1

CANADIAN THESES ON MICROFICHE

I.S.B.N.

THESES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Division

Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

59992

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

- Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

ERIC CHARLES SAMSON

Date of Birth — Date de naissance

Sept. 4, 1957

Country of Birth — Lieu de naissance

CANADA

Permanent Address — Résidence fixe

58 EASTPARK DR.
OTTAWA, ONTARIO
K1B 4A1 CANADA

Title of Thesis — Titre de la thèse

A MIXED ANALOG DIGITAL APPROACH TO AN
LSI CHIP DESIGN FOR AUTOMATIC WORD RECOGNITION

University — Université

CARLETON UNIVERSITY

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

Doctor of Philosophy

Year this degree conferred — Année d'obtention de ce grade

1983

Name of Supervisor — Nom du directeur de thèse

Dr. M.A. Copeland

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication, ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur

Date

Sept. 13, 1983

Signature

Eric Samson

A MIXED ANALOG DIGITAL APPROACH TO AN
LSI CHIP DESIGN FOR AUTOMATIC WORD RECOGNITION

by

© Eric Charles Samson

A thesis submitted to the Faculty of Graduate Studies and
Research in partial fulfillment of the requirements for the
degree of Doctor of Philosophy.

Faculty of Engineering
Carleton University
Ottawa, Ontario
September, 1983

The undersigned hereby recommend to the Faculty of
Graduate Studies and Research acceptance of the thesis,
submitted by Eric Charles Samson

B.Eng., M.Eng.,
in partial fulfilment of the requirements for
the degree of Doctor of Philosophy

.....

Chairman, Department of Electronics

.....

Thesis Supervisor

.....

External Examiner

Carleton University

September 2, 1983

ABSTRACT

A novel single chip mixed analog digital proposal for the realization of an automatic word recognition algorithm is described in this thesis. The work was approached by considering the interaction between integrated circuit implementation technology and the functions required by speech recognition methods. A class matching based word recognition algorithm, rather than a feature matching algorithm, was developed since class matching algorithms tend to require less data processing, and are therefore more suited for single LSI chip design. Furthermore, the use of switched capacitor techniques for performing class decisions is proposed. Suggestions as to how the main obstacle to the use of these techniques - dc offset caused by op amp voltage offset and clock feedthrough - can be circumvented are made. The means with which the class decisions may be manipulated so as to yield good recognition results is also described. Finally, results of real time breadboarding of the proposed chip are presented which validate both the circuit and system concepts of the complete design.

ACKNOWLEDGEMENTS

The author wishes to express his indebtedness to his thesis supervisor, Professor M.A. Copeland, and to A. Chowaniec, whose guidance and aid have been of crucial importance. The author also wishes to thank L. McCready, D. Ribner and K. Lam for many interesting discussions, to N. Mikhail for his assistance with computer hardware, and to D. Glendinning for providing some useful graphics software.

Financial support from Telesat Canada, the Faculty of Graduate Studies and Research, the National Sciences and Engineering Research Council of Canada , and my parents is gratefully acknowledged.

TABLE OF CONTENTS

	PAGE
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Automatic Word Recognition	1
1.2.1 Design Goals for Automatic Word Recognition Systems	1
1.2.2 Design Constraints for Automatic Word Recognition Systems	3
1.3 Thesis Organization	7
CHAPTER 2 ISOLATED WORD RECOGNITION	8
2.1 Introduction	8
2.2 General Characteristics of Speech	10
2.3 Speech Parameters	17
2.3.1 Fourier Spectrum Based Coefficients	17
2.3.2 Linear Predictive Coding Methods	21
2.3.3 Other Types of Coefficients	23
2.3.4 Feature Selection for AWR	25
2.4 Word Recognition Algorithms	26
2.5 Summary of Past AWR Work	32
2.6 Time Normalization and the Distance Measure in Template and Class Matching	39
2.7 Class Extraction	50
2.7.1 Introduction	50
2.7.2 Class Extraction Via Linear Discriminant Analysis	52
2.7.3 Character String Error Correction Rules	56
2.7.4 Distance Matrix in Class Matching	56
2.7.5 Proposed Class Matching Algorithm	58
2.8 Conclusion	61
CHAPTER 3 DESCRIPTION OF AN INTEGRATED CIRCUIT FOR AWR	62
3.1 Introduction	62
3.2 Preprocessing and Class Extraction	68
3.2.1 Introduction	68
3.2.2 Sound to Electrical Signal Conversion	69
3.2.3 Antialiasing Requirements	70
3.2.4 Amplitude Normalization Via Automatic Gain Control	70
3.2.5 Feature Extraction	73
3.2.6 Class Extraction	80
3.3 Word Detection and Decision	85
3.4 Reference Pattern Training	88

	PAGE
3.5 Conclusion	88
CHAPTER 4 EXPERIMENTAL RESULTS OF THE REAL TIME SIMULATION OF THE AWR SYSTEM	94
4.1 Introduction	94
4.2 Experimental Test Set-Up	96
4.3 Weight Vector Training and Classes	98
4.4 Recognition Accuracy	105
4.4.1 Effect of Vocabulary Size and Complexity	107
4.4.2 Effect of Reference Pattern Averaging	110
4.4.3 Effect of Thresholding	112
4.4.4 Effect of Time Warping	114
4.5 Conclusion	120
CHAPTER 5 CONCLUSION	123
5.1 Introduction	123
5.2 Summary of Thesis and List of Contributions	123
5.3 Suggestions for Future Work: Circuit Implementation	126
5.4 Suggestion for Future Work: System Implementation	127
5.4.1 Algorithm Tuning	127
5.4.2 Class Selection	128
5.5 Extrapolation of Class Matching to Other Applications	129
5.5.1 Continuous Speech Recognition	129
5.5.2 Speaker-Related Recognition Systems	130
REFERENCES	131

LIST OF SYMBOLS AND ABBREVIATIONS

A	LPC filter amplitude
A/D	Analog to digital
A(j,i)	Correlation function of LPC coefficients
a, a _Q , a _K	Constants indicating capacitor ratios
a(i)	i th bit of a word, i = 1 for MSB
a(k)	LPC coefficients
AGC	Automatic gain control
AWR	Automatic word recognition
BMDP	Biomedical Data Processing
BPF	Bandpass filter
b _j	j th backward prediction residual
C	Lower bound on AGC counter output Capacitor
C _{AGC,f}	AGC total feedback capacitance
C _I	Correlated sampling capacitor
C(1)	Feature covariance vector for class 1
CP6	Carleton University computer
C _{si}	Storage capacitor for i th channel
COUNT	Counter output for AGC
CR	Reference word class vector

CT	Test word class vector
dB	Decibel
dB(B)	Sound pressure level, B weighting
DFT	Discrete Fourier transform
DSP	Digital signal processing
D(n,m) A	Accumulated distance up to point (n,m)
D(),Dq	Distance function
D T	Total distance
DTW	Dynamic time warping
d	Local distance Discriminant function
E	Expected value
FT(i)	th i feature of the test word
Fw(i)	th th i feature of the w reference word
f	Number of classes in a class vector
f j	th j forward prediction residual
f 0i	th Center frequency of i bandpass filter
f s	Sampling frequency
G	AGC gain
G(i)	Discriminant function for class i
HWR	Halfwave rectifier
Hz	Hertz (i.e., cycles per second)

IDFT	Inverse discrete Fourier transform
KHz	Kilohertz
k_m	k_m^{th} PARCOR coefficient
L	Number of time sample taken of test word
LPC	Linear predictive coding
LPF	Lowpass filter
LSI	Large scale integration
M	Number of time samples in a test word after normalization
MDAC	Multiplying digital to analog converter
MOS	Metal oxide semiconductor
MUX	Multiplexer
$m(i)$	Mean feature vector for class i
$\min []$ $\{w\}$	Find the minimum over all the values of w of the function
MSB	Most significant bit
N	Number of time samples in a reference word Average duration of a word pattern Number of bits in AGC counter Number of classes available for a decision
N _D	Number of local distance calculations
N(W)	Normalization factor
p	Number of feature in a feature vector Probability of a correct bit
p'	Probability of a correct bit after averaging
P(1)	A priori probability of event 1
PARCOR	Partial Correlation

Q	Stored charge
RAM	Random access memory
RE	Real
RMS	Root mean square
ROM	Read only memory
$r(i,k)$	Correlation coefficient at k^{th} frame of test word
SC	Switched capacitor
SR	Reference word segment vector
S/H	Sample and hold
ST	Test word segment vector
START	Address of first time sample of a test word
T	Period Test Utterance
T_1, T_2	Rejection thresholds
Tw	th Template of w word
t	Transpose
USART	Universal synchronous/asynchronous receiver/transmitter interface
V	Number of vocabulary words Voltage
VLSI	Very large scale integration
V_{DD}	Upper rail voltage supply
V_{in}	Input voltage
V_{SS}	Lower rail voltage supply
$V(z)$	z-transform of vocal tract impulse response
$W(k)$	th Weighting function of the k arc of a path

w	Word assignment variable
w(i,j,k)	Weighting coefficient of k th feature for class pair (i,j)
-1	
z	z-transform of unit delay
	Magnitude
Σ	Summation
φ, φ̄	Two phases of a 50 percent duty cycle clock
φ ₁ , φ ₁₁ , φ ₂	Clock pulses
φ _{latch}	Latching clock
φ _{reset} , φ _r	Reset clock
φ _s	Sampling clock
μ	Micro

CHAPTER 1

INTRODUCTION

1.1 Introduction

This work explores the requirement and potential of sampled analog design for an automatic word recognition system which is to be capable of realization on one integrated circuit chip. This is done by studying a speech recognition algorithm in terms of its hardware implementation. It is shown, for example, how the nature of a sampled analog design affects the choice and performance characteristics of the recognition algorithm to be used for an automatic word recognition system. Additionally, the requirements of minimum area design also influence the nature of the recognition algorithm which can be used.

1.2 Automatic Word Recognition

1.2.1 Design Goals for Automatic Word Recognition Systems

An important requirement for many machines is to have an efficient interface for interaction with man. In particular, the task of an automatic word recognition system is to make this

interface more natural, i.e., to make the machine adapt to the user rather than the user adapt to the machine. The development of speech synthesis algorithms has reached the point where it is practical for machines to be made to talk, i.e., there is one-way speech communication. However, two-way speech communication, while commercially available, is likely to be as expensive as the rest of the machine being manipulated [1-3]. Although there is the inevitable proportional relationship between the cost of automatic word recognition systems and their capabilities, there are still many applications for limited systems - for tasks where the hands are busy, where mobility is required during data entry, where the environment is too severe to allow the use of a keyboard, and where only a limited vocabulary is required (telephone dialing, word processing, games, menu-based data bases...). For example, a voice data entry system used in an assembly line environment is described by Lerman [4]. Only isolated word recognition systems with a fixed preselected vocabulary have been implemented on a single integrated circuit. More sophisticated commercially available recognition systems require the interaction of several digital VLSI circuits with some software run by a mini- or micro-computer system. Thus, there is still room to reduce cost, power consumption, weight and volume in automatic word recognition systems with no loss in capability. The minimum capabilities to be surpassed by a new

integrated circuit design are then those of a fixed preselected isolated word recognition system.

1.2.2 Design Constraints for Automatic Word Recognition Systems

One of the major design constraints to be taken into account is that there is only so much signal processing possible on an integrated circuit - its architecture must be made as flexible and compact as possible.

The object of this work is to arrive at a single LSI chip circuit architecture enabling more flexible isolated word recognition. The system is restricted to isolated word recognition so that the more complex levels of computation required by more sophisticated (and expensive) recognition systems (see Figure 1.1) need not be considered. An isolated word recognition system has no need for a syntax analysis (sentence grammar) or semantic analysis (word sequence meaning). A flexible but limited vocabulary sets the context of a conversation (pragmatics). There remains the problem of acoustic analysis, i.e., phonetics and phonology. Acoustic analysis must be used: 1) to determine when a word begins or ends; 2) to code a spoken word into a form suitable for automatic recognition; and 3) to train the automatic word recognition system for a particular speaker (or group of speakers) or vocabulary.

Figure 1.1 Levels of speech analysis.

Figure 1.2 Block diagram of the general architecture for word recognition.

Although it is not strictly necessary, adaptation of any isolated word recognition architecture to continuous word recognition should also be kept in mind.

The architecture for electronic speech recognition can be divided into three main functional groups (as shown in Figure 1.2). The first function is that of feature extraction from the speech waveform (i.e., the speech must be coded into some form convenient for analysis). This basically involves carrying out some form of filtering on the speech waveform. The filtering can either be implemented digitally (after an analog to digital conversion) or by purely analog means. In the past few years, switched capacitor techniques have been developed which enable efficient monolithic sampled-analog circuit design of filters [5-6], and these techniques are of prime importance for feature extraction in this thesis.

The second function group in Figure 1.2 performs analysis on the speech and arrives at a word decision. When feature extraction is carried out digitally, the analysis function is naturally implemented digitally. This is especially so if a speech recognition algorithm based on template matching using dynamic time warping [7] is chosen. Similarly, when feature extraction is by analog means, an analog analysis technique which does not require an analog to digital conversion may be the most convenient. This is especially so if a speech recognition

algorithm based on class extraction [8-10] is chosen. These points will be discussed in succeeding chapters.

The third and final functional group shown in Figure 1.2 is that of data storage. Due to the long term but real-time access requirements, data must be stored digitally.

The ideal, of course, is to use analog techniques whenever analog techniques are more convenient and efficient than digital techniques and vice versa. Analog architectures tend to be useful for tasks which can be performed in parallel (such as the filter bank and correlation), whereas digital architectures tend to be useful in tasks requiring sequential processing (such as the word detection and decision algorithms). Thus the experimental architecture described in this work uses a combination of digital and analog design which is compatible with standard MOS fabrication processes. Recognition of elemental speech units by this architecture is achieved with a form of filtering and correlation which is a mixture of analog and digital techniques. The actual word decision is derived digitally. Thus the choice of recognition algorithm is a direct consequence of the possibility of a mixed digital and analog approach.

1.3 Organization

The contents of this work are divided into five chapters. Chapter 1 is the introduction. Chapter 2 deals with the characteristics of speech and the basic speech recognition algorithms. A recognition algorithm is developed with a mixed digital-analog implementation in mind. Chapter 3 is devoted to the circuit design of the experimental word recognition system developed. Chapter 4 contains the test results obtained using the suggested word recognition architecture. Chapter 5 is the concluding chapter. It provides a summary of the contributions of this work and mentions areas where improvements for the word recognition system may be possible. As well, it contains suggestions for alternative uses for the proposed word recognition system which were not tested out in this work and which should also be investigated.

CHAPTER 2

ISOLATED WORD RECOGNITION

2.1 Introduction

Speech recognition is a subset of pattern recognition. Figure 2.1 shows a general conceptual organization of a speech recognition system. First there is a sensing stage where the isolated words from a particular vocabulary are converted into an electronic signal. These words must then go through a preprocessing stage where a set of characteristic features for the speech waveform are derived. These features are a set of measurements obtained from the speech waveform. For example, a filter bank can be used to determine the energy content of the speech in different frequency bands. The resulting measurements (i.e., features) are characteristic of the speech waveform. Optimally, the features are a noise free set of measurements preserving only the relevant information necessary to classify the speech waveform. Unfortunately, the main problem in automatic word recognition (AWR) is the determination of exactly what constitutes the relevant information. Hence a discussion of the speech models and model parameters is given.

The next two speech recognition stages in Figure 2.1 are

Figure 2.1 The general organization of a word recognition system.

those of feature analysis and word classification, where the extracted features are subjected to an analysis which enables the classification of the spoken word. In automatic word recognition two basic schemes have evolved for these stages - template and class analysis. These two main approaches to AWR are described by discussing their theoretical basis and capabilities. Reasons are given as to why class matching was chosen for mixed digital and sampled-analog LSI implementation.

2.2 General Characteristics of Speech

The acoustic characteristics of speech are based on the mechanism of speech production. A system model for speech production is shown in Figure 2.2(a). Figure 2.2(b) illustrates the physiologic aspects of this model. There are basically three excitation sources for speech sounds: 1) periodic vocal cord vibration (at a fundamental frequency called the pitch frequency) which is represented in the system model by the output of an impulse train through the glottal pulse shape filter; 2) vocal cord obstruction, which is represented in the system model by the output of a random noise generator; and 3) vocal cord closure, which is represented by the gain control in the system model. The sound resulting from these excitations are often referred to as voiced, unvoiced, and plosive sounds, respectively. During

Figure 2.2 Speech Production: (a) simple model
(b) diagram of mechanism of speech production.

the excitation process the vocal tract (Figure 2.2(b)) is continuously adjusting its shape to produce the different speech sounds. The combinations of the different excitations and articulatory movements produce the different phonemes (those sounds that distinguish one word from another). The different gradations in excitations and vocal tract shapes to produce a single phoneme in turn produce the allophonemes (variations on a phoneme).

