Speaker independent isolated word recognition

SPEAKER INDEPENDENT ISOLATED WORD RECOGNITION

by

ELIJAH MWANGI

A Doctoral Thesis
Submitted in partial fulfilment of the requirements
for the award of
DOCTOR OF PHILOSOPHY
of the
Loughborough University of Technology

1987

Supervisor: Dr C S Xydeas
Department of Electronic and Electrical Engineering
Loughborough University of Technology

ABSTRACT

The work presented in this thesis concerns the recognition of isolated words using a pattern matching approach. In such a system, an unknown speech utterance, which is to be identified, is transformed into a pattern of characteristic features. These features are then compared with a set of pre-stored reference patterns that were generated from the vocabulary words. The unknown word is identified as that vocabulary word for which the reference pattern gives the best match.

One of the major difficulties in the pattern comparison process is that speech patterns, obtained from the same word, exhibit non-linear temporal fluctuations and thus a high degree of redundancy. The initial part of this thesis considers various dynamic time warping techniques used for normalizing the temporal differences between speech patterns. Redundancy removal methods are also considered, and their effect on the recognition accuracy is assessed.

Although the use of dynamic time warping algorithms provide considerable improvement in the accuracy of isolated word recognition schemes, the performance is ultimately limited by their poor ability to discriminate between acoustically similar words. Methods for enhancing the identification rate among acoustically similar words, by using common pattern features for similar sounding regions, are investigated.

Pattern matching based, speaker independent systems, can only operate with a high recognition rate, by using multiple reference patterns for each of the words included in the vocabulary. These patterns are obtained from the utterances of a group of speakers. The use of multiple reference patterns, not only leads to a large increase in the memory requirements of the recognizer, but also an increase in the computational load. A recognition system is proposed in this

thesis, which overcomes these difficulties by (i) employing vector quantization techniques to reduce the storage of reference patterns, and (ii) eliminating the need for dynamic time warping which reduces the computational complexity of the system.

Finally, a method of identifying the acoustic structure of an utterance in terms of voiced, unvoiced, and silence segments by using fuzzy set theory is proposed. The acoustic structure is then employed to enhance the recognition accuracy of a conventional isolated word recognizer.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor, Dr C S Xydeas, for his guidance, encouragement and assistance, especially in the preparation of this thesis. His knowledge in the field of Digital Processing of Speech Signals was invaluable. Thanks are also due to my sponsors, the Royal Commission for the Exhibition of 1851, and the Committee of Vice Chancellors and Principals of the UK Universities and Colleges for the financial support which made it possible for me to pursue this research programme.

I would also like to thank my colleagues, Dr Koh Soo Ngee, Dr David Allott and Mr N Gouvianakis, of the Speech and Image Processing Laboratory, Loughborough University, with whom I have shared many interesting and useful discussions. Special thanks go to the PDP11 Computer Manager, Mr A F Erwood, for ensuring a reliable and efficient computer service, and Dr P Maheswaran for his useful advice and encouragement.

I am also grateful to the following subjects who took part in the recording sessions at the beginning of the research programme: David Allott, Tony Erwood, Murray Holt, Leslie Beeley, Christine Johnson, Suzanne Whyman and Steven Afterlak.

Finally, I would like to thank Janet Smith for her patience in typing this thesis.

LIST OF PRINCIPAL SYMBOLS AND ABBREVIATIONS

ARPA	Advanced Research Projects Agency
BW	Bandwidth
D(A,B)	Distance between word patterns A and B
$d(a_i,b_j)$	Distance between speech frames a_i and b_j
$D_N(M)$	Average distortion of a training set of N vectors by an M entries VQ codebook
DTW	Dynamic Time Warping
E_p	Minimum prediction error
ERR	Normalized prediction error
FIR	Finite Impulse Response
g(i,j)	Accumulated distance at the grid point (i,j)
kNN	k Nearest Neighbours
LE	Logarithmic Energy
L(G)	Language generated by the Grammar G
LP1	1st Linear Prediction Coefficient
LPC	Linear Predictive Coding
LPF	Low Pass Filter
MAX(.)	Maximum value of function in (.)
MIN(.)	Minimum value of function in (.)
MKM	Modified K Means
R(i)	i^{th} autocorrelation coefficient
SPLIT	Sequence of Phoneme Like Templates
V_N	Non-terminal symbol
VQ	Vector Quantization
VUS	Voiced-Unvoiced-Silence
V_T	Terminal Symbol
ZCR	Zero Crossing Rate
$[\ ]^t$	Transpose matrix operator

CONTENTS

Page No

Page No

CHAPTER 1

INTRODUCTION

The concept of speech communication between man and machines is not new, being found in the folklore of many ancient civilizations. The first major achievement towards such a goal can be attributed to the work of A.G. Bell [1] on the conversion of sound stimulus into electrical signals. However, it is only with the advent of electronics and modern computers that the subject has developed from myth into reality.

Man-machine communication by voice can be subdivided into three regions:

i) voice response systems
ii) speaker recognition systems
iii) speech recognition systems

Voice response systems translate the output of a device into a spoken message, thus providing a speech communication in one direction only i.e. from the device to man. In speaker recognition systems, the speech communication is from man to machine and the task is to verify or identify a speaker from a given list. Speech recognition systems also use voice communication from man to machine. The aim is either to recognize or 'understand' a spoken utterance. The art of 'understanding' means that the device responds correctly to what was spoken.

A combination of the three systems would enable a speaker to hold a 'conversation' with a machine. Such a possibility would not only give rise to a lot of curiosity and academic interest, but also it can be usefully applied in a number of ways which are briefly described in the following sections.

1.1 THE VALUE OF MAN-MACHINE SPEECH COMMUNICATION

In the 20th century man has developed machines and in particular computers in order to satisfy the demand in efficiency of the activities of modern life. There is a call for the need to optimize the link between the machine and the operator. Traditionally, the link has been mainly mechanically oriented, but recent advancement in speech processing has exposed the great potential and the suitability of voice as a link. Speech is man's most natural means of communication.
Speech is also man's highest communication capacity output channel.

Several investigators [2], [3], [4] have examined the relative information carrying capabilities of some of the common techniques, among them: speaking, handwriting, typing, and touch-tone methods. Table 1.1 is a summary of the results obtained. These results strongly suggest that speech, as an information conveying medium, is unsurpassed by the other common modes of communication. Therefore, if computers could recognize human speech, they would exploit the potential offered by the speech communication and also become available for use by a large section of the population. In general, man-machine speech communication would find applications in such fields as:

i) commercial
ii) military
iii) social
iv) scientific

Commercial Applications

Speech technology will probably have its greatest use in commercial applications. Already voice input systems have become operational in quality control, automated material handling and stock control. In most quality control tasks, the operators' eyes and hands are usually busy with inspection. By using a voice data entry system, the operator is able to record his observations and measurements as he is

TABLE 1.1: INFORMATION RATE FOR COMMON COMMUNICATION MODES

COMMUNICATION MODE	INFORMATION RATE (words/sec)
1. Speaking [2]	2.0-3.6
2. Handwriting [2]	0.4
3. Typing (skilled subjects) [2]	1.6-2.5
4. Typing (unskilled subjects) [2]	0.2-0.4
5. Touch-tone [3]	1.2-1.5
6. Typing (skilled subjects in a problem solving situation) [4]	0.6
7. Speaking in a problem solving situation [4]	2.9

doing his normal work. The operator is thus able to provide a fast, timely and accurate inspection report.

Telephone services currently provided by attendants, can also be automated and offered at a low cost by employing speaker and speech recognition systems. Such services are based on the retrieval of information from computerized data banks and include ordering and verifying credit cards, telephone banking, catalogue ordering, travel reservations, stock market quotations, weather forecast information, etc.

Military Applications

Speech recognition has also become attractive to the military especially for the purpose of security, surveillance, command and control.

i) Security:
Security precautions require the identification or the verification of persons before getting access into certain installations. Speaker recognition techniques can be used in conjunction with traditional methods like magnetic cards, badge readers etc, to enhance security.

ii) Surveillance:
Surveillance of enemy communication channels is undoubtedly of major interest to military intelligence. One of the aims is to recognize a keyword or a set of keywords embedded in narrow bandwidth conversation speech as found in a radio link. In the surveillance of lengthy speech conversations there is a need for a quick method of editing and scanning. The automatic recognition of keywords would perform this function.

Channels can also be surveilled with the aim of identifying the language being used. Linguistic chains formed from the phonetic transcription of speech have been shown [5] to be a powerful means of discrimination between different languages.

iii) Command and control:
In the aeroplane cockpit, the pilot is continuously monitoring and manipulating a diverse number of instruments. The use of his voice in issuing commands for the control of some devices can be a most important advantage.

Social Applications

The disabled would probably be the largest social group to enjoy the benefits of speech recognition technology. People who are paralysed from the neck down could use their power of speech to control a wheelchair. Systems which convert the voice output into a visual signal have been proposed as speech training aids for the deaf [6].

Speech recognition can also play a useful role in language learning and translation. Devices that can check spellings for a limited number of words, or translate some phrases from one language to another are already available.

Other applications which would interest the general public are voice-controlled domestic items like TV sets, radios and toys. Voice output messages in fire alarm systems, in motor vehicles, and in personal items like watches, could also prove to be popular.

Speaker recognition or verification can be put to use in criminal investigations. Already, voice identification results have been presented as evidence in some United States courts of law [7].

Scientific Applications

The development of a computer capable of understanding human speech, under any circumstances and conditions, would be a novel scientific invention. The achievement of such a goal would require a wealth of knowledge about the nature of speech and the manner in which human beings understand spoken language. Such a system would have to model

aspects of human intelligence and would thereby serve as a testing ground for the effectiveness of the theories of artificial intelligence.

1.2 HUMAN FACTORS IN MAN-MACHINE COMMUNICATION

There exists a number of human factors that can affect the usefulness of voice as a medium of communication with a computer for data entry and control applications. Failure to attend to these factors would be detrimental to the tasks involved.

i) Recognition accuracy

The recognition accuracy in a practical speech recognition system must be reasonably high so as not to hinder the accomplishment of the intended task and to eliminate any loss of confidence by the user. Even where provision for error correction exists, high error rates tend to frustrate the user and he will not wish to use the system. It has also been reported that [8] recognition accuracy tends to decrease when the user senses that something is wrong with the system, and loses his confidence.

ii) Error correction [9]

Since the speech recognition process is generally error prone, it becomes necessary to provide convenient error correction procedures. One common method is to store the recognition output in a buffer stage and only transmit it to the output device upon reception of a verification command. Other commands can be used for erasing part of the data or for clearing the entire buffer. The problem with these commands is that they can also be misrecognized resulting in a frustrating exercise and in the addition of more errors to be corrected.

iii) Response time

In order for voice data entry to be competitive with other data entry media such as keyboards, it is necessary for the recognition process to be as fast as possible. Immediate feedback of the recognition

results must also be given to the user, usually as a direct echo of the words entered. Efficient use of the feedback for verification requires the delay to be minimized.

1.3 ORGANIZATION OF THE THESIS

The thesis presents a research work in the speaker independent recognition of isolated words. The vocabulary of interest is composed of the 36 alpha numeric digits and the 14 control words: YES, NO, SET, ADD, DELETE, STORE, MULTIPLY, CONTROL, READ, INPUT, OUTPUT, LOAD, WRITE, END.

The first chapter in the thesis discusses the role of speech as an input/output mode in man-machine communication. The difficulties encountered in achieving the goal of this interaction, which is the motivation of the research are briefly outlined.

Chapter 2 opens with a discussion of the basic structure and mechanism of human speech production system, the nature of the speech signal and human perception system. The rest of the chapter is devoted to a review of the methods used in speech recognition systems. Isolated, connected, continuous and speech understanding systems are covered in a general sense, rather than a detailed description, with a minimum of mathematics. However, essential information is included to give the reader an understanding of the basic techniques involved.

Chapter 3 is concerned with the modelling of non-linear time variations of the speech utterances. This poses one of the major problems in pattern matching based word recognition since it involves the comparison of speech patterns of different temporal lengths. Several speech pattern time normalization methods are examined and compared on the basis of their word recognition performance.

The use of filter bank features in word recognition is examined in Chapter 4. The way the performance of different filter banks, i.e. number of filters, type of filters, frequency spacing, etc affect the

accuracy of word recognition is investigated. Techniques for improving recognition performance are also examined.

Chapter 5 is concerned with word recognition based on linear prediction coding (LPC) coefficients. The problem of improving recognition performance, reducing memory requirements and attaining low computational complexity in recognition systems is explored. Finally, a recognition system termed the LPC/VQ/SPLIT which is a hybrid of some two well established recognizers is presented. The proposed recognizer has the advantage of requiring far less memory storage and maintains comparable performance to the established schemes.

In Chapter 6, the application of the fuzzy set theory in the segmentation of speech into voiced, unvoiced and silence is proposed and compared with an established method. The resulting segmentation process enables the broad acoustic structure of an utterance to be identified. The rest of the chapter deals with improving the performance of the LPC based recognizer by incorporating the information about the acoustic structure of the utterance as side information.

The final chapter provides a recapitulation of both the novel recognition schemes proposed in this thesis and the main results obtained experimentally by computer simulations. Suggestions for further research work are also made.

CHAPTER 2

REVIEW OF SPEECH RECOGNITION SYSTEMS

2.1 INTRODUCTION

In this chapter, the concepts and theories underlying speech recognition systems are presented. The chapter commences with a discussion on the theory of speech production and the nature of speech signals. A brief historical survey of the developments in speech recognition is also presented.

Speech recognition systems can be considered as belonging to one of the three categories: (i) isolated word systems, (ii) connected word systems, or (iii) continuous speech systems, depending on the form of speech input they are expected to accept. The general techniques applied in each of these three categories are discussed in order to expose the difficulties involved in the recognition task, and the aspects of the problems still to be solved.

2.2 THE THEORY OF SPEECH PRODUCTION [10][11]

The human speech production mechanism consists of an excitation source and a time varying resonant cavity formed by the vocal tract. Figure 2.1 illustrates the cross-sectional view of the vocal system.

The vocal tract is a non-uniform tube with an average length of 17 cm. It is terminated at one end by the glottis, which is an opening between the vocal cords, and at the other end by the lips. The cross-sectional area of the vocal tract varies along its length, from a complete closure to about 20 sq cm as determined by the movement of the lips, jaws, tongue and velum. The nasal cavity which begins at the velum and terminates at the nostrils can be coupled to the vocal tract by the action of the velum to produce nasal sounds. Otherwise during the generation of non-nasal sounds, the velum seals off the vocal tract from the nasal cavity.

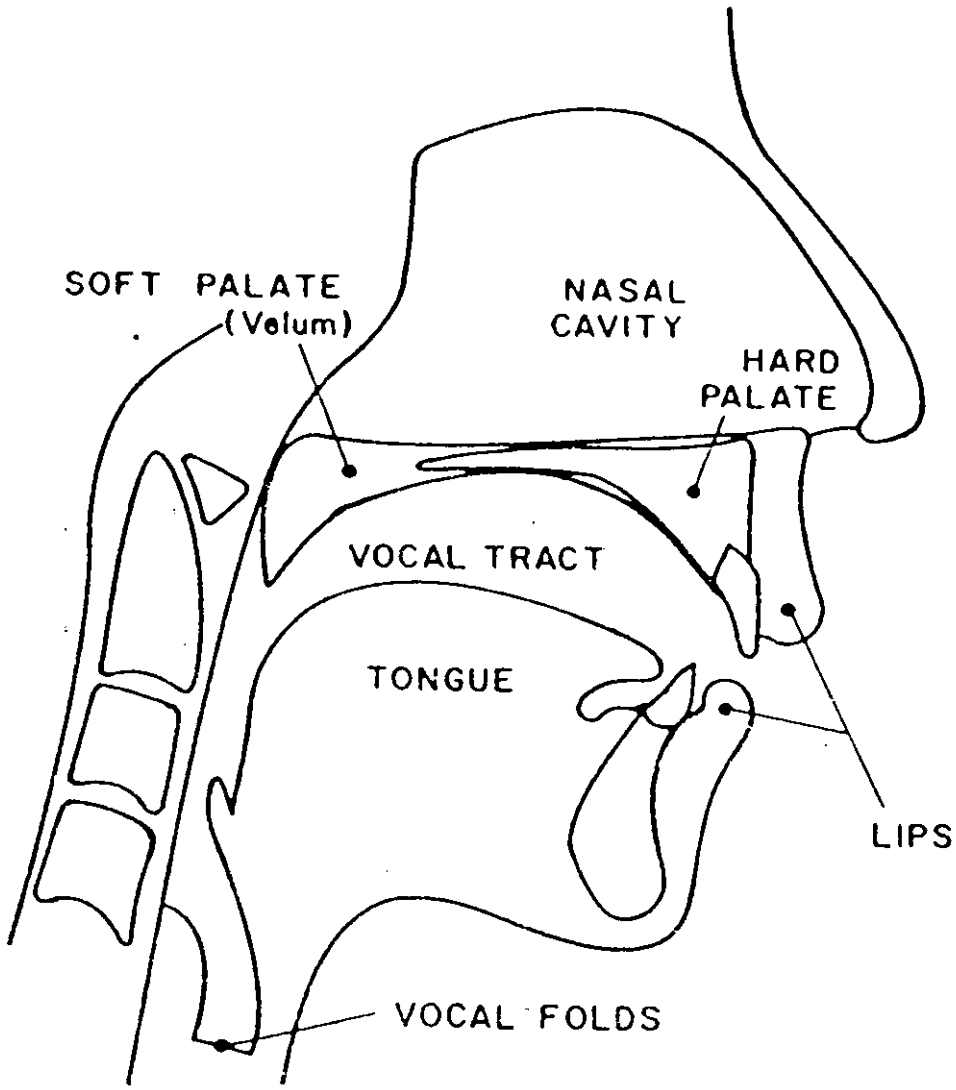


FIGURE 2.1: A CROSS-SECTIONAL VIEW OF THE VOCAL SYSTEM [11]

There are three basic ways in which sounds can be generated by the vocal system. Voiced sounds are produced by forcing the air from the lungs through the glottis with the tension of the vocal cords adjusted so that they vibrate in a relaxed oscillation. The interrupted air flow produces quasi-periodic broad spectrum pulses which excite the vocal tract. The English vowels a, e, i, o, u are produced in this manner.

Fricative sounds such as /f/, /th/, /s/, /sh/ etc, are generated by using the articulators to form a constriction at some point on the tract and then forcing air through the constriction to produce turbulent flow. With both a constriction and vocal cord vibration, voiced fricatives such as /v/ are generated.

Plosive sounds such as /p/, /g/ and /t/ are generated by making a complete closure usually towards the lips end of the vocal tract, building up air pressure behind the closure, and then suddenly releasing the air.

Click sounds unique to certain South African languages [12] are generated by the concurrence of two points of closure on the tongue, the back one always being velar. The air enclosed between the two points undergoes a rarefaction by the backward and downward movement of the tongue. When the front closure is released the air rushes into the mouth.

All these techniques of speech production involve the modification of the frequency spectrum of the excitation source by the vocal tract. The rather loose interaction between the vocal system and the sound sources, can be approximately represented as linearly separable. Figure 2.2 shows the source tract model of speech production with the underlying linear systems theories. The sound radiated from the lips s(t), can be approximated as the convolution of the vocal tract impulse response h(t), and the excitation signal g(t), i.e.

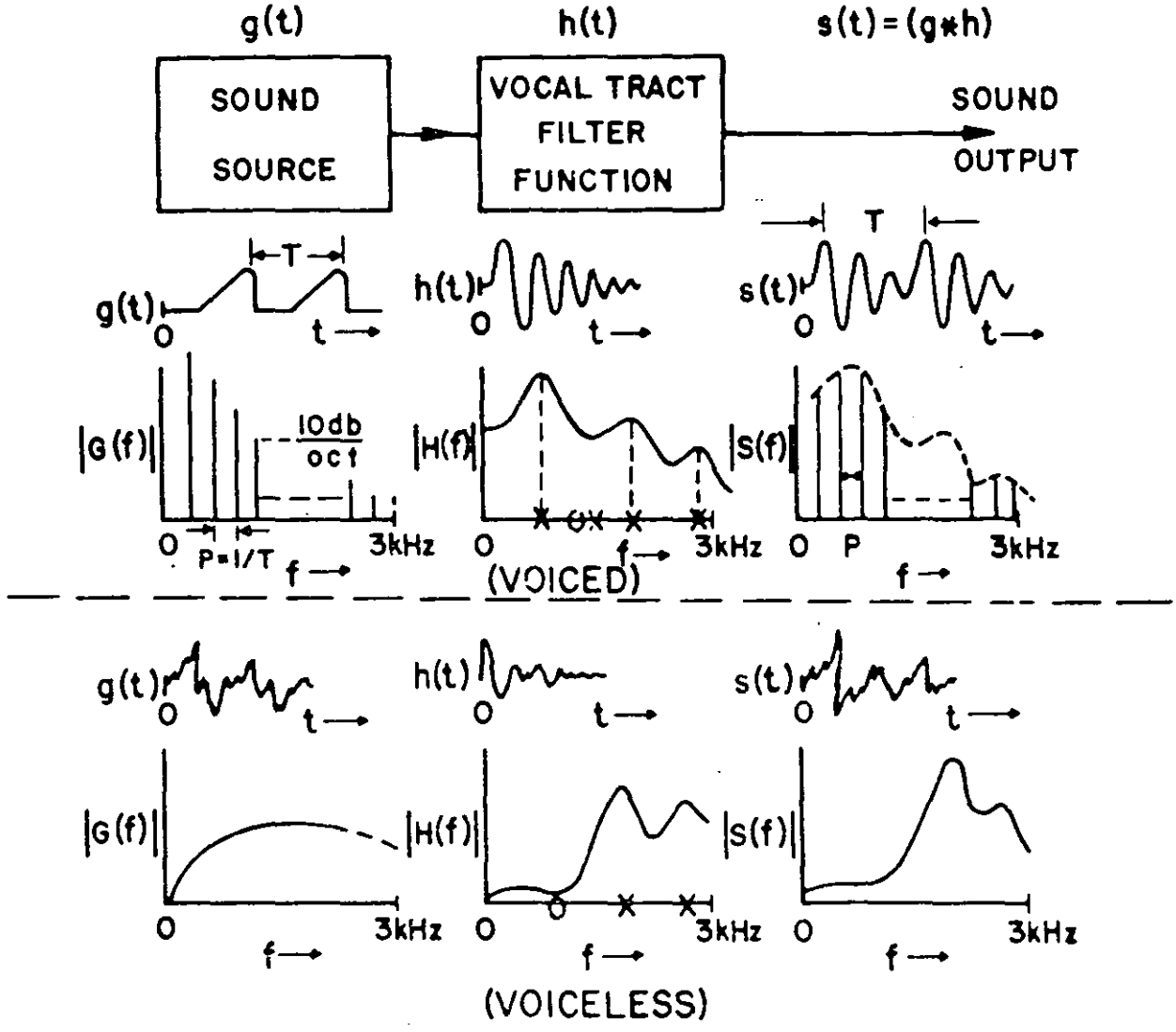


FIGURE 2.2: THE SOURCE TRACT MODEL OF SPEECH PRODUCTION [10]

$$s(t) = g(t) * h(t) \qquad 2.1$$

The impulse response h(t) is time varying, since the various articulators in the vocal tract are continuously in motion. The excitation source g(t) too is time varying. However, both the source and vocal tract responses can be considered as stationary in the short term. This concept is the basis of most recognition systems in which characteristic features are extracted from temporal segments of a speech signal.

2.3 CHARACTERISTICS OF SPEECH SOUNDS

Figure 2.3(a) shows a time waveform of voiced speech. The fine pseudo-periodic structure observed in the waveform arises from the psuedo-periodic excitation source.

From the short time frequency spectrum of the voiced signal shown in Figure 2.3(b), it can be observed that most of the spectral energy is concentrated in the lower frequency spectrum. The envelope of the spectrum exhibits resonances arising from the frequency response of the vocal tract. It is usual to find at least three dominant resonant peaks below 4 kHz. The resonances are referred to as formants.

A time waveform of unvoiced speech and its short term spectrum are shown in Figure 2.4(a) and (b) respectively. Both the responses appear to be 'noise-like' owing to the nature of the excitation signal. The spectrum does not exhibit the fine resonant structure observed with voiced sounds.

Speech signals are approximately stationary within short time intervals and non-stationary over a long time duration. Voiced and unvoiced sounds can further be subdivided into elemental speech units by taking into consideration their spectrographic characterization as well as the manner and place of articulation. These elemental speech

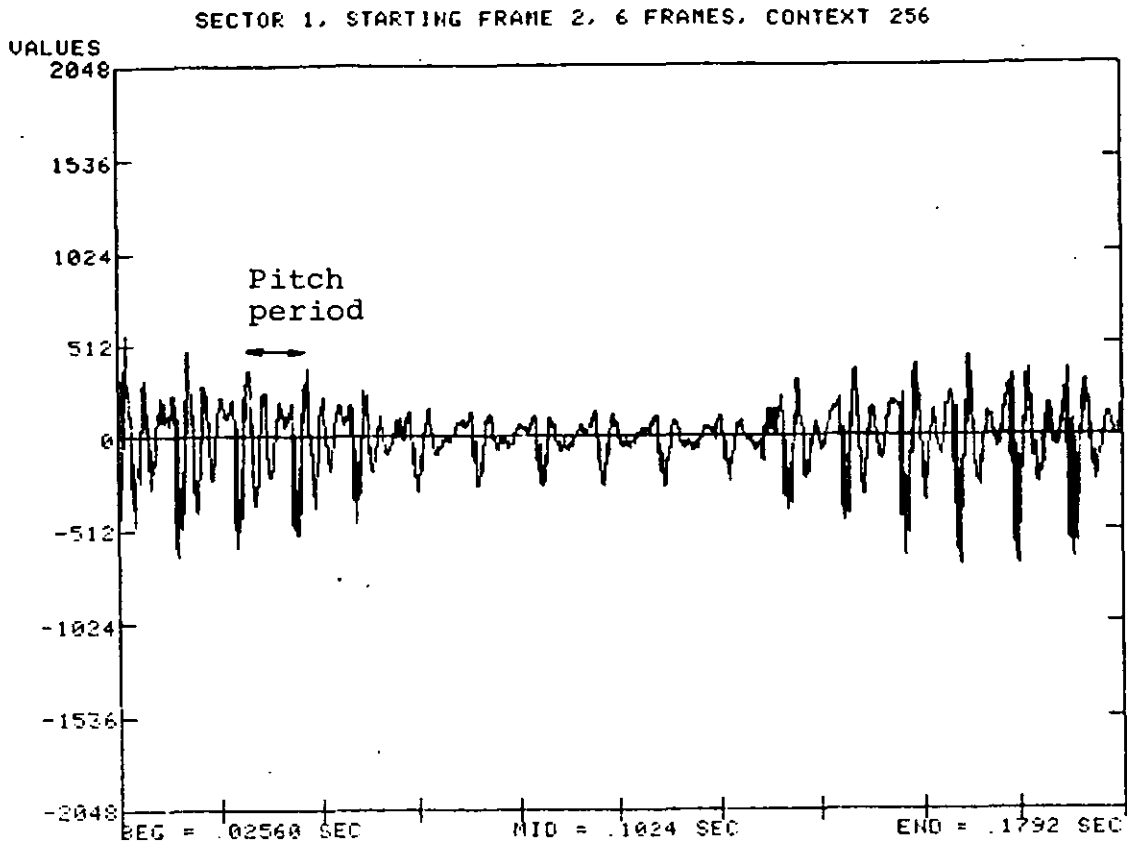


FIGURE 2.3(a): TIME WAVEFORM OF VOICED SPEECH

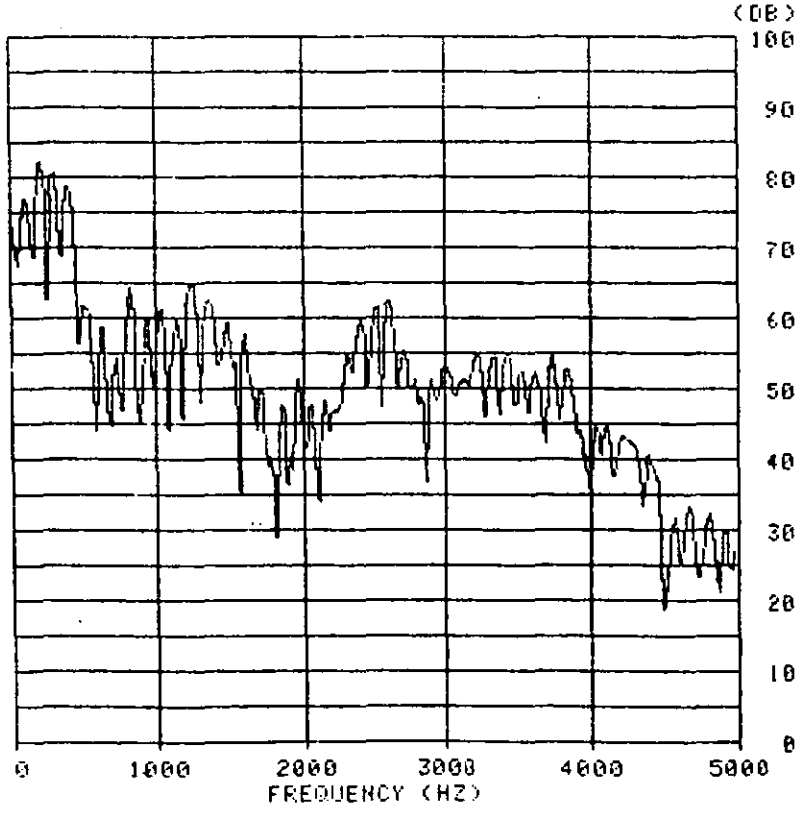


FIGURE 2.3(b): SHORT TIME FREQUENCY SPECTRUM OF A 32 msec VOICED SPEECH SEGMENT

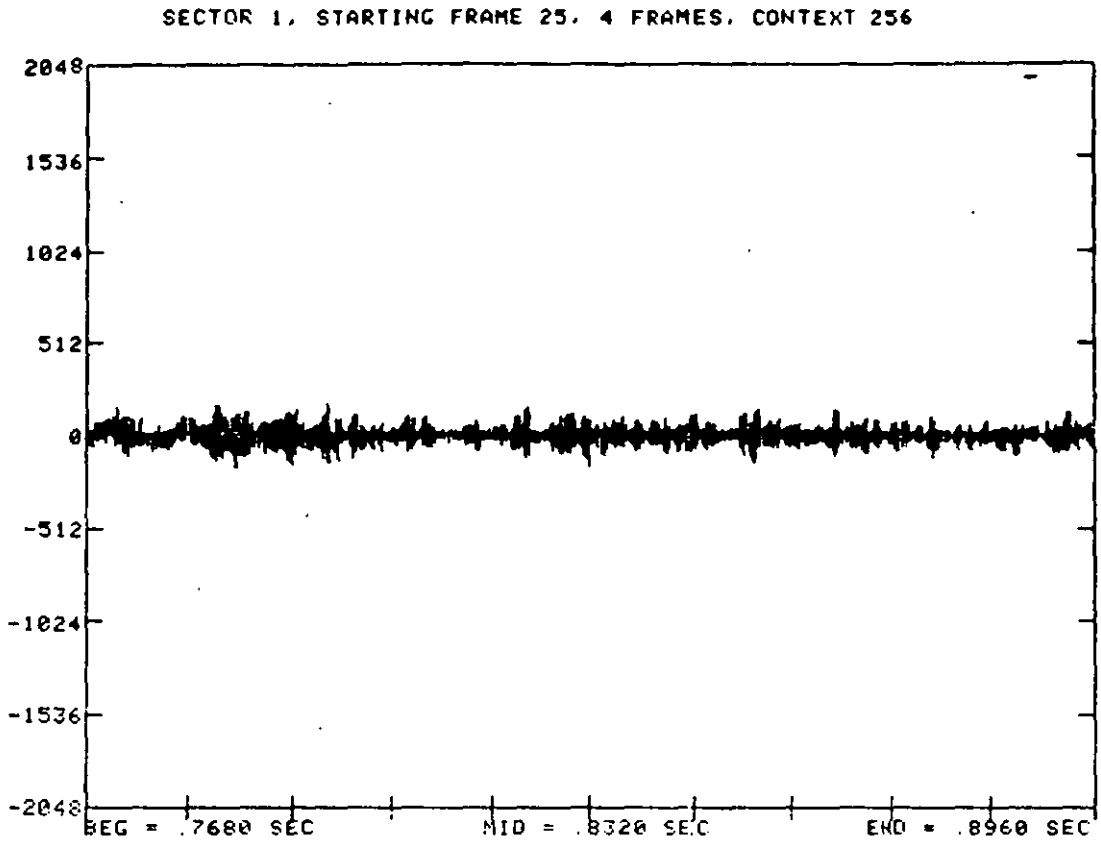


FIGURE 2.4(a): TIME WAVEFORM OF UNVOICED SPEECH

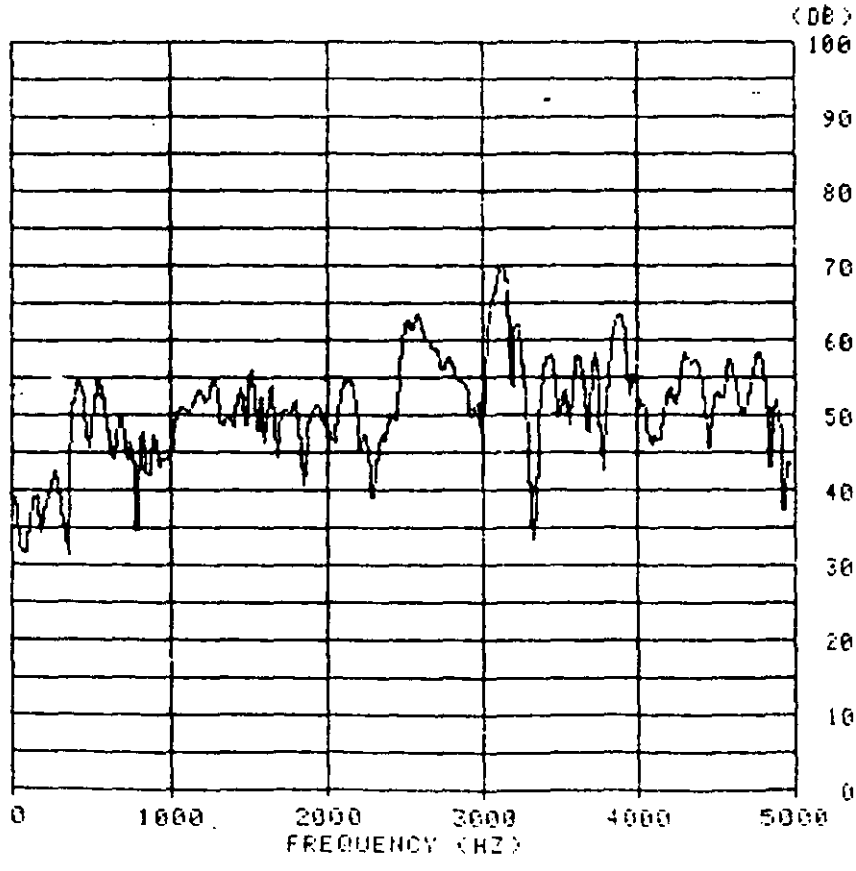


FIGURE 2.4(b): SHORT TIME FREQUENCY SPECTRUM OF A 32 msec UNVOICED SPEECH SEGMENT

units, known in linguistics as phonemes, are the 'building blocks' of spoken words and play a similar role to letters in written language. The set of phonemes in any language is quite small compared with the set of all the possible words. In the English language, there are about 42 phonemes grouped into the four broad categories: vowels, semivowels, diphthongs and consonants [13].

2.4 A BRIEF HISTORY OF AUTOMATIC SPEECH RECOGNITION

Automatic speech recognition is generally defined as the process of extracting the message in a speech utterance. Most of the research work in this field has been done within the last thirty years with the early attempts having been initiated by the following two factors:

i) the major developments in electronics, after the invention of the tube in 1930, gave rise to the availability of new electrical circuits for signal analysis;
ii) the introduction of the vocoder by Dudley [14] in 1939 and the sound spectrograph by Potter et al [15] in 1947, gave a better understanding of the information bearing elements in a speech signal. The sound spectrograph, displaying a 3-dimensional plot of the speech energy - time axis - and frequency, for the first time gave scientists the chance of measuring and observing the changing acoustic cues in the speech signal along its time axis. This helped to increase the knowledge on the nature of speech signals and their method of production thus opening the way for future speech technology.

2.4.1 The Early Work in the 1950s

The first successful recognizer, reported by Davis, Biddulph and Balashek [16] of the Bell Laboratories in the USA in 1952, was concerned with the identification of the ten digits spoken in isolation. Their method of analysis was based on dividing the frequency spectrum of the speech signal into two bands, one above and the other below 900 Hz. A count was made on the number of times the

signal level in each of the two bands crossed the zero amplitude axis (i.e. zero crossings count).

This indicated the approximate frequency range in which the energy in the speech signal is concentrated. Displaying the two measurements on a horizontal and vertical axis yielded what is now known as a first formant-second formant (F1-F2) plot. The resulting pattern of an input digit was then cross-correlated with each of the pre-stored patterns of the ten digits, zero to nine. The digit pattern which had the highest cross-correlation with the input was chosen as the identity of the digit. The device could recognize which of the ten digits was spoken, with an accuracy of over 90%, if the pre-stored patterns had been formed from the samples of that particular speaker. With the speech of a different speaker, however, the accuracy could be as low as 50%.

Apart from its historical significance, the Davis recognizer introduced the technique of reducing the input speech signal into a pattern and then comparing it with pre-stored reference patterns, a method which is still in force today.

Dudley and Balashek [17] in 1958 developed a recognizer that analyzed the speech signal by splitting into ten frequency bands and extracting certain features whose durations were compared with pre-stored reference patterns of the vocabulary words. A major aspect of the approach by Dudley, and his contemporaries like Fry and Denes [18], was the attempt to segment words into phonetic units. Fry and Denes used a phonetic set comprising four vowels and nine consonants and stored estimates on the probability of a given phoneme following other phonemes. The overall performance of these early recognizers, especially in a speaker independent mode, was not impressive, ranging from 24 to 44%. Nevertheless, these early attempts at recognition did demonstrate the value of using the spectrograph as a useful tool in the study.

2.4.2 The 1960s

The use of the digital computer in speech recognition was first employed in the early 1960s by Denes and Matthew [19]. They reported a system of recognizing the ten digits using a 17 channel spectrum analyzer to obtain word patterns. The resulting time-frequency patterns of a number of utterances for each digit were averaged and stored as reference patterns.

The time-frequency pattern, from an input digit utterance to be identified, was compared by a cross-correlation process with each stored pattern. The digit was classified as the reference pattern giving the best match. An important concept in the Denes and Matthew recognizer, was the introduction of time normalization of the speech patterns. Short versions of an utterance, that were spoken at a faster rate than the reference utterances, were stretched out to equal the duration of the reference utterance. On the other hand, slowly spoken long versions of an utterance were compressed, to match the length of the reference utterance. Experiments showed that better recognition rates were obtained with the time normalization than without it.

Another early use of computers in speech recognition was in the work reported by Forgie and Forgie [20] of the Lincoln Laboratories in the USA, dealing with the identification of fricatives in initial and final positions of isolated words.

The introduction of the Fast Fourier Transform (FFT), in the mid-1960s by Cooley and Tukey [21], made it possible to achieve complex mathematical analysis of speech waveforms with reasonable computational effort and also paved the way for fully digital speech recognition systems. This, along with the desire to market small scale recognition products, led to the development of special purpose hardware. At about the same time, recognizers were also developed for other languages such as Japanese by Nagata et al [22] and German by Musman et al [23].

At the end of the 1960s speech scientists had begun to expand their domain to the recognition of continuous speech. Early attempts were reported in 1968, by Otten [24], who proposed the application of syllabic units, prosodics, and finite state language to represent the structure of speech dialogue with a machine.

2.4.3 The Early 1970s

By 1970, the interest in continuous speech recognition had developed to such a stage that the Information Processing Technology Office of the Advanced Research Projects Agency (ARPA) of the United States Department of Defence, found it necessary to initiate a five year research programme [25]. The objective of the research was to make a breakthrough in the speech understanding capability that could be used in a practical man-machine communication system. The ARPA study group emphasized that the recognition of continuous speech needed to use, not only the advanced techniques achieved in acoustic analysis, but also required a methodological approach with the inclusion of grammatical, semantic and prosodic constraints, together with phonological rules which govern a given human language. These constraints would represent various knowledge sources that could be brought to attain successive understanding of the message in speech. The ARPA project called for a system that would accept continuous speech from any cooperative speaker. The language was limited to a vocabulary of 1000 words and allowed to have an artificial syntax appropriate to a limited task situation, e.g. data management, chess playing etc.

When the ARPA project ended in 1976 a number of task dependent systems: HARPY, HEARSAY, HWIM and SDC, which could understand spoken utterances within a given context had been developed [25]. Many of the present day continuous speech recognition systems still employ the techniques investigated during the ARPA project.

Apart from the ARPA project, the work reported by Baker [26] and simultaneously by Jenelik [27] has also contributed immensely to the

speech recognition research. The DRAGON system proposed by Baker, models the knowledge sources necessary for automatic recognition of continuous speech, as a probabilistic function of a Markov process.

In conjunction with the ARPA project and the DRAGON system, two other major developments in the early 1970s which helped to accelerate the pace in speech recognition, especially for isolated words, were the introduction of Linear Prediction Coding (LPC) and Dynamic Programming (DP) techniques by Itakura in 1975 [28]. LPC, which is based on the speech synthesis mode, is particularly suitable since it describes efficiently the spectral characteristics of speech. On the other hand, DP provides an extremely useful technique in optimizing the temporal differences between speech utterances.

2.5 ISOLATED WORD RECOGNITION SYSTEMS

Isolated word recognition systems deal with speech input in the form of words spoken in isolation. Since distinct pauses exist between words, the problem of separating one word from another does not arise. The recognition process of isolated words begins by digitizing the speech utterance which still results in a large number of data points. It then becomes necessary to employ a data reduction technique whereby the large set of data points is transformed into a small set of features which are equivalent in the sense that they faithfully describe the properties of the speech waveform. Usually a data reduction ratio between 100 and 10 is generally acceptable.

2.5.1 Feature Extraction

Different sets of features for representing speech signals have been proposed, ranging from simple measurements such as zero crossing rates to the more complex short time spectral parameters. The motivation for choosing one feature set over another is often dependent on the constraints imposed on the system in terms of cost, speed and recognition accuracy. Some of the commonly used features are discussed below:

i) Filter bank features

A popular set of features used in speech recognition is the output of a bank of filters. The speech utterance is passed through a bank of bandpass filters covering the speech bandwidth. The energy in the speech signal in a given frequency band is estimated from the output of the respective bandpass filter. For a given time instant on the speech utterance, a set of energy features define an Nth order feature vector, where N is the number of bandpass filters employed. Thus the whole utterance can be expressed as a pattern of discrete Nth order feature vectors.

ii) Linear prediction features

Another widely used set of features, is based on the linear prediction coding of speech, and was first proposed for recognition purposes by Itakura [28]. Linear prediction is based on the assumption that a speech sample can be approximated as a linear combination of a number of immediately preceding samples.

For each sample, a prediction error e(n), is defined as follows:

$$e(n) = x(n) - \bar{x}(n) = x(n) - \sum_{i=1}^{p} a(i)\, x\,(n-i) \qquad 2.2$$

where $\bar{x}(n)$ is the linearly predicted sample and x(n) is the actual sample.

On minimizing the mean square prediction error e(n), over a finite interval, a unique set of predictor coefficients, a(i), i = 1,2, ...p, can be obtained. These coefficients give a good short term spectral estimate. Thus, an utterance can be represented as a sequence of discrete vectors of linear prediction coefficients.

iii) Articulatory parameters

An ideal set of features for describing speech sounds would be the parameters giving the position of the tongue, lips, jaws and the velum as functions of time. Since these articulators take an infinite number of positions, a statistical analysis of X-ray data [29] can be done to determine an effective representation of the articulator movements in a reduced dimension space. The parameters can also be estimated from the speech signal [30][31].

2.5.2 The Pattern Matching Model for Isolated Word Recognition

Figure 2.5 shows the typical pattern matching model employed in the majority of isolated word recognition systems. It consists of three main stages:

i) feature extraction stage
ii) pattern comparison stage
iii) decision rules stage

The input to the model is an isolated utterance which is to be identified within a given vocabulary. After an analogue to digital conversion, features like those described in Section 2.5.1 are extracted from temporal segments of the utterance. The resulting pattern is compared with pre-stored patterns of reference vocabulary words.

A decision rule is used to identify the input word as the reference vocabulary word giving the best match. This pattern matching model has been widely used with the following three advantages:

i) the model is invariant for different vocabularies, feature sets, pattern comparison measures and decision rules
ii) it is easily implemented
iii) it has been found to give satisfactory performance in practice.

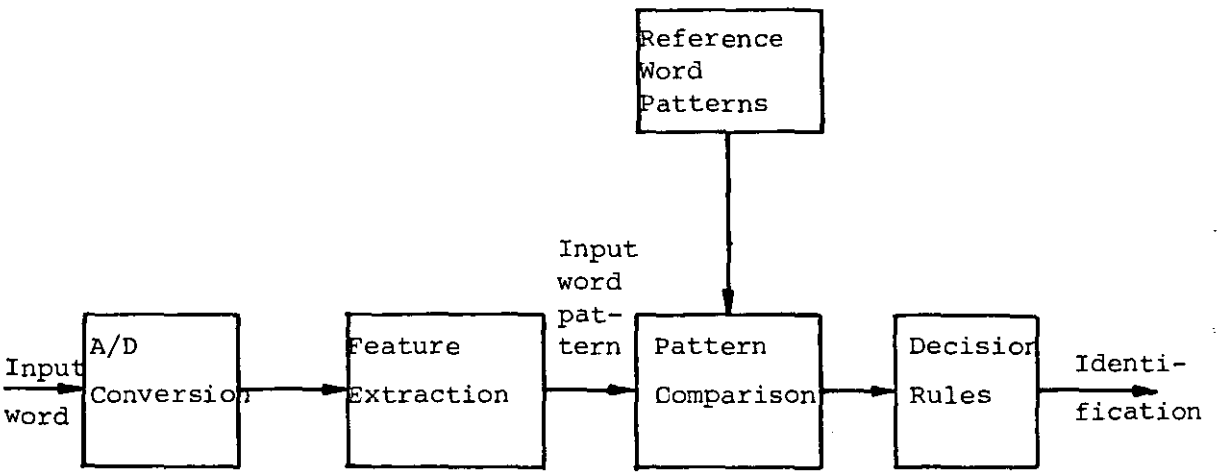


FIGURE 2.5: A PATTERN MATCHING MODEL FOR ISOLATED WORD RECOGNITION

One major difficulty in the pattern comparison stage is that speech utterances are rarely of equal temporal length. Their duration are dependent on the speaker, his rate of speaking and on the circumstances. The same is true, even if the speaker is the same person who recorded the reference patterns. One solution to this problem is to map the time axis of one pattern onto the other such that maximum coincidence is attained.

The performance of the word recognizer is severely degraded if the input and the reference speech utterances are obtained from different speakers. However, these inter-speaker differences can be reduced by employing a large number of reference patterns obtained from different speakers per vocabulary word. As a consequence of using multiple reference patterns per vocabulary word, the response time and the memory requirements of the system are greatly increased. In such situations clustering techniques, like vector quantization [32] in which a group of similar feature vectors are mapped into a single vector, can be applied to the reference patterns in order to reduce the memory requirements.

2.5.3 Stochastic Modelling Approach for Isolated Word Recognition [33]

It would be quite natural to consider speech as being generated by a stochastic process of the type described by hidden Markov chains. A hidden Markov process consists of two inter-related mechanisms, an underlying Markov chain having a finite number of states, and a set of random functions, one of which is associated with each state. At a given time instant, the hidden Markov process is in a unique state and an observation is generated by the random function associated with the state. This causes the underlying Markov chain to change state in accordance with its transition probabilities. These states cannot be observed directly, only the outputs of the random functions at each state are seen.

In speech production, the vocal tract and the various articulators can be approximated as being in one of a finite number of articulatory configurations, or states. In a given state, a short time speech signal which can only have one of a finite number of spectral shapes is generated. Thus the short term spectrum of the speech signal is determined by the current state, while the spectral variation with time is determined by a transition probability distribution of the underlying Markov chain.

Let the underlying Markov chain have the N states:

$$q_1, q_2, \ldots\ldots, q_N$$

and let the set V of K spectral shapes, also referred to as symbols, be

$$(v_1, v_2, \ldots, v_k, \ldots, v_K)$$

The underlying Markov chain can be specified in terms of an initial state distribution vector $\pi = (\pi_1, \pi_2, \ldots \pi_N)$ and a state transition matrix $A = [a_{ij}]$, $1 \leqslant i \leqslant N$, $1 \leqslant j \leqslant N$. π_i is defined as the probability of observing state q_i at time t=1. The value of a_{ij} is the probability of a transition to state q_j given the current state q_i

i.e. $$a_{ij} = \text{prob } (q_j \text{ at } t+1 \,|\, q_i \text{ at } t) \qquad 2.3a$$

The random process associated with each state can be collectively represented by the stochastic matrix $B = [b_{jk}]$, $1 \leqslant j \leqslant N$, $1 \leqslant k \leqslant K$. The value b_{jk} is the probability of observing the spectral shape v_k given the current state q_j and is denoted as:

$$b_{jk} = \text{prob } (v_k \text{ at } t \,|\, q_j \text{ at } t) \qquad 2.3b$$

A hidden Markov model, M, is thus specified by the set (π, A, B). Efficient methods for estimating parameters in the matrices A and B have been proposed by Baum [34].

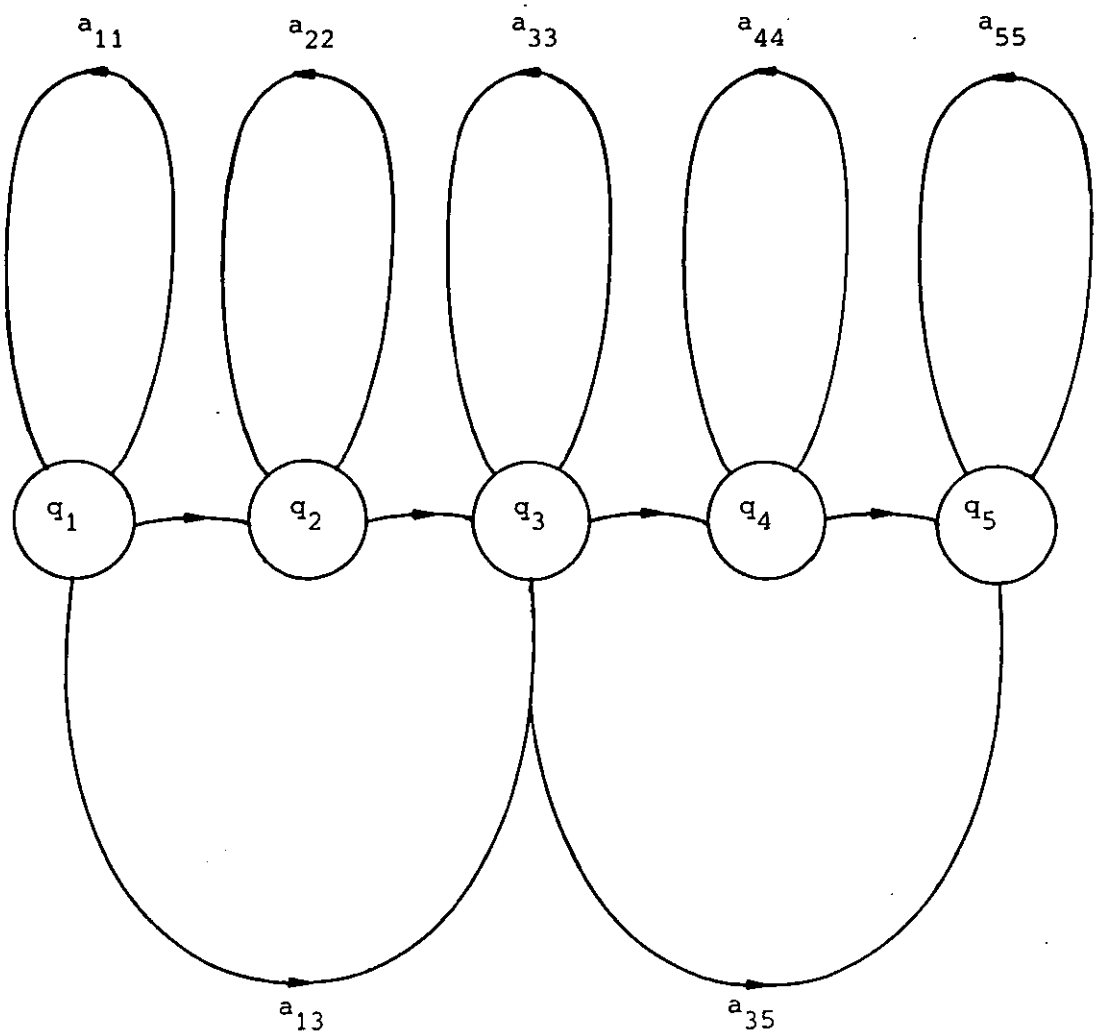
Figure 2.6 illustrates the structure of a hidden Markov model with 5 states, i.e. N=5. The model starts in state q_1 and terminates in state q_5 by progressing from the left to the right without re-visiting states which have been left. In the model, transitions within the state, i.e. a_{11}, a_{22} etc are allowed, so are transitions that skip neighbouring states, i.e. a_{13}, a_{35} etc. By imposing different constraints on the transitions, other variants of the hidden Markov models can be obtained.

In order to use the hidden Markov models to perform speech recognition, it is necessary first to generate the set of spectral shapes V, usually by vector quantization of LPC coefficients. Next a large number of repetitions of each vocabulary word are used as a training set to derive the hidden Markov model for each vocabulary word.

The speech recognition problem is thus specified as follows. Given a set W of R words vocabulary, $W = (w_1, w_2, \ldots, w_R)$ and a set of hidden Markov models for each word, $M_1, M_2, \ldots, M_R$. For an unknown word $w_i \in W$, with an observation sequence $O_i = (O_1, O_2, \ldots O_\ell \ldots, O_L)$, where each $O_\ell \in V$, $1 \leqslant \ell \leqslant L$.

The probability, P_i, that this sequence was generated by the model M_i is given by:

$$P_i = \text{prob}\,(O_i/M_i) = \sum_{q_1 q_2 \cdots q_N} \pi_{q_1} b_{q_1}(O_1)\, a_{q_1 q_2} b(O_2) \cdots a_{q_{N-1} q_N} b_{q_N}(O_L) \qquad 2.4$$



b_{11}	b_{21}	b_{31}	b_{41}	b_{51}
b_{12}	b_{22}	b_{32}	b_{42}	b_{52}
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
b_{1k}	b_{2k}	b_{3k}	b_{4k}	b_{5k}

FIGURE 2.6: A HIDDEN MARKOV MODEL

The interpretation of equation 2.4 can be made as follows. The initial state q_1 exists with a probability π_{q_1}. The spectral shape O_1 is generated with probability $b_{q_1}(O_1)$. Then a transition is made to state q_2 with probability $a_{q_1 q_2}$ to generate spectral shape O_2 with probability $b_{q_2}(O_2)$. The process is continued until the last transition from state q_{N-1} to state q_N with the probability $a_{q_{N-1} q_N}$ and the spectral shape O_L generated with probability $b_{q_N}(O_L)$.

A computationally efficient algorithm for evaluating equation 2.4 has also been proposed by Baum [34].

The unknown utterance is classified as W_i if, and only if,

$$P_i \geqslant P_j \quad \text{for} \quad 1 \leqslant j \leqslant R \qquad 2.5$$

2.6 CONNECTED WORD RECOGNITION SYSTEMS

In connected word recognition, the speech is a sequence of words from a specified vocabulary. Typical examples include the digit strings of telephone numbers, identification codes etc. where the vocabulary is the 10 digit set (0-9) and connected letters, like in word spellings, where the vocabulary is the 26 letter set (A-Z).

The recognition of connected words can be performed by applying the pattern matching techniques of isolated word recognition. Thus, as illustrated in the block diagram in Figure 2.7, the connected speech recognition system is almost identical to the isolated word recognizer, except for the introduction of a section in which connected patterns are synthesized.

Let an input speech pattern consisting of words of unknown length, be expressed as the discrete sequence, C, of length I:

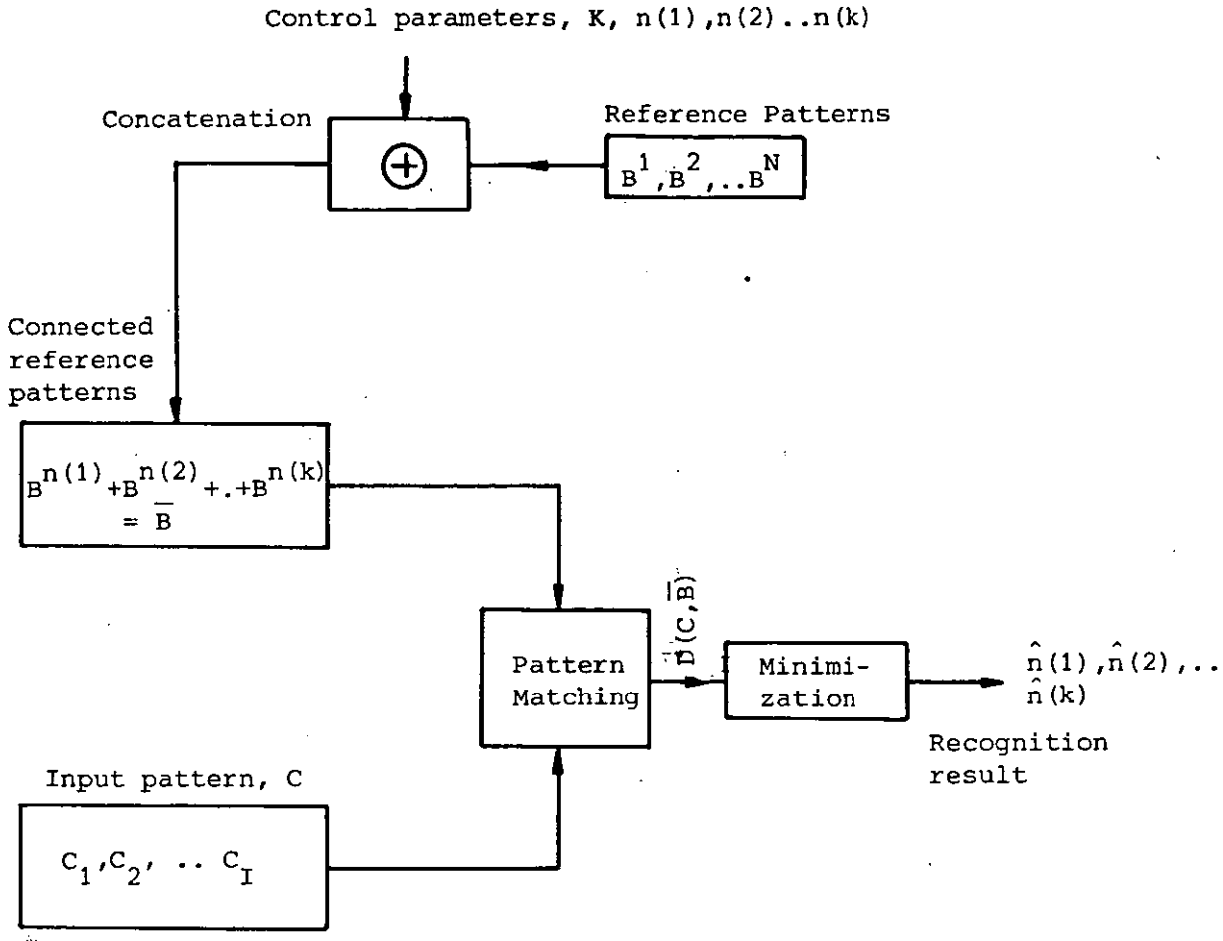


FIGURE 2.7: A CONNECTED WORD RECOGNITION SYSTEM [35]

$$C = c_1, c_2, \ldots, c_i, \ldots c_I \qquad 2.6a$$

and let the vocabulary be the set of N words (1, 2, ... n,....N). The reference pattern of word m is represented as a discrete sequence, B^m of length J_m

i.e. $$B^m = b_1^m, b_2^m, \ldots, b_j^m, \ldots b_{J_m}^m \qquad 2.6b$$

The concatenation of two speech patterns B^m and B^n is denoted as $B^m \oplus B^n$ and is the sequence:

$$B^m \oplus B^n = b_1^m, b_2^m, \ldots, b_{J_m}^m, b_1^n, b_2^n, \ldots, b_{J_n}^n \qquad 2.7a$$

A connected pattern $\bar{B}$ of words n(1), n(2), ... n(k) is synthesized by concatenating their reference patterns as follows:

$$\bar{B} = B^{n(1)} \oplus B^{n(2)} \oplus \ldots \oplus B^{n(k)} \qquad 2.7b$$

The unknown input speech pattern C, is matched with the synthesized reference pattern $\bar{B}$, to give a distance $D(C,\bar{B})$. The matching process is repeated, changing the number of words k and the indexes n(1),n(2),... n(k), until all the permutations of the indexes are exhausted.

