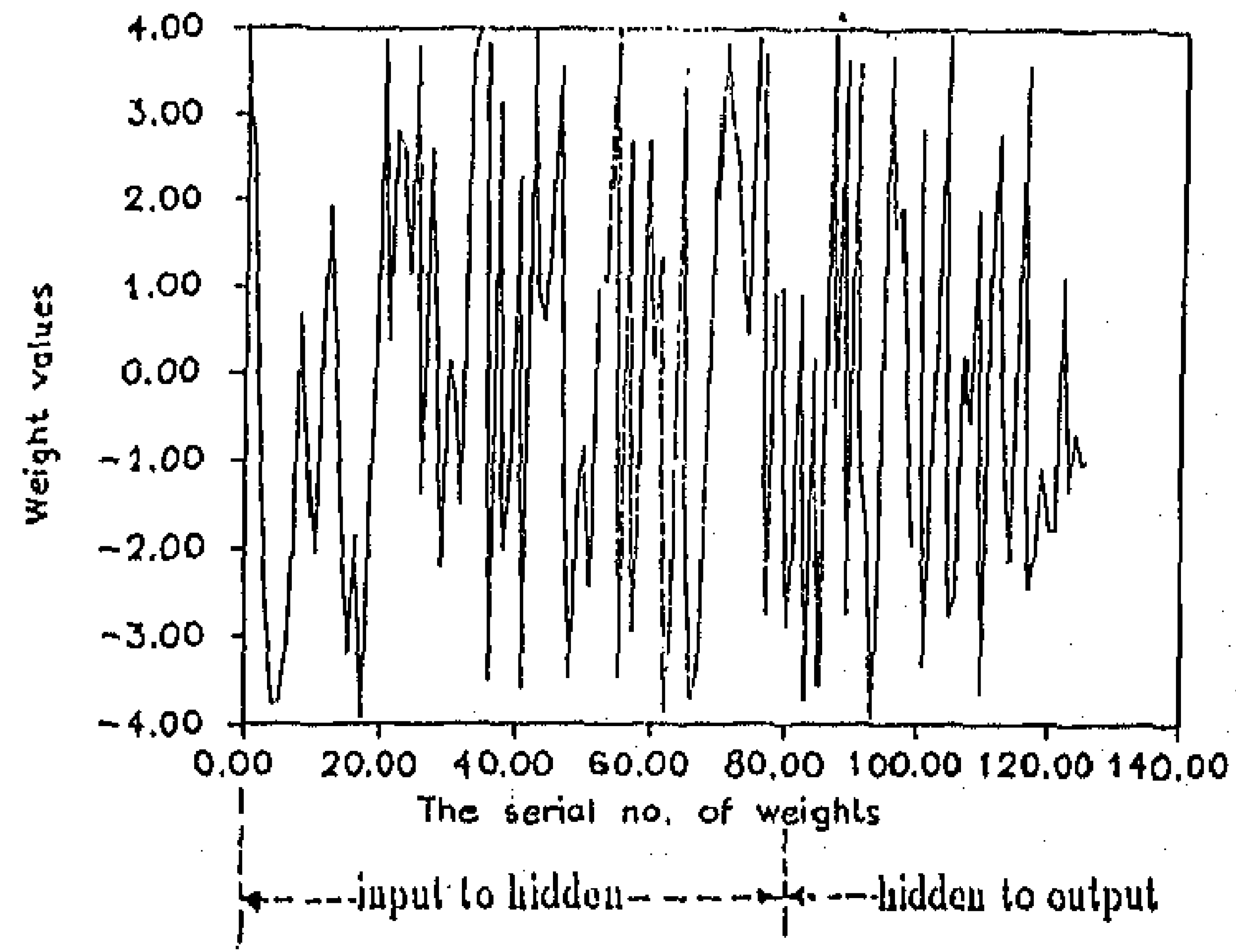


Fig. 6.5(a): Weight Plot Without Regularization

Feature2 (F_2) of Figure 2.9

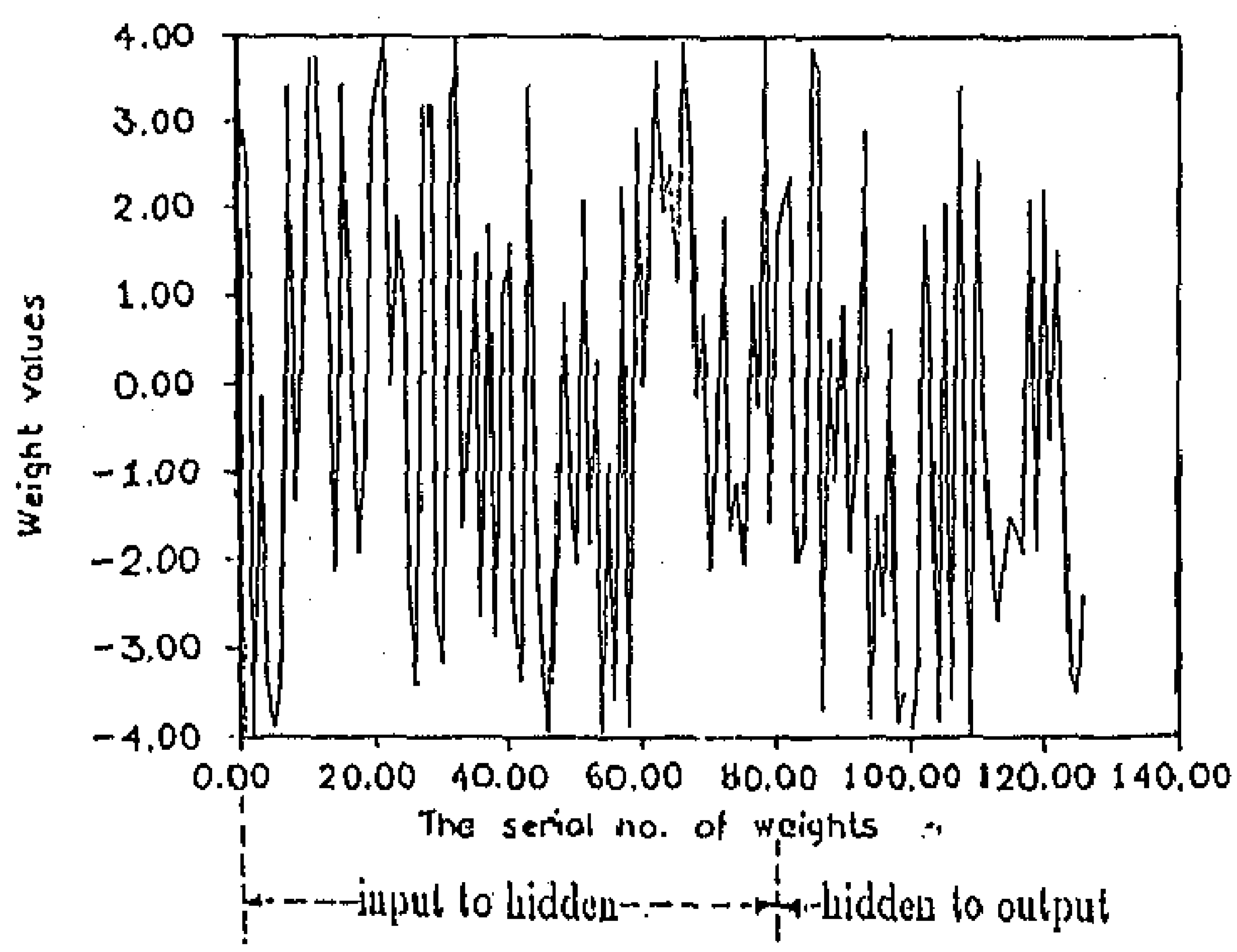


Fig. 6.5(b): Weight Plot Without Regularization

Feature1 (F_1) of Figure 2.9

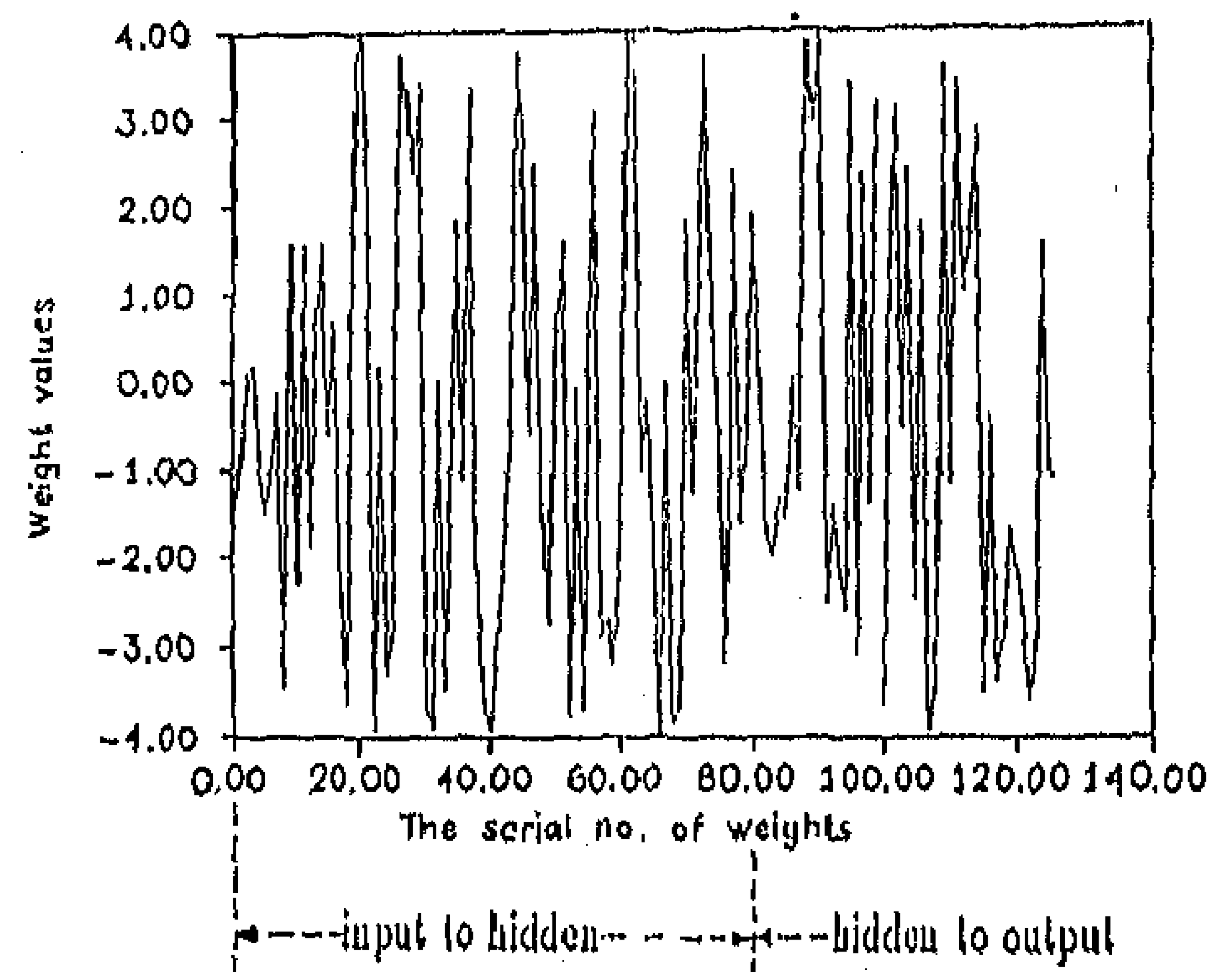


Fig. 6.5(a): Weight Plot With Regularization (Contd.)