In particular, for Canadian english [11], there are about 42 phonemes. These have been subdivided by various researchers into groups, the composition and names of these groups depending on the motivation and criteria of the researchers. The theory of distinctive features [12-14] provides one method of subdivision. It assumes that every phoneme is composed of several non-mutually exclusive distinctive classes (where classes are characterizing attributes of a word). These are determined so as to require the minimum number of binary selections necessary for the specification of any particular phoneme. Table 2.1 contains examples of some of these classes (as proposed by Jacobson [13]), along with their corresponding acoustic and physiological correlates. The phoneme 'r', for example, is described by the classes vocalic, consonantal, and compact whereas the phoneme 'f' is described by the classes not

TABLE 2.1

SOME CLASSES PROPOSED AS DISTINCTIVE FEATURES

Classes (binary opposite in brackets)	Acoustic Correlate
Vocalic (non-vocalic)	Sharply defined formant structure
Consonantal (non-consonantal)	Low total energy
Compact (diffuse)	Higher concentration of energy in a narrow central region of spectrum
Tense (lax)	Higher total amount of energy, a larger spread of energy in spectrum and in time
Voiced (voiceless)	Presence of pitch
Nasal (oral)	Spreading of energy over wider frequency regions by a reduction in intensity in certain formants and the introduction of nasal formants

vocalic, consonantal, oral, not interrupted, diffuse, not acute, and voiced.

Another phoneme grouping, based on the study of place and manner of articulation combined with spectrographic characterization, is shown in Figure 2.3 [15,16].

The vowel and vowel-like sounds are produced by the passage of voiced excitation through different transfer functions generated by the flexible vocal cavities. The transfer function consists of a few major resonances (formants) whose exact frequencies and bandwidths are primarily determined by the positions of the tongue, jaws, and lips. The different gradations in tongue and lip positions provide a basis for the further subdivision of vowels into front, middle and back vowels.

The vowel-like sounds can be divided into two groups. The semivowels are of vowel-like nature but are characterized by a gliding transition in the vocal tract shape between adjacent phonemes. Thus the acoustic characteristics of these sounds are strongly influenced by context. The nasals are also vowel-like in nature but the velum (Figure 2.2(b)) is lowered so that air flows through the nasal cavity, giving rise to additional poles and zeroes [11,123].

Diphthongs are gliding monosyllabic speech items that

FIGURE 2.3 A phoneme grouping scheme

start at the articulatory position of one vowel and move to the position of another [16].

The names of the remaining phoneme classes are self-descriptive, i.e., the voiced and unvoiced plosives (or stops), and the voiced and unvoiced fricatives.

There are at least two disadvantages to these types of groupings. One is that they are based on idealized physiological processes of sound formation, which processes are not always present in all the allophones of a given phoneme [17,18]. Moreover, the influence of neighboring phonemes cannot be properly accounted for at the phoneme level since different phoneme combinations require differing transitional movements from the articulators [15]. There are indications that these transitional movements are used as cues in human speech recognition [19]. In addition, a speaker can sometimes distort the acoustic properties of speech sounds quite severely so that they do not exhibit the usual acoustic characteristics. The other disadvantage is that the characteristics required to determine the existence of a class may be very hard to measure and hence not very practical. This has been pointed out to be the case for the distinctive classes [14].

2.3 Speech Parameters

The vocal tract model of Figure 2.2(a) can be represented in many different ways, providing coefficients such as the cepstrum, linear predictive and partial correlation coefficients, etc., which can be used as speech parameters. The underlying idea for many of these coefficients is that speech production can be viewed as the convolution of an excitation function with a slowly varying vocal tract impulse response. Thus the time varying sequence of a particular model's coefficients could be used as the speech recognition algorithm features. Whether or not a coefficient is used as a recognition feature depends on two factors: 1) how well it performs in terms of recognition accuracy; and 2) how easily it can be derived with an integrated circuit design. The next few sections are devoted to brief descriptions of coefficients which have been used as speech recognition features in the past [20,21].

2.3.1 Fourier Spectrum-Based Coefficients

There are basically two ways in which the excitation function effect on a speech spectrum can be removed: via deconvolution (using cepstrum coefficients) and via smoothing along the frequency axis of short-time power spectra. In both cases the resulting spectrum envelope

carries phoneme information [20].

Cepstrum coefficients are obtained by separating the components of the convolution mentioned earlier [22]. This is done by first converting the time domain information into the frequency domain so that the convolution takes the form of an multiplication. Taking the logarithm of the frequency spectrum then converts the multiplication into an addition of the vocal tract response and the excitation function. These are in different frequency ranges of the resulting spectrum and hence easily separated. A complete deconvolution system is shown in Figure 2.4. It can be seen that both a discrete Fourier transform and an inverse Fourier transform are required.

Short-time power spectra are usually obtained by means of a filter bank or a Fourier transform. Filter banks have the advantage that their bandwidths at different center frequencies can be different. This property enables a larger frequency range to be covered with fewer coefficients than is generally the case with discrete Fourier transforms, which are generally of constant bandwidth. Another advantage of the filter bank approach is that its analog LSI implementation has already been shown feasible. A switched capacitor filter bank is available commercially [23,24] in the form of a bank of pairs of cascaded second order state variable filters, as shown in Figure 2.5. Rosshart [25] has

FIGURE 2.4 Cepstrum extraction for speech

Figure 2.5 Diagram of a filter bank.

implemented twenty second order audio filters using only two multiplexed operational amplifiers on 2.5 square millimeters.

2.3.2 Linear Predictive Coding Methods

One derivation of the linear predictive coding (LPC) coefficients is based on modelling the vocal tract impulse response as an all-pole filter with z-transform

$$V(z) = \frac{A}{1 - \sum a(k)z^{-k}} \quad (2.1)$$

where the LPC coefficients are the $a(k)$'s. One approach for obtaining the LPC coefficients is based on the adaptive least mean square gradient algorithm [26,27]. However, the coefficient convergence rate using this algorithm tends to be too slow for speech recognition [28-31]. Another approach, which does not have the convergence problem is based on calculating the autocorrelation values of incoming speech. Autocorrelation values directly transform into LPC coefficients using Durbin's recursion relation [32,33]. A circuit implemented by Fellman et. al. [43] for the autocorrelation approach is shown in Figure 2.6.

The lattice structure shown in Figure 2.7 can also be used for LPC speech processing as a linear predictor [16,32,34,35]. When the lattice coefficients are adapted to minimize the output mean-square error, they are called the

Figure 2.6 Autocorrelation coefficient extraction.

Figure 2.7 Analysis stages for the PARCOR coefficients.

partial correlation coefficients. The PARCOR coefficient algorithms [32,34-41] have been shown to converge quickly enough for speech processing applications [35]. Moreover, the coefficients have the advantage that they are amplitude insensitive. However, while the analog LSI implementation of the lattice LPC analysis filter required for PARCOR extraction has been implemented as shown in Figure 2.8 (requiring 11.6 square millimeters), complete LSI implementation of the PARCOR analysis algorithm has yet to be demonstrated (the digital coefficient extraction circuitry is estimated to require an additional 11.3 square millimeters) [42,43].

2.3.3 Other Types of Coefficients

Formant frequencies can be determined in a variety of ways. A peak searching algorithm can be applied to a frequency spectrum obtained from either a Fourier transform or an linear predictive analysis [44]. The formant frequencies can also be calculated directly from the LPC coefficients by calculating the poles of Eq. 2.1. Obviously, these are computationally intensive methods. Spectral enhancement techniques using phase locked loops [45] and Kalman filtering [46] have not yet been successfully applied to formant analysis although Martin's frequency controlled locked loop appears promising [44].

(a)

$$k_m = \frac{2E[I_{m-1}(1) b_{m-1}(1-r)]}{E[I_m^2(1)] + E[b_m^2(1)]}$$

$$\frac{A_{\text{res}m}}{A_{\text{res}m-1}} = \frac{1-k_m}{1+k_m} = \frac{E[I_{m-1}(1) - b_{m-1}(1-r)]^2}{E[I_{m-1}(1) + b_{m-1}(1-r)]^2}$$

(b)

Figure 2.8 Lattice adaptive filter: (a) switched capacitor implementation of a lattice filter, (b) digital portion of lattice adaptive filter.

Zero crossing analysis is also a much less computationally involved method of obtaining coefficients. These coefficients are derived from information pertaining to the time location of successive zero-crossings of the speech time waveform. Under certain conditions, the coefficients also contain first and second formant information [47]. Unfortunately, not much has been published on the effectiveness of these coefficients for speech recognition [21].

2.3.4 Feature Selection for AWR

Since no theory exists which can preselect the information bearing features of a pattern, feature selection is heuristic and uses ad hoc procedures [48]. Only actual experience can evaluate the value of a feature set. Hints along these lines can be abstracted from the workings of the ear, however. For example, while the hearing process is not fully understood, it is known that the ear physically behaves as a bank of nearly constant bandwidth filters below 1000 Hertz, and constant Q filters above 1000 Hertz [8,49-50]. Furthermore, vocoder experience suggests that 16 to 23 channels are sufficient to achieve the necessary spectral resolution and frequency coverage [16,51].

2.4 Word Recognition Algorithms

After preprocessing, the speech is represented by a time sequence of feature vectors. It is evident that some form of comparison to previously obtained representations for the words in the recognition vocabulary must occur. However, even in the most carefully pronounced repetitions of a word by one speaker the feature values and the number of features are likely to be different. This section describes two methods for handling this variance.

In template matching, a set of templates (i.e., feature patterns) are derived which are to be compared to the features of an unidentified utterance T (the unidentified utterance is represented by the variable T to indicate it is to be tested) via a distance measure. A distance measure is related to the similarity between two patterns - the smaller the distance, the greater the similarity. The Euclidean distance is probably the most well known measure, and is well suited to Euclidean space. Other spaces require different distance measures. Recapping, a distance measure is some mathematical relationship between the features of the unidentified utterance and those of the templates which is a minimum when the features of the unidentified utterance T are compared to the template of the correct identity of T . Note that there is no intermediate process of analysis or

classification of the word, but only the derivation of a measure which enables a final word choice. This is an essential difference from class matching, which follows.

The classes of a word are characterizing attributes of the word. For example, the recognition that a given part of a word is a vowel created with the tongue and jaw in a relatively high position could allow to classify that part of the word as having that class. A given word then has a set of classes common to all recognizable correct pronunciations of the word. The elements of the classes which are common to all words under consideration (i.e., in the vocabulary) carry no discriminatory information and may be ignored. There are classes, however, which represent the differences between words. Once a complete set of discriminatory classes for each word in a vocabulary has been determined from the measured data (i.e., the features), then the recognition and classification of the words is reduced to a simple matching process, or table look-up scheme. This is the basis of the name "class matching". While classes may be attributes of an entire word, they can also be distributed in a word in such a manner as to allow segmentation of the word. A word segment is defined as a portion of a word where a uniform set of classes prevail (the set may also consist of only one class). Literate english speakers, for example,

have no problem segmenting the word 'cab' into 'c', 'a', and 'b' segments. In other words a class is, at least, a partial classification which can be extracted from the features of the speech data which allows to categorize the utterance T to some extent.

The example below illustrates the difference between class and template analysis.

Assume that the test word to be recognized is a member of the vocabulary set {'i', 'a', 'o'}. Furthermore, assume that each word is composed of only one class and, when sampled, yields only one feature vector. Let the energy content of the words in progressively higher center frequency bandpass regions correspond to the features F(1), F(2), and F(3), respectively, which make up the feature vector. Thus, for this example, the preprocessing stage of Figure 2.1 outputs three values, FT(1), FT(2), and FT(3), which are a function of the test word. Figure 2.9 illustrates the two analysis procedures within this context.

The templates (i.e., representative feature patterns) for the words 'i', 'a', and 'o' can therefore be expressed in

(a)

WORD	w	TEMPLATE	DISTANCE MEASURE	ALGORITHM
'i'	1	$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$	$D(w) = \sum Fw(i) - FT(i) $	min $[D(w)]$ {w}
'a'	2	$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$		
'o'	3	$\begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$		

(b)

WORD	CLASS	ALGORITHM
'i'	front vowel	<pre> graph TD START[START] --> DV1{front vowel} DV1 --> i[i] DV1 --> DV2{middle vowel} DV2 --> a[a] DV2 --> DV3{back vowel} DV3 --> o[o] DV3 --> NoWord[No Word] </pre>
'a'	middle vowel	
'o'	back vowel	

FIGURE 2.9 Examples of two automatic word recognition algorithms;
(a) Template matching, (b) class matching.

the form of vectors such as

$$T_w = \begin{bmatrix} F_w(1) \\ F_w(2) \\ F_w(3) \end{bmatrix} \quad (2.2)$$

where $T(w)$ is the template of the w word in the vocabulary and $P(i, w)$ is the value of the i feature of the w word. Let the templates of 'i', 'a', and 'o' be

$$T_1 = (1, 0, 1)^t \quad T_2 = (1, 1, 1)^t \quad T_3 = (2, 0, 0)^t \quad (2.3)$$

respectively. A good distance measure is given by the expression

$$D(w) = \sum_{i=1}^3 |F_w(i) - F_T(i)| \quad (2.4)$$

where $D(w)$ is the Chebyshev distance between the w template and the test word feature pattern. The template matching algorithm decides that the utterance T is the word corresponding to the template giving the minimum distance $D(w)$ ($w = 1, 2, 3$), as long as this number is under a certain threshold. It is easily seen that the choice of distance measure is dependent on the features chosen.

In class matching, a set of classes are derived for the vocabulary which enable the comparison of the classes of T with the classes of all the words in the vocabulary. For a vocabulary word to contain a particular class, the features of this word must have certain properties. That is, the different classes correspond to different regions within the feature space.

Various pattern recognition methods can be applied to determine the class membership of a particular feature vector: the nearest neighbor decision rule, discriminant analysis, the maximum likelihood decision rule, etc.. Several of these methods can be expressed in terms of weighted sums of features, hence dividing the feature space into subspaces (i.e., classes) with a set of hyperplanes. For example, the 'front vowel' class corresponds to the property that the weighted sum

$$G1(FT(1)) = .25 FT(1) - FT(2) + FT(3) - 1.25 \quad (2.5)$$

is greater than or equal to zero. Similarly the classes 'middle vowel' and 'back vowel' correspond to

$$G2(FT(1)) = .25 FT(1) + 2 FT(2) + FT(3) - 3.25 > 0 \quad (2.6)$$

and

$$G3(FT(1)) = FT(1) + 0 FT(2) + FT(3) - 2 > 0 \quad (2.7)$$

respectively. Let the classes corresponding to the words 'i', 'a' and 'o' be the 'front vowel', 'middle vowel', and 'back vowel' classes, respectively. The test word is identified by finding the vocabulary word which has matching classes. Thus if T has the property of the 'front vowel' class as, can be determined by using Eq. 2.5, the unknown utterance must be 'i'.

Once a set of classes have been determined for a test word, they must be compared to those of the words in the reference vocabulary. The table look-up procedure required can

be formulated using a distance measure:

$$D(w) = -Gw(FT(i)) \quad (2.8)$$

and is usually performed as such. The conceptual difference between class and template matching lies in the category assignment process of class matching, which drastically reduces the amount of data to be processed to obtain a recognition decision. This gain is obtained at the expense of a large amount of preprocessing during a preliminary training stage to derive the appropriate distance measure.

Classes need not necessarily be mutually exclusive. However, the use of mutually exclusive classes can lead to simpler word classification schemes since only one class at a time need be processed. As is the case for the distance measure in template matching, the choice of classes is dependent on the features chosen to represent a word.

2.5 Summary of Past AWR Work

Generally, template matching as applied to AWR involves the use of a set of features such as mentioned in Section 2.3, a distance measure, and a time normalization procedure. Time normalization is required because, unlike the example in the preceding section, words generally require more than one time sample of the features for characterization. This implies that

the templates are usually multidimensional matrices rather than vectors. Time normalization is then required to ensure that the events in a test utterance's features are properly aligned in time with the corresponding events of the appropriate template. This must be done since the same word spoken at different times is spoken at different rates and therefore the template corresponding to a test word is generally not of the same dimension as the test word's feature pattern.

