INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

University Microfilms International
300 North Zeeb Road
Ann Arbor, Michigan 48106 USA
St. John's Road, Tyler's Green
High Wycombe, Bucks, England HP10 8HR

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/4446

7824582

ANDREWS, HAROLD LEE
ABNORMAL SPEECH RECOGNITION,

THE UNIVERSITY OF OKLAHOMA, PH.D., 1978

University Microfilms International 300 N. ZEEB ROAD, ANN ARBOR, MI 48106

© 1978

HAROLD LEE ANDREWS

ALL RIGHTS RESERVED

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ABNORMAL SPEECH RECOGNITION

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

BY

HAROLD LEE ANDREWS

Norman, Oklahoma

1978

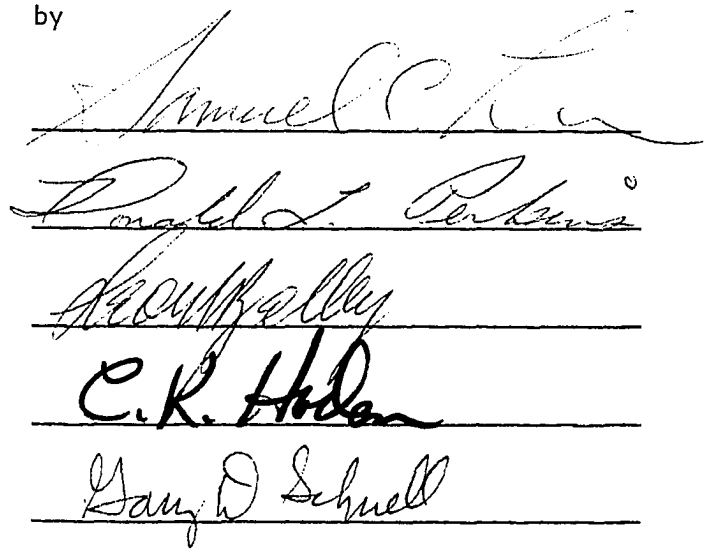
ABNORMAL SPEECH RECOGNITION

A DISSERTATION

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

AND COMPUTING SCIENCES

by



DEDICATION

This dissertation is dedicated to my wife, Amour, and my daughter, Anita.

ACKNOWLEDGEMENT

We are fortunate when others will assist us with no expected reward except our gratitude.

I am deeply grateful to Virgil Danforth, Barbi Carter, Camille Zink, and Charles Roberts who took time from the business of fighting to master their illness to help me.

I express my appreciation to Ms. Joyce Brandeis for allowing me into her classroom. Also, I thank Mr. Harper Orth for allowing me access to the students at the Cerebral Palsy Center in Norman, Oklahoma. I acknowledge the assistance of Mr. Edward Malone and Mr. Richard Coleman.

I am grateful to Ms. Patricia Sullivan for having done an excellent job of typing.

My sincere gratitude goes to members of my committee, Dr. Gary Schnell, Dr. Donald Perkins, and Dr. Clovis Haden. I am particularly grateful to Dr. Leon Zelby for his encouragement, support, and sober advice. Finally, this dissertation could not have been completed without the technical guidance and support of Dr. Samuel Lee.

TABLE OF CONTENTS

		Page
	ABSTRACT	vi
	GLOSSARY	viii
I.	INTRODUCTION	1
	Design Considerations and Background Information	5
	Vocabulary	5
	Real-Time Feature Extraction Techniques	6
II.	BACKGROUND AND RECENT DEVELOPMENTS	8
	Technical Considerations	11
	Recall of Word Prototypes	13
	Recent Advances	15
	Fundamental Problems	19
	Improving System Performance	23
III.	PATTERN RECOGNITION (An Overview)	28
	Deterministic Pattern Recognition	30
	The Perceptron Approach	30
	Piecewise Linear Classifiers	32
	The Potential Function Approach	33
	Statistical Pattern Recognition	37
	Regression Functions	37
	Stochastic Approximation for Decision Function Estimation	41
	Increment - Correction Algorithm	44
	The Method of Potential Functions	45
IV.	LINEAR PREDICTIVE CODING OF SPEECH	49
	An Autocorrelation Method of Linear Prediction	52
	The Inner Product Formulation	55
	The Recursive Solution	58
	Markel's Algorithm	63
	Lumped Linear Prediction	65

TABLE OF CONTENTS--Continued

Page

V. EXPERIMENTAL RESULTS 67

Experiments 67
Results 71
Linear Error Correction (LEC). 71
LEC With Pattern Deletion. 71
Piecewise Linear Error Correction (PWL). 72
Fractional Error Correction (FEC). 73
The Potential Function Approach. 73
Increment Correction (Statistical) 74

VI. DISCUSSIONS. 75

The Effectiveness of Coding Words
Using Lumped Linear Predictive Coding. 75
Comparison of Various Learning Algorithms 76
Piecewise Linear Error Correction. 76
Linear Error Correction. 77
Fractional Error Correction and Potential Functions 77
Statistical Algorithms 77
End-Point Detection 78
Other Considerations. 79
Pattern Deletion During Training. 80

VII. CONCLUSIONS. 83

APPENDIX A: FLOWCHART AND PROGRAM FOR LUMPED LINEAR PREDICTION. 85

APPENDIX B: FLOWCHARTS OF RABINER AND SAMBUR'S END-POINT LOCATION ALGORITHM. 91

APPENDIX C: FLOWCHART AND EXAMPLES OF PATTERN RECOGNITION ALGORITHMS. 96

BIBLIOGRAPHY 122

ABSTRACT

This dissertation addresses the problem of machine recognition of the speech of people afflicted with cerebral palsy. At the time of this writing, machine recognition of speech has not developed to the point of general applicability for use by government, industry or the general public. Progress has been made to the extent that there are isolated word recognition machines that are in use by some companies, and some special purpose devices are being marketed.

Abnormal speech recognition allows the opportunity for a fresh look at machine recognition of speech, and, as a consequence of the intended limited application of the approaches presented in this paper, novel approaches were examined that ultimately will result in the development of a machine for abnormal word recognition.

It must be made clear that no attempt was made to spectrally distinguish abnormal speech from normal speech. Both types of speech are comprised of stochastic signals that contain vast amounts of information. Although it is recognized that the general population finds it extremely difficult to understand handicapped speech, there is no reason to believe that the speech of people with cerebral palsy cannot be coded for recognition by machines. The utterances of cerebral palsy victims are distinguishable, and this is demonstrated by the fact that they do communicate with people who learn to understand their speech.

Coding of speech was a major task and the method used in this papaper, called Lumped Linear Prediction, is considered to be the major contribution of this research to speech recognition. Lumped linear prediction applies linear predictive coding to the entire speech wave of an utterance. The difference is in application. Other researchers use linear prediction, a form of Wiener filtering, to code segments of speech using a small number (8 to 16) of coefficients for each segment. The segments are 100ms to 300ms in duration. Lumped linear prediction, in this particular research, uses eight coefficients to code an entire utterance. This is a dramatic reduction in the dimensionality of training patterns used for pattern recognition. Rather than searching for a means to efficiently classify several hundred or even thousand coefficients, the efforts here were directed toward determining a suitable pattern recognition scheme for words that were coded with just eight coefficients. Good results were obtained for a small vocabulary (ten words). Extension to larger vocabularies, although dependent upon other factors, can be achieved by using more coefficients.

With respect to pattern recognition, an important observation was made that the number of iterations required for convergence to solution weight vectors is tremendously influenced by the deletion of patterns during the training process. Patterns are deleted based on the condition that discriminant conditions are satisfied during training; that is, for a pattern $\overline{X}$ belonging to class ω_i, if it is determined that $D_i(x) > D_j(x)$ (where $D_j(x) = \overline{W}_j \cdot \overline{X}$, $j = 1, \ldots, M$, $j \neq i$), then this pattern is deleted from the training set.

GLOSSARY

Autocorrelation

The correlation of a function with itself. Assuming that a sequence of N speech samples $\{s(n)\} = \{s(0), s(1), \ldots, s(N-1)\}$ is available, an autocorrelation sequence $r(\ell)$ is generated as follows:

$$r(\ell) = \sum_{n=0}^{N-1-\ell} s(n)\, s(n+\ell) \quad \text{for } \ell \geq 0$$

Autoregressive Filter

When a time series $\{\nu_k\}$ is generated from a time series $\{u_k\}$ in accordance with the equation

$$\nu_k = A_o\, u_k + A_1\, u_{k-1} + \ldots + A_r\, u_{k-r} - \beta_1\, \nu_{k-1} - \beta_2\, \nu_{k-2} - \ldots - \beta_s\, \nu_{k-s},$$

the term autoregressive is used when there are no A terms other than A_o. This is a recursive filter when the kth value of ν depends on the preceeding k-s values of ν.

Convergence

To approach a limit as the number of term increases without limit. Specifically, in solving for solution weight vectors $\overline{W}(K)$ for discriminant functions, the limit is reached when $\overline{W}(K+1) = \overline{W}(K)$, where K is the number of iterations.

Correlation

The correlation between wave forms is a measure of the similarity or

relatedness between the waveforms.

Criterion Function

A function so chosen that if its minimum value is achieved when $\overline{W} \cdot \overline{X}_i > 0$, where $\overline{X}_i$ is the ith row of an N x (n+1) matrix $\overline{X}$ of a system of inequalities

$$\overline{X} \cdot \overline{W} > 0$$

then finding the minimum of the function for all i, i = 1,2, ..., N, is equivalent to solving the given system of linear inequalities. The criterion function is denoted as $J(\overline{W}, X)$.

Decision Surface

The n dimensional surface that is generated by the equation

$$D(x) = \overline{W}^{(i)} \cdot \overline{X}, \text{ where } \overline{W}^{i} = (W_1, W_2, \ldots, Wn + 1)' \text{ and}$$

$$\overline{X} = \begin{bmatrix} X_1 \\ X_2 \\ \cdot \\ \cdot \\ \cdot \\ X_n \end{bmatrix}$$

when n = 2, the equation is that of a line; it is the equation of a plane when n = 3 and of a hyperplane when n > 3.

Deterministic

Based on the assumption that pattern classes are separable, deterministic refers to algorithms that are developed without making any assumptions concerning the statistical properties of the pattern classes.

Dichotomization

The act of dividing into classes or groups.

Discriminant Functions

Scaler and single-valued functions of pattern vectors $\overline{X}$ that define decision boundaries which separate pattern classes on the basis of observed vectors.

Disjoint

Classes A or B having no common elements - they are non-intersecting.

Fricatives

Sounds characterized by frictional passage of the expired voice or voiceless breath against a narrowing at some point in the vocal tract; such as, f, u, th, s, z, sh, zh and h.

Glottal

Pertaining to the glottis.

Glottis

The vocal apparatus of the larynx, consisting of the true vocal cords (plica vocalis) and the opening between them (rima glottidis).

Hyperplane

(See Decision Surface)

Inverse Filter

A filter that is defined by the equation

$$A(z) = \sum_{i=0}^{M} a_o z^{-i} \qquad (a_o = 1)$$

with $M \geq 2K + 1$, where K is the number of formants for a speech wave. A(z) is an all-zero filter.

Nasal

Sounds that are uttered through the nose with the mouth passage occluded (as m, n, ng); also, sounds uttered with the mouth open, the soft palate lowered and the nose passage producing a resonance.

Phoneme

A member of the set of the smallest units of speech that serve to distinguish one utterance from another in a language or dialect.

Phonological

Relating to the science of speech sounds (especially the theory of sound changes in a language).

Plosives

Utterances produced by the expelling of breath after the closing of the oral passages in the production of a stop consonant, as after the b in bat.

Recursive Filter

(See Autoregressive Filter)

Segmentation

The formation of several units of a speech sample for purposes of analysis of the speech wave or speech recognition.

Separable

Classes of patterns that are capable of being disassociated.

