SPEAKER INDEPENDENT REAL-TIME SPEECH RECOGNITION SYSTEM

by

ABID M. JINDANI, B.S.E.E., B.S.C.S.

A THESIS

IN

ELECTRICAL ENGINEERING

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

Approved

August, 1998

ACKNOWLEDGEMENTS

During the course of my work, there have been many people who were responsible for the successful completion of this thesis. Without the help and support of these people, this long and rocky road would never have been traversed. I would like to take this opportunity to acknowledge some of them. I wish to express my sincere gratitude and appreciation to Dr. Micheal Parten, my graduate advisor, for his guidance, valuable suggestions, encouragement and moral support. I am also grateful to Dr. Sunanda Mitra and Dr. Donald Gustafson for serving as members of my thesis committee. I would also like to thank the Department of Electrical Engineering for their financial support throughout the course of my graduate studies. I also appreciate the moral support of all my friends and colleagues who have not been named here individually. I am most grateful to my parents and my family for their unending support, encouragement, guidance, and most of all patience through my whole life. It is to my family to whom I wish to dedicate this thesis.

TABLE OF CONTENTS

ABSTRACT

This thesis attempts to develop a real-time speaker-independent Automatic Speech Recognition (ASR) system. The system recognizes isolated utterances from a limited vocabulary, and is small and cost-efficient to be incorporated into a consumer appliance. The recognition is based on zero crossings and energy content measurement on the speech waveforms. The algorithm is based on segmenting the speech waveform into ten equally spaced intervals and performing a match with the patterns in a reference template. The system was implemented on an IBM Personal Computer and achieved an error rate of 0% on a vocabulary of four words from an initial ten-word database of 16 speakers (8 male and 8 female). The system recognized unknown utterances in less than 0.3 seconds.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Technological breakthroughs have been occurring in society at an unprecedented rate. It is nearly impossible for a person to go by a day without interacting with some sort of machine or appliance. From videocassette recorders (VCR) to television sets to microwave ovens, humans are totally submerged in technology. However, the interaction between machines and humans is still very primitive. Most people don't like to press an untold number of buttons to accomplish a task. As a result the appliances or other daily consumer equipment is not utilized to its fullest extent. For example, many people still have VCRs that flash the infamous 12:00 as their time. So it is only natural that if people could "talk" to these machines their lives would be even more comfortable. Most consumer appliances today have some sort of electronic or computer control, this feature paves the way to realize the goal of "talking" to these appliances.

Human-Computer Interaction (HCI) is one of the areas of research that has gained considerable attention in the recent years. With the advent of such technologies as the Graphical User Interface (GUI) and others, the goal of bridging the gap between the human and computer is just over the horizon. Historically the most common form of input has been the traditional keyboard. Recently the computer has seen the mouse, touch-pads, remote control, digitizing pad and other similar inputs. Computer processing power has been growing almost exponentially with the arrival of smaller, faster and

cheaper microprocessors. As a result most of the time the computer is waiting for the person to input data, rather than the human waiting for the machine to respond.

All of the technologies mentioned above have a common component, specifically the hands. The most natural and efficient form of exchanging information among humans is speech. So it is only logical that the next technological development be the natural language speech recognition for HCI.

1.2 Speech Recognition

The goal of speech recognition is for a machine to be able to "hear," "understand," and "act upon" spoken information. This goal for now remains in the distant future. However, new advances in the field of speech recognition have shown considerable progress towards that goal.

Modern speech recognition systems typically can be classified in three ways:

1. Speaker dependent or speaker independent.
2. Isolated words or continuous speech.
3. Small or large vocabulary.

The most complex of these systems is the continuous speech, speaker-independent system with a large vocabulary and the simplest is the isolated word, speaker-dependent system with a small vocabulary. In each case the speech-processing task is computationally intensive. Advances in computer architecture and very large scale integration (VLSI) design have led to very powerful computer systems that are adept at

handling such a task. However it will be quite sometime before natural language speech recognition becomes a reality.

The earliest speech recognition systems were first attempted in the early 1950s. In 1952, at Bell Laboratories, Davis, Biddulph and Balashek developed an isolated digit recognition system for a single speaker [1]. The system worked on measuring spectral resonances in the vowel region of each digit.

In the 1960s several fundamental ideas in speech recognition emerged and were published with the Japanese leading the hardware effort. One of the early efforts was the hardware vowel recognizer developed by Suzuki and Nakata [2]. This system involved a filter bank spectrum analyzer whose output from each of the channels was fed to a vowel-decision circuit, and a majority decision logic scheme was used to choose the spoken vowel. Another system developed by the Japanese was a hardware speech segmenter along with zero-crossing analysis on different sections of the speech to recognize the phoneme [3]. Perhaps the most notable attempt came from another Japanese group led by Nagata at NEC Laboratories in 1963 [4]. Nagata produced a hardware spoken digit recognizer that set the stage for automatic speech recognition systems to come.

The 1960s also saw the development of dynamic time warping algorithms by a number of researchers in the United States and the former Soviet Union. In the United States, Martin and his colleagues at the RCA Laboratories researched the problem of nonuniformity of time scales in speech events [5]. The Russian effort led by Vintsyuk developed the dynamic programming methods for time aligning a pair of speech utterance [6]. Vintsyuk's work remained largely unknown and did not come to light until

the 1980s, by which time more formal methods were proposed and implemented. One of the important projects in speech recognition in the 1960s was the pioneering research of Reddy in the field of continuous speech recognition by Dynamic tracking of phonemes [7].

In the 1970s a number of advances in speech recognition were achieved. The most notables are isolated word recognition, pattern-recognition, dynamic programming methods, and linear predictive coding (LPC) [8,9,10]. AT&T Bell Labs made a number of advancements in the area of speaker independent speech recognition systems. The research was conducted under the supervision of Rabiner where his team used clustering techniques for speaker-independent speech recognition [11].

The 1980s was characterized by a shift in analysis from template-based approaches to statistical modeling methods, especially the Hidden Markov Model (HMM) approach. The technique became so popular that virtually every speech recognition laboratory in the world started applying the technique to speech recognition systems. Another popular technique based on statistical methods that also gained popularity was Neural Networks. Several researchers were able to apply the principles of neural networks to speech recognition system [12,13].

The 1980s also saw significant development in the area of large vocabulary continuous speech recognition through the Defense Advanced Research Projects Agency (DARPA) community, which sponsored a number of large research programs at various academic and research institutions. The DARPA project also developed a number of tools

and resource management database to help researchers in the area of speech recognition [20].

Speech recognition although still a developing field has become a viable technology and many commercial systems are currently available for the consumer. Dictation systems for the personal computer include software from numerous vendors including Dragon System, IBM and Kurzweil.

Dragon System's NaturallySpeaking Deluxe is a very advanced speaker dependent continuous speech recognition system [15]. The software runs on an IBM personal computer and boasts a vocabulary of 60000 words, a recognition rate of 160 words per minute and accuracy of 95% or higher. The shortfalls of the software are very large system requirements that include 96 MB of random access memory (RAM), a Pentium 133MHz processor and 55 MB of hard disk space. Although the software is speaker dependant it has the ability to store as many users as the system has storage space but each additional user requires 15MB of hard disk space. The software only allows one active user at a time and requires the system be notified if a new user is active. Dragon Systems has another product called DragonDictate, which is an isolated word speech recognition software. The software requirements are comparable to NaturallySpeaking but has the added feature of speaker adaptation. The speaker adaptation model lies somewhere between speaker independent and speaker dependent. In speaker adaptation, the system adapts itself to a new user with little training. IBM and Kurzweil both have software very similar to Dragon Systems' and only vary slightly in vocabulary size and system requirements.

The systems that have been discussed above all have shortfalls of one kind or another. The major shortcoming of these systems is that they all require training of some kind on the part of the user. The training time maybe as short as couple of minutes to as long as a few hours. In many applications the user requires that a system be available immediately with little or no interaction on the part of the user to setup the system.

The training aspect of the system limits the number of users that can use the system at any time to one. This is a problem that severely restricts the number of applications in which speech recognition can be deployed. Most consumer level appliances or computing machinery is frequently used by more than one person. Training each and every individual using the system can become a daunting and inefficient task.

Another problem that exists with these systems is that they are all PC based systems. This aspect of these systems also restricts them to only one type of environment. Everyday electronic or electrical appliances do not have the computing power or storage to warrant such a system.

Although a few hardware-based systems do exist for commercial applications, they too have problems of their own. Most of the hardware-based systems are used in conjunction with a host PC. A common Digital Signal Processor (DSP) based system is the Dialogic Antares 2000 series of DSP ISA bus based speech recognition boards. These boards contain four Texas Instruments TMS320C31 32-bit floating-point DSPs running at 50 MHz and also include 2MB of dedicated memory. The host operating systems supported are WindowsNT and multiple flavors of UNIX. The speech recognition system supports both speaker independent and speaker dependant mode, and can also recognize

both discrete and continuous speech. The system is intended to be used in computer telephony applications as a telephone attendant for a company. The prohibitive cost and computing requirements of the system do not make it an attractive option for consumer level applications.

Another system that is available for hardware based speech recognition systems is SmartSpeak by Advanced Recognition Technologies. This is actually a software system intended to be developed on microcontrollers and low cost DSPs. The software also requires a host PC for speech acquisition and preprocessing. Although a cost-effective system, it only supports speaker dependent operation.

1.3 Statement of the Problem

Although many systems exist for speech recognition, none of them address the needs for consumer level applications. In order for a system to be incorporated in the everyday needs of a consumer, the system must be speaker independent, fast, low-cost, require no training and small enough to be fit inside a consumer appliance. Such a system will move speech recognition from the domain of the academic or industrial application to that of a common home user.

The above system can be implemented using current technology once a certain number of compromises are made. For example, let's say a speech recognition system is to be developed so that it can be incorporated into a home microwave oven. One can immediately see that there is no need to have a 60,000 word vocabulary for such a system, a dozen words including the digits are sufficient for its operation. The system

could be further simplified if one does not allow the user to change the number of words in the vocabulary. The Second aspect of the system is that it does not have to accept continuous speech. For example, a common command may be "Cook.... Two.... Minutes.... Start." Recognition accuracy of 90% or above is acceptable since most people would not mind repeating a command to the system one out of ten times or less.

The system can be implemented using one of the common DSP processors with a small amount of dedicated memory and an analog-to-digital converter to accept the input speech. The system would be fast, small and cost-efficient to be incorporated into a wide variety of consumer electronics. The aim of this thesis is therefore to develop a speaker independent, isolated word, limited vocabulary speech recognition system that is small enough to fit in a small household appliance and that can be operated in real-time.

The second chapter provides information on basic speech processing techniques. Chapter 3 describes the entire system that has been developed. Chapter 4 explains all the algorithms used in the system and provides a mathematical basis for them. The results of system evaluation are also reported in this chapter. The final chapter summarizes the results of the system and also suggests further improvements and future research directions.

CHAPTER 2

SPEECH PROCESSING TECHNIQUES

Many different techniques are available for processing of speech signals, most of them can be classified under two broad categories: time-domain and frequency-domain methods. Time-domain methods involve the waveform of the speech signal directly. Frequency-domain methods look at how the signal behaves in the frequency spectrum of the signal.

2.1 Digital Representation of Speech Signal

It is important to understand how the speech signal is represented in digital form. The conversion of the analog speech waveform into digital form is usually called speech coding. Figure 2.1 shows a block diagram of how an analog signal is represented in digital form.

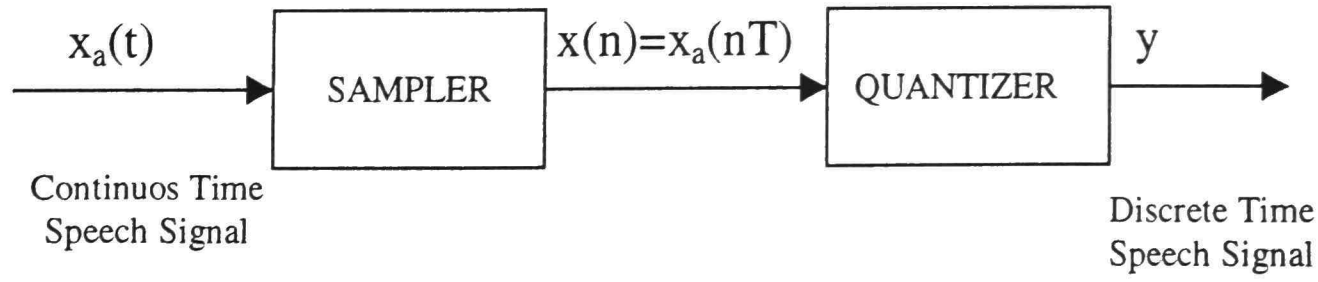

Figure 2.1 General block diagram of a digital waveform representation.

The very well known sampling theorem (Nyquist Theorem) [26] states that a bandlimited signal can be represented by samples taken periodically in time – provided that the samples are taken at a high enough rate. The samples are usually taken at least twice the Nyquist frequency, which is the highest frequency component in the signal under consideration. The frequency bandwidth for typical voice signal ranges from about 100Hz to 3500KHz. Assuming a Nyquist frequency of 4Khz a sampling rate of 8Khz is typical for speech processing.

The second step in acquiring the signal is the quantization of the samples taken periodically in time. In typical applications of speech processing the quantization levels and ranges are generally distributed uniformly. In uniform quantizers there are only two parameters: the number of levels and quantization step size, Δ. The number of levels is generally chosen to be of the form 2^B so as to make the most efficient use of B-bit binary code words. Together, Δ and B must be chosen so as to cover the range of input samples.

2.2 Algorithms in Speech Recognition

Many different approaches exist in recognizing human speech. Algorithms such as template matching come under the pattern recognition approach. Algorithms that depend on knowledge sources including the stochasticity of speech signals and neural networks are based on the artificial intelligence approach. Stochastic modeling using Hidden Markov Models (HMM) has become especially popular in modern speech recognition systems.

2.2.1 Zero-Crossing and Energy-Based Speech Recognition

Speech recognition using zero crossing and energy content has been demonstrated by a number of researchers. Rabiner and Sambur developed a speaker-independent digit-recognition system using energy and zero-crossing measures [16]. The system works by first segmenting the unknown word into three regions and then making categorical judgments as to which of six broad acoustic classes each segment falls into. It was observed that the zero-crossing rate at the beginning of words starting with strong fricatives is higher than for words starting with weak consonants. The system was reported to have an error rate of 2.7 percent. No information regarding the processing platform or recognition speed was provided. However, the simplicity of the algorithm suggests that the system could be implemented with relative ease on a DSP. Although the system only recognized digits, it is not too difficult to extend the system to other words of the vocabulary.

2.2.2 Template-Based Speech Recognition

Template-based speech recognition systems include a database of speech patterns that define the vocabulary. The database is generated during the training phase of the system. In the recognition mode, an input speech sample is compared to the stored templates in the database and a decision is made based upon a best match. Since the rate at which words are spoken vary greatly, it is important that some form of alignment be made between the incoming speech and the stored templates. The alignment can be thought of as a mapping of input speech to that of stored frames and the problem reduces to a minimization problem. One algorithm that greatly rectifies this situation is the

Dynamic Time Warping (DTW) algorithm. DTW algorithms have been incorporated in a number of speech recognition systems. A variety of DTW algorithms with varying constraints have been proposed and incorporated for use in speech recognition. One algorithm called the Constrained Endpoints, 2-to-1 Range of Slopes (CE2-1) proposed by Itakura [17] have the constraint that the starting and ending points are assumed to be in perfect registration, and the dynamic path is assumed to be in a fixed parallelogram whose slopes are 2 and ½ at the edges. Another algorithm called the Unconstrained Endpoints, 2-to-1 Range of Durations (UE2-1) has the condition that starting and ending points are unconstrained but must not go beyond a few speech frames. The dynamic path is again assumed to lie within a fixed parallelogram whose slopes are 2 and ½ at the edges. A third algorithm Unconstrained Endpoints, Local Minimum (UELM) has again the constraints on endpoints relaxed, and the allowable region of dynamic paths is constrained to follow the locally optimum path to within a few frames. A diagram illustrating all three algorithms is shown in Figure 2.2. The system developed by Itakura was speaker-dependent isolated-word speech recognition with a 200-word vocabulary. The system had a recognition rate of 97%.

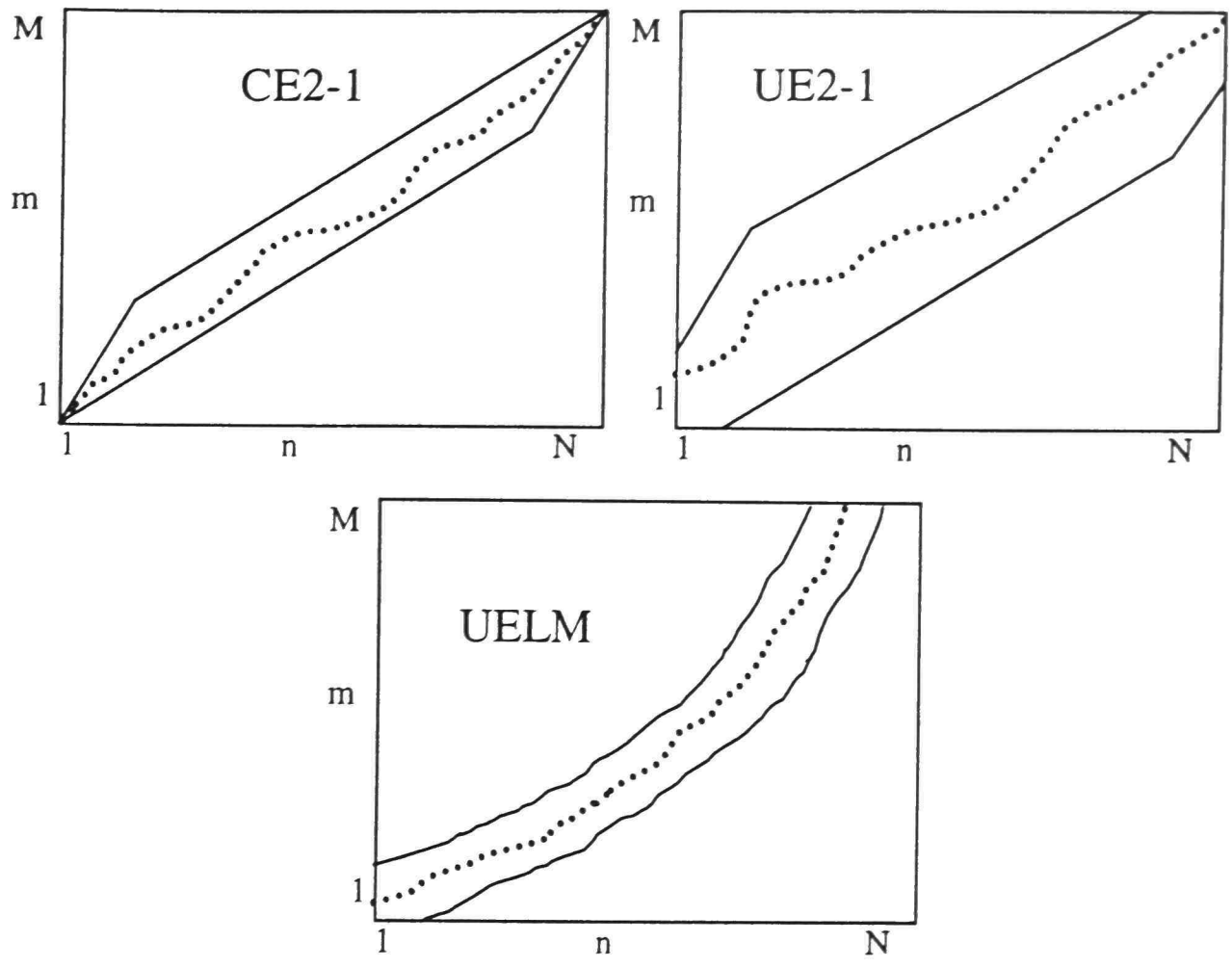

Figure 2.2 Typical warping paths for the three dynamic time-warping techniques.

2.2.3 Stochastic Speech Recognition

Systems based on stochastic models such as HMM deal with incomplete information or uncertainty. The HMM uses states that model generic speech sounds and transitions between states with associated transition probabilities to model the temporal behavior of speech. The system assumes that a hidden Markov process produced the speech. Although HMMs can provide substantially better recognition rates the system is computationally very expensive and template based system can provide much faster results. Rabiner et al. [18] developed a Vector Quantization based HMM speech recognition system that was speaker-independent, isolated-word with a limited vocabulary. The system was implemented using SUN workstations. Although the system

could recognize words from any speaker, the system had several drawbacks, which was the immense amount of time it took to train the system to a given set of words. A 10-word vocabulary took more than 15 hours to train the system and recognition of words also took longer than is acceptable in a real-time environment. The hardware and software requirements of the system also prohibit it as a viable consumer system.

2.3 Speech Recognition System

After examining a number of these approaches the zero crossing and energy content-based approach seems to be the most viable solution to the problem stated. This approach requires the least number of computations, relatively few memory storage elements and near real-time performance. Advances in computer architecture and VLSI have led to the development of the digital signal processors (DSP). In recent years DSPs have seen a considerable growth in many areas of signal processing from multimedia applications and digital data transfer to speech recognition. A DSP chip is 10 to 50 times more powerful than micro-processing computer chips in handling math intensive tasks, such as those involved in compressing and processing voice and video signals. It enables data to be processed in real time, which would otherwise not be achievable. The use of a DSP would make the system very cost-efficient and fast enough to operate in real time. This system would be small in size, since a DSP does not require several peripherals and numeric processors to generate control signals and perform calculations. Secondly, the enormous computing power of a DSP would help in the real time aspect of the speech recognition system.

As stated earlier the goal of this thesis is to develop a stand-alone speaker-independent, isolated speech recognition system with a small vocabulary that is real-time, small and cost-efficient. The Texas Instruments TMS32C30 series of DSP can fulfill this requirement. An attempt is made to develop such a system using the techniques of template matching algorithm with zero crossing and energy content measures. In this approach the test utterance is divided into a number of sections and the zero-crossing rate and energy content is measured for each of the sections. The results are then compared to the stored reference patterns and a decision is made. The decision criteria achieved through two different approaches. The first is a bayesian classification criterion and the other is a modified form of the Euclidean distance measure. The performance of both the algorithms is provided in Chapter 5.

CHAPTER 3

FRAMEWORK OF THE RECOGNITION SYSTEM

This chapter describes the development of the entire speech recognition system that has been proposed. All the algorithms that have been developed and implemented are discussed briefly. Chapter 4 provides a more detailed discussion of the actual algorithm along with the pseudo code.

3.1 Acoustic Phonetics

The elements of most languages, including English, can be described by a set of distinctive sounds, or phonemes. The phonemes can be further classified into four broad categories [21]:

1. Vowels,
2. Diphthongs,
3. Semivowels,
4. Consonants.

The vowels are further classified into front (/i/, /I/, /e/, /ɛ/, and /æ/), middle (/ə/, /Λ/), and back vowels (/u/, /U/, /OW/, and /o/). It is also convenient to subdivide the consonants into the categories noise-like (fricatives, plosives) and vowel-like (nasals, glides).

A recognition system must use a set of robust measurements to classify the words in a manner suitable for correct identification. The requirements for a recognition parameter to be selected as being a robust measurement are:

i. The parameter can be simply and unambiguously measured.

ii. The parameter can be used to grossly characterize a large proportion of speech sounds.

iii. The parameter can be conveniently interpreted in a speaker-independent manner.

The zero-crossing rate and energy content measure fit the above criterion. The general acoustic properties of the words can be effectively characterized by these measurements. For example, noise-like sounds have a relatively high zero-crossing rate and relatively low energy.

3.2 Speech Recognition Databases

In the investigation of speech recognition it is important that some sort of standard database for system testing and analysis be available. Although many different standard databases, including the Defense Advance Research Projects Agency (DARPA) Resources Management Database (DRMD) [20] are available to researchers, the Texas Instruments 46-word Speaker-Dependent Isolated Word Corpus [14] was chosen for this system. Texas Instruments in collaboration with the National Institute of Standards and Technology (NIST) designed this database. The corpus contains 16 speakers (8 males and 8 females) and includes 46 words per speaker, which include the ten digits, 26 letters of the alphabet and 10 computer-related words. Only the 10 computer-related words from both male and female speakers were chosen for this system. The 10 computer-related words are "enter", "erase", "go", "help", "no", "rubout", "repeat", "stop", "start", and "yes".

Table 3.1 shows the sound class characteristics of the ten words used in this system. The phonetic transcription of the words in the table uses the United States Advanced Research Projects Agency's (ARPA) all uppercase *ARPAbet* for the phonemes.

