An Approach to
A Robust Speaker Recognition System

by

Michael Tran

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
in
Electrical Engineering

APPROVED:

Dr. Hugh F. VanLandingham, Chairman

William T. Baumann
Dr. William T. Baumann

Dr. Walling R. Cyre

Dr. John S. Bay

Dr. Frederick H. Lutze

December, 1994
Blacksburg, Virginia

An Approach to
A Robust Speaker Recognition System

by
Michael Tran
Department of Electrical Engineering
Dr. Hugh F. VanLandingham, Chairman

(ABSTRACT)

This dissertation presents a design of a robust, automatic speaker recognition (ASR) system. The ASR system is designed to work with both text-independent and text-dependent speaker recognition. Several speaker spectral features are studied to determine their contribution in term of accuracy to the system. A new algorithm is designed to label a speaker voice as either male-type voice or female-type voice. Following this division, the processing time of the speaker identification for the ASR system will be reduced by about half. Rectangular window, Hamming window, first order preemphasis filter, and many proposed spectral distances are also investigated. The principal components analysis is used to achieve high degree of female-type and male-type separation as well as the speaker recognition accuracy. Spectral features are combined to improve the recognition performance of the system. In addition, many other system components such as speech endpoint detection, automatic noise thresholds, etc. are required to build correctly in order to achieve high speaker recognition accuracy. Multi-stage decision process is used both to improve and to speed up the decision if certain criteria are met. Finally, TIMIT acoustic continuous speech corpus is used to evaluate the speaker recognition performance and the robustness of the system.

Acknowledgments

I would like to express my deepest gratitude and thanks to my advisor, Dr. Hugh F. VanLandingham, for his time, advice, support, and guidance throughout my M.S. and Ph.D. research at Virginia Tech. I am very fortunate to work for him in all my graduate years and have benefited greatly from his advice.

I would like to thank Dr. Walling R. Cyre for his valuable time to serve on both my M.S. and Ph.D. advisory committee. Dr. Cyre has given me much helpful advice.

I also would like to thank Dr. William T. Baumann, Dr. John S. Bay, and Dr. Frederick H. Lutze to serve on my advisory committee. Their time and suggestions toward my dissertation are sincerely appreciated.

Finally, I want to dedicate this dissertation to my dearest family. Without the love, constant support, and encouragement from my parents, my education would have never been possible. I also like to thank my friends for making my graduate years at Virginia Tech the most enjoyable ones.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Human's abilities both to understand the speech and to recognize the speakers from their voices have inspired and amazed many scientists to research in this field. Prior to the mid 1960's, most of the speech processing systems were based on analog hardware implementations. Since the advent of inexpensive digital computers in the early 1960's and pulse code modulation (PCM), the speech area has undergone many significant advances. Successful speech processing systems require knowledge in many disciplines including acoustic wave spectrum, pattern recognition, and artificial intelligence techniques. In general, speech technology includes the following areas : speech enhancement, speaker separation, speech coding, speech recognition, speech synthesis, speaker recognition, and language modeling/ identification. The area of speaker recognition can be divided into the speaker verification and the speaker identification. The transfer of speech technology from research laboratories to the commercial market is only the first step. Abundant challenges in speech research are still waiting to be conquered. With today's availably inexpensive, high quality sound boards, high quality microphones, speech databases, and fast computers, speech research is more accessible and has greater promise for the complete understanding of the fascinating human auditory system in the near future. This dissertation's main emphasis is on the speaker recognition problem.

1.1 Background and Motivation

Identifying a person is a requirement for controlling access to secure facilities, personal information (medical records), services like banking, credit checks, etc. Today, an average person may use many different security items such as PINs for automatic teller machines, phone cards, credit cards, and memberships. These can be lost, stolen, or counterfeited. Biometric systems are the automated methods for verifying a person's identity based on physiological characteristics like handwriting, fingerprints, and voice. Some techniques are expensive, and others are invasive. Voices of different individuals do not sound alike and may be the most natural and inexpensive biometric system to use for personal identity verification. Most automatic speaker recognition systems assume that the speakers are cooperative. Therefore, speaker recognition could be useful in many services, and can play a major role in preventing telecommunications fraud.

Speaker verification is one area of general speaker recognition which also includes speaker identification. *Speaker verification* is used to determine whether an unknown utterance was spoken by the claimed speaker. On the other hand, *speaker identification* is the labeling of an unknown utterance among utterances of *known* speakers. Speaker verification can be sub-divided into text-dependent and text-independent speaker verification. In *text-dependent* speaker verification, the task is to verify the same utterance both in training and later in testing, where the utterances in training and in testing are not necessarily the same for *text-independent* speaker verification.

Speaker differences that both enable and hinder speaker recognition include interspeaker and intraspeaker variations. Interspeaker variations, i.e. between speakers are due to the physical aspect of differences in vocal cords and vocal tract shape, and to the behavioral aspect of differences in speaking styles among speakers. Intraspeaker variations are the differences in the same utterance spoken by the same speaker : speaking rate, his emotional state, his health, etc. Variations in voices translate to variations in acoustic parameters. Good speaker recognition system should capture these variations. Therefore, it is desirable to select those acoustic features that have the following characteristics [1] :

- High interspeaker and low intraspeaker variabilities.
- Easy to measure and reliable over time.
- Occur naturally and frequently in speech.
- Stable in different transmission environments.
- Difficult to imitate.

Decades of research have determined several useful features as functions of time such as pitch, speech intensity, formant frequencies, nasal coarticulation, linear predictive coefficients (LPC), LPC derived coefficients, and many pattern matching techniques.

1.2 Survey of Previous Work

Many papers and textbooks have described and proposed many different techniques to extract speaker features and to build automatic speaker recognition (ASR) systems with different assumptions and environments [1] - [129]. Most ASR systems use either the template matching method or probabilistic modeling of the features of the speakers. In the template matching method, the reference template of the claimed speaker created during the training phase is compared with the unknown template. Probabilistic models employ long-term statistical feature averaging. In general, signal processing front ends are used to detect speech endpoints, and to convert raw speech signal into parametric spectral features as function of time for later processing. Speech spectral features include the filter bank outputs, pitch, intensity, short-time spectrum, formant frequencies, bandwidths, nasal coarticulation, spectral correlations, timing, speaking rate, linear predictive coefficients (LPC), LPC orthogonal parameters, LPC cepstrum, a host of LPC derived coefficients, and real cepstrum. Next, dynamic time warping (DTW), dynamic frequency warping, pattern comparison, normalization, vector quantization, hidden Markov models (HMM), and a host of spectral dissimilarity distances are employed to identify the speaker from these parametric acoustic feature contours. Neural network technology together with recent acoustic theory has been applied to extract speaker-dependent speech features in text-independent speaker recognition, etc. Time-delay neural networks have been used successfully in both speech recognition and speaker recognition. Clustering techniques using neural networks are used for matching speaker features.

Most automatic speaker recognition (ASR) systems employed from simple to sophisticated endpoint detection algorithms to find endpoints of an utterance [2], [21], [53], [70], [83,84,85,103,113,120,124]. Simple endpoint detection uses the short-term energy and zero-crossing rate algorithms. Sophisticated endpoint detection employs the short-term energy, pattern comparison, adaptive level quantization, energy pulse detection, and decision rules for a noisy environment. Filter bank outputs, as functions of time spanning the useful frequency range employed by the ASR system, are used to approximate the short-term Fourier spectrum which provides a complete description of the acoustical characteristics of speech [19], [22]. Short-term analysis, formants, pitch, and cepstrum together with two stage statistical measurements and minimum risk classification are presented in [5], [10]. Pitch and intensity contours related to the speaker's glottal source [13], [95], long-term parameter averaging of pitch, gain, reflection coefficients [57], and fundamental frequency contours [40] are effective spectral features in ASR systems. Coarticulation of nasal consonants was found to give more reliable clues than the nasal spectrum alone, and has been used to identify speakers [112].

Linear predictive parameters and their derived parameters related to the speaker's vocal tract are the most important features to date. By an appropriate eigenvector analysis of the LPC parameters, a set of orthogonal parameters are obtained that are highly indicative of the speaker identity [98], [99], [107]. Orthogonal parameters together with adaptive noise cancelling algorithms are used to achieve high accuracy in noisy environments [5]. Other LPC derived methods include LPC cepstrum phonetic based methods [102], LPC instantaneous and transitional spectral information [110], LPC cepstral coefficients [4], [6], [117], cepstral coefficients expanded by an orthogonal polynomial [29], LPC derived principle spectral components (PSCs) [64], [66], LPC derived parameters with frequency warped spectral distance [72], mel-scale warped cepstrum designed to place less emphasis on high frequencies before taking the inverse FFT [32], and LPC derived parameters with hidden Markov models [101], [126]. All LPC techniques are operated under the assumption that the speech is stationary within a short period of time (frame interval). This is a relatively good assumption, but at times can be inaccurate, depending on the speaker's utterance.

Vector quantization representing spectral features [54], [93], [109], many spectral dissimilarity distance measures [41], [86], [106], [117], [123], statistical methods [54], [78], perceptual based features [123], clipped autocorrelation function [71], and artificial neural network (ANN) methods help to improve ASR accuracy. Dynamic time warping and normalization algorithms using dynamic programming techniques are employed to stress, compress, and align acoustic contour patterns based on allowable paths (heuristic base) and minimum dissimilarity distance [86]. ANN techniques, paradigms, and algorithms including preditive neural networks (PNN), self-structuring hidden control (SHC) models, self-structuring Pi-Sigma (SPS) neural models, time-delay neural networks (TDNN), clustering techniques using neural networks, and others are presented in [17], [37], [47], [48], [49], [51], [77], [90], [111], [127].

1.3 Objectives and Scope of Research

The objectives of this research are to investigate how unique an individual speech features from others, to extract speaker features, and ultimately to design a robust, effective, and reliable real-time text-independent/ text-dependent speaker recognition system. The system performance goal is to discriminate between and among speakers, and stable over time. The following main issues considered in this research are :

- How to extract and select effective speaker features.
- How to speed up the speaker identification process.
- How to design and build a robust ASR system with high recognition accuracy.
- System performance assessment.

1.3.1 Scope of the research

The scope of this research is described by the following tasks :

- To study the effect of human vocal tract.
- To design a robust ASR system.
- To achieve high recognition accuracy.
- To select and evaluate the effectiveness of individual spectral feature.
- To propose a highly accurate speaker recognition algorithm.
- To build a robust ASR system with high recognition accuracy.

1.3.2 Research Approaches

Spectral features of the sampled speech signal can be obtained by the following methods :

- Short-term Fourier transform.
- Linear predictive coding (LPC) coefficients.
- LPC derived coefficients : PARCOR, log area, cepstrum, ...

- Filter banks and wavelet transforms.
- Least squares coefficients.

In this research, the linear predictive coding (LPC) **a**, the LPC PARCOR **k**, and the LPC cepstrum **c** are employed to represent the spectral features of speech. Linear predictive coding method models the present speech signal as linear combination of the past values of the speech signal. This all-pole transfer function are also used to model the human vocal tract. Background noise and equipment variations are evaluated carefully to generate noise and speech thresholds.

The robustness of the ASR system can be accomplished by a method of labeling a voice as either female-type or male-type. Many different spectral distances are used to measure the dissimilarity between two spectral patterns of speech signal. Special speech endpoint detection is used to enhance the recognition accuracy. Long-term statistics of spectral features are employed. Principal components analysis is also employed to enhance the female-type / male-type voice as well as the overall recognition accuracy. The weighted combination of spectral features also improves the recognition rate significantly.

1.4 Contribution and Work Organization

The following contributions are resulted from the research work :

- A robust ASR system with high recognition rate is successfully built with 94.4 % speaker identification accuracy and 97.6% speaker verification accuracy on 462 speakers at 8 kHz sampling frequency.
- A new combination of features are proposed to improve the recognition rate.
- A new algorithm is designed to separate male-type voices from female-type voices successfully. Following this division, the speaker identification time is reduced by about half.
- Simple and effective speech endpoint detection algorithm for this ASR system.

- Many useful analysis of spectral features, windows, preemphasis filters, spectral distances, visual clusters, and paramters are presented.

The research work is divided into seven chapters. Speaker recognition background, the motivation, a survey of previous research works, the objectives and scopes of the research are presented in chapter 1. Chapter 2 describes the fundamentals of human speech production system, human auditory system, and speech phonemes together with speech sound classification. Some speech signals and their spectrograms are used to illustrate the raw speech signal and the quasi-periodic nature of speech signals. Chapter 3 provides some important speech processing techniques which are derived from many interdisciplinary areas such as digital signal processing, pattern recognition, and neural network technology. Signal processing techniques are used to process the speech signals and to extract their features. Speech spectral clustering, pattern matching, pattern comparison on the basis of feature contours are the pattern recognition techniques. Neural network techniques are also employed to extract speaker-dependent features and to recognize among the speakers. Chapter 4 described the sound components and evaluations and the system components design overview : sound board specifications, room noise and equipment evaluation, choice of windows, window length, and the speech endpoint detection. The automatic speaker recognition design is presented in Chapter 5. In Chapter 5, methods of extracting speaker features, building the reference speaker feature database, and computing moments of a distribution are described in details. Several ASR design approaches using long-term average statistics of spectral features and different spectral distances are evaluated. In Chapter 6, an algorithm for female-type / male-type voice separation is designed to reduce the identification time by about half. The final ASR design is presented and its performance is evaluated. Chapter 7 is the conclusion and recommendations.

Chapter 2

Fundamentals of Speech Science

To communicate with other people, a speaker must formulate his message into words of their common language and then execute these words through a series of neuromuscular movements to produce acoustic sound waves via the human speech production system. The listener's auditory system receives and translates these radiating acoustic sound waves back into neurological signals that can be understood by their brains. The acoustic waves also provide the essential feedback to the speaker's auditory system to monitor his own speech. This feedback and repetition at the beginning of the learning stage are the keys in helping a person to learn to speak correctly and to reinforce his or her learning process.

2.1 Human Speech and Auditory Systems

The human speech production system is a complex mechanism to produce speech sounds. It basically consists of pharynx (throat) cavity, nasal cavity, oral cavity, vocal cords, velum, and lung as shown in Figure 2.1 [20]. The oral cavity and throat cavity are referred to as the vocal tract which plays a major role in producing unique speech sounds so that one can identify a person from his voice. However, sometimes nasal coupling produced by nasal cavity can significantly modify the frequency characteristics of the acoustic sound waves in such way that it is impossible to recognize a person from his voice.

The human auditory system used to process speech is shown in Figure 2.2. The ear has three different regions : the outer ear, the middle ear, and the inner ear. The outer ear consists of the pima and the auditory canal that leads to the eardrum. In the middle ear, the small bone (malleus or hammer) attached to the eardrum makes contact with another

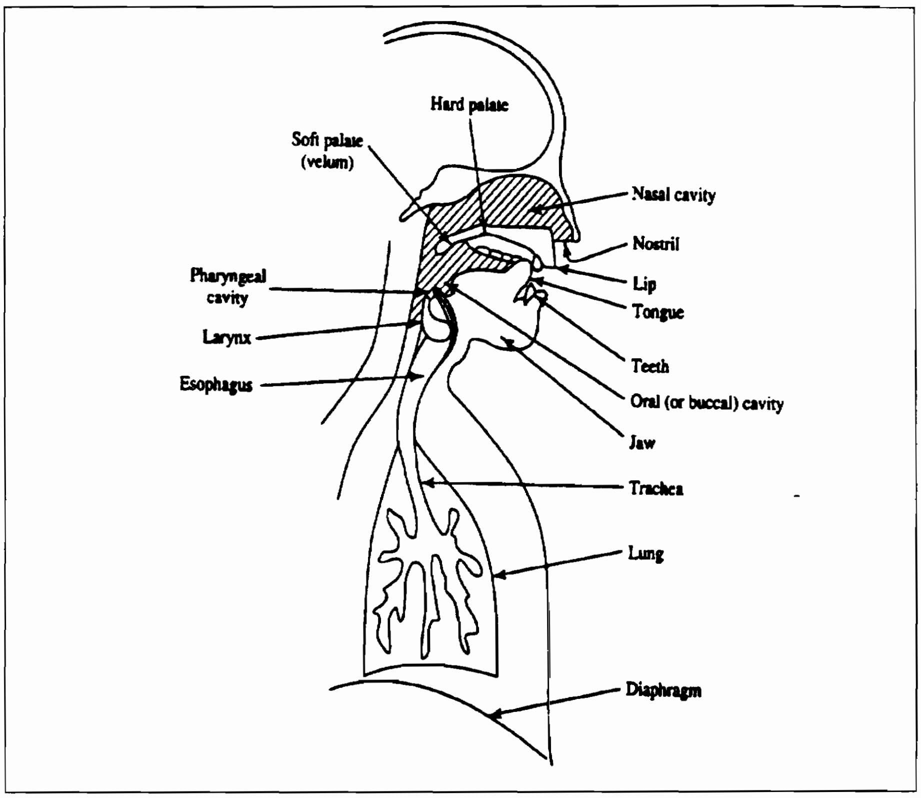


Figure 2.1 Human speech production system.

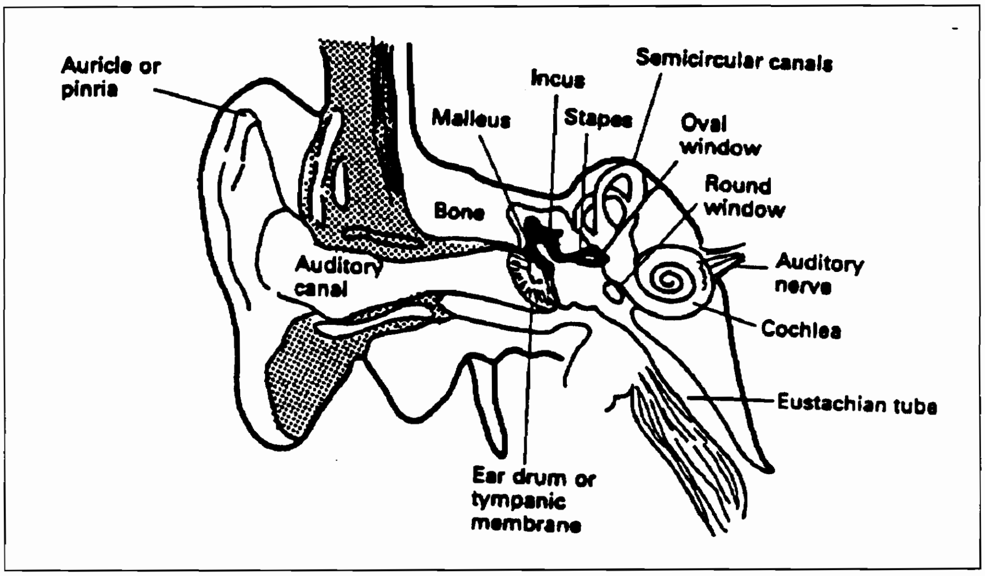


Figure 2.2 Human auditory system.

bone (anvil or incus) to transmit the vibrations of the eardrum to the oval window of the inner ear. The outer ear is used to guide the sound waves to the middle ear impinging upon the eardrum to make it vibrate. The inner ear consists of the cochlea. The fluid-filled cochlea is separated by the basilar membrane and the auditory nerve. The oval window vibrations result in pressure waves that propagate through the cochlear fluid and cause the basilar membrane to deflect at different points along its length. The basilar membrane can be characterized by a set of frequencies at different points along the membrane. The cochlea can be modeled as a bank of filters. Inner hair cells along the basilar membrane are sensors which convert mechanical motion to neural activity. Each inner hair cell is connected to about 10 nerve fibers. There are about 30,000 nerve fibers total, linking the inner hair cells to the auditory nerve which is connected to the brain where speech perception is performed [86]. The human auditory system has the ability to selectively listen to a particular speaker's voice by the mismatch of the arriving sounds.

2.2 Speech Production Model

The speech signal can be produced by exciting the vocal tract with either quasi-periodic pulses, random pulses or mixed pulses. Glotal pulses (periodic pulses) are used to produce voice sounds. Random pulses and mixed pulses produce unvoiced and plosive sounds. The vocal tract shape is known to change slowly with time to produce different sounds and at any time can be characterized by its formant frequencies. Therefore, the vocal tract can be modeled as an all-poles time-varying discrete-time filter.

$$H(z)=\frac{G}{1-\sum_{n=1}^{P}a_n z^{-n}}=\frac{G}{\prod_{n=1}^{P}(1-p_k z^{-1})} \tag{2.1}$$

The speech waveform can be divided into segments of stationary signals. The short-time Fourier transform is used to find the spectrum of each speech segments. A typical discrete-time model and a simple model of the speech production system are shown in Figures 2.3 and 2.4. This model is an over simplification of the human speech production system since there will be some coupling effect with the nasal cavity.

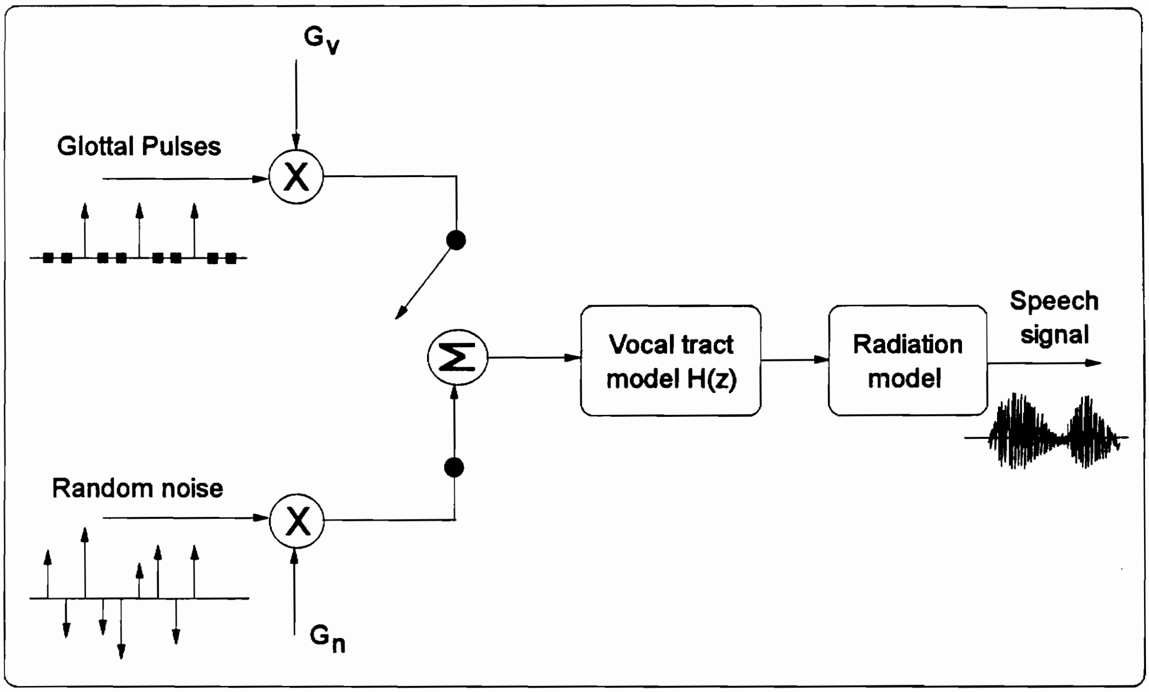


Figure 2.3 A discrete-time model of the speech production system.

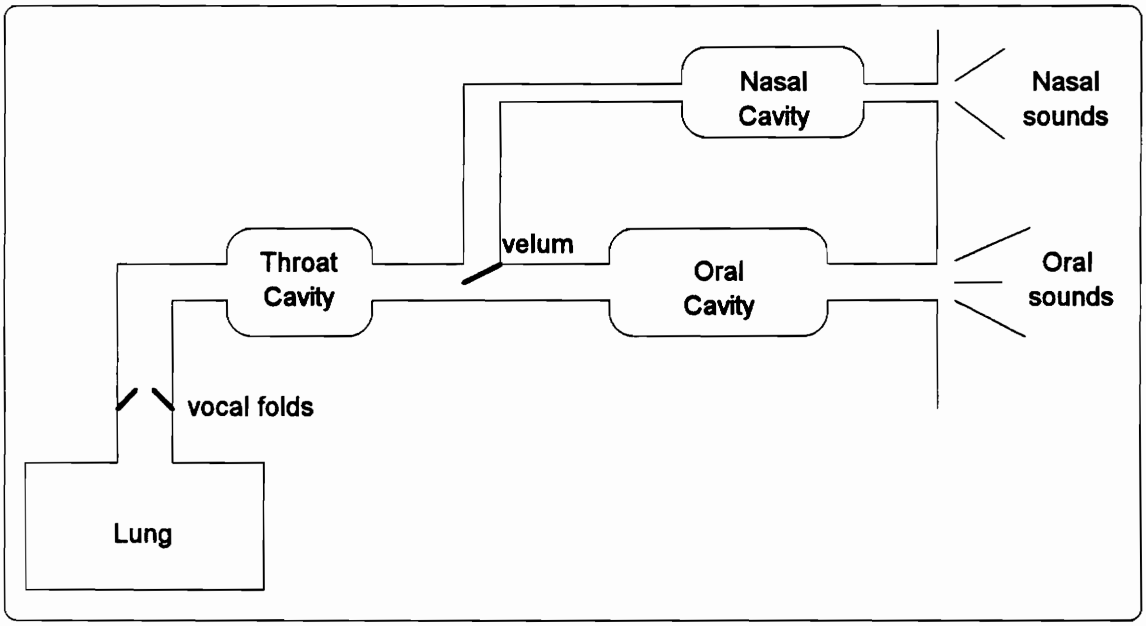


Figure 2.4 Simple model of human speech production system.

2.3 Phonemes and Sound Classification

Phonemes can be considered as fundamental units used to pronounce a word. The single-symbol version of ARPAbet (Advance Research Project Agency) of about 47 phonemes in American English made up of vowels, semivowels, diphthongs, and consonants (nasals, stops, fricatives, affricates) are presented in Table 2.1.

Table 2.1. Single-symbol version of ARPAbet for American English [20].

Symbols	Examples	Symbols	Examples	Symbols	Examples
i	heed	X	roses	m	mad
I	hid	p	plot	n	nice
e	bait	b	but	G	sing
E	head	t	time	l	love
@	had	d	deep	L	cattle
a	nod	k	kick	M	some
c	bought	g	guy	N	son
o	boat	f	five	F	batter
U	hood	v	vice	Q	(glottal stop)
u	boot	T	thing	w	win
R	bird	D	then	y	yard
x	ago	s	sign	r	run
A	mud	z	zoo	C	church
Y	hide	S	show	J	judge
W	down	Z	azure	H	when
O	boy	h	help	ju	you

These phonemes can be divided into 12 vowels, 4 diphthongs, 4 semivowels, and 27 consonants as shown in Figure 2.5 [86].

Speech sounds are produced by exciting the vocal tract with a wideband excitation. In general, speech sounds can be classified into three broad sound categories :

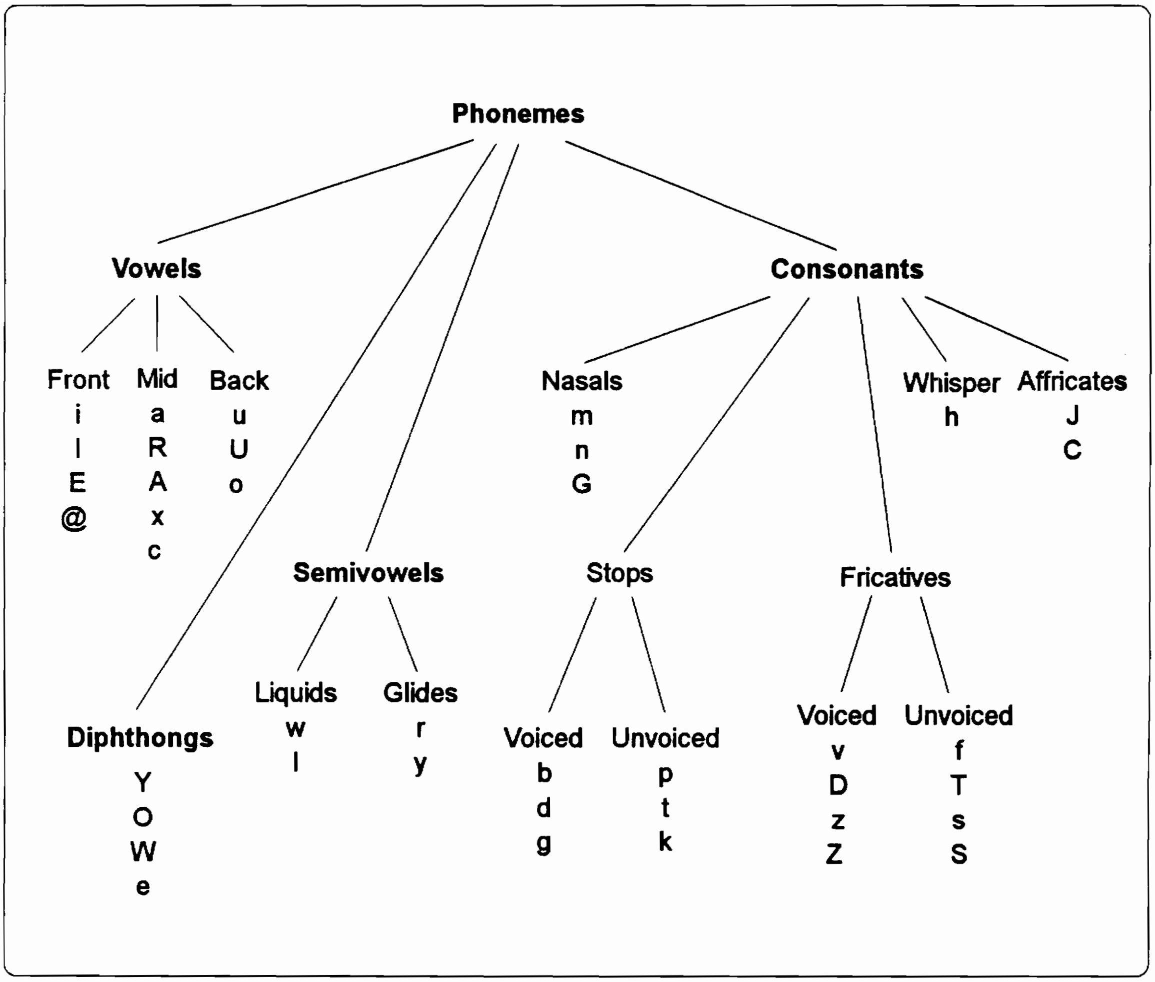


Figure 2.5 Chart of phonetic classification of American English.

- Voiced sounds are produced by exciting the vocal tract with quasi-periodic airflow pulses.
- Fricative sounds are produced by exciting the vocal tract with steady airflow that becomes turbulent at some point.
- Plosive sounds are produced by building up the air pressure behind the vocal tract followed by a suddenly release of the pressure.

The speech sound spectrum represents a slowly time-varying (nonstationary) signal that can be divided into stationary sound segments over a short period of time. Each sound segment possesses similar acoustic properties. This leads to the classification of phonemes constructed from their properties related to the time waveform, frequency characteristics, manner of articulation, place of articulation, and the type of excitation [20].

2.3.1 Vowels

There are 12 vowels in the phonemes of American English consisting of 4 front vowels, 5 mid vowels, and 3 back vowels. Vowels are produced by exciting the vocal tract with quasi-periodic pulses of airflow through the vibration of vocal folds. The different vowel sounds can be determined from the positions of the tongue, jaw, and lips. In general, the vowel waveforms have longer time-duration and larger amplitude than the consonant waveforms. Due to quasi-periodic excitation, different vowels can be determined by the first three formant frequency locations of their spectrogram plots. The fourth and higher formant frequencies remain relatively constant regardless of changes in articulation. Table 2.2 shows the average formant frequencies F1, F2, and F3 of typical vowels. Furthermore, the average bandwidths of formant frequencies also help to contribute to the recognition of the vowels. From Table 2.2, the formants of the front vowels occur at high frequency band, the formants of the back vowels locate at low-frequency band, and the formants of the mid vowels locate in between. However, the measured formants for a given vowel sound can vary greatly among different speakers, and it is not easy to determine exactly their formant peaks. Also, the phoneme boundaries together with allophones, breath noise, and background noise can cause problems in classifying the vowel sounds.

Table 2.2. Average formant locations for vowels [86].

Vowels	F1 (Hz)	F2 (Hz)	F3 (Hz)
/i/	270	2290	3010
/I/	390	1990	2550
/E/	530	1840	2480
/@/	660	1720	2410
/a/	730	1090	2440
/c/	570	840	2410
/U/	440	1020	2240
/u/	300	870	2240
/A/	640	1190	2390
/R/	490	1350	1690

2.3.2 Diphthongs

A diphthong is a transitional sound produced when the vocal tract starts at the articulatory position of one vowel and ends in the position of another vowel. There are four diphthongs in American English : /Y/ in buy, /W/ in down, /O/ in boy, and /e/ in bait. It is difficult to distinguish a diphthong from a sequence of two vowels

2.3.3 Semivowels

Semivowels consisting of /w/, /l/, /y/, /r/ are vowel-like sounds. Semivowels are classified into liquids (/w/, /l/) and glides (/r/, /y/). Liquid semivowels possess spectral characteristics similar to vowels, but they are weaker in amplitude due to their more constricted vocal tract. Glide semivowels are generally characterized by a transition sound in vocal tract between adjacent phonemes. Thus, the frequency characteristics of the semivowels may vary significantly depending on the context in which they occur.

2.3.4 Consonants

The consonants consist of the nasal consonants, the voiced stop consonants, the unvoiced stop consonants, the voiced fricative consonants, the unvoiced fricative consonants, the whisper consonant, and the affricate consonants.

Nasal consonant sounds consisting of /m/, /n/, and /G/ are another class of steady-state voiced speech. Nasal consonant sounds are produced by the airflow pulses through the open nasal cavity with the closed mouth cavity. They can be characterized by their energy, the broader F1 formant bandwidths, and the zeros (nasalization) induced into the vocal tract transfer function. Their waveforms are similar to those of vowels with weaker amplitudes. Thus, formant frequency peaks of the nasal consonants and nasalized vowels are not well-defined. For /m/ the constriction is at the lips; for /n/ the constriction is behind the teeth; and for /G/ the contriction is just forward of the velum.

The unvoiced fricative consonants consisting of /f/, /T/, /s/, and /S/ are produced by exciting the vocal tract with a steady airflow that becomes turbulent at a constriction. For the unvoiced fricatives, the constriction is near the lips for /f/, near the teeth for /T/, near the middel of the oral tract for /s/, and near the back of the oral tract for /S/.

The voiced fricative consonants consisting of /v/, /D/, /z/, and /Z/ are produced by vibrating the vocal tract with mixed excitation which becomes turbulent at some point of constriction.

The voiced stop consonants consisting of /b/, /d/, and /g/ are produced by building up air pressure behind the vocal tract and then suddenly releasing the pressure. The constriction is at the lips for /b/, at the back of the teeth for /d/, and near the velum for /g/. The voiced stop sounds are dynamical in nature and are highly influenced by the following vowel. Thus, it is relatively difficult to recognize the voiced stop consonants.

The unvoiced stop consonants consisting of /p/, /t/, and /k/ are produced by building up air pressure followed by a sudden release with the unvibrate vocal tract. The duration of frication for unvoiced stops are usually longer than for voiced stops.

The whisper consonant /h/ is a stressed exhale. The affricate consonants consisting of /C/, and /J/ are a concatenation of a stop and a fricative.

2.3.5 Variations in phoneme

Allophones are two or more variants of the same phoneme. For example, the aspirated "p" of <pin> and the nonaspirated "p" in <spin> are allophones of the phoneme "p". The spectral patterns of allophones are slightly different from each other [20].

Prosodic features include the tonal and rhythmic aspects of the speech. These features cause significant variations in speech waveforms, duration time, intensity, as well as frequency spectrum of individual phonemes.

Coarticulation occurs when the phoneme at the end of one word and that at the beginning of the next word are articulated only once. Phoneme articulations and coarticulations usually overlap each other in time and vary in duration. Thus, it is not easy to determine exactly the boundaries of each phoneme in continuous speech.

2.3.6 Speech Signals and Spectrograms

This brief introduction to speech science will be concluded by some illustrations of the complexity involved in viewing phonemes graphically. The consistency of these formant frequencies of a voice is the important factor in recognizing speakers from their voices. Figures 2.6, 2.7, and 2.8 show the plots of the following phoneme groups and their spectrograms :

- Nasal consonant <n> (no) and its spectrogram.
- Voiced stop consonant <g> (go) and its spectrogram.
- Voiced fricative <z> (zoo) and its spectrogram.
- Glide semivowel <r> (read) and its spectrogram.
- Liquid semivowel <w> (we) and its spectrogram.
- Vowel <o> (obey) and its spectrogram.

All the speech signals are recorded at 8 kHz, monoral channel, and 16 bits per sample. The signal is preemphasized with a first order filter with a pole at 0.99. Using a Hamming window length of 30 msec and 50 % overlap, the short-time speech spectrum is calculated and plotted as a function of time and frequency.

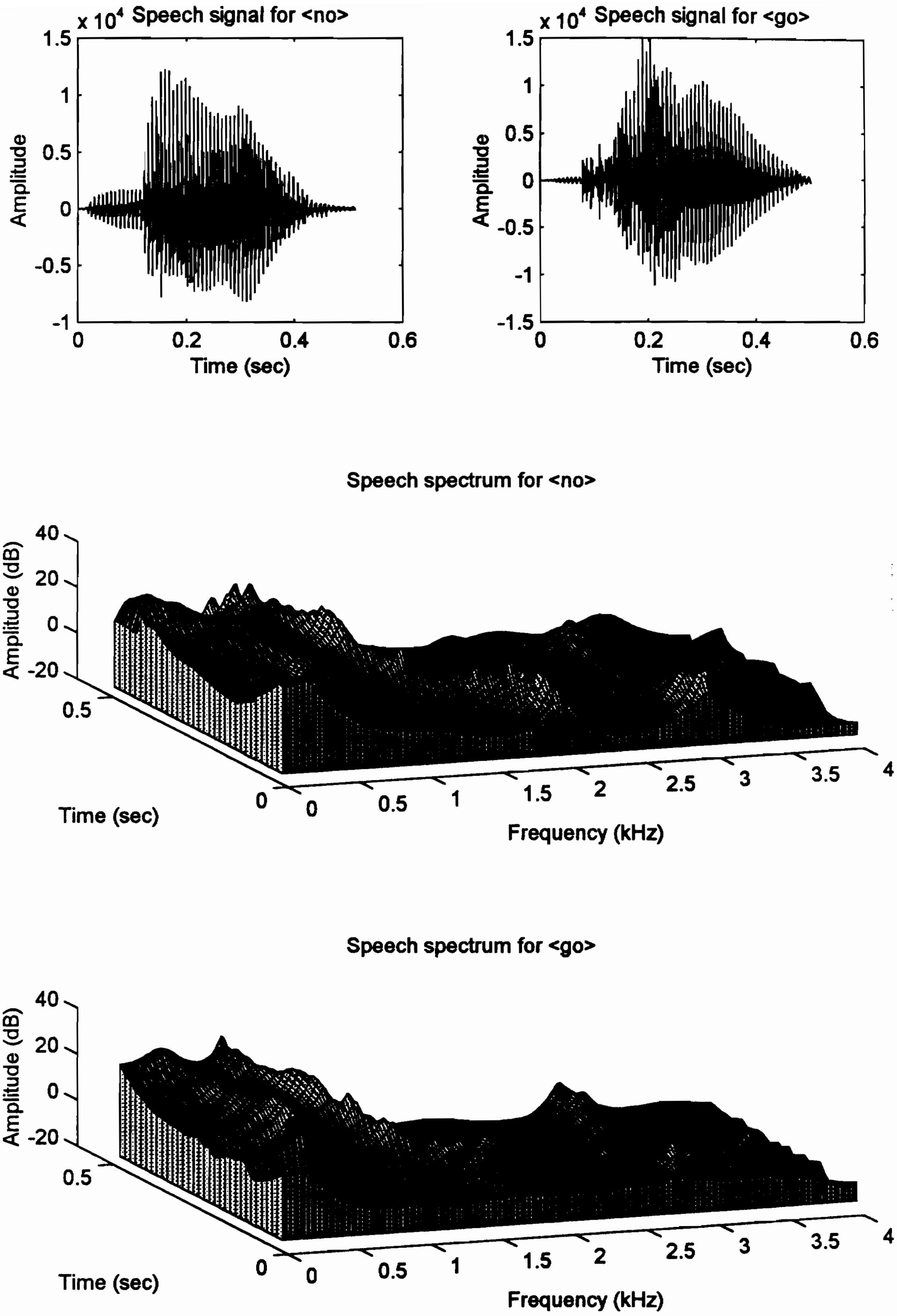


Figure 2.6 : Speech signals and their spectral plots for <no> and <go>.

