SPEECH SYSTEM FOR A VOICE-IMPAIRED PERSON

by

RAVI KUMAR ANJANI SIRIGINEEDI, B.Tech.

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

December, 1999

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude and appreciation to Dr. Micheal Parten, my graduate advisor, for his guidance, valuable suggestions and encouragement. I am grateful to Dr. Sunanda Mitra and Dr. Donald Bagert for serving as members of my thesis committee. I specially thank Dr. Donald Wunsch for fostering my curiosity and my interest in neural networks and for providing a wealth of interesting conversation and reasoned advice. I would also like to thank the Department of Electrical Engineering and Dr. Donald Wunsch (again) 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 girlfriend for their unending support, encouragement, and guidance. It is to my family to whom I wish to dedicate this thesis.

TABLE OF CONTENTS

ABSTRACT

This thesis attempts to develop a speaker-dependent speech system for voice-impaired people. The system recognizes isolated utterances from a limited vocabulary, and is small and cost-efficient enough to be incorporated into a hand-held system. A 20-dimensional feature vector was generated based on zero crossing and energy content measurements of the speech waveforms. The generated feature vectors were used to train a neural network and the trained network was tested with known and unknown utterances. The system was implemented on an IBM Personal Computer and achieved a recognition rate of 76% on a ten-word database of 16 speakers (8 male and 8 female). A test database, which mimics a voice-impaired person's speech, was developed, and a recognition rate of 60% was observed. The system recognized utterances at an average rate of 0.15 seconds/recognition.

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Speech is a natural mode of communication for people. They learn all the relevant skills during their early childhood, without instruction, and they continue to rely on speech communication throughout their lives. It comes so naturally to them, that they do not realize how complex a phenomenon speech is. The human vocal tract and articulators are biological organs with nonlinear properties, whose operation is not just under conscious control but also affected by factors ranging from gender, upbringing, to emotional state. As a result, vocalizations can vary widely in terms of their accent, pronunciation, articulation, roughness, nasality, pitch, volume, and speed; moreover, during transmission, their irregular speech patterns can be further distorted by background noise and echoes, as well as electrical characteristics (if telephones or other electronic equipment are used). All these sources of variability make speech recognition, a very complex problem, even more than speech generation.

Yet people are so comfortable with speech that they would also like to interact with computers via speech, rather than having to resort to primitive interfaces such as keyboards and pointing devices. A speech interface would support many valuable applications, for example, telephone directory assistance, spoken database querying for novice users, "hands-busy" applications in medicine or fieldwork, office dictation devices, or even automatic voice translation into foreign languages. Such tantalizing

applications have motivated research in automatic speech recognition since the 1950's. Great progress has been made so far, especially since the 1970's, using a series of engineered approaches that include template matching, knowledge engineering, and statistical modeling. Yet computers are still nowhere near the levels of human performance at speech recognition, and it appears that further significant advances will require some new insights.

People, undoubtedly are much better at recognizing speech, the human brain is known to be wired differently than a conventional computer; in fact it operates under a radically different computational paradigm. While conventional computers use a very fast and complex central processor with explicit program instructions and locally addressable memory, by contrast the human brain uses a massively parallel collection of slow and simple processing elements (neurons), densely connected by weights (synapses) whose strengths are modified with experience, directly supporting the integration of multiple constraints, and providing a distributed form of associative memory.

The brain's impressive superiority at a wide range of cognitive skills, including speech recognition, has motivated research into its novel computational paradigm since the 1940's, on the assumption that brain like models may ultimately lead to brain like performance on many complex tasks. This fascinating research area is now known as connectionism, or the study of artificial neural networks. The history of this field has been erratic (and laced with hyperbole), but by the mid-1980's, the field had matured to a point where it became realistic to begin applying connectionist models to difficult tasks like speech recognition. By 1990, many researchers had demonstrated the value of neural

networks for important subtasks like phoneme recognition and spoken digit recognition, but it was still unclear whether connectionist techniques would scale up to large speech recognition tasks. This thesis demonstrates that neural networks can indeed form the basis for speaker-dependent speech recognition system, and that neural networks offer some clear advantages over conventional techniques.

1.2 Speech Recognition

The current state of the art in speech recognition is a complex issue, because a system's accuracy depends on the conditions under which it is evaluated. Under sufficiently narrow conditions almost any system can attain humanlike accuracy, but it's much harder to achieve good accuracy under general conditions. The conditions of evaluation, and hence the accuracy of any system can vary along the following dimensions.

- **Vocabulary size and confusability.** As a general rule, it is easy to discriminate among a small set of words, but error rates naturally increase as the vocabulary size grows. For example, the 10 digits "zero" to "nine" can be recognized essentially perfectly [1], but vocabulary sizes of 200, 5,000, or 100,000 may have error rates of 3%, 7%, or 45% [2, 3, 4]. On the other hand, even a small vocabulary can be hard to recognize if it contains confusable words. For example, the 26 letters of the English alphabet (treated as 26 "words") are very difficult to discriminate because they contain so many confusable words (most notoriously, the E-set: "B, C, D, E, G, P, T, V, Z"); an 8% error rate is considered good for this vocabulary [5].

- **Speaker dependence versus independence.** By definition, a speaker dependent system is intended for use by a single speaker, but a speaker independent system is intended for use by any speaker. Speaker independence is difficult to achieve because a system's parameters become tuned to the speaker(s) that it was trained on, and these parameters tend to be highly speaker-specific. Error rates are typically 3 to 5 times higher for speaker independent systems than for speaker dependent ones [6]. Intermediate between speaker dependent and independent systems, there are also multi-speaker systems intended for use by a small group of people, and speaker-adaptive systems which tune themselves to any speaker given a small amount of their speech as enrollment data.
- **Isolated, discontinuous, or continuous speech.** Isolated speech means single words; discontinuous speech means full sentences in which words are artificially separated by silence; and continuous speech means naturally spoken sentences. Isolated and discontinuous speech recognition is relatively easy because word boundaries are detectable and the words tend to be cleanly pronounced. Continuous speech is more difficult, however, because word boundaries are unclear and their pronunciations are more corrupted by co-articulation, or the slurring of speech sounds, which for example causes a phrase like "could you" to sound like "could jou." In a typical evaluation, the word error rates for isolated and continuous speech were 3% and 9%, respectively [7].
- **Task and language constraints.** Even with a fixed vocabulary, performance will vary with the nature of constraints on the word sequences that are allowed during

recognition. Some constraints may be task-dependent, (for example, an airline-querying application may dismiss the hypothesis "The apple is red"); other constraints may be semantic (rejecting "The apple is angry"), or syntactic (rejecting "Red is apple the"). Constraints are often represented by a grammar, which ideally filters out unreasonable sentences so that the speech recognizer evaluates only plausible sentences. Grammars are usually rated by their perplexity, a number that indicates the grammar's average branching factor (i.e., the number of words that can follow any given word). The difficulty of a task is more reliably measured by its perplexity than by its vocabulary size.

- **Read versus spontaneous speech.** Systems can be evaluated on speech that is either read from prepared scripts, or speech that is uttered spontaneously. Spontaneous speech is vastly more difficult, because it tends to be peppered with disfluencies like "uh" and "um", false starts, incomplete sentences, stuttering, coughing, and laughter; and moreover, the vocabulary is essentially unlimited, so the system must be able to deal intelligently with unknown words (e.g., detecting and flagging their presence, and adding them to the vocabulary, which may require some interaction with the user).
- **Adverse conditions.** A range of adverse condition [8] can also degrade a system's performance. These include environmental noise (e.g., noise in a car or a factory); acoustical distortions (e.g., echoes, room acoustics); different microphones (e.g., close-speaking, omnidirectional, or telephone); limited frequency bandwidth (in

telephone transmission); and altered speaking manner (shouting, whining, speaking quickly, etc.).

In each of the above cases, 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.

1.3 Research Trends

The earliest speech recognition systems were attempted in the early 1950s. In 1952, at Bell Laboratories, Davis, Biddulph and Balashek developed an isolated digit recognition system for a single speaker [9]. The system worked on measuring spectral resonance 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 Suziki and Nakata [10]. 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 of different sections of the speech to recognize the phoneme [11]. Perhaps the most notable attempt came from another Japanese group led by Nagata at NEC Laboratories in 1963 [12]. 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 non-uniformity of time scales in speech events [13]. The Russian effort led by Vintsyuk developed the dynamic programming methods for time aligning a pair of speech utterance [14]. 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 [15].

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) [16, 17, 18]. 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 [19].

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 systems [20, 21].

The 1980s also saw a 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 [22].

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 [23]. The software runs on an IBM personal computer and boasts a vocabulary of 60,000 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 96MB of random access memory (RAM), a Pentium 133MHz processor and 55MB of hard disk space. Although the software is speaker dependent 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 to 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 requirement is 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.

All the systems that have been discussed above 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 a 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 50MHz and also include 2MB of dedicated memory. The host operating systems supported are Windows NT and multiple flavors of UNIX. The speech recognition system supports both speaker independent and speaker dependent 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 Smart Speak by Advanced Recognition Technologies. This is actually a software system intended to be developed on micro-controllers 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.

1.4 Statement of the Problem

The decade of the 1990s is already coming to an end and now is the time to address the issue of priorities and to set tangible and feasible goals for human beings' entry into the 21st century. Voice will be an important and vital technology and one, which will clearly be of assistance to disabled and non-disabled individuals alike. Better algorithms are being developed to predict speech output for people with vocal

dysfunction thereby encouraging more discourse and facilitating communication. An early example of this was the Bliss system developed by Carlson et al. and Hunnicutt [24], which contained about 500 symbols that could be translated into common words and longer utterances. This system was helpful in encouraging individuals with severe speech and language impairments to use speech synthesis.

The aim of this thesis is to develop a completely speaker dependent speech system with the help of a neural network architecture. At later stages, this system can be hardware developed using a common DSP with a small amount of directed memory and an analog-to-digital converter that can be incorporated as a compact hand-held system for the voice impaired individuals. A simple Endpoint detection algorithm to be used for the preprocessing of speech signals is investigated. A supervisory-learning, multi-layer feed-forward back propagation (FFBP) neural network is examined and its performance on the specific task of speech recognition is assessed.

Chapter II provides a basic introduction to neural networks and an overview of their development, theory and training. FFBP architecture and its advantages are outlined in detail. In Chapter III, the use of Endpoint detection algorithms at the input to a neural network is discussed along with several other speech recognition algorithms. Chapter IV describes the entire system that has been developed and explains the implementation of that system. The results of system evaluated 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 II

NEURAL NETWORKS FOR SPEECH RECOGNITION

2.1 Introduction

Neural networks can solve problems that conventional methods cannot, or at least not within acceptable cost or performance criteria. Unlike conventional computing techniques, which handle problems through mathematical models and algorithms, neural network techniques require little knowledge of the system as they are taught by example. They are trained, not programmed. Neural networks are very good at pattern recognition, pattern matching and classification tasks, they are suitable to solve non-linear or ill-defined problems, and very well adapted to process noisy or corrupted input data.

A brief history of the development of neural networks and a basic introduction to their theory is outlined in this chapter. Feed-forward backpropagation (FFBP) architecture is described in detail.

2.2 The Perceptron

The idea of the simple neuron model first emerged in the 1940s with the work of McCulloch and Pitts [27]. The cybernetics movement that ensued attempted to combine biology, psychology, engineering and mathematics resulting in architectures for networks of neurons, which would perform a number of tasks. In 1949, Hebb's book [28] put forward the theory of neural networks developing "internal representations" related to experience.

In the 1950s, research continued initially into the development of networks to perform specific tasks but this changed and the goal became to develop machines that could learn. By the end of that decade, there had been a lack of significant developments and work in this field diminished considerably.

In the 1960s, interest was revived with the publication of a book by Rosenblatt [29] where he defined the concept of the perceptrons and laid down many theories about them. In their simplest form, these processing elements, also known as, nodes or artificial neurons, have the structure illustrated in Figure 2.1. It was proved theoretically that a perceptron could learn to perform any task as long as it is possible to program it, to do so.

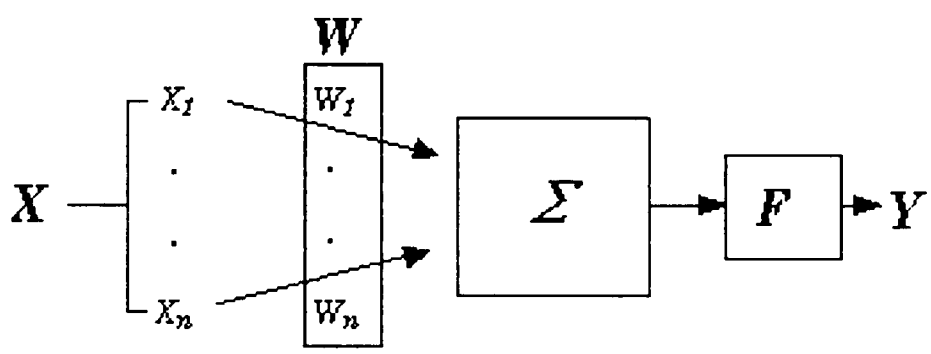

Fig 2.1 The general structure of a Perceptron

A set of inputs (X_1 to X_n) is applied to each node representing the inputs from the outside world or, alternatively, they may be outputs from other nodes. Each of the input is multiplied by a weight (W_1 to W_n) associated with the node input to which it is connected and the weighted inputs are then summed together.

A threshold value C, local for each node is added to the weighted summation and

the resulting sum is then passed through a hard limiting activation function F. The output of a node is therefore

$$Y = F\{\sum_{n=1}^{N}(X_n * W_n) + C\}.$$

The perceptron effectively splits the input patterns into two distinct regions with one region being represented by a 1 on the output and the other a 0. Rosenblatt's training algorithm for the perceptron would converge if the input patterns to the perceptron were linearly separable. The perceptron would therefore approximate the decision boundary between the two classes of outputs.

Perceptrons were successfully trained to perform certain tasks but there were failures that could not be overcome. Minsky and Papert pointed out the serious shortcomings of perceptrons [30] and interest in the study of neural networks again declined.

The Exclusive-Or (Ex-Or) function is a major illustration of the limitation of perceptrons. For the ex-or function an output of 1 is generated if the inputs are {0,1} or {1,0} and an output of 0 is generated if the inputs are {0,0} or {1,1}. This is not a linearly separable function so the perceptron cannot learn it. A more complicated decision surface is required here and it was found that a curved decision surface is required to separate the two classes of inputs.

2.3 The Multi-Layer Perceptron

Minsky and Papert had proposed a solution to the problem posed by functions such as the ex-or. They suggested that an extra layer of nodes with non-linear activation functions could be introduced. The output would now be a non-linear combination of the inputs so more complicated decision surfaces could be represented. The problem that remained was that no training algorithm was available to train such a network of perceptrons at the time.

During the 1970s, more research turned towards the representation of knowledge and away from learning and many new ideas were developed. Then, in the 1980s, there was a resurgence of interest in neural networks and it was during this time that an effective algorithm, called backpropagation, for the training of multi-layer perceptron (MLP) structures was developed [31].

2.3.1 The Feed-Forward Backpropagation Algorithm

Feed-forward backpropagation [48] is a gradient descent algorithm where weights and biases are adjusted to minimize a cost function equal to the mean square error in the network. This network consists of a number of layers of neurons. Signals travel only from the input layer through one or more hidden layers to the output layer neurons. Only during the training session, the error information is passed back to the network in the form of modifications to the weight factors. When the training is done the weight factors are fixed and the network behavior does not change anymore, until it is re-trained.

For a 3-layer neural network with N input nodes and M output nodes, the network's weights are initially set to small random values. An input/output vector pair *p* is presented to the network with input vector

$$x_{p0}, x_{p1}, \ldots, x_{pN-1}$$

and target output vector

$$t_0, t_1, \ldots, t_{M-1}.$$

From this input vector, an output vector is produced by the network, which can then be compared to the target output vector. If there is no difference between the produced and target output vectors no learning takes place. Otherwise the weights are changed to reduce the difference. The weights are adapted using a recursive algorithm which starts at the output nodes and works back to the hidden layer. The error in the network when training pair *p* is presented is defined as

$$E_p = \frac{1}{2}\sum (t_{pj} - o_{pj})^2 \qquad (2.1)$$

where:

t_{pj} is the target value for the j^{th} element of the output pattern from the training pair *p*.

o_{pj} is the actual value produced by the network for the j^{th} element of the output pattern when the input pattern from the training pair *p* is presented to its input.

The overall error is therefore

$$E = \sum_p E_p \,. \tag{2.2}$$

The input to node j is

$$net_{pj} = \sum_i w_{ji} o_{pi} \tag{2.3}$$

where:

$w_{ji}(t)$ is the weight from the i^{th} node of the previous layer to the j^{th} node at time t when the input/output pair p is presented to the network

o_{pi} is the output of the i^{th} node of the previous layer.

A non-linear activation function is employed in each node such that the output of node j is

$$o_{pj} = f_j(net_{pj}) \,. \tag{2.4}$$

To implement a gradient descent the negative derivative of Ep with respect to w_{ij} must be proportional to the change in the weight w_{ij}, $\Delta_p w_{ji}$. Therefore,

$$\Delta_p w_{ji} \propto -\frac{\delta E_p}{\delta w_{ji}} . \tag{2.5}$$

Applying the chain rule to (2.5) gives

$$\frac{\delta E_p}{\delta w_{ji}} = \frac{\delta E_p}{\delta net_{pj}} \frac{\delta net_{pj}}{\delta w_{ji}} . \tag{2.6}$$

From (2.3), it can be seen that

$$\frac{\delta net_{pj}}{\delta w_{ji}} = \frac{\delta}{\delta w_{ji}} \sum_i w_{ji} o_{pi} = o_{pi} \,. \tag{2.7}$$

Applying the chain rule to (2.6) gives

$$\frac{\delta E_p}{\delta net_{pj}} = \frac{\delta E_p}{\delta o_{pj}} \frac{\delta o_{pj}}{\delta net_{pj}} . \qquad (2.8)$$

From (2.1), it can be seen that

$$\frac{\delta E_p}{\delta o_{pj}} = \frac{\delta}{\delta o_{pj}} \frac{1}{2} \sum (t_{pj} - o_{pj})^2 = -(t_{pj} - o_{pj}) = -\delta_{pj} . \qquad (2.9)$$

From (2.4), it can be seen that

$$\frac{\delta o_{pj}}{\delta net_{pj}} = f_j^{'}(net_{pj}) . \qquad (2.10)$$

Substituting (2.7), (2.9) and (2.10) into (2.6) gives

$$-\frac{\delta E_p}{\delta w_{ji}} = -(-\delta_{pj}) f_j^{'}(net_{pj}) o_{pi} = \delta_{pj} f_j^{'}(net_{pj}) o_{pi} . \qquad (2.11)$$

As mentioned earlier, the negative derivative of E_p with respect to w_{ij} must be proportional to the change in the weight w_{ij}, D_p to implement a gradient descent. Therefore,

$$\Delta_p w_{ji} \alpha \delta_{pj} f_j^{'}(net_{pj}) o_{pi} . \qquad (2.12)$$

Let,

$$\Delta_p w_{ji} = \eta \delta_{pj} f_j^{'}(net_{pj}) o_{pi} \qquad (2.13)$$

where η is a small constant known as the learning rate.

Then,

$$w_{ji}(t+1) = w_{ji}(t) + \eta \delta_{pj} f_j^{'}(net_{pj}) o_p \qquad (2.14)$$

where $w_{ji}(t+1)$ is the weight from the i^{th} node to the j^{th} after adjustment.

Equation (2.14) is known as the standard delta rule and defines how weights are changed after the presentation of a training pair. The activation function must be non-

linear since, otherwise, the neural network would perform a linear transformation at each layer and could therefore be reduced to its equivalent single layer network. The effectiveness of the extra layer of perceptrons is then lost. The activation must also be differentiable as required by equation (2.11). The sigmoid function is the one most often used since it meets all the requirements.

2.3.2 The Fully Connected Neural Network

The most common form of neural network is the 3-layer, fully connected, feed forward MLP (multi-layer perceptron), an example of which is shown in Figure 2.2. The nodes are arranged in 3 layers: an input layer, a hidden layer and an output layer with inputs flowing in the forward direction from input layer to output layer through the hidden layer except during training. In this type of network, the inputs of every node in the hidden layer are connected to the outputs of every node in the input layer and the inputs of every node in the output layer are connected to the outputs of every node in the hidden layer. The nodes in the input layer are used to monitor the external signals input to the neural network and the neurons in the output layer are used to make the final decision and transmit the signal produced to the outside world.

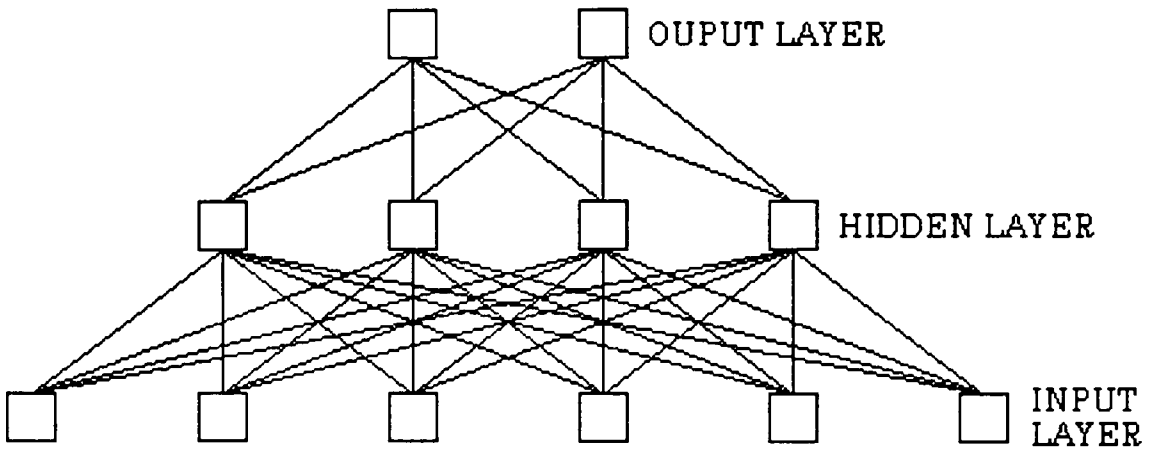

Fig 2.2 Fully connected neural network architecture

The hidden layer is where the major problem solving occurs. The values of the weights on the nodes of the hidden layer constitute the "internal representations" of the net and are based on the network's experience of the training data. The more hidden units there are the more complex a decision surface is formed. However, if there are too many hidden units, the network will possess too much modeling power and will learn the training set along with any inherent idiosyncrasies. It will then have poor generalization and may not be able to recognize new data not contained in the training set. Some rules of thumb are offered in determining a good number of hidden units for neural networks but few seem to be based on any solid theory. The most common method of determining an optimal number of hidden units was trial and error. Much research has been carried out since and other methods have been presented as an alternative. Most go hand-in-hand with methods to reduce the connectivity of the network.