Statistical

This is in reference to algorithms that are equivalent to Baye's decision functions

$$di(X) = p(^{\omega}i/\overline{X})$$

These are distinguished from the deterministic case in that only the patterns of class ω_i are considered in the estimation of $p(\overline{X}/^{\omega}i)$. No learning takes place since patterns of other classes do not influence the estimation process.

Template

A pattern that is characteristic (having all of the important attributes) of the patterns belonging to a given class. The template is used for pattern matching. No training is required.

Training

The process of determining decision functions through a series of adjustments. Arbitrary decision functions are initially assumed, and through a sequence of iterative steps, these decision functions are made to approach optimum or satisfactory forms.

Vocal Tract

An accoustical tube which is non-uniform in cross-sectional areas - terminated by the lips at one end and by the vocal cord constriction at the other end. It is approximately 17 cm long in the adult male and is deformed by movement of the articulators; that is, the lips, jaw, tongue and velum. The cross-sectional area of the forward portion of the tract can be varied from zero to approximately 20 cm^2.

Voiced

Uttered with vocal cord vibration.

Bayes' Classifier

A classifier which minimizes total expected loss, $r_i(\overline{X}) = \sum_{i=1}^{M} L_{ij}\, p(^{\omega}i/X)$. That is, a pattern $\overline{X}$ may belong to any of M classes and the expected loss

incurred in assigning observation $\overline{X}$ to class ω_j is given by $r_j(\overline{X})$.

The Bayes' classifier assigns a pattern $\overline{X}$ to the class with the lowest value of r.

CHAPTER I

INTRODUCTION

Cerebral palsy is "a persisting qualitative disorder appearing before the age of three years, due to a nonprogressive damage to the brain." [6] It has many forms with varying degrees of severity of effect on motor activity. The patterns of paralysis are paraplegic, diplegic, and pseudobulbar.

Damage to the nervous system is often recognized at birth or soon thereafter by some abnormality of breathing, sucking and swallowing, color of mucous membrane, or responsiveness.

A particular form is Congenital Choreoathetosis (Double Athetosis) wherein every voluntary act is marred by intense involuntary movements, leaving the patient nearly helpless. The tongue may extrude from the mouth with unsightly drooling, and the face is contorted in a never ending series of grimaces. Speech is slurred or inarticulate and punctuated by grunts and unpleasant throat sounds. The hands are engaged in constant writhing, and all attempts to use the limbs results in a slow, spreading spasm of the entire limb or all of the musculature. Patients are many times erroneously classified as mentally defective because of the motor and speech impairment. No doubt, in some instances this is

correct, but others retain intellectual function and can be educated. The less severly affected patients often make successful occupational adjustments.

Corrective operations and therapy will often have significant effect towards overcoming the handicaps of the cerebral palsied to the extent that once the communications gap is bridged, in a cerebral palsy center for instance, students often demonstrate more than adequate cognitive skills; however, communications remains for many of them a significant social and learning barrier once they venture outside the walls of a rehabilitation center. Of course, the limit to which the severity of affliction can be corrected or ameliorated will vary from individual to individual; so that, a student who might have barely made intelligible utterances as a pre-teen could progress to the point of normal speech while remaining confined to a wheel chair. Another student might never overcome the speech handicap.

Approximately six out of every one-thousand newly born and fifteen out of every one-thousand people in this country are afflicted with some form of cerebral palsy. There are approximately nine-thousand victims in the state of Oklahoma. Sixty to sixty-five percent of those diagnosed as cerebral palsy victims are mentally retarded [25]. The remaining thirty to thirty-five percent of the victims may have severe physical handicaps and their speech in general is difficult to understand, but once a listener has learned to correctly decipher their utterances, he is able to carry on a conversation.

Using the microprocessor, it is expected that a machine can be designed that will recognize and display the speech of the voice handi-

capped. Although suitable techniques have been in existence for a number of years, the idea of a machine for this purpose would have had little attraction because of the expense involved and the size and therefore limited accessibility of the machine. It is also important to point out that such a machine would have very little if any utility for people with normal speech.

There is a sizable literature covering speech recognition, voice recognition, speech synthesis, vocoders, etc. The digital computer has allowed research in these areas to grow by leaps and bounds as a consequence of its speed and memory capabilities. Present techniques in speech analysis and synthesis will be markedly enhanced by machine improvements alone. There are instances in which machines have been designed and programmed to accept limited verbal instructions. In fact, pattern recognition techniques exist not only for spoken but for optical input, but there has been no published research on abnormal speech recognition. Existing techniques are almost uniformly limited because they lack general application. For example, in speech recognition, there is no technique at the time of this writing that maintains a high level of accuracy as the number of speakers increases. There are other techniques that will maintain a reasonably high, ninety-two to ninety-five percent, recognition accuracy for a relatively large number of male speakers using a limited vocabulary. Women and children with their normally higher fundamental voice frequencies cause a significant increase in recognition error when they are added to a predominantly adult male speaker population [7].

The performance requirements for a speech machine for the voice

handicapped are less stringent than would be the requirements for a machine intended for general application; therefore, the prospect of ultimate success is encouraging. The object of this effort is to investigate and develop a speech trainer/learner that will allow cerebral palsy victims to self-adjust their speech so that their speech is compatible with machine recognition. Simply stated, the machine would learn predictive coefficients of the speech waveforms from the trainee while the trainee learns to exercise greater muscle control. There is no intent of significant speech rehabilitation. To the extent that rehabilitation does occur, it would be considered purely a bonus of the training process.

The machine would not be affected by the quality of speech (unlike people); however, not having perceptual skills, the machine will rely completely on the speaker to provide it with input that would statistically match what is contained in memory. Also the machine's vocabulary would largely depend on the user, with new words being added as progress is made.

Users of the machine, speech therapists, teachers and parents would have to understand that ultimate proficiency with the machine would result only after long and tedious work. In many respects, working with the machine would not significantly differ from working with any other rehabilitative therapy. The ultimate benefit would result when the cerebral palsy victim ventures out into the world with a device that would make communications with the outside world a little easier.

Design Considerations and Background Information

Vocabulary. It will be necessary to develop a vocabulary having attributes such as maximal spectral separability and broad language utility in order to have high machine recognition reliability. The first attribute would require that such features as frequency and amplitude patterns of words be distinquishable. Although the speech wave of a given word may not be exactly reproducible with repeated enunciations, a correlation of the speech waves for the utterances must be shown to exist. This sameness or correlation must be unique for a given word, but distinct for different words. Learning algorithms require that patterns be separable in order for weight vector convergence to occur.

Secondly, the words must be useful. The trainer/learner will not necessarily have a vocabulary that would be adaptable to grammatically and syntactically correct phrases and sentences. Users must be able to make themselves understood.

To illustrate the above, consider the word list that is used by beginning students at the Cerebral Palsy Center in Norman, Oklahoma. The list is happy, eat, hungry, drink, thirsty, potty, bed, sleep, sad, hot, go, play, T.V., stop, cold, home, Daddy, Mama, brother, sister, I love you, letter, candy, clothes, and coat. Cold and coat when spoken by one of these children might sound alike, say as "'Kō". One needs not look at a speech spectrograph or speech wave in order to know that this would be unacceptable to the machine. Although more advanced children and adults might have considerably larger vocabularies, the initial vocabulary of the trainer/learner must be such that users will have the same kind of utility that this beginning list has.

Consideration must be given to the sounds (syllables, words, and phonemes) that are more easily uttered by the handicapped. These sounds cannot be uniquely determined for a broad category of people. The training process would determine words for each individual that should be modified or deleted. For example, if the "th" of thirsty could not be easily spoken, the machine could easily be programmed to recognize "irsty" and display thirsty. In some instances, total sound substitutions might be made; so that cold, for example, might be assigned "Ka" for machine recognition as cold. These considerations apply primarily to the initial vocabulary used in the early stages of training. Ultimately each user would design his own vocabulary.

Real-Time Feature Extraction Techniques. A means of generating patterns from spoken words in real time with a minimal amount of computer memory is a major technical task. There are numerous techniques that have been highly successful in isolated word recognition. On a large computer, these techniques might be used indiscriminately. However, microprocessors are not easily adaptable to performing some tasks primarily because of memory size and smaller sets of instructions codes. Yet they are very attractive from both cost and size standpoints.

The object in this project is to simply match patterns, the technique used to characterize information for pattern recognition need not be elaborate. To summarize from Schafer and Rabiner [28]: The main decisions to be made in the design of word recognition systems are 1). How to normalize for variations in speech; 2). What is the parametric representation; 3). How does the system adapt to a new speaker or new vocabulary; 4). How does one measure the similarity of two utterances; and 5).

How to speed up matching. These items are defined and elaborated upon in the following chapters.

Handicapped speech differs from what might be called normal speech. The analytical or linquistic difference is not considered in this paper. Handicapped speech can be understood by people with some effort, and with some additional effort machines can be designed which will likewise recognize handicapped speech.

CHAPTER TWO

BACKGROUND AND RECENT DEVELOPMENTS

> "Nature, as we often say, makes nothing in vain, and man is the only animal whom she endowed with the gift of speech. And whereas mere voice is but an indication of pleasure or pain, and is therefore found in other animals, the power of speech is intended to set forth the expedient and inexpedient, and therefore the just and the unjust. And it is a characteristic of man that he alone has any sense of good and evil, of just and unjust, and the like, and the association of living beings who have this sense makes a family and a state".
>
> Aristotle, Politics.

Communication, particularly over distances, has been a technical preoccupation of man over the ages. "The ancient Greeks are known to have used intricate systems of signal fires which they placed at judiciously selected mountains to relay messages between cities." [9] Quoting further from Flanagan, "History records other efforts to overcome the disadvantages of acoustic transmission. In the sixth century B.C., Cyrus the Great of Persia is supposed to have established lines of signal towers on high hilltops, radiating in several directions from his capital. On these vantage points he stationed leather-lunged men who shouted messages along, one to the other. Julius Caesar reportedly used similar voice towers in Gaul." Our progress has been substantial, and technology promises that this progress will continue.

One of the applications in speech research is the design of a computer and accompanying software that will accomplish mechanical translation from one or several languages to a given language. Taube states that, "it is tacitly assumed that a one-to-one correspondence exists between the language of the original text and that of the translation. If this assumption is correct then it is possible to envisage a purely mechanical process--in the broad sense--which if applied to the input text will result in an output translation, and which if reapplied to the translation will reproduce the original input test" [31]. It is clear that in general this one-to-one correspondence does not exist. Mechanical translation is further complicated by the polysemy of words and contextual considerations.

"No purely automatic procedure is available or is in view that would enable presently existing (non-learning) computers to resolve the polysemy of the word 'pen' in such sentences as 'the pen is in the box,' and 'the box is in the pen,' within the same contexts that would enable a human reader (or translator) to resolve it immediately and unerringly [31]."

Taube notes that as of 1961 the expenditure of research in mechanical translation, exclusive of machine costs, was approximately $3,000,000 per year. This amount would have been sufficient to hire 300 full-time translators at $10,000 per year, or to train 300 translators at a $10,000 training cost per translator. Over the years this would produce a reasonably large body of competent translators--especially if it is added to the present annual expenditure for training in languages. Furthermore training in languages may remove the

need for translation.

The above represents a research effort involving speech that is at best on the outskirts of man's ability to solve using the computer. However, other objectives, some of which are even broader in scope and benefit, fall well within the realm of reasonable and productive research efforts. The remaining overview is based on White's paper [35].

Isolated word recognition has been shown to be realizable. Some examples of systems currently in use follow:

1. Data entry by quality control inspectors (Owens-Illinois Corp. uses it for inspection of T.V. faceplates; Ford Motor Company for assembly line inspection of cars; Continental Can for inspecting pull ring can lids; Tecumseh products for compressor inspection; Union Carbide for manipulating nuclear products at Oakridge, Tenn.; a semiconductor manufacturer uses it with microscopes, inspecting microscopic components);
2. Control of materials handling equipment (United Airlines for baggage handling; Kresge for control of package routing system);
3. Special purpose computer programming (a manufacturer uses isolated word recognition to program automatic machine tools); and
4. Editing of financial information (EMI, Ltd., England uses it to collect numerical data from a variety of sources to prepare monthly financial statements) [35].