Feature2 (F_2) of Figure 2.9

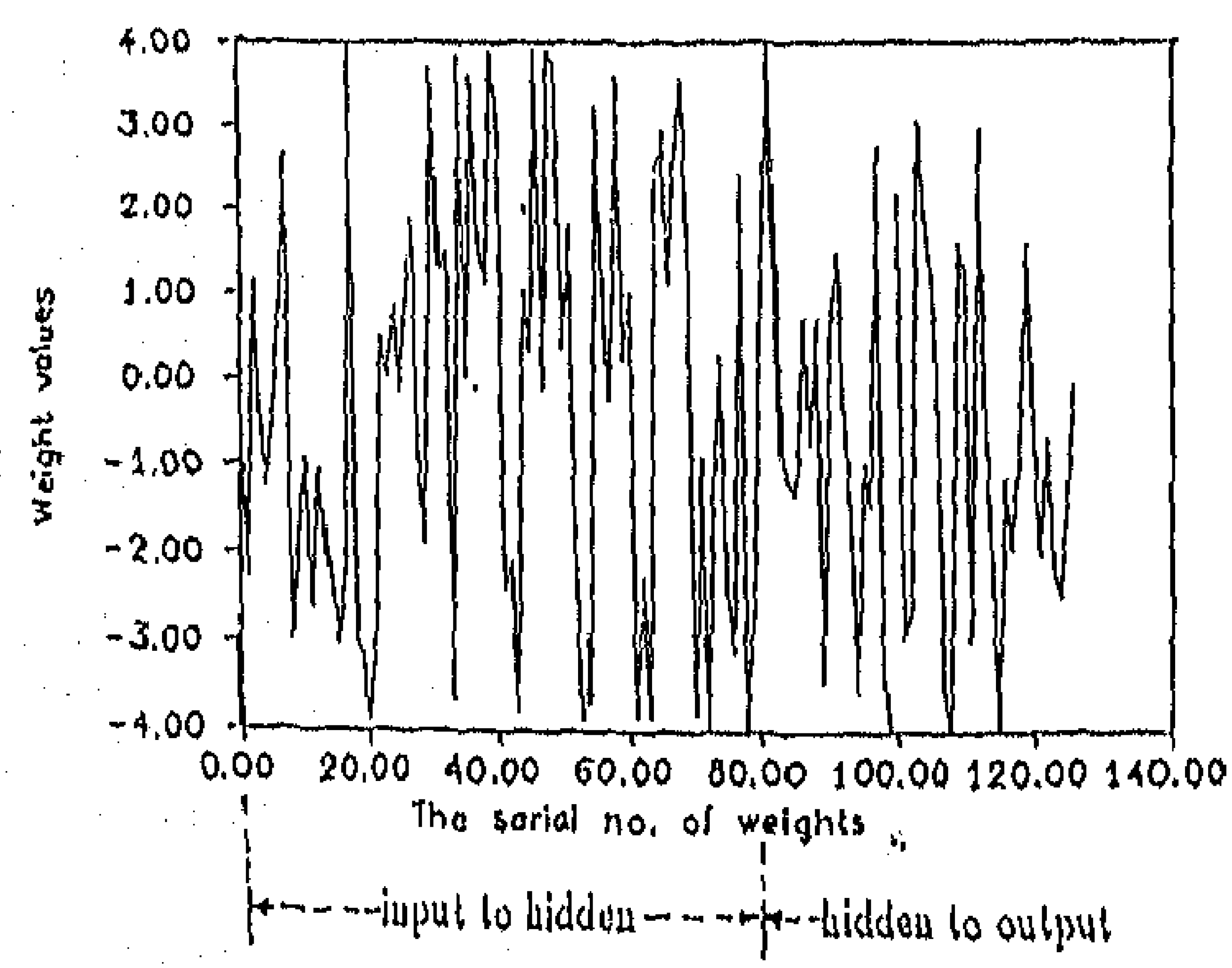


Fig. 6.5(b): Weight Plot With Regularization (Contd.)

Table 6.12: Results of Fig. 2.9

Linguistic values learned (With Regularization)

	Antecedent	Consequent						
		A_1	A_2	A_3	B	C	D	E
NN_1	If F_1 is Small Then	.87	.029	.013	.013	.05	.00	.00
	If F_1 is Medium Then	.04	.99	.97	.93	.99	.01	.00
	If F_1 is Big Then	.06	.93	.02	.69	.71	.00	.01
NN_2	If F_2 is Small Then	.90	.88	.00	.07	.10	.01	.01
	If F_2 is Medium1 Then	.65	.14	.00	.91	.09	.00	.00
	If F_2 is Medium2 Then	.94	.06	.00	.04	.91	.00	.00
	If F_2 is Big Then	.88	.87	.00	.05	.13	.00	.01

Recognition Score (Hard Partitioning)

class A	class B	class C	overall score
93.3 %	68.04 %	100%	89.6 %

Table 6.13: Results of Fig. 2.10

Linguistic values learned (With Regularization)

	Antecedent	Consequent					
		A_1	A_2	B_1	B_2	C	D
NN_1	If F_1 is Small Then	.93	.03	.00	.01	.00	.02
	If F_1 is Medium Then	.20	.87	.67	.04	.01	.02
	If F_1 is Big Then	.05	.06	.68	.96	.02	.01
NN_2	If F_2 is Small Then	.95	.94	.01	.95	.00	.00
	If F_2 is Medium Then	.96	.13	.94	.08	.00	.00
	If F_2 is Big Then	.11	.96	.04	.06	.01	.01

Recognition Score (Hard Partitioning)

class A	class B	overall score
92.1 %	76.2 %	84.6 %

Table 6.14: Results of Bengali Vowels of Fig. 2.21

Linguistic values learned (With Regularization)

	Antecedent	Consequent								
		u	o	ə	a	ae	e	i	g	h
NN_1	If F_1 is Small Then	.89	.62	.04	.05	.09	.58	.92	.00	.00
	If F_1 is Medium Then	.04	.92	.58	.06	.01	.69	.10	.00	.01
	If F_1 is Big Then	.00	.06	.92	.95	.99	.06	.02	.00	.00
NN_2	If F_2 is Small Then	.94	.93	.95	.05	.02	.01	.04	.00	.00
	If F_2 is More or Less Medium Then	.91	.94	.97	.84	.02	.00	.04	.00	.01
	If F_2 is Medium Then	.12	.00	.01	.00	.97	.90	.05	.00	.00
	If F_2 is Big Then	.13	.06	.00	.00	.05	.66	.91	.00	.00

Recognition Score

Hard Partitioning

class u	class o	class ə	class a	class ae	class e	class i	overall score
97.1 %	62.5 %	36.9 %	1 %	88.8 %	83.9 %	41.1 %	59.4 %

Fuzzy Partitioning

class u	class o	class ə	class a	class ae	class e	class i	overall score
100 %	84.7 %	90.7 %	91.8 %	97.2 %	100 %	96.4 %	94.07 %