Table 3.1 – Sound classes characteristic of the words

Word	Sequence of Sound Classes
ENTER (/AE/ /N/ /T/ /ER/)	FV→VLC→UVNLC→MV
ERASE (/AE/ /R/ /EY/ /S/)	FV→VLC→FV→UVNLC
GO (/G/ /OW/)	VNLC→MV
HELP (/HH/ /EH/ /L/ /P/)	VLC→FV→VLC→UVNLC
NO (/N/ /OW/)	VLC→MV
RUBOUT (/R/ /UW/ /B/ /AW/ /T/)	VLC→BV→VNLC→D→UVNLC
REPEAT (/R/ /IY/ /P/ /IY/ /T/)	VLC→FV→UVNLC→FV→UVNLC
STOP (/S/ /T/ /AA/ /P/)	UVNLC→UVNLC→MV→UVNLC
START (/S/ /T/ /AA/ /R/ /T/)	UVNLC→UVNLC→MV→VLC→UVNLC
YES (/Y/ /EH/ /S/)	VLC→FV→UVNLC

VNLC = Voiced, noise-like consonant.
UVNLC = Unvoiced, noise-like consonant.
VLC = Vowel-like consonant.
FV = Front vowel.
MV = Middle vowel.
BV = Back vowel.
D = Diphthong

Figures 3.1 and 3.2 show the acoustic waveform, energy and zero crossing for the word 'ENTER.' The plot for the rest of the words is given in Appendix A.

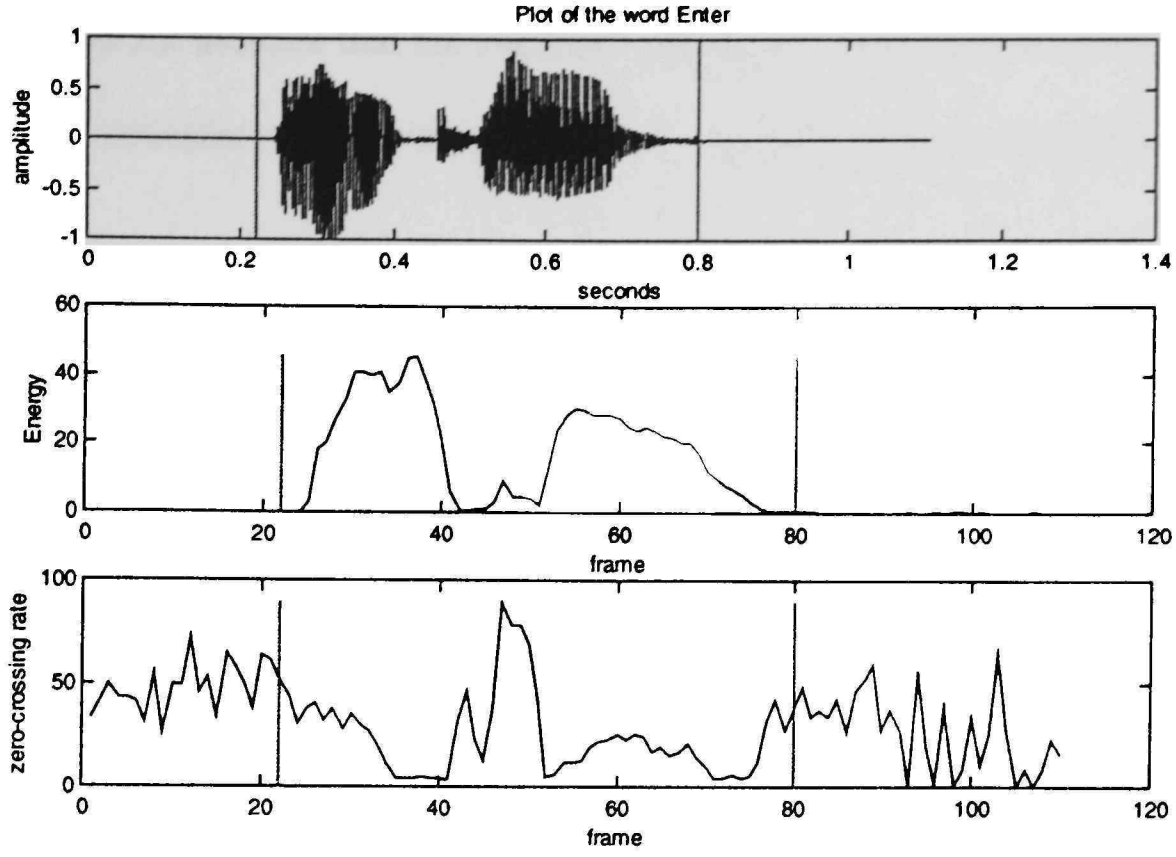

Figure 3.1. Short time energy and zero crossing data for the word "Enter" by a female speaker.

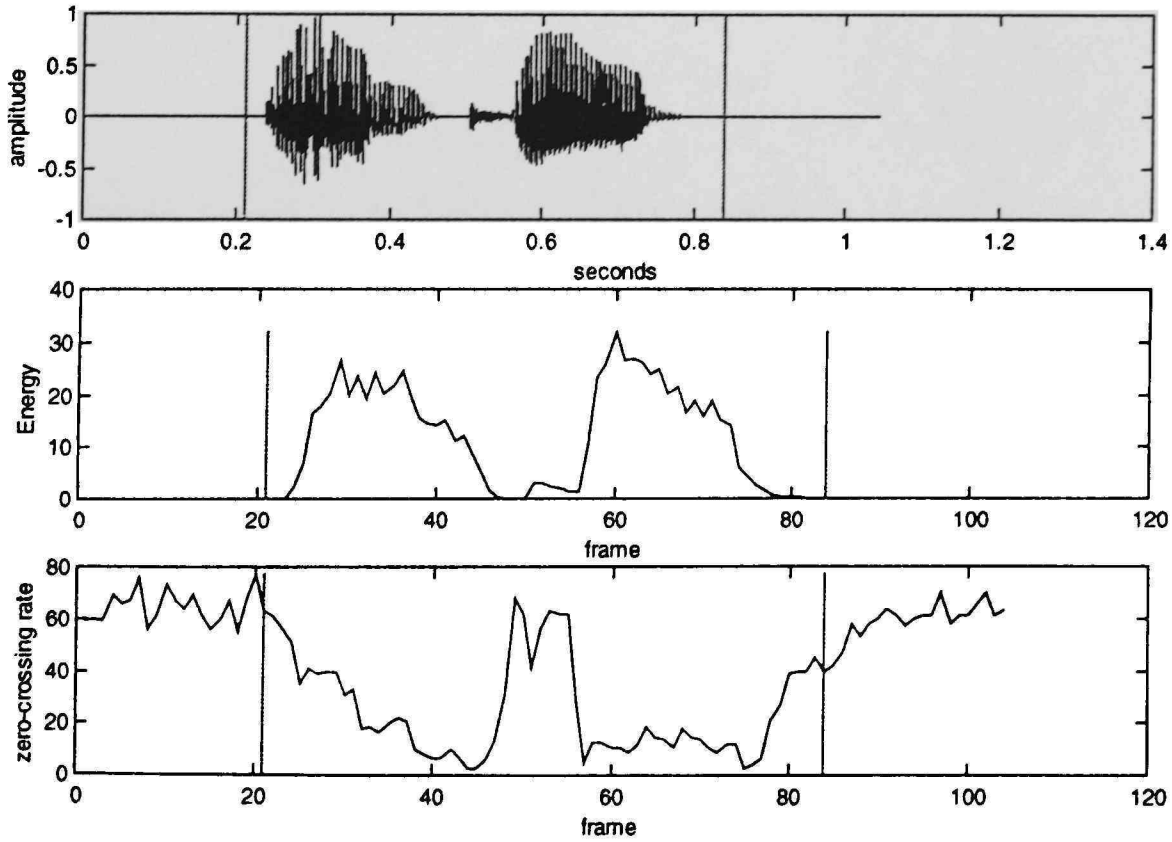

Figure 3.2. Short time energy and zero crossing data for the word "Enter" by a male speaker.

Figures 3.1 and 3.2 indicate that the variation among male and female speakers is small enough that a representative waveform can be obtained for proper recognition of the words.

3.3 Speech Recognition System

Figure 3.3 shows a block diagram of a speech recognition system that has been implemented. The system shown here consists of two separate modes of operation. In the first mode a set of input utterances is used to create a reference template for each of the words in the vocabulary. Once the reference template is created it is not modified any further. In the second mode, an unknown input speech is compared against the reference template and a decision is made.

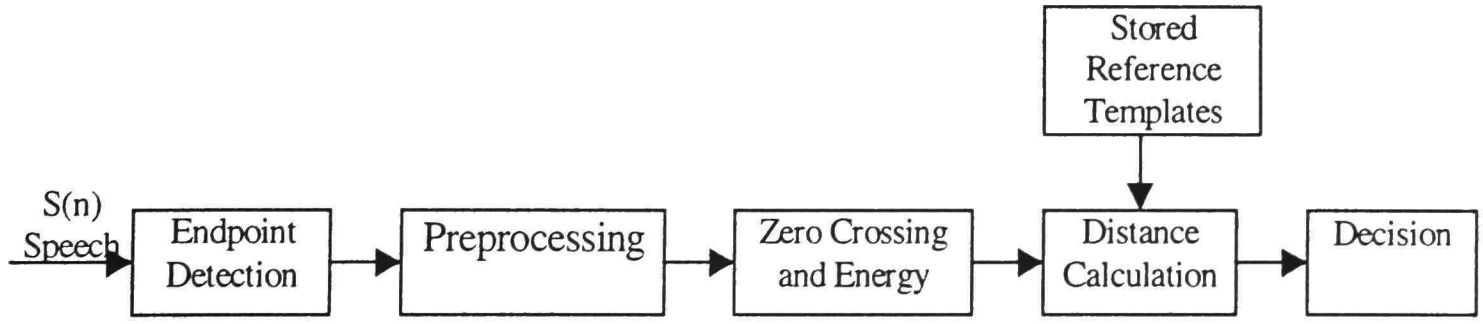

Figure 3.3 Block Diagram of the speech recognition system.

The operating modes of the system are independent and are implemented separately. The reference template creation is done entirely on a computer, whereas the recognition mode takes place in a DSP based system.

3.4 Endpoint Detection

It is important in a speech recognition system that the beginning and the ending of an utterance are accurately known. This not only reduces the amount of data that needs to be processed but also discriminates the utterance against background noise. The problem of detecting endpoints would seem to be a relatively trivial, but, in fact it is has been found to be very difficult in practice, except in cases of very high signal to noise ratios. Some of the problems that plague endpoint detection are weak fricatives (/f/, /T/, /h/) or voiced fricatives that become unvoiced at the end ("has"), weak plosives at either end (/p/, /t/, /k/), nasals at the end ("gone"), and trailing vowels at the end ("zoo").

In order to solve the problem of endpoint detection, a number of signal processing techniques must first be established. The underlying assumption in most speech processing schemes is that the properties of the speech signal change relatively slowly with time. This assumption leads to a variety of "short-time" processing methods in which short segments of the speech signals are isolated and processed. These short segments, usually called analysis frames, often overlap one another.

3.4.1 Energy Content Measure

A typical quantity that is calculated is the short-time energy. A simple definition of the short-time energy is

$$E_n = \sum_{m=n-N+1}^{n} x^2(m). \qquad (3.1)$$

The major significance of E_n is that it provides a basis for distinguishing voiced speech segments from unvoiced speech segments.

Figure 3.4 shows a plot of a speech sample and the corresponding short-time energy. As seen in the figure, the values of E_n for the unvoiced segments are significantly smaller than for voiced segments. The energy function can also be used to locate the approximate time at which voiced speech becomes unvoiced, and vice versa. For very high quality speech (high signal-to-noise ratio), the energy can be used to distinguish speech from silence.

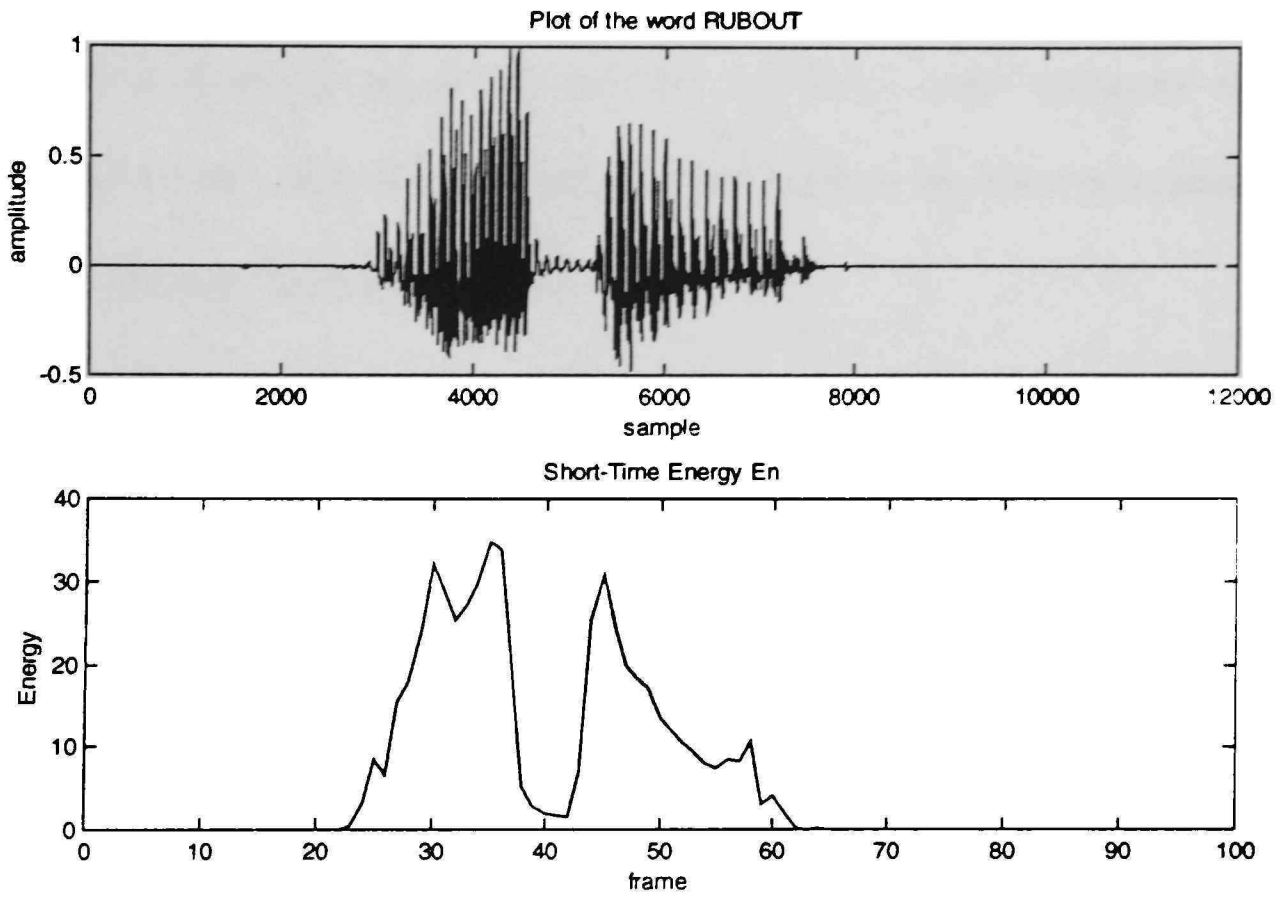

Figure 3.4 Acoustic waveform and short-time energy function for the word RUBOUT.

3.4.2 Zero-Crossing Rate

Another useful measure in signal processing is the zero-crossing rate. A zero crossing is said to occur if successive samples have different algebraic signs. The rate at which zero-crossings occur is a simple measure of the frequency content of a signal. For example, a sinusoidal signal of frequency F_0, sampled at a rate F_S, has F_S/F_0 samples per

cycle of the sine wave. Each cycle has two zero crossings so that the long-time average rate of zero-crossings is

$$Z = 2\frac{F_0}{F_S}, \text{crossings/sample.} \tag{3.2}$$

Thus, the average zero-crossing rate gives a reasonable way to estimate the frequency of a sine wave.

Since speech signals are broadband signals, the interpretation of average zero-crossing rate is therefore much less precise. However, rough estimates of spectral properties can be obtained using a representation based on the short-time average zero-crossing rate. An appropriate definition is

$$Z_n = \sum_{m=-\infty}^{\infty} \left|\text{sgn}[x(m)] - \text{sgn}[x(m-1)]\right| w(n-m) \tag{3.3}$$

where

$$\begin{aligned} \text{sgn}[x(n)] &= 1 \quad x(n) \geq 0 \\ &= -1 \quad x(n) < 0 \end{aligned} \tag{3.4}$$

and

$$\begin{aligned} w(n) &= \frac{1}{2N} \quad 0 \leq n \leq N-1 \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{3.5}$$

Equation (3.3) makes the computation of Z_n appear more complex than it really is. All that is required is to check samples in pairs to determine where zero-crossings occur and compute the average over N consecutive samples.

Since high frequencies imply high zero crossing rates, and low frequencies imply low zero crossing rates, there is a strong correlation between zero crossing rate and

energy distribution with frequency. A reasonable generalization is that if the zero-crossing rate is high, the speech signal is unvoiced, while if the zero crossing rate is low the speech signal is voiced.

Figure 3.5 shows the acoustic waveform, the short-time energy and average zero crossing rate for the word REPEAT. As can be seen from the figure, the voiced and unvoiced regions are quite prominent.

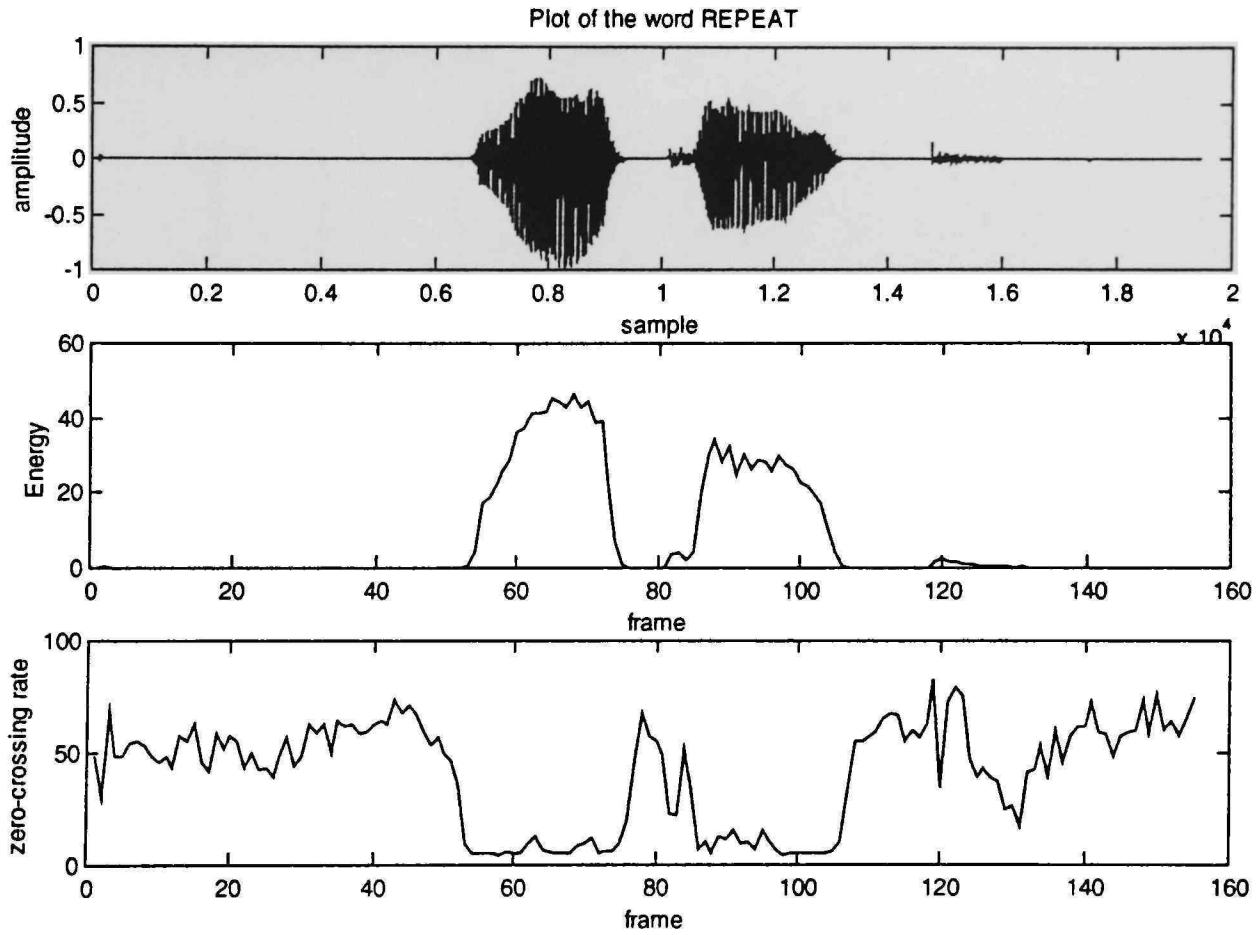

Figure 3.5 Acoustic waveform, short-time energy function and zero-crossing rate.

There are a number of practical considerations in implementing a representation based on the short-time average zero-crossing rate. The zero-crossing rate is strongly affected by DC offset in the analog-to-digital converter, 60 Hz hum in the signal, and any noise that may be present in the digitizing system. Therefore, extreme care must be taken

in the analog processing prior to sampling to minimize these effects. For example, it is often preferable to use a bandpass filter, rather than a low pass filter, as the anti-aliasing filter to eliminate DC and 60Hz components in the signal.

3.4.3 Endpoint Detection Algorithm

Rabiner and Sambur [19] have proposed an endpoint detection algorithm using zero crossing and energy measures. Figure 3.6 shows the short-term zero crossing and energy measures plotted for the word "four."

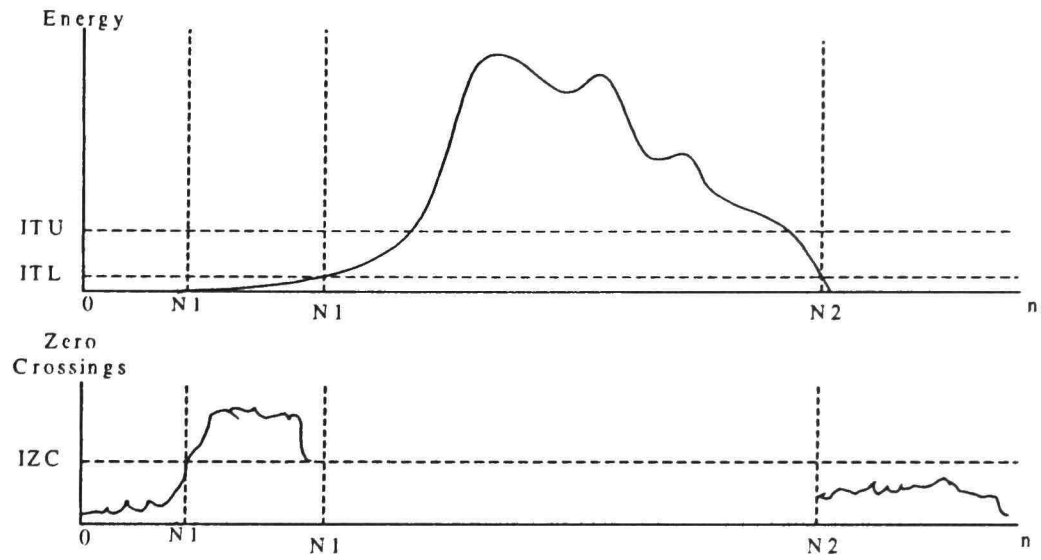

Figure 3.6 Short time energy and zero crossings data for the word "four."

The curves were obtained by calculating the energy content and zero crossing rate every 10 msec on frames of length 10 msec. It was assumed that the first 10 frames are background noise. They are used to find the mean and variance of each of the features. These measurements are then used to set the "upper" and "lower" thresholds, τ_u and τ_l, as shown in Figure 3.6. The energy curve is then searched to find the first crossing of the upper threshold τ_u moving toward the middle of the segment from each end. The algorithm then goes back to the nearest crossing of τ_l in each case. This process yields

tentative endpoints N_1 and N_2 in Figure 3.6. The double-thresholding procedure prevents the false indication of endpoints by dips in the energy curve. Next the algorithm moves towards the ends from N_1 and N_2 for no more than 25 frames, examining the zero crossing rate to find three occurrences of counts above the threshold τ_{zc}. If these are not found, the endpoint remains at the original estimate. If three occurrences are found, then the endpoint estimate is moved backward (or forward) to the time of the first threshold crossing. This is the case for N_1 (moved to N_1') in the figure.

3.5 Reference Template Creation

There are a number of problems that need to be adressed before a reference template for each of the words in the vocabulary of the system can be constructed. The first problem is that the utterances of a given word are not temporally aligned. A popular technique to solve the problem is the Dynamic Time Warping (DTW) algorithm. This algorithm was addressed in the last chapter. DTW algorithms work extremely well but can become computationally expensive and therefore unsuitable in a realtime environment. A simpler approach is to take the test utterance and break it down into an equal number of sections and calculate the features within the sections. Since the zero crossing rate and the energy content are calculated in frames, the average of the parameters is calculated within the sections. Figure 3.7 and Figure 3.8 show how this can be accomplished.