The optimum parameters, $k=\hat{k}$ and $n(x) = \hat{n}(x)$, $x = 1,2,\ldots\hat{k}$ which give the minimum distance D(C,B) are determined. The unknown input pattern C, is identified as the $\hat{k}$ connected words $\hat{n}(1), \hat{n}(2),\ldots\hat{n}(\hat{k})$. The minimization problem can be expressed as the solution to the following equation:

$$D_{min} = \underset{k,n(x)}{MIN} \left[D(C,\ B^{n(1)} \oplus B^{n(2)} \oplus \ldots + B^{n(x)} \oplus \ldots + B^{n(k)}) \right] \qquad 2.8$$

Unfortunately, except for trivial cases with short sequences and small vocabularies, the exhaustive solution of equation 2.8 is impractical due to the excessive computation involved. A suitable approach for solving equation 2.8 has been proposed by Sakoe [35], in which the minimization problem is considered in two stages, one for the word level and the other for the whole connected words level.

A partial pattern $C(\ell, m)$, of the input pattern C is defined as:

$$C(\ell, m) = C_{\ell+1},\ C_{\ell+2},\ \ldots,\ C_m;\quad \ell \geqslant 0,\ m \leqslant I \qquad 2.9$$

In splitting the pattern C into k partial word patterns, the (k-1) word boundaries, $\ell(1)$, $\ell(2)$, ..., $\ell(k-1)$ are assumed. Thus:

$$C = C(\ell(0),\ell(1)) \oplus C(\ell(1),\ell(2)) \oplus \ldots \oplus C(\ell(k-1),\ell(k)) \qquad 2.10$$

where $\ell(0) = 0$ and $\ell(k) = I$.

The distance between C and a concatenated reference pattern $B^m \oplus B^n$, is given by:

$$D(C,\ B^m \oplus B^n) = \underset{\ell}{MIN}\ D(C(0,\ell),\ B^m) + D(C(\ell, I),\ B^n)) \qquad 2.11$$

Inserting equation 2.10 into 2.8 and applying the relationship defined in 2.11 gives:

$$D_{min} = \underset{k,\ell(x)}{MIN} \{ \sum_{x=1}^{k} \underset{n(x)}{MIN} \; D\,[C(\ell(x-1),\ell(x)),\; B^{n(x)}]\} \qquad 2.12$$

Equation 2.12 is solved by dynamic programming methods. Other techniques for solving the connected word recognition problem have been proposed by Myers et al [36].

2.7 THE RECOGNITION OF CONTINUOUS SPEECH

In a continuous speech recognition system, the input speech is in the form of naturally spoken words in a given language. The aim of the recognition system is either to identify the words or to decode the message in the input speech. The latter is also referred to as speech understanding.

A study of human speech perception can provide a useful insight into the modelling of a computer recognition system for continuous speech.

2.7.1 The Human Speech Perception [37]

The human speech perception process comprises several stages of analysis, namely:

i) auditory
ii) phonetic
iii) phonological
iv) prosodical
v) lexical
vi) syntactic
vii) semantic
viii) pragmatic

The speech signal heard by the listener is transformed by the cochlea, an organ located in the inner ear, into a time varying pattern whose main features are the concentration of energy in a frequency-time

space. In the cochlea too, there are neurological systems for extracting features such as the fundamental frequency, intensity, spectral shape, duration of the speech signal and representing them by psychoacoustical sensations like pitch, loudness and timbre.

The phonetic stage involves the extraction of features such as labiality, nasality, voicing and frication which serve to discriminate between specific speech sounds. The auditory and phonetic stages of speech processing appear to be closely inter-related since certain consonantal speech sounds are known to be perceived as of being a particular phonetic type group without being discriminated within the group. Thus, the auditory processing stage reduces the speech signal into a continuous parametric representation of frequency and time, on which the listener imposes a categorization on the space of sounds, from his linguistic knowledge, to give a string of phonemes.

The phonological stage serves to bring aspects of the variability in pronunciation of words in the particular language, to bear on the perception process. Phonetic sequences of many words are markedly reorganized when the words are used in certain phrases. For example [38], for the phrase ' would you' which when spoken fast appears as 'wujeu', the listener must invert the generative rule:

/wud/ + /ju/ = /woje/

so as to perceive correctly.

In the prosodical stage, information on stress patterns, intonation and pauses are extracted from the speech signal. This information gives a clue as to whether the message in speech is a question, statement, command etc.

Lexical, syntactic and semantic information are invoked to give the phonetic string a meaningful message. The lexical information pertains to the vocabulary words in the language known to the listener. The syntax is the grammatical structure of the language,

which describes not only the way words are concatenated to form sentences, but also the manner in which phonemes form syllables, and syllables form words. Semantics is the meaning of the words. Pragmatics refers to the context of the conversation. Using these four knowledge sources, a sentence can be rejected if it is inconsistent with one of the knowledge sources. For example, [39], consider the following sentences:

i) Sleep roses dangerously young colourless
ii) Colourless yellow ideas sleep furiously
iii) Colourless paper packages crackle loudly

The first sentence is syntactically and consequently semantically unacceptable. The second sentence is syntactically correct but meaningless. The last sentence, though syntactically and semantically correct, would be rejected because it is pragmatically inconsistent. Generally, human listeners do experience difficulties in decoding the pragmatic concepts of a language unless they are well conversant with the context of conversation.

If the knowledge is incomplete or inaccurate, human listeners tend to make hypotheses. In many cases an unambiguous interpretation is possible on the basis of incomplete phonetic representation, by generating a hypothesis that represents the listener's expectation of the continuation of the utterance. These observations have been reported by Warren [40] as the phoneme restoration effect, in which selected phonemes were removed from words in sentences and replaced by various forms of noise and listeners still continued to 'hear' the missing sounds.

In another experiment described by Reddy [39], subjects were asked to listen to a sentence and then write down what they heard. The results obtained are given in Table 2.1 and generally show that the listeners try to form their own hypothesis as to what was said. There is also the failure to detect the end of one word and the beginning of the other, which contributes to erroneous hypothesis.

TABLE 2.1: HYPOTHESIS GENERATION IN HUMAN LISTENERS [39]

ACTUAL PHRASE	RESPONSE
in mud eels are, in clay none are	1st subject: in muddies sar, in clay mannar
	2nd subject: in my deals are, en clannannar
	3rd subject: in my ders, en clain
	4th subject: in model sar, in claynanar

Despite these setbacks, it is evident that the study of human speech perception can be of immense benefit to continuous speech recognition.

2.7.2 The Continuous Speech Recognition Model [25]

The continuous speech recognition system can be modelled as a two level hierarchy consisting of an acoustic processor as the first level and a language analyzer as the second level. Such a model is illustrated in the block diagram in Figure 2.8.

i) The acoustic analyzer

The first step, in the processing of the continuous input speech signal, is to transform it to a discrete symbol string. The symbols may be phonemic, syllabic or actual words depending on the segmentation process employed in the system. The transformation of speech into phonemic symbols involves feature detection, segmentation and labelling. The features commonly extracted are energy and fundamental frequency which serves to detect frication, voicing, silence and stress in the speech signal within the segment. Each segment is then labelled with the closest phonemic symbol. Before the symbol sequence can be applied to the language analyzer it is necessary to apply phonological rules to combine segments, change labels based on context, and delete transitional segments. Syllabic symbols can be extracted by detecting energy dips in the speech signal.

When words in a sentence are separated with brief pauses, the process of detecting the beginning and ending points for the words can be done on the basis of temporal variation of energy in the speech waveform. Energy minima of sufficient duration would indicate word boundaries.

Let the language L, in the recognition task, be limited to a vocabulary V consisting of M symbols (i.e. words, syllables or phonemes) V_1, V_2,... V_M. An arbitrary sentence W in the language can be expressed as a string of symbols:

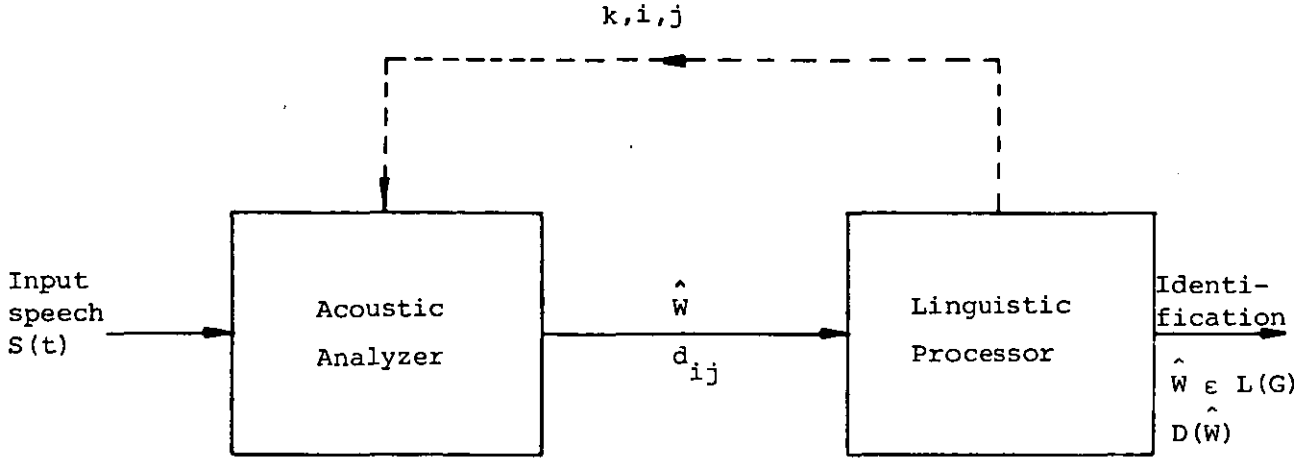


FIGURE 2.8: A TWO LEVEL CONTINUOUS SPEECH RECOGNITION MODEL

$$W = W_1 W_2 \ldots W_i \ldots W_{K'} \qquad 2.13$$

where $W_i \varepsilon$ V, for $1 \leq i \leq K'$.

The sentence W, as defined in 2.13 is encoded in s(t), the input speech to the acoustic analyzer. The acoustic analyzer decodes the speech to obtain an input sentence $\tilde{W}$, which is a corrupted form of W, i.e.

$$\tilde{W} = \tilde{W}_1 \tilde{W}_2 \ldots \tilde{W}_i \ldots \tilde{W}_K \qquad 2.14$$

where $\tilde{W}_i \varepsilon$ V for $1 \leq i \leq K$; but $\tilde{W}$ is not necessarily a sentence in L. The acoustic analyzer also computes a distance matrix D.

$$D = [d_{ij}] \quad 1 \leq i \leq K, \quad 1 \leq j \leq M \qquad 2.15$$

where d_{ij} is the distance between the ith symbol $\tilde{W}_i$ and the vocabulary symbol V_j.

ii) Linguistic processor

The language processor accepts the symbol string $\tilde{W}$ and distance matrix D from the acoustic processor and then produces the string $\hat{W}$:

$$\hat{W} = \hat{W}_1 \hat{W}_2 \ldots \hat{W}_K \qquad 2.16$$

for which the distance $D(\hat{W})$ is given by:

$$D(\hat{W}) = \text{MIN} \sum_{i=1}^{K} (d_{ij_i}) \quad 1 \leq j_i \leq M \qquad 2.17$$

where d_{ij_i} is the distance between W_i and all the symbols V_j that may appear in the ith position of sentences in L. Equation 2.17 is minimized subject to the constraint that $\hat{W}$ is a sentence in the language L.

Without any constraints, given a vocabulary consisting of M different words and admitting sentences having a maximum of K words each, the number of possible sentences that can be formed would be of the order of M^K. The exponential growth in the number of sentences can be constrained by imposing a grammatical structure, as defined by the syntax, and invoking the relationship between objects and events in accordance with semantic rules. This process can be approximated by elementary formal language theory and has been advantageously exploited in the computer recognition of continuous speech.

A language which is generated by a grammar G is denoted as L(G). The grammar G is a function of four arguments:

$$G = G(V_T, V_N, S, P),\ V_N \cap V_T = 0 \qquad 2.18$$

where V_T is a finite set of symbols out of which sentences are formed i.e. vocabulary of possible words which are also designated as terminal symbols. V_N is another set of symbols disjoint from V_T, but whose members define V_T.+ Symbols in V_N are also described as non-terminal symbols and would refer to generalized parts of a sentence like a predicate, verb, adjective, etc. S, which is a member of V_N ($S \varepsilon V_N$), is designated as a starting symbol, and would refer to a complete sentence. P is a finite set of transformations, termed production rules. Typically each production rule expresses a possible way of transforming a non-terminal symbol into a sequence of one or more symbols (terminal, non-terminal or both) as indicated below:

+ For example, V_N could be thought of as a phrase and V_T as a word

$$a \rightarrow B,\ a,\ B \in (V_N \cup V_T)^* \qquad 2.19$$

the asterisk * stands for the set of all strings of elements in the designated set.

A sentence expressed as a string of symbols $W = W_1, W_2, \ldots W_i, \ldots W_K$, where $W_i \in V$, for $1 \leq i \leq K$, is said to belong to the language L(G), if and only if there exists a sequence of production rules which can derive W from a starting symbol S, i.e.

$$W \in L(G), \text{ if } S \rightarrow a_1,\ a_1 \rightarrow a_2, \ldots,\ a_M \rightarrow W \qquad 2.20$$

The use of different production rules leads to languages of different properties and complexities as formulated by Chomsky [14]. If the production rules are of the form:

$$\begin{array}{ll} & A \rightarrow a\ B \\ \text{or} & C \rightarrow b \end{array} \qquad 2.21$$

where $V_N = (A, B, C)$ and $V_T = (a, b)$, then the grammar is classified as a Chomsky type 3, also known as Regular grammar.

Regular grammars may be used to generate or analyze a subset of a natural language appropriate to a certain task and can be represented by a state transitional diagram as illustrated in Figure 2.9 for an airline reservation system [42]. In the diagram, transition from one state to another is dependent on the production rules in the grammar G. The edges are labelled with the terminal symbols in V_T, which are the vocabulary words. The finiteness of the language is implied in the state diagram by disallowing any path from starting and ending at the same state.

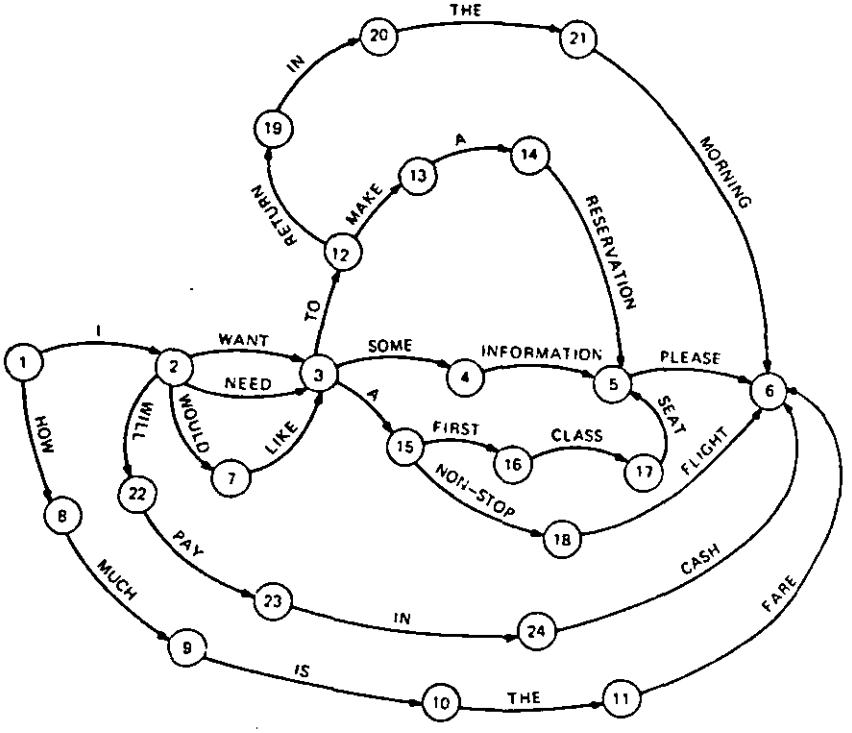


FIGURE 2.9: A STATE TRANSITIONAL DIAGRAM [42]

Sentences or symbols in V_N can be generated by starting from node 1 and following any path to final node 6. As each transition is made, the vocabulary word associated with the transition is added to the rest of the string already formulated.

The particular problem addressed by the formal language theory which is relevant to speech recognition is that of parsing sentences in a language. This specifically means determining whether $W \varepsilon L(G)$, using a sequence of production rules for the derivation of $S \rightarrow W$. The acoustic processor provides the distance matrix D, and the language analyzer finds the sentence $W \varepsilon L(G)$ which satisfies equation 2.17. For example, if the acoustic transcription gives the sentence $\hat{W}$ in equation 2.14 as 'WOULD MUCH IS TO FARE', it would be clear from the state diagram in Figure 2.9 that the sentence is invalid since there is no path whose edges are so labelled. The parsing algorithm described by Levinson [42] aims to produce from $\tilde{W}$, a string of words with a valid path in the state diagram and with a minimum total distance. The production rules of the grammar are invoked in order to achieve the process.

iii) Syntax directed approach to continuous speech recognition

In practice it is difficult to determine accurately the phonemic or syllabic boundaries in speech.

Phonemes are not easily determined acoustically due to co-articulation. Furthermore, some sounds can belong equally to more than one phoneme. It has been estimated [39] that phoneticians seem to agree only 51% of the time, when labelling continuous speech in unfamiliar languages.

The problem with syllabic segmentation is that sometimes the boundaries cannot be specified uniquely. For example [38], for the word 'common' it is not clear whether the syllables are /kom/ + /on/ or /ko/ + /mon/.

In continuous speech too, pauses do not normally exist between words, and detecting the beginning and the ending points of words is an error prone process.

One of the methods of recognizing continuous speech in the presence of unreliable segmentation is to use syntax to direct the matching of continuous speech symbols to reference prototype words [42]. In this case the word segmentation problem is overcome by generating many hypothetical segmentations and choosing the best one. This 'hypothesis and test' approach requires the establishment of a feedback link between the acoustic processor and the language analyzer as shown in Figure 2.8. The language analyzer provides the acoustic processor with a list of those symbols which can occur at a given node in the sentence being processed and the acoustic processor computes only those entries in the distance matrix. Based on the parsing stage in the sentence, the linguistic analyzer also specifies the approximate time t_s when the vocabulary symbol v should occur.

In the acoustic processor, the output speech string beginning at t_s is matched to the reference prototype word v to obtain the distance D_v and a computed endpoint t_e. The endpoint will then be used as the nominal beginning point for the next word in the candidate sentence. The parsing process is implemented by a dynamic programming recursion identical to equation 2.17, except for a pointer which is included to keep track of the segmentation.

2.8 SPEECH DATA BASE AND EQUIPMENTS

The speech data base in this research work was recorded in a silent room from the utterances of four male subjects SM1, SM2, SM3, SM4 and three female subjects SF1, SF2, SF3. All the subjects were native speakers of the English language. The subjects read the vocabulary of 50 words in the order listed in Table 2.2, in a casual and cooperative manner. It was considered important to arrange the digit and alphabet words set in a random manner so as to reduce the co-articulation between adjacent words.

TABLE 2.2

THE LIST OF VOCABULARY WORDS AS READ BY THE SUBJECTS DURING THE RECORDING SESSION

Order	Vocabulary Word	Order	Vocabulary Word
1	DELETE	26	FOUR
2	NINE	27	STORE
3	INPUT	28	L
4	F	29	G
5	O	30	A
6	W	31	V
7	Z	32	Y
8	K	33	NO
9	THREE	34	E
10	ZERO	35	I
11	WRITE	36	Q
12	END	37	FIVE
13	SIX	38	READ
14	J	39	U
15	D	40	X
16	S	41	TWO
17	LOAD	42	P
18	N	43	EIGHT
19	ONE	44	C
20	ADD	45	T
21	M	46	YES
22	H	47	R
23	B	48	SEVEN
24	SET	49	MULTIPLY
25	CONTROL	50	OUTPUT

Each vocabulary word was spoken twice by each male subject and once by each of the female subjects. The recorded speech was bandlimited to 5 kHz+ and digitized at 10 kHz using a 12 bit A/D converter and subsequently transferred to the hard disk of a computer.

The results presented in this thesis were obtained by simulations using the DEC PDP 11/34 and PDP 11/73 computers.

2.9 DISCUSSION

In this chapter, an attempt has been made to review developments and techniques used in speech recognition. A comparison of the recognition accuracy of the various systems is difficult to make. This is because the accuracy of a given system is not only dependent on the design techniques, but also on a number of diverse factors such as: the recognition vocabulary, noise level in the speech signal, speech signal bandwidth, etc. As a general rule, however, isolated word systems can, and do, achieve better performance than connected word systems which in turn have a better performance than continuous speech systems.

There is still a demand for an increase in performance in isolated word recognition systems which can be achieved by solving the following problems:

i) speaker independence
ii) reduction in computation time
iii) reduction in memory requirements
iv) discrimination of acoustically similar words in the vocabulary.

The rest of this thesis is primarily concerned with the recognition of isolated words, and attention is focused on solving the above four problems.

+ The 5 kHz bandwidth was selected to provide comparability with other research work

CHAPTER 3

TIME NORMALIZATION IN SPEECH PATTERNS

3.1 INTRODUCTION

The variation in the speaking rate, which is mainly dependent on the manner of the speaker and on his emotional status, means that different repetitions of the same word will rarely be of equal temporal length. The elimination of these fluctuations in speaking rate, or time normalization as it is often called, is a central issue in speech recognition systems based on comparison of patterns of unequal length. A linear transformation of the time axis, in order to eliminate the temporal differences between speech patterns, will prove inadequate to deal with the highly non-linear fluctuations of the speaking rate. In this Chapter, several non-linear time normalization algorithms are discussed, and their performance is assessed with the aim of selecting the algorithm to be employed in the proposed isolated word recognition systems described in subsequent chapters.

3.2 DYNAMIC TIME WARPING

Temporal differences between two speech patterns, can be eliminated by warping the time axis of one of the patterns onto the other, such that maximum coincidence is attained. This requires the modelling of the time axis fluctuations by a non-linear function of some specified properties.

Let A(t) and B(t) be two speech patterns referred to as input and reference patterns respectively, which are not necessarily of equal temporal length. The time normalization problem is to find a function F(t) which maps the pattern A(t) onto the corresponding parts of B(t) such that the distance, D(A,B), between the two patterns is minimized.

Thus, F(t) is such that:

$$D(A,B) = \text{MIN} \int_{t_o}^{t_a} D(A(t), B(F(t))).G(t,F(t),\dot{F}(t))\, dt \qquad 3.1$$

where t_o and t_a are points on the time axis which indicate the starting and ending point of the input pattern A(t). F(t) is specified to be a monotonically increasing and continuously differentiable function. $\dot{F}(t)$ is the derivative of F(t). D(A(t),B(F(t))) is the distance of an individual point in A at time t from a point in B at time F(t). G is a weighting function which is dependent on t, F(t) and $\dot{F}(t)$.

Unfortunately, there is no simple solution to the continuously variable problem of equation 3.1 and the only alternative is to use discrete functions. If the two speech patterns are time sampled with a constant and common sampling period, then the time warping function, F(n), can now be determined as the solution to the problem:

$$D(A,B) = \underset{F(n)}{\text{MIN}} \sum_{n=1}^{N} D(A(n), B(F(n))) \qquad 3.2$$

where D(A(n),B(F(n))) is the distance between the n^{th} discrete frame of the input pattern and the frame F(n) of the reference pattern.

Dynamic programming methods can be used efficiently to define the optimum time warping function, F(n), according to equation 3.2, which minimizes the total distance between the two patterns. The optimization process is known as Dynamic Time Warping (DTW). A variety of DTW algorithms can be obtained by imposing different restrictions on the warping path.

3.2.1 The Sakoe and Chiba DTW Algorithms [43]

Let the two speech patterns A(t) and B(t) be expressed as a sequence of discrete multi-dimensional vectors, i.e.

$$A = a_1, a_2, \ldots, a_i, \ldots, a_I \qquad 3.3a$$

$$B = b_1, b_2, \ldots, b_j, \ldots, b_J \qquad 3.3b$$

where I and J are the number of vector frames in pattern A and B respectively.

A matrix of distances, d(i,j) is computed as:

$$d(i,j) = d(a_i, b_j),\ 1 \leqslant i \leqslant I, \quad 1 \leqslant j \leqslant J \qquad 3.3c$$

The distance between pattern A and B, can be defined along a path F, in the i-j plane as illustrated in Figure 3.1.
Thus:

$$F = f(1), f(2), \ldots, f(k), \ldots f(K); \quad 1 \leqslant k \leqslant K \qquad 3.4$$

where f(1) = (1,1) and f(K) = (I,J).

Let the grid point (i,j) at f(k), be denoted as (i(k), j(k)), and the distance between the two feature vectors a_i and b_j at this point as d(f(k)) = d(i,j). Then, the weighted sum of the distances along the warping function is given by:

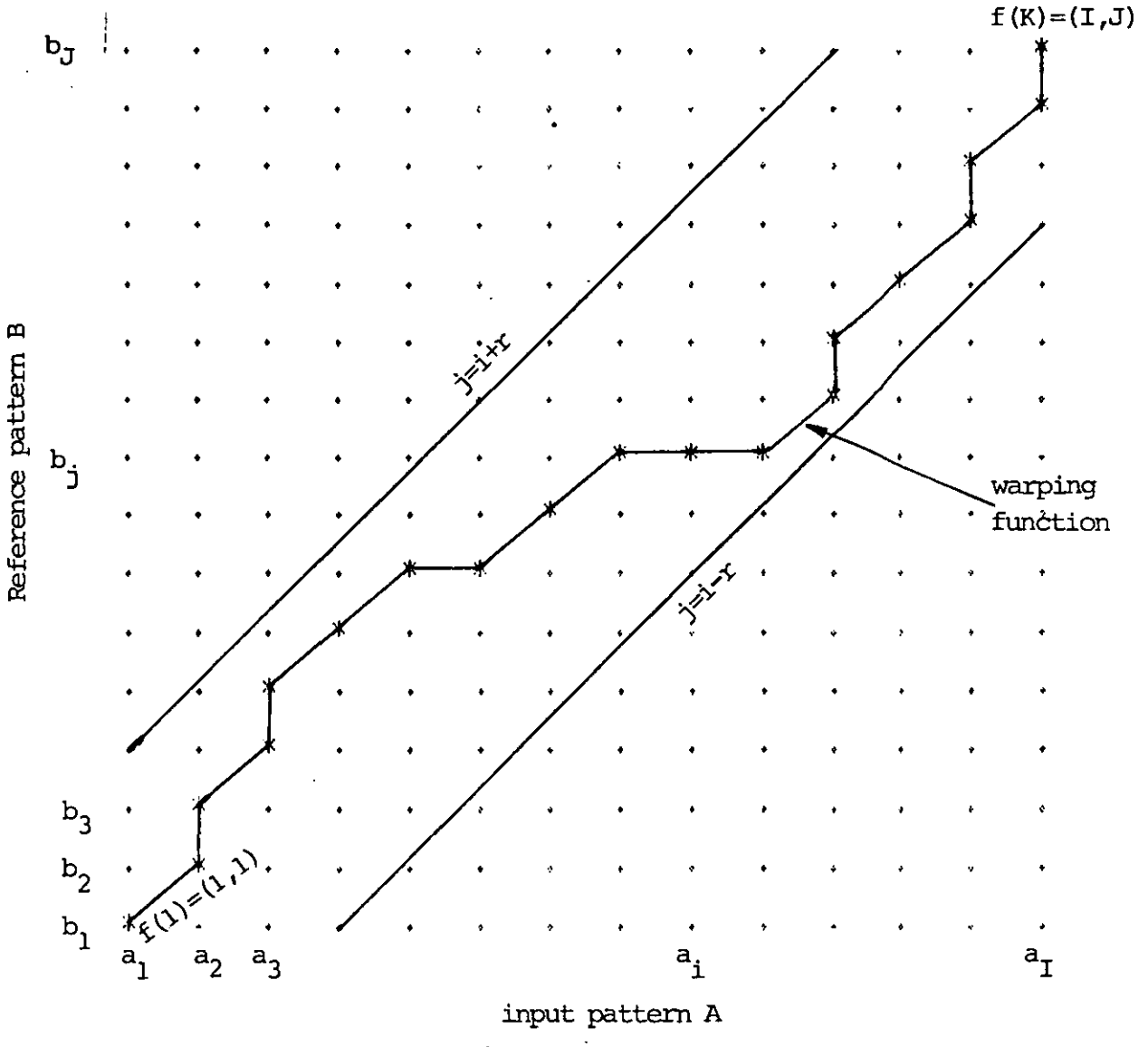


FIGURE 3.1: THE TIME WARPING FUNCTION

$$D(F) = \sum_{k=1}^{K} d\,(f(k)).\ w(k) \qquad 3.5$$

where w(k) is a positive weighting coefficient related to the length of the path from f(k-1) to f(k). D(F) attains a minimum value when the warping function is chosen to provide an optimum time alignment between the two patterns. Then the time normalized distance, D(A,B), between patterns A and B, as given by equation 3.2 can be expressed as:

$$D(A,B) = \underset{F(n)}{\text{MIN}} \left[\sum_{k=1}^{K} d(f(k)).w(k) \right]/N \qquad 3.6$$

where N is a normalization constant used to compensate for the number of points on the warping function.

i) Restriction on the warping path

If the assumption that the input speech pattern A, and the reference pattern B coincide precisely at the initial frame and at the final frame, i.e. f(1) = (1,1) and f(K) = (I,J), then the solution to the DTW equation 3.2 is equivalent to finding the 'best' path through a finite set of grid points, and as such any 'path finding' technique can be used. However, the warping path is a model of the time axis fluctuations of the speech and, accordingly, it should reflect these fluctuations by preserving essential linguistic structures. In speech patterns these structures are continuity, monotonicity, and limitation on the duration of acoustic segments, and can be realised on the warping function by imposing the following conditions:

a) Monotonic conditions:

$$i(k-1) \leqslant i(k) \qquad \text{and} \qquad j(k-1) \leqslant j(k) \qquad 3.7$$

b) Continuity conditions:

$$i(k) - i(k-1) \leqslant 1 \quad \text{and} \quad j(k) - j(k-1) \leqslant 1 \qquad 3.8$$

c) Boundary conditions:

$$i(1) = 1, \quad j(1) = 1 \qquad 3.9a$$

and

$$i(K) = I, \quad j(K) = J \qquad 3.9b$$

d) Window length condition:

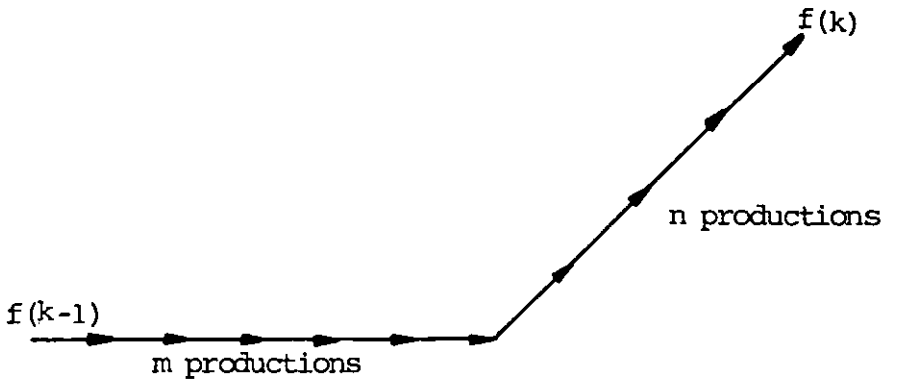
$$|j(k) - i(k)| \leqslant r \qquad 3.10$$

where r is a suitable positive integer termed the adjustment window length. This condition removes the possibility of an excessive time difference between the two speech patterns. The maximum value of r is the absolute difference in frames between the input and the reference patterns.

e) Warping path gradient condition:

The gradient of the warping path should not be allowed to be too steep, nor too gentle, since it can result in the unrealistic correspondence between a short segment of one pattern with a long segment of the other pattern under comparison. A situation like this would occur if a short segment of a phoneme transition in one speech pattern is in good coincidence with an entire vowel steady state segment in the other speech pattern. Thus, it is necessary to restrict the warping function gradient to a certain range which will not cause undesirable time axis warping.

Consider two consecutive points f(k-1) and f(k) on the warping function as illustrated in Figure 3.2. The point f(k) is derived from the point f(k-1) by either horizontal, vertical or diagonal



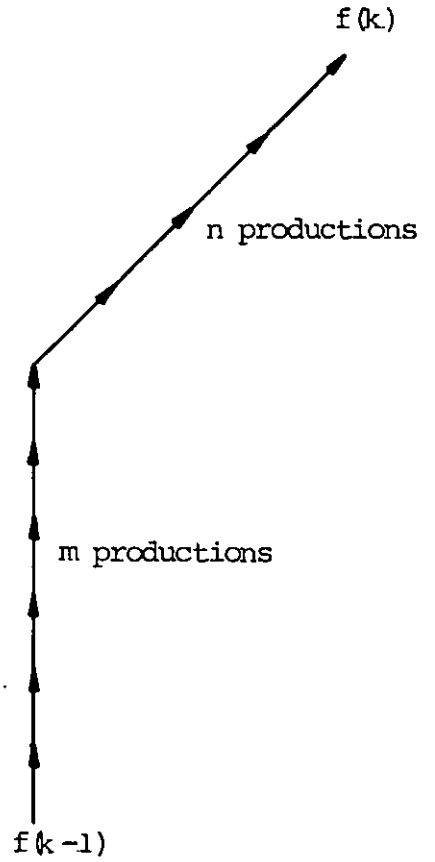


FIGURE 3.2: THE WARPING PATH GRADIENT CONSTRAINT

productions or a series of their combinations. A minimum gradient condition can be realised by requiring n diagonal productions to be preceded by m horizontal productions. Similarly, a maximum gradient condition can be realised by having the n diagonal productions preceded by m vertical productions. So the gradient of any prospective path between f(k-1) and f(k) is restricted by these two values.

The intensity of the gradient constraint can be expressed as a measure P, given as:

$$P = n/m \qquad 3.11$$

Figure 3.3 illustrates the various series of productions for the different values of P. For example, in Figure 3.3c, the contribution to the grid point (i,j) comes either from (i-3, j-2), or (i-1, j-1), or (i-2, j-3). The minimum gradient from f(k-1) to f(k) has two diagonal productions and one horizontal production, and the maximum gradient has two diagonal productions and one vertical production. The gradient constraint measure P for this case is 2. The larger the value of P, the more restricted the gradient of the warping function. In the case where $P = \infty$ the warping path would be restricted to the diagonal line i = j and the non-linear time normalization is not achieved. When P = 0, there is no restriction on the gradient of the warping function.

ii) The weighting coefficient and the normalized constant

The computation of the time normalized distance D(A,B) between the two speech patterns $A = \{a_1, a_2, \ldots a_i, \ldots a_I\}$ and $B = \{b_1, b_2, \ldots, b_j, \ldots, b_J\}$ as given in equation 3.6, requires the specification of the weighting coefficient w(k), and the normalization coefficient N, which is dependent on w(k). Several

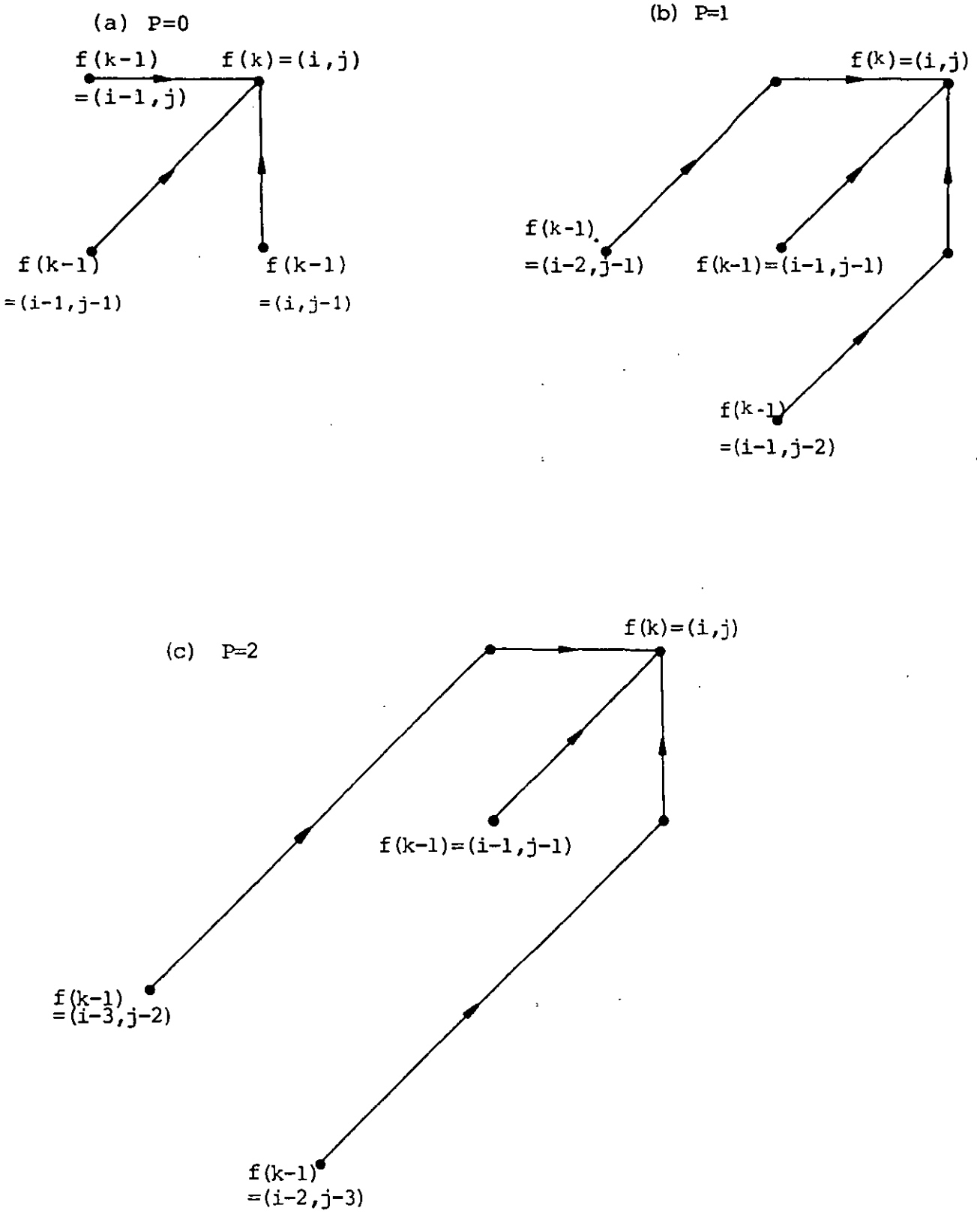


FIGURE 3.3: WARPING PATH PRODUCTIONS FOR P=0, P=1, AND P=2

weighting functions which depend only on the local productions of the warping path have been proposed [43] and are of the form:

Type a: $w(k) = i(k)-i(k-1)$ 3.12a

Type b: $w(k) = j(k)-j(k-1)$ 3.12b

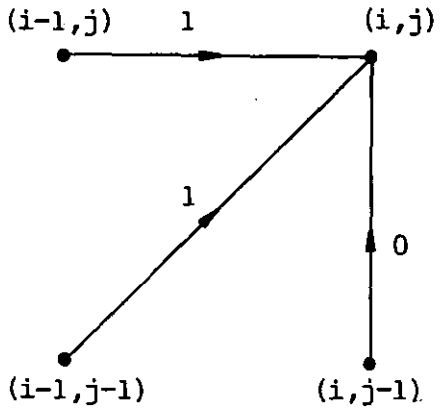
Type c: $w(k) = (i(k)-i(k-1) + j(k)-j(k-1))$ 3.12c

Type d: $w(k) = MAX\ [(i(k)-i(k-1), (j(k)-j(k-1))]$ 3.12d

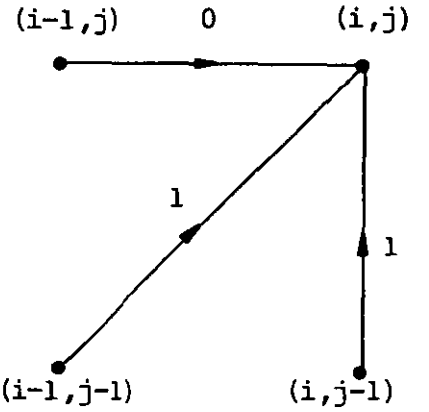
Figures 3.4(a), (b) and (c) give a pictorial illustration of these weights on each production of a warping function with gradient constraint P=0, P=1 and P=2 respectively.

It can be observed that the weighting function type 'a', weighs the productions according to the distance moved along the i axis; type 'b' according to the distance moved along the j axis; type 'c' according to the sum of the distances moved in both i and j directions. Type 'd' weighs all the productions equally. For types 'a' and 'b' weighting functions, the zero weights on some productions may result in the exclusion of some features in the speech pattern from the comparison process. Since the effect of these weighting functions is to map the time axis of one pattern onto the time axis of the other, the time normalization is referred to as of asymmetrical form. Type 'c' weighting function ensures that all the frames of both speech patterns are used in the comparison process, and is referred to as symmetrical form of time normalization. Symmetrical time normalization can be seen as a process whereby the time axis of both speech patterns are mapped onto a temporarily defined common axis.

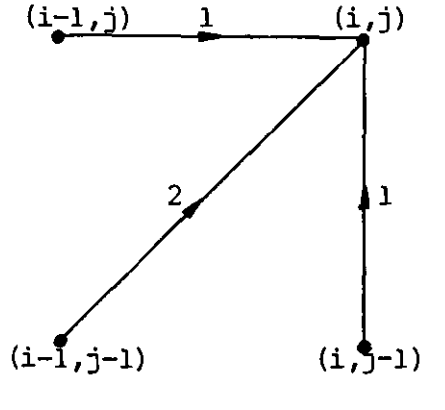
The normalization coefficient N, is determined by the requirement that the total distance, D(A,B), should be the average local distance along the warping path and is expected to be independent of both the



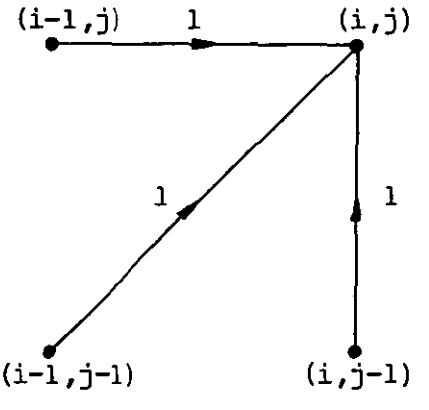
Type 'a' constraint



Type 'b' constraint

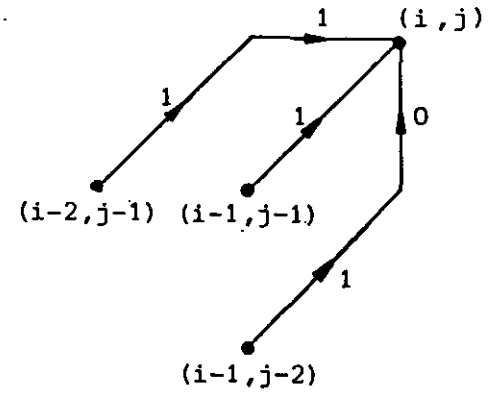


Type 'c' constraint

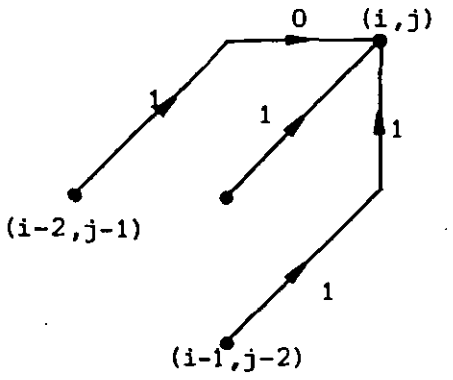


Type 'd' constraint

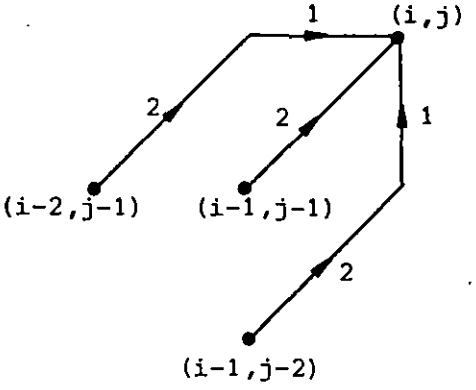
FIGURE 3.4(a): WEIGHTING FUNCTIONS FOR A WARPING PATH WITH GRADIENT CONSTRAINT P=0



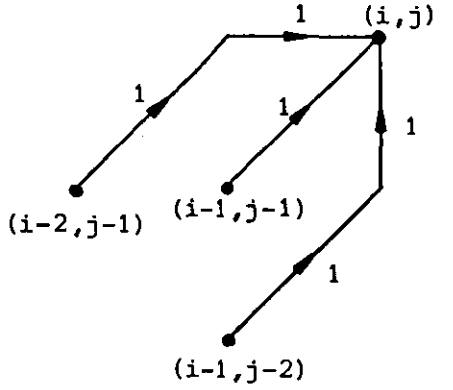
Type 'a' constraint



Type 'b' constraint

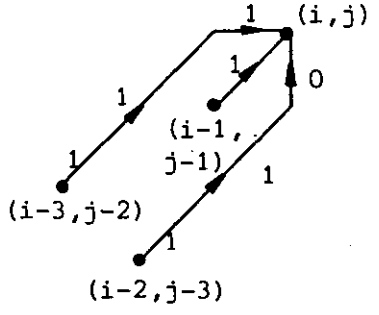


Type 'c' constraint

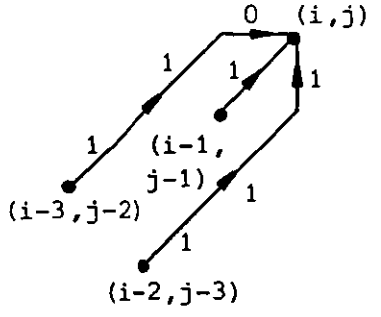


Type 'd' constraint

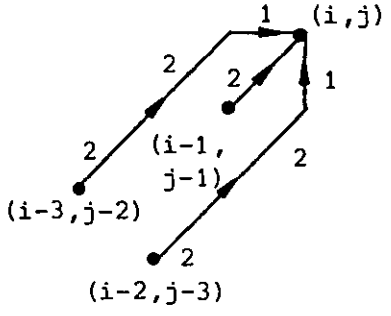
FIGURE 3.4(b): WEIGHTING FUNCTION FOR A WARPING PATH WITH GRADIENT CONSTRAINT P=1



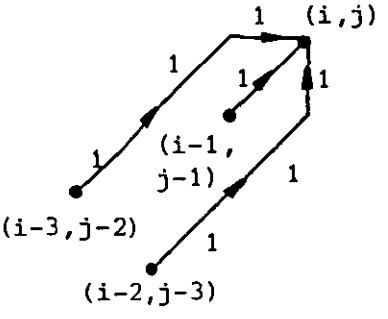
Type 'a' constraint



Type 'b' constraint



Type 'c' constraint



Type 'd' constraint

FIGURE 3.4(c): WEIGHTING FUNCTIONS FOR A WARPING PATH WITH GRADIENT CONSTRAINT P=2

path and the temporal length of the two patterns and is of the form:

$$N = \sum_{k=1}^{K} w(k) \qquad 3.13a$$

For the type 'a' weighting function, the normalized constant is reduced to:

$$N = \sum_{k=1}^{K} w(k) = \sum_{k=1}^{K} [i(k)-i(k-1)] = I \qquad 3.13b$$

while N, for type 'b' weighting function, becomes:

$$N = \sum_{k=1}^{K} [j(k) - j(k-1)] = J \qquad 3.13c$$

and for type 'c':

$$N = \sum_{k=1}^{K} [i(k)-i(k-1) + j(k)-j(k-1)] = I+J \qquad 3.13d$$

However, the type 'd' weighting function is dependent on the time alignment path. This dependence is illustrated in the example of Figure 3.5, which shows two possible time alignment paths for speech patterns A and B. For simplicity, it is assumed that the two patterns are of equal length L. Path 1 is the straight line joining the initial and the final end points, and path 2 is on the edge of the acceptable region in which the warping functions should lie for an adjustment window length r. For example, using values of L=7, and r=2, the time normalization constants for the path are:

path 1: N = 6
path 2: N = 8

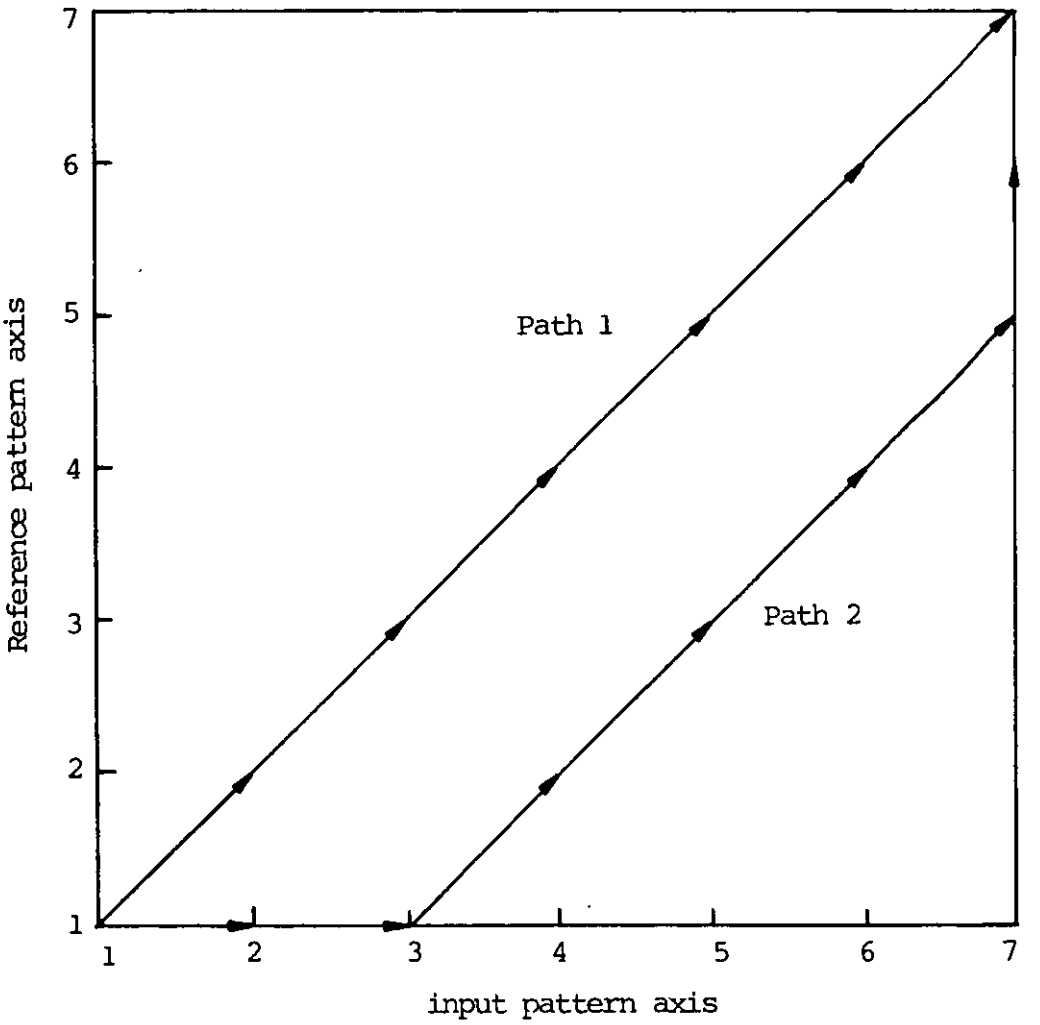


FIGURE 3.5: TWO POSSIBLE PATHS THROUGH A GRID

The dependence of type 'd' normalization constant on the warping path, is a serious disadvantage. This is because the computation of the time normalized distance, D(A,B), is done by a dynamic programming method which relies on a local minimization process to arrive at a final solution iteratively and the optimum path can only be determined at the end of the computation. Thus a normalization constant dependent on the path is clearly unsuitable. The alternative is to choose an arbitrary value for the normalization constant, say L, at the expense of a bias in the DTW process for certain paths over others. The bias may result in a non-optimal path, thus affecting the performance of the warping process.

iii) **The DP matching algorithm**

Dynamic programming (DP) methods can be effectively applied to compute the time normalized distance, D(A,B), between the two speech patterns A and B which are defined as in equations 3.3a and 3.3b. The procedure is recursively implemented by defining the minimum accumulated distance at a given point on the warping function as the sum of the accumulated distance at the preceding point plus the weighted local distance. Starting from the initial grid point (1,1) up to the final grid point (I,J), the final accumulated distance on the warping path can be found using a DP method.

Step 1:
Initial condition

$$g_1\,(f(1)) = d(f(1)).w(1) \qquad 3.15$$

where $g_1(f(1))$ is the minimum accumulated distance at f(1).

Step 2:
The DP equation is:

$$g_k\,(f(k)) = \text{MIN}\,\{g_{k-1}\,(f(k-1)) + d(f(k)).w(k)\} \qquad 3.16$$

Step 3:
Do step 2 for k = 2, 3, ...K. Restrict the warping function to the region $j-r \leqslant i \leqslant j+r$.

Step 4:
The time normalized distance, D(A,B), is given by:

$$D(A,B) = (1/N)\, g_K(f(K)) \qquad 3.17$$

where N is the normalization constant.

The flowchart in Figure 3.6 illustrates the computations in the above algorithm.

By imposing the restrictions on the warping function described in Section 3.2.1(i) and substituting equation 3.12 for the weighting coefficient w(k) in the DP equation in step 3, several practical algorithms can be realised. For example, for the asymmetrical form with no slope constraint (P=0) as illustrated in Figure 3.3a, the DP equation reduces to:

$$g(i,j) = \text{MIN} \begin{cases} g(i,j-1) + d(i,j) \cdot 0 \\ g(i-1,j-1) + d(i,j) \cdot 1 \\ g(i-1,j) + d(i,j) \cdot 1 \end{cases} \qquad 3.18$$

Table 3.1 contains several DP equations for both symmetrical and asymmetrical forms of time normalization for various warping function gradient constraints.

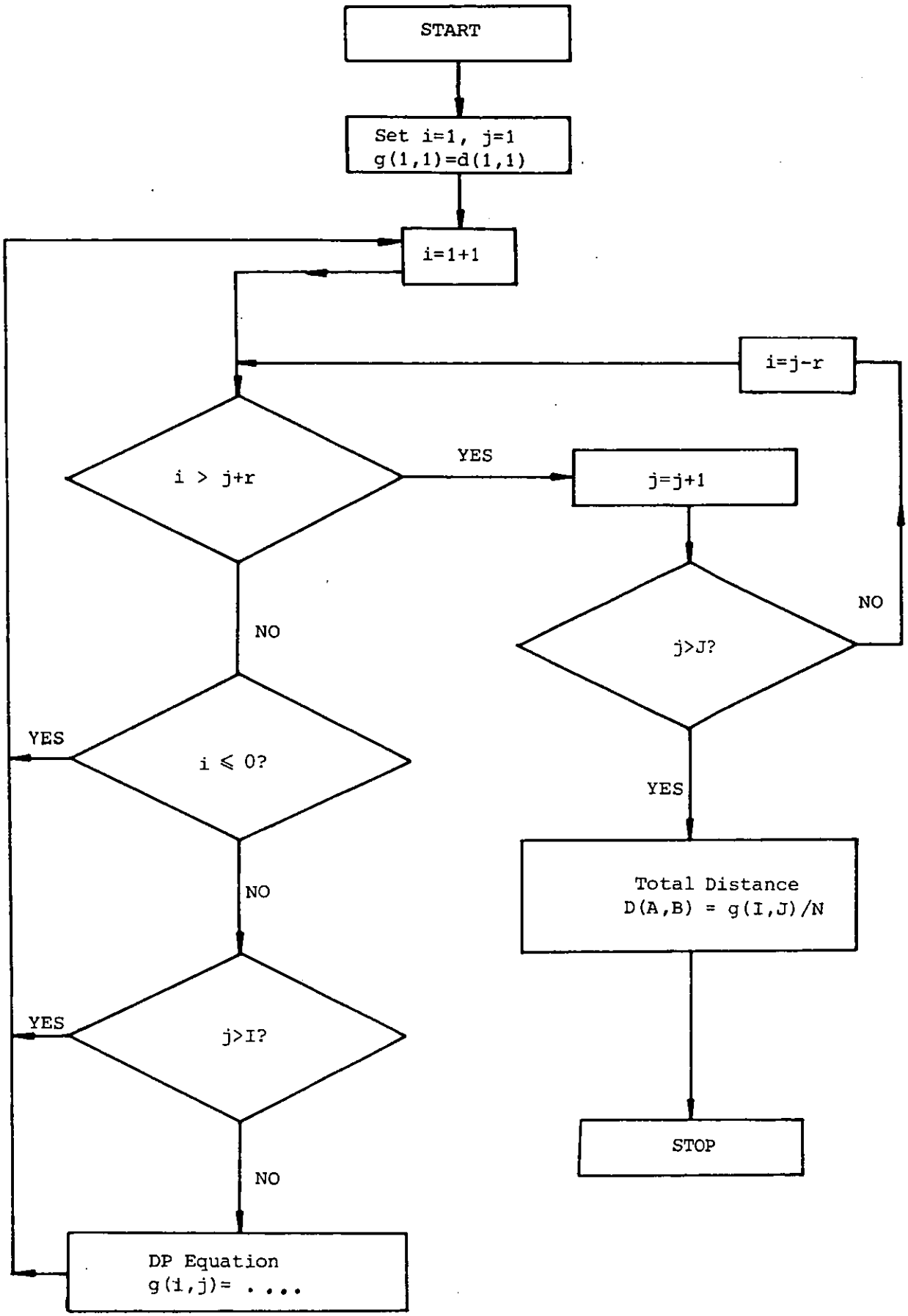


FIGURE 3.6: THE DP MATCHING ALGORITHM FLOWCHART

TABLE 3.1

SAKOE AND CHIBA'S DTW ALGORITHMS [43]

Gradient Constant P	Symmetrical/ Asymmetrical	DP Equation g(i,j) = ...
0	Symmetrical	$\text{MIN}\begin{cases} g(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-1,j)+d(i,j) \end{cases}$
0	Asymmetrical (Type a constraint)	$\text{MIN}\begin{cases} g(i,j-1) \\ g(i-1,j-1)+d(i,j) \\ g(i-1,j)+d(i,j) \end{cases}$
1	Symmetrical	$\text{MIN}\begin{cases} g(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-2,j-1)+2d(i-1,j)+d(i,j) \end{cases}$
1	Asymmetrical (Type a constraint)	$\text{MIN}\begin{cases} g(i-1,j-2)+d(i,j-1) \\ g(i-1,j-1)+d(i,j) \\ g(i-2,j-1)+d(i-1,j)+d(i,j) \end{cases}$
2	Symmetrical	$\text{MIN}\begin{cases} g(i-2,j-3)+2d(i-1,j-2)+2d(i,j-1)+d(i,j) \\ g(i-1,j-1)+2d(i,j) \\ g(i-3,j-2)+2d(i-2,j-1)+2d(i-1,j)+d(i,j) \end{cases}$
2	Asymmetrical (Type a constraint)	$\text{MIN}\begin{cases} g(i-2,j-3)+d(i-1,j-2)+d(i,j-1) \\ g(i-1,j-1)+d(i,j) \\ g(i-3,j-2)+d(i-2,j-1)+d(i-1,j)+d(i,j) \end{cases}$

3.2.2 The Itakura DTW Algorithm [26]

The Itakura DTW algorithm is realised by imposing different restrictions on the warping path gradient from those described in the Sakoe and Chiba's algorithms. The monotonic, continuity and boundary conditions of the warping function remain the same. Figure 3.7a illustrates the relationship between adjacent points on the warping function. The accumulated distance g(i,j) at the grid point (i,j) is the sum of the local distance between the ith and the jth frames of the input and the reference pattern respectively, and the minimum accumulated distance to the grid point (i-1,q), i.e.

$$g(i,j) = d(a_i,b_j) + \underset{j-2\leqslant q\leqslant j}{MIN} \{g(i-1,q)\} \qquad 3.19$$

The path to the grid point (i,j) can only originate from the three points: (i-1,j), (i-1,j-1) and (i-1,j-2). A further constraint on the warping path is that two successive horizontal productions are not allowed. Thus equation 3.19 takes the form:

$$g(i,j) = d(i,j) + MIN \begin{cases} g(i-1,j) & .W(i-1,j) \\ g(i-1,j-1) & \\ g(i-1,j-2) & \end{cases} \qquad 3.20a$$

where $W(i-1,j) = \infty$ if $f(i-1) = f(i-2)$

$= 1$ otherwise

Dynamic Programming is used to compute equation 3.20a starting from the grid point (1,1) to the point (I,J) to give the final solution:

$$D = g(I,J)/N \qquad 3.20b$$

where N is the number of points on the warping function.