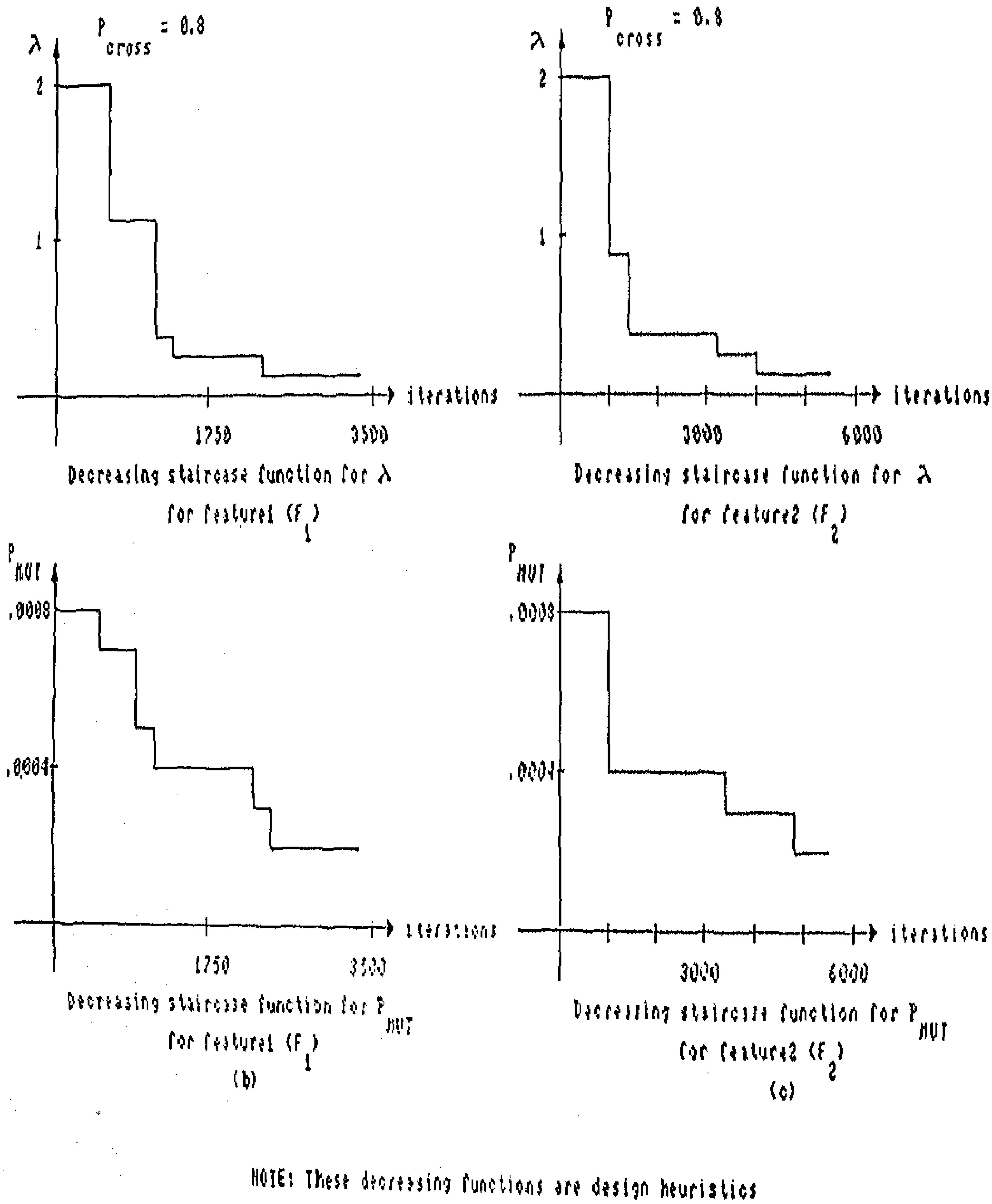
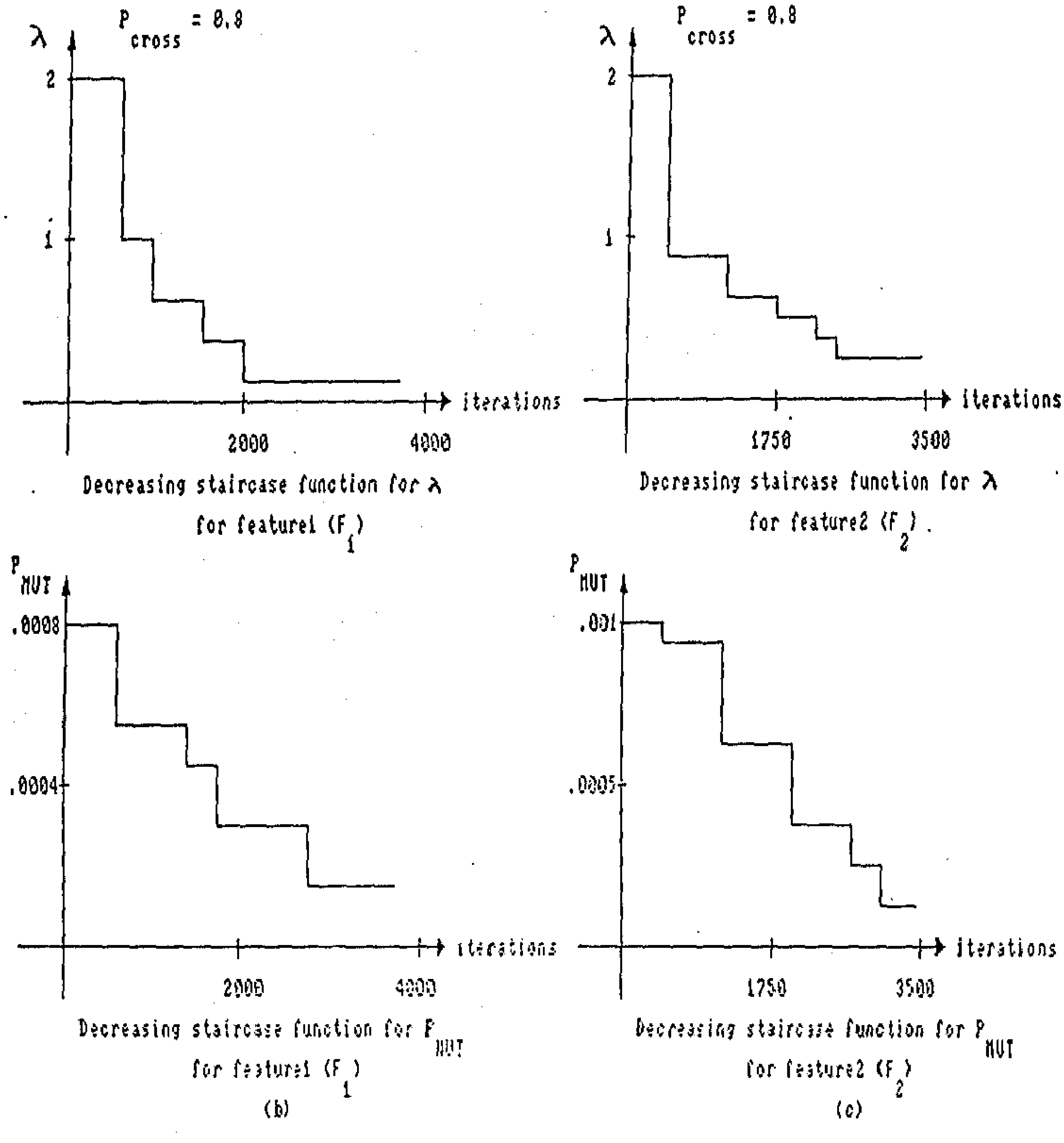
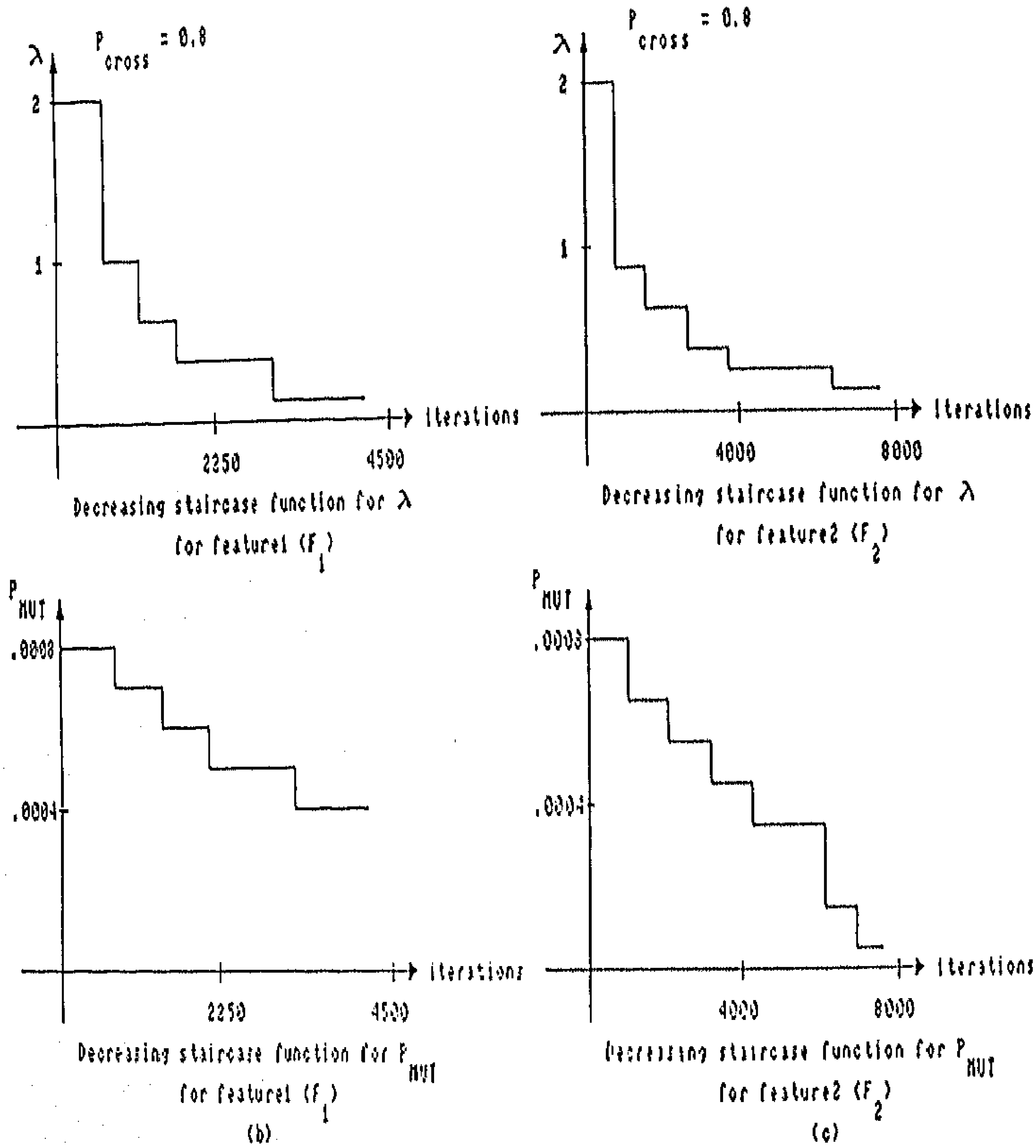


Fig. 6.6: Design heuristics for Figure 2.9



NOTE: These decreasing functions are design heuristics

Fig. 6.7: Design heuristics for Figure 2.10



NOTE: These decreasing functions are design heuristics

Fig. 6.8: Design heuristics for Figure 2.21

6.9 Conclusion

We have developed a successful fusion technology using GA. We test the validity of the fusion technology on object recognition and pattern classification problems and achieve promising results. By increasing the population size (N) and the length of each substring (q) we can achieve much faster convergence than we have reported here. We have already experienced this fact during our experimental studies but not reported here. However depending on the limitation of the resources and need of the problem, we always have to take a judicious decision about the population size (N) and the length of the substring (q) as we have done in the present cases. The concept of regularization has been used for further improvement of MLP performance. Thus, from Chapter 3 to Chapter 6 we have developed fusion of the first kind which produces very effective results for pattern classification and occluded object recognition problems. Note that, fusion of the first kind, which is treated in Chapter 3 is based on interpretation (2.14(a)) of a MFI of Section 2.9 of Chapter 2. Whereas from Chapter 4 to Chapter 6 we used the interpretation (2.16) of a MFI of Section 2.9 of Chapter 2. But the essential difference between the approaches adopted in Chapter 4 and Chapter 5 and approach adopted in Chapter 6 occurs at the time we learn the weights of the neural net. At Chapter 4 and Chapter 5 we use backpropagation algorithm for learning the weights of the neural net. In Chapter 6 the concept of backpropagation is replaced by GA learning algorithm. In the next chapter we try to investigate the concept of fusion of thee second kind as mentioned in Chapter 1 (see Table 1.1).

Chapter 7

Logical Neural network: A fusion of the second kind

In this chapter we further consider a neuro fuzzy technique based on the modified interpretation (Equation (2.16)) of a MFI for pattern classification / occluded object recognition problems. But here we consider fusion (i.e. neuro fuzzy fusion) of second kind as mentioned in Chapter 1 (see Table 1.1). The basic network configuration is same as Figure 4.1 of Chapter 4. But we introduce logical structure in the network computations as shown in Figure 7.3. That means, we essentially design two three layered neural network models each of which consists of 'logic' neurons. The network configuration of Figure 7.3 is valid for pattern classification / occluded object recognition on $\Re^2$. In the present fusion approach (i.e. fusion of the second kind) we may extend the network of Figure 7.3 for classification of patterns / objects on $\Re^n$ as we have already stated in Chapter 4 and 5. The logic neurons employ AND and OR operations of multiple-valued logic, in place of the weighted sum function in the ordinary neuron model. A kind of 'backpropagation' method, which includes some logical operations, is used to train the layered neural network model for pattern recognition. The advantages of the 'logic' neuron models are as follows:

a) satisfactory separability and

b) simple as well as fast operation due to the use of AND and OR operators.

The AND - OR circuits are readily realizable with hardware technology. Hassoun and Nabha [27] reported an efficient hardware implementation capable of doing the AND - OR computations. Pedrycz [71] used *logical operators MAX and MIN for* designing neural networks that are capable of solving one - output problems. This was extended by Pedrycz [73] to handle multiclass problems using a different performance index. The concept of reference neurons was introduced in Pedrycz [74] to design pattern classifiers. The inputs to these networks consisted of all logical combinations of the input variables and the *Lukasiewicz* implication operator was used. Later, Di Nola, Pedrycz and Sessa [7] developed the concepts of single and multi-level relational structures. Hirota and Pedrycz [29] incorporated the use of fuzzy clustering for developing the geometric constructs leading to the design of knowledge based networks. Pedrycz and Rocha [75] used basic aggregation neurons (AND / OR) and referential processing units to design knowledge based networks. Pedrycz, Lam and Rocha [76] formed a distributed modeling that develops individual fuzzy models and determines the essential logical relationships between the local methods. The models themselves are implemented in the form of logic processors being regarded as specialized fuzzy neural networks.

7.1 Fusion of the second kind based on two-valued logic

Watanabe [103] proposed logic neurons the schematic diagram of which is shown in Figure 7.1. Two sets of weight vectors are employed for this model. He proposed a scheme of backpropagation type of neural network for a three - layered network. This network was utilized for a two valued logic. This is an example of the second kind of fusion as mentioned in Table 1.1 of Chapter 1 where the logic is introduced into the neural network structure. The outputs for this type of network is binary in nature. According to the desired response d, the weights are to be adjusted. The output z_is are generated *through forward-propagation* of responses of the consisting neurons to the input pattern vectors. For training of the logic neural network, he developed a kind of 'backpropagation' method which is described in [103]. He simulated a logic neural network which consists of six AND neurons in the hidden layer and two OR neurons in the output layer for the pattern recognition task. The performance results were encouraging.