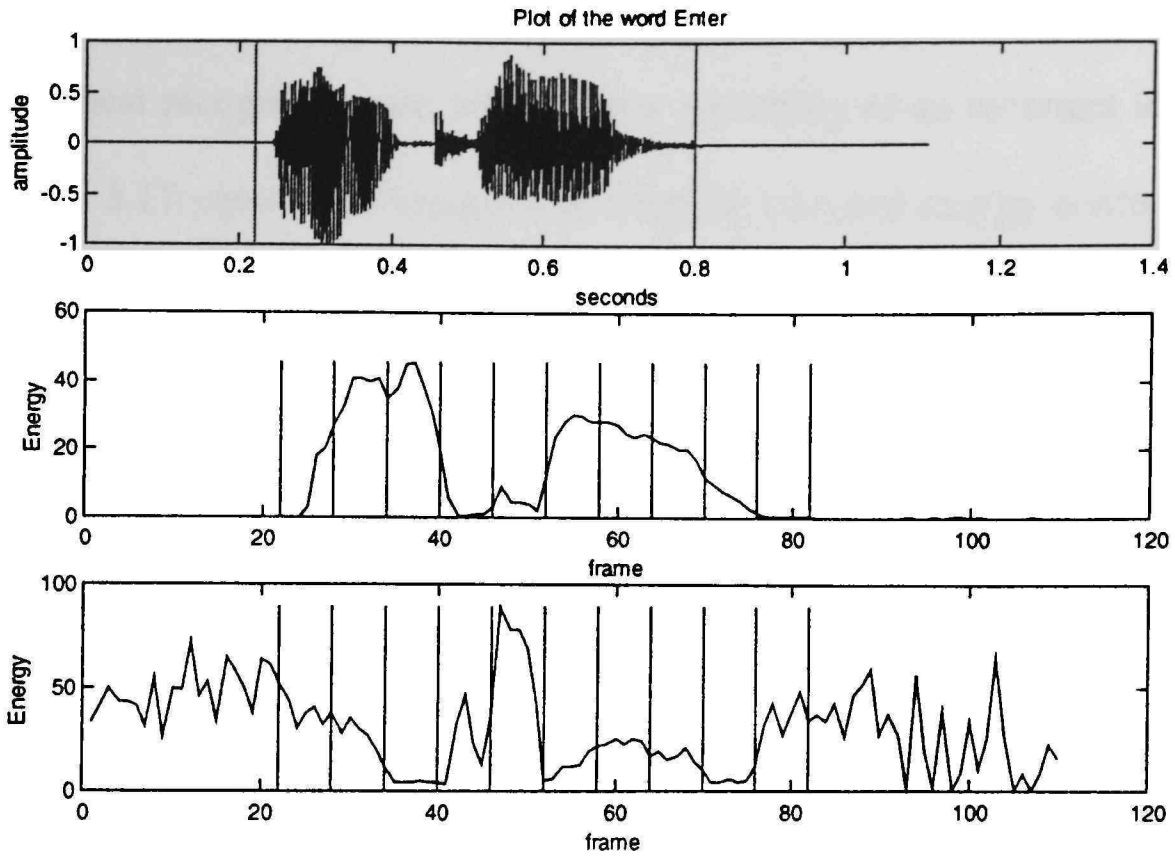

Figure 3.7 Plot of the word 'ENTER' showing equally spaced sections

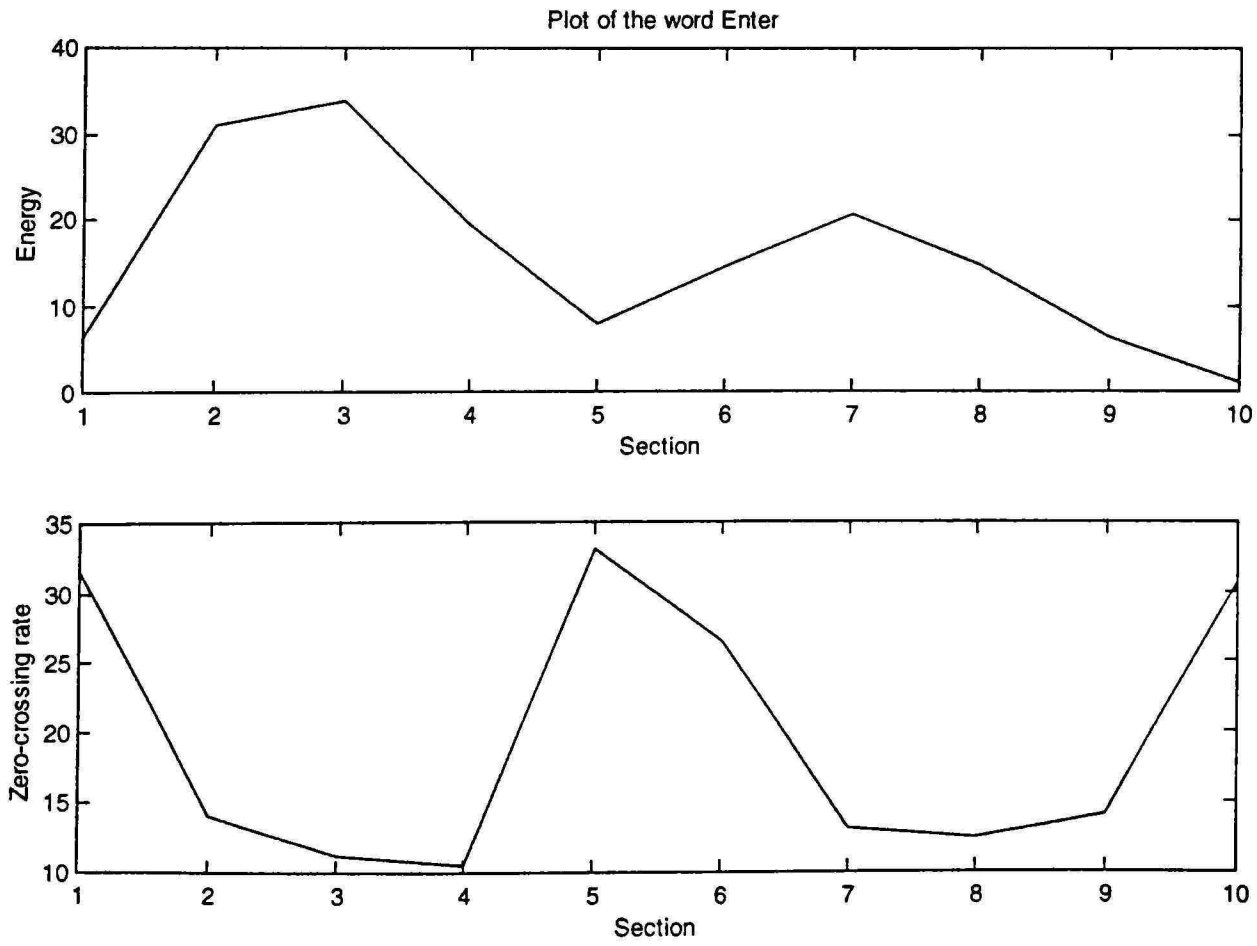

Figure 3.8 Average values of zero crossing and energy content for the word "Enter".

Another problem that needs to be addressed is how to choose the words so as to obtain the highest recognition rate and the least possibility of an incorrect identification. Figures 3.9 to 3.17 show the average zero crossing rate and energy content of the ten words in ten equally spaced sections. The reference template was obtained by first calculating the average zero crossing rate and the energy content in each of the ten sections. A feature vector was obtained by combining the energy and zero crossing rate into a single vector of twenty dimensions. This process was repeated for all the utterance of a given word and the feature vectors were averaged together to form the reference template. The standard deviation in each element of the feature vector for all the utterances of a given word was also calculated. This vector was designated as the radius of the "circle of confidence" and stored in the reference template along with the averaged feature vector.

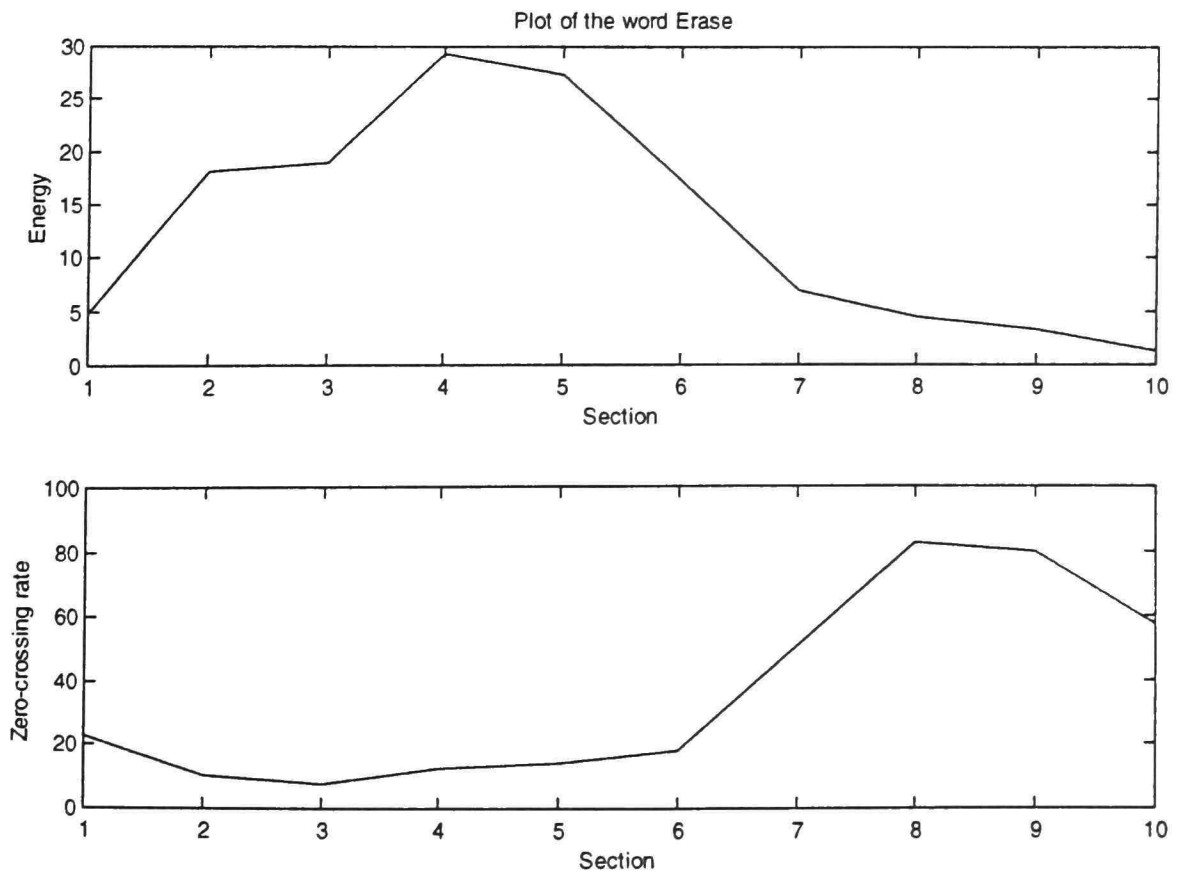

Figure 3.9 Average values of zero crossing and energy content for the word "Erase".

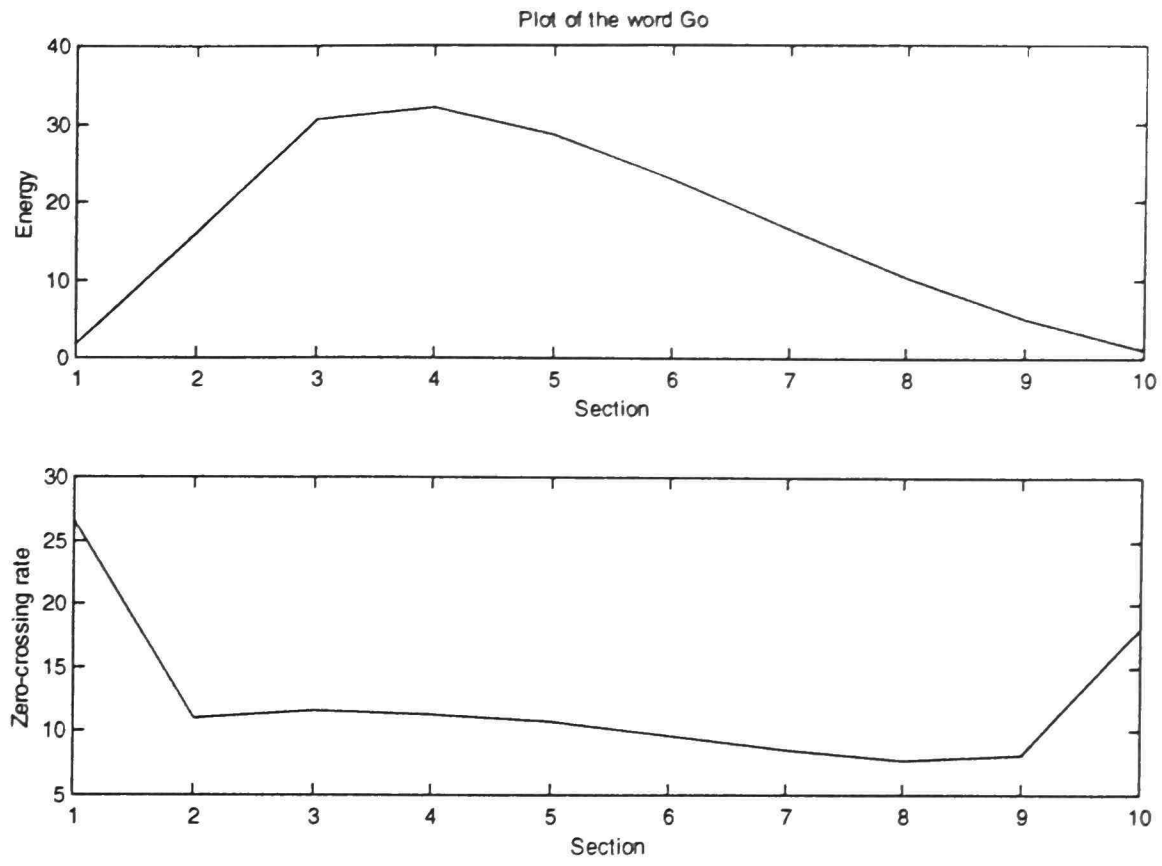

Figure 3.10 Average values of zero crossing and energy content for the word "Go".

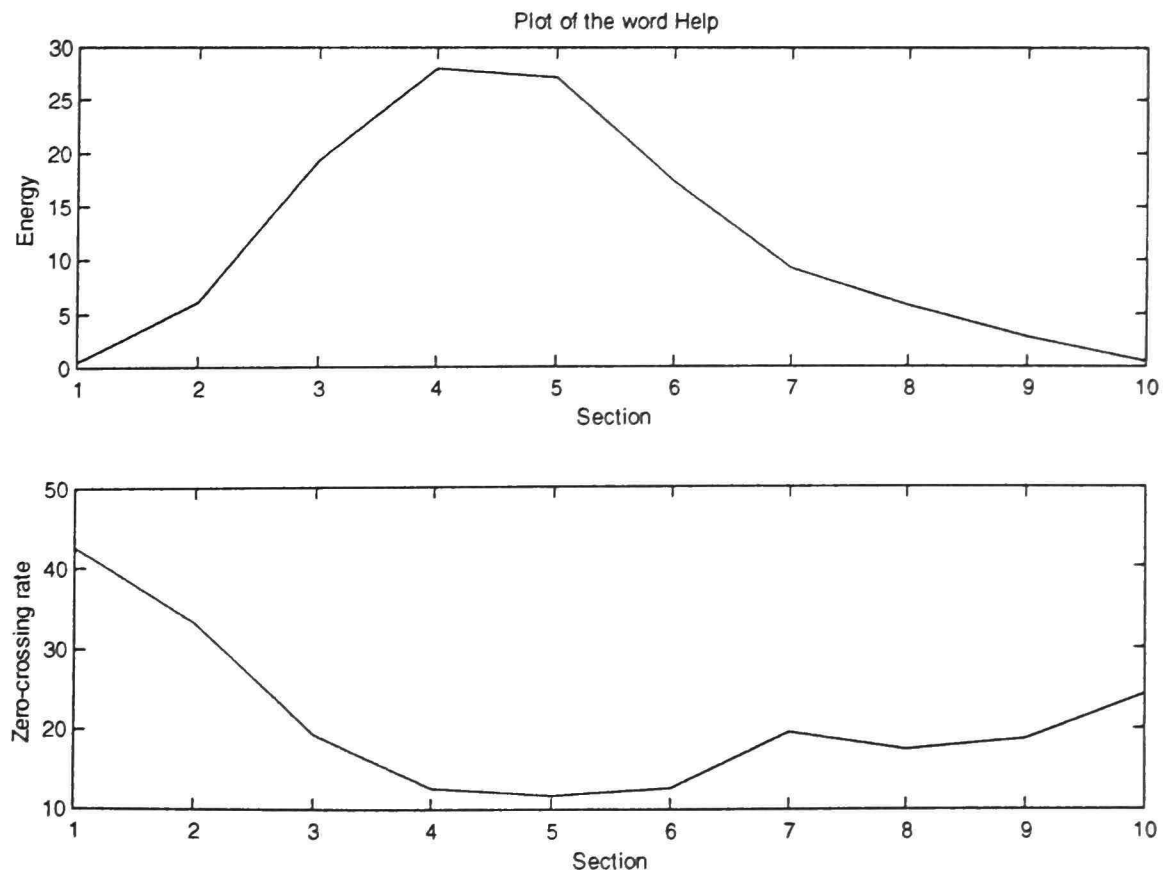

Figure 3.11 Average values of zero crossing and energy content for the word "Help".

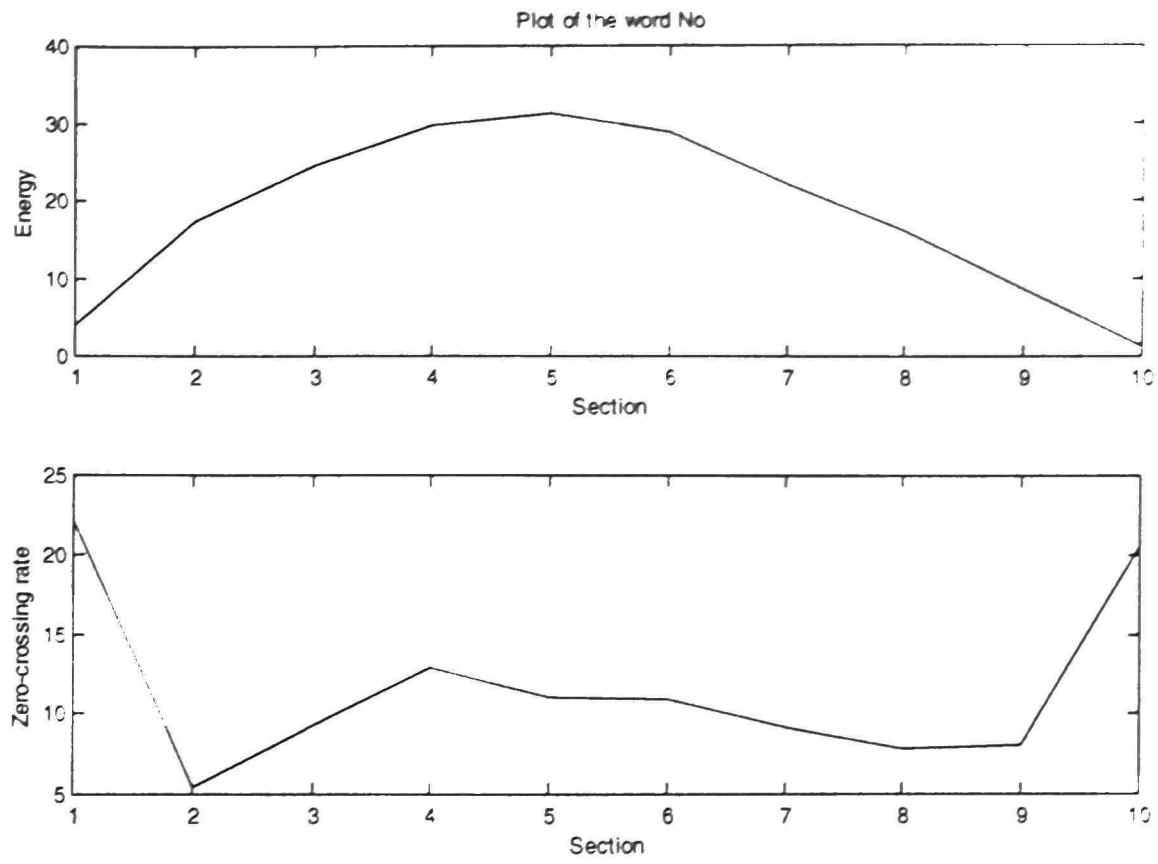

Figure 3.12 Average values of zero crossing and energy content for the word "No".

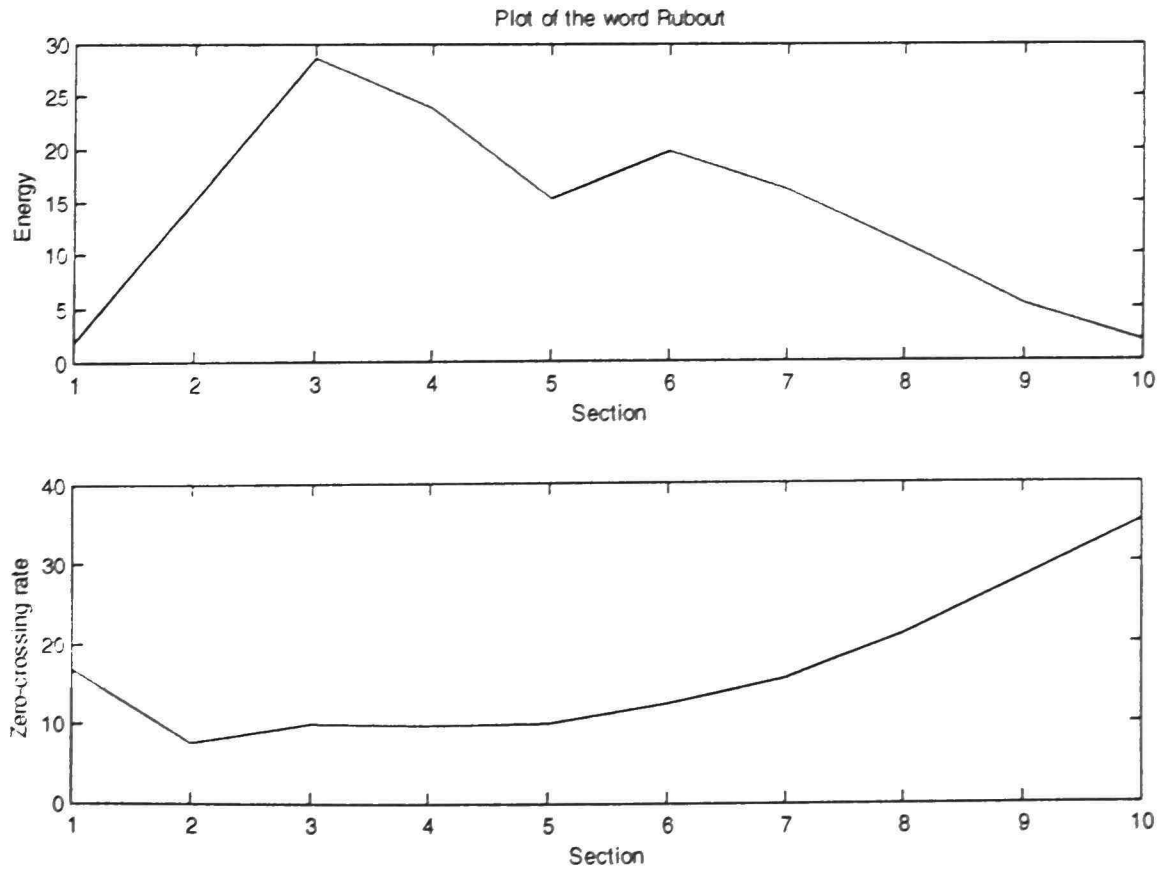

Figure 3.13 Average values of zero crossing and energy content for the word "Rubout".