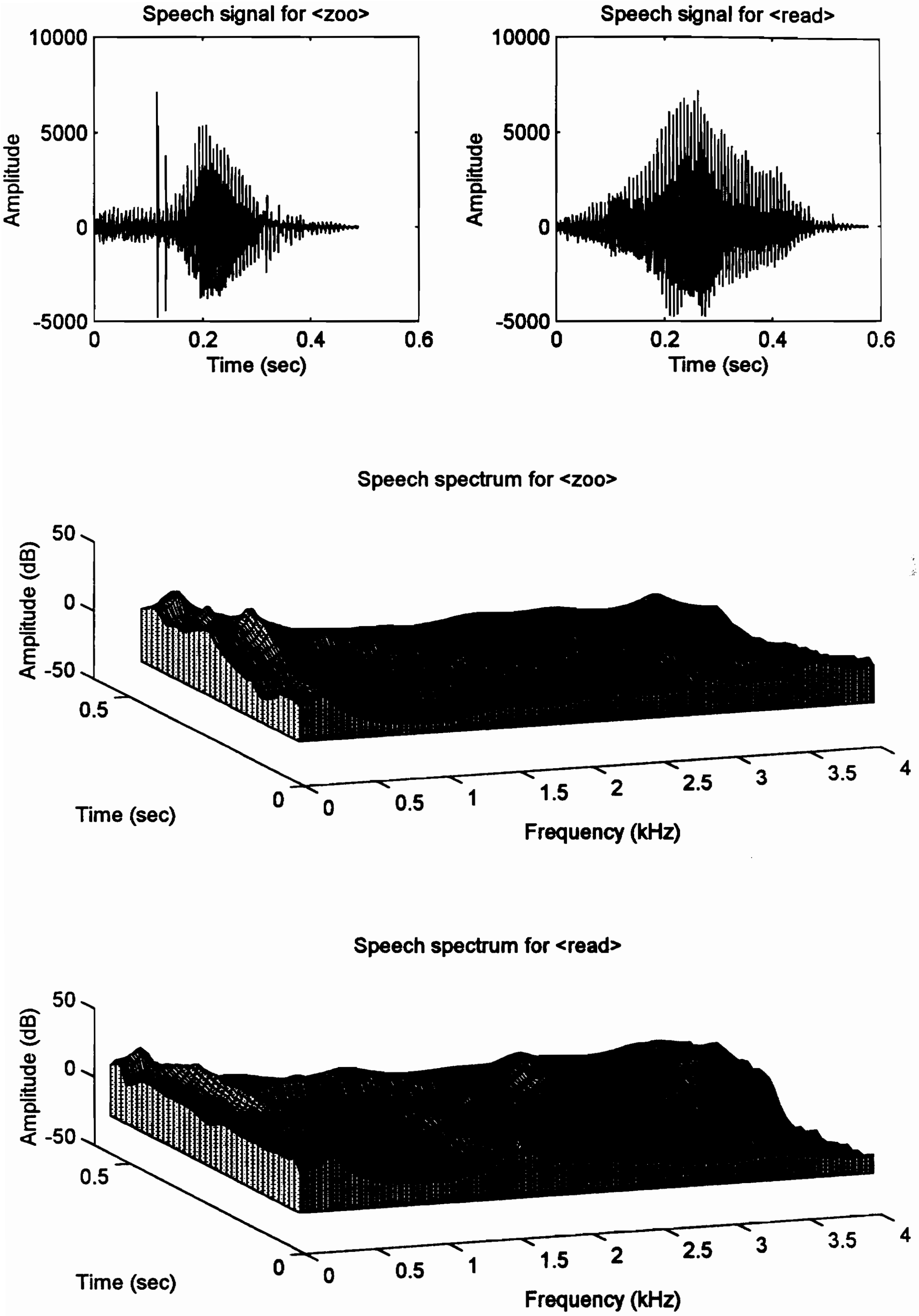


Figure 2.7 : Speech signals and their spectral plots for <zoo> and <read>.

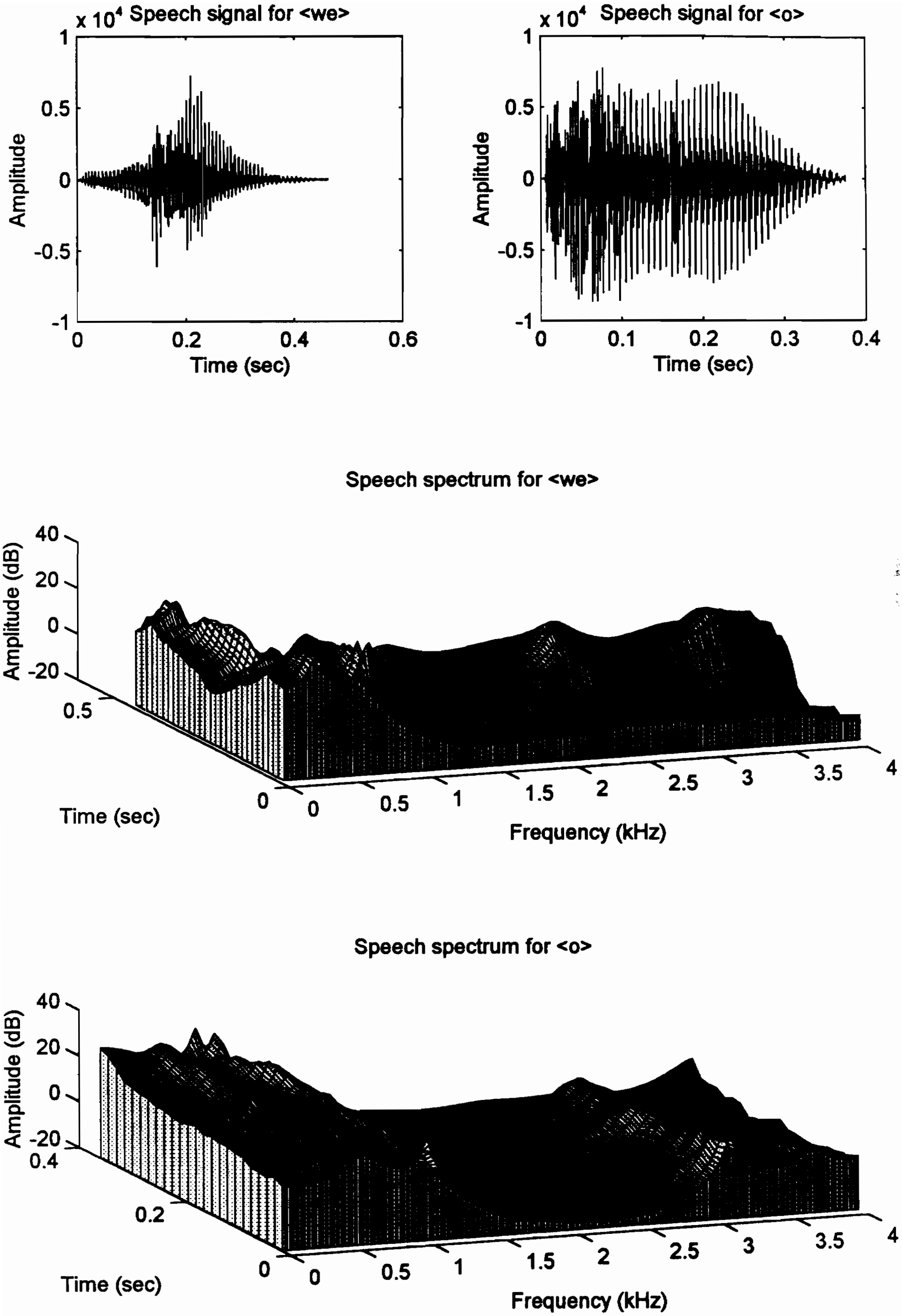


Figure 2.8 : Speech signals and their spectral plots for <we> and <o>.

Chapter 3

Speech Processing Techniques

Most speaker verification systems employ signal processing, pattern recognition, and neural network techniques. Signal processing techniques are used to process the raw speech signal and to extract unique speaker features. Pattern recognition techniques are used to compare reference templates, test templates, and to make decisions.

3.1 Signal Processing Techniques

Basic signal processing tools include the Fourier transform, discrete Fourier transform, z-transform, smoothed windows, energy, zero-crossing rates, etc. Short-time Fourier transforms, filter-bank spectra, and linear predictive coding (LPC) spectra are used to extract important speech features [20], [87], [86].

3.1.1 Basic Concepts in Digital Signal Processing

If x(n) is the discrete sequence obtained by sampling the continuous signal x(t), its Fourier transform (FT) and inverse Fourier transform (IFT) are defined as :

$$X(w) = \sum_{n=-\infty}^{\infty} x(n)e^{-jwn} \tag{3.1}$$

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(w)e^{jwn}dw \tag{3.2}$$

Its z-transform and inverse z-transform are given below :

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{3.3}$$

$$x(n) = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz \tag{3.4}$$

where C is a counterclockwise contour in the region of convergence and encircling the origin in the z-plane.

For a finite sequence x(n) of length N, the discrete Fourier transform (DFT) and the inverse discrete Fourier transform (IDFT) are defined as :

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \qquad k = 0,1,\ldots,N-1 \tag{3.5}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi nk/N}, \qquad n = 0,1,\ldots,N-1 \tag{3.6}$$

where the frequency ω is evaluated at N equally-spaced points, ω = 2πk/N.

3.1.2 Windows

A window is used in speech processing to divide continuous speech into segments which are assumed to be stationary in a short period of time. Several commonly used windows include rectangular, Hanning, Hamming, and Blackman windows. These windows of length N are defined by the following equations :

- *Rectangular*

$$w(n) = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \textit{otherwise} \end{cases} \tag{3.7}$$

- *Hamming*

$$w(n) = \begin{cases} 0.54 - 0.46\cos(2\pi n / N), & 0 \le n \le N-1, \\ 0, & \textit{otherwise} \end{cases} \tag{3.8}$$

- *Hanning*

$$w(n) = \begin{cases} 0.5 - 0.5\cos(2\pi n / N), & 0 \le n \le N-1, \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

- *Blackman*

$$w(n)=\begin{cases} 0.42-0.5\cos(2\pi n/N)+0.08\cos(4\pi n/N), & 0\le n\le N-1, \\ 0, & otherwise \end{cases} \qquad (3.10)$$

3.1.3 Short-Time Fourier Transform

If the sequence s(n) is a set of samples from the speech signal and the sequence v(n) is a window function of length N, then its short-time Fourier transform is given below :

$$S_n(w)=\sum_{k=n-N+1}^{n} s(k)v(n-k)e^{-jwk} \qquad (3.11)$$

The short-term Fourier transform can also be implemented as a filter bank :

$$S_n(w)=\left[s(n)e^{-jwn}\right]*v(n) \qquad (3.12)$$

It can be computed by the use of the discrete Fourier transform (DFT) in each window by evaluating the frequency ω at N equal space points on the unit circle below :

$$S(k)=\sum_{n=0}^{N-1} s(n)v(n)e^{-j2\pi kn/N}, \qquad k=0,1,\ldots,N-1 \qquad (3.13)$$

Since continous speech is assumed to be stationary over a short period of time, the speech spectrum can be considered to be a collection of many short-time Fourier transforms of windowed speech segments.

3.1.5 Short-Term Energy and Zero-Crossing Rate

Short-term energy is used to measure the energy level of a segment of speech. This energy level together with other techniques are used to determine speech endpoints, voiced speech, unvoiced speech, and silence. The short-term energy for a frame of length N ending at time m is defined below :

$$E_m = \sum_{n=m-N+1}^{m} s^2(n) \tag{3.14}$$

The zero-crossing rate measures the number of times the speech signal crosses the zero value in a frame of length N, and is defined below :

$$Z(m) = \frac{1}{2N} \sum_{n=m-N+1}^{m} \left| sgn[s(n)] - sgn[s(n-1)] \right| \tag{3.15}$$

where $sgn[s(n)] = 1, \quad s(n) \geq 0$

$\qquad = -1, \quad s(n) < 0.$

It is an important parameter for voiced/unvoiced classification because the zero-crossing rate for unvoiced speech is much higher than that of voiced speech.

3.1.6 Filter Banks

Since the vocal tract can be characterized by a set of formant frequencies, a filter bank can be used as the front end in the speech recognition system to detect these resonant frequencies. A typical output of the i th bandpass filter in a bank of Q filters is given below

$$s_i(n) = s(n) * h_i(n) = \sum_{k=0}^{K_i - 1} h_i(k) s(n-k) \tag{3.16}$$

where $h_i(n)$ is the impulse response of the i th bandpass filter with a K_i sample duration.

This output signal $s_i(n)$ is rectified and low-pass filtered to measure the energy of the speech signal in this frequency band and to reject undesired frequencies. Sampling rate reduction and amplitude compression are used to improve the output representation. The block diagram of a filter bank is presented in Figure 3.1. Since the fastest rate of motion of speech harmonics is about 20-30 Hz, the outputs from low-pass filters can be resampled at the rate of 40-60 Hz. These new channel signals can be compressed using logarithmic encoding or μ-law encoding. The bit rate of the resulting channel signals is much less than that of the original speech.

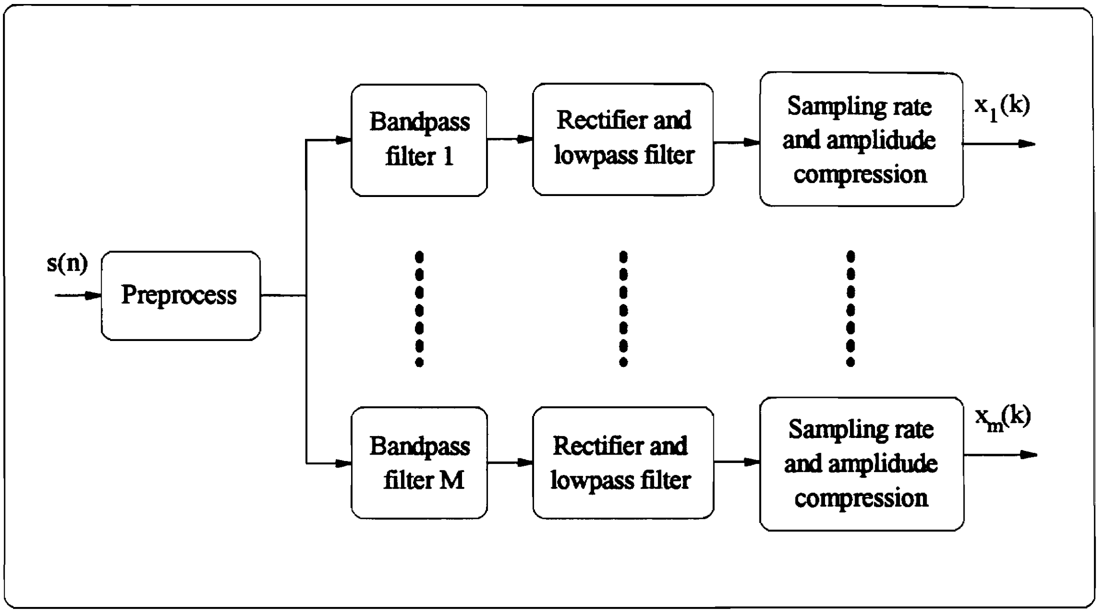

s(n)
Preprocess
Bandpass filter 1
Rectifier and lowpass filter
Sampling rate and amplidude compression
$x_1(k)$
Bandpass filter M
Rectifier and lowpass filter
Sampling rate and amplidude compression
$x_m(k)$

Figure 3.1 Block diagram of the filter bank.

For a uniform filter bank, the bandwidth of all bandpass filters are equal, K_i = N. The filter bank in Eq. (3.16) can be implemented in term of the short-time Fourier transform using fixed window w(n) of length N below :

$$s_i(n) = \underbrace{e^{jw_i n}\sum_{k=0}^{N-1} s(k)w(n-k)e^{-jw_i k}}_{S_n(e^{jw_i})} = e^{jw_i n}S_n(e^{jw_i}) \qquad (3.17)$$

The discrete Fourier transform of Eq. (3.17) is given below :

$$s_i(n) = e^{j(2\pi i/N)n}\sum_{k=0}^{N-1} s(k)w(n-k)e^{-j(2\pi i/N)k} \qquad (3.18)$$

Most-practical filter bank systems use FIR bandpass filters, the number of filters Q is between 8 and 32 with full or half wave rectifiers, and IIR lowpass filters. Preprocessors and postprocessors are used in filter banks to improve the recognition. Preprocessors include signal preemphasis (to equalize the inherent spectral tilt in speech), noise elimination, and signal enhancement. Postprocessors include frequency smoothing of individual filter-bank outputs.

The IIR lowpass filter is represented by the following difference equation :

$$y(n)=\sum_{k=1}^{N}a_k y(n-k)+\sum_{r=0}^{M}b_r x(n-r) \tag{3.19}$$

The present value of the output is calculated from the past values of the output and the present and past values of the input recursively. Due to the recursive formula, the IIR filter can be designed to achieve the same specification with lower filter order than the FIR filter. The following classical designs are used widely to implement IIR filters :

- Butterworth design : the magnitude response is maximally flat in both passband and stopband.
- Chebyshev design : the magnitude response is equiripple in either passband or stopband.
- Elliptic design : the magnitude response is equiripple in both passband and stopband.

The difference equation of the FIR filter is given below :

$$y(n)=\sum_{r=0}^{M}b_r x(n-r) \tag{3.20}$$

The linear phase property of the FIR filter is very useful in speech processing applications where precise time alignment is necessary [87]. The three general techniques for FIR design are listed below :

- Window design technique.
- Frequency sampling technique.
- Equiripple technique.

However, the linear phase property comes with the penalty of much higher order filters.

3.1.7 Linear Predictive Coding

Linear predictive coding attempts to model the present speech signal as a linear combination of past signal values, which translates into an all-pole transfer function representation of the speech production system. Therefore, the speech signal *s(n)* can be represented by the difference equation below :

$$s(n) = Gu(n) + \sum_{k=1}^{P} a_k s(n-k) \qquad (3.21)$$

where *u(n)* is an unknown excitation and G is a gain of the excitation.

The transfer function $H(z)$ associated with Eq. (3.21) is given by

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{P} a_k z^{-k}} \qquad (3.22)$$

The estimated signal $\tilde{s}(n)$ and the prediction error $e(n)$ are defined as

$$\tilde{s}(n) = \sum_{k=1}^{P} a_k s(n-k), \qquad (3.23)$$

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{P} a_k s(n-k) \qquad (3.24)$$

Then the total mean square error is given by

$$E = \sum_n e^2(n) = \sum_n \left[s(n) - \sum_{k=1}^{P} a_k s(n-k) \right]^2 \qquad (3.25)$$

Using the method of least squares, the parameters a_k can be determined by setting

$$\frac{\partial E}{\partial a_i} = 0, \qquad i = 1, 2, ..., P \qquad (3.26)$$

to get

$$\boxed{\Phi_{io} = \sum_{k=1}^{P} a_k \Phi_{ik}\,, \qquad i = 1, 2, ..., P} \qquad (3.27)$$

where $\Phi_{ik} = \sum_n s(n-i)s(n-k)$ (3.28)

There are two principle methods to solve Eq. (3.27) for the a_k coefficients: the autocorrelation method and the covariance method.

- ***The autocorrelation method***

The autocorrelation method assumes that the signal exists inside a window of length N, and equals zero outside the window. Usually, the signal is weighted by one of the commonly used windows above, and is defined below :

$$s(m) = \begin{cases} s(m+n)w(m), & 0 \le m \le N-1 \\ 0, & otherwise. \end{cases} \tag{3.29}$$

Now, the problem is to minimize the following mean squared error below :

$$E_n = \sum_{m=0}^{N-1+P} e_n^2(m) \tag{3.30}$$

The autocorrelation solution to Eq. (3.27) can be expressed as

$$\sum_{k=1}^{P} a_k R(|i-k|) = R(i) \qquad 1 \le i \le P \tag{3.31}$$

$$G = \sqrt{R(0) - \sum_{k=1}^{P} a_k R(k)} \tag{3.32}$$

where $$R(i) = \sum_{n=0}^{N-1-i} s(n)s(n+i) \tag{3.33}$$

The matrix form of Eq. (3.31) is given by

$$\begin{bmatrix} R_o & R_1 & \cdots & R_{P-1} \\ R_1 & R_o & \cdots & R_{P-2} \\ . & . & & . \\ . & . & & . \\ R_{P-1} & R_{P-2} & \cdots & R_o \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_P \end{bmatrix} = \begin{bmatrix} \mathrm{R}_1 \\ \mathrm{R}_2 \\ . \\ . \\ \mathrm{R}_\mathrm{P} \end{bmatrix} \tag{3.34}$$

This special (P x P) matrix with equal diagonal elements and symmetry is called a Toeplitz matrix. There are several efficient algorithms employed to solve the Toeplitz matrix equation. Durbin's recursive algorithm in Table 3.1 can be used to compute the values of a_k once $R(i)$ has been calculated. The order of P is usually between 8 to 20.

Table 3.1 Durbin recursive algorithm.

```
1. Compute :
      $R(i) = \sum_{n=0}^{N-1-i} s(n)s(n+i)$      for $i = 0, 1, 2, ..., P$.
2. Set :
      $E_o = R(0)$
3. For $i = 1, 2, ..., P$
      $k_i = \dfrac{R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j)}{E_{i-1}}$
      $\alpha_i^{(i)} = k_i$
      For $j = 1, ..., i-1$
          $\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}$
      end
      $E_i = (1 - k_i^2) E_{i-1}$
   end
4. Solution :
   $a_k = \alpha_k^{(P)}$      for $k = 1, 2, ..., P$
```

- ***The covariance method***

An alternative to using a window is to use a fixed interval of n = 0, 1, ..., N-1 to compute the mean-squared error and to use the unweighted speech directly :

$$E = \sum_{n=0}^{N-1} e^2(n) \tag{3.35}$$

The solution can be expressed as

$$\begin{bmatrix} \Phi_{(1,1)} & \Phi_{(1,2)} & \cdots & \Phi_{(1,P)} \\ \Phi_{(2,1)} & \Phi_{(2,2)} & \cdots & \Phi_{(2,P)} \\ . & . & & . \\ . & . & & . \\ \Phi_{(P,1)} & \Phi_{(P,2)} & \cdots & \Phi_{(P,P)} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_P \end{bmatrix} = \begin{bmatrix} \Phi_{(1,0)} \\ \Phi_{(2,0)} \\ . \\ . \\ \Phi_{(\mathrm{P},0)} \end{bmatrix} \tag{3.36}$$

where $\Phi_{(i,k)} = \sum_{n=0}^{N-1} s(n-i)s(n-k), \qquad i = 1, 2, \ldots, P$ and $k = 0, 1, \ldots, P$ (3.37)

The Cholesky decomposition method is used to solve for the LPC coefficients.

Other LPC derived parameters include the log area ratio coefficients, cepstral coefficients, PARCOR coefficients, and parameter weighting cepstral coefficients [81].

- *The log area ratio coefficients*

$$g_m = \log\left(\frac{1-k_m}{1+k_m}\right) \qquad 1 \le m \le P \tag{3.38}$$

where k_m are the PARCOR coefficients and g_m are log area ratio coefficients.

- *The LPC cepstral coefficients*

$$c_o = \ln(G^2) \tag{3.39a}$$

$$c_m = a_m + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k} \ , \qquad 1 \le m \le P \tag{3.39b}$$

$$c_m = \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k} \ , \qquad m > P \tag{3.39c}$$

where G is the gain term in the LPC model.

Since the low order cepstral coefficients are sensitive to the overall spectral slope, and the high order cepstral coefficients are sensitive to noise, it is desirable to weight these parameters to minimize these effects.

- *The parameter weighting cepstral coefficients*

$$w_m = 1 + (Q/2)\sin(m\pi/Q) \tag{3.40}$$

$$h_m = w_m c_m , \qquad\qquad 1 \le m \le Q \tag{3.41}$$

where w_m is a weighting window known as the bandpass lifter.

- *The first order temporal cepstral derivative coefficients*

$$c'_m(t) \approx \mu \sum_{k=-K}^{K} k c_m(t+k), \qquad\qquad 1 \le m \le Q \tag{3.42}$$

where μ is a normalization constant and K = 3.

3.1.8 Real Cepstral Analysis

If a low frequency signal is corrupted by the addition of high frequency noise, a simple lowpass filter can be designed to remove the unwanted noise. However, the speech signal is modeled by the convolution of the vocal tract transfer function and the glottal input signal, and can be expressed by :

$$s(n) = u(n) \otimes h(n) \tag{3.43}$$

It is not possible to separate the vocal tract from the glottal input signal using frequency filters. Real cepstral analysis in Figure 3.2 can be employed to separate the vocal tract signal from the glottal signal. This process is called the *liftering operation*. The cepstral analysis procedure of Figure 3.2 is presented by the following equations :

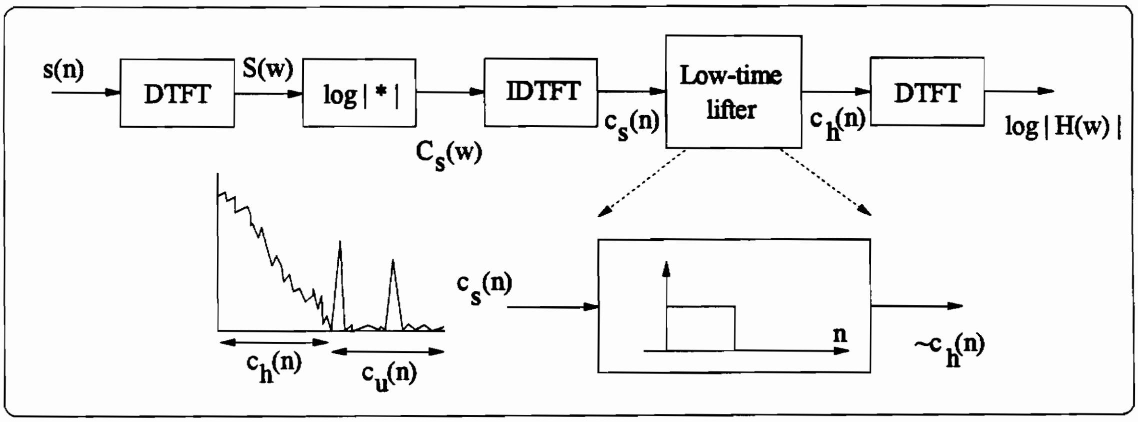


Figure 3.2 Low-time liftering cepstral analysis.

$$\text{DTFT } \{ s(n) \} : \qquad S(w) = U(w)H(w) \qquad (3.44)$$

$$\text{Log } \{ S(w) \} : \qquad \underbrace{\log|S(w)|}_{C_s(w)} = \underbrace{\log|U(w)|}_{C_u(w)} + \underbrace{\log|H(w)|}_{C_h(w)} \qquad (3.45)$$

$$\text{IDTFT } \{ C_S(w) \} : \qquad c_s(n) = c_u(n) + c_h(n) \qquad (3.46)$$

Since the $C_h(w)$ is the spectrum of the slowly varying vocal tract, and $C_u(w)$ is the spectrum of the fast varying glottal, low time lifter can be used to separate $c_u(n)$ from $c_h(n)$.

3.2 Pattern Recognition Techniques

Speech endpoint detection, spectral distortion measures, pattern matching, decision logic, time alignment and normalization using dynamic programming are among pattern-comparison techniques. They are the powerful tools in the area of speech processing.

3.2.1 Speech Endpoint Detection

Detecting the presence of speech in a background noise is an important task to achieve high accuracy in speaker recognition system. For the detection of the endpoints of speech signals, several algorithms have been proposed [2, 21, 53, 70, 83, 84, 85, 103, 113, 120, 124]. Simple endpoint detection uses the short-time energy and zero crossing rate algorithms. Sophisticated endpoint detection employs the short-time energy, pattern comparison, adaptive level quantization, and a decision rule for noisy environments. Short time energy is used to distinguish among voiced, unvoiced, or background silence speech. The zero crossing rate (ZCR) also provides a rough voiced / unvoiced classification feature since unvoiced speech has a much higher ZCR than voiced speech. In speaker recognition systems, speech endpoints are used to removed silence, pauses, and other unwanted unvoiced speech which helps to reduce the amount of processing of the speech data. Figures 3.3, 3.4, and 3.5 summarize three different approaches to most of the

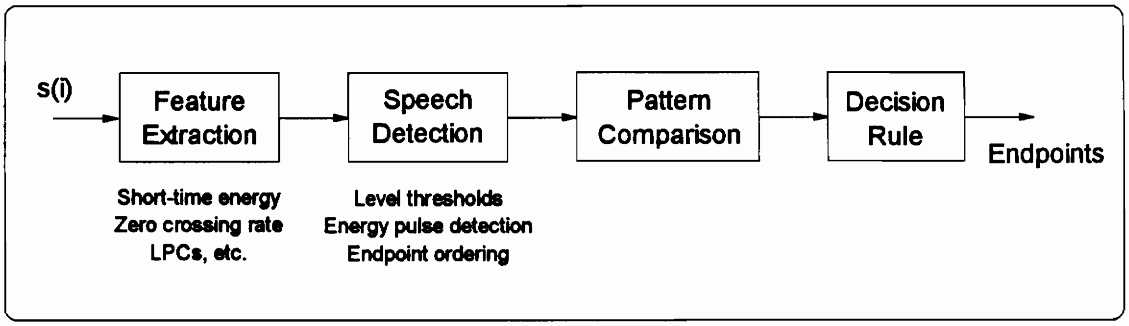


Figure 3.3 The explicit approach.

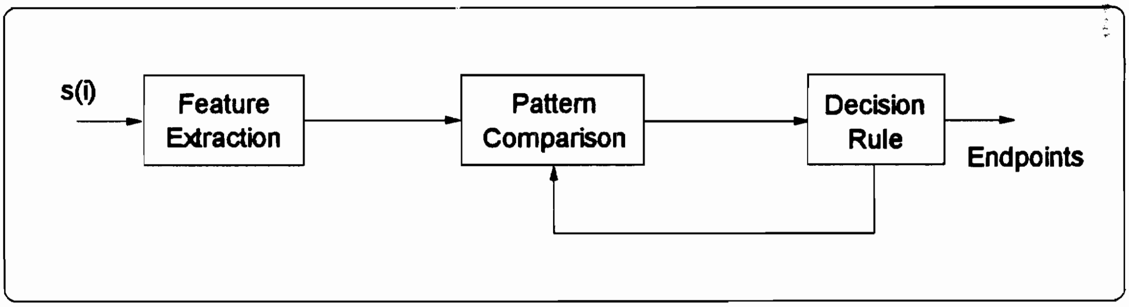


Figure 3.4 The implicit approach.

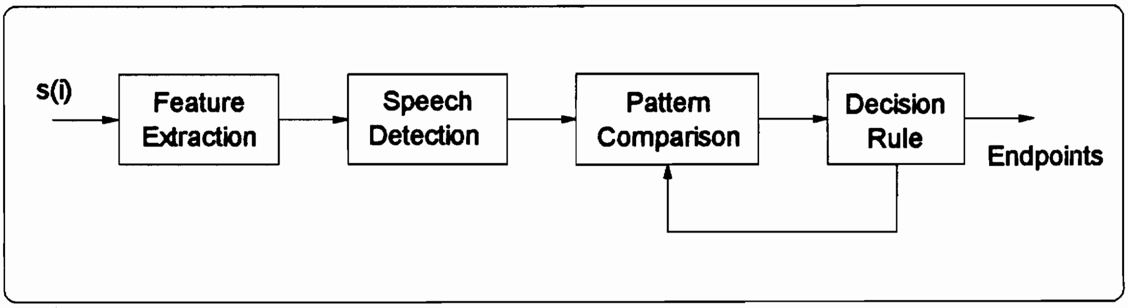


Figure 3.5 The hybrid approach.

speech endpoint detection algorithms: the explicit approach, the implicit approach, and the hybrid approach [86]. The explicit approach fails in noisy environments. On the other hand, the implicit approach provides higher accuracy with the penalty of high computational load. The hybrid approach, providing both high accuracy and low computational load, requires setting the best threshold values (by extensive testings).

3.2.2 Spectral Distortion Measures

Spectral distortion measures are used to compute the overall dissimilarity between two feature spectral patterns. The goal is to have a large distance for two perceptually different sounds, and a small distance for two perceptually similar sounds. Before discussing spectral distances, it is important to know the following concepts [86]:

- Spectral changes that do not fundamentally change the perceived sound include : spectral tilt, highpass filtering, lowpass filtering, and notch filtering. The associated spectral distance should be small.
- Spectral changes that perceptually lead to different sounds include : significant differences in formant locations, and significant differences in formant bandwidth. The associated distance should be large.

To determine the dissimilarity between speech signals s(n) and s'(n), the direct Euclidean distance between the above two signals is not appropriate due to the variations in speaking rate, unknown time alignment of the two signals, and variable degrees of loudness. Now let $S(w)$ and $S'(w)$ represent the power spectrum of s(n) and s'(n). Then the log of S(w) can be expressed in terms of cepstral coefficients below :

$$\log S(w) = \sum_{m=-\infty}^{\infty} c_m e^{-jmw} \tag{3.47}$$

Using Parseval's theorem, the rms log spectral distance can be expressed as :

$$d_2^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\log S(w) - \log S'(w)\right|^2 dw = \sum_{m=-\infty}^{\infty} (c_m - c'_m)^2 \tag{3.48}$$

where c_m and c'_m are the cepstral coefficients of $S(w)$ and $S'(w)$ respectively [86].

Since the cepstrum is a decaying sequence, the truncated cepstral distance is adequate and is defined below :

$$d_c^2 = \sum_{m=1}^{p} (c_m - c'_m)^2 \tag{3.49}$$

where the first p cepstral coefficients in (3.47) are used.

The following additional spectral distortion measures commonly employed in many speaker recognition systems are listed below :

- *Root power sum distance*

$$d_{cw}^2 = \sum_{m=1}^{p} (nc_m - nc'_m)^2 \tag{3.50}$$

- *Weighted cepstral liftering distance*

$$d_{cw}^2 = \sum_{m=1}^{p} w_m^2 (c_m - c'_m)^2 \tag{3.51}$$

  where w_m is defined in equation (3.40).

- *Mahalanobis distance*

$$d_m^2 = (\mathbf{c} - \mathbf{c}')^T \mathbf{R}^{-1} (\mathbf{c} - \mathbf{c}') \tag{3.52}$$

  where **R** is the intra-speaker covariance matrix of **c**. The inverse covariance matrix is used to decorrelate and normalize the cepstral coefficients.

- *Likelihood ratio distance*

$$d_{LR}^2(\mathbf{a},\mathbf{b}) = \frac{\mathbf{b}^T \mathbf{R}_a \mathbf{b}}{\mathbf{a}^T \mathbf{R}_a \mathbf{a}} - 1 \tag{3.53}$$

  where $\mathbf{R}_a$ is the autocorrelation matrix of the vector **a**.

- *Inverse variance distance*

$$d_{iv}^2 = \sum_{m=1}^{p} [w_m (c_m - c'_m]^2 \tag{3.54}$$

  where $$w_m = \frac{1}{\mathrm{var}(c_m)} \tag{3.55}$$

- *Correlation distance*

$$d^2(\mathbf{a},\mathbf{b}) = \frac{\mathbf{b}^T\mathbf{a}}{\left[(\mathbf{b}^T\mathbf{b})(\mathbf{a}^T\mathbf{a})\right]^{1/2}} \tag{3.56}$$

- *Smoothed Group Delay Spectrum*

$$d_{sg}^2(\mathbf{a},\mathbf{b}) = \sum_{m=1}^{p}\left[w_m(a_m - b_m)\right]^2 \tag{3.57}$$

where
$$w_m = me^{\left[\frac{-m^2}{2\kappa^2}\right]} \tag{3.58}$$

$\kappa \approx 13.33.$

3.2.3 Time Alignment and Normalization

Since automatic segmentation of utterances into meaningful linguistic units is not easy, speech utterances are usually divided into equal speech segments called frames, and the acoustic feature vectors are computed from these frames. A series of feature vectors that characterize the behavior of the speech signal is called a template. Templates of utterances are usually compared frame by frame which leads to a time alignment problem since speech events in two utterances of the same text spoken by the same speaker are seldom aligned in time due to differences in speaking rates. Therefore, time alignment needs to be done before comparing speech utterances of the same text. Simple approximate time synchronization shown in Figure 3.6 can be achieved by linearly aligning the beginning and the end of the two utterances, called a linear time alignment. In linear time alignment the spectral distortion between two speech patterns X and Y can be written as [86] :

$$d(\mathrm{X},\mathrm{Y}) = \sum_{nx=1}^{M} d(\mathbf{x}_{nx}, \mathbf{y}_{ny}) \tag{3.59}$$

where
$$\mathbf{X} = \{(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M)\}, \quad \mathbf{Y} = \{(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)\}$$

$$nx = \frac{N}{M}ny \tag{3.60}$$

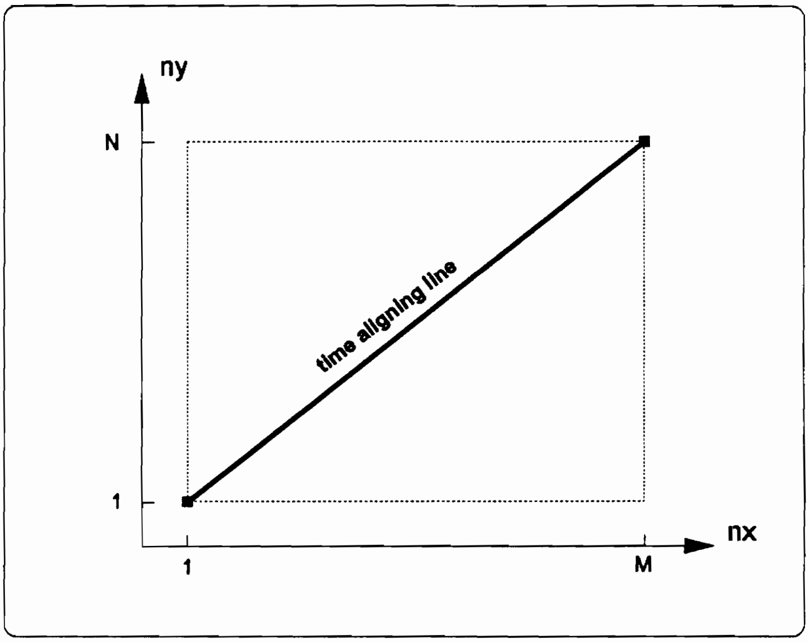


Figure 3.6 Linear time alignment of two speech patterns.

From equations (3.56), (3.60), the speaking rate change is linearly proportional to the duration of the utterance and independent of the sound being spoken. Accurate time alignment is important for an automatic speaker recognition system. The constraint of invariant speaking rate is not sufficient to align speech events of utterances because the effects of speaking rate variations are nonlinear : vowels and stressed syllables tend to expand or contract more than consonants and unstressed syllables [76]. Enough misaligned speech events can result in large distortion distances which leads to the incorrect decision to reject an utterance spoken by same person. The need for nonlinear time alignment of speech events is obvious.

Nonlinear time alignment warps one speech template in an attempt to align similar acoustic segments to the reference template, known as a dynamic time warping (DTW) procedure. The DTW method uses one time warping function m = w(n) to normalize the test pattern time to the reference pattern time nonlinearly. This DTW procedure usually employs a dynamic programming technique that uses an optimum time expansion / compression function for nonlinear alignment shown in Figure 3.7. The DTW aligns templates by finding a time warping that minimizes the total distance measure. From

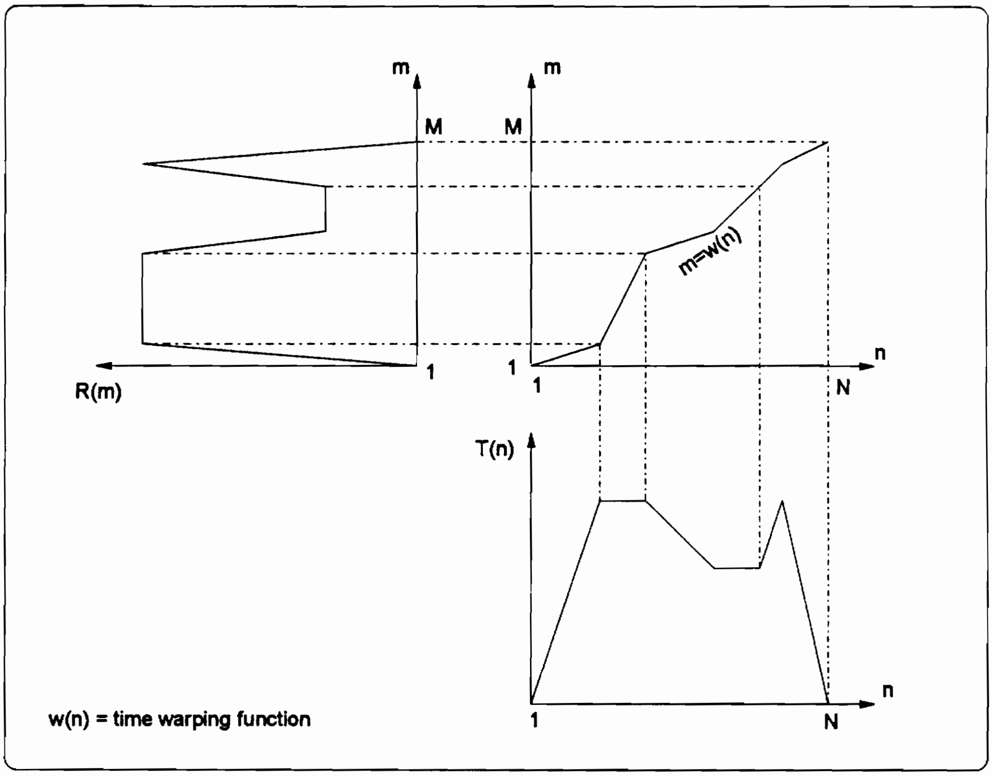


Figure 3.7 Nonlinear time alignment of test T(n) with reference R(m).