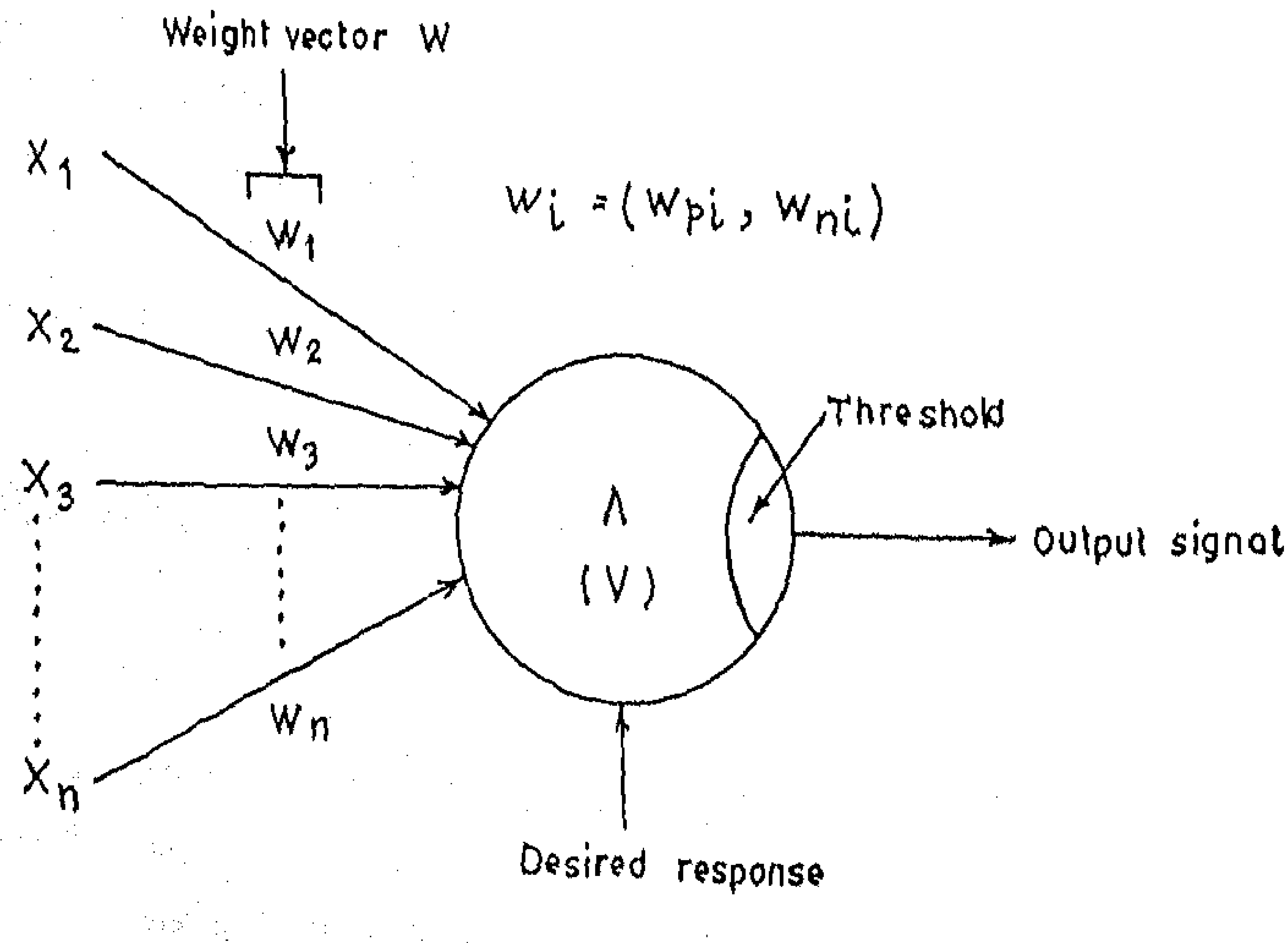


Fig. 7.1: Circuit symbol of AND (OR) neurons

7.2 Fusion of the second kind based on fuzzy logic

In the type of fusion proposed by Pedrycz [73], the author tried to realize the fuzzy relations between input and output by learning the relations through neural network. By doing this, he wanted to draw some analogy between fuzzy relational matrix and learned neural network. In his approach, the neural network is treated as a MAX-MIN network. The output of the network is given by (see Fig. 7.2):

$$Y(y) = f(\Sigma_{i=1}^{n} X(x_i) R(x_i, y) + v(y))$$

where Y(y) is the output, $X(x_i)$ is the i^{th} input node, v(y) is the bias for the y^{th} node. f is a nonlinear function that takes the form in its MAX-MIN counterpart as,

$$Y(y) = max[max_{x_i}(min(X(x_i), R(x_i, y)), v(y)]$$

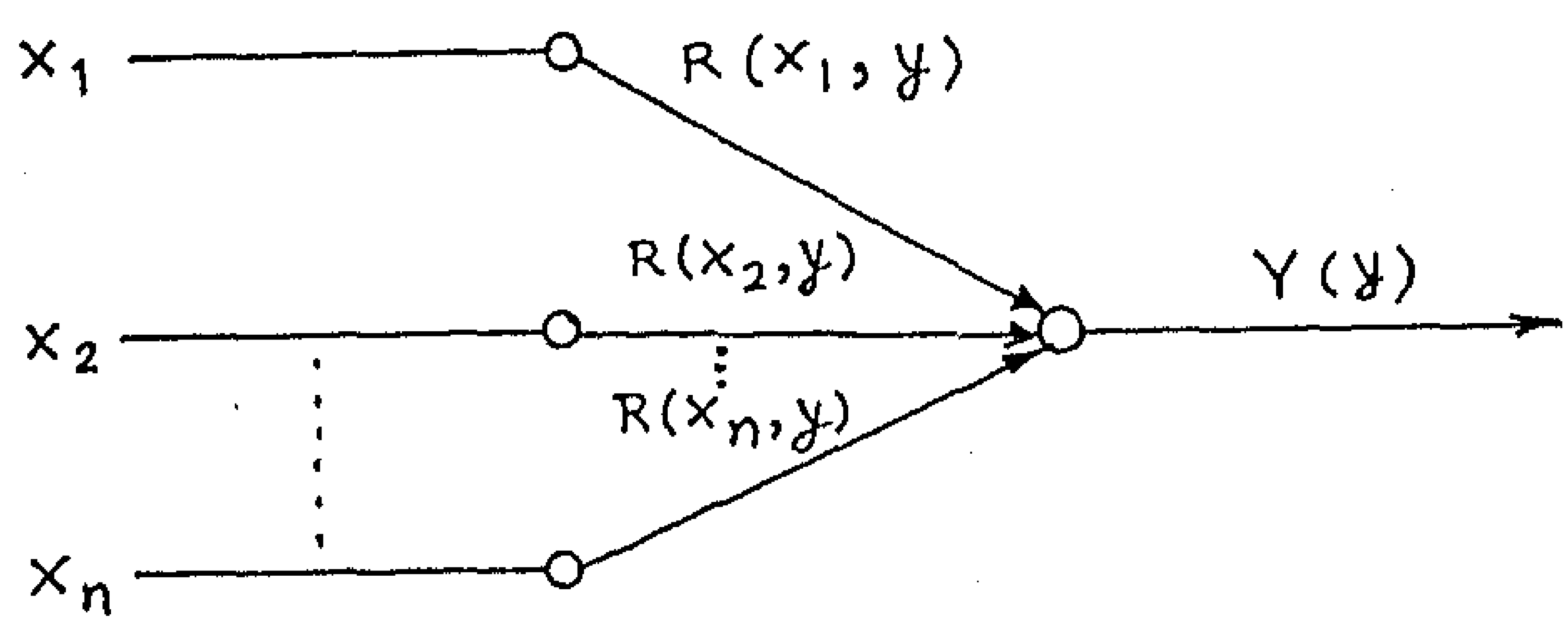


Fig. 7.2: A basic n-input single output node of the network

The objective function is given by the following form,

$$T(y) \equiv Y(y) = \frac{1}{2}[(T(y) \rightarrow Y(y)) \wedge (Y(y) \rightarrow T(y)) + (\bar{T}(y) - \bar{Y}(y)) \wedge (\bar{Y}(y) - \bar{T}(y))]$$

where T(y) denotes the target value at the node Y. For easy hardware implementation point of view, he used Lukasiewicz implication for fuzzy implications. The parameters of the neural net (weights and biases) proposed by Pedrycz [73] updates by the following formula,

$$W(k+1) = W(k) + \psi(k)[\frac{\Delta W(k+1)}{N} + \eta\frac{\Delta W(k)}{N}]$$

where η is the momentum coefficient, k is the number of iteration, $\psi(k)$ is a nonincreasing function of k and N is the number of training patterns.

If the output of the net is denoted by

$$y = (x \wedge a) \vee v$$

then derivative in his method can be calculated by

$$\frac{\partial}{\partial a}[(x \wedge a) \vee v] = \begin{cases} 1, \; if & x \wedge a \geq v, \\ & x \geq a \\ 0 \quad otherwise & \end{cases}$$

The fast convergence rate of the proposed method is another plus point.

7.3 Proposed method

The network structure of our proposed scheme is shown in Figure 7.3. In our approach the inputs to the network of Figure 7.3 are same as inputs to the network of Figure 4.1 of Chapter 4. The basic steps involved in learning and testing are same as stated in Chapter 4. The only difference between the network of Figure 4.1 and the network of Figure 7.3 occurs at the neuron levels of different layers of each individual network. In our network configuration we assume min-max structure (called ANDing) for each neuron of the hidden layer and max-min structure (called ORing) for each neuron of the output layer. The AND and ORs have many definitions like, max-product, max-drastic product etc. Previous researchers have used different types of nomenclature for AND and OR operations as stated here.

The weight parameters from the input layer to hidden layers are denoted by the vector W and the weight parameters from the hidden layer to the output layer are denoted by the vector V. Hence, the output of the hidden nodes are given by, $y'_j = f(x_i, w_{ji}; v_j)$ where the node of input layer is denoted by i, that in hidden layer by j and the node of output layer by k. v_j is the bias for a node of hidden layer. The output y'_j is hence given by,

$$\bigwedge_{i=1}^{N}(x_i \vee w_{ji}) \wedge v_j$$

where N is the number of inputs and

$$net_j = f(x_i, w_{ji}) = \bigwedge_{i=1}^{N}(x_i \vee w_{ji}) \ .$$

The output y_k at the output layer is given by

$$\begin{aligned} y_k &= f(y'_j, V_{kj}; \mu_k) \\ &= \bigvee_{j=1}^{X}(y'_j \wedge V_{kj}) \vee \mu_k \end{aligned}$$

where X is the number of hidden layer nodes and μ_k is the bias for the k^{th} unit at the output layer and

$$net_k = f(y'_j, V_{kj})$$

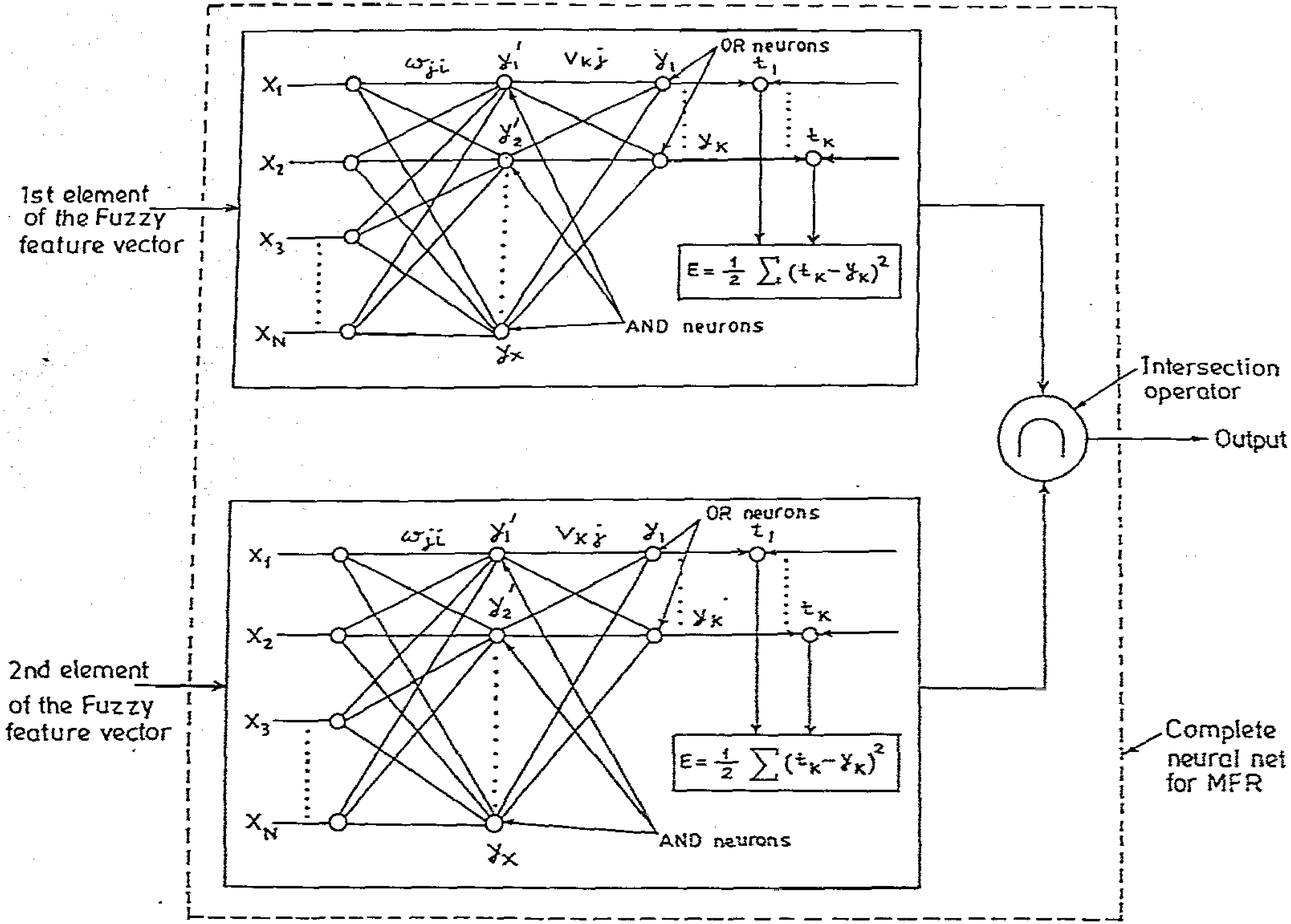


Fig. 7.3 : Architecture of the proposed method.