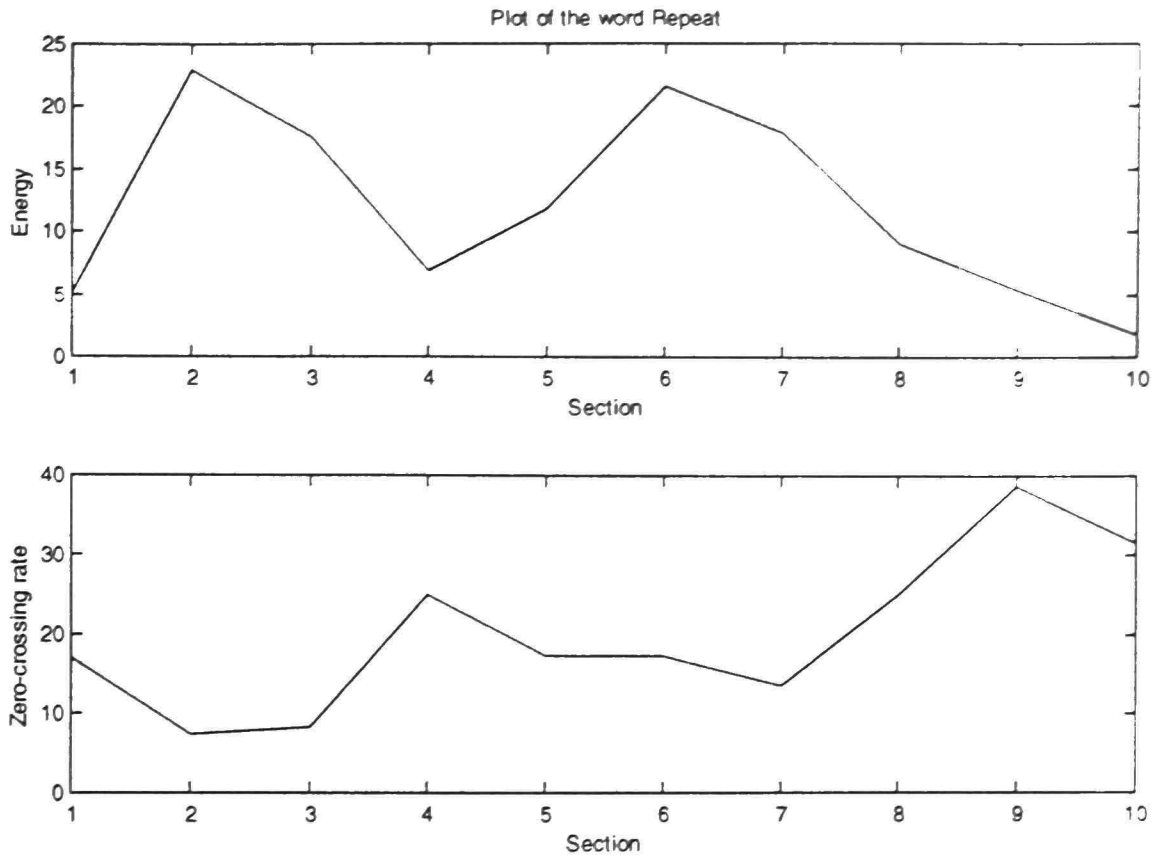

Figure 3.14 Average values of zero crossing and energy content for the word "Repeat".

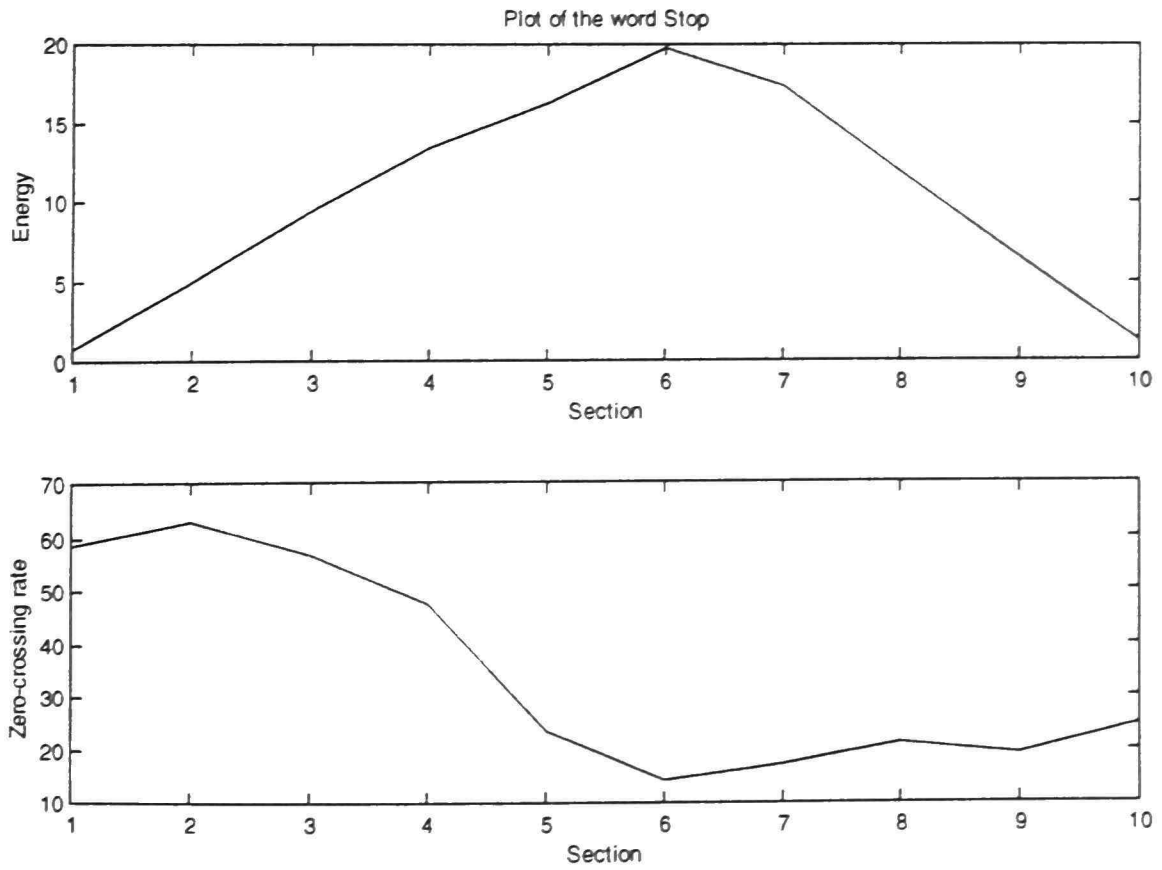

Figure 3.15 Average values of zero crossing and energy content for the word "Stop".

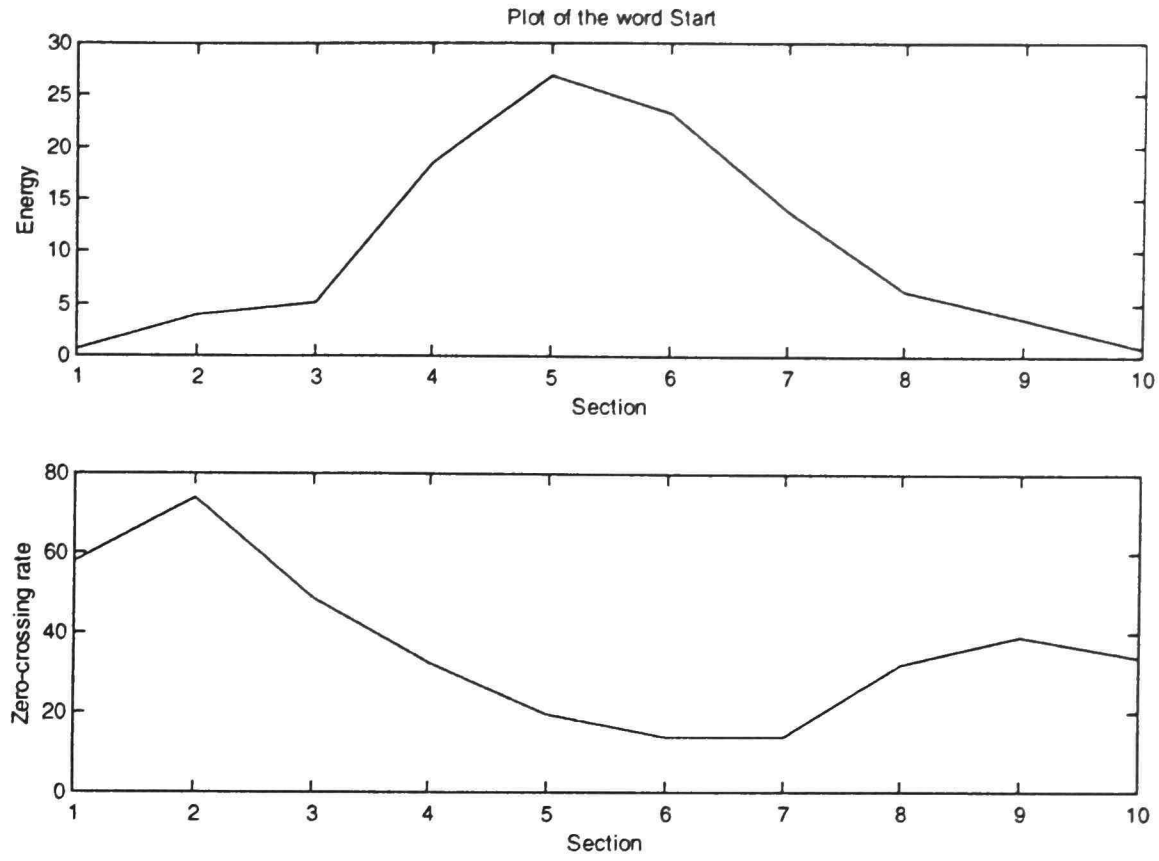

Figure 3.16 Average values of zero crossing and energy content for the word "Start".

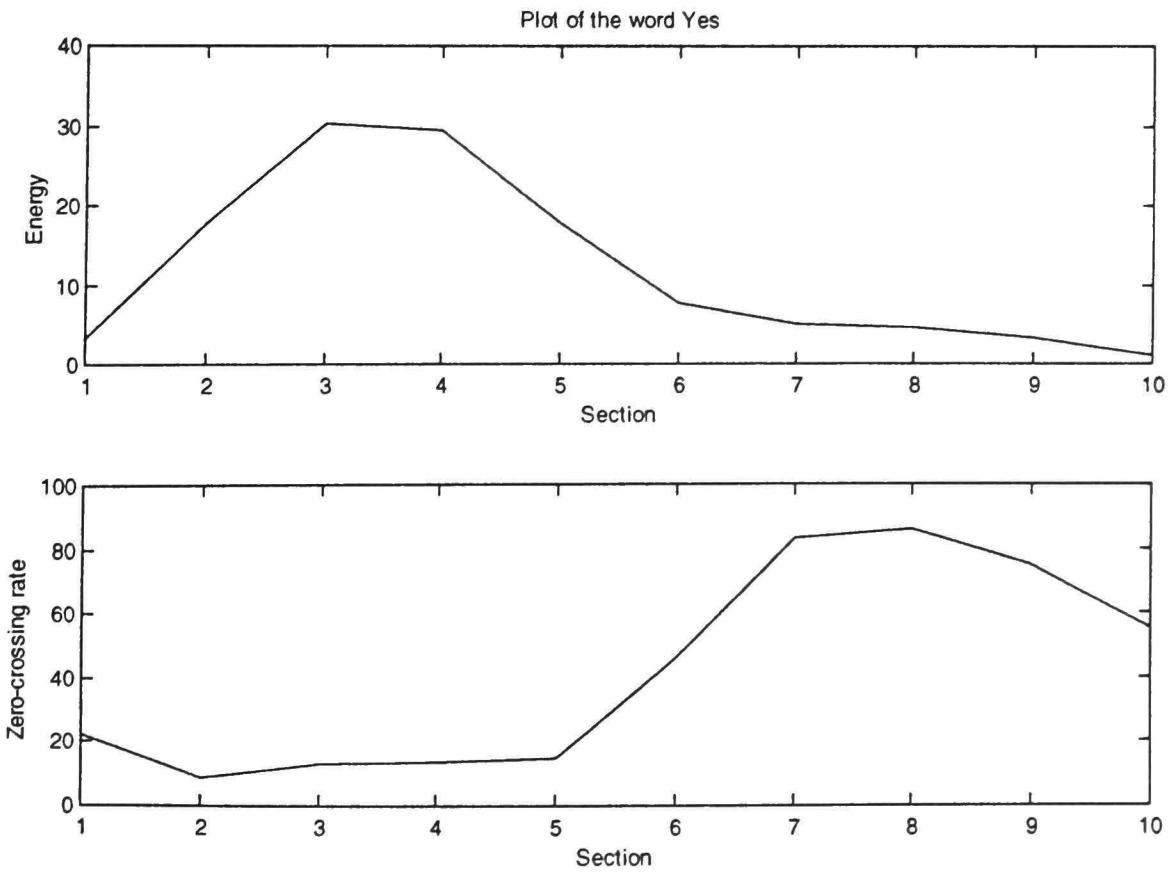

Figure 3.17 Average values of zero crossing and energy content for the word "Yes".

Figures 3.9 through 3.17 indicate that a few of the words are similar to each other and hence would not make good candidates for the speech recognition system. A more quantitative approach to the problem is provided in the next chapter.

3.6 Distance Measures

The last stage after the creation of the reference template is the actual recognition of an unknown utterance against the reference template. A number of different techniques exist to simplify the decision criteria. Most decision criteria involve some form of a distance measure. Given two vectors **x** and **y** in a multidimensional space, a metric $d(\cdot,\cdot)$ can be defined in the *N-dimensional real Cartesian space*, denoted $\Re^N$. The metric on $\Re^N$ is a real-valued function with three properties. For all

$\mathbf{x}, \mathbf{y}, \mathbf{z} \in \Re^N$,

1. $d(\mathbf{x}, \mathbf{y}) \geq 0$.
2. $d(\mathbf{x}, \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$.
3. $d(x,y) \leq d(x,z) + d(z,y)$.

Most metrics used in speech processing are special cases of the Minkowski metric. The Minkowski metric is defined as

$$d_s(\mathbf{x}, \mathbf{y}) \equiv \sqrt[s]{\sum_{k=1}^{N} |x_k - y_k|^s}, \tag{3.6}$$

where s is the *order of the Minkowski metric,* or the l_s *metric* and x_k is the kth component of the N-vector **x**. Two particular cases of the Minkowski metric are

1. The l_1 or *city block* metric,

$$d_1(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{N} |x_k - y_k|. \tag{3.7}$$

2. The l_2 or *Euclidean* metric,

$$d_2(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{k=1}^{N} |x_k - y_k|^2}. \tag{3.8}$$

A few other distance measures used in statistical pattern recognition are the maximum likelihood distance and probabilistic distance measures. These measures are computationally more expensive as they require calculations of the covariance matrix, determinants, probability density function (pdf), integrals and logarithms.

The Euclidean metric was utilized in this speech recognition system for the final decision. The decision is made by calculating the distance metric between the unknown utterance and all the words in the vocabulary. The minimum distance that falls within the cluster radius of the word is chosen as the correct word.

CHAPTER 4

IMPLEMENTATION OF THE SYSTEM

The previous chapter described the speech recognition system that has been developed. The algorithms for speech processing and recognition were also described. The next stage in the development of the speech recognition system is to implement all the algorithms on a platform and to test their performance to ascertain their proper operation before its final implementation in a digital signal processor.

The system was first implemented on a PC using a high level language. Mathworks Inc.'s MATLAB 5 was used to realize the speech recognition system. MATLAB is an integrated technical computing environment that combines numeric computation, advanced graphics and visualization, and a high-level programming language. MATLAB includes numerous tools for data and algorithm analysis. Most of these tools appear in the form of *toolboxes*. In order to keep the system implementation as simple as possible, none of the MATLAB's special purpose routines were utilized. This also facilitates the final implementation on a DSP.

The software implementation of the system on the PC consists of two main modules. The first module is the reference template creator and the second is the recognition module. Both these modules use a third module called the feature vector generator to complete their tasks. Figure 4.1 depicts the interaction between each of these modules.

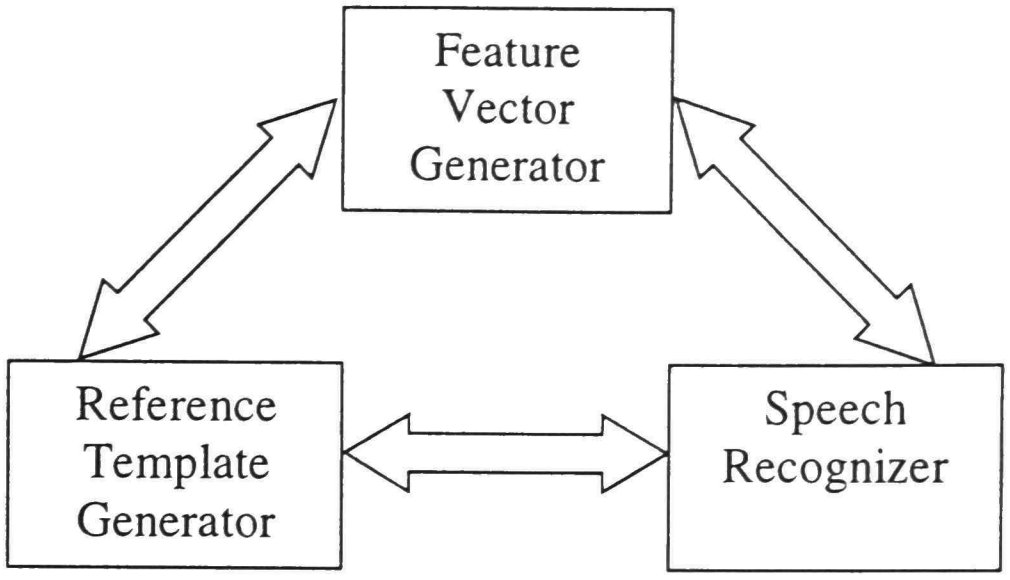

Figure 4.1 Interaction of software speech recognition modules.

The system essentially operates in one of two modes: reference template creation and speech recognition. In the reference template creation mode, the system accepts speech input and calls the feature vector generator to obtain the feature vector of the speech. The system then creates the reference template from a number of speech inputs. In the speech recognition mode, the system accepts an unknown speech and makes an attempt to recognize the speech based on the reference template created earlier.

One of the disadvantages of MATLAB is its inability to acquire sound in real-time. Most of today's PCs include a soundcard capable of acquiring sound from a number of sources including a microphone. Most soundcards have sampling rates as low as 4KHz to as high as 48Khz, and can sample at both 8 and 16 bits per sample. The PC based system instead accepts prerecorded speech as its input. The system assumes that the input speech has been sampled at 12.5KHz and at 16 bits per sample. Recording and collecting a number of speech utterances from a number of speakers for different words would have

been a daunting task. Instead one of the standard databases was chosen to expedite the process.

4.1 Speech Acquisition and Database

As mentioned earlier, the TI-46 database was used to test and evaluate the system. The speech utterances in the database were recorded in a low noise sound isolation booth, using an Electro-Voice RE-16 cardoid dynamic microphone, positioned two inches from the speaker's mouth and out of the breath stream. The speech signals were digitized at 16 bits per sample and sampled at 12500 samples per second. The signals were preprocessed at Texas Instruments to remove DC offset, filtering and other signal conditioning techniques were also applied to remove noise.

The database uses the NIST Speech Header Resources (SPHERE) file format for encoding the speech data [24]. The NIST SPHERE header is a 1024-byte American Standard Code for Information Interchange (ASCII) which is prepended to the waveform data.

The MATLAB software is not capable of decoding the NIST Sphere format. It can only read and decode Microsoft Windows Pulse Code Modulated (PCM) waveforms. Therefore it was necessary to convert all the files in the database to the Microsoft PCM format. The FMJ-Software's Awave [23] audio format converter/editor was used to accomplish this task. This software is capable of reading the NIST Sphere files and converting them to Microsoft Windows PCM WAV files.

The final database consisted of 32 utterances for each of the ten words. Sixteen of those utterances were classified as reference data and the rest were classified as test data.

The test and reference utterances consisted of utterances from 16 different male and 16 different female speakers. The entire database consisted of 320 speech samples of which 160 were designated as reference and the rest as test data.

4.2 Feature Vector Generator

After the acquisition of speech signal, the next step is to calculate the feature vector for the speech sample. As described in section 3.5 the speech waveform is divided into a number of equally spaced intervals. The zero crossing rate and energy content in each interval is measured.

The feature vector generator begins by first finding the endpoints of the utterance. It then divides the waveform into 10 equally spaced intervals. For each interval, the module calculates the average zero crossing and energy content for the frames in the interval. The following is the pseudo code for the Feature Vector Generator (FVG).

```
read wave file
calculate endpoints
calculate number of frames in each interval

for each interval do
      for each frame do
            calculate zero crossing
            calculate energy content
      end;
      store zero crossing and energy content for each interval
end

for each interval do
      calculate mean of zero crossing
      calculate mean of energy content
end

store mean zero crossing and mean energy content in a single
array

return array
```

Figures 3.5, 3.6 and 3.7 show the flowcharts for the endpoint algorithm.

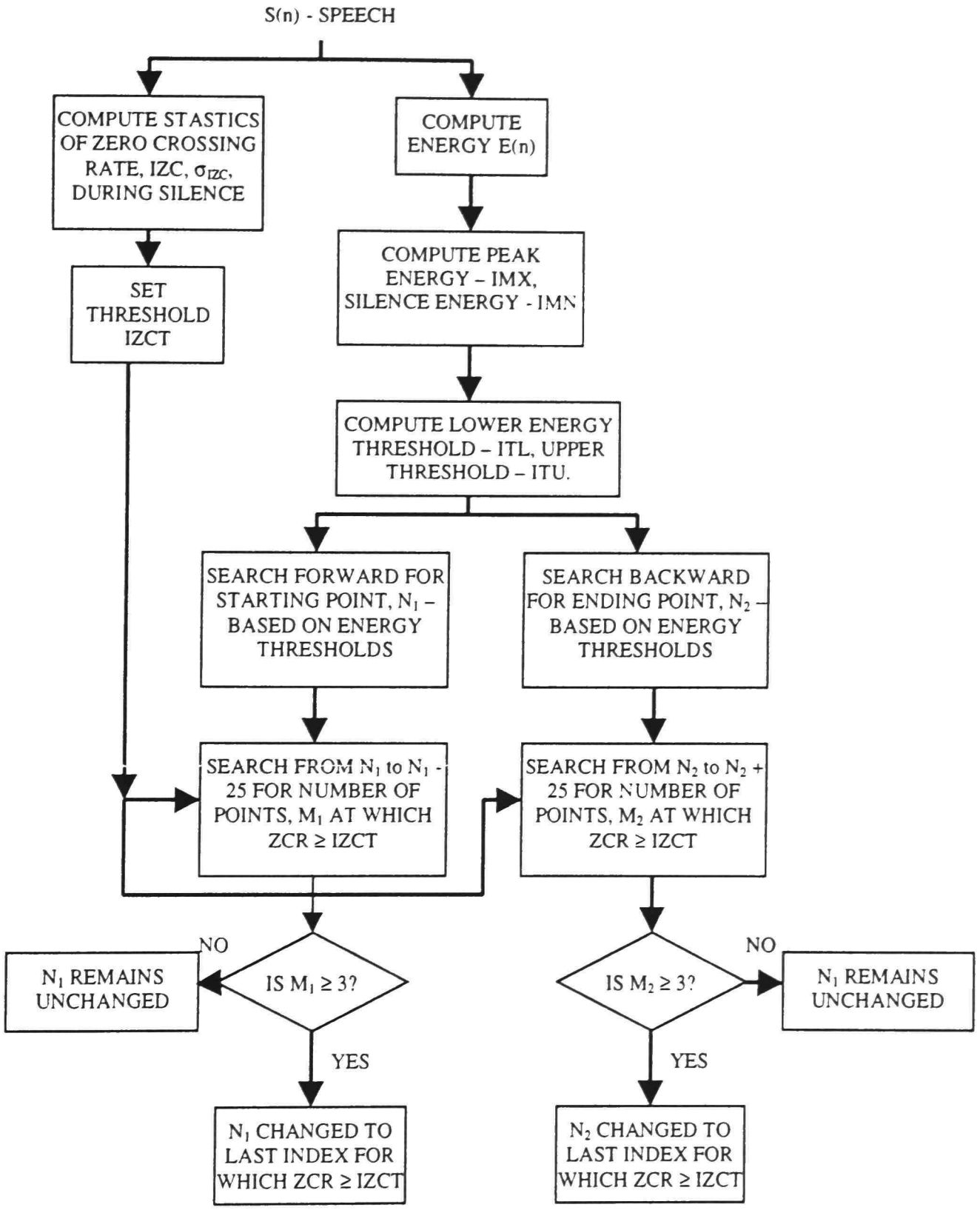

Figure 4.2 Flowchart for the endpoint algorithm.

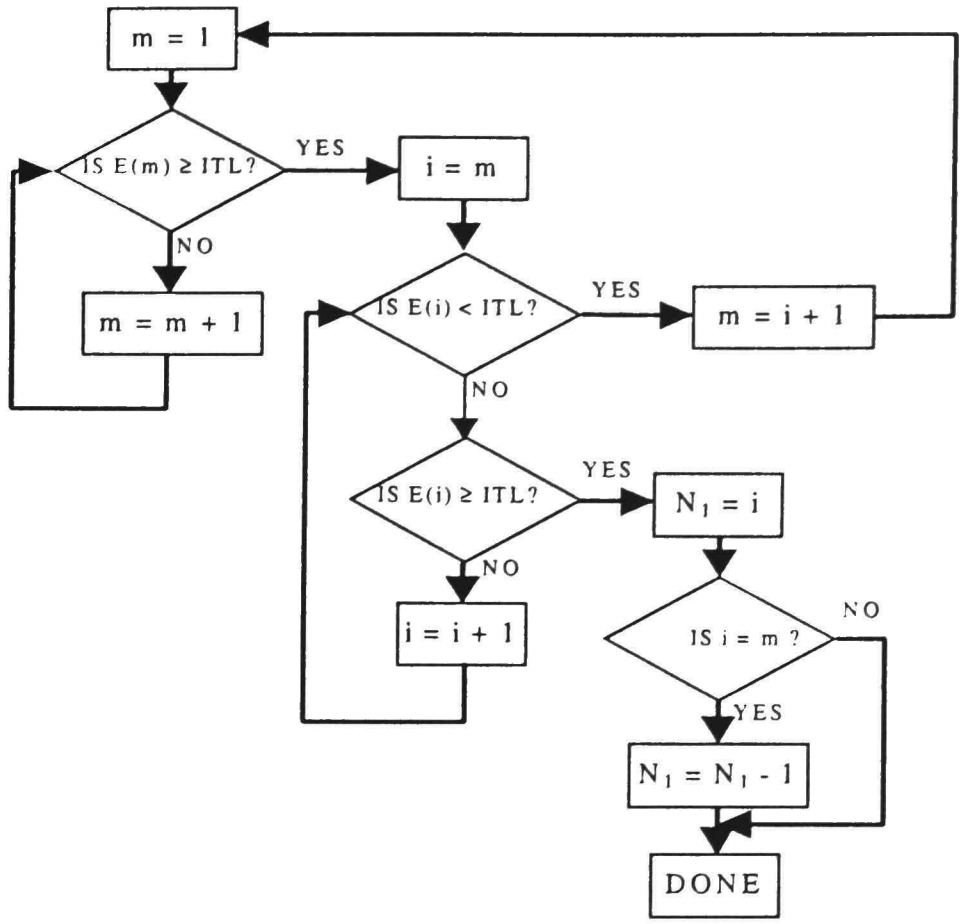

Figure 4.3 Flowchart for the beginning point initial estimate based on energy.