Figure 3.7, the reference pattern R(m) has M frames, and the test pattern T(n) has N frames. DTW searches for the best frames in the reference template to match against the frames in the test template. The time warping function *w*(n) is derived from the solution of an optimization problem :

$$D = \min_{w(n)} \left[\sum_{n=1}^{N} d(T(n), R(w(n))) \right] \tag{3.61}$$

where *d*(.,.) is a frame distance between the test pattern *T*(n) and the reference pattern *R*(*w*(n)). The resulting *w*(n) is the "optimal path" solution that minimizes the total distance D.

The principle of optimality which is the basis of a class of computational algorithms for the above optimization problem is described in the theorem below :

Theorem 3.1 : *The Principle of Optimality.*

An optimal policy has the property that, whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

The principle of optimality translates into the equation below :

$$\varphi(i,j) = \min_{m}\left[\varphi(i,m) + \varphi(m,j)\right] \tag{3.62}$$

The mathematical formula is described as the minimum cost of moving from point i to point j in one step is equal to the minimum cost of moving from i to j in any partial optimal sequence. The dynamic programming algorithm based on this principle of optimality is usually employed to find the optimal time warping path w(n).

However, a more general time alignment employs two warping functions defined by:

$$nx = \Phi_x(k) \tag{3.63}$$

$$ny = \Phi_y(k) \tag{3.64}$$

where k = 1, 2, ..., T; nx = 1, 2, ..., M; and ny = 1, 2, ..., N.

To align two utterances, the typical DTW constraints include the following [86] :

- endpoint constraints.
- monotonicity conditions.
- local continuity constraints.
- global path constraints.
- slope weighting.

- *Endpoint constraints* :

Usually utterances are limited to fix temporal endpoints, leading to a set of constraints for the warping functions :

$$\Phi_x(1) = 1, \qquad \Phi_y(1) = 1 \tag{3.65}$$

$$\Phi_x(T) = N, \qquad \Phi_y(T) = M \tag{3.66}$$

- *Monotonicity Conditions* :

Monotonicity conditions are used to maintain the temporal order of the spectral sequence in the speech pattern and to eliminate the possibility of time reverse warping which does not make sense. The conditions are defined below :

$$\Phi_x(k+1) \geq \Phi_x(k) \tag{3.67}$$

$$\Phi_y(k+1) \geq \Phi_y(k) \tag{3.68}$$

- *Local Continuity Constraints* :

The local continuity constraints are employed to ensure the proper time alignment while keeping the loss of information to minimum. The local continuity constraints are derived in many different forms. One set of the local continuity constraints is written below :

$$\Phi_x(k+1) - \Phi_x(k) \leq 1 \tag{3.69}$$

$$\Phi_y(k+1) - \Phi_y(k) \leq 1 \tag{3.70}$$

A typical type II set of local continuity constraints and the resulting paths is illustrated in Figure 3.8.

- *Global Path Constraints* :

Global path constraints are presented in Figure 3.9. All allowable paths must move within the shaded parallelogram area in Figure 3.9. The effects of global path constraints prevent any path that involves an excessive time stretch or compression since there must be a deviation limit on the variations in rate for a speaker uttering the same word or sentence on different occasions.

- *Slope Weighting* :

The slope weighting function *m*(k) controls the contribution of each short-time distortion d(.,.). Larger weight values are used for less preferable paths since a higher distortion represents a less likely match. Figure 3.10 shows some slope weighting forms.

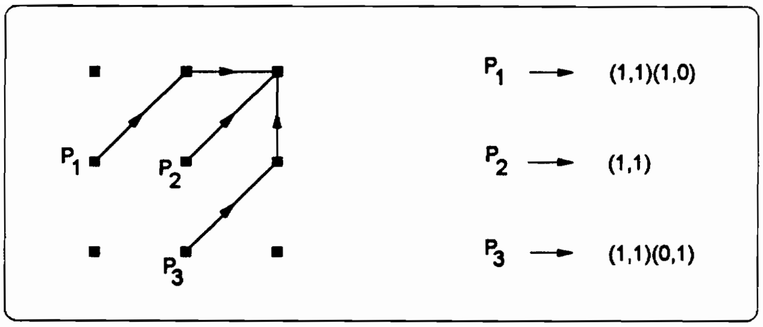


Figure 3.8 Type II, local continuity constraints.

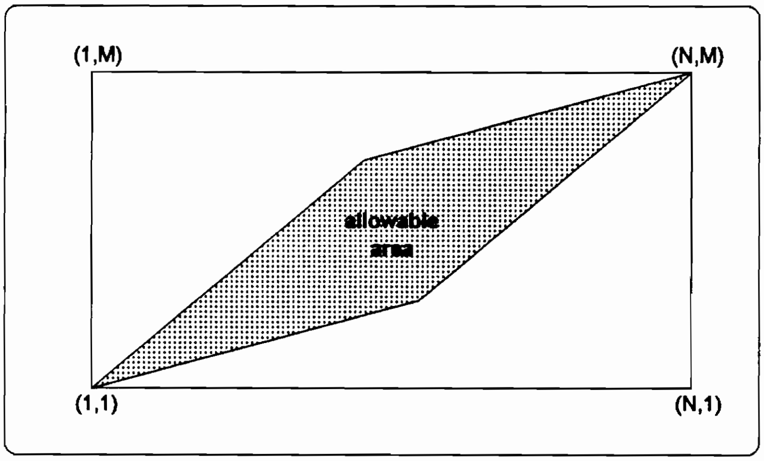


Figure 3.9 Global path constraints _ shaded area.

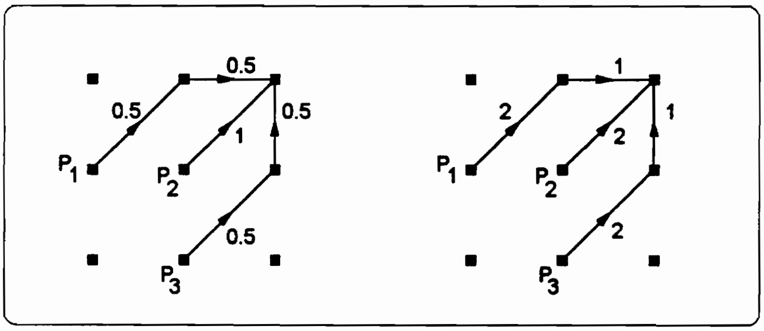


Figure 3.10 Two forms of slope weighting.

3.3 Neural Network Techniques

The working of the human brain is still a mystery to man. The brain is so powerful that it can perform a wide variety of solving many problems from thinking, talking, remembering, feeling, and learning. This powerful device of the human has amazed and inspired many scientists to attempt modeling its operations. In the brain, a neuron is the main cellular unit of the nervous system. Each neuron receives and combines signals from many other neurons and produces an output signal to the axon to perform certain actions. Neurocomputing has emerged from this inspiration.

In an effort to model certain capabilities of the brain, Warren McCulloch and Walter Pitts established a simplified model of a biological neuron in "A logical calculus of ideas imminent in nervous activity" paper in 1943. Later Hebb (1949) suggested a method whereby the parameters of the McCulloch-Pitts neuron model could self-adjust. These early studies of biological neural networks laid the foundations for what was to become known as artificial neural networks (ANNs).

3.3.1 Neuron Model

The McCulloch-Pitts model for a biological neuron consists of many inputs, corresponding to dendrites (cell connections to other neurons) connected through synaptic junctions (variable weights). The model is described by :

$$Y = f(\sum_{i=0}^{N} x_i w_i) \qquad (4.1)$$

where x_i = inputs, $i = 0, 1, 2, \ldots, N,$
x_o = bias = 1,
w_i = variable weights.

In this model, x_o is used to provide a bias to the activation function $f(.)$. McCulloch and Pitts did not provide any method through which the neuron could adapt its weights in a "learning" process. In 1949, Hebb postulated a simple mathematical formula to change the neuron weights in proportion to the activations of the neuron :

$$\Delta w_i = \mu Y(\mathbf{x})x_i, \qquad i = 0,1,\ldots,N \tag{3.72}$$

where **x** represents the vector of (N+1) inputs and μ is the learning parameter.

Figure 3.11 illustrates the McCulloch-Pitts neuron model used widely in many different ANN paradigms.

3.3.2 The Perceptron

In 1958, Rosenblatt demonstrated some practical applications using the perceptron. The perceptron is a single level connection of McCulloch-Pitts neurons. The perceptron is capable of linearly separating the input vectors into pattern classes by a hyperplane. Figure 3.12 shows the perceptron of N-features (inputs) and M-classes (outputs). This perceptron can be described by :

$$y_i = f\left\{\sum_{j=0}^{N} w_{ij}x_j\right\}, \qquad i = 1,2,\ldots,M \tag{3.73}$$

where the function $f(x) = 1(x)$, the unit-step function.

Rosenblatt derived a learning rule based on weight adjusted in proportion to the error between the output neurons and the target outputs. The weight adjustments are given by :

$$\Delta w_{ij} = \mu(y_i^d - y_i)x_j \tag{3.74}$$

where $i = 1, 2, \ldots, M$, $j = 0, 1, \ldots, N$, and y^d is the desired output vectors.

The perceptron represented a major step in the application of ANNs. Still, many problems cannot be solved with two-layer perceptrons.

3.3.3 Neural Network Operation and Benefits

There are two distinct operation modes of neural networks. They are the *learning* mode and the *recall* mode. The *learning* mode is the process of modifying the network weights in response to a set of inputs in such a way that the outputs produced by the

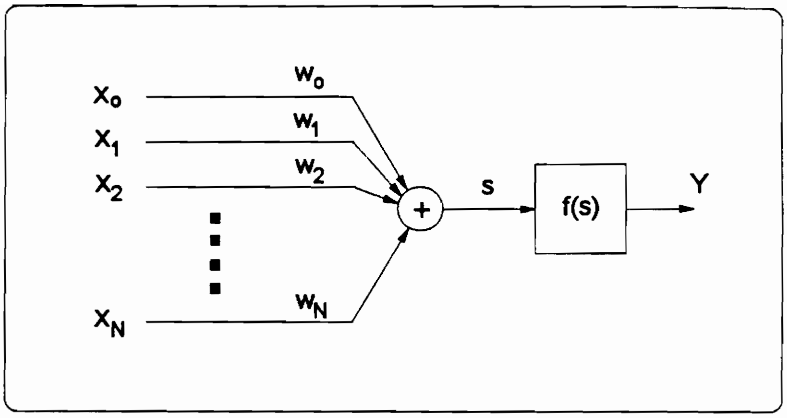


Figure 3.11 The McCulloch-Pitts Neuron.

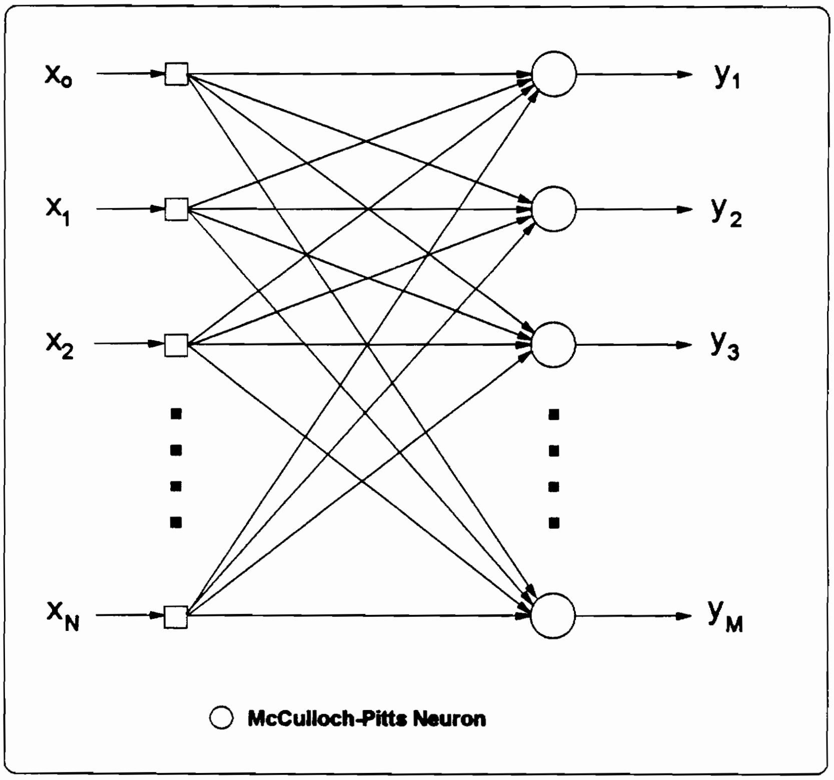


Figure 3.12 Single-layer perceptron.

network should follow the desired outputs to an acceptable level of error. There are three types of learning : *supervised*, *unsupervised*, and *reinforcement*. When the desired outputs are used to train the network, the learning process is called *supervised*. When the outputs are not presented to train the network, unsupervised learning process is observed. The last type of learning is *reinforcement* learning where only grades of good or bad are presented to the network during training.

The *recall* mode is the process of finding how well the network has learned. Often, the network will be presented with a different set of data from the training data. The outputs produced by the neural network are compared against the desired outputs to determine the network's performance.

Several useful properties of ANNs include :

- learning by example : the ability to create their rules by learning from training patterns presented to the networks.
- fault tolerance : the response of the networks only change slightly if some processing elements are defective or damaged due to the fact that the information is not stored in one place. It is distributed across its numerous weights.
- input-output nonlinear mapping.
- VLSI massively parallel implementability.
- Adaptivity : the neural networks have the ability to change their weights to adapt to a new environment.

3.3.4 Multi-layer Perceptrons

The capabilities of single-layer perceptrons are limited to linear decision boundaries and simple logic functions. The single-layer perceptrons cannot realize the simple XOR problem. This led to the development of multi-layer perceptrons. In general, multi-layer perceptrons (MLPs) consist of an input layer, one or more hidden layers, and an output layer. Figure 3.13 illustrates a feed-forward three layer MLP network. The MLP networks overcome many of the limitations of the single-layer perceptrons. Besides the hard-limiting activation function, other nonlinear smoothed functions also are employed

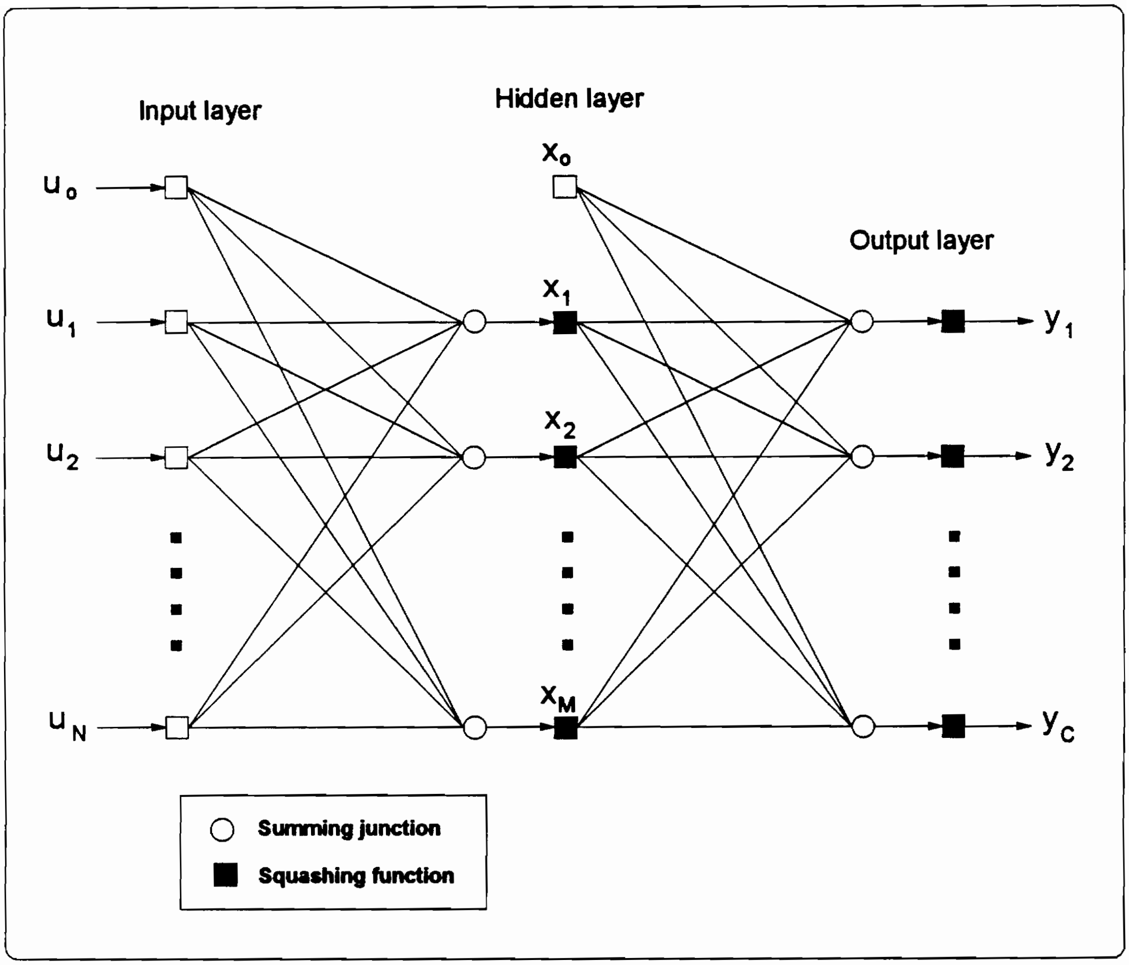


Figure 3.13 Three-layer feedforward perceptrons.

such as the sigmoid, the hyperbolic tangent, etc.

- Sigmoid function :

$$f(v) = \frac{1}{1+e^{-v}}$$

- Hyperbolic tangent function :

$$f(v) = \tanh(v) = \frac{e^{v}-e^{-v}}{e^{v}+e^{-v}}$$

Using perceptrons in parallel and cascade structures, more complex decision boundaries and arbitrary Boolean functions can be modeled. However, many different neural network structures can be used to model the mapping. Finding the optimal structure that yields the best performance is not easy to determine. Therefore, there exists rules and algorithms to find an appropriate structure.

The back-propagation algorithm is usually used to train MLP networks. The back-propagation training algorithm employs an iterative gradient-descent method which minimizes the mean squared error between the desired output and the MLP output. The algorithm procedure in Table 3.2 is presented in 10 steps.

Table 3.2 : Backpropagation algorithm.

- ***Step 1*** : Define structure

$$\hat{\mathbf{x}} = \mathbf{u}\mathbf{W}^{(x)}$$
$$\hat{\mathbf{y}} = \mathbf{x}\mathbf{W}^{(y)}$$

where $\boldsymbol{u}$ = vector of N-1 inputs,
$\boldsymbol{x}$ = vector of M-1 hidden nodes,
$\boldsymbol{y}$ = vector of C outputs,
$\mathbf{W}^{(x)}$ = hidden layer weight matrix,
$\mathbf{W}^{(y)}$ = output weight matrix.

Both x_o and u_o are the bias and equal to 1.

- ***Step 2*** : Initialize all weight matrices to random numbers between -0.1 and 0.1.
- ***Step 3*** : Present training pairs (**u**, **d**) to the network.

- ***Step 4*** : Compute output vector of the hidden layer,

$$\mathbf{x} = f(\hat{\mathbf{x}}) = f(\mathbf{u}\mathbf{W}^{(x)})$$

where *f*(.) is an activation function.

- ***Step 5*** : Compute vector of the output layer,

$$\mathbf{y} = f(\hat{\mathbf{y}}) = f(\mathbf{x}\mathbf{W}^{(y)})$$

- ***Step 6*** : Compute the output errors,

$$\mathbf{E}^{(y)} = f'(\hat{\mathbf{y}})(\mathbf{d} - \mathbf{y})$$

Sigmoid derivative : $f'(\hat{\mathbf{y}}) = \mathbf{y}(1 - \mathbf{y})$

Tanh derivative : $f'(\hat{\mathbf{y}}) = (1 + \mathbf{y})(1 - \mathbf{y})$

- ***Step 7*** : Compute hidden layer errors,

$$\mathbf{E}^{(x)} = f'(\hat{\mathbf{x}})\mathbf{W}^{(y)}\mathbf{E}^{(y)}$$

- ***Step 8*** : Update output layer weight matrix,

$$\mathbf{W}^{(y)} = \mathbf{W}^{(y)} + \Delta\mathbf{W}^{(y)}$$

$$\Delta\mathbf{W}_k^{(y)} = \eta\mathbf{x}\mathbf{E}^{(y)} + \beta\Delta\mathbf{W}_{k-1}^{(y)}$$

where η = learning rate parameter.

β = momentum rate parameter.

- ***Step 9*** : Update hidden layer weight matrix,

$$\mathbf{W}^{(x)} = \mathbf{W}^{(x)} + \Delta\mathbf{W}^{(x)}$$

$$\Delta\mathbf{W}_k^{(x)} = \eta\mathbf{u}\mathbf{E}^{(x)} + \beta\Delta\mathbf{W}_{k-1}^{(x)}$$

- ***Step 10*** : Return to step 3 until all training data pairs have been used. Many repeating set of training data may be needed to train the network to a desirable error.

3.3.5 Radial Basis Function Neural Network

Radial basis function (RBF) neural networks have some advantages over other neural network structures :

- For adjusting the parameters in a RBF neural networks, training times are often shorter than for other networks.
- Processes in nature often tend to be Gaussian, and RBF networks can model processes with Gaussian functions.

The RBF network basically consists of an input layer, a hidden layer, and the output layer. Each node in the hidden layer employing *radial basis functions* produces a localized output with respect to the input signals. The outputs are the weighted sums of the outputs of the hidden layer. A three layer RBF neural network is presented in Figure 3.14. The most common radial basis function is the Gaussian kernel function of the form :

$$\Phi_m(\mathbf{u}) = \exp\left[-\frac{(\mathbf{u}-\mathbf{c}_m)^T(\mathbf{u}-\mathbf{c}_m)}{2\sigma_m^2}\right], \qquad m = 1,2,\ldots,M \qquad (3.75)$$

Another common variation on the basis functions is to increase their functionality using the Mahalanobis distance in the Gaussian kernel. The above equation becomes :

$$\Phi_m(\mathbf{u}) = \exp\left[-(\mathbf{u}-\mathbf{c}_m)^T\mathbf{R}_m^{-1}(\mathbf{u}-\mathbf{c}_m)\right], \qquad m = 1,2,\ldots,M \qquad (3.76)$$

where $\mathbf{R}_m^{-1}$ is the inverse of the covariance matrix of **u** associated with hidden node m.

Other functions are the multiquadratic function

$$\Phi_m(\mathbf{u}) = \left(\|\mathbf{u}-\mathbf{c}_m\|^2 + \kappa^2\right)^{1/2} \qquad (3.77)$$

and the inverse multiquadratic function

$$\Phi_m(\mathbf{u}) = \left(\|\mathbf{u}-\mathbf{c}_m\|^2 + \kappa^2\right)^{-1/2} \qquad (3.78)$$

Different basis functions can be utilized for producing different localized outputs.

Learning in the hidden layer is performed using an unsupervised method, typically a clustering algorithm, or some heuristic clustering algorithm to find the cluster centers $\mathbf{c}_m$ such that the weights into hidden node m are the compoments of the "center" vector $\mathbf{c}_m$.

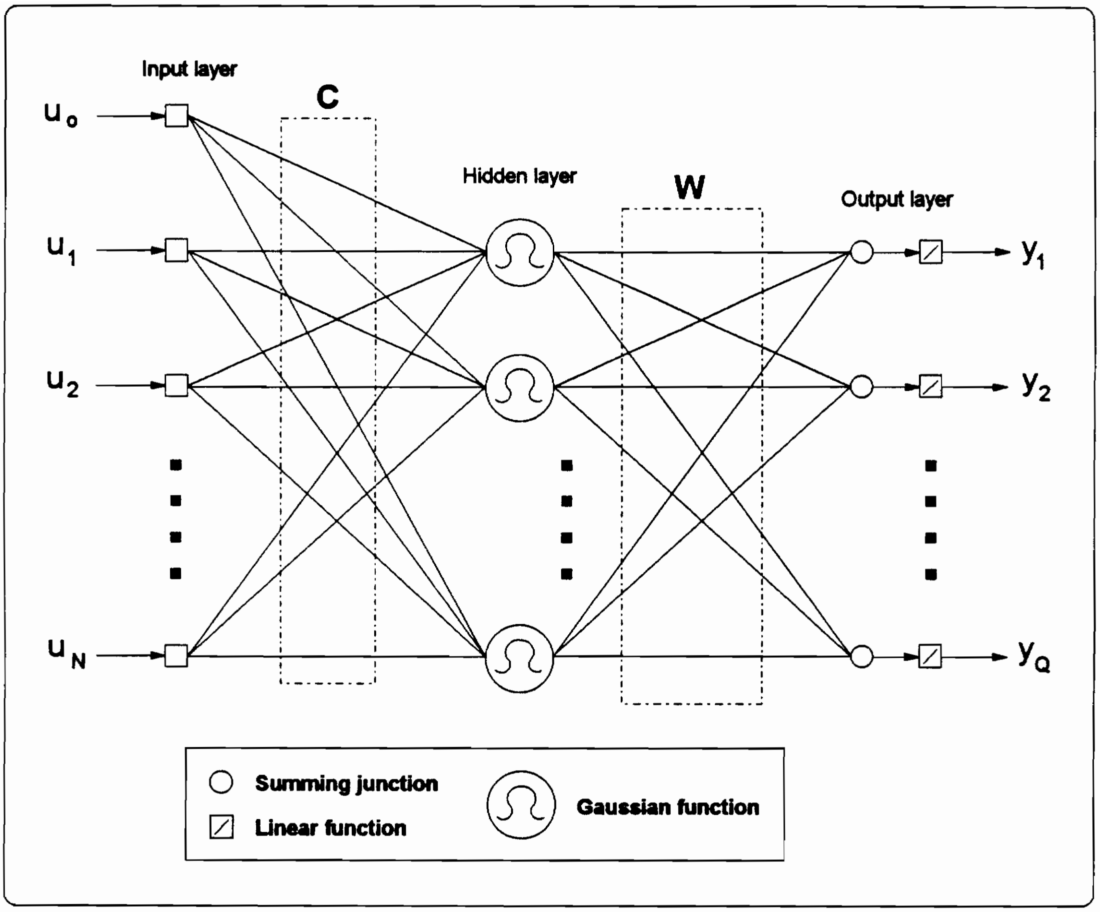


Figure 3.14 Three layer radial basis function neural network.

When representing an input vector **u** to the network, the network implements

$$\mathbf{y} = \mathbf{W} \cdot \Phi(\|\mathbf{x} - \mathbf{c}\|) \tag{3.79}$$

Given some training data with desired responses, the weights **W** in the output layer are adjusted using the least mean square (LMS) algorithm which can be realized iteratively or noniteratively. A noniterative method to solve for W is the singular value decomposition (SVD).

Clustering algorithms are classified as direct and indirect algorithms. Direct (heuristic) approachs attempt to isolate patterns without the presence of a criterion function while the indirect approachs optimize the classification based on some criterion functions. The most common clustering algorithm used to train the hidden layer RBF network is the generalized Lloyd algorithm or the K-means clustering algorithm and is summaried in Table 3.3.

Once the clustering algorithm is complete, the parameters σ_m^2 are computed. They represent a measure of the spread of the data associated with each node. A common choice is to set them equal to the average distance between all training data :

$$\sigma_m^2 = \frac{1}{M_m} \sum_{\mathbf{u} \in \Theta_m} (\mathbf{u} - \mathbf{c}_m)^T (\mathbf{u} - \mathbf{c}_m) \tag{3.80}$$

where Θ_m is the set of training patterns grouped with cluster center $\mathbf{c}_m$, and M_m is the number of patterns in Θ_m .

Selecting the cluster centers randomly is the easiest initialization method. However, this method cannot be guaranteed that the selection made has been representative. Placing the cluster centers on a uniform grid assures that centers are uniformly located. Again, this method only produces good results if the input data is distributed uniformly as well.

Table 3.3 : K-means clustering algorithm.

K-means clustering algorithm

```
• Initialize the cluster centers $\mathbf{c}_m$ , $m$ = 1, 2, ..., M : either randomly
  or the first M training patterns.
• Repeat
          /* Group all patterns with the closest cluster center. */
            for all $\mathbf{u}_n$ do
                  Assign $\mathbf{u}_n$ to $\Theta_{m^*}$ , where $\|\mathbf{u}_n - \mathbf{c}_{m^*}\| = \min_m \|\mathbf{u}_n - \mathbf{c}_m\|$ .
            end;
            /* Compute cluster centers. */
            for all $\mathbf{c}_m$ do
                  $\mathbf{c}_m = \frac{1}{M_m} \sum_{\mathbf{u}_n \in \Theta_m} \mathbf{u}_n$
            end;
• until there is no change in cluster assignments.
```

Chapter 4

ASR Design : System Components

In order to design and build a successful ASR system, many other components need to be designed and built correctly in addition to a good speaker recognition method. In general, the following components are required to be understood and designed first :

- Selection of sound equipment such as sound boards and microphones.
- Design and build an audio wave recorder with many built-in functions such as record, stop, play, save data, open data, frequency, mode, and resolution selections.
- Test background noise in different noisy environments with different sound boards and microphones in order to set various noise level thresholds in the speech endpoint detection design.
- Understand the description of TIMIT speaker database.
- Selection of recording parameters.
- Study the effects of the type of windows and the window length on speech spectrum, and the choices of window length and type.
- Study methods of creating the ASR speaker database and the procedure of speaker registration.
- Understand the conversion of 16 kHz, 16 bits, mono TIMIT waveform to 8 kHz, 16 bits, monoral waveform for telephone bandwidth consideration.
- Study the effects of the first order preemphasis filter on both noise and speech signals.
- Design the speech endpoint detection algorithm with auto noise level estimation, and present its performance evaluation on both TIMIT database and recorded signal.

4.1 Sound Components and Evaluations

The sound components in an ASR system include the sound equipment, an audio wave recorder, and the TIMIT speaker database. An audio wave recorder is programmed to record the speaker voice data via a microphone. The following sound components were used to build a low cost PC-based ASR system : Media Vision ProAudio spectrum 16 sound board, the Audio-Technica ATM11 unidirectional microphone, and a personal computer. In addition, a Ensoniq soundscape 16 sound board and a Realistic unidirectional microphone are employed to test for the equipment variations in different environments. These variations are important factors in designing the speech endpoints.

4.1.1 Sound Board Specifications

A Media Vision ProAudio spectrum 16 sound board was installed to record voice in a normal office environment. The sound board has the following specifications :

- 16 bit linear ADC serial.
- High performance 16 bit DMA computer interface.
- 4 watts / channel stereo amplifier.
- 8, 12, and 16 bit PCM data.
- 4 kHz - 44 kHz sampling frequency.
- Master volume : 0 to -62 dB (1dB/step).
- Input mixer : 1 to -60 dB (2dB/step).
- Sampled audio PCM = 90 dB.
- Total harmonic distortion = 0.05%
- Frequency response : flat 30 Hz - 20 kHz (± 3 dB).
- Shielded circuitry and dynamic filtering 2 kHz - 22 kHz to reduce noise.
- Drive level = 11.5 V (p-p).
- Load impedance = 8 ohms.
- SCSI CD-ROM interface : 690 kB/sec.

This sound board provides a high signal to noise ratio, quality sound, and a flat spectrum over the range of the frequency response.

4.1.2 Audio Wave Recorder

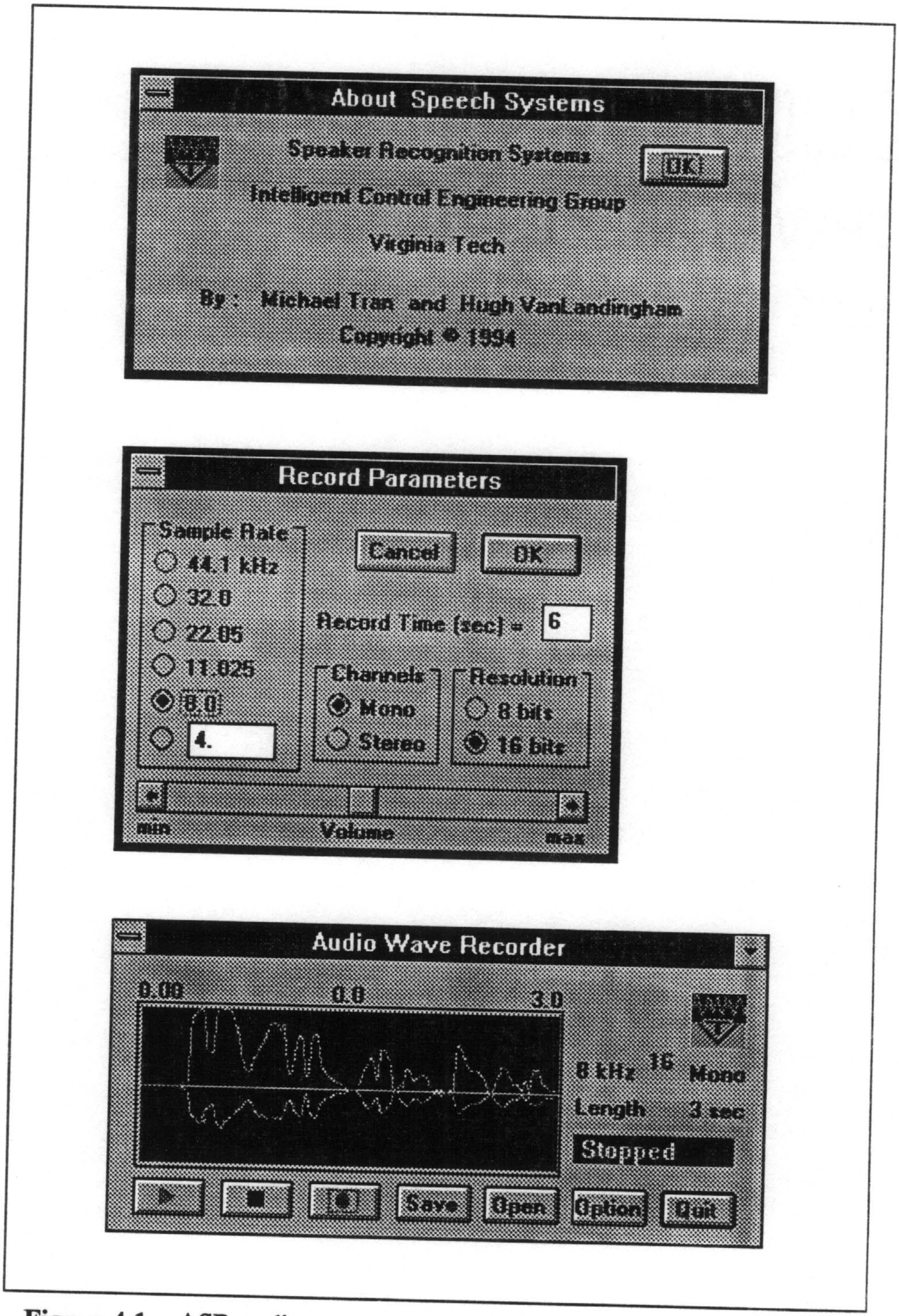
An audio wave recorder is used to record speech waveforms, an absolute requirement for building an ASR system. The audio wave recorder shown in Figure 4.1 provides the following built-in features :

- **Play** : to send the speech signal to a pair of Sony speakers.
- **Stop** : to stop the recorder during playing or recording sessions.
- **Record** : to record the speaker audio voice.
- **Save** : to save a speaker voice to a file to be included in the speaker database.
- **Open** : to open a voice file.
- **Option** : to provide changes in recording parameters.
- **Quit** : to quit the recording session.

The <Audio Wave Recorder> menu also displays the desired recording length, sampling frequency, sample resolution, mono/ stereo mode, the envelope graph of speech signal as well as the real-time play back speech. The <Option> button is used to access the <Record Parameters> dialog menu in order to change the sampling frequency from 4 kHz to 44.1 kHz, the mono/ stereo mode, the sample resolution (8 bits per sample or 16 bits per sample), the speaker volume, and the desired recording time in seconds. The recorder stop automatically at the desired recording time. The recorder will read and save the audio file in a "RIFF" audiowave format. The "RIFF" header format is described below :

• "RIFF"	4 bytes (unsigned char).
• xxxx	4 bytes (unsigned long) = total file size after this line read.
• "WAVE"	4 bytes (unsigned char).
• "fmt "	4 bytes (unsigned char).
• xxxx	4 bytes (unsigned long).
• xx	2 bytes (unsigned int) = format tag PCM = 1.
• xxxx	4 bytes (unsigned long) = sampling rate.
• xxxx	4 bytes (unsigned long) = average bytes per second.
• xx	2 bytes (unsigned int) = block alignment.
• xx	2 bytes (unsigned int) = bits per sample.
• "data"	4 bytes (unsigned char).

- xxxx 4 bytes (unsigned long) = total data size in bytes.
- <speech data>



About Speech Systems

Speaker Recognition Systems

Intelligent Control Engineering Group

Virginia Tech

By : Michael Tran and Hugh VanLandingham

Copyright © 1994

Record Parameters

Sample Rate
○ 44.1 kHz
○ 32.0
○ 22.05
○ 11.025
● 8.0
○ 4.

Record Time (sec) = 6

Channels
● Mono
○ Stereo

Resolution
○ 8 bits
● 16 bits

Audio Wave Recorder

8 kHz 16 Mono

Length 3 sec

Stopped

Figure 4.1 ASR audio wave recorder menu.

4.1.3 Room Noise and Equipment Evaluation

The following noise tests are used to measure the effects of different microphones and sound boards in the office room :

- noise measure with both microphone and speaker off.
- noise measure with microphone on, close to computer.
- noise measure with both microphone and speaker on, close to computer.
- noise measure with both microphone and speaker on, far from computer.

Audio-technica ATM11 unidirectional microphone, Realistic unidirectional microphone, Media Vision ProAudio Spectrum 16, and Ensoniq Soundscape 16 are among the equipment to be evaluated. All background noise test files are recorded at 10 kHz, 16 bits per sample, and monoral channel.

After many measurement tests, it is determined that the speaker on/off has a minimal effect on the measurement level. Tables 4.1, 4.2, and 4.3 summarize the measurement results of the "silent" room noise on three tests. The first test in Table 4.1 measures the mean, standard deviation, minimum energy, maximum energy, and average energy of the room noise with microphone off. The second test measures the room noise with microphone close to the running computer and the results are shown in Table 4.2. Table 4.3 shows the results of the last test with the microphone far from the running computer. In general, the Media Vision ProAudio Spectrum 16 sound board with either the Realistic or Audio-Technica ATM11 microphone has smaller mean values over the Ensoniq Soundscape 16 sound board with Realistic microphone. All measurements in Tables 4.1, 4.2, and 4.3 are based on 80 point frames.

Table 4.1 : Silent room noise with microphone off.

Raw signal	Mean	Std.	Min energy	Max energy	Avg. energy
MV & Realistic mike	-52.11	6.53	52.54 dB	53.95 dB	53.43 dB
MV & ATM11 mike	-63.39	21.56	53.76 dB	56.85 dB	55.51 dB
ES & Realistic mike	2635.8	197.45	79.36 dB	87.61 dB	87.43 dB

Table 4.2 : Silent room noise with microphone on close to running computer.