Then the learning process proceeds according to formulas as follows:

$$\text{i) } w_{ji_{new}} = w_{ji_{old}} - \alpha_1 \frac{\partial E}{\partial w_{ji}}$$
$$\text{ii) } V_{kj_{new}} = V_{kj_{old}} - \alpha_2 \frac{\partial E}{\partial V_{kj}}$$
$$\text{iii) } v_{j_{new}} = v_{j_{old}} - \alpha_3 \frac{\partial E}{\partial v_j}$$
$$\text{iv) } \mu_{k_{new}} = \mu_{k_{old}} - \alpha_4 \frac{\partial E}{\partial \mu_k}$$

where E is the objective function and is equal to $\frac{1}{2}\Sigma(y_k - t_k)^2$ and $\alpha_1, \alpha_2, \alpha_3$ and α_4 are learning coefficients. For our learning scheme, we keep them same and constant. We involve no 'momentum' term in our proposed method. The initialization for the weights as well as biases are random as before .

We calculate the necessary derivatives as follows:

$$\frac{\partial E}{\partial V_{kj}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial V_{kj}}$$
$$\frac{\partial E}{\partial y_k} = (y_k - t_k)$$
$$\frac{\partial y_k}{\partial V_{kj}} = \begin{cases} 1, \ if & \bigvee_{i=1}^{N} (y'_i \wedge V_{ki}) \geq \mu_k \ \& \\ & \bigvee_{\substack{i \neq j \\ i=1}}^{N} (y'_i \wedge V_{ki}) \leq y'_j \wedge V_{kj} \ \& \\ & V_{kj} \leq y'_j \\ 0 \quad otherwise & \end{cases}$$
$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial y'_j} \frac{\partial y'_j}{\partial w_{ji}}$$
$$\frac{\partial y'_j}{\partial w_{ji}} = \begin{cases} 1, \ if & \bigwedge_{\substack{k \neq i \\ k=1}}^{N} (x_k \vee w_{jk}) \leq v_j \ \& \\ & \bigwedge_{k=1}^{N} (x_k \vee w_{jk}) \geq x_i \vee w_{ji} \ \& \\ & w_{ji} \geq x_i \\ 0 \quad otherwise & \end{cases}$$

$$\frac{\partial E}{\partial y'_j} = \bigvee_k \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial y'_j}$$
$$= \bigvee_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial net_k} \frac{\partial net_k}{\partial y'_j}$$

$$\frac{\partial net_k}{\partial y'_j} = \begin{cases} 1, \ if & \bigvee_{\substack{i \neq j \\ i=1}}^{X} (y'_i \wedge V_{ki}) \leq (y'_j \wedge V_{kj}) \\ & y'_j \leq V_{kj} \\ 0 \ \ otherwise & \end{cases}$$

$$\frac{\partial y_k}{\partial net_k} = \begin{cases} 1, \ if & net_k \geq \mu_k \\ 0, \ otherwise & \end{cases}$$

$$\frac{\partial E}{\partial \mu_k} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial \mu_k}$$

$$\frac{\partial y_k}{\partial \mu_k} = \begin{cases} 1, \ if & \bigvee_{j=1}^{X} (y'_j \wedge V_{kj}) \leq \mu_k \\ 0 \ \ otherwise & \end{cases}$$

$$\frac{\partial E}{\partial v_j} = \frac{\partial E}{\partial y'_j} \frac{\partial y'_j}{\partial v_j}$$

$$\frac{\partial y'_j}{\partial v_j} = \begin{cases} 1, \ if & \bigwedge_{i=1}^{N} (x_i \vee w_{ji}) \geq v_j \\ 0 \ \ otherwise & \end{cases}$$

Now all the above mentioned computations are based on the operators max and min. In our experimental study we also have adopted other combinations of operators. For instance the results reported in this chapter are based on max-product operator. Hence we have to make appropriate changes in all the above mentioned computations for max-product operators. Now, in the above computations we have shown that under max-min operators, the derivative values become either 1 or 0. For faster convergence of the net we avoid this 0 - 1 status of the derivatives and fit a smooth sigmoidal function between 0 and 1 as proposed in [32].

7.4 Numerical Examples

To test the effectiveness of the proposed scheme, simulations were run for Figures 2.8,2.9,2.10,2.11,2.19,2.21,2.23. The rules, distribution pattern, number of hidden nodes are as shown in Table 4.3,4.6,4.8,4.9,4.12,4.15,4.18. The recognition scores shown below are quite satisfactory. We used the fuzzy masking of .2 .5 .7 1 .7 .5 .2 as before for fuzzification of the features.

Table 7.1(a): Recognition Scores (in %) (Hard Partitioning) for Figure 2.8

A	B	C	D	E	F	Overall
87.1	90.5	41	31.5	86.1	17.4	62.5

Table 7.1(b): Recognition Scores (in %) (Fuzzy Partitioning) for Figure 2.8

A	B	C	D	E	F	Overall
99.13	95.7	73.1	95.5	100	98.3	94.9

The first feature is learned in 16 iterations and the second feature is learned in 15 iterations.

Table 7.2: Recognition Scores (in %) for Figure 2.9

A	B	C	Overall
70.7	46.4	100	71.9

The first feature is learned in 29 iterations and the second feature is learned in 39 iterations.

Table 7.3: Recognition Scores (in %) for Figure 2.10

A	B	Overall
82.5	93.1	87.4

The first feature is learned in 8 iterations and the second feature is learned in 11 iterations.

Table 7.4: Recognition Scores (in %) for Figure 2.11

A	B	Overall
55.2	95.7	81.6

The first feature is learned in 11 iterations and the second feature is learned in 11 iterations.

Table 7.5(a): Recognition Scores (in %) (Hard Partitioning) for Telugu data (Figure 2.19)

a	e	i	o	u	ə	Overall
75.9	58	90.9	35.3	60.7	66.6	63.8

Table 7.5(b): Recognition Scores (in %) (Fuzzy Partitioning) for Telugu data (Figure 2.19)

a	e	i	o	u	ə	Overall
84.3	100	98.4	39.6	91.1	93.9	86.1

The first feature is learned in 44 iterations and the second feature is learned in 10 iterations.

Table 7.6(a): Recognition Scores (in %) (Hard Partitioning) for Bengali data (Figure 2.21)

u	o	ə	a	ae	e	i	Overall
75.4	45.8	89.2	3.7	72.2	100	14.3	59.1

Table 7.6(b): Recognition Scores (in %) (Fuzzy Partitioning) for Bengali data (Figure 2.21)

u	o	ə	a	ae	e	i	Overall
100	98.6	89.2	96.7	94.4	100	96.4	90.0

The first feature is learned in 11 iterations and the second feature is learned in 9 iterations.

Table 7.7(a): Recognition Scores (in %) (Hard Partitioning) for Assamese data (Figure 2.23)

u	ū	o	ə	a	ae	e	i	Overall
100	83.3	26.6	37.5	75	81.8	100	10.5	59.4

Table 7.7(b): Recognition Scores (in %) (Fuzzy Partitioning) for Assamese data (Figure 2.23)

u	ū	o	ə	a	ae	e	i	Overall
100	83.3	100	81.3	91.7	100	100	84.2	92.3

The first feature is learned in 9 iterations and the second feature is learned in 10 iterations.

7.5 Conclusion

We have applied logic into the neural network structure. We have found that the best results (with respect to the recognition score) which have been reported here are obtained with max-product operator. Also the convergence of the net with max-product operator is much faster than the net with max-min operator. We tried several other combinations like min-max, max-bounded product, max-drastic product etc. But, as the convergence of the net is much slower in these cases, it needs further investigations with respect to our network configuration. The same network configuration i.e. Figure 7.3 is applicable for occluded object recognition provided we use a decision table like Table 5.1 of Chapter 5.

Chapter 8

Conclusion and Scope for future work

The present thesis demonstrates some results on the effectiveness of the neuro fuzzy reasoning for pattern classification and two dimensional occluded object recognition. Fuzzy sets have been introduced at the input and output level of the neural network to tackle and represent the impreciseness of our reasoning technique (neuro fuzzy) for pattern classification and two dimensional occluded object recognition problems. We primarily consider two kinds of fusions as mentioned in Chapter 1 (see Table 1.1). Between these two kinds of fusions, the first approach (i.e. replacement of logic by learning) is based on two different interpretations of multi dimensional fuzzy implication (see Equation (2.14(a)) and (2.16) of Chapter 2). The replacement of logic by learning means the replacement of the logical interpretation of the implication operator (see Table 2.1) between the antecedent clauses and consequent clause of a fuzzy **If Then** rule by the learning rule of backpropagation algorithm or genetic algorithm. Having a good exposure in fuzzy reasoning technique (based on the interpretation of Equation (2.14(a))) for pattern classification (see Chapter 2) from Chapter 3 to 6 we switched over to the concept of fusion of the first kind. Note that fusion of the first kind has been realized first by interpretation of Equation (2.14(a)) (see Chapter 3) and then by Equation (2.16) (see Chapter 4, 5 and 6). In Chapter 7 we consider the fusion of the second kind (i.e. introduction of logical structure into neural network). Note that, in the fusion of the second kind the relation between the antecedent clauses and consequent clause is also learnt either by backpropagation scheme or by genetic algorithm. That means also in fusion of the second kind we replace the logical interpretation of the implication operator (see Table 2.1) by learning scheme. But the neurons of the net perform logical operation (AND - OR operation) instead of arithmetic operation (addition - multiplication). The input of each logical neuron is not further mapped through any nonlinear expression like sigmoid function, radial basis function etc. Rather, it is compared with the bias parameter [73] and then produces the output of the logic neuron. In all neural network learning schemes we follow the generalized delta rule for backpropagation neural network except for Chapter 6 where genetic algorithm has been introduced for learning. Our reasoning approach (neuro fuzzy) for pattern classification and occluded object recognition is general. But to test the effectiveness of our algorithm throughout the thesis we have considered synthetic data, vowel classification data and two dimensional occluded scene data. As our reasoning technique for classification / occluded object recognition is general, it can be applied to any other type of classification / occluded object recognition problems. Throughout the thesis we have compared the performance of our approaches with that of some existing methods and have achieved very promising results.

8.1 Contributions of this work

The work reported in this thesis, makes several contributions in the field of fuzzy rule based approach to pattern classification and two dimensional occluded object recognition.

Chapter 2 sheds some light on the application of the method of approximate reasoning to pattern classification problems. We study different interpretations of fuzzy implications to pattern classification problems. As the primary fuzzy reasoning techniques proposed by Zadeh, Mizumoto and Zimmermann [108,61,62]are independent of any kind of applications and developed much earlier than the development of reasoning based on MFI [91,100] we initially started with the concept of Zadeh, Mizumoto and Zimmermann for pattern classification problems and then demonstrate its (approach proposed by Zadeh, Mizumoto and Zimmermann) relation with a particular interpretation of a MFI [91,100]. In the context of MFI we have introduced a new definition of fuzzy feature vector / pattern vector which effectively explains how fuzzy **If Then** rules can locate patterns in the pattern space. At the end of Chapter 2, we have developed a new model for fuzzy reasoning (based on conventional interpretation of a MFI) for classification of patterns under missing components.