Neural networks are trained using training pairs, which consist of an attribute or feature vector and a class vector. The input feature vector is applied to the network and

the output of the network calculated. The output is compared with the target output (the class vector) and the difference between the two is fed back to the network. A training algorithm is used to adjust the weights of each node to minimize the error. Feature vectors are applied sequentially and the weights are adjusted until the error for the whole training set is at an acceptably low level. During the training process, weights will gradually converge towards the values, which will match the input feature vector to the desired output. The most common approach to training is to use a gradient descent algorithm such as the error backpropagation algorithm described earlier in section 2.3.1.

Neural networks have many potential benefits. They are massively parallel, which provides high computation rates. They contain a degree of robustness compared to sequential computers due to the large number of nodes whose connections are primarily local. Overall performance need not be impaired significantly due to damage to a few nodes or links. There is also the improvement in performance with time that makes them suitable for tasks such as the one in question here of speech recognition [32]. Two of the major problems in speech recognition have been due to fluctuations on the speech pattern time axis and spectral pattern variation [33]. Neural networks have proved to be useful tools for the task of discriminating between different categories of spectral patterns such as those obtained for speech signals.

Many of these advantages are not applicable to neural networks, which are implemented on sequential computers, as is most often the case. The benefits are realized when parallel-computing technology is employed. There has been much research on the implementation of neural networks on highly parallel computing architectures and typical

neural network algorithms map successfully onto SIMD (Single Instruction stream. Multiple Data streams) architectures [34]. It should be noted the HMMs have also been implemented in parallel bringing many of the associated advantages.

There are now several different types of neural networks and training algorithms. Some like Kohonen's self-organizing network [35], have been applied to speech recognition. However, the work in this thesis is concerned with a principle of supervised learning known as the feed-forward backpropagation (FFBP) which is being used for the speaker-dependent speech recognition problem. FFBP neural networks are suggested to work very well for classification problems and it is the prime reason why FFBP architecture was chosen for this problem of speech recognition. More practical details of the architecture are given in Chapter IV where the total implementation of the speech system is discussed.

CHAPTER III

SPEECH PROCESSING

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.

3.1 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 3.1 shows a block diagram of how an analog signal is represented in digital form.

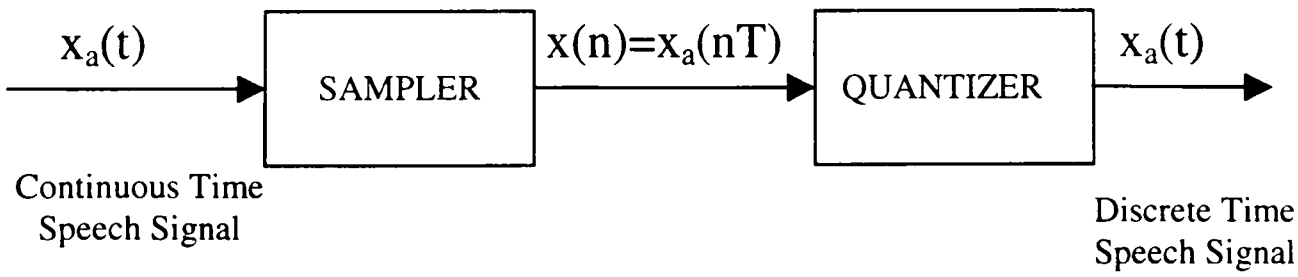

Figure 3.1 General block diagram of a digital waveform representation.

The very well-known sampling theorem (Nyquist Theorem) [36] states that a band-limited 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.

3.2 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. In this thesis, concepts from both the pattern recognition and artificial intelligence approaches are used for developing the speaker dependent speech system. Stochastic modeling using Hidden Markov Models (HMM) has become especially popular in modern speech recognition systems.

3.2.1 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 measure [37]. 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.

3.2.2 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 the 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 [38] 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 1/2 at the edges. Another algorithm called the Unconstrained Endpoints, 2-to-1 Range of Duration (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 1/2 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 3.2. 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%.

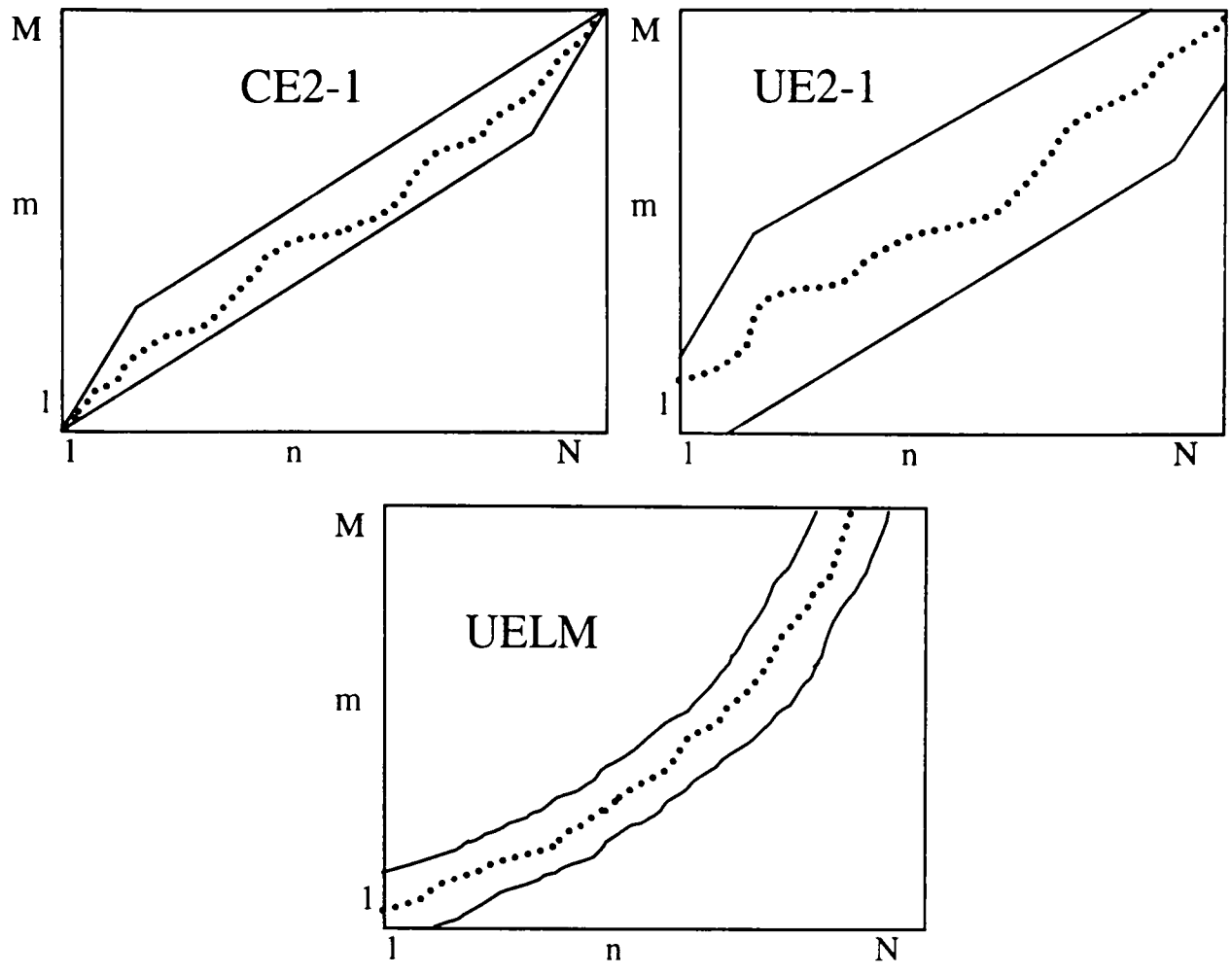

Figure 3.2 Typical warping paths for the three dynamic time-warping techniques.

3.2.3 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 a template based system can provide much faster results. Rabiner et al. [39] 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 hand-held 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 speaker-dependent speech recognition system with a small vocabulary that is real-time, small and cost-efficient. 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 following paragraphs describe the development of the entire speech recognition system that has been proposed. All the algorithms that have been developed and implemented are discussed briefly.

3.3 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 [40]:

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

The vowels are further classified into front (/i/, /I/, /e/, /ɛ/, and /æ/), middle (/ɜ/, /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.

3.4 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) [41] are available to researchers, the Texas Instruments 46-word Speaker-Dependent Isolated Word Corpus [42] 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 classes 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 3.3 and 3.4 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.

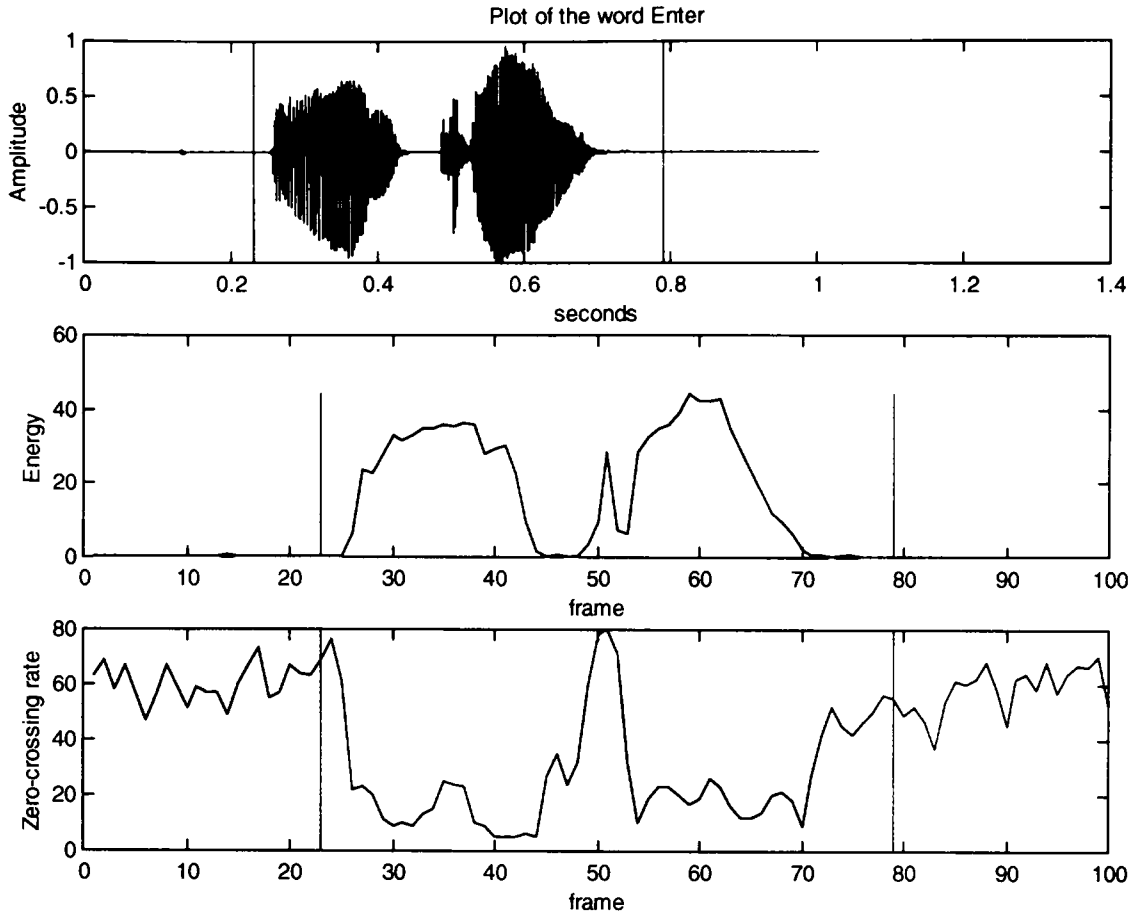

Figure 3.3. Short-time energy and zero-crossing data for the word "Enter" by a female speaker

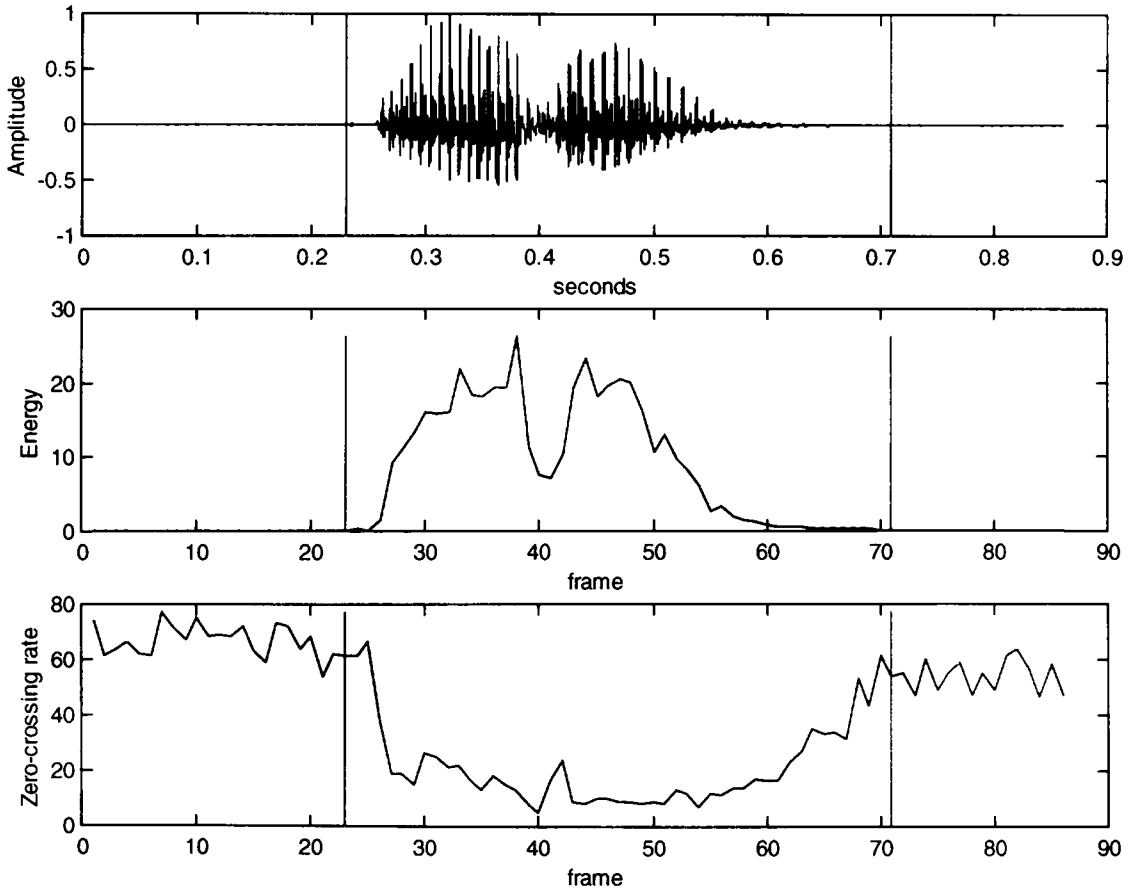

Figure 3.4. Short-time energy and zero-crossing data for the word "Enter" by a male speaker

Fig 3.3 and 3.4 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.

3.5 Speech Recognition System

Figure 3.5 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 (target vectors) 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. The operating modes of the system are independent and are implemented separately.

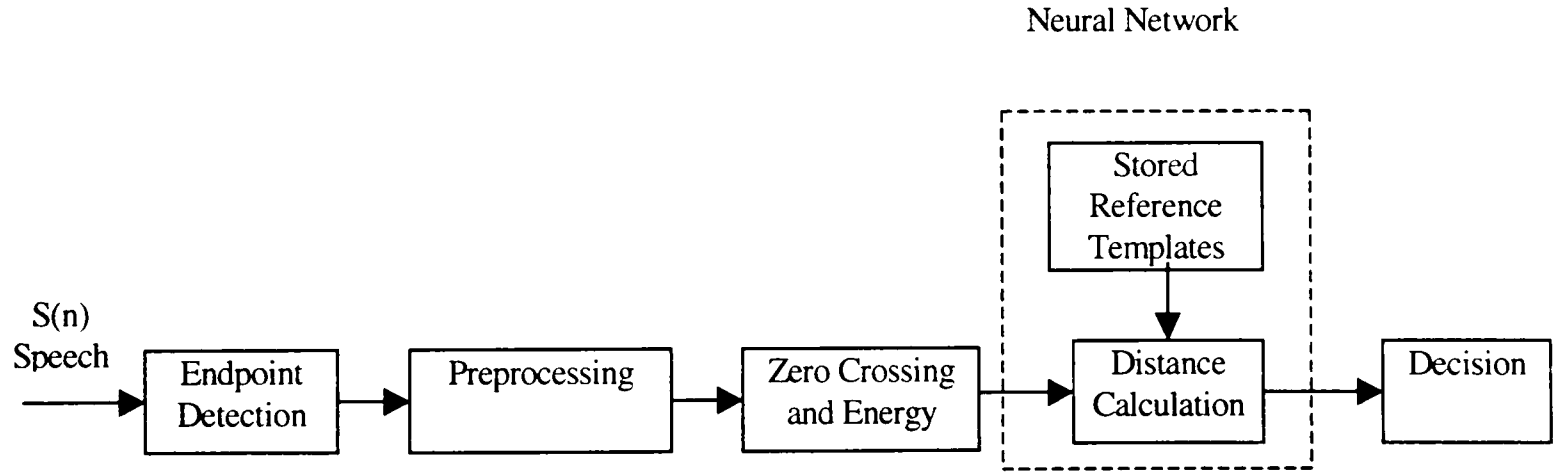

Figure 3.5 Block Diagram of the speech recognition system.

3.6 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 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.

3.6.1 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 3.6 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.

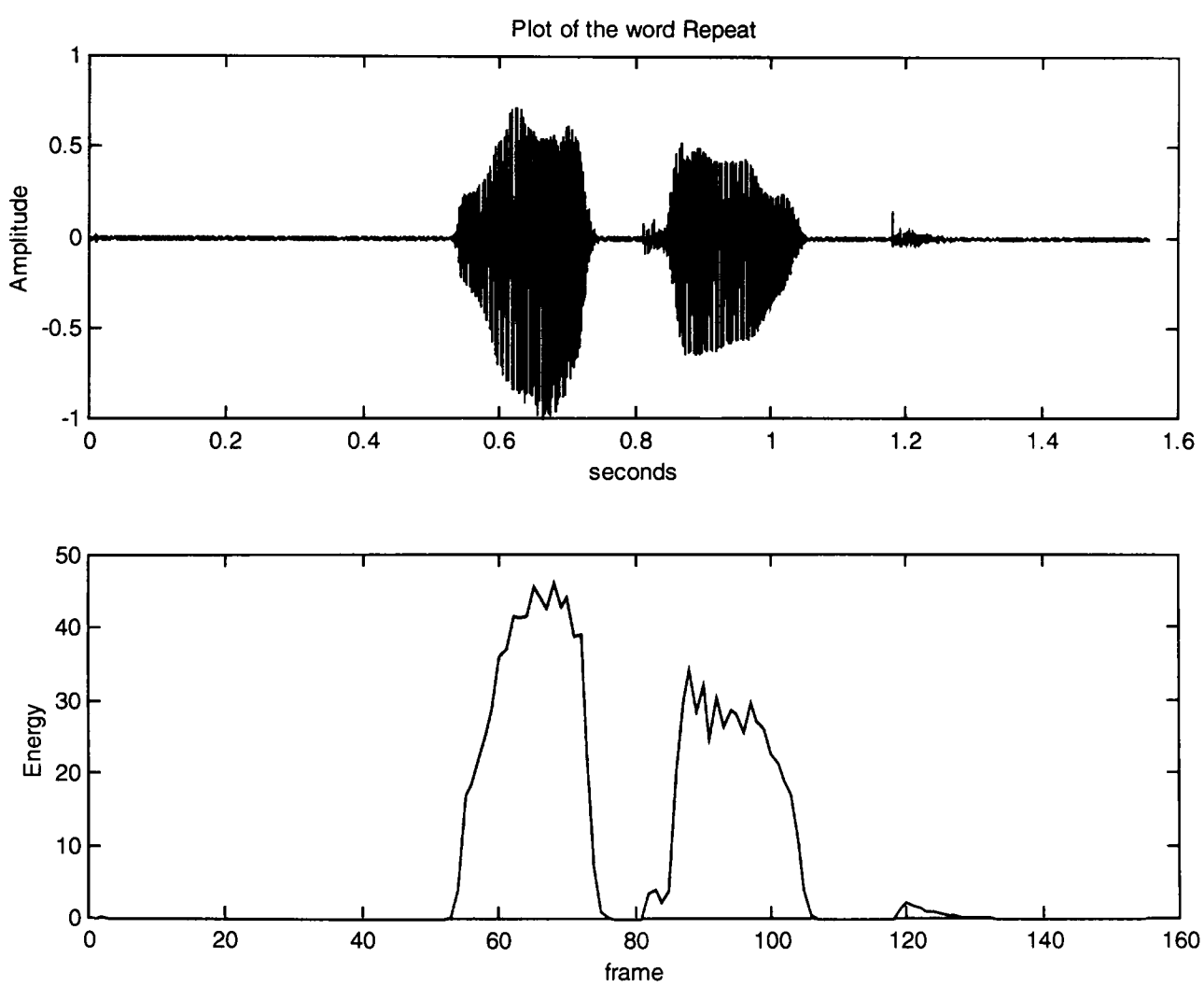

Figure 3.6 Acoustic waveform and short-time energy function for the word "Repeat".

3.6.2 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 [37] 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

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

and

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

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 3.7 shows the acoustic waveform and the average zero crossing rates for the word "Repeat". As can be seen from the figure, the voiced and unvoiced regions are quite prominent.