The choice of distance measure can significantly affect recognition accuracy in template matching. For example, it has been demonstrated that other distance measures than the log likelihood ratio for the LPC coefficients drastically reduce their recognition performance [35]. PARCOR coefficients using a Euclidean measure, cepstrum coefficients using a correlation measure, and multichannel smoothed log spectrum coefficients using the Chebyshev distance have yielded recognition accuracies better than 98 percent [20,52]. Nearly all template matching algorithms reported in the literature have been software based [7,53-60,62] due to the complexity of the time normalization procedure used. Notable exceptions have used a VLSI dynamic time warping processor [63-65,128], a digital speech processing peripheral (basically a TRW multiplier-accumulator with discrete component state controller)[66], which has been replaced by a

high speed digital signal processing microprocessor [141] in a more recent version [61], or highly simplified feature coding with linear time warping [67,68]. Table 2.2 describes the components required for two state of the art DTW-based speech recognition systems.

Class matching, on the other hand, generally uses a wider variety of features (zero crossing rate, RMS energy, LPC coefficients, etc...), a word segmentation procedure, and a word decision procedure which compares the labelled segments of the test utterance against the previously labelled reference vocabulary. Words are usually composed of more than one segment. When the segmentation approach is used the classes are usually composed of phonemes or phoneme groups, as described in Section 2.2. Typical class matching limited vocabulary systems isolated word AWR systems use 3 to 40 features and extract 3 to 42 classes. There are both software and hardware oriented versions. In many software versions [9,17,69] the classes are recognized by being the output from decision trees (such as the one shown in Figure 2.10 [9]), where the path down the tree is determined by comparison of the features using either fixed or self-adjusting thresholds. For many software systems with very small preselected vocabularies (ten digits), the path down the decision tree also yields a word decision (as shown in Figure 2.11 [69]).

TABLE 2.2 CHIP DESCRIPTION OF TWO COMMERCIAL WORD RECOGNITION SYSTEMS.

Texas Instruments		NEC	
Component	Comment	Component	Comment
Host Processor	with external I/O devices	Host Processor	With external I/O devices
TMS 320 microcomputer	2 44 mm 3 μ NMOS	NEC μ PD7720 DSP	2 28 mm 3 μ NMOS
Data RAM	16 K by 16 bit External	Data RAM	External
External Program RAM	Amount not specified	DTW chip	2 37 mm 2.5 μ NMOS
A/D	8 Khz	A/D

Figure 2.10 Becker and Poza's classification algorithm.

Figure 2.11 Rabiner and Sambur's decision tree for the digit recognition algorithm.

In the hardware equivalents of ~~these~~ the decision trees are replaced by various combinations of combinatorial, sequential, and threshold logic circuits [70-78]. More sophisticated AWR systems [10,69,79-92], after applying a set of rewriting rules on the class sequence for the purpose of time normalization and/or error correction either search for an exact match amongst the reference vocabulary or apply a distance measure between the class sequence of the test utterance and the reference vocabulary. Usually, the distance measure is used to implicitly perform the time normalization and error correction tasks [93-95]. For example, the distance measure can consider multiple class guesses for a word segment, thereby allowing for incorrect class decisions [91]. All of these sophisticated AWR systems were software oriented. Only one large vocabulary hardware class matching system has been well documented, that of Martin [96,97]. His approach, however was based on a very large number of discrete components (such as resistors, capacitors, and binary and threshold logic gates) and therefore requires considerable modification for integrated circuit design.

Table 2.3 describes the performance of several commercial automatic word recognition systems as tested by Doddington and Schalk of Texas Instruments [142-144] . In this evaluation a twenty word vocabulary was used consisting of the ten digits and

TABLE 2.3 EVALUATION OF SPEECH RECOGNIZERS

Manufacturer	Model	Nominal * Price (K\$)	T.I Error ** Rate (%)
Verbex	1800	65	0.2
Nippon Electric	DP-100	40	1.2
Nippon Electric	DP-200	21	1.2
Threshold Tech.	T-500	10	1.4
Interstate Elec.	VRM	1.5	2.9
Heuristics	7000	5	5.9
Centigram	MIKE 4725	4	7.1
Scott Instr.	VET/1	0.91	12.6

* Actual price may vary. Prices here are a general guide

**

No rejections allowed (i.e., substitutions errors only)

- ! The VET/1 is designed as a home-computer peripheral.
The price includes only the software and peripheral hardware.

ten command words: start, stop, yes, no, go, help, erase, rubout, repeat, and enter. For training, ten tokens of each word were collected.

Assuming that the class and template matching algorithms use the same feature set, insight into their relative implementation complexities can be obtained by examining the relative complexities of their feature analysis procedures (i.e., examining their distance measures and word decision processes).

2.6 Time normalization and the Distance Measure in Template and Class Matching

An over-simplistic view of time normalization in template matching would be to make all of the utterances of the same duration via a linear time transformation. However, differences in speaking rate affects speech events in a non-linear way, so that linear transformation is not sufficient [52,87,98]. White and Neely [52], for example, demonstrated that, while linear time alignment and non-linear time warping give comparable recognition accuracies for a monosyllabic vocabulary (alphabet and digits), non-linear time warping gave ten percent better accuracy for a polysyllabic vocabulary. This is a significant difference since it has been found that error rates worse than five percent [2]

cause user frustration and dissatisfaction.

The dynamic time warping algorithm is a non-linear time normalization procedure which has recently become popular [7,53-56]. The dynamic time warping algorithm calculates a distance measure between a test pattern and a reference pattern while performing the time warp. If it is assumed that the beginning and ending points of the test and reference words are known, then the problem of dynamic time warping can be formulated as a path finding problem over a finite grid, as shown in Figure 2.12. In template matching the test word can be written as a sequence of M p -dimensional feature vectors $FT(m)$ ($m=1,2,\dots,M$), where k corresponds to consecutive time samples and p is the number of features. Similarly the template can be written as a sequence of N p -dimensional vectors $FR(n)$ ($n=1,2,\dots,N$). Thus the test and reference words are represented by different dimension sequences.

In class matching the test word can be written as a sequence of M f -dimensional class vectors $CT(m)$ ($m=1,2,\dots,M$), where m corresponds to consecutive time samples and f is the number of different classes per time sample. Similarly a reference word class pattern can be written as a sequence of N f -dimensional class vectors $CR(n)$ ($n=1,2,\dots,N$). Similarly, the test and reference words can be written as a sequence of Q and L

Figure 2.12 Plot of the time relationship between a reference word and a test word.

f-dimensional segment vectors ST and SR, respectively. The dynamic time warp algorithm attempts to find an optimal path which minimizes a total distance function D_T of the form

$$D_T = \frac{\sum_{k=1}^K d(i(k), j(k)) W(k)}{N(W)} \quad (2.9)$$

for template, class, and segment matching respectively.

The function $d(i(k), j(k))$ is the local distance between frames $i(k)$ of the reference and $j(k)$ of the test, $W(k)$ is a weighting

function of the k th arc of the path, and $N(W)$ is a normalization factor which is a function of the weighting function W . Note that the time axes (m and n) have been expressed parametrically as functions of k , i.e.,

$$\begin{aligned} n &= j(k) & k &= 1, 2, \dots, K \\ m &= i(k) & k &= 1, 2, \dots, K \end{aligned} \quad (2.10)$$

where K is the length of the common time axis. The path which minimizes this distance results in a distance value which measures the degree of similarity between the test and reference words when their time scales are warped so as to make them as similar as possible. The dynamic time warping algorithm is a recursive method for minimizing Eq. 2.9 efficiently. This efficiency is achieved because it is assumed that the optimal path to a given grid point (n, m) only depends on earlier grid points and that the globally optimum path is also locally

optimum. Thus it is possible to define an accumulated distance

$D(n, m)$ in terms of an intermediate point (n', m') as

$$D(n, m) = \min_{\{m', n'\}} D(n', m') + \sum_{l=0}^{L-1} d(i(K'-1), j(K'-1)) W(K'-1) \quad (2.11)$$

where

$$\begin{aligned} i(K') &= n & j(K') &= m \\ i(K'-L) &= n' & j(K'-L) &= m' \end{aligned} \quad (2.12)$$

The minimized total distance function $D_T (= D_A(N, M)/N(W))$

is evaluated recursively using this relation [63].

For each reference word, the total distance can be calculated and the decision on the word spoken made on the basis of $\min_{\{\text{vocabulary}\}} D_T$. It is evident that the dynamic warping algorithm can be computationally demanding. Myers et al [53] reported several template matching dynamic time warping (DTW) algorithms having error rates between four and six percent. The average number of distance calculations for these algorithms varied from 492 to 781 per reference word. An approximation for the number of local distance calculations required to recognize a word using an efficient DTW algorithm [53] is

$$\frac{N}{D} = \frac{V \cdot N}{3} \quad (2.13)$$

where N is the average number of local distance calculations, N_D

is the average duration (in frames) of a word pattern. and V is

the number of vocabulary words. Moreover, Myers noted that the template matching DTW algorithm took from 50 to 90 percent of the overall computation time (depending on the number of vocabulary words). This implies that the complexity of the DTW template matching recognition systems is indicated by the complexity of the DTW algorithm.

The complexity of the DTW algorithm can be estimated by looking at the area requirements of DTW chips. The University of California's design [65] occupied 14.7 square millimeters (with test pads, 5 μ single poly NMOS), but used a large amount of external memory to simplify some of the functions. NEC's design [64] occupied 37 square millimeters (excluding test pads, 2.5 μ NMOS), a fairly large part of this being occupied by 8K bits of dynamic RAM with refresh.

In order to reduce the computational requirements of the DTW algorithm, Tappert and Das [99] suggest that the average duration in frames (i.e., N) of a word pattern be reduced as much as possible. Generally, due to the nature of speech, differences in adjacent (in time) features are small during sustained sounds. Figure 2.13 summarizes their results where only feature set samples significantly different from the last stored one are used. It is apparent that storing only about 45 percent of the

Figure 2.13 Effect of feature data reduction scheme on recognition accuracy.

time samples does not cause degradation in recognition performance. A 50 percent reduction in data can yield up to a factor of 4 saving in the time normalization computation. Considering classes, the difference can be non-existent, in which case the non-changing portion of the word defines a word segment. Thus the word pattern length (in frames) can be reduced by storing only those time samples (whether features or classes) which differ significantly from the last stored one. However the cost of calculating which samples to eliminate would then also have to be considered.

This data reduction approach can drastically simplify the DTW procedure as applied to class matching, especially when the classes are mutually exclusive and classification nearly error free (or if error correcting rules are applied before DTW). A description of the resulting segmentation approach to time normalization is shown in Figure 2.14. Every 5 to 20 milliseconds, a new set of feature values T becomes available. The class extraction stage calculates a sequence of class vectors $CT(1), CT(2), \dots, CT(i), CT(i+1), \dots, CT(M-1), CT(M)$ from the features. Depending on the details of the class extraction process, it may or may not be possible for consecutive classes to be the same (note that class extraction processes where consecutive classes are never the same can be referred to as

Figure 2.14 A block diagram of the sedimentation approach. The class sequence $CT(i-1), CT(i), CT(i+1)$ yields the sediment $ST(j)$.

segment extraction processes).

Assuming that in this particular class extraction process consecutive class decisions may be the same, then for each test word a class sequence (such as shown in Figure 2.15) is generated. Storing only those classes which differ from the preceeding class, the segment sequence ST1(1), ST4(2), ST²(3), ST3(4), ST4(5) is generated. ST^j(i) indicates the ith segment in the test word, and j indicates the particular class type corresponding to the segment. If it is assumed that a class must last at least two time frames before it can be labelled as a segment, then the corrected segment sequence is ST1(1), ST2(2), ST3(3), ST4(4). ST4(5) can be considered a glitch since it exists for too short a time (see Figure 2.15). It is this sort of rewriting rule which is referred to in Section 2.5. The dynamic warping of such short segment sequences is a much simpler task than normalizing much longer feature or class sequences since the number of local distance calculations required is reduced by approximately 98 percent (N in feature matching is typically 40 while for segment sequences it is about 6, thus the average number of local distance calculations is reduced by a factor of 1600/36).

In conclusion, then, it is seen that the time

CT1(1)CT1(2)CT1(3)CT1(4)CT4(5)CT1(6)CT2(7)CT2(8)CT2(9)CT3(10)CT3(11)CT3(12)CT4(13)CT4(14)

Figure 2.15 Features versus time for a word.

normalization procedure for template matching is much more demanding computationally than that of class matching with reliable segmentation. Thus, the remaining factor which affects recognition algorithm complexity is that of class extraction.

2.7 Class Extraction

2.7.1 Introduction

Different classes are generally identified by the existence of a high correlation between a set of particular feature values for each class. A correlation requires the implementation of a weighted sum such as $\sum x(i)y(i)$, where $x(i)$ and $y(i)$ are the two variables being correlated. If one of these variables is analog and the other is digital then the use of a multiplying digital to analog converter (MDAC) to perform the multiplication is likely to be most convenient [100].

The MDAC naturally implements a mixed analog-digital multiplication and provides an analog product. An MDAC can easily be implemented in a MOS process. A resettable integrator can then be used to implement the summation required by the correlation. Since the efficient means (from an integrated circuit viewpoint) of performing correlations shown in Figure 2.16 is available, a class matching algorithm which was correlation based was developed.

Figure 2.16 MDAC based correlation.

2.7.2 Class Extraction via Linear Discriminant Functions

In the appropriate feature space, given mutually exclusive classes, measurements of the features of these classes will cluster around specific locations, as shown in Figure 2.17(a). The class extraction problem is to derive a procedure which will determine the boundaries of a particular class. This can be done implicitly by using a distance measure, or explicitly by actually defining the boundaries. Figure 2.17(b) illustrates the concept. First, assume the case of explicit boundaries, as shown in Figure 2.17(b). Unlike the clusters shown in Figure 2.17(a), the clusters shown in figure 2.17(b) cannot be separated from each other using only one linear boundary. Rather, it requires two distinct linear boundaries to separate each cluster from every other individual cluster. Establishing the presence of a given class requires a choice between pair groupings for all possible pair groupings. For a set of N pairwise separable classes, there are $N(N-1)/2$ such groupings, of which $N-1$ will be required to make a class decision.

Now consider the implicit boundaries implied by the Mahalanobis distance [139]

$$(F - m(i))^T C(i)^{-1} (F - m(i)) \quad (2.14)$$

where each class i has its corresponding mean vector $m(i)$ and

Figure 2.17 Clusters in feature space; (a) each class is separable from rest by a single boundary, (b) classes are pairwise separable.

Figure 2.18 Class extraction algorithm decision sequence.

covariance vector $C(i)$. Feature vector FT is considered to be in the class i for which $d(i, FT)$ is a minimum. Therefore, establishing the presence of a given class requires the computation of Eq. 2.14 at most $N-1$ times [101] and the computation of the corresponding number of comparisons required to find the minimum distance. Note that when the covariance matrices for all of the classes are identical, the feature space is implicitly divided by linear boundaries.

Ease of computation is of prime consideration in this work. Therefore, unless there are overwhelming advantages in terms of recognition accuracy, techniques which require computation of higher order expressions should not be used. Since it has been demonstrated that hyperplanes can be used to divide the feature space into class-like regions [9,91,96], the use of linear discriminant functions [139] seems promising.

Linear discriminant functions are of the form

$$d(i, j, n) = \sum_{k=1}^P w(i, j, k) FT(k, n) \quad (2.15)$$

where $d(i, j, n)$ is the discriminant function for class pair $\{i, j\}$ at time n , and $w(i, j, k)$ is a weighting coefficient

which is being multiplied with the k^{th} feature at time n .

Discriminant functions have the property that if a sample test

feature vector $T(n)$ corresponds to class i , then

$$d(i,j,n) > 0 \text{ for all } i = j \quad (2.16)$$

This property can be used to test the class of feature vectors between class pairs. A succession of such operations, as indicated by the tree in Figure 2.18, completes the class assignment process for a feature vector. The process is reminiscent of successive approximation for analog to digital converters. Each junction in the tree corresponds to a calculation of $d(i,j,n)$. If $d(i,j,n)$ is greater than zero then the left index (i) of the next discriminant function calculation remains i (if $d(i,j,n)$ is less zero then the left index becomes j) and the right index becomes $j+1$. Thus at each junction $d(i,j,n)$ the next junction is either $d(i,j+1,n)$ or $d(j,j+1,n)$. After $N-1$ such comparisons, the indices of the discriminant function indicate the class decision for time n . Since each comparison requires $p+1$ multiplications, $(N-1)(p+1)$ multiply-accumulate operations are required for each class decision.