Chapter 3 deals with a fusion of the first kind based on the interpretation of Equation (2.14(a)) of a MFI. The only difficulty we have realized is that when the number of features of a pattern (object) is more than two then the dimension (breadth) of the neural network configuration becomes very large which introduces complexity in learning the weights of the neural network. Moreover when the number of features is more than two then representation of fuzzy feature vector / pattern vector on the pattern space becomes difficult causing further difficulty to have an estimate of the fuzzy set C which is the consequent part of a MFI and which is estimated by looking at the relative position of the fuzzy feature vector / pattern vector and the well defined cover of the pattern space. Hence we switch over to the interpretation of Equation (2.16) of a MFI for our reasoning process for pattern classification / occluded object recognition problems.

Chapter 4 deals with the new interpretation i.e. Equation (2.16) of a MFI and successfully implements multi dimensional fuzzy reasoning on neural networks for pattern classification problems. In Chapter 4 we have discussed an ad-hoc approach for management of uncertainty for vowel classification which is experimentally verified.

In Chapter 5 we straightaway apply the methods developed in Chapter 4 with an additional decision table (Table 5.1 of Chapter 5) to accommodate two dimensional occluded object recognition problem. From the design study of Chapter 4 and 5 we understand that the MFR approach which is implemented on neural network is general for tackling pattern classification and two dimensional occluded object recognition problems.

In Chapter 6 we have highlighted some existing difficulties of backpropagation learning scheme and adopted genetic algorithm for learning keeping all other features of the neural network, which are discussed in Chapter 4 and 5, same. In Chapter 6, we further introduce the concept of regularization to generalize the performance of the neural network.

The treatment of Chapter 4 to 6 does not suffer from the curse of large dimensions of features as it happen in case of Chapter 3. That means, if we try to classify patterns / objects on $\Re^2$ then either we may follow the method adopted in Chapter 3 or the methods adopted in Chapter 4, 5 or 6. But when we consider classification problems on $\Re^n$, n > 2 it is advisable to adopt the methods proposed in Chapter 4 to 6.

Overcoming the problem of large dimensions of features, in Chapter 7 we consider fusion of the second kind.

The success and failure stories of our pattern classification / occluded object recognition problems are thoroughly discussed in each chapter.

8.2 Limitations and Scope for future work

In Section 2.10 of *Chapter 2 we have* proposed a new model for fuzzy reasoning based on Equation (2.14(a)) of Chapter 2. So far the inferencing technique is concerned, the model works very satisfactorily. But the problem which we attempt (pattern classification under missing component) deserves more attention than it has been given in the present thesis simply because we need to develop a complete nonmonotonic reasoning for pattern classification based on default logic [88]. The aspect of *nonmonotonicity* is needed to revise the default *belief* / *tentative* belief which was initially considered for classification. Due to the failure of sensors etc. some of the components of the fuzzy feature vector / pattern vector may be missing at the time of classification. But this missing component may be recovered (due to the repair of the sensors) at any subsequent stage of classification when revision of our earlier tentative belief / default belief based on which the earlier classification was done may be needed. Thus a clear open area is left for future work.

The fusion methodology of Chapter 3 is equally applicable for object recognition problem provided we incorporate one decision table (see Table 5.1 of *Chapter 5*) *at an* appropriate stage of our recognition scheme. To learn the weights of the neural network proposed in Chapter 3 if we see that the usual drawbacks of the backpropagation learning (see the discussion of Section 6.3.2 of Chapter 6) hampers the training of the net we may adopt the genetic algorithm as we have done in Chapter 6.

In Chapter 4, for management of uncertainties in pattern classification, we have experimentally designed a triangular fuzzy masking. But such masking should be designed based on the type of data of Appendix B and using fuzzy statistics. This is another open area which one should consider for management of uncertainties in pattern classification as discussed in Section 4.7 of Chapter 4.

In Chapter 5, for recognition of two dimensional occluded objects, we did not consider the aspect of positioning the object on the scene. Some additional research attempt is needed to incorporate the positioning scheme in our present algorithm [85,86,87]. We simply recognize the occluded object using a voting scheme [55,56,57,58] instead of hypothesis generation and verification paradigm [85,86,87]. Also note that the algorithm stated in Chapter 4 is meant for recognition of two dimensional occluded objects. But its treatment is so generalized that we may extend it for recognition of three dimensional occluded objects [86,87].

In Chapter 6 we may further use genetic algorithm to generate rules from the learned neural network and verify the original set of rules. At the time of learning the set of weights of neural network by genetic algorithm we used SUN 3/60 as the simulator which limits the number of strings in a population. The convergence may be faster if we could have allowed populations with larger number of strings in it. Moreover, genetic algorithm can be utilized to generate rules from the learnt neural net and verify the original set of rules.

It is claimed by some researchers that, the efficiency of training of neural net can be improved by hybrid learning approach. That means initially we should use genetic algorithms for learning to achieve the optimal region fast and then apply the gradient search to find the optimal solution. This sort of hybrid approach can avoid local minima. An effort can be made to test such approach for our models of Chapter 6.

In Chapter 7 we use an objective function which is conventionally used for backpropagation algorithm and which has been used in Chapter 2 to 7 in our thesis. Instead of using this objective function we may consider the kind of objective function mentioned in [73] or some variations of the objective function of [73] using Table 1 of [72]. The objective function originally used by Pedrycz [73] has to be maximized to learn the weights of the neural network. Whereas we always minimize an error function to learn the weights of our neural network. When we use genetic algorithm which may also be used in Chapter 7, the said objective function has to be converted to a form which has to be maximized as per the requirements of genetic algorithm [16]. This conversion from minimization problem to a maximization problem can be avoided if we use the objective function proposed by Pedrycz [73] and can successfully use the genetic algorithm for learning the weights of the neural network.

Bibliography

[1] Attneave F.(1954). Some informational aspects of visual perception, *Psychol. Rev.*, 61, pp. 183 - 193.

[2] Batchelor B. G. and D. J. Hand(1975). Classification of incomplete pattern vectors using orthogonal function methods, *Proc. 3rd Int. Congress on Cybernetics and System Research*, Bucharest, August, pp. 425 - 435.

[3] Belew R. K., J. McInerney and N. N. Schraudolph(1990). Evolving networks: Using the genetic algorithm with connectionist learning. *Technical report CS 90-174*, University of California, San Diego.

[4] Besl P. and R. C. Jain(1988). Segmentation through variable-order surface fitting,*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2), pp. 167-192.

[5] Chalmers D. J.(1990). The evolution of learning: an experiment in genetic connectionism. In D.S. Touretzky, J.L. Elman, T.J. Sejnowski and G.E. Hinton, editors,*Proceedings of the 1990 Connectionist models Summer School*, San Mateo, California, Morgan Kaufmann.

[6] Chen D. S. and R. C. Jain(1994). A robust backpropagation learning algorithm for function approximation, *IEEE Transactions on Neural Networks*, 5(3), pp. 467-479.

[7] Di Nola A., W. Pedrycz and S. Sessa(1995). Fuzzy relational structures: the state of art,*Fuzzy Sets and Systems*, 75, pp. 241 - 262.

[8] Driankov D., H. Hellendoorn and M. Reinfrank(1993). *Introduction to Fuzzy Control*, Springer Verlag, Berlin.

[9] Dutta Majumder D., A. K. Dutta & N. R. Ganguli(1978). Acoustic features of Telugu vowels, *Acoustica*, Vol. 41, pp. 55 - 64.

[10] Fant G.(1960). *Acoustic theory of speech production.* Mouton & Co.,'S-Gravenhage.

[11] Franzini, M. A.(1987). Speech recognition with back propagation. *Proceedings of 9^{th} Annual Conference of the Engineering in Medicine and Biology Society*, IEEE, New York, pp. 1702 - 1703.

[12] Freeman H.(1961). On the encoding of arbitrary geometric configurations, *IEEE Trans. Elec. Comput.*, 10, pp. 260-268.

[13] Freeman H.(1978). Shape description via the use of critical points, *Pattern Recognition*, 10, pp. 159-166. .

[14] Furuya T., A. Kokubu & T. Sakamoto(1988). NFS: Neuro fuzzy inference system, *Trans. IPSJ*, Vol. 30, No. 6, pp. 795 - 798.

[15] Geyer Schulz Andreas(1995). *Fuzzy rule based expert systems and genetic machine learning*, Physica Verlag, Germany.

[16] Goldberg D. E.(1989). *Genetic Algorithms: Search, Optimization and Machine Learning*, (Addison - Wesley, Reading MA).

[17] Gruau F.(1992). Genetic synthesis of boolean neural networks with a cell rewriting developmental process. *Technical Report*, Laboratoire de l'Informatique du Parallélisme.

[18] Gupta M. M. and G. K. Knopf(1990). Fuzzy neural network approach to control systems, *Proc. of first Int. Symp. on uncertainty modeling and analysis*, Maryland, Dec. 3-5, pp. 483 - 488.

[19] Gupta M. M. and G. K. Knopf(1993). Dynamic neural network for fuzzy inference, *SPIE's Conf. on Applications of Fuzzy Logic technology*, Boston, Sept., pp. 7 - 10.

[20] Gupta M. M. and D. H. Rao(1994). On the principles of fuzzy neural networks, *Fuzzy Sets and Systems*, 61, pp. 1 - 18.

[21] Gupta M. M. and J. Qi(1991). On fuzzy neuron models, Int. Joint Conf. on Neural Networks, *IJCNN*, Seattle, July, pp. 431 - 456.

[22] Gupta M. M.(1991). Uncertainty and Information: The emerging paradigms, *Int. J. of Neuro and Mass-Parallel Computing and Information Systems*, 2, pp. 65 - 70.

[23] Gupta M. M.(1992). Fuzzy neural computing systems, 2^{nd} *Int. Conf. on Fuzzy Logic and Neural Networks*, Japan, July, pp. 17 - 22.

[24] Harp S. A., T. Samad and A. Guha(1989). Towards the genetic synthesis of neural networks, *Proc. ICGA-89*, Morgan Kaufmann, pp. 360 - 369.

[25] Gupta L. and A. M. Upadhye(1991). Nonlinear alignment of neural net outputs for partial shape classification, *Pattern Recognition*, 24, pp. 943 - 948.

[26] Hand D. J.(1977). Classification of incomplete vectors, *Ph. D. Thesis*, University of Southampton.

[27] Hassoun M. H. and A. M. Nabha(1993). Implementation of O(n) Complexity Max/Min circuits for fuzzy and connectionist computing, *IEEE - 93 International Conference on Neural Networks*, San Francisco, CA, USA, March 1993, pp. 998 - 1003.

[28] Hayashi I., H. Nomura, H. Yamasaki and N. Wakami(1992). Construction of fuzzy inference rules by NDF and NDFL, *International Journal of approximate reasoning*, Vol.6, pp. 241 - 266.

[29] Hirota K. and W. Pedrycz(1993).*Proceedings of 1993 International Joint Conference on Neural Networks*, pp. 1867 - 1870.

[30] Hu M. K.(1962). Visual pattern recognition by moment invariants, *IRE Trans. Inform. Theory*, Vol. 17-18, pp. 179 - 187.

[31] Hudson D.L. & M. E. Cohen(1991). Combination of rule based and connectionist expert systems, *International J. of Microcomputer applications*, Vol. 10, pp. 36 - 41.

[32] Ikoma N., W. Pedrycz & K. Hirota(1993). Estimation of fuzzy relational matrix by using probabilistic descent method, *Fuzzy Sets and Systems*, Vol. 57, pp. 335 - 349.

[33] Jain R.(1976). Decision making in the presence of fuzzy variables, *IEEE Transactions on Systems, Man & Cybernetics*, 6, pp. 698 - 703.

[34] Jain R.(1977). Tolerance analysis using fuzzy sets, *Int. Jour. System Sci.*, Vol. 7, No. 12, pp. 1393 - 1401.

[35] Jain R.(1977). A procedure for multiple aspect decision making using fuzzy sets, *Int. Jour. System Sci.*, Vol. 8, pp. 1 - 7.