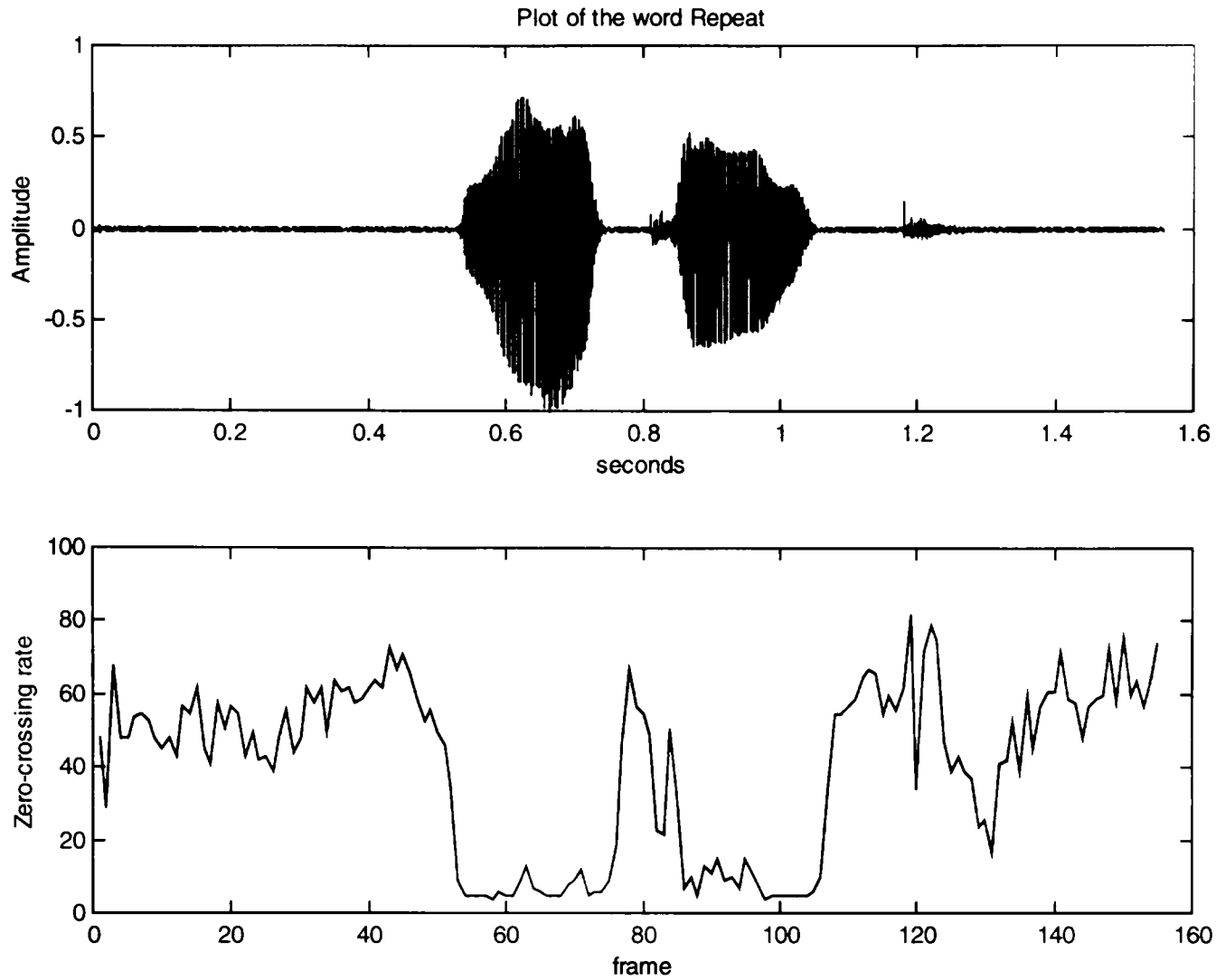

Figure 3.7 Acoustic waveform and zero-crossing rate for the word "Repeat".

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.

3.6.3 Endpoint Detection Algorithm

Rabiner and Sambur [43] have proposed an endpoint detection algorithm using zero crossing and energy measures. Figure 3.8 shows the short-term zero crossing and energy measures plotted for the word "four."

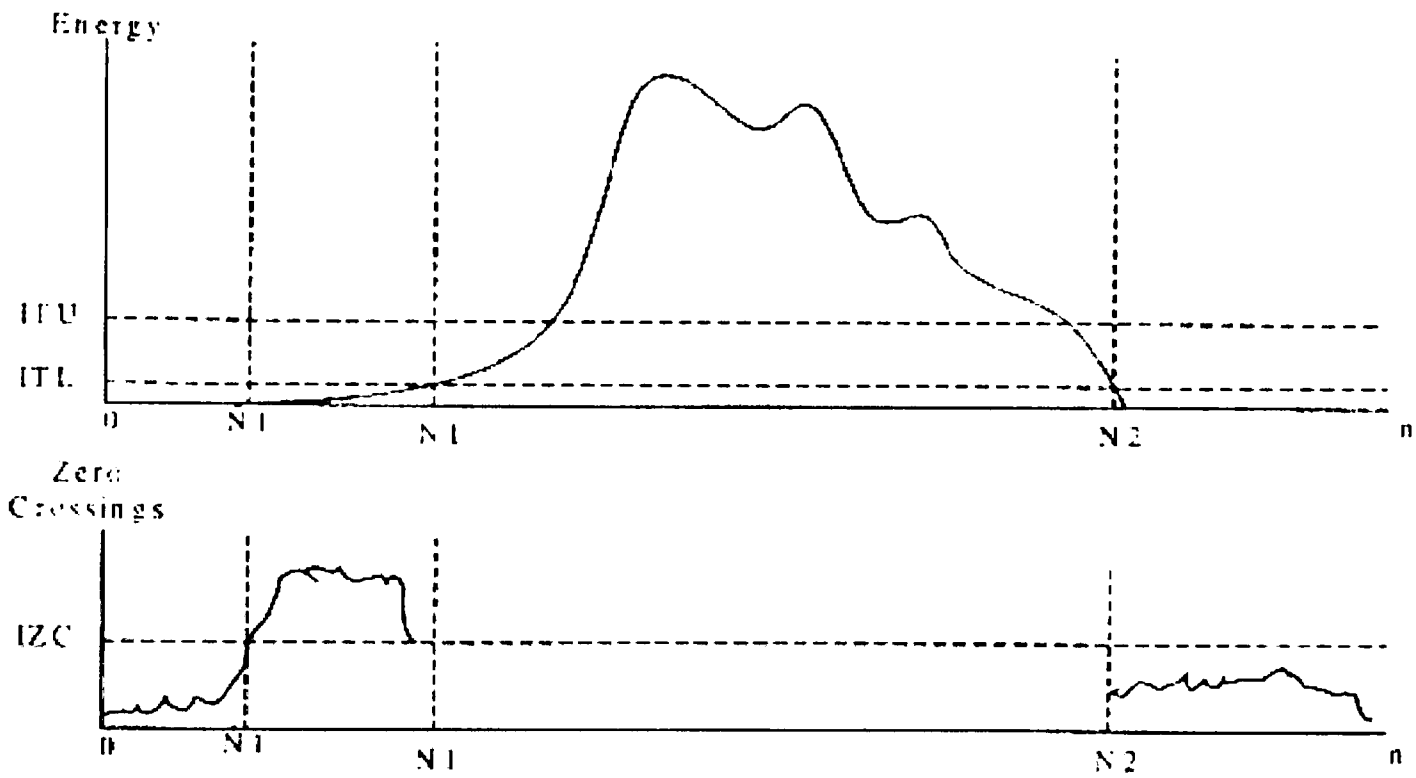

Figure 3.8 Short-time energy and zero crossings data for the word "four."

The curves were obtained by calculating the measure 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 the figure. 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

the figure. 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.

3.7. 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 earlier in this chapter. DTW algorithms work extremely well but can become computationally expensive and therefore unsuitable in a realtime environment. A simpler approach would be 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 3.9 and 3.10 show how this can be accomplished. The average values of zero crossings and energy content for the other words are given in Appendix B.

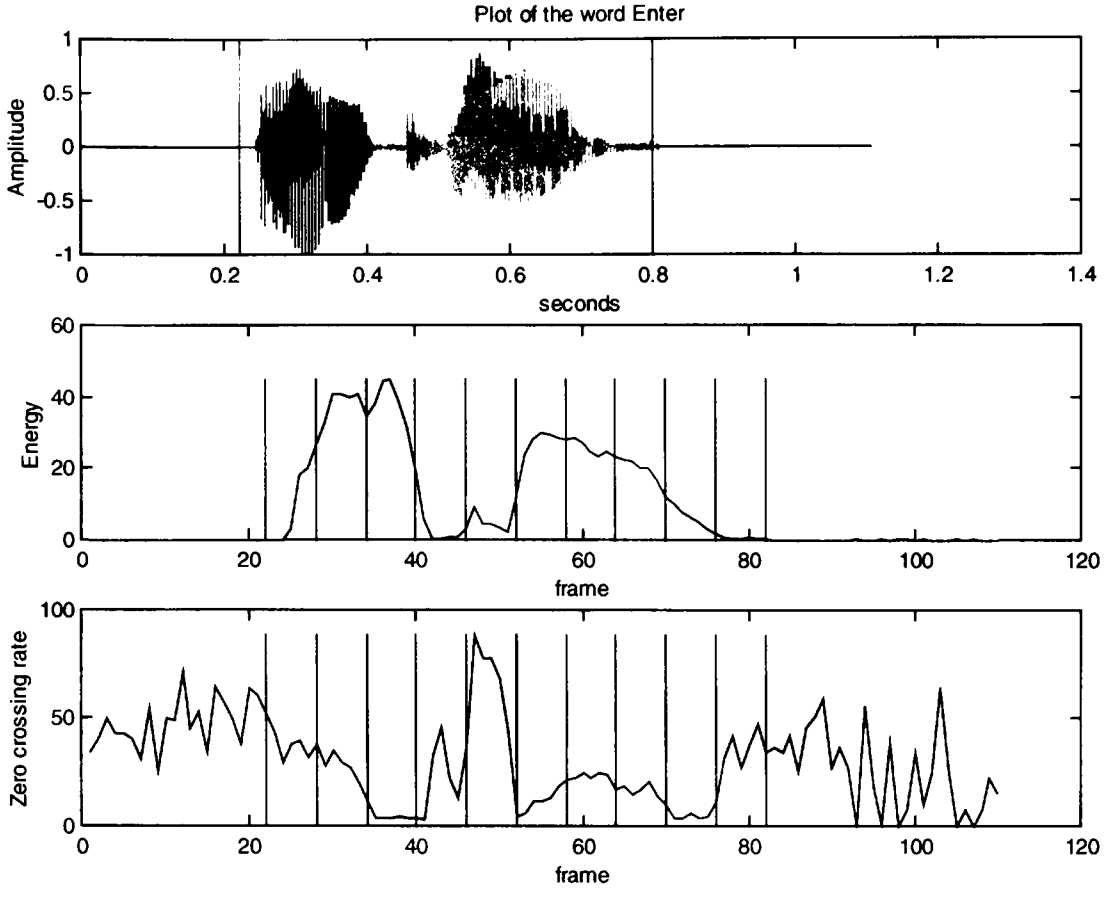

Figure 3.9 Plot of the word 'ENTER' showing equally spaced sections

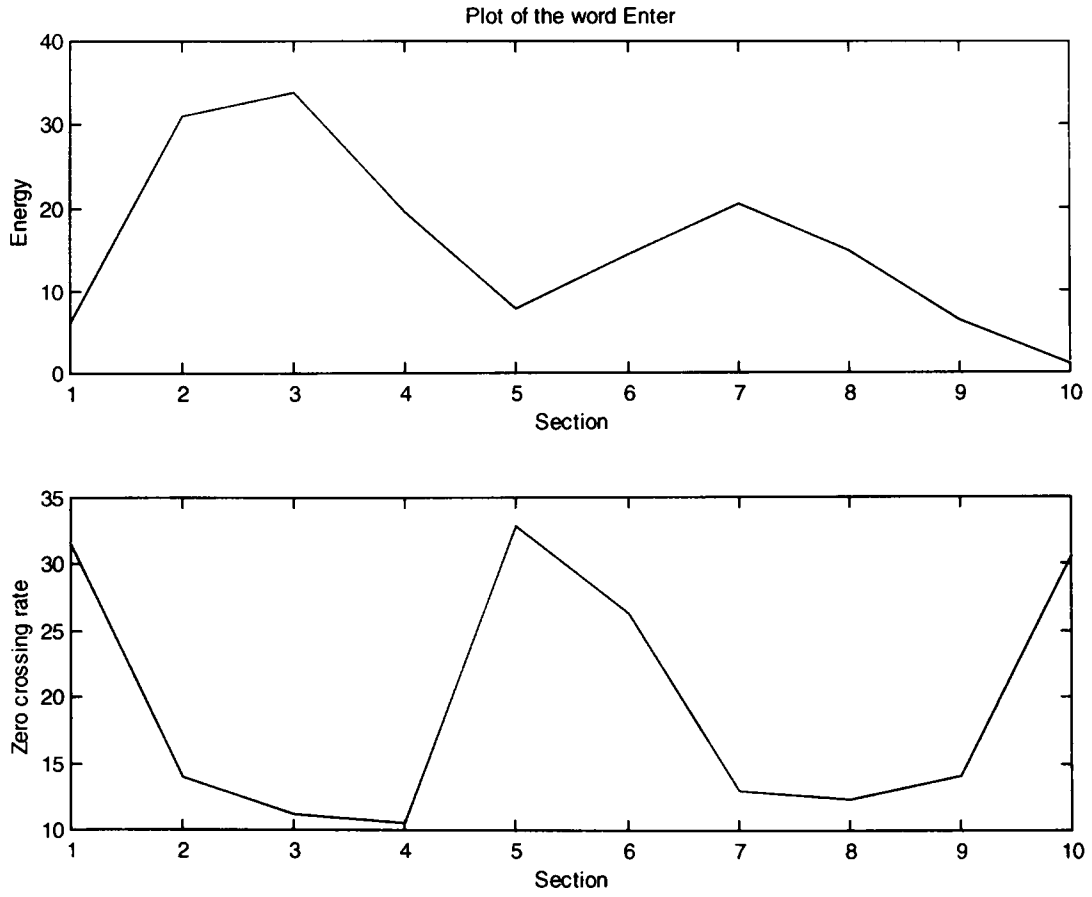

Figure 3.10 Average values of zero-crossing and energy content for the word "Enter".

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. These graphs 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. As seen in the figures, 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.

3.8. Template Matching

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 $\mathfrak{R}^N$. The metric on $\mathfrak{R}^N$ 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 $\mathbf{x}$. 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.

A Euclidean metric is utilized in the speech recognition system for the final decision. Template matching is done by calculating the distance metric between the

unknown utterance and the reference templates. The unknown word is classified into one of the reference templates if the corresponding Euclidean distance is minimum.

Chapter IV provides a more detailed discussion of the actual algorithm along with the pseudo code.

CHAPTER IV

IMPLEMENTATION OF THE SPEECH 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.

The system was 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. The software implementation of the system on the PC consists of two main modules. The first module is the reference template creator, which includes preprocessing of the speech signals and the training of the neural network with the generated feature vectors. The second is the recognition module, which preprocesses the unknown utterance and makes a decision as to which word the unknown utterance belongs to. The setup of the above-mentioned modules is pictorially described in Figure 4.1.

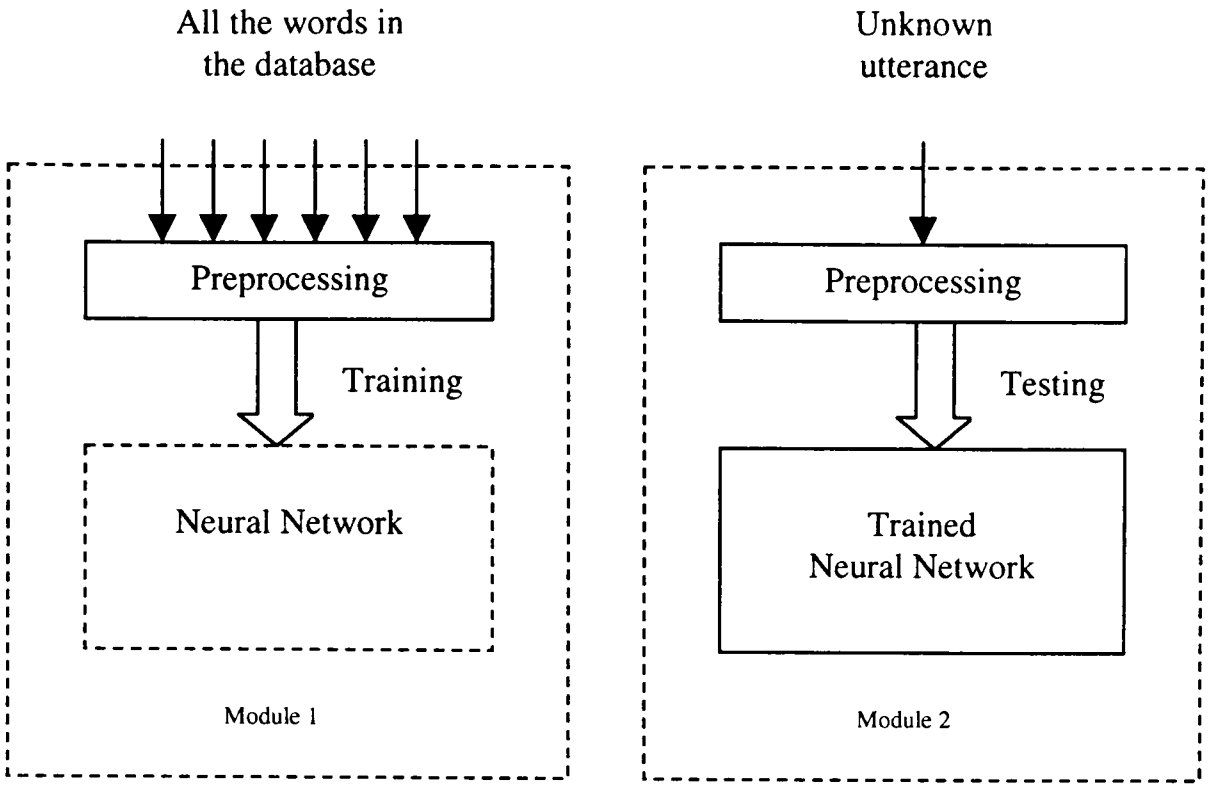

Figure 4.1 Speech recognition modules.

4.1 Speech Acquisition and Database

The TI-46 database was used to develop and test the software implementation of the system. The sound was 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 signal in the database was 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 [44]. 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. The program can only work on 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 [45] audio format converter/editor was used to accomplish this task. The 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. 16 of those utterances were classified as template creation data and the rest were classified as test data. The 16 test and template creation utterances consisted of utterances from 8 different male and 8 different female speakers. The entire database consisted of 320 speech samples.