Raw signal	Mean	Std.	Min energy	Max energy	Avg. energy
MV & Realistic mike	-52.01	59.67	48.78 dB	61.89 dB	56.18 dB
MV & ATM11 mike	-62.88	41.39	53.65 dB	59.13 dB	56.45 dB
ES & Realistic mike	2639	389	80.07 dB	89.13 dB	87.47 dB

Table 4.3 : Silent room noise with microphone on far from running computer.

Raw signal	Mean	Std.	Min energy	Max energy	Avg. energy
MV & Realistic mike	-51.44	21.22	52.40 dB	55.26 dB	53.91 dB
MV & ATM11 mike	-61.50	20.04	54.15 dB	56.12 dB	55.23 dB
ES & Realistic mike	2648	199	79.44 dB	87.70 dB	87.47 dB

where MV = Media Vision ProAudio Spectrum 16 sound board.

ES = Ensoniq Soundscape 16 sound board.

From the three tables above, the following observations are noted :

- The average value of background noise varies greatly among sound boards.
- The mean of background noise is not close to zero (biased).
- The noise variation increases with microphone on.
- The background noise energy variation is about 7 dB versus its average energy and about 13 dB versus its max energy.

Figure 4.2 shows graphs of the raw noise signal with ATM11 microphone off, with ATM11 microphone on, and the spectral plots of different sound boards and microphones. In Figure 4.2, the spectral plots are generated from the short-time FFT performed over each 256 point Hamming window of the signal. The FFT spectrum of the background noise is flat over frequency range and there is no indication of the 60 Hz AC power interference in both sound boards. As we will see later, the knowledge of the background noise, equipment, and variations are very important for the speech endpoint detection and, in turn, to the success of the speaker recognition system.

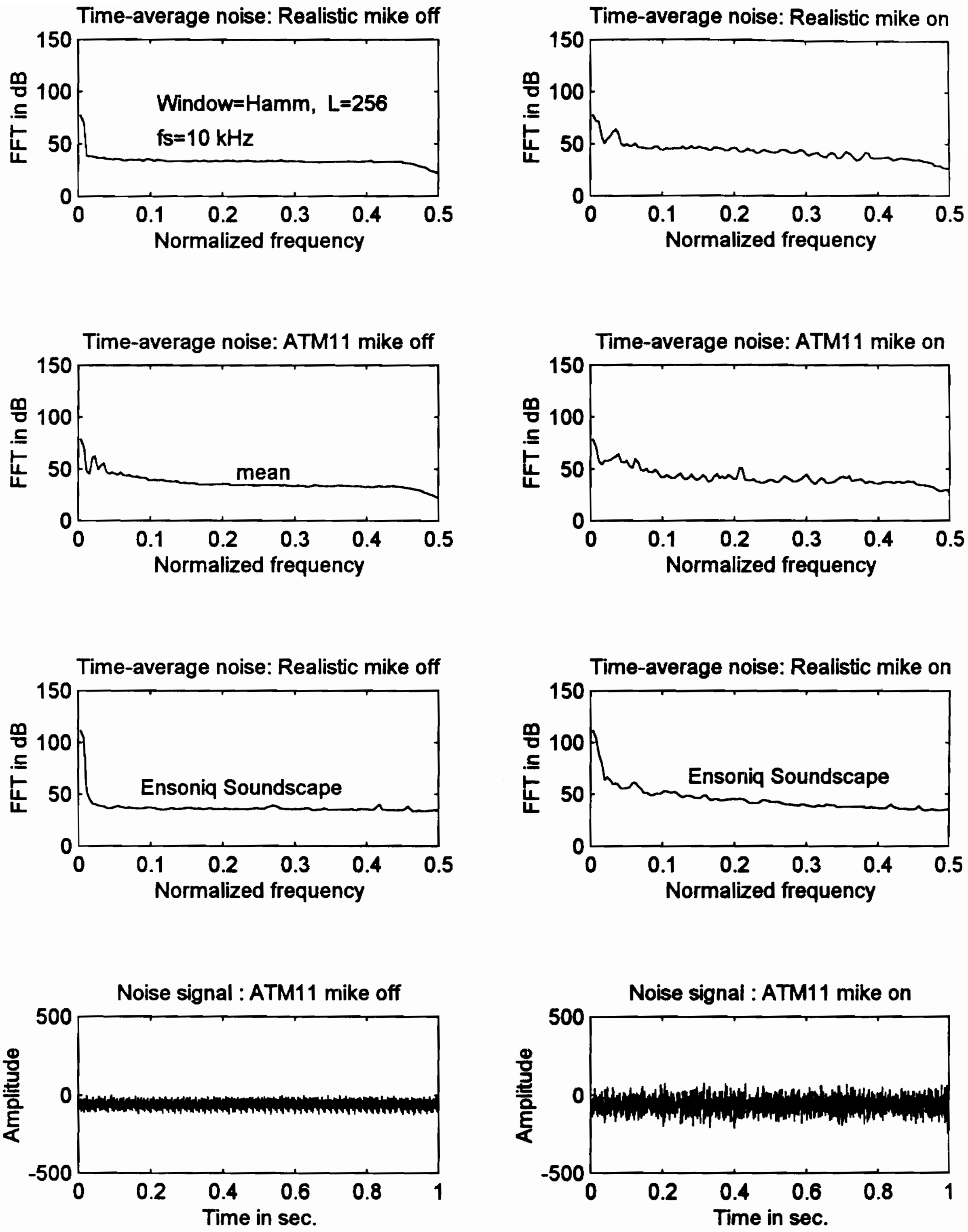


Figure 4.2 Plots of background noise signals and their FFT spectra.

4.1.4 TIMIT Database

The Texas Instruments/ Massachusetts Institute of Technology (TIMIT) database is used to evaluate the text-independent speaker recognition system. In the TIMIT database, 630 speakers from 8 major dialect regions of the United States, each spoke 10 utterances for a total of 6300 utterances. There are 70% male speakers and 30% female speakers to make up the speaker population. The text material of the database includes 2 dialect sentences (SA*.WAV), 450 phonemically-compact sentences (SX*.WAV), and 1890 phonetically-diverse sentences (SI*.WAV). Each speaker speaks 2 identical dialect sentences, 5 phonemically-compact sentences, and 3 phonetically diverse sentences. The SA sentences are used to expose dialectal variants of the speakers. The two SA sentences are "She had your dark suit in greasy wash water all year." and "Don't ask me to carry an oily rag like that." The SX sentences are designed to provide a good coverage of pairs of phones. The SI sentences are used to add diversity in sentence types and phonetic contexts. The dialect distribution of speakers is presented in Table 4.4.

There is a information header of fixed 1024 bytes preappended to the wave data file. In general, all speech utterances are recorded at a 16 kHz sampling rate, monoral channel, and 16 bits per sample. The number of samples in the file, sample min, and sample max are also provided in the header.

Table 4.4 : Dialect distribution of speakers.

Dialect Region (DR) Name	DR Code	# Male Speakers	# Female Speakers	Total # Speakers
New England	1	31 (63%)	18 (27%)	49 (8%)
Northern	2	71 (70%)	31 (30%)	102 (16%)
North Midland	3	79 (67%)	23 (23%)	102 (16%)
South Midland	4	69 (69%)	31 (31%)	100 (16%)
Southern	5	62 (63%)	36 (37%)	98 (16%)
New York City	6	30 (65%)	16 (35%)	46 (7%)
Western	7	74 (74%)	26 (26%)	100 (16%)
Army Brat	8	22 (67%)	11 (33%)	33 (5%)
Total # Speakers	1-8	438 (70%)	192 (30%)	630 (100%)

4.2 System Components Design Overview

In this section, the following system design components to be used in the design of the automatic speaker recognition system are presented :

- Voice recording parameters.
- ASR speaker database generation and registration.
- TIMIT waveform down sampling from 16 kHz to 8 kHz.
- Choice of window type and window length.
- First order preemphasis filter.
- Speech endpoint detection algorithm.

All system design components contribute greatly to the accuracy of the final ASR system.

4.2.1 Voice Recording Parameters

A high sampling frequency of speech signals results in higher fidelity voice signals for each speaker and in turn a higher recognition rate. The costs associated with a high sampling frequency are higher storage of voice data required, more memory to process data, and a longer recognition time. The most important cost is that it cannot be used with a telphone channel where an automatic speaker recognition (verification) system has many useful applications such as in the banking service. Therefore, it is desirable to design an ASR system that works in the telephone frequency range.

In this ASR system, the audio wave recorder is used to record a speaker voice with the following parameters :

- Sampling frequency : 8 kHz.
- Sample resolution : 16 bits.
- Recording channel : mono.

The above recording parameters are close to the telephone range and are commonly used in most ASR system. The design algorithm also works well with a higher sampling frequency.

4.2.2 ASR Speaker Database Registration

In general, the audio wave recorder is used to record the speaker voice first. Next, the speaker features are extracted from the voice data. Then the speaker features are added to the ASR speaker database. The above procedure is used to register a speaker to the speaker database and is shown in Figure 4.3. For the purpose of complete evaluations of the system accuracy, the ASR system also reads the TIMIT speaker database, then converts the 16 kHz, 16 bits, monoral voice data to 8 kHz, 16 bits, monoral voice data. The conversion is done by passing the TIMIT signal through an anti-aliasing lowpass filter with a cutoff frequency of 4 kHz and then decimating the sequence data by two. Next, speaker features are extracted from the new wave data and are added to the ASR speaker database as shown in Figure 4.4.

A total of 462 speakers consisting of 136 female speakers and 326 male speakers in the train section of the TIMIT database is used to determine the accuracy of the ASR system. The distribution of the speakers is provided in Table 4.5. For each speaker, the five phonemically-compact sentences (SX*.WAV) are combined into one training utterance for speaker registering purpose, while the other five sentences (2 SA*.WAV & 3 SI*.WAV) are used for testing purposes (identification / verification).

Table 4.5 : Distribution of ASR speaker database.

Dialect Region (DR) Name	DR Code	# Male Speakers	# Female Speakers	Total # Speakers
New England	1	14	24	38
Northern	2	23	53	76
North Midland	3	20	56	76
South Midland	4	15	53	68
Southern	5	25	45	70
New York City	6	13	22	35
Western	7	18	59	77
Army Brat	8	8	14	22
Total # Speakers	1-8	326	136	462

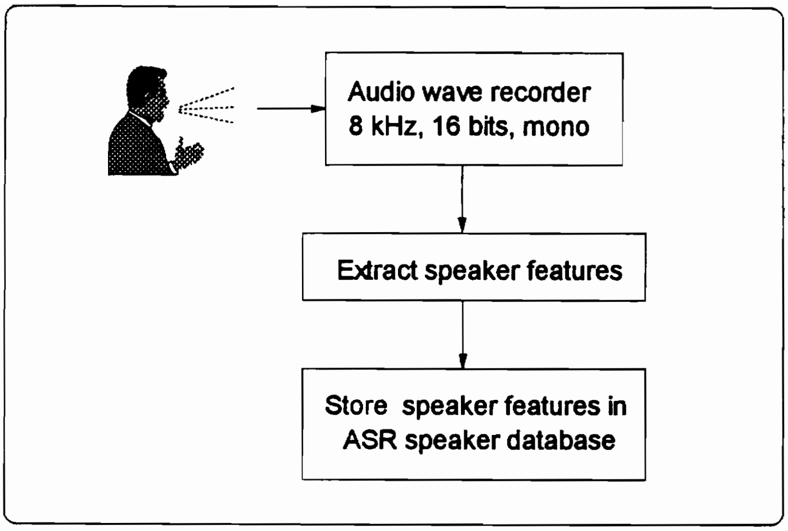


Figure 4.3 ASR speaker database using audio wave recorder.

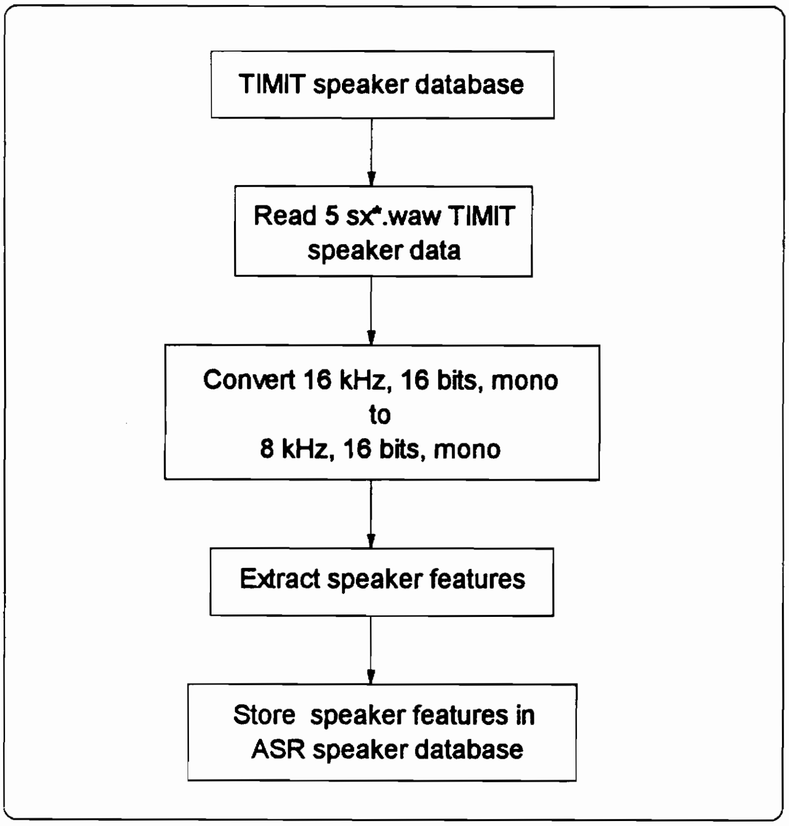


Figure 4.4 ASR speaker database from TIMIT.

4.2.3 Frequency Decimation

In Figure 4.4 above, the 16 kHz sampling rate in the TIMIT data is decreased to an 8 kHz sampling rate. This process is called ***decimation*** because the original sample set is reduced in number. The decimation procedure is depicted in Figure 4.5 where the original signal is passed through a 50 point Hamming FIR lowpass digital filter with a cutoff frequency of 4 kHz, and the output of the lowpass digital filter is decimated by 2. The resulting resampled output sequence is two times shorter than the original sequence. The frequency response of the anti-aliasing lowpass filter is plotted in Figure 4.6. The flat unity amplitude (0 db) with no ripple in the passband and linear phase response of the FIR digital filter are preferred since it will not distort any frequency spectrum below 4 kHz.

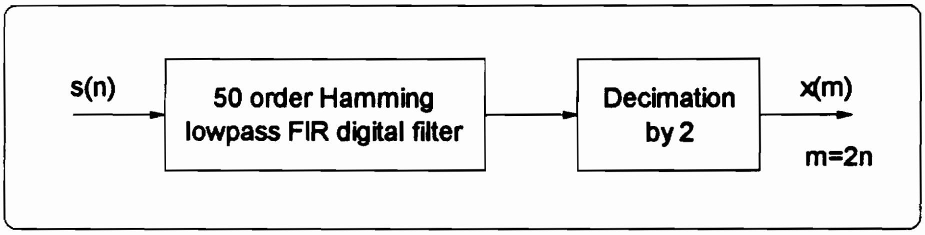


Figure 4.5 Frequency decimation by 2.

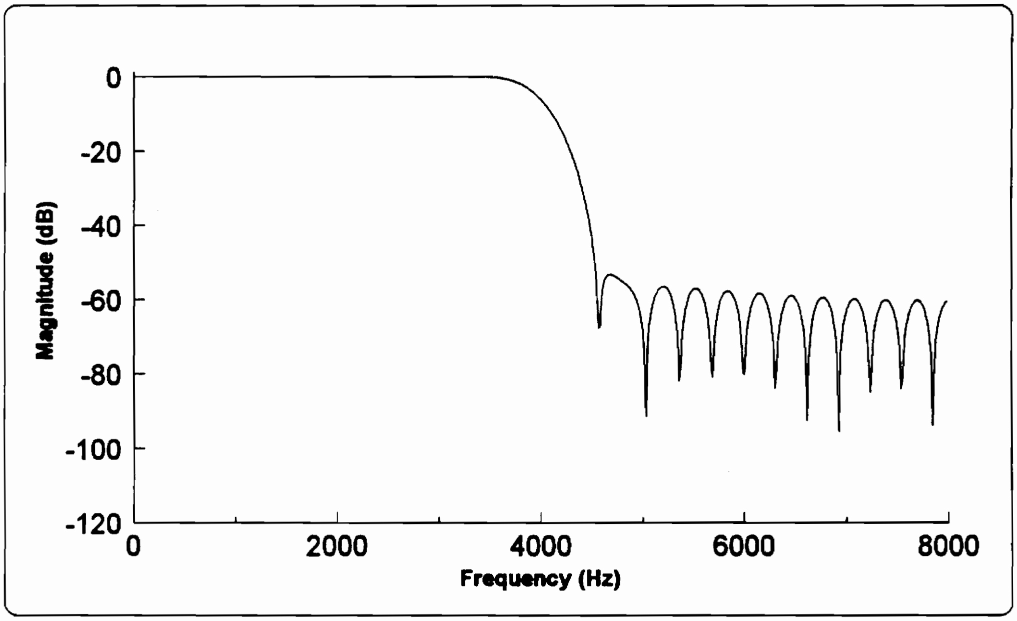


Figure 4.6 Magnitude response of a 50 order Hamming FIR lowpass digital filter.

4.2.4 Choice of Windows

Since the vocal tract shape changes slowly with time to produce different sounds, a window is used in speech processing to divide continuous speech into segments (called frames) which are assumed to be stationary in a short period of time. The purpose of the bell-shaped windows is to taper the signal near n = m and near n = m+N-1 so as to minimize the errors at section discontinuity boundaries. The choice of window and its duration are important in speech analysis because the frequency response of both the signal and window are convolved together in the frequency domain. It is desirable to have a narrow main-lobe bandwidth and a large attenuation in the sidelobes. Narrower main lobe bandwidth provides sharper spectral details, and large attenuation in the sidelobes reduces the spectral effects from the discontinuities at the ends of the window interval. Commonly used windows in speech analysis include the rectangular window, Hamming window, Hanning window, and Blackman window. Figure 4.7 shows the magnitude plots of these windows and their magnitude spectra. Hamming window is chosen due to the combination of its narrow main-lobe bandwidth and its -41 dB peak sidelobe attenuation.

4.2.5 Choice of Window Length

In general, the speech spectrum is a collection of many short-time Fourier transforms of windowed speech segments. There is a tradeoff in choosing a size of the window. A large window length N provides high frequency resolution to resolve individual formant frequencies at the cost of low time resolution. A small window length M has high time resolution for good estimation of the speech spectral envelope at the cost of low frequency resolution. Figure 4.8 shows the FFT plots of 15, 30, and 60 msec Hamming windowed and rectangular windowed speech of first letter <o> of <obey>. A 30 ms window length equivalent to a frequency of 33 Hz is chosen since it has enough resolution to cover the male pitch range (50-250 Hz) and the female pitch range (120-500 Hz). From Figure 4.8, the spectra use of either window provides similar results. However, the spectra of Hamming windowed segments of speech is smoother than the spectra of the rectangular window segments.

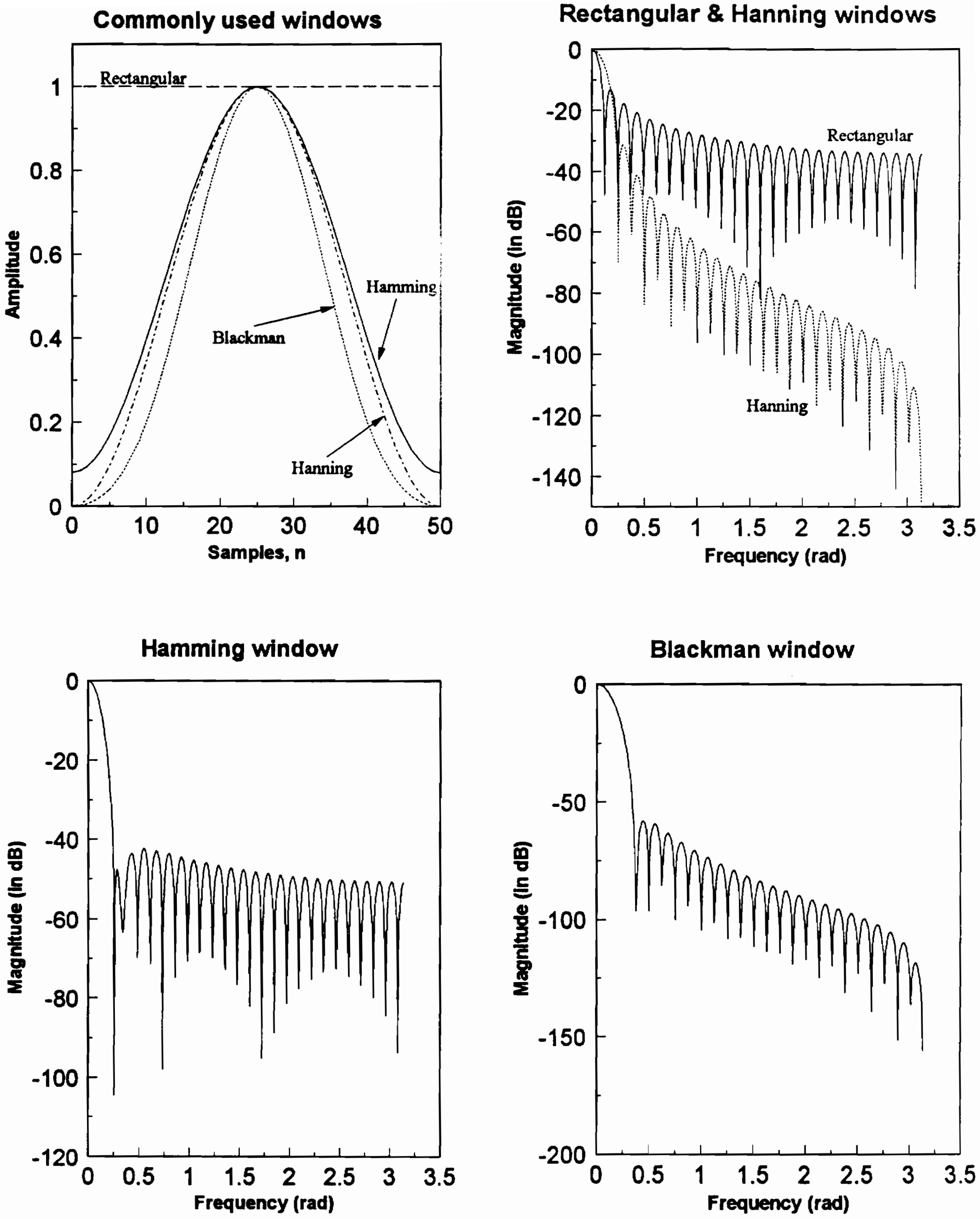


Figure 4.7 Commonly used windows and their Fourier transforms.

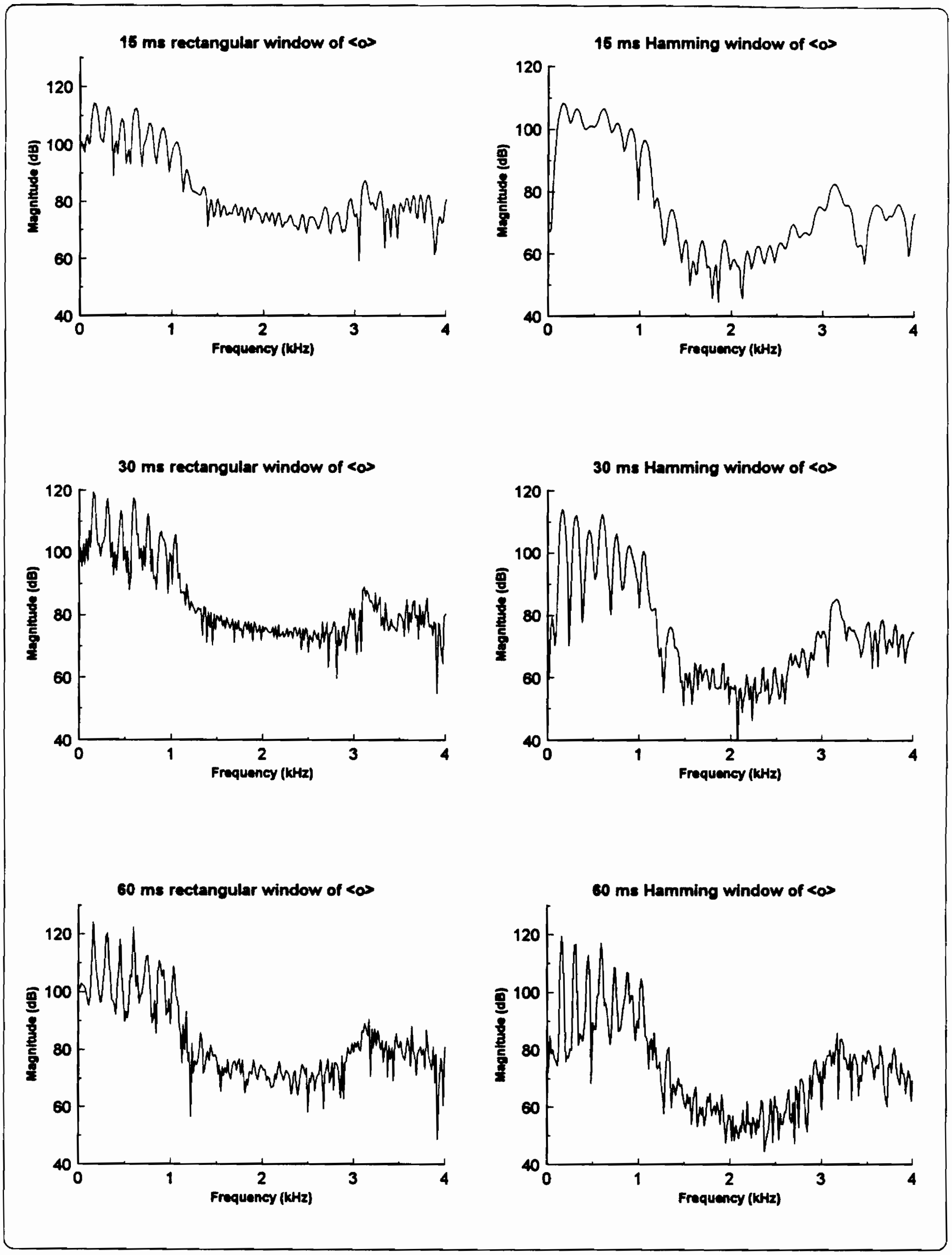


Figure 4.8 FFT plots of different window length using both rectangular and Hamming windows.

4.2.6 Signal Preemphasis

The first order preemphasis filter is used to equalize the inherent spectral tilt in speech and has the following form :

$$H(z) = 1 - az^{-1}, \qquad 0.9 \le a \le 1.0 \qquad (3.14)$$

This filter introduces a zero near w = 0 Hz, and a 32 dB boot at w = π in magnitude over that at w = 0 as illustrated in Figure 4.9. There are two reasons to use this filter. First, the filter output signal is spectrally flatter and thus is less susceptible to finite precision effects. Second, since the minimum phase component of the glotal signal can be modeled by a simple two poles near z = 1, the lip radiation effect and this preemphasis filter introduce two zeros near z = 1 to eliminate the above two poles. Now, the signal is assumed to be characterized by the vocal tract only. In Figure 4.9, the preemphasis filter also reduces the noise level and gives a boost to the high frequency spectrum of the letter <o>.

The effects of the preemphasis filter on the noise levels with different environments and equipments are summaried in Tables 4.6 and 4.7. In general, all measurements are reduced significantly when compared to measurements in Tables 4.1, 4.2, and 4.3.

Table 4.6 : Room noise with microphone on with preemphasis of **a = 0.95**.

Raw signal	Mean	Std.	Min energy	Max energy	Avg. energy
MV & Realistic mike	-2.6	20.65	40.87 dB	50.89 dB	45.15 dB
MV & ATM11 mike	-3.15	19.42	42.71 dB	47.33 dB	44.82 dB
ES & Realistic mike	132.15	33.57	56.50 dB	63.44 dB	61.63 dB

Table 4.7 : Room noise with microphone on with preemphasis of **a = 1**.

Raw signal	Mean	Std.	Min energy	Max energy	Avg.energy
MV & Realistic mike	0.0027	21.25	40.43 dB	51.09 dB	45.17 dB
MV & ATM11 mike	-0.0052	19.81	42.53 dB	47.49 dB	44.88 dB
ES & Realistic mike	0.232	28.68	45.01 dB	57.54 dB	47.83 dB

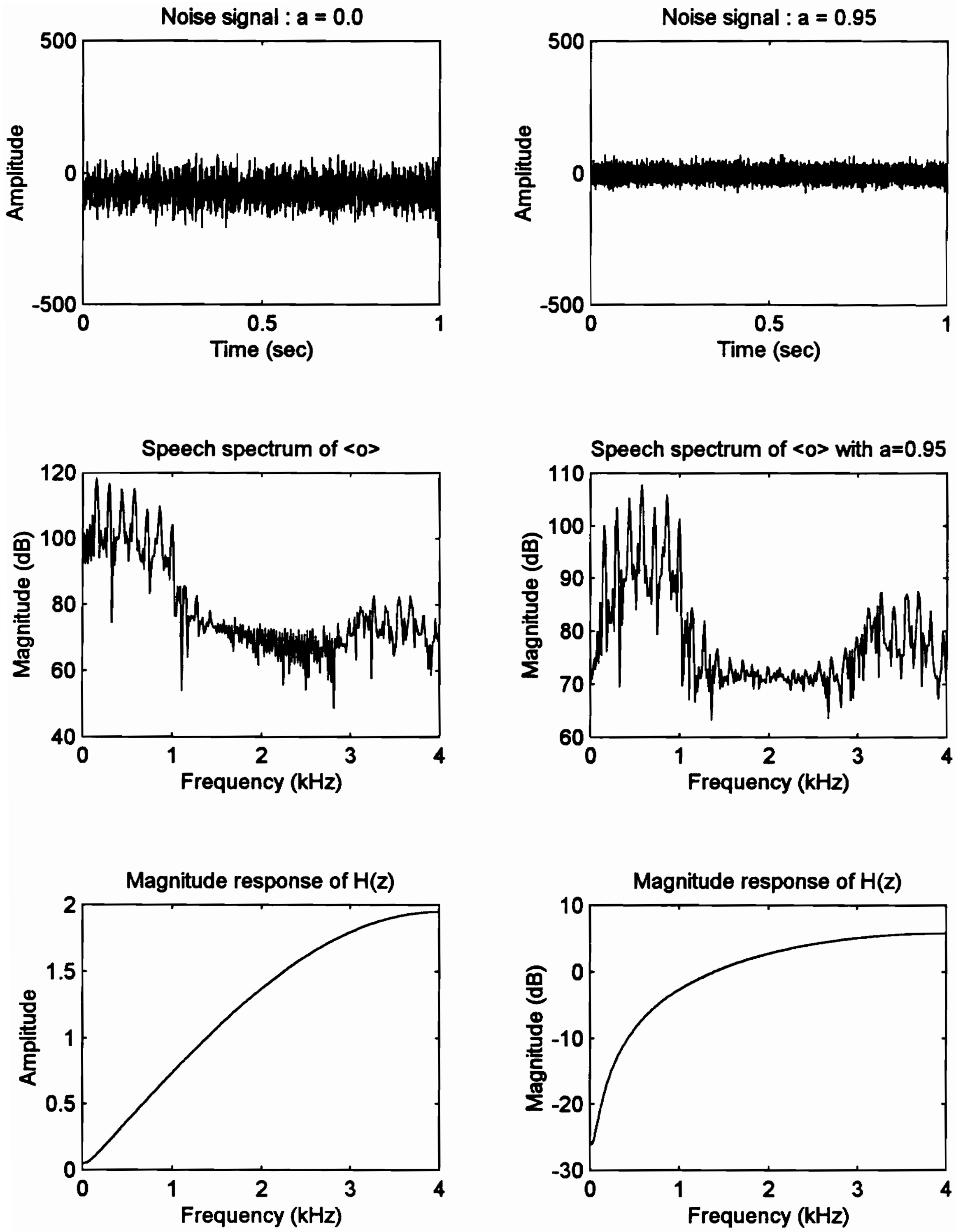


Figure 4.9 Plots of noise and speech spectrum with / without preemphasis filter, and plots of the magnitude response of H(z).

4.2.7 Speech Endpoint Detection

In this ASR system speech endpoint detection is used to detect the presence of speech, to remove pauses and silences in a background noise. The block diagram of the design of the speech endpoint detection is presented in Figure 4.10. The speech endpoint detection will read the noise data in a specific file to determine the threshold of the maximum noise energy. This noise energy level along with empirical noise energy variations in different environments and different equipments, and minimum valid speech energy level are combined to set various thresholds and timing in the detection algorithm. First the speech signal is divided into 10 ms frames with 50 % overlap. The detection algorithm goes through frame by frame, keeping the valid speech signal frame, and throwing away the silence/ pause frame according to conditions of various setpoints and time criteria. After processing all frames, all valid speech signal frames are joined together sequentially to create the new all-speech data for speaker features extraction later.

The performance of the speech endpoint detection is illustrated in Figure 4.11 through the plots of two examples: a TIMIT speech signal and a ATM11 recorded speech signal. In each example, both the original speech signal and the new speech signal with the removal of pauses/silences are presented. The 'x' denotes the start of the valid speech signal, and the 'o' denotes the end of the valid speech signal in Figure 4.11. This ASR speech endpoint detection is tested on about 600 TIMIT sentences, and the plots of speech detection of each sentence are examined visually to assure its accuracy. In general, it performs well. In addition, this endpoint detection is designed to step over very low signals and weak unvoiced sounds for better speaker recognition performance. The frame energy is computed using the equation below :

$$E_m = \sum_{n=m-L+1}^{m} s^2(n) \qquad (4.1)$$

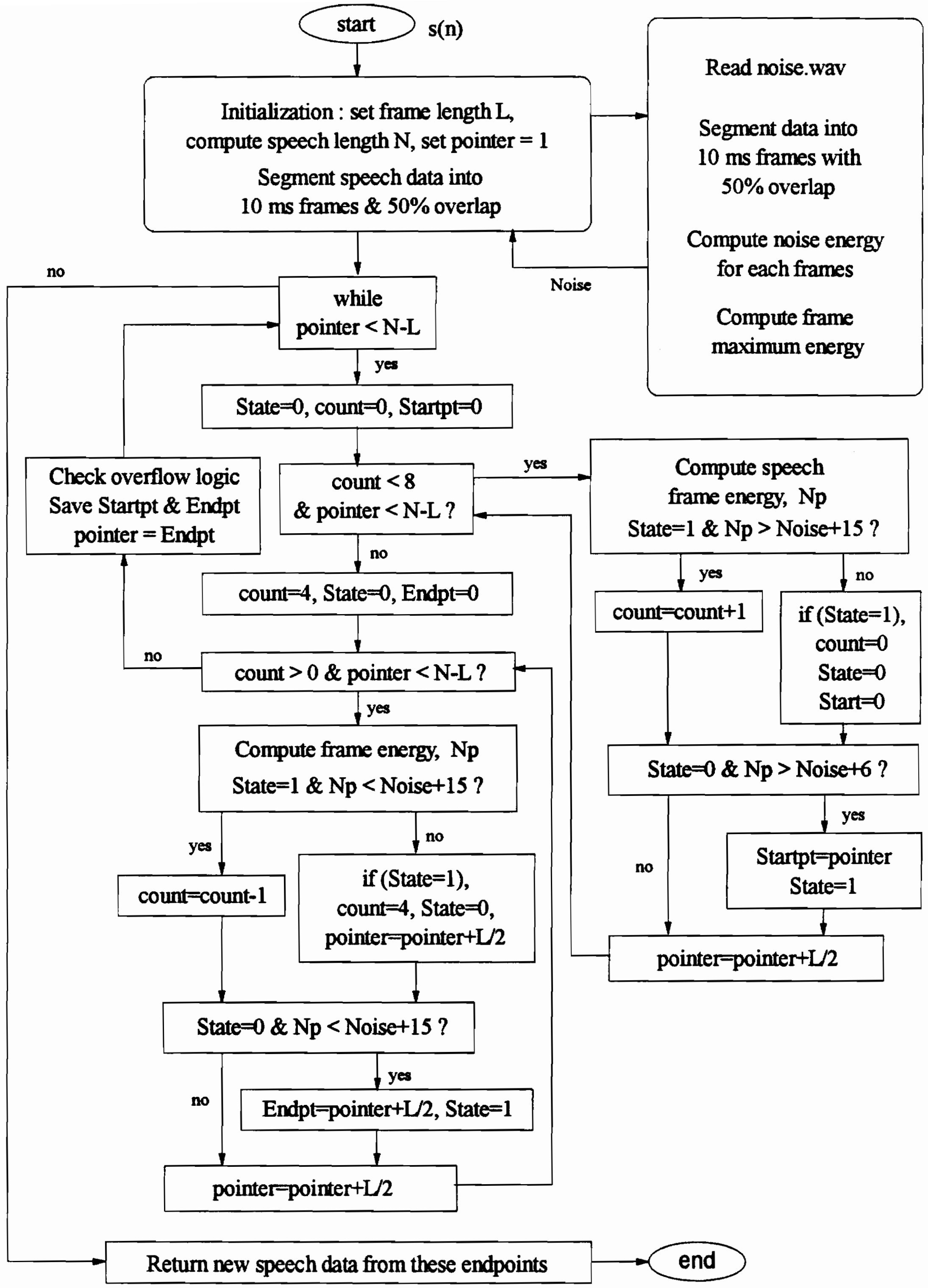


Figure 4.10 Block diagram of the ASR speech endpoint detection.

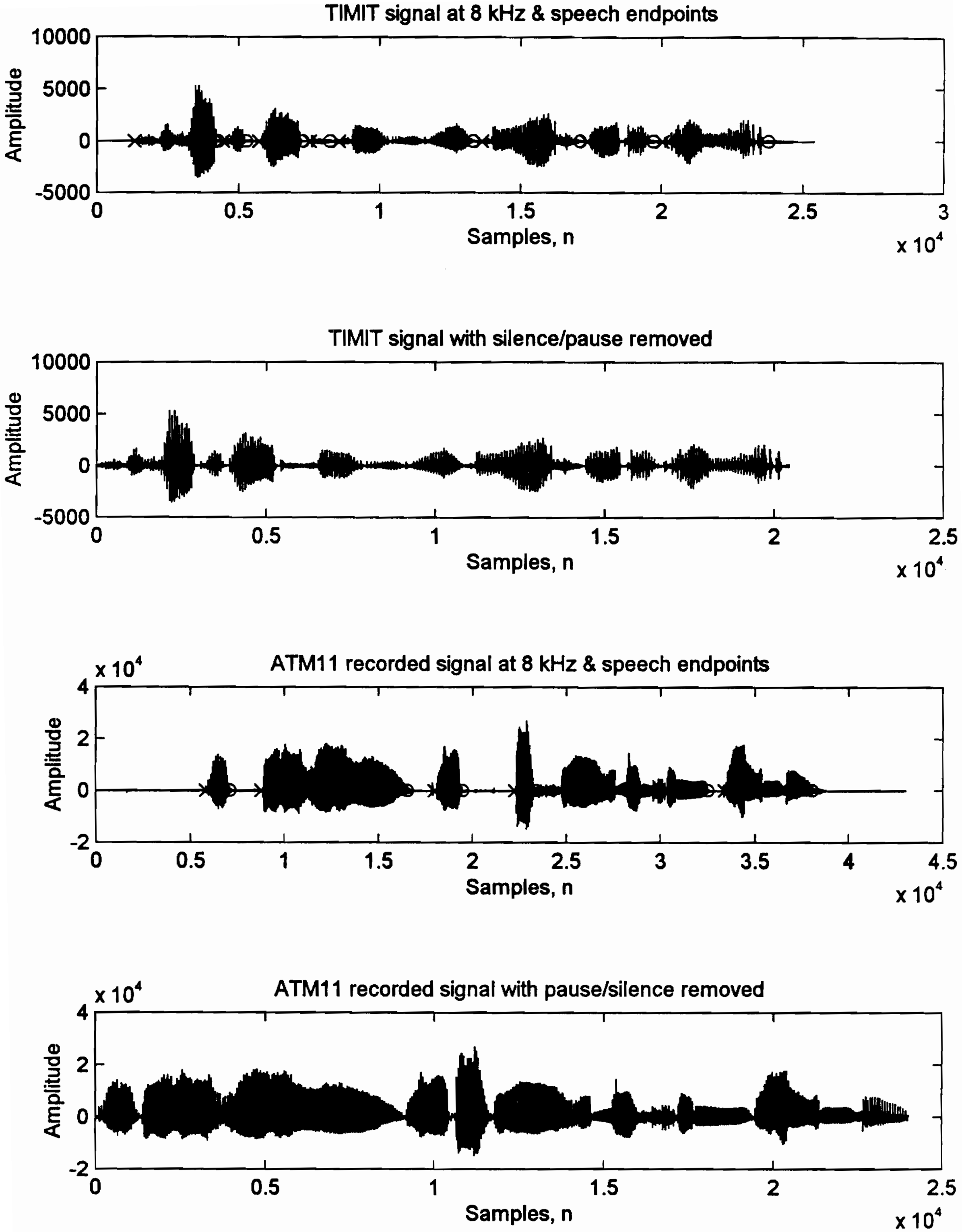


Figure 4.11 Plots of two original speech signals vs. two no silence/pause speech signals, where 'x' = speech start & 'o' = speech end.