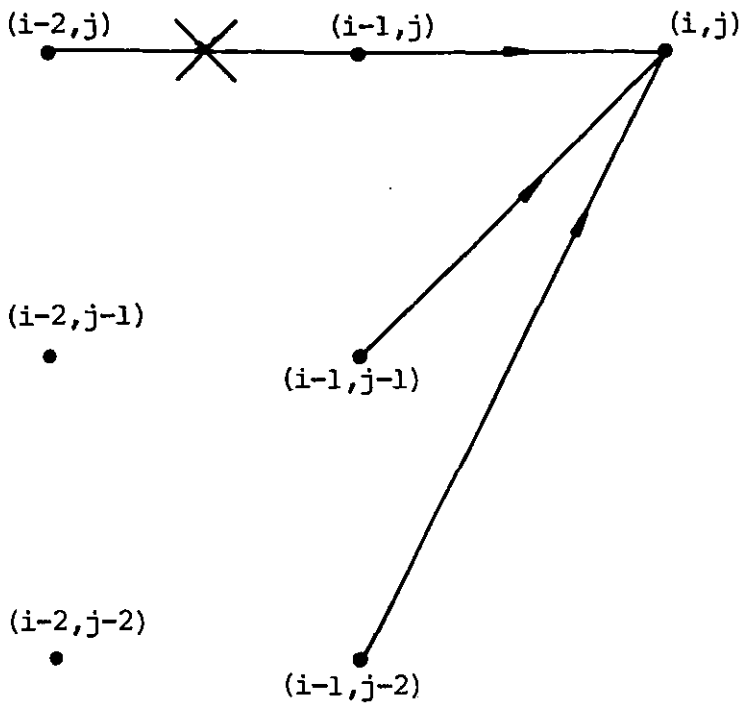


FIGURE 3.7(a): POSSIBLE PRODUCTIONS TO THE GRID POINT (i,j)

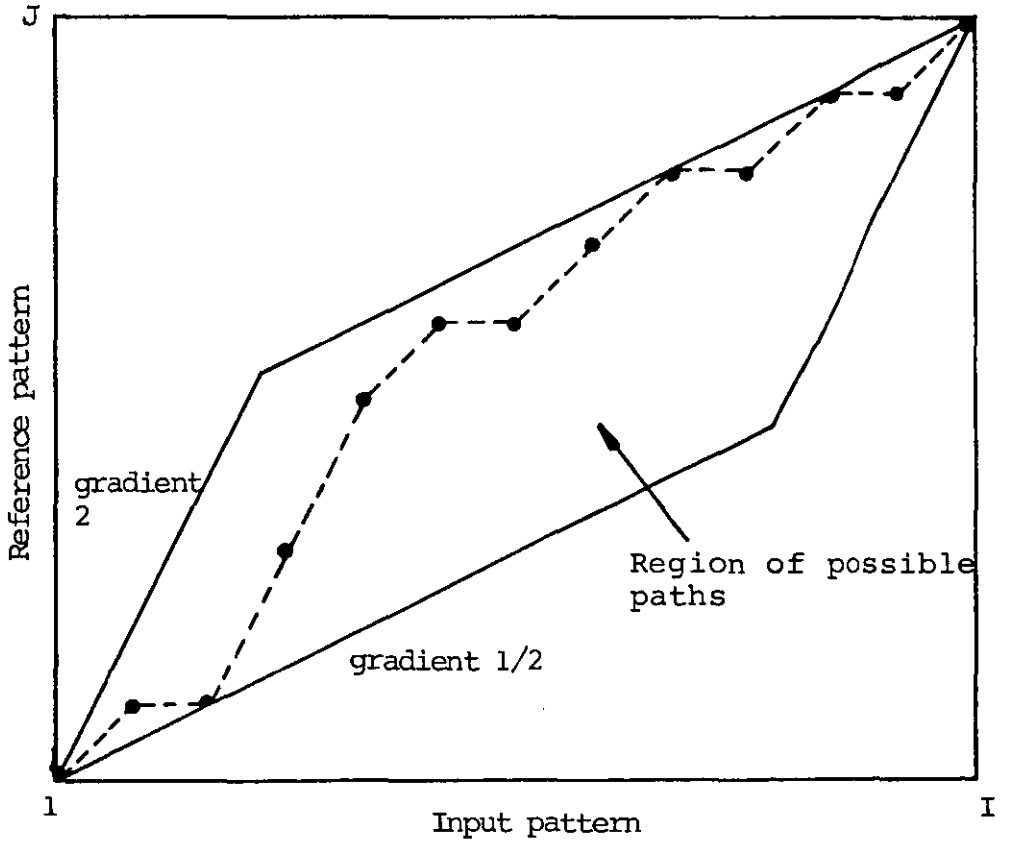


FIGURE 3.7(b) REGION OF POSSIBLE WARPING PATH FOR THE ITAKURA DTW ALGORITHM

The local constraints imposed on the warping function plus the end points boundary conditions result in a search area in the i-j plane in which the optimal path lies as illustrated in Figure 3.7b. The region is a parallelogram whose two extreme corners are the grid points (1,1) and (I,J) with lines of gradient 1/2 and 2. The parallelogram lines with gradient 1/2 arises from the condition prohibiting two successive horizontal productions, so that no path to the grid point (i,j) can originate from a grid point higher than (i-2,j-1). The lines with gradient of 2, are likewise determined by the condition that the path to the grid point (i,j) cannot originate from a lower grid point than (i-1,j-2).

3.2.3 Results

The performances of Sakoe/Chiba and Itakura DTW algorithms were evaluated in a pattern matching based isolated word recognition system. The speech patterns used were discrete sequences of 14th order LPC feature vectors, extracted from Hamming window weighted speech data in 25.6 msec segments. Two sets of 5 kHz bandlimited speech, consisting of a group of acoustically similar words (confusion set), and a group of dissimilar words (dissimilar set) were used in the experiments. The local distance for comparing the input and the reference pattern frames is the distortion measure proposed by Itakura [26] and will be discussed further in Chapter 5 of this thesis. The input pattern to the recognizer is matched with all the reference patterns and is recognized as the reference word with the smallest distance D(A,B). Details of the input and reference patterns are given below:

a) **Confusion set:**

The confusion set of words was composed of the acoustically similar vocabulary words: G, B, C, D, E, V, P, T, spoken by the three male subjects SM1, SM2, SM3 and the two female subjects SF1, SF2. Each male speaker spoke each of the eight words twice, and the female

speakers only once. With the speech patterns of SF1 as the references, the DTW algorithms were tested with the speech utterances of SM1, SM2, SM3 and SF2 as the input. The test procedure was repeated with the speech patterns of SM1 as reference and the input patterns from SM1, SM3, SF1 and SF2. The percentage of correct identifications of the input words is defined as the recognition rate.

b) **Dissimilar set:**

The dissimilar set of words was composed of the vocabulary words: NINE, THREE, WRITE, CONTROL, STORE, FIVE, YES, SEVEN, which all appear to be acoustically different. These words were spoken by the three male subjects SM1, SM2, SM3, and the female subjects SF1, SF2. As with the confusion set, the three male speakers uttered the vocabulary words twice and the two female speakers once. The experimental procedure described above with the confusion set, was also repeated using the dissimilar set.

The recognition test results using the Sakoe and Chiba DTW algorithms are shown in Table 3.2. Of interest here is the comparison in performance of different Sakoe/Chiba DTW algorithms. Therefore, the actual value of the adjustment window length, r, used in the algorithms is not important as long as it is fixed for a given speech pattern pair. In the experiment, the value of r was fixed to four frames (i.e. 102.4 msec), otherwise r was set to the value of the absolute difference in frames between the input and the reference patterns. The recognition test results for the Itakura DTW algorithms are also shown in Table 3.2, for both the confusion and the dissimilar vocabulary sets.

3.3 MODIFIED DTW ALGORITHMS

3.3.1 The Endpoints Adjustment

The assumptions in the DTW algorithms discussed in the preceding sections that the input and the reference speech patterns are in

TABLE 3.2

RECOGNITION TEST RESULTS USING (i) SAKOE AND CHIBA'S
(ii) ITAKURA'S DTW ALGORITHMS

Type of DTW Algorithm	Recognition Rate (%)	
	Test Vocabulary	
	Confusion Set	Dissimilar Set
1. Sakoe & Chiba P=0 Asymmetrical $r \geqslant 4$	42.9	78.6
2. Sakoe & Chiba P=0 Symmetrical $r \geqslant 4$	58.9	82.2
3. Sakoe & Chiba P=1 Asymmetrical $r \geqslant 4$	60.7	78.6
4. Sakoe & Chiba P=1 Symmetrical $r \geqslant 4$	66.1	80.4
5. Sakoe & Chiba P=2 Asymmetrical $r \geqslant 4$	53.6	73.2
6. Sakoe & Chiba P=2 Symmetrical $r \geqslant 4$	53.6	71.4
7. Itakura	53.6	78.6

complete time synchronism at the initial and final frames, can only be justified where accurate determination of the beginning and the endpoints of both patterns can be made. Usually, the detection of the end frames of an utterance is a difficult task because of their similarity with silence frames. At the beginning of an utterance, breathing noise is usually present and it is easily confused with speech. Weak fricative sounds at the beginning or ending of an utterance, or vowel tails which appear at the end of an utterance, can sometimes be identified incorrectly as silence. Usually in a practical speech recognition system, the endpoints of the reference speech pattern are accurately determined manually and stored in memory. Thus it is only for the input utterance that endpoints have to be determined during the recognition process. If accurate detection of the endpoints of an utterance cannot be made, then the performance of the DTW process is degraded. An alternative approach is to relax the boundary conditions of the time warping algorithm as proposed by Rabiner et al [44]. Their method is based on retaining the restrictions on the warping path described in Section 3.2.1(i), with the exception of the boundary conditions which are modified as follows:

$$1 \leqslant i(1) \leqslant 1+\delta \qquad 3.21a$$

and

$$I-\delta \leqslant i(K) \leqslant I \qquad 3.21b$$

where δ is a positive constant representing the number of frames within which the endpoint is to be found. The non-zero value of δ increases the area of the allowed search region in the i-j plane in which the optimum path can lie. A value of $\delta = 1$ has been suggested as generally suitable [44].

3.3.2 Paliwal's Modification over the Sakoe and Chiba DTW Algorithm [45]

In the Sakoe and Chiba's DTW algorithm, a given frame in the input speech pattern is compared with a limited number of frames in the reference speech pattern, in order to remove the possibility of excessive time differences not normally present in speech. The allowable number of frames mismatch, r, is defined as in equation 3.10, i.e:

$$|i-j| \leqslant r \qquad 3.22$$

and is the width of the adjustment window in the warping path. This means that the grid endpoint (I,J) is outside the region in which the optimal path can be found if the absolute time difference, $|J-I|$, between the two patterns is larger than r. Thus the algorithm is limited to patterns whose temporal differences are less than a certain value of r. This limitation is undesirable in word recognition systems because many of the speech patterns can be of diverse temporal lengths, and an accurate distance measure is still required between patterns of large temporal differences.

Paliwal et al [45] have proposed some modification on the Sakoe and Chiba algorithm in order to enable the comparison of patterns of any temporal difference. The modified algorithm uses the same adjustment window length and restricts the warping path to the region in the i-j plane bounded by two lines parallel to the diagonal line joining the initial grid point (1,1) to the final grid point (I,J). The adjustment window is given by:

$$|i - (j/s)| \leqslant r \qquad 3.23$$

where s = J/I is the gradient of the diagonal line.

The adjustment window condition limits the warping path in the region bounded by lines j=si+r and j=si-r and ensures the inclusion of the endpoint (I,J) in the distance computation. Figure 3.8 is an illustration of the warping path region for both the Sakoe and Chiba's algorithms and the Paliwal's algorithm when $|I-J| > r$.

3.3.3 Myers' Algorithm [46]

The performance of the Itakura's DTW algorithm, like the Sakoe and Chiba's algorithms, becomes inadequate when large temporal differences exist between the speech patterns to be compared. Myers et al [46] have examined the effects of the various ratios of the input to reference pattern lengths on the Itakura's DTW algorithm.

Figure 3.9 illustrates the relationship between the search area in which the optimum path can be found and the ratio I/J of the input to reference pattern length. The search area is maximum when the input and the reference patterns are of equal length. The area shrinks considerably when the input/reference pattern ratio is 2/3 and at a ratio of 1/2 only a single path, which is the straight line joining the grid points (1,1) and (I,J), is valid. Such a situation is merely a linear expansion of the reference axis and does not exploit any of the advantages offered by DTW.

The larger the search area, the less the restriction on the warping path resulting in many paths among which the best can be selected. Thus it would be reasonable to expect the DTW algorithm to give better performance for patterns of equal length since the search area will be maximum. Myers has proposed a scheme whereby both reference and input patterns are reduced to a standard length before applying the DTW algorithm. Thus the input speech pattern A, given as:

$$A = \{a_1, a_2, \ldots, a_i, \ldots, a_I\}$$

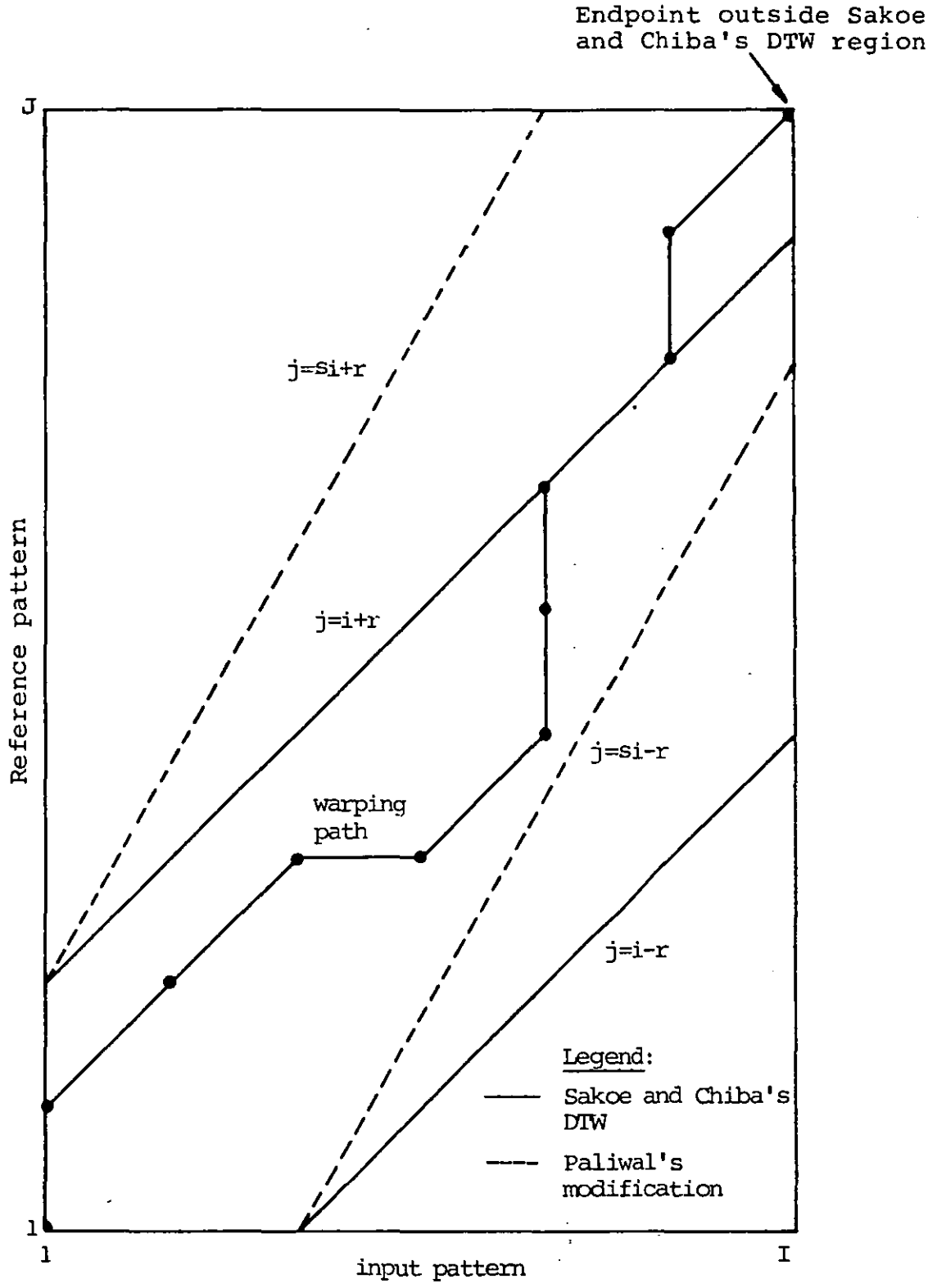
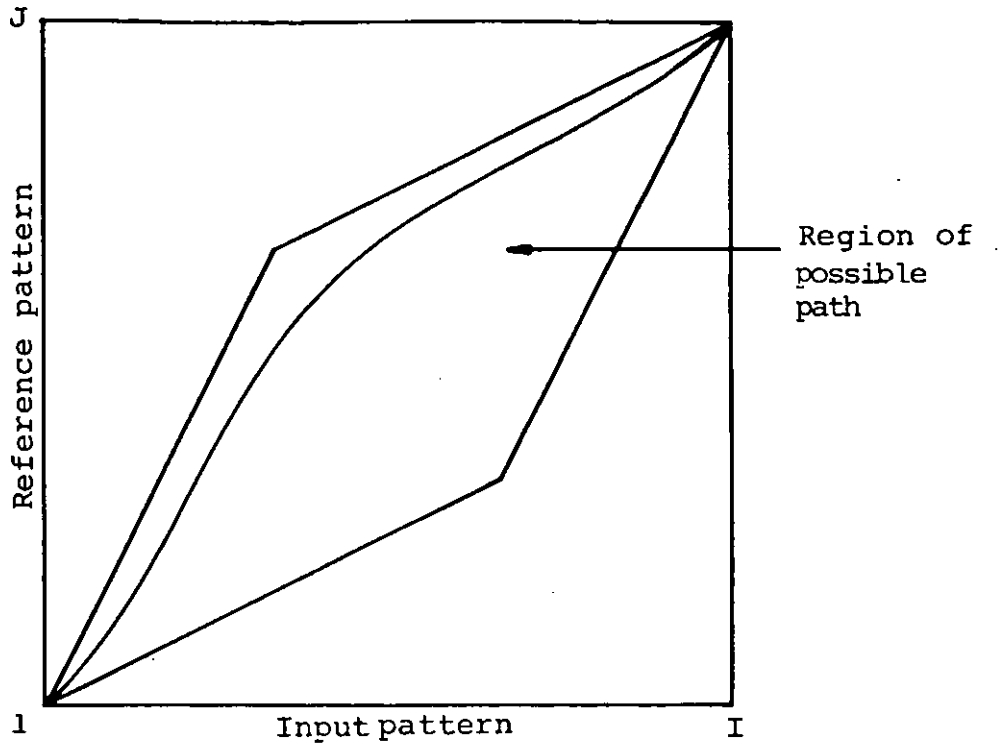
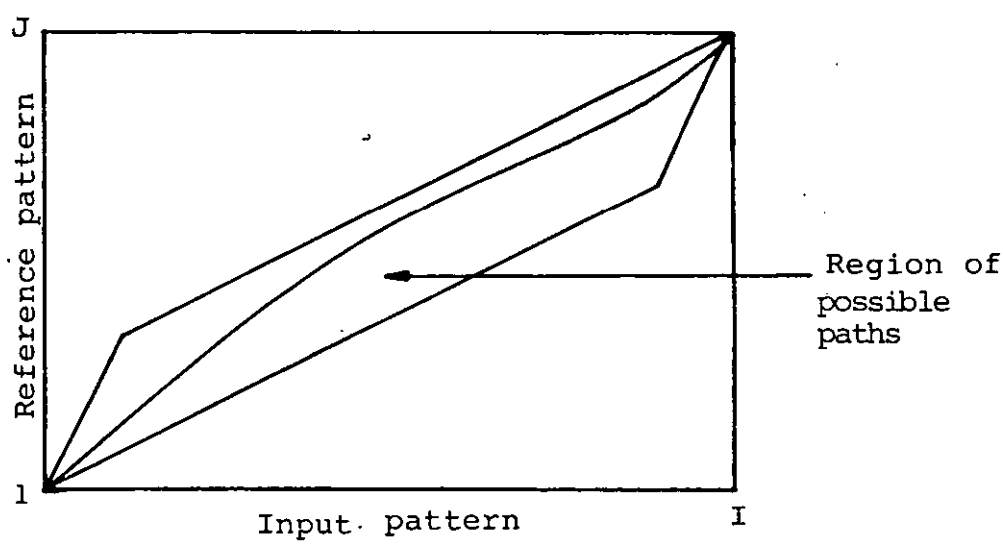


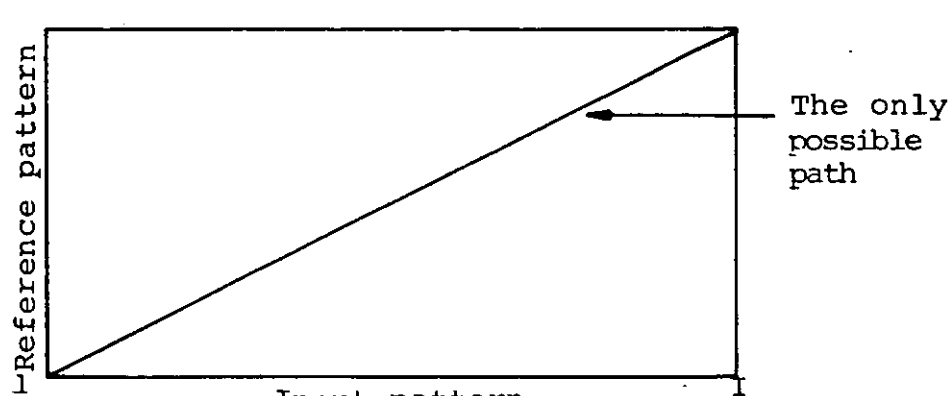
FIGURE 3.8: AN ILLUSTRATION OF THE WARPING PATH REGION FOR BOTH THE SAKOE AND CHIBA'S DTW ALGORITHM, AND THE PALIWAL'S MODIFICATION WHEN $|I-J| > r$



(a) Region of possible paths for I=J



(b) Region of possible paths for J=2I/3



(c) Region of possible paths for J=I/2

FIGURE 3.9: THE EFFECTS OF THE TEST REFERENCE PATTERN LENGTH RATIO ON THE RANGE OF THE WARPING PATH USING ITAKURA'S DTW ALGORITHM

is transformed into a pattern A of length I' such that:

$$A = \{\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{i'}, \ldots, \bar{a}_{I'}\}$$

where the feature vectors $\bar{a}_{i'}$ are related to the original vectors a_i as follows:

$$\bar{a}_{i'} = (1-M)\, a_i + M\, a_{i+1}, \qquad i' = 1,2,\ldots, I' \qquad 3.24$$

where

$$i \text{ is the largest integer} < [(i'-1)(I-1)/(I'-1)] + 1 \qquad 3.25a$$

and

$$M = [(i'-1)(I-1)/(I'-1)] + 1-i \qquad 3.25b$$

for example, a sequence $A = \{a_1, a_2, \ldots, a_8\}$, with 8 frames, is transformed into a sequence A' with 6 frames given by:

$$A' = \{\bar{a}_1 = a_1,\ \bar{a}_2 = 0.6a_2 + 0.4a_3,\ \bar{a}_3 = 0.2a_3 + 0.8a_4,$$

$$\bar{a}_4 = 0.8a_5 + 0.2a_6,\ \bar{a}_5 = 0.4a_6 + 0.6a_7,\ \bar{a}_6 = a_8\}$$

Similarly the reference pattern $B = \{b_1, b_2, \ldots, b_j, \ldots, b_J\}$ is transformed into a pattern of the same length I' as the input pattern. Following the pattern length normalization, the DTW algorithm is then applied.

3.3.4 Results

The modifications provided by the Paliwal and by Myers' approach on the DTW algorithms of Sakoe and Chiba, and Myers' approach on the Itakura DTW algorithm, were evaluated by their performance in an LPC-based pattern matching isolated word recognition system. Both the confusion words set (B, C, D, E, G, P, T, V), and the dissimilar words set (NINE, THREE, WRITE, CONTROL, STORE, FIVE, YES, SEVEN) were used in the experiments.

i) Variation of the recognition rate with the adjustment window length

The Sakoe and Chiba DTW algorithm, with the warping path gradient constraint P=1, and using the symmetrical form of matching, was found in Section 3.2.3 to give a higher performance than the other algorithms. The variation of the recognition rate, with the adjustment window length employed in the algorithm was investigated using the confusion set data. The results obtained are shown in Figure 3.10. The variation obtained on incorporating Paliwal and also Myers' approach are also shown in Figure 3.10.

The results obtained on repeating the experiments with the dissimilar data set are shown in Figure 3.11.

ii) Paliwal's modification as applied to the Sakoe and Chiba's DTW algorithm

The modified algorithm used a fixed window length, r, of four frames and employed the endpoint relaxation method discussed in Section 3.3.1 with δ=1. The recognition test results obtained for different constraints are tabulated in Table 3.3.

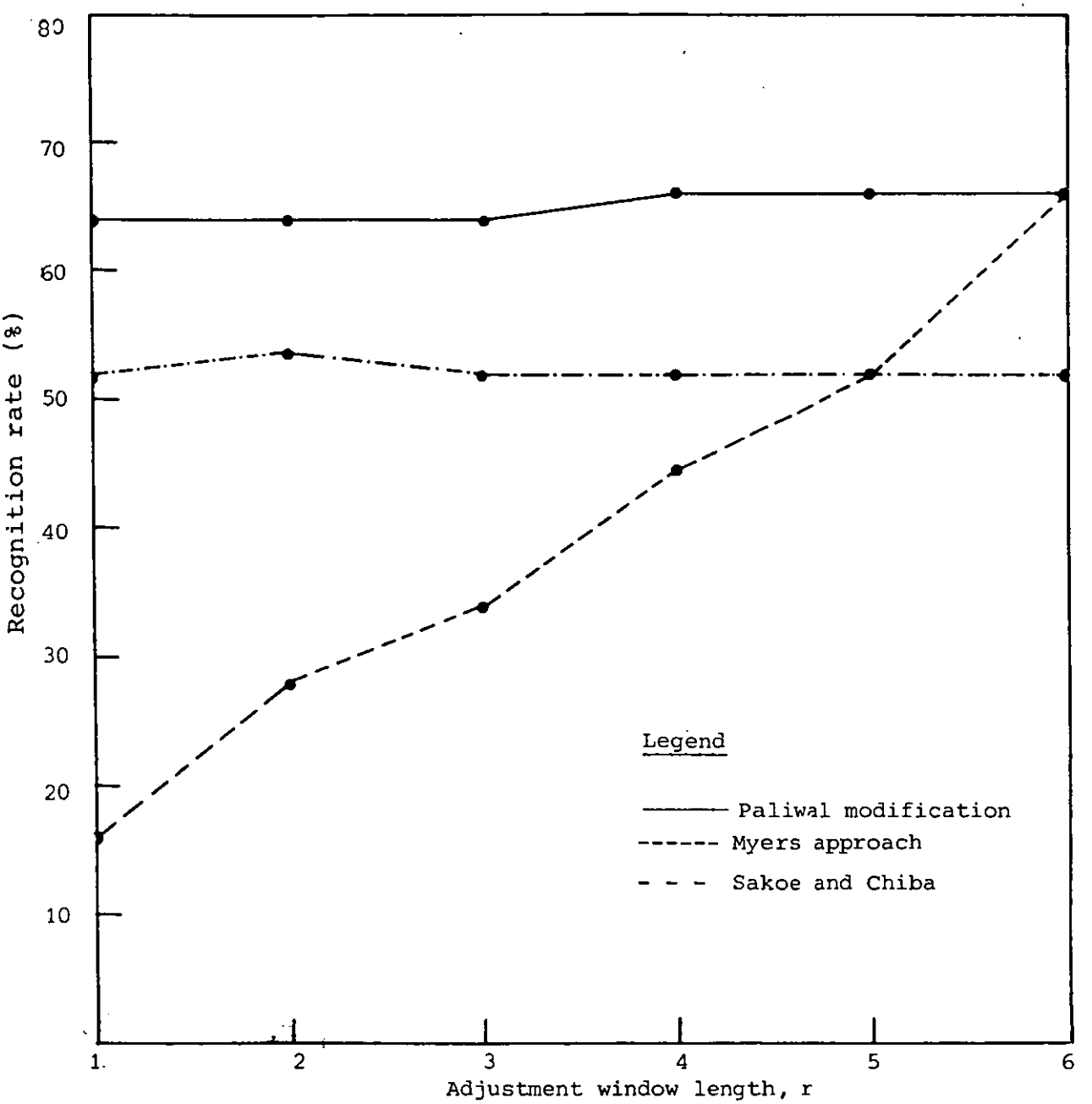


FIGURE 3.10: VARIATION OF THE RECOGNITION RATE WITH THE ADJUSTMENT WINDOW LENGTH FOR A SET OF ACOUSTICALLY SIMILAR WORDS

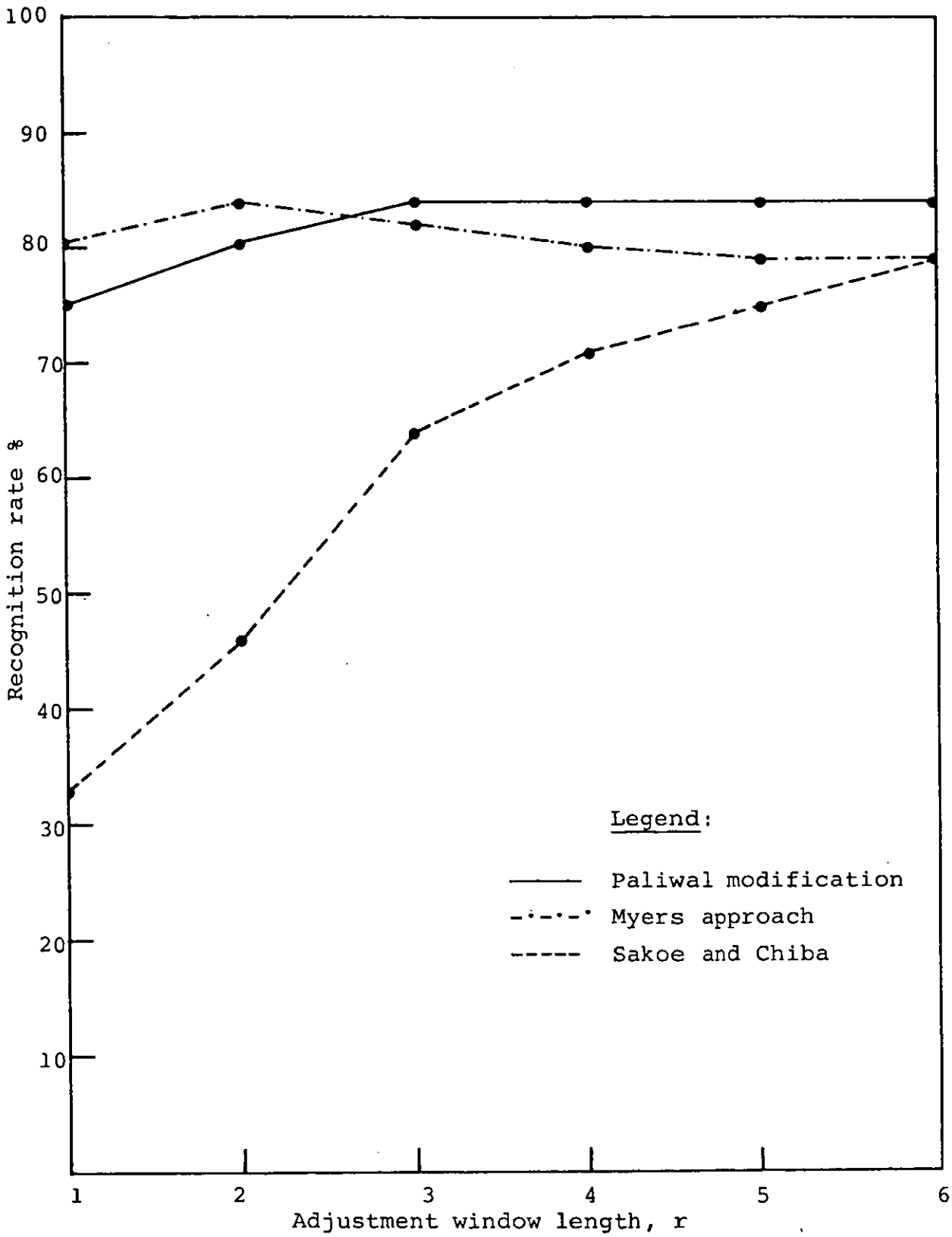


FIGURE 3.11: VARIATION OF THE RECOGNITION RATE WITH THE ADJUSTMENT WINDOW LENGTH FOR A SET OF ACOUSTICALLY DISSIMILAR WORDS

TABLE 3.3

RECOGNITION TEST RESULTS OBTAINED ON USING THE PALIWAL'S MODIFICATION OVER THE SAKOE AND CHIBA'S DTW ALGORITHM

Sakoe & Chiba's DTW algorithms constraints	Recognition Rate (%)	
	Test Vocabulary	
	Confusion Set	Dissimilar Set
1. P=0 Asymmetrical r=4	53.6	76.8
2. P=0 Symmetrical r=4	58.9	80.4
3. P=1 Asymmetrical r=4	60.1	80.4
4. P=1 Symmetrical r=4	66.1	83.9
5. P=2 Asymmetrical r=4	53.6	69.6
6. P=2 Symmetrical r=4	44.6	66.1

iii) **Myers' approach as applied to the Sakoe and Chiba's algorithm**

Using Myers' algorithm, all the speech patterns were normalized to a fixed length which is the average length of the patterns within the recognition vocabulary. For the confusion set the normalized length was 15 frames and 20 frames for the dissimilar set. Since the speech patterns were already normalized to a fixed length, a smaller window length r=2, was employed in the Sakoe and Chiba's algorithms. Endpoint relaxation for the speech patterns was employed with δ=1. The recognition test results are shown in Table 3.4.

iv) **Myers' approach as applied to the Itakura's DTW algorithm**

The speech patterns were processed as in (iii) above, using the Myers' algorithm and the same endpoint relaxation figure of δ=1. The recognition results are also tabulated in Table 3.4.

3.4 THE ORDERED GRAPH SEARCH TECHNIQUE

The DTW methods discussed above have been found to give reliable time alignment between input and reference speech patterns, but have the disadvantage of involving heavy computation. As a consequence, several alternative procedures for reducing the computation have been proposed [44][47] but are mainly based on imposing more tight restrictions on the warping path constraints at the expense of losing optimality. Brown and Rabiner [48] have proposed a novel procedure based on the search algorithms described by Nilsson [49]. In their method the DTW process is modelled as an ordered graph search (OGS) through a constrained grid, in order to find the path with minimum cost as discussed below.

3.4.1 Path Cost Estimation

Consider the graph in Figure 3.1. Let each grid point in the search area be termed a node since the optimal path may pass through it. A node is designated by its coordinates in the i-j plane. Thus the

TABLE 3.4

RECOGNITION RESULTS USING MYERS' PATTERN LENGTH NORMALIZATION ALGORITHM WITH (i) SAKOE & CHIBA'S DTW ALGORITHM (ii) ITAKURA'S DTW ALGORITHM

Type of DTW Algorithm	Recognition Rate (%)	
	Test Vocabulary	
	Confusion Set	Dissimilar Set
1. Sakoe & Chiba P=0 Asymmetrical r=2	50.0	80.4
2. Sakoe & Chiba P=0 Symmetrical r=2	50.0	82.1
3. Sakoe & Chiba P=1 Asymmetrical r=2	48.2	83.9
4. Sakoe & Chiba P=1 Symmetrical r=2	53.6	83.9
5. Sakoe & Chiba P=2 Asymmetrical r=2	50.0	85.7
6. Sakoe & Chiba P=2 Symmetrical r=2	53.6	82.2
7. Itakura	58.9	78.6

starting node s=(1,1) and the ending node e = (I,J). For any path in the grid passing through the node n, the path cost q(n) in terms of total accumulated distances at the node is given by:

$$q(n) = g(n) + h(n) \qquad 3.26$$

where g(n) is the path cost from the starting node s to n and h(n) is the minimum cost of the path from node n to the end node e. Since the search starts from node s, towards node e, the minimum cost g(n) is known exactly, but the cost h(n) can only be estimated. Thus an estimate of the minimum path cost $\bar{q}(n)$ at n for the path is given as:

$$\bar{q}(n) = g(n) + \bar{h}(n) \qquad 3.27$$

where $\bar{h}(n)$ is an estimate of h(n).

In the OGS algorithm, it is required to satisfy the condition $\bar{q}(n) \leq q(n)$. Thus $\bar{h}(n)$ must underestimate the true path cost h(n), from node n to the terminal node e, i.e.

$$\bar{h}(n) \leq h(n) = \sum_{k=i+1}^{I} (D(A(k), B(f(k))) \qquad 3.28$$

where n = (i,j) and e = (I,J). A and B are the two speech patterns under consideration. The true path cost, from node n to the terminal node, is the sum of the local distances along the path. Considering the asymmetrical form of time warping where the number of grid points along the path is (I-i), $\bar{h}(n)$ can be bound above by:

$$\bar{h}(n) = (I-i).\bar{d} \qquad 3.29$$

where $\bar{d}$ is a constant, small enough to ensure that $\bar{h}(n) < h(n)$. For acoustically different speech patterns, equation 3.29 has been found to be a gross underestimate resulting in a large number of nodes to be searched. In order to overcome these difficulties, an adaptive estimator has been proposed [48], and is of the form:

$$\bar{h}(n) = (I-i).\ g(n)/i \qquad 3.30$$

For similar sounding words, g(n)/i, tends to be small, thus giving an estimate of h(n) in the correct range. For patterns of acoustically dissimilar words, the overall effect is to increase $\bar{h}(n)$. Although there is no theoretical guarantee that $\bar{h}(n)$ of equation 3.30 underestimates h(n) in all cases, it has been found to give a suitable estimate in practice [48].

3.4.2 The Search Algorithm

In the standard DTW approach, the computation of all the local distances in the search region is required since all the possible warping paths are searched, subject to the imposed constraints. The gain in the computation efficiency in the graph search method is achieved by considering only the warping paths which appear to be likely candidates for the optimal path and as such, the local distances for some grid points are not required.

A path starting from s to the node n is characterised by the nodal state at n given by:

i) The node coordinates (i,j)
ii) The production from node n-1 to node n
iii) The estimated cost $\bar{h}(n)$
iv) The exact cost g(n).

At the given node n the path is searched by expanding the production for which the estimated cost is minimum, the other productions are terminated into 'open' nodes. These nodes are termed 'open' since if the chosen production is later found to be illegal or suboptimal then they can be re-visited and expanded for the optimal path search. As the path search is carried out, an 'open list' of nodal states of potential paths is maintained. The 'open list' is arranged such that the path with the lowest estimated cost g(n) is on top of the list. The minimum cost path through the grid is found by removing the node on the top of the open list and expanding it to generate all legal successor nodes for which new path costs are estimated. The successor nodes are also sorted into the 'open list' and again the minimum cost node is removed from the list and expanded. If the following conditions are satisfied, then the first path to terminate at node e will be the minimum cost node.

i) The node expansion operation is the same for all nodes
ii) $g(n)$ is monotonic and > 0 for $\forall$ n=s
iii) $\bar{q}(n) \leqslant q(n)$ for $\forall$ n
iv) $\bar{h}(n)$ is monotonic, $h(n) > 0$, $\forall$ n $\neq$ e.

The various steps in the algorithm are as follows:

Step 1: Start with node s = (1,1) on open list
Step 2: Remove node from open list having lowest estimated cost $\bar{q}(n)$ and place in closed list for expansion
Step 3: Generate successor node subject to local constraints
Step 4: If successor node is illegal go to step 2
Step 5: If the successor node already exists in the open list or in closed list go to step 3
Step 6: If successor node does not satisfy search constraints go to step 3
Step 7: Save nodal state
Step 8: If terminal node reached go to step 10
Step 9: Compute the estimated cost $\bar{q}(n)$ and sort the node into the open list. Go to step 3

Step 10: The minimum path cost is obtained from the nodal state as g(I).

The above algorithm is illustrated in Figure 3.12, in which an input pattern, with 7 frames, is compared with a reference pattern with 6 frames. The search begins at node s = (1,1), which is expanded using horizontal, vertical and diagonal productions to grid point (1,2), (2,2), and (2,1), where an estimate of the path cost g(n) from equation 3.27 is made. The grid point (2,2), with the smallest cost g(n) is expanded, the cost at the other grid points being placed in an open list. The grid point (2,2) is expanded to points (3,3), (3,2) and (2,3), and the path costs estimated. On expanding the grid point (2,3), which has the lowest estimated path cost, to grid points (3,3), and (3,4), (the grid point (2,4) is illegal because it falls outside the search region), the path cost estimates show that an earlier grid point (3,2) has the smallest path cost estimate among all the open list nodes. Therefore, the estimated costs at (3,4) and (3,3) are put in the open list and instead the grid point (3,2) is expanded. The process continues until grid point (7,6) is reached.

3.4.3 Results

The OGS technique was compared with the DTW algorithm by performing recognition tests using the confusion set and the dissimilar set of vocabulary words of Sections 3.2.3 and 3.3.4. Only the asymmetrical form of Sakoe and Chiba's algorithms were considered. This is because the path cost estimation in the OGS algorithm, as given in equations 3.29 or 3.30, requires the asymmetrical path constraints. The recognition rate results and the number of distance computations done are given in Table 3.5.

3.5 DISCUSSION

From the results presented in the previous sections, the following points can be deduced:

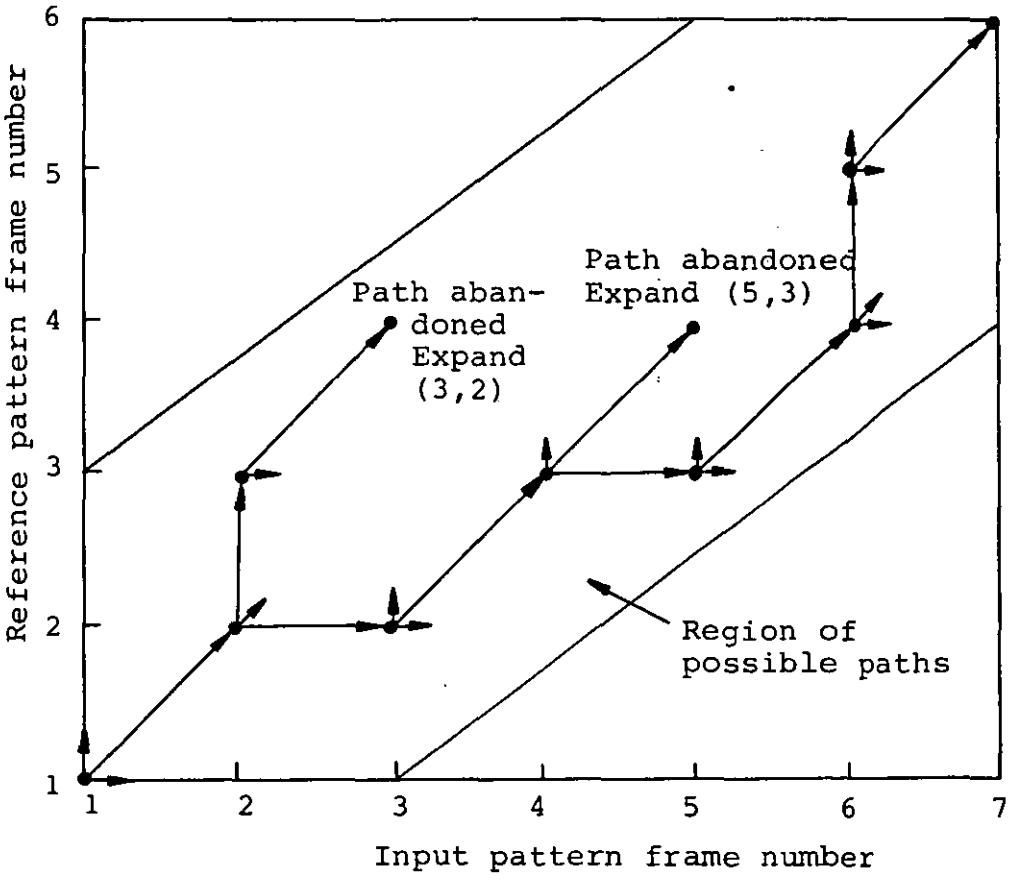


FIGURE 3.12: AN ILLUSTRATION OF THE GRAPH SEARCH TECHNIQUE

TABLE 3.5

A COMPARISON OF THE RECOGNITION RATE OBTAINED WITH THE ASYMMETRICAL FORM OF SAKOE AND CHIBA'S ALGORITHM WHEN DTW AND WHEN OGS TECHNIQUES ARE EMPLOYED

		Recognition Rate (%)		Average number of distance computations per word	
Algorithm	DTW/ OGS	Confusion Set	Dissimilar Set	Confusion Set	Dissimilar Set
Sakoe & Chiba r=4 P=0 Asymmetrical	DTW	53.6	76.8	88	123
	OGS	50.0	76.8	34	48
Sakoe & Chiba r=4 P=1 Asymmetrical	DTW	60.1	80.4	88	123
	OGS	60.1	78.6	31	44
Sakoe & Chiba r=4 P=2 Asymmetrical	DTW	53.6	69.6	88	123
	OGS	53.6	66.1	26	45

i) The recognition rate performance of the Sakoe and Chiba's DTW algorithm, tabulated in Table 3.2, is seen to be heavily dependent on the warping path gradient constraint P. The symmetrical form of matching tends to give a better performance than the asymmetrical form of matching, for a given gradient constraint P. This would be expected since in the asymmetrical form of matching, the possibility of some of the frames in the speech patterns being excluded in the distance computation may arise. The symmetrical algorithm with the constraint P=1, gives better recognition rate than the other types of warping algorithms under consideration, including the Itakura DTW algorithm. These results are in agreement with the findings of Sakoe and Chiba [43].

ii) The variation of the recognition rate with the adjustment window length in the Sakoe and Chiba's symmetrical algorithm with the gradient constraint P=1 is illustrated in Figures 3.10 and 3.11. An improvement in the recognition rate is obtained as the adjustment window length r is increased. For low values of r, the recognition rate is poor since the possibility that r is less than the absolute difference between input and reference patterns arises.

Using Paliwal's modification on the Sakoe and Chiba algorithm, has the effect of significantly improving the recognition rate, as has been reported by Paliwal et al [45]. As the value of r is increased from unity, the recognition rate increases to a maximum value. Likewise using Myers' algorithm to normalize the patterns to equal temporal length and then applying the Sakoe and Chiba DTW algorithm, improves the recognition rate, but as the value of r is increased a drop in the recognition rate may occur.

iii) Results obtained by applying Paliwal's modification on the various types of Sakoe and Chiba's algorithms are given in Table

3.3. Again the algorithm, using the gradient constraint P=1, and of the symmetrical form of matching, shows a significant superior performance over the other algorithms.

iv) Results obtained when the input and reference speech patterns are normalized to a fixed temporal length using the Myers' algorithm, and then applying the Sakoe and Chiba's DTW algorithm in the matching process, are shown in Table 3.4. The symmetrical matching algorithm, with gradient constraint P=1, again gives better performance than the other DTW algorithms. The effect of using the Myers' normalized patterns in the Itakura DTW algorithm gives a slight improvement in the recognition rate as indicated by the results in Tables 3.2 and 3.4. Myers et al [46] also reported similar characteristics. The advantage of using Myers' algorithm would be significant when the reference and input patterns have large temporal differences.

v) Table 3.5 shows the recognition results obtained when the Sakoe and Chiba asymmetrical algorithm were computed using the DTW method and when the OGS method was employed. The OGS method requires far less local distance computations than the DTW, but this is realised at the expense of a slight drop in the recognition rate. Brown and Rabiner [48] have reported similar conclusions in their comparison of the OGS method with the Itakura DTW algorithm. Since there are alternative ways, which will be discussed in Chapter 5, of reducing the computation time in a word recognizer with negligible loss in recognition accuracy, the OGS method does not offer any advantages.

In conclusion, from the results obtained in this Chapter, Paliwal's modification on the Sakoe and Chiba DTW algorithm with gradient constraint P=1, and of the symmetrical form was found to offer the best performance in an isolated word recognizer and as such it is employed in the recognizers discussed in subsequent chapters.

CHAPTER 4

THE USE OF FILTER BANK ENERGY FEATURES IN ISOLATED WORD RECOGNITION

4.1 INTRODUCTION

In the human speech perception system, speech signals undergo a bandpass filtering process in the cochlea, an organ located in the inner ear. Energy is extracted from the speech in frequency bands spaced according to the natural, or as is often called, the critical band frequency scale of the ear. From the distribution of energy in the frequency bands, important acoustic cues which enable the decoding of the message in the speech signal are determined.

Similarly, in computer based speech recognition systems an utterance can be represented as a discrete pattern of energy values obtained from the various frequency bands of a bandpass filtering process. In this Chapter, different filter banks, characterized by the number of filters, type of filters, filter passbands and filter spacing, are designed and used in the translation of a speech utterance into a pattern of energy parameter values. The effect of the type of filter bank employed in processing the speech utterances, on the accuracy of a word recognition system, is investigated. Finally, methods for reducing the redundancy present in speech energy patterns are suggested as a way of improving the word recognition accuracy.

4.2 FILTER BANK FEATURE EXTRACTION

The filter bank feature extraction process is illustrated in Figure 4.1. A speech utterance, S, is passed through a bank of Q bandpass digital filters which partition the frequency spectrum of the signal into various frequency bands. The passbands of the filters are usually designed to be continuous over the signal frequency spectrum, so that the composite spectrum of the overall filter bank does not

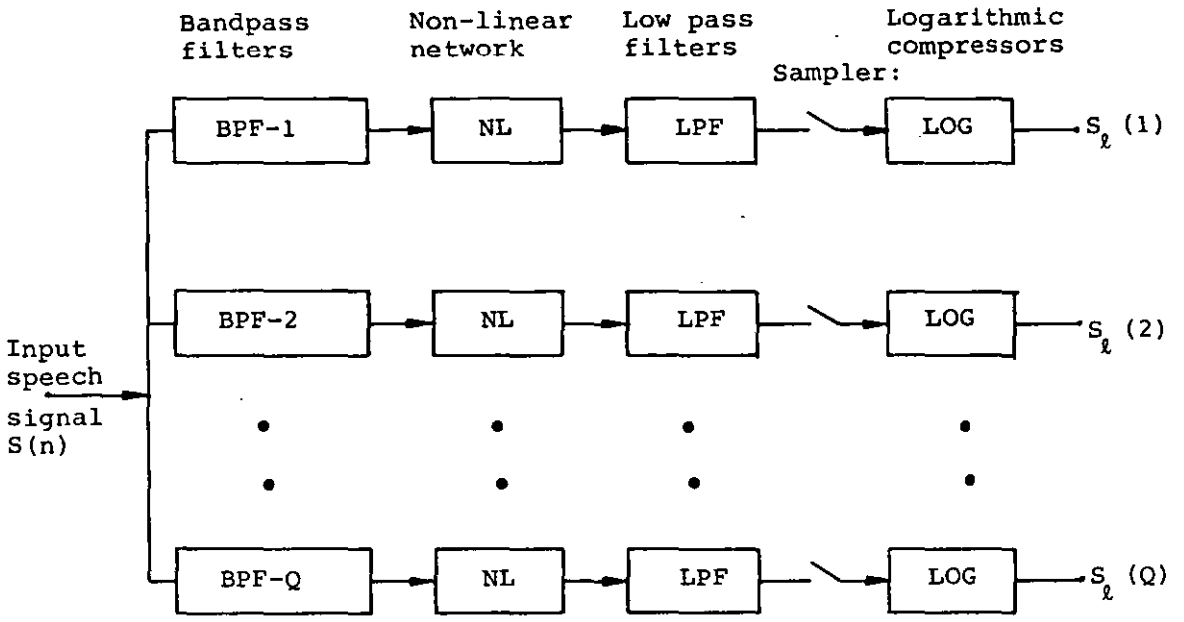


FIGURE 4.1: THE FILTER BANK FEATURE EXTRACTION

have sharp dips between adjacent filters. The frequency spacing can be determined in various ways. They can be placed on a uniform scale, a logarithmic scale or along a critical band frequency scale which is related to the human speech perception system.

Each bandpass output is passed through a non-linear network such as a square law detector, or a full wave rectifier. The non-linearity has the effect of non-uniformly distributing the original band-limited signal energy over the entire frequency spectrum. However the energy at low frequency in the resultant signal, is generally proportional to the total bandlimited energy. Thus, the output of the low pass filter (LPF) which follows the non-linear network, is a measure of the energy in the speech signal in a particular frequency band. The sampler decimates the output of the low pass filter at a rate twice the LPF cut-off frequency. The set of energy values obtained from the Q channels generally have a large dynamic range. For this reason, a logarithmic compressor is applied to reduce the range.

The set of energy values, for a given time instant, constitute a Q-dimensional vector. Thus the input speech utterance, S, is reduced to a temporal pattern of L vectors of energy values:

$$S = \{s_1, s_2, \ldots, s_\ell, \ldots, s_L\} \qquad 4.1a$$

where

$$s_\ell = [s_\ell(1), s_\ell(2), \ldots s_\ell(Q)], \quad 1 \leq \ell \leq L \qquad 4.1b$$

is a vector of logarithmic energy values measured from the Q channels at a time instant ℓ. In speech recognition terminology, the vector s_ℓ is also known as a frame. The output speech pattern from the filter bank, is then subjected to a time and frequency normalization procedure, as described below.

4.2.1 Channel Thresholding

The purpose of channel thresholding is to clamp low level noise signals in the channels, at instances when essentially no speech signal is present. This is accomplished by applying a threshold value, so that energy values below a certain level are clamped at the threshold value. In this manner, the operation of the system is less sensitive to background noise present in the input speech signal. The peak signal energy, for the utterance S, in the ith channel is given as:

$$s(i)^{max} = \underset{1 \leq \ell \leq L}{MAX} [s_\ell(i)] \qquad 4.2$$

and the threshold, $\bar{T}(i)$, for the ith channel is set at:

$$\bar{T}(i) = s(i)^{max} - T \qquad 4.3$$

The major effect of a finite value of T, is to eliminate errors arising due to widely varying energy values in bands with no speech energy. As such, its actual value is not important, as long as it is at least equal to the average signal to noise ratio of the input speech. In practice, the peak signal to noise ratio of the input speech has been suggested [50] as a suitable choice of T.

4.2.2 Energy Normalization

Energy normalization attempts to compensate for the variation in the gain level of speech from one utterance to another utterance. For the ith frame, the average energy value $\bar{s}_\ell$ is given by:

$$\bar{s}_\ell = (1/Q) \sum_{i=1}^{Q} s_\ell(i), \text{ for } 1 \leq \ell \leq L \qquad 4.4$$

The normalized feature vector of the frame is then given by:

$$\bar{s}_{\ell}(i) = s_{\ell}(i) - \bar{s}_{\ell}, \quad 1 \leqslant i \leqslant Q \qquad 4.5$$

The whole discrete sequence, S, is thus transformed into a new sequence of normalized values.

4.3 THE DIGITAL FILTER BANKS

Finite Impulse Response (FIR) filters, were chosen for the bandpass filters in Figure 4.1, owing to their linear phase properties and stability. These filters were designed using the window approach technique (see Appendix A), although the equiripple approximation and the frequency sampling method are known [51] to give better performance in terms of the filter passband-stopband transition width for a given passband and stopband ripple factor. The main advantage of the window design approach is its relative simplicity, and its flexibility in specifying the length of the filter impulse response.

4.3.1 Filter Bank Spacing

Four different methods were investigated in dividing the input signal frequency range into continuous bands, that is: uniform spacing, octave spacing, 1/3 octave spacing, and critical band spacing were investigated.

i) **Uniform spacing**

The uniformly spaced filter bank is obtained by dividing the frequency spectrum uniformly into the required number of channels. The centre frequency, f_i of the ith channel is given by:

$$f_i = (F_S/N) \quad i; \quad i = 1, 2, \ldots Q \qquad 4.6$$

where F_s is the sampling frequency, N is the number of filters that span the baseband frequency of the speech signal. Thus, Q satisfies the property

$$Q \leqslant N/2$$

since the channels for i > N/2 are mirror images of the first N/2 channels.

The bandwidth Δf_i of the ith channel is given by:

$$\Delta f_i = (f_i - f_{i-1}).\ k, \quad k \geqslant 1 \qquad 4.7$$

when the factor k=1, the filters are placed "end to end". For k > 1, there is overlap between adjacent filters.

ii) **Ideal octave and 1/3 octave spacing**

An alternative filter bank spacing is to arrange the channel bandwidth equally along a logarithmic scale. The bandwidth of the ith channel, ΔF_i is given by:

$$\Delta F_i = a\ \ \Delta F_{i-1}, \quad i = 2, 3, \ldots, Q \qquad 4.8a$$

$$\Delta F_1 = C \qquad 4.8b$$

where a and C are fixed constants.

The centre frequency, F_i, of the ith channel, is given by the relation:

$$F_i = F_0 + \sum_{\ell=1}^{i-1} \Delta F_\ell + \frac{\Delta F_i}{2} \qquad 4.9a$$

where $$F_0 = F_1 - (\Delta F_1/2) \qquad 4.9b$$

is the lower cut-off frequency of the first channel.