4.2 Feature Vector Generator

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
```

The FVG module starts by finding the endpoints of the utterance, where the beginning and ending of the unknown word is determined. Figures 4.2, 4.3 and 4.4 show the flowcharts for the endpoint algorithm.

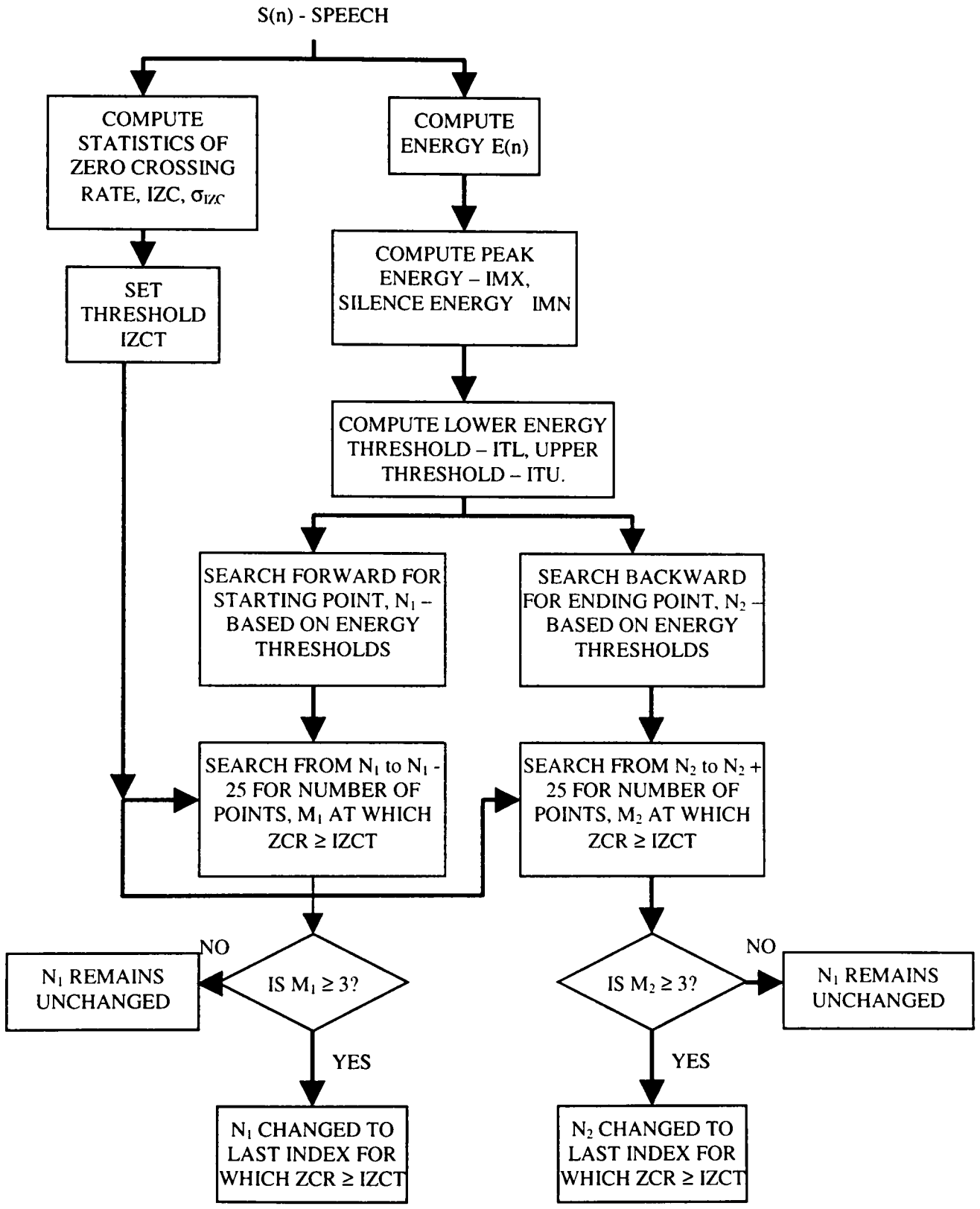

Figure 4.2 Flowchart for the endpoint algorithm.

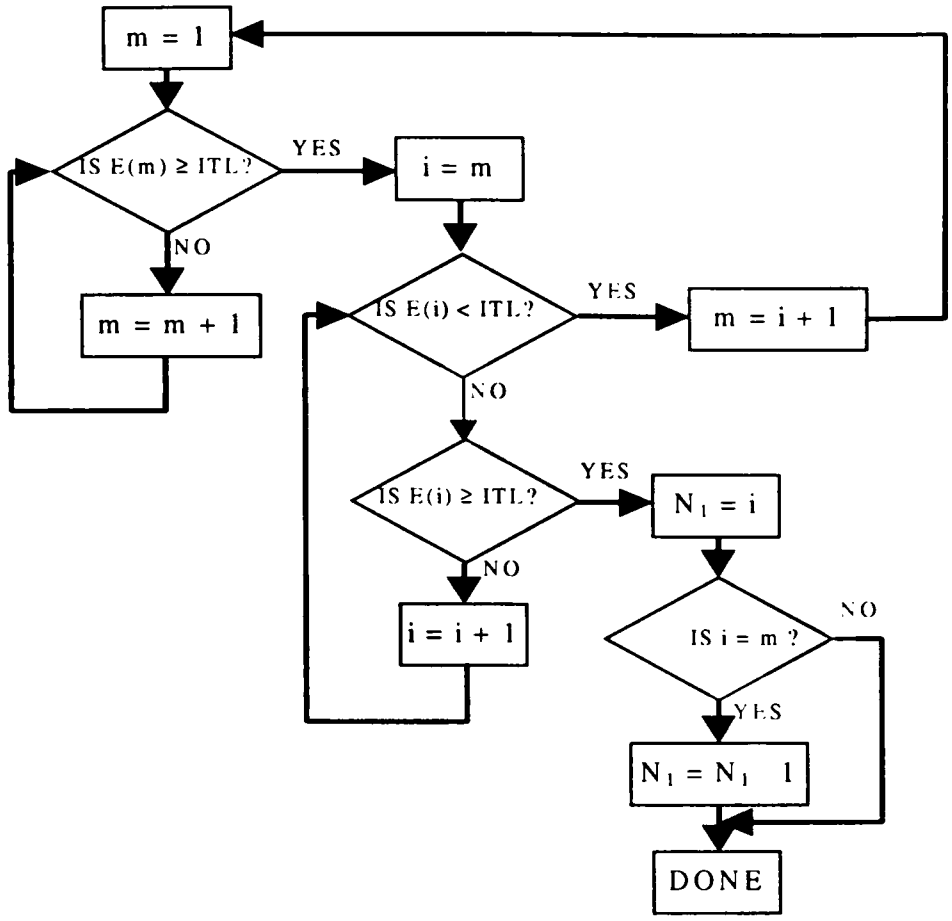

Figure 4.3 Flowchart for the beginning point initial estimate based on energy.

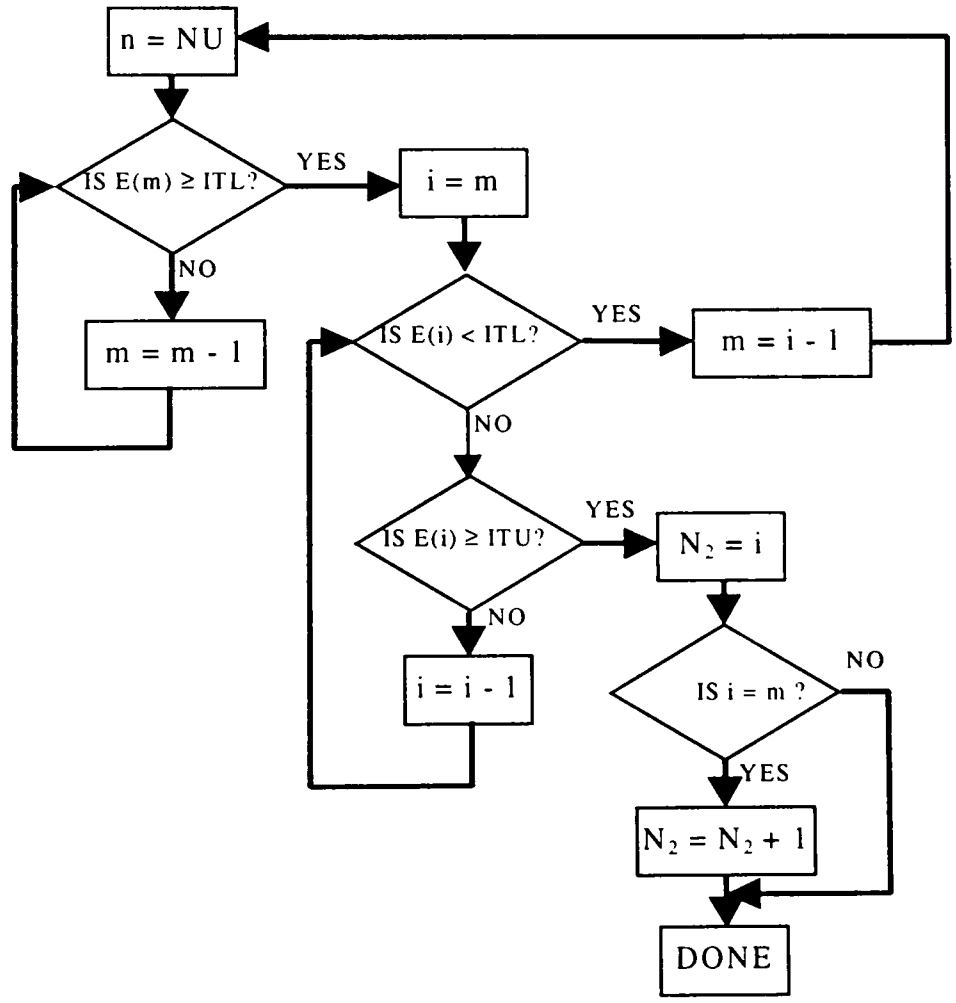

Figure 4.4 Flowchart for the ending point initial estimate based on energy

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 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 3.7.1 and 3.7.2. The final result is a feature vector of 20 dimensions for any given input utterance.

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 joint central moment of two random variables x and y may be written as

$$c_{xy} = E\{(x-\mu_x)(y-\mu_y)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\mu_x)(y-\mu_y)f_{xy}(x,y)dxdy. \tag{4.1}$$

The correlation coefficient is then defined as

$$\rho_{xy} = \frac{c_{xy}}{\sigma_x \sigma_y}. \tag{4.2}$$

Figure 4.5 shows pictorially the correlation between the different words. The graph was generated by calculating the correlation coefficient between words from their feature vectors, which was calculated by FVG.

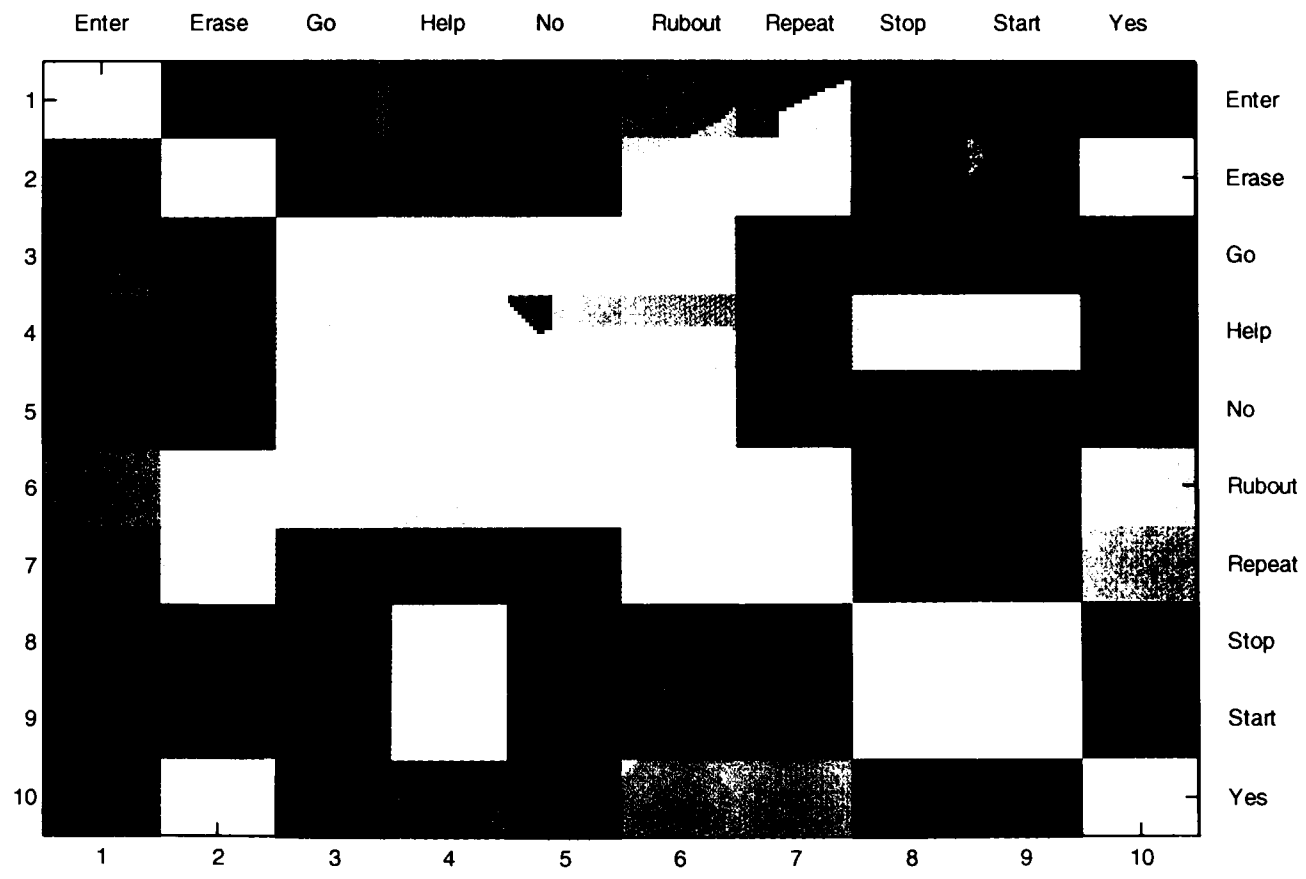

Figure 4.5 Correlation between words of the vocabulary.

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 and then calculate the correlation coefficient against other words. The word that produces the lowest correlation against all the other words in the vocubulary would make the best candidate for the system.

4.4 Reference Template Creation (Training)

The reference template creation module starts by asking the user for which words to create the reference templates (these reference templates will be the target vectors for the neural network). It then asks for the number of utterances in each word from which to create the reference template. The names of the files have been chosen in such a way to facilitate this aspect. The filename start with 'wd' and is followed by a word number and then 's' and finally the utterance number. For example, 'wd4s6.wav' refers to word number 4, which is "Help" and utterance number 6, which correspond to a female speaker in the test data.

The program then calls the FVG module that calculates the feature vector for that utterance. The program repeats this process until the feature vectors for all the utterances of a given word have been calculated. These feature vectors are inputs to the FFBP neural network. For the database that is dealt with in this thesis, there are 10 words 16 utterances. If the user chooses to train the network with all the words and for all the utterances, there will be 160 (number of words x number of utterances) feature vectors.

The mean of the feature vectors for all the utterances of an individual word is calculated, this calculation results in the target vectors. For this choice of database, there will be 10 target vectors. The neural network gets trained with the feature vectors with reference to the target vectors. 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 feature vectors from all the
    utterances
end

for each word in the vocabulary do
  for each utterance of the word do
      train the neural network with reference to all the target
      vectors
  end
end

store the trained neural network.
```

4.5 Speech Recognition Module (Testing)

The speech recognition module accepts an unknown or known utterance and tries to identify the utterance from the reference template. The module accepts a Microsoft Windows WAV file of sampling rate 12.5KHz and 16-bits per sample. The module calls the FVG module to calculate the feature vector of the utterance and this vector is fed as an input to the trained neural network. This feature vector propagates through the network and a series of weight updations take place in an attempt to match with one of the target vectors. An output vector is generated in this process, the Euclidean distance between the output vector and each of the target vectors is calculated. The program

makes a decision by choosing the word with the minimum distance. The following is the pseudo code for the speech recognition module.

```
load trained network
read unknown utterance
calculate feature vector by using FVG
feed the trained neural network with the generated feature vector

for all the words in the vocabulary do
  calculate distance between the resultant output vector and the
  target vector
end

find word with minimum distance
declare that recognized word to the corresponding template.
```

4.6 Speech Recognition Results

The system is tested with individual utterance of each word and the number of times the word was recognised was recorded. Table 4.1 summarizes the results of this experiment. The columns report the number of correct recognitions.

Table 4.1 Recognition results for all the utterances in the database

Word	Male (8 utterances)	Female (8 utterances)	Random (16 utterances)	% of correct matches
Enter	6	7	12	72
Erase	6	7	12	78
Go	5	6	11	69
Help	7	8	12	84
No	7	7	12	81
Rubout	6	6	11	72
Repeat	7	8	13	87
Stop	6	7	12	78
Start	6	6	11	72
Yes	6	7	10	72

Table 4.1 reports an average recognition rate of 76.5%. The results shown above correspond to a FFBP, which had 5 nodes in the input layer, 10 nodes in the output layer and no hidden layers. The algorithm used for training was *trainlm* (Levenberg-Marquardt backpropagation)[46], t*rainlm* is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization. Training occurs according to the *trainlm's* training parameters, which are shown in Table 4.2 with their default values:

Table 4.2 Default values of the training parameters and the values chosen for experimentation

Function name	Default values	Chosen values For Experimentation
net.trainParam.epochs	10 (Maximum number of epochs to train)	500
net.trainParam.goal	0 (Performance goal)	20
net.trainParam.lr	0.01 (Learning rate)	0.01
net.trainParam.max_fail	5 (Maximum validation failures)	5
net.trainParam.mem_reduc	1 (Factor to use for memory/speed trade off)	1
net.trainParam.min_grad	1e-10 (Minimum performance gradient)	1e-10

The time taken for training the network was approximately 13.5hours/training cycle and the time taken for testing the network with individual utterances was 0.15sec/word. Changing the training parameters shown in Table 4.2 can alter this recognition rate. For example the learning rate is increased to 0.1 and a drop of 6% in the recognition rate was observed. Thus for this thesis which aims at a strictly speaker dependent system, it is advised that the network is trained until an optimum recognition rate is obtained, once an optimum value is obtained the network should be made available only for testing. The recognition rate obtained shows an improvement of 12% over the

recognition rate reported in the work done by Abid Zindani in his thesis [47]. In his work, the author uses zero crossing and energy content for the reference template generation but a different technique for speech recognition was used. A circle of confidence was defined and if the unknown utterance falls within the radius of that circle, then the word was reported as recognized. Thus comparing both techniques from the result perspective, neural networks prove to be a better technique than this conventional speech recognition technique. The primary aim of this research is to develop a hand-held system for the voice-impaired people, where the speech utterances by the speaker are quite indistinct when compared to a normal person's speech. Better recognition techniques that are capable of identifying the subtle differences in the speech utterances are desired. The computations in neural network models resemble the human intelligent thinking system; they are adaptable, context-sensitive, error-tolerant, large memory capacity, and real-time capability. Thus neural network techniques have been studied and applied for recognizing the voice-impaired person's speech. The results tabulated above and the comparison with other work done with conventional methods show that neural network techniques work well for the 16-speaker database, which is a normal person's speech. To test the capability of neural networks for recognizing indistinct speech, a random database of speech utterances mimicking a voice-impaired person's speech was generated. There was no standard voice-impaired speech database available. Four words out of the standard 10-word database used previously were chosen and five utterances for each word were recorded to try to mimic a voice-impaired person's speech. This database was used to train the neural network and then it was tested with the words within the database. The recognition rate observed was 60%, which is 16.5% less than the recognition rate

obtained in the normal person's speech database. The results for the test database are tabulated in Table 4.3.

Table 4.3 Recognition results for the test database

Word	S1	S2	S3	S4	S5	% of correct matches
Enter	√	√	√	x	x	60
Erase	√	√	√	x	√	80
Go	√	√	x	x	√	60
Rubout	x	√	√	x	x	40

√ - correct recognition of the word
x - incorrect recognition of the word

A standard database with a real voice-impaired speech database, more number of words and speakers will allow the neural network to learn a lot of information about the speech signals and enhance the network's memory. Thus it can be optimistically stated that considering the above factors and a few changes in the network parameters will considerably bring back the system's performance to the expected recognition rate.

CHAPTER V

CONCLUSIONS

The speech system described in this thesis is an attempt at developing a hand-held system for the speaking-impaired people. The approach presented here describes a variety of robust measurements and techniques that can be implemented on a platform suitable for hand-held systems. The speech recognition system discussed was implemented using the techniques of zero-crossing rate and energy content for preprocessing the input speech signal. The speech of a voice-impaired person is quite unclear when compared to a normal person's speech, thus a better technique for recognition was sought after and neural network proved to be handy because of its high-memory capacity, adaptability to the data and clear classification capability. Thus, in the recognition phase a feed forward neural network was employed.

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 76.5% correct recognition on a test database of 16 different speakers with a vocabulary of 10 words. The system also reported a 60% correct recognition on another test database (which mimics a voice-impaired person's speech, created for the sole purpose of testing it on this speech system) with 5 speakers and a vocabulary of 4 words. The system was able to recognize the words from both the databases at an average rate of 0.15sec/word, but the system has to be trained before it can be tested. The average training periods were 13.5hours/training cycle and 2hours/training cycle for the 16-

speaker and 5-speaker databases, respectively. The memory required to implement the speech recognition system on a DSP is estimated approximately to 80KB. The speed, accuracy and memory requirements make this system a cost-effective approach to hand-held speech recognition devices for the voice-impaired people.

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 in section 3.2.2, 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 the speed of DSP processors can realize the problem in real time in the near future.

Another problem that needs to be addressed is the neural network architecture chosen for developing the recognition system. In the network described for the system there were no hidden layers chosen during the training phase. The choice of the number of hidden layers in the feed-forward structure design often involves considerable engineering adjustment. Often, trade-offs between training time and mapping accuracy lead to iterative adjustment of the network using simulation. For the given speech recognition problem, "the more hidden layers the better" could be used to guide sizing of the hidden layer, since the number of hidden layers controls the flexibility of decision boundaries. However, extensively large numbers of hidden layers may be counter-productive. The network training time is influenced by the size of the hidden layer. Thus

an optimum number of hidden layers chosen will greatly enhance the decision criteria. The feed forward architecture exemplifies supervised learning. Networks that are used to determine natural clusters or feature similarity from unlabeled samples, commonly called self-organizing networks, can be classified as unsupervised learning algorithms. These algorithms have some measure of pattern associativity or similarity, which is used to guide the learning process. These unsupervised architectures can substitute the feed forward architecture to obtain an improvement in the recognition rate for a real, massive voice-impaired database.

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

REFERENCES

[1] Doddington, G. (1989) "Phonetically Sensitive Discriminants for Improved Speech Recognition," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing.*

[2] Itakura, F. (1975) "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 23(1): 67-72

[3] Miyatake, M., Sawai, H., and Shikano, K. (1990) "Integrated Training for Spotting Japanese Phonemes Using Large Phonemic Time-Delay Neural Networks," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing.*

[4] Kimura, S. (1990) "100,000-Word Recognition Using Acoustic-Segment Networks," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing.*

[5] Hild, H. and Waibel, A. (1993) "Connected Letter Recognition with a Multi-State Time Delay Neural Network," *Advances in Neural Information Processing,* Morgan-Kaufmann Publishers, San Francisco.

[6] Lee, K.F. (1988) "Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System," *Ph.D. Dissertation, Carnegie Melon University.*

[7] Bahl, L., Bakis, R., Cohen, P., Cole, A., Jelinek, F., Lewis, B., and Mercer, R. (1981) "Speech Recognition of a Natural Text Read as Isolated Words," *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing.*

[8] Furui, S. (1993) "Towards Robust Speech Recognition under Adverse Conditions," *Proc. of the ESCA Workshop on Speech Processing and Adverse Conditions*, pp. 31-41, Cannes-Mandelieu, France.

[9] K. H. Davis, R. Biddulph, and S. Balashek, (1952) "Automatic Recognition of Spoken Digits," *Journal of Acoustic Society of America,* 24(6): 637-642.

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

[11] T. Sakai and S. Doshita (1962) "The Phonetic Typewriter Information Processing 1962," *Proceedings of the IFIP Congress, Munich*

[12] K. Nagata, Y. Kato, S. Chiba (1963) "Spoken Digit Recognizer for Japanese Language," *NEC Research and Development*, No.6

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

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

[15] D. R. Reddy (September 1966) "An Approach to Computer Speech Recognition by Direct Analysis of the Speech Wave," *Technical Report No. C549, Computer Science Department, Stanford University.*

[16] V. M. Velichko and N.G. Zagoruyko (June 1970) "Automatic Recognition of 200 Words," *International Journal of Man-Machine Studies*, 2:223

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

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

[19] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg and J. G. Wilpon (August 1979) "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Trans. Acoustics, Speech, Signal Processing*, ASSP-27 (1): 336-349

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

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

[22] Price, P. J., W. Fisher, J. Bernstein et al. (1988) "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.

[23] 21st Century Eloquence http://voicerecognition.com/1998/trends/

[24] Carlson, R., Granstrom, B., and Hunnicutt, S. (1982) "Bliss Communication with Speech or Text Output," *Proceedings of ICASSP 82*, 747-750

[27] Nadeu, C., Lieida, E, and Santamaria, M. E. (1985) "Trace Segmentation in A LPC-Based Isolated Word Recognition System," *Proceedings of MELECON 1985*, Volume 2: Digital Signal Processing, pp. 111-113, Madrid.

[28] Hebb, D. O. (1949) *The Organization of Behavior,* John Wiley & Sons, New York.

[29] Rosenblatt, R. (1959) *Principles of Neurodynamics*, Spartan Books, New York.

[30] Minsky, M. L., and Papert, S. A. (1988) *Perceptrons* (Expanded Edition), MIT Press, Cambridge.