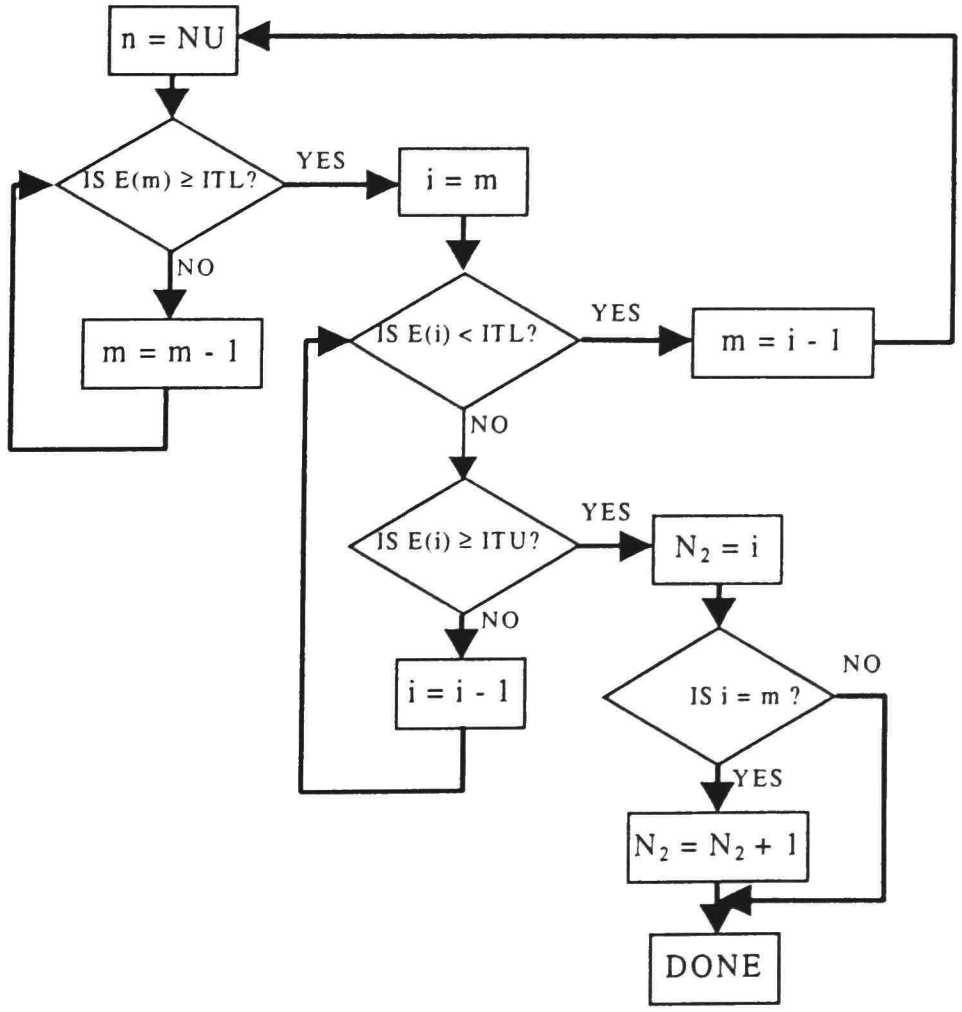

Figure 4.4 Flowchart for the ending point initial estimate based on energy

During the silence region of the word (first 10 frames) a zero crossing threshold, *IZCT*, is chosen as the minimum of a fixed threshold, *IF* (25 crossings per 10msec), and the sum of the mean zero crossing rate during silence, *IZC*, plus twice the standard deviation of the zero crossing rate during silence, i.e.,

$$IZCT = MIN(IF, IZC + 2\sigma_{IZC}). \tag{4.1}$$

The energy function for the entire interval *E(n)*, is then computed. The peak energy, *IMX*, and the silence energy, *IMN*, are used to set two thresholds, *ITL* and *ITU*, according to the rule

$$I1 = 0.03 * (IMX - IMN) + IMN \tag{4.2}$$
$$I2 = 4 * IMN \tag{4.3}$$
$$ITL = MIN(I1, I2) \tag{4.4}$$
$$ITU = 5 * ITL. \tag{4.5}$$

After the silence regions have been eliminated the program divides the waveform into ten equally spaced sections. The program accomplishes this by calculating the number of frames needed in each section to divide the signal data into ten equally spaced sections. Since the number of frames in each section may not be a whole number the program rounds up the number. This step will make the last section have fewer or more frames than the other sections. This aspect does not adversely affect the overall performance of the system. The zero crossing rate and energy content in each of the sections is calculated as discussed in sections 3.4.1 and 3.4.2. The final result is a feature vector of 20 dimensions for any given input utterance.

The next task in the development of the speech recognition system is to choose a number of words suitable for the system to recognize. Although it may be intuitively

apparent that similar sounding words will yield high error rates, a more quantitative method for word selection criteria is needed. A simple and efficient algorithm for word selection process is described in the next section.

4.3 Word Selection Process

In order to obtain a quantitative criteria for which words are good candidates, the correlation coefficient among the words was calculated. The join central moment of two random variables x and y maybe written as

$$c_{xy} = E\{(x-\mu_x)(y-\mu_y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\mu_x)(y-\mu_y)f_{xy}(x,y)dxdy. \qquad (4.6)$$

The correlation coefficient is then defined as

$$\rho_{xy} = \frac{c_{xy}}{\sigma_x \sigma_y}. \qquad (4.7)$$

Figure 4.5 shows pictorially the correlation between the different words. The picture was generated by calculating the correlation coefficient between feature vectors of different words.

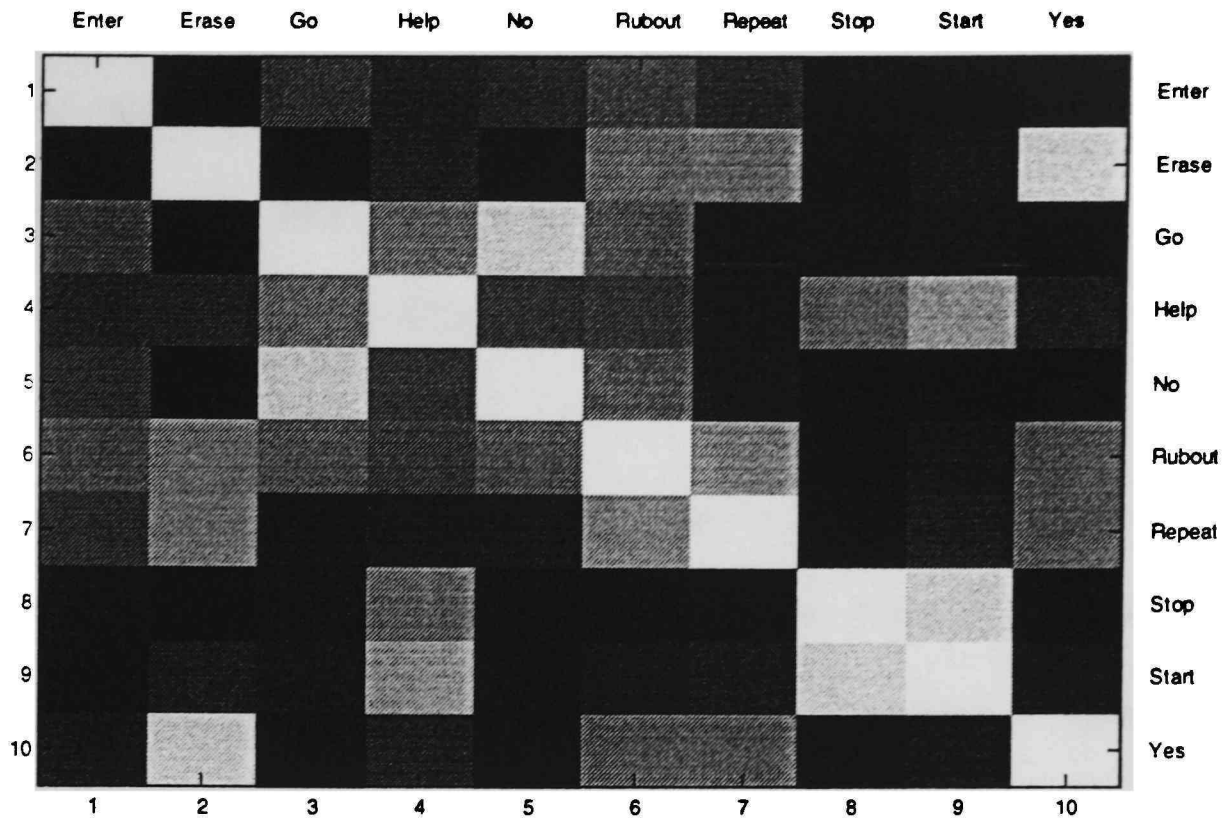

Figure 4.5 Correlation between words of the vocabulary.

The darkest areas in the graph show lowest correlation whereas the light areas are of high correlation. The graph is symmetric about its diagonal axis. This graph can be used to choose words of lowest correlation which would make the best candidates for the speech recognition system. With this in mind, it becomes quite clear that in order to add more words to the system one only need to obtain the feature vector for a new word and then calculate the correlation coefficient against other words in the vocabulary. The word that produces the lowest correlation against all the other words in the vocubulary would make the best candidate for the system.

From Figure 4.5 a few three and four-word combinations were chosen to test the system. The following are the combination of words used:

Set I. 'ERASE', 'GO', and 'START'

Set II. 'GO', 'REPEAT', and 'STOP'

Set III. 'ENTER', 'NO', and 'YES'

Set IV. 'NO', 'STOP' and 'YES'

Set V. 'ENTER', 'NO', 'STOP', and 'YES'

Set VI. 'ERASE', 'GO', 'REPEAT', and 'STOP'

Now that a method of obtaining the feature vectors and a suitable vocabulary is established, the next step is to create a reference template from the different words and their utterances. The reference template will then be used to recognize unknown utterances.

4.4 Reference Template Creation

The reference template creation module starts by asking the user for which words to create the reference templates. It then asks for the number of utterances in each word. The names of the files have been chosen in such a way to facilitate this aspect. The filename starts with 'wd' and is followed by a word number and then 's' and finally the utterance number. For example, 'WD5S24.WAV' refers to word number 5, which is 'NO' and utterance number 24, which correspond to a male speaker in the test data.

The reference creation module then calls the FVG module that calculates the feature vector for an utterance of one of the words. The program repeats this process until the feature vectors for all the utterances of a given word have been calculated. The program then calculates the mean and the standard deviation of all the feature vectors for a given word. Each word is associated with a particular radius, which is calculated by finding the Euclidean distance between the mean vector and another vector that is one

standard deviation away from the mean. This radius serves the purpose of a "circle of confidence" around a word, which allows an unknown word to be recognized at a later time.

This entire process is repeated until a reference template is created for all the words in the vocabulary. The reference template finally contains a representative feature vector along with a radius of "circle of confidence" for all the words in the vocabulary. The following is the pseudocode for the reference template creation module.

```
for each word in the vocabulary do
  for each utterance of the word do
      calculate feature vector using FVG module
  end
end

for each word in the vocabulary do
      calculate the mean of all feature vectors from all the
utterances
end

for each word in the vocabulary do
      calculate the standard deviation of all feature vectors
from all the utterances
end

for each word in the vocabulary do
      Add the standard deviation to the mean in another variable
called wordp1s
end

/* This is the radius associated with each word */
for each word in the vocabulary do
      calculate the euclidean distance between the mean and
wordp1s
end

return euclidean distance
return word mean
```

The MATLAB code for the reference template creator is provided in Appendix B.

4.4 Speech Recognition Module

Once a reference template is created for the words in the vocabulary, the system is now ready to recognize unknown input utterances. The speech recognition module accepts any prerecorded unknown utterance and tries to identify it from the reference template. The module calls the FVG module to calculate the feature vector of the utterance and then calculates the Euclidean distance between the unknown utterance and the feature vector of all the words in the reference template. The program makes an initial guess by choosing the word with the minimum distance. It then determines if the distance is within a circle of the radius associated with the word in the reference template. If the unknown utterance falls within that radius, the utterance is identified as belonging to that word, otherwise the utterance is reported as not recognized. This process is depicted pictorially in Figure 4.6

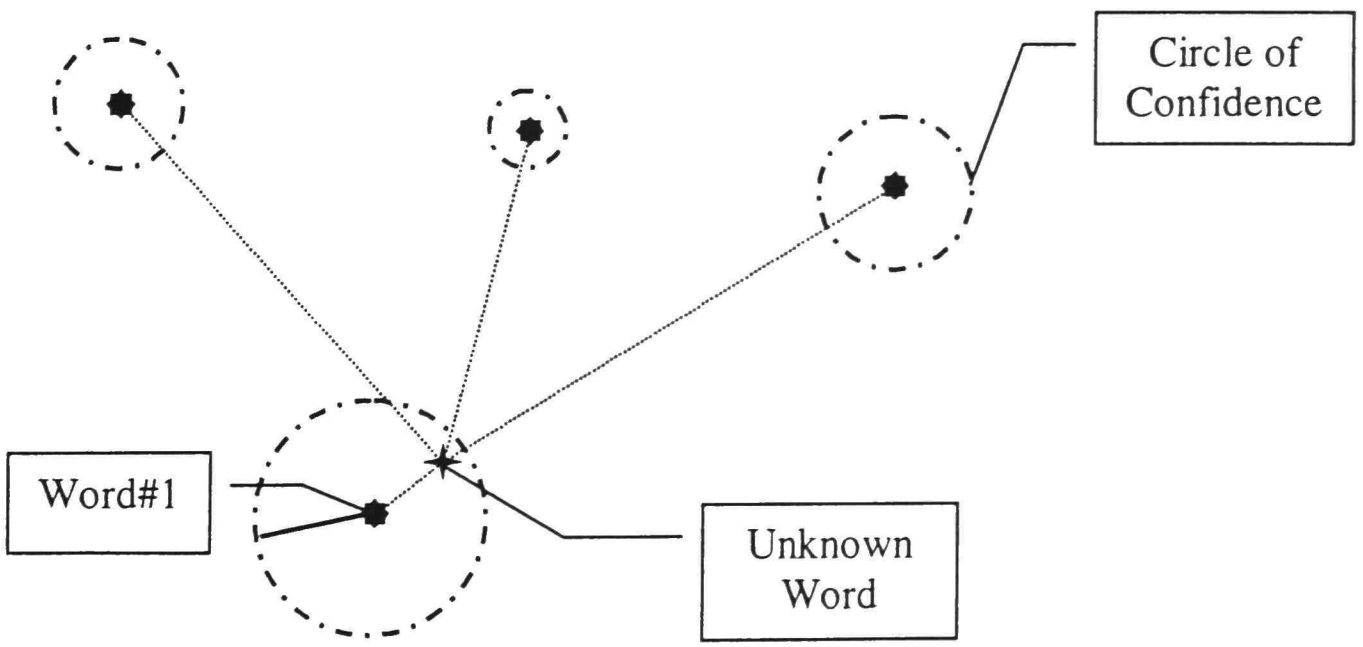

Figure 4.6 Pictorial representation of Unknown word decision.

As can be seen from the diagram that the unknown word belongs to word#1. The following is the pseudo code for the speech recognition module.

```
load reference template
read unknown utterance

calculate feature vector by using FVG

for all the word in the vocabulary do
      calculate distance between unknown feature vector and the
      words in the vocabulary
end

find word with minimum distance

if distance < radius of word
      the unknown is identified as the word
else
      the unknown was unrecognized
end
```

The MATLAB code for the entire speech recognition system is provided in Appendix B.

4.5 Speech Recognition Results

Initially the system was programmed to recognize one word at a time. A reference template consisting of one word was created from 16 utterances, designated as reference data, of that word (8 male and 8 female). The system was then tested against all the utterances designated as test data in the database for performance evaluation. The number of correct and incorrect recognition for all the utterances was recorded. Table 4.1 summarizes the results of the experiment. The columns report what was the programmed word and rows is the number of correct utterances the system identified when 16 different utterances, 8 male and 8 female, were provided as input.

Table 4.1 Recognition results for one word recognition.

	The programmed word									
	Enter	Erase	Go	Help	No	Rubout	Repeat	Stop	Start	Yes
Enter	10	0	0	2	0	2	5	3	1	0
Erase	0	11	0	0	0	0	0	0	0	7
Go	6	0	11	7	11	6	4	0	0	0
Help	0	0	2	11	1	0	1	6	5	0
No	3	0	7	5	11	6	5	0	0	0
Rubout	7	3	1	2	3	11	10	0	0	0
Repeat	0	0	0	0	0	0	11	0	0	0
Stop	0	0	0	2	0	1	1	9	6	0
Start	0	0	0	2	1	0	1	8	12	0
Yes	0	7	0	0	0	1	0	0	0	12
Correct (%)	63	69	69	69	69	69	69	56	75	75

In many respects the above table is very similar to Figure 4.5 as it shows exactly the combination of words that would not work well at all in the system.

Tables 4.2 through 4.4 show the results of the speech recognition system for sets of data described in section 4.3. The reference template consisting of the words in the sets was created from the utterances in the reference data. In each case the system was provided with 16 utterances, 8 male and 8 female, of each word in the test data.

Table 4.2 Speech Recognition Results for Sets I and II

	Erase	Go	Start
Erase	11		
Go		10	
Start			12
Overall Correct (%)	69		
Incorrect (%)	0		

	Go	Repeat	Stop
Go	11		
Repeat		10	
Stop			9
Overall Correct (%)	63		
Incorrect (%)	0		

Table 4.3 Speech Recognition Results for Sets III and IV

	Enter	No	Yes
Enter	9		
No		11	
Yes			12
Overall Correct (%)	67		
Incorrect (%)	0		

	No	Stop	Yes
No	10		
Stop		9	
Yes			12
Overall Correct (%)	65		
Incorrect (%)	0		

Table 4.4 Speech Recognition Results for Sets V and VI

	Enter	No	Stop	Yes
Enter	10			
No		11		
Stop			9	
Yes				12
Overall Correct (%)	66			
Incorrect (%)	0			

	Erase	Go	Repeat	Stop
Erase	11			
Go		11		
Repeat			10	
Stop				9
Overall Correct (%)	64			
Incorrect (%)	0			

As can be seen from the results the speech recognition system made no errors in misidentifying words and only reported a few words as not being recognized. As more words were added to the vocabulary, the perfomance of the system was greatly reduced. One of the main reason for the performance degradation is the limited vocabulary in the database from which the system was built. New words can be added to the system if ones of low correlation as described earlier in the chapter can be found.

4.6 DSP Implementation of the system.

Although the final system was not fully implemented on a DSP platform, a number of comparisons can be made. The Texas Instrument's TI320C3x DSP was chosen as a baseline system. The Texas Instrument's TI320C3x DSP is a single cycle processor. This allows the processor to complete any floating point operation in a single clock cycle. In addition, the TI DSP is based on a Single Instruction Multiple Data (SIMD) architecture. The SIMD architecture allows the DSP to perform a floating point operation on multiple pieces of data. This can significantly improve the speed of common signal processing task. The TI DSP can also multiply and add data items simultaneously. Multiply-Add operations are fairly common in signal processing tasks. Figure 4.7 shows the block diagram of the TI TMS320c3x DSP.

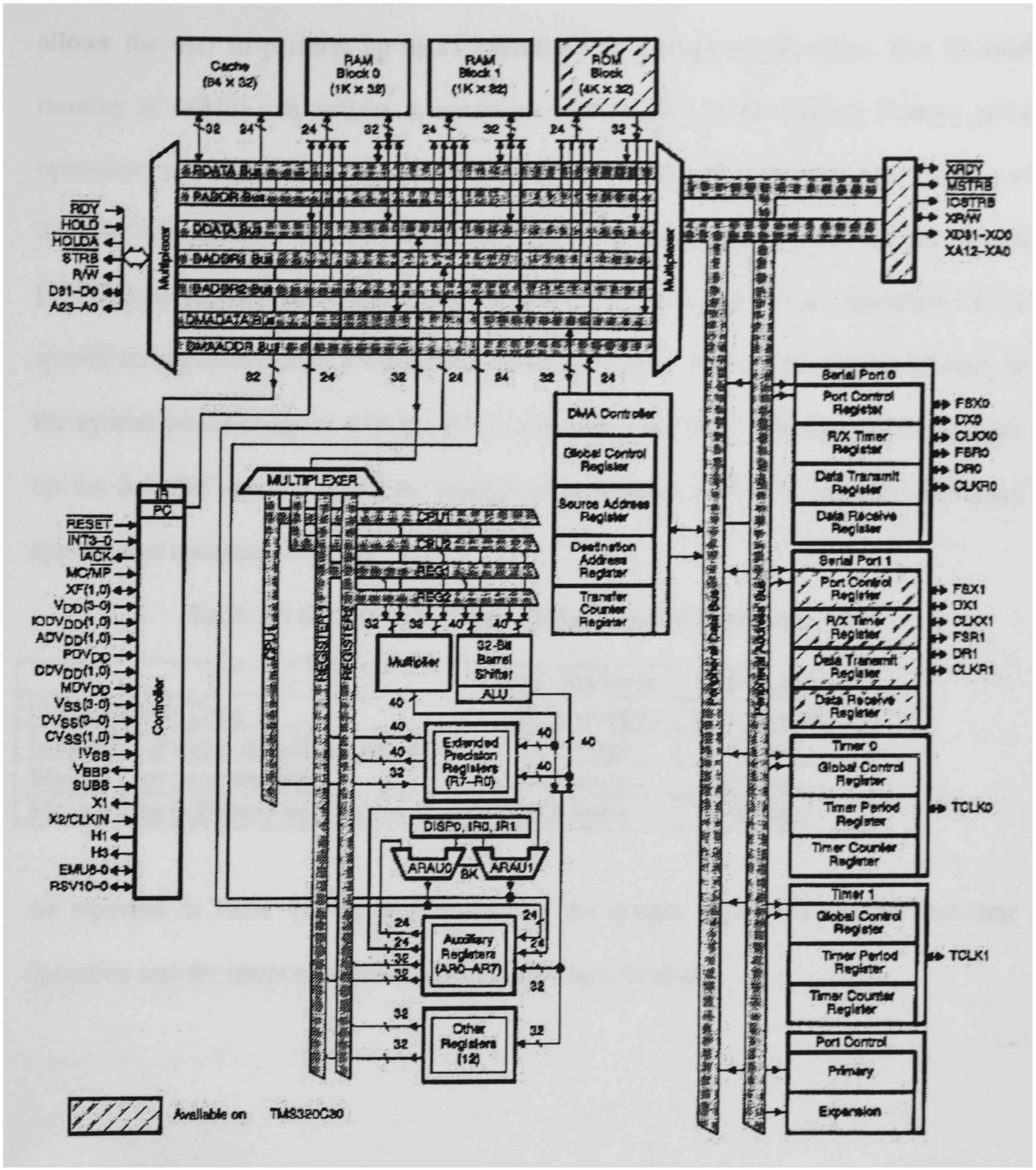

Figure 4.7 Block Diagram of Texas Instrument's TMS32C3X DSP.

Figure 4.7 shows that the TMS32C3x also employs a Harvard architecture with its separate instruction and data memory space. The processor also incorporates multiple address and data buses for both instruction and data memory. This extensive parallelism

allows the user to perform up to 11 operations in a single clock cycle. The TI DSP running at 40Mhz can sustain a maximum performance of 60 million floating point operations per second [24]. In comparison an Intel Pentium II microprocessor running at 233Mhz can perform approximately 100 million floating point operations per second [25]. Hence the two platforms are comparable in speed as far as the simulation of the speech recognition system is concerned. Table 4.4 shows how a DSP implementation of the system would compare with the PC version described here. The comparison is made on the basis of memory space for storage of algorithms, reference template and input speech, and computation speed.