The other extreme, that is, real-time, continuous speech recognition systems accepting unrestricted vocabularies with unknown speakers are not as easily realized with today's technology and knowledge. Continuous speech recognition/understanding systems limited in vocabulary size and number of speakers and requiring carefully pronounced speech on a particular topic are within the realm of possibility. The Advanced Research Projects Agency of the U.S. Department of Defense (ARPA) is sponsoring a $15 million five-year speech understanding project to accomplish this intermediate problem of recognition/understanding with a limited vocabulary. The research has been performed at Bolt, Beranek and Newman; Carnegie-Mellon University; Lincoln Laboratory of MIT; Stanford Research Institute; Systems Development Corp.; Haskins Laboratories; Speech Communications Research Laboratory; Sperry-Rand; and the University of California at Berkeley. The goal of the ARPA speech understanding project is to achieve continuous speech understanding for a one-thousand word vocabulary for a specific task (e.g. making airline reservations) for a small number of speakers in near real-time on a "big" computer (e.g., PDP-10).

Technical Considerations

Explicitly or implicitly, spectral analysis is involved in speech recognition. This may be in terms of encoding of speech using autocorrelations of the amplitude variations of speech, in terms of linear predictive coding, zero crossing statistics, spectral analysis, per se, or other analytical techniques.

The initial parametric representation of speech may contain redundant or irrelevant information that can be safely removed without requiring recognition of any speech sounds--this is data compression. Data compression reduces the computational load for all subsequent processing. Formant tracking is the most common approach to speech data compression. "Formant tracking is the monitoring of time evolution of the major peaks of the power spectrum of speech. The formants are produced by the resonances of the vocal cavity." [35] Speech spectrography has shown that the patterns of formants is the dominant feature of speech spectrograms.

Linear predictive coding has made formant tracking a favorite technique among speech researchers. Fast Fourier transforms, as a counter example, requires considerably more computational power. Also, zero-crossing techniques are less efficient in general because of the great variety in the speech wave.

Formant tracking is often used for word recognition. However, the recognition of subpatterns in speech may be more promising for continuous speech recognition. When subpatterns can be recognized, only the name of the subpattern need be saved, and the detailed data representing the subpattern can be discarded.

The similarity between the machine's word prototypes and an unknown word can be measured using correlation functions, filter functions, and geometric distance functions. Geometric functions, as a case in point, operate in N-dimensional spaces defined by N parameters of parametric representations of speech, and are typified by Euclidean, Chebyshev, and Hamming distance functions. For example, the similarity

between two steady state sounds can be defined to be inversely proportional to the Euclidean distance between the sounds, where the coordinates of each sound are defined to be its average pass band energies in a bandpass filter bank.

The approach to word recognition of exhaustively matching all prototypes is inefficient and can perhaps be improved by using subword prototypes. In considering a more efficient scheme a look at phoneme recognition is appropriate. Spoken words can be represented as strings of phonemes, a basic sound unit of speech. In spoken English about 38 phonemes (16 vowels and 22 consonants) are typically used. The two basic advantages in phonemic recognition are that phonemes make possible (1) selective recall of word prototypes and (2) reduction of memory requirements to store word prototypes (data compression). Selective recall reduces the number of prototypes that need to be processed, and data compression reduces memory requirements.

Recall of Word Prototypes. In the word recognition schemes described above, templates are serially processed until the correct template is found. Theoretically, a much more efficient process would be to find some sort of key that could alter the order in which templates are processed so that the correct template would more likely be processed earlier.

Phonemes might provide such a key. The idea is to recognize phonemes first and use them to form words which then cause likely word templates to be processed. This is a "hypothesize and test" paradigm. A rough classification decision is made on the basis of phonemic spelling and this rough classification is verified by resorting to template

matching. The computational expense of recognizing phonemes is small relative to that required for word recognition because there are many more words than phonemes, and phonemes are smaller than words. If phoneme recognition were perfect, then there would be no need to match word prototypes at all; properly spelled English words could be retrieved directly from memory using a dictionary indexed with phonemically spelled words--or English could be learned phonemically. However, phonemic strings are often full of errors, so it is necessary to be able to resort to template matching to verify tentative spellings.

Consider how phonemic encoding achieves data compression. The general principle is that data compression results from storing the names of phonemes themselves. The original data presumably could represent many more phonemes than need to be distinguished by having different names. More specifically, if a given phoneme is a common element in a group of patterns, then data compression is achieved by storing the detailed information about that phoneme only once and then replacing the phoneme by its name wherever else it occurs. The name of the phoneme can then be used to find the detailed information about the phoneme whenever necessary.

The field of pattern recognition has many mechanisms for generating features and transforming their representation. Whatever the strategy for generating intermediate features, the functions of a feature must be to remove redundancy and irrelevant information from speech. This makes the ultimate classification decisions easier since the information that must be analyzed presumably has less noise.

Recent Advances

Some recent advances in speech recognition are [34].

1. Linear predictive coding (LPC) for data compression and spectral smoothing (Atal, Markel, Marloul, Itakura)
2. Vocal tract area function and vocal tract length from LPC model of vocal tract (Wakita)
3. Linear predictive residual as a speech sound similarity measure for steady-state sounds (Itakura)
4. Dynamic programming for time alignment of unknown and reference utterances (Saito et al.,Itakura, White and Neely)
5. Cepstral and Cosh distance functions for speech sound similarity measurements (Markal and Gray)
6. Markov models for representing the time evolution of speech understander's state of knowledge (Baker)
7. Dimensionality reduction for speech data by principal component analysis shows that parametric representations other than formant frequencies and bandwidths are just as adequate for the representation of speech data (Pols, et al.)
8. Pitch tracking by center clipping, infinite peak clipping, and autocorrelation analysis (Dubnowski et al.)
9. Convex hulls in speech energy profiles for syllable segmentation (Mermelstein); and
10. Beginning and ending of utterance detection by amplitude, zero crossing, and back-tracking (Rabiner and Sambur)

Linear predictive coding (LPC), a form of Wiener filtering, is a very important development in automatic speech recognition. Its primary use is to represent speech in a highly compressed form and to produce spectral smoothing that clearly reveals formants. The power of LPC does not lie in the accuracy of its representation of speech or speech spectra but in that it is a computationally efficient means of getting formant peaks and in that it produces a compact representation of speech.

Linear predictive coding implicitly imposes a model of a nonbranching acoustic tube on speech. Wakita [33] derives the important result that both the cross-sectional area and the length of the vocal tube can be simply calculated along with the LPC coefficients. By normalizing a random speaker's vocal tract shape and length to some standard value, a major step forward is achieved toward independence from the need to retrain recognition systems for new speakers.

The linear predictive residual (LPR) is the error that remains when a linear predictive filter is applied to a time series representation of speech. This is also known as "matched filtering" in pattern recognition. Itakura [12] gives a computationally efficient formula for calculating the log ratio of two linear predictive residuals. He proceeds to show experimentally that such a ratio gives an excellent speech sound similarity measure when the unknown residual is used in the denominator and different template sounds are used in the numerator. The Itakura similarity function makes it possible to achieve significant computation economies in implementation of word recognition by making it possible to perform all preprocessing and similarity

measurements relatively simply in the time domain.

Dynamic programming is a well-known technique in operations research which, like LPC, has only recently made an impact on speech recognition. It is an extremely useful technique for achieving nonlinear time adjustment (warping) to align multiple syllabic utterances. Although the technique had been used prior to 1970 in Japan, Russia, and France, the speech community in the U.S. did not start using it until 1974 when Itakura introduced it along with the linear predictive residual during his stay at Bell Labs. The power of dynamic programming was revealed more forcefully when White and Neely showed that, on the one multisyllabic vocabulary they tested, a 20-fold reduction in error was obtained with dynamic programming compared to the best linear time alignment strategies known to them.

The CEPSTRAL and COSH distance functions were proposed and tested as time domain similarity measures for speech sounds by Markel and Gray [10, 15]. Their properties make them slightly preferrable to the Itakura LPR measure.

Dimensionality reduction with principal components analysis is a standard technique in pattern recognition. It is essentially the same as the Kurhunen Loeve technique (K.S. Fu). Pols [26] was able to use principal components analysis to identify Dutch vowel sounds as accurately as could be done by hand analysis of formant frequencies and bandwidths. This demonstrated that entirely automatic techniques are capable of performing as well at identifying vowels as laborious hand analysis. Dimensionality reduction with principal components analysis can actually improve recognition as well as reduce data

rates for representing speech.

Pitch tracking by center clipping, infinite peak clipping, and autocorrelation analysis is important for its accuracy and simplicity. The accuracy of this approach is reported to be as good as the CEPSTRAL approach--meaning that it is as good as the best known. Peak and center clipping produce a speech wave with only three states: -1, 0, 1. This reduces autocorrelation analysis to addition and subtraction, and it brings about a dramatic reduction in the computational burden associated with pitch tracking [8].

Syllable segmentation with convex hulls is a novel approach to an important problem. Almost every speech researcher tries his hand on syllable or phoneme segmentation at one time or another. Paul Mermelstein [19] has proposed a simple algorithm that will work on any continuous approximately convex function and will find the important dips and valleys. He applies his "convex hull" strategy to speech energy profiles and does a good job of segmenting syllables in speech. It would be misleading to suggest that the use of convex hulls solves the segmentation problem in general. The most reliable way to segment speech is to recognize its constituent sounds, and even this approach has errors. The value of convex hulls is their simplicity and the fact that good syllable segmentation is obtained most of the time.

Detecting the beginnings and endings of utterances is an old problem that must be faced by every speech recognition system builder. The solution presented by Rabiner and Sambur [27] is one of the simpler and more accurate ones in the literature.

Fundamental Problems

The fundamental problems of speech recognition are exciting because they are general, and their solutions will have an impact upon all aspects of machine perception. The fundamental problems arise from the fact that the information needed to identify a speech sound is often not spread over a large time interval; that is, it is often a function of context. This phenomenon is extensive; it occurs frequently in all types of speech sounds: in phonemes, in syllables, in words and even in sentences. This is an example of a fundamental problem in artificial intelligence in which local ambiguity is inherent in the data and can be removed only with information from global sources. Speech research is contributing to the solution of this general problem. White [35] illustrates the nature of local ambiguity and methods for dealing with it through the example of phonemes.

If the approximately 38 phonemes of general American English could be recognized accurately, then it would be relatively easy to recognize an unlimited vocabulary of English words. Unknown speech would first be converted to strings of phonemes which would be converted to standard orthography by looking up words in a dictionary based on phoneme spelling. This approach has considerable appeal.

Phoneme recognition has been attempted with and without feedback from the context surrounding the phoneme. Those attempts which have used only local information have failed except for highly artificial speech. The information needed to identify a phoneme is often not present in the phoneme itself. For example, if vowels are preceded by

the consonants "l" or "r", the vowels are changed so much that they are quite often misclassified if taken out of context. Phonemes are strongly affected by neighboring phonemes because of the physical inertia of the tongue and other articulators. This inertia is exacerbated in people with motor problems. Evidently speech is composed of a series of sound targets that speakers usually fail to reach but with no loss of intelligibility because listeners can properly interpret the gestures toward the sound targets by using contextual information.

Attempts to circumvent the local phonemic ambiguity problem lead to the development of phonological rules governing the effects of phonemes on neighboring phonemes. Phonological rules represent a type of syntactic rule for phonemes (instead of words) telling what sequences of phonemes are legal. Phonological rules are usually intended to cope with commonly occuring coarticulation phenomena. It does not appear to be practical to design phonological rules to deal with erratic personal speaking idosyncracies and mumblings. There are nonetheless very real sources of acoustic ambiguity with which people have no trouble coping. So, we look further than phonological rules in our search for a solution to the "local acoustic ambiguity" problem.