[31] Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986) "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Vol.1. Foundations, pp. 318-362, MIT Press.

[32] Lippman, R. P. (April 1987) "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, pp. 4-22.

[33] Sakoe, H, Isotani, R., Yoshida, K., Iso, K. and Watanabe, T. (1989) "Speaker-Independent Word Recognition Using Dynamic Programming Neural Networks," *Proceedings of the IEEE Int. Conf. On Acoustics, Speech and Signal Processing*, ICASSP '89, pp. 29-32.

[34] Nordstrom, T. and Svensson, B. (1992) "Using and Designing Massively Parallel Computers For Artificial Neural Networks," *Journal of Parallel and Distributed Computing*, vol. 14, no.3, pp. 260-285.

[35] Kohonen, T. (1988) "The Neural 'Phonetic' Typewriter," *Computer*, pp.11-22, March 1988.

[36] Ferrel G. Stremler (1990), Introduction to Communication Systems. Third Edition Reading, Addison-Wiley, Massachusetts.

[37] M.R.Sambur, L.R.Rabiner (Jan 1975), "A speaker-Independent Digit-Recognition System," *The Bell System Technical Journal* Vol. 54, No. 1.

[38] Itakura F., (Feb 1985), Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23.

[39] Rabiner L. R., Levinson S. E and Sondhi M. M., (1983), "On the application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition," *The Bell System Technical Journal.*

[40] Lawrence R. Rabiner, Ronald W. Schafer,(1978), *Digital Processing of Speech Signals*, Prentice Hall, Englewood Cliffs, New Jersey.

[41] Price, P.J., W. Fisher, J.Bernstein et al. (1988), "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.

[42] Texas Instruments (TI) (Sep 1991) and the National Institute of Standards and Technology (NIST), *TI 46-Word Speaker-Dependent Isolated Word Corpus*. NIST Speech Disc7-1.1.

[43] Rabiner, L.R., Sambur, M.R., (Feb 1975), "An Algorithm for Determining the Endpoints of Isolated Utterances," *The Bell System Technical Journal*, vol. 54, No.2.

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

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

[46] *Neural Network Toolbox*, Matlab version 5.2.0, The Mathworks, Inc.

[47] Jindani, Abid M., (1998), "Speaker independent real-time speech recognition system" Thesis (MS in EE), Texas Tech University.

[48] Mary Jo Creaney-Stockton, (1996), "Isolated Word Recognition Using Reduced Connectivity Neural Networks With Non-Linear Time Alignment Methods" Ph.D. dissertation, University of Newcastle, Upon-Tyne.

APPENDIX A

SHORT TIME ENERGY AND ZERO CROSSINGS

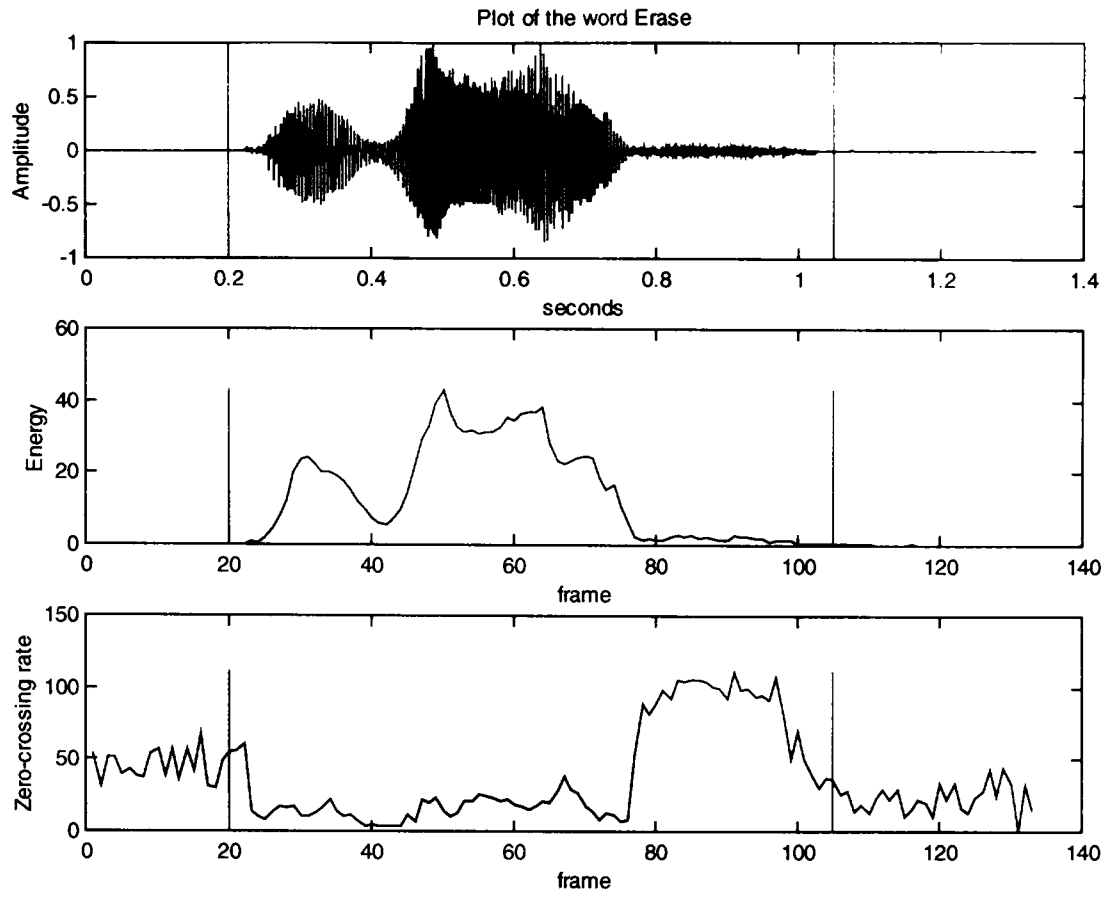

Figure A.1 Short-time energy and zero-crossing data for the word "Erase" by a female speaker

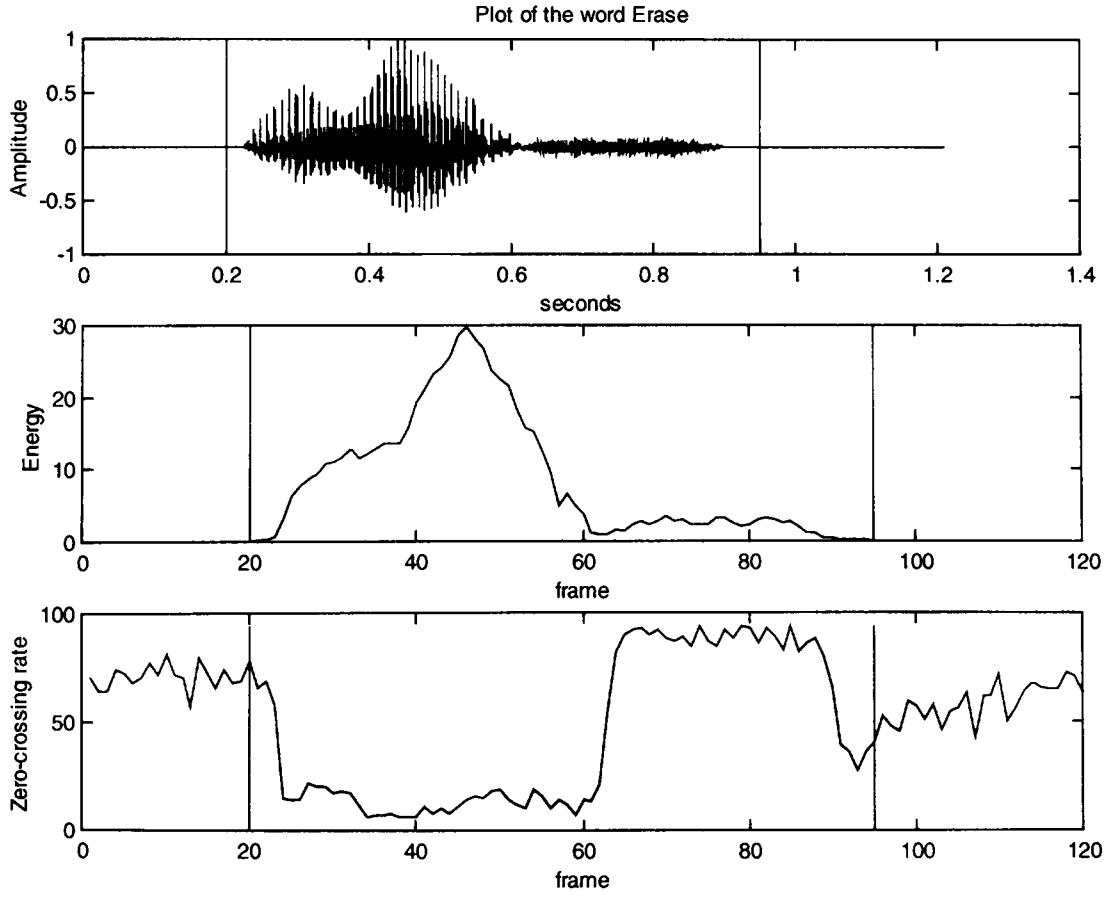

Figure A.2 Short-time energy and zero-crossing data for the word "Erase" by a male speaker

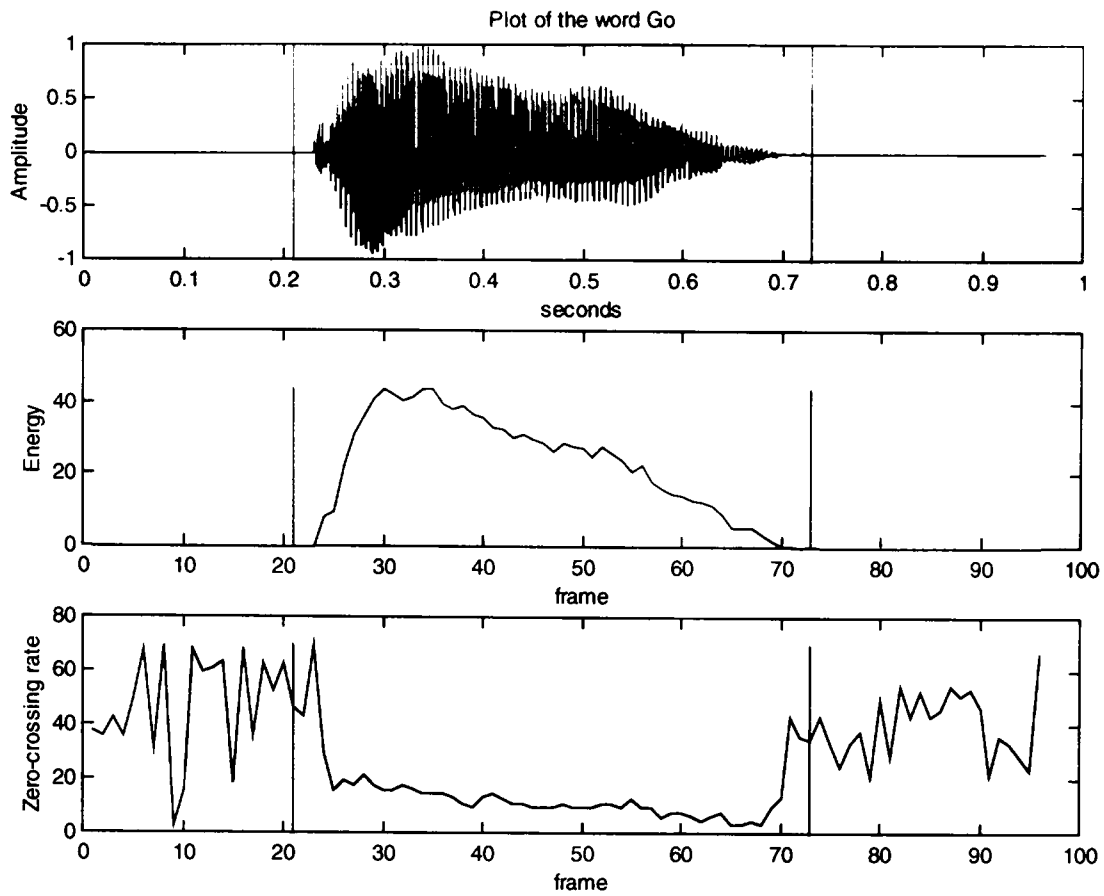

Figure A.3 Short-time energy and zero-crossing data for the word "Go" by a female speaker

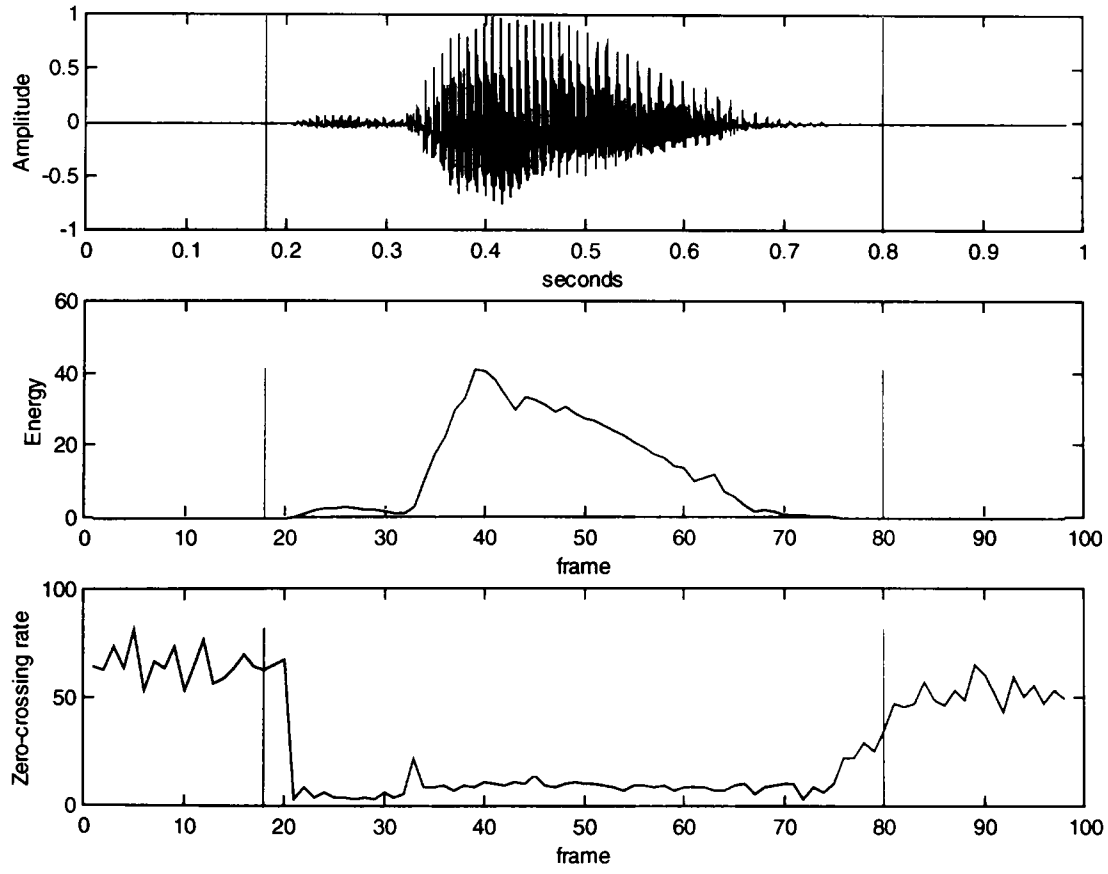

Figure A.4 Short-time energy and zero-crossing data for the word "Go" by a male speaker

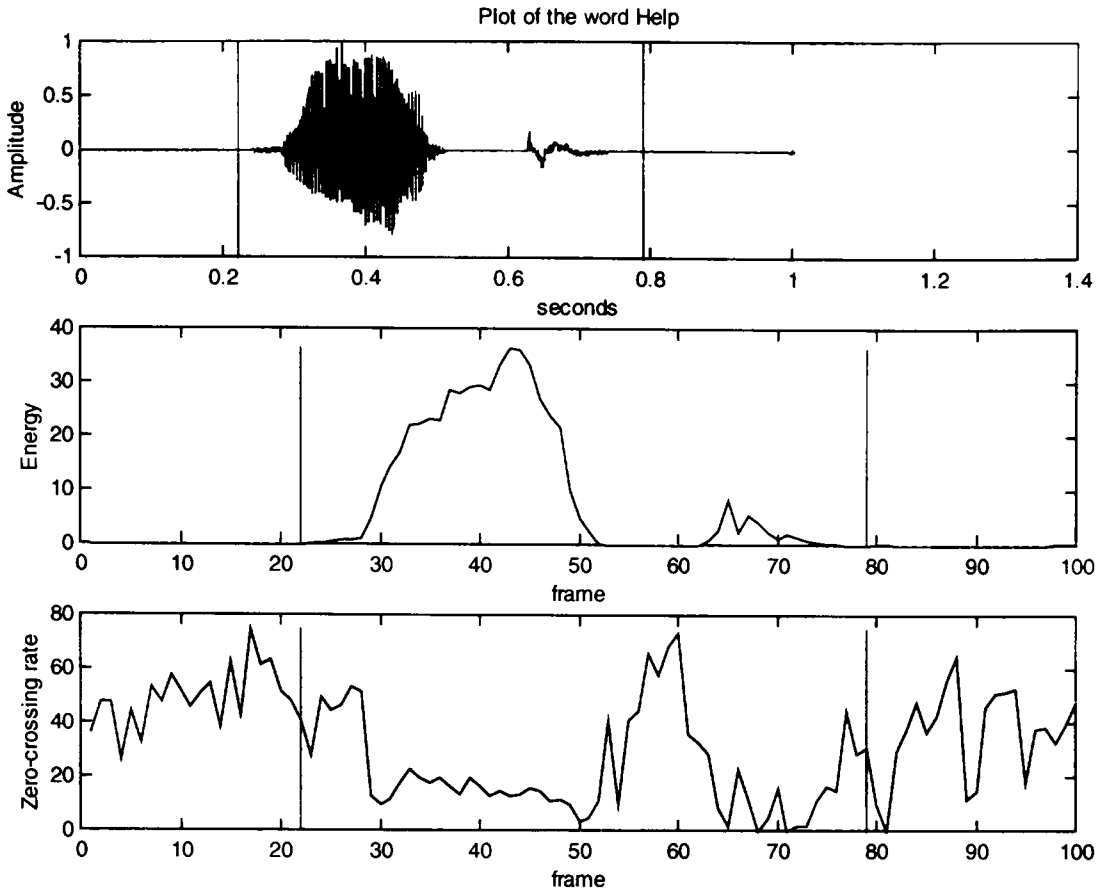

Figure A.5 Short-time energy and zero-crossing data for the word "Help" by a female speaker

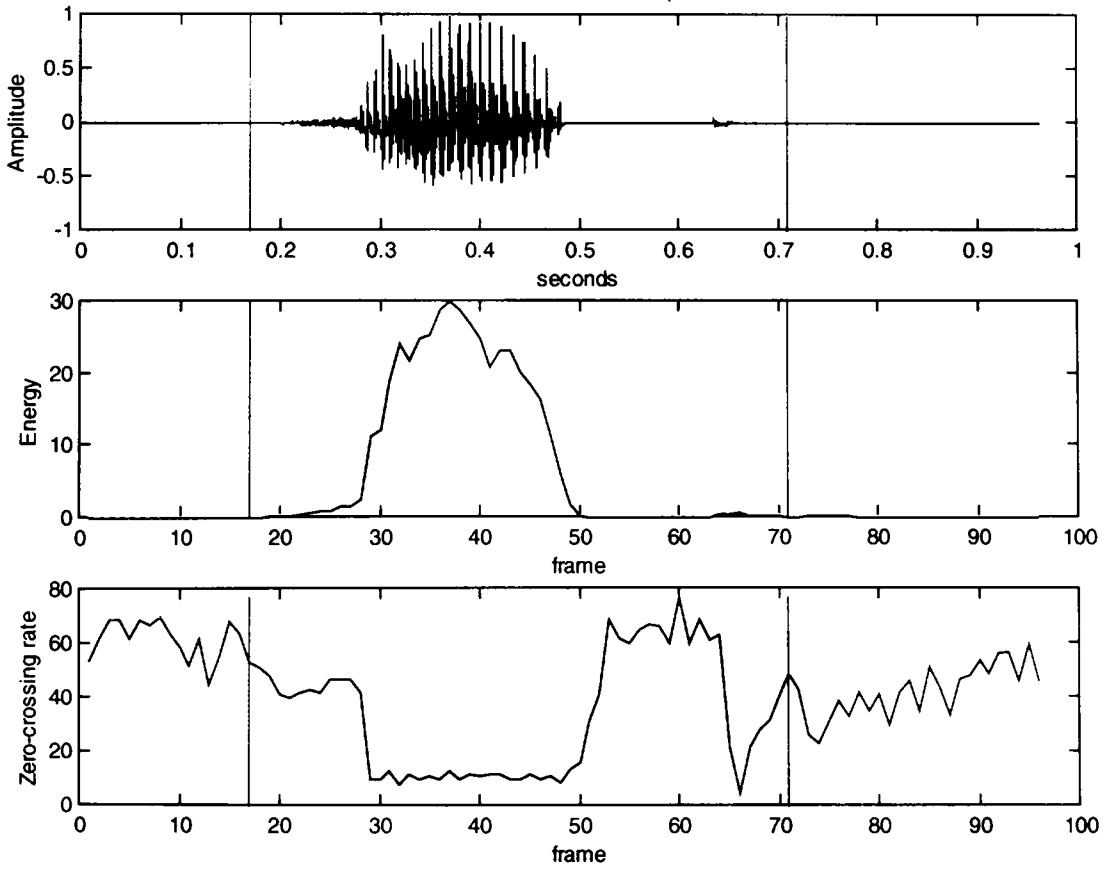

Figure A.6 Short-time energy and zero-crossing data for the word "Help" by a male speaker