Table 4.5 Comparison of Speech Recognition Implementation

	PC Platform	DSP Platform
Computation Time	0.32 seconds	0.32 seconds
Memory for input speech(1.5 seconds)	38 Kbytes	38 Kbytes
Memory for program code	200 Kbytes	16 Kbytes
Memory for reference template (4 words)	16 Kbytes	16 Kbytes

As reported in Table 4.5 the performance of the system is fast enough for real-time operation and the memory requirements are also very modest.

CHAPTER 5

CONCLUSION

The speech recognition system described in this thesis is an attempt at developing a system for a consumer level application. The approach presented here describe a variety of robust measurements and techniques that can be implemented on a platform suitable for consumer appliances. The speech recognition system discussed was implemented using the techniques of zero crossing rate and energy content with a template-matching algorithm. An algorithm for a word selection process, based on correlation coefficient, was also presented.

The system was initially developed and implemented on a PC platform. The flexibility of the system allows it to be easily implemented on a DSP. The PC-based system reported a 0% error rate on the test database of 16 different speakers with a vocabulary of four words. The system was able to recognize unknown utterances in less than 0.4 seconds. The speech recognition system developed here allows new words to be added with relative ease. A simple procedure based on the correlation coefficient to ensure the suitability of a new word was also provided. The memory required to implement the speech recognition system on a DSP is just 60 Kbytes. The speed, accuracy and memory requirements make this system a cost-effective approach to speech recognition for the masses.

The speech recognition system described here can be improved in a number of ways. One of the areas where the system can benefit the most is the temporal alignment

problem. The techniques of Dynamic Time Warping (DTW) can help resolve the problem and improve the recognition rate of the system. As discussed earlier, DTW attempts to "shrink" or "expand" the input speech waveforms to match the one in the reference template. The problem reduces to that of a minimization of an input to reference mapping. Although computationally expensive, the exponential growth in speed of DSP processors can realize the problem in real time in the near future.

Another problem that needs to be addressed is the decision algorithm. The existing system calculates the Euclidean distance without regard to the "weighting" of the energy content and the zero crossing rate. A suitable distance measure would incorporate appropriate weights for the measures. The weighting requires further understanding of the relationship between energy content and zero crossing rates. Other common distance measures include cepstrum-style log similarity measures [27].

The system also assigns a circle of confidence whose radius is only one standard deviation away from the average. This radius was chosen by a trial and error process, a more analytical process needs to be develop to help determine the radius. A more formal method would be to cluster the reference data using a clustering algorithm. Some common clustering algorithms include the chainmaps, the shared nearest neighbor procedure, the k-means iteration, Isodata, and numerous algorithms based on Fuzzy Logic [11]. The clustering algorithm can also help create more than one reference template per word. This procedure inherently incorporates information regarding the various nuances in the pronunciation of a given word among a number of different speakers [27].

Finally, a study of various new DSP for speech processing and recognition should also be undertaken. Many new multimedia processors incorporate a number of different modules into a single processor. This could lead to more cost-efficient and near real-time implementations of robust speech recognition systems.

REFERENCES

[1] K.H. Davis, R. Biddulph, and S. Balashek, "Automatic Recognition of Spoken Digits," *J. Acoust. Soc. Am.,* 24 (6): 637-642, 1952.

[2] J. Suzuki and K. Nakata, "Recognition of Japanese Vowels – Preliminary to the Recognition of Speech," *J. Radio Res. Lab,* 37 (8): 193-212, 1961.

[3] T. Sakai and S. Doshita, "The Phonetic Typewriter, Information Processing 1962," *Proc. IFIP Congress,* Munich, 1962.

[4] K. Nagata, Y. Kato, and S. Chiba, "Spoken Digit Recognizer for Japanese Language," *NEC Res. Develop.,* No. 6, 1963.

[5] T. B. Martin, A. L. Nelson, and H. J. Zadell, "Speech Recognition by Feature Abstraction Techniques," Tech. Report AL-TDR-64-176, Air Force Avionics Lab, 1964.

[6] T. K. Vintsyuk, "Speech Discrimination by Dynamic Programming," *Kibernetika,* 4 (2): 81-88, Jan.-Feb. 1968.

[7] D. R. Reddy, "An Approach to Computer Speech Recognition by Direct Analysis of the Speech Wave," Tech. Report No. C549, Computer Science Dept., Stanford University, September 1966.

[8] V. M. Velichko and N. G. Zagoruyko, "Automatic Recognition of 200 Words," *Int. J. Man-Machine Studies,* 2: 223, June 1970.

[9] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.,* ASSP-26 (1): 43-49, February 1978.

[10] F. Itakura, "Minimum Prediction Residual Applied to Speech Recognition," *IEEE Trans. Acoustics, Speech, Signal Proc.,* ASSP-23 (1): 67-72, February 1975.

[11] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Trans. Acoustics, Speech, Signal Proc.,* ASSP-27 (4): 336-349, August 1979.

[12] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE Acoustics, Speech, and Signal Processing Magazine,* 4 (2): 4-22, April 1987.

[13] A. Weibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," *IEEE Trans. Acoustics, Speech, Signal Proc.*, 37: 393-404, 1989.

[14] Texas Instruments (TI) and the National Institute of Standards and Technology (NIST), *TI 46-Word Speaker-Dependent Isolated Word Corpus*, NIST Speech Disc 7-1.1, September 1991.

[15] 21st Century Eloquence http://voicerecognition.com/1998/trends/

[16] M. R. Sambur and L. R. Rabiner, "A Speaker-Independent Digit-Recognition System," *The Bell System Technical Journal* Vol. 54, No. 1, January 1975.

[17] Itakura F, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 23, February 1985,* pp 57-72.

[18] 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," *The Bell System Technical Journal*, 1983.

[19] Rabiner, L. R. and Sambur, M. R., "An Algorithm for Determining the Endpoints of Isolated Utterances," *The Bell System Technical Journal,* Vol. 54, No. 2, February 1975.

[20] Price, P. J., W. Fisher, J. Bernstein et al. "A database for continuous speech recognition in a 1000-word domain," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, vol. 1, pp.651-654, 1988.

[21] Lawrence R. Rabiner and Ronald W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, New Jersey, 1978.

[22] NIST Sphere file format
http://squid.eng.cam.ac.uk/~ajr/wsjcam0/node11.html#SECTION00082000000000000000

[23] FMJ-Software, Awave, http://hem.passagen.se/fmj/fmjsoft.html

[24] *Texas Instruments TMS320C3x Reference Manual.* Dallas, TX: Texas Instruments, 1995.

[25] Doug Rasor, "Can DSP and NSP coexist in multimedia?," *Texas Instruments WWW page*, http://www.ti.com/sc/docs/integrat/95may/dspcol.htm

[26] Ferrel G. Stremler, *Introduction to Communication Systems*, Third ed. Reading, Massachusetts: Addison-Wesley, 1990.

[27] Lawrence R. Rabiner, "On Creating Reference Templates for Speaker Independent Recognition of Isolated Words," *IEEE Transactions on Acousticcs, Speech, And Signal Processing,* vol. ASSP-26, No. 1, February 1978.

APPENDIX A

SHORT TIME ENERGY AND ZERO CROSSING DATA

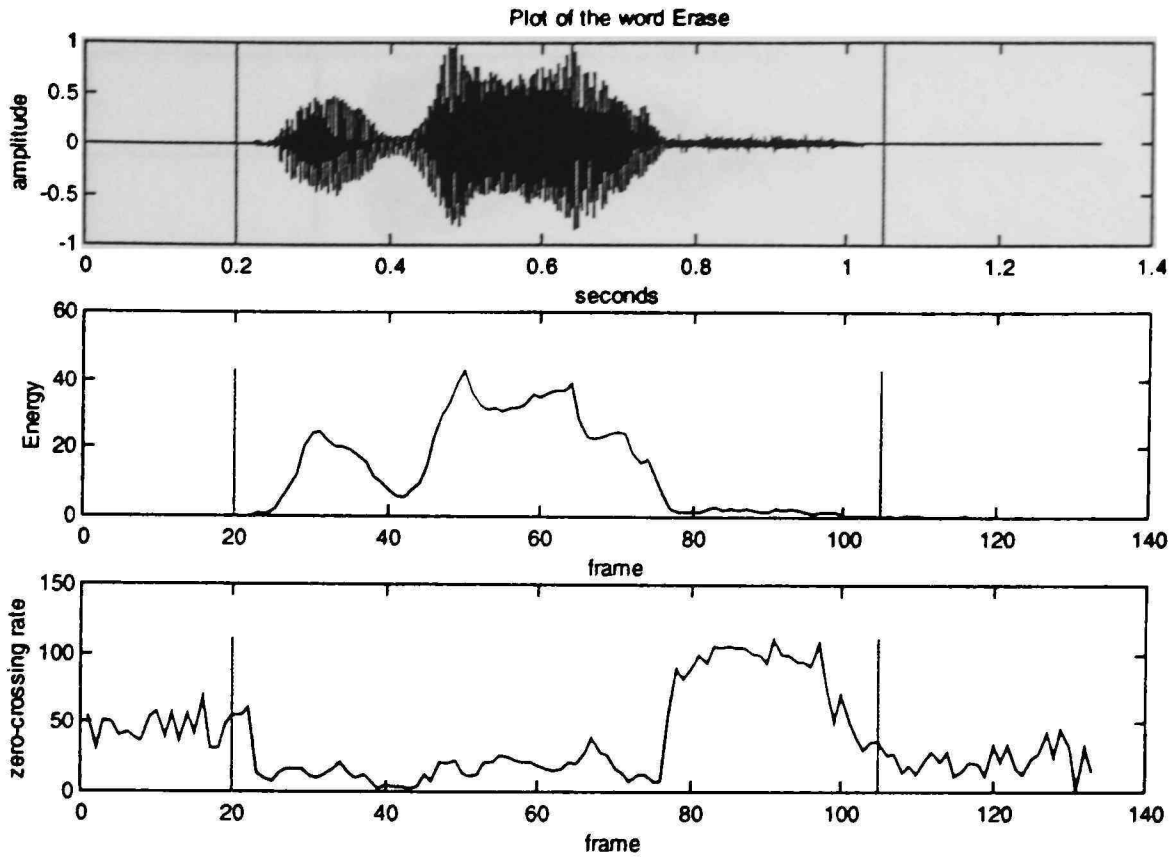

Figure A.1 Short time energy and zero crossing data for the word "Erase" by a female speaker.

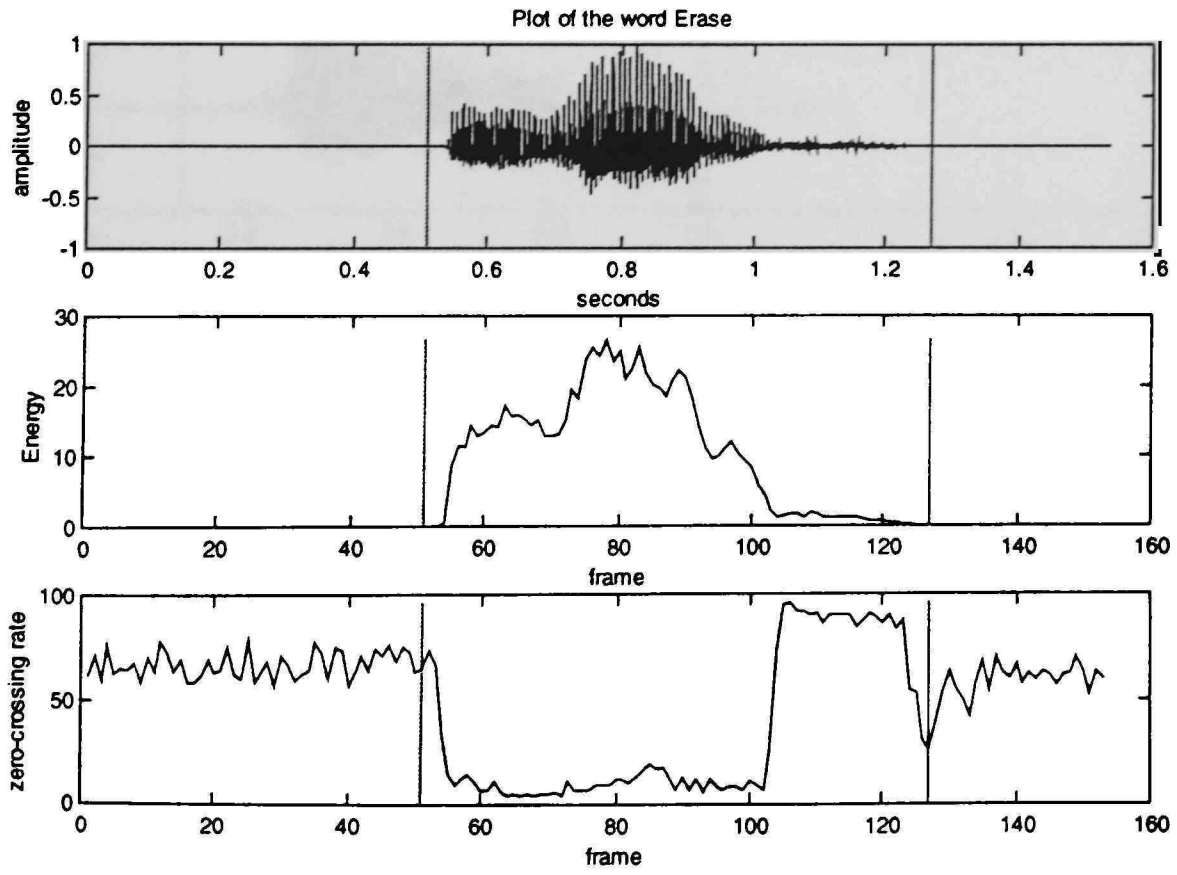

Figure A.2 Short time energy and zero crossing data for the word "Erase" by a male speaker.

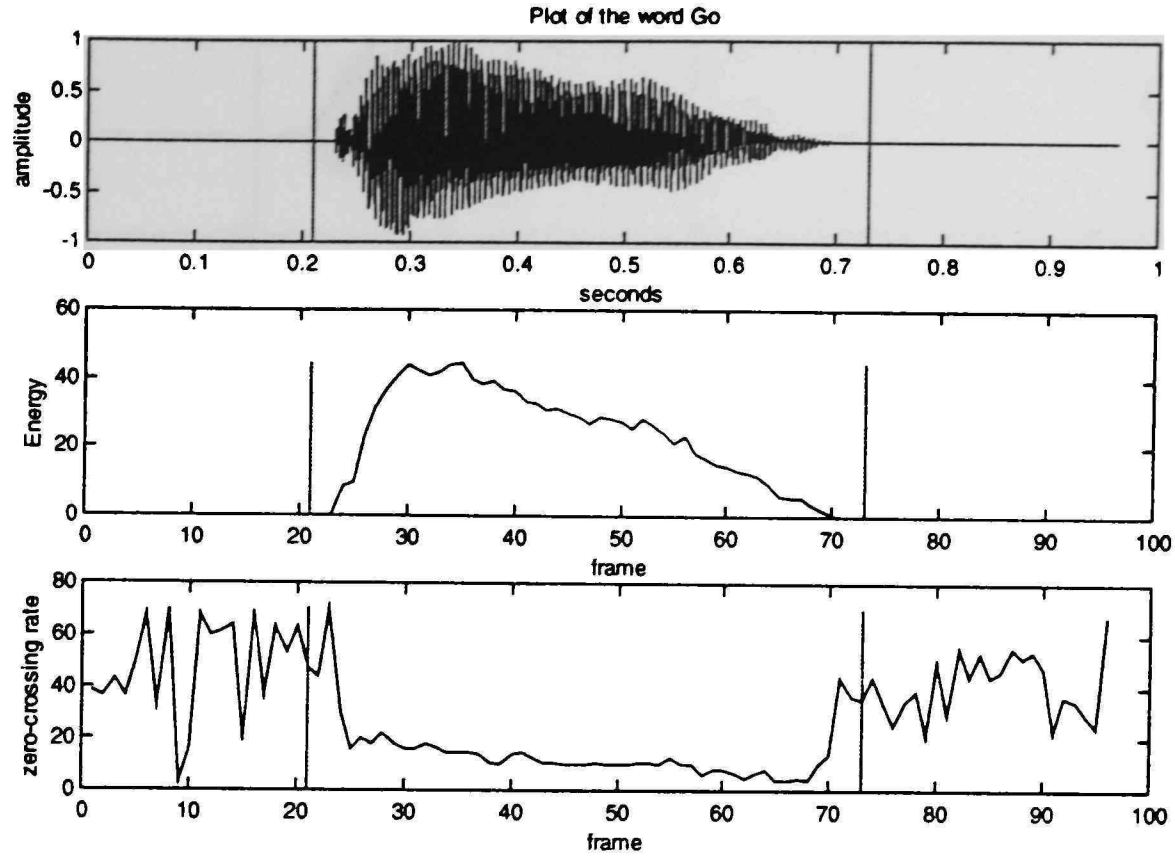

Figure A.3 Short time energy and zero crossing data for the word "Go" by a female speaker.

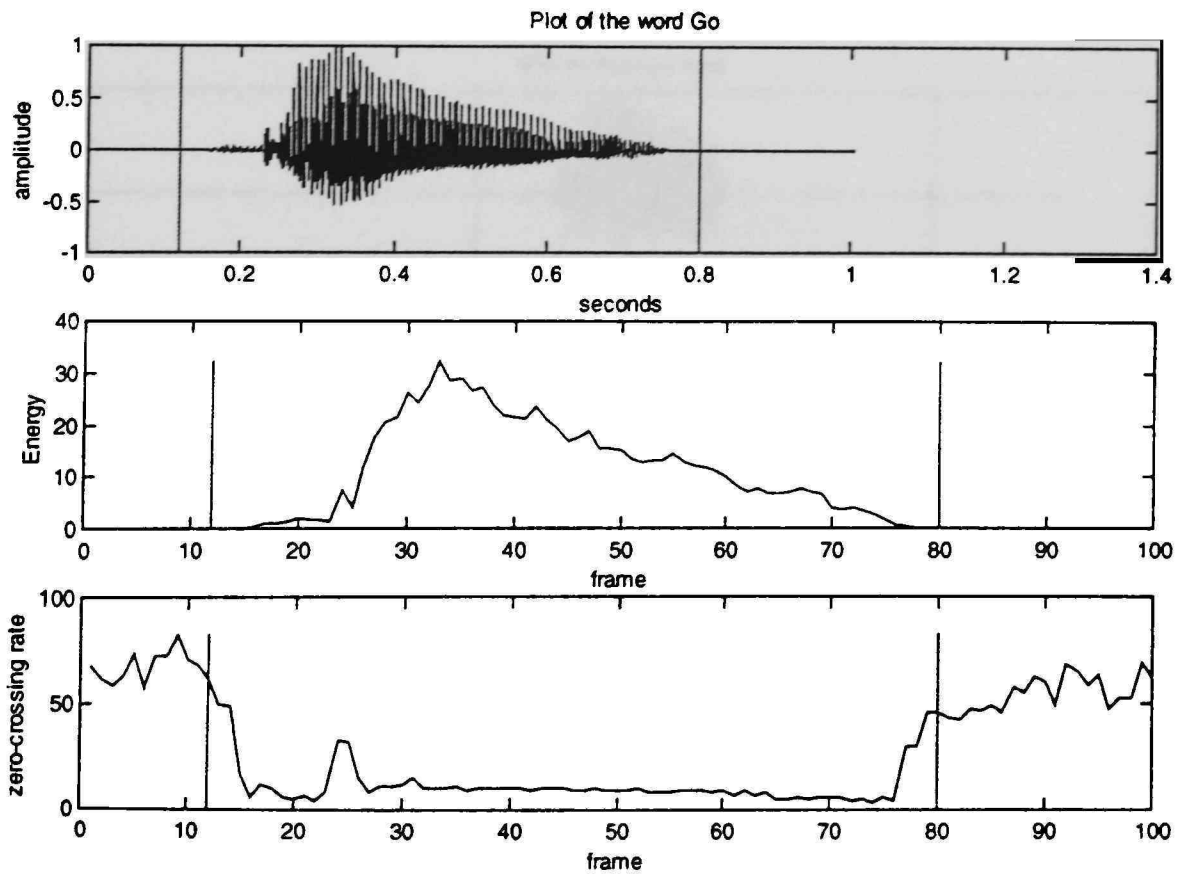

Figure A.4 Short time energy and zero crossing data for the word "Go" by a male speaker.

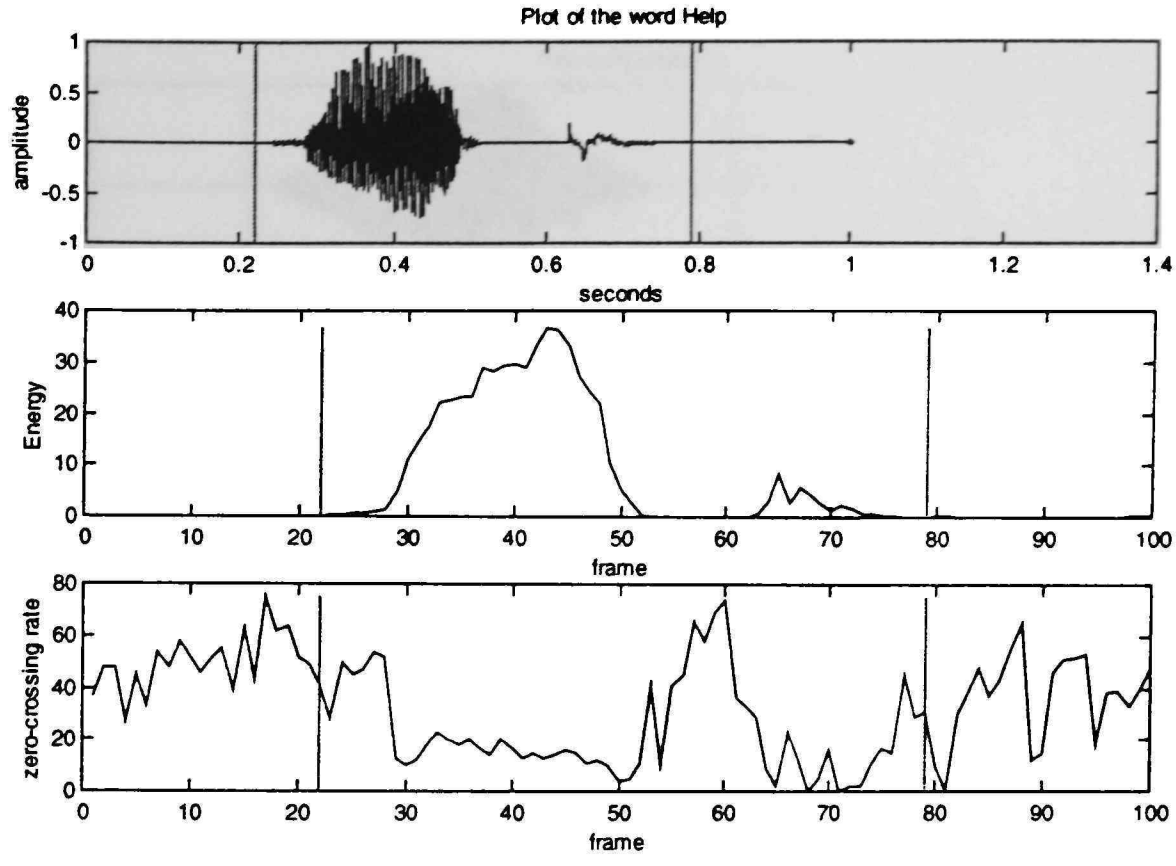

Figure A.5 Short time energy and zero crossing data for the word "Help" by a female speaker.