For a value of a = 2, the spacing is ideal octave, and for a = 4/3, the spacing is 1/3 octave.

iii) **Critical band spacing**

Critical band spacing is based on the human auditory perception system, where the ear is known to process speech using a filter bank type of analysis [52][53]. The filter spacing is highly non-uniform, and with characteristics that would be difficult to obtain with conventional filter design techniques. Figure 4.2 is an illustration of the amplitude/frequency characteristics of the auditory system filters [54]. The centre frequency F_i, the lower and upper cut-off frequencies F_L and F_H can be approximated by first transforming the linear frequency scale into a non-linear Barks [54] scale, as follows:

$$B(F) = \begin{cases} 0.01\ F & 0 \leqslant F \leqslant 500 \\ 0.007F + 1.5, & 500 \leqslant F \leqslant 1200 \\ 6\ \ln F - 32.6, & F \geqslant 1220 \end{cases} \qquad 4.10$$

where F is the frequency in Hz, B is the frequency in Barks. For a given filter, centred at frequency F_i, the critical bandwidth is obtained by first evaluating:

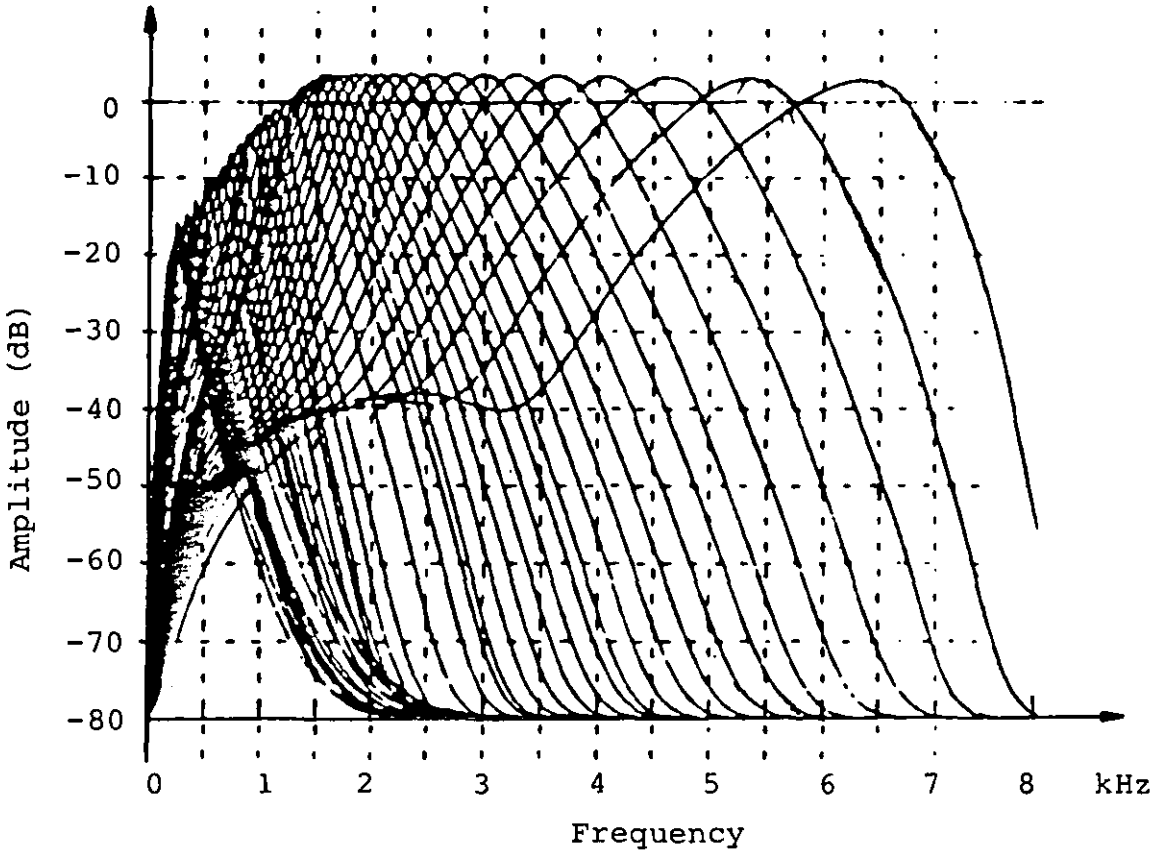


FIGURE 4.2: FREQUENCY CHARACTERISTICS OF THE AUDITORY SYSTEM FILTER [54]

$$B_0 = B(F_i) \qquad 4.11$$

The lower cut-off frequency F_L, the upper cut-off frequency F_H, and the critical bandwidth ΔF_i are given by [54]:

$$F_L = F\ (B_0 - 0.5)$$

$$F_H = F\ (B_0 + 0.5)$$

$$\Delta F_i = F_H - F_L \qquad 4.12$$

Thus, using equations 4.10, 4.11 and 4.12, it can be deduced that the lower frequency channels, i.e. $200 \leqslant F_i \leqslant 500$, have a bandwidth of 100 Hz approximately.

4.3.2 Bandpass Filter Design Results

The individual bandpass filters in the filter bank were designed by truncating the infinite response of the ideal bandpass response with a Hamming window as described in Appendix A. An impulse response of 128 samples in length was found to give reasonably sharp cut-off filters in the bandwidth range 100 Hz - 2.5 kHz.

i) **Uniform filter bank**

Filter banks with 5, 8, 10, 12 and 16 channels equally spaced along the 5 kHz signal spectrum and with no overlap were designed. The spacing of these filters is given in Table 4.1. Figure 4.3 illustrates the characteristics (log magnitude against frequency) of the individual filters in the 8-channel filter bank with no overlapping between adjacent filters.

TABLE 4.1

UNIFORMLY SPACED FILTER BANKS WITH 5, 8, 10, 12 AND 16 CHANNELS

	PASSBANDS (Hz)				
Channel No.	5 Channel Filter Bank Bandwidth (BW) = 920 Hz	8 Channel Filter Bank Bandwidth (BW) = 580 Hz	10 Channel Filter Bank Bandwidth (BW) = 460 Hz	12 Channel Filter Bank Bandwidth (BW) = 400 Hz	16 Channel Filter Bank Bandwidth (BW) = 285 Hz
1	200-1120	150- 730	200- 660	150- 550	200- 485
2	1120-2040	730-1310	660-1120	550- 950	485- 770
3	2040-2960	1310-1890	1120-1580	950-1350	770-1055
4	2960-3880	1890-2470	1580-2040	1350-1750	1055-1340
5	3880-4800	2470-3050	2040-2500	1750-2150	1340-1625
6		3050-3630	2500-2960	2150-2550	1625-1910
7		3630-4210	2960-3420	2550-2950	1910-2195
8		4210-4790	3420-3880	2950-3350	2195-2480
9			3880-4340	3350-3750	2480-2765
10			4340-4800	3750-4150	2765-3050
11				4150-4550	3050-3335
12				4550-4950	3335-3620
13					3620-3905
14					3905-4190
15					4190-4475
16					4475-4760

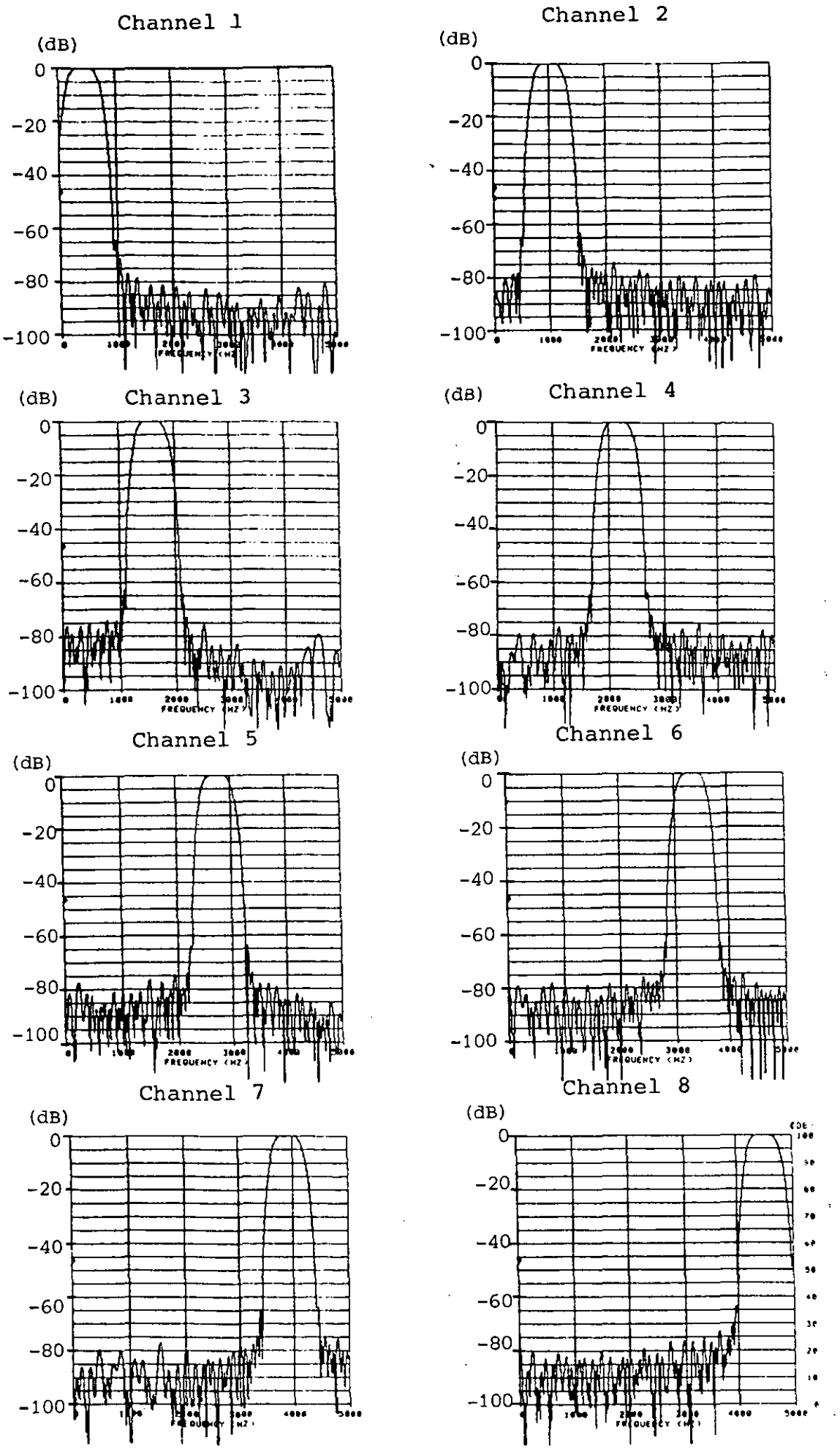


FIGURE 4.3: FREQUENCY RESPONSE OF THE 8-CHANNEL UNIFORM SPACED FILTER BANK

ii) **Ideal octave spacing**

Assigning a bandwidth of the order of 100 Hz for the low frequency channel, the 5 kHz frequency spectrum range is divisible into 5 channels spaced along an ideal octave frequency scale. The filter spacings are given in Table 4.2. The frequency response of the individual channels is shown in Figure 4.4.

iii) 1/3 octave spacing

The low frequency channel is assigned a bandwidth of the order of 100 Hz. The 5 kHz frequency spectrum range then yield 8 channels spaced along a 1/3 octave frequency scale as shown in Table 4.3. The frequency response of the individual channels is shown in Figure 4.5.

iv) Critical band spacing

The sixteen channels which cover the 5 kHz frequency spectrum on a critical band scale are given in Table 4.4. The frequency response of the individual channels is shown in Figure 4.6.

4.4 THE WORD RECOGNITION SYSTEM

4.4.1 System Description

The block diagram of the word recognition system based on the filter bank analysis is shown in Figure 4.7. The input speech samples, S(n), n=1,2, ..., N, are passed through the bank of bandpass filters. A filtered signal, $y_i(n)$, is obtained at the output of the ith bandpass filter as a convolution of the input signal with the filter impulse response thus:

$$y_i(n) = \sum_{m=0}^{M-1} h_i(m)\, S(n-m); \quad 1 \leq i \leq Q \qquad 4.13$$

TABLE 4.2

IDEAL OCTAVE FILTER BANK OVER THE BASEBAND 150-4800 Hz

Channel No.	Passband (Hz)	Bandwidth (Hz)
1	150- 300	150
2	300- 600	300
3	600-1200	600
4	1200-2400	1200
5	2400-4800	2400

TABLE 4.3

1/3 OCTAVE FILTER BANK OVER THE 150-4860 Hz BASEBAND

Channel No.	Passband (Hz)	Bandwidth (Hz)
1	150- 325	175
2	325- 555	230
3	555- 866	311
4	866-1280	414
5	1280-1830	550
6	1830-2570	740
7	2570-3550	980
8	3550-4860	1310

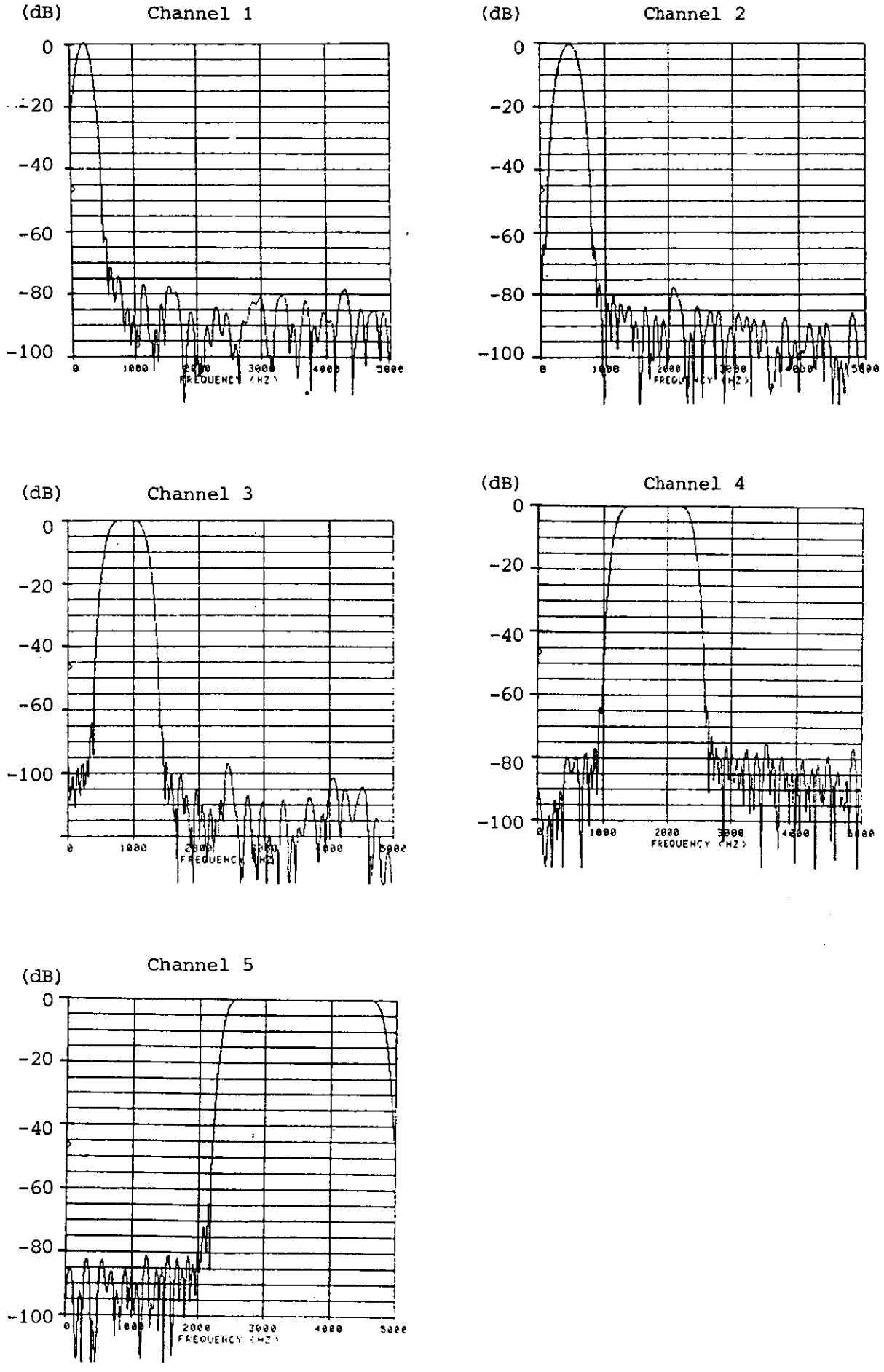


FIGURE 4.4: FREQUENCY RESPONSE OF THE 5 CHANNEL IDEAL OCTAVE FILTER BANK

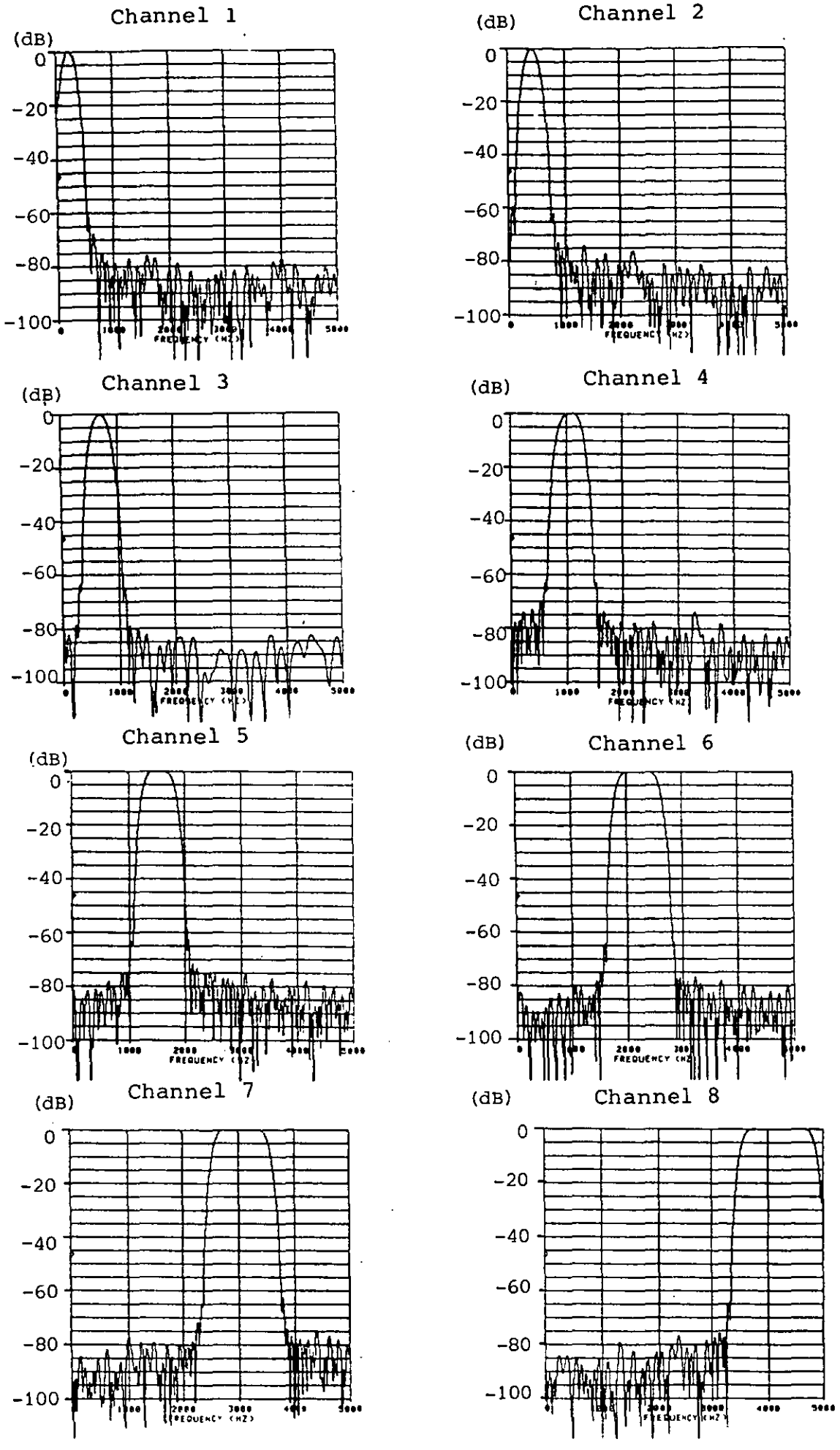


FIGURE 4.5: FREQUENCY RESPONSE OF THE 8 CHANNEL 1/3 OCTAVE FILTER BANK

TABLE 4.4

THE CRITICAL BAND SPACED FILTER BANK

Channel No.	Passband (Hz)	Bandwidth (Hz)
1	250- 350	100
2	350- 450	100
3	450- 550	100
4	550- 690	140
5	690- 830	140
6	830- 970	140
7	970-1110	140
8	1110-1255	145
9	1255-1480	225
10	1480-1750	270
11	1750-2070	320
12	2070-2450	380
13	2450-2900	450
14	2900-3430	530
15	3430-4060	630
16	4060-4800	740

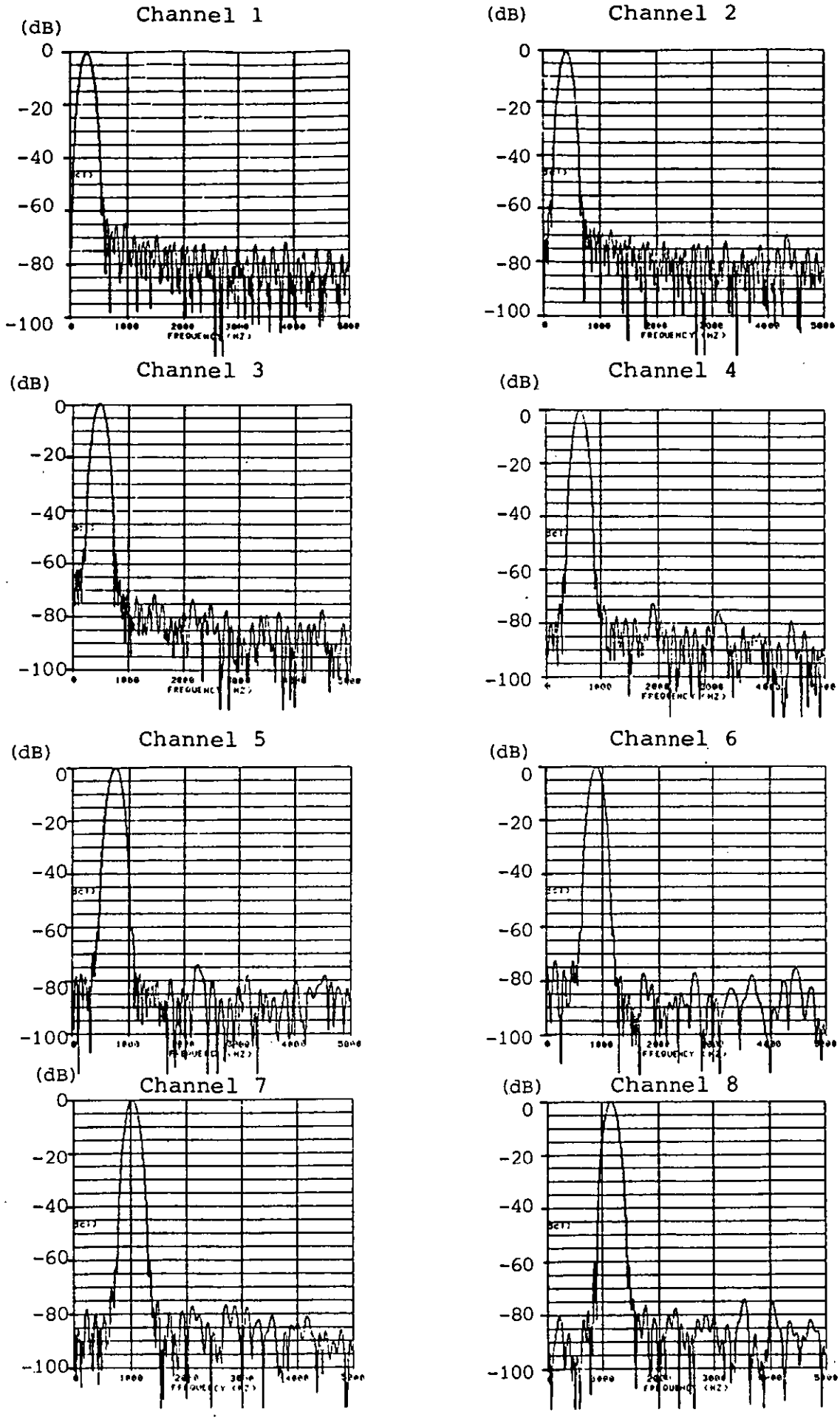


FIGURE 4.6(a): FREQUENCY RESPONSE OF THE 16-CHANNEL CRITICAL BAND FILTER BANK

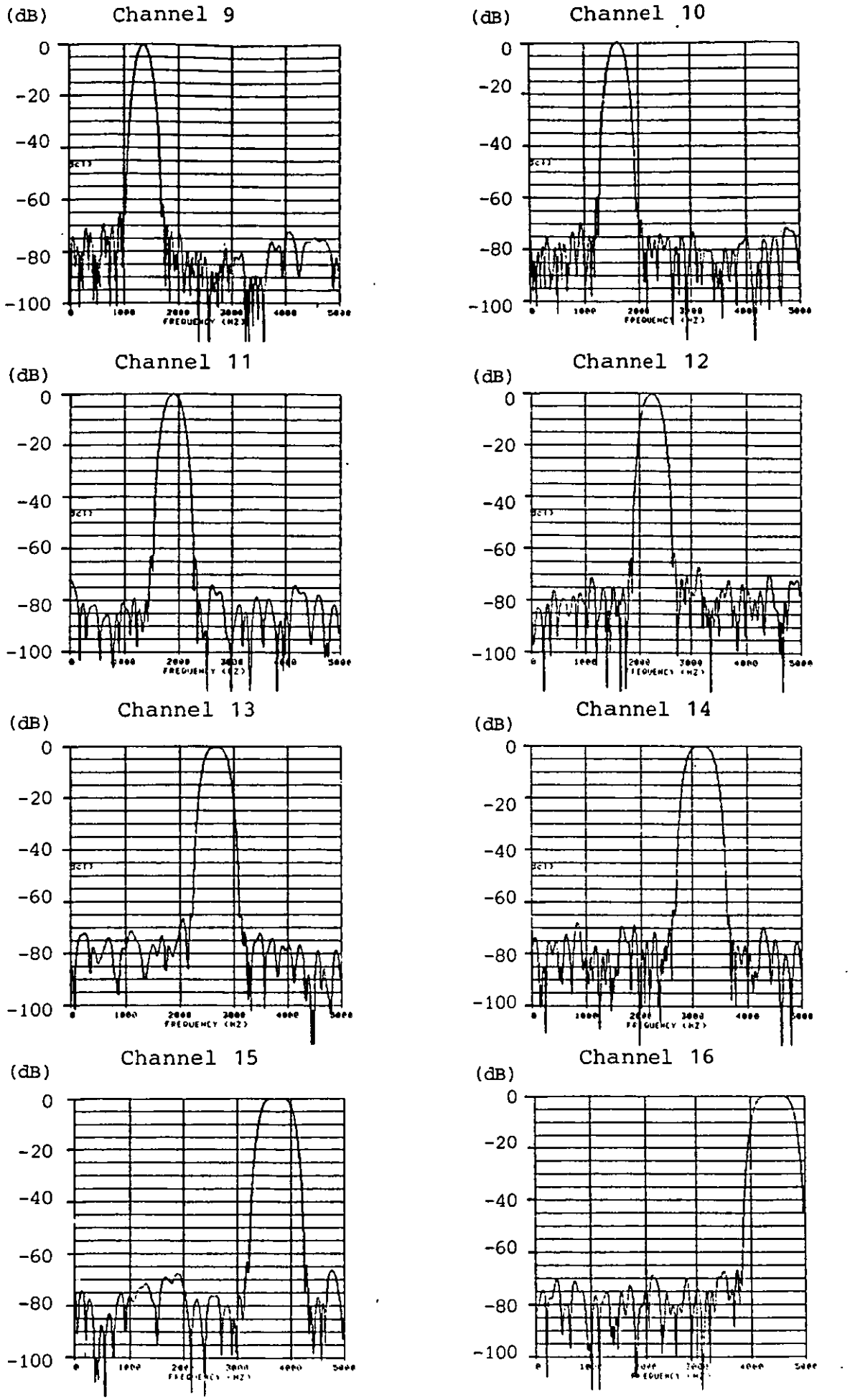


FIGURE 4.6(b): FREQUENCY RESPONSE OF THE 16-CHANNEL CRITICAL BAND FILTER BANK

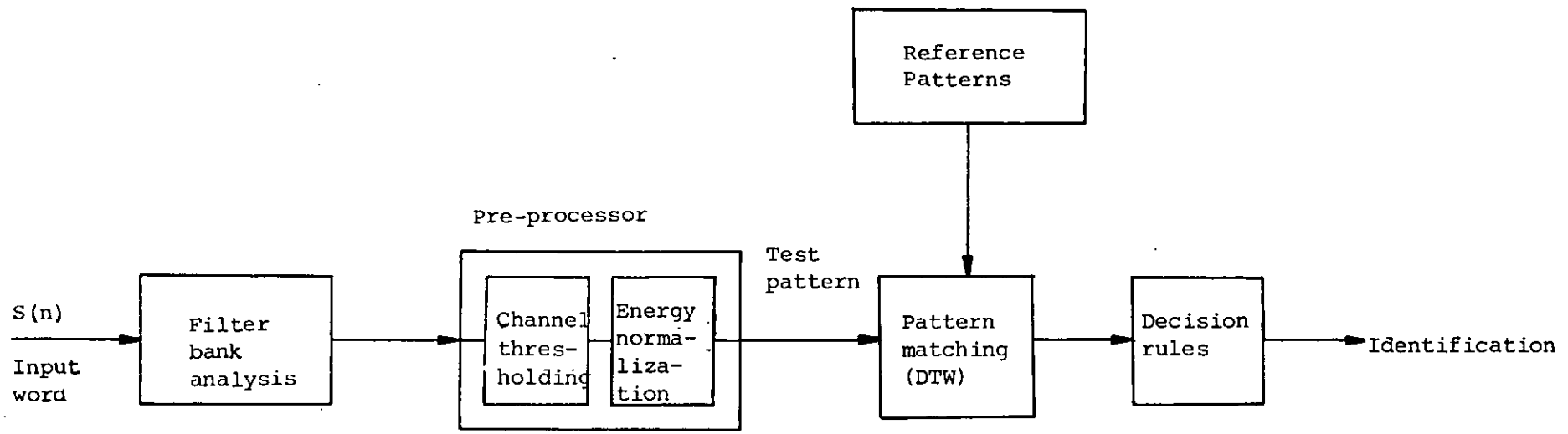


FIGURE 4.7: THE WORD RECOGNITION SYSTEM

where M is the duration of the filter impulse response h(m). The signal, $y_i(n)$, is passed through a square law detector to obtain a signal sequence $\bar{y}_i(n)$. At the discrete time instant, n, the energy $s_n(i)$, in the ith channel can be estimated by low pass filtering the signal $\bar{y}_i(n)$, i.e.

$$s_n(i) = \sum_{k=0}^{K-1} \bar{y}_i(n-k)\, w(k) \qquad 4.14$$

where $$\bar{y}_i(n) = (y_i(n))^2, \quad 1 \leq n \leq N+M-1$$

A Hamming window, w(k), with a length K = 256, was employed as a LPF. Thus, at each time instant, n, the output of the filter bank gives a Q-dimensional vector of energy values. By sampling $s_n(i)$, at suitable temporal intervals, the input signal can be represented as a pattern of L vectors of Q dimensions, as given in equation 4.1. The sampling interval of 25.6 msec was used in the experiments described here. The speech pattern is then subjected to a pre-processing stage where a channel threshold of 50 dB below peak signal energy, (i.e. T = 50 dB) is employed. An energy normalization procedure as described by equations 4.4 and 4.5, is then performed. The resultant speech pattern of normalized energy vectors is then compared with pre-stored reference patterns of the vocabulary words. The pattern comparison yields a set of distance scores which are passed over to a decision stage, where the identification of the input word is made.

i) Training session

The aim of a training session is to create reference patterns to be used in the word recognition process. In a single reference pattern recognition system, each vocabulary word is represented by one reference pattern formed from the utterances of a given speaker. In a multiple reference pattern system, each vocabulary word is

represented by patterns formed from the repetitions of the word by different speakers.

ii) Testing session

In the testing session, the performance of the recognizer in correctly identifying an input word is assessed. The input word is reduced to a discrete sequence of channel energy values and compared with all the pre-stored set of reference patterns of the vocabulary words. Let the input word be represented by the pattern A, of I frames, and that each frame contains a set of Q energy values.

i.e. $$A = \{a_1, a_2, \ldots, a_i, \ldots, a_I\} \qquad 4.15a$$

where $$a_i = [a_{i1}, a_{i2}, \ldots, a_{iq}, \ldots a_{iQ}] \qquad 4.15b$$

Likewise, a reference pattern B is the sequence of J frames:

$$B = \{b_1, b_2, \ldots, b_J\} \qquad 4.16a$$

where $$b_j = [b_{j1}, b_{j2}, \ldots, b_{jq}, \ldots b_{jQ}] \qquad 4.16b$$

The matching process of pattern A to B requires the definition of a local distance between their frames. The absolute norm has been suggested as a suitable measure, and is used here mainly because of its simplicity and its reported satisfactory performance [55].

Thus, a distance $d(a_i, b_j)$ between frames a_i and b_j is given as

$$d(a_i, b_j) = \sum_{q=1}^{Q} | a_{iq} - b_{jq} | \qquad 4.17$$

The Euclidean distance measure can also be used, as well as other alternative distance measure metrics proposed by Klatt [56].

The total distance D(A,B) between pattern A and B is obtained using the Sakoe and Chiba dynamic time warping algorithm with Paliwal's modification, as discussed earlier in Section 3.2 of Chapter 3. In a single reference pattern per vocabulary word system, the input word is recognized as the vocabulary word which gives the minimum error matching, i.e. the nearest neighbour rule.

Let $R = \{R_1, R_2, \ldots, R_v, \ldots, R_V\}$ be the set of reference patterns of the V vocabulary words. The DTW matching stage computes the distance, $D(A,R_v)$, $v = 1, 2, \ldots V$, between the input word and the vth reference pattern. The input word is recognized as the vocabulary word r represented by the rth pattern such that

$$D(A,R_r) = \min_{1 \leq v \leq V} D(A, R_v) \qquad 4.18$$

When multiple patterns are used for each vocabulary word, then the 'k nearest neighbour' (kNN) rule is used to identify the input word. Let each vocabulary word be represented by M patterns. Thus, the reference patterns form the set $\{R_1^m, R_2^m, \ldots, R_v^m, \ldots R_V^m\}$, where $m = 1,2, \ldots M$. R_v^m is the mth occurrence of the vth vocabulary word.

Let the DTW distance between the input word pattern, A, and the reference pattern R_v, be denoted as D_v. For each vocabulary word there are M distances. If these distances are re-ordered so that:

$$D_v^{A,1} \leq D_v^{A,2} \leq \ldots \leq D_v^{A,M} \qquad 4.19$$

Then, for the kNN rule the average distance D_v is computed from:

$$\bar{D}_v = \sum_{k=1}^{K} D_v^{A,k}, \quad K < M \qquad 4.20$$

The input word is recognized as the vocabulary word represented by the rth reference word which gives:

$$\bar{D}_r = \min_{1 \leq v \leq V} \bar{D}_v \qquad 4.21$$

4.4.2 Recognition Results

A series of experiments were performed in order to assess the influence of the different filter bank designs on the recognition accuracy. The investigations involved the use of: (a) uniformly spaced filter banks with 5, 8, 10, 12, and 16 channels; (b) a third octave spaced filter bank with 8 channels; (c) ideal octave spaced filter bank with 5 channels, and (d) critical band spaced filter bank with 16 channels.

i) **Recognition performance in systems using single reference pattern per vocabulary word**

Table 4.5 gives the results obtained in different recognition systems employing single reference pattern per vocabulary word. Utterances from the male speakers SM1 and SM3, and from the female speaker SF1 were used to test the recognizer in a speaker independent mode. The utterances of speaker SM2 were used in generating the reference patterns.

ii) **Recognition performance in systems using multiple reference patterns per vocabulary word**

In these experiments, each vocabulary word was represented by four patterns obtained from different speakers. The recognizer was tested with speech utterances from speakers who did not contribute to the

TABLE 4.5

PERFORMANCE OF THE WORD RECOGNIZER WITH VARIOUS FILTER BANK SYSTEMS: SINGLE REFERENCE PATTERN/VOCABULARY WORD

FILTER BANK SYSTEM		RECOGNITION ACCURACY (%)			
Filter spacing	No. of Channels	Test 1 Test Speaker: SM1 Ref Speaker: SM2	Test 2 Test Speaker: SM3 Ref Speaker: SM2	Test 3 Test Speaker: SF1 Ref Speaker: SM2	Average
Uniform	5	52	56	42	50.0
Uniform	8	60	66	54	60.0
Uniform	10	62	68	56	62.0
Uniform	12	62	62	56	60.0
Uniform	16	58	60	54	57.3
Ideal Octave	5	64	66	52	60.6
1/3 Octave	8	68	70	62	66.6
Critical Band	16	64	62	58	61.3

generation of the reference patterns i.e. speaker independent mode. The decision process in the recognizer employed a kNN rule with k = 3, to identify an input word, as described in Section 4.4.1. The experimental results are given in Table 4.6.

4.5 THE EFFECT OF SPEECH SIGNAL REDUNDANCY SUPPRESSION ON RECOGNIZER PERFORMANCE

In a word recognition system, a speech utterance, A, is expressed as a discrete sequence of points in a multi-dimensional feature space. Since the speech signal consists of stationary and transitional regions during which a rapid change in the characteristic of the spectrum occurs, some points along the discrete sequence representation will be spectrally similar to their immediate predecessors, and others will show large differences. In this section, two methods of transforming the points in A into a new sequence B, where the points have either less redundancy or are distributed equidistantly in a multi-dimensional space are discussed. The transformed speech patterns are then used in the word recognition system.

4.5.1 A Simple Redundancy Removal Method

Given a speech utterance, A, as a sequence of I multi-dimensional feature vectors, a less redundant sequence B can be formed by neglecting vectors in A which are less than a certain threshold distance from their immediate predecessors.

Let, $$A = \{a_1, a_2, \ldots, a_i, \ldots, a_I\} \quad 4.22a$$

then $$B = \{b_1, b_2, \ldots, b_i, \ldots, b_{I'}\} \quad 4.22b$$

where $I' < I$

and $b_i = a_i$ if $d(a_{i-1}, a_i) > d_T$

TABLE 4.6

PERFORMANCE OF THE WORD RECOGNIZER WITH VARIOUS FILTER BANKS
SYSTEMS: MULTIPLE REFERENCE PATTERNS/VOCABULARY WORD

FILTER BANK SYSTEM		RECOGNITION ACCURACY (%)			
Filter spacing	No. of Channels	Test 1 Test Speaker: SM1 Ref Speakers: SM2,SM4,SF2,SM3	Test 2 Test Speaker: SM3 Ref Speakers: SM2,SM4,SF2,SM1	Test 3 Test Speaker: SF1 Ref Speakers: SM2,SM4,SF2,SM3	Average
Uniform	5	62	72	64	66.0
Uniform	8	72	78	72	74.0
Uniform	10	74	86	80	80.0
Uniform	12	70	80	76	75.3
Uniform	16	74	70	72	72.0
Ideal Octave	5	70	84	80	78.0
1/3 Octave	8	74	90	82	82.0
Critical Band	16	72	88	82	80.6

d_T is a threshold value dependent on the required length I' of B. If N frames are to be purged from A, then from the ordered list of inter-frame distances, $d_1 \leq d_2 \leq \ldots d_N \leq \ldots d_{I-1}$, d_T is set equal to d_N. The original sequence is said to be compressed by a factor of (I'/I) in obtaining the new sequence B.

4.5.2 The Trace Segmentation method [57][58]

The vectors a_1, a_2, ..., a_I of the speech pattern A, (equation 4.2a), can be considered as describing a trace of I points in the multi-dimensional feature space. The idea behind trace segmentation is to re-distribute the I points along the trace, into a fewer number of points which are equidistantly spaced. This is achieved by partitioning the trace into S segments, and thereafter using the S+1 segment boundaries as the new points which describe the trace. The total accumulated distance D, along the discrete sequence A is computed as:

$$D = \sum_{i=1}^{I-1} d(a_i, a_{i+1}) \qquad 4.23$$

where $d(a_i, a_{i+1})$, is the absolute distance between vectors a_i and a_{i+1}.

The distance, D, is to be distributed equally along the S segments of the trace. Thus, once S is fixed, the distance D_L between two consecutive points in the transformed sequence will be given by:

$$D_L = D/S \qquad 4.24$$

The transformed sequence $B = \{b_1, b_2, \ldots, b_{S+1}\}$, is formed from the sequence A, as follows:

The first and last points in A are retained in the transformed sequence, i.e.

$$b_1 = a_1 \qquad \text{and} \qquad b_{S+1} = a_I$$

Then, starting with b_1 as the first point in the transformed sequence, the distance between consecutive points in A are computed until the following conditions are met:

$$d(b_1, a_2) + d(a_2, a_3) + \ldots + d(a_{k-1}, a_k) \geqslant D_L \qquad 4.25$$

$$d(b_1, a_2) + d(a_2, a_3) + \ldots + d(a_{k-2}, a_{k-1}) < D_L \qquad 4.26$$

From equations 4.25 and 4.26, there exists a point b_2, which belongs to the space between a_{k-1} and a_k, such that:

$$d(b_1, a_2) + d(a_2, a_3) + \ldots + d(a_{k-1}, b_2) = D_L \qquad 4.27$$

Equation 4.27 can be rearranged as:

$$d(a_{k-1}, b_2) = D_L - d(b_1, b_2) - d(a_2, a_3), - \ldots - d(a_{k-2}, a_{k-1}) \qquad 4.28$$

Let the Q energy values, in say vector a_j, be denoted as:

$$a_j^1, a_j^2, \ldots a_j^q, \ldots, a_j^Q$$

The elements of vector b_2, can be located by linear interpolation:

$$(b_2^q - a_{j-1}^q)/D_L = (a_j^q - a_{j-1}^q)/d(a_{j-1}, a_j) \qquad 4.29$$

where q = 1,2, ..., Q.

$$b_2^q = a_{j-1}^q + \{(a_j^q - a_{j-1}^q) \cdot D_L/d(a_{j-1}, a_j)\} \qquad 4.30$$

Hence, starting from b_2, the algorithm is repeated to obtain:

$$b_3, b_4, \ldots, b_{S+1} \qquad 4.31$$

Thus, the original sequence is replaced with the new sequence:

$$B = \{b_1, b_2, b_3, \ldots, b_{S+1}\} \qquad 4.32$$

where $b_1 = a_1$ and $b_{S+1} = a_I$.

The trace segmentation procedure compresses the original pattern, by a factor of (S/I), into points which are not temporally equidistant, but spaced according to spectral changes along the utterance. In the new pattern more points will be allocated on the transition regions of the speech utterance where more accurate description is required, at the expense of stationary regions where the signal is more redundant.

4.5.3 Results

The recognition system employing an 8 channel, 1/3 octave spaced filter bank is shown, in Table 4.6, to give better performance in a speaker independent manner, than the other systems under consideration. The effect of using speech utterances with reduced redundancy as described in Sections 4.5.1 and 4.5.2, on the performance of a word recognizer was then investigated for this type of filter bank. Figure 4.8 shows the recognition results, expressed as a percentage of correct identifications of the input word, obtained in three testing sessions. In these experiments, each vocabulary word was represented by four patterns from utterances of different speakers. The recognizer was tested with speech utterances from speakers who did not contribute to the generation of reference patterns, as given in Table 4.6. Both the simple redundancy removal and the trace segmentation methods were used with varying compression factors on the test and reference speech patterns.

4.6 DISCUSSION

In this Chapter, the use of filter bank features in word recognition is examined. Filter banks with different numbers of channels were designed and their frequency characteristics are shown in Figures 4.3, 4.4, 4.5 and 4.6. Each filter was designed by truncating the infinite response of an ideal bandpass filter with a Hamming window of 128 samples in length. The filters possess a reasonably sharp cut-off rate, and also provide an attenuation of at least -60 dB in the stop band.

On the performance of the word recognition system using various filter banks as given in Tables 4.5, 4.6 and Figure 4.8, the following points can be deduced:

i) For a given filter bank system, the recognition accuracy is greatly enhanced by using multiple reference patterns per vocabulary word, rather than a single reference pattern per

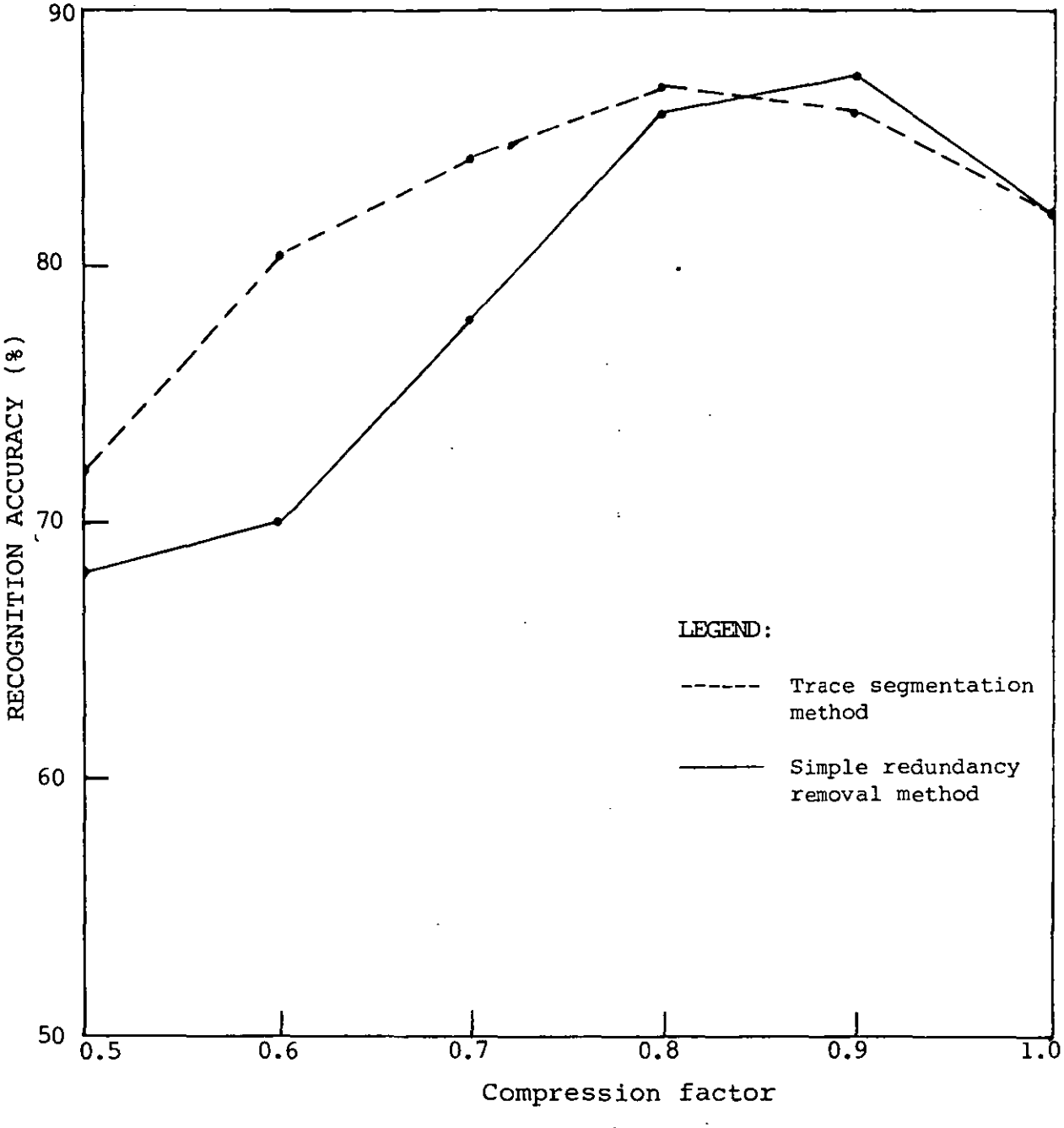


FIGURE 4.8: THE VARIATION IN THE PERFORMANCE OF THE WORD RECOGNIZER USING THE 8 CHANNEL 1/3 OCTAVE FILTER BANK WITH THE PATTERN COMPRESSION FACTORS

vocabulary word. This is to be expected, since the multiple patterns from different speakers represent a wider variability of the vocabulary word.

ii) On using uniformly spaced filter banks, the recognition performance degrades when the number of channels is too few (in the range of 5) or too many (in the range of 16). The recognition accuracy obtained with 10 channels was the highest. The reason for the degradation in performance when the number of channels is high is that the individual filters become so narrow in bandwidth that the energy estimation is poor due to large fluctuations in the spectrum. With a low number of channels, the system has a very poor frequency resolution which leads to inability to discriminate between words. Similar results have been reported by Dautrich et al [50].

iii) Given the same number of channels, filters spaced on a non-uniform scale give a better performance than those spaced on a uniform scale, i.e. the 5 channel ideal octave vis-à-vis the 5 channel uniform spaced filter bank, the 8 channel 1/3 octave vis-à-vis the 8 channel uniformly spaced filter bank, and the 16 channel critical band vis-a-vis the 16 channel uniform filter bank. The reason for this can be attributed to the fact that the uniform filter bank spacing weighs all the regions in the spectrum equally, whereas the non-uniform filter banks have a bias toward the lower spectrum range. From subjective listening tests, it is known that the spectral range below 3 kHz is more important than the upper frequency range (3 kHz-5 kHz) in the identification of speech utterances consisting mainly of voiced sounds. Probably, it is this bias in the non-uniform filter banks, which gives them a superior performance over the uniform filter banks, in the recognition task. Note that the 50-word recognition vocabulary consists of utterances with mainly voiced sounds.

iv) The highest word recognition accuracy was obtained using the 8 channel 1/3 octave filter bank.

v) An improvement in the accuracy of the word recognition system can be achieved by using the proposed simple redundancy removal, or the trace segmentation methods, as shown in Figure 4.8. When a compression factor of 0.5 is used, the performance is severely degraded since useful information in the speech patterns is lost. A compression factor of 0.8 or 0.9 produces a significant improvement in recognition accuracy. This is because a bias is introduced, in which more features are extracted from transitional regions, as opposed to stationary regions in the speech signal. A major setback with the simple redundancy removal and the trace segmentation methods, is the difficulty involved in estimating the level of redundancy in a speech utterance, and hence the optimal compression factor to be used.

CHAPTER 5

THE USE OF LPC FEATURES IN ISOLATED WORD RECOGNITION

5.1 INTRODUCTION

Linear prediction analysis has been established as a predominant method in the estimation of speech parameters such as pitch, formants, vocal tract area functions, spectra etc, with a reasonable accuracy and a low computational load. The ability to describe the vocal tract transfer function with a small number of parameters, is of fundamental importance in many aspects of speech processing. The short time spectral estimation of speech using linear prediction, also provides a suitable representation of the signal for recognition purposes.

This chapter commences with a presentation of the linear prediction theory and its applicability in speech recognition. The performance of several LPC-based word recognition systems is then assessed by computer simulations. In order to achieve speaker independent performance, multiple reference patterns per vocabulary word are employed in the recognition system. However, an increase of the recognition accuracy in such a system is realized at the expense of a large increase in the computational load. A method, based on clustering the reference pattern into a small number of disjoint groups is suggested as a means of reducing the computational load.

The need to reduce the memory requirements in the recognizer, leads to the use of vector quantization techniques. Word recognition systems which employ vector quantization are therefore examined, and their performance compared. A new system, termed the LPC/VQ/SPLIT recognizer, which has a low memory requirement and still maintains a recognition accuracy comparable with established systems is finally proposed.

5.2 LINEAR PREDICTION OF SPEECH [11][59][60]

The main idea behind linear prediction is that a given speech sample can be approximated as a linear combination of its immediately preceding samples. Such a representation leads to a simple all-pole filter that can model the short term vocal tract transfer function with a reasonable accuracy.

5.2.1 Basic Principles

The block diagram in Figure 5.1 is an illustration of the basic all-pole speech synthesis model; a time-variant digital filter excited either by a periodic pulse train or by random noise. The steady state transfer function of the digital filter is of the form:

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a(k)\, z^{-k}} \qquad 5.1$$

where G is a gain parameter
and $a(k)$, $1 \leqslant k \leqslant p$ are the filter coefficients.

This transfer function, H(z) is a simplification of the filter in the source filter model of speech production, first proposed by Fant [61] in which the combined spectral contributions from the vocal tract, glottal excitation and the radiation of the lips are represented by a single all-pole time varying filter.

Voiced sounds are modelled by exciting the filter with pulses separated by a pitch period. Unvoiced sounds are modelled with random noise as the input. Nasals and fricative sounds are not well modelled by this simplified system since the acoustics of these sounds are described by a vocal tract transfer function containing zeros and poles. Nevertheless, for a high order of filter coefficients, p, a good representation of all kinds of sounds can be obtained with the all-pole synthesis model. A major advantage of the

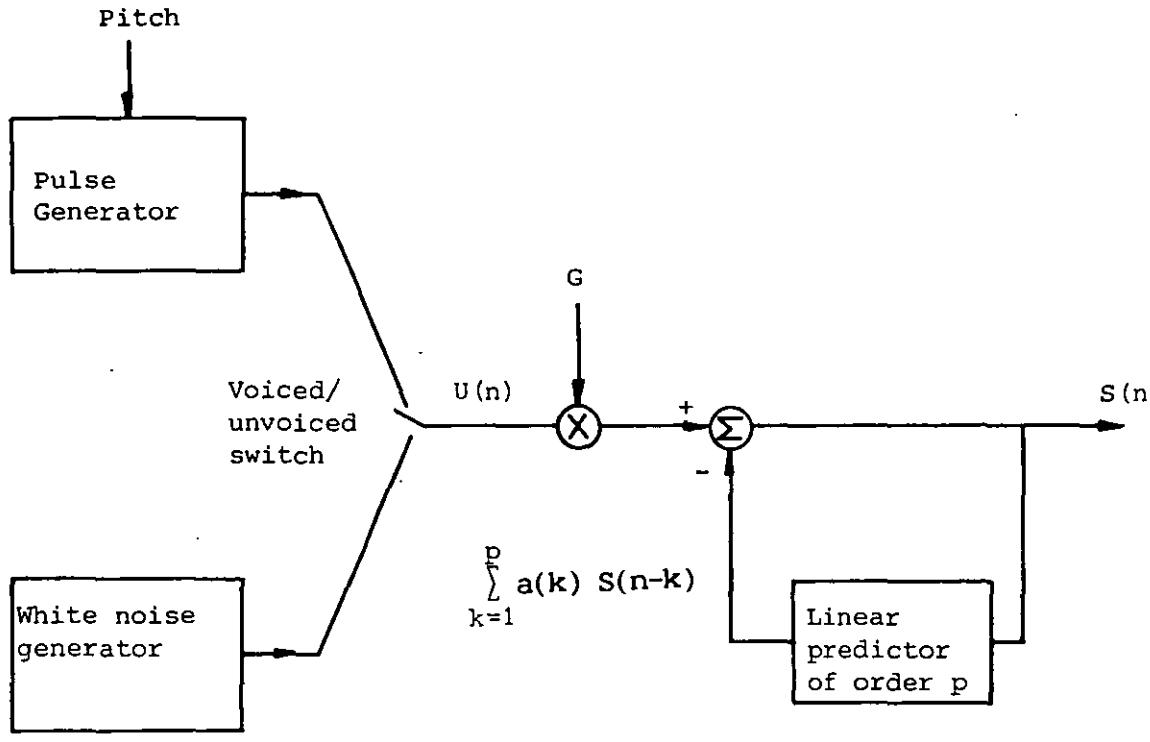


FIGURE 5.1: THE SOURCE-FILTER SPEECH SYNTHESIS MODEL

model is that the filter parameters, i.e. the gain and the filter coefficients, can be determined efficiently by the linear prediction analysis method.

Linear prediction, as its name implies, is a method of predicting a speech sample, $\bar{S}(n)$, from a linear combination of a number of immediately preceding samples, i.e.

$$\bar{S}(n) = \sum_{k=1}^{p} a(k)\ S(n-k) \qquad 5.2$$

where S(n-k), k = 1,2, ...p, are the p preceding samples. The weighting coefficients, a(k), k = 1,2, ... p, are optimized by minimizing the sum of the squares of differences between actual speech samples and the linearly predicted ones. These coefficients are known as the prediction coefficients, and p is termed the order of prediction.

Let the error between the actual value of a sample, S(n), and its predicted value, $\bar{S}(n)$, be e(n).
Then,

$$e(n) = S(n) - \bar{S}(n) = S(n) - \sum_{k=1}^{p} a(k)\ S(n-k) \qquad 5.3$$

The prediction coefficient are obtained by minimizing the total squared error, E, given by:

$$E = \sum e^2(n) = \sum (S(n) - \sum_{k=1}^{p} a(k)\ S(n-k))^2 \qquad 5.4$$

Depending on the range of summation in equation 5.4, there arises two distinct methods for the estimation of the prediction coefficients, namely the autocorrelation and the covariance methods.

5.2.2 The Autocorrelation Method

In the autocorrelation method of computing predictor coefficients, equation 5.4 is minimized by considering only speech samples within a finite duration, and that outside this duration, the speech samples have zero values. This can be accomplished by weighting the speech samples, S(n) using the rectangular window, w(n). The windowed speech samples, S(n) are given by:

$$S'(n) = \begin{cases} S(n)\, w(n), & w(n) = 1, \quad 1 \leqslant n \leqslant N-1 \\ 0, & \text{otherwise} \end{cases} \qquad 5.5$$

The length, N, of the window function w(n) is set to a suitable duration, since the speech signal is approximately stationary within short time segments. Since S(n) is non-zero only in the time interval $0 \leqslant n \leqslant N-1$, the error e(n) for the p^{th} order predictor will be non-zero over the interval $0 \leqslant n \leqslant N-1+p$. From equation 5.3, it can be observed that the error, e(n), will be large at the beginning of the interval, i.e. $0 \leqslant n \leqslant p-1$, because some of the predicted samples are set to zero. Likewise, e(n) can also be large at the end of the interval i.e. $N \leqslant n \leqslant N-1+p$, because the actual speech samples are set to zero. Therefore, a window function such as Hamming or Hanning, which gradually reduces the speech samples at the beginning and at the end of the interval, is generally used.

The total squared error, E, is given by:

$$E = \sum_{-\infty}^{+\infty} e^2(n) = \sum_{n=0}^{N-1+p} e^2(n) = \sum_{n=0}^{N-1+p} \left(S(n) + \sum_{k=1}^{p} a(k).S(n-k)\right)^2 \qquad 5.6$$

$$E \text{ is minimized by setting, } \frac{\partial E}{\partial a(i)} = 0, \quad 1 \leqslant i \leqslant p \qquad 5.7$$

which gives:

$$\sum_{n=0}^{N-1+p} \left(S(n) + \sum_{k=1}^{p} a(k)\, S(n-k)\right) \sum_{i=1}^{p} S(n-i) = 0 \qquad 5.8$$

or

$$\sum_{n=0}^{N-1+p} S(n)\, S(n-i) = - \sum_{k=1}^{p} a(k) \sum_{n=0}^{N-1+p} S(n-k)\, S(n-i) \qquad 5.9$$

$$i = 1,2, \ldots p$$

The minimum total squared error, E_p, also known simply as the minimum prediction error, is obtained by substituting equation 5.9 into 5.6 giving:

$$E_p = \sum_{n=0}^{N-1+p} S(n)^2 + \sum_{k=1}^{p} a(k) \sum_{n=0}^{N-1+p} S(n)\, S(n-k) \qquad 5.10$$

Equations 5.9 and 5.10 reduce to:

$$\sum_{k=1}^{p} a(k)\, R(i-k) = - R(i), \quad 1 \leqslant i \leqslant p \qquad 5.11$$

and

$$E_p = R(0) + \sum_{k=1}^{p} a(k)\, R(k) \qquad 5.12a$$

The normalized prediction error, V_p, is obtained by normalizing E_p with R(0):

i.e.
$$V_p = \frac{E_p}{R(0)} = 1 + \sum_{k=1}^{p} a(k)\, R(k)/R(0) \qquad 5.12b$$

where

$$R(i) = \sum_{n=-\infty}^{\infty} S(n)\, S(n-i) = \sum_{n=0}^{N-1} S(n)\, S(n-i), \quad 0 \leqslant i \leqslant p \qquad 5.13$$

The coefficients, R(i-k) in equation 5.11 are the autocorrelation coefficients of the speech signal, hence the name given to the analysis method.

The set of equations defined by equation 5.11, can be expressed in a matrix form as follows:

$$\begin{bmatrix} R(0) & R(1) & R(2) & \dots & R(p-1) \\ R(1) & R(0) & R(1) & \dots & R(p-2) \\ R(2) & R(1) & R(0) & \dots & R(p-3) \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ R(p-1) & R(p-2) & R(p-3) & & R(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ a(3) \\ \cdot \\ \cdot \\ a(p) \end{bmatrix} = - \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \cdot \\ \cdot \\ R(p) \end{bmatrix} \qquad 5.14$$

The p by p matrix of the autocorrelation coefficients is Toeplitz i.e. the elements along any given diagonal are equal, a property which can be advantageously exploited in the computation of the predictor coefficients.

5.2.3 The Covariance Method

In the second approach, the minimization of the total squared error, E, is done over a fixed interval, i.e. the error signal, e(n), is windowed but the speech samples are not.

$$E = \sum_{n=0}^{N-1} e^2(n) = \sum_{n=-p}^{N-1} \left(S(n) + \sum_{k=1}^{p} a(k)\, S(n-k)\right)^2 \qquad 5.15$$

Equation 5.9 now becomes:

$$\sum_{n=-p}^{N-1} S(n)\,S(n-k) = -\sum_{k=1}^{p} a(k) \sum_{n=-p}^{N-1} S(n-k)\,S(n-i), \quad 1 \leqslant i \leqslant p \qquad 5.16$$

Equation 5.16 is very similar to 5.9, except for the range of summation which uses values of S(n) in the interval $-p \leqslant n \leqslant N-1$, rather than over the interval $0 \leqslant n \leqslant N-1$. Although these differences seem to be minor, the set of linear equations eventually derived, has significantly different properties that affect the method of solution and leads to different predictor parameters.