Chapter 5

ASR Design : Features and Analysis

In this chapter, various methods and spectral distances are used to search for speaker differences that enable text-independent speaker recognition. Primarily, the linear predictive coefficients and their derived coefficients are used as speaker features in this ASR system particularly the LPC parameters, **a**, the LPC PARCOR, **k**, and the LPC cepstral, **c**. To evaluate the performance of the ASR system on different methods, it is necessary to build the ASR train speaker features database first. Then, long-term spectral means, variances, and medians are employed to evaluate the effectiveness of each feature. Next, these moments of speaker features together with different proposed spectral distances are used to compute the errors of the speaker recognition system. Different methods of combining spectral features are also exploited to improve the performance of the system. Furthermore, the system accuracies for both female speakers and male speakers on both text-independent speaker identification and text-independent speaker verification are also presented.

5.1 ASR Speaker Feature Databases

This section will describe the method to build the ASR speaker feature datbases from TIMIT database or from the audio wave recorder. The ASR train and test speaker databases are built for 462 speakers in the train section of TIMIT database. The autocorrelation method and Durbin's recursive algorithm are used to extract the linear predictive parameters. Moments of a distribution, including mean, median, variance, and absolute deviation, are introduced and employed to generate the speaker features. Next, different windows and preemphasis filters are used to evaluate their effectiveness.

5.1.1 ASR Train and Test Speaker Databases

In the train section of the TIMIT database, 462 speakers from 8 major dialect regions of the United States, each spoke 10 sentences for a total of 4620 sentences. The 2310 SX sentences are used to create the ASR train speaker database and the 2310 SA & SI sentences are used to create the ASR test speaker database. In general, 5 SX sentences from each speaker are combined into one train sentence for this speaker which is converted to an 8 kHz, 16 bits, monoral signal sentence, and stored in the ASR train speaker database. The same procedure is applied to the other 5 SA & SI sentences of the speaker to create the ASR test speaker database as shown in Figure 5.1. From these two databases, speaker features are extracted to build the train / test speaker features databases to evaluate the accuracy of the ASR system.

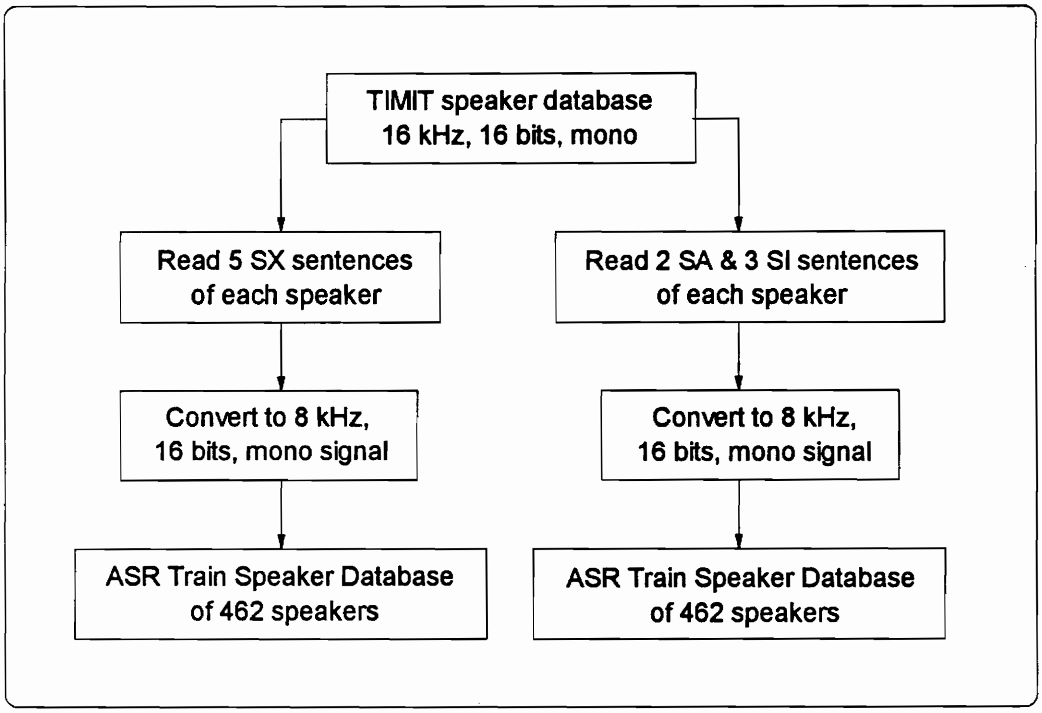


Figure 5.1 ASR Train Speaker Database and Test Speaker Database.

5.1.2 Speaker Spectral Features

The ASR system employs the following spectral predictive parameters to estimate the speaker spectral features :

- 12^{th} order LPC parameters **a**.
- 12^{th} order LPC PARCOR parameters **k**.
- 12^{th} order LPC cepstral parameters **c**.

In this ASR system, the linear predictive coding is selected to model the present speech signal as a linear combination of past signal values :

$$s(n) = Gu(n) + \sum_{k=1}^{12} a_k s(n-k) \tag{5.1}$$

As mentioned in Chapter 3, using the method of least squares, the parameters a_k can be determined from :

$$\Phi_{io} = \sum_{k=1}^{12} a_k \Phi_{ik} \tag{5.2}$$

where $\Phi_{ik} = \sum_{n} s(n-i)s(n-k)$ (5.3)

Two principle methods to solve Eq. (5.2) for the a_k coefficients are the autocorrelation method and the covariance method. The autocorrelation method assumes that the signal exists inside a window of length N, and equals to zeros outside the window which can be achieved by multiplying the signal s(n) by a window w(n) :

$$s(m) = \begin{cases} s(m+n)w(m), & 0 \le m \le N-1 \\ 0, & otherwise. \end{cases} \tag{5.4}$$

In this ASR system, the autocorrelation method is selected to solve Eq. (5.2) to give

$$\sum_{k=1}^{12} a_k R(|i-k|) = R(i) \qquad 1 \le i \le 12 \tag{5.5}$$

where $R(i) = \sum_{n=0}^{N-1-i} s(n)s(n+i)$ (5.6)

N = window length.

and the Durbin's recursive algorithm in Table 3.1 are used to solve the Toeplitz matrix Eq. (5.4), and to compute the LPC coefficients **a**, and the PARCOR coefficients **k**.

The 12th order LPC cepstral coefficients are computed as shown below :

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m} \right) c_k a_{m-k} \ , \qquad 1 \le m \le 12 \tag{5.6}$$

Moments of these LPC derived spectral features are used to characterize each speaker in the ASR speaker features database, and will be discussed next.

5.1.3 Moments of a Distribution

Moments of a distribution, such as mean, variance, standard deviation, absolute deviation, skewness, and so forth, are usually employed to characterize a set of data that has strong tendency to cluster around some particular value. There are many advantages of using these parameters. Depending on the conditions of the data set, these parameters can accurately describe the characteristics of the data set and can reduce the required amount of storage space. The second advantage is the ease of the computation of these parameters, making them useful for comparison.

The *mean* value of a set of N data values is the estimate value around which central clustering occurs and is defined by :

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{5.7}$$

The mean value may be a poor estimate for data values drawn from a probability distribution with very broad tails. The altermative estimator is the median.

The median of a distribution is estimated from N sample values by finding the value x_{med} which has equal numbers of values above it and below it. If the N samples of data are sorted in ascending order, then the median is defined by :

$$x_{med} = x\left(\frac{N+1}{2}\right), \quad \text{N odd} \tag{5.8a}$$

or

$$x_{med} = \frac{1}{2}\left[x\left(\frac{N}{2}\right) + x\left(\frac{N}{2}+1\right)\right], \quad \text{N even} \tag{5.9b}$$

The median estimator fails only if the area in the tails is large, while the mean fails if the first moment of the tails is large even though their area is negligible.

The *variance* value of a set of N data values is used to measure the spread of the data around its mean value, and is defined by :

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 \tag{5.10}$$

The square root of the variance is called the *standard deviation.*

The mean depends on the first moment of the data, and the variance depends on the second moment of the data. There exists a more robust estimator of the spread of the data called mean absolude deviation. The *mean absolute deviation* is defined by :

$$mad_x = \frac{1}{N}\sum_{i=1}^{N}|x_i - \overline{x}| \tag{5.11}$$

Historically, this parameter has not been popular because it is analytically less tractable and makes theorem proving difficult. In recent years, it has become popular and an important estimate for broad distributions with significant numbers of "outlier" points.

The skewness, or third moment is used to estimate the degree of asymmetry of a distribution around its mean, and is defined by :

$$skew_x = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{x_i - \overline{x}}{\sigma}\right]^3 \tag{5.12}$$

A positive value of the skewness denotes a distribution whose tail extends towards more positive x. A negative value of the skewness denotes a distribution whose tail extends towards more negative x.

5.1.4 ASR Train Speaker Features Databases

The ASR system will read each train utterance of each speaker from the ASR train speaker database of 462 speakers. Then the 8 kHz, 16 bits, monoral utterance signal s(n) is passed thru the speech endpoint detection algorithm to remove pauses, silences, and weak unvoiced sound signals. The resulting signal is then passed separately to a 30 msec rectangular window, 30 msec Hamming window, and first-order preemphasis filter of a=0.95 and 30 msec Hamming window as shown in Figure 5.2. Next, the 12^{th} order LPC **a**, **k**, and **c** are computed every 30 msec (240 points window) in each 15 msec frames. These LPC features are continued to store until the end of the speech utterance. Finally, the ASR system will compute the mean, variance, skewness, median, and mean absolute deviation for each branch, and store these spectral moments in the ASR speaker features database. The same procedure is used for all 462 speakers. In short, Figure 5.2 presents three different ways to generate speaker features :

- Rectangular window : **A, K, C** moments for all 462 speakers.
- Hamming window : **A, K, C** moments for all 462 speakers.
- First order preemphasis filter and Hamming window : **A, K, C** moments for all 462 speakers.

where **A, K, C** are matrices storing **a**, **k**, and **c** for all speakers.

The varieties of different ways to collect features and different feature types are used to analyze the importance and the usefulness of each feature. The typical PARCOR feature **K** plots of clusters of speaker means, variances, absolute deviations, and skewness are presented in Figure 5.3. Figure 5.3 also provides plots of typical female/ male values of these moments from the PARCOR **K**. Figure 5.4 shows the plots of clusters of medians, and typical female/ male medians for **A**, **K**, and **C**. From these two figures, the dimension of recognizing speakers from their voices does not appear to be easy since features of all speakers' clusters are close to each other. The question is which windows, and parameters among **A, K, C** provide high recognition accuracy. Does the first-order preemphasis filter make a significant difference? The next several sections will propose different methods for measuring the spectral dissimilarity, and analyze the importance of each feature and the different combinations of features.

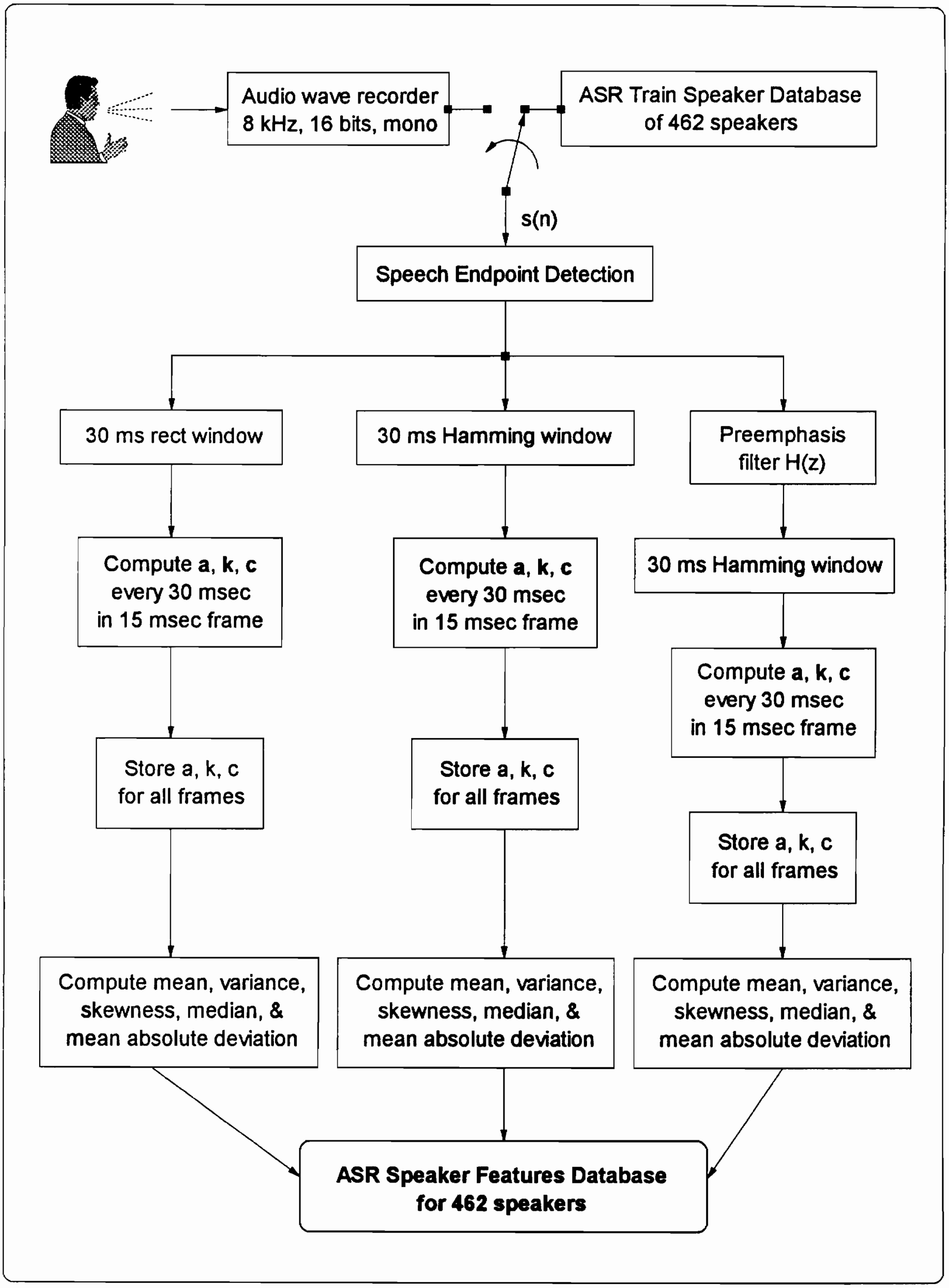


Figure 5.2 ASR Speaker Features Database.

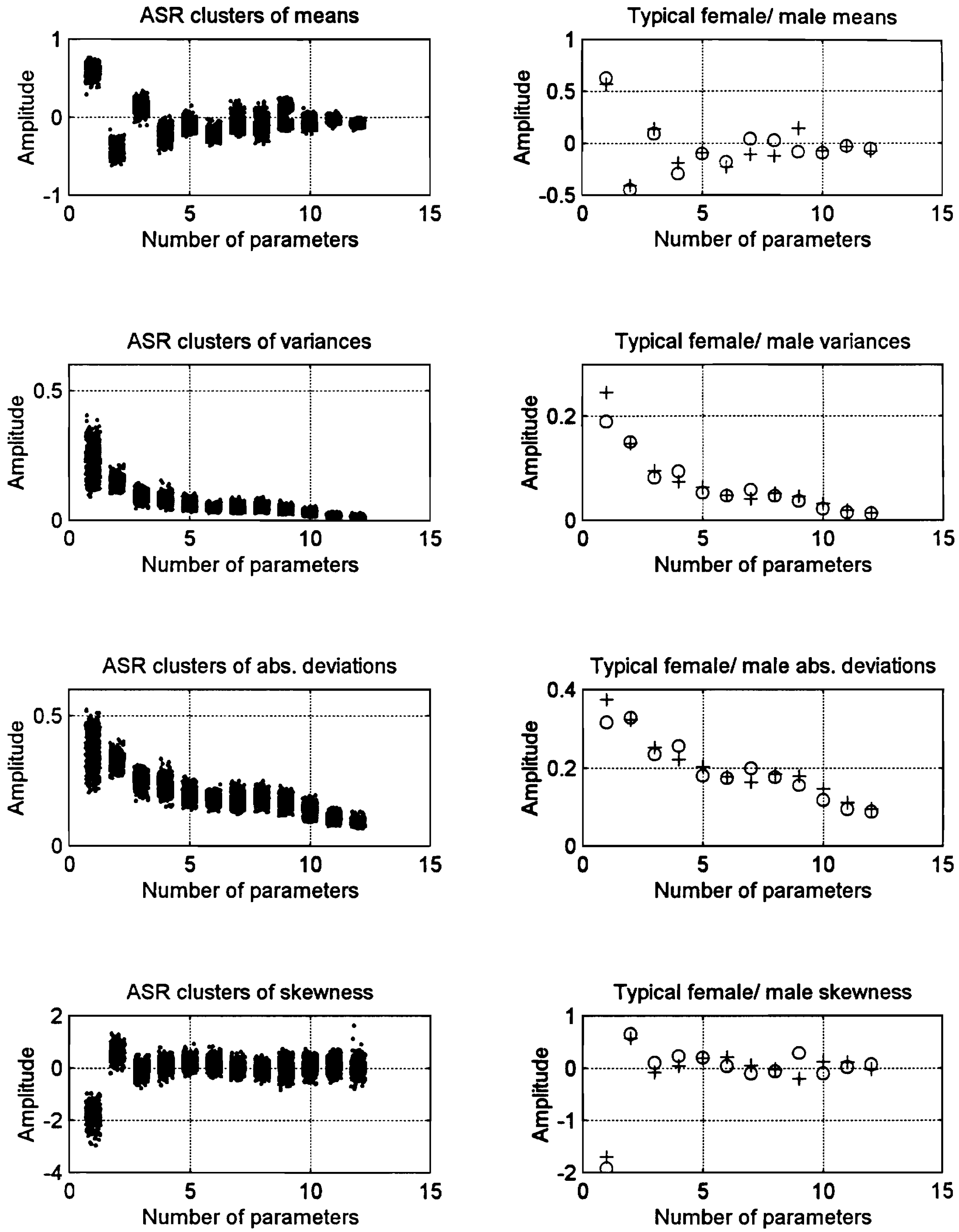


Figure 5.3 : Plots of ASR clusters of moments and typical male/ female moments.

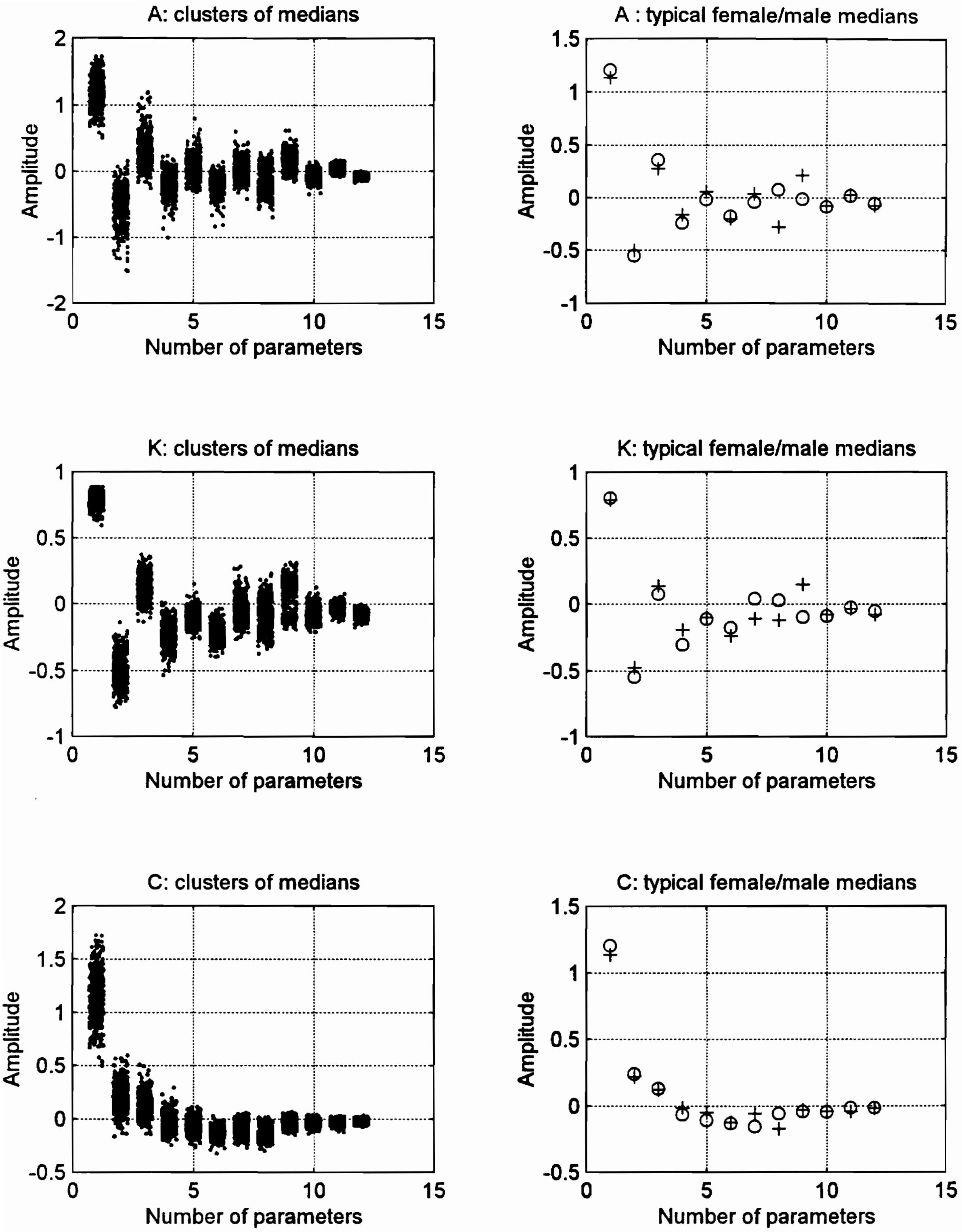


Figure 5.4 : Clusters of medians and typical female/male medians for A, K, and C.

5.2 ASR Design : Spectral Means and Medians

In this section, different spectral distances are proposed. The feature means and feature medians, and spectral distances are employed to design the automatic speaker recognition system. The system performance is evaluated and the effectiveness of each feature is presented. Block diagrams of both the text-independent speaker verification procedure and the text-independent speaker identification procedure are presented. The following criteria are designed to evaluate the speaker recognition performance :

- The performance of the ASR system based on means and variances of features.
- The performance of the ASR system based on medians and variances of features.
- The performance of the ASR system based on the combined features.

Following the development, the effectiveness of each feature set is discussed.

5.2.1 Distances for Spectral Means

In general, spectral distances are used to compute the overall dissimilarity between two feature spectral patterns. The goal is to have a large distance for two perceptually different sounds, and a small distance for two perceptually similar sounds. In this ASR system the following spectral distances are proposed to measure the spectral dissimilarities among the speakers :

$$d_1 = \sum_{k=1}^{12} \frac{(\mu_k^i - \overline{x}_k)^2}{(\sigma_k^i)^2} \tag{5.13}$$

$$d_2 = \sum_{k=1}^{12} \frac{(\mu_k^i - \overline{x}_k)^2}{(mad_k^i)^2} \tag{5.14}$$

$$d_3 = \sum_{k=1}^{12} \frac{|\mu_k^i - \overline{x}_k|}{\sigma_k^i} \tag{5.15}$$

$$d_4 = \sum_{k=1}^{12} \frac{|\mu_k^i - \overline{x}_k|}{mad_k^i} \tag{5.16}$$

where μ_k^i = train kth parameter mean of speaker *i*,

$(\sigma_k^i)^2$ = train kth parameter variance of speaker *i*,

mad_k^i = train kth parametere mean absolute deviation of speaker *i*,

$\overline{x}_k$ = test kth parameter speaker mean.

As mentioned in Chapter 3, the direct Euclidean measurements on the two signals s(n) and s'(n) are not appropriate due to variations in speaking rate, unknown time alignment of the two signals, and variable degrees of loudness. If *S*(*w*) represents the power spectrum of s(n), then the log of *S*(*w*) can be expressed in terms of cepstral coefficients below :

$$\log S(w) = \sum_{m=-\infty}^{\infty} c_m e^{-jmw} \tag{5.17}$$

Using Parseval's theorem, the rms log spectral distance between *S*(*w*) and *S*'(*w*) can be expressed as :

$$d_2^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|\log S(w) - \log S'(w)\right|^2 dw = \sum_{m=-\infty}^{\infty} (c_m - c'_m)^2 \tag{5.18}$$

where c_m and c'_m are the cepstral coefficients of *S*(*w*) and *S*'(*w*) respectively [86].

Since the cepstrum is a decaying sequence, the truncated cepstral distance is adequate and is defined below :

$$d_c^2 = \sum_{m=1}^{12} (c_m - c'_m)^2 \tag{5.19}$$

The proof for Eqn. (5.18) is presented below [86] :

Proof :

$$d_2^2(S,S') = \int_{-\pi}^{\pi} \left|\log S(w) - \log S'(w)\right|^2 \frac{dw}{2\pi}$$

$$= \int_{-\pi}^{\pi} \left| \sum_{n=-\infty}^{\infty} c_n e^{-jwn} - \sum_{n=-\infty}^{\infty} c'_n e^{-jwn} \right|^2 \frac{dw}{2\pi}$$

$$= \int_{-\pi}^{\pi} \left| \sum_{n=-\infty}^{\infty} (c_n - c'_n) e^{-jwn} \right|^2 \frac{dw}{2\pi}$$

$$= \int_{-\pi}^{\pi} \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} (c_n - c'_n)(c^*_m - c'^*_m) e^{-jw(n-m)} \frac{dw}{2\pi}$$

$$= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} (c_n - c'_n)(c^*_m - c'^*_m) \int_{-\pi}^{\pi} e^{-jw(n-m)} \frac{dw}{2\pi}$$

$$= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} (c_n - c'_n)(c^*_m - c'^*_m) \delta(n-m)$$

$$= \sum_{n=-\infty}^{\infty} \left| c_n - c'_n \right|^2 = \sum_{n=-\infty}^{\infty} (c_n - c'_n)^2$$

where $\delta(n-m) = \begin{cases} 1 & , \quad n = m \\ 0 & , \quad n \neq m \end{cases}$

* denotes complex conjugate notation.

The spectral distances between speakers are the sum of the squares of the difference between each corresponding parameters. Therefore, this distance uses equal weighting for all parameters and does not account for the fact that the variation of each parameter of the same speaker before the calculation of moments of the distribution varies greatly. Figure 5.5 presents plots of typical female/ male on train/ test PARCOR **K** clusters and train/ test means of the **K** clusters. Since the train means and the test means of this female voice are similar, it implies that means of spectral features can be used to distinguish among speakers. The same observation can be made on the male case. Another important fact in Figure 5.5 is that the variation of each parameter is quite different. The large variation of a parameter implies that it is not necessary to have close values in both train and test features of the same speaker. If equal weights are used for each parameter, then the large variation of that parameter may point to an incorrect speaker. The need to

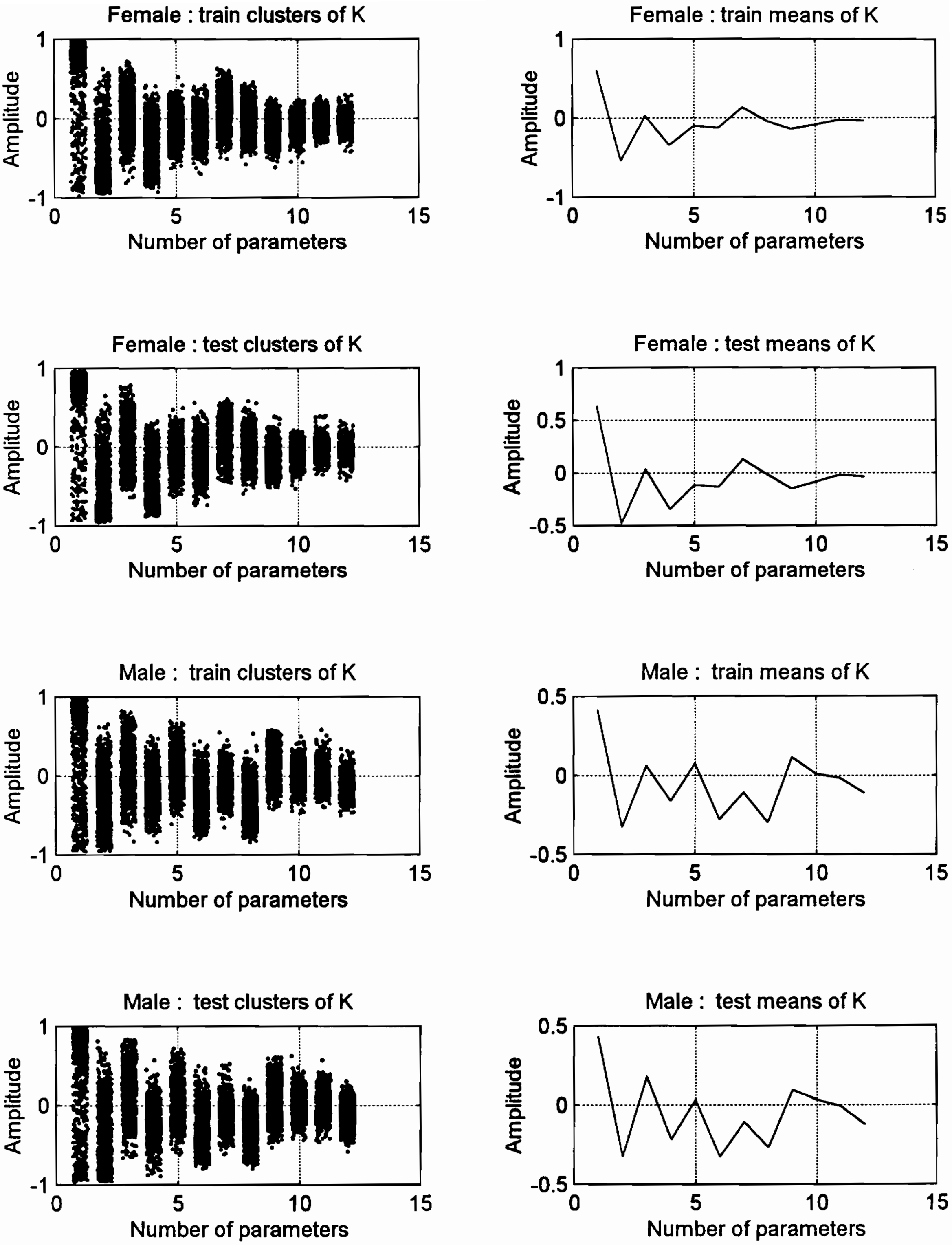


Figure 5.5 : Train/ test clusters and means of K for a female speaker and a male speaker.

include the variation of each parameter into the spectral distances is essential. From the spectral distance in Eqns. (5.13 - 5.16), the contribution of each parameter distance to the overall spectral distance is inversely proportional to its variations. Parameters of small variation contribute more weight to the overall distance measure than parameters of large variation.

5.2.2 ASR Design : Use of Spectral Means

In Figure 5.5, the means of the clusters of the PARCOR parameters in the train (reference) speaker features database for a male voice and a female voice are plotted together with the means of their test features. The male voice patterns generated from the test database match well with the voice patterns of the same male in the reference (train) speaker features database. Female voice patterns in both reference and test sets are also similar. Therefore, the means of spectral clusters provide a good tool for recognizing speakers. The above four spectral distances are used to find out how good means of spectral features for speakers do in recognizing speakers. As presented in Chapter 1, speaker recognition includes speaker identification and speaker verification. The block diagram of the ASR speaker identification procedure is presented in Figure 5.6. The ASR system will read an unknown utterance either from an audio wave recorder or from the ASR test speaker database. The speech signal s(n) is passed thru the speech endpoint detection, a 30 msec Hamming window frame with 50% overlapping. The system then computes the speaker features **a**, **k**, and **c** for each frame and stores in **A**, **K**, and **C**. Next, the system computes means, variances, mean absolute deviations, and skewnesses of **A**, **K**, and **C**. The spectral distances of these moments are computed against all 462 speakers. The system selects a speaker in the reference (train) database with the smallest distance, and labels the unknown utterance as the utterance of that speaker. Since the texts of the train utterances are different from the texts of the test utterances in the TIMIT database, the identification procedure in Figure 5.6 is known as *text-independent* speaker identification which is a superset of text-dependent speaker identification. Since it is impossible to time-align speaker features in the text-independent case, the overall means and the overall variances of the speaker features are used to distinguish among speakers.

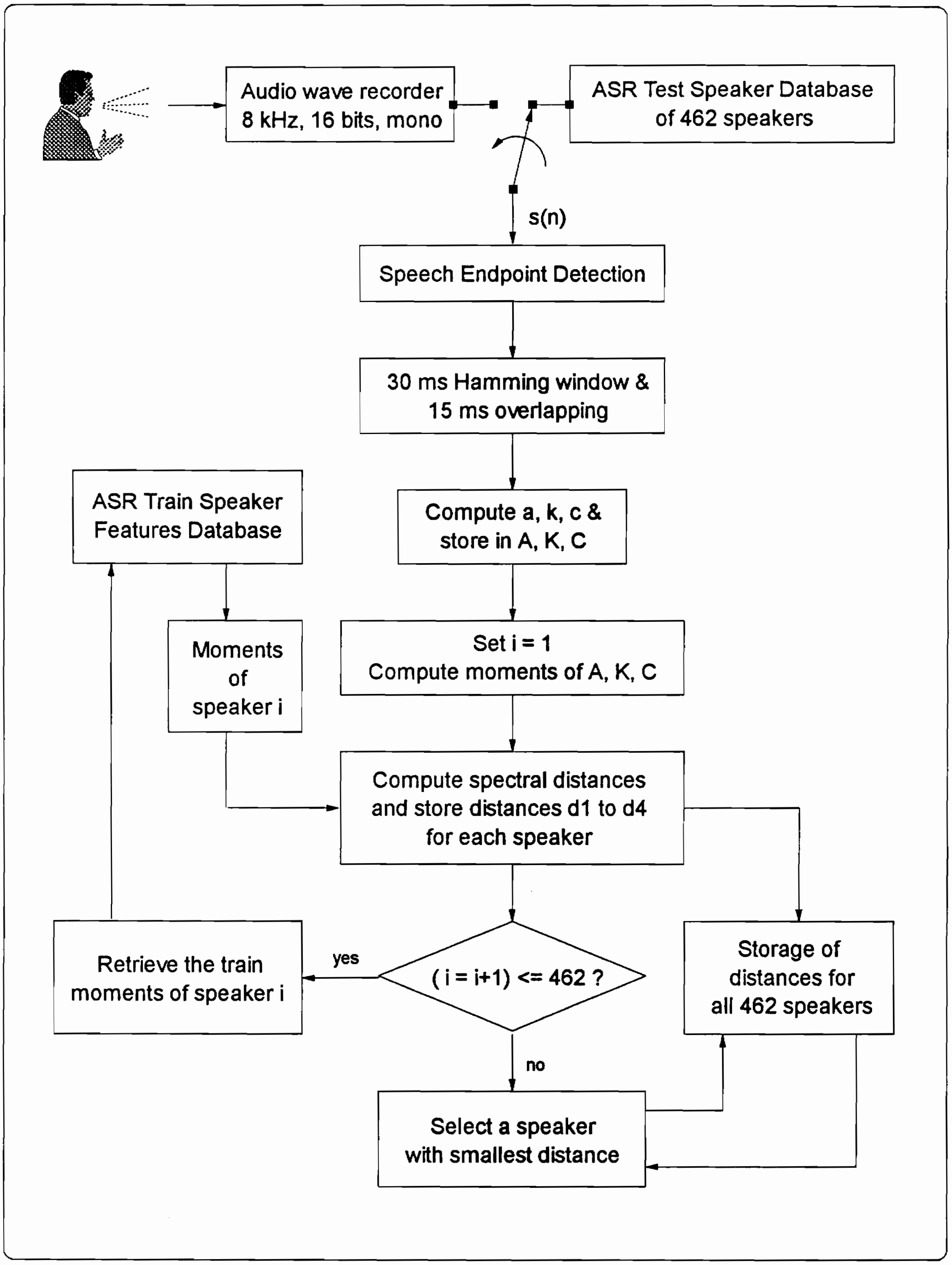


Figure 5.6 ASR speaker identification procedure.

These overall means can be considered as the model of the vocal tract of the speaker. As mentioned in Chapter 1, speaker differences that enable speaker recognition include interspeaker and intraspeaker variations. Interspeaker variations are due to the physical aspect of differences in vocal cords and vocal tract shape, and to the behavioral aspect of differences in speaking styles among speaker. Therefore, the ASR system at hand utilizes the concept of physical aspect to recognize speakers. It is very difficult for a professional mimic to fool a recognition system that bases its decisions on the physical aspect of the speakers. A system based on behavioral aspect of the speakers, however, can be fooled by professional mimics.

Speaker verification is used to verify if the unknown utterance is from the claimed speaker. The block diagram of the ASR speaker verification procedure is illustrated in Figure 5.7. The procedure is similar to that of speaker identification. In short, the speaker features are extracted from the unknown utterance and are compared against the reference features of the claimed speaker. If the difference is less than the threshold of the claimed speaker, the system accepts the speaker. Otherwise, it rejects the speaker. In general, it is preferred to set the threshold tightly because accepting a wrong person may have more adverse affects.

Since the system attempts to recognize speakers based on the means of their spectral feature parameters, it is more informative to see the frequency spectra of the means of speaker features. Plots of reference feature means versus test feature means for the same female, plots of reference feature spectrum versus test feature spectra for the same female are illustrated in Figure 5.8. Similar sets of plots for the same male is also provide in Figure 5.8. Now, the relation between the spectral means and their frequency spectra can be observed in Figure 5.8. It is very interesting to note that the spectra of male speakers have more spectral peaks than the spectra of female speakers. The amplitude and frequency locations of the peaks are different for this female speaker and the male speaker.

The resulting recognition errors using speaker feature means, variances based on the four spectral distances are summaried in Tables 5.1, 5.2, and 5.3. In each table, the speaker recognition errors are divided into identification errors and verification errors. Both identification errors and verification errors are then categoried into female/ male identification errors and female/ male verification errors. The speaker recognition errors

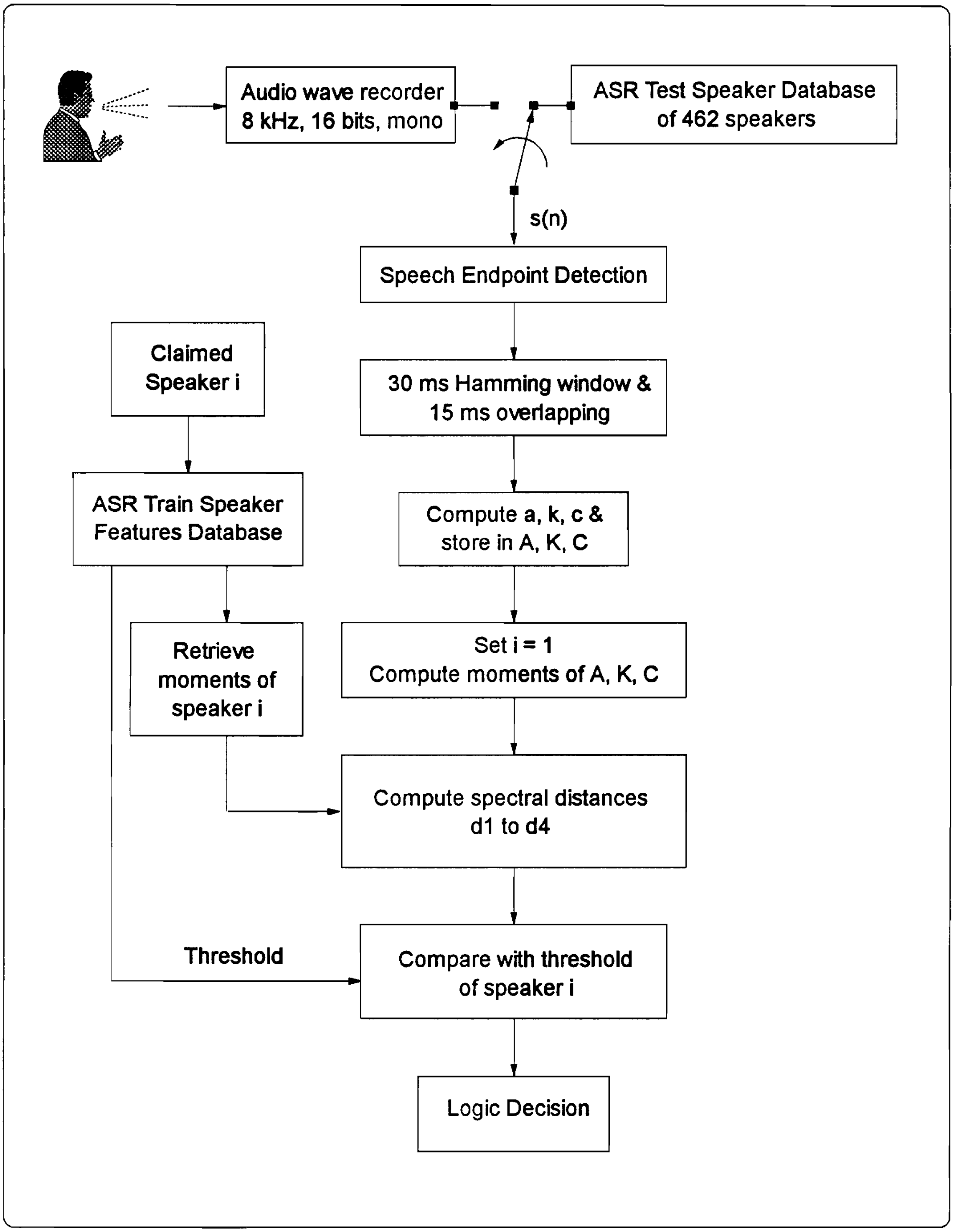


Figure 5.7 ASR speaker verification procedure.