[36] Jain R. and W. Stallings(1978). Comments on fuzzy-set theory versus Bayesian statistics. *IEEE Transactions on Systems, Man & Cybernetics*, 8, pp. 332 - 333.

[37] Kandel A.(1975). Fuzzy hierarchical classifications of dynamic patterns, *NATO ASI Pattern recognition and Classification*, France, Sept.

[38] Kandel A.(1975). Properties of fuzzy matrices and their applications to hierarchical structures, 9^{th} *Asilomar Conf. Circuits, Syst. and Comput.*, Pacific Grove, CA, Nov.

[39] Kandel A.(1978). Fuzzy statistics and forecast evaluation, *IEEE Transactions on Systems, Man & Cybernetics*, 8(5), pp. 396 - 401.

[40] Kandel A.(1980). On the modelling of uncertain systems, *Proc. Int. Congr. on Appl. Syst. Res. and Cybern.*, Acapulco, Mexico, Dec.

[41] Kandel A.(1982). *Fuzzy techniques in pattern recognition*, John Wiley and Sons.

[42] Kandel A.(1986). *Fuzzy mathematical techniques with applications*, Addison Wesley, (Reading MA).

[43] Kandel A. and G. Langholz(1992). *Hybrid architectures for intelligent systems*, CRC Press.

[44] Keller J. & D. J. Hunt(1985). Incorporating fuzzy membership functions into the perceptron algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 7, pp. 693 - 699.

[45] Kim Y. S. and S. Mitra(1994). An adaptive integrated fuzzy clustering model for pattern recognition, *Fuzzy Sets and Systems*, 65, pp. 297 - 310.

[46] Kiszka J. B. & M. M. Gupta(1990). Fuzzy logic neural network, *BUSEFAL*, 4, pp. 104 - 109.

[47] Kitano H.(1990). Designing neural networks using genetic algorithms with graph generation system, *Computer Systems*, 4, pp. 461 - 476.

[48] Klir G. J. & T. Folger(1989). *Fuzzy Sets, Uncertainty and Information.* Reading MA: Addison Wesley.

[49] Klir G. J. & B. Yuan(1995). *Fuzzy Sets and Fuzzy Logic: theory and applications*. Prentice Hall.

[50] Koza J. R.(1990). A paradigm for genetically breeding populations of computer programs to solve problems. *Technical Report, STAN-CS-90-1314*, Dept. of Computer Science, Stanford University, CA.

[51] Ladefoged P.(1967). *Three axes of experimental phonetics*, Oxford University Press, Oxford.

[52] Lindblom B.(1968). On the production and recognition of vowels. *Thesis*, Stockholm University.

[53] Lootsma F. A.(1991). Scale sensitivity and rank preservation in a multiplicative variant of the analytic hierarchy process, *Rep. 91-20*, Faculty of Technical Mathematics and Informatics, University of Delft, The Netherlands.

[54] Mamdani E. H.(1977). Application of fuzzy logic to approximate reasoning using linguistic systems,*IEEE Trans. Comput.*, 26, pp. 182 - 191.

[55] Martin W. N. and J. K. Aggarwal(1978). Dynamic scene analysis: a survey, *Comput. Graph. Image Process.*, 7(3), pp. 356 - 374.

[56] Martin W. N. and J. K. Aggarwal(1979). Computer analysis of dynamic scenes containing curvilinear figures, *Pattern Recognition*, 11, pp. 169 - 178.

[57] McKee J. and J. K. Aggarwal(1975). Finding the edges of the surfaces of three dimensional curved objects by computer, *Pattern Recognition*, 7, pp. 25 - 52.

[58] McKee J. and J. K. Aggarwal(1977). Computer recognition of partial views of curved objects, *IEEE Trans. Comput.*, C-26, 8, pp. 790 - 800.

[59] Miller G. F., P. M. Todd and S. U. Hegde(1989). Designing neural networks using genetic algorithms, *Proc. International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 379 - 384.

[60] Mitra S., Y. S. Kim and S. Pemmaraju, Adaptive pattern recognition by self-organizing neural networks, in: S. Mitra, M.M. Gupta and W. Kraske Eds.) *Neural and Fuzzy Systems*, Ch 8 (SPIE Inst. series).

[61] Mizumoto M.(1985). Extended fuzzy reasoning, *Approximate reasoning in Expert Systems*, Gupta, Kandel, Bandler and Kiszka, Eds, North-Holland, Amsterdam, pp. 71 - 85.

[62] Mizumoto M. and H. J. Zimmermann(1982). Comparison of Fuzzy Reasoning methods, *Fuzzy Sets and Systems*, 8(3), pp. 253 - 283.

[63] Morita A., Y. Imai and M. Takegaki(1988). A method to refine fuzzy knowledge model of neural network type, *SICE '88 27^{th} meeting JS33-3*, pp. 347 - 348.

[64] Mühlenbein H. and J. Kindermann, The dynamics of evolution and learning - towards genetic neural networks, In: R. Pfeifer et. al. (eds.), *Connectionism in Perspective*, North-Holland, pp. 173 - 197.

[65] Nafarieh, A.(1988). A new approach to inference in approximate reasoning and its applications to computer vision, *Ph. D. Dissertation*, University of Missouri-Columbia.

[66] Nakanishi S. and T. Takagi(1990). Pattern recognition by neural networks and fuzzy inference, in *Proceedings of the 1990 International Conference on Fuzzy Logic and Neural Networks, IIZUKA (Japan)*, pp. 183 - 186.

[67] Newton S. C. and S. Mitra(1991). Self-organizing leader clustering in a neural network using a fuzzy learning rule, *SPIE Proc. 1565*, pp. 331 - 337.

[68] Nomura H., I. Hayashi and N. Wakami(1989), Self tuninng method of fuzzy reasoning by Hopfield neural network, 5^{th} *Fuzzy Syst. Symp.*, pp. 177 - 182.

[69] Pao Y. H.(1989). *Adaptive Pattern Recognition and Neural Networks.* Addison - Wesley Publishing Company.

[70] Pedrycz W.(1989). *Fuzzy Control and Fuzzy Systems*, (Research Studies Press Ltd.), England.

[71] Pedrycz W.(1990). Relational structures in fuzzy sets and neurocomputations, *Proceedings of the 1990 International Conference on Fuzzy Logic and Neural Networks, Iizuka*, Japan, pp. 235 - 238.

[72] Pedrycz W.(1990). Direct and Inverse Problem in Comparison of Fuzzy Data, *Fuzzy Sets and Systems*, Vol. 34, pp. 223 - 235.

[73] Pedrycz W.(1991). Neurocomputations in relational systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13, pp. 289 - 297.

[74] Pedrycz W.(1992). Fuzzy neural networks with reference neurons as pattern classifiers, *IEEE Transactions on Neural Networks*, 3, pp. 770 - 775.

[75] Pedrycz W. and A. F. Rocha(1993). Fuzzy set based models of neurons and knowledge-based networks, *IEEE Transactions on Fuzzy Systems*, 1, pp. 254 - 266.

[76] Pedrycz W., P. C. F. Lam and A. F. Rocha(1995). Distributed fuzzy system modeling, *IEEE Transactions on Systems, Man and Cybernetics*, 25(5), pp. 769 - 780.

[77] Persoon E. and K. S. Fu(1977). Shape discrimination using Fourier descriptor, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 7, No. 3, pp. 179.

[78] Price K. E.(1984). Matching closed contours, in *Proc. Seventh Int. Conf. Pattern Recognition*, Montreal, P.Q., Canada, July 30 - Aug. 2, pp. 990 - 992.

[79] Ray Kumar S. and J. Ghoshal(1995). Neuro linguistic approach to pattern recognition, *Fuzzy Sets and Systems*, 74, pp. 277 - 305.

[80] Ray Kumar S. and J. Ghoshal(1996). Approximate reasoning approach to pattern recognition, *Fuzzy Sets and Systems*, 77, pp. 125 - 150.

[81] Ray Kumar S. and J. Ghoshal(1996). Neuro genetic approach to pattern recognition, *IIZUKA Proceedings*, Vol. 1, pp. 221 - 224.

[82] Ray Kumar S. and J. Ghoshal(1997). Neuro fuzzy approach to pattern recognition, *Neural Networks*, Vol. 10, No 1, pp. 161 - 182.

[83] Ray Kumar S. and J. Ghoshal(1997). Neuro fuzzy reasoning for occluded object recognition, *Fuzzy Sets and Systems* (to appear).

[84] Ray Kumar S. and J. Ghoshal(1997). A neuro genetic approach for multi dimensional fuzzy reasoning for pattern classification, *Fuzzy Sets and Systems*, (Revised Version communicated).

[85] Ray Kumar S. and D. Dutta Mazumder(1989). Application of differential geometry to recognize and locate partially occluded objects, *Pattern Recognition Letters*, 9, pp. 351 - 360.

[86] Ray Kumar S. and D. Dutta Mazumder(1991). Recognition and positioning of partially occluded 3-D objects, *Pattern Recognition Letters*, 12, pp. 93 - 108.

[87] Ray Kumar S. and D. Dutta Mazumder(1994). Application of Hopfield neural networks and Canonical perspectives to recognize and locate partially occluded 3-D objects, *Pattern Recognition Letters*, 15(8), pp. 815 - 824.

[88] Rich E.(1983). *Artificial Intelligence*, McGrawHill International.

[89] Rosenfeld A. and E. Johnston(1973). Angle Detection on digital curves, *IEEE Trans. Comput*, C-22(9), pp. 875 - 878.

[90] Sugeno, M. and G. T. Kang(1988). Structure identification of fuzzy model, *Fuzzy Sets and Systems*, 28(1), pp. 15 - 33.

[91] Sugeno M. and T. Takagi(1983). Multi-dimensional fuzzy reasoning, *Fuzzy Sets and Systems*, 9, pp. 313 - 325.

[92] Takagi H., N. Suzuki, T. Koda and Y. Kojima(1992). Neural networks designed on approximate reasoning architecture and their applications, *IEEE Transactions on Neural Networks*, Vol. 3, No. 5, pp. 752 - 760.

[93] Takagi H. and I. Hayashi(1991). NN-driven fuzzy reasoning, *International Journal of approximate reasoning*, Vol. 5, pp. 191 - 212.

[94] Tamir D. E., D. G. Schwartz and A. Kandel(1991). A pattern recognition interpretation of implications. *Information Sc.*, Vol. 57 - 58, pp. 197 - 215.

[95] Tamura R., R. Fujioka, H. Isibuchi & H. Tanaka(1990), Identification of interval valued membership function using neural networks, *3^{rd} Sign. Syst. Contr. Symp. of ISCIE*, pp. 97 - 100.

[96] Tikhonov A. N.(1977). *Solutions of Ill-Posed Problems*, (Washington DC: V.H. Winston & Sons).

[97] Tou J. T. and R. C. Gonzalez(1974). *Pattern Recognition Principles*, Addison Wesley.

[98] Triantaphyllou E.(1993). A quadratic programming approach in estimating similarity relations, *IEEE Transactions on Fuzzy Systems*, 1(2), pp. 138 - 145.

[99] Tsang P. W. M., K. H. Wong and K. Y. Lo(1988). A feature extraction technique using dynamic programming, it Electronic Imaging, 88, *Internal Electronic Imaging Exposition and Conference*, pp. 194 - 198.

[100] Tsukamoto, Y.(1979). An approach to fuzzy reasoning method. In M.M.Gupta, R.K.Ragade & R.R.Yager (Eds.), *Advances in Fuzzy Set Theory and Applications*, North-Holland, Amsterdam, pp. 137 - 149.

[101] Urbakh V. Yu.(1976). Discriminant diagnostic with incomplete symptomatics, *Kybernetics*, Vol. 5, pp. 113 - 118.

[102] Watanabe N., A. Kawamura, R. Masuoka, Y. Ohwada and K. Asakawa (1990). Fuzzy control using neural network, *40^{th} of meeting IPSJ*, 3C - 10, pp. 148 - 149.