Attempts to avoid the local acoustic ambiguity inherent in phoneme use leads to the use of larger units such as syllables or words. The use of larger units is dictated by the fact that larger units tend to have smaller amounts of internal ambiguity. Syllables are more robust than phonemes, and words are more robust than syllables. A significant source of experimental evidence demonstrating that larger models promote accuracy comes from the fact that the most accurate speech

recognition systems known today use utterance prototypes rather than smaller units. For instance, White and Neely [36] were able to achieve 99.6% correct recognition for a 91 word vocabulary using utterance models (templates). This is significantly better than any system based on a smaller unit. However, such high recognition scores apply only to isolated utterance recognition where all acoustic information necessary to identify a word is present in the word itself. Continuous speech often contains words that are acoustically ambiguous, which people recognize easily with contextual information but which machines using only word templates would fail to recognize. So, the ultimate solution is not to be found with the use of utterance-sized units.

It is becoming increasingly clear that there is no single speech unit, nor is there a single set of rules that satisfactorily avoids all commonly occuring acoustic aberrations in speech. The solution to the problem seems to be approachable only asymptotically through the use of ever increasing numbers of large units, models, rules and other sources of knowledge. A significant challenge to artificial intelligence is the need to combine large numbers of models and rules.

Model is used to mean the same thing as "source of knowledge". The problem is how to use models differing in reliability and computation expense to optimize efficiency. Because large models are typically more expensive to store and use, it is White's [35] opinion that larger units should be used in a "feedback" mode and would become involved only by the partial recognition of smaller units. This is based on the extremely important idea that the recognition of computationally cheap sound units can eliminate the need to recognize some computationally expensive sound

units.

A solution to the fundamental problem of local acoustic ambiguity is to have models of speech sounds at all levels (phoneme, syllable, word, phrase) and a strategy allowing lower models to call on higher models to resolve local ambiguities and a strategy allowing the higher models to call on lower models to request further analysis. The higher level models are needed for conservation of computational resources (both memory and computation).

Note that higher level models may be embodied in rules as well as templates. For instance, phonological rules and language syntax perform the function of removing local ambiguity by reference to a larger context. A particularly interesting set of rules is that governing the application of "check morphemes"--suffixes, prefixes, or articles--added to words or inserted in phrases which require agreement in person, gender, and temporal reference between words.

Examples that illustrate the above ideas follow. To start, let us consider a detailed example of problems encountered in attempting to use models for words only. In this example, no models are allowed for phonemes or syllables or other subword units. Word prototypes are encoded with representations derived from signal processing techniques.

According to White, the best signal-processing speech compression techniques known today require approximately 1000 bits per second to produce marginally intelligible speech. Given the typical computational requirements to match prototypes and unknowns, it is possible to show that general purpose computers are two orders of magnitude too slow to match an unknown utterance against several thousand utterance prototypes

in real time. The memory-to-processor bandwidth is too small and the processor speed itself is insufficient. It is also true that there is not enough storage in most random access memories to store templates for several thousand words if the templates are stored in the original parametric representation of the signal processor. Thus, if word templates are to be encoded in terms of signal-processing representation, they must be stored on disks and used sparingly. In other words, the use of word prototypes may produce high accuracy, but encoding the prototypes in the representation produced by the signal processor requires so much storage and processor power that this approach must be limited to isolated word recognition for small vocabularies and to machines with significant computational power.

There are two general ways to reduce the disparity between data processing needs and abilities. One way is to rely on faster hardware. The other way is to use artificial intelligence/pattern recognition techniques to optimize performance of existing hardware. Artificial intelligence/pattern recognition techniques can be used in conjunction with a number of "sources of knowledge" ("intermediate models" or features") so that computationally inexpensive knowledge controls the application of more expensive knowledge.

Improving System Performance

White [35] asserts that ultimate improvements in speech recognition will not result from more accurate identification of short speech sounds, but that the major gains will come from being able to resolve local acoustic ambiguity with information arising from larger speech segments.

The important problems seem to be how to represent and use sources of knowledge arising from global environments.

There are at least four categories of mechanisms for reducing the disparity between data rate processing needs and abilities: (1) special purpose hardware, (2) improved speech data compression techniques, (3) robust speech representation through normalization techniques, and (4) dictionary compression and directed search through the use of hierarchy of speech subunits. In this latter category, information from larger speech intervals is allowed to affect the resolution of local acoustic ambiguity in speech subunits. In giving estimates of performance improvement, White used the performance of an isolated utterance system using utterance prototypes and an exhaustive search of the dictionary of prototypes as a standard of comparison.

Special Purpose Hardware. An order of magnitude increase in data rate processing ability can be expected from new computer architecture by incorporating higher memory-to-processor bandwidths and special purpose processors. Alternative approaches include the use of parallel processing, custom LSI, and/or optical computers.

Speech Encoding and Data Compression. Speech compression is not likely to reduce data rates by more than a factor of four. According to White, the information theoretic minimum mean number of bits required to transmit speech information, including the speaker's emotional state, identify, and semantic context, is probably between 100 and 200 bits per second. So speech compression is already within an order of magnitude of its theoretical limit.

Improved Representation Through Normalization Techniques. Whatever

prototypical speech units are used, the representation for them can be made less variable by normalization techniques. The use of these techniques means essentially that fewer templates or special rules are needed to represent a speech sound faithfully. This means that less processing needs to be done to match any unknown to a prototypical sound. These techniques include normalization by (1) velocity, (2) amplitude, (3) time, (4) speaker spectra, (5) dynamic range, and (6) noise subtraction.

1) Velocity normalization is the shortening of steady state spectra segments to remove artificial variations in sound duration due to variations in speaking rate. ("Velocity" refers to the time rate of change of the spectra).

2) Amplitude normalization is the removal of speech amplitude as a parameter in speech sound similarity measurement. This ensures that a sound that varies in energy but not in its spectral composition is still interpreted as the same sound.

3) Time normalization is the stretching or shrinking of the length of time elapsed between given speech segments. The goal is to align the time of occurance of unknown speech events relative to reference speech events to see how well they match. Dynamic programming is an excellent way of achieving this.

4) Speaker spectra normalization is the transformation of the power spectral density function in order to remove the effects of differing vocal tract length. This is required only for systems that attempt

to recognize speech from different speakers without retraining. It can be achieved by using the LPC model as shown by Wakita [32] or by using bandpass filtered data as shown by Pols et al.

5) Dynamic range normalization is the determination of the energy variations of the speech in order to adjust thresholds to allow energy to be used in segmentation and segment labeling.

6) Noise subtraction normalization is the determination of the energy of ambient noise and the subtraction of that energy from the input signal so that only the speech signal is left.

Improved Dictionary Compression and Search Strategies. Dramatic increases in data processing capabilities can be expected from using a hierarchy of speech sound subunits to promote dictionary compression and "directed search" dictionary retrieval strategies. Two orders of magnitude improvement can be expected for a vocabulary of 1000 words or more when the baseline system uses exhaustive search. The basic idea of directed search is that the identification of computationally cheap sound units makes the more expensive units easier to find in a dictionary. Computationally more expensive sound units are those that require more bits to store and more processing to match. White calls these "larger units". When the correct larger units are recalled from memory, the quality of the match between the larger units and the unknown speech will reveal which of the retrieved units is correct.

For instance, the partially correct spelling of a word unit in terms of phonemic subunits might make it possible to find the word in a dictionary without an exhaustive search. An imperfect match dictionary retrieval method could be achieved by changing the spelling of the word

until it hits an entry in the dictionary. Control of spelling changes could come from phonological rules, secondary spellings produced by acoustic analysis, or reference to more general sound classes.

By using a hierarchy of sound units, the effective reduction in recognition time can be expected to be approximately a log function of vocabulary sizes rather that a linear function. Thus, the relative savings will be larger for larger vocabularies.

Hierarchal organization provides contextual constraints by virtue of the fact that every subunit has the context provided by the larger units above it in the hierarchy. This contextual information can control the operation of pattern classifiers to achieve significant improvements in classifier efficiency. Properly applied contextual information may be the most promising method to increase the apparent computational power of pattern recognizing machines.

Finally, hierarchically organized speech subunits and "directed search" are techniques that permit utilization of contextual information to control processing. These techniques are general and provide an example of techniques developed for speech recognition that apply to a great many problems in pattern recognition.

CHAPTER THREE

PATTERN RECOGNITION

(An Overview)

Even with normal speech, that is speech not affected by motor problems, the variations in utterances are significant. The human brain is capable of determining the most likely utterance on the basis of the acoustic wave, the context of the speech, the facial expression and body gestures. All of these signals are present to some extent in the communication of the cerebral palsy victim, but they are grossly modified, thus making the recognition process more difficult for humans.

Accepting the experience of others as a guide [2], [14], twelve to sixteen linear predictor coefficients per speech segment are optimum for formant analysis around 10 KHz. Assuming in the worst case that this number of coefficients might be necessary for pattern recognition, it may be desirable to find a means of reducing the dimensionality of the pattern vectors by extracting as much meaningful information as possible from the pattern vectors in order to optimize the pattern recognition process. This "feature selection" process would make "learning" a computationally slower process. Preliminary investigations indicate that considerably fewer predictor coefficients are required, and feature selection will not be used initially.

Pattern recognition can be subdivided into three major categories: deterministic, statistical and syntactical. Syntactical will not be used here. Deterministic and statistical require that discriminant or decision functions be determined that will allow the dichotomization of pattern classes. Let $\omega_1, \omega_2, \ldots, \omega_m$ be designated as the m possible pattern classes to be recognized, and let

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_m \end{bmatrix}$$

be the feature measurement vector where x_i represents the $i^{\underline{th}}$ feature measurement. Then, the discriminant function $D_i(\bar{x})$ associated with pattern class ω_i, i = 1, ..., m, is such that if the input pattern represented by the feature vector $\bar{x}$ is in class ω_i, denoted as $\bar{x} \in \omega_i$, then the value $D_i(\bar{x})$ must be the largest. That is, for all $\bar{x} \in \omega_i$,

$$D_i(x) > D_j(x), \quad i, j = 1, \ldots, m, \quad i \neq j \tag{3-1}$$

Thus, in the N dimensional feature space Ω_x, the boundary or partition, called the decision boundary, between regions associated with classes ω_i and ω_j, respectively, is expressed by the equation

$$D_i(\bar{x}) - D_j(\bar{x}) = 0 \qquad (3\text{-}2)$$

Deterministic Pattern Recognition

Deterministic pattern classifiers are those whose decision functions are generated from training patterns by means of iterative "learning" algorithms. Once a type of decision function has been specified, the problem becomes the determination of the coefficients. Deterministic algorithms are capable of "learning" the solution coefficients from the training sets whenever these training pattern sets are separable by the specified decision functions. Deterministic algorithms are developed without making any assumption concerning the statistical properties of the pattern classes. On the other hand, statistical algorithms attempt to approximate $p(\omega_i/\bar{x})$, the conditional density function of class ω_i, which can then be used as Bayes [10],[23],[32] decision functions.

The Perceptron Approach. The basic perceptron or linear error correction model is an implementation of a linear decision function. The response of the machine is proportional to the weighted sum of the associative array features; that is, if we let x_i denote the ith feature and w_i the corresponding weight for that feature, the response is given by

$$D = \sum_{i=1}^{n+1} w_i x_i = \bar{w}^{(i)} \cdot \bar{X}, \text{ where } \bar{w}^{(i)} = \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_{n+1} \end{bmatrix} \qquad (3\text{-}3)$$

For the two class case, if $D > 0$, the pattern belongs to the class ω_1, if

$D < 0$, it belongs to ω_2.