Note that the first decision (i.e., $d(1,2,n)$) is critical: a class 1 pattern incorrectly assigned to class 2 at the first junction in the tree can never be classified correctly. A similar problem is possible when the distance measure is used for classification, i.e., one of the trial classes can yield a

misleading distance. There are at least two ways in which this problem can be taken into account.

2.7.3 Character String Error Correction Rules

Note that the class sequence derived from a test word can be considered to be a character string, where each class corresponds to a different character. The application of heuristic and statistical error correction rules combined with dynamic programs using character string distance measures (Hamming, Levenshtein) has been successful in the more sophisticated word recognition systems [92,94,102,103]. Less sophisticated recognitions systems usually ignored the problem or else only used error correcting rules [9,17,69-78].

2.7.4 Distance Matrix in Class Matching

Rather than comparing the calculated classes to the reference classes directly, a log likelihood matrix is calculated for a test word in which each row corresponds to a sampled feature and each column corresponds to a mutually-exclusive class, as shown in Figure 2.19(a). The log likelihood, $l(j,k)$, can be evaluated using the relationship [85]

$$l(j,k) = -\log \left\{ \sum_{i=-N}^N A(j,i) r(i,k) \right\} \quad (2.17)$$

(a)

(b)

Figure 2.19 Distance matrices (a) log likelihood matrix
(b) binary distance matrix.

where $r(i,k)$ is the correlation coefficients of the test word, i is the correlation time delay, j is the class, k is the test word frame, and $A(i,j)$ is the correlation function of the linear prediction coefficients. A dynamic programming procedure can be used to find the maximum likelihood of the test word over a reference vocabulary composed of the order and typical duration of each class for each possible word [85,104]. A generalization of this procedure suggests itself. That is, a distance matrix can be obtained in a similar fashion using a distance measure. Furthermore, note that a binary distance matrix can be obtained if thresholds are applied to the distance measure.

2.7.5 Proposed Class Matching Algorithm

The DTW methods described so far have the disadvantage that they are computationally demanding because of the large number of possibilities which must be considered, even when dynamic programming techniques are used. However, unlike the case of template matching, it has not been established by anyone that complex procedures such as non-linear time warping are necessary in order to obtain good performance from a class matching word recognition system. Consider that, in general, a word is composed of noticeably fewer class segments than feature vectors. That is, while the average word can be represented by a sequence of approximately 6 phonemes [15], approximately 40

feature vectors are required [56]. Each word segment must, therefore, represent a proportionately much larger time period (8 times) than a feature vector. To start the DTW process for template matching, a linear time transformation can be done on the test word to make it contain the same number of feature vectors as the reference word. This is done to simplify the total distance normalization calculation. The remaining time non-linearity amounts to less than 37.5 milliseconds [56], especially in monosyllabic words [52]. Thus, if linear time normalization is performed on the feature sequence, the resulting class sequence should be much less affected by non-linear speaking rates than the feature sequence, therefore requiring much simpler manipulation than DTW to correct for time non-linearities.

For the above reason a word recognition algorithm based on the distance matrix but using linear time normalization was developed and considered for integrated circuit implementation. The algorithm is based on the fact that mutually exclusive classes can be treated as if they were not mutually exclusive. This can be done by letting each row of the distance matrix correspond to one of the elements from a well selected subset of the linear discriminant functions. Since each discriminant function yields a yes/no decision, a binary distance matrix can

be obtained (see Figure 2.19(b)) similar to that described above, but with much less computational effort. A binary matrix element is given by

$$BT(j,k) = \begin{cases} 1 & \text{if } d(j,k) > 0 \\ 0 & \text{if } d(j,k) < 0 \end{cases} \quad (2.18)$$

where $d(j,k)$ is the discriminant function for class j evaluated at time k . The redundant information in the matrix can be used to reduce the effect of classification errors. (Alternatively, the discriminant functions of non-mutually exclusive classes could serve in the same manner).

According to the maximum a posteriori criterion, the distance

$$D(w) = \sum_{j=1}^{ROW} \sum_{k=1}^{COL} d(BT(j,k), Bw(j,k)) \quad (2.19)$$

can be used to make a word decision. $BT(j,k)$ and $Bw(j,k)$ correspond to the bits at the j^{th} row and k^{th} column of the w^{th} reference word and the test word, respectively. ROW and COL are the number of rows and columns in the distance matrix, respectively, and $d(\cdot)$ is the Hamming distance. The test word then corresponds to the reference vocabulary word giving the minimum distance $D(w)$ (or equivalently, having the minimum number of bits different). Note that this class matching technique

manages to encode some of the transient speech information implicitly.

2.8 Conclusion

Feature extraction for both class and template matching recognition algorithms are basically identical processes. Filter bank coefficients are well suited for AWR since they have given good recognition results [20]. Moreover, they are amenable to sampled analog realization, which is one of the goals of this thesis.

Generally, recognition algorithms requiring the use of dynamic programming techniques have very high speed and memory requirements due to the large amount of data to be processed. In feature template matching, very little data reduction can be performed on the features to help alleviate this problem. On the other hand, class matching can be used to perform a significant amount of data reduction, thereby drastically simplifying the recognition process and allowing LSI implementation.

CHAPTER 3

DESCRIPTION OF AN INTEGRATED CIRCUIT FOR AWR

3.1 Introduction

The details of a proposed mixed analog-digital integrated circuit design for a class matching AWR system are described in this chapter. These details are dependent on the capabilities desired from the system, as well as the functional requirements imposed by the recognition algorithm. This proposal has resulted from extensive breadboarding and real time simulation, described in Chapter 4, using circuits which prototype those appropriate for a mixed sampled analog and digital design. The recognition algorithm has been refined by an interplay between the results of the prototyping and the system considerations described in Chapter 2.

A block diagram of the complete AWR system using the proposed recognition I.C. is shown in Figure 3.1. The host processor can be any general purpose microcomputer. It can be used for implementing tasks based on the output from the AWR chip, for retrieving reference vocabularies for particular tasks, and for performing some of the low data rate user controllable computations required by the recognition algorithm. For example,

Figure 3.1 Example of the use of the proposed AWR chip in a recognition system.

the user can use the host processor to perform threshold bounds on the recognition score depending on the desired level of error rejection.

The capabilities desired from the system aid in defining the AWR I.C.'s I/O pins via internal and external processing requirements. Obviously, several pins must be assigned for power supplies and clocking. An analog input for the test words and pins for the digital indication of the recognition results are also necessary. Since a sampled analog digital process is being assumed, the signal conditioning circuitry for the analog input can be on chip.

The design described here is to be capable of recognizing a word selected from a user specified menu, where the menus are defined during a vocabulary training stage. The maximum menu size and the number of menus is not a system limit. This implies that reference word storage is external to the I.C., which in turn implies that the appropriate address pins for this external memory, as well as input and output data pins must be provided. The write or read capability of these data pins is determined by the status of the user set training/recognition mode pin. Other pins can be provided to indicate the best correlation score for the vocabulary, which the host processor can use to make the final word decision under user programmable control. Finally,

due to the nature of the word detection process, an output pin for indicating that the I.C. is ready for word input is convenient.

Another major influence on the design process is given by knowledge of the required dynamic ranges necessary at each step in the AWR algorithm. While no attempt was made to find the relationship between dynamic ranges and recognition accuracy, lower bounds on dynamic ranges are implied by those used during the real time simulation of the recognition algorithm that was performed by the author if better recognition performance is desired. This will be described in Chapter 4.

Keeping these guidelines in mind, the description of the internal details of the AWR I.C. is divided into four algorithmic stages, as shown in Figure 3.2. First is the preprocessing and class extraction stage. Preprocessing is required to extract the feature set from which the class decisions are derived. Due to the nature of the computations involved (filtering and correlation) switched capacitor techniques are well suited for the implementation of this stage. The second stage is that of word detection and time normalization, i.e., calculating the test word's binary distance matrix and storing it in internal memory. During the recognition mode of operation, the binary distance

Figure 3.2 Algorithmic stages of AWR chip.

Figure 3.3 Block diagram of feature and class extraction stages.

matrix is then processed by the word decision stage to calculate the recognition scores of the most likely reference vocabulary words. The time normalization and word decision stages require a relatively irregular set of functions, so that a flexible architecture which can be programmed for different functions must be considered. Moreover, since it is envisioned that further refinement of the AWR algorithm is possible (for example, the word detection algorithm is a bit cumbersome) and since nearly all the decisions required are sequential in nature, a digital stored program architecture is suitable for the implementation of these stages. The fourth and last stage, the vocabulary training process, requires the interaction of the same digital and analog subsystems as during the recognition phase, but under the control of a different host processor program. During the vocabulary training mode, rather than storing the binary matrix for a word in internal RAM, the AWR chip channels it to external memory so that the host processor can perform whatever type of reference word averaging the user prefers.

As opposed to being a final product description, the circuits described in this chapter are intended to indicate feasibility. Therefore, some of the less critical subsystems (in terms of the possibility of being able to perform the required functions) need not be described in detail (i.e., the

digital processor).

3.2 Preprocessing and Class Extraction

3.2.1 Introduction

The major function of the preprocessor is to extract the speech features. As discussed in Section 2.3.4, a sixteen channel filter bank can be used. The class extraction circuitry must compute 16 correlations every 15 milliseconds. A block diagram of these two functions is shown in Figure 3.3. Dynamic ranges of 50 dB (including dc) for the correlation, and 42 dB (including dc) at the peak-picking outputs are suitable.

There are several sources of noise in switched capacitor circuits, most of which are not impediments towards meeting these specifications. The thermal noise in the switches and the transistors' $1/f$ noise are easily kept well below 50 dB [6]. Signal offsets due to parasitic charge injection and op amp offset voltage [43] are the major factors limiting the performance of SC circuitry for the recognition application since sufficiently large voltage offsets in the features can affect the results of the weighted sum and comparisons required in preprocessing and class extraction. Therefore this chapter will go into some detail on the proposed solutions to this problem. These offsets can be significantly reduced by using the proper

clocks and circuit layout [25,105,106,108].

Due to the sampling inherent in SC circuits, aliasing is also a potential source of noise. To avoid this problem the input can be sampled at a much higher frequency than its bandwidth, or an analog lowpass filter must be used to remove the unwanted frequencies. Unfortunately, the lower the cutoff frequency the greater is the area required for implementation of the filter's passive components. High sampling rates induce a similar problem in SC filters since the ratio of cutoff to sampling frequency of these filters is proportional to the ratio in the size of the capacitors which must be switched [109] and therefore more area must be devoted to capacitors. Thus the sampling frequency of SC filters must be minimized, keeping in mind the effect of a lower sampling frequency on the antialiasing filter.

3.2.2 Sound to Electrical Signal

The noise background in which an AWR system must operate is a primary factor that affects performance. There are several ways to reduce environmental noise, such as noise cancelling microphones [96], sound booths, and adaptive noise cancellation [110,111]. For this work, a Telex electret noise cancelling microphone was used, providing a 50 to 10 KHz frequency range and

at least 60 dB dynamic range in a room with background noise at 65 dB(B). The input signal to the AWR system is therefore of good quality even in a moderately noisy environment.

3.2.3 Antialiasing Requirements

In order to prevent aliasing during waveform sampling it is necessary to ensure that no significant frequency content exists in the signal above one half the sampling frequency. Since speech is naturally bandlimited (40 dB down at 8 KHz [112]) and since microphones can be mechanically adjusted for frequency response [113], the requirements for an antialiasing filter are not critical. Thus a lowpass filter with a cutoff frequency of 8 KHz operating at a much higher sampling frequency than the circuitry sampling its output (i.e., decimation [114-116]) is sufficient

3.2.4 Amplitude Normalization Via Automatic Gain Control

During normal speech, amplitude levels may vary over a wide range. Since the feature values are a function of the input amplitude level, some means must be provided to obtain a normalized output amplitude over the range of the input level. This can be accomplished by the use of an automatic gain control circuit. Another reason for which compression may be useful has been suggested by Nierderjohn [117]. Compression increases the

power of consonants relative to vowels. Since consonants are of prime importance as cues to speech recognition [96,111] dynamic range compression of speech then allows an improved intelligibility of speech in noise.

Figure 3.4 shows a switched capacitor circuit which can be used to perform the automatic gain control function. In order to achieve two to one dynamic range compression, the gain of the circuit must be proportional to the average output amplitude. A ramp type analog to digital conversion determines the counter output for this condition to be true. By using combinatorial logic to put a lower bound on the counter output, the circuit gain can be designed such that small input signals have unity gain and large signals have reduced gains. More specifically, it can be shown that when the ratio of the input capacitance, C_{in} , to the total feedback capacitance, $C_{AGC,f}$, is equal to

$$\frac{C_{in}}{C_{AGC,f}} = \frac{C}{2^N} \quad (3.1)$$

then the circuit gain G is given by

$$G = C / \text{COUNT} \quad (3.2)$$

where C is the lower bound on the counter output, COUNT is the actual counter output, and N is the number of bits used for the count. Although the AGC circuit was not used to obtain the

Figure 3.4 Automatic gain control circuit.

experimental results described in Chapter 4, it was breadboarded and the above equations were validated.

3.2.5 Feature Extraction

Voltages corresponding to the energy levels in different frequency bands are the desired features to be supplied to the class extraction circuitry. Rather than obtaining true rms levels, it is much easier to obtain the peak values of the
 * bandpass filter outputs.

Table 3.1 shows the center frequencies and quality factors of the second order bandpass filters which can be used to evaluate the different features as per the reasoning described in Section 2.3.4. The schematic of a second order SC bandpass filter which can be used to implement these filters is shown in Figure 3.5, which shows a Tow-Thomas biquad filter [118]. The filter output can be made operational amplifier offset free (i.e., d.c. offset reduced by more than 45 dB [122]) by appropriately sampling the op amp voltage offsets on one clock phase so that they cancel themselves out on the next [119]. The Tow-Thomas biquad has the property that its gain, quality factor and center frequency can be adjusted. The filter's center frequency is approximately proportional to a , its bandwidth to a and its gain to a (where the different a 's correspond to K Q

TABLE 3.1 CENTER FREQUENCIES AND QUALITY FACTORS FOR FEATURE FILTERS

CHANNEL NUMBER	CENTER FREQ. (Hertz)	QUALITY FACTOR	CHANNEL NUMBER	CENTER FREQ. (Hertz)	QUALITY FACTOR
1	200	2	9	1270	4.32
2	300	3	10	1600	4.32
3	400	4	11	2016	4.32
4	500	5	12	2540	4.32
5	600	6	13	3200	4.32
6	700	7	14	4032	4.32
7	800	8	15	5080	4.32
8	1008	4.32	16	6400	4.32

Figure 3.5 Offset-free continuous output bandpass filter.

normalized capacitor sizes as seen in Figure 3.5). The spectra of these filters are shown in Figure 3.6.

The circuit shown in Figure 3.7 illustrates the concept used to perform the peak-extraction. A comparison is made between the present estimate of the peak value, which is stored on capacitor C, and the current input. If the current input is larger than the stored value, the switch turns on, thus updating the peak value. Periodically, the circuit must be reset. Figure 3.8(b) illustrates the output of such a peak-picking circuit, where the input (Figure 3.8(a)) originates from channel one of the filter bank. The word spoken by the author was "cap" and the silent portion of the plosive can be seen in the center of the figure. It is evident from this figure that the reset frequency of the peak-extractor must be fractionally lower than the lowest expected speech pitch frequency (i.e., 60 Hertz [146]).

The voltages extracted from the peak-extractor for a filter channel must be supplied to the correlator for as many times as there are correlations to perform. If each feature came from a different sample and hold circuit, the varying voltage offsets from the sample and holds would reduce the correlation signal to noise ratio. In order to reduce this fixed pattern noise effect, the circuit shown in Figure 3.7 can be modified to

Figure 3.6 Frequency Response of the second order bandpass filters for feature extraction.

Figure 3.7 Peak-extraction circuit.