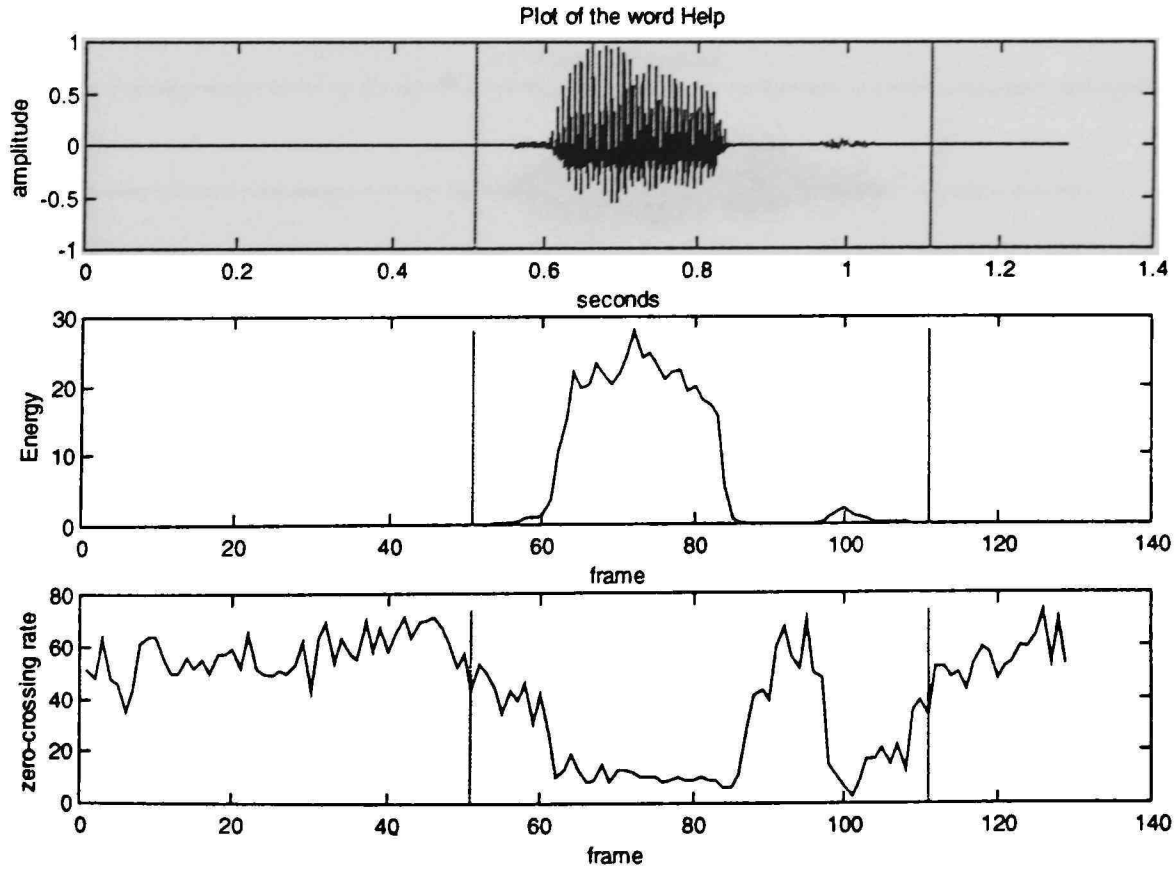

Figure A.6 Short time energy and zero crossing data for the word "Help" by a male speaker.

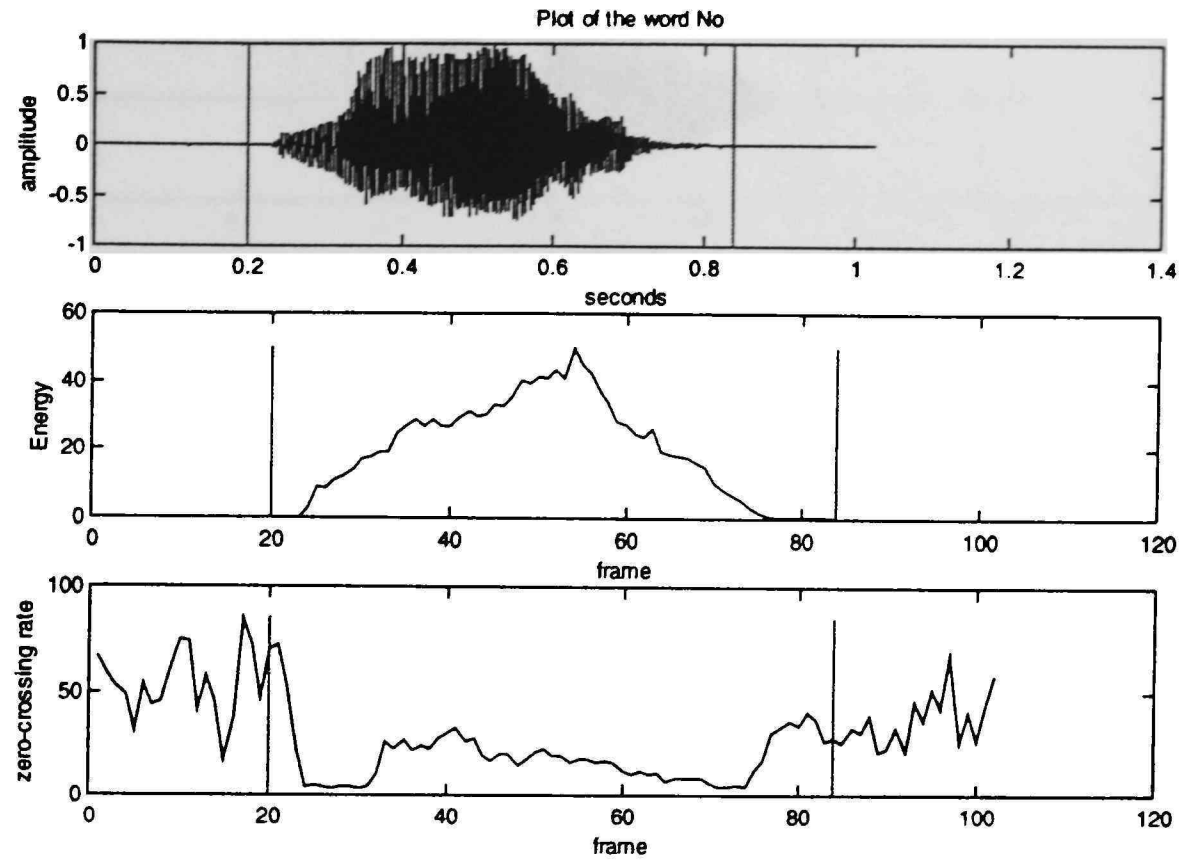

Figure A.7 Short time energy and zero crossing data for the word "No" by a female speaker.

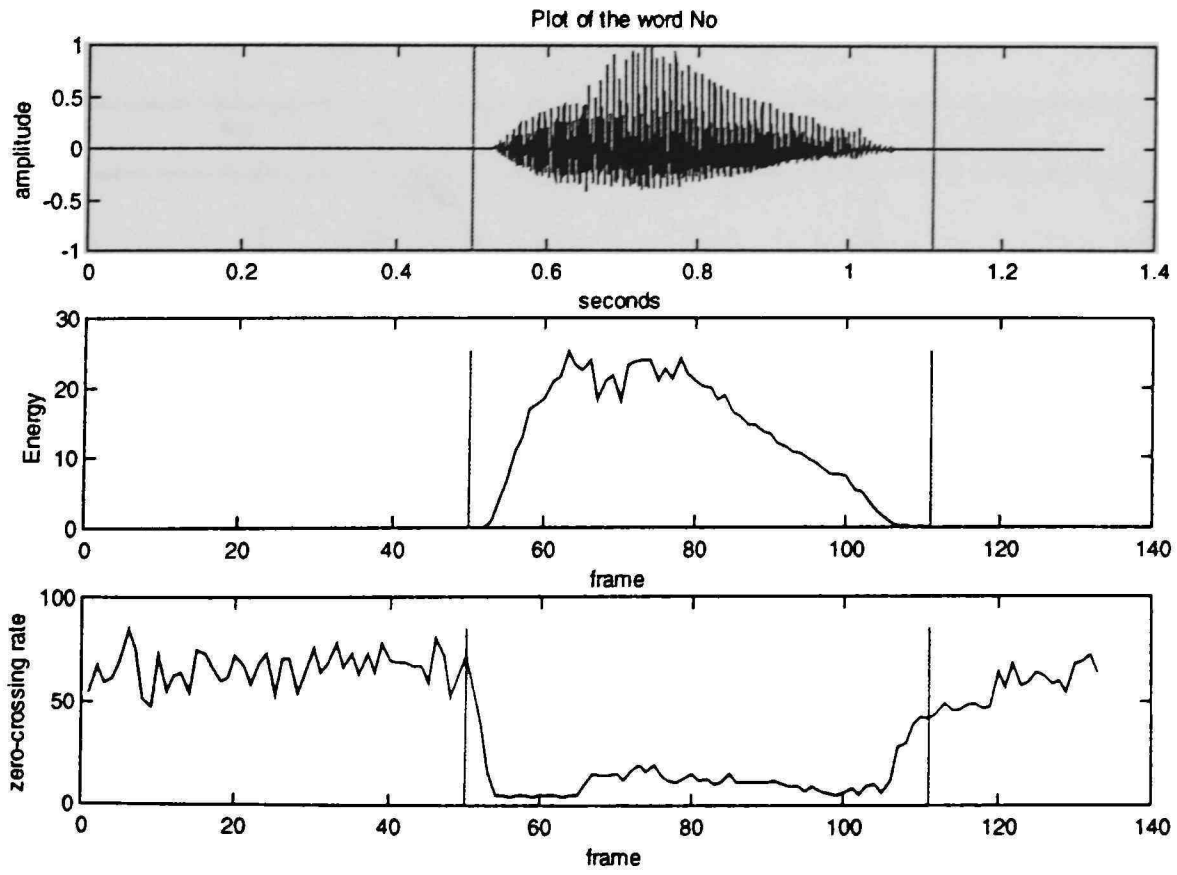

Figure A.8 Short time energy and zero crossing data for the word "No" by a male speaker.

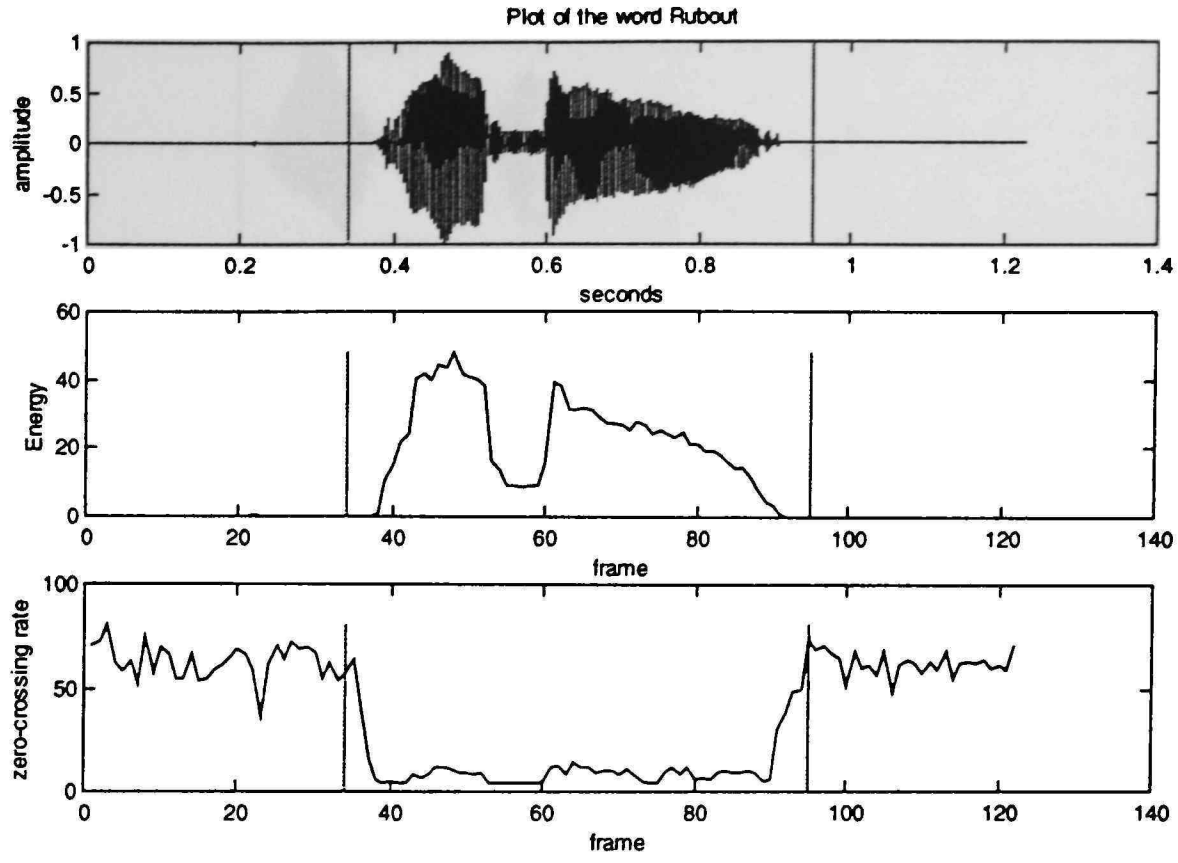

Figure A.9 Short time energy and zero crossing data for the word "Rubout" by a female speaker.

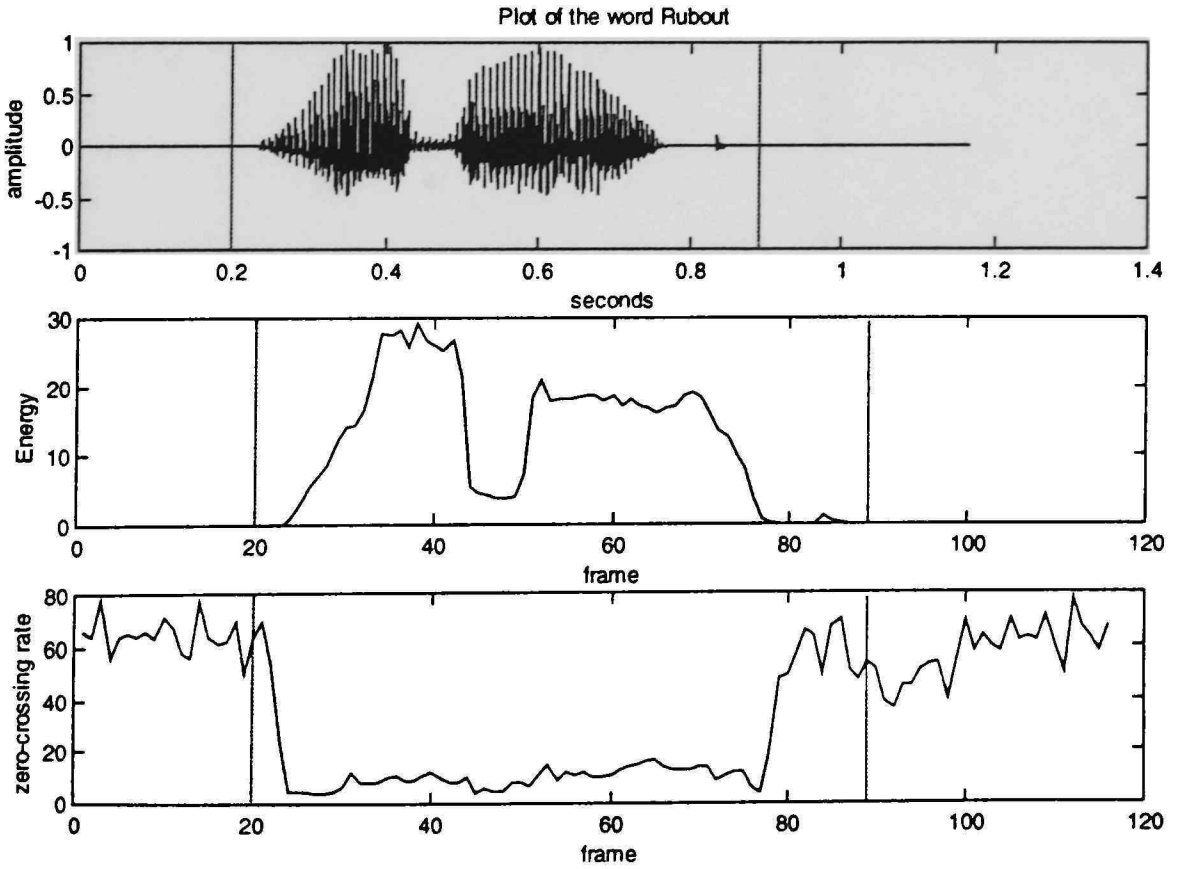

Figure A.10 Short time energy and zero crossing data for the word "Rubout" by a male speaker.

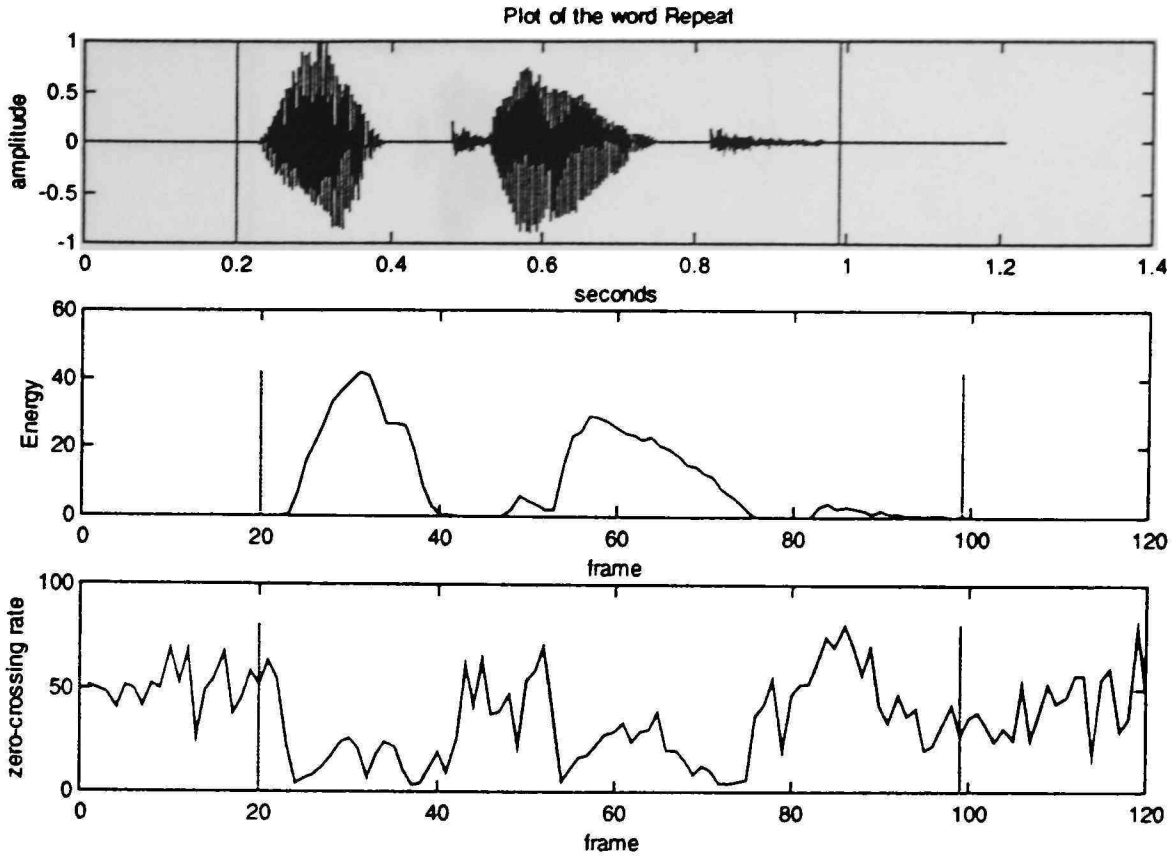

Figure A.11 Short time energy and zero crossing data for the word "Repeat" by a female speaker.

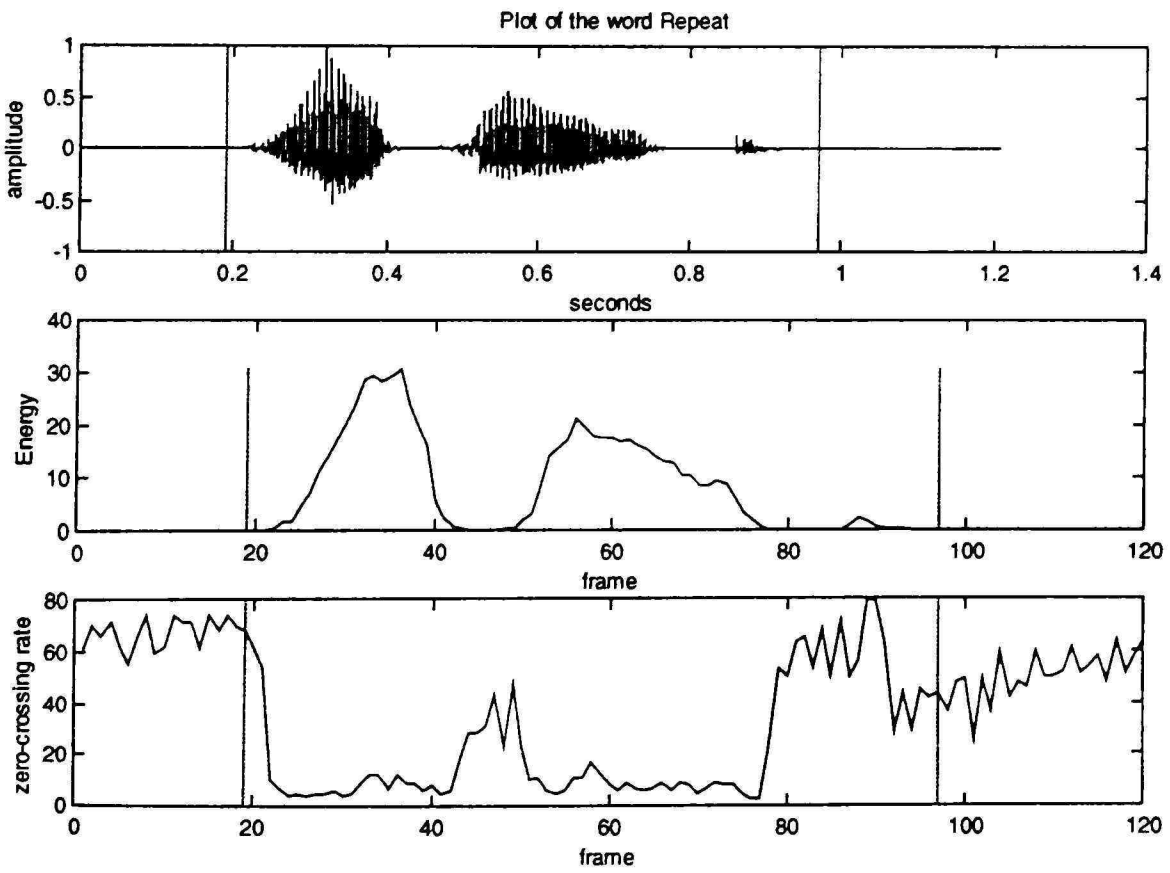

Figure A.12 Short time energy and zero crossing data for the word "Repeat" by a male speaker.

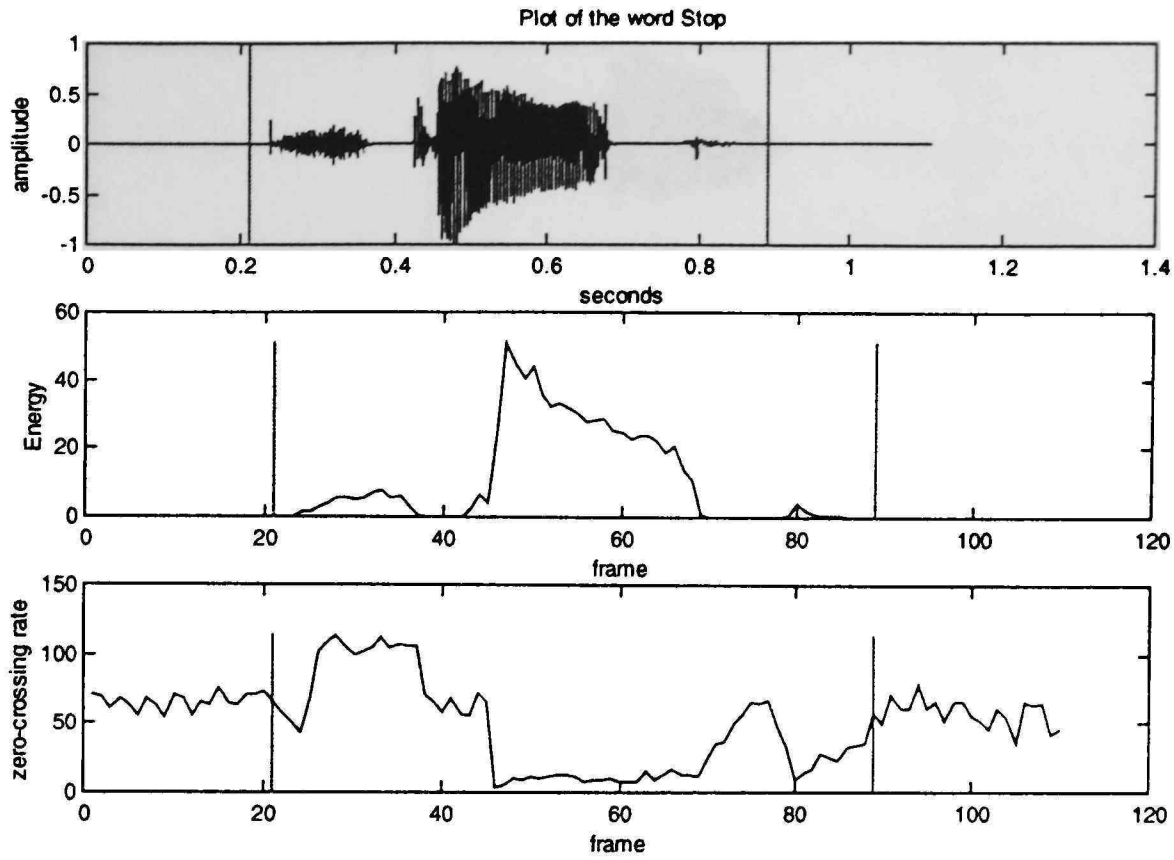

Figure A.13 Short time energy and zero crossing data for the word "Stop" by a female speaker.