Equation 5.16 reduces to:

$$\sum_{k=1}^{p} a(k)\ \phi(i,k) = -\ \phi(0,i), \quad 1 \leqslant i \leqslant p \qquad 5.17$$

where, $\phi(i,k) = \sum_{n=-\infty}^{\infty} w_e(n)\,S(n-i)\,S(n-k)$, gives the cross-correlation of the speech samples windowed by the function $w_e(n)$, usually a rectangular window.

The relationship defined by equation 5.17 can be expressed in matrix form as:

$$\begin{bmatrix} \phi(1,1) & \phi(1,2) & \dots & \phi(1,p) \\ \phi(2,1) & \phi(2,2) & \dots & \phi(2,p) \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \phi(p,1) & \phi(p,2) & \dots & \phi(p,p) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ \cdot \\ \cdot \\ a(p) \end{bmatrix} = \begin{bmatrix} \phi(1,0) \\ \phi(2,0) \\ \cdot \\ \cdot \\ \phi(p,0) \end{bmatrix} \qquad 5.18$$

The above covariance matrix is symmetrical, i.e. $\phi(i,k) = \phi(k,i)$ but unlike the autocorrelation matrix it is not Toeplitz.

The minimum prediction eror, E_p, is given by:

$$E_p = \phi(0,0) + \sum_{k=1}^{p} a(k)\ \phi(0,k) \qquad 5.19$$

5.2.4 Computation of the Predictor Coefficients

The solution to equations 5.11 and 5.17 for the predictor coefficients can be obtained using any of the established methods for solving p linear equations in p unknowns, e.g. the Gauss-Siedel method [62]. Generally these methods would require heavy computation, but utilising the properties of the coefficient matrices of these equations leads to more efficient and faster computation. For example, the symmetrical nature of the covariance matrix enables the use of Cholesky's decomposition solution [63]. For the autocorrelation method, the coefficient matrix is not only symmetrical but also Toeplitz and for this special case, Levinson [64] and Durbin [65] developed a recursive technique to efficiently compute the prediction coefficients for a given order. Their algorithm is as follows:

Step i: Given the matrix equation $\sum_{k=1}^{p} a(k)\ R(i-k) = -R(i)$, for $1 \leqslant i \leqslant p$, it is desired to solve for the predictor coefficients $\{a_k\}$, $k = 1, 2, \ldots p$

Step ii: Let $E_o = R(0)$

Step iii: Compute

$$k_i = -\left[R(i) + \sum_{j=1}^{i-1} a^{i-1}(j)\ R(i-j)\right]/E_{i-1} \qquad 5.20$$

Step iv: $a^i(i) = k_i$ 5.21

Step v: $a^i(j) = a^{i-1}(j) + k_i\, a^{i-1}(i-j); \quad 1 \leqslant j \leqslant i-1$ 5.22

Step vi: $E_i = (1 - k_i^2)\, E_{i-1}$ 5.23

Step vii: Steps (iii), (iv) and (v) are solved recursively for i = 1, 2, ...p and the final solution for the predictor coefficients is given by:

$$a(j) = a^p(j) \quad 1 \leqslant j \leqslant p \qquad 5.24$$

In the algorithm, the computation for predictor order p, is preceded by the computation for the solution of all predictors of order less than p. The algorithm also computes the minimum error, E_i at every step, which decreases as the order of prediction increases, i.e.

$$0 \leqslant E_i \leqslant E_{i-1}, \qquad E_o = R(0) \qquad 5.25$$

The intermediate quantities, k_i, are referred to as reflection coefficients, or partial correlation coefficients, and are always less than unity in magnitude, i.e.

$$-1 \leqslant k_i \leqslant 1, \quad 1 \leqslant i \leqslant p \qquad 5.26$$

Equation 5.26 has been shown [67] to be the necessary and sufficient condition for the all-pole filter to be stable, i.e. all the poles inside the unit circle. This is a major advantage of the autocorrelation method over the covariance in computing the predictor coefficients. As long as the autocorrelation coefficients are normalized, the Levinson-Durbin recursive algorithm always produces a stable filter, a condition which is not guaranteed in the covariance method. Apart from the autocorrelation and the covariance methods,

many other formulations of the linear prediction exist, e.g. maximum likelihood [66], lattice [59] and inverse filter methods [11].

5.2.5 The Gain of the Synthesis Model

From the transfer function of the synthesis model, given in equation 5.1, the output speech samples S(n) are related to the excitation signal U(n) as follows:

$$S(n) = \sum_{k=1}^{p} a(k)\, S(n-k) + G\, U(n) \qquad 5.27$$

Since the prediction error signal e(n) is defined as:

$$e(n) = S(n) - \bar{S}(n) = S(n) - \sum_{k=1}^{p} a(k)\, S(n-k) \qquad 5.28a$$

or

$$S(n) = \sum_{k=1}^{p} a(k)\, S(n-k) + e(n) \qquad 5.28b$$

If the speech samples are defined exactly by the model, then:

$$e(n) = G\, U(n) \qquad 5.29$$

i.e. the error signal is proportional to the excitation signal, where the constant of proportionality is the gain, G. In practice, it is not possible to solve for the gain, G, directly from the error signal. Instead, a reasonable assumption that the energy in the error signal is equal to the energy in the excitation signal, is made:

i.e. $$\sum_{n=0}^{N-1} e^2(n) = G^2 \sum_{n=0}^{N-1} U(n) \qquad 5.30$$

and that $$\sum_{n=0}^{N-1} U(n) = 1$$

Thus,

$$G^2 = E_p \qquad 5.31$$

5.2.6 Spectral Properties

The autocorrelation approach is directly suitable for the frequency domain interpretation of linear prediction and therefore will be used in the following discussion.

The total squared error, E, expressed in the time domain by equation 5.6, can be regarded as the output obtained by filtering the speech signal with an all-zero filter whose transfer function, A(z), is given by:

$$A(z) = 1 + \sum_{k=1}^{p} a(k)\, z^{-k} \qquad 5.33$$

Let $E(\omega)$ be the Fourier transform of the error signal, e(n), and $S(\omega)$ be the Fourier transform of the speech signal, S(n), in a given time interval.

Using Parseval's theorem:

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(\omega)|^2 \, d\omega = \frac{1}{2\pi} \int_{-\pi}^{+\pi} |S(\omega)|^2 \; |A(\omega)|^2 \, d\omega \qquad 5.34$$

Substituting, $H(\omega) = \frac{G}{A(\omega)}$ into equation 5.34, gives:

$$E = \frac{G^2}{2\pi} \int_{-\pi}^{+\pi} \frac{|S(\omega)|^2}{|H(\omega)|^2} d\omega \qquad 5.35$$

or, in terms of the power spectrum:

$$E = \frac{G^2}{2\pi} \int_{-\pi}^{+\pi} \frac{P(\omega)}{\hat{P}(\omega)} d\omega \qquad 5.36$$

where $P(\omega)$ is the power spectrum of the speech signal and $\hat{P}(\omega)$ the power spectrum of the model defined by $H(z)$.

Thus, minimizing the total square error in the frequency domain, is equivalent to minimizing the integrated ratio of the signal spectrum, $P(\omega)$, to its approximation, $\hat{P}(\omega)$. The total square error is large when $\hat{P}(\omega) < P(\omega)$, and small, for $\hat{P}(\omega) > P(\omega)$. Since the power spectrum contains resonances at the formant frequencies, it means that for a quasi-periodic signal the spectral approximation is far superior at the harmonics than between harmonics. These properties are illustrated in Figure 5.2, which shows the signal spectrum of a vowel sound modelled by a 28-pole linear prediction spectrum. The original signal spectrum was obtained by an FFT analysis on a 25.6 msec segment of voiced speech.

It can be shown [see Appendix B], that the autocorrelation coefficient of the speech segment and the autocorrelation coefficient of the impulse response corresponding to the system function $H(z)$, are equal for the first $(p+1)$ values. As $p \to \infty$, all the autocorrelation coefficients are equal, and leads to the following relationship:

$$\lim_{p \to \infty} |H(\omega)|^2 = |S(\omega)|^2 \qquad 5.37$$

This means that, for a large value of p, the signal spectrum is closely approximated by the all-pole model function, $H(z)$, with an

arbitrary small error. This is illustrated in Figure 5.3, which shows the original speech segment input signal, its FFT derived spectrum and the linear prediction spectrum for various values of the prediction order p.

5.2.7 Limitation of Linear Predictive Analysis

In the linear predictive analysis, it is assumed that all the speech sounds can be generated by exciting the all-pole filter with either quasi-periodic pulses or random noise. This means that nasal sounds and voiced fricatives cannot be suitably modelled using the all-pole model. Fortunately, human perception is more sensitive to the location of resonances than anti-resonances and so the synthesized speech is generally acceptable as of good quality.

The normalized prediction error, V_p, defined in equation 5.12b, has been shown to be dependent on the shape of the model spectrum [67], by expressing it as:

$$V_p = \frac{\exp\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} \log \hat{P}(\omega)\, d\omega\right]}{\frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{P}(\omega)\, d\omega} \qquad 5.38$$

where $\hat{P}(\omega)$ is the power spectrum defined by the linear prediction model.

V_p can be seen as a measure of the spectral flatness of the model spectrum. It attains a maximum value of 1 if $\hat{P}(\omega)$ is flat, and tends to zero when $\hat{P}(\omega)$ exhibits large fluctuations.

Thus, the spectrum of a voiced speech segment, which is characterized by resonances at a number of formant frequencies, is well modelled by linear prediction.

Conversely, the spectrum of unvoiced sounds tend to be flat, and the prediction error becomes higher. Of course, in both cases the

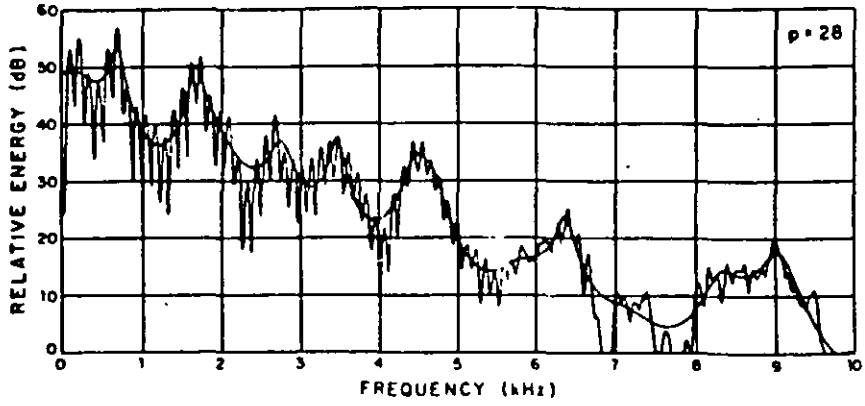


FIGURE 5.2: A 28 POLE FIT TO AN FFT SIGNAL SPECTRUM [60]

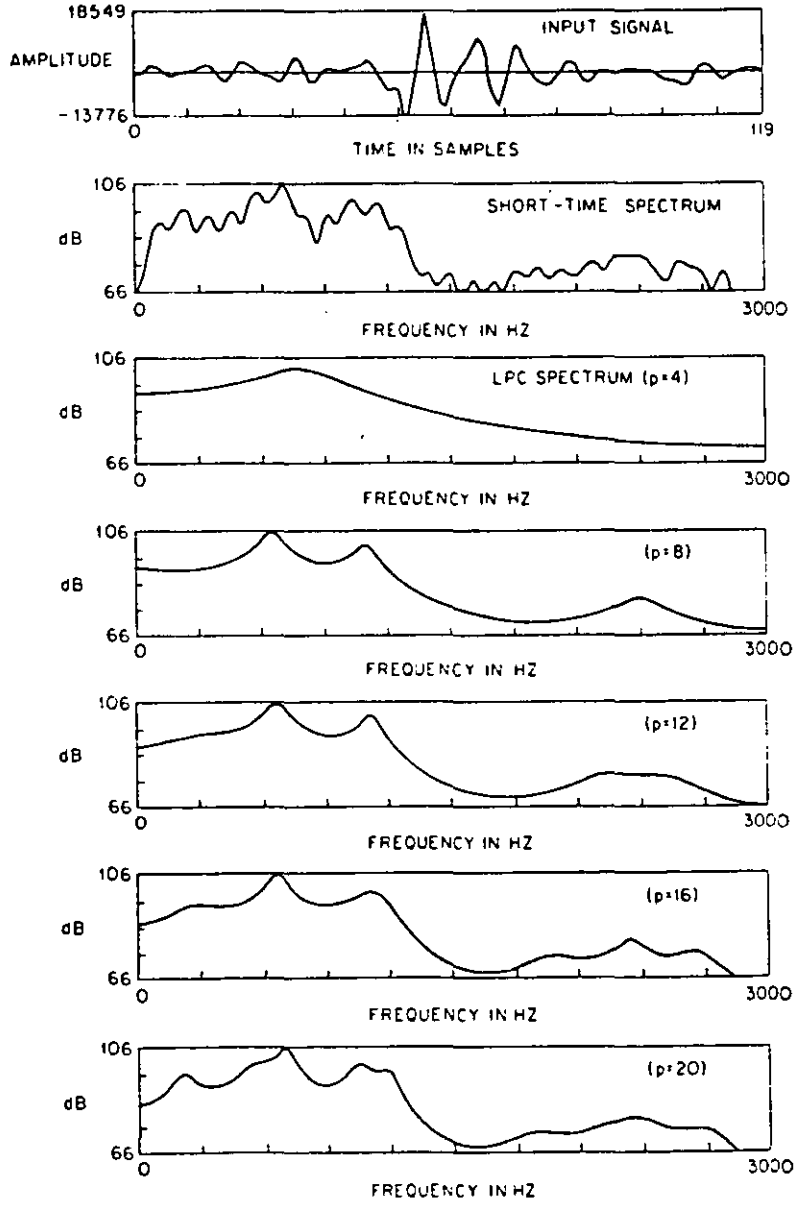


FIGURE 5.3: SPECTRA FOR /a/ VOWEL SAMPLED AT 6 kHz FOR SEVERAL VALUES OF PREDICTOR ORDER, p [59]

prediction error is also a factor of the order of prediction, p. The variation of the normalized prediction error V_p, as a function of p, for both voiced and unvoiced sounds is shown in Figure 5.4.

The fact that linear prediction spectral estimation of a quasi-periodic speech signal, is far more accurate at the harmonics than between harmonics, results in a better model for male speech than female speech. In female speech, the spectral harmonics are further apart than in male speech due to the higher pitch, thus giving a poorer fit. Children's speech results in even less accurate spectral estimation because the pitch frequency is much higher.

5.2.8 Extracting LPC Coefficients for Speech Recognition

The short-time power spectrum has been used as one of the main features in the description of speech segments. Since linear prediction coefficients give a good estimate of the short-time spectrum, their use in speech recognition becomes very attractive. The manner in which the LPC parameters are extracted from a speech utterance is illustrated in the block diagram of Figure 5.5.

The digitized speech utterance, S(n), is first pre-emphasized by using a first order non-recursive filter with a transfer function, $H(z) = 1 - az^{-1}$, to obtain the signal, $S'(n) = S(n) - a\,S(n-1)$. A suitable estimate of the pre-emphasis factor, 'a', is given by the ratio, R(1)/R(0), [60]. The aim of pre-emphasis is to reduce the spectral dynamic range of the signal.

In the next stage, the signal is segmented into frames, each of N samples. The temporal length of the frames should be of the order of a few pitch periods in the speech signal. Typical frame sizes range from 15 msec to 50 msec which correspond to values of N from 150 to 500 samples, at a sampling range of 10 kHz. Consecutive frames are spaced M samples apart. When $0 < M < N$, there is an overlap of the blocks.

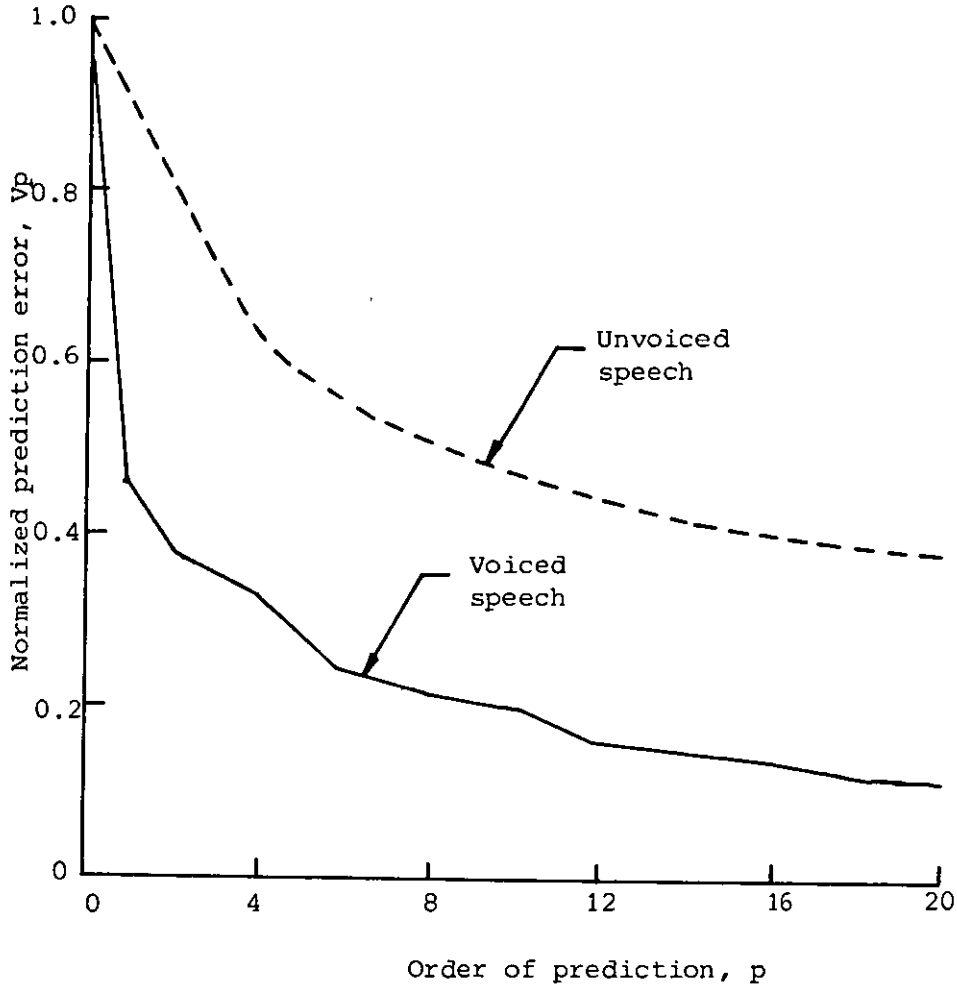


FIGURE 5.4: VARIATION OF THE NORMALIZED PREDICTION ERROR WITH THE ORDER OF PREDICTION, p [60]

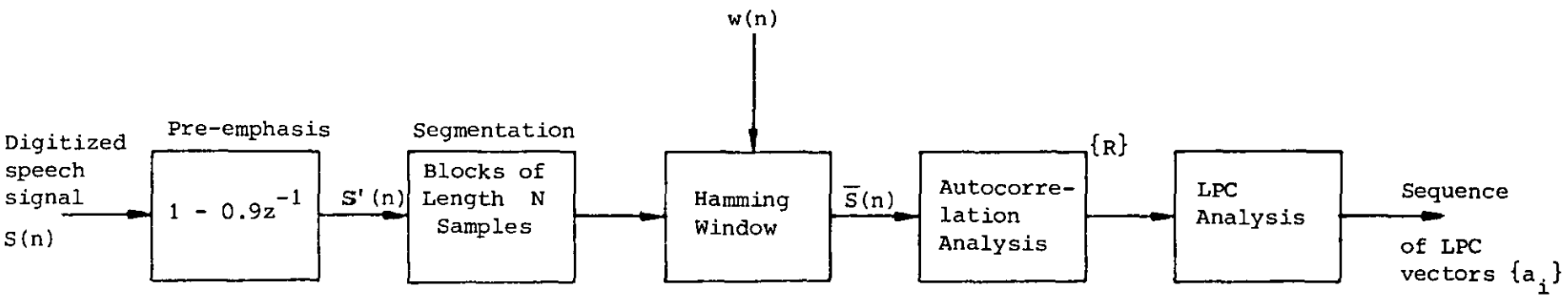


FIGURE 5.5: EXTRACTION OF LPC FEATURES FOR RECOGNITION

The speech data in each block is windowed by a function, w(n), so as to gradually taper the samples to zero, starting from the centre of the frame and proceeding towards the frame edges. A frequently used window in LPC analysis, is the Hamming window, given by:

$$w(n) = \begin{cases} 0.54 - 0.46 \cos(2\pi n/N-1), & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \qquad 5.39$$

Autocorrelation coefficients, $R(\ell)$, $\ell = 0, 1, \ldots, p$, are computed from the Hamming windowed speech samples, using the relation:

$$R(\ell) = \sum_{n=0}^{N-1-\ell} \frac{S(n)\, S(n+\ell)}{N}; \quad 0 \leq \ell \leq p+1 \qquad 5.40$$

where p is the order of prediction.

The autocorrelation coefficients are usually normalized by the zeroth delay autocorrelation coefficient, R(0). Levinson and Durbin's recursive algorithm can then be employed to derive the values of linear prediction coefficients of the desired order.

In this manner, the speech signal is reduced to a discrete sequence of LPC vectors which describe the short-time spectral shape of the signal.

5.3 DISTANCE MEASURES FOR LPC COEFFICIENTS

The use of linear prediction in speech recognition has found wide acceptance since Itakura and Saito [66][67] first proposed a suitable distance measure in the comparison of two speech frames expressed in the LPC domain. To be useful, a distance measure d(X,Y) between the speech frames X and Y should satisfy the following properties:

i) d(X,Y) should be positive definite, i.e.

$$d(X,Y) > 0 \text{ for } X \neq Y$$
$$d(X,X) = 0 \qquad 5.41$$

and be subjectively meaningful in the sense that small and large distance measures correspond to similarity and dissimilarity respectively.

ii) d(X,Y) should have a physically meaningful interpretation in the frequency domain.

iii) The distance measure should be efficiently computable.

The conventional squared error and the absolute norm distance measures discussed in Chapter 4 do not appear to be subjectively meaningful when applied to LPC coefficient sets. For this reason, a number of distance measures which have meaningful frequency domain interpretation have been proposed [67][68][69] and some are briefly discussed below.

5.3.1 The Log Spectral Measure [69][70]

Consider two spectral models,

$$G/(1 + \sum_{k=1}^{p} a(k)\, z^{-k}), \text{ and } \bar{G}/(1 + \sum_{k=1}^{p} \bar{a}(k)\, z^{-k})$$

The spectral difference, $V(\theta)$, between these models on a log magnitude versus frequency scale is given by:

$$V(\theta) = \log_e \{G^2/|1 + \sum_{k=1}^{p} a(k)e^{-j\theta k}|^2\} - \log_e \{\bar{G}^2/|1 + \sum_{k=1}^{p} \bar{a}(k)e^{-j\theta k}|^2\} \qquad 5.42$$

where θ is the frequency on a scale normalized by the factor $F_s/2\pi$. F_s is the sampling frequency.

One set of logical choices for a measure of distance based on $V(\theta)$, is the L_p norms defined by d_p, where,

$$d_p = \left\{ \int_{-\pi}^{+\pi} |V(\theta)|^p \frac{d\theta}{2\pi} \right\}^{1/p} \qquad 5.43$$

When a value of p=1 is used, d_p defines the absolute log spectral measure. For p=2, the rms log spectral measure is defined and for p approaching infinity the peak log spectral measure is obtained.

The L_p measures exhibit linearity, in the sense that multiplication of $V(\theta)$ by a scalar constant results in a multiplication of d_p by the same constant. In addition, the L_p measures are symmetric and positive definite. However the main problem with the above distance measures is the computational load required to obtain sufficient values of $V(\theta)$ in order to approximate the integral in equation 5.43 by a summation.

5.3.2 The Itakura-Saito Distance Measure [67]

Let $P_T(\omega)$ and $P_R(\omega)$ be two power spectra of a test and a reference speech frame described respectively by the LPC sets, $a_T = \{1, a(1), a(2), \ldots a(p)\}$ and $a_R = \{1, \bar{a}(1), \bar{a}(2) \ldots, \bar{a}(p)\}$. Then, the Itakura-Saito distance, d_{IS}, between the two spectra is defined as:

$$d_{IS}(P_T, P_R) = \int_{-\pi}^{+\pi} \left[\frac{P_T}{P_R} - \log_e \left(\frac{P_T}{P_R}\right) - 1\right] \frac{d\omega}{2\pi} \qquad 5.44$$

The theoretical significance of d_{IS}, comes from the formulation of linear prediction as an approximate maximum likelihood estimation. Since of concern here is the power spectrum estimates of P_T and P_R which are of the form:

$$P_T = \frac{G^2}{|1 + \sum_{k=1}^{p} a(k) z^{-k}|^2} \quad \text{and} \quad P_R = \frac{\bar{G}^2}{|1 + \sum_{k=1}^{p} \bar{a}(k) z^{-k}|^2} \qquad 5.45$$

Equation 5.44 can be expressed as [see Appendix C]:

$$d_{IS}(P_T\ P_R) = \frac{\alpha}{\bar{G}^2} + \log_e(\bar{G}^2) - \log_e(G^2) - 1 \qquad 5.46$$

where $\alpha = r(0)\ r_a(0) + 2 \sum_{n=1}^{p} r(n)\ \bar{r}_a(n)$

and $r_a(n) = \sum_{i=0}^{p-n} \bar{a}(i)\ \bar{a}(i+n)$

and where $r(n)$ are the time-domain autocorrelation coefficients of $P_T(\omega)$.

For a given power spectrum, $P(\omega)$, and a scaled version of itself $\lambda P(\omega)$, the Itakura-Saito distance between the two spectra, as defined in equation 5.44, is simplified to:

$$d_{IS}(P, \lambda P) = \frac{1}{\lambda} + \log_e \lambda - 1 \qquad 5.47$$

Thus d_{IS} is a gain sensitive distance measure, a characteristic that is completely undesirable in the comparison of speech frames. However, two gain insensitive versions of d_{IS} are available and are referred to as the gain optimized and the gain normalized measures.

i) The gain optimized Itakura-Saito distance measure:
The gain optimized Itakura-Saito distance measure, d_{GO}, is given by:

$$d_{GO}(P_T, P_R) = \underset{\lambda \to 0}{\text{MIN}}\ d_{IS}(P, \lambda P) \qquad 5.48$$

$$= \log_e \int_{-\pi}^{\pi} \frac{P_T}{P_R} \frac{d\omega}{2\pi} - \int_{-\pi}^{\pi} \log_e \left(\frac{P_T}{P_R}\right) \frac{d\omega}{2\pi} \qquad 5.49$$

For an all-pole power spectra, equation 5.49 can be expressed as,

$$d_{GO}(P_T, P_R) = \log_e(\alpha) - \log_e(G^2) \qquad 5.50$$

ii) The gain-normalized Itakura-Saito distance measure:
The gain normalized Itakura-Saito distance, d_{GN}, for spectra of the all-pole form, is defined as:

$$d_{GN}(P_T, P_R) = d_{IS}\left(\frac{P_T}{G^2}, \frac{P_R}{\bar{G}^2}\right) \qquad 5.51$$

$$= \frac{\alpha}{G^2} - 1 \qquad 5.52$$

The three distance measures are inter-related as follows:

$$d_{GO} = \log_e(1 + d_{GN}) \qquad 5.53a$$

$$d_{GO} = \log_e\left[\frac{\bar{G}}{G^2}\left(d_{IS} + \log_e\frac{G^2}{\bar{G}^2} + 1\right)\right] \qquad 5.53b$$

An interpretation of the Itakura-Saito distance measure:
The gain normalized Itakura-Saito distance measure, d_{GN} can also be rewritten in matrix form as follows [see Appendix C]:

$$d_{GN}(P_T, P_R) = \frac{[\bar{a}][R_T][\bar{a}]^t}{[a][R_T][a]^t} - 1 \qquad 5.54$$

where $[R_T]$ is the autocorrelation coefficient matrix of the test frame speech data, and $[a]^t$ denotes the transposed vector of $[a]$. An interpretation of the distance measure is illustrated in Figure 5.6.

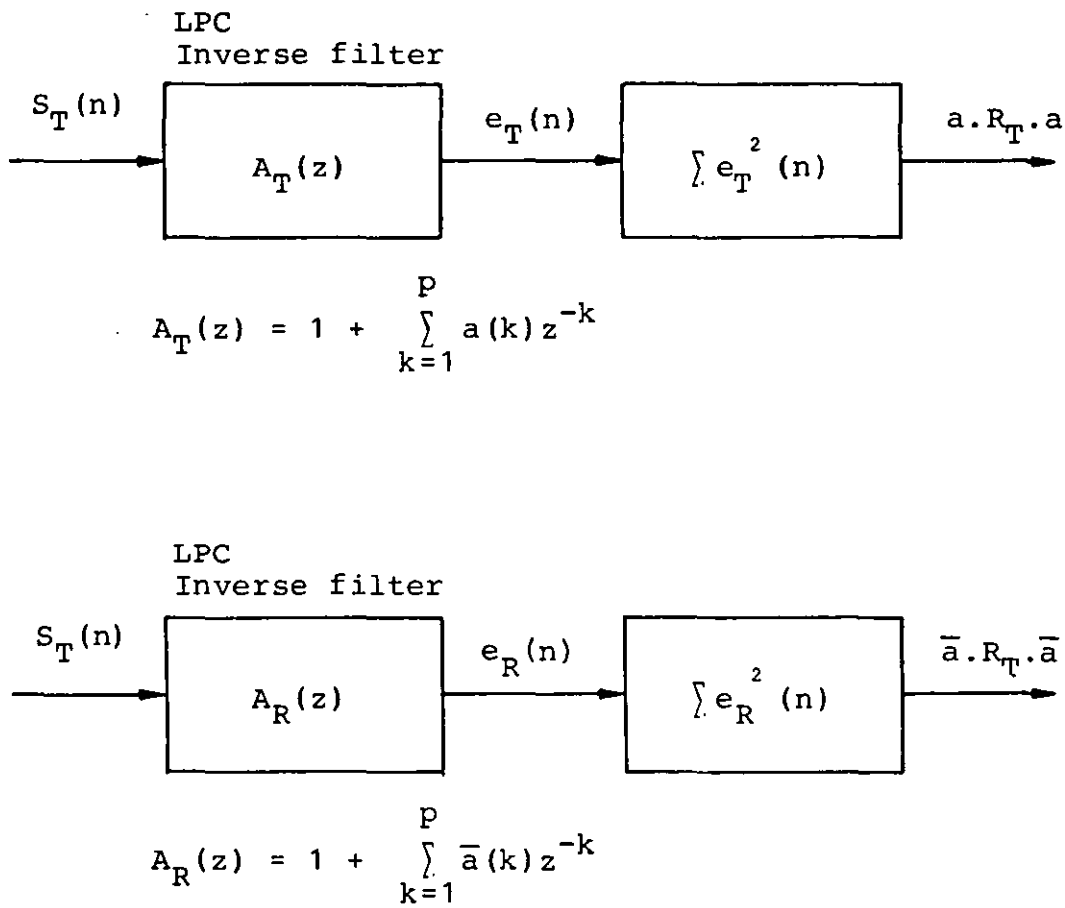


FIGURE 5.6: AN INTERPRETATION OF THE ITAKURA-SAITO DISTANCE MEASURE

If a test speech frame, $S_T(n)$, is passed through its own inverse LPC filter $A_T(z)$, the energy of the error signal, $e_T(n)$ is equal to G^2 (equation 5.30), which can also be expressed in matrix form as $[a][R_T][a]^t$ (see Appendix C). If the same test speech frame is passed through the inverse LPC system described by the parameters of the reference speech frame, the energy of the error signal, $e_R(n)$, will be given by $[\bar{a}]\,[R_T][\bar{a}]^t$. The minimum energy in the error signal will occur only when the signal is passed through its own inverse LPC system, since the LPC parameters are optimized for the frame.

The ratio, $[\bar{a}][R_T][\bar{a}]^t/[a][R_T][a]^t$, thus defines a measure of difference between test and reference speech frames on their spectra. The ratio is equal to unity only when the two frames are identical, otherwise it is always greater than unity.

Some criticisms have been made on the three forms of Itakura-Saito distance measure; especially that they do not satisfy the properties of a true metric,

i.e. $$d_{IS}\,(X,Y) \neq d_{IS}\,(Y,\,X) \qquad 5.55$$

It has been shown by de Souza [71], that the Itakura-Saito distance measure is not a χ^2-distribution with p degrees of freedom (where p is the order of prediction), and therefore it is not optimal as a test statistic. However, despite these objections, the Itakura-Saito distance measures have been used in many practical applications with excellent results.

5.4 THE LPC-BASED WORD RECOGNITION SYSTEM

The block diagram of a conventional word recognition system based on the LPC analysis is shown in Figure 5.7. An input speech utterance, S(n) is passed through a pre-emphasis network with transfer function, $1-0.90z^{-1}$. After an autocorrelation analysis on 25.6 msec Hamming

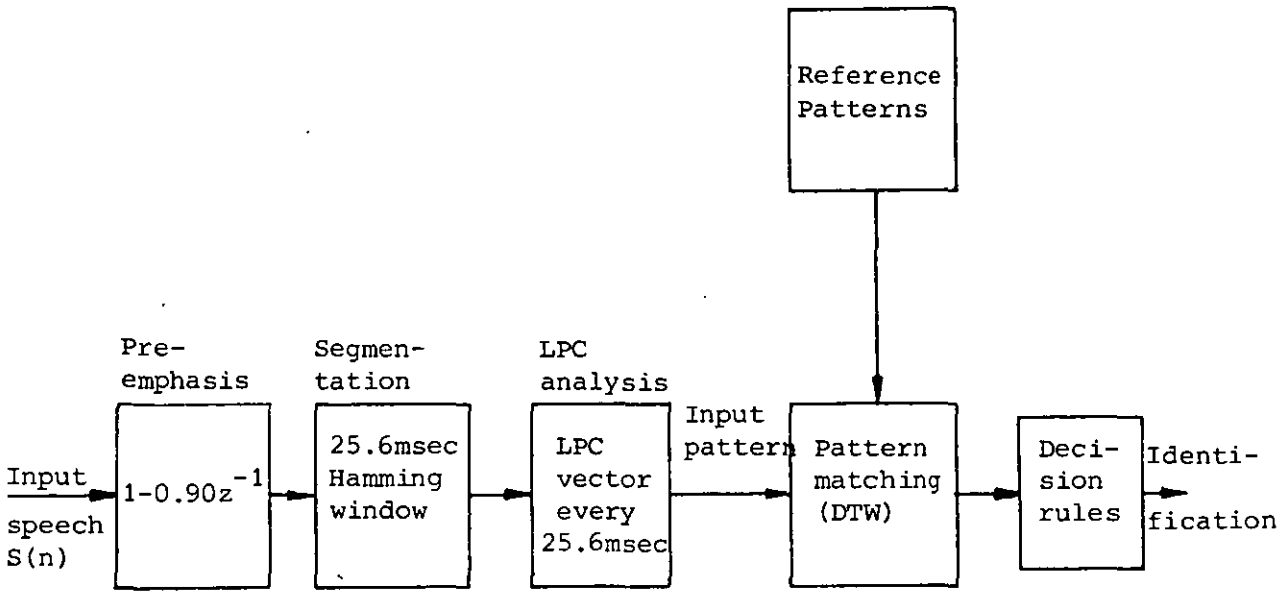


FIGURE 5.7: THE BLOCK DIAGRAM OF AN LPC-BASED WORD RECOGNIZER

windowed speech segments, LPC coefficients are extracted once every 25.6 msec, employing the Levinson-Durbin's recursive algorithm. The input word is thus expressed as a discrete pattern of LPC vectors. The recognition task requires the input word pattern to be matched with a pre-stored set of reference vocabulary word patterns. These reference patterns are generated a priori during a training session. The dynamic time warping technique proposed by Sakoe and Chiba, including Paliwal's modification, as discussed earlier in Section 3.2, is employed to obtain a time normalized distance between the input word pattern and each reference pattern. The gain-normalized Itakura-Saito distance measure was used for the local distance between the frames of the patterns. In order to achieve speaker independent performance, each vocabulary word is represented by multiple reference patterns. Hence, the input word is identified from the vocabulary using the k nearest neighbour (kNN) rule, discussed earlier in Section 4.4 of Chapter 4.

i) **Training procedure**

In the training session, reference patterns of the vocabulary words are generated by an LPC analysis. Each vocabulary word is represented by patterns formed from the repetitions of the word by four speakers.

ii) **Recognition results**

A series of tests were carried out in order to investigate the performance of the recognizer in correctly identifying input words. The input words to the recognizer were taken from a speaker who did not contribute to the generation of reference patterns. Each of the 50 vocabulary words, was represented by four reference patterns while the input word was identified using the kNN rule, with k=3. The results obtained on the recognition accuracy, as a percentage of correct identifications of the input words, are given in Table 5.1. The order of prediction, p, was varied from 6 to 14.

TABLE 5.1

PERFORMANCE OF THE LPC-BASED WORD RECOGNIZER WITH VARYING PREDICTION ORDERS

	RECOGNITION ACCURACY (%)			
Order of Predictor p	Test 1 Test speaker SM1 Ref speakers: SM2, SM4, SF2, SM3	Test 2 Test speaker SM3 Ref speakers: SM2, SM4, SF2, SM1	Test 3 Test speaker SF1 Ref speakers: SM2, SM4, SF2, SM3	Average
p = 6	74	64	60	66.0
p = 8	78	70	64	70.6
p = 10	84	80	72	78.6
p = 12	90	88	84	87.3
p = 14	94	90	84	89.3

5.4.1 The Use of Discriminative Patterns

Although the dynamic time warping algorithm achieves considerable success in word recognition, its performance is ultimately limited by its poor ability to discriminate between acoustically similar words. The problem arises because all local differences between a test and a reference pattern are assumed to be of equal importance. For acoustically similar words, some local differences are crucial to the correct identification of the input word, whereas some other local differences are irrelevant. For example, in the vocabulary under consideration, the words set {B, C, D, E, G, P, T, V} have all got a common ending sound /e/, and differ only in their initial regions. If in the recognition of a word 'B', the ending /e/ sound happens to be more similar to the /e/ region in the reference pattern 'E', than the /e/ region in the reference pattern 'B', then it is quite possible for a misrecognition to occur. A pattern matching technique in which attention is focussed on those regions in the pattern that serve to distinguish it from similar words, would provide a solution to this problem.

Moore et al [72], have proposed the following method of re-structuring the reference patterns, so that similar regions are represented by common frames. Consider a speech pattern A, of I frames, and a pattern B, of J frames,

i.e. $A = \{a_1, a_2, \ldots, a_i, \ldots a_I\}$ and $B = \{b_1, b_2, .., b_j, \ldots, b_J\}$

and the distances d(i,j), along the optimal time registration path obtained by a DTW procedure.

If patterns A and B represent different words which have some similar sounding regions, then some distances d(i,j), will be large and others small. A probability distribution of these distances can be obtained by using a large training set of the word pairs. The local

distance d(i,j) can then be replaced with a probabilistic measure, L(i,j/d), which gives the likelihood that frame a_i belongs to the class of frame a_j, given the distance d(i,j). From the variation of L(i,j/d) along the time registration path, a suitable threshold, below which frames can be considered to be similar, is determined. Thus, the similar frames in pattern A can be replaced with the corresponding frames in pattern B. Moore et al [79], successfully used this approach to discriminate between the acoustically similar set of words pairs, {FIVE, NINE} {K, J}, {B, D}, {D, T}, {STALACTITE, STALAGMITE} {RIDER, WRITER}. They reported a reduction in recognition error rate, from 26.8% to 7.0%, on employing the discriminative reference patterns.

However, the above procedure requires a large training set of the acoustically similar words, in order to obtain the probability distribution of the inter-frame distances. In the absence of such data, an approximate procedure is used here, in which the distances d(i,j), of one word pair, are used to identify the similar regions.

i) Training Session

The reference patterns of the words set {B, C, D, E, G, P, T, V}, obtained from the same speaker were considered. Using Myer's algorithm, each word pattern was normalized to a length equal to that of word pattern 'E'. The Sakoe-Chiba asymmetric DTW algorithm with a gradient constraint, P=1, was employed to obtain the inter-frame distances along the warping path, between each word pattern and the word pattern 'E'. Figure 5.8 depicts the distances along the time registration path, obtained with the above words set, uttered by the male speaker SM1. A threshold level is set by observation since the end regions of the word pair are expected to display strong similarity. The use of the symmetric DTW process, discussed in Chapter 3, ensures that consecutive points on the optimal time registration path, correspond to different frames in pattern E.

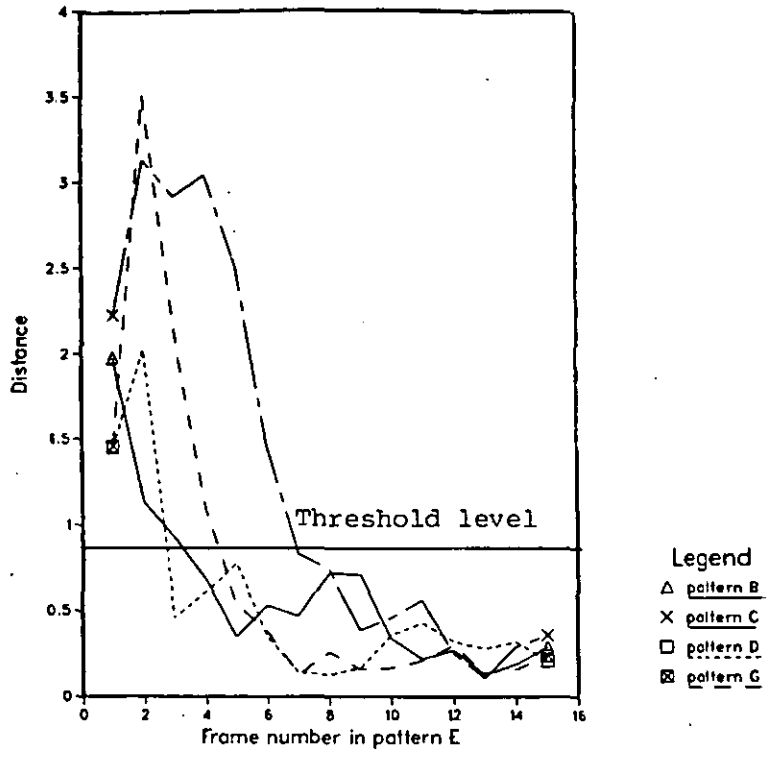


FIGURE 5.8(a): LOCAL DISTANCES ALONG THE OPTIMAL PATH, FOR THE WORD PATTERN PAIRS {B,E}, {C,E}, {D,E} AND{G,E}

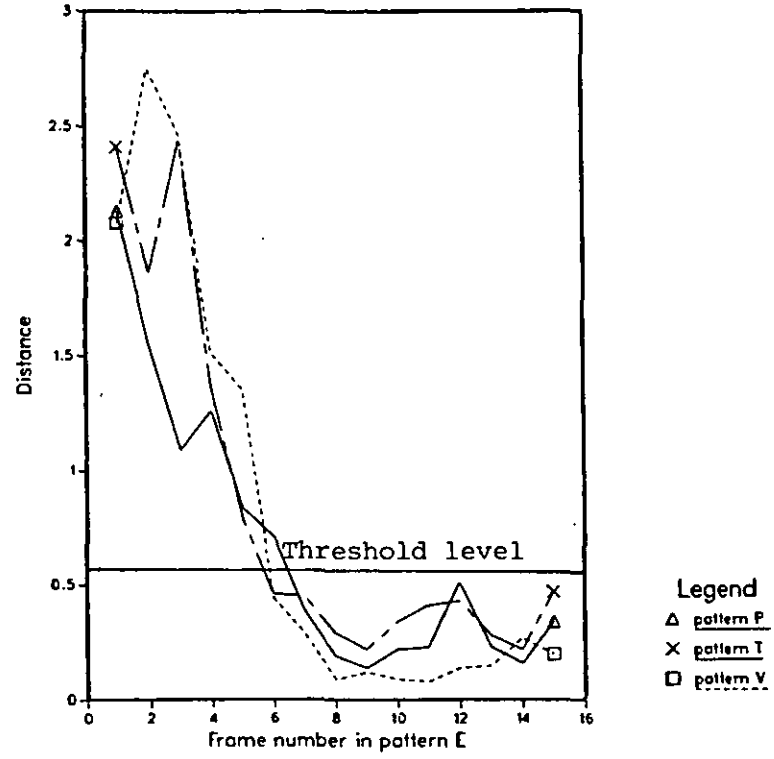


FIGURE 5.8(b): LOCAL DISTANCES ALONG THE OPTIMAL WARPING PATH, FOR THE WORD PATTERN PAIRS {P,E}, {T,E} AND {V,E}

ii) **Results**

Tests were carried out to assess the influence of the reference pattern structure discussed above, on the word recognition accuracy. The input words in each test were obtained from a speaker who did not contribute to the reference pattern generation. Each vocabulary word was represented by four reference patterns. The results obtained are given in Table 5.2, as a percentage of correct identification of the input words. For comparison purposes, the results obtained without using the discriminative reference patterns are also given in the same table.

5.4.2 A Computation Cost Reduction Method

The error rate of a speaker independent isolated word recognition system can be decreased by using reference vocabulary patterns which reflect the inter-speaker variations for a given word. This is achieved when each vocabulary word is represented by multiple patterns of the same word uttered by different speakers. Such an approach, while improving the recognition error rate, results in a huge increase in computation, since an input utterance to be classified must be compared with a greatly increased number of reference patterns, as opposed to the case of a single reference pattern per vocabulary word system.

It would be of interest, therefore, to reduce the computational load in the pattern matching stage of a recognition system by limiting the number of reference patterns which are compared with the input utterance. This can be achieved by using a clustering procedure which partitions the reference patterns of the vocabulary words into a small number of disjoint groups. For each group, a representative pattern, termed a cluster centroid, is determined. The input utterance to be identified, is first compared with all the cluster centroids, and then only with the reference patterns associated with the closest centroid. The reduction in computational cost of the proposed recognition system is dependent on the number of clusters and their occupancy.

TABLE 5.2

THE PERFORMANCE OF THE LPC-BASED WORD RECOGNIZER USING DISCRIMINATIVE REFERENCE PATTERNS FOR THE WORDS SET {B, C, D, E, G, P, T, V}

	RECOGNITION ACCURACY (%)			
Recognition System (Predictor order, p=14)	TEST 1 Test speaker:SM1 Ref speakers:SM2 SM4, SF2, SM3	TEST 2 Test speaker:SM3 Ref speakers:SM2 SM4, SF2, SM1	TEST 3 Test speaker:SF1 Ref speakers:SM2 SM4, SF2, SM3	Average
With discriminative reference pattern for the set {B,C,D,E, G,P,T,V}	94	92	88	91.3
Without discriminative patterns	94	90	84	89.3

Wilpon and Rabiner [73], have recently proposed a Modified K Means (MKM), algorithm for use in the selection of a small number of reference patterns from a large training set composed of repetitions of the same word by different speakers. Their aim was to obtain a small set of patterns which represent the major diversities of the vocabulary word. These patterns are then employed to enhance the speaker independent performance of a recognition system. Here, the MKM algorithm is employed to solve a different problem. By clustering all the reference patterns of the entire vocabulary, into small disjoint groups, a computationally faster recognition process can be realized. The algorithm can be described in the following steps.

Step 1: Given: A set $W = \{W_1, W_2, \ldots, W_V\}$ of V isolated words of different temporal lengths, and each word is a discrete sequence of LPC vectors. The distance matrix of the entire set, $D(W_i, W_j)$, $1 \leqslant i$ $j \leqslant V$, is computed using a DTW process. The objective is to cluster the entries in W into M disjoint groups such that words within each group exhibit a certain level of similarity. Set the convergence check parameter, D_{conv}, to a large value.

Step 2: Find the two entries W_K and W_L which are most dissimilar. Set N, the number of clusters to 2 and let the two initial cluster centroids $\bar{W}^1$ and $\bar{W}^2$ be W_K and W_L.

Step 3: Classify each entry in W, to the nearest of the previously defined centroids.

Step 4: For each cluster, find the pattern whose maximum distance to any other pattern within the cluster is minimum i.e. the minimax centre given by:

$$\underset{1 \leqslant L \leqslant p^m}{\text{MAX}} \; D(\bar{W}^m, W_L^m) = \underset{1 \leqslant i \leqslant p^m}{\text{MIN}} \left[\underset{1 \leqslant L \leqslant p^m}{\text{MAX}} \; D(W_i^m, W_L^m) \right] \qquad 5.56$$

where p^m is the occupancy of cluster m and $\bar{W}^m$ is the minimax centre and also the centroid. $W_i^m \varepsilon$ cluster m.

Step 5: Compute the average intracluster distance $\bar{D}^m$ for each cluster and the average intercluster distance, D

$$\bar{D}_m = \frac{1}{p^m} \sum_{L=1}^{p^m} D(\bar{W}^m, W_L^m) \qquad 5.57a$$

and

$$D = \sum_{m=1}^{N} (\bar{D}_m/N) \qquad 5.57b$$

Step 6: If $(D_{conv} - D)/D_{conv} > 0.05$, reset D_{conv} to D, and exit if N= M, otherwise go to Step 3. If $(D_{conv}-D)/D_{conv} \leqslant 0.05$, continue.

Step 7: Set N = N+1. Identify the largest cluster g, $1 \leqslant g \leqslant N-1$. The centroid of the new cluster W^N is the pattern of cluster g furthest from the centroid $\bar{W}^g$. Go to step 3.

A flowchart illustrating the above steps in the algorithm, is shown in Figure 5.9.

i) Testing Session:

Using the MKM clustering algorithm, the reference vocabulary is clustered into M disjoint groups, whose centroids are the set of $\{\bar{W}^1, \bar{W}^2, \ldots, \bar{W}^M\}$.

An unknown input word, X, is then compared with this set of the centroids and classified into the cluster associated with the nearest centroid, i.e.

$$\text{Classify X in L if } D(X, \bar{W}^L) = \underset{1 \leqslant L \leqslant M}{\text{MIN}} \; D(X, \bar{W}_L) \qquad 5.58$$

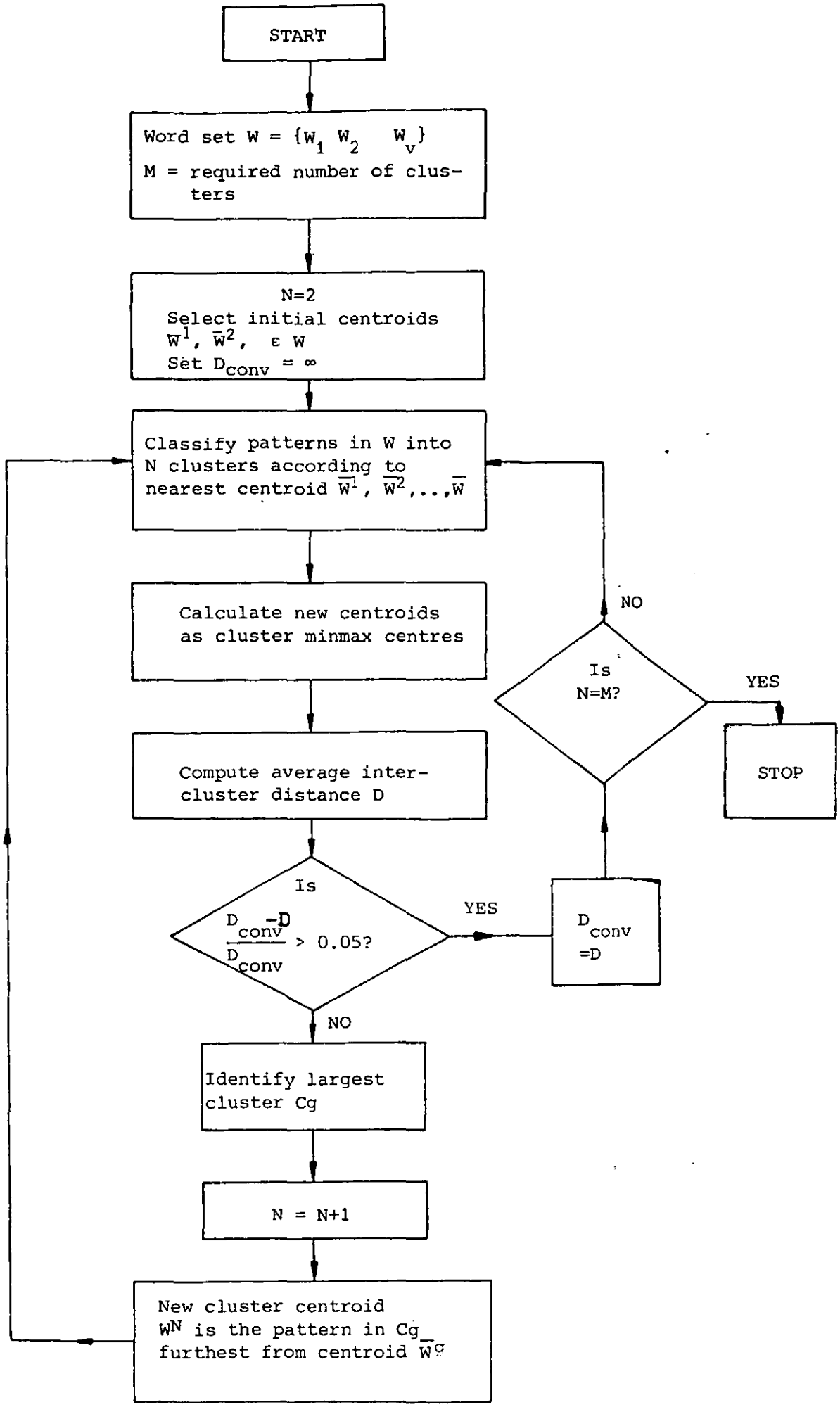


FIGURE 5.9: FLOW CHART FOR THE MKM ALGORITHM

Next, the pattern X, is compared with all the J entries in cluster L and is identified as the reference vocabulary word whose pattern gives the minimum distance.

i.e. Identify X from

$$\underset{1 \leq j \leq J}{\text{MIN}} \quad D(X, C_j^L), \; C_j^L \; \epsilon \text{ cluster L} \qquad 5.59$$

The number of pattern comparisons needed for the identification of X are (M+J), and is thus dependent on the cluster occupancy.

In general, the highest number of pattern comparisons that need to be carried out in identifying an input word, is given by $(M+J_L)$, where J_L is the size of the largest cluster. Thus, the reduction in computational cost, obtained on using the above clustering process on reference patterns, is given by the ratio, $(M+J_L)/V$. Also the computation involved in the MKM clustering procedure is considered as negligible overhead cost since it is only done once during the training session of the recognizer.

ii) Results

The training set for the MKM clustering algorithm was composed of 150 patterns which were derived from repetitions of each vocabulary word by the speakers SM3, SM4 and SF2. Figure 5.10 illustrates the properties of the clusters generated in terms of their average intercluster distance.

The speech utterances from the speakers SM1, SM2 and SF1 were used to test the performance of the recognition system with 4, 6, 8, 10, 12 and 15 reference pattern clusters. Figure 5.11 illustrates the recognition error rate and the computational cost reduction obtained using different numbers of clusters for the reference patterns, as compared with the recognition system where the reference patterns are not clustered.

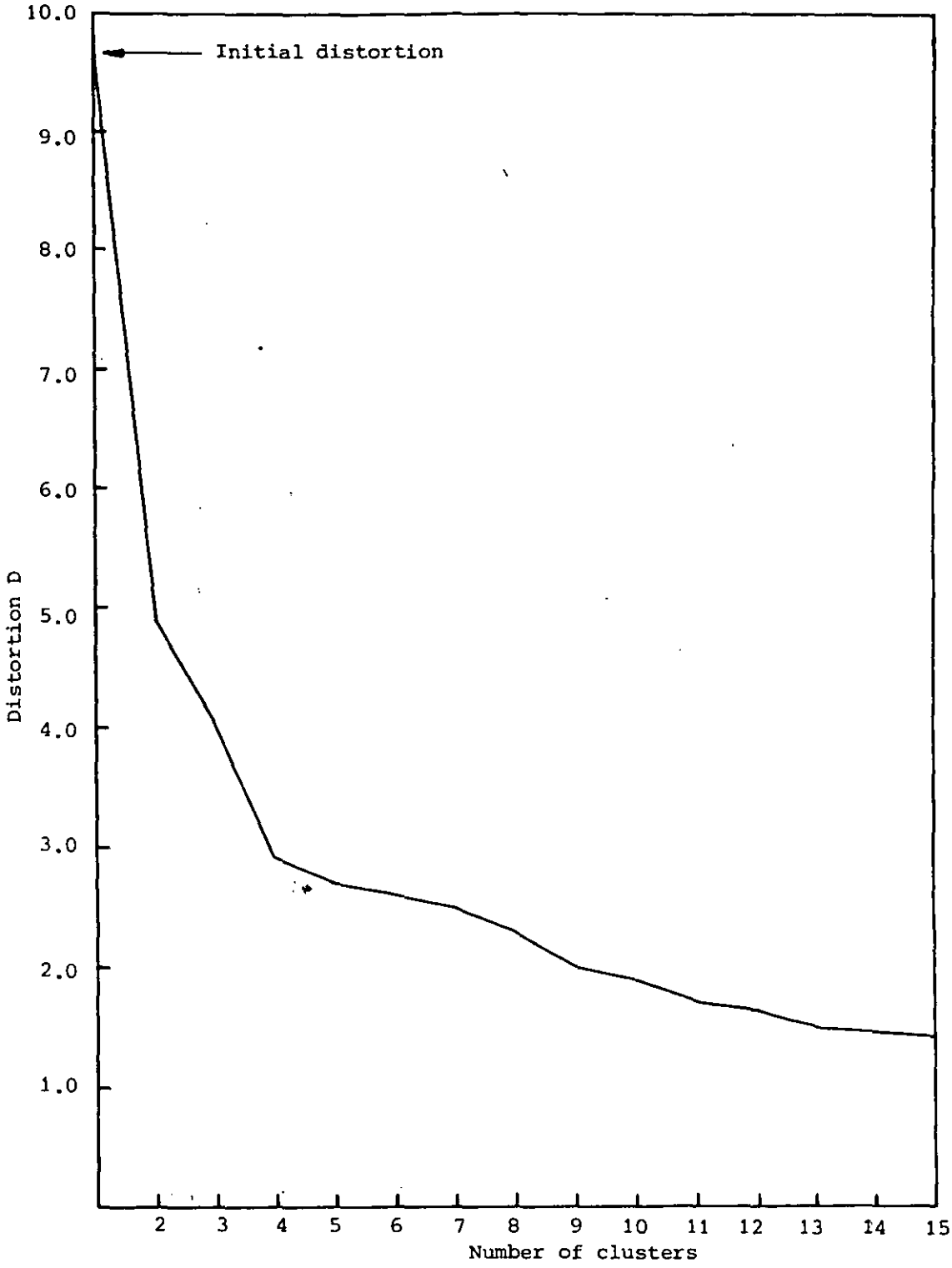


FIGURE 5.10: AVERAGE INTERCLUSTER DISTORTION D, AS A FUNCTION OF THE NUMBER OF CLUSTERS

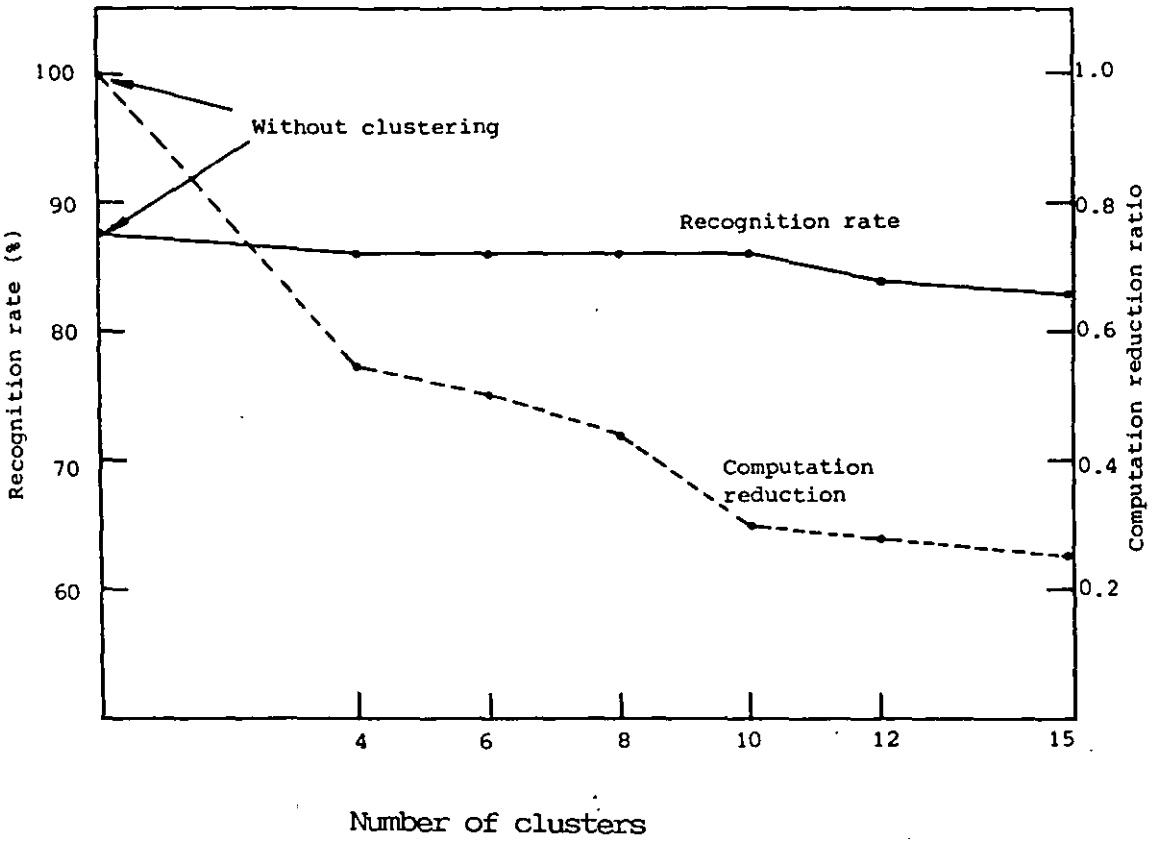


FIGURE 5.11: RECOGNITION RATE AND THE COMPUTATION REDUCTION RATIO VERSUS THE NUMBER OF CLUSTERS IN THE REFERENCE VOCABULARY

5.5 VECTOR QUANTIZATION IN WORD RECOGNITION

In the word recognition systems presented in the preceding sections, a large amount of memory is required to store the multiple reference patterns of the vocabulary words, which are employed in order to achieve speaker independence. These word recognizers also employ the computationally expensive DTW process in the pattern matching stage. As such, techniques which would reduce the memory requirements or eliminate the need for a DTW process, without seriously degrading the recognition accuracy, could be very useful. Vector quantization offers such a possibility, and is the subject of discussion in this section. Two word recognition systems, LPC/SPLIT and LPC/VQ, which employ VQ techniques to reduce memory requirements and to eliminate the need for the DTW process respectively are investigated. A hybrid system, the LPC/VQ/SPLIT, which combines the advantages offered by the two recognition systems is then proposed.