Figure 3.8 Input and output of peak-extraction circuit: (a) channel 1 bandpass filter output for the word "cap" (b) peak-extraction output. Horizontal scale is 7.4 milliseconds per centimeter.

the form shown in Figure 3.9(a). The use of a multiplexed op amp output helps to eliminate op amp voltage offset as a source of fixed pattern noise and the differential architecture is used to reduce the clock feedthrough effect [120].

The circuit operates using the clock sequence shown in Figure 3.9(b) for a two channel peak-extractor. On ϕ_r , the capacitors C_1 to C_4 are discharged. On ϕ_{21} high, the current peak value for channel 1 (stored on C_{s1}) is supplied to the comparator. The capacitors C_1 to C_4 are discharged again during ϕ_r . Also on ϕ_r , ϕ' is high and the present input and the past peak value are compared. Conditional updating (depending on the result of the comparison, ϕ_s) of the peak value is therefore performed during the next ϕ_1 clock. The same process is repeated for the other multiplexed channel by using the data from C_{s1} and C_{s2} in succession.

To prevent aliasing, the peak extraction update time for all channels is made inversely proportional to the bandpass filters' sampling frequency, f_s . Since some classes last for as short as 30 milliseconds [15], all class decisions must be made.

Figure 3.9 Combined peak-extraction and sample and hold for 2 channels: (a) circuit diagram (b) timing diagram.

within 15 milliseconds. Thus there is a conflict in the timing requirement of the peak extractor, i.e., to catch the correct peaks one must update the peak estimates continuously for 15 milliseconds, at the same time feeding the correlator the true peak values. This problem can be solved by cycling two multiplexed op amp peak extractors, such that when one of these is performing a peak update, the other is supplying features to the correlator at the required rate.

3.2.6 Class Extraction

It has been found experimentally (see Chapter 4) that the class extraction algorithm works when 16 correlations and comparisons are computed every 15 milliseconds. A block diagram of a simple system which can do these processes is shown in Figure 3.10. The speech features come from the multiplexed peak-extractor in a serial sampled analog stream. The multiplying digital-to-analog converter (MDAC) is used to multiply a feature vector ($FT(n)$) with a digital weight vector corresponding to the class being tested. A switched capacitor resettable integrator performs the correlation sum, which is compared to a threshold (analog ground - the test is for the sign of $d(i,j,n)$). The result of the comparison indicates a class decision, which is stored in RAM for further processing (Section 3.3).

Figure 3.10 Class extraction circuitry.

The rate at which the MDAC must perform its multiplications is determined by the number of features p , the number of classes N , and the time available T . Thus the MDAC must perform at least $(p+1)N/T$ multiplications per second. Therefore the serial sampled-analog stream to the MDAC from the peak extractor should be at least at this rate. For 16 classes and 17 features, this is not a limitation for the circuit shown in Figure 3.11(a).

The multiplication of a digital weight by an analog input is based on the principle of charge redistribution [121]. Complementary clocks are required. During the first phase (Figure 3.11(b)) (ϕ high), the top plate of the largest capacitor is switched to either the input voltage or analog ground, depending on the polarity of the digital weight. The top plate of all the other capacitors is connected to ground and their bottom plates, depending on the value of the weight, are switched to either the input or ground. This results in a stored charge on the top plates of all the capacitors of Q :

$$Q = CV(in) \left\{ \frac{a(1)}{2} - \sum_{i=2}^9 \frac{a(i)}{2} \right\} \quad (3.3)$$

where $a(i)$ indicates the particular bit pattern representing the

Figure 3.11 Offset free MDAC and comparator: (a) circuit
(b) clocks for MDAC and comparator.

digital weight, and $V(in)$ is the analog input voltage. During ϕ low the top plates of all the capacitors are connected together and floated, while all the bottom plates are tied to ground. Since the charge on the top plates is conserved, it redistributes itself amongst all the capacitors (including the integrator's SC resistor). On the next clock phase this charge is redistributed to the integrator capacitor. Thus the multiplication product V is given by

$$V = V(in) \left\{ \frac{a(1)}{2} - \sum_{i=2}^9 \frac{a(i)}{2} \right\} \{ 2^{10} / (2^{10} + 1) \} \quad (3.4)$$

After repeating this process $p+1$ times with the appropriate weights, a correlation is stored on the integrating capacitor. Note that due to the action of capacitor C_I the correlation is op amp offset free [122]. Furthermore, since the correlation result is checked for sign only, absolute gain accuracy is not important for the summation and small capacitors can be used for the integrator.

The circuit described for performing the peak extraction and correlation processes is very similar to a programmable transversal filter developed by Bell Northern Research [120] and should therefore have similar specifications. The peak extraction circuit, of course, requires a novel adaptation of the

2 2
OF / DE

sample and holds used in BNR's filter. BNR's 31-tap filter achieves a dynamic range greater than 70 dB with 9 bits coefficient accuracy, and the die size is less than 11.2 square millimeters (including clock generation and interface synchronization logic).

In addition to the area required for the peak extractor and correlator, ROM space of 2.5 Kbits ($=9N(p+1)$) is required for the digital weights.

3.3 Word Detection and Decision

The basis for the AWR algorithm was described in Section 2.7. Word recognition is performed by processing the class extraction results so as to produce a binary distance matrix for a test word which can be used to compute the closest match to the reference vocabulary via Eq. 2.19, i.e.,

$$D(w) = \sum_{j=1}^N \sum_{m=1}^M d(BT(j,m), Bw(j,m)) \quad (3.5)$$

$BT(j,m)$ and $Bw(j,m)$ correspond to the bits at the j^{th} row and m^{th} column of the binary distance matrices of the w^{th} reference word and the test word, respectively. N is the number of classes and M is the number of time samples the distance matrix, and $d(,)$ is the Hamming distance.

Figure 3.12 illustrates the algorithm in more detail.

After class extraction, a test word is represented in RAM as an array of N class decisions by L time samples, where L varies for each utterance. Thus a maximum number of time samples must be allocated for word storage in order to minimize chip RAM area. Time linearization is used to convert the stored word to a binary distance matrix occupying N by M bits in RAM. In the real time simulation described in Chapter 4, N is 16, M is 15, and the maximum value for L is 40. The reference templates are also N by M bits. The time linearization is performed by choosing M equally spaced samples in time from the stored class decisions by using the relationship

$$\text{ADDRESS}(I) = \text{START} + \frac{L \cdot I}{M+1} \quad \text{for } I = 1 \text{ to } M \quad (3.6)$$

where START corresponds to the address of the first time sample of the test word, and ADDRESS corresponds to the addresses of the time linearized samples. Word correlation consists of computing Eq. 3.5 for each reference word. The results of the word correlation can then be used by the host processor to make a word decision or indicate that a retrieval is required.

In the real time simulation of the AWR system, 65% of the total word decision and detection computation time (about 4 msec. per reference word) was required for calculating Eq. 3.5.

Figure 3.12 Word detection and decision algorithm flowchart.

This large proportion was required because an instruction set unsuited to the application had to be used, i.e., MN shift lefts and add carry to register. Figure 3.13 illustrates the program segment used to calculate Eq. 3.5. If desired, custom design of the digital processor could easily reduce the time taken for this operation by converting the shift and add carry instructions into a single higher speed instruction.

3.4 Reference Pattern Training

Since reference pattern training does not require real time response, and since there is more than one way of performing this function, the host processor can be used to perform the required data processing. The task of the AWR chip during the training phase is then to output the time normalized binary matrices, along with the appropriate synchronization pulses and addresses, to offchip RAM (rather than onchip). For use during the real time simulation, the reference training process performed, for each vocabulary word, a majority vote decision for each bit position of the binary distance matrices stored in RAM.

3.5 Conclusion

Figure 3.14 illustrates the AWR chip in terms of the subsystem circuits described in this chapter. The estimated area

```

CLR D(1)           ;Initialize D(1)
.REPT M             ;DO until .ENDR{M} M times
    MOV    (R4)+,R3 ;Move the next reference
                  ; word class decision vector
                  ; into register R3
    MOV    @(R1)+,R2 ;Move the next normalized
                  ; test word decision vector
                  ; into register R2
    XOR    R2,R3     ;Store the bit by bit
                  ; exclusive or of these
                  ; registers into R3
.REPT N             ;DO until .ENDR{N} N times
                  ;Count the number of bits
                  ; which are set by
    ASL    R3        ; shifting the MSB of R3
                  ; into the carry bit and
    ADC    D(1)      ; adding the carry bit to D(1)
.ENDR{N}
.ENDR{M}

```

FIGURE 3.13 Program segment used to calculate D(w).

TABLE 3.2 AREA DISTRIBUTION OF PROPOSED AWR CHIP

Subsystem	Comments	Area - mm ²
AGC		4
Filter Bank	16 fourth order	<20 [145]
Peak Extractor, Correlator, and Comparator	Excluding weight ROM	<11 [120]
Weight ROM	2.5 Kbits	1 [67]
Data RAM	1 Kbits, includes 24 sixteen bit registers for the digital processor	10 [67]
Digital Processor	Includes interconnection requirements of entire chip	>10 [67]

Figure 3.14 Block diagram of subsystems of AWR chip.

for such a chip (assuming a 5 μ channel CMOS process), which is capable of speaker dependent isolated word vocabulary recognition, is 54 square millimeters. Table 3.2 gives the area distribution. These are estimates intended to give an indication of the probable maximum areas involved. Note that significant reduction in the total area may be possible. Rather than implementing the filter bank as fourth order filters in parallel, much area could be saved by multiplexing, if noise performance could be guaranteed. This approach appears very promising since another highly multiplexed filter (shown in Figure 2.8) had a 50 dB dynamic range [106].

The mixed analog digital architecture described in this work can be compared to all-digital architectures. All-analog architectures are not well suited to the type of sequential operations required by the word decision algorithm and are therefore not discussed further. Due to the wide variety in the type of functions required by the binary matrix algorithm, and considering the number of multiplications required for digital feature extraction, it is reasonable to expect that a custom digital design for implementing the algorithm would be similar to that of a general purpose digital signal processor such as Texas Instruments' TMS 320 [141]. However, it can be shown that the TMS 320 is not sufficiently fast to implement the binary matrix

algorithm in real time just by considering the instruction rate required for the sixteen fourth order bandpass filters required for feature extraction. From the z-transform of a second order Tow-Thomas bandpass filter, it is evident that a fourth order bandpass filter can be implemented using eight multiplications and eight additions for each output time sample (when implemented as a cascade of two second order filters). Therefore for all sixteen fourth order bandpass filters, 256 arithmetic operations must be performed within the signal sampling period (50 usec.). Thus a minimum of 5.12 million arithmetic operations per second are required, not including data input/output time. The TMS 320 can only process 5 million arithmetic operations per second and therefore is not suitable for real time implementation of the binary matrix algorithm. The Fujitsu digital signal processor [147], however, is fast enough to perform the entire AWR algorithm in real time, but at the expense of silicon area -

2
91 mm on a 2.3 μ silicon gate process.

Although hitherto the primary emphasis of VLSI technology has been on the optimization of digital circuits, work is being performed to shrink analog circuits to VLSI dimensions [129-132]. Again, if noise performance can be guaranteed, further area reduction is possible, making the proposed mixed analog digital AWR chip even more competitive with the all digital VLSI chips

being used for speech recognition.

CHAPTER 4

EXPERIMENTAL RESULTS OF THE REAL TIME SIMULATION OF THE AWR SYSTEM

4.1 Introduction

This chapter is divided into three sections. First, a description of the experimental test set up used to perform the real time simulation of the binary matrix class matching word recognition algorithm is given. The weight training procedure is described in the second section. Finally, the third section describes the behavior of the binary matrix word recognition algorithm.

Since the experimental test set up was used as a developmental tool, it needs to be much more adaptable than the proposed AWR chip design. The reasons are:

- i) hardware changes were indicated during development
- ii) the experimenter wants to be able to tailor the system to more than one application. For example, the training program needed to allow correlation coefficients (weight vectors) to be ascribed to the various classes for the particular speaker, speakers, or vocabulary should be run on the same system that is used for reference pattern training. Both these reasons imply

that it is useful during training and future development to have a readily adaptable experimental set-up.

In class matching speech recognition systems, there are two levels of training. The first level preprocesses speech to extract information which enables class decision. This level of training needs to be done only once and is basically user transparent in a completed recognition system. In particular, for the binary matrix word recognition algorithm, a set of classes and their corresponding weight vectors must be chosen appropriate to the system requirements (i.e., speaker dependent or independent, a preselected or unspecified vocabulary). The next level of training is that which generates the reference word patterns. In this chapter the emphasis is on weight vector training, although the effect of reference averaging is also presented.

Recognition accuracy is affected by several factors: vocabulary size, reference matrix averaging, distance thresholding, and time warping technique. The effect of some of these factors on the recognition accuracy helped to determine the makeup of the proposed recognition algorithm and are therefore presented in this chapter.

4.2 Experimental Test Set Up

Figure 4.1 shows the experimental test set following the philosophy of Chapter 3 for the AWR system, plus the data acquisition systems required by the weight vector and reference pattern training procedures.

Feature extraction is performed by bandpass RC active filtering, followed by peak extraction using the circuit shown in Figure 3.7. A multiplying digital to analog converter and switched capacitor resettable integrator are used to perform the series of correlations necessary for the class extraction algorithm. The output of the comparator monitoring the correlation is used by the state controller to make a nonsilence detection decision. During weight vector training, the non-silence decision initiates sampling and analog to digital conversion of the features, which are sampled by a minicomputer (PDP 11/03) under the control of a microprocessor (8085). During reference pattern training and word recognition, the non-silence decision initiates minicomputer sampling of the class decisions, under the control of a counter-ROM based state controller. The minicomputer contains the programs which process the incoming data. Thus, during the real time simulation of the AWR chip and during reference pattern training, the minicomputer performs the function of both the host processor and the AWR chip's word

Figure 4.1 Experimental test set

detection and decision logic.

Table 4.1 provides a list of the dynamic ranges at dc obtained at two stages in the test set up. Table 4.2 contains the results of a non real time analysis of the digits vocabulary performed on the mainframe computer (CP6) after data acquisition by the test set up. The table shows the effect of weight vector coefficient quantization and correlator output dynamic range on binary matrix bit decision uncertainty. The data in these tables can serve as guidelines during circuit development.

4.3 Weight Vector Training and Classes

In this work, a speaker dependent set of weight vectors was derived due to limited computing resources and the logistics involved in acquiring data from many speakers. This is not an inherent system limitation. In order to compete with existing commercial systems, the trainable vocabulary option was chosen.

The first step in the preparation of weight vectors was to determine which classes could be extracted from the feature set described in Section 2.2. The elements from the set of phonemes were chosen as the candidate for the classes. Since the feature set is not necessarily complete it is expected that a varying degree of success will be obtained for each class. Once

TABLE 4.1 DC DYNAMIC RANGE (MEASURED VALUES)

At Output of	Dynamic Range (dB)
Feature	42
Correlation	50

TABLE 4.2 EFFECT OF WEIGHT QUANTIZATION AND CORRELATION
DYNAMIC RANGE ON BINARY CLASS DECISION MATRIX

Weight Coefficient Quantization (Bits)	Correlation Dynamic Range (dB)	Fraction of Bit Decisions Uncertain (%)
8	36	14.2
8	42	6.6
8	48	3.1
8	54	1.6
8	60	0.8
9	36	13.0
9	42	6.3
9	48	3.0
9	54	1.5
9	60	0.8

the set of classes is chosen, the weight vectors can be programmed into the speech recognition hardware to determine how well the recognition algorithm performs on feature data which is not the same as that used in the training stage.