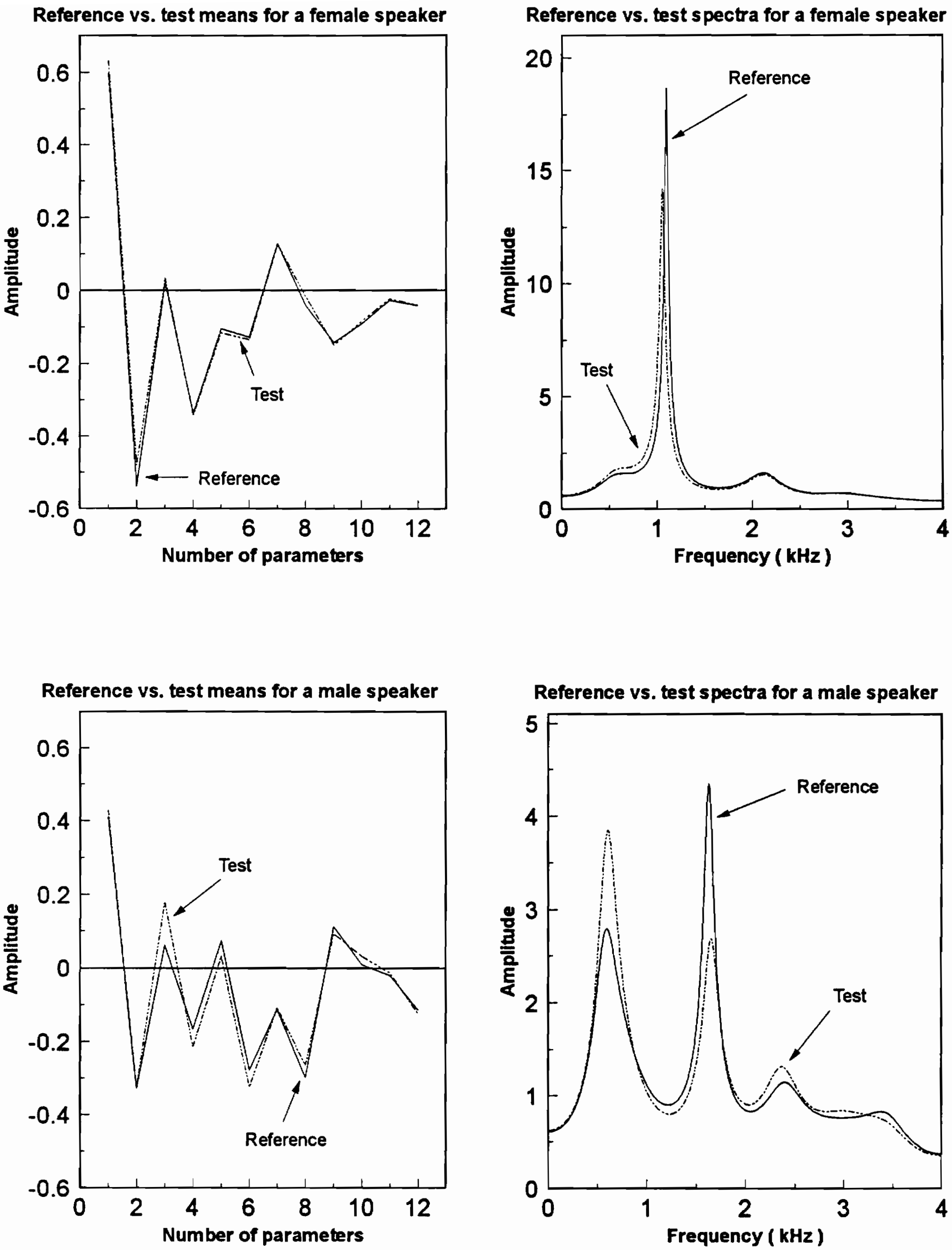


Figure 5.8 Plots of reference / test means and spectra for a female and a male speaker.

based on the speaker features using a rectangular window are presented in Table 5.1. Table 5.2 provides the speaker recognition (SR) errors based on speaker features using a Hamming window. Table 5.3 provides the SR errors based on the speaker features using both a first order preemphasis filter and a Hamming window. Before discussing the errors in these tables, it is essential to know that there are 136 female speakers and 326 male speakers in the database. Each table provides SR errors using the four spectral distances on LPC features **A**, LPC PARCOR features **K**, and LPC cepstral features **C**. In general, LPC PARCOR features **K** outperform LPC **A**, and LPC cepstral **C** in both identification and verification cases. The spectral distances d_1 and d_2 outperform the other two spectral distances d_4 and d_5. LPC features **A** have the worst results of approximately 50% for both identification and verification. The highlight values indicate the "best" performance values in each tables. In general, the "best" female speaker identification and verification errors are 30.1% and 29.4% respectively. The "best" male speaker identification and verification errors are 31.9% and 32.5% respectively.

Table 5.1 : Speaker recognition errors using rectangular window.

		Identification errors		**Verification errors**	
		Female	Male	Female	Male
A	d_1	71 (52.2%)	156 (47.9%)	68 (50.0%)	162 (49.7%)
	d_2	75 (55.1%)	154 (47.2%)	69 (50.7%)	162 (49.7%)
	d_3	74 (54.4%)	167 (51.2%)	76 (55.9%)	165 (50.6%)
	d_4	73 (53.7%)	164 (50.3%)	75 (55.1%)	165 (50.6%)
K	d_1	**42 (30.9%)**	**104 (31.9%)**	**43 (31.6%)**	**107 (32.8%)**
	d_2	**41 (30.1%)**	**104 (31.9%)**	**41 (30.1%)**	**106 (32.5%)**
	d_3	51 (37.5%)	123 (37.7%)	47 (34.6%)	113 (34.7%)
	d_4	53 (39.0%)	122 (37.4%)	47 (34.6%)	115 (35.3%)
C	d_1	56 (41.2%)	147 (45.1%)	59 (43.4%)	139 (42.6%)
	d_2	55 (40.4%)	147 (45.1%)	57 (41.9%)	140 (42.9%)
	d_3	61 (44.9%)	164 (50.3%)	58 (42.6%)	157 (48.1%)
	d_4	61 (44.9%)	168 (51.5%)	58 (42.6%)	157 (48.1%)

Table 5.2 : Speaker recognition using Hamming window.

		Identification error		**Verification error**	
		Female	Male	Female	Male
A	d_1	84 (61.7%)	191 (58.6%)	81 (59.6%)	191 (58.6%)
	d_2	83 (61.0%)	193 (59.2%)	82 (60.3%)	191 (58.6%)
	d_3	87 (64.0%)	192 (58.9%)	82 (60.3%)	186 (57.1%)
	d_4	87 (64.0%)	192 (58.9%)	84 (61.8%)	188 (57.7%)
K	d_1	**42 (30.9%)**	**114 (35.0%)**	**41 (30.1%)**	**117 (35.9%)**
	d_2	**42 (30.9%)**	**113 (34.7%)**	**40 (29.4%)**	**119 (36.5%)**
	d_3	51 (37.5%)	123 (37.7%)	41 (30.1%)	126 (38.6%)
	d_4	53 (39.0%)	127 (39.0%)	44 (32.6%)	128 (39.3%)
C	d_1	52 (38.2%)	134 (41.1%)	48 (35.3%)	131 (40.2%)
	d_2	51 (37.5%)	135 (41.4%)	51 (37.5%)	133 (40.8%)
	d_3	55 (40.4%)	141 (43.3%)	53 (39.0%)	153 (46.9%)
	d_4	56 (41.2%)	144 (44.2%)	55 (40.4%)	157 (48.2%)

Table 5.3 : Speaker recognition error using preemp. filter & Hamming window.

		Identification error		**Verification error**	
		Female	Male	Female	Male
A	d_1	65 (47.8%)	170 (52.1%)	64 (47.1%)	184 (56.4%)
	d_2	64 (47.1%)	170 (52.1%)	64 (47.1%)	186 (57.1%)
	d_3	71 (52.2%)	178 (54.6%)	70 (51.5%)	188 (57.7%)
	d_4	72 (52.9%)	182 (55.8%)	70 (51.5%)	187 (57.4%)
K	d_1	**44 (32.4%)**	**104 (31.9%)**	**41 (30.1%)**	**110 (33.7%)**
	d_2	**45 (33.1%)**	**104 (31.9%)**	**42 (30.9%)**	**111 (34.0%)**
	d_3	43 (31.6%)	116 (35.6%)	48 (35.3%)	123 (37.7%)
	d_4	42 (30.9%)	117 (35.9%)	47 (34.6%)	122 (37.4%)
C	d_1	60 (44.1%)	156 (47.9%)	53 (39.0%)	145 (44.5%)
	d_2	59 (43.4%)	157 (48.2%)	55 (40.4%)	143 (43.9%)
	d_3	66 (48.5%)	159 (48.8%)	62 (45.6%)	166 (50.9%)
	d_4	64 (47.1%)	155 (47.5%)	60 (44.1%)	168 (51.5%)

Since the text-independent speaker recognition errors are large, long-term means and variances of the speaker features are not providing enough information to recognize all speakers from their voices even though the means of speaker features in reference and in test are consistent. This implies that there are other speakers with better similar test patterns to the correct speaker test patterns to the correct speaker reference patterns. Two examples of incorrect speaker recognition on two female speakers and two male speaker is illustrated in Figure 5.9. Figure 5.9 provides means plots and spectra plots of reference means of a female speaker versus two test means of the same female and another female. Similar plots for male speakers are also presented in Figure 5.9. In both male and female cases, all means and spectra plots are too close to call. We can see from these figures that the task of recognizing speakers from their voices is not an easy one.

5.2.3 ASR Design : Use of Spectral Medians

From the previous section, it can be seen that only a small variation in the values of the feature means causes the ASR system to select the incorrect speaker. The other alternative is to use the medians of speaker features instead of the means of speaker features. The following spectral distances are used to evaluate the performance of the medians in the speaker recognition system and are defined below :

$$d_5 = \sum_{k=1}^{12} \frac{(xmed_k^i - xmed_k)^2}{(\sigma_k^j)^2} \qquad (5.20)$$

$$d_6 = \sum_{k=1}^{12} \frac{(xmed_k^i - xmed_k)^2}{(mad_k^i)^2} \qquad (5.21)$$

$$d_7 = \sum_{k=1}^{12} \frac{\left|xmed_k^i - xmed_k\right|}{\sigma_k^j} \qquad (5.22)$$

$$d_8 = \sum_{k=1}^{12} \frac{\left|xmed_k^i - xmed_k\right|}{mad_k^i} \qquad (5.23)$$

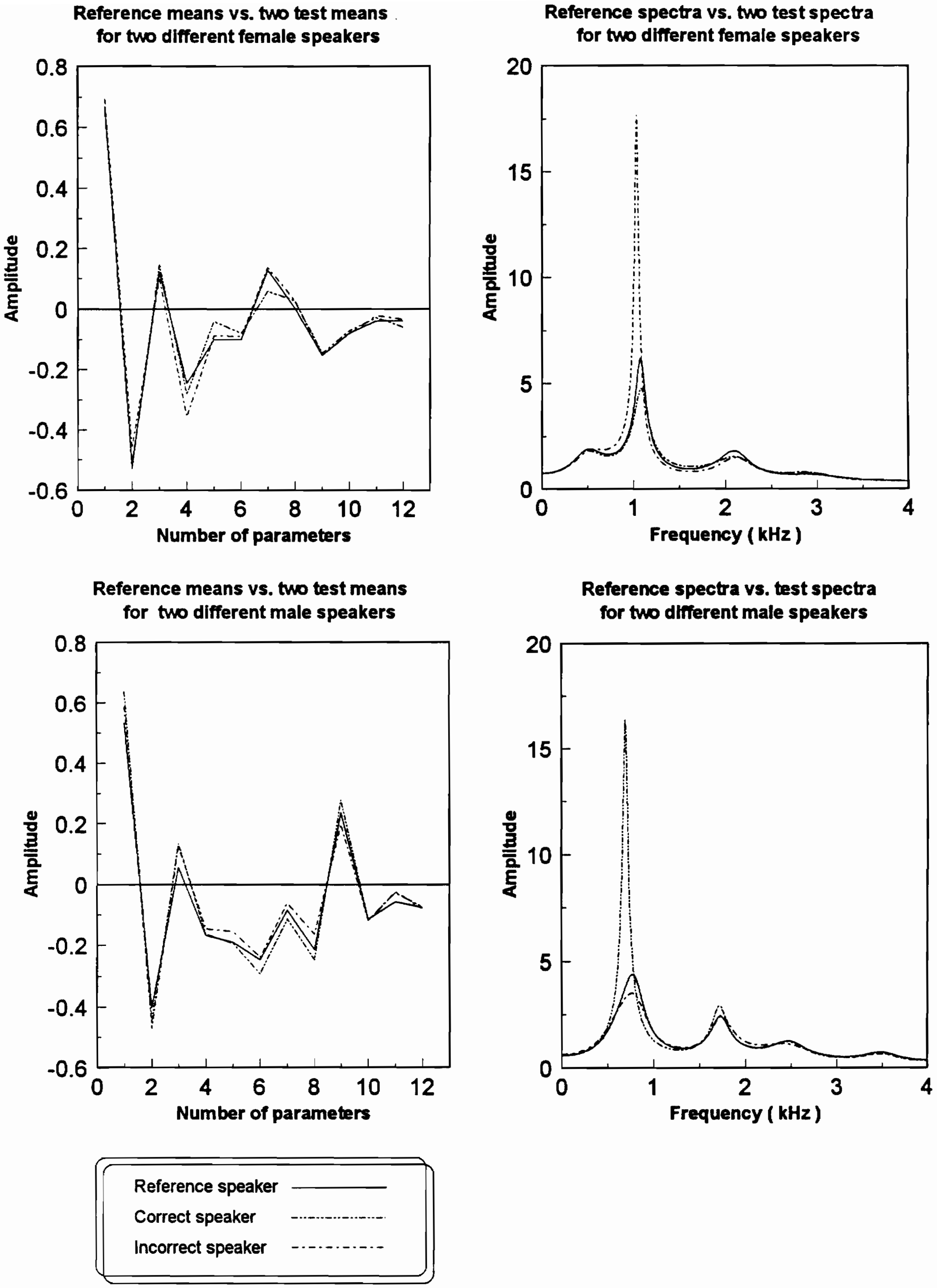


Figure 5.9 Reference vs. two test means and spectra for two female and two male speakers.

where $xmed_k^i$ = train kth parameter median of speaker i,

$(\sigma_k^i)^2$ = train k^{th} parameter variance of speaker *i*,

mad_k^i = train k^{th} parametere mean absolute deviation of speaker *i*,

$xmed_k$ = test k^{th} parameter speaker median.

Since the means of the same male speakers on both the reference database and the test database are quite different as shown in Figure 5.8 of previous section, it is informative to plot the frequency spectra of the same male speaker and female speaker based on the medians of speaker features. Figure 5.10 provides plots of reference feature medians versus test feature medians, and the respected frequency spectra plots for the same female and male speakers. From Figure 5.10, neither the spectrum nor the median plots of a female and a male speaker change much or seem to help male spectral patterns. Therefore, the only way to find out how good the ASR system based on medians of features is to evaluate the train and test speaker database.

The resulting recognitions errors using medians of features and the four spectral distances above are tabulated in Tables 5.4, 5.5, and 5.6. Each table provides the speaker recognition errors for only LPC features **K**, and LPC cepstral features. The LPC features **A** are not included in the tables since its performance is mediocre. The highlight values indicate the "best" row performance values in each table. The spectral distance d_2 outperforms the other three spectral distances. However, the performance decreases when compared to the results in Tables 5.1, 5.2, and 5.3. In general, the "best" female identification and verification errors are 36.0% and 33.8% respectively. The "best" male identification and verification errors are 40.2% and 39.6% respectively. Again, the LPC PARCOR features **K** outperforms the LPC cepstral features **C**.

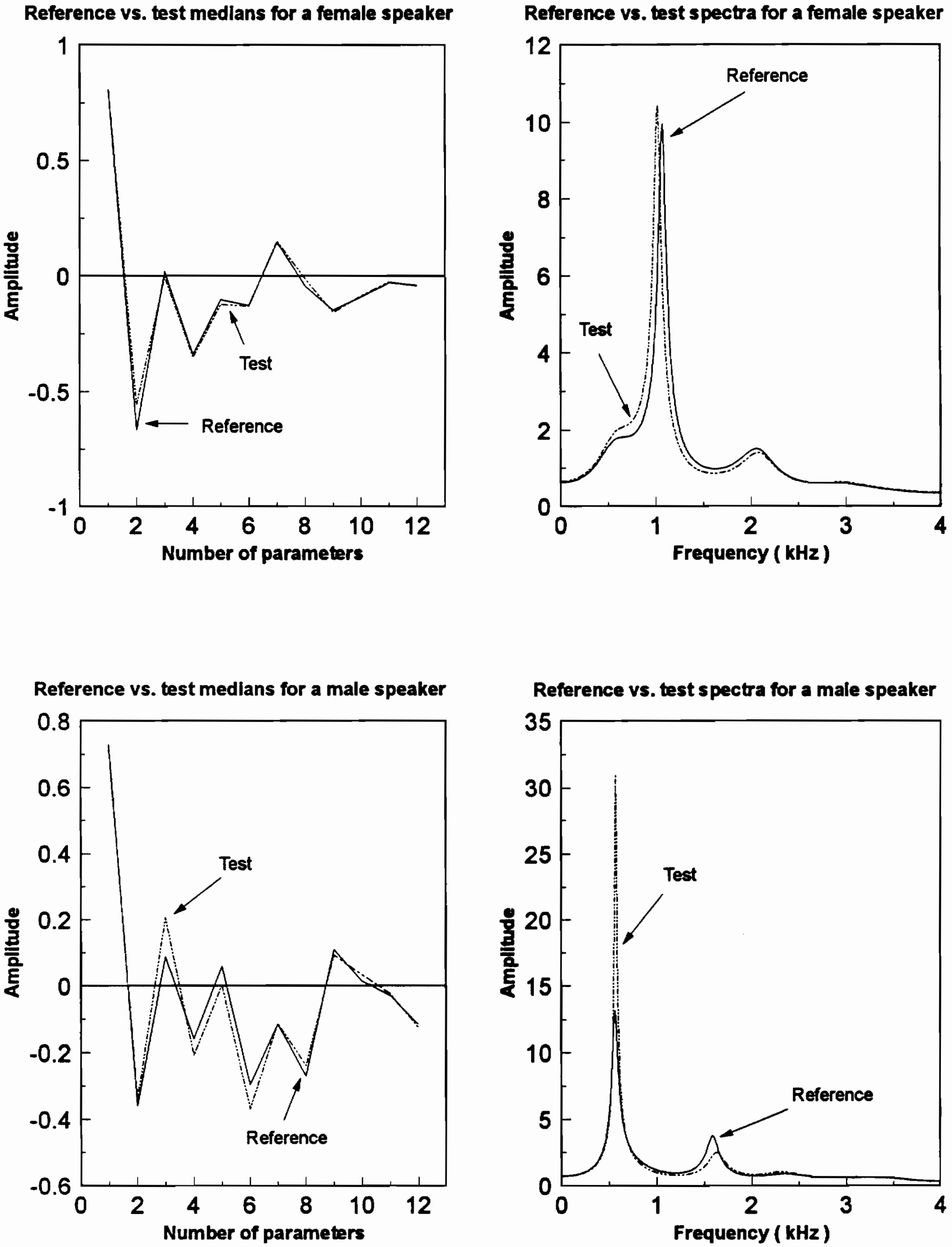


Figure 5.10 Plots of reference / test medians and spectra for a female and a male speaker.

Table 5.4 : Speaker recognition errors using medians & rectangular window.

		Identification error		Verification error	
		Female	Male	Female	Male
K	d_1	53 (39.0%)	139 (42.6%)	50 (36.8%)	135 (41.4%)
	d_2	**53 (39.0%)**	**135 (41.4%)**	**46 (33.8%)**	**129 (39.6%)**
	d_3	52 (38.2%)	142 (43.6%)	52 (38.2%)	146 (44.8%)
	d_4	49 (36.0%)	140 (42.9%)	51 (37.5%)	142 (43.6%)
C	d_1	63 (46.3%)	160 (49.1%)	62 (45.6%)	173 (53.1%)
	d_2	63 (46.3%)	161 (49.4%)	61 (44.9%)	174 (53.4%)
	d_3	68 (50.0%)	183 (56.1%)	65 (47.8%)	182 (55.8%)
	d_4	68 (50.0%)	182 (55.8%)	66 (48.5%)	182 (55.8%)

Table 5.5 : Speaker recognition errors using medians & Hamming window.

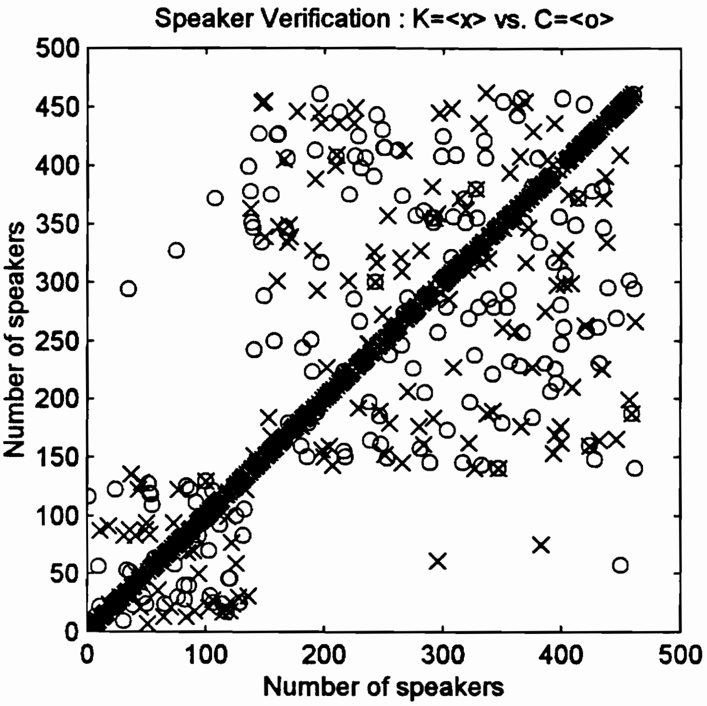
		Identification error		Verification error	
		Female	Male	Female	Male
K	d_1	62 (45.6%)	135 (41.4%)	51 (37.5%)	147 (45.1%)
	d_2	**61 (44.9%)**	**131 (40.2%)**	**50 (36.8%)**	**145 (44.5%)**
	d_3	60 (44.1%)	142 (43.6%)	53 (39.0%)	147 (45.1%)
	d_4	60 (44.1%)	141 (43.3%)	51 (37.5%)	144 (44.2%)
C	d_1	60 (44.1%)	158 (48.5%)	56 (41.2%)	167 (51.2%)
	d_2	61 (44.9%)	156 (47.9%)	55 (40.4%)	164 (50.3%)
	d_3	63 (46.3%)	160 (49.1%)	62 (45.6%)	170 (52.1%)
	d_4	60 (44.1%)	160 (49.1%)	61 (44.9%)	168 (51.5%)

Table 5.6 : ASR errors using preemphasis filter & Hamming window.

		Identification error		Verification error	
		Female	Male	Female	Male
K	d_1	58 (42.6%)	131 (40.2%)	56 (41.2%)	142 (43.6%)
	d_2	**57 (41.9%)**	**132 (40.5%)**	**55 (40.4%)**	**139 (42.6%)**
	d_3	56 (41.2%)	133 (40.8%)	59 (43.4%)	151 (46.3%)
	d_4	55 (40.4%)	133 (40.8%)	58 (42.6%)	149 (45.7%)
C	d_1	63 (46.3%)	167 (51.2%)	59 (43.4%)	169 (51.8%)
	d_2	61 (44.9%)	167 (51.2%)	59 (43.4%)	170 (52.1%)
	d_3	69 (50.7%)	175 (53.7%)	64 (47.1%)	184 (56.4%)
	d_4	68 (50.0%)	174 (53.4%)	64 (47.1%)	183 (56.1%)

5.3 ASR Design : Means and Medians of K and C

In previous sections the errors of the speaker recognition system based on **a**, **k**, and **c** alone are large. Both **k** and **c** features have better performance than the **a** features. If both **k** and **c** select same incorrect speakers most of the time, then the combination of **k** and **c** features will not improve the recognition performance of the ASR system. Otherwise, the combination of **k** and **c** features may provide better recognition performance. The speaker verification based on **k** features versus the speaker verification based on **c** features is illustrated in Figure 5.11 where 'x' and 'o' belong to **k** and **c** respectively. The x-axis corresponds to the correct speaker which the system supposes to select. The y-axis corresponds to the speaker which the system selects. If all 'x' and 'o' lie exactly along the diagonal, then perfect speaker recognition is achieved. If 'x' and 'o' lie outside the diagonal line, the incorrect speakers are selected. If all 'x' and 'o' outside the diagonal line lie on top of each other, then the combination of features **c** and **k** does not provide "much better" speaker recognition than **c** or **k** alone. Since most of the 'x' and 'o' off the diagnonal line are separate most of the time, this event indicates that the correlation of 'x' and 'o' off the diagonal line is low, and that the combination of the **k** and **c** features



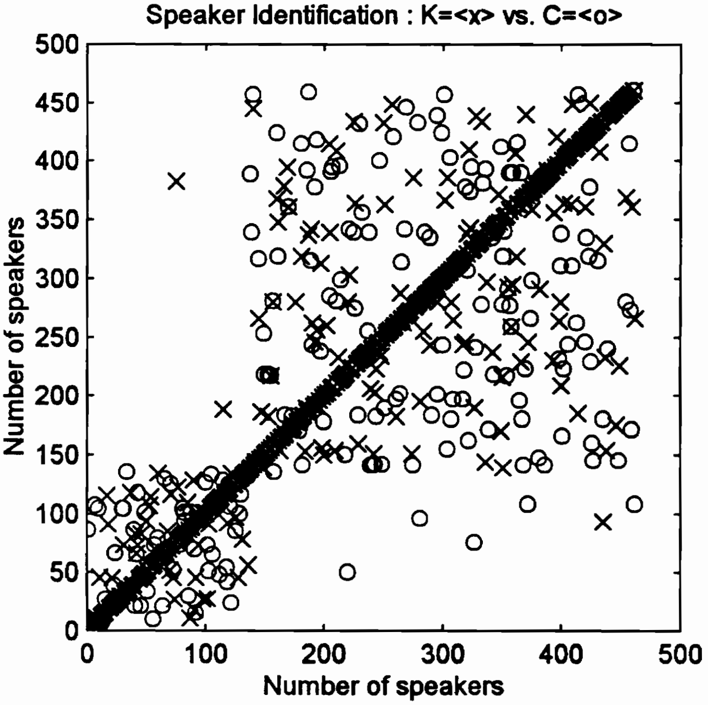


Figure 5.11 : Speaker verification and identification based on LPC **k** and LPC **c** alone.

may provide better speaker recognition than **k** features or **c** features alone. Next, it is necessary to verify if **c** and **k** features can be utilized to improved the speaker recognition performance. The plots of reference speaker means, the test means of the correct speaker, and the test means of the system-selected speaker for both **k** and **c** features are illustrated in Figure 5.12. For the female speakers, the test means of **k** for the selected speaker are closer to the reference means of **k** than the test means of **k** for the correct speaker. However, the test means of **c** for the correct speaker are closer to the reference means of **c** than the test means of **c** for the selected speaker. Similar situations are observed for the male speakers from Figure 5.12. Therefore, **c** and **k** features can be combined to improve the speaker recognition performance. The next logical step is to build an ASR system based on both **k** and **c** features, and to evaluate the system performance.

5.3.1 ASR Design : Use of Means of k and c

Since the spectral distance d_1 and the spectral distance d_2 outperform the other two spectral distances, only d_1 and d_2 distances are considered in this section. Again, the spectral features **a** will not be employed in this section due to its poor performance. The ASR speaker verification and speaker identification procedures of the new system are similar to the procedures in Figures 5.6 and 5.7 except that the system will add the spectral distance computed from features **k** to spectral distance computed from features **c**, and stores the resulting distance. The system then selects the speaker with smallest distance for the speaker identification. For speaker verification, the system accepts the speaker if the resulting distance is less than the reference threshold, or rejects the speaker if the resulting distance is larger than the reference threshold.

The resulting speaker recognition errors using speaker means of a combination of **k** and **c** features based on the two spectral distances d_1 and d_2 are tabulated in Tables 5.7, 5.8, and 5.9. The resulting spectral distance is computed according to the following equation :

$$D = d_i(\mathbf{k}) + w d_i(\mathbf{c}) \tag{5.24}$$

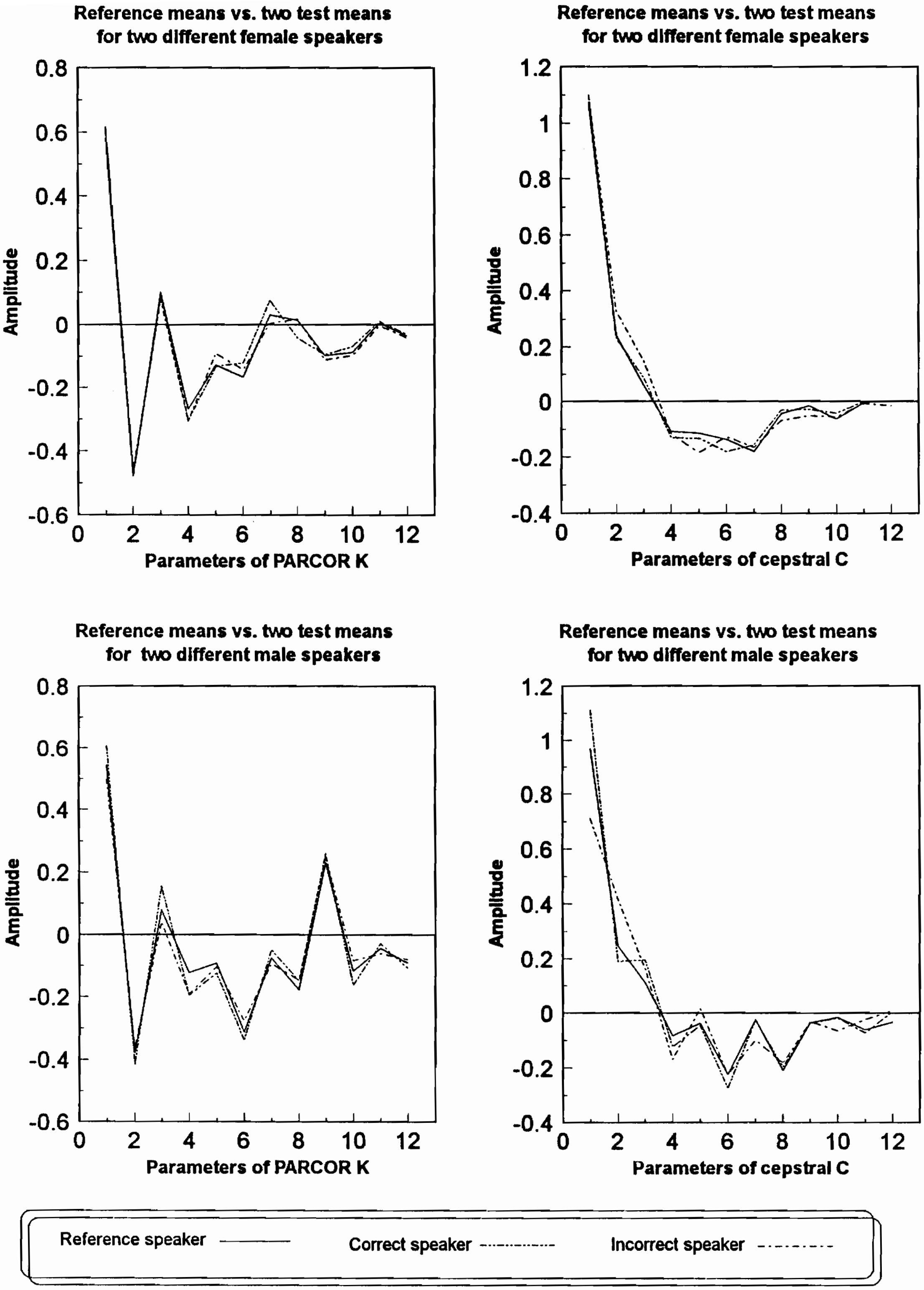


Figure 5.12 Reference vs. two test means of K & C for two female and two male speakers.

where i = 1, 2.

w = a scaling constant.

d_i (*) = the spectral distance 1 or 2 of either **k**, or **c**.

For each table, the "best" performance in each of the four categories are highlighted. The "best" identification errors in Table 5.7 correspond to the use of $w = 0.8$, while the "best" verification errors in the same table correspond to the use of $w = 0.9$. Similar observations are made in Tables 5.8 and 5.9. It is interesting to note that different scaling constants can be used to improve the recognition performance of the system. In general, it makes sense to use a higher constant value w for speaker verification than for speaker identification.

Now, the "best" female speaker identification and verification errors for the new system are **11.8% and 14.0%** respectively compared to **30.1% and 29.4%** in previous section. The "best" male speaker identification and verification errors are **19.3% and 20.9%** respectively compared to **31.9% and 32.5 %** in previous system. However, the errors is still too high. Another observation is that the spectral distance d_2 slightly outperforms the spectral distance d_1.

Table 5.7 : SR errors using means of **k**, **c** & rectangular window.

		Identification errors		**Verification errors**	
		Female	Male	Female	Male
w=1.0	d_1	23 (16.9%)	72 (22.1%)	20 (14.7%)	70 (21.5%)
	d_2	22 (16.2%)	71 (21.8%)	21 (15.4%)	68 (20.9%)
w=0.9	d_1	21 (15.4%)	72 (22.1%)	**19 (14.0%)**	**69 (21.2%)**
	d_2	20 (14.7%)	71 (21.8%)	20 (14.7%)	69 (21.2%)
w=0.8	d_1	21 (15.4%)	70 (21.5%)	20 (14.7%)	68 (20.9%)
	d_2	**20 (14.7%)**	**70 (21.5%)**	20 (14.7%)	70 (21.5%)

Table 5.8 : SR errors using means of k, c & Hamming window.

		Identification errors		**Verification errors**	
		Female	Male	Female	Male
w=1.0	d_1	16 (11.8%)	71 (21.8%)	21 (15.4%)	73 (22.4%)
	d_2	**16 (11.8%)**	**69 (21.2%)**	21 (15.4%)	73 (22.4%)
w=0.9	d_1	18 (13.2%)	71 (21.8%)	**20 (14.7%)**	**71 (21.8%)**
	d_2	19 (14.0%)	69 (21.2%)	20 (14.7%)	72 (22.1%)
w=0.8	d_1	19 (14.0%)	71 (21.8%)	20 (14.7%)	72 (22.1%)
	d_2	18 (13.2%)	71 (21.8%)	20 (14.7%)	71 (21.8%)

Table 5.9 : SR errors using means of k, c, preemphasis filter & Hamming window.

		Identification errors		**Verification errors**	
		Female	Male	Female	Male
w=1.0	d_1	24 (17.6%)	66 (20.2%)	23 (16.9%)	72 (22.1%)
	d_2	23 (16.9%)	66 (20.2%)	23 (16.9%)	68 (20.1%)
w=0.9	d_1	22 (16.2%)	66 (20.2%)	23 (16.9%)	71 (21.8%)
	d_2	22 (16.2%)	65 (19.9%)	23 (16.9%)	68 (20.1%)
w=0.7	d_1	21 (15.4%)	65 (19.9%)	**19 (14.0%)**	**70 (21.5%)**
	d_2	**21 (15.4%)**	**63 (19.3%)**	21 (15.4%)	71 (21.8%)

5.3.2 ASR Design : Use of Medians of k and c

Similar procedures are applied to the system that makes use of medians of **k** and **c** instead of means of **k** and **c**. For each table, the "best" performance in each of the four categories are highlighted. The "best" identification errors in Table 5.10 correspond to the use of w = 1.0, while the "best" verification errors in the same table correspond to the use of w = 0.7. Similar observations are made in Tables 5.11 and 5.12. Now, the "best" female speaker identification and verification errors for the new system are **22% and 18.4%** respectively compared to **11.8% and 14.0%** in previous section. The "best" male speaker identification and verification errors are **22.1% and 26.7%** respectively compared to **19.3% and 20.9%** in previous system. Again, the spectral distance d_2 slightly outperforms the spectral distance d_1. In general, the overall system performance based on spectral medians is lagged behind the spectral means counterpart. Even the "best" system performance in this chapter is still high. In the next chapter, different methods are used to analyze the speech features and to improve both the speed and the performance of the speaker recognition system.

Table 5.10 : SR errors using medians of **k**, **c** & rectangular window.

		Identification errors		**Verification errors**	
		Female	Male	Female	Male
w=1.0	d_1	32 (23.5%)	86 (26.4%)	33 (24.3%)	99 (30.4%)
	d_2	**32 (23.5%)**	**83 (25.5%)**	30 (22.1%)	100 (30.7%)
w=0.9	d_1	33 (24.3%)	85 (26.1%)	32 (23.5%)	100 (30.7%)
	d_2	32 (23.5%)	85 (26.1%)	32 (23.5%)	100 (30.7%)
w=0.7	d_1	33 (24.3%)	87 (26.7%)	32 (23.5%)	100 (30.7%)
	d_2	31 (22.8%)	86 (26.4%)	**29 (21.3%)**	**99 (30.4%)**

Table 5.11 : SR errors using medians of **k**, **c** & Hamming window.

		Identification errors		Verification errors	
		Female	Male	Female	Male
w=1.0	d_1	32 (23.5%)	77 (23.6%)	26 (19.1%)	94 (28.8%)
	d_2	30 (22.0%)	79 (24.2%)	**25 (18.4%)**	**95 (29.1%)**
w=0.9	d_1	33 (24.3%)	78 (23.9%)	27 (19.8%)	94 (28.8%)
	d_2	31 (22.8%)	79 (24.2%)	26 (19.1%)	95 (20.1%)
w=0.8	d_1	33 (24.3%)	80 (24.5%)	27 (19.8%)	93 (28.5%)
	d_2	**30 (22%)**	**77 (23.6%)**	27 (19.8%)	95 (29.1%)

Table 5.12 : SR errors using medians of **k**, **c**, preemp filter & Hamming window.