5.5.1 The Theory of Vector Quantization [74][75]

Vector quantization, (VQ), which was first applied to low bit rate coding of speech signals, is a fundamental result of Shannon's rate distortion theorem [76], which states that for a given rate or distortion function, a source can always be more accurately represented by coding vectors rather than scalars. Although Shannon's rate distortion theorem expounds the optimality of vector based coding, it does not provide an insight as to how such a system can be designed. Furthermore, the traditional scalar coders often yield satisfactory performance. As a result, few design techniques for vector coders were considered prior to the late 1970's, when it was found that a simple algorithm proposed by Lloyd [77] for the design of pulse code modulation systems, provided a suitable technique for the design of VQ Codebooks of data sources such as speech waveforms, speech parameter vectors, images etc. The main application of VQ has been in minimization of communication channel capacity and in the reduction of memory requirements for data storage, the latter

application being directly relevant to the speech recognition problem.

Vector quantization, can be defined simply as a system for mapping a group of similar vectors into a single entity.

Let $T = \{t_1, t_2, \ldots, t_N\}$, be a large set of LPC vectors, obtained from the reference patterns of vocabulary words. The main idea behind vector quantization is to create an optimum set of LPC vectors $\{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_M\}$, $M \ll N$, termed a codebook, such that, for a given value of M, the error in replacing a vector in the set T by the closest entry in the codebook, is minimized.

The optimization problem can be expressed as:

$$D_N(M) = \underset{\{\hat{a}\}}{\text{MIN}} \left[\frac{1}{N} \sum_{i=1}^{N} \underset{1 \leq m \leq M}{\text{MIN}} \; d(t_i, \hat{a}_m)\right] \qquad 5.60$$

where $D_N(M)$ is the average distortion of the training set containing N LPC vectors when the codebook has M entries. $d(t_i, \hat{a}_m)$, is the gain-normalized Itakura-Saito distance between training set vector t_i and codebook entry $\hat{a}_m$.

Equation 5.60 can be solved efficiently by the so called binary splitting methods [78][79][80].

5.5.2 The Binary Splitting VQ Algorithm

There are two forms of the binary splitting algorithms, namely the full-search and the tree search algorithms.

i) **The full-search algorithm**

The full-search algorithm begins by finding an optimum solution for a codebook with two entries (i.e. M=2), and then splits each of the entries into two components; hence the name binary split. The

algorithm, computes the optimum solution for this 4 entries codebook, and continues the iterations until the codebook size, M, is as large as desired or until the rate of decrease in the average distortion $D_N(m)$ of the training set satisfies a predetermined threshold. The algorithm can be described in the following steps.

Step 1: Start with a training set, $T = \{t_1, t_2, \ldots, t_N\}$, of a large number of LPC vectors of the reference patterns.

Step 2: Select two vectors from the set T, to be the initial codebook entries, $\{\hat{a}_1, \hat{a}_2\}$, i.e. m=2. Set D_o, the initial average distortion of the training set to a large value.

Step 3: Compute the distance between each vector in T and the codebook entries. The average distortion $D_N(m)$, of the training set is given by:

$$D_N(m) = \frac{1}{N} \sum_{i=1}^{N} \underset{1 \leq m' \leq m}{\text{MIN}} \; d(t_i, \hat{a}_{m'}) \qquad 5.61$$

Step 4: If the decrease in average distortion $(D_o - D_N(m)/D_o) < \varepsilon$, set $D_o = D_N(m)$ and go to step 6. ε is a pre-set threshold.

Step 5: Update the codebook entries by clustering the vectors in the training set T, into m clusters. Each vector in $t_i \varepsilon T$, is assigned to cluster m' according to the nearest neighbour rule, i.e.

$$\text{Classify } t_i \text{ in cluster } m' \text{ if } d(t_i, \hat{a}_{m'}) = \underset{1 \leq m' \leq m}{\text{MIN}} \; d(a_i, \hat{a}_{m'})$$

where $\{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{m'}, \ldots \hat{a}_m\}$, are the m codebook entries.

Determine the centroid, $C_{m'}$, m' = 1,2, ... m of each cluster. $C_{m'}$ is the LPC vector corresponding to the average autocorrelation

coefficients of the vectors in cluster m'. Use the centroids as the new codebook entries. Go to step 3.

Step 6: Exit if the desired size, M, of the codebook has been achieved. Otherwise, split each centroid into two components by perturbing its elements by a small quantity, δ, i.e. the centroid, $\hat{a}_{m'}$, is split into two vectors $a_{m1} = (1+\delta)\hat{a}_{m'}$ and $a_{m2} = (1-\delta)\hat{a}_{m'}$. Set m=2m and go to step 3.

In the above algorithm, the initial selection of the codebook entries can either be made by picking two vectors arbitrarily from the training set, or by calculating the centroid of the whole training set, and then splitting the centroid to give two vectors which are spectrally dissimilar. In running the algorithm, each training set vector is compared with every codebook entry, hence the name full search. When a group of training vectors are determined to belong to the same cluster, the procedure in step 5 of the algorithm, aims to obtain a single vector that represents the whole cluster with minimum error. This vector is termed the centroid of the cluster and is computed by averaging the corresponding autocorrelation vectors, and then deriving the LPC vector of this averaged autocorrelation vector. The algorithm terminates when the desired number of entries, M, in the codebook is achieved or when the average distortion falls below a pre-set threshold.

A flow chart which illustrates the full search binary split algorithm is given in Figure 5.12.

ii) **The tree-search algorithm**

The tree-search VQ algorithm starts with an optimum 2-entries (i.e. m=2) codebook which has been generated by a full search algorithm. The two entries in the codebook are split to give a 4-entries codebook. Instead of running a full search procedure on the training set, each of the 4-entries searches only the training vectors in the cluster associated with its parent entry. The complete algorithm can be described as follows:

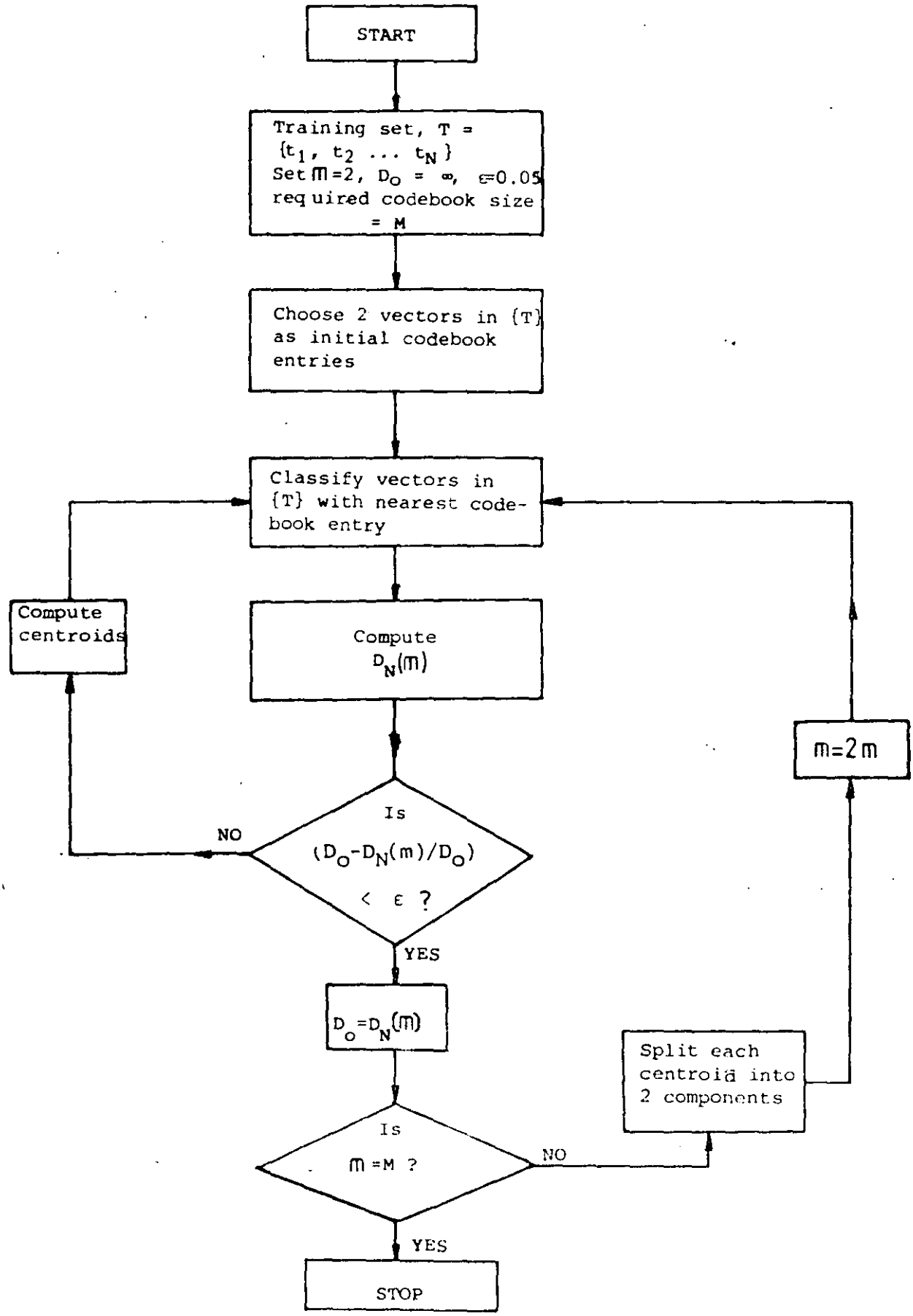


FIGURE 5.12: THE FLOW CHART OF THE VQ FULL-SEARCH ALGORITHM

Step 1: Start with a training set, $T = \{t_1, t_2, ..., t_N\}$, of a large number of LPC vectors obtained from the reference patterns.

Step 2: Select two vectors from the set T, to be the initial codebook entries $\{a_1, a_2\}$, i.e. m=2. Set D_o, the initial average distortion of the training set to a large value.

Step 3: If m=2, compute the distance $D_N(m)$ as in Step 3 of the full search algorithm. For m > 2, compute the distance between an entry and the training vectors coded by its parent entry, to obtain the average distortion of the training set.

Step 4: Code each of the training set vectors by the nearest codebook entry, based on nearest distance, i.e. vector $t_i \in T$ belongs to cluster m' if

$$d(t_i, C_{m'}) = \underset{1 \leq m' \leq m}{\text{MIN}} \; d(t_i, C_{m'}) \qquad 5.62$$

Step 5: If the decrease in average distortion, $(D_o - D_N(m))/D_o$, is less than a pre-set threshold ε, set D_o equal to $D_N(m)$ and go to step 7.

Step 6: Compute the centroid of each of m clusters in the training set. Use the centroids as the new codebook entries. Go to step 4.

Step 7: Exit if the desired size, M, of the codebook has been achieved, otherwise split each entry into two components. Set m=2m and go to step 3.

5.5.3 VQ Experimental Results

The vector quantizer training set {T}, was derived from a speech data base formulated from the utterances of five speakers: SM2, SM3, SM4, SF2 and SF3, on the 50 words vocabulary. Each vocabulary word was

represented by eight reference patterns; two patterns from each of the male subjects SM2, SM3, SM4 and one pattern from each of the subjects SF2, SF3. After passing the speech signal through a pre-emphasis network, $H(z) = 1-0.90z^{-1}$, a 14th order LPC analysis was performed on 25.6 msec Hamming windowed speech segments and a set {T}, of 7658 LPC vectors was obtained.

Both the full-search and the true-search VQ algorithms used the above training set to generate codebooks of sizes 8, 16, 32, 64 and 128. In the algorithms, the initial two entries in the codebook were selected arbitrarily from the training set.

During the iterative process, the decrease in average training set distortion, $D_N(m)$ is monitored, using a pre-set distortion threshold $\epsilon = 0.05$. A centroid, $\hat{a}_m$, is split into two vectors, $\hat{a}_{m1}$ and $\hat{a}_{m2}$, by retaining the centroid as $\hat{a}_{m1}$, and generating $\hat{a}_{m2}$ as the slightly perturbed centroid by an arbitrary factor, of say, 0.98, i.e.

$$\hat{a}_{m1} = \hat{a}_m \quad \text{and} \quad \hat{a}_{m2} = 0.98\ \hat{a}_m$$

Results obtained on the average distortion, $D_N(m)$ of the training set, in the generation of codebook of sizes 2, 4, 8, 16, 32, 64 and 128 entries, for both full-search and tree-search algorithms, is shown in Figure 5.13. With the full search method the 128 entries codebook was generated after 73 iterations, compared to 58 iterations for the tree-search method. Thus, as would be expected, the tree-search method converges faster than the full-search method. The gain in convergence rate is obtained at the expense of a higher training set distortion level.

The cluster occupancy, N_i, is defined as the number of training vectors in the ith cluster, i.e. the cluster represented by the ith codebook entry. Figure 5.14 shows a histogram of the number of clusters and their occupancy for the 128 entries codebook generated

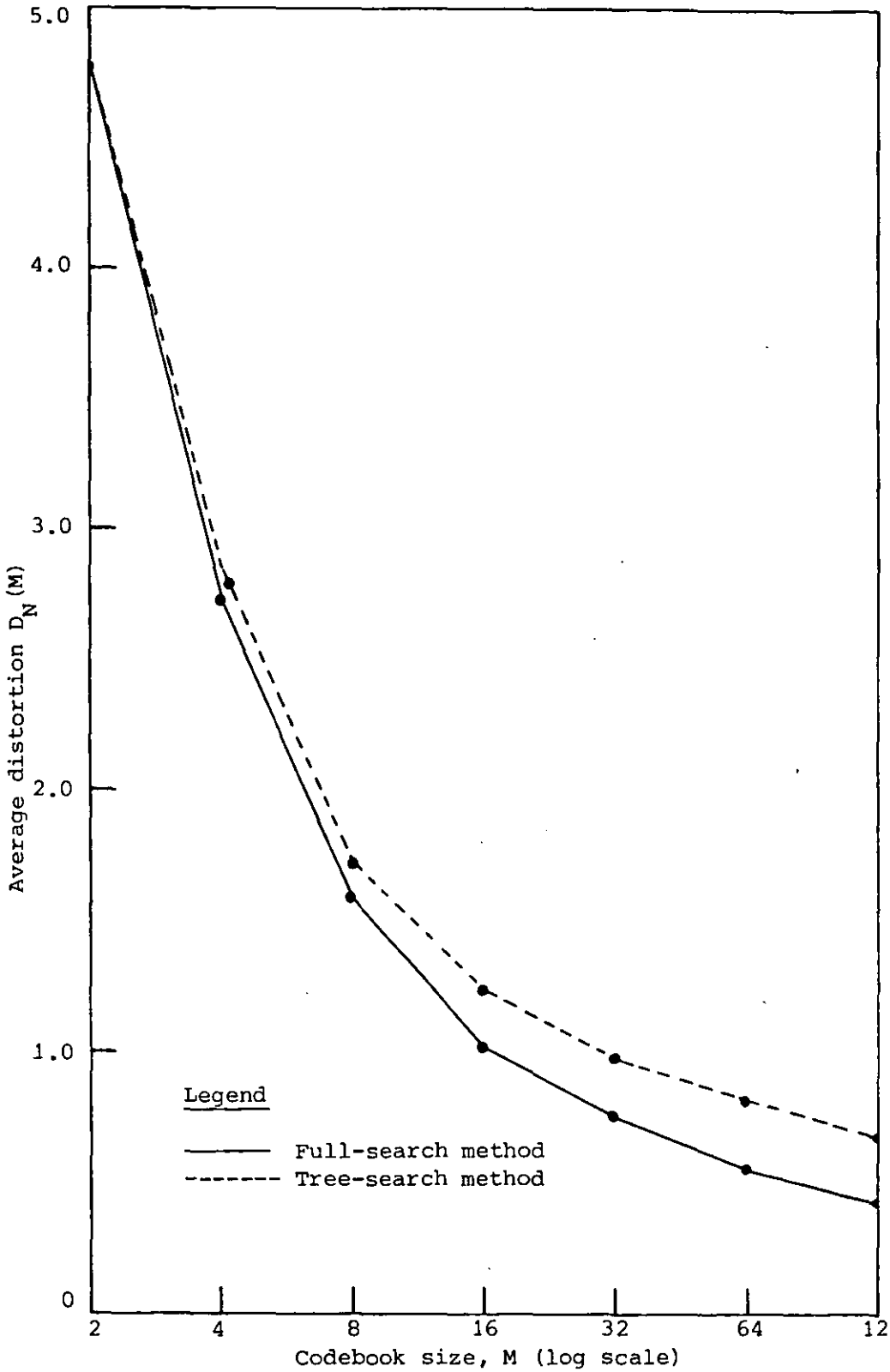


FIGURE 5.13: VARIATION OF THE AVERAGE DISTORTION OF THE TRAINING SET WITH CODEBOOK SIZE

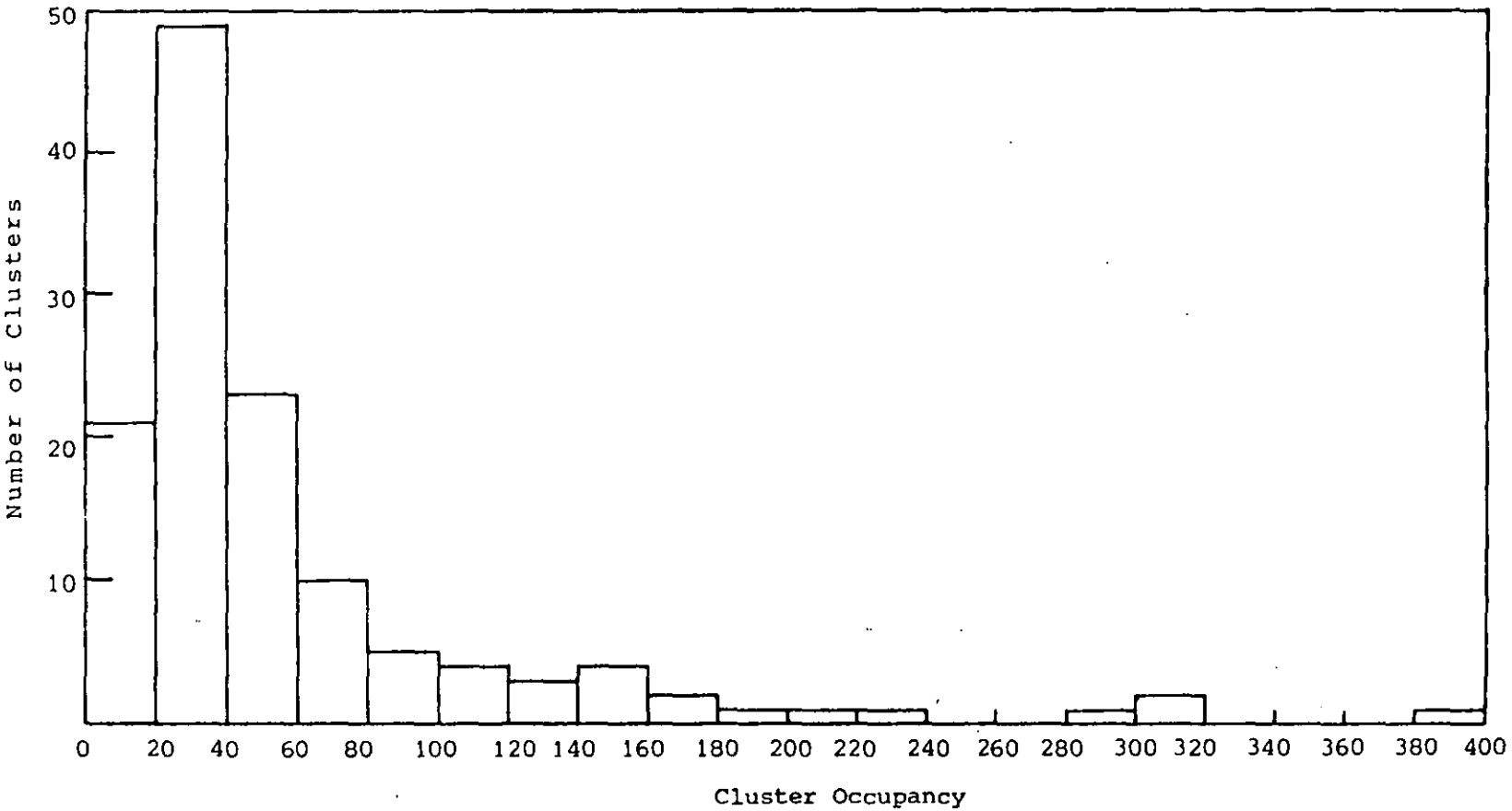


FIGURE 5.14: THE NUMBER OF CLUSTERS IN THE TRAINING SET, AND THEIR OCCUPANCY, FOR A 128 ENTRIES CODEBOOK GENERATED BY THE FULL-SEARCH METHOD

by the full-search method. The largest cluster had 381 training vectors, and the smallest cluster contained 10 training vectors. A similar characteristic for the 128 entries codebook generated by a tree-search method is shown in Figure 5.15. The largest cluster had 323 training vectors and the smallest cluster had only one training vector. These characteristics indicate that empty clusters would more easily arise in the tree-search rather than in the full-search method.

5.5.4 The LPC/SPLIT Recognizer

The memory storage in a word recognizer is mainly used for the reference patterns of vocabulary words. In order to achieve speaker independence, multiple reference patterns per vocabulary word are usually employed. This results in a large increase in memory requirements. Sugamura et al [81], proposed the LPC/SPLIT recognition system which uses vector quantization techniques to reduce the memory requirements without severely degrading the recognition accuracy of the system. Figure 5.16 is an illustration of the LPC/SPLIT recognition system.

i) **The training procedure**

A set {T}, of thousands of LPC vectors is extracted from a speech data base consisting of all the vocabulary words uttered by different speakers. Using a full search VQ process on the set {T}, a codebook C_x of M entries is generated. The entries in codebook C_x, can be regarded as 'phoneme-like' or pseudo-phoneme since they exhibit distinct spectral properties. Each reference pattern is then expressed as a 'sequence of phoneme-like templates', hence the name SPLIT, by (a) computing the spectral distance, between each reference pattern LPC vector, and all the phoneme-like templates, (b) substituting the LPC vector of each segment for the corresponding phoneme-like template which offers minimum spectral distance, i.e.

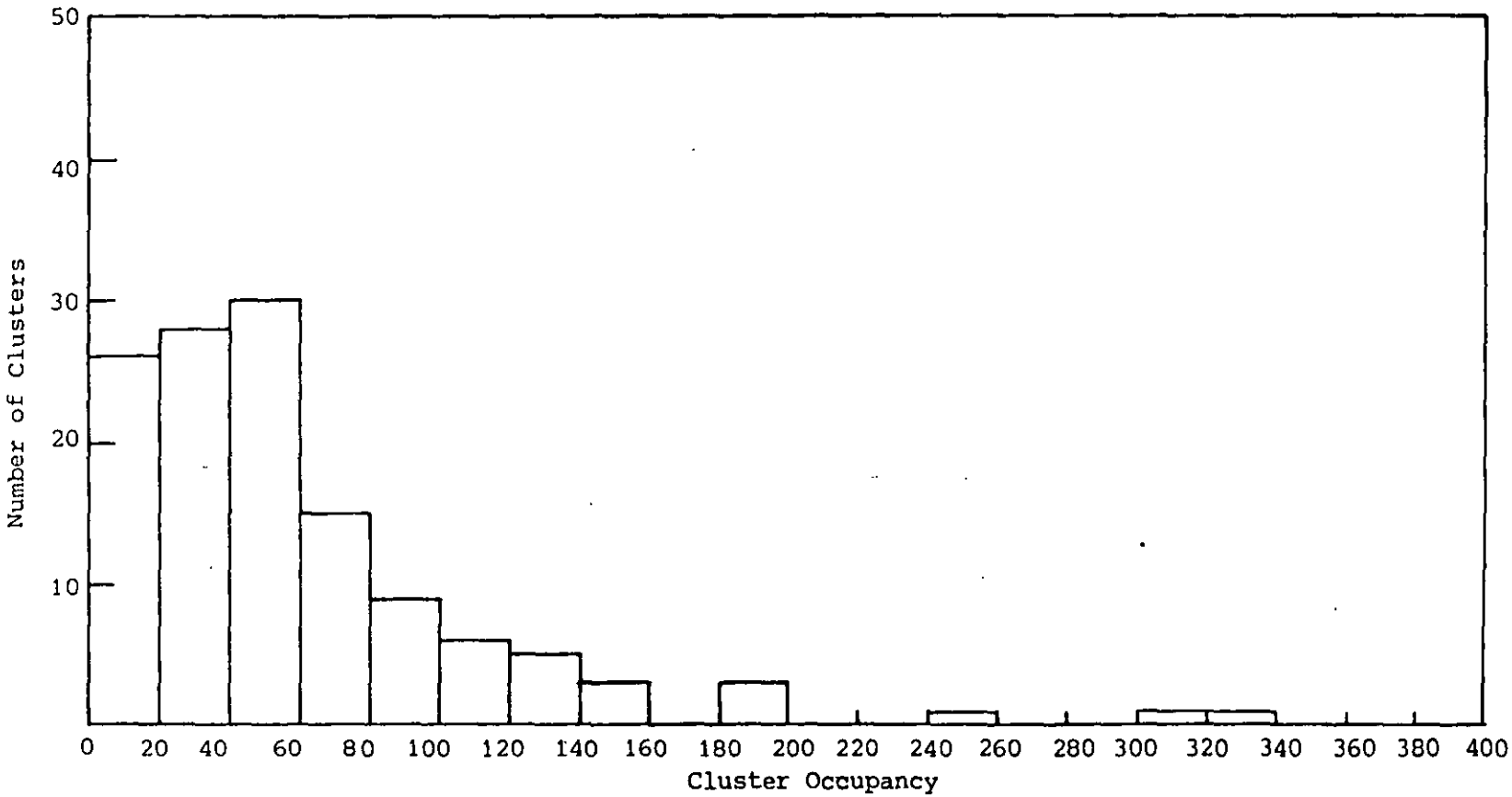


FIGURE 5.15: THE NUMBER OF CLUSTERS IN THE TRAINING SET, AND THEIR OCCUPANCY, FOR A 128 ENTRIES CODEBOOK GENERATED BY A TREE-SEARCH METHOD

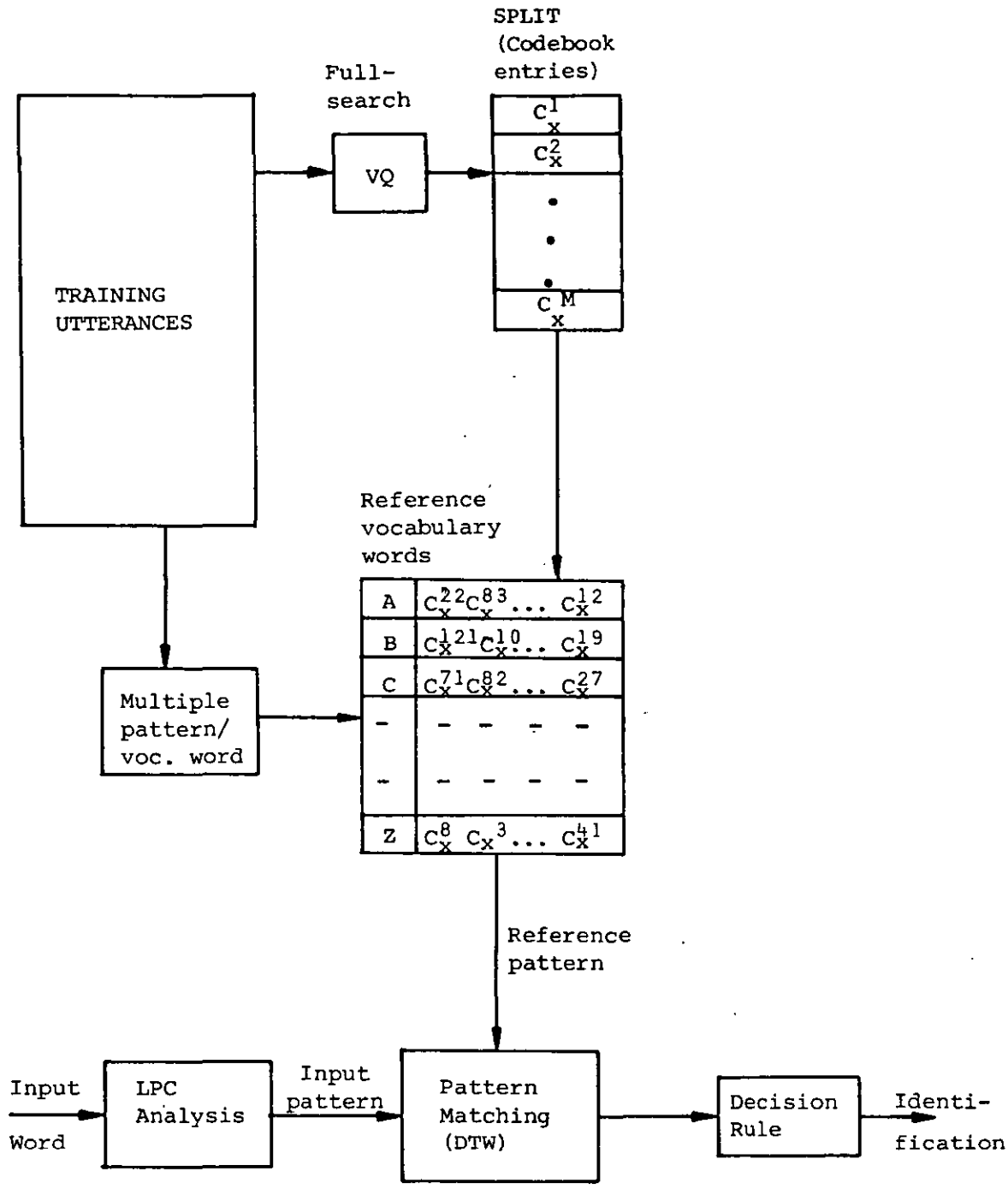


FIGURE 5.16: THE LPC/SPLIT RECOGNIZER

Let the reference pattern, B, be the sequence of J, LPC vectors

$$B = \{b_1, b_2, \ldots, b_j, \ldots, b_J\} \qquad 5.63a$$

and the set of phoneme-like templates C_x, be

$$C_x = \{C_x^1, C_x^2, \ldots, C_x^m, \ldots, C_x^M\} \qquad 5.63b$$

where C_x^m is the mth phoneme-like template.

The pattern B is expressed as a sequence of J phoneme-like templates as follows:

$$B = \{C_x^{m1}, C_x^{m2}, \ldots, C_x^{mj}, \ldots C_x^{mJ}\} \qquad 5.63c$$

where

$$d(b_j, C_x^{mj}) = \min_{1 \leq m \leq M} (b_j, C_x^m), \; 1 \leq j \leq J \qquad 5.63d$$

Thus, in the LPC/SPLIT recognizer, it will only be necessary to store the set of phoneme-like templates, C_x, and the sequence of indices that define each reference pattern.

ii) Recognition procedure

During the recognition procedure, the input word is expressed as a sequence of LPC vectors and compared with the reference patterns which have already been expressed as a sequence of phoneme-like templates. A dynamic time warping process is used to obtain a time

normalized distance between the input and reference patterns. Subsequently, the input word is identified as the vocabulary word whose reference pattern is associated with minimum distance.

The speaker independent performance of the LPC/SPLIT system can be enhanced by using multiple reference patterns per vocabulary word, and then using a kNN rule, as discussed in Section 4.4 of Chapter 4, to identify the input word.

5.5.5 The LPC/VQ Recognizer

Most isolated word recognition systems which employ the pattern matching approach, require some form of time normalization procedure in order to eliminate the temporal differences between reference and input speech patterns. These time normalization procedures are generally computationally expensive, and as such, techniques capable of obliviating their use, would be desirable. Shore and Burton [82], have proposed a recognition system, which uses reference patterns whose time sequence information has been removed. The system, therefore, is able to achieve pattern comparison without the need for a DTW procedure. The Shore-Burton recognizer, referred to as the LPC/VQ recognizer in this thesis, uses a VQ technique to generate reference patterns without a temporal axis.

Figure 5.17 is a block diagram which illustrates the structure of an LPC/VQ recognition system.

i) **Training session**

Each vocabulary word is represented by multiple reference patterns obtained from a group of speakers. These patterns provide a short training sequence of LPC vectors, which are used to generate a codebook for the vocabulary word. Thus, each vocabulary word will be represented by a unique codebook whose entries no longer possess the time sequence information. For example, a codebook for the vocabulary word "X" is generated by running a full-search VQ

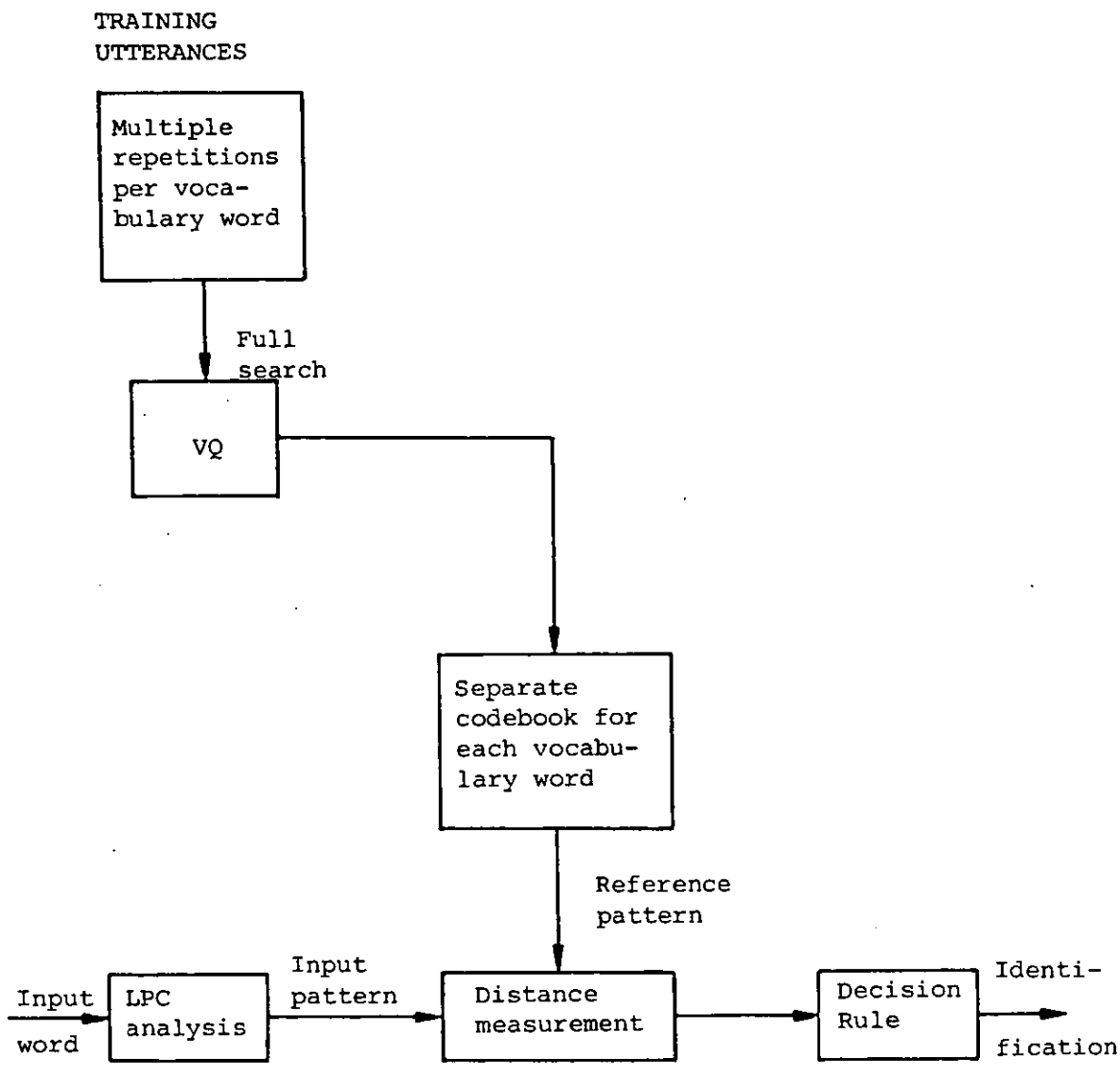


FIGURE 5.17: THE LPC/VQ RECOGNIZER

procedure on the available versions of the word 'X'. The codebooks can be designed to be of fixed size or fixed distortion.

Fixed size codebooks have a predetermined size M, and the VQ algorithm generates the M entries that minimize the training set distortion rate to an acceptable threshold. Therefore, fixed size codebooks of the various vocabulary words, will display different distortion levels.

In the fixed-distortion codebooks, the VQ algorithm produces a codebook that encodes the training data with a pre-set average distortion. Fixed-distortion codebooks of the different vocabulary words will not necessarily be of equal size.

Fixed-size codebooks of 8 entries and of 16 entries were generated for every vocabulary word.

ii) **Testing session:**
Let an unknown input utterance, A, be represented as the discrete sequence of I, LPC vectors:

$$A = \{a_1, a_2, \ldots, a_i, \ldots a_I\} \qquad 5.64a$$

and let R be the number of words in the recognition vocabulary. Then there are R codebooks C_r, r = 1, 2, ..., R. The size of the rth codebook C_r, is denoted N_r.

The average distortion obtained on encoding this input pattern with the rth codebook is given by:

$$d_r = \frac{1}{I} \sum_{i=1}^{I} \underset{1 \leq j \leq N_r}{\text{MIN}} \; d(a_i, C_r^j) \qquad 5.64b$$

where C_r^j is the jth entry of the codebook C_r, and $d(a_i, C_r^j)$ is the gain-normalized Itakura-Saito distortion between a_i and C_r^j.

The decision logic that follows, classifies the input pattern as the vocabulary word associated with the codebook that has the minimum weighted average distance measure D_m, defined as:

$$D_m = \underset{1 \leq r \leq R}{\text{MIN}} \quad (d_r/L_r) \qquad 5.64c$$

where L_r is the number of distinct entries used in the rth codebook to obtain d_r.

5.5.6 The LPC/VQ/SPLIT Recognizer [83]

The proposed LPC/VQ/SPLIT recognizer combines the design philosophies of the two previous systems, i.e. the LPC/SPLIT and the LPC/VQ recognizers, to yield an efficient isolated word recognizer with improved memory and computational complexity characteristics. The system is illustrated in the block diagram in Figure 5.18.

i) Training session

The LPC/VQ/SPLIT recognizer employs a separate reference codebook per vocabulary word, in the same way as the LPC/VQ recognizer. Each codebook is based on one word uttered a number of times by different speakers, as in the LPC/VQ recognizer training session. A codebook of M pseudo-phonemes is also generated by a full search VQ process in a long training sequence of vocabulary words. In addition, the entries of each reference codebook are replaced with the nearest of the M pseudo phonemes. That is, given an R word vocabulary, the rth codebook C_r of size N_r,

$$C_r = \{C_r^1, C_2^2, \ldots, C_r^j, \ldots, C_r^{N_r}\} \qquad 5.65a$$

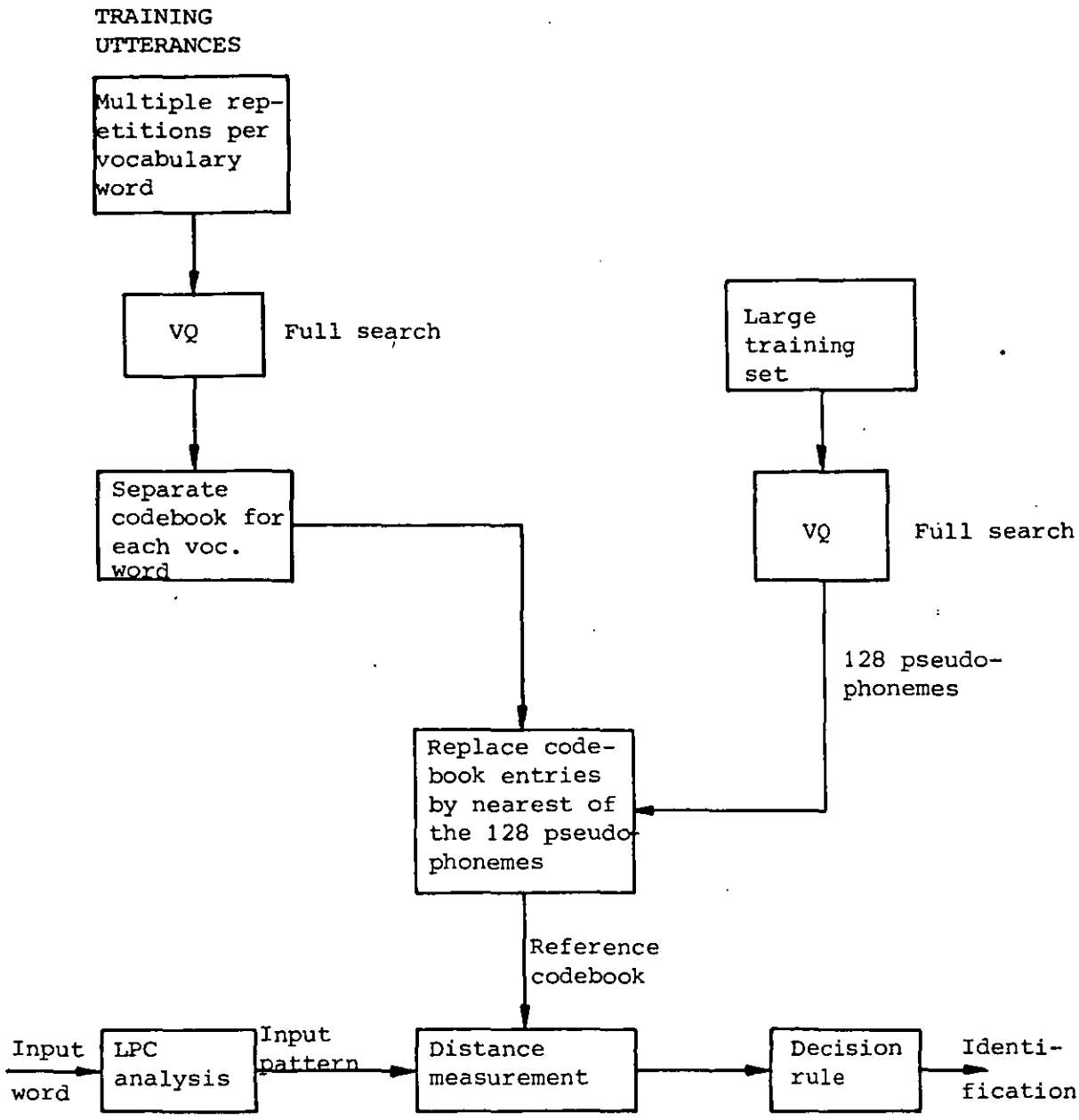


FIGURE 5.18: THE LPC/VQ/SPLIT RECOGNIZER

where C_r^j is the jth entry
and, given also, the phoneme-like templates codebook, X

$$X = \{X_1, X_2, \ldots, X_k \ldots X_M\} \qquad 5.65b$$

the modified codebook $\hat{C}_r$ is defined as:

$$\hat{C}_r = \{\hat{C}_r^1, \hat{C}_r^2, \ldots, \hat{C}_r^j \ldots, \hat{C}_r^{N_r}\} \qquad 5.65c$$

where $\hat{C}_r^j = X_k$, and the index k minimizes the distortion measure $d(C_r^j, X_k)$ for k = 1,2, ...M.

Fixed-size codebooks, $\hat{C}_r$, with 8 entries and 16 entries were generated.

ii) **Testing session**

An input word, A, which is expressed as a sequence of I, LPC vectors is compared with each of the modified reference codebooks $\hat{C}_r$ to give a distance d_r, r = 1,2, ... R i.e.

$$d_r = \frac{1}{I} \sum_{i=1}^{I} \underset{1 \leq j \leq N}{\text{MIN}} \; d(a_i, \hat{C}_j^r) \qquad 5.65d$$

where $d(a_i, \hat{C}_j^r)$, is the gain-normalized Itakura-Saito distance between the ith LPC vector of A, and the jth entry of $\hat{C}_r$.

The input word is recognized as the word which corresponds to that codebook giving the minimum weighted average distance D_m, as in equation 5.64c.

5.5.7 Results

i) Recognition accuracy

The recognition accuracy of the LPC/SPLIT, LPC/VQ and LPC/VQ/SPLIT systems operating on the 50 word vocabulary was assessed by computer simulations. The training speech data base was composed of the utterances of three male subjects who spoke each word twice, and the utterances of the two female subjects who spoke each word once. The speech utterances were bandlimited to 5 kHz and then sampled at 10 kHz. The 12-bit per sample, digitized signal, was segmented into 25.6 msec frames, from which LPC vectors were obtained by a 14th order analysis on the pre-emphasized and Hamming windowed speech every 12.8 msec.

The LPC/SPLIT recognizer used a 128 entries codebook generated by a full-search VQ procedure, on the whole training data sequence as described in Section 5.5.2 (ii).

The LPC/VQ recognizer, used a codebook per vocabulary word generated from the eight repetitions of the word. Fixed size codebooks of 8 entries and 16 entries were obtained for every vocabulary word. Table 5.3 gives the average distortion, $D_N(M)$, on the training data from subjects SM1, SM2, SM4, SF2 and SF3, in the generation of the codebooks.

The proposed LPC/VQ/SPLIT recognizer, employed both the 128 entries codebook of the LPC/SPLIT recognizer, and the fixed size codebooks of the LPC/VQ recognizer.

Table 5.4 shows the recognition performance of the three systems in three testing sessions. The results are based on the utterances spoken by a male and a female subject who did not contribute to the training data used to generate the reference codebooks.

TABLE 5.3

FIXED SIZE, REFERENCE CODEBOOKS DISTORTION, GENERATED BY THE FULL-SEARCH METHOD

Vocabulary Word	No. of Training Set Templates	Distortion for 8/16 size Codebooks	Vocabulary Word	No. of Templates in Training Set	Distortion for 8/16 entries Codebooks
One	268	0.940/0.565	P	258	0.502/0.311
Two	258	0.743/0.596	Q	294	0.760/0.570
Three	286	0.697/0.408	R	278	0.558/0.345
Four	272	0.463/0.395	S	302	0.537/0.402
Five	334	0.726/0.415	T	248	0.418/0.348
Six	384	0.541/0.379	U	274	0.645/0.398
Seven	332	0.601/0.372	V	260	0.457/0.319
Eight	302	0.845/0.539	W	326	0.795/0.590
Nine	322	0.685/0.462	X	354	0.537/0.590
Zero	348	0.706/0.442	Y	298	1.143/0.584
A	246	0.481/0.274	Z	272	0.594/0.414
B	222	0.411/0.338	Delete	336	0.631/0.487
C	318	0.460/0.374	Input	356	1.023/0.743
D	230	0.339/0.246	Write	332	1.095/0.630
E	236	0.367/0.282	End	302	0.664/0.446
F	274	0.285/0.217	Load	324	0.868/0.508
G	258	0.459/0.266	Add	238	0.742/0.449
H	336	0.680/0.445	Set	336	0.589/0.383
I	264	0.676/0.382	Control	426	1.041/0.705
J	282	0.536/0.358	Store	368	0.731/0.531
K	254	0.442/0.302	No	304	0.737/0.495
L	238	0.569/0.387	Read	332	0.978/0.422
M	234	0.662/0.397	Yes	356	0.719/0.493
N	246	0.562/0.383	Multiply	486	0.947/0.685
O	246	0.654/0.435	Output	398	0.769/0.620

TABLE 5.4

PERFORMANCE OF THE LPC/SPLIT, LPC/VQ, LPC/VQ/SPLIT RECOGNIZERS

	RECOGNITION ACCURACY (%)			
Recognition System	Test 1 Test speaker SM1	Test 2 Test speaker SM3	Test 3 Test speaker SF1	Average
	Ref Speakers: SM2, SM3, SM4, SF2	Ref Speakers: SM1, SM2, SM4, SF2	Ref Speakers: SM1, SM2, SM4, SF2	
LPC/SPLIT (4 ref patterns/ voc. word)	86	86	78	83.1
	Ref Speakers: SM2, SM3, SM4, SF2, SF3	Ref Speakers: SM1, SM2, SM4, SF2, SF3	Ref Speakers: SM1, SM2, SM4, SF2, SF3	
LPC/VQ 8 entries codebook	92	90	88	90.0
LPC/VQ 16 entries codebook	96	96	90	94.0
LPC/VQ/SPLIT 8 entries codebook	90	88	84	87.3
LPC/VQ/SPLIT 16 entries codebook	96	92	90	92.6

ii) **Memory requirements**

The memory required in each system, for storing the reference patterns can be easily computed for an R words vocabulary. Assuming that the rth codebook C_r, consists of N_r, p-dimensional LPC vectors, and that each vector element is represented with an average of N_a bits, then the LPC/VQ recognizer requires a memory size, S_{VQ}, given by:

$$S_{VQ} = \sum_{r=1}^{R} N_r.p\ N_a \text{ bits} \qquad 5.66$$

The LPC/SPLIT recognizer requires to store the set of phoneme-like templates, and the index sequences of the reference patterns. Thus, using N reference patterns for each vocabulary word and assuming Q_i frames in the ith reference pattern, the memory required, S_{SPLIT}, is,

$$S_{SPLIT} = M.p.\ N_a + \sum_{i=1}^{N.R} Q_i\ \log_2 M \text{ bits} \qquad 5.67$$

where M is the number of pseudo-phonemes.

Similarly the memory size, $S_{VQ/SPLIT}$, required in the LPC/VQ/SPLIT system, is given by:

$$S_{VQ/SPLIT} = M.p.N_a + \sum_{i=1}^{R} N_r.\log_2 M \text{ bits} \qquad 5.68$$

The memory characteristics of the three systems are shown in Figure 5.19, as a ratio of the memory size in the LPC/VQ/SPLIT recognizer with R = 10, M = 128 and N_r = 16.

iii) **Computational complexity**

To a first approximation, a comparison of the computational complexity of the systems can be based on the complexity of their

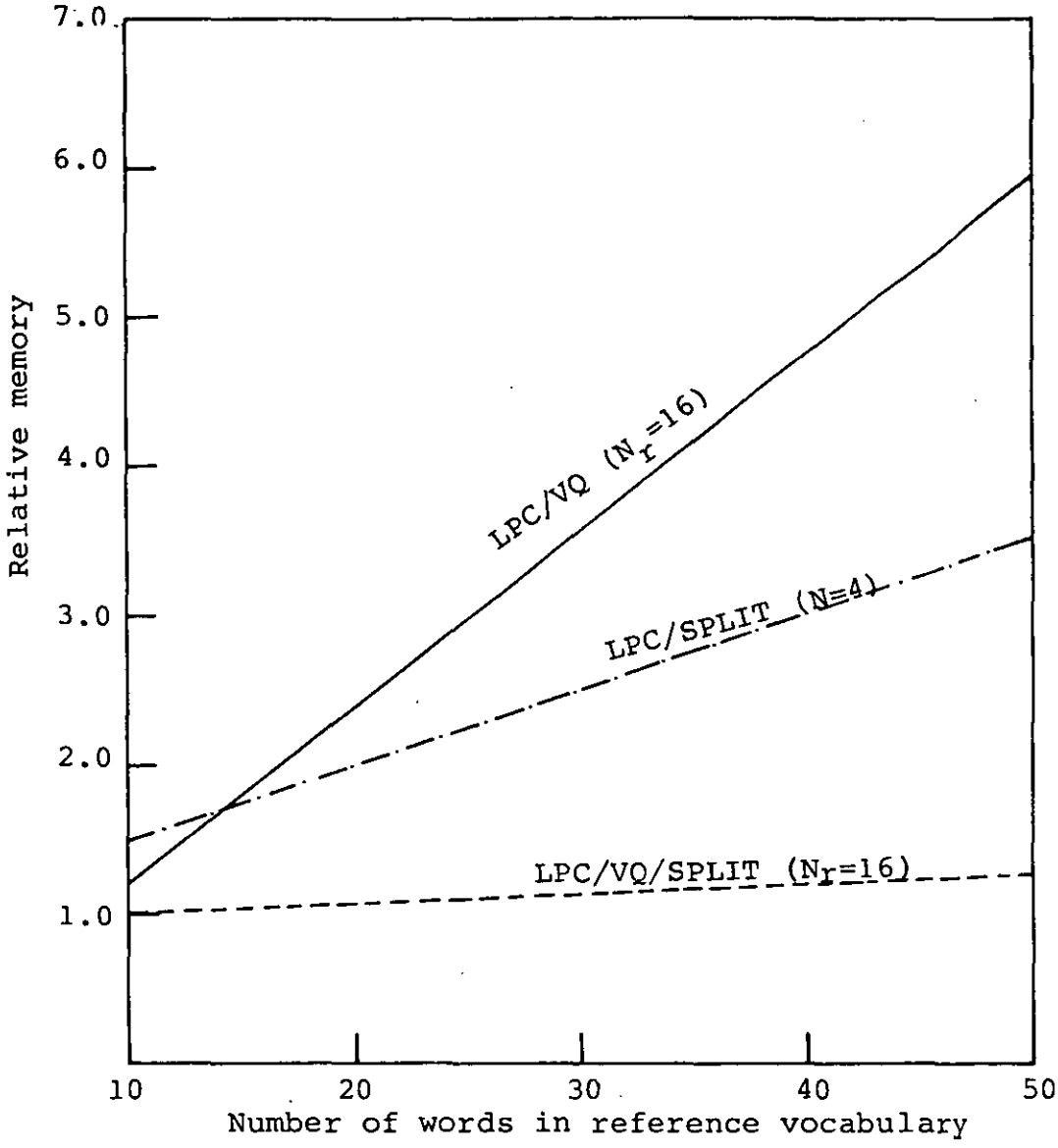


FIGURE 5.19: MEMORY REQUIREMENTS IN THE RECOGNITION SYSTEMS AS A RATIO OF THE MEMORY SIZE IN THE LPC/VQ/SPLIT SYSTEM WITH R=10, M=128 AND N_r = 16

pattern matching stage of the recognition process, and in particular on the computational effort involved in forming the required spectral distances between the LPC vectors.

Consider the LPC/SPLIT recognizer, with an average of J frames per reference pattern, and using an adjustment window in the DTW process of r frames wide. If the input pattern has an average of I frames, and N reference patterns per vocabulary word are used, then the number of spectral distances, D_1, required is:

$$D_1 = [I.J - (I-r)(J-r)]N.R \qquad 5.69$$

Similarly, in the LPC/VQ, and in the LPC/VQ/SPLIT recognizers, the number of spectral distances, D_2, to be computed is given by:

$$D_2 = I.N_r.R \qquad 5.70$$

where N_r is the average number of entries in a codebook. For example, using typical values, such as I = 40, J = 40, N_r = 16, N = 4 and r = 8, gives an estimate of the ratio D_1/D_2, as:

$$4 \leqslant (D_1/D_2) \leqslant 8 \qquad 5.71$$

5.6 DISCUSSION

In this Chapter, word recognition systems using patterns expressed as discrete sequences of LPC feature vectors, are examined. The initial investigations were concerned with the effects of the LPC prediction order on the recognition accuracy. Since the short-time spectral estimation of a speech segment improves while increasing the

prediction order p, the recognition accuracy is expected to display the same characteristic. As the order of prediction was varied from 6 to 14, the recognition results shown in Table 5.1, indicate a monotonic improvement. However, the difference between results obtained with prediction order, p = 12 and p = 14, show only a slight improvement. This is because for values of prediction larger than 12, the rate of decrease in the prediction error of the system is small, as illustrated in Figure 5.4. The results obtained with the female subject show a markedly lower accuracy in comparison with the male subjects, as shown in Tables 5.1, 5.2 and 5.4 The reason can be attributed to the inferior LPC modelling of female speech.

The use of discriminative reference patterns, for the similar words set {B,C, D, E, G, V, P, T}, shows an improvement in word recognition accuracy, as shown in Table 5.2. This would be expected, since attention is focussed on those regions in which a speech pattern differs from patterns of acoustically similar words.

The heavy computational cost, in word recognition systems employing multiple reference patterns, leads to the investigation of the use of clustering techniques to reduce this cost. The method of clustering reference patterns into small disjoint groups, using the MKM algorithm was found to be effective in this respect. As shown in Figure 5.11, the method provides a reduction in computational load by about a half, at the expense of a slight degradation in recognition accuracy.

The multiple reference patterns, which are employed in the word recognition system to achieve speaker independence, increases the memory requirements of the system. In addition, the DTW procedure, used to provide a non-linear time alignment between input and reference patterns, is a computationally expensive process. Thus, vector quantization techniques, which can be used to solve these two problems were also investigated. The generation of vector quantization codebooks from a large training data set was performed using both the full-search and the tree-search binary split methods.

From the results given in Figures 5.13, 5.14 and 5.15, the superiority of the full-search method over the tree-search is exposed. The codebook obtained with the full-search method gives a lower training data set distortion than the codebook of similar size generated with the tree-search method. The full-search method also partitions the training set into better distributed clusters, than the tree-search method. The smallest cluster in the full-search method contains 10 training set vectors, whereas with the tree-search, clusters containing single vectors were found. This means that the tree-search method is more likely to give rise to empty clusters, which would be an undesirable situation. The only advantage with the tree-search is its faster convergence rate.

Three isolated word recognition systems, LPC/SPLIT, LPC/VQ and LPC/VQ/SPLIT, were then studied. From the computer simulation results given in Section 5.5.7, the advantages of the proposed LPC/VQ/SPLIT recognition system over the other two established systems is clearly evident. The LPC/VQ system offers the highest recognition accuracy, but its memory requirement proves to be prohibitive for large vocabularies. Although the memory requirements are relaxed in the LPC/SPLIT system, the recognition accuracy is relatively poorer. Only the LPC/VQ/SPLIT system offers high recognition rate, with low memory/computational complexity characteristics.