The weight vector training vocabulary was composed of consonant/vowel/consonant sequences whose elements were chosen from the set of classes shown in Table 4.3. The resulting training words are shown in Table 4.4. To simplify the weight training procedure, most of the words are synthetic (i.e., no English meaning). The training procedure consists of two steps. First, representative feature vectors (a total of 1769) for each phoneme class were carefully selected from the training vocabulary. A stepwise discriminant analysis program from the BMDP software package [124] was available which could be used to perform discriminant analysis between all pair combinations (a total of 210) of the classes. The program attempts minimum error-rate classification via the use of the discriminant function

$$G(i) = (C^{-1} m(i))' F T - 1/2 m(i)' C^{-1} m(i) + P(\text{class } i) \quad (4.1)$$

That is, the program assumes that all the class densities are normally distributed and with equal covariance matrices (this is equivalent to using the Mahalanobis distance, as described in Section 2.7.2). In general, however, the phoneme classes do not

TABLE 4.3 CLASSES AND THEIR PRONOUNCIATION

Class Symbol	Nearest English Equivalent	Class Symbol	Nearest English Equivalent
EE	Eve	T	Top
AY	hAte	S	Sun
A	Ask	F	Fast
AW	fAther	M	Man
I	It	N	Noon
O	Obey	Z	Zoo
OO	bOOt	SH	SHe
U	Up	TH	THin
B	Bad	V	Voice
D	baD	VB	VOICEBAR
P	PoP	SI	SILENCE

TABLE 4.4 LIST OF SYNTHETIC WORDS USED TO DERIVE THE WEIGHT VECTORS

BAB	DAD		VAV
BIB	DID	...	VIV
BAWB	DAWD	...	VAWV
BAYB	DAYD	...	VAYV
BEEB	DEED	...	VEEV
BOB	DOD	...	VOV
BOOB	DOOD	...	VOOV
BUB	DUD	...	VUV

have equal covariance matrices. Nonetheless, the program was still used due to its convenience. Unfortunately, the package was not written so as to enable file storage of the weight vectors between pairs of classes. Instead, it was more convenient to calculate the discriminant functions between larger groups of classes. The resulting confusion matrix for the classes of Table 4.3 is shown in Table 4.5. Not too surprisingly, some classes performed much better than others.

An example of the implication of inaccurate classification is shown for the word 'seven' in Figure 4.2. The points in the figure were derived using the classification technique described in Section 2.7 and the discriminant functions corresponding to the classes of Table 4.3. For comparison, the phonetically correct representation for 'seven' is [S], [AY], [V], [N]. One way to obtain accurate word classification despite the incorrect phoneme classifications would be to use the complex error correcting methods described in [102,125]. Starting from phoneme error rates of 61 and 77 percent, word recognition rates of 93 and 98 percent were obtained, respectively. One could also replace or add some features. For example, the confusion amongst the voiced and unvoiced plosives could be reduced by the introduction of the 'voice onset time' feature [49]. A much easier way to handle classification errors is to combine some of

Figure 4.2 Spectrograph and classification results for the word "seven".

TABLE 4.5 CONFUSION MATRIX FOR TWENTY-TWO CLASSES

CLASS	PERCENT CORRECT	NUMBER OF SAMPLES CLASSIFIED INTO EACH CLASS																					
		SI	B	D	P	T	A	I	AW	AY	EE	O	OO	U	VB	S	F	M	N	Z	SH	TH	V
SI	100	87	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
B	58	0	15	1	0	0	0	0	0	0	0	0	0	0	3	0	0	1	5	0	0	1	
D	81	2	2	29	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	
P	73	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	3	2	3	0	0	4	
T	66	0	0	8	6	42	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	5	
A	77	0	0	0	0	0	56	0	3	2	0	0	0	12	0	0	0	0	0	0	0	0	
I	71	0	0	0	0	0	0	49	0	15	5	0	0	0	0	0	0	0	0	0	0	0	
AW	79	0	0	0	0	0	2	0	69	0	0	5	0	11	0	0	0	0	0	0	0	0	
AY	79	0	0	0	0	0	0	14	0	60	1	0	0	0	0	0	0	0	0	0	0	0	
EE	94	0	0	0	0	0	0	1	0	1	73	0	0	0	0	0	0	0	0	0	0	3	
O	76	0	0	0	0	0	0	0	8	0	0	59	0	11	0	0	0	0	0	0	0	0	
OO	85	0	1	0	0	0	0	1	0	1	1	2	58	0	0	0	0	3	1	0	0	0	
U	64	0	0	0	0	0	14	0	4	0	0	9	0	47	0	0	0	0	0	0	0	0	
VB	96	2	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0	0	
S	57	0	0	6	0	1	0	0	0	0	0	0	0	0	0	91	4	0	0	40	7	10	
F	77	0	0	10	11	2	0	0	0	0	0	0	0	0	0	0	138	0	0	0	15	3	
M	71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	48	14	0	0	6	
N	58	0	0	0	1	0	0	0	0	0	0	0	0	0	11	0	0	24	54	0	0	3	
Z	79	0	0	0	0	2	0	0	0	0	0	0	0	0	0	16	0	1	0	74	0	1	
SH	92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	2	97	0	
TH	59	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	16	1	1	0	0	32	
V	40	0	0	0	11	0	0	0	0	0	0	0	9	0	0	0	11	10	2	0	0	41	
TOTAL	74	91	18	54	68	47	72	65	84	79	80	75	67	81	59	114	174	90	80	116	104	81	70

TABLE 4.6 REDUCED CLASS SET

Class Symbol	Members
T	T
A	A AW O U
I	I EE OO AY
N	N VB
SZ	S Z
F	F TH V
SI	SI

TABLE 4.7 CONFUSION MATRIX FOR SEVEN CLASSES

CLASS	PERCENT CORRECT	NUMBER OF SAMPLES CLASSIFIED INTO CLASSES						
		SI	T	A	I	N	SZ	F
SI	100	87	0	0	0	0	0	0
T	94	0	60	0	0	1	0	3
A	98	0	0	307	5	0	0	0
I	97	0	1	2	281	5	0	2
N	96	2	0	0	0	134	0	4
SZ	89	0	14	0	0	0	226	13
F	88	4	9	0	9	20	0	294
TOTAL	94	93	84	309	295	160	226	316

the more easily confused classes together. The binary matrix approach can then be used to compensate for the effects of misclassification errors and the reduced number of classes.

Table 4.6 shows a reduced set of classes made in such a manner. The corresponding confusion matrix (Table 4.7) demonstrates much better behavior. The classification redundancy required by the binary matrix algorithm is easily obtained by letting the weight vectors be equal to the difference between pair combinations of the discriminant functions computed with the BMDP program. The weight vectors used to arrive at the word decision in the following section are listed in Table 4.8.

4.4 Recognition Accuracy

In accordance with the discussion on Chapter 2, the recognition algorithm divides the signal space into a set of V disjoint decision regions $\{I(w)\}$; any test word $BT(j,k)$ is assigned to $I(k)$ if

$$D(k) < D(i) \quad \text{for all } i \neq k \quad (4.2)$$

It is evident that the probability of error for the algorithm is a complex function of the vocabulary. Typical results obtained using real speech (i.e., the vocabularies shown in Table 4.9)

TABLE 4.8 WEIGHT COEFFICIENTS FOR CORRELATION CALCULATIONS

Class J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	Constant
1	.117	.120	.049	-.670	.334	-.226	.000	.173	.718	-.052	.000	-.401	-.184	.000	.268	-.123	-4.904
2	.200	.009	.380	-.809	.532	-.298	.000	-.406	1.649	-1.153	.000	1.541	-1.516	.000	.521	-.304	-32.025
3	.187	.049	.271	-.738	.557	-.358	.000	.158	.714	-.431	.000	.505	-.482	.000	.230	-.361	-25.090
4	.024	.196	.189	-.624	.436	-.112	.000	-.233	1.122	-.405	.000	.049	.103	.000	.526	-1.097	-2.954
5	.177	.099	.310	-.762	.523	-.343	.000	.065	.923	-.384	.000	.170	-.318	.000	.487	-.509	-26.760
6	.083	-.111	.331	-.139	.198	-.072	.000	-.579	.931	-1.101	.000	1.942	-1.332	.000	.253	.181	-27.121
7	.070	-.071	.222	-.068	.223	-.132	.000	-.015	-.004	-.379	.000	.906	-.298	.000	.038	-.238	-20.186
8	-.093	.076	.140	.046	.102	.114	.000	.406	.404	-.353	.000	.450	.287	.000	.258	-.974	-1.950
9	.060	-.021	.261	-.092	.189	-.117	.000	-.108	.205	-.332	.000	.571	-.134	.000	.219	-.386	-21.856
10	-.013	.040	-.109	.071	.025	-.060	.000	.564	.935	.722	.000	-1.036	1.034	.000	.291	-.057	6.935
11	-.176	.187	-.191	.185	-.096	.186	.000	.173	-.529	.748	.000	-1.492	1.619	.000	.005	-.793	29.071
12	-.023	.090	-.070	.047	-.009	-.045	.000	.471	-.726	.769	.000	-1.371	1.198	.000	.034	-.205	5.265
13	-.163	.147	-.082	.114	-.121	.246	.000	-.391	.408	.026	.000	-.456	.585	.000	.396	-.736	22.138
14	-.009	.050	.039	-.024	-.034	.015	.000	-.093	.209	.047	.000	-.335	.164	.000	.257	-.148	-1.670
15	.153	-.097	.121	-.138	.087	-.231	.000	.298	-.199	.021	.000	.121	-.421	.000	-.039	.588	-23.806

for one speaker follow.

4.4.1 Effect of Vocabulary Size and Complexity

The vocabularies and their corresponding recognition accuracies shown in Table 4.9 were obtained with no rejection decisions allowed, but each reference pattern being the average of three repetitions of the vocabulary words. Twenty-two repetitions of each vocabulary were used to obtain the recognition results. The effect of vocabulary complexity is reflected by the fact that the 10 word vocabulary gave inferior results compared to the larger 13 word vocabulary. Figure 4.3 shows the expected [126,127] constant increase in error rate corresponding to increasing vocabulary sizes. The smallest vocabulary used to make this plot was a subset of the next largest, and so on until the largest. The digits vocabulary was an exception to this procedure and therefore was not considered when drawing the line fitting the points. Moreover, the words in these vocabularies were chosen such that no two words were obviously too similar (generally, more than 8 percent of the bits were different between any two of the reference patterns). It also turned out that no two words were different by more than half the total number of bits. The results presented are for one speaker only. For different speakers, different curves such as described in [126,127] (i.e., having different slopes and

TABLE 4.9 VOCABULARIES AND THEIR RECOGNITION ACCURACIES

Vocabulary Size	10 Words	11 Words	13 Words	20 Words	35 Words
Vocabulary Words	One Two Three Four Five Six Seven Eight Nine Zero	Add Subtract Multiply Divide Save Core Replace Insert Store Delete List	Add Minus Times Divide Sine Cosine Tangent Root Equal Inverse Store Delete List	Add Subtract Multiply Point Save Make Core Replace Insert Store Delete List Number Skip Address Output Input Compare Byte Control	Add Subtract Multiply Divide Save Sine Core Cosine Tangent Replace Root Equal Insert Inverse Store Delete List Number Skip Address Control Output Input Compare Byte Erase Decimal Plus To Word Octal Minus Degree Radian Register
Recognition Accuracy	90%	94%	93%	91%	83%

* Recognition results are those obtained with no rejections allowed, and with only three averages of the reference templates.

Figure 4.3 Effect of vocabulary size on recognition accuracy.

different y-intercepts), should result.

4.4.2 Effect of Reference Pattern Averaging

Due to variations caused by the speaker in the way a word is pronounced, and by incorrect class decisions due to circuit noise and imperfect class boundaries, the observations of the reference word bit patterns are noisy. Reference matrix averaging has the benefit that the chances of an incorrect matrix are reduced. Therefore the probability of an incorrect decision is also reduced.

For a binary pattern, averaging by checking each bit for a majority occurrence in an odd number of trials, n , suggest itself. The probability that a correct assignment is made to a particular bit is given by the probability that more than $n/2$ bits are the same in n trials. This relationship is shown in Figure 4.4, where it is assumed that p is the probability of a correct bit before averaging, and p' is the probability of a correct bit after averaging. As long as the probability of a bit being correct before averaging is greater than one half, averaging should improve recognition performance. Figure 4.5 shows the effect of such an averaging scheme for the thirteen word vocabulary of Table 4.9. Reference pattern averaging is seen to improve recognition performance by reducing both the error mean

Figure 4.4 Effect of averaging on bit probability.

Figure 4.5 Effect of averaging on recognition accuracy.

and variance.

4.4.3 Effect of Thresholding

Depending on the penalty attached to an incorrect decision, a tradeoff can be made between error rate and rejection rate. According to the maximum a posteriori criterion (with fixed Bayes cost for rejection decisions) a rejection decision should be made if

$$\min_{\{w\}} D(w) > T_1 \quad (4.3)$$

where T_1 is a threshold level which is a function of the test

word. It has been found to be more convenient to make T_1 a

constant, but a user of the proposed AWR chip could write a host processor program where T_1 is a function of the test word.

Another possible rejection criteria can be applied between the reference word giving the smallest distance and the reference word giving the second smallest distance. That is, a rejection decision can be made if

$$\begin{aligned} & D(\text{word giving smallest distance}) - \\ & D(\text{word giving second smallest distance}) < T_2 \end{aligned} \quad (4.4)$$

Figure 4.6 illustrates the effect of these thresholds on the error and rejection rates for the 13 word vocabulary of Table

Figure 4.6 Effects of T_1 and T_2 on recognition performance.

4.9. It is seen that, up to a point, the higher T the higher

the error rate (the error rate started at zero percent, although this is not visible in the portion of the graphs shown) and the smaller the rejection rate. Conversely, it is seen that the higher T the lower the error rate, but the higher the rejection rate.

4.4.4 Effect of Time Warping

The two determinants for selecting the type of time warping to be used in the AWR chip are the implementation cost and the resulting recognition accuracy. Since calculating the local distances in the real time simulation occupied such a large proportion of the total computation time, the number of local distance computations required by different algorithms gives a crude estimate of the implementation cost. This relationship is reinforced by the fact that, generally, as the number of local distance computations increases, more overhead is required to control the flow of data. Table 4.10 summarizes the results of four different time warping procedures. The recognition scores shown are those achieved assuming no rejection decisions allowed and were obtained for the 13 word reference vocabulary (made using 11 repetitions). The time warpings allowed for the different algorithms are illustrated in Figure

TABLE 4.10 RESULTS OF FOUR TIME WARPING PROCEDURES

Warp Type	Figure	Error (%)	Local Distances Required
Linear	Figure 4.7(a)	3.4	15
Linear with time shift	Figure 4.7(b)	3.4	45
Nonlinear	Figure 4.7(c)	3.4	45
DTW	Figure 4.7(d)	1.6	62

TABLE 4.11 COMPARISON OF VOCABULARY COMPLEXITIES

DISTANCE RANGE	NUMBER OF WORD PAIRS WITHIN DISTANCE RANGE	
	TEN WORD VOCABULARY	ELEVEN WORD VOCABULARY
10-20	3	0
20-30	5	0
30-40	8	4
40-50	10	12
50-60	9	7
70-80	2	8
80-90	6	14
90-100	1	3
100-110	0	5

4.7(a) to 4.7(d).

Figure 4.7(a) illustrates time warping via the use of the linear transformation (Eq. 3.6)

$$\text{ADDRESS}(i) = \text{START} + \frac{L \cdot i}{M+1} \quad \text{for } i = 1 \text{ to } M \quad (4.5)$$

$D(w)$ for each reference word is calculated recursively as

$$\begin{aligned} i &= 1 \text{ to } M, \quad i=j, \quad D(0,0) = 0 \\ D(i,j) &= D(i-1,j-1) + d(Bw(i), BT(j)) \\ D(w) &= D(M,M) \end{aligned} \quad (4.6)$$

where $d(,)$ is the local distance calculation.