		Identification errors		Verification errors	
		Female	Male	Female	Male
w=1.0	d_1	30 (22.1%)	76 (23.3%)	29 (21.3%)	90 (27.6%)
	d_2	**32 (23.5%)**	**72 (22.1%)**	29 (21.3%)	89 (27.3%)
w=0.9	d_1	32 (23.5%)	76 (23.3%)	29 (21.3%)	90 (27.6%)
	d_2	33 (24.3%)	76 (23.3%)	29 (21.3%)	88 (27.0%)
w=0.8	d_1	31 (22.8%)	79 (24.2%)	29 (21.3%)	92 (28.2%)
	d_2	32 (23.5%)	79 (24.2%)	**29 (21.3%)**	**87 (26.7%)**

Chapter 6

Robust ASR System

It is desirable to build a robust automatic speaker recognition (ASR) system with a high level of accuracy. The robust speaker recognition algorithm presented in this chapter will reduce the identification processing time in half due to the ability of the female/ male-type voice separation. The following techniques are used to investigate and to design the final male-type voice / female-type voice algorithm :

- Female / male overall means and variances approach.
- Female / male parameter separation approach.
- Female / male separation approach using the principal components analysis (PCA) technique.

The final robust ASR system design is also presented in this chapter. The following topics are described in the final robust ASR system :

- ASR reference speaker feature database.
- Spectral distances to measure the spectral distortion between two feature patterns.
- Speaker recognition procedure.
- Final robust ASR design.
- Robust identification algorithm and verification algorithm.
- Final system performance and evaluation.

6.1 Classification of Male and Female Types

The human auditory system can reliably recognize male voices from female voices. If the ASR system can somehow reliably perform a similar task, then the ASR system need only identify a speaker as either male or female. Following this division, the processing time for the ASR system will be reduced by about half. The first task is therefore to find which features separate most male voices from female voices, and which algorithm to use.

6.1.1 Female/ Male Overall Means and Variances Approach

The most natural approach is to compute the overall means and variances of female voices in the train speaker features database and the overall means and variances of male voices on the same database. These overall means and variances are defined below :

$$\mu_m = \frac{1}{N_m}\sum_{i=1}^{N_m}\mu_i^m \tag{6.1}$$

$$\mu_f = \frac{1}{N_f}\sum_{i=1}^{N_f}\mu_i^f \tag{6.2}$$

$$\sigma_m^2 = \frac{1}{N_m}\sum_{i=1}^{N_m}(\sigma_i^m)^2 \tag{6.3}$$

$$\sigma_f^2 = \frac{1}{N_f}\sum_{i=1}^{N_f}(\sigma_i^f)^2 \tag{6.4}$$

where μ_m : overall means of male voices,

μ_f : overall means of female voices,

σ_m^2 : overall variances of male voices,

σ_f^2 : overall variances of female voices,

$\mu_i^m, (\sigma_i^m)^2$: means and variances of the i^{th} male speaker,

$\mu_i^f, (\sigma_i^f)^2$: means and variances of the i^{th} female speaker,

N_m : number of male speakers in the reference database,

N_f : number of female speakers in the reference database.

Now, if the unknown means of a speaker are presented to the system, these unknown means will be compared against the overall means of male voices and the overall means of female voices using the following distance equations :

$$D_m = \sum_{k=1}^{12} \frac{(x_k - (\mu_m)_k)^2}{(\sigma_m^2)_k} \tag{6.5}$$

$$D_f = \sum_{k=1}^{12} \frac{(x_k - (\mu_f)_k)^2}{(\sigma_f^2)_k} \tag{6.6}$$

D_m measures how far the unknown means are from the overall male means and variances. Similarly, D_f measures how far the unknown means are from the overall female means and variances. If D_m is greater than D_f, the unknown means are closer to female regions than to male regions. The unknown means can be labeled as female means. However, there are two types of errors associated with the use of these distances. Male means may be labeled as female means, and female means may be labeled as male means. Typical LPC PARCOR plots of the overall male / female means and variances are presented in Figure 6.1. The resulting performance using the above method are tabulated in Tables 6.1, 6.2, and 6.3.

Table 6.1 : Male / female separation with rectangular window.

Features	Correct Females	Males as Females	Correct Males	Females as Males
LPC A	125	11	315	11
LPC **K**	**130**	**7**	**319**	**6**
LPC C	129	8	318	7

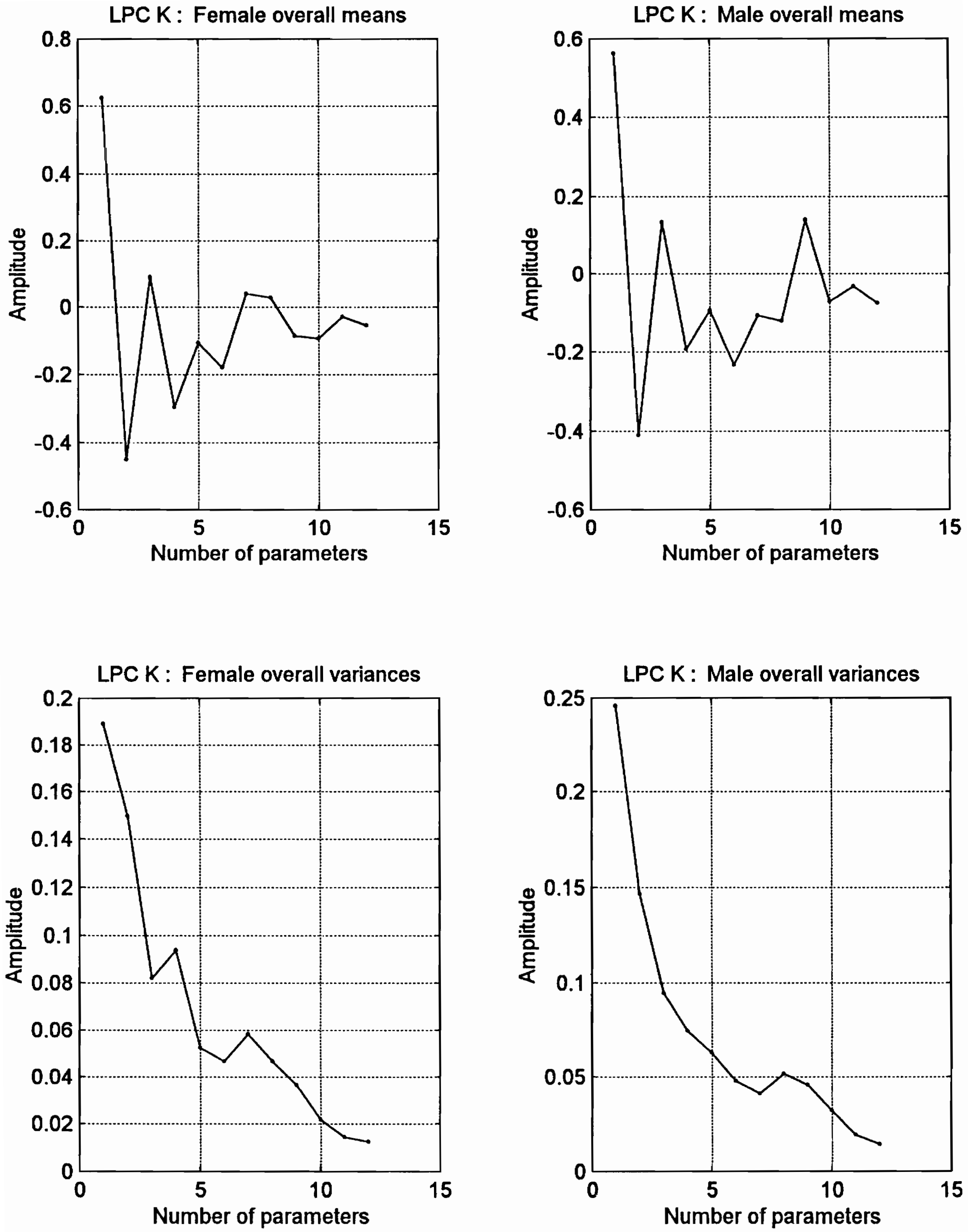


Figure 6.1 Plots of female/ male overall means and variances of LPC K.

Table 6.2 : Male / female separation with Hamming window.

Features	Correct Females	Males as Females	Correct Males	Females as Males
LPC **A**	127	19	307	9
LPC **K**	**131**	**9**	**317**	**5**
LPC **C**	129	10	316	7

Table 6.3 : Male / female separation with preemp filter & Hamming window.

Features	Correct Females	Males as Females	Correct Males	Females as Males
LPC **A**	129	8	318	7
LPC **K**	**132**	7	**319**	**4**
LPC **C**	128	8	318	8

The "best" performance is highlighed in each table. In general, the LPC **K** features always provide better performance than the other two LPC features. The "best" overall performance comes from the use of a first-order preemphasis filter and Hamming window. However, better performance is needed for a *high accuracy* ASR system.

6.1.2 Female/ Male Parameter Separation Approach

First, the plots of 12 dimensional clusters are designed to see the overall effects of each parameter of spectral features **a**, **k**, and **c**. Both female and male clusters of means, variations, and absolute deviations are plotted in the following Figures 6.2, 6.3, and 6.4 for each spectral feature **a**, **k**, and **c** respectively. These plots help to visualize the overall

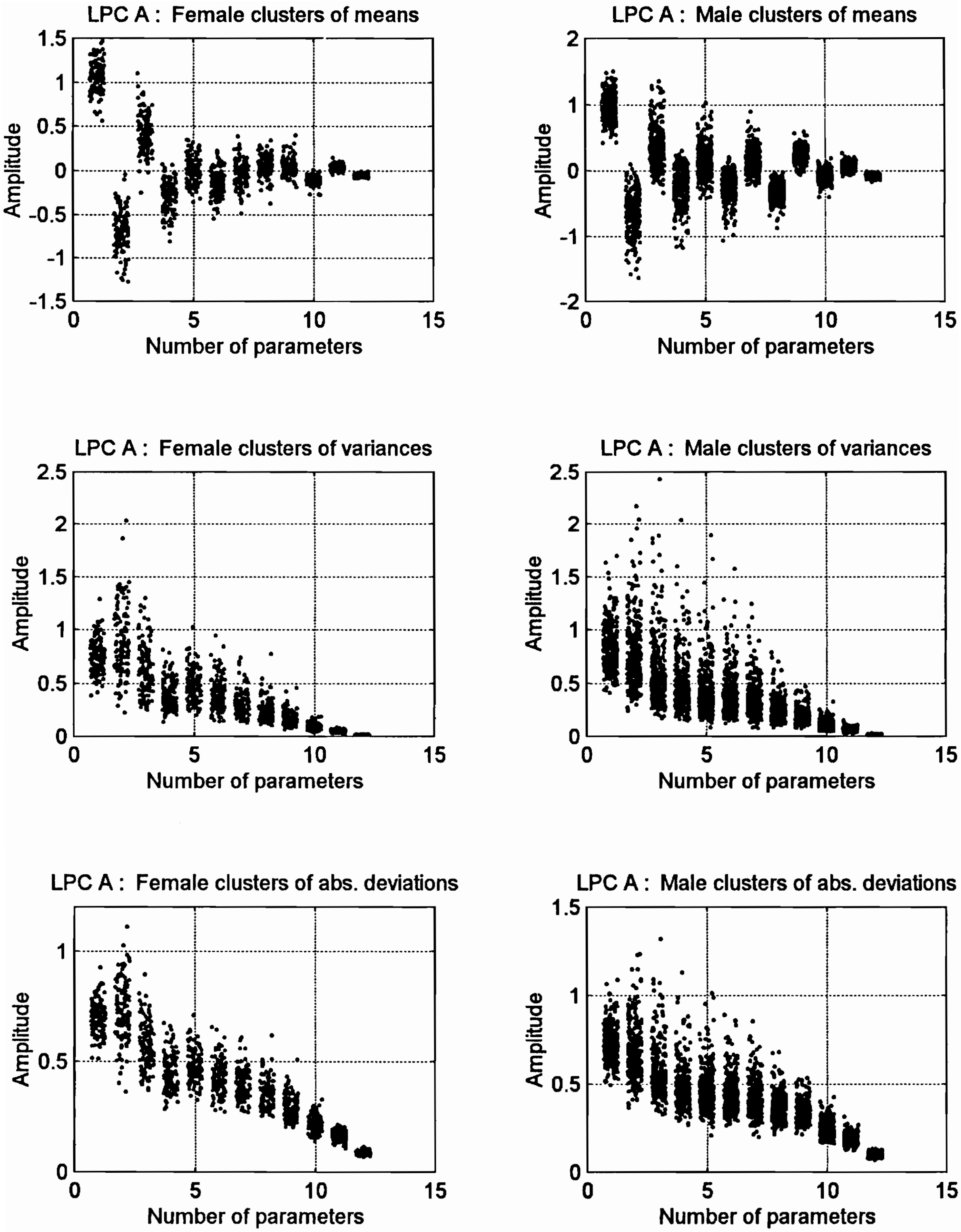


Figure 6.2 Female/ male clusters of means, variances, and absolute deviations of LPC A.

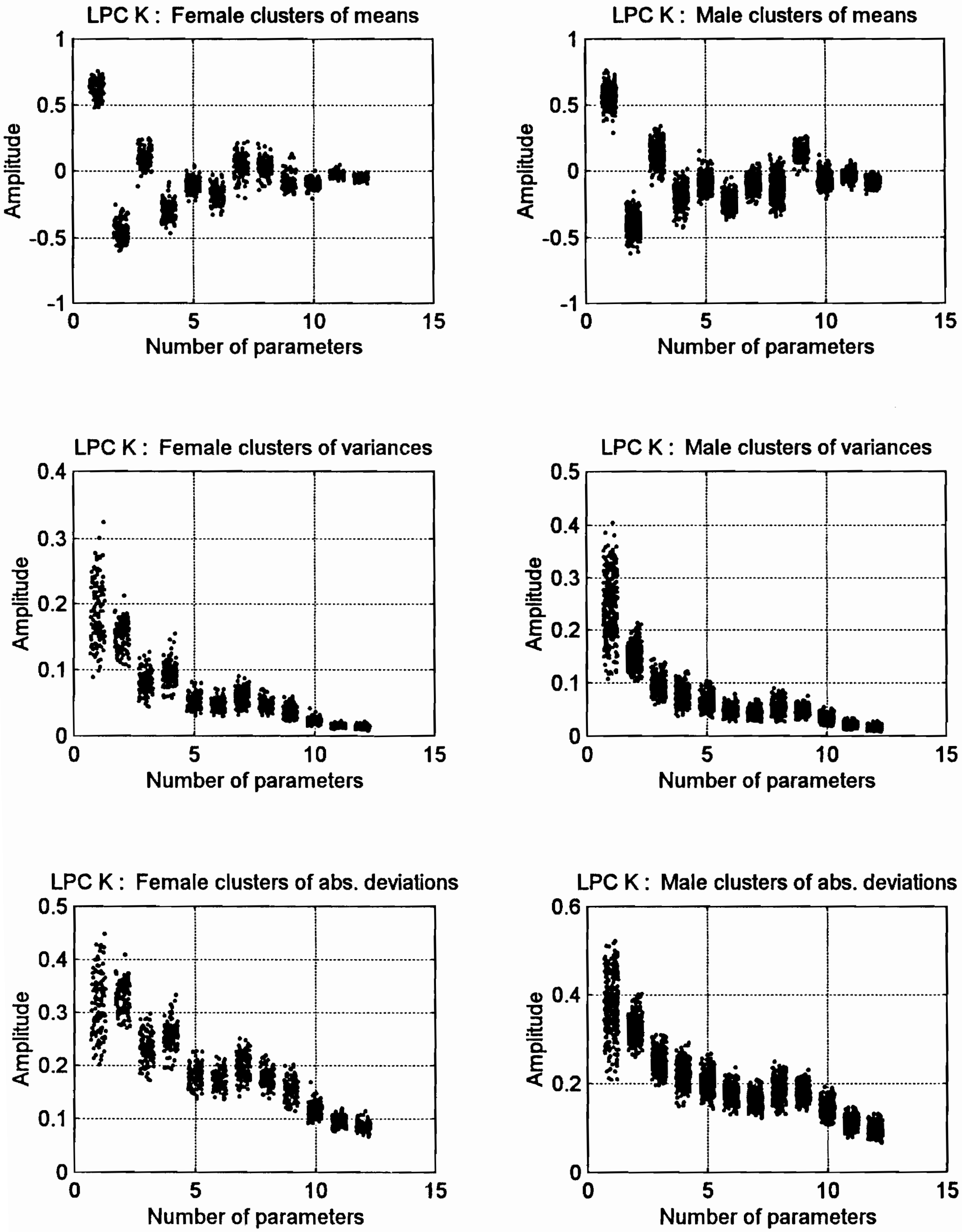


Figure 6.3 Female/ male clusters of means, variances, and absolute deviations of LPC K.

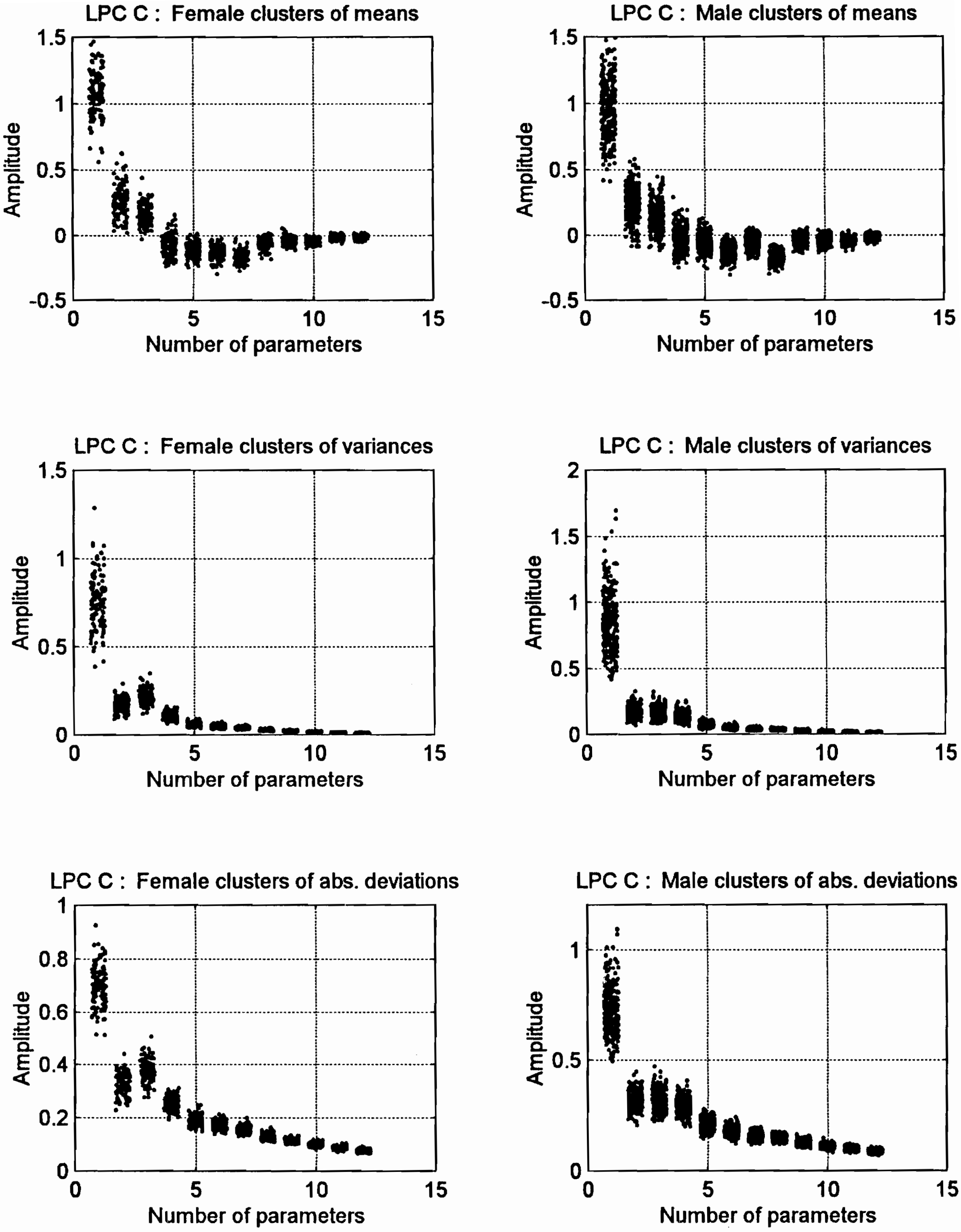


Figure 6.4 Female/ male clusters of means, variances, absolute deviations of LPC C.

locations of male voices and female voices, and to act as an aid to the search for the right parameter that would separate most male voices and female voices. They also provide a visual big picture of how these parameters fit together, and the intraspeaker variability of each parameter. A careful examination of these plots show that there is no parameter in both LPC **A** and LPC **C** that can definitely separate most male speakers from female speakers. Parameter 8 of the cepstral features has some separation power. However, there is one parameter in Figure 6.3 that clusters the male voices far from the same parameter clusters of the female voices. It is the parameter 9 of the means of the LPC PARCOR coefficients in Figure 6.3 that provides this high separation power. The plot of the parameter 9 of the means of the PARCOR features for all 462 speakers is provided in Figure 6.5. From Figure 6.5, it is clear that the values of most of the female voices are below 0.05, and the values of most of the male voices are above 0.05. There are only a few overlapping individual male voices and female voices. In addition, the *seventh* and *eighth* parameters of the means of the PARCOR features also provide large female/ male separation as shown in Figure 6.5. Table 6.4 summaries several important thresholds of parameters which can be used to separate male voices from female voices. The question that arises immediately is how to separate the overlapping voices.

The female/ male separation performance based on a combination of the 9^{th} parameter and a few others of the LPC PARCOR **K** and the use of the D_m and D_f in previous section is summarized in Tables 6.5, 6.6, and 6.7. The "best" performance in each table is highlighted. From these tables, the female / male separation performance of just the ninth parameter of feature **k** outperforms the overall means and overall variances method in previous section. The combination of this parameter with the eighth parameter provides an even better result. In general, the "best" performance from these tables yields a total of 8 male/ female mislabeled out of 462 speakers, **a error of 1.7 %**. However, a better result can be achieved by making use of some of the information provided in Table 6.4. For example, there is no male speaker with the k9 value below or even close to -0.05, but there are 114 female voices with the k9 value below -0.05. Next, the principal components analysis (PCA) will be applied to speaker features in order to improve the performance of separating male voices from female voices. First, the principle of components analysis will be introduced in the next section.

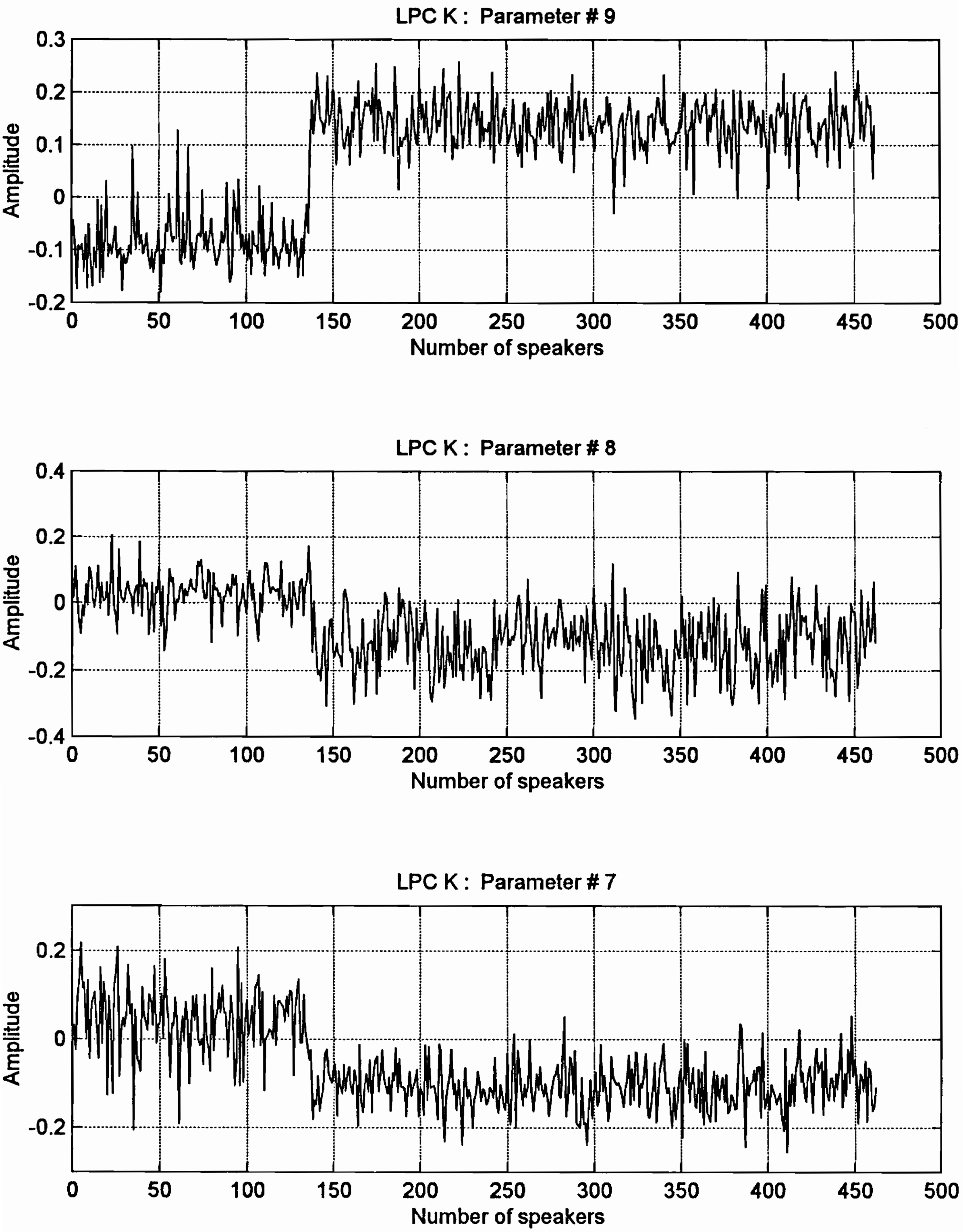


Figure 6.5 Plots of parameter *9th*, *8th*, and *7th* of the means of LPC K.

Table 6.4 Several important contribution thresholds of each parameters.

Parameter #	Thresholds	# Females	# Males
k4	> -0.2	6	174
	< -0.3	63	27
k6	< -0.3	4	33
k7	> 0.05	62	0
	> 0.00	97	8
	< -0.1	8	191
k8	> 0.1	15	1
	> 0.0	97	32
	< 0.0	39	294
	< -0.1	**3**	**189**
k9	**< -0.05**	**114**	**0**
	< 0.0	125	3
	> 0.05	**3**	**317**
	> 0.1	1	263
k10	> 0.0	0	36
	> -0.05	10	105
k12	< -0.1	0	61

Table 6.5 : Male / female separation using parameters of K with rect. window.

Features	Correct Females	Males as Females	Correct Males	Females as Males
k9 alone	128	6	320	8
k9+k6	130	5	321	6
k9+k6+k10	**130**	**3**	**323**	**6**

Table 6.6 : Male / female separation using parameters of K with Hamm. window.

Features	Correct Females	Males as Females	Correct Males	Females as Males
k9 alone	129	6	320	7
k9+k8	133	6	320	3
k9+k8+k12	**133**	**5**	**321**	**3**

Table 6.7 : M / F separation using parameters of K with filter & Hamm. window.

Features	Correct Females	Males as Females	Correct Males	Females as Males
k8 alone	129	3	323	7
k8+k6	131	3	323	5
k8+k10	**131**	**3**	**323**	**5**

6.1.3 Principal Components Analysis

Feature selection or feature extraction from the data is a key issue in statistical pattern recognition. The process of transforming a data space into a feature space that has exactly the same dimension as the original data space is referred to as *feature selection.* The transformation is designed to represent the original data in an effective way. Principal components analysis (PCA), also known as the *Karhunen-Loeve transformation,* maximizes the rate of decrease of variance which is optimum in the mean-square error sense [36]. Principal components analysis is the best-known technique in multivariate analysis. Pearson (1901) who first introduced the method, used it in a biological context to recast linear regression analysis into a new form. Independently, Karhunen (1947) used it in the setting of probability theory, and it was generalized by Loeve in 1963.

Let **X** represent a random data matrix of (N x p) dimension with nonzero mean values, and let **V** be the transform matrix of (p x p) dimension on which the matrix **X** is to be projected. The projection matrix **P** (N x p) is defined below :

$$\mathbf{P} = \mathbf{XV} \tag{6.7}$$

subject to the constraint

$$\mathbf{V}^T\mathbf{V} = \mathbf{I} \tag{6.8}$$

where **I** = (p x p) identity matrix.

The mean vector of the projection matrix P is defined by :

$$\overline{\mathbf{P}} = E[\mathbf{P}] = E[\mathbf{X}]\mathbf{V} \tag{6.9}$$

Then the variance of the matrix P is given below :

$$\begin{aligned}\text{var}\ (\mathbf{P}) &= E\left[(\mathbf{P}-\overline{\mathbf{P}})^2\right] \\ &= E\left[(\mathbf{P}-\overline{\mathbf{P}})^T(\mathbf{P}-\overline{\mathbf{P}})\right] \\ &= E\left[(\mathbf{XV}-E(\mathbf{XV}))^T(\mathbf{XV}-E(\mathbf{XV}))\right] \\ &= E\left[\mathbf{V}^T(\mathbf{X}-E(\mathbf{X}))^T(\mathbf{X}-E(\mathbf{X}))\mathbf{V}\right] \\ &= \mathbf{V}^T E\left[(\mathbf{X}-E(\mathbf{X}))^T(\mathbf{X}-E(\mathbf{X}))\right]\mathbf{V} \\ &= \mathbf{V}^T\mathbf{RV}\end{aligned} \tag{6.10}$$

The (p x p) symmetric matrix **R** is the covariance matrix of the data matrix **X**, and is defined below :

$$\mathbf{R} = E\left[(\mathbf{X}-E(\mathbf{X}))^T(\mathbf{X}-E(\mathbf{X}))\right] \tag{6.11}$$

The next question is to find the matrix **V** along which var(**P**) has local maxima or minima, subject to a constraint on the Euclidean norm of **V**. The solution to this problem is well known, and lies in the eigenstructure of the covariance matrix **R**.

In short, the columns of the projection matrix **V** are the eigenvectors of **R**, and the variance of the matrix data **X**, the eigenvalues of **R**. Singular value decomposition (SVD) is used to find the eigenvalues and eigenvectors of the covariance matrix **R**. The next method will make use of the principal components analysis to increase the performance of female / male separation considered in previous sections.

6.1.4 Female/ Male Separation Approach Using PCA Technique

In previous sections, the "best" approach resulted in a total of 8 female/ male mislabeled. The need to search for a perfect male/ female separation is very difficult since one female voice in the database has a high value of the ninth parameter of **k** placing it inside the male region. However, since the goal of female/ male separation is to increase the speed of the speaker recognition system, it is only necessary to find the exact male voice or female voice. In other words, it suffices to find some way to group the speech voices into male-type voice and female-type voice. From Figure 6.5, the following criteria are used to determine the male-type voice and the female-type voice :

- If $k_9 > 0.1$ ===> label the voice as male-type voice.
- If $k_9 < -0.05$ ===> label the voice as female-type voice.
- Otherwise, male/ female will be labeled by the algorithm developed below.

where k_9 is the 9th parameter of LPC **k**.

In this ASR system, it is assumed that the female/ male voices in the reference speaker feature database are known at the registration time. If this information is known, the analysis of female/ male separation is possible. From Figure 6.5, there is one female voice with $k_9 > 0.1$. This female voice will be labeled as male-type voice, and grouped with other male-type voices. There is no male speaker with $k_9 < -0.05$. Table 6.8 describes an algorithm to generate the male-type voice and female-type voice templates.

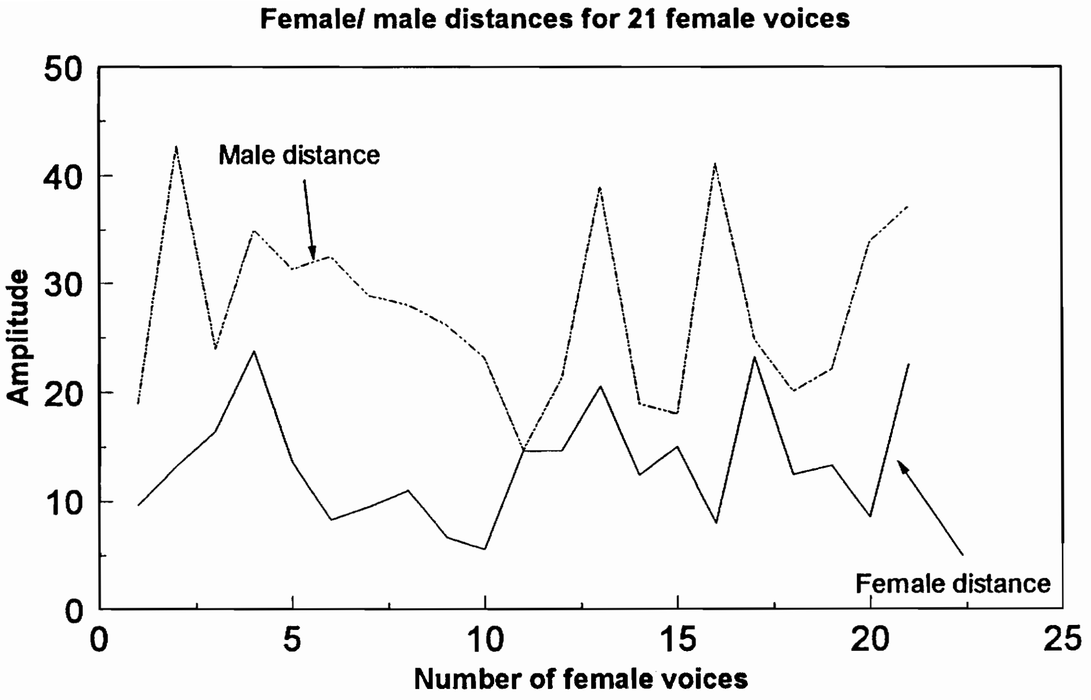
Table 6.8 : Algorithm to generate male-type & female-type templates.

Step 1 :	Read data matrix of (462 x 12) of LPC K from train features database.
Step 2 :	Separate into matrix of male-type voice and matrix of female-type voice.
Step 3 :	Keep only speakers with $k_9 > -0.1$.
Step 4 :	Compute the covariance matrix from the female-type matrix of means. Compute the eigenvalues and eigenvectors of the covariance matrix. Project the female-type matrix using the above eigenvectors. Compute overall female-type means from the new projected matrix. Store and label these eigenvectors, eigenvalues, and overall female-type means as tools for labeling the female-type voice later.
Step 5 :	Compute the covariance matrix from the male-type matrix of means. Compute the eigenvalues and eigenvectors of the covariance matrix. Project the male-type matrix using its eigenvectors. Compute overall male-type means from the new projected matrix. Store and label these eigenvectors, eigenvalues, and overall male-type means as tools for labeling the female-type voice later.

Once the male-type and female-type templates are created, the process of labeling an unknown voice into male-type voice and female-type voice is described by the algorithm in Table 6.9. In general, these algorithms use principal components analysis, several thresholds, and a Mahalanobis distance to improve the female/ male-type separation performance. Then these two algorithms are evaluated on the test speaker feature database. After making use of step 2 in Table 6.9, voices of 400 speakers can be labeled as either male-type voice or female-type voice, leaving 62 voices to be labeled using the rest of the algorithm. In the end, all male-type and female-type voices are identified correctly. A **99.9% accuracy** is achieved. Figure 6.6 shows the results of the female / male separation of the 62 voices which cannot be identified from the thresholds. There are 21 female voices out of the 62 voices. When a female voice is presented to the system, the distance of this voice to the female overall means is smaller than the distance of this voice to the male overall means as shown in the top part of Fig. 6.6. Similarly, the distance of the male voice is closer to the male overall means than to the female overall means. Since the algorithm can label female/ male very accurately, the number of comparisons in speaker identification is reduced considerably. Thus, a robust ASR system is achieved. The next several methods are proposed to improve the recognition accuracy which was not achieved in previous sections.

Table 6.9 : Algorithm to label a voice as male-type or female-type voice.

Step 1 :	Read a speaker voice and extract the necessary means of LPC **k** using methods described in Chapters 4 & 5.
Step 2 :	If k_9 < -0.05 : return < female-type voice >. If k_9 > 0.10 : return < male-type voice >. Otherwise, continue.
Step 3 :	Project the voice data matrix using female-type eigenvectors. Project the voice data matrix using male-type eigenvectors
Step 4 :	Compute the distance of the female-type projected matrix from the overall female-type means and the female-type eigenvalues.
Step 5 :	Compute the distance of the male-type projected matrix from the overall male-type means and the male-type eigenvalues.
Step 6 :	If the female-type resulting distance < the male-type resulting distance, return < female-type voice >. else, return < male-type voice>.



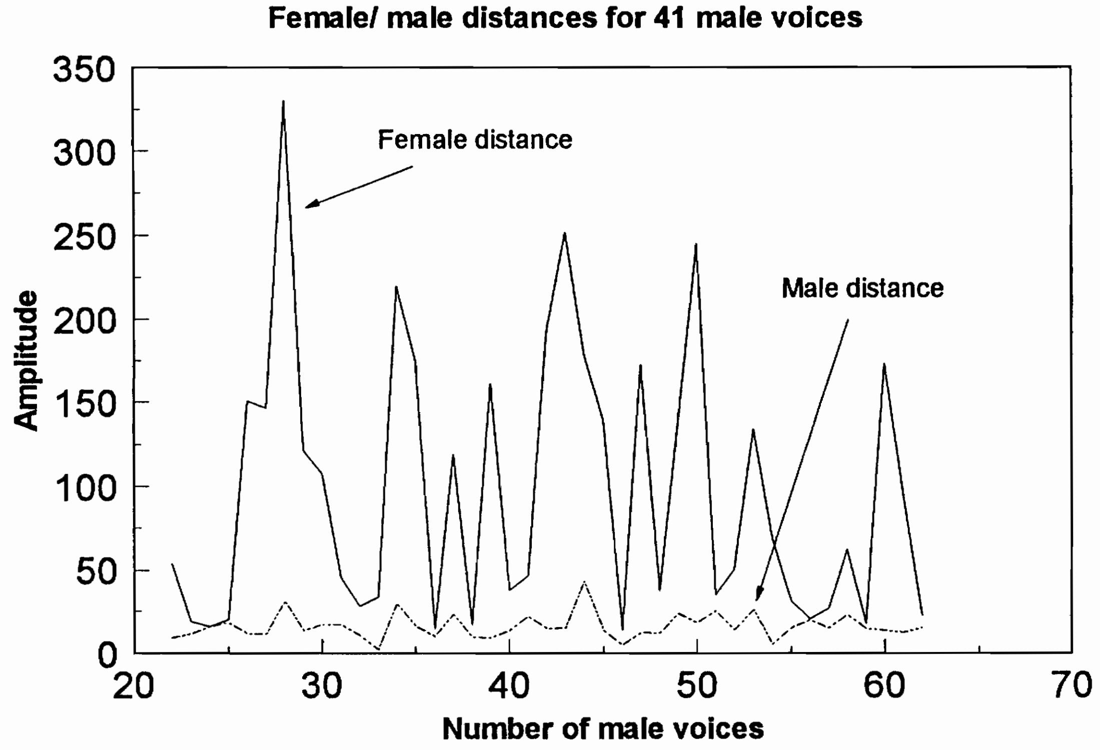


Figure 6.6 Plots of female/ male separation performance.

6.2 Robust ASR System Design

In previous sections, linear predictive coefficients (LPC) **a** have consistently resulted in a lower recognition performance than their counterparts, particularly the LPC PARCOR **k**, and the LPC cepstral **c.** The rectangular window does not appear to provide good results and will not be used. The preemphasis filter also will not be employed in the final design version. Therefore, the final design of the robust ASR system will consist of only the Hamming window version of the LPC **k** and LPC **c** as speaker features.

The following topics are presented in this section :

- A new way to generate the ASR reference speaker feature database.
- Two new spectral dissimilarity measurements.
- The robust speaker identification algorithm.
- The robust speaker verification algorithm.
- The reduction of the identification processing time in half.

The new speaker features are based on the orthogonal linear prediction method [98], and are extended with the weighted combination of features k, and c to improve the overall accuracy of the automatic speaker recognition system.