5.7 NOTE ON PUBLICATION

A paper entitled, "The use of phoneme-like templates in isolated word recognition without time alignment", in co-authorship with the Supervisor, Dr C S Xydeas, has been published in the Proceedings of the 3rd European Signal Processing Conference (EUSIPCO), held at the Hague, the Netherlands on 2-5 September 1986. The paper is based on the work presented in Section 5.5.

CHAPTER 6

THE USE OF VOICED, UNVOICED, AND SILENCE CLASSIFICATION OF SPEECH SEGMENTS IN WORD RECOGNITION

6.1 INTRODUCTION

An improvement in the accuracy of an isolated word recognition system can be achieved using the techniques discussed earlier in Chapters 4 and 5 namely: multiple reference patterns per vocabulary word, redundancy suppression in the speech utterances, discriminative reference patterns for similar sounding words, and vector quantization. An alternative approach for improving the recognition rate, is to detect the broad acoustic structure of an utterance, and then use the information to supplement a conventional recognizer. The acoustic structure obtained using the three voiced, unvoiced and silence classes, can give a strong indication as to the identity of an unknown utterance within the recognition vocabulary. For example, in a 'digit' recognition system, if the unknown input utterance is detected to begin with an unvoiced fricative, then it is obvious that the word cannot be a 'ONE', 'EIGHT', or a 'NINE', and thus the pattern comparison would be limited to the other seven reference candidate words. If the input word is further determined to have the acoustic structure, 'unvoiced-voiced-silence-unvoiced', then the most likely candidate is the word 'SIX'.

The information of the acoustic structure of an utterance can also be exploited in order to discriminate between utterances which have similar sounding regions. For example, the word sets {X,SIX}, {YES,S}, may result in a very close distance measure in the LPC-based recognizer and hence misclassification will occur, if say, an input utterance 'SIX' has its 'X' portion more similar to the reference word 'X' than to the reference word 'SIX'. Since 'X' and 'SIX' have quite different acoustic structures, it would be advantageous to use the acoustic structure as an aid in the recognition process.

In this Chapter, a new method on the voiced-unvoiced-silence classification of speech segments using the fuzzy set theory, is first presented. The proposed algorithm is subsequently employed in obtaining the acoustic structure of the input utterances which in turn is used to enhance the accuracy of a conventional word recognizer.

6.2 VOICED-UNVOICED-SILENCE CLASSIFICATION OF SPEECH

The need to classify successive segments of speech as Voiced, Unvoiced, or Silence (VUS), arises in speech recognition, as well as in the areas of voice synthesis and the reduction of acoustic noise which has been added to speech signals. A number of existing VUS classification methods are based on five parameters measured from the input signal, namely:

i) the zero-crossing rate
ii) the logarithmic energy
iii) the first autocorrelation coefficient
iv) the first LPC coefficient
v) the normalized prediction error.

The choice of these specific parameters, hereafter referred to as the 'VUS parameters', to determine the voiced, unvoiced or silence nature of speech segments can be attributed to experimental observations as well as to speech synthesis theory. Given the value of these parameters, the question arises however as how to use the information for an accurate VUS classification of an input speech segment. Atal and Rabiner [84] for example, assumed that the five parameters are distributed according to a multidimensional Gaussian probability density function whose mean and covariances are obtained using a training procedure. A minimum distance rule was subsequently employed to classify the speech segments as voiced, unvoiced or silence.

In this section, an alternative approach to VUS classification using fuzzy set theory is proposed [85] and its performance compared with the Atal and Rabiner's method.

The classification of speech segments into VUS classes can be suitably modelled by fuzzy algorithms, since the various classes are defined in an inexact manner by the five parameters. In the proposed scheme, a training procedure is used, in which speech segments from the voiced, unvoiced and silence classes are manually selected and analyzed to derive the average values of the mentioned five parameters in each class. A decision rule which characterizes each class, is formulated based on the values of the parameters. For example, an unvoiced segment is characterized by a 'high' zero crossing rate count, a 'medium' logarithmic energy, a 'low' first delay autocorrelation coefficient, a 'high' first LPC coefficient, a 'medium' normalized prediction error. The linguistic terms, 'low', 'medium' and 'high' are relative and do not possess sharp boundaries, and hence are vaguely defined. The parameters of an input speech utterance to be classified, are considered as elements of a fuzzy set, whose membership grades are distributed according to the so called π or S functions. The VUS parameters obtained in the training procedure are used to specify the π and S functions. Modelling the parameters of the input speech segment with the decision rule for each class serves to identify its 'closeness' to that class. The 'closeness' can be defined as a number in the interval, [0,1]. In the following sections, a discussion on the theory of fuzzy sets and its applicability in the classification of speech signals is presented. The VUS classification method proposed by Atal and Rabiner, is also discussed. The performance of the two methods, in classifying speech segments from subjects who did not contribute to the training procedure, is compared.

6.2.1 Elements of the Fuzzy Set Theory [86][87][88]

The fuzzy set theory is an algebra based on imprecision, whereby each object under consideration as an element of a set, is assigned a

membership grade which expresses its degree of belongingness to that set. Thus, a gradual transition from non-membership of the set to full membership is provided.

A fuzzy set A, of a universe of discourse U is thus characterized by a membership function:

$$\mu : U \rightarrow [0,1] \qquad 6.1$$

which assigns a membership grade μ, for every element of U, in the interval [0,1]. Full membership grade is designated as 1, and non-membership as grade 0. A cross-over point in the set A, is the element y whose grade of membership is 0.5. A fuzzy singleton is a fuzzy set consisting of a single element. If A is a fuzzy singleton, and y is its element in U, with the membership grade μ, then A is expressed with the denotation:

$$A = \mu/y \qquad 6.2$$

A fuzzy set may be seen as the union of its singletons:

i.e.
$$A = \int_U \mu(y)/y \qquad 6.3$$

and, if A has a finite number of singletons, then:

$$A = \mu_1/y_1 + \mu_2/y_2 + \ldots + \mu_i/y_i + \ldots + \mu_n/y_n \qquad 6.4$$

where μ_i, i = 1,2, ..., n, is the membership grade of y_i in A.

In many situations, it is appropriate to express the membership function of a fuzzy set in terms of a standard function whose

parameters are adjustable to approximately fit a specified membership function. Figures 6.1 and 6.2 illustrate the standard functions, S and π, which are commonly used. A function is S type, if it is monotonically increasing (S^+), or decreasing (S^-) and is defined as follows:

$$S^+(y:\alpha,\beta,\gamma) = \begin{cases} 0 & \text{for } y \leqslant \alpha \\ 2\,((y-\alpha)/(\gamma-\alpha))^2 & \text{for } \alpha \leqslant y \leqslant \beta \\ 1 - 2\,((y-\gamma)/(\gamma-\alpha))^2 & \text{for } \beta \leqslant y \leqslant \gamma \\ 1 & \text{for } y \geqslant \gamma \end{cases} \qquad 6.5a$$

and

$$S^-(y:\alpha,\beta,\gamma) = 1 - S^+(y:\alpha,\beta,\gamma) \qquad 6.5b$$

where $\beta = (\alpha+\gamma)/2$, is the cross-over point.

A function is π type, if there exists only a single point at which monotonicity changes direction, and is defined as:

$$\pi(y:\beta,\gamma) = \begin{cases} S^+(y;\ \gamma-\beta,\ \gamma-\beta/2,\ \gamma) & \text{for } y \leqslant \gamma \\ 1-S^+(y;\ \gamma,\ \gamma+\beta,\ \gamma+\beta) & \text{for } y \geqslant \gamma \end{cases} \qquad 6.6$$

where β is the bandwidth, i.e. the separation between the two cross-over points of π. γ is the point at which π is unity.

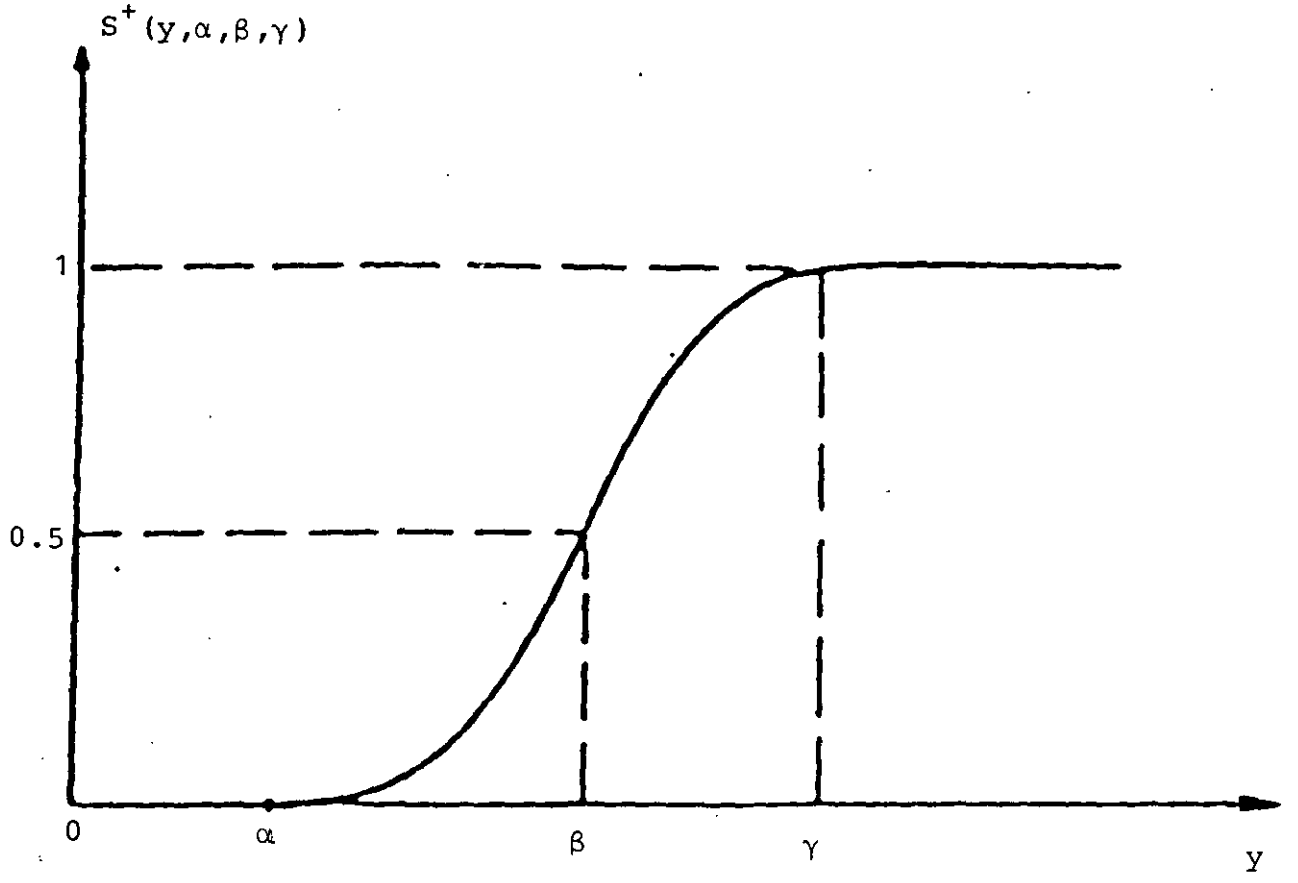


FIGURE 6.1a: PLOT OF THE S^+ FUNCTION

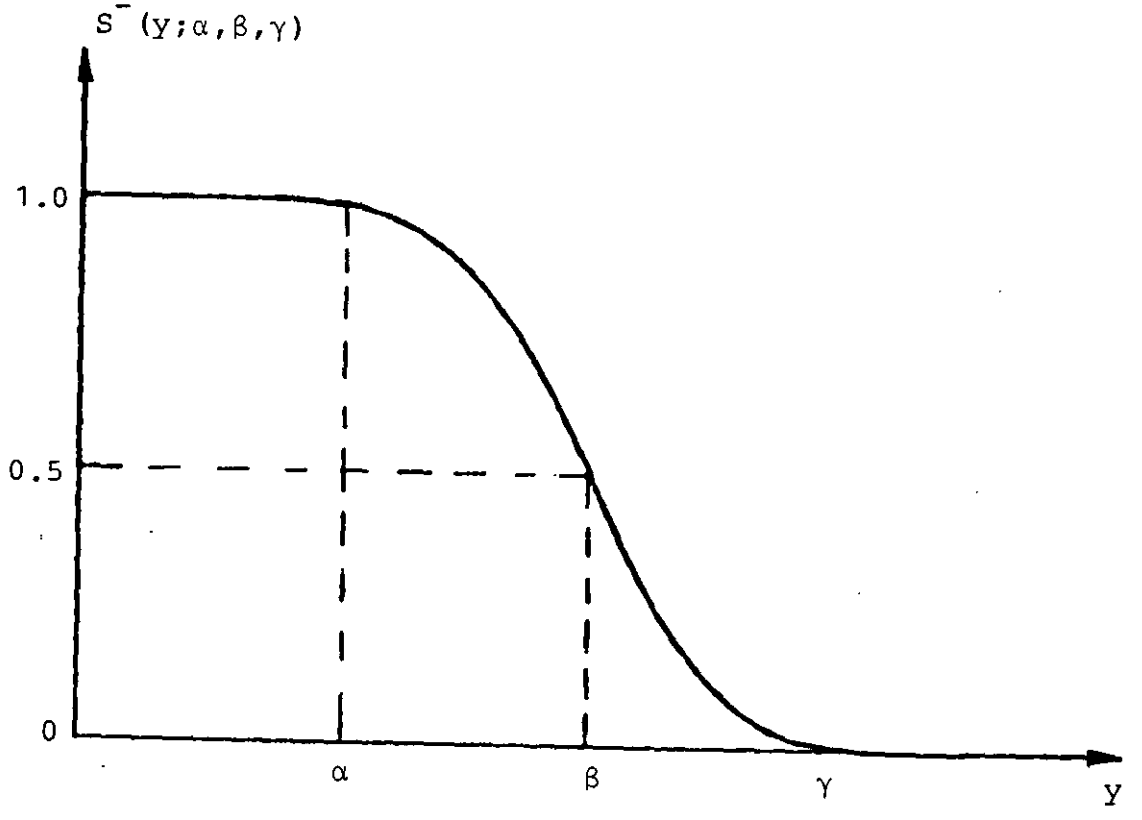


FIGURE 6.1b: PLOT OF THE S^- FUNCTION

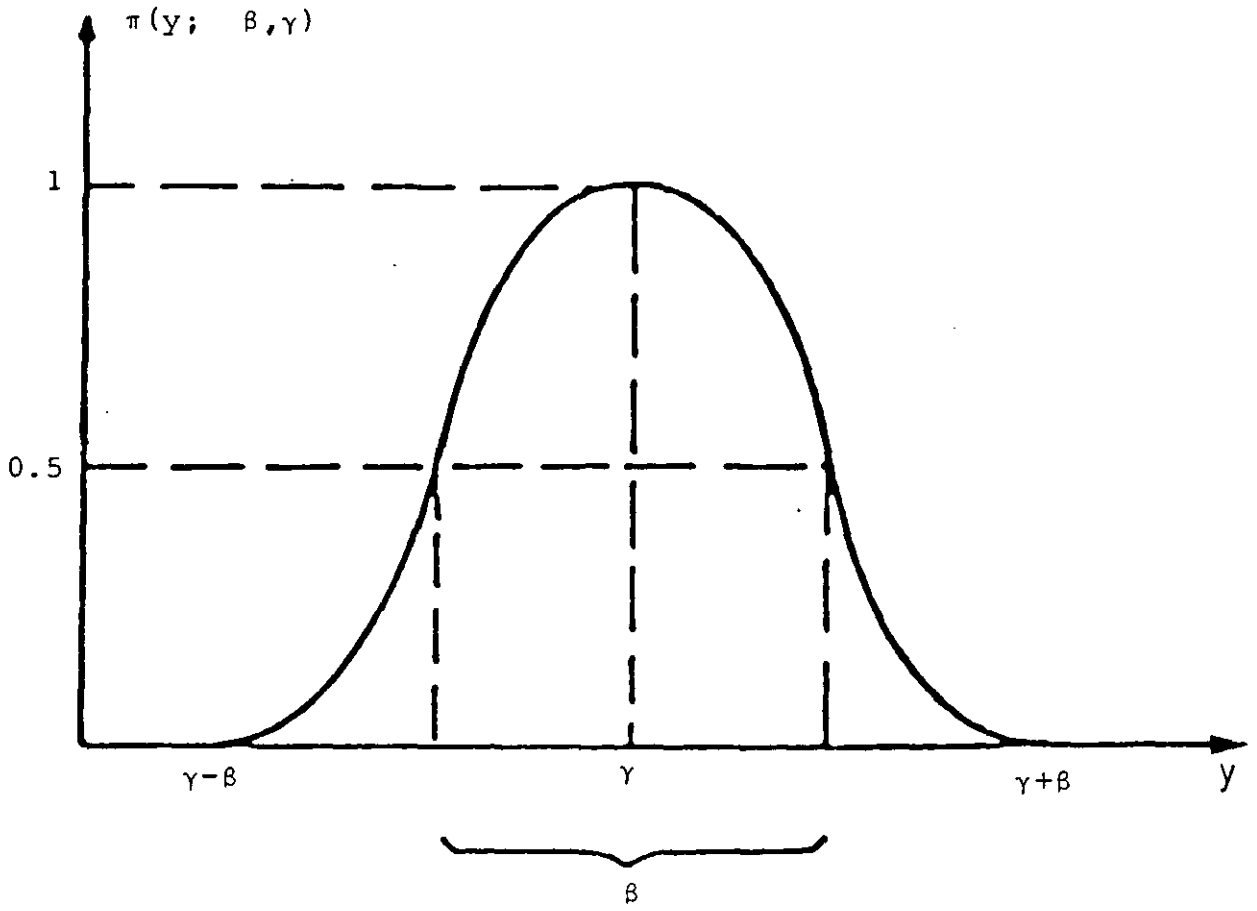


FIGURE 6.2: PLOT OF THE π FUNCTION

6.2.2 The VUS Parameters

For the VUS classification of speech segments, it is desirable to use parameters which are easy to extract from the signal and efficient in discriminating between the voiced, unvoiced and silence classes. The following parameters are known to satisfy these requirements:

i) The zero crossing rate count (ZCR) is related to the number of sample polarity changes in the speech segment and indicates the frequency at which spectral energy is concentrated. Spectral energy is concentrated at low frequency for voiced sounds, and at high frequency for unvoiced sounds. Depending on the background noise, the ZCR count for silence frames is generally higher than that of voiced segments, but lower than the ZCR of unvoiced segments.

ii) The logarithmic energy, LE, in the voiced speech segments is considerably higher than in unvoiced segments. Silence frames contain the least energy.

iii) The unit sample delay autocorrelation coefficient R(1), lies by definition, between -1 and +1. Since in voiced sounds, the energy is concentrated in the low frequency range, adjacent speech samples are highly correlated, and R(1) takes a value close to +1. The correlation is close to zero for unvoiced sounds.

iv) The first LPC coefficient is identical to the value of the Cepstrum of the signal at unit sample delay [see Appendix D]. Since the spectrum of these three classes have such considerable difference, so should the first LPC coefficient.

v) The prediction error is an indication of the uniformity of the speech spectrum. Voiced sounds have a spectrum with resonances which result in a smaller prediction error than for unvoiced sounds. The prediction error E_p, discussed in Section 5.2.2, is

normalized with the frame energy LE, to give the logarithmic normalized prediction error, ERR; i.e.

$$ERR = LE - 10 \log_{10} [R(0) + \sum_{i=1}^{p} a(i)\, R(i)] \qquad 6.7$$

where a(i) is the ith LPC coefficient, R(i) the ith autocorrelation coefficient, p is the order of prediction, and LE is the logarithmic energy of the speech samples in the frame. The value of ERR will be high for voiced sounds and low for unvoiced sounds.

6.2.3 The Decision Process

The classification of speech segments into VUS classes can be suitably modelled by fuzzy reasoning since these classes are defined in an inexact manner by the five parameters. Rules can be formulated relating the classes to these parameters using fuzzy linguistic terms. For example, the rule for voiced class indicates its characterization as low zero crossing rate count (ZCR), a high logarithmic energy (LE), a high unit delay autocorrelation coefficient (R(1)), a low first LPC coefficient (LP1) and a high logarithmic normalized prediction error (ERR). The complete set of rules are as follows:

Rule 1: Voiced class = low ZCR + high LE + high R(1) + low LP1 + high ERR

Rule 2: Unvoiced class = high ZCR + medium LE + low R(1) + high LP1 + medium ERR

Rule 3: Silence class = medium ZCR + low LE + medium R(1) + medium LP1 + low ERR

i) **Training session**

The linguistic terms 'high', 'low', 'medium', can be described by the S^+, S^- and π functions of Figures 6.1a, 6.1b and 6.2 respectively with the thresholds α and γ determined during a training session.

The speech utterance 'At the side of the rock, a small stream flowed into the river', spoken by a female subject, bandlimited at 3.4 kHz and sampled at 8 kHz was manually classified into voiced, unvoiced and silence intervals. Speech segments of 16 msec duration, from each of the three classes, were selected and analyzed to derive the average values of the mentioned five VUS parameters. The logarithmic normalized prediction error was calculated using a 12th order filter.

For example, the fuzzy set 'low ZCR' is described by the S^- function illustrated in Figure 6.1b. The threshold α is the average value of the voiced segment ZCR count, and γ is the average unvoiced ZCR count. In Table 6.1, the parameter values obtained in the training procedure and the thresholds used in the various fuzzy sets are shown.

ii) Classification

The speech utterances to be classified are bandlimited to 3400 Hz, sampled at 8 kHz and segmented at intervals of 16 msec. In each segment, the five VUS parameters are obtained.

Using rule 1, the 'closeness' of the measured VUS parameters to the voiced class can be determined. The grade of membership of the parameters in the respective fuzzy sets are evaluated, e.g. the grade of membership, v_1, of the ZCR count in the fuzzy set 'low ZCR' is obtained using the S^- function. Rule 1 gives the following relation, defined as the fuzzy set V_x,

$$V_x = v_1/\text{low ZCR} + v_2/\text{high LE} + v_3/\text{high R(1)} + v_4/\text{low LP1} + v_5/\text{high ERR} \qquad 6.8$$

where v_1, v_2, v_3, v_4, v_5 are the membership grades of the VUS parameters, ZCR, LE, R(1), LP1, ERR, in the respective fuzzy sets.

The 'closeness' of the same VUS parameters to the unvoiced class is evaluated in a similar manner, using rule 2. The fuzzy set, U_x,

TABLE 6.1: THE VUS PARAMETER VALUES, FUZZY SET THRESHOLDS, AND COVARIANCE MATRICES OBTAINED IN THE TRAINING SESSION

CLASS	PARAMETERS				
	ZCR	LE	R(1)	LP1	ERR
Voiced	19.58	49.89	0.86	-1.67	13.77
Unvoiced	86.52	29.26	-0.51	0.80	6.62
Silence	26.52	3.86	0.65	-0.70	4.09

FUZZY SET THRESHOLDS

S^- Sets	S^+ Sets	π Sets
Low ZCR (α= 19.58, γ= 86.52)	High ZCR (α= 19.58, γ= 86.52)	Medium ZCR (γ= 26.52, β= 3.47)
Low LE (α= 3.86, γ= 47.89)	High LE (α= 3.86, γ= 47.89)	Medium LE (γ= 29.26, β= 9.32)
Low R(1) (α= -0.51, γ= 0.86)	High R(1) (α= -0.51, γ= 0.86	Medium R(1) (γ= 0.65, β= 0.80)
Low LP1 (α= -1.67, γ= 0.80)	High LP1 (α= -1.67, γ= 0.80	Medium LP1 (γ= -0.70, β= 0.48)
Low ERR (α= 4.09, γ= 13.77)	High ERR (α= 4.09, γ= 13.77)	Medium ERR (γ= 6.62, β= 1.26)

COVARIANCE MATRICES

Voiced Class

1.00	0.691	-0.930	0.237	-0.768
0.691	1.00	-0.810	0.153	-0.890
-0.930	-0.810	1.00	-0.141	0.847
0.237	0.153	-0.141	1.00	-0.437
-0.768	-0.890	-0.847	-0.437	1.00

Unvoiced Class

1.00	-0.252	-0.981	0.659	0.234
-0.252	1.00	0.245	0.177	0.234
-0.981	0.245	1.00	-0.725	-0.290
0.659	0.177	-0.725	1.00	0.755
0.234	0.324	-0.290	0.755	1.00

Silence Class

1.00	-0.840	-0.641	0.760	-0.742
-0.840	1.00	0.665	-0.937	-0.884
-0.641	0.665	1.00	-0.793	-0.881
0.760	-0.937	-0.793	1.00	-0.970
-0.742	-0.884	-0.881	-0.970	1.00

obtained in this case is given by:

$$U_x = u_1/\text{high ZCR} + u_2/\text{medium LE} + u_3/\text{low R(1)} + u_4/\text{high LP1} + u_5/\text{medium ERR} \quad 6.9$$

where u_1, u_2, u_3, u_4, u_5 are the membership grades of the VUS parameters in the respective fuzzy sets.

Finally, rule 3 is used to determine the 'closeness' of the VUS parameters to the silence class. The fuzzy set S_x obtained in this case is given by:

$$S_x = s_1/\text{medium ZCR} + s_2/\text{low LE} + s_3/\text{medium R(1)} + s_4/\text{medium LP1} + s_5/\text{low ERR} \quad 6.10$$

where s_1, s_2, s_3, s_4, s_5 are the membership grades in the respective fuzzy sets.

Since the grade of membership of an element in a fuzzy set indicates the degree of belongingness to the concept expressed by that set, it can also be interpreted as a measure of 'truthfulness'. Absolute truth would be indicated by a membership grade 1, and absolute falsity by 0. The decision for VUS classification can be based on the degree of 'truth' in each of the three sets V_x, U_x, and S_x. The sum of membership grades in each of the sets V_x, U_x and S_x lies between a maximum of 5 and a minimum of 0. The S^+ function can be used as a 'truth' distribution with the two threshold extremes α and γ set to 0 and 5 respectively. The class yielding the highest 'truth' value is interpreted as the correct class for the speech segment, i.e. from the relationships expressed in equations 6.8, 6.9 and 6.10.

Let,

$$y_1 = \sum_{j=1}^{5} v_j, \quad y_2 = \sum_{j=1}^{5} u_j, \text{ and } y_3 = \sum_{j=1}^{5} s_j \qquad 6.11$$

The grade of membership, μ_i, of y_i in the 'truth' distribution set is given by:

$$\mu_i = S^+ (y_i: \alpha, \beta, \gamma) \qquad i = 1,2,3 \qquad 6.12a$$

The speech segment is classified in i, if

$$\text{Class } i \triangleq \text{MAX}(\mu_i), \quad i = 1,2,3 \qquad 6.12b$$

Class 1,2,3 refers to voiced, unvoiced and silence classes respectively.

6.2.4 Atal and Rabiner's Method [84]

The same five parameters are also used in Atal and Rabiner's method to classify the speech segment into VUS classes. To make this decision, a classical minimum probability of error decision rule is employed which assumes a multidimensional Gaussian distribution of the parameters, with a mean and covariance matrix obtained from a training session. The mean VUS parameter values, and the corresponding covariance matrices for 16 msec segments of the training speech utterance of Section 6.2.3 are given in Table 6.1. Specifically, let x be an L-dimensional column vector representing the five parameters in a speech segment to be classified. Then the L-dimensional Gaussian density function, $g_i(x)$, for x with mean

vector m_i and covariance matrix W_i, for the ith class, is given by:

$$g_i(x) = (2\pi)^{-L/2} \, |W_i|^{-\frac{1}{2}} \, \exp[-\frac{1}{2} (x-m_i)^t \, W_i^{-1} \, (x-m_i)] \qquad 6.13$$

where W_i^{-1} is the inverse of W_i, $|W_i|$ is the determinant of W_i and $(x-m_i)^t$ is the transpose of $(x-m_i)$.

The mean m_i and covariance matrix W_i, for the class i are given by:

$$m_i = \sum_{n=1}^{N} x(n)/N \qquad 6.14$$

and

$$W_i = \sum_{n=1}^{N} (x(n)x^t(n) - m_i m_i^t)/N \qquad 6.15$$

where N is the number of training vectors for class i, i = 1,2,3. Class 1,2,3 refers to voiced, unvoiced and silence classes respectively.

For multidimensional Gaussian distribution, a discriminant function, $d_i(x)$, which classifies feature vectors with a minimum error is given as [89]:

$$d_i(x) = (x-m_i)^t \, W_i^{-1} \, (x-m_i) \qquad 6.16$$

That is, to classify a feature vector x, the weighted distance, $d_i(x)$, from x to each of the class mean vectors, m_i, is computed. Vector x is assigned to the class which gives the nearest distance.

6.2.5 Results

The speech utterance, 'Industrial shares were mostly a trifle higher', from a male subject, and the utterance 'Joining hands, they danced in excitement around the fire', from a female speaker were used to test the performance of both the fuzzy set and the Atal and Rabiner's classification methods. The input speech signal was bandlimited to 3.4 kHz and sampled at 8 kHz. The results obtained in classifying 16 msec speech segments, using both algorithms, are given in Tables 6.2 and 6.3 in the form of a confusion matrix of correct and incorrect identifications. For example in Table 6.2, using the fuzzy set method, 260 voiced segments were correctly classified, 2 voiced segments were classified as unvoiced and 7 voiced segments were classified as silence. Figure 6.3 shows the time waveform of the utterance 'Industrial shares were mostly a trifle higher', and the classification of its segments into VUS classes using the fuzzy set method. These results show that the fuzzy set method provides a classification accuracy comparable to the Atal and Rabiner's method. In addition, the absence of matrix multiplications in the fuzzy method, unlike in Atal and Rabiner's method, serves to simplify the computational load.

6.3 ACOUSTIC SEGMENTATION

A speech utterance can be expressed as a sequence of segments belonging to the voiced, unvoiced or silence classes by dividing the utterance into segments of suitable temporal durations and classifying the signal in each interval as voiced, unvoiced or silence. For recognition purposes, however, this 'fine' segmentation results in a cumbersome system. This is because versions of the same word will have a different number of segments, hence giving rise to acoustically different patterns. As such, a coarse segmentation process which gives the general acoustic structure of the utterance is desirable. Attempts to use methods based on rules for combining or deleting classified segments in order to obtain a coarse acoustic structure of an utterance, proved unfruitful. The main problem was

TABLE 6.2: RESULTS FROM THE FUZZY ALGORITHM

Identified as:	Actual Class		
	Voiced	Unvoiced	Silence
VOICED	260	0	6
UNVOICED	2	59	3
SILENCE	7	5	58

TABLE 6.3: RESULTS FROM ATAL AND RABINER'S ALGORITHM

Identified as:	Actual Class		
	Voiced	Unvoiced	Silence
VOICED	263	2	5
UNVOICED	3	60	3
SILENCE	3	2	59

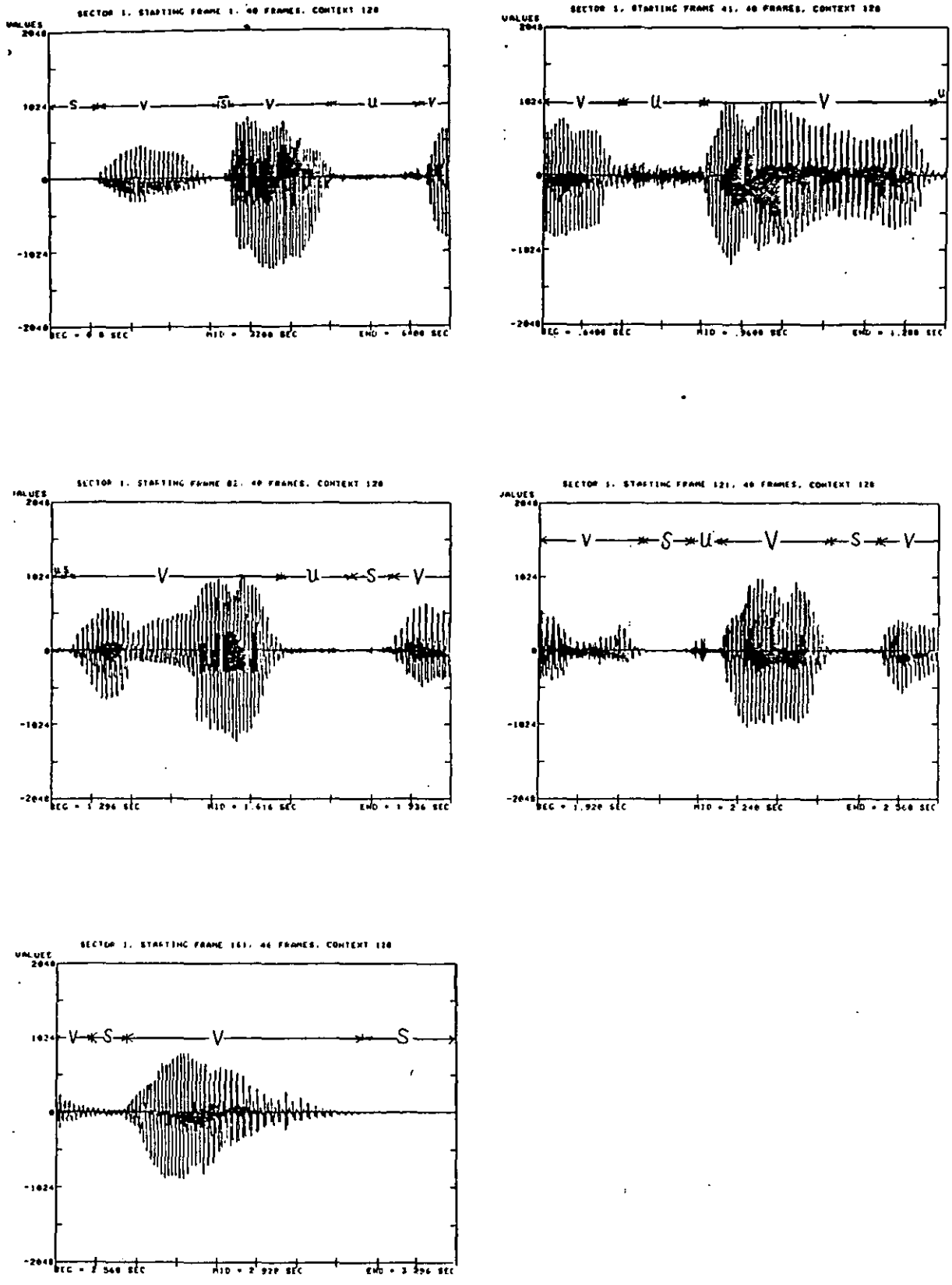


FIGURE 6.3: CLASSIFICATION OF THE SEGMENTS OF THE UTTERANCE 'INDUSTRIAL SHARES WERE MOSTLY A TRIFLE HIGHER' USING THE FUZZY SET THEORY METHOD

to obtain a fixed number of coarse segments per utterance. In addition, such rules can lead to erroneous decisions. For example, the occurrence of a short silence duration, preceded and succeeded by a larger voiced duration, can be an actual possibility. Thus, using rules to obtain coarse segments, by deleting the silence frames in such a situation, would result in an erroneous decision. A suitable method for obtaining a coarse structure from the fine segmentation is to optimally divide an utterance into a given number of regions, and then classifying each region according to the nature of segments it contains. The procedure is discussed below.

6.3.1 Bridle's Algorithm [90]

An optimum procedure for segmenting a speech utterance into a given number of regions has been proposed by Bridle et al [90], and is as follows:

Let a speech utterance be represented by the discrete sequence of multidimensional feature vectors, $\{a_1, a_2, \ldots a_N\}$, and that it is desired to divide the utterance into M regions, where $M < N$. The speech pattern has N-1 junctions, numbered 1, 2, ..., N-1 between feature vectors where the boundary of the regions might be placed. Let the fixed boundaries before a_1 and after a_N, be numbered 0 and N respectively. The division of the utterance into M regions now reduces to selecting the M-1 of the interior junctions $i_1, i_2, \ldots, i_{M-1}$, and keeping the fixed boundaries, i.e. $i_0 = 0$ and $i_M = N$.

In the algorithm, a 'segment evaluation function', f(i,j), is defined as the error introduced by representing the region of the utterance between junction i and junction j, as a single feature vector and is given by:

$$f(i,j) = \begin{cases} \sum_{k=i+1}^{j} d(a_k, \bar{a}_{ij}) \\ 0, \text{ if } i = j \end{cases} \qquad 6.17$$

where $\bar{a}_{ij} = \sum_{k=i}^{j} a_k/(j-i)$, is the mean vector of the feature vectors in the region, and $d(a_k, \bar{a}_{ij})$ is a distance measure.

A global segmentation criterion, G, which is a function of the sequence of junctions chosen as the new boundaries is defined as the sum of errors introduced in each portion of the utterance and is given as:

$$G(i_0, i_1, \ldots, i_m) = \sum_{k=1}^{M} f(i_{k-1}, i_k) \qquad 6.18$$

It is of significance that equation 6.18 can be defined recursively as:

$$G\{i_0, i_1, \ldots, i_M\} = G\{i_0, i_1, \ldots, i_{M-1}\} + f\{i_{M-1}, i_M\} \qquad 6.19$$

The aim of the algorithm is to obtain the sequence $\{i_0, i_1, \ldots, i_M\}$, which minimizes G. Let F(m,n) be the minimum value of G obtained in dividing the first n segments of the utterance into m sections:

$$\text{i.e.} \qquad F(m,n) = \underset{i_1, i_2, \ldots i_{m-1}}{\text{MIN}} \; G\{0, i_1, \ldots, i_{m-1}, n\} \qquad 6.20$$

using equation 6.19, then equation 6.20 can be expressed as:

$$F(m,n) = \text{MIN}\; G\{0, i_1, \ldots, i_{m-1}\} + f(i_{m-1}, n) \qquad 6.21$$

which simplifies to:

$$F(m,n) = \min_{i} [F(m-1, i) + f(i,n)] \qquad 6.22$$

Equation 6.22, allows the computation of the approximate error for the best division of the whole utterance into M sections. During the computation, the values F(1,N), F(2,N), ..., F(M-1,N) which are the minimum errors if fewer sections are required, are also produced. At every stage in the computation, the junction number i, which minimizes equation 6.22 is stored in an array P(m,n). After F(M,N) is obtained, the optimal section boundaries can be recovered by starting with i_M = M, and then tracing back through the array P(m,n).

6.3.2 Results

Bridle's segmentation algorithm was tested with the three 5 kHz bandlimited speech utterances, 'SIX', 'X', 'INPUT' obtained from the subject SM1. Each utterance was segmented into 25.6 msec frames and then a 14th order LPC analysis carried out. Bridle's algorithm was then used to divide the utterance into a required number of regions. Since the LPC coefficients in each segment are available, the segment evaluation function of equation 6.15 employed the gain-normalized Itakura-Saito distance measure.

Figure 6.4 shows the time waveform of the utterance, 'SIX', consisting of 23 segments of 25.6 msec duration each. The result of dividing the utterance into 3, 4, 5 or 6 regions using Bridle's algorithm is shown in the figure. For example, the boundaries of a four region division of the utterance are: 0, 6, 10, 16, 23. It can be seen from the figure that these boundaries tend to correspond to acoustic changes in the utterance. Also shown in the figure is the voiced, unvoiced, silence classification of the 23 segments as obtained with the fuzzy set theory approach. Similarly results obtained with the utterances 'X' and 'INPUT' are shown in Figures 6.5 and 6.6 respectively.

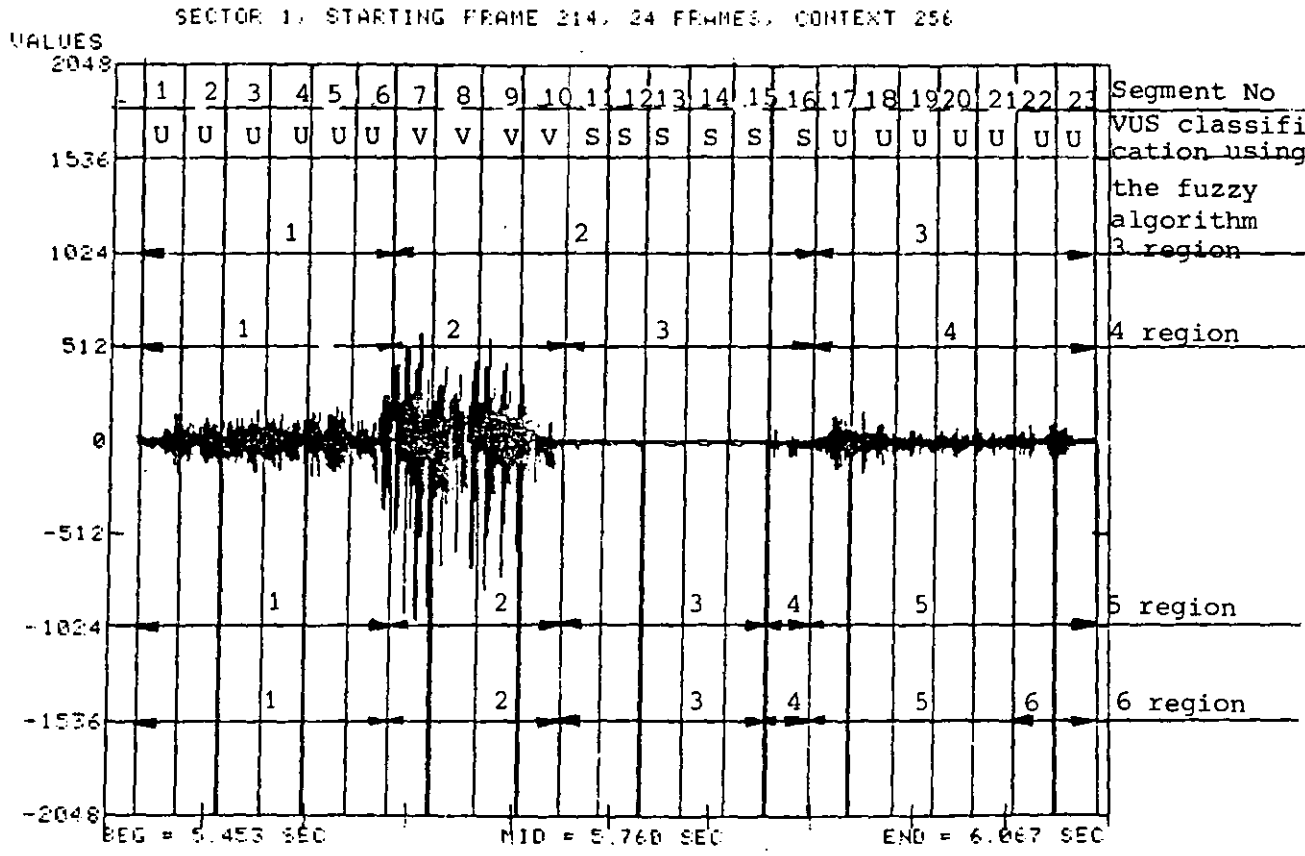


FIGURE 6.4: SEGMENTATION OF THE UTTERANCE 'SIX' INTO 3, 4, 5 AND 6 REGIONS USING BRIDLE'S ALGORITHM

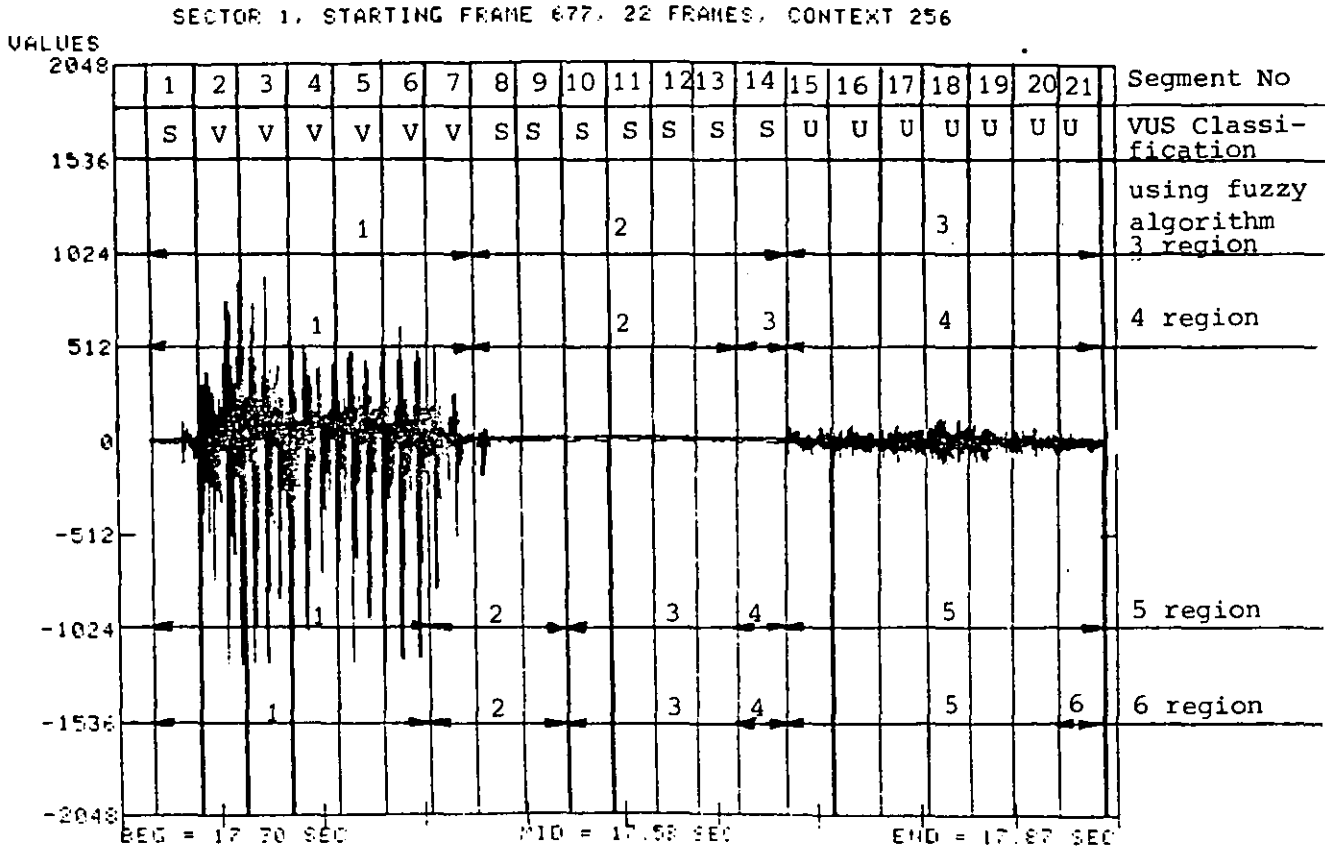


FIGURE 6.5: SEGMENTATION OF THE UTTERANCE 'X' INTO 3, 4, 5 AND 6 REGIONS USING BRIDLE'S ALGORITHM

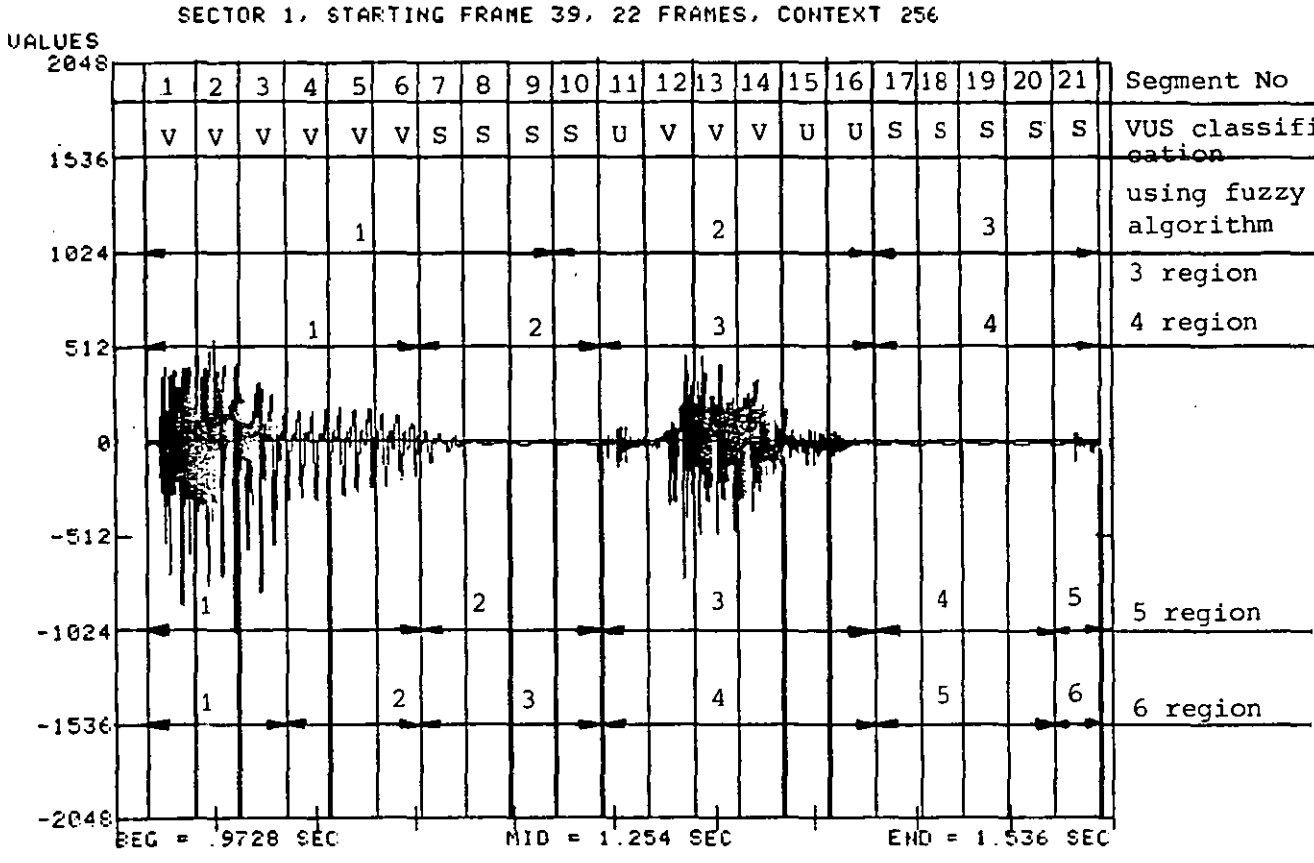


FIGURE 6.6: SEGMENTATION OF THE UTTERANCE 'INPUT' INTO 3, 4, 5 AND 6 REGIONS USING BRIDLE'S ALGORITHM

Most of the 50 words in the recognition vocabulary (apart from words like MULTIPLY, CONTROL, SEVEN) can be suitably described by 4 or less coarse regions. As such, a division of the utterances into 4 regions, was used in the word recognition systems described in subsequent sections.

6.4 THE WORD RECOGNITION SYSTEMS

The detection of the broad acoustic structure of speech utterances can be used in enhancing the performance of the word recognizer. The acoustic identity of an utterance is incorporated in a word recognizer either as a first pass section or as a parallel section, as described below.

6.4.1 Word Recognizers with a First Pass VUS-Based Recognizer

The structure of the recognition system is shown in Figure 6.7. The first pass is a VUS-based recognizer which outputs the identity of vocabulary words having a similar VUS structure to the input utterance. During the second pass, a conventional recognition process is employed, in which the input word pattern is matched only to those reference patterns identified in the first stage as likely candidates. The detailed recognition process is as follows.

An input word is segmented into 25.6 msec frames and 14 LPC coefficients are extracted after Hamming windowing the speech segments which have already been pre-emphasized through a first order network with a transfer function, $1-0.9z^{-1}$. Each frame is then classified as voiced, unvoiced or silence. This is accomplished by extracting from the speech segment, the five VUS parameters: zero-crossing rate count, the logarithmic energy, the unit delay autocorrelation coefficient, the first LPC coefficient and the normalized prediction error, and then applying the fuzzy set classification method. The fuzzy set thresholds which define the linguistic terms, low, medium, high, were obtained in a training session as described in Section 6.2. The next step in the VUS-based

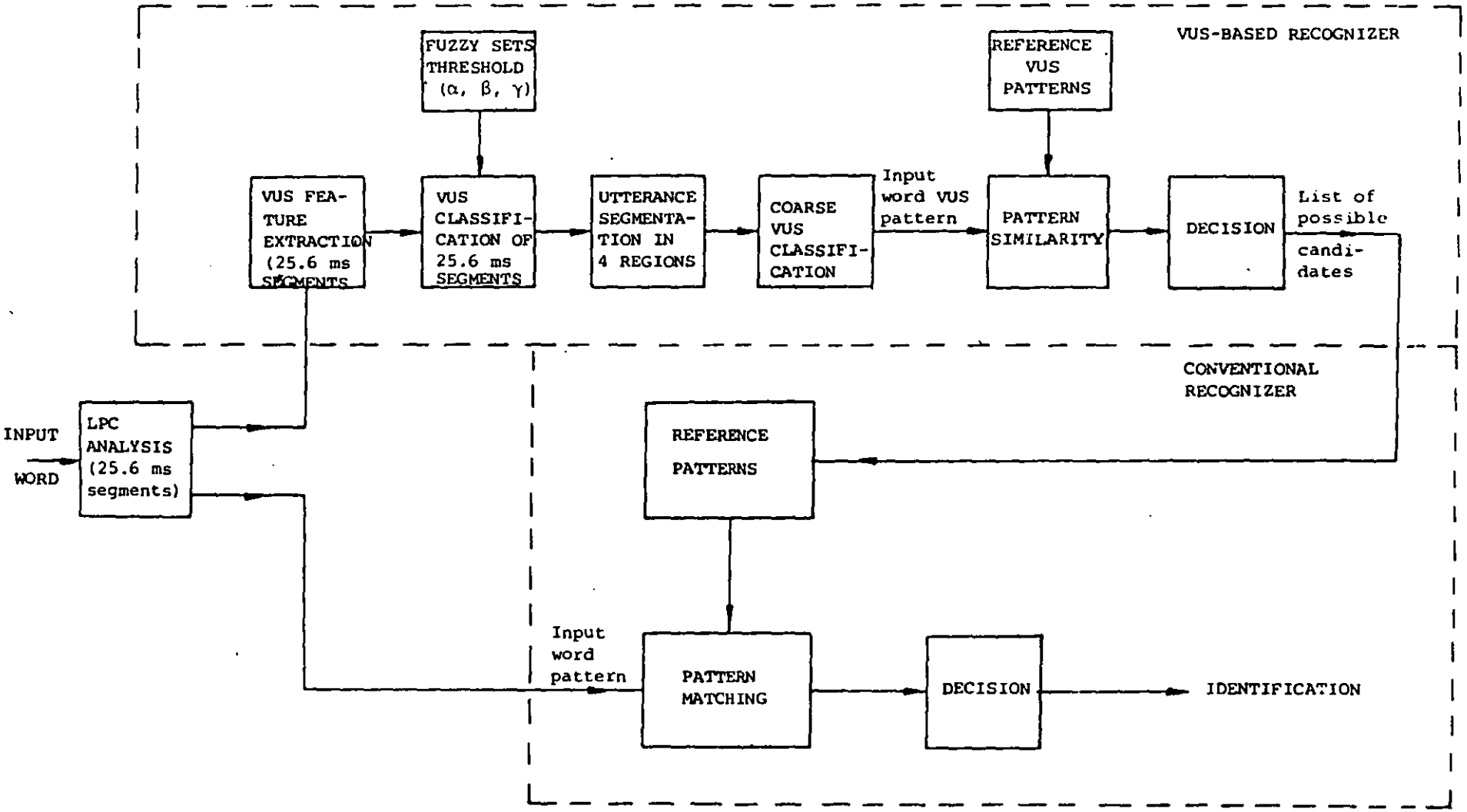


FIGURE 6.7: A CONVENTIONAL WORD RECOGNIZER WITH A FIRST PASS VUS-BASED WORD RECOGNIZER

recognizer, involves the division of the input word pattern into four regions, by employing Bridle's algorithm to locate the junctions in the sequence where the region boundaries are to be placed. Since the speech utterance is also expressed as a sequence of LPC vectors, the segment evaluation function in Bridle's algorithm employs the gain-normalized Itakura-Saito distance measure. Each of the four regions is then classified as voiced, unvoiced or silence, according to the identity of the majority of the segment contained within the region. The result of the above procedure, is that the input word is expressed as a sequence of four voiced, unvoiced or silence labels that indicate the broad acoustic structure of the word. This broad VUS pattern of the input word is compared with reference VUS patterns of vocabulary words generated in a similar manner during a training session. All the reference words, whose VUS patterns have the same structure as that of the input word, are identified as potential candidates.

The second pass of the recognition system, can employ any of the recognizers discussed in Chapter 5. The input utterance, described as a pattern of LPC vectors, is compared only with reference words identified as potential candidates in the first pass. The input word is then identified according to the decision rules in use by the particular recognizer, i.e. the nearest neighbour rule, or the KNN rule.

6.4.2 Word Recognizers with a Parallel VUS-Based Recognizer Section

The VUS-based recognizer can also be used as a parallel section to a conventional word recognizer. The composite recognition system is shown in the block diagram in Figure 6.8. The reference word patterns used in the recognition system are obtained during a training session.

During the testing session, an input word, A, is expressed as a sequence of LPC vectors and applied to the conventional recognizer to obtain an output V_1. The same word, A, is partitioned into four

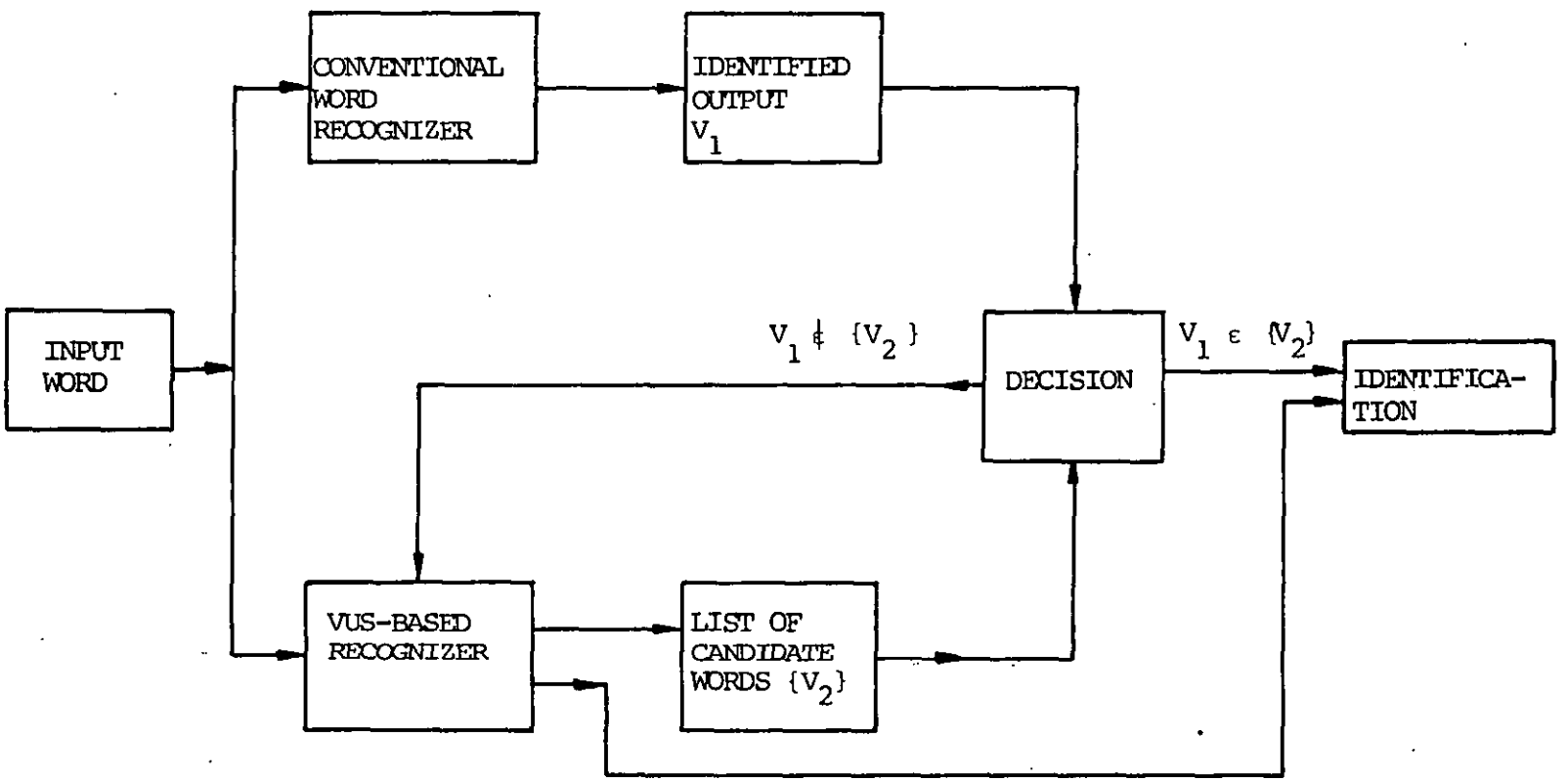


FIGURE 6.8: A CONVENTIONAL WORD RECOGNIZER WITH A PARALLEL VUS-BASED RECOGNIZER

regions using Bridle's algorithm and its VUS sequence identified using the fuzzy set theory method. The input word, as a VUS pattern, is applied to the VUS-based recognizer, which gives the output set of words, $\{V_2\}$, whose VUS structures are identical to that of the input. The outputs of both the conventional recognizer and the VUS-based recognizer are passed over to a decision stage. The input word is identified as V_1, if $V_1 \in \{V_2\}$, otherwise a feedback to the VUS-based recognizer is made if $V_1 \notin \{V_2\}$. Each word in the set $\{V_1 \cup V_2\}$, is expressed both as a sequence of LPC vectors, and as a sequence of four VUS segments. The following method was used to identify the input word in the set $\{V_1 \cup V_2\}$.