Figure 4.7(b) also uses a linear transformation, but two extra class decision vectors of the test word are taken using

$$\text{ADDRESS}(i) = \text{START} + \frac{L \cdot i}{M+1} \quad \text{for } i = 0 \text{ to } M+1 \quad (4.7)$$

Thus the test word is represented by $M+2$ decision vectors, and it is possible to extract three M by N vector sequences (time shifted with respect to each other) for the test word. $D(w)$ for each reference word is calculated recursively as

$$\begin{aligned} i &= 1 \text{ to } M, \quad j=i+q \text{ where } q = \{-1, 0, 1\}, \quad D(0, 0+q) = 0 \\ D_q(i,j) &= D(i-1, j-1) + d(Bw(i), BT(j)) \\ D(w) &= \min_{\{q\}} D_q(M, M+q) \end{aligned} \quad (4.8)$$

Figure 4.7(c) illustrates the use of the linear transformation of Eq. 4.7, combined with a nonlinearizing warping

(a) linear

(b) linear with time shift

(c) nonlinear

(d) DTW

Figure 4.7 Four time warping schemes. Possible paths for each scheme are indicated.

procedure, to evaluate $D(w)$. $D(w)$ is evaluated recursively as

$$\begin{aligned}
 i &= 1 \text{ to } M, \quad i=j, \quad D(0,0) = 0 \\
 D(i,j) &= D(i-1,j-1) + \min \{d(Bw(i),BT(j-1)), \\
 &\quad d(Bw(i),BT(j)), \\
 &\quad d(Bw(i),BT(j+1))\} \\
 D(w) &= D(M,M)
 \end{aligned} \tag{4.9}$$

Finally, Figure 4.7(d) illustrates a feature vector dynamic time warp suggested by Ackroyd [133] adapted to class decision vectors. Again, after a linear time transformation, $D(w)$ is evaluated recursively as

$$\begin{aligned}
 i &= 1 \text{ to } M, \quad |i-j| < 3 \\
 D(1,1) &= d(Bw(1),BT(1)) \\
 D(2,1) &= d(Bw(2),BT(1)) \\
 D(1,2) &= d(Bw(1),BT(2)) \\
 \text{if } |i-j| > 2 &\text{ then } D(i,j) = \infty \\
 D(i,j) &= \min \{D(i-1,j-1) + d(Bw(i),BT(j)), \\
 &\quad D(i-1,j) + 5, \\
 &\quad D(i,j-1) + 5\} \\
 D(w) &= D(M,M)
 \end{aligned} \tag{4.10}$$

As expected, the dynamic time warping algorithm gives the best recognition results, but is relatively computationally expensive. Observation of the time warp paths resulting from this dynamic time warping algorithm indicates that a linear transformation results for correct decisions, as shown in Figure 4.8. It is therefore not surprising that straightforward linear time transformation gives acceptable recognition accuracy. Due to its relatively low implementation cost it is the choice proposed for the AWR chip.

The linear transformation combined with time shifting was

(a)

(b)

Figure 4.8 Time warp paths resulting from the dynamic time warping algorithm (16000 is used to represent infinity). The accumulated distances are shown for (a) a correct decision where the test and reference word are matched and (b) a correct decision given non-matching test and reference words.

performed to eliminate word boundary detection error as the source for the difference between the DTW and the linear transformation algorithms. Since the time shifting linear algorithm yielded the same recognition accuracy as the straightforward linear transformation algorithm, it would appear that the performance difference is due to the non-compensated nonlinearities in the test words. As can be seen from the recognition results, non-compensated nonlinearities only affect a small percentage of the total number of words tested.

It was hoped that, by adding some of the properties of the DTW algorithm to the linear transformation, a tradeoff between implementation complexity and recognition accuracy would result (i.e., the nonlinear transformation of Figure 4.7(c)). Unfortunately, all such schemes developed by the author did not yield better performance.

4.5 Conclusion

The purpose of this chapter is not to provide absolute recognition performance results, but rather: 1) to demonstrate that the binary matrix algorithm can actually be used for the word recognition task, 2) to provide quantitative guides to the circuit performance required for accurate implementation of the algorithm, and 3) to provide qualitative guides to the

controllable factors affecting recognition performance.

The first of these purposes is provided by the recognition accuracies reported. The second is provided by Table 4.2, which emphasizes the effect between circuit dynamic range and algorithm performance. The final purpose is covered by the descriptions of the effects of thresholding and reference pattern training on recognition performance.

There are basically three causes of word recognition error which can be deduced from the test results: 1) non-compensated nonlinearities, 2) the similarity of some of the words in the recognition vocabulary, 3) finite circuit dynamic range values, and 4) other causes, such as incorrectly pronounced words. The effect of non-compensated nonlinearities should be suffered due to the reduced implementation cost of the linear time warping algorithm. Word similarity results if the classes used yield similar binary decision matrices for some of the reference words. Table 4.11 shows the distance scores between the words in the reference vocabularies for the ten and eleven word vocabularies of Table 4.9. It is seen that the ten word vocabulary, which has more relatively similar words than the eleven word vocabulary, also did not yield as good a recognition rate. However, improvements to the class selection and reference vocabulary training procedures are possible since it was not the purpose of

this work to optimize these processes. Since they affect only data, such improvements need not affect the proposed circuit implementation for the AWR chip. As seen from Tables 4.1 and 4.2 the dynamic ranges of the breadboarded AWR algorithm were low enough that some bit errors could result, although at a low rate. Integrated circuit implementation should improve dynamic range and therefore also recognition performance. Finally, other causes for recognition error, such as mispronounced words, tend to be uncontrollable and very hard to detect. Fortunately, their frequency of occurrence seems to be low.

CHAPTER 5

CONCLUSION

5.1 Introduction

This chapter is divided into four parts. First, a summary of the work performed in this thesis is provided. Second, possible improvements to the circuit implementation are discussed. Next, possible improvements from a systems viewpoint are presented. Finally, other applications of the proposed AWR chip are discussed.

5.2 Summary of thesis and list of contributions

By considering the data reduction inherently possible in class matching based word recognition algorithms, and the efficiency of analog techniques for filtering and correlation, the binary distance matrix word recognition algorithm was developed. It was then shown that the performance of the binary matrix algorithm is dependent on circuit dynamic range at d.c. The manner by which various circuit techniques which reduce the effect of this d.c. noise can be incorporated into the algorithm's functional blocks were then described. Finally, real time experimental results obtained from a breadboarding of the

binary matrix algorithm were obtained. From these it was possible to show that both recognition score thresholding and reference pattern training can be used to adjust recognition accuracy. Moreover, it was demonstrated that linear time warping for a class matching based algorithm yields recognition performance only slightly inferior to that provided by much more complex non-linear time warping algorithms. Finally, and most importantly, the recognition performance obtained validated both the binary matrix's algorithm and proposed circuit implementation.

A novel mixed analog digital LSI realization of an automatic word recognition algorithm developed by the author has been proposed in this thesis. The principle contributions required for this achievement are listed below.

1. The description of a possible circuit design for an isolated word recognition chip. This required

- (a) the novel use of the multiplying digital to analog converter and switched capacitor integrator for the efficient implementation of class decisions

- (b) the adaptation of an existing multiplexed sample and hold circuit for use in peak-extraction

- (c) the development of a mixed analog and digital

automatic gain control circuit

(d) the identification of d.c. offset as a major obstacle to analog implementation of the class matching algorithm, and the proposal of circuit solutions.

2. The development of the binary matrix class matching algorithm for non-preselected vocabulary isolated word recognition. This required

(a) an overview of the field of speech recognition and identification of the main system problems to be overcome, which included speech amplitude and rate variance, and feature and class set selection. In addition, it was necessary to reconcile the fact that while the ear appears to make class decisions using data from different instants in time [134], in terms of implementation it is easier to make class decisions based on individual time instants.

(b) the determination of the tradeoffs implicit between the class and feature matching algorithms which can be used as circuit design aids, i.e.,

(i) integrated circuit area versus processing bandwidth considering analog and digital implementations of the binary matrix recognition algorithm

(ii) ease of implementation versus recognition accuracy: the type of arithmetic operations and control

over data flow required

(iii) memory requirements for test and reference pattern storage

3. Experimental real time verification of the proposed recognition algorithm.

5.3 Suggestions for Future Work: Circuit Implementation

There are several ways by which it may be possible to reduce the area of the AWR chip.

The automatic gain control circuit described in Chapter 3, for example, may be implemented in a different manner. Automatic gain control using a completely analog circuit or a first order adaptive filter as suggested by Chiba [135] may have more efficient implementations.

One of the main factors to be considered in the switched capacitor design of the bandpass filters used for feature extraction is dc offset. Unfortunately, while the dc offset of a second order bandpass filter can be made better than 40 dB down from the rails, more work is required to determine the dc noise performance of a multiplexed implementation of a filter bank. As mentioned by Bosshart [25], significant area savings may result.

Another avenue through which area savings may result is by

the use of either higher density analog LSI or VLSI fabrication processes. While it appears the trade-off between noise (clock feedthrough, $1/f$ and kT/C) and area for switched capacitor circuits [107] will set a limit to integration density, area reduction may be made before this tradeoff becomes important for the word recognition application.

5.4 Suggestions for Future Work: System Implementation

5.4.1 Algorithm Tuning

It may be possible to improve several of the procedures making up the word recognition algorithm. Three examples are described below.

In the current version of the AWR algorithm word endpoint determination is performed after sampling the test word a fixed number of time samples after the end-of-silence flag is set. A search is then made through the resulting class decision vectors to eliminate test word candidates which are obviously incorrect. For short duration words this procedure has the drawback of slowing down the recognition process since extra "silence" decision vectors must be sampled before the word decision process can begin. Thus a more dynamic endpoint detection algorithm would be beneficial.

The linear transformation used to obtain the binary class decision matrix does not quite take care of all the time warping present in some of the test words. While no simple non-linear algorithm for this purpose has yet been found by the author, this does not mean that none such can be found.

Only majority count type averaging for reference pattern training has been demonstrated in this work. Other potential reference pattern training algorithms include: using more than one representation of a word in the reference vocabulary, using DTW to enable a more accurate majority average, or using the best choice of time shift to apply to each version of a reference word before averaging.

5.4.2 Class Selection

In this work, the class selection process was not automatic and therefore cumbersome. Moreover, since only six classes (see Table 4.5) were chosen from which to derive the weight vectors, binary matrix similarity between different words tends to reduce recognition performance. There are several ways with which these situations may be improved. A partial solution, i.e., choosing more classes, is not a problem (see Table 4. for a wider selection). Choosing additional classes efficiently, however, is. A more comprehensive solution may be to make use

of unsupervised feature space partitioning techniques, such as those used in algorithms for vector quantizer design [135-139].

Yet another approach is to search for a feature set which would permit a better class set. For example, distinctive classes [13,14] would be an ideal set of non-mutually exclusive classes for the binary matrix recognition algorithm since they enable phoneme identification using only twelve bits per class decision vector.

5.5 Extrapolation of Class Matching to other Applications

5.5.1 Continuous Speech Recognition

While performing feature template matching using DTW is possible and can yield good results, the number of local distance calculations grows as $V \cdot L \cdot N \cdot M / 3$ [53], where N is the average duration (in frames) of a reference pattern, L is the maximum number of reference patterns required for a phrase, V is the number of number of reference patterns, and M is the duration of a test phrase. This number very quickly becomes unmanageable. Unfortunately, straightforward linear time transformation is not expected by the author to perform well for continuous speech, so that the binary matrix class matching algorithm is not suitable. A more promising approach may be to use just one class decision

per frame and no time transformation (as described in Section 2.7.2 and Figure 4.2) and to use syntactical and phonological knowledge for error correction. This would require some modification of the proposed AWR chip, but mainly to the algorithm implemented by its digital processor. The reduced dimensions of the problem may then make continuous speech recognition using some form of class string matching DTW algorithm more practical, so that even a host processor program would be capable of the real time performance required.

5.5.2 Speaker-Related Recognition Systems

When using weight vectors and reference word patterns obtained from one speaker, it has been observed by the author that the recognition systems performance is speaker sensitive. This fact can serve as the basis for a text dependent speaker identification/verification system using the proposed AWR chip and an appropriate host processor program.

On the other hand, speaker independent word recognition may be achieved by the use of reference pattern clustering techniques [140], where repetitions of each reference word from many speakers are clustered. Thus the proposed AWR chip with the appropriate host processor programs should be able to perform speaker independent recognition.

REFERENCES

- (1) Reddy, D.R. "Speech Recognition by Machine: A Review", IEEE Proc., Vol. 64, No. 4, April 1976.
- (2) Welch, J.R., "Automatic Speech Recognition - Putting it to work in Industry", IEEE Computer Magazine, p. 65, May, 1980.
- (3) Lea, W.A., ed., "Trends in Speech Recognition", Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980.
- (4) Lerman, L., "Traceability via Voice Data Entry", IEEE Trans. On Comp., Hybrids, and Man. Tech, Vol. CHMT-3, p.360, 1980.
- (5) Warren, B., "Applications of LSI to Speech Recognition", IEEE 1981 COMPCON Spring, pl38.
- (6) Copeland, M.A., et al., "Sampled Analog Filtering Using Switched Capacitors as Resistor Equivalents", IEEE JSSC, Vol.12, No.6, p.592, 1977.
- (7) Itakura, F., "Minimum Prediction Residual Principle Applied to Speech Recognition", IEEE Trans. ASSP-23, p. 67, 1975.
- (8) Reddy, D., "Segmentation of speech Sounds", JASA, Vol. 40, p. 307, 1966.
- (9) Poza, F., Becker, R., "Acoustic Phonetic Research in Speech Understanding", IEEE Trans. on ASSP-23, p. 416, 1975.
- (10) Walker, D., "The SRI Speech Understanding System", IEEE Trans. on ASSP, Vol. 23, p. 397, 1975.
- (11) Flanagan, J.L., "Speech Analysis, Synthesis, and Perception", Springer-Verlag, N.Y., 1972.
- (12) Halle, M., JASA, Vol. 28., p.303, 1954.
- (13) Jacobson, R., "Fundamentals of Language", Mouton and Co., The Hague, 1956.

- (14) Fant, G., "Speech Sounds and Features", MIT Press, Cambridge, Mass., 1973.
- (15) Potter, R., et. al., "Visible Speech", Dover Pub., Inc., N.Y., 1966.
- (16) Schafer, R., Rabiner, L., "Digital Processing of Speech Signals", Prentice-Hall, 1978.
- (17) Reddy, D.R., "Phoneme Grouping for Speech Recognition", JASA, Vol. 41, p. 1295, 1967.
- (18) Reddy, D.R., "Computer Recognition of Connected Speech", JASA, Vol. 42, p. 329, 1968.
- (20) Ichikawa, A., et. al., "Evaluation of Various Parameter Sets in Spoken Digit Recognition", IEEE Trans., AU-21, No. 3, p. 202, 1973.
- (21) Nierderjohn, R., Castelaz, P., "A Comparison of Linear Prediction, FFT, and Zero-Crossing Analysis Techniques for Vowel Recognition", IEEE Conf. Rec., p. 541, 1978.
- (22) Schafer, R., Oppenheim, A., "Digital Signal Processing", Prentice-Hall Inc., Englewood Cliffs, N.J., 1975.
- (23) Lyon, T., Cox, D., Tseng, H., "A Monolithic Audio Spectrum Analyzer for Speech Recognition Systems", ISSCC 1982, p. 272.
- (24) Lyon, T.L., Cox, D.B., et. al., "A Monolithic Audio Spectrum Analyzer", IEEE Trans. on ASSP, Vol. 31, p. 288, 1983.
- (25) Bosshart, P., "A Multiplexed SC Filter Bank", IEEE JSSC-15, p. 939, 1980.
- (26) Widrow, B., et. al., "Adaptive Noise Cancelling: Principles and Applications", Proc. of IEEE, Vol. 63, No. 12, p. 1692, 1972.
- (27) Morgan, D.R., Craig, S., "Real-Time Adaptive Linear Prediction Using The LMS Gradient Algorithm", IEEE Trans. on ASSP, Vol. 24, No. 6, p. 494, 1976.

- (28) White, M., et. al., "Charge Coupled Device Adaptive Discrete Analog Signal Processing", IEEE JSSC-14, p. 132, 1979.
- (29) Ahuja, B., "A Sampled Analog MOS LSI Adaptive Programmable Transversal Filter", Ph. D. Thesis, Carleton University, 1978.
- (30) Chan, D., "Investigation of a Mixed Digital Analog Adaptive Filter", M.Eng. Thesis, Carleton University, 1980.
- (31) Reed, F., "A Comparison of LMS Adaptive Cancellers Implemented in the Frequency Domain and the Time Domain", IEEE Trans on ASSP-29, p. 770, 1981.
- (32) Markel, J.D., Gray, A., "Linear Prediction Of Speech", Springer-Verlag, N.Y., 1976.
- (33) Makhoul, J., "Linear Prediction - A Tutorial", IEEE Proc. Vol. 63, p. 561, 1975.
- (34) Honig, M., Messershmitt, D., "Convergence Properties of An Adaptive Digital Lattice Filter", IEEE Trans. on ASSP, Vol. 29, No. 3, p. 642, 1981.
- (35) Makhoul, J., Cosell, L., "Adaptive Lattice Analysis of Speech", IEEE Trans. on ASSP, Vol. 29, No. 3, p. 654, 1981.
- (36) Brodersen, R.W., et. al., "MOS-LSI Analog/Digital Techniques for Linear Prediction", IEEE 1980 ICC, p. 57.1.1.
- (37) Lee, D., et. al., "Recursive Least Squares Ladder Estimation Algorithms", IEEE Trans. on ASSP, Vol. 29, No. 3, p. 627, 1981.
- (38) Griffith, L., Medaugh, R., "A Comparison of Two Fast Linear Predictions", IEEE 1981 Conf. Rec., p. 293.
- (39) Buckley, K., Rao, S., "A Comparison of adaptive Gradient and Adaptive Least-Square Algorithms", Int. Conf. on ASSP 1981, p. 283.
- (40) Makhoul, J., "Stable and Efficient Lattice Methods

for Linear Prediction", IEEE Trans. on ASSP-25, p. 423, 1977.