For multiclass problems, where M is the number of classes, responses $D_1, D_2 \ldots, D_m$ are observed, and the pattern is assigned to class ω_i, if $D_i > D_j$ for all $j \neq i$.

The perception algorithm is stated as follows: [22]

1) From the training pattern set and with an arbitrarily selected set of weight vectors, calculate $D_i(\bar{x}) = \bar{w}^{(i)} \cdot \bar{x}$ for $i = 1, \ldots, M$, where $\bar{w}^{(i)}$ is the weight vector for class ω_i.

2) If a pattern $\bar{x}$ belonging to category is presented to the machine with the result that some decision function, say the jth $(i \neq j)$ is larger than the ith, the weight vectors for the ith and j^{th} decision functions are then modified by the addition and subtraction respectively of the pattern vector $\bar{x}$. Let the ith and j^{th} weight vectors prior to modification be denoted by $\bar{w}^{(i)}$ and $\bar{w}^{(j)}$ respectively. The adjusted weight vectors $\bar{w}^{(i)'}$ and $\bar{w}^{(j)'}$, where the prime indicates an adjusted vector, are then

$$\bar{w}^{(i)'} = \bar{w}^{(i)} + c\bar{x} \qquad (3\text{-}4)$$

$$\bar{w}^{(j)'} = \bar{w}^{(j)} - c\bar{x} \qquad (3\text{-}5)$$

All other weight vectors remain unchanged.

The coefficient c is the correction increment and can be any of the following:

a) a positive constant so that the distance moved by a discriminant function toward a particular decision surface is always the same.

b) a fraction so chosen that the distance moved toward a deci-

sion surface is some fixed fraction of the original distance of the weight vector from the decision surface; that is

$$c = \frac{\lambda|\bar{w}^{(i)} \cdot x|}{\bar{x} \cdot \bar{x}}, \quad \lambda = 0, 1 \text{ or } 2 \qquad (3\text{-}6)$$

In this instance, the initial weight vectors are non-zero.

For $\lambda = 0$, the weight point is not moved.

For $\lambda = 1$, the weight point is moved to the pattern decision surface.

For $\lambda = 2$, the weight point is reflected across the pattern decision surface to a point an equal distance on the other side.

Piecewise Linear Classifiers. [23] This is an extension of the perceptron classifier. A piecewise linear classifier consists of R banks of subsidiary discriminators with each bank corresponding to one of the pattern classes. That is, in the perceptron approach, one discriminator is used to adjust weight vectors for all classes; whereas, in this instance, several discriminators for groups of classes perform the adjustments. The term "subsidiary" is used because the bank of discriminators can be thought of as one large discriminator. A pattern class is presented to the machine and the values of all of the subsidiary discriminants are calculated. The pattern is then placed in the class corresponding to the bank containing the highest valued subsidiary discriminant.

The weight vectors are calculated and determined in the same manner as in the perceptron approach. The difference in training has to do with the creation of subclasses. In this research, pairs of

classes were used to determine the subsidiary discriminant functions. Larger subdivisions require more computation time for pattern matching without the benefit of lower recognition error.

The Potential Function Approach. If sample pattern points are likened to potential energy sources, the potential at any of these points attains a peak value and then decreases at any point away from the sample pattern point $\bar{x}_k$. Using this analogy, we may visualize the presence of equipotential contours which are described by a potential function $K(\bar{x},\bar{x}_k)$. For patterns in pattern class ω_i, we may imagine that the cluster of sample patterns forms a "plateau" with the sample points located at peaks of a group of hills. The plateaus of the various classes are separated by valleys in which the potential is said to drop to zero. The potential functions dichotomize the pattern hyperspace and can, therefore, be considered decision functions.

The potential for any sample pattern point can be characterized by the expression

$$K(\bar{x},\bar{x}_k) = \sum_{k=1}^{\infty} \lambda_i^2 \phi_i(\bar{x}) \phi_i(\bar{x}_k) \qquad (3\text{-}7)$$

where $\phi_i(\bar{x})$, $i = 1, \ldots, m$, are orthonormal functions over the region of definition of the patterns. The λ_i, $i = 1, \ldots,$ are real numbers different from zero and chosen in such a way that the potential function $K(\bar{x},\bar{x}_k)$ is bounded for $x_k \varepsilon\ \omega_1 U \omega_2 U \ldots U \omega_m$.

Potential functions are computed successively as patterns are presented. The cumulative potentials at the kth iterative step are

determined by the aggregate of individual potential functions. This cumulative potential, which will be denoted $k_i(\bar{x})$, where i is the number of iterations, is determined in such a way that if the training pattern $\bar{x}^i_{\ell+1}$ is incorrectly classified, the cumulative potential is modified. If the pattern is classified correctly, the cumulative potential is unchanged at this step. To clarify, the cumulative potential is simply the adjusted potential after several iterations.

Orthonormal and Orthogonal Functions. For multivariate functions, the orthonormality condition in vector form is expressed as

$$\int_{\bar{x}} u(\bar{x})\phi_i(\bar{x})\phi_j(\bar{x})d\bar{x} = \delta_{ij} \tag{3-8}$$

where for n variables, the weighting function is $u(\bar{x}) = u(x_1,x_2 \ldots, x_n)$ $\phi_i(\bar{x}) = \phi_i(x_1,x_2, \ldots, x_n)$, $\int_{\bar{x}}$ denotes the multiple integral,

$$\int_{x_1=a}^{b} \int_{x_2=a}^{b} \cdots \int_{x_n=a}^{b}, \text{ and } \delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases} \tag{3-9}$$

However, the functions are used in their orthogonal form because the orthonormal form is numerically more complex. It should be noted that theoretical developments of potential functions require that the orthonormal form be used.

A complete system of orthogonal functions of n variables, $x_1 \ldots x_n$ may be constructed as follows: Groups of n functions from the one-variable set are multiplied together after proper substitution of the variables $x_1,x_2, \ldots, x_n$. If the original functions are orthogonal in the interval $a \leq x \leq b$, the resulting n-variable functions $\phi_1, \phi_2, \ldots$ are orthogonal over the hypercube $a \leq x_j \leq b$, j = 1, 2, ..., n. For example,

the functions of a multivariate set with n = 4, are formed as follows:

$$\begin{aligned}
\phi_1(\bar{x}) &= \emptyset_1(x_2)\emptyset_1(x_2)\emptyset_1(x_3)\emptyset_1(x_4) \\
\phi_2(\bar{x}) &= \emptyset_1(x_1)\emptyset_1(x_2)\emptyset_1(x_3)\emptyset_2(x_4) \\
\phi_3(\bar{x}) &= \emptyset_1(x_1)\emptyset_1(x_2)\emptyset_2(x_3)\emptyset_1(x_4) \\
\phi_4(\bar{x}) &= \emptyset_1(x_1)\emptyset_1(x_3)\emptyset_3(x_4)\emptyset_2(x_4) \\
\phi_5(\bar{x}) &= \emptyset_1(x_1)\emptyset_2(x_2)\emptyset_1(x_3)\emptyset_1(x_4)
\end{aligned} \tag{3-10}$$

The Legendre, Laguerre and Hermite Polynomials constitute three sets of polynomials well-suited to pattern recognition. They are easy to generate and they satisfy the Weierstrass approximation theorem, which states that any function which is continuous in a closed interval $a \leq x \leq b$ can be uniformly approximated within any prescribed tolerance over that interval by some polynomial.

The orthogonal Legendre polynomial functions may be recursively generated by the equation

$$(k + 1)P_{k+1}(x) - (2k + 1)xP_k(x) + kP_{k-1}(x) = 0, \quad k \geq 1 \tag{3-11}$$

where $P_o(x) = 1$ and $P_1(x) = x$. These functions are orthogonal within the interval $-1 \leq x \leq 1$, and they are orthogonal with respect to the weighting function $u(x) = 1$.

The Laguerre polynomials may be generated using the recursion relation

$$L_{k+1}(x) - (2k+1-x)L_k(x) + k^2 L_{k-1}(x) = 0, k \geq 1 \qquad (3\text{-}12)$$

where $L_o(x) = 1$ and $L_1(x) = 1-x$. These polynomials are orthogonal in the interval $0 \leq x < \infty$ with respect to the weighting function $u(x) = e^{-x}$.

The Hermite Polynomial functions are generated by the recursion relation

$$H_{k+1}(x) - 2xH_k(x) + 2kH_{k-1}(x) = 0, \; k \geq 1. \qquad (3\text{-}13)$$

where $H_o(x) = 1$ and $H_1(x) = 2x$. These functions are orthogonal with respect to $u(x) = e^{-x^2}$ on the interval $-\infty < x < \infty$, hence the range of variables is of no concern with these polynomials.

The algorithm for the potential function case is as follows:

At the beginning of training, the initial cumulative potentials $k_o^{(1)}(\bar{x})$, $k_o^{(2)}(\bar{x}), \ldots k_o^{(m)}(\bar{x})$ are assumed to be zero. The superscripts indicate the class membership. Suppose that at the (k + 1)st iterative step a sample pattern $\bar{x}_{k+1}$ belonging to class ω_i is presented. If

$$K_k^{(i)}(\bar{X}_{k+1}) > K_k^{(j)}(\bar{x}_{k+1}) \text{ for all } j \neq i \qquad (3\text{-}14)$$

the potentials are not changed, that is,

$$K_{k+1}^{(i)}(\bar{x}) = K_k^{(i)}(\bar{X})_1; \; i = 1,2,\ldots,M \qquad (3\text{-}15)$$

However, if $\bar{x}_{k+1} \in \omega_i$ and for some ℓ

$$K_k^{(i)}(\bar{X}_{k+1}) \leq K_k^{(\ell)}(\bar{X}_{k+1}) \qquad (3\text{-}16)$$

then the following corrections are made

$$K_{k+1}^{(i)}(\bar{x}) = K_k^{(i)}(\bar{x}) + K(\bar{x},\bar{x}_{k+1}) \qquad (3\text{-}17)$$

$$K_{k+1}^{(\ell)}(\bar{x}) = K_k^{(\ell)}(\bar{x}) - K(\bar{x},\bar{x}_{k+1})$$

$$K_{k+1}^{(j)}(\bar{x}) = K_k^{(j)}(x),\ j = 1,\ldots,L,\ \ldots,\ M,\ j \neq i,\ j \neq 1.$$

The decision functions are the potential functions and are denoted as $D_k^{(i)}(\bar{x})$.

Statistical Pattern Recognition

"By means of statistical considerations it is possible to derive a classification rule which is optimal in the sense that, on an average basis, its use yields the lowest probability of committing classification error."[10] The Bayes classification rule,

$$D_i(\bar{x}) = p\,(^{\bar{\omega}}i/\bar{x}) = \frac{p(\bar{\omega}_i)p(^{\bar{X}}/_{\omega_i})}{p(\bar{x})},\ i = 1,\ldots,m \qquad (3\text{-}18)$$

sets the standard of optimum classification performance and this is the basis of statistical formulations for pattern classification algorithms.

In equation (3-18), $p(^{\omega}i/\bar{x})$ is the conditional density function of class ω_i. $P(\omega_i)$ is the a priori probability of class ω_i, $p(^{\bar{x}}/_{\omega_i})$ is the probability density function of $\bar{x}$ when $\bar{x}$ belongs to ω_i and $p(\bar{x})$ is the probability density function of $\bar{x}$.

Regression Functions. Stochastic approximation methods are employed to find the roots of a regression function. If the regression function represents the derivative of a properly formulated criterion function, finding the root of the derivative function yields the minimum of the criterion function.

Let $g(w)$ be a function of w having a single root $\hat{w}$ so that $g(\hat{w}) = 0$.

Assume that g(w) is negative for all values of w less than $\hat{w}$ and positive for all values of w greater than $\hat{w}$. Most functions not satisfying this condition can be made to do so by multiplying by -1.