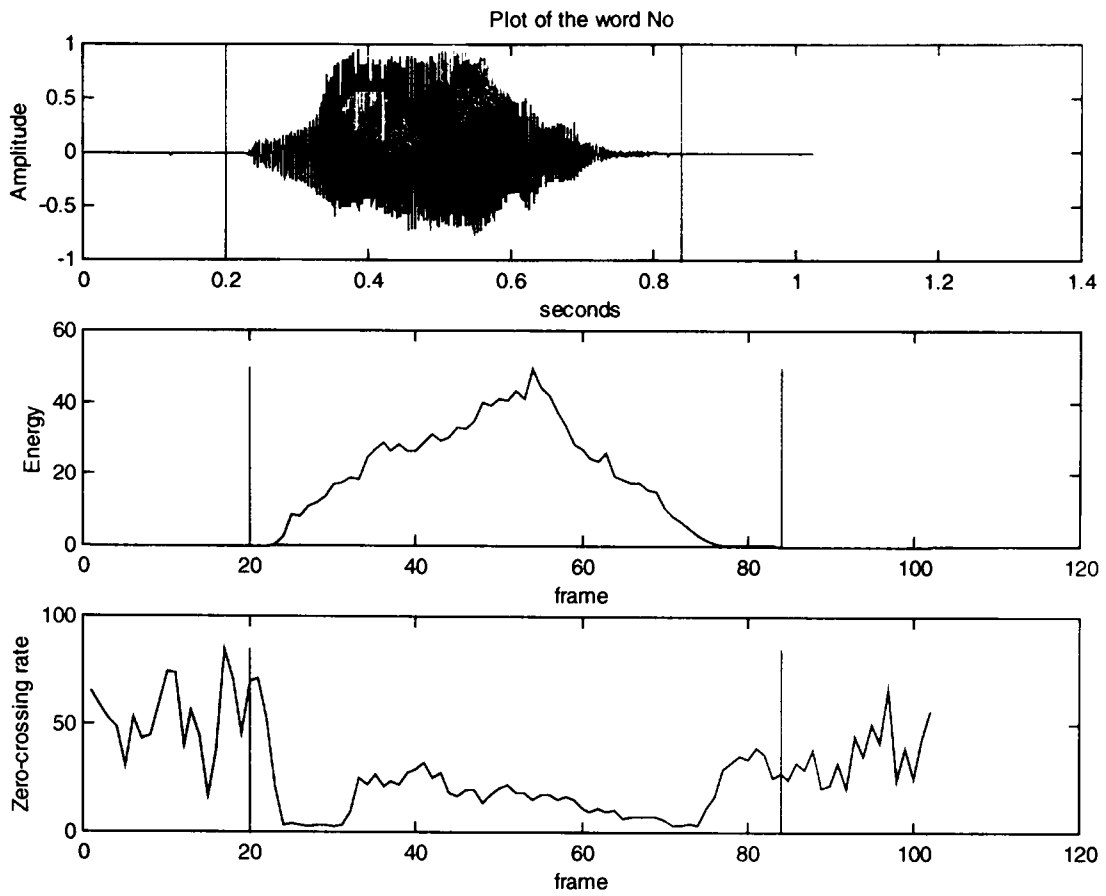

Figure A.7 Short-time energy and zero-crossing data for the word "No" by a female speaker

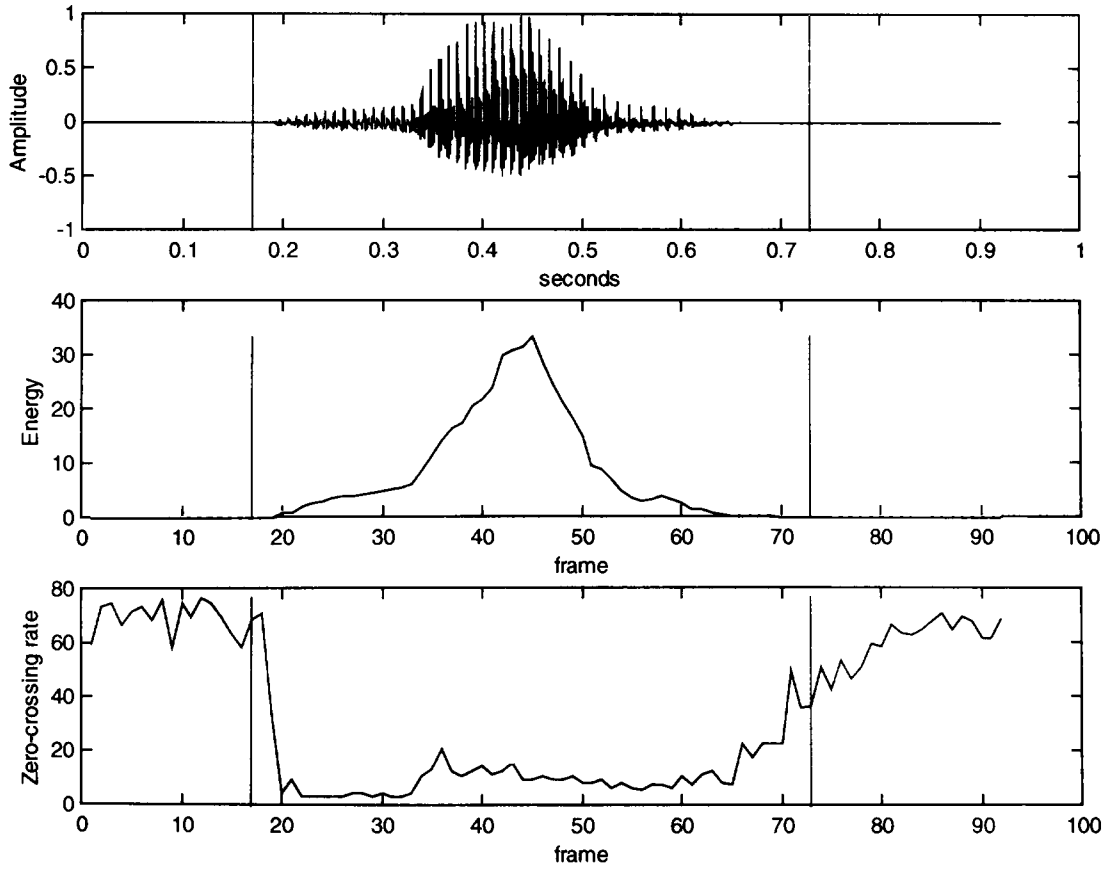

Figure A.8 Short-time energy and zero-crossing data for the word "No" by a male speaker

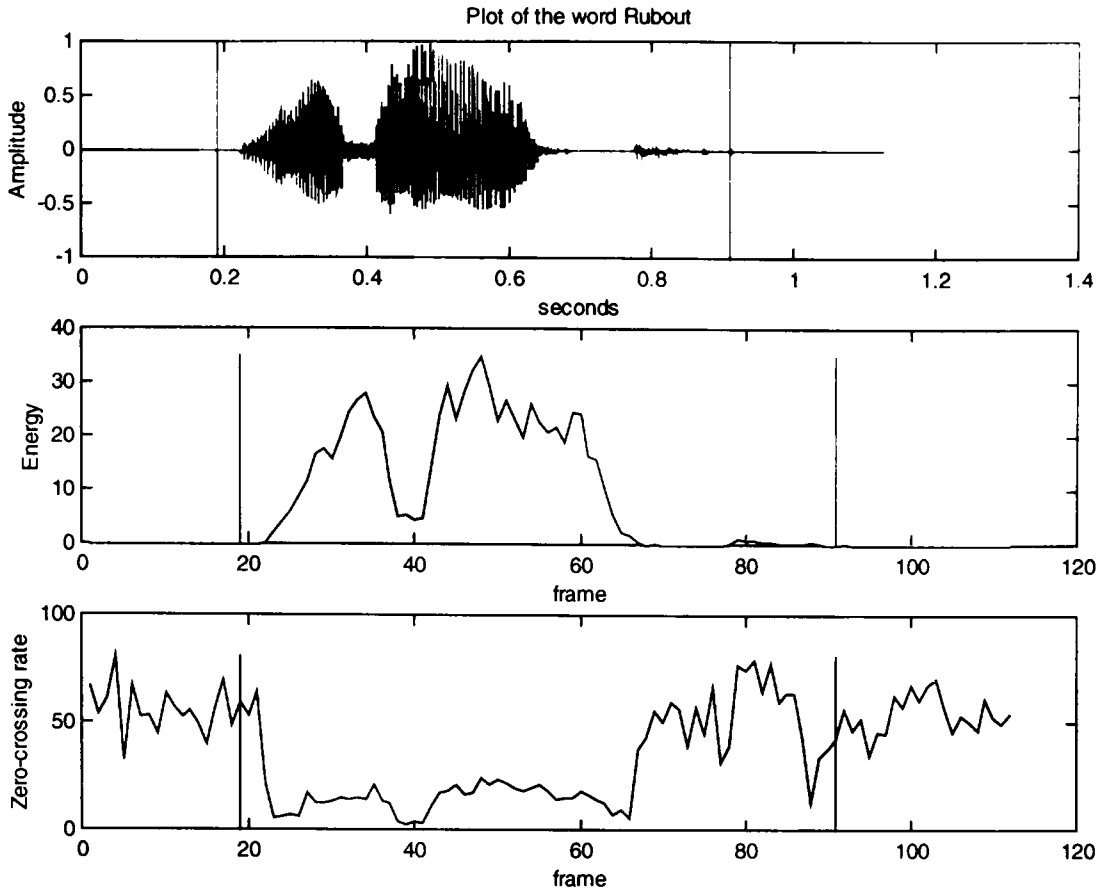

Figure A. 8 Short-time energy and zero-crossing data for the word "Rubout" by a female speaker

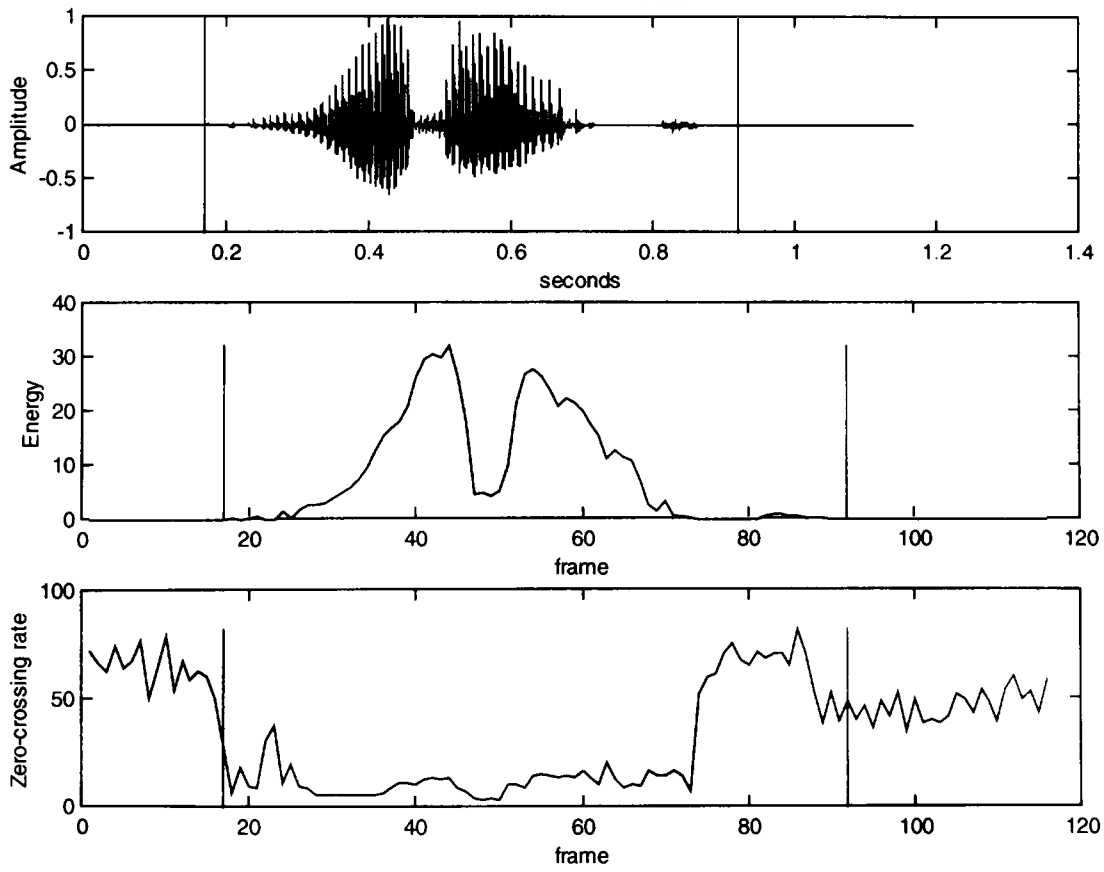

Figure A.9 Short-time energy and zero-crossing data for the word "Rubout" by a male speaker

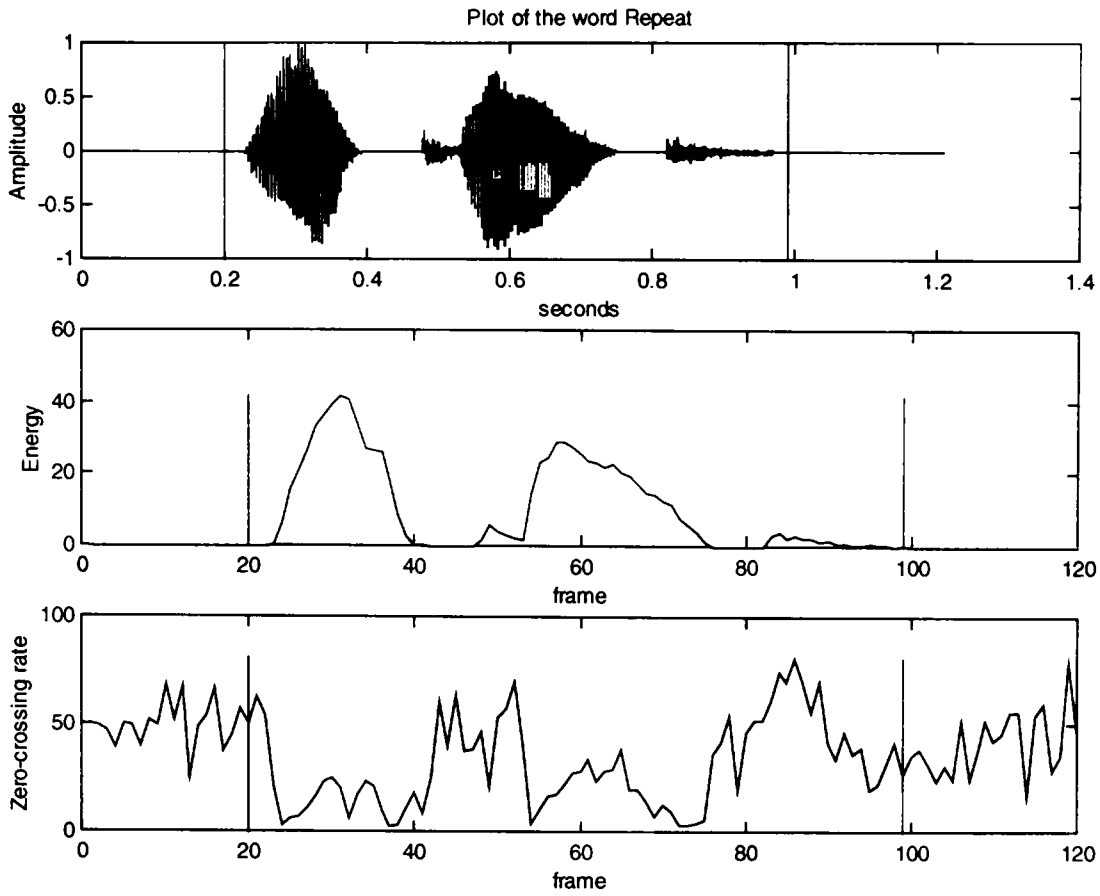

Figure A.11 Short-time energy and zero-crossing data for the word "Repeat" by a female speaker

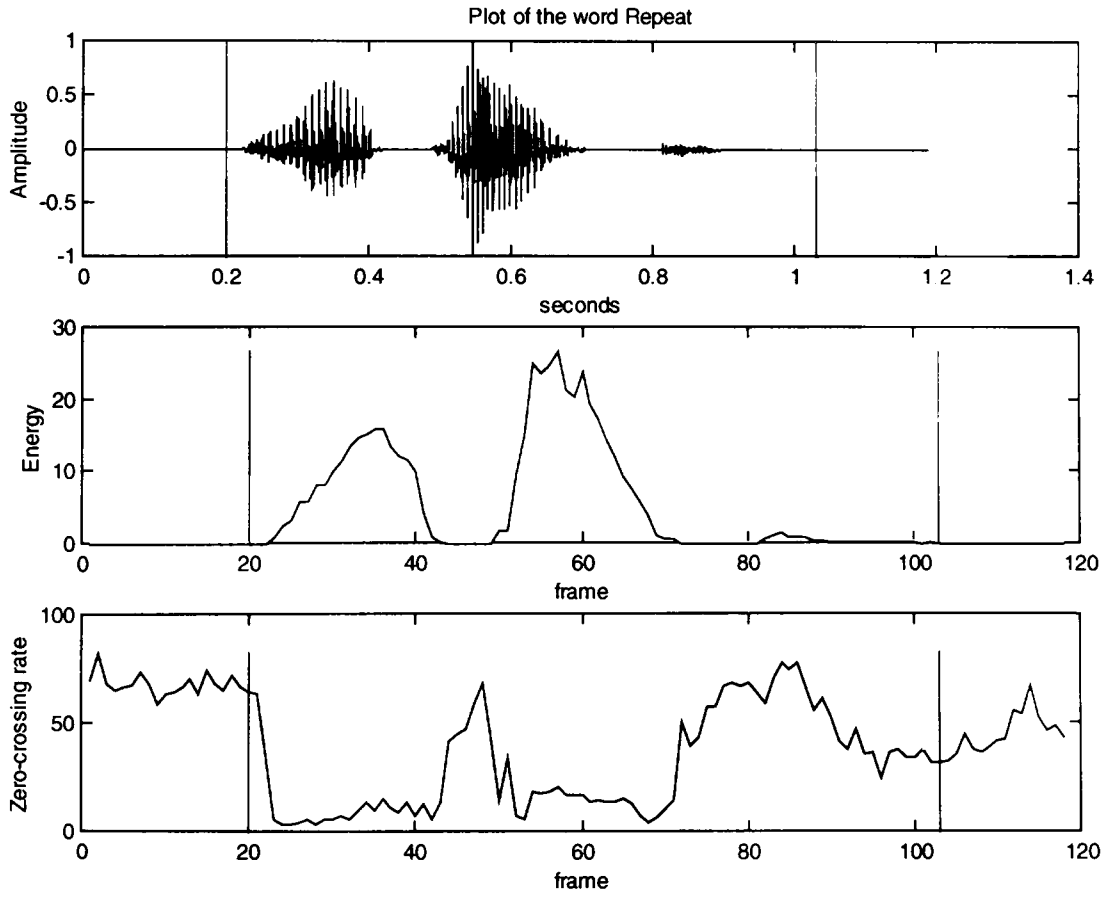

Figure A.12 Short-time energy and zero-crossing data for the word "Repeat" by a male speaker

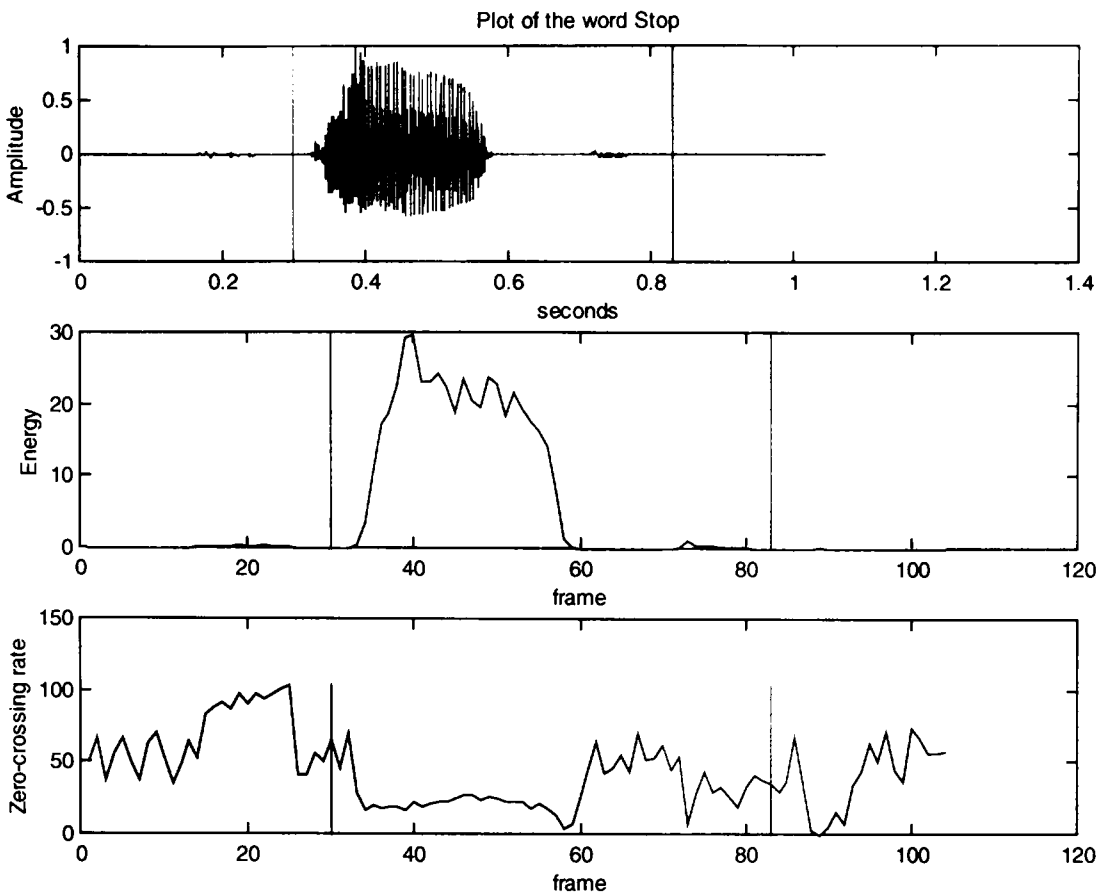

Figure A.13 Short-time energy and zero-crossing data for the word "Stop" by a female speaker