[103] Watanabe T., M. Matsumoto and T. Hasegawa(1990). A layered neural network model using logic neurons, *Proceedings of the First International Conference on fuzzy logic and neural networks, Iizuka*, Japan, pp. 675 - 678.

[104] Whitley D. and T. Hanson(1989). Optimizing neural networks using faster, more accurate genetic search. In: David Schaffer, editor, *Proceedings of the Third International Conference on Genetic algorithms*, San Mateo, California, pp. 391 - 396.

[105] Yamaguchi T., M. Tanabe and J. Murakami(1989). Fuzzy control using LVQ unsupervised learning, *SICE Joint Symp. of 15^{th} Syst. Symp. and 10^{th} Knowl. Engn. Symp.*, pp. 179 - 184.

[106] Yamaguchi T., N. Imasaki and K. Haruki(1990). Fuzzy rule realization on associative memory system, *IJCNN '90*, pp. 720 - 723.

[107] Yamakawa T. and S. Tomoda(1989). A fuzzy neuron and its application to pattern recognition, *Proc. of the third IFSA Congress*, Seattle, Aug., pp. 30 - 38.

[108] Zadeh L. A.(1970). Theory of approximate reasoning, in: J.E. Hayes, Donald Michie and L.I. Mikulich, Ed., *Machine Intelligence*, (Ellis Horwood), pp. 149 - 194.

[109] Zadeh L. A.(1973). Outline of a new approach to the analysis of complex systems and decision processes, *IEEE SMC*, 1, pp. 28 - 44.

[110] Zadeh L.A.(1996). Fuzzy Logic = Computing with words, *IEEE Transactions on Fuzzy Systems*, 4(2), pp. 103 - 111.

[111] Zahn C. and R. Roskies(1972). Fourier descriptors for plane closed curves, *IEEE Trans. Comput.*, Vol. 21, pp. 269 - 281.

[112] Zimmermann H. J.(1976). Description and optimization of Fuzzy Systems, *International Journal of General Systems*, 2, pp. 209 - 215.

[113] Zimmermann H. J.(1980). Testability and measuring of mathematical models in Social sciences, *Mathematical Modeling*, 1, pp. 123 - 139.

[114] Zimmermann H. J. and P. Zysno(1983). Decisions and Evaluations by Hierarchical Aggregation of Information, *Fuzzy Sets and Systems*, 10, pp. 243 - 266.

[115] Zimmermann H. J.(1987). Fuzzy Sets in Pattern Recognition, in: P.A. Devijer and J. Kittler, Ed., *Pattern Recognition Theory and Applications*, (Springer-Verlag, Berlin), pp. 383 - 391.

[116] Zimmermann H. J.(1987). *Fuzzy Sets, Decision Making and Expert Systems*, (Kluwer, Boston).

[117] Zimmermann H. J.(1991). *Fuzzy Set Theory - And its applications*, 2^{nd} Rev., Ed. (Kluwer, Boston).

[118] Zimmermann H. J. and P. Zysno(1980). Latent connectives in human decision making, *Fuzzy Sets and Systems*, 4, pp. 37 - 51.

[119] Zimmermann H. J.(1993). *First European Congress on Fuzzy and Intelligent Technilogies - EUFIT '93*, Vol. 1, Aachen, September 1993. Verlag der Augustinus Buchhandlung.

[120] Zimmermann H. J.(1993). *First European Congress on Fuzzy and Intelligent Technilogies - EUFIT '93*, Vol. 2, Aachen, September 1993. Verlag der Augustinus Buchhandlung.

[121] Zimmermann H. J.(1993). *First European Congress on Fuzzy and Intelligent Technilogies - EUFIT '93*, Vol. 3, Aachen, September 1993. Verlag der Augustinus Buchhandlung.

Appendix A

To demonstrate the capacity of our algorithm for management of uncertainty in pattern classification (see Section 4.7 of Chapter 4) we have taken the vowel data from Bengali language for three speakers. The shape and size of the clusters for different vowels uttered by different speakers vary due to the difference in pronunciation and change in vocal tract dynamics. We highlight these portions of clusters where the shape and size vary from speaker to speaker (see Figure 4.4, 4.5 and 4.6 of Chapter 4). Our present algorithm along with the fuzzy masking concept is able to identify many of those distorted portions of clusters (see Figure 4.4, 4.5 and 4.6 of Chapter 4) those were not present during the learning of the rule based system. The clusters whose shape and / or position differ widely from one another are cluster a, æ, e and i. For cluster a, the vowels occurring in the zone 700 - 800 Hz. for F_1 and 1100 - 1300 Hz. for F_2 are much distorted in the pattern space of speaker2 and 3 than that for speaker1. For cluster æ, the samples occurring in the zone 550 - 650 Hz. for F_1 and 1500 - 1700 Hz. for F_2 for speaker 2 and 3 are much distorted than that for speaker1. In the pattern space for speaker2, for this vowel, one lone point occurring at 700,2300 is also correctly classified. Clusters e and i, the shapes and locations of which have changed drastically, in speaker3, can be classified correctly by utilizing the spread of fuzzy masking.

Appendix B

The table below represents the informant wise variances of the first two formant frequencies for different vowels. The vowels /ɐ/ and / æ/ show consistent low informant wise coefficient of variation for the first formant. The low height of the tongue for the production of these vowels may be noted in this connection. The variation is also low for the vowel / ∂/. Extraordinarily large variation is observed for vowel /*o*/ for all informants except the second female informant. Apart from this, high vowels /*u*/ and /*i*/ show consistently large variations for the first formant frequency. Male informants exhibit consistently high variations for the vowel /*e*/. These variational data conform well with those for Telugu vowels [9]. As regards the coefficient of variation for second formant frequency, the low value for the front vowel /*i*/ is somewhat consistent. This consistent behavior is also exhibited by the high variation for the back vowel /*u*/. The combined ranking shows that front vowels occupy the low variation end and the back vowels occupy the high variation end. This behavior is not consistent with Telugu data. In fact, Telugu /*e*/ exhibited largest variation while /*o*/ exhibited least variation. However, the fact that the vowels with low tongue position exhibit the lowest F_1 variation and those with retracted tongue position exhibit largest F_2 variation tends to conform the hypothesis that the inaccuracy of positioning the tongue for a vowel is mainly responsible for the variation of formants. Vowel wise variations of the first two formant frequencies are in most cases larger for female informants than for male informants.

This is in conformity with the Swedish vowels [10].

Table B.1: Informant wise variation of the first two formant frequencies for different vowels of Bengali language

Phon Symbol		Male				Female				Male / Female Pooled
	Formant	1	2	3	1,2,3	1	2	3	1,2,3	
		σ (Hz.)	σ (Hz.)	σ (Hz.)	σ (Hz.)	σ (Hz.)	σ (Hz.)	σ (Hz.)	σ (Hz.)	σ (Hz.)
/ u/	F_1	41.9	32.3	23.2	38.5	47.1	69.0	49.6	48.2	53.0
	F_2	128.8	110.4	111.0	122.9	105.7	97.9	47.1	102.9	117.0
/ o/	F_1	92.0	66.7	62.4	83.0	57.3	49.5	75.6	63.8	85.6
	F_2	88.6	124.3	37.4	110.9	82.6	87.9	49.5	90.1	111.6
/ɔ/	F_1	51.9	44.4	60.2	50.1	54.0	64.3	75.2	74.0	71.1
	F_2	56.6	94.7	83.7	62.8	53.9	114.1	141.4	93.7	115.0
/ɐ/	F_1	39.0	50.0	50.0	45.1	36.3	69.2	82.4	60.5	71.7
	F_2	82.3	123.9	110.2	133.4	117.6	194.7	102.6	164.5	217.9
/ æ/	F_1	40.5	46.3	59.8	50.6	28.9	48.7	38.2	62.4	72.7
	F_2	105.5	103.0	71.5	96.5	256.2	65.8	119.7	203.4	230.5
/ e/	F_1	62.5	39.1	43.7	62.1	37.4	42.6	0	56.5	71.3
	F_2	97.8	99.9	70.4	108.3	143.9	130.0	110.8	152.3	270.4
/ i/	F_1	31.7	30.6	18.6	26.5	33.3	30.7	41.5	44.1	25.7
	F_2	88.1	49.0	86.1	103.5	73.2	86.2	116.6	108.7	336.9

Appendix C

Here, the internal angle vs. curvature of various occluded scenes (depicted in Chapter 5) have been given.

Table C.1 : Internal Angle Vs. Curvature of occlusion1

Scene point	Internal Angle	Curvature
6	110.51	4.34
17	48.71	8.20
23	89.96	7.50
24	89.96	9.00
25	104.93	4.69
26	117.63	3.89
28	52.10	7.99
29	59.01	7.56
30	37.85	8.88
40	75.93	6.50
41	94.82	5.32
42	127.82	8.69
43	82.84	6.07
44	115.94	4.00
45	66.09	7.11

Table C.2: Internal Angle Vs. Curvature of occlusion2

Scene point	Internal Angle	Curvature
0	60.92	7.44
15	69.52	7.89
22	60.92	7.44
25	68.35	7.97
26	85.10	5.93
41	108.39	5.96
45	42.04	8.62
46	52.10	7.99
47	53.10	7.93
49	44.98	9.64
50	126.10	3.36
51	58.21	7.61
52	102.29	4.85
53	99.12	5.05
54	59.01	7.56
55	103.99	7.60
56	53.47	7.90
65	69.41	6.91

Table C.3: Internal Angle Vs. Curvature of occlusion3

Scene point	Internal Angle	Curvature
0	65.31	7.16
16	123.64	3.52
22	56.52	7.71
30	110.18	5.81
32	84.52	5.96
33	121.62	4.16
34	56.50	7.71
36	70.31	10.96
40	66.11	7.11
41	52.10	7.99
42	8.80	39.19

Table C.4: Internal Angle Vs. Curvature of occlusion4

Scene point	Internal Angle	Curvature
10	129.07	3.18
15	116.51	3.96
16	123.64	3.52
17	128.60	5.13
18	54.44	10.46
25	126.81	8.86
31	107.35	5.18
32	74.90	10.50
33	63.40	8.32
34	68.17	6.98
37	44.98	8.43
38	101.40	4.91
40	118.21	3.86
43	55.38	7.78
54	59.99	7.50

Table C.5: Internal Angle Vs. Curvature of occlusion5

Scene point	Internal Angle	Curvature
0	60.92	7.44
6	89.96	5.62
7	135.95	2.75
8	148.97	3.10
9	146.25	5.62
10	143.07	4.61
12	143.07	4.61
14	130.54	6.18
15	74.26	7.55
16	149.59	2.53
20	60.92	7.44
23	68.35	7.97
24	88.81	7.59
25	52.10	10.65
26	146.25	5.62
30	146.25	5.62
35	42.49	8.59
36	146.25	2.81
38	146.25	5.62
39	146.25	5.62
41	134.94	11.26
45	138.83	2.57
46	138.83	2.57
60	108.39	5.96
61	146.82	3.31
62	42.64	8.58
63	52.10	7.99
64	51.46	8.03
66	44.98	9.64
67	59.01	8.64
68	102.48	4.84
69	96.48	5.21
70	62.82	7.32
72	48.86	8.19
73	146.25	3.37
74	134.94	3.75
75	134.94	3.75
76	139.91	4.00
78	138.25	2.60
83	69.41	6.91

Table C.6: Internal Angle Vs. Curvature of occlusion6

Model point	Internal Angle	Curvature
1	143.07	4.61
6	123.64	9.39
8	133.94	2.87
13	109.39	4.41
14	123.64	3.52
15	118.36	3.85
17	134.94	11.26
18	142.88	2.65
19	134.94	3.75
20	134.94	11.26
23	93.49	5.40
25	128.60	5.13
32	109.60	7.03
33	149.42	2.18
34	149.42	2.18
39	112.78	4.20
41	70.63	6.83
42	101.40	4.91
43	79.57	6.27
50	143.07	6.15
51	143.07	6.15
55	63.98	7.25
65	146.25	5.62
68	146.25	5.62
77	147.93	2.00
79	39.07	10.06
80	46.89	9.50
81	33.67	9.14