Consider that instead of g(w) we are able only to observe noisy values of g(w), denoted h(w). The error between the true values and the noisy observation at any point w is given by g(w) - h(w). It is assumed that h(w) is unbiased, that is

$$E\{h(w)\} = g(w), \tag{3-19}$$

and that the variance of the observation h(w) from g(w) should be finite for all values of w; that is,

$$\sigma^2(w) = E\{[g(w) - h(w)]^2\} \tag{3-20}$$

It is assumed that $\sigma^2(w) < L$ for all w, where L is a finite, positive constant. This latter assumption precludes observations so far from the true value of g(w) that the root seeking procedure would never be able to recover. In other words, the noisy observations should be reasonably well behaved.

With the above assumption, the Robins-Monro algorithm [23],[10] can be used to seek the root $\hat{w}$, of the function g(w). If w(1) represents the initial, arbitrary estimate of $\hat{w}$, and w(k) the estimate at the kth iterative step, the Robbins-Monro algorithm updates the estimate according to the relation

$$w(k+1) = w(k) - \alpha_k h[w(k)] \tag{3-21}$$

where k is the interation count and α_k is a member of a sequence of positive numbers satisfying the conditions

$$\lim_{k \to \infty} \alpha_k = 0$$

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad (3\text{-}22)$$

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty$$

An example is the harmonic series

$$\alpha_k = \{1/K\} = \{1, \tfrac{1}{2}, 1/3, \ldots\}$$

The Robbins-Monro (R-M) algorithm makes corrections on the estimates which are proportional to the previous observations h [w(k)]. Large overcorrections are avoided by assuming that g(w) is bounded by a straight line on either side of the root. The bounding function that is used is given by

$$|g(w)| < A|w - \hat{w}| + B < \infty \qquad (3\text{-}23)$$

where A is the slope of the lines and $\pm$ B are the values of g(w) just to the right and left of $\hat{w}$ respectively. From the figure below, it is evident that as long as the root lies in some finite interval, the existence of an A and B which will satisfy expression (3-23) can always be assumed.

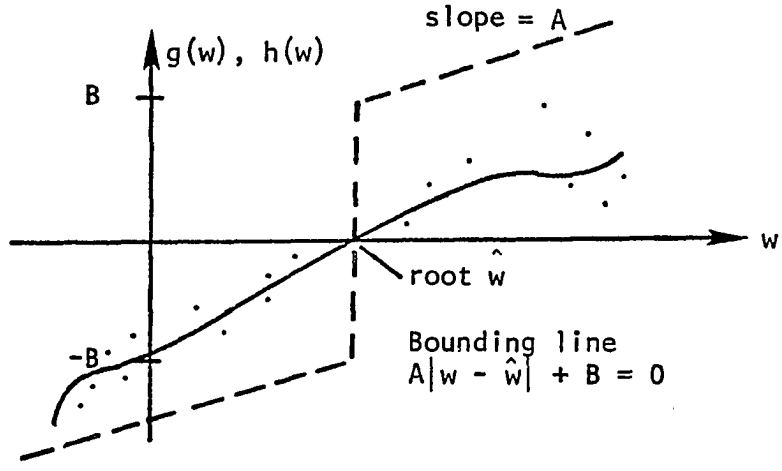


Figure 3-1

With the conditions of (3-19), (3-20), (3-22) and (3-23) being satisfied, the R-M algorithm converges to $\hat{w}$ in the mean-square sense, that is,

$$\lim_{k \to \infty} \{E|w(k) - \hat{w})^2|\} = 0 \qquad (3\text{-}24)$$

Expression (3-24) says that as the number of iterations approaches infinity, the variance of the estimate w(k) from the root $\hat{w}$ will approach zero.

The decreasing significance of the correction factors α_k with increasing k, the number of iterations, has the effect of decreasing the magnitude of the adjustment with successive iterations. Since any sequence $\{\alpha_k\}$ satisfying Equation (3-22) must decrease with increasing k, the R-M algorithm is generally slow to converge. To accelerate convergence, α_k should be kept constant during steps in which h[w(k)] has the same sign. This procedure is based on the fact that changes in the sign of h[w(k)] tend to occur more often in the vicinity of the root $\hat{w}$. For points far away from the root large corrections are desired. The corrections should be smaller as the root is approached.

For the multidimensional case, we have that the weight vector $\overline{w}$ is $\overline{w} = (w_1, w_2, \ldots, w_n, w_{n+1})$, where w_i are associative weights for features x_2 and w_{n+1} corresponds to an appended 1. It is desired to find the root of a regression function $g(\overline{w})$ from the noisy observations $h(\overline{w})$. With $\overline{w}(1)$ representing the initial (arbitrary) estimate of the root $\hat{\overline{w}}$, and $\overline{w}(k)$ the estimate at the kth iterative step, the multidimensional R-M algorithm updates the estimate according to

$$\overline{w}(k+1) = \overline{w}(k) - \alpha_k h[\overline{w}(k)] \qquad (3\text{-}25)$$

where α_k is the same as described earlier. Satisfying the same conditions as for the two dimensional case, the multidimension R-M algorithm converges in the mean-square sense; that is, if the noisy observations are unbiased, their variation from g(w) is finite, and if the regression function is bounded, then we have, as was the case for equation (3-24),

$$\lim_{k \to \infty} E\{||\bar{w}(k) - \hat{\bar{w}}||^2\} = 0 \qquad (3\text{-}26)$$

and $\text{Prob}\{||\lim_{k \to \infty} \bar{w}(k) - \hat{\bar{w}}||^2\} = 1$, where

$||\bar{w}(k) - \hat{\bar{w}}||^2$ is the magnitude squared of the vector $[\bar{w}(k) - \hat{\bar{w}}]$.

Stochastic Approximation for Decision Function Estimation. The densities $p(\omega_i/\bar{x})$ are to be estimated for implementation of the Bayes' decision functions $D_i(\bar{x}) = p(\omega_i/\bar{x})$, $i = 1, \ldots, M$. The approach taken by Tou and Gonzalez [31] is to expand the decision functions over a set of known functions according to the relation

$$D_i(\bar{x}) = p(\omega_i/\bar{x}) \simeq \sum_{j=1}^{K+1} w_{ij}\phi_j(x) = \bar{w}_i \cdot \bar{\phi}(x) \qquad (3\text{-}27)$$

where

$$\bar{\phi}(\bar{x}) = [\phi_1(\bar{x}), \phi_2(\bar{x}), \ldots, \phi_k(\bar{x}), 1].$$

A vector $\bar{x}^*$ can be defined such that

$$\bar{x}^* = \begin{bmatrix} \phi_1(\bar{x}) \\ \phi_2(\bar{x}) \\ \vdots \\ \phi_k(x) \\ 1 \end{bmatrix} \qquad (3\text{-}28)$$

Tou and Gonzales use several mathematically expedient manipulations to obtain a more useful form of equation (3-27). The argument that

follows is the most acceptable. One of the most commonly used types of generalized decision function is that in which the functions $\{\phi_i(\bar{x})\}$ are of a polynomial form. In the simplest case these functions are linear; that is, if $\bar{x} = (x_1, x_2, \ldots, x_n)'$, then $\phi_i(\bar{x}) = x_i$, with $K = n$. Under this condition we obtain

$$P(\omega_i/\bar{X}) = \bar{w}_i' \cdot \bar{x}$$

where $\bar{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in}, w_{i,n+1})'$.

The only information that is available during training is the class membership of each pattern vector. For each class, let us define a random classification variable, $r_i(\bar{x})$, with the following property,

$$r_i(\bar{x}) = \begin{cases} 1, & \text{if } \bar{x} \in \omega_i \\ 0, & \text{otherwise} \end{cases} \tag{3-30}$$

any distinct values can be used. One and zero are arbitrary.

Since we desire knowledge of $p(^{\omega}i/\bar{x})$ only for classification purposes, let us interpret $r_i(\bar{x})$ as being a noisy observation of $p(^{\omega}i/\bar{x})$, that is,

$$r_i(\bar{x}) = p(\omega_i/\bar{x}) + \eta \tag{3-31}$$

where η is a noise factor which is assumed to have zero expected value, so that $E\{r_i(\bar{x})\} = E\{p(^{\omega}i/\bar{x})\}$. The idea is to seek an approximation to $p(^{\omega}i/\bar{x})$ of the form $\bar{w}_i' \cdot \bar{x}$ by observing values of $r_i(\bar{x})$. Consider the criterion function $J(\bar{w}_i, \bar{X}) = E\{|r_i(\bar{x}) - w_i'(x)|\}$. The minimum of the function is zero, and it occurs when $\bar{w}_i'(\bar{x}) = r_i(\bar{x})$. In other words, the minimum occurs when the pattern $\bar{x}$ is classified correctly. This follows from the fact that $r_i(\bar{x})$ is a known classification variable

during training. Therefore, if $\overline{W}_i\overline{X} = r_i(\overline{x})$ for all patterns of the training set, $\overline{w}_i$ is capable of classifying all of these patterns correctly.

Since it is assumed that $E\{r_i(\overline{x})\} = E\{p(^{\omega}i/\overline{x}\ \}$, $J(\overline{w}_i,\overline{X})$ can also be expressed as $J(\overline{w}_i,\overline{x}) = E\{|p(^{\omega}i/\overline{x}) - \overline{w}_i(\overline{x})|\}$. This equation states that finding the minimum of $J(\overline{w}_i,\overline{x})$ corresponds to finding an average approximation to $p(\omega_i/\overline{x})$. In other words, the approximation is such that the expected value of the absolute difference between the function $p(\omega_i/\overline{x})$ and its approximation is zero.

We are interested in finding the minimum of a function $J(\overline{w},\overline{x})$ which is the expected value of some other function $f(\overline{w},\overline{x})$, that is,

$$J(\overline{w},\overline{x}) = E\{f(\overline{w},\overline{x})\} \qquad \text{and} \tag{3-32}$$

$$\frac{\partial J(\overline{w},\overline{x})}{\partial w} = E\ \frac{\partial f(\overline{w},x)}{\partial w} \tag{3-33}$$

The root of $\dfrac{\partial J(w,\overline{x})}{\partial \overline{w}}$ can now be successively estimated by invoking the R-M algorithm with

$$h[\overline{w}(k)] = \left.\frac{\partial f(\overline{w},\overline{x})}{\partial \overline{w}}\right|_{\overline{w} = \overline{w}(k)} \tag{3-34}$$

Using $\overline{w}\ (K+1) = w(k) - \alpha_k h[\overline{w}(k)]$, we obtain the general algorithm,

$$\overline{w}(k+1) = \overline{w}(k) - \alpha_k \left.\frac{\partial f(\overline{w},\overline{x})}{\partial w}\right|_{\overline{w} = \overline{w}(k)} \tag{3-35}$$

where $\overline{w}(1)$ is arbitrarily chosen.

It is worth emphasizing that the statistical algorithm will converge to the approximation regardless of whether or not the classes are strictly separable or not. The price is the rate with which the statistical algorithm achieves convergence.

Increment-Correction Algorithm. Let the criterion function be given as

$$J(\bar{w}_i, \bar{x}) = E\{|r_i(\bar{x}) - \bar{w}' \cdot \bar{x}|\} \tag{3-36}$$

where

$$r_i(\bar{x}) = \begin{cases} 1 \text{ if } \bar{x} \in \omega_1 \\ 0 \text{ Otherwise} \end{cases}$$

The minimum of $J(\bar{w}_i, \bar{x})$ with respect to $\bar{w}_i$ is achieved when the patterns are classified correctly.

The partial of J with respect to $\bar{w}_i$ is

$$\frac{\partial J}{\partial w_i} = E\{-x \text{ sgn } [r_i(\bar{x}) - \bar{w}_i' \cdot \bar{x}]\} \tag{3-37}$$

where sgn (arg) = 1 or -1 depending on whether or not the argument is greater than zero.