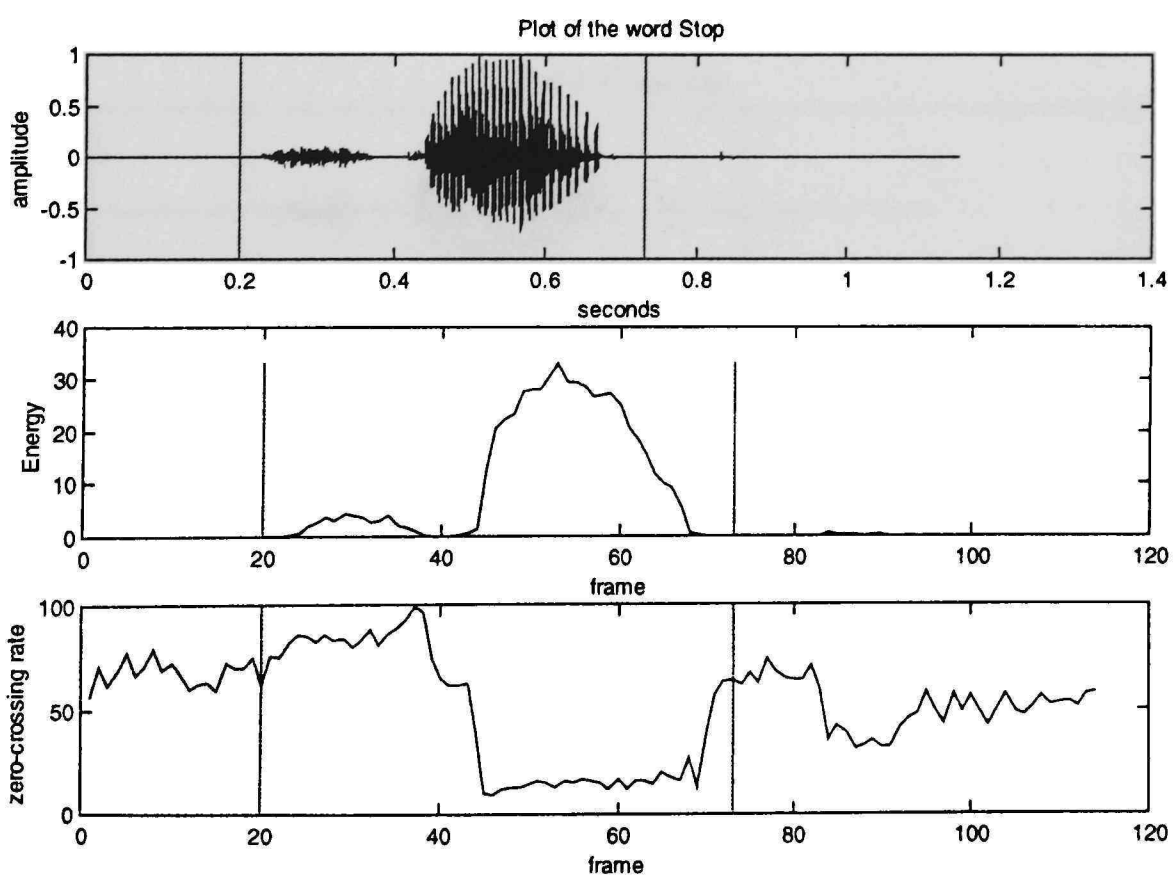

Figure A.14 Short time energy and zero crossing data for the word "Stop" by a male speaker.

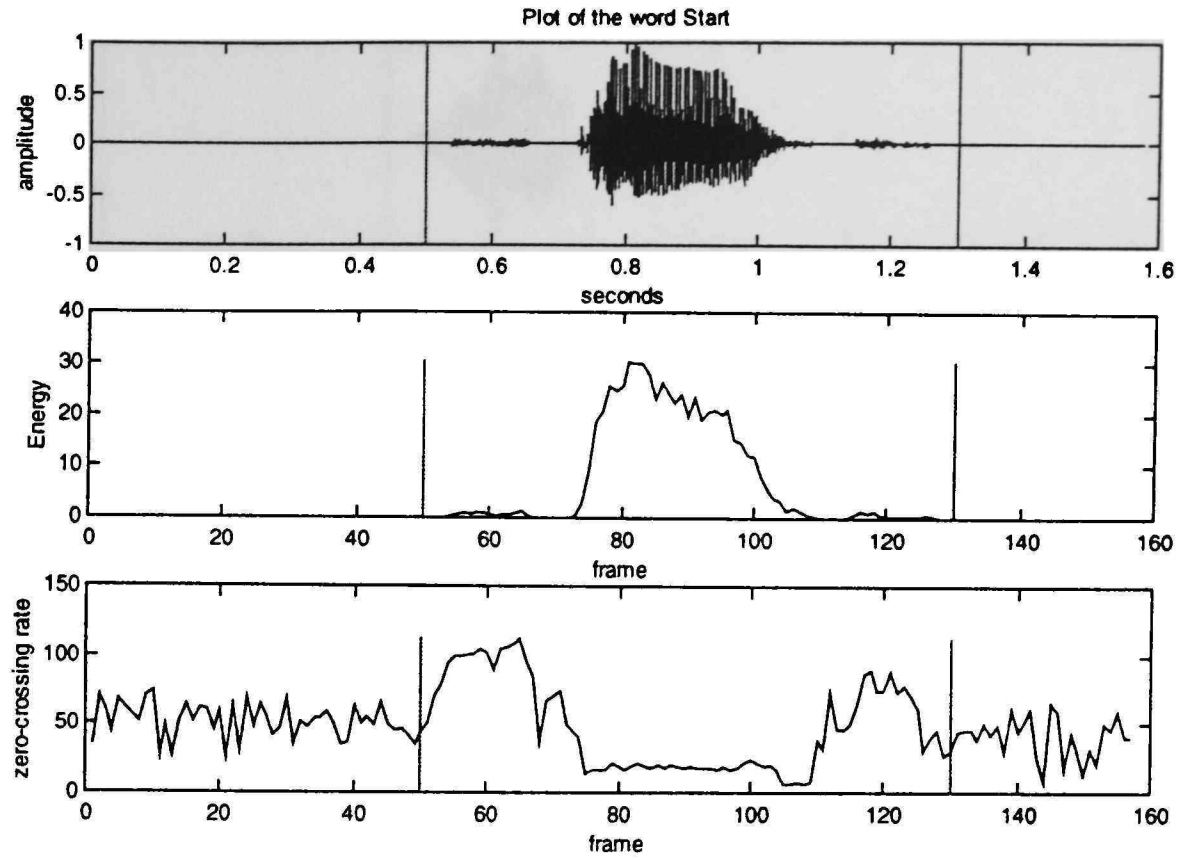

Figure A.15 Short time energy and zero crossing data for the word "Start" by a female speaker.

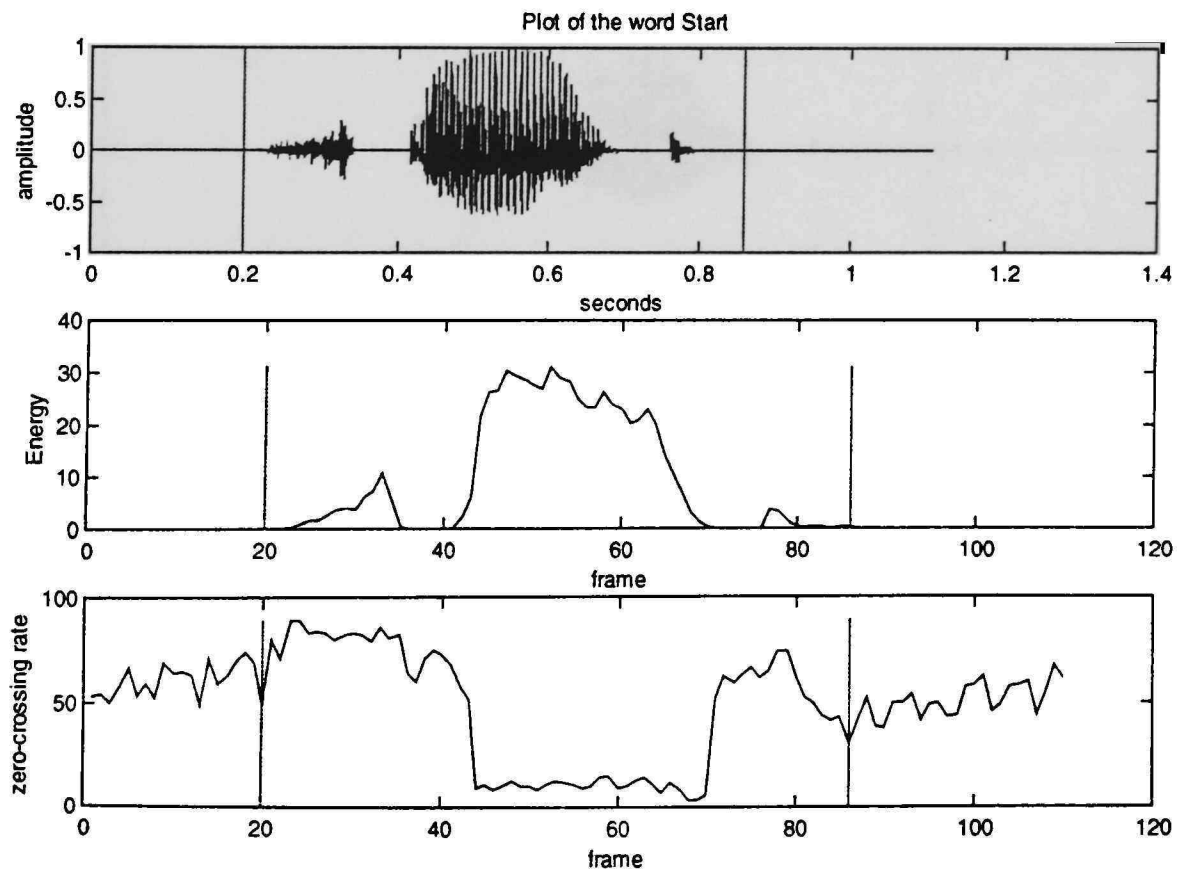

Figure A.16 Short time energy and zero crossing data for the word "Start" by a male speaker.

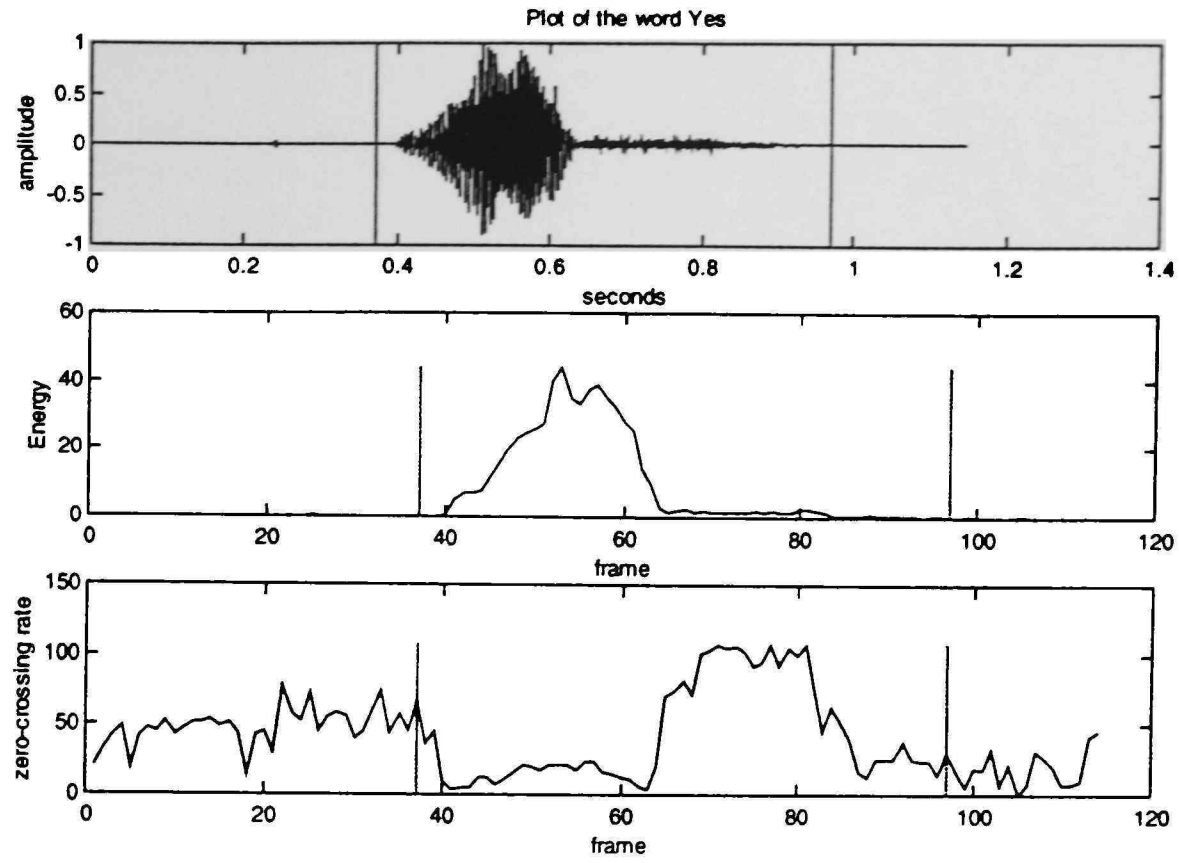

Figure A.17 Short time energy and zero crossing data for the word "Yes" by a female speaker.

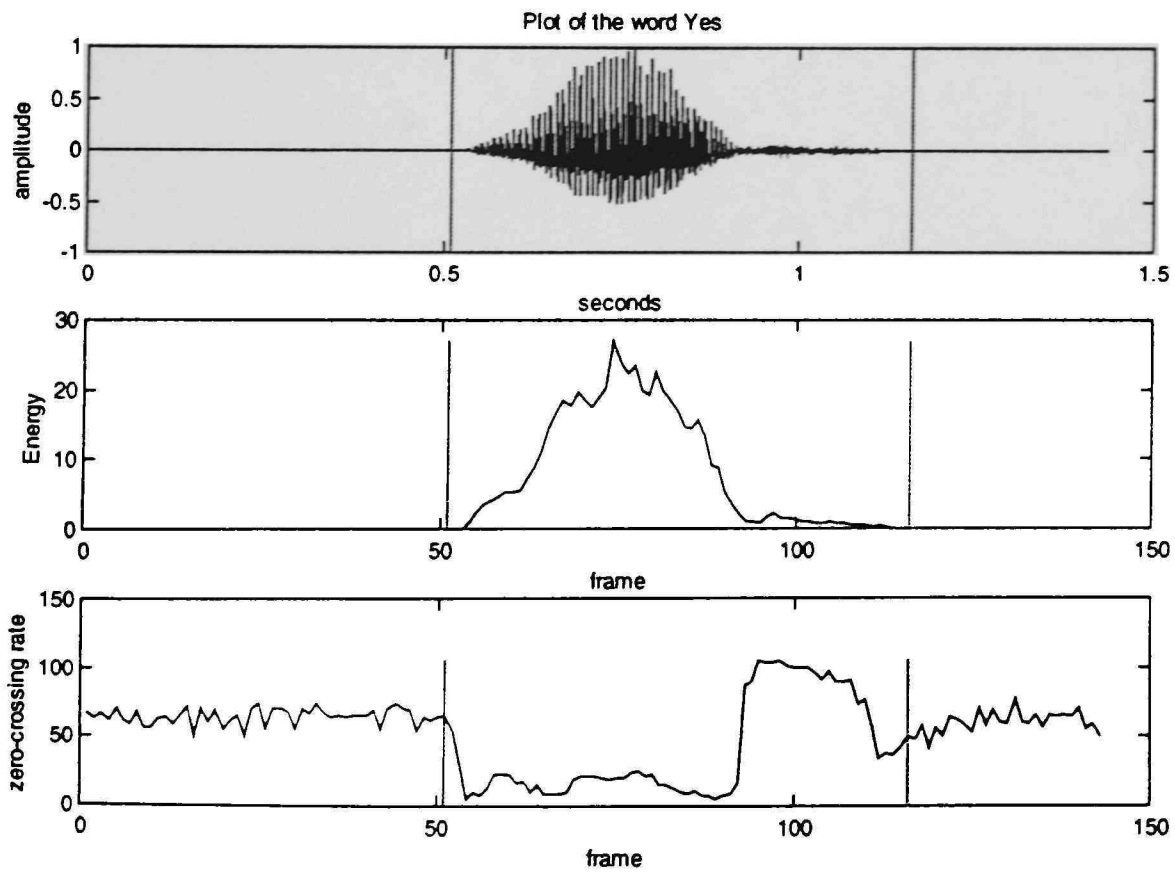

Figure A.18 Short time energy and zero crossing data for the word "Yes" by a female speaker.

APPENDIX B

MATLAB CODE FOR THE SPEECH RECOGNITION SYSTEM

MATLAB M-code for Feature Vector Generator.

```
% This M-file calculates the Feature vector of an utterance.
% The function accepts a Microsoft Windows WAV file as input
% and returns a feature vector of 20 dimensions.

function [ev] = genpara(file)
file=[file,'.wav'];

% Read wav file
a=wavread(file);
b=length(a);
d=max(abs(a));

% calculate total number of frames in data.
tf=floor(b/125);

% calculate Energy and Zero Crossing of the entire sample
tempID=[];
tempE=[];
for i=1:tf;
   E=0;
   ID=0;
   for j=(i-1)*125+2:(i-1)*125+125
      if (sign(a(j)) ~= sign(a(j-1)))
      ID = ID + 1;
      end;
      E=E+abs(a(j));
   end;
   tempE=[tempE, E];
   tempID=[tempID, ID];
end

% Calculate threshold levels ITL, ITU and IZCT

ID_TH = mean(tempID(1:10));
IF = 25;
IZCT=min(IF, ID_TH + 2*std(tempID(1:10)));
IMX = max(tempE);
IMN = mean(tempE(1:10));
I1 = 0.03*(IMX - IMN) + IMN;
I2 = 4*IMN;
ITL = min(I1, I2);
ITU = 5*ITL;

% Initial guess of Start point
m=1;
Em=0;
while (Em < ITU)
   Em=0;
   for j=(m-1)*125+1:(m-1)*125+125
      Em=Em + abs(a(j));
   end;
```

```
    m = m + 1;
end
m=m-1;
while (Em > ITL)
    Em=0;
    for j=(m-1)*125+1:(m-1)*125+125
        Em=Em + abs(a(j));
    end;
    m = m - 1;
end
m=m+1;
N1=m;
N1old=N1;

% Initial guess of end point
m=tf;
Em=0;
while (Em < ITU)
    Em=0;
    for j=(m-1)*125+1:(m-1)*125+125
        Em=Em + abs(a(j));
    end;
    m = m - 1;
end
m=m+1;
while (Em > ITL)
    Em=0;
    for j=(m-1)*125+1:(m-1)*125+125
        Em=Em + abs(a(j));
    end;
    m = m + 1;
end
m=m-1;
N2=m;
N2old=N2;

% compare end point with Zero Crossing to keep or change endpoints

if (N1<=25)
  k1=1;
else
  k1=N1-25;
end
M1=0;
Ntemp=0;
for i=N1:-1:k1
    ID=0;
    for j=(i-1)*125+2:(i-1)*125+125
        if (sign(a(j)) ~= sign(a(j-1)))
            ID = ID + 1;
        end;
    end
    if (ID>=IZCT)
      M1=M1+1;
```

```
      Ntemp=i;
    end
    if (M1>=3)
      break
    end
end
if (M1>=3)
  N1old=N1;
  N1=Ntemp;
end

M2=0;
Ntemp=0;
if (N2>=(tf-25))
  k2=tf;
else
  k2=N2+25;
end
for i=N2:k2
   ID=0;
   for j=(i-1)*125+2:(i-1)*125+125
      if (sign(a(j)) ~= sign(a(j-1)))
         ID = ID + 1;
      end;
   end
   if (ID>=IZCT)
     M2=M2+1;
     Ntemp=i;
   end
   if (M2>=3)
     break
   end
end
if (M2>=3)
  N2old=N2;
  N2=Ntemp;
end

% End points have been determined

start_fr=N1;
end_fr=N2;

% Calculate number of frames in each of the ten intervals
numinter=10;
interval_fr=round((end_fr-start_fr)/numinter);

%Calculate zero crossings in each interval

zc=[];
n=0;
for k=1:(numinter-1)
   n=0;
   for i=start_fr+(k-1)*interval_fr:start_fr+(k*interval_fr)-1
```

```
      ID=0;
      for j=(i-1)*125+2:(i-1)*125+125
            if (sign(a(j)) ~= sign(a(j-1)))
            ID = ID + 1;
         end;
      end
      n=n+1;
      zc(k,n)=ID;
   end
end
% Last interval may not contain enough frames as compared to the actual
% number of frames in each interval

zclast=[];
for i=start_fr+(numinter-1)*interval_fr:end_fr
   ID=0;
   for j=(i-1)*125+2:(i-1)*125+125
      if (sign(a(j)) ~= sign(a(j-1)))
               ID = ID + 1;
      end;
   end
   zclast=[zclast, ID];
end

% Caculate energy content in each interval
er=[];
n=0;
for k=1:(numinter-1)
   n=0;
   for i=start_fr+(k-1)*interval_fr:start_fr+(k*interval_fr)-1
      e=0;
      for j=(i-1)*125+1:(i-1)*125+125
         e=e+abs(a(j));
      end
      n=n+1;
      er(k,n)=e;
   end
end

% Last interval may not contain enough frames as compared to the actual
% number of frames in each interval
erlast=[];
for i=start_fr+(numinter-1)*interval_fr:end_fr
   e=0;
   for j=(i-1)*125+1:(i-1)*125+125
      e=e+abs(a(j));
   end
   n=n+1;
   erlast=[erlast, e];
end

evtempz=[];
evtempe=[];
```

```
% Find mean of zero crossings within the intervals

for i=1:(numinter-1)
  evtempz=[evtempz, mean(zc(i,:))];
end
evtempz=[evtempz,mean(zclast)];

% Find mean of Energy content within the intervals

for i=1:(numinter-1)
  evtempe=[evtempe, mean(er(i,:))];
end
evtempe=[evtempe,mean(erlast)];

% Final feature vector
ev=[evtempz evtempe];
```

MATLAB M-code for Reference Template Generator

```
% M-file for Reference Template Generator

clear all;
close all;

% Ask user for which words to recognize is an array
% for example if words number 2 5 7 and 9 need to be recognized
% the user will enter [2 5 7 9] as input

% The program then asks the user for number of utterances in each word

words=input('Enter which words to recognize: ');
numutter=input('Enter number of utterances for each word: ');
numword=length(words);

wordpara=[];

% The program calls the feature vector generator for each utterance of
% each word to be recognized
for i=1:numword
   for j=1:numutter
      t1=num2str(words(i));
      t2=num2str(j);
      wordfile=strcat('wd',t1,'s',t2);
      disp(sprintf('Word %s is being processed',wordfile))
      wordpara(i,j,:)=genpara(wordfile);
   end
end
[x,y,numpara]=size(wordpara);


% The mean of all utterances of a given word is calculated
wordmean=[];
for i=1:numword
   for j=1:numpara
     wordmean(i,j)=mean(wordpara(i,:,j));
   end
end

% The standard deviation of all utterances of a given word is
calculated.
wordstd=[];
for i=1:numword
   for j=1:numpara
     wordstd(i,j)=std(wordpara(i,:,j));
   end
end

% A new vector is created that is one standard deviation away from the
% mean. This serves the purpose of "radius of confidence" for that
% word.
```

```
wordp1s=[];
for i=1:numword
  for j=1:numpara
     wordp1s(i,j)=wordmean(i,j)+wordstd(i,j);
  end
end

% Here the Euclidean distance between the mean and the word that is one
% standard deviation away is calculated. This is the radius of
% confidence.

wvreg=[];
for k=1:numword
  D1=0;
  for l=1:numpara
    D1=D1+(wordmean(k,l)-wordp1s(k,l))^2;
  end
  wvreg=[wvreg, sqrt(D1)];
end

% save all calculated parameters in a datafile. This is now the
% reference template.
save datafile.mat words wordpara wordmean wvreg
```

MATLAB M-file for Speech Recognition module.

```
% M-file for the speech recognition system

% Load the reference template into memory

load datafile.mat

% obtain parameters from reference template
[numword,numutter,numpara]=size(wordpara);

% Ask user for an input utterance to recognize
testfile=input('input test file to recognize: ','s');

results=[];
correct=0;

% Calculate feature vector for the unknown utterance
wordev=genpara(testfile);

% Calculate Euclidean distance between the unknown utterance
% and all the words in the reference template.

wv=[];
for k=1:numword
   D=0;
   for l=1:numpara
      D=D+(wordmean(k,l)-wordev(l))^2;
   end
   wv=[wv, sqrt(D)];
end

% Find the word with minimum distance
[wdist, wnum]=min(wv);

% check to see if the distance is within the words radius of confidence
% if it is then thats the word otherwise report not recognized.
if wdist < wvreg(wnum)
  disp(sprintf('The utterance was identified as %d',words(wnum)))
else
  disp(sprintf('The word was not recognized'))
end
```

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University or Texas Tech University Health Sciences Center, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Agree (Permission is granted.)

Student's Signature _____ Date _____

Disagree (Permission is not granted.)

Student's Signature _____ Date _____