Let the word pattern $X \in \{V_1 \cup V_2\}$ be expressed as the discrete sequence of I, LPC vectors:

i.e. $$X = x_1, x_2, \ldots, x_i, \ldots, x_I \qquad 6.23$$

The Bridle's algorithm is used to partition X into four regions. Let the mth region, where $1 \leqslant m \leqslant 4$, contain the L, LPC vectors

$$x_i, x_{i+1}, \ldots, x_{i+L-1} \qquad 6.24$$

The mth region is then represented by the vector x_m, obtained from the autocorrelation coefficients vector $\bar{R}_m$ given by:

$$\bar{R}_m = \frac{1}{L} \sum_{j=1}^{L} R_{i+j-1} \qquad 6.25$$

where R_i is the autocorrelation coefficients vector which gives the LPC vector x_i.

Thus, the whole utterance, X, is represented by four LPC vectors, $(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4)$. The input pattern A is also reduced to a discrete pattern of four LPC vectors as described above. The distance between the input word pattern $A = \{\bar{a}_1, \bar{a}_2, \bar{a}_3, \bar{a}_4\}$ and the reference pattern X in $V_1 U \{V_2\}$, is given by:

$$D(X,A) = \sum_{i=1}^{4} d(\bar{x}_i, \bar{a}_i) \qquad 6.26$$

where $d(\bar{x}_i, \bar{a}_i)$ is the gain-normalized Itakura-Saito distance measure.

The unknown input word is identified as the reference word in the set $V_1 U \{V_2\}$, which gives the minimum distance.

6.4.3 Results

The influence of a first pass VUS-based recognizer, on the performance of conventional isolated word recognizers was investigated. The conventional recognizers used are those presented in Chapter 5, i.e:

i) The LPC-based recognizer with multiple reference patterns
ii) The LPC/SPLIT recognizer with multiple reference patterns
iii) The LPC/VQ recognizer with 16 entry reference codebooks
iv) The LPC/VQ/SPLIT recognizer with 16 entry reference codebooks.

The input words, spoken by a subject who did not contribute to the generation of reference patterns, were used for testing the recognition system. The recognition results obtained, as a percentage of correct identification of the input words, are given in Table 6.4. Similarly, the influence of the VUS-based recognizer, as a parallel section, on the performance of the above conventional recognizers was also assessed. The recognition results are given in Table 6.5.

TABLE 6.4

PERFORMANCE OF CONVENTIONAL WORD RECOGNIZER
WITH A FIRST-PASS VUS BASED RECOGNIZER

	RECOGNITION ACCURACY (%)			
RECOGNITION SYSTEM	Test 1 Test speaker SM1	Test 2 Test speaker SM3	Test 3 Test speaker SF1	Average
	Ref Speakers: SM2, SM3, SM4 SF2	Ref Speakers: SM1, SM2, SM4 SF2	Ref Speakers: SM1, SM2, SM4 SF2	
LPC-based recognizer (4 ref patterns per voc. word)	86	82	80	82.7
LPC/SPLIT (4 ref patterns/ voc word)	84	82	72	79.3
	Ref Speakers: SM2, SM3, SM4 SF2, SF3	Ref Speakers: SM1, SM2, SM4 SF2, SF3	Ref Speakers: SM1, SM2, SM4 SF2, SF3	
LPC/VQ 16 entries codebook	88	90	84	87.3
LPC/VQ/SPLIT 16 entries codebook	88	86	82	85.3

TABLE 6.5

PERFORMANCE OF CONVENTIONAL RECOGNIZERS
WITH A PARALLEL VUS BASED RECOGNIZER

	RECOGNITION ACCURACY (%)			
RECOGNITION SYSTEM	Test 1 Test speaker SM1	Test 2 Test speaker SM3	Test 3 Test speaker SF1	Average
	Ref Speakers: SM2, SM3, SM4 SF2	Ref Speakers: SM1, SM2, SM4 SF2	Ref Speakers: SM1, SM2, SM4 SF2	
LPC-based recognizer (4 ref patterns per voc. word from subjects SM1, SM2, SM4, SF2)	94	90	90	91.3
LPC/SPLIT (4 ref patterns/ voc. word from subjects SM1, SM2, SM4, SF2)	92	90	86	89.3
	Ref Speakers: SM2, SM3, SM4 SF2, SF3	Ref Speakers: SM1, SM2, SM4 SF2, SF3	Ref Speakers: SM1, SM2, SM4 SF2, SF3	
LPC/VQ 16 entries codebook	96	98	92	95.3
LPC/VQ/SPLIT 16 entries codebook	96	98	92	95.3

6.5 DISCUSSION

The Chapter commenced with a presentation of the VUS classification of speech segments. For this purpose, the five parameters - zero crossing rate count, energy, first delay autocorrelation coefficient, first LPC coefficient, normalized prediction error, were extracted from the speech segments. The results given in Tables 6.1 and 6.2 shows the VUS classification of speech segments in utterances, spoken by subjects who did not contribute to the training process, using the fuzzy set theory method and the Atal and Rabiner's method respectively. These results indicate that the accuracy obtained with fuzzy set method is comparable to Atal and Rabiner's method. The classification process using Atal and Rabiner's method, requires the computation of the matrix equation defined in equation 6.16, whereas for the fuzzy set approach, only the evaluation of the membership grades as defined in equation 6.8 are required. Thus the fuzzy set method offers a computationally simpler approach. However, both classification processes are still error prone. Most of the errors arise from the confusion between weak voiced segments and silence segments. Other errors occur in classifying segments in which the speech samples are changing from one class to another, i.e. when speech signal transitions are present within a segment. In such a case, the speech segment consists of samples from more than one class, and thus classifying the whole segment becomes ambiguous.

For recognition purposes, it would be desirable to obtain a general acoustic structure of the speech utterance, from its VUS classification of fine segments. This was achieved by using Bridle's algorithm of segmenting an utterance into a specific number of regions. Figures 6.3, 6.4 and 6.5 show the segmentation of the words 'SIX', 'X', 'INPUT', respectively, into 3, 4, 5 and 6 regions using Bridle's algorithm. The algorithm can be seen to be a powerful method of partitioning a speech utterance into a required number of regions, as the boundaries are placed at junctions where speech characteristics are in transition, subject to the frame size.

The identification of the VUS structure of an utterance was usefully exploited in enhancing the accuracy of a word recognition system. When the VUS-based recognizer was used as a parallel section to a conventional recognizer, the recognition accuracy was significantly improved. These improvements are shown in Table 6.6, which is a comparison of the results in Tables 5.1, 5.4 and 6.5. However, employing the VUS-based recognizer as a first pass section, results in a drop in recognition accuracy as shown in Table 6.6. The reason for this poor performance, is that any errors made in the first pass section are passed over to the conventional recognizer, which will subsequently make an erroneous identification. Such a situation would not arise when the VUS based recognizer is employed, as a parallel section to a conventional recognizer. This is because classification errors made in one section will not influence the decision process in the other section. Furthermore, when the parallel section is used, there is a provision for a feedback path if the outputs of the two sections are not in agreement. The LPC/VQ and the LPC/VQ/SPLIT systems both gave an accuracy of 95.3% when employing a parallel VUS-based recognizer section. This recognition accuracy was the highest among the systems under consideration in Table 6.6.

6.6 NOTE ON PUBLICATION

A paper entitled "Voiced-Unvoiced-Silence classification of speech using fuzzy set theory", in co-authorship with the supervisor, Dr C S Xydeas, has been published in the Proceedings of IEEE/ Mediterranean Electrotechnical Conference, held in Madrid, Spain, from 7-10 October, 1985, pp 123-126. The paper is based on the work presented in Section 6.2.

TABLE 6.6

A COMPARISON OF THE WORD RECOGNITION ACCURACY IN THE THREE SYSTEMS:

i) a conventional recognizer
ii) a conventional recognizer employing a parallel VUS-based recognizer
iii) a conventional recognizer employing a first pass VUS-based recognizer

CONVENTIONAL RECOGNIZER	Conventional recognizer without VUS-based recognizer	Conventional recognizer with parallel VUS-based recognizer	Conventional recognizer with first pass recognizer
LPC-based recognizer (4 reference patterns per vocabulary word)	89.3	91.3	82.7
LPC/SPLIT (4 reference patterns per vocabulary word)	83.1	89.3	79.3
LPC/VQ (16 entries codebook)	94	95.3	87.3
LPC/VQ/SPLIT (16 entries codebook)	92.6	95.3	85.3

CHAPTER 7

RECAPITULATION

7.1 INTRODUCTION

In this thesis, isolated word recognition systems based on pattern matching techniques have been studied and new improved techniques have been developed. The main objectives throughout have been to achieve a high recognition rate, whilst keeping the computational complexity and memory requirements at a minimum. Several word recognition techniques were investigated, and their performance in a speaker independent mode was assessed by computer simulations.

The initial work was mainly concerned with the problem of modelling the non-linear temporal fluctuations in a speech signal. This enabled the comparison of patterns of diverse temporal durations to be achieved. Speech utterances can be represented in a suitable form, for recognition purposes, by characteristic spectral features extracted from short temporal speech segments. Two spectral feature sets were considered, namely: filter bank features and LPC features. From simulation results, it was found that systems using speech utterances expressed as patterns of LPC features give a higher recognition accuracy than those using filter bank spectral estimates. This gave an impetus to further study of recognition systems which use speech patterns described by LPC features. However, it was quickly realised that the use of multiple reference patterns per vocabulary word, in order to achieve a speaker independent system raised considerably the memory requirements of the recognizer. Furthermore, the DTW techniques used in the pattern matching stage of the recognizer, were computationally expensive. The desire to solve these problems, led to the use of vector quantization techniques. A new recognition system, termed the LPC/VQ/SPLIT recognizer, was developed. In this system, reference patterns are stored as

sequences of Codebook entries, and in the pattern matching stage the need for using the computationally complex DTW is eliminated.

The accuracy of word recognizers can be enhanced further by taking into consideration the broad acoustic structure of input words. For this purpose, a new method of speech segment classification into voiced, unvoiced, and silence categories using the fuzzy set theory was proposed.

The general conclusions relating to the work carried out in each Chapter are reviewed in the following sections, followed by suggestions for further work and closing remarks.

7.2 TIME NORMALIZATION IN SPEECH PATTERNS

The main reason for exploring time normalization techniques in Chapter 3 was to obtain a suitable method that would be employed in the word recognition experiments. The Sakoe and Chiba DTW algorithm was found to be inadequate for the vocabulary words under consideration. This is because temporal differences between the input and reference patterns, exceeded the number of allowed frames mismatch in the algorithm. As such, Paliwal's modification on Sakoe/Chiba DTW algorithm was employed to correct the inadequacies. The DTW algorithm proposed by Itakura, was also investigated and found to exhibit similar deficiencies as the Sakoe/Chiba algorithm. That is, for example, an input word pattern cannot be matched with a reference pattern if the ratio of their temporal lengths is greater than 2. Myers' method, in which word patterns are transformed into patterns of fixed lengths, was used with Itakura's DTW algorithm and it was found to overcome these problems.

Myers' method was also used to provide fixed length patterns for the Sakoe and Chiba's algorithm. Two sets of vocabulary words, one of which is composed of acoustically similar words and the other of acoustically dissimilar words, were used to test the performance of

the following schemes: (i) Sakoe/Chiba DTW algorithm, (ii) Paliwal's modification as applied to Sakoe/Chiba DTW algorithm, (iii) Myers' method with Sakoe/Chiba DTW algorithm, and (iv) Itakura's DTW algorithm with fixed lengths patterns generated with Myers' method.

From a comparison of the word recognition results obtained with the above schemes, Paliwal's modification as applied to the Sakoe/Chiba DTW algorithm gave the highest accuracy. For this reason the word recognition systems described in subsequent Chapters, employed this particular algorithm in the pattern matching stage.

The concern for the heavy computational complexity of the DTW algorithm led to a consideration of Brown and Rabiner's graph search technique in Section 3.5. From the simulation results, the graph search technique, while offering less computations, resulted in a drop in word recognition accuracy as opposed to the DTW methods. As such, the graph search method was abandoned in favour of the DTW method. However, other computational load reduction methods were investigated and are described in Chapter 5.

7.3 THE USE OF FILTER BANK FEATURES IN WORD RECOGNITION

The representation of speech utterances as discrete patterns of energy values in selected frequency bands, and their subsequent use in a word recognition system, was considered in Chapter 4. The FIR filter bank systems, designed to cover the 0-5 kHz bandwidth of the speech signals are: (i) an 8-channel, 1/3 octave spaced filter bank, (ii) a 5-channel, ideal octave spaced filter bank, (ii) 5, 8, 10, 12 and 16 channel uniformly spaced filter banks, (iv) 16 channel critical band spaced filter bank. The accuracy obtained in the isolated word recognizer using speech utterances processed by different filter banks was compared. First, the word recognition systems were tested in a speaker independent mode using a single reference pattern per vocabulary word, and then in a multiple reference pattern per vocabulary word situation. The results of these experiments show an overwhelming superiority of multiple

reference patterns over single reference pattern systems. In addition the 8 channel, 1/3 octave spaced filter bank gave a better recognition accuracy (82.0%) than the other filter bank systems under consideration.

Attempts were also made to improve on the accuracy of multiple reference systems, by suppressing the redundancy present in speech patterns. Two redundancy suppression methods, namely: trace segmentation and a simple redundancy removal method, were employed as discussed in Section 4.5. In the 8 channel, 1/3 octave filter bank word recognition system, an improvement in recognition accuracy by 5.5% is obtained when the simple redundancy method is used to compress the speech patterns by a factor of 0.9, as shown in Figure 4.8. Although these redundancy suppression methods lead to improvement in recognition accuracy, their use would be impeded by the difficulty involved in estimating the level of redundancy in the speech pattern.

7.4 THE USE OF LPC FEATURES IN WORD RECOGNITION

The word recognition system using patterns of LPC features was considered in Chapter 5. The first issue was to assess the influence of the prediction order of the LPC model on the recognition accuracy. An average recognition accuracy of 89.3%, as shown in Table 5.1, was obtained when a 14th order LPC analysis was used. A comparison of the average accuracy of the word recognizer employing an 8-channel, 1/3 octave filter bank (Table 4.6), and the LPC based recognizer with 14th order LPC coefficient (Table 5.1), reveals the superior performance of the LPC based system. It was thus decided to proceed with further investigations of word recognizers employing LPC features.

The acoustic similarity of some vocabulary words is an obvious source of recognition errors. A method of generating discriminative reference patterns for similar sounding words in the vocabulary was used to reduce such errors. An average recognition accuracy of 91.3%

was obtained on using the discriminative reference patterns for the words set {B, C, D, E, G, P, T, V}. However, the use of multiple reference patterns in the recognition systems, imposes a high computational load in the recognition process. For this purpose, the computational cost reduction method of Section 5.4.2 was proposed. In this method, reference patterns are clustered into a disjoint number of groups, and each group is represented by a cluster centroid. The input word is first compared with the cluster centroids of the various groups, and then only with the reference patterns associated with the best match cluster centroid. The reduction in computational cost is dependent on the number of clusters and their occupancy, as illustrated in Figure 5.9. From the simulation results, a computational reduction of 10:3 was obtained at a slight drop in recognition accuracy by 1.5%.

Next, attention was focussed on (i) large memory requirements of the LPC based word recognizer in storing the reference patterns, and (ii) on the complexity of the DTW algorithm during the pattern matching process. These considerations led to the use of vector quantization techniques. Two established recognizers, termed the LPC/SPLIT and the LPC/VQ systems, were studied by computer simulations. The LPC/SPLIT word recognition system operates with a reduced memory requirement but still uses the DTW process, whereas the LPC/VQ system requires a large memory space, but has the advantage of eliminating the need for the DTW process. Based on the characteristics of these two recognizers, a hybrid system termed the LPC/VQ/SPLIT system in which the advantages of both recognizers were preserved, was developed. From the computer simulation results given in Table 5.4, and the memory characteristics illustrated in Figure 5.17, the advantages offered by the LPC/VQ/SPLIT can be deduced, i.e. a high recognition accuracy (92.6%) and low memory/computational complexity characteristics.

7.5 THE USE OF VOICED, UNVOICED AND SILENCE CLASSIFICATION OF SPEECH SEGMENTS IN WORD RECOGNITION

In Chapter 6, a method for enhancing the accuracy of a word recognition system by identifying the broad acoustic structure of the input speech utterance is formulated. The method requires as a prerequisite, a reliable classification of speech segments into voiced, unvoiced and silence classes. A new and simple technique which applies the fuzzy set theory to obtain such classification was developed, and found to offer comparable performance to the established but complex statistical method of Atal and Rabiner (Tables 6.1 and 6.2). The segmentation of an utterance into temporal durations of the order of 25.6 msec, and the subsequent classification into voiced, unvoiced and silence classes yields a 'fine' acoustic structure which is not suitable for recognition purposes. Bridle's algorithm for dividing an utterance into a few regions was therefore employed in obtaining the coarse acoustic structure. Two strategies of supplementing a conventional word recognition system with the coarse VUS structure of speech utterances were then investigated. The VUS-based recognizer was initially used as a first pass section and a conventional recognizer as the second pass. It was observed that, in such a system, the recognition accuracy actually decreased, as indicated by results in Tables 5.4 and 6.4. The deterioriation in recognition accuracy is due to errors in the first pass stage being carried over to the conventional recognizer.

It was therefore decided to investigate the effects of using the VUS-based recognizer as a parallel section to the conventional word recognizer. The results of these investigations are given in Table 6.5. A comparison of the accuracy obtained with conventional systems, as given in Tables 5.1 and 5.4, and when a parallel VUS recognizer is used, as given in Table 6.5, reveals an improvement in recognition accuracy as follows: (i) 2% increase in the LPC based word recognizer, (ii) 6.2% increase in the LPC/SPLIT system, (iii) 1.3% increase in the LPC/VQ system employing 16 entry codebooks, and

(iv) 2.7% increase in the LPC/VQ/SPLIT system employing 16 entry codebooks.

The LPC/VQ/SPLIT and the LPC/VQ systems employing 16 entry codebooks gave an identical recognition accuracy of 95.3%, when the VUS-based recognizer was used as a parallel section. The recognition accuracy was higher than obtained with the other systems under consideration.

7.6 SUGGESTIONS FOR FURTHER WORK

As a further extension to the work discussed in this thesis, the following suggestions are made:

i) The use of trace segmentation and redundancy removal methods, in word recognition systems, as discussed in Section 4.5, provided an improvement in recognition accuracy for small compression factors. Such results serve to strengthen the premises that the information contained in transitional regions of an utterance play a more important role than the stationary regions in the recognition process. Thus, if one could extract more features during transitions rather than in stationary regions, it can be envisaged that the recognition accuracy could be enhanced. Such an approach has been considered by Watari et al [91], but still requires an accurate detection of transitional regions in the speech signal.

ii) It is considered that further research into certain aspects of the design of vector quantization codebooks may yield improved results. In Section 5.5, the centroid of a group of LPC vectors was computed from the average autocorrelation vector of the whole group. A better centroid, in terms of the distance to any vector within the group, would probably be obtained as the vector whose maximum distance to any other vector is minimum, i.e. minimax. Such centroids would give rise to codebooks that represent the training set with less distortion. Also, the VQ design methods employed in the same section,

namely: the binary splitting algorithms, may possibly be improved (in terms of convergence rate and not distortion level) by the use of more sophisticated 'multiple splitting' strategies.

iii) The VUS-based recognizer was used either as a first-pass section or as a parallel section added to a conventional recognizer. It could not be used in isolation because its acoustic description of an utterance is too generalized. A more exact representation of the acoustic structure would require the classification of segments into more than the three classes. For example, the voiced class can be split into vowels, semivowels, diphthongs, voiced-stops, voiced fricatives. The unvoiced class can similarly be split into unvoiced fricatives and unvoiced stops. Features which can help to identify these subgroups have been studied by Ruske [37] and by Zue et al [92]. The fuzzy set theory can be applied to model the classification into a similar manner as proposed in this thesis. However, such work would require the use of a spectrograph, especially for the training procedures.

7.7 CLOSING REMARKS

The last 15 years have seen spectacular and significant advances in the general area of speech recognition and in particular isolated word systems. The formulation of the Itakura-Saito distance measure and the DTW techniques are landmarks in isolated, as well as connected word recognition systems. The strategy, emphasized in the ARPA programme, of combining several knowledge sources in order to attain successful understanding of speech sentences, has been accepted as the key to continuous speech recognition.

However, in spite of the significant progress, there are still several problems to be solved. These include the recognition of: speech degraded by noise, telephone bandwidth speech, speech from uncooperative subjects, speech distorted by the environment e.g.

'helium' speech. The incorporation of certain knowledge sources like pragmatics and prosody in continuous speech recognition is yet to be solved. The solution to these problems will require the cooperation with scientists in other fields such as artificial intelligence, linguistics, ergonomics and psychology.

The developments in microtechnology have produced powerful signal processing chips like NEC uPD7720 and Texas Instrument's TMS320. These chips have been available for the last five years and can be programmed to perform a number of speech processing algorithms in real time. The most recent speech recognition chip available from NEC is the uPD7764. This new chip has been designed specifically for speech recognition yet comprises two independent general purpose processors (labelled the D-processor and the G-processor) holding their programs in RAM. The calculation of the distance between two vectors of filter bank energy features are performed in the D-processor. The DTW algorithm is performed in the G-processor. However, the extraction of the feature vectors is not performed by the 7764 but by a 7763 spectrum analyzer. With these new chips, NEC claims the implementation of an isolated word recognition system, with a 380 word vocabulary, operating with a response time of 300 msec. The features used are 16 dimensional filter bank energies of 16 msec speech segments in the spectral range 250 Hz-5400 Hz. Many other manufacturers can also be expected to develop similar devices possessing great potential for speech recognition. Thus, the widespread use of speech recognition will most likely be held up by theoretical rather than technical aspects.

The author hopes that the effort put in the work described in this thesis makes a contribution to the future development of isolated word recognition.

REFERENCES

1. WEINSTEIN, S.B:
"Scenes from the life of Alexander Graham Bell", A digest of news and events of interest to the IEEE Communication Society, Vol 14, No 1, Jan 1976, pp 13-20.

2. TURN, R:
"The use of speech for man-computer communication". Rand Report 1386-ARPA, Rand Corp., Santa Monica, CA, 1974.

3. NEWEL, A; BARNETT, J; FORGIE, J; GREEN, C; KLATT, D.H; LICKLIDER, J.C.R; MUNSION, J; REDDY, D.R. and WOODS, W.A:
"Speech understanding systems", Final report of a study group, Carnegie-Mellon University, Pittsburg PA, 1971.

4. CHAPANIS, A; PARRISH, R.N; OCHSMAN, R.B. and WEEKS, G.O:
"Studies in interactive communication II: the effects of four communication modes on the linguistic performance of teams during cooperative problem solving", Human Factors 19, No 2, 1977, pp 101-126.

5. LEONARD, R.G. and DODDINGTON, G.R:
"Automatic classification of languages", Rome Air Development Centre, Rome NY, Tech Report, TR 75-264, 1975.

6. WATANABE, A; UEDA, Y. and SHIGENADA, A:
"Colour display system for connected speech to be used for the hearing impaired". IEEE Trans ASSP, Vol 33, 1985, pp 164-173.

7. TOSI, O:
"Voice identification: theory and legal applications", Univ Park Press, Baltimore, MD, USA, 1979.

8. MARTIN, T.B. and WELCH, J.R:
"Practical speech recognizers and some performance effective parameters". Trends in speech recognition, W.A. Lea (Ed), Prentice-Hall Inc., Englewood Cliffs, NJ, 1980, pp 24-38.

9. WILPON, J.G. and ROBERTS, L.A:
"The effects of instruction and feedback on speaker independent consistency for automatic speech recognition", IEE Inter Conf on Speech Input/Output, No 258, Mar 1986, pp 242-247.

10. FLANAGAN, J:
"Voices of men and machines". Electronic speech synthesis. G. Bristow (Ed), Granada Publishing Ltd, 1984, pp 48-69.

11. MARKEL, J.D. and GRAY, A.H:
Linear prediction of speech". Springer-Verlag, Berlin, Heidelberg, New York, 1976.

12. GIANNINI, A:
"Spectrographic and electro-aerometric analysis on aspirated and unaspirated clicks in Zulu". IEE Inter Conf on Speech Input/ Output, No 258, March 1986, pp 298-303.

13. FLANAGAN, J.L:
"Speech analysis, synthesis and perception". 2nd ed., New York, Springer-Verlag, 1972.

14. DUDLEY, H:
"Remaking speech". J. Acoustic Soc of Amer, Vol 11, 1939, pp 169-177.

15. POTTER, R; KOPP, G. and GREEN, H:
"Visible speech", New York, Van Nostrand, 1947.

16. DAVIS, K.H; BIDDULPH, R. and BALASHEK:
"Automatic recognition of spoken digits". J. Acoustic Soc of Amer, Vol 24, No 6, 1952, pp 637-642.

17. DUDLEY, H. and BALASHEK, S:
"Automatic recognition of phonetic patterns in speech", J. Acoustic Soc of Amer, Vol. 30, No 8, 1954, pp 721-733.

18. FRY, D.B. and DENES, P:
"The solution of some fundamental problems in mechanical recognition of speech". Language and Speech, Vol 1, 1959, pp 35-38.

19. DENES, P. and MATTHEWS, M.V:
"Spoken digit recognition using time-frequency pattern matching". J. Acoustic Soc of Amer, Vol. 32, No, 11, 1960, pp 1450-1455.

20. FORGIE, J.W. and FORGIE, C.D:
"Results obtained from vowel recognition computer program", J. Acoustic Soc of Amer, Vol. 31, No 11, 1959, pp 1480-1489.

21. COOLEY, J.W. and TUKEY, J.W:
"An algorithm for the machine calculation of complex Fourier series". Math Computers, Vol. 19, 1965, pp 297-301.

22. NAGATA, K; KATO, Y. and CHIBA, S:
"Spoken digit recognizer for Japanese language", NEC Res Develop, No 6, 1963.

23. MUSMANN, H.G. and STEINER, K.H:
"Phonetische Addiermaschine", Arch Elek Ubertagung, Vol. 19, 1965, pp 502-510.

24. OTTEN, K.W:
"Automatic recognition of continuous speech". Technical report No AFAL-TE-66-408. A.F. Avionics Laboratory, Wright-Patterson AFB, Ohio, USA.

25. KLATT, D.H:
"Review of the ARPA speech understanding project". J Acoustic Soc of Amer, Vol 62, Dec 1977, pp 1345-1366.

26. BAKER, J.K:
"The DRAGON system - an overview". IEEE Trans. ASSP, Vol 23, Feb 1975, pp 24-29.

27. JELINEK, F; BAHL, L.R. and MERCER, R.L:
"Design of a linguistic statistical decoder for the recognition of continuous speech". IEEE Trans. Inform. Theory, Vol 21, May 1975, pp 250-256.

28. ITAKURA, F:
"Minimum prediction residual principle applied to speech recognition". IEEE Trans. ASSP, Vol 23, Feb 1975, pp 67-72.

29. HARSHMAN, R; LADEFOGED, P. and GOLDSTEIN, L:
"Factor analysis of tongue shapes", J. Acoustic Soc Amer, Vol 62, 1977, p 693.

30. SHIRAI, K. and HONDA, M:
"Estimation of articulatory parameters from speech waves". Trans Inst Elect and Comm Eng (Japan), Vol E61, No 5, 1978, pp 382-383.

31. COKE, C.M. and FUJIMURA, O:
"Model for specification of the vocal-tract area function". J. Acoustic Soc Amer, Vol 40, 1966, p 1271.

32. GERSHO, A:
"On the structure of vector quantizers". IEEE Trans Inf Theory, Vol 28, No 2, March 1982, pp 157-166.

33. RABINER, L.R; LEVINSON, S.E. and SONDHI, M.M:
"On the application of vector quantization and hidden Markov models to speaker-independent isolated word recognition". BSTJ, Vol 162, No 4, April 1983, pp 1075-1105.

34. BAUM, L.E:
"An equality and associated maximization technique in statistical estimation of probabilistic functions of Markov processes". Inequalities, Vol 3, 1972, pp 1-8.

35. SAKOE, H:
"Two level DP-matching - a dynamic programming based pattern matching algorithm for connected word recognition". IEEE Trans ASSP, Vol 27, Dec 1979, pp 588-595.

36. MYERS, C.S. and RABINER, L.R:
"Connected digit recognition using a level building DTW algorithm", IEEE Trans. ASSP, Vol 29, June 1981, pp 351-363.

37. RUSKE, G:
"Auditory perception and its application to computer analysis of speech". Computer analysis and perception, Vol II: Auditory signals, C.Y. Suen and R. de Mori (Eds), CRC Press Inc, Boca Raton, FIA, 1982, pp 1-42.

38. SHROUP, J.E:
"Phonological aspects of speech recognition", in Trends in speech recognition. W.A. Lea (Ed), Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1980, pp 125-138.

39. REDDY, D.R:
"Speech recognition by machine - a review". Proc IEEE, Vol 64, Apr 1976, pp 501-531.

40. WARREN, R.M:
"Auditory perception". Pergamon Press, New York, 1982.

41. CHOMSKY, N:
"On certain formal properties of grammar". Inform. Control, Vol 2, 1959, pp 137-167.

42. LEVINSON, S.E:
"The effects of syntactic analysis on word recognition accuracy". BSTJ Vol 57, May-June 1978, pp 1627-1644.

43. SAKOE, H. and CHIBA, S:
"Dynamic programming algorithm optimization for spoken word recognition" IEEE Trans. ASSP, Vol 26, Feb 1978, pp 43-49.

44. RABINER, L.R; ROSENBERG, A.E. and LEVINSON, S.E:
"Considerations in dynamic time warping algorithms for isolated word recognition". IEEE Trans. ASSP, Vol 26, Dec 1978, pp 575-582.

45. PALIWAL, K.K; AGARWALL, A. and SINHA, S.S:
"A modification over Sakoe and Chiba's dynamic time warping algorithm for isolated word recognition". Proc. Inter. Conf. ASSP, May 1982, pp 1259-1261.

46. MYERS, C; RABINER, L.R. and ROSENBERG, A.E:
"Performance trade-offs in dynamic time warping algorithms for isolated word recognition". IEEE Trans. ASSP, Vol. 28, Dec 1980, pp 623-635.

47. SILVERMAN, H.F. and DIXON, N.R:
"State constrained dynamic programming (SCDP) for discrete utterance recognition". Proc. Inter. Conf. ASSP, April 1980, pp 169-172.

48. BROWN, M.K. and RABINER, L.R:
"An adaptive, ordered, graph search technique for dynamic time warping for isolated word recognition". IEEE Trans. ASSP, Vol 30, Aug 1982, pp 535-544.

49. NILSSON, N.J:
"Problem-solving methods in artificial intelligence". McGraw Hill Book Company, New York, 1971.

50. DAUTRICH, B.A; RABINER, L.R. and MARTIN, T.B:
"On the effects of varying filter bank parameters on isolated word recognition". IEEE Trans. ASSP, Vol 31, No 4, Aug 1983, pp 793-807.

51. OPPENHEIM, A.V. and SCHAFER, R.W:
"Digital signal processing", Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1975.

52. BEKESY, G:
"Experiments in hearing", McGraw Hill, New York, 1960.

53. WILSON, J.P:
"Basilar membrane data and their relation to theories of frequency analysis", in Facts and Models in hearing. E. Zwicker and E. Terhardt (Eds), Springer-Verlag, Heidelberg, 1974.

54. SENEFF, S:
"A computational model for the peripheral auditory system". Inter. Conf. ASSP, April 1986, pp 1983-1986.

55. SILVERMAN, H.F. and DIXON, N.R:
"A comparison of several speech-spectra classification methods". IEEE Trans. ASSP, Vol 24, Aug 1976, pp 289-295.

56. KLATT, D.M:
"Prediction of perceived phonetic distance for critical band spectra". Proc. Inter. Conf. ASSP, May 1982, pp 1278-1281.

57. KUHN, M.H; TOMASCEWSKI, H. and NEY, H:
"Fast nonlinear time alignment for isolated word recognition". Inter. Conf. ASSP, Mar/April 1981, pp 736-740.

58. PIERACCINI, R:
"Pattern compression in isolated word recognition". Signal Processing, Vol 7, No 1, 1984, pp 1-15.

59. RABINER, L.R. and SCHAFER, R.W:
"Digital processing of speech signals". Prentice-Hall Inc, Englewood Cliffs, New Jersey, 1978.

60. MAKHOUL, J:
"Linear prediction: a tutorial review". Proc. IEEE, Vol 63, April 1975, pp 561-580.

61. FANT, G:
"Acoustic theory of speech production". Gravenhage Mouton and Co., The Hague, 1960.

62. HILDEBRAND R.B:
"Introduction to numerical analysis". McGraw Hill Book Company, New York, 1956.

63. FADDEEV, D.K. and FADDEEVA, V.N:
"Computational methods of linear algebra". San Francisco, California, Freeman, 1963.

64. LEVINSON, N:
"The Wiener RMS error criterion in filter design and prediction". J. Math. Phys. Vol 25, No 4, 1947, pp 261-278.

65. DURBIN, J:
"The fitting of time-series models". Rev. Inter. Inst. Statistics, Vol 28, No 3, 1960, pp 233-243.

66. ITAKURA, F. and SAITO, S:
"Analysis synthesis telephony based on the maximum likelihood method". Reports of the 6th Int. Congr. Acoustics, Vol II, c-5-4, 1968.

67. ITAKURA, F. and SAITO, S:
"A statistical method for the estimation of speech spectral density and formant frequencies". Electronic Comm. Japan, Vol 53-A, 1970, pp 36-46.

68. MATSUYAMA, Y; BUZO, A. and GRAY, R.M:
"Spectral distortion measures for speech compression". Information Systems Lab., Stanford University, California, Tech. Rep. 6504-3, April 1978.

69. GRAY, A.H. and MARKEL, J.D:
"Distance measure for speech processing". IEEE Trans. ASSP, Vol 24, Oct 1976, pp 380-391.

70. GRAY, R.M; BUZO, A; GRAY, A.H. and MATSUYAMA, Y:
"Distortion measures for speech processing". IEEE Trans. ASSP, Vol 28, No 4, Aug 1980, pp 365-376.

71. de SOUZA, P.V:
"Statistical tests and distance measures for LPC coefficients". IEEE Trans. ASSP, Vol 25, Dec 1977, pp 554-559.

72. MOORE, R.K; RUSSELL, M.J. and TOMLINSON, J:
"Discriminative network: a mechanism for focusing recognition in whole-word pattern matching". Inter. Conf. ASSP, 1983, pp 1041-1043.

73. WILPON, J.G. and RABINER, L.R:
"A modified K-means clustering algorithm for use in isolated word recognition". IEEE Trans. ASSP, Vol 33, No 3, June 1985, pp 587-594.

74. GRAY, R.M:
"Vector quantization". IEEE ASSP Magazine, April 1984, pp 4-29.

75. GERSHO, A. and CUPERMAN, V:
"Vector quantization - a pattern matching technique for speech coding". IEEE Communications Magazine, Dec 1983, pp 15-21.

76. SHANNON, C.E:
"A mathematical theory of communication". BSTJ Vol 27, 1948, pp 379-423.

77. LLOYD, S.P:
"Least squares quantization in PCM". Unpublished Bell Lab. Tech. Note, 1957. Published in IEEE Trans. Information Theory, Vol IT-28, No 2, March 1982, pp 129-137.

78. LINDE, Y; BUZO, A. and GRAY, R.M:
"An algorithm for vector quantizer design". IEEE Trans. on Comm. Vol 28, Jan 1980, pp 84-95.

79. BUZO, A; GRAY, A.H; GRAY, R.M. and MARKEL, J.D:
"Speech coding based upon vector quantization". IEEE Trans. ASSP, Vol 28, No 5, Oct 1980, pp 562-574.

80. RABINER, L.R; SONDHI, M.M. and LEVINSON, S.E:
"Note on the properties of a vector quantizer for LPC coefficients". BSTJ Vol 62, Oct 1983, pp 2603-2616.

81. SUGAMURA, N;: SHIKANO, K. and FURUI, S:
"Isolated word recognition using phoneme-like templates". Inter. Conf. ASSP, Vol 1, 1983, pp 723-726.

82. SHORE, J.E. and BURTON, D:
"Discrete utterance speech recognition without time alignment". IEEE Trans. IT, Vol 29, No 4, July 1983, pp 473-491.

83. MWANGI, E. and XYDEAS, C.S:
"The use of phoneme-like templates in isolated word recognition". Proc. 3rd European Signal Process. Conf. (EUPISCO), Sept 1986, pp 561-564.

84. ATAL, B. and RABINER, L.R:
"A pattern recognition approach to voiced-unvoiced-silence classification with application to speech recognition". IEEE Trans. ASSP, Vol 24, No 3, June 1976, pp 201-212.

85. MWANGI, E. and XYDEAS, C.S:
"Voiced-unvoiced-silence classification of speech using the fuzzy set theory". Proc. IEEE/Mediterranean Electrotechnical Conf. Oct, 1985, pp 123-126.

86. ZADEH, L.A:
"Fuzzy sets". Information and Control. Vol 8, 1965, pp 338-353.

87. ZADEH, L.A; FU, K.S; TANAKA, K. and SHIMURA, M. (Eds):
"Fuzzy sets and their application to cognitive and decision process". London, Academic Press, 1975.

88. KANDEL, A:
"Fuzzy techniques in pattern recognition", John Wiley and Sons, New York, 1982.

89. DUDA, R.O. and HART, P.E:
"Pattern classification and scene analysis". John Wiley and Sons Inc, New York, 1973.

90. BRIDLE, J.S. and SEDGEWICK, N.C:
"A method for segmenting acoustic patterns with applications to automatic speech recognition". Proc. IEEE Inter. Conf. ASSP, 1977, pp 656-659.

91. WATARI, M; AKABANE, M. and SAKO, Y:
"A speaker independent word recognition based on transient matching". Inter. Conf. ASSP, Vol 2, April 1983, pp 715-718.

92. ZUE, V.W. and SCHWARTZ, R.M:
"Acoustic processing and phonetic analysis". Trends in Speech Recognition (W.A. Lea, Ed), Englewood Cliffs, N.J. Prentice Hall, 1980, pp 101-124.

93. BLACKMAN, R.B. and TUKEY, J.W:
"The measurement of power spectra". Dover Publications Inc, New York, 1958.

94. KAISER, J.F:
"Non-recursive digital filter design using the I_0-sinh window function". Proc. IEEE Inter. Symp. on Circuits and Systems, San Francisco, April 1974, pp 20-23.

95. GIBBS, A.J:
"On the frequency domain responses of causal digital filters". PhD Thesis, University of Wisconsin, USA, 1969.

APPENDICES

APPENDIX A

THE WINDOW DESIGN METHOD FOR FIR FILTERS

The system function of an FIR filter is of the form:

$$H(z) = \sum_{n=0}^{N-1} h(n)\, z^{-n} \qquad \text{A.1}$$

where $h(n)$, $0 \leqslant n \leqslant N-1$, is the impulse response.

The window design technique starts with a specification of the required frequency response, $H(e^{j\omega})$, of the filter. Figure A.1(a) shows the amplitude/frequency characteristics of an ideal low pass filter (LPF). Since the frequency response of any digital filter is periodic in frequency, then it can be expanded as a Fourier series as follows:

$$H'(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h'(n) e^{-j\omega n} \qquad \text{A.2}$$

where $h'(n)$ is the corresponding impulse response sequence

i.e.

$$h'(n) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} H'(e^{j\omega})\, e^{j\omega n}\, d\omega \qquad \text{A.3}$$

The impulse response is shown in Figure A.2(b).

A finite duration of the impulse response can be obtained from $h'(n)$ by a simple truncation process, as follows:

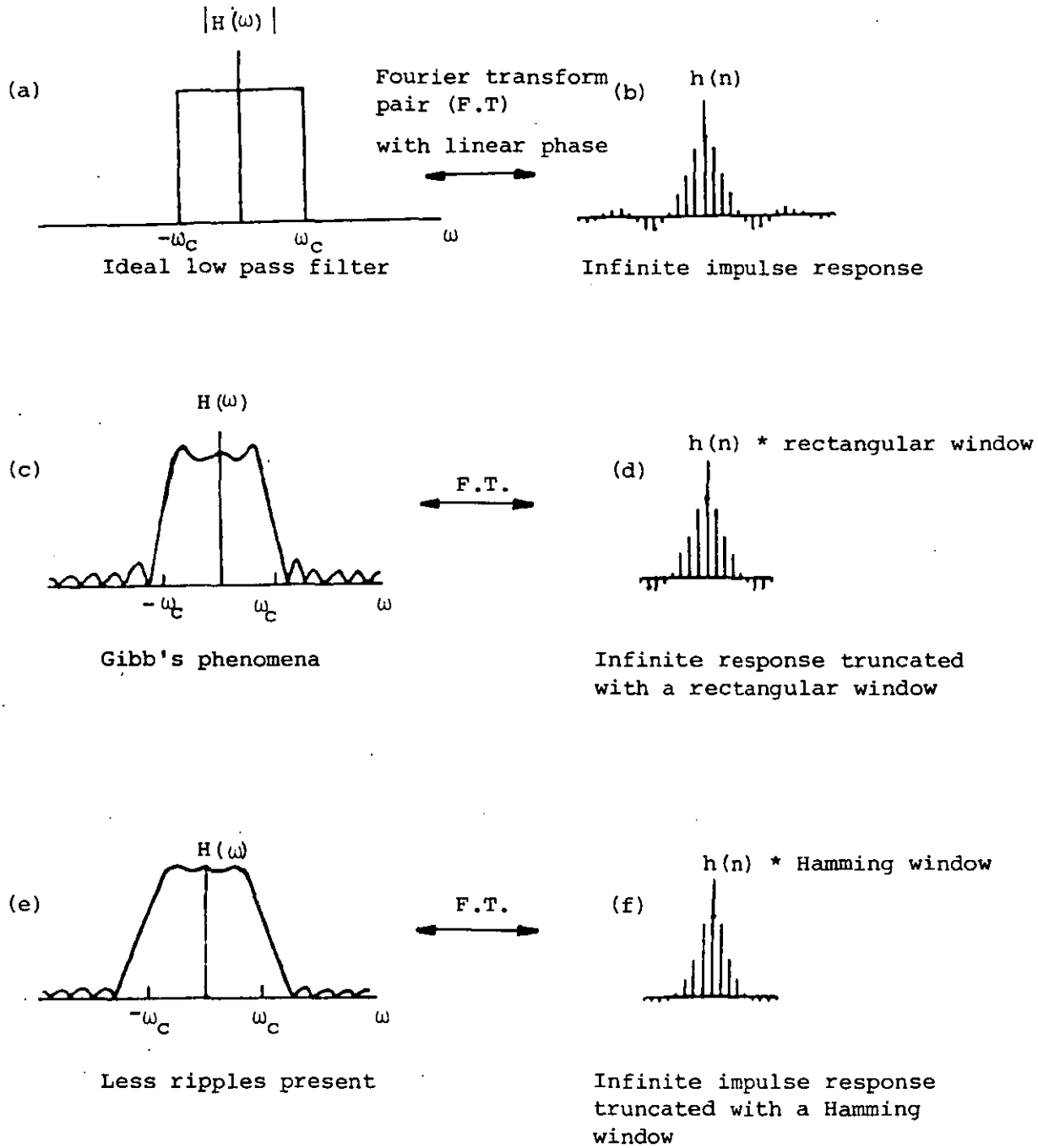


FIGURE A.1: THE DESIGN OF AN IDEAL LOW PASS FILTER USING THE WINDOW APPROACH TECHNIQUE

$$h(n) = \begin{cases} h'(n) , & 0 \leq n \leq N-1 \\ 0 , & \text{otherwise} \end{cases} \qquad \text{A.4}$$

In general, h(n) can be represented as a product of the desired impulse response, h'(n) and a 'window', w(n), of finite duration, i.e.

$$h(n) = w(n)\, h'(n) \qquad \text{A.5}$$

In equation A.4, w(n) is a rectangular window defined as:

$$w(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{otherwise} \end{cases} \qquad \text{A.6}$$

The resultant frequency response, $H(e^{j\omega})$, shown in Figure A.1(c) is given by:

$$H(e^{j\omega}) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} H'(e^{j\theta})\, W(e^{j(\omega-\theta)})\, d\theta \qquad \text{A.7}$$

i.e. it is the periodic continuous convolution of the desired frequency response with the Fourier transform, $W(e^{j\omega})$ of the window. Although the rectangular window gives a sharp cut-off frequency, the presence of ripples in the passband is undesirable. A number of window functions (Hamming, Hanning, Blackman, Kaiser etc) have been proposed [93,94], and are used in order to smooth out the ripples i.e. to reduce the effects of the Gibbs phenomena [95]. However,

these window functions achieve a moderation in the Gibbs phenomena at the expense of a wider passband-stopband transition region.

Figure A.1(e) shows the frequency response obtained on using the Hamming window in the impulse response truncation process. A Hamming window, W(n) is defined as follows:

$$W(n) = 0.54 - 0.46 \cos (2\pi n/N-1), \quad 0 \leqslant n \leqslant N-1 \qquad A.8$$

APPENDIX B

PROPERTIES OF THE AUTOCORRELATION COEFFICIENTS OF THE IMPULSE RESPONSE OF THE ALL-POLE MODEL

For the all-pole filter, defined by the transfer function, H(z),

$$H(z) = \frac{G}{1 + \sum_{K=1}^{p} a(k)\ z^{-k}} \qquad \text{B.1}$$

where G is the gain, and a(k), k = 1,2, ... p, are the predictor coefficients. The impulse response, h(n) of the filter is

$$h(n) = \begin{cases} 0 & , \text{ for } n < 0 \\ G & , \text{ for } n = 0 \\ \Sigma a(k)\ h(n-k), & \text{for } n > 0 \end{cases} \qquad \text{B.2}$$

The autocorrelation function, $\bar{R}(i)$, of the impulse response is given by:

$$\bar{R}(i) = \sum_{n=-\infty}^{\infty} h(n)\ h(n+|i|) = \sum_{n=0}^{\infty} h(n)\ h(n+|i|), \text{ for all } i \qquad \text{B.3}$$

Substituting B.2 into B.3, gives

$$R(i) = \sum_{k=1}^{p} a(k)\ \bar{R}(|i-k|), \quad \text{for} \quad 0 < i < \infty \qquad \text{B.4}$$

and

$$R(0) = G^2 + \sum_{k=1}^{p} a(k)\ \bar{R}(k) \qquad \text{B.5}$$

The autocorrelation coefficients, R(i), for speech samples was shown earlier in Section 5.2.2 to obey the following equations:

$$R(i) = \sum_{k=1}^{p} a(k)\ R(i-k) \qquad 1 \leqslant i \leqslant p \qquad \text{B.6}$$

$$R(0) = G^2 + \sum_{k=1}^{p} a(k)\ R(k) \qquad \text{B.7}$$

Except for the range of delay order, i, the two sets of equations, B.4 and B.6, are of the same form. Therefore, for the range $0 \leqslant i \leqslant p$, the two autocorrelation coefficient sets are related by a constant, c

i.e. $$\bar{R}(i) = c\ R(i) \qquad 0 \leqslant i \leqslant p \qquad \text{B.8}$$

Since the total energy in h(n) must equal that in the speech sample, then:

$$\bar{R}(0) = R(0) \qquad \text{B.9}$$

From B.8 and B.9, the constant c must be unity, and hence:

$$\bar{R}(i) = R(i) \qquad 0 \leqslant i \leqslant p$$

Therefore, the first (p+1) coefficients of the autocorrelation of the impulse repsonse of H(z) are identical to the corresponding coefficients of the signal.

APPENDIX C

THE ITAKURA-SAITO DISTANCE MEASURE [66][67]

If $P_T(\omega)$ and $P_R(\omega)$ are the power spectra of a test and a reference speech frame, then the Itakura-Saito distance, d_{IS}, between them is defined as:

$$d_{IS}(P_T,P_R) = \int_{-\pi}^{+\pi} [\frac{P_T}{P_R} - \log_e (\frac{P_T}{P_R}) - 1] \frac{d\omega}{2\pi} \qquad C.1$$

where

$$P_T = \frac{G^2}{|1 + \sum_{k=1}^{p} a(k)\ z^{-k}|^2} = \frac{G^2}{|A_T(z)|^2} \qquad C.2$$

and

$$P_R = \frac{\bar{G}^2}{|1 + \sum_{k=1}^{p} \bar{a}(k)\ z^{-k}|^2} = \frac{\bar{G}^2}{|A_R(z)|^2} \qquad C.3$$

G and $\bar{G}$ are gain parameters, a(k) and $\bar{a}(k)$, where k = 1,2,...p are the prediction coefficients. Both a(0) and $\bar{a}(0)$ are defined as equal to 1.

If A(z) has all its zeros within the unit circle, then A(1/z) will be analytic on and within the unit circle, since all its zeros are outside the unit circle. Residue calculus can be used to show that:

$$\int_{-\pi}^{+\pi} \log_e \{|A(e^{j\omega})|^2\} \frac{d\omega}{2\pi} = \int_{-\pi}^{+\pi} \log_e \{|A(e^{-j\omega})|^2\} \frac{d\omega}{2\pi} \qquad C.4$$

$$= 2\ \text{Real} \left(\int_{\pi}^{\pi} \log_e \{A(e^{-j\omega})\} \frac{d\omega}{2\pi} \right) \qquad C.5$$

$$= 2\ \text{Real} \left(\oint \log_e \{A\ (1/z)\} \frac{dz}{2\pi j z} \right) \qquad C.6$$

$$= 2\ \text{Real}\ \{\log_e \{A(\infty)]\} = 2\ \text{Real}\ [\log_e(1)]$$

$$= 0 \qquad C.7$$

Thus, equation C.1 reduces to:

$$d_{IS}(P_T,P_R) = \int_{-\pi}^{+\pi} \frac{P_T}{P_R} \frac{d\omega}{2\pi} - \int_{-\pi}^{+\pi} \log_e \left[\frac{G^2/|A_T(z)|^2}{\bar{G}^2/|A_R(z)|^2}\right] \frac{d\omega}{2\pi} - 1 \qquad C.8$$

$$= \frac{1}{\bar{G}^2} \int_{-\pi}^{+\pi} P_T |A_R(z)|^2 \frac{d\omega}{2\pi} + \log_e(\bar{G}^2) - \log_e(G^2) - 1 \qquad C.9$$

The first term on the right hand side of equation C.9 can be simplified as follows:

$$\frac{1}{\bar{G}^2} \int P_T |A_R(z)|^2 \frac{d\omega}{2\pi} = \frac{1}{\bar{G}^2} \int_{-\pi}^{+\pi} \left| \sum_{k=0}^{p} \bar{a}(k)\, e^{-jk\omega} \right|^2 P_T(\omega) \frac{d\omega}{2\pi} \qquad C.10$$

$$= \frac{1}{\bar{G}^2} \int_{-\pi}^{+\pi} \left(\sum_{k=0}^{p} \bar{a}(k)\, e^{-jk\omega} \right)^2 P_T(\omega) \frac{d\omega}{2\pi} \qquad C.11$$

$$= \frac{1}{\bar{G}^2} \int_{-\pi}^{+\pi} \sum_{k=0}^{p} \sum_{\ell=0}^{p} \bar{a}(k)\, \bar{a}(\ell)\, e^{j(k-\ell)}\, P_T(\omega) \frac{d\omega}{2\pi} \qquad C.12$$

Put $P_T(\omega) = \sum R(n)\, e^{-jn\omega}$ and $R(n) = \int_{-\pi}^{+\pi} P_T(\omega)\, e^{jn\omega} \frac{d\omega}{2\pi}$

i.e. the spectral density $P_T(\omega)$ is a non-negative even function of ω, whose Fourier coefficients R(n) define an autocorrelation sequence.

Thus,

$$\frac{1}{\bar{G}^2} \int P_T |A_R(z)|^2 \frac{d\omega}{2\pi} = \frac{1}{\bar{G}^2} \sum_{k=0}^{p} \sum_{\ell=0}^{p} \bar{a}(k)\, \bar{a}(\ell)\, R(k-\ell) \qquad C.13$$

$$= \frac{[\bar{a}][R][\bar{a}]^t}{\bar{G}^2} = \frac{\alpha}{\bar{G}^2} \qquad C.14$$

where [a] is the LPC coefficient vector and [R] the matrix of autocorrelation coefficients.

The numerator, α, is usually expressed in a more computationally efficient form as follows:

$$\alpha = R(0)\,\bar{R}_a(0) + 2\sum_{i=1}^{p} R(i)\,\bar{R}_a(i) \qquad \text{C.15}$$

$$\text{where } \bar{R}_a(i) = \sum_{k=0}^{p-i} \bar{a}(k)\,\bar{a}(k+i) \qquad \text{C.16}$$

Equation C.15 can be obtained from C.14 using the following steps:

$$\sum_{\ell=0}^{p}\sum_{k=0}^{p} \bar{a}(k)\,\bar{a}(\ell)\,R(k-\ell) = \sum_{i=-k}^{p}\sum_{k=0}^{p} \bar{a}(k+i)\,\bar{a}(k)\,R(i) \qquad \text{C.17}$$

$$= \sum_{k=0}^{p} \{\bar{a}(0)\,\bar{a}(k)\,R(k) + \bar{a}(1)\,\bar{a}(k)\,R(k-1) + \ldots + \bar{a}(p)\,\bar{a}(k)\,R(k-p)\} \qquad \text{C.18}$$

$$= \bar{a}(0)\,\bar{a}(0)\,R(0) + \bar{a}(1)\,\bar{a}(0)\,R(1) + \ldots + \bar{a}(p)\,\bar{a}(0)\,R(p)$$

$$+ \bar{a}(0)\,\bar{a}(1)\,R(1) + \bar{a}(1)\,\bar{a}(1)\,R(0) + \ldots + \bar{a}(p)\,\bar{a}(1)\,R(p-1)$$

$$+ \bar{a}(0)\,\bar{a}(2)\,R(2) + \bar{a}(1)\,\bar{a}(2)\,R(1) + \ldots + \bar{a}(p)\,\bar{a}(2)\,R(p-2)$$

$$+ \ldots \quad \ldots \quad \ldots$$

$$+ \bar{a}(0)\,\bar{a}(p)\,R(p) \qquad \text{C.19}$$

By inspection equation C.19 can be expressed as:

$$R(0) \sum_{k=0}^{p} \bar{a}(k)\,\bar{a}(k) + 2\,R(1) \sum_{k=0}^{p-1} \bar{a}(k)\,\bar{a}(k+1)$$

$$+ 2\,R(2) \sum_{k=1}^{p-2} \bar{a}(k)\,\bar{a}(k+2) + \ldots + 2\,R(p)\,\bar{a}(0)\,\bar{a}(p) = \alpha \qquad \text{C.20}$$

Therefore the Itakura-Saito distance measure, d_{IS} can be expressed in a simplified form as:

$$d_{IS}\,(P_T,\ P_R) = \frac{\alpha}{\bar{G}^2} + \log_e\,(\bar{G}^2) - \log_e\,(G^2) - 1 \qquad \text{C.21}$$

The gain-normalized Itakura-Saito distance measure, d_{GN}, is defined as:

$$d_{GN}\,(P_T,\ P_R) = \frac{\alpha}{G^2} - 1 \qquad \text{C.22}$$

The parameter, G^2, in equation C.22 can be expressed in a matrix form as follows:

$$G^2 = [a][R][a]^t \qquad \text{C.23}$$

This relationship can be shown in the following steps:

$$[a][R][a]^t = \sum_{i=0}^{p} \sum_{k=0}^{p} a(i)\,a(k)\,R(i-k) \qquad \text{C.24}$$

$$= \sum_{i=0}^{p} \{ \sum_{k=1}^{p} a(i)\,a(k)\,R(i-k) + a(i)\,a(0)\,R(i) \} \qquad \text{C.25}$$

$$= \sum_{i=0}^{p} \sum_{k=1}^{p} a(i)\, a(k)\, R(i-k) + \sum_{i=0}^{p} a(i)\, R(i) \qquad \text{C.26}$$

$$= \sum_{i=1}^{p} \sum_{k=1}^{p} a(i)\, a(k)\, R(i-k) + \sum_{k=1}^{p} a(0)\, a(k)\, R(k) + \sum_{i=0}^{p} a(i) R(i) \qquad \text{C.27}$$

The relationship between predictor coefficients and autocorrelation coefficients is given by:

$$\sum_{k=1}^{p} a(k)\, R(i-k) = -\, R(i), \quad 1 \leqslant i \leqslant p \qquad \text{C.28}$$

Substituting C.28 into the first term on the right hand side of equation C.27, gives:

$$- \sum_{i=1}^{p} a(i)\, R(i) + \sum_{k=1}^{p} a(k)\, R(k) + \sum_{i=0}^{p} a(i)\, R(i) \qquad \text{C.29}$$

$$= \sum_{i=0}^{p} a(i)\, R(i) = R(0) + \sum_{i=1}^{p} a(i)\, R(i) = G^2 \qquad \text{C.30}$$

Thus

$$d_{GN} = \frac{[\tilde{a}][R][\tilde{a}]}{[a][R][a]^t} - 1 \qquad \text{C.31}$$

APPENDIX D

THE RELATIONSHIP BETWEEN LPC COEFFICIENTS AND CEPSTRAL COEFFICIENTS

From the LPC filter defined by the transfer function,

$$H(z) = \frac{G}{1 + \sum_{k=1}^{p} a(k) z^{-k}} \qquad D.1$$

where a(k), k = 1,2, ...p are the prediction coefficients and p the order of prediction. G is the gain of the filter.

The Cepstrum, C(z) is defined in the z domain by taking the log of the transfer function, H(z), i.e.

$$\log_e H(z) = C(z) = \sum_{n=1}^{\infty} C(n)\, z^{-n} \qquad D.2$$

where C(n) are the cepstral coefficients.

Substituting equation D.1 into D.2 and taking the derivatives of both sides with respect to z^{-1}, gives

$$\frac{d}{dz^{-1}} \{\log_e G - \log_e [1 + \sum_{k=1}^{p} a(k)\, z^{-k}]\} = \frac{d}{dz^{-1}} \sum_{n=1}^{\infty} C(n) z^{-n} \qquad D.3$$

D.3 can be simplified to give

$$- [\sum_{k=1}^{p} k\, a(k)\, z^{-k+1}]/[1 + \sum_{k=1}^{p} a(k)\, z^{-k}] = \sum_{n=1}^{\infty} n\, C(n)\, z^{-n+1} \qquad D.4$$

or

$$-\sum_{k=1}^{p} k\, a(k)\, z^{-k+1} = \left(1 + \sum_{k=1}^{p} a(k)\, z^{-k}\right) \sum_{n=1}^{\infty} n\, C(n) z^{-n+1} \qquad D.5$$

Equating the constant term and the various powers of z^{-1}, on the left and the right hand sides of equation D.5, gives the relationship between C(n) and a(n), i.e.

$$C(1) = -\, a(1) \qquad D.6$$

$$C(n) = \sum_{k=1}^{n-1} (1 - k/n)\, a(k)\, C(n-k) + a(n) \qquad 1 \leqslant n \leqslant p \qquad D.7$$

Thus, the cepstral coefficient at unit delay is identical to the first LPC coefficient.