- (41) Barnwell, T., "Recursive Autocorrelation Computation for LPC Analysis", IEEE ASSP Conf. Rec., p. 1, 1977.
- (42) Fellman, R., Brodersen, R., "A Switched-Capacitor Lattice Filter", IEEE Trans. on ASSP, Vol. 31, p. 294, 1983.
- (43) Fellman, R., Hurst, P., Brodersen, R., "Switched Capacitor Circuits for Adaptive Filtering and Autocorrelation", ISSCC 1983, p. 126.
- (44) Martin, K., "A Switched Capacitor Realization of a Spectral Line Enhancer", ISCASP 82, p. 229.
- (45) Gupta, S., "Phase-Locked Loops", IEEE Proc., Vol. 63, p. 291, 1975.
- (46) Brown, K., et. al., "IQ-Locked Loop Tracker", Int. Conf. on ASSP 19 77, p. 293.
- (47) Morris, L., "The Role of Zero Crossings in Speech Recognition and Processing"
- (48) Tou, J., Gonzalez, R., "Pattern Recognition Principles", Addison-Wesley Pub. Co., 1974.
- (49) Searle, C., et. al., "Stop Consonant Discrimination Based on Human Audition", JASA, Vol. 65, p. 799, 1979.
- (50) Zwicker, E., Terhardt, E., "Automatic Speech Recognition using Psychoacoustic Models", JASA, Vol. 65, p. 487, 1979.
- (51) Li, K., et. al., "Correlation Characteristics and Dimensionality of Speech Spectra", JASA Vol. 46, p. 1019, 1969.
- (52) White, A., Neely, R., "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering, and Dynamic Programming", IEEE Trans. on ASSP, Vol. 24, No. 2, p. 183, 1976.
- (53) Myers, C., Rabiner, L., "A Level Building DTW Algorithm for Connected Word Recognition", IEEE

Trans. ASSP-29, p. 284, 1981.

- (54) Sakoe, H., Chiba, S., "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", IEEE Trans. ASSP-26, p. 43, 1978.
- (55) Rabiner, L., et. al., "Considerations in DTW Algorithms for Discrete Word Recognition", IEEE Trans. ASSP-26, p. 575, 1978.
- (56) Myers, C., et. al., "Performance Tradeoffs in DTW Algorithms for Isolated Word Recognition", IEEE Trans. ASSP-28, p. 623, 1980.
- (57) Myers, C., Rabiner, L., "Connected Digit Recognition Using a Level-Building DTW Algorithm", IEEE Trans. on ASSP, Vol. 29, p. 351, 1981.
- (58) Rabiner, L., Schmidt, C., "A Connected Digit Recognizer Based on DTW and Isolated Digit Templates", Int. Conf. on ASSP 1980, p. 194.
- (59) Rabiner, L., et. al., "On the Effects of Varying Analysis Parameters on an LPC Based Isolated Word Recognizer", BSTJ, Vol. 60, p. 893.
- (60) Sakoe, H., "Two-Level DP-Matching - A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition", IEEE Trans. on ASSP, Vol. 27, p. 588, 1979.
- (61) Dusek, L., et al., "Voice Recognition Joins Speech on Programmable Board", Electronics, p. 128, April 21, 1983.
- (62) Alderfeld, B., et. al., "Automated Directory Listing Retrieval System Based on Isolated Word Recognition", IEEE Proc. Vol. 68, p. 1364.
- (63) Telecommunications, Jan. 1983, p. 15, (Advertisement)
- (64) Iwata, T., et. al., "A Speech Recognition Processor", ISSCC 1983, p. 120.
- (65) Lowy, M., et. al., "An Architecture for a Speech Recognition System", ISSCC 1983, p. 118.

- (66) Ackenhusen, J., Rabiner, L., "Microprocessor Implementation of an LPC-Based Isolated Word Recognizer", Int. Conf. on ASSP 1981, p. 746.
- (67) Bui, C., Monbaron, J., Michel, J., "An Integrated Voice Recognition System", IEEE Trans. on ASSP, Vol. 31, p. 323, 1983.
- (69) Rabiner, L., Sambur, M., "Some Preliminary Experiments in the Recognition of Connected Digits", IEEE Trans. on ASSP-24, p. 170, 1976.
- (70) Teacher, C., et. al., "Experimental Limited Vocabulary Speech Recognizer", IEEE Trans. on AU-15, p. 127, 1967.
- (71) King, J., Tunis, C., "Some Experiments in Spoken Word Recognition", IBMJ of R. and D, Vol. 10, p. 65, 1966.
- (72) Biancomano, V., "Limited Speech Recognition", QST, Oct., 1972, p. 36.
- (73) De Mori, R., "Experiments in Automatic Segmentation of Continuous Speech", Acustica, Vol. 34, p. 158, 1976.
- (74) Gilli, L., Meo, A., "Sequential System for Recognizing Spoken Digits in Real Time", Acustica, Vol. 19, p. 38, 1967/68.
- (75) Dammann, J., "Applications of Adaptive Threshold Elements of the REcognition of Acoustic Phonetic States", JASA, Vol. 38, p. 213, 1965.
- (76) Wiren, J., Stubbs, H., "Electronic Binary Selection System for Phoneme Classification", JASA, Vol. 28, p. 1082, 1956.
- (77) Dudley, H., "Phonetic Pattern Recognition Vocoder for Narrow-Band Speech Transmission", JASA Vol. 30, p. 733, 1958.
- (78) Davis, K., et. al., "Automatic Recognition of Spoken Digits", JASA Vol. 24, p. 637, 1958.
- (79) Belar, H., Olson, H., "Speech Processing Techniques

- and Applications", IEEE Trans on AU-15, p. 120, 1967.
- (80) Denes, P., Mathews, M., "Spoken Digit Recognition Using Time-Frequency Pattern Matching", JASA, Vol. 32, p. 1450, 1960.
 - (81) Bakis, R., Sholtz, P., "Spoken Digit Recognition Using Vowel-Consonant Segmentation", JASA, Vol. 34, p. 1, 1962.
 - (82) Kido, K., et. al., "Segmentation and Classification of Speech by Use of Low Q Single Tuned Filters", Int. Conf. on ASSP 1972, K3, p. 396, I4, p. 327.
 - (83) Keller, T.G. von, "An On-Line Recognition System for Spoken Digits", JASA Vol. 49, p. 1288, 1971.
 - (84) Tanaka, K., "A Dynamic Processing Approach to Phoneme Recognition - Feature Extraction", IEEE Trans. on ASSP, Vol. 27, p. 596, 1979.
 - (85) Furui, S., "A Training Procedure for Isolated Word Recognition Systems", IEEE Trans. on ASSP, Vol. 28, p. 129, 1980.
 - (86) Regel, P., "A Module for Acoustic-Phonetic Transcription of Fluently Spoken German Speech", IEEE Trans. on ASSP, Vol. 30, p. 440, 1982.
 - (87) Rabiner, L., et. al., "Speaker Independent Recognition of Isolated Words Using Clustering Techniques", IEEE Trans. ASSP-27, p. 336, 1979.
 - (88) Medress, M., et. al., "An Automatic Word Spotting System for Conversational Speech", Int. Conf. on ASSP 1978, p. 712.
 - (89) Baudry, M., Dupeyrat, B., "Speech Segmentation and Recognition Using Syntactic Methods", Int. Conf. on ASSP 1979, p. 101.
 - (90) Tappert, C., "Experiments with a Tree Search Method for Converting Noisy Phonetic Representation into Standard Orthography", IEEE Symp. on Speech Recognition, p. 261, 1974.
 - (91) Lowerre, B., "The Harpy Speech Recognition System",

Computer Sci Ph. D Thesis, Carnegie-Mellon University, 1976.

- (92) Kohonen, T., et. al., "A Thousand Word Recognition System Based on the Learning Subspace Method and Redundant Hash Processing", Int. Conf. on Pattern Recognition, p. 158, 1980.
- (94) Okuda, T., et. al., "A Method for the Correction of Garbled Words Based on the Levenshtein Metric", Computer, Vol. 25, p. 172.
- (95) Kashyap, R., Mittal, M., "A New Method for Error Correction in Strings with Application to Spoken Word Recognition", Int. Conf. on ASSP 1977, p. 675.
- (96) Martin, T., "Acoustic Recognition of a Limited Vocabulary in Continuous Speech", Ph.D. Thesis, University of Pennsylvania, 1970.
- (97) Herscher, M., Cox, R., "An Adaptive Isolated Word Speech Recognition System" IEEE 1972 Conf. on Speech Communication and Processing, p. 89.
- (98) Schroeder, M., "Similarity Measure for Automatic Speech and Speaker Recognition", JASA, Vol. 43, p. 375, 1968.
- (99) Tappert, C., Das, S., "Memory and Time Improvements in a Dynamic Programming Algorithm for Matching Speech Patterns", IEEE Trans. ASSP-26, p. 583, 1978.
- (100) Roy, D., Menard, P., "High Performance Digitally Programmable Analog Transversal Filter", Electronic Letters, Vol. 14, No. 20, p. 671, 1978.
- (101) Feiveson, A., "Classification By Thresholding", IEEE Trans. PAMI-5, p. 48.
- (102) Kashyap, R., Mittal, M., "Recognition of Spoken Words and Phrases in Multitalker Environment Using Syntactic Methods", Computer, Vol. 27, p. 442.
- (103) Gonzalez, R.C., Thomason, M.G., "Syntactic Pattern Recognition", Addison-Wesley Pub. Co., 1978.
- (104) Sakoe, H., Chiba, S., "A DP Approach to Continuous

Speech Recognition", The Seventh Int. Cong. On Acoustics (Budapest), p. 20C13, 1971.

- (105) Gregorian, R., Wegner, G., "A CMOS ADM CODEC for PABX Applications", ISSCC 1983, p. 72.
- (106) Fellman, R., Brodersen, R., "A Switched Capacitor Lattice Filter", IEEE JSSC-18, p. 46, 1983.
- (107) Einspruch, N., ed., "VLSI Electronics, Vol. 2", Academic Press, Toronto, p. 189, 1981.
- (108) Ribner, D., Private Communication, Nov., 1981, and May, 1983.
- (109) Rahim, C., "An Investigation of MOS Switched Capacitor Filter Circuits", Ph.D. Thesis, Carleton University, 1978.
- (110) Pulsifer, D., "Application of Adaptive Noise cancellation to Noise Reduction in Audio Signals", Ph. D. Thesis, University of Utah, 1979.
- (111) Lim, J., Oppenheim, A., "Enhancement and Bandwidth Compression of Noisy Speech", IEEE Proc. Vol. 67, p. 1586, 1979.
- (112) Dunn, H., White, S., "Statistical Measurements on Conversational Speech", JASA Vol. 11, p. 278, 1940.
- (114) Martin, K., Sedra, A., "Easing Prefiltering Requirements of SC Filters", Elect. Letters, Vol. 16, p. 613, 1980.
- (115) Gregorian, R., Nicholson, W., "Switched Capacitor Decimation and Interpolation Circuits", IEEE Trans. CAS-27, p. 509, 1980.
- (116) Crochiere, R., Rabiner, L., "Interpolation and decimation of Digital Signals", IEEE Proc. Vol. 69, p. 300, 1981.
- (117) Nierderjohn, R., Grotelueschen, J., "The Enhancement of Speech Intelligibility in High Noise Levels By High Pass Filtering Followed By Amplitude Compression", IEEE Trans. ASSP-24, p. 277, 1976.

- (118) Sedra, A., Brackett, P., "Filter Theory and Design: Passive and Active", Matrix Pub. Inc., Champaign, Ill., 1978.
- (119) Gregorian, R., "An Offset Free Switched Capacitor Biquad", Microelectronics Journal, No. 13, p. 37, 1982.
- (120) Sunter, S., Girczyc, E., Chowaniec, A., "A Programmable Transversal Filter of Voice Frequency Application", IEEE JSSC, Vol. 16, No. 4, p. 367, 1981.
- (121) McCreary, J., Gray, P., "All MOS Charge Redistribution Analog-to-Digital Conversion Techniques", IEEE JSSC-10, p. 371, 1975.
- (122) Lam, K., Private Communication, May 1983.
- (123) Bruckert, E., et. al., "Three-Tiered Software and VLSI Aid Developmental System to Read Text Out Aloud", Electronics, p. 133, April 21, 1983.
- (124) Dixon, W., ed., "BMDP Statistical Software", University of California Press, Los Angeles, 1981.
- (125) Mercier, G., Callec, J., et. al., "Automatic Segmentation, Recognition of Phonetic Units and Training in the Keal Speech Recognition System", Int. Conf. on ASSP 1982, p. 2000.
- (126) Rosenberg, A., Rabiner, L., Wilpon, J., "Isolated Word Recognition for Large Vocabularies", BSTJ, Vol. 61, p. 2989.
- (127) Rosenberg, A., Rabiner, L., Wilpon, J., "Speaker Trained Recognition of Large Vocabularies", Int. Conf. on ASSP 1982, p. 2018.
- (128) Kawakami, Y., et. al., "A Single-Chip DSP for Voiceband Applications", IEEE ISSCC 1980, p. 40.
- (129) Gillingham, P., "Switched Capacitor Circuits for VLSI: A Design Study", M.Eng. Thesis, Carleton University, Dec., 1982.
- (130) Wong, S., Salama, A., "Impact of Scaling on MOS Analog Performance", IEEE JSSC-18, p. 106, 1983.

- (131) Iwata, A., "A Single-Chip Codec with Switched Capacitor Filters", IEEE JSSC-16, p. 315, 1981.
- (132) Matsui, K., "2 u CMOS Switched Capacitor Circuits for Analog Video LSI", ISCAS 1982 Proc., p. 241.
- (133) Ackroyd, M., "Isolated Word Recognition using the Weighted Levenshtein Distance", IEEE Trans. on ASSP-28, p. 243.
- (134) Kewley-Port, D., "Time Varying Features as Correlates of Place of Articulation in Stop Consonants", JASA, Vol. 73, p. 322.
- (135) Chiba, ., et al, "A speaker Independent Word Recognition System", ICPR 1978, p. 995.
- (136) Fontana, R., Fox, M., "A Composite Source Model for Speaker and Isolated Word Recognition", ICASSP 1981, p. 741.
- (137) Shore, J., Gray, R., "Minimum Cross-Entropy Pattern Classification and Cluster Analysis", IEEE PAMI-4, p. 11, 1982.
- (138) Shore, J., Burton, D., "Discrete Utterance Speech Recognition Without Time Normalization", ICASSP 1982, p. 907.
- (139) Tou, J., Gonzalez, R., "Pattern Recognition Principles", Addison-Wesley Pub. Co., Inc., 1974.
- (140) Rabiner, L., Wilpon, J., "Considerations in Applying Clustering Techniques to Speaker Independent Word Recognition", JASA Vol. 65, 1979.
- (141) Magar, S., Caudel, E., Leigh, A., "A Microcomputer with Digital Signal Processing Capability", ISSCC 1982, p. 32.
- (142) Doddington, G., Schalk, T., "Speech Recognition: Turning Theory to Practice", IEEE Spectrum, Sept. 1981, p. 26.
- (143) Nahow, A., "Automatic Speech Recognition: A Tutorial", G.E Computer Management Operation, TIS

81 CMO 114, Aug., 1981.

- (144) Hieronymous, J., Enea, H., "Evaluating the Performance of Commercial Speech Recognizers", IEEE 1982 Compcon Spring Digest of Papers, p. 207.
- (145) Bell Northern Research, unpublished notes, 1978.
- (146) Fry, D.B., "The Physics of Speech", Cambridge University Press, Cambridge, p. 68, 1979.
- (147) Kikuchi, H., Inaba, T., et. al., "A 23K Gate CMOS DSP with 100 ns Multiplication", ISSCC 1983, p. 128.
- (148) Clark, J., Collins, P., Lowerre, B., "A Formalization of Performance Specifications for Discrete Utterance Recognition Systems", ICASSP 1981, p. 753.

END

0	6	1	0	9	H	8	4
---	---	---	---	---	---	---	---

FIN