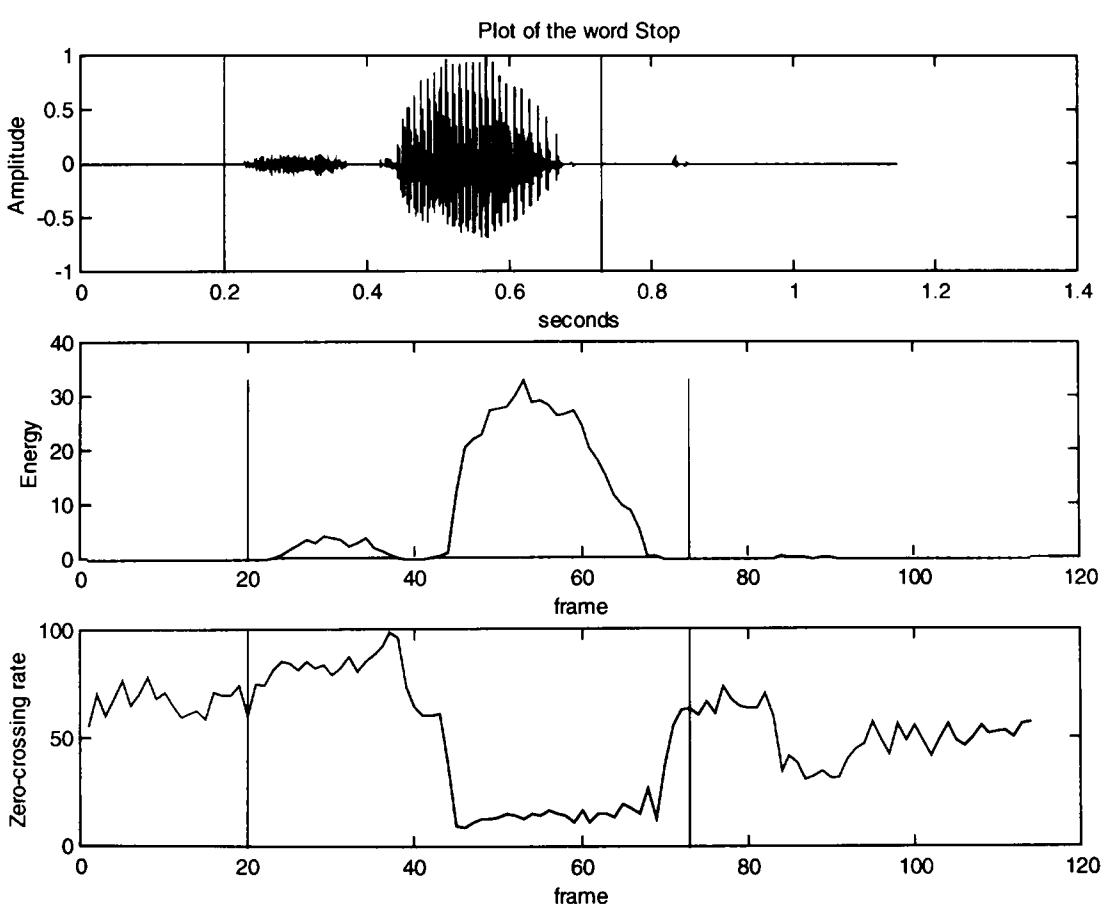

Figure A.14 Short-time energy and zero-crossing data for the word "Stop" by a male speaker

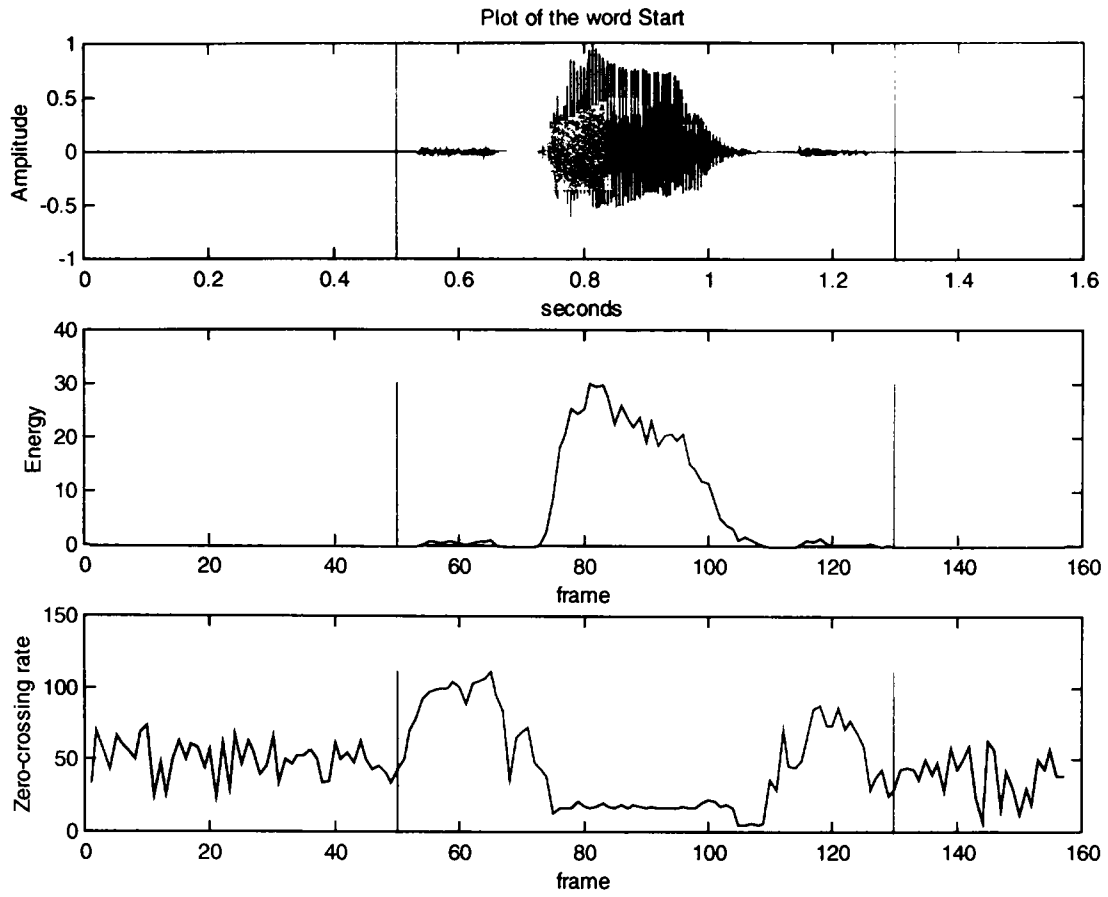

Figure A.15 Short-time energy and zero-crossing data for the word "Start" by a female speaker

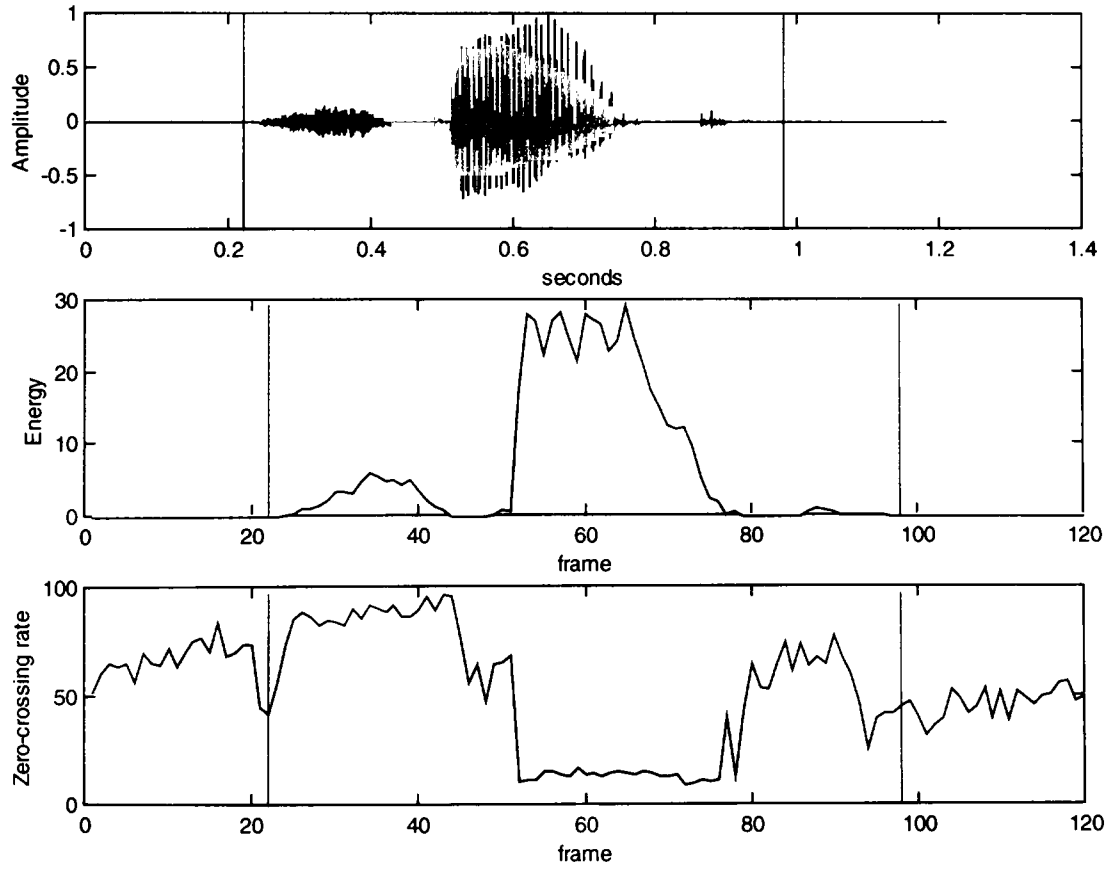

Figure A.16 Short-time energy and zero-crossing data for the word "Start" by a male speaker

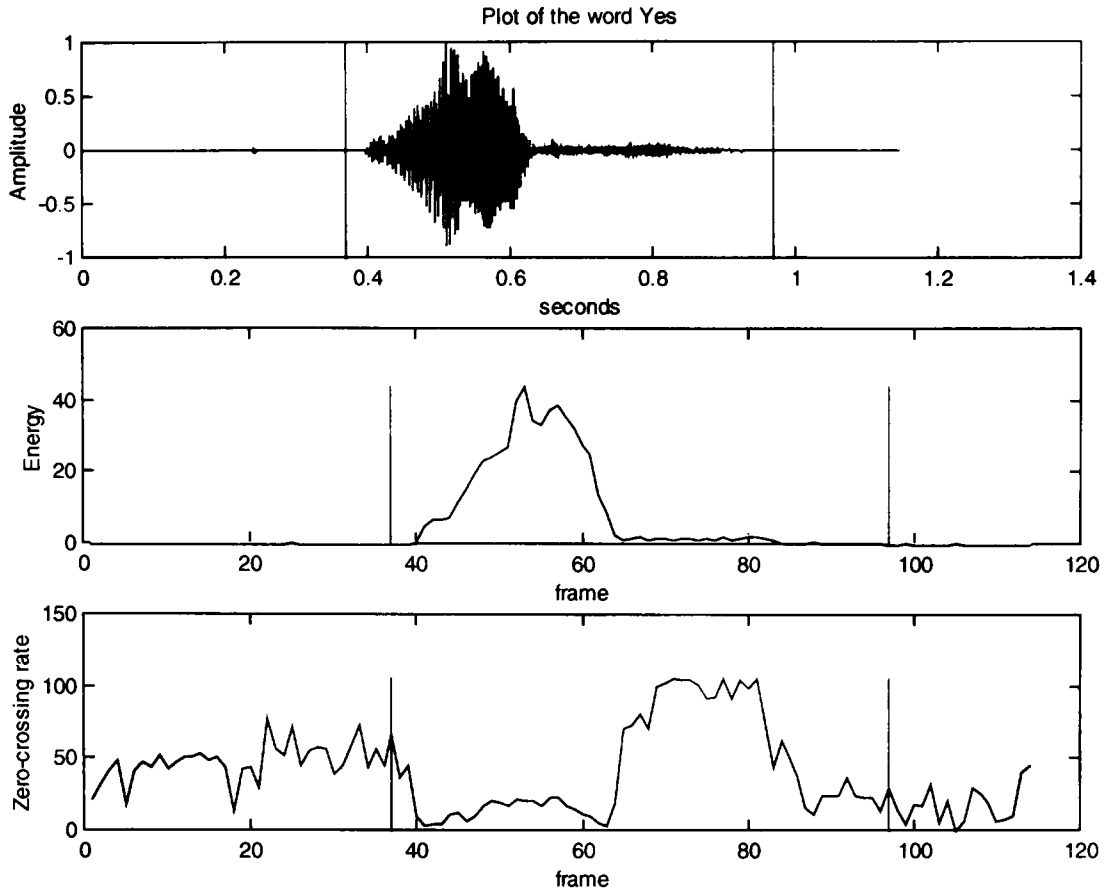

Figure A.17 Short-time energy and zero-crossing data for the word "Yes" by a female speaker

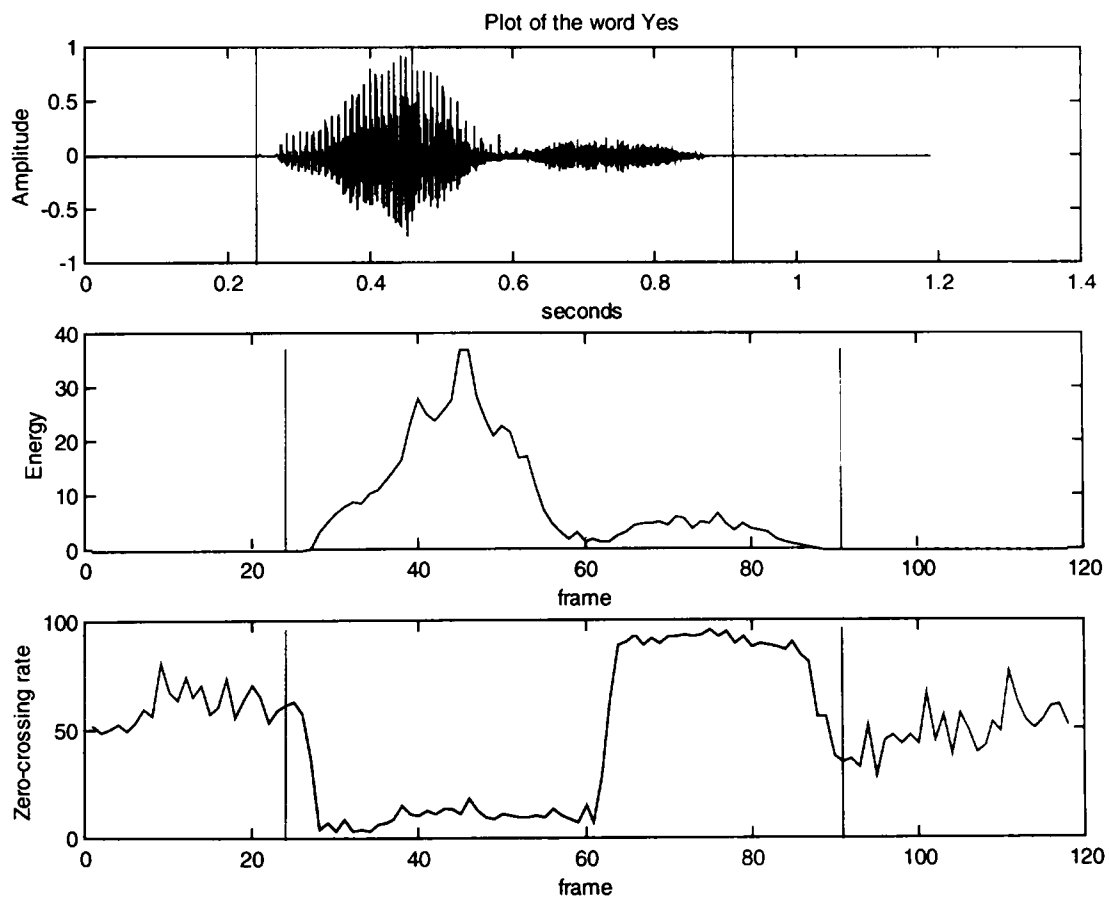

Figure A.18 Short-time energy and zero-crossing data for the word "Yes" by a male speaker

APPENDIX B

AVERAGE VALUES OF ZERO CROSSINGS AND ENERGY CONTENT

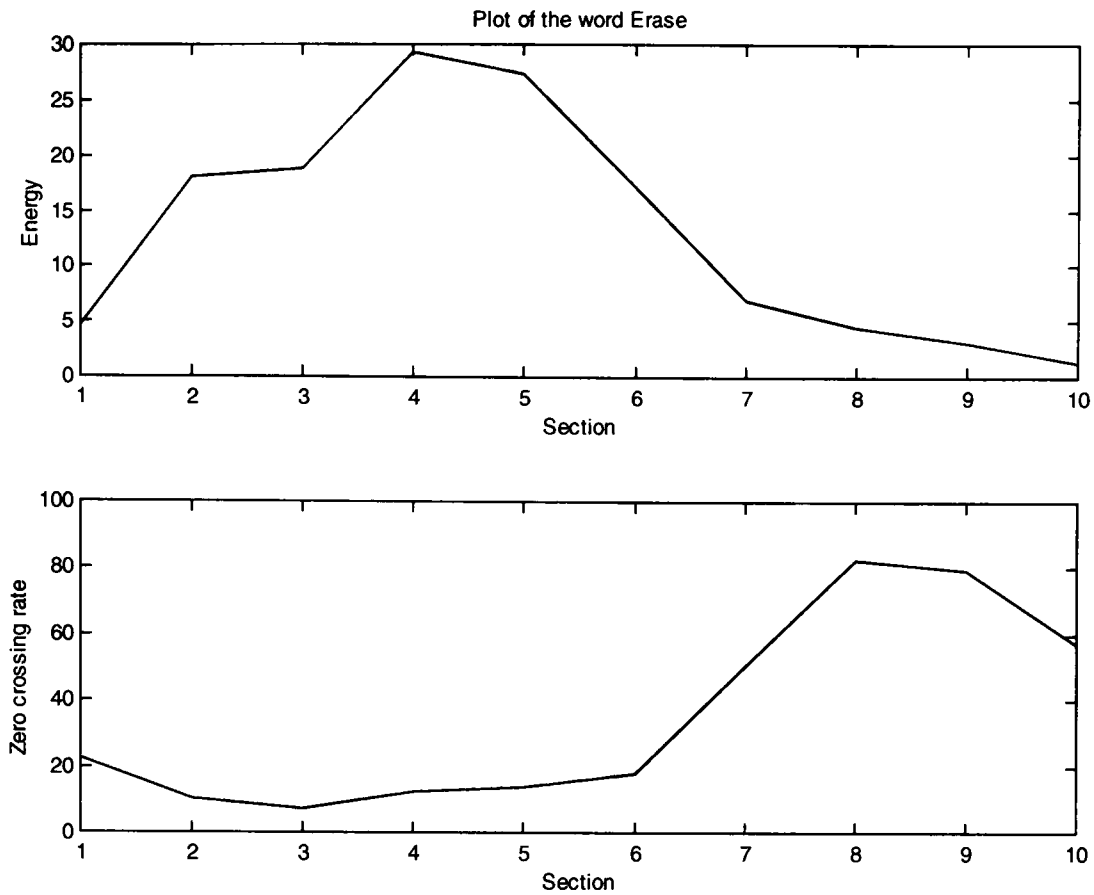

Figure B.1 Average values of zero-crossing and energy content for the word "Erase"

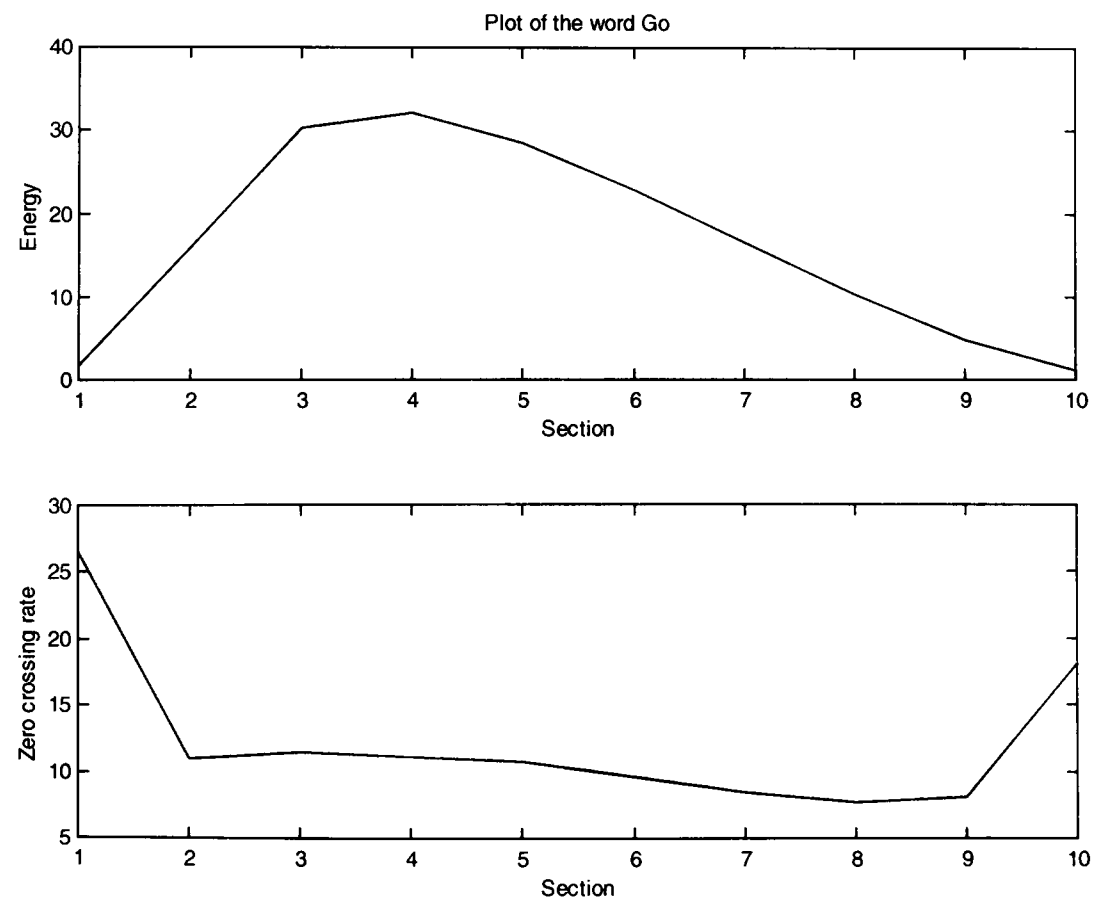

Figure B.2 Average values of zero-crossing and energy content for the word "Go"