Letting $h(\bar{w}_i) = -\bar{x} \text{ sgn } [r_i(\bar{x}) - \bar{w}_i' \cdot x]$ and substituting into the general algorithm of (3-35) yields

$$\bar{w}_i(k+1) = \bar{w}_i(k) + \alpha_k(\bar{x}(k) \text{ sgn } \{r_i[\bar{x}(k)] - \bar{w}_i'(k) \cdot \bar{x}(k)\} \tag{3-38}$$

$w_i(1)$ is arbitrarily chosen and k is the iteration number. Using the definition of sgn, (3-38) may be written as

$$\bar{w}_i(k+1) = \begin{cases} \bar{w}_i(k) + \alpha_k \bar{x}(k), \text{ if } \bar{w}_i(k)\bar{x}(k) < r_i[\bar{x}(k)] \\ \bar{w}_i(k) - \alpha_k \bar{x}(k), \text{ if } \bar{w}_i(k)x(k) \geq r_i[\bar{x}(k)] \end{cases} \tag{3-39}$$

This algorithm makes an adjustment on the weight vector at every step. This is in contrast with the perceptron algorithm, where a correction is made only when a pattern is misclassified.

The iterative procedure of (3-38) or (3-39) is said to have

converged to an error-free solution when all training patterns of ω_i, $i = 1,\ldots,M$, have been correctly classified. In the strictest sense this means that $\bar{w}_i' \cdot \bar{x} = r_i(\bar{x})$, i.e., $\bar{w}_i \cdot \bar{x} = 1$ if $\bar{x} \in \omega_i$ and $\bar{w}_i' \cdot \bar{x} = 0$ otherwise. In terms of correct recognition, it is sufficient to require that for all patterns of class ω_i, $D_i(\bar{x}) > D_j(\bar{x})$ for all $j \neq i$ where

$$D_i(\bar{x}) = \bar{w}_i \cdot \bar{x} \text{ and } D_j(\bar{x}) = \bar{w}_j \cdot \bar{x}. \qquad (3\text{-}40)$$

When the classes under consideration are not strictly separable with the specified decision functions, we are assured that in the limit the solution will converge to the absolute-value approximation of $P(\omega_i/\bar{x})$, as indicated by the criterion function of (3-36). Since the Bayes decision functions, are identically equal to these probability density functions we are therefore guaranteed an absolute-value approximation to the Bayes classifier.

The Method of Potential Functions. Observed data can belong to either class ω_i or ω_j, but cannot belong to both. In view of this assumption, partition boundaries can be generated to categorize the pattern classes. The major problem of pattern classification lies in the generation of partition boundaries on the basis of the observed sample patterns known to belong to a certain class. It might be that sample patterns taken from different pattern classes do not form disjoint sets. Consequently, no partition boundaries can be generated to completely separate the pattern classes. For each pattern class, only a probability can be determined for assignment to class ω_i or class $\bar{\omega}_i$. The problem of probabilistic classification lies in training the machine to determine correctly the probability that new patterns belong to a par-

ticular pattern class on the basis of individual observations during the training process when the association of the sample pattern and the corresponding classes is given a priori.

With stochastic patterns, the classification of new patterns is based on the set of conditional probabilities $P(\omega_i/\bar{x})$, $i = 1,2,\ldots,M$, which are in effect the recognition functions. If $P(\omega_i/\bar{x}) > P(\omega_j/\bar{x})$ for all $j \neq i$, the new pattern $\bar{x}$ is assigned to class ω_i. The recognition function can be estimated iteratively from the training sample pattern by application of the potential function method. Let the recognition function $P(\omega_i/x)$ be approximated by $\tilde{f}_k(\bar{x})$. The function $\tilde{f}_k(\bar{x})$ is defined as follows

$$\tilde{f}_k(\bar{x}) = \begin{cases} 0 & , \text{if } -\infty < \hat{f}_k(\bar{x}) < 0 \\ \hat{f}_k(x) & , \text{if } 0 \leq \hat{f}_k(\bar{x}) \leq 1 \\ 0 & , \text{if } 1 < f_k(\bar{x}) < \infty \end{cases} \tag{3-41}$$

where

$$\hat{f}_k(\bar{x}) = \sum_{j=1}^{m} c_j(k)\phi_j(\bar{x}) \tag{3-42}$$

In this expansion, the functions $\phi_j(x)$ are given, and $c_{j(k)}$ are unknown coefficients determined during training. The potential function associated with any pattern point x_k is given, as in the deterministic case, by

$$K(\bar{x},\bar{x}_k) = \sum_{j=1}^{m} \lambda_j^2 \phi_j(\bar{x})\phi_j(\bar{x}_k) \tag{3-43}$$

The recursive algorithm for the determination of the approximation function $\hat{f}_k(\bar{x})$ may be stated as follows. Starting with $\hat{f}_o(\bar{x}) = 0$, when a sample pattern $\bar{x}_1$ is presented to the machine, the potential function associated with any pattern point $\bar{x}_k$ is $K(\bar{x},\bar{x}_k)$, and three situations may

arise:

1) If $\overline{x}_1 \in \omega_1$ and $\hat{f}_o(\overline{x}_1) > 0$, or $x_1 \notin \omega_1$ and $\hat{f}o(\overline{x}_1) < 0$, then $\hat{f}_1(\overline{x}) = \hat{f}_o(x)$. In other words if the machine makes a correct classification for pattern $\overline{x}$, $\hat{f}_o(\overline{x})$ remains unchanged. Note that this is a mathematical expedient--the situation cannot occur.

2) If $\overline{x}_1 \in \omega_i$ and $\hat{f}_o(\overline{x}_1) < 0$, then $\hat{f}_1(x) = \hat{f}o(x) + \alpha_1 k(\overline{x},\overline{x},)$

3) If $\overline{x}_1 \notin \omega_i$ and $\hat{f}_o(x_1) > 0$, then $\hat{f}_1(x) = f_o(\overline{x}) - \alpha_1 K(x_1 x_1)$.

After the presentation of all sample patterns to the machine, the potential function associated with $\overline{x}_{k+1}$ is $K(\overline{x},\overline{x}_{k+1})$. If $\overline{x}_{k+1} \in \omega_i$ and $\hat{f}_k(X_{k+1}) > 0$, or $\overline{X}_{k+1} \notin \omega_i$ and $\hat{f}_k(x_{k+1}) < 0$, then

$$\hat{f}_{k+1}(\overline{x}) = \hat{f}_k(\overline{x}), \tag{3-44}$$

If $\overline{X}_{k+1} \in \omega_i$ and $\hat{f}_k(\overline{x}_{k+1}) < 0$, then

$$\hat{f}_{k+1} = \hat{f}_k(\overline{x}) + \alpha_{k+1} K(\overline{X},\overline{X}_{k+1}), \tag{3-45}$$

If $\overline{X}_{k+1} \notin w_i$ and $\hat{f}_k (x_{k+1}) > 0$, then

$$\hat{f}_{k+1}(\overline{x}) = \hat{f}_k(\overline{x}) - \alpha_{k+1} k(\overline{x},\overline{x}_{k+1}) \tag{3-46}$$

The coefficient α_k, $k = 1,2,\ldots$, form a sequence of positive numbers satisfying the conditions

$$\lim_{k \to \infty} \alpha_k = 0 \tag{3-47}$$

$$\sum_{k=1}^{\infty} \alpha_K = \infty \tag{3-48}$$

$$\sum_{k=1}^{\infty} \alpha_k^2 < \infty \qquad (3\text{-}49)$$

The harmonic series satisfies these conditions.

Tou and Gonzalez state that for the range from zero to one, the function $\hat{f}_k(\overline{x})$ converges to the recognition $p(^{\omega}i/\overline{x})$ with increasing k.

CHAPTER FOUR

LINEAR PREDICTIVE CODING OF SPEECH

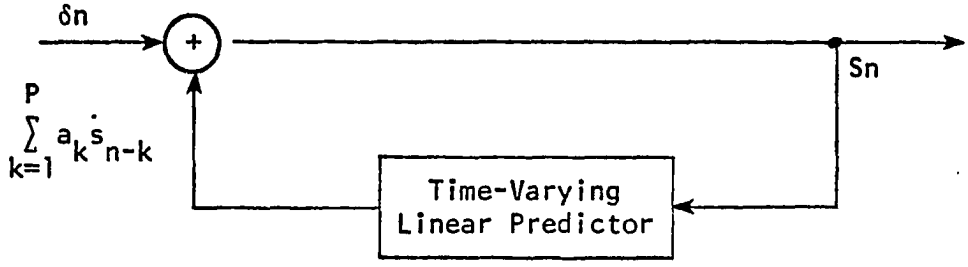
A major concern in speech recognition is the parametric representation of the speech. Linear predictive coding has been highly successful in various speech analysis and recognition projects. Markel's algorithm is straightforward and fast, so that linear predictive coding as formulated by Markel [14] is the basis for the parametric representation used in this project.

Atal and Hanauer [2] state that the vocal tract can be represented as a discrete time-varying linear filter. If it is assumed that the variations with time of the vocal tract can be approximated with sufficient accuracy by a succession of stationary shapes, it then would be possible to define a transfer function in the complex z-domain for the vocal tract.

Nonnasal voiced sounds have no zeros and are therefore adequately represented by an all pole recursive filter. Unvoiced and nasal sounds usually include anti-resonances (zeros) as well as the resonances (poles). The zeros lie within the unit circle in the z-plane; hence each factor in the numerator of the transfer function can be approximated by multiple poles in the denominator. Atal and Hanauer further note that zeros in most cases contribute only to the spectral balance and that an all pole model of the vocal tract can approximate the affect of

of anti-resonances on the speech wave in the frequency range of interest to any desired degree of accuracy.

The combined contribution of the glottal flow, the vocal tract movement and radiation are represented by a single recursive filter as shown below.



The output of the linear filter at the nth sampling instant is

$$s_n = \sum_{k=1}^{P} a_k s_{n-k} + \delta_n, \qquad (4\text{-}1)$$

where the predictor coefficients a_k account for the filtering action of the vocal tract, the radiation and the glottal flow, and δ_n represents the nth sample of the vocal-cord excitation (a pulse). Equation (4-1) says that the value of the speech wave at the nth instant is determined by the past p samples of the speech.

The transfer function of the linear filter is

$$T(z) = \frac{1}{(1 - \sum_{k=1}^{P} a_k z^{-k})} \qquad (4\text{-}2)$$

For stability, there are p poles of T(z) which are real or occur in conjugate pairs.

The number of coefficients p required to represent any speech segment adequately is determined by the number of resonances and

anti-reasonances of the vocal tract in the frequency range of interest, the nature of the glottal volume flow function, and the radiation.

For analysis, Atal and Hanauer suggest twelve coefficients at a sampling rate of 10 KHz. The predictor coefficients a_k, together with the pitch period, the rms value of the speech sample, and a binary parameter to indicate whether the speech is voiced or unvoiced provide an excellent approximation of the speech wave over a time interval during which the vocal tract shape is assumed to be constant. The vocal tract shape is not constant during speech production, so Atal and Hanauer adjust these parameters every five to ten milliseconds.

Historical Development. [14] Speech analysis using the maximum likelihood estimation method was developed by S. Saito and L. Itakura in 1966. In 1968, B.S. Atal and M. R. Schroeder published a method for linear prediction of the speech wave. In 1975, J. D. Markel observed that both of the basic analysis equations, independently developed, were derivable as special cases of R. Prony's method originally formulated in 1795 and extended to a least square formulation as early as 1924 [13]. From the analysis equations, moderate bit-rate speech transmission systems have been developed. Markel's paper [17] shows that the basic analysis approach is transformable into a formant extraction algorithm and, moreover demonstrated that high quality formant trajectory estimation is possible even for the more difficult problem of closely spaced formants and fast transitions. Markel developed an algorithm that is linear, fast and accurate. Because of the speed and accuracy of Markel's algorithm, it is extremely attractive for the task of phoneme, word, phrase or sentence coding.