6.2.1 ASR Reference Speaker Feature Database

The reference speaker feature database generated for the robust ASR system is different from the previous reference speaker feature database in that the system will store a projection matrix, a variance vector, as well as the mean vector of the orthogonal parameters. The block diagram of the procedure to create the reference speaker feature database is presented in Figure 6.7. First, the system will read a reference speech signal s(n) either from an audio wave recorder or from the ASR reference (train) speaker database. The signal s(n) is passed thru the speech endpoint detection where noise, silence, pauses, and weak unvoiced sounds are removed. Next, the LPC **k** and **c** are computed on a basis of a 30 ms (240 point) Hamming window with 50% overlapping frames using the autocorrelation method and Durbin's recursive algorithm. The resulting spectral features **k**, and **c** for each frame are stored in **K** and **C** matrices respectively for all frames in an utterance. For each of the **K** and **C** matrices, the principal components analysis technique is employed to transform these matrices into a new data space in an effective way, maximizing the rate of decrease of variance which is optimum in the mean-square sense. The new projected feature data are computed as following for matrix K :

- Compute the covariance **R** of the matrix **K** (N x p) :

$$\mathbf{R} = E\left[(\mathbf{K} - E(\mathbf{K}))^T (\mathbf{K} - E(\mathbf{K}))\right] \tag{6.11}$$

- Compute the eigenvalues and eigenvectors of the covariance matrix **R** (p x p) :

$$[\mathbf{V},\mathbf{D}] = SDV(\mathbf{R}) \tag{6.12}$$

where *SVD* = single value decomposition method.

V = eigenvectors of **R**.

D = eigenvalues of **R**.

- Compute the new projected data :

$$\mathbf{P} = \mathbf{KV} \tag{6.13}$$

- Compute the mean vector of the new projected matrix **P** (N x p) :

$$\bar{p}_m = \frac{1}{N}\sum_{k=1}^{N} p_{km} \quad , \qquad m = 1, 2, \ldots, 12 \tag{6.14}$$

$$\bar{\mathbf{p}} = [\bar{p}_1 \quad \bar{p}_2 \; \cdots\cdots \; \bar{p}_{12}] \tag{6.15}$$

- Store the following parameters as a speaker features for the LPC **K** :

$$\{\mathbf{V},\mathbf{D},\bar{\mathbf{p}}\} \tag{6.16}$$

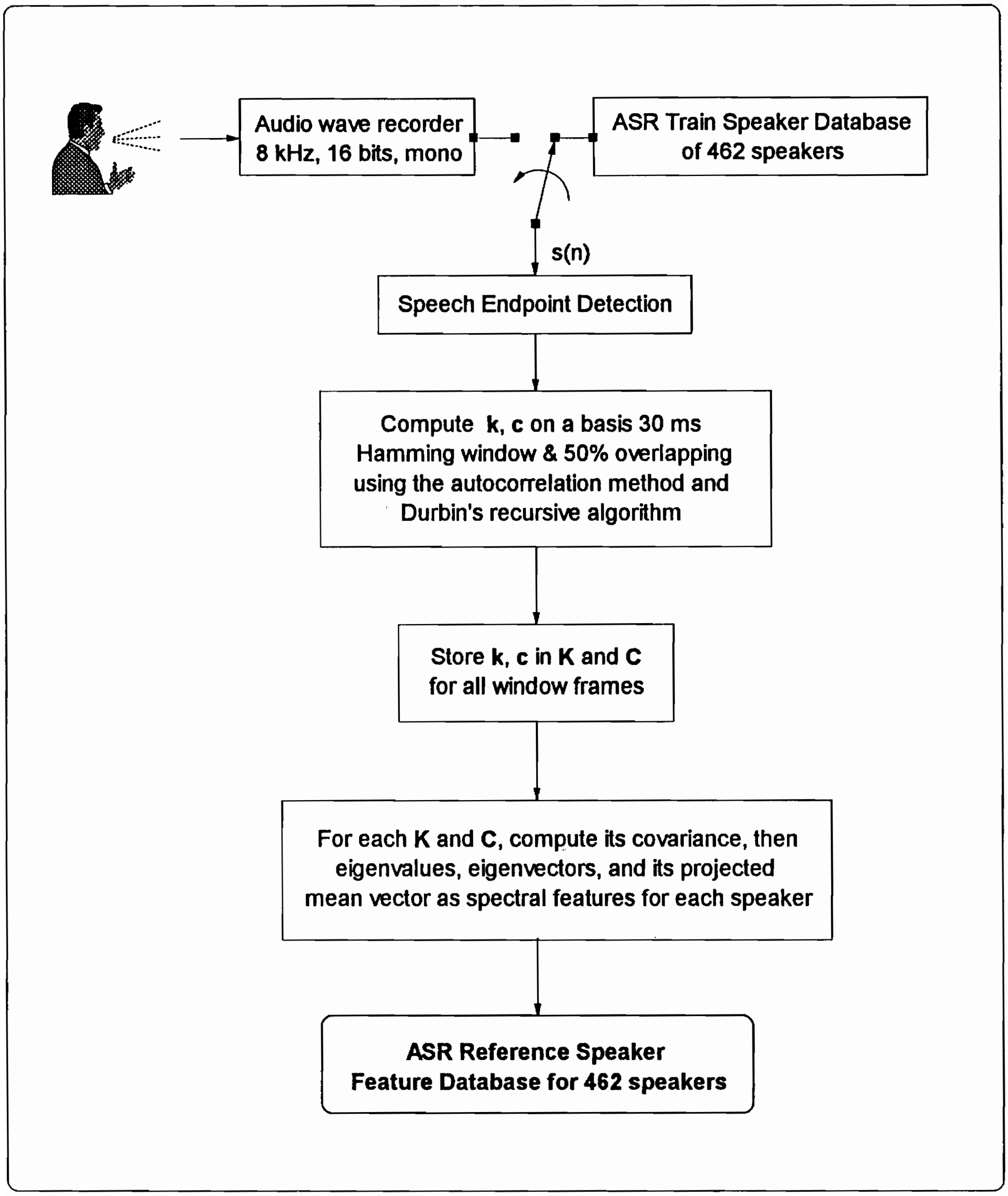

Audio wave recorder
8 kHz, 16 bits, mono
ASR Train Speaker Database
of 462 speakers
s(n)
Speech Endpoint Detection
Compute k, c on a basis 30 ms
Hamming window & 50% overlapping
using the autocorrelation method and
Durbin's recursive algorithm
Store k, c in K and C
for all window frames
For each K and C, compute its covariance, then
eigenvalues, eigenvectors, and its projected
mean vector as spectral features for each speaker
ASR Reference Speaker
Feature Database for 462 speakers

Figure 6.7 ASR Reference Speaker Feature Database for 462 speakers.

The process continues until the features of all 462 speakers have been created and stored in the reference speaker feature database. In a summary, the following spectral features are stored for each speaker in the ASR reference speaker feature database :

- $\{\mathbf{V},\mathbf{D},\bar{\mathbf{p}}\}$ from LPC **K** features.
- $\{\mathbf{V},\mathbf{D},\bar{\mathbf{p}}\}$ from LPC **C** features.

6.2.2 Spectral Distances

The following spectral distances are used to compute the spectral distortion of the two spectral patterns **r** and **t** :

$$d_1(\mathbf{r},\mathbf{t}) = \sum_{k=1}^{12} \frac{(r_k - t_k)^2}{\lambda_{rk}} \tag{6.17}$$

$$d_2(\lambda,v) = \sum_{k=1}^{12} \left(\frac{v_k - \lambda_k}{\lambda_k} \right)^2 \tag{6.18}$$

- d_1 represents the distance between the reference pattern **r** and the test pattern **t** using the variance of the reference pattern.
- d_2 represents the distance between the variance λ of the reference pattern **r** and the variance v of the test pattern **t** using the variance of the reference pattern.

The reference pattern is the speaker feature set in the ASR reference speaker feature database, whereas the test pattern is usually generated from an unknown speaker voice. The variance of the reference pattern is the eigenvalue set of the speaker features.

6.2.3 Speaker Recognition Procedure

The block diagram of the speaker recognition procedure is presented in Figure 6.8. The speaker voice is first digitized and endpointed. The speaker features are extracted

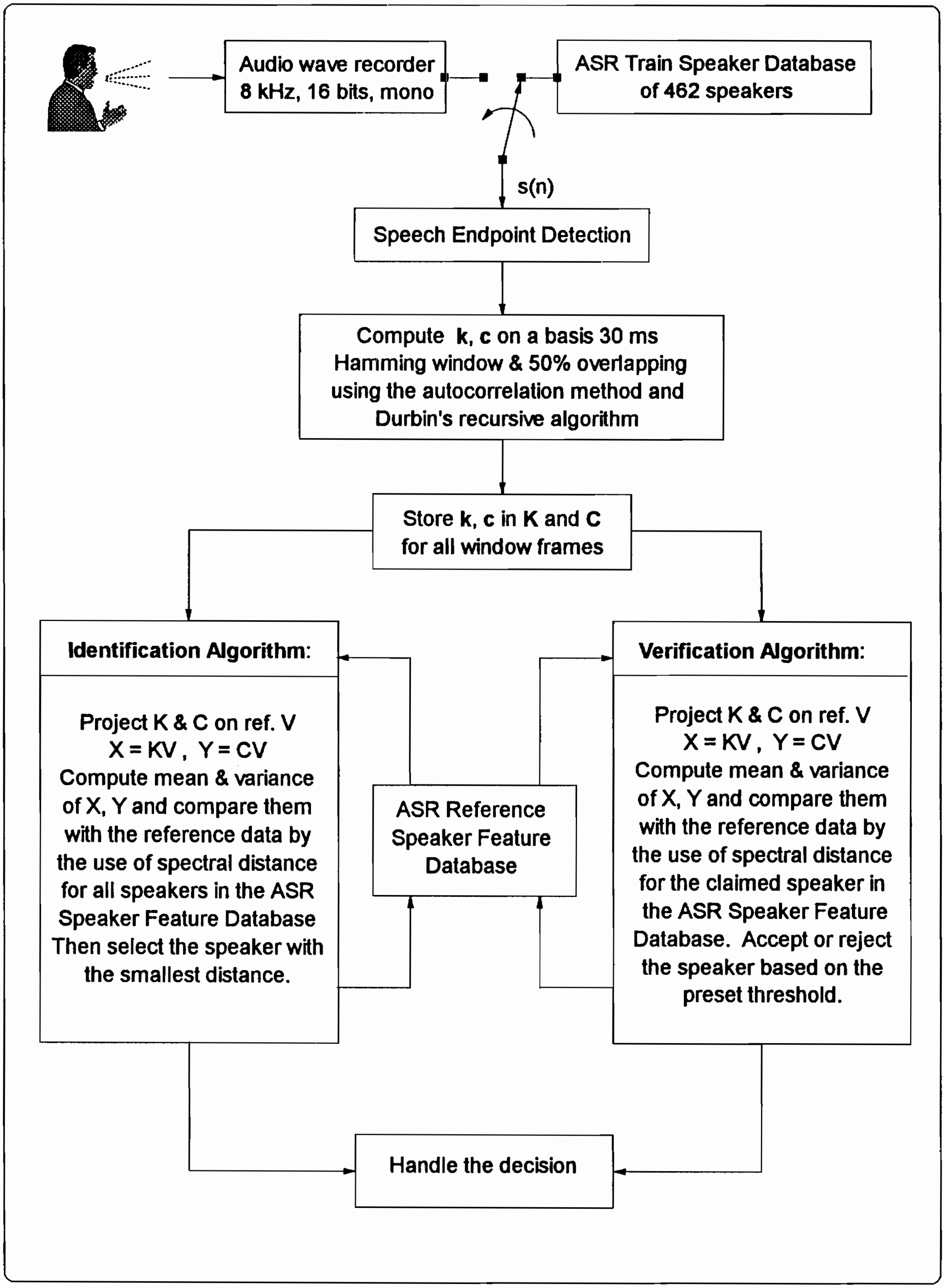


Figure 6.8 Speaker recognition procedure.

from the processed signal for K and C. The identification algorithm in the ASR system is summaried in Table 6.10.

Table 6.10 : Speaker identification algorithm.

1. Set i = 1.
2. Retrieve the i^{th} speaker features : eigenvalues, eigenvectors, and feature means from the ASR reference speaker feature database.
3. Project the **K** & **C** on the above reference eigenvectors **V**,
   $$\mathbf{X} = \mathbf{K}\,\mathbf{V} \quad \& \quad \mathbf{Y} = \mathbf{C}\,\mathbf{V}.$$
4. Compute the means and variances of matrices **X** and **Y**.
5. Compute and store the spectral distances d_1 and d_2 against the reference patterns of the i^{th} speaker features.
6. while ((i=i+1) <= 462), repeat steps 2 to 5.
7. Select the speaker with the smallest distance.

Table 6.11 : Speaker verification algorithm.

1. Retrieve the claimed speaker features : eigenvalues, eigenvectors, and feature means from the ASR reference speaker feature database.

2. Project the **K** & **C** on the claimed-speaker reference eigenvectors **V**,

$$\mathbf{X} = \mathbf{K}\,\mathbf{V} \quad \& \quad \mathbf{Y} = \mathbf{C}\,\mathbf{V}.$$

3. Compute the means and variances of matrices **X** and **Y**.

4. Compute and store the spectral distances d_1 and d_2 against the reference patterns of the claimed-speaker features.

9. Compare the resulting distances with the preset threshold. If the distances are less than the threshold, it is a correct speaker. Otherwise, it is an imposter.

6.2.4 Robust ASR Design

The following features are used in building a final design of the robust ASR system : the speaker features generated from the LPC PARCOR K, and the speaker features generated from the LPC cepstral C. The following distance is used to compare two speakers patterns :

- Distance for features derived from K :

$$d_K(\mathbf{r}_K, \mathbf{t}_K) = d_1(\mathbf{r}_K, \mathbf{t}_K) + 0.225 d_2(\lambda_K, v_K) \tag{6.19}$$

- Distance for features derived from C :

$$d_C(\mathbf{r}_C, \mathbf{t}_C) = d_1(\mathbf{r}_C, \mathbf{t}_C) + 0.15 d_2(\lambda_C, v_C) \tag{6.20}$$

- Total distance separation of reference speaker features and test speaker features :

$$d_T(\mathbf{r},\mathbf{t}) = d_K(\mathbf{r}_K,\mathbf{t}_K) + 0.65d_C(\mathbf{r}_C,\mathbf{t}_C) \qquad (6.21)$$

where $\mathbf{r}_c$ = reference speaker features derived from C,

$\mathbf{t}_c$ = test speaker features derived from C,

$\mathbf{r}_k$ = reference speaker features derived from K,

$\mathbf{t}_k$ = test speaker features derived from K.

λ_c = variance (eigenvalues) of reference speaker features derived from C,

ν_c = variance of test speaker features derived from C,

λ_k = variance (eigenvalues) of reference speaker features derived from K,

ν_k = variance of test speaker features derived from K.

The final robust speaker recognition algorithms are listed in Tables 6.12, and 6.13. The performance of the algorithms are summaried in Table 6.14.

Table 6.12 : Final robust speaker identification algorithm.

1. Pass test speaker feature K to male/female-type separation algorithm. If the return value is 'female-type', then use the 'female-type' reference speaker feature database only.

2. Compute $d_K(\mathbf{r}_K, \mathbf{t}_K)$ and $d_C(\mathbf{r}_C, \mathbf{t}_C)$ for all speakers of the same type.

3. Select the speaker with smallest distance among d_k , and select the speaker with the smallest distance among d_c .

4. If both selected speakers are the same person, return <person identity> with 99.01% confidence. Otherwise, continue.

5. Compute the total distance d_T for all speakers of the same type, then select the speaker with the smallest distance.

Table 6.13 : Final robust speaker verification algorithm.

1. Compute $d_K(\mathbf{r}_K, \mathbf{t}_K)$ and $d_C(\mathbf{r}_C, \mathbf{t}_C)$ for the claimed speaker and the test speaker.

2. If distance d_k < threshold$_{\mathrm{k}}$, and distance d_c < threshold$_{\mathrm{c}}$, then accept the speaker with 99.9% confidence. Otherwise, continue.

3. If the total distance d_T < threshold$_{\mathrm{T}}$, then accept the speaker with 97.6% accuracy. Otherwise reject.

Table 6.14 : Speaker recognition errors using the final robust ASR design.

		Identification errors		**Verification errors**	
		Female	Male	Female	Male
K&C	d_T	7 (5.1%)	19 (5.8%)	4 (2.9%)	7 (2.1%)

Therefore, this robust ASR system will reduce the identification processing time in half, along with a 94.4% overall identification accuracy, and a 97.6% overall verification accuracy.

The computation cost and memory requirement are listed below :

- For each speaker, a (12 x 12) eigenvector matrix, 12 eigenvalues, and 12 mean values for **k**, and **c** are stored in the speaker feature database. The total memory requirement for each speaker in the database is 1,344 bytes. Therefore, the total storage memory for all speakers are 620,928 bytes.
- A typical raw 10 seconds of speech data for speaker recognition requires 160,000 bytes of memory.
- The processing time is about 10 seconds for speaker verification, and 2-3 minutes for speaker identification based on a 90 MHz pentium computer.

Chapter 7

Conclusion

The robust ASR system above reported a 94.4% overall speaker identification accuracy, and a 97.6% overall speaker verification accuracy based on 462 speakers at 8 kHz sampling frequency. Howcver, there are 7 out of 11 errors in the verification process that can be recovered because the distance next to the smallest distance selects those 7 speakers. Similarly, there are 11 out of 26 errors in identification that can be recovered because the distance next to the smallest distance selects those 11 speakers. If these situations can be solved, a 96.8% overall identification accuracy and a 99.1% overall verification accuracy can be achieved.

Even though the final combination of features has the "best" result compared to result from individual features, the spectral distance using the cepstral features correctly identify 2 out of 4 mis-identified female speakers in the verification process, and correctly identify 2 out of 7 mis-identified male speakers in the verification process.

Here are the results from several other papers on the TIMIT database :

- The speaker recognition rate on 24 male speakers [111] are 75% using HC models, 70.8% using SHC models, and 54.2% using PS models.
- The speaker recognition rate on 420 speakers [6] are 99.5% at 16 kHz sampling frequency.
- The speaker recognition rate on 420 speakers [62] are 95.3% at 16 kHz sampling frequency.

Based on this research work, the best recognition results come from the "optimal" spectral distance and the combination of spectral features. The spectral distance is designed to match the training and testing features. Clustering techniques using neural

networks have been used successfully in pattern matching tasks. Clustering has advantages over the weighted distance approach being that the weighted distance approach uses a single (scalar) distance threshold, whereas the threshold used in clustering is defined by a cluster boundary located in the multidimesional feature space. Consequently, the cluster boundary threshold has more degrees of freedom and can be chosen based on the separation of the clusters. In this light, if the principal components analysis employed after clustering the data into useful clusters may help the recognition accuracy.

The linear predictive coding coefficients **a** have lower recognition rate than their counterparts, namely the LPC PARCOR **k** and the LPC cepstral **c**. Principal components analysis has proven helpful in separating male-type voice and female-type voice as well as the speaker recognition performance.

References

[1] B.S. Atal, "Automatic Recognition of Speakers from Their Voices," *Proc. IEEE*, vol. 64, no. 4, pp. 460-475, April 1976.

[2] B.S. Atal and L.R. Rabiner, "A Pattern Recognition Approach to Voiced-Unvoiced-Silence Classification with Application to Speech Recognition," *IEEE Trans. ASSP*, pp. 201-212, June 1976.

[3] B.S. Atal, "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification," *Journal of the Acoustical Society of America*, vol. 55, pp. 1304-1312, June 1974.

[4] J.B. Attili, M. Savic, and J.P. Campbell, "A TMS32020 Based Real-Time, Text-Independent, Automatic Speaker Verification System," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 599-602, 1988.

[5] M. Baraniecki and M. Shridhar, "A Speaker Verification Algorithm for Speech Utterances Corrupted by Noise with Unknown Statistics," *Proc. Intl. Conf. Acoustics, Speech & Signal Processing*, pp.904-907, 1980.

[6] F. Bimbot, L. Mathan, A. DeLima, and G. Chollet, "Standard and Target Driven AR-Vector Models for Speech Analysis and Speaker Recognition," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. II, pp.5-8, 1992.

[7] R. Bogner, "On Talker Verification via Orthogonal Parameters," *IEEE Trans. ASSP*, ASSP-29, pp. 1-12, 1981.

[8] G.B. Brown and P.Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, (John Wiley and Sons, New York, 1992).

[9] E. Bunge, "Automatic Speaker Recognition System AUROS for Security Systems and Forensic Voice Identification," *Proc. Intl. Conf. on Crime Countermeasures _ Sci. & Eng.*, pp. 1-7, 1977.

[10] E. Bunge, U. Hofker, P. Jesorsky, B. Kriener, and D. Wesseling, "Statistical Techniques for Automatic Speaker Recognition," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 772-775, 1977.

[11] C.F. Chan and K.W. Law, "Thinned Lattice Filter for LPC Analysis," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. I, pp. 117-120, 1992.

[12] C. Charalambous, "Conjugate Gradient Algorithm for Efficient Training of Artificial Neural Networks", *IEE Proceedings-G*, Vol. 139, No. 3, pp 301-310, 1992.

[13] S.H. Chen and M.T. Lin, "On the Use of Pitch Contour of Mandarin Speech in Text-Independent Speaker Identification," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 1418-1421, 1987.

[14] R. Cheung and B. Eisenstein, "Feature Selection via Dynamic Programming for Text-Independent Speaker Identification," *IEEE Trans. ASSP*, ASSP-26, pp. 397-403, 1978.

[15] E. Chilton and B.G. Evans, "The Spectral Autocorrelation Applied to The Linear Prediction Residual of Speech for Robust Pitch Detection," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 358-361, 1988.

[16] J.Y. Choi, M. Tran, P. Huynh, and H.F. VanLandingham, "Optimal Control of Nonlinear Systems Using Neural Networks," Proceedings of the 1992 Southeastern Symposium on System Theory, N.C.A.&T University, Greensboro, NC.

[17] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, (John Wiley & Sons Ltd., 1993).

[18] P. Corsi, "Speaker Recognition : A Survey," *Automatic Speech Analysis and Recognition*, pp. 277-308, 1982.

[19] S.K. Das and W.S. Mohn, "A Scheme for Speech Processing in Automatic Speaker Verification," *IEEE Trans. Audio Electroacoust.*, vol. AU-19, pp. 32-43, March 1971.

[20] J.R. Deller, J.G. Proakis, and J.H.L. Hansen, *Discrete-Time Processing of Speech Signals*, (Macmillan Publishing Co., New York, 1993).

[21] E.S. Dermatas, N.D. Fakotakis, and G.K. Kokkinakis, "Fast Endpoint Detection Algorithm For Isolated Word Recognition in Office Environment," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 733-736, July 1991.

[22] G.R. Doddington, "Personal Identity Verification Using Voice," *Proc. Electro-76*, pp.22-24, 1976.

[23] B. Doval and X. Rodet, "Fundamental Frequency Estimation and Tracking Using Maximum Likelihood Harmonic Matching and HMMs," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. I, pp.221-224, 1993.

[24] J.J. Dubnowski, R.W. Schafer, and L.R. Rabiner, "Real-time Digital Hardware Pitch Detector," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-24, pp. 2-8, Feb. 1976.

[25] A.M. Engebretson, "Benefits of Digital Hearing Aids," *IEEE Engineering in Medicine and Biology magazine*, pp. 238-248, April / May 1994.

[26] J.A. Freeman and D.M. Skapura, *Neural Networks: Algorithms, Applications, and Programming Techniques*, (Addison-Wesley Co., Reading Massachusetts, 1991).

[27] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification," *IEEE Trans. ASSP*, ASSP-29, pp. 254-272, 1981.

[28] S. Furui, "Comparison of Speaker Recognition Methods Using Statistical Features and Dynamic Features," *IEEE Trans. ASSP*, ASSP-29, pp. 342-350, 1981.

[29] S. Furui and A.E. Rosenberg, "Experimental Studies in a New Automatic Speaker Verification System Using Telephone Speech," *Proc. ICASSP*, pp. 1060-1062, 1980.

[30] W.A. Gardner, *Introduction to Random Processes*, (McGraw-Hill, Inc., 1990).

[31] J. Garofolo, et al. , "DARPA TIMIT : Acoustic-Phonetic Continuous Speech Corpus", Gaithersburg, MD.: *U.S. Department of Commerce*, 1993.

[32] H. Gish and M. Schmidt, "Text-Independent Speaker Identification," *IEEE Signal Processing Magazine*, pp. 18-32, Oct. 1994.

[33] B. Gold and L.R. Rabiner, "Parallel Processing Techniques for Estimating Pitch Periods of Speech in The Time-Domain," *J. Acoust. Soc. Amer.*, vol. 46, pp. 442-448, Aug. 1969.

[34] M.H. Goldstein, Jr., "Auditory Periphery As Speech Signal Processor," *IEEE Engineering in Medicine and Biology magazine*, pp. 186-196, April / May 1994.

[35] H. Hattori, "Text Independent Speaker Recognition Using Neural Networks," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. II, pp. 153-156, 1992.

[36] S. Haykin, *Neural Networks*, (MacMillan/IEEE Press, 1994).

[37] R. Hecht-Nielsen, *Neurocomputing*, (Addison-Wesley, Menlo Park, CA., 1990).

[38] P. Hedelin and D. Huber, "Pitch Period Determination of Aperiodic Speech Signals," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 361-364, 1990.

[39] X. Huang and K.F. Lee, "On Speaker-Independent, Speaker-Dependent, and Speaker-Adaptive Speech Recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 1, no. 2, pp. 150-157, April 1993.

[40] M.J. Hunt, J.W. Yates, and J.S. Bridle, "Automatic Speaker Recognition for Use over Communication Channels," *IEEE Intl. Conf. Record on Acoust., Speech & Signal Processing*, pp. 764-767, 1977.

[41] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-23, pp. 67-72, 1975.

[42] L.B. Jackson, *Digital Filters and Signal Processing*, (Norwell, M.A.: Kluwer Academic Publishers, 1986).

[43] P. Jesorsky, U. Hofker, and Maati Talmi, "Extraction of Speaker Specific Features from Spoken Code Sentences," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 279-282, 1978.

[44] C. Johnson, H. Hollien, and J. Hicks, "Speaker Identification Utilizing Selected Temporal Speech Features," *Journal of Phonetics*, no. 12, pp. 319-326, 1984.

[45] Y.H. Kao, J.S. Baras, and P.K. Rajasekaran, "Robustness Study of Free-Text Speaker Identification and Verification," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. II, pp. 379-382, 1993.

[46] Y.H. Kao, P.K. Rajasekaran, and J.S. Baras, "Free-Text Speaker Identification Over Long Distance Telephone Channel Using Hypothesized Phonetic Segmentation," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. II, pp. 177-180, 1992.

[47] Y. Kato and M. Sugiyama, "Speaker-Independent Features Extracted by a Neural Network," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. I, pp. 553-556, 1993.

[48] B. Kosko, *Neural Networks and Fuzzy Systems*, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1992).

[49] B. Kosko, *Neural Networks for Signal Processing*, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1992).

[50] W.B. Kuhn, "A Real-Time Pitch Recognition Algorithm for Music Applications," *Computer Music Journal*, vol. 14, no. 3, pp.60-71, Fall 1990.

[51] S.Y. Kung, *Digital Neural Networks*, (PTR Prentice Hall, 1993).

[52] S.Y. Kwon, A.J. Goldberg, D. Ng, and K. Ouellette, "A Robust Realtime Pitch Extraction From the ACF of LPC Residual Error Signals," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 403-406, August, 1985.

[53] L.F. Lamel, L.R. Rabiner, A.E. Rosenberg, and J.G. Wilpon, "An Improved Endpoint Detector For Isolated Word Recognition," *IEEE ASSP*, vol. 29, pp. 777-785, August 1981.

[54] K.P. Li and E.H. Wrench, "An Approach to Text-Independent Speaker Recognition with Short Utterances," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 555-558, 1983.

[55] W.C. Lin and S.K. Pillay, "Feature Evaluation and Selection for an On-Line, Adaptive Speaker Verification System," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 734-737, 1976.

[56] J.D. Markel, "The SIFT Algorithm for Fundamental Frequency Estimation," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 367-377, Dec. 1972.

[57] J.D. Markel, B.T. Oshika, and A.H. Gray, "Long-Term Feature Averaging for Speaker Recognition," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-25, pp. 330-337, August 1977.

[58] S.L. Marple, *Digital Spectral Analysis with Applications*, (Englewood Cliffs, N.J. : Prentice-Hall, Inc., 1987).

[59] C. McGonegal, A. Rosenberg, and L.R. Rabiner, "The Effects of Several Transmission Systems on an Automatic Speaker Verification System," *Bell Sys. Tech. Journal*, no. 58, pp. 2071-2087, 1979.

[60] N.J. Miller, "Pitch Detection by Data Reduction," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-23, pp. 72-79, Feb. 1975.

[61] N. Mohankrishnan, M. Shridhar, and M.A. Sid-Ahmed, "A Composite Scheme for Text-Independent Speaker Recognition," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 1653-1656, 1982.

[62] C. Montacie, P. Deleglise, F. Bimbot, and M.J. Caraty, "Cinematic Techniques for Speech Processing : Temporal Decomposition and Multivariate Linear Prediction," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. I, pp. 153-156, 1992.

[63] D.P. Morgan and C.L. Scofield, *Neural Network and Speech Processing*, (Boston, M.A.: Kluwer Academic Publishers, 1991).

[64] J.M. Naik, "Speaker Verification : A Tutorial", *IEEE Communications Magazine*, pp. 42-48, January 1990.

[65] J.M. Naik, L.P. Netsch, and G.R. Doddington, "Speaker Verification Over Long Distance Telephone Lines," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 524-527, 1989.

[66] J.M. Naik and G.R. Doddington, "High Performance Speaker Verification Using Principal Spectral Components," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 881-884, 1986.

[67] K.S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. on Neural Networks*, vol. 1, no. 1, pp. 4-27, March 1990.

[68] H. Ney and R. Gierloff, "Speaker Recognition Using a Feature Weighting Technique," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 1645-1648, 1982.

[69] H. Ney, "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition," *IEEE Trans. ASSP*, ASSP-32, pp. 263-271, 1984.

[70] H. Ney, "An Optimization Algorithm for Determining the Endpoints of Isolated Utterances," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 720-723, 1981.

[71] H. Ney, "Telephone-Line Speaker Recognition Using Clipped Autocorrelation Analysis," *Proc. Intl. Conf. Acoustics, Speech & Signal Processing*, pp. 188-191, 1980.

[72] H. Noda, "Frequency-Warped Spectral Distance Measures for Speaker Verification in Noise," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 576-579, 1988.

[73] K.A. Oh and C.K. Un, "A Performance Comparison of Pitch Extraction Algorithms for Noisy Speech," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, 1984.

[74] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1989).

[75] A.V. Oppenheim, editor, *Applications of Digital Signal Processing*, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1978).

[76] D. O'Shaughnessy, "Speaker Recognition", *IEEE ASSP Magazine*, pp. 4-17, October 1986.

[77] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, (Addison Wesley Co., 1989).

[78] J.E. Paul, A.S. Rabinowitz, J.P. Riganati, and J.M. Richardson, "Development of Analytical Methods for A Semi-Automatic Speaker Identification System," *Carnahan Conf. on Crime Countermeasures*, pp. 52-64, May 1975.

[79] G.E. Peterson and H.L. Barney, "Control Methods Used in a Study of The Vowels," *Journal of the Acoustical Society of America*, vol. 24, pp. 175-184, 1952.

[80] J. Picone, G.R. Doddington, and B.G. Secrest, "Robust Pitch Detection In a Noisy Telephone Environment," *Proc. Intl. Conf. Acoust., Speech & Signal Proc.*, pp. 1442-1445, 1987.

[81] D.P. Prezas, J. Picone, and D.L. Thomson, "Fast and Accurate Pitch Detection Using Pattern Recognition and Adaptive Time-Domain Analysis," *Proc. Intl. Conf. Acoust., Speech & Signal Proc.*, pp. 109-112, April 1986.

[82] L.R. Rabiner, A.E. Rosenberg, and S.E. Levinson, "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," *IEEE Trans. ASSP-26*, pp. 575-582, December 1978.

[83] L.R. Rabiner and M.R. Sambur, "An Algorithm for Determining the Endpoints of Isolated Utterances," *Bell Syst. Tech. J.*, vol. 54, no. 2, pp. 297-315, Feb. 1975.

[84] L.R. Rabiner, C.E. Schmidt, and B.S. Atal, "Evaluation of a Statistical Approach to Voiced-Unvoiced-Silence Analysis for Telephone-Quality Speech," *Bell System Tech. J.*, pp. 455-487, Mar. 1977.

[85] L.R. Rabiner and M.R. Sambur, "Voiced-Unvoiced-Silence Detection Using the Itakura LPC Distance Measure," *IEEE ASSP*, 1977.

[86] L.R. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1993).

[87] L.R. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1978).

[88] L.R. Rabiner, M.J. Cheng, A.E. Rosenberg, and C.A. McGonegal, "A Comparative Performance Study of Several Pitch Detection Algorithms," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-24, pp. 399-418, Oct. 1976.

[89] P.A. Regalia, S.K. Mitra, and P.P. VaiDyanathan, "The Digital All-Pass Filter: A Versatile Signal Processing Building Block," *Proceedings of the IEEE*, vol. 76, no. 1, pp.19-37, Jan. 1988.

[90] H. Ritter, T. Martinetz, and K. Schulten, *Neural Computation and Self-Organizing Maps*, (Addison Wesley Co., 1992).

[91] R.C. Rose, E.M. Hofstetter, and D.A. Reynolds, "Integrated Models of Signal and Background with Application to Speaker Identification in Noise," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 2, pp. 245-257, April 1994.

[92] A.E. Rosenberg and K.L. Shipley, "Speaker Identification and Verification Combined With Speaker Independent Word Recognition," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 184-187, 1981.

[93] A.E. Rosenberg and F.K. Soong, "Evaluation of a Vector Quantization Talker Recognition System in Text Independent and Text Dependent Modes," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 873-876, 1986.

[94] A.E. Rosenberg, "Automatic Speaker Verification : A Review," *Proc. IEEE*, vol. 64, pp. 475-487, April 1976.

[95] A.E. Rosenberg and M.R. Sambur, "New Techniques for Automatic Speaker Verification," IEEE Trans. Acoust., Speech, and Signal Processing, vol. ASSP-23, pp.169-176, April 1975.

[96] M.J. Ross, H.L. Shaffer, A. Cohen, R. Freuberg, and H.J. Manley, "Average Magnitude Difference Function Pitch Extractor," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-22, pp. 353-362, Oct. 1974.

[97] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-26, pp.43-49, 1978.

[98] M.R. Sambur, "Speaker Recognition Using Orthogonal Linear Prediction," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-24, pp. 283-289, August 1976.

[99] M.R. Sambur, "Text Independent Speaker Recognition Using Orthogonal Linear Prediction," *Proc. Intl. Conf. Acoustics, Speech & Signal Processing*, pp. 727-729, 1976.

[100] M.R. Sambur, "Selection of Acoustic Features for Speaker Identification," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-23, pp. 176-182, April 1975.

[101] M. Savic and S.K. Gupta, "Variable Parameter Speaker Verification System Based on Hidden Markov Modeling," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp.281-284, 1990.

[102] M. Savic and J. Sorensen, "Phoneme Based Speaker Verification," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. II, pp. 165-168, 1992.

[103] M.H. Savoji, "A Robust Algorithm for Accurate Endpointing of Speech Signals," *Speech Communication 8*, pp. 45-60, 1989.

[104] B.G. Secrest and G.R. Doddington, "Postprocessing Techniques for Voice Pitch Trackers," *Proc. Intl. Conf. Acoust., Speech & Signal Proc.*, pp. 172-175, 1982.

[105] B.G. Secrest and G.R. Doddington, "An Integrated Pitch Tracking Algorithm for Speech Systems," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp.1352-1355, June, 1983.

[106] M. Shridhar, N. Mohankrishnan, and M.A. Sid-Ahmed, "A Comparison of Distance Measures for Text-Independent Speaker Identification," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp.559-562, 1983.

[107] M. Shridhar and M. Baraniecki, "Accuracy of Speaker Verification via Orthogonal Parameters for Noisy Speech," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 785-788, 1979.

[108] P.K. Simpson, *Artificial Neural Systems : Foundations, Paradigms, Applications, and Implementations*, (Pergamon Press, 1990).

[109] F.K. Soong, A.E. Rosenberg, L.R. Rabiner, and B.H. Juang, "A Vector Quantization Approach to Speaker Recognition," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 387-390, 1985.

[110] F.K. Soong and A.E. Rosenberg, "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp.877-880, 1986.

[111] H.B.D. Sorensen and U. Hartmann, "Pi-Sigma and Hidden Control Based Self-Structuring Models for Text-Independent Speaker Recognition," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. I, pp. 537-540, 1993.

[112] L.S. Su, K.P. Li, and K.S. Fu, "Identification of Speakers by Nasal Coarticulation," *Journal of the Acoustical Society of America*, vol. 156, pp. 1876-1882, Dec. 1974.

[113] C. Tsao and R.M. Gray, "An Endpoint Detector for LPC Speech Using Residual Error Look-Ahead for Vector Quantization Applications," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, section 18B.7.1, May 1984.

[114] H.F. VanLandingham, S. Bingulac, and M. Tran, "A Comparison of Conventional and Neural Network Approaches to System Identification," *Control-Theory and Advanced Technology, Journal of MITA Press*, vol. 9, no. 1, pp.77-97, 1993.

[115] H.F. VanLandingham, M. Tran, and J.Y. Choi, "Multivariate Process Identification," Proceedings of the IEEE Systems, Man, and Cybernetics, Le Touquet, France, vol. 3, pp. 435-440, Oct. 1993.

[116] H.F. VanLandingham, M. Tran, and J.Y. Choi, "System Identification with Multilayer Perceptrons," Proceedings of the IEEE Systems, Man, and Cybernetics, Le Touquet, France, Oct. 1993.

[117] G. Velius, "Variants of Cepstrum Based Speaker Identity Verification," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 583-586, 1988.

[118] L.X. Wang, *Adaptive Fuzzy Systems and Control*, (PTR Prentice Hall, 1994).

[119] M.L. Whitehead, B.B. Stagner, B.L. Lonsbury-Martin, and G.K. Martin, "Measurement of Otoacoustic Emissions For Hearing Assessment," *IEEE Engineering in Medicine and Biology magazine*, pp. 210-226, April / May 1994.

[120] J.G. Wilpon, L.R. Rabiner, and T. Martin, "An Improved Word-Detection Algorithm for Telephone Quality Speech Incorporating Both Syntactic and Semantic Constraints," *AT&T Bell Laboratories Technical Journal*, vol. 63, no. 3, pp. 479-498, March 1984.

[121] J.G. Wilpon, L.R. Rabiner, C.H. Lee, and E.R. Goldman, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. 38, no. 11, pp. 1870-1878, November 1990.

[122] R.E. Wohlford, E.H. Wrench, and B.P. Landell, "A Comparison of Four Techniques for Automatic Speaker Recognition," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 908-911, 1980.

[123] L. Xu, J. Oglesby, and J.S. Mason, "The Optimization of Perceptually Based Features for Speaker Identification," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 520-523, 1989.

[124] G.S. Ying, C.D. Mitchell, and L.H. Jamieson, "Endpoint Detection of Isolated Utterances Based On a Modified Teager Energy Measurement," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, vol. 2, pp. 732-735, April 1993.

[125] Y. Zhao, "A Speaker-Independent Continuous Speech Recognition System Using Continuous Mixture Gaussian Density HMM of Phoneme-Sized Units," *IEEE Trans. on Speech and Audio Processing*, vol. 1, no. 3, pp.345-361, July 1993.

[126] Y.C. Zheng and B.Z. Yuan, "Text Dependent Speaker Identification Using Circular Hidden Markov Models," *Proc. Intl. Conf. Acoust., Speech & Signal Processing*, pp. 580-582, 1988.

[127] D.R. Hush and B.G. Horne, "Progress in Supervised Neural Networks," *IEEE Signal Processing Magazine*, pp. 8-39, January 1993.

[128] "Technical Tutorial Seminar : Research Issues in Text-to-Speech Conversion," *IEEE Home Video Tutorial*, Product no. HV0089-3.

[129] "Technical Tutorial Seminar : Neural Network Based Speech Recognition Systems," *IEEE Home Video Tutorial*, Product no. HV0088-5.

Vita

Born in Saigon, Vietnam, Michael Tran came to the United States in 1979, and graduated from Wakefield High School in 1983. He then attended and later graduated from Virginia Tech with a B.S. degree in Electrical Engineering in 1986. Upon graduation, he joined Texas Instruments as an electrical design engineer for two years. Taking leave on educational absence, he enrolled in the Electrical Engineering graduate program at Virginia Tech where he later completed the M.S. and Ph.D. degrees. He also expects to complete the M.B.A. degree. Mr. Tran is the recipient of a Pratt Presidential Fellowship, Texas Instruments Quality Excellence Award, and the Renssenlaer Medal Award. He is a member of IEEE, Phi Eta Sigma, Tau Beta Pi, and Eta Kappa Nu. He enjoys traveling, collecting postcards, playing volleyball, and roller-skating.