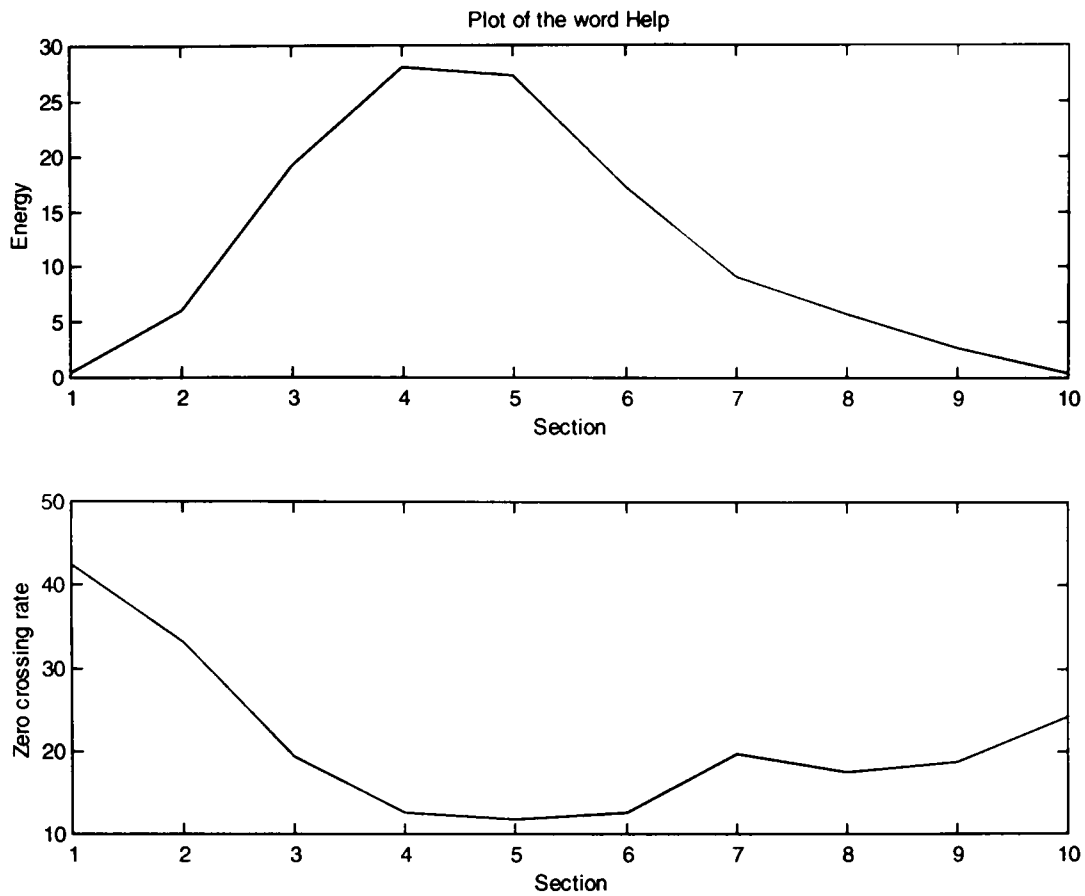

Figure B.3 Average values of zero-crossing and energy content for the word "Help"

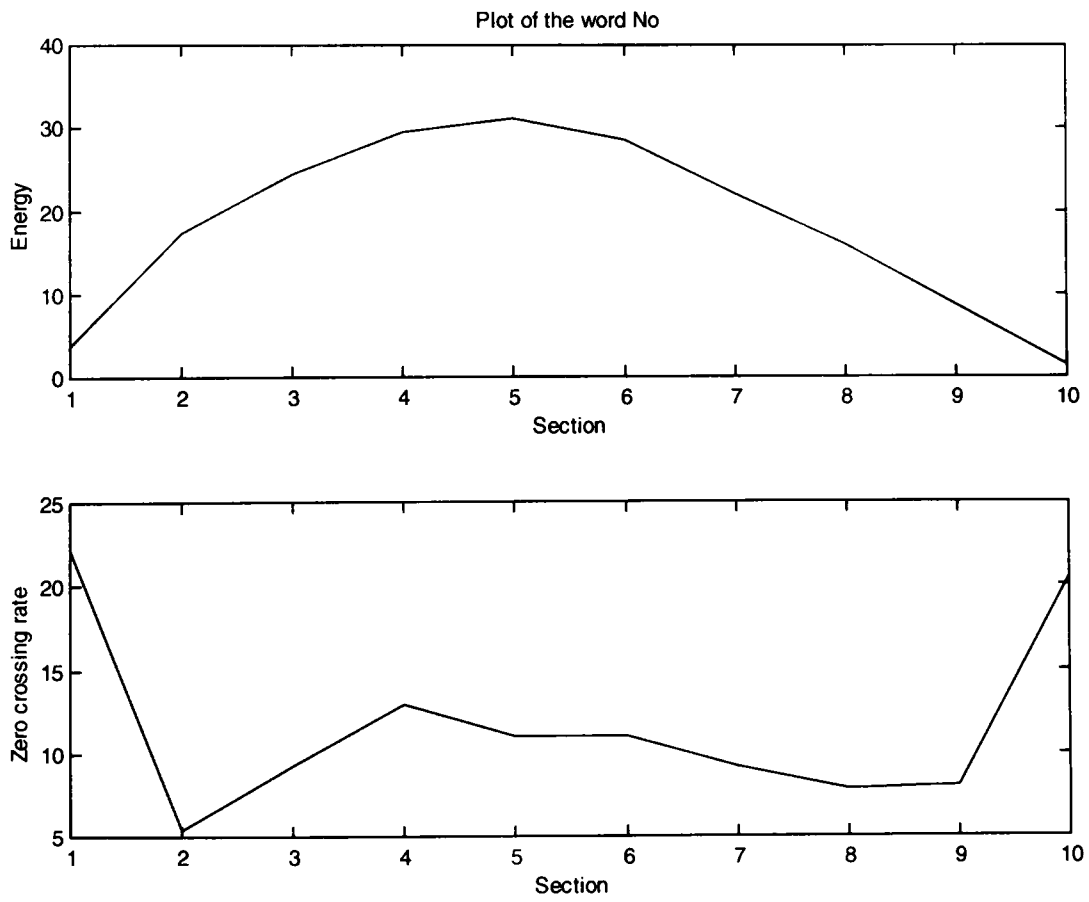

Figure B.4 Average values of zero-crossing and energy content for the word "No"

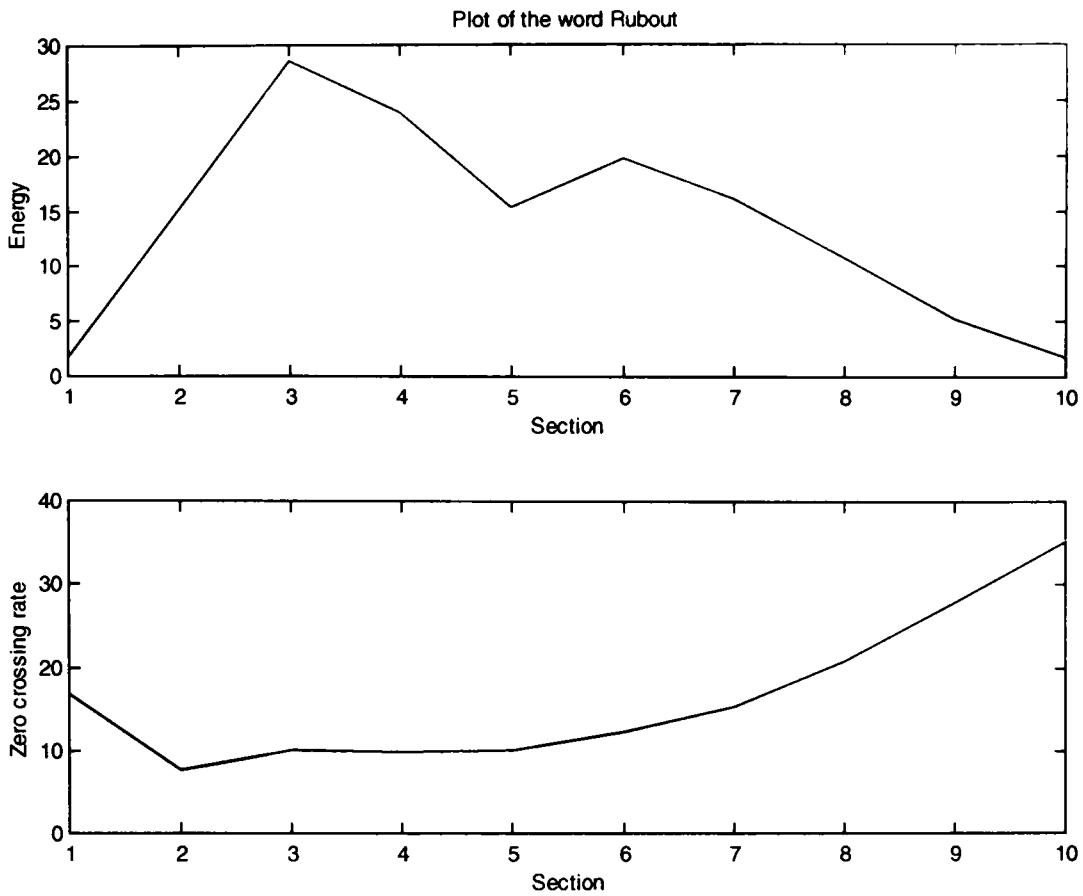

Figure B.5 Average values of zero-crossing and energy content for the word "Rubout"

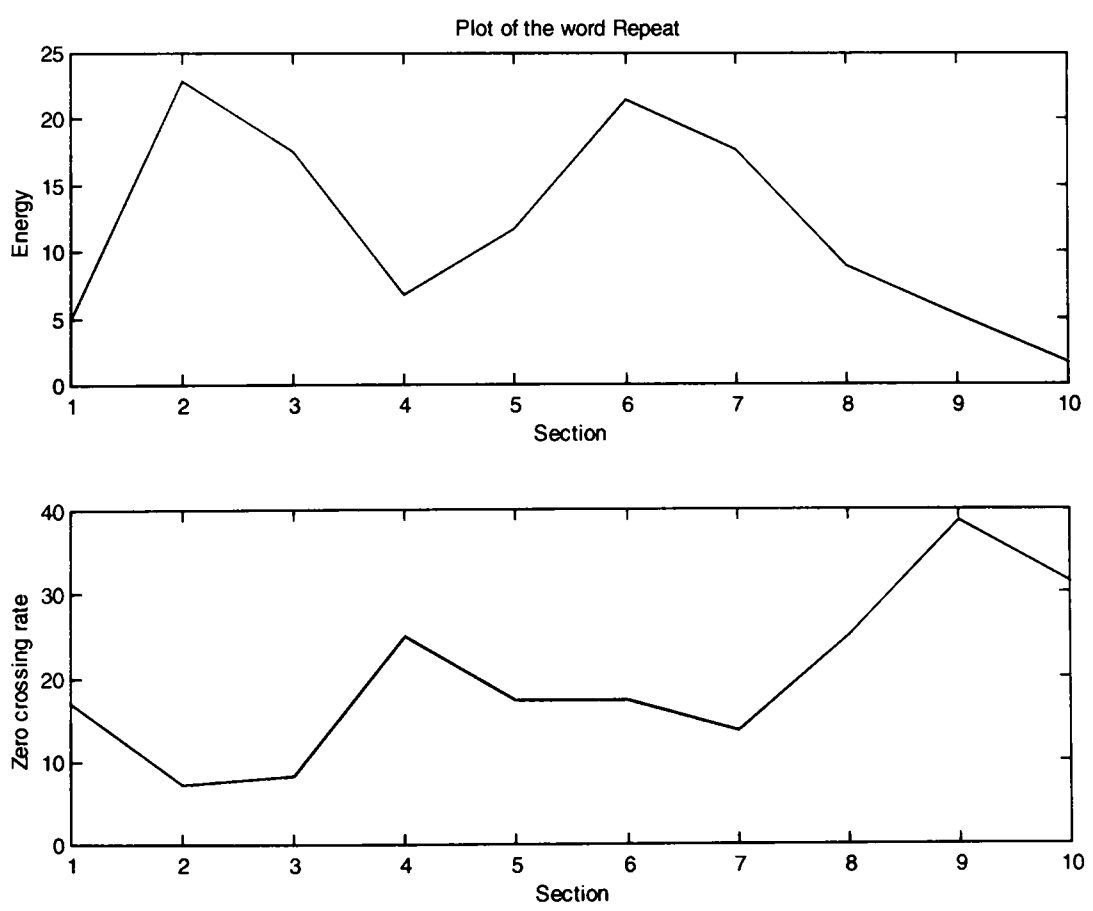

Figure B.6 Average values of the zero-crossing and energy content for the word "Repeat"

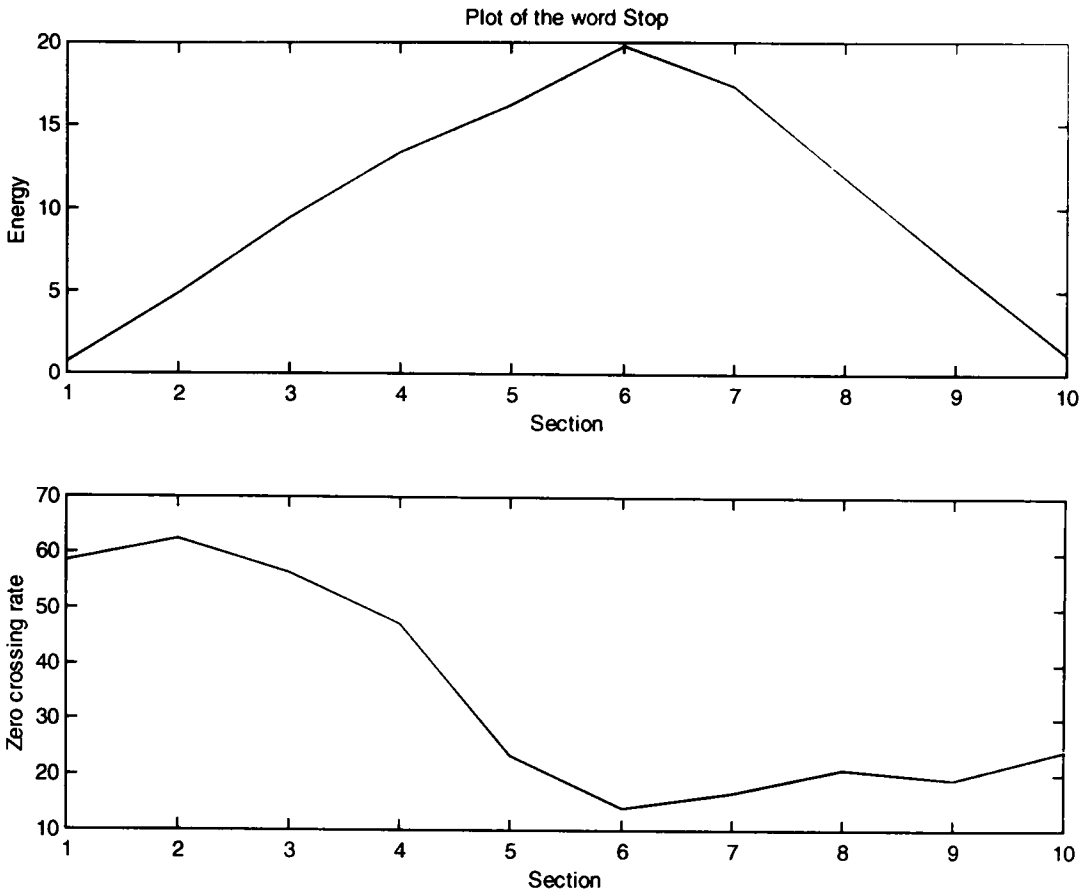

Figure B.7 Average values of the zero-crossing and energy content for the word "Stop"

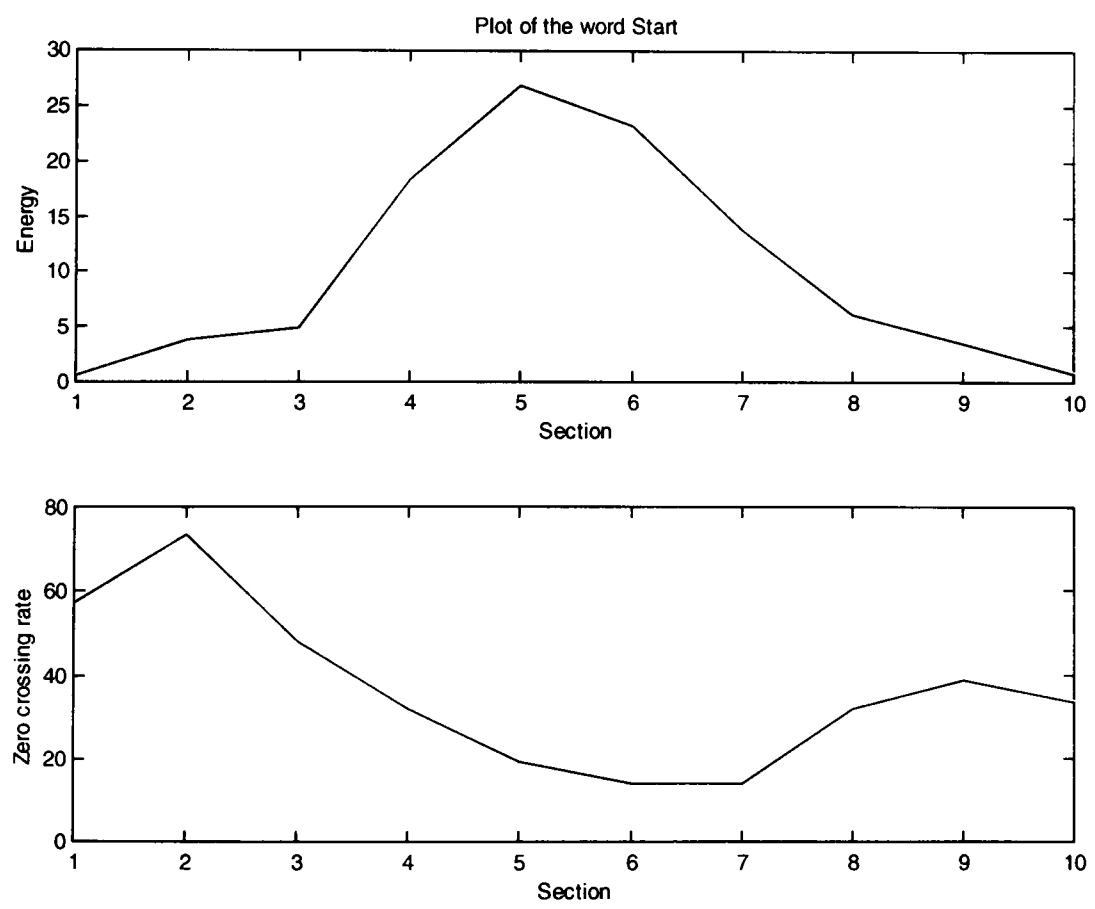

Figure B.8 Average values of zero-crossing and energy content for the word "Start"

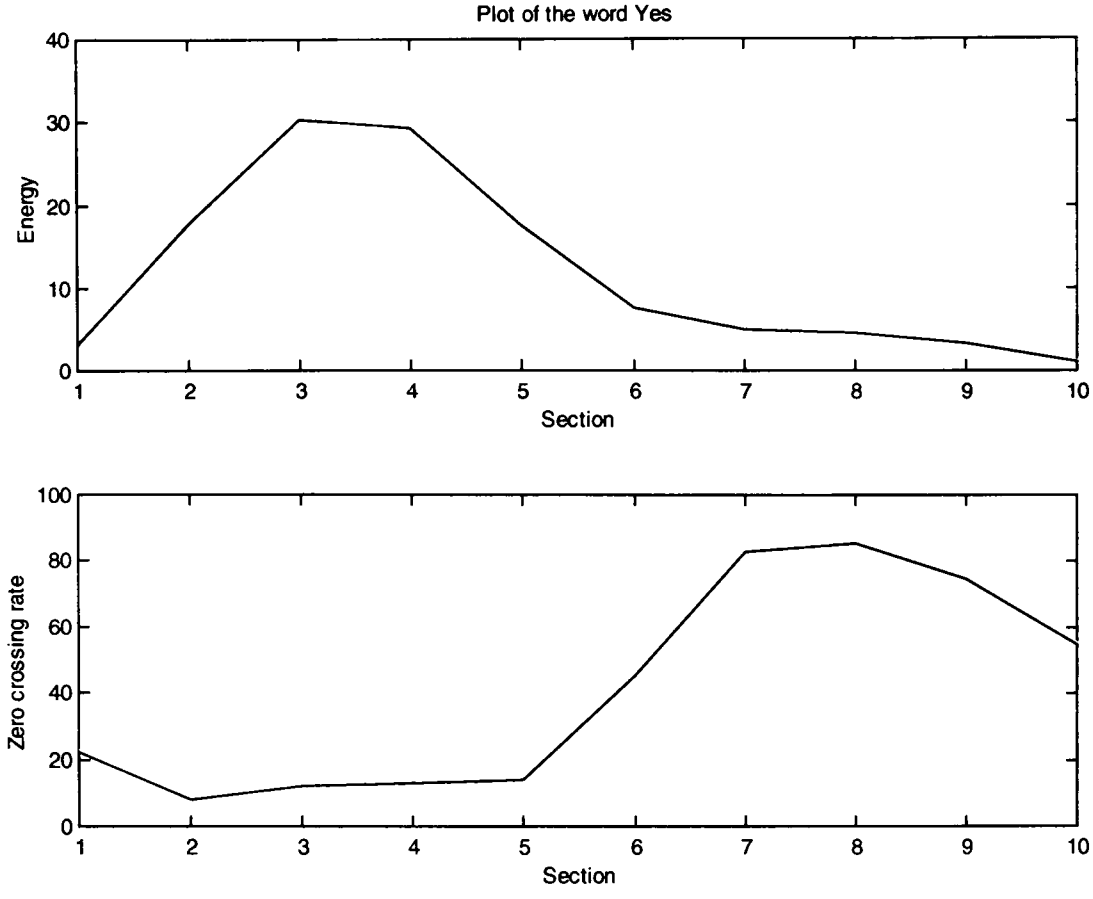

Figure B.9 Average values of the zero-crossing and energy content for the word "Yes"

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 _____