An Autocorrelation Method of Linear Prediction

In 1973, Markel and Gray published an article containing a recursive solution for the predictor coefficients [14]. Given a set of N data points $\{x_n\}$ not all equal to zero, where $x_n = 0$ for $n < 0$ or $n > N-1$, the correlation sequence for these data will be denoted by $\{r_k\}$ where

$$r_k = r_{-k} = \sum_{n=0}^{N-1-|k|} x_n x_{n+|k|} \tag{4-3}$$

To estimate the discrete spectrum of their autocorrelation sequence, an autoregression filter is defined with a transfer function given by definition as

$$H(z) = \frac{\sigma}{A(z)} \tag{4-4}$$

where σ is a gain term,

$$A(z) = 1 + \sum_{\ell=1}^{M} a_\ell z^{-\ell} = \sum_{\ell=0}^{\Lambda} a_\ell z^{-\ell}, \quad a_o = 1 \tag{4-5}$$

M is the number of coefficients, and a_ℓ are the predictor coefficients. The autoregression filter has a unit sample response that satisfies the equation

$$h_k = \sigma\delta_{ko} = \sum_{\ell=1}^{m} a_\ell h_{k-\ell} \tag{4-6}$$

where δ_{ko} is the Kroneker delta.

Assuming stability of the filter, its autocorrelation sequence can be expressed as

$$\rho_{\ell - n} = \sum_{k=-\infty}^{\infty} h_{k-\ell} h_{k-v} = \sum_{k=-\infty}^{\infty} h_k h_{k+|\ell - v|} \tag{4-7}$$

Assuming causality of the filter ($h_k = 0$ for $k < 0$), the lower limit in each of the sums can be replaced by a finite term rather than minus

infinity.

If equation (4-6) is rewritten as

$$\sigma\delta_{ko} = \sum_{\ell=0}^{m} a_\ell h_{k-\ell},\ a_o = 1 \tag{4-8}$$

multiplied by $h_{k-\nu}$ and summed over all values of k, the result

$\sigma h_{-\nu} = \sum_{\ell=0}^{m} a_\ell \rho_{\ell-\nu}$ is obtained. Since the filter is causal, $h_{-\nu} = 0$ for $\nu > 0$. From equation (4-6) it is seen that $h_o = \sigma$. Thus, equation (4-6) can be rewritten as

$$\sum_{\ell=0}^{m} a_\ell \rho_{\ell-\nu} = 0, \text{ for } \nu > 0 \tag{4-9a}$$

$$\sum_{\ell=0}^{m} a_\ell \rho_\ell = \sigma^2, \text{ for } \nu = 0 \tag{4-9b}$$

Since there are M + 1 parameters in the autoregression filter, a_1 through a_m and σ, a set of M + 1 requirements must be met. In particular, it is required that the first M + 1 values of the filter autocorrelation values match the first M + 1 data autocorrelation values; that is,

$$r_k = \rho_k, \text{ for } K = 0,1,\ldots,M,$$

This requirement along with equations (4-9a) and (4-9b) yields

$$\sum_{\ell=0}^{m} a_\ell r_{\ell-\nu} = 0,\ \nu = 1,2,\ldots,M \tag{4-10a}$$

and

$$\sum_{\ell=0}^{m} a_\ell r_\ell = \sigma \tag{4-10b}$$

Equations (4-10a) and (4-10b) are identical to those obtained in the formulation of the inverse filter and its equivalent problems. Equations (4-10b) gives an expression for a gain term to be used in

approximating the spectrum of the original error.

The coefficients $a_1, a_2, \ldots, a_m$ can be obtained by either solving the set of M simultaneous autocorrelation equations indicated by equation (4-10a) or by minimizing the quadratic form

$$Q = \sum_{\ell=0}^{m} \sum_{\upsilon=0}^{m} a_\ell \, r_{\ell-\upsilon} \, a_\upsilon, \text{ with } a_o = 1 \tag{4-11}$$

Since $\frac{\partial Q}{\partial a_\upsilon} = 0$ gives precisely (4-10a)

Introducing matrix notation, equations (4-10) and (4-11) are rewritten respectively as

$$\underset{m}{\overline{R}} \, \overline{A} = \overline{B} \tag{4-12a}$$

and

$$\overline{Q} = \overline{A}^T \, \overline{R}_M \, \overline{A} \tag{4-12b}$$

where

$$\overline{R}_m = \begin{bmatrix} r_0 & r_1 & \cdots & & r_m \\ r_1 & r_0 & r_1 & \cdots & r_{m-1} \\ \cdot & r_1 & r_0 & & r_{m-2} \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ r_m & r_{m-1} & r_{m-2} & \cdots & r_0 \end{bmatrix} \tag{4-13}$$

$$A^T = [1, a_1, a_2, \ldots, a_m], \text{ and} \tag{4-14a}$$

$$B^T = [\sigma^2, 0, 0, \ldots, 0] \tag{4-14b}$$

Markel and Gray observed that the quadratic forms Q and $\overline{Q}$, expressed in equations (4-11) and (4-12) along with the positive definite property of the covariance or autocorrelation matrix, $\overline{R}$, suggested the introduction of an inner product definition such that Q can be a norm square of a vector in that space. This approach lead to a compact derivation of the recursive solution to equation (4-10a).

The Inner Product Formulation. Let polynomials $F(z)$, $F^{+}(z)$, $G(z)$, and $U(z)$ be defined by the summations

$$F(z) = \sum_{k=-\infty}^{\infty} f_k z^{-k}, \qquad F^{+}(z) = \sum_{k=-\infty}^{\infty} f_k^{*} z^{k}, \qquad (4\text{-}15a)$$

and

$$G(z) = \sum_{k=-\infty}^{\infty} g_k z^{-k} \qquad U(z) = \sum_{-\infty}^{\infty} u_k z\text{-}k \qquad (4\text{-}15b)$$

Where

$F^{+}(z) = F(\frac{1}{z})$, for f_k real and f_k^{*} is the complex conjugate of f_k.

If $R(z)$ is the Z-transform of the correlation sequence $\{r_k\}$, then by application of the Z-transform inversion integral, we have that

$$r_k = r_{-k} = \frac{1}{2\pi j} \int_{\Gamma} R(z) z^{-k-1} dz, \qquad (4\text{-}16)$$

where Γ is a simple closed contour which encircles the origin of the Z-plane. If the unit circle is chosen as the contour, then,

$$r_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} R(e^{j\theta}) e^{-jk\theta} d\theta, \qquad (4\text{-}17)$$

where $j = \sqrt{-1}$. Now, if the inner product is defined as

$$\langle F(z), G(z) \rangle = \frac{1}{2\pi j} \int_{\Gamma} R(z) F^*(z) G(z) z^{-1} dz \qquad (4\text{-}18a)$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} R(e^{j\theta}) F^*(e^{j\theta}) G(e^{j\theta}) d\theta, \qquad (4\text{-}18b)$$

then it is clear that the autocorrelation sequence r_{i-k} can be written as

$$r_{i-k} = \langle z^{-i}, z^{-k} \rangle \qquad (4\text{-}19)$$

Thus, the inner product formulation allows the autocorrelation equation (4-10a) to be rewritten as

$$\sum_{i=0}^{m} a_i \langle z^{-i}, z^{-k} \rangle = 0, \quad k = 1, 2, \ldots, M \qquad (4\text{-}20a)$$

or from linearity and the assumption of real coefficients $\{a_i\}$,

$$\langle \sum_{i=0}^{m} a_i z^{-i}, z^{-k} \rangle = 0, \quad k = 1, 2, .., M, \qquad (4\text{-}20b)$$

or by application of equation (4-5)

$$\langle A(z), z^{-k} \rangle = 0, \quad K = 1, 2, \ldots, M \qquad (4\text{-}20c)$$

Hence, A(z) is orthogonal to z^{-k} for K = 1, 2, ..., M.

Markel and Gray show that equations (4-18) are a valid inner product definition by showing that

1) Conjugate symmetry holds, that is

$$\langle F(z), G(z) \rangle = \langle G(z), F(z) \rangle^*$$

2) Linearity holds, that is

$$\langle cF(z),\ aG(z) + bU(z)\rangle =$$

$$ac^{*} \langle F(z),\ G(z)\rangle + bc^{*}\langle F(z),\ U(z)\rangle$$

where a, b, and c are complex constants, and

3) the inner product has a positive norm, that is

$$\langle F(z),\ F(z)\rangle > 0, \text{ for } F(z) \neq 0$$

Other useful properties are that

$$\langle Z^{-n}F(z),\ Z^{-n}G(z)\rangle = \langle F(z),\ G(z)\rangle \tag{4-21}$$

and

$$\langle F(z),\ G(z)\rangle = \langle 1,\ F^{+}(z)\ G(z)\rangle \tag{4-22}$$

Equations (4-18) are expressed as transform domain relationships. They can also be expressed in the discrete sample domain as

$$\langle F(z),\ G(z)\rangle = \left\langle \sum_{k=-\infty}^{\infty} f_k z^{-k},\ \sum_{\ell=-\infty}^{\infty} z\ g_\ell\ ^{\ell} \right\rangle$$

$$= \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} f_k g_\ell \langle z^{-k},\ z^{-\ell}\rangle$$

$$= \sum_{k=-\infty}^{\infty} \sum_{\ell=-\infty}^{\infty} f_k^{*} g_\ell r_{k-\ell} \tag{4-23}$$

The quadratic Q given in equation (4-11) can be equivalently written in inner product notation as

$$Q = \langle A(z),\ A(z)\rangle \tag{4-24}$$

and the gain term σ^2 given in equation (4-10b) as

$$\sigma^2 = < A(z), 1 > \tag{4-25}$$

It follows also that Q can now be expressed as the norm square

$$\| A(z) \|^2 = < A(z), A(z)> = Q \tag{4-26}$$

of the vector A(z) in the inner product space.

The Recursive Solution

To recursively solve for the vector a(z), start with an inverse filter of degree M and proceed to an inverse filter of degree M + 1 with a final result $A(z) = A_m(z)$. Let $A_m(z)$ represent the inverse filter of degree M

$$A_m(z) = \sum_{k=0}^{M} a_{mk} z^{-k}, \tag{4-27}$$

where $a_{mo} = 1$ for all M. Initially it is assumed that all coefficients a_{mk} are real. It will be later shown that this assumption is justified. By applying equations (4-21), (4-22) and (4-16) to the orthogonality property (4-20), it is possible to obtain a polynomial $B_m(z)$ of degree m + 1 in terms of the polynomial $A_m(z)$ of degree m that is also orthogonal to powers of z^{-1}.

Thus,

$$\begin{aligned} <A_m(z), z^{-\ell}> &= < z^{\ell}, A_m\left(\frac{1}{z}\right) > \\ &= < z^{-m-1+\ell}, z^{-m-1} A_m\left(\frac{1}{z}\right) > \qquad (4\text{-}28) \\ &= 0, \ \ell = 1, 2, \ldots, m \end{aligned}$$

Defining an index $-k = -m-1 + \ell$ and a polynomial

$$B_m(z) = z^{-(m+1)} A_m(1/z). \tag{4-29}$$

a new orthogonality relationship

$$\langle z^{-k}, B_m(z) \rangle = 0, \quad k = 1, 2, \ldots, m \tag{4-30}$$

is obtained. From equations (4-27) and (4-29)

$$B_m(z) = z^{-(m+1)} \sum_{k=0}^{m} a_{mk} z^k \tag{4-31a}$$

or

$$B_m(z) = \sum_{k=1}^{m+1} b_{mk} z^{-k} \tag{4-31b}$$

where $b_{mk} = a_{m,\, m+1-k}$ and $b_{m,\, m+1} = a_{mo} = 1$. At this point, two polynomials $A_m(z)$ of degree m and $B_m(z)$ of degree m+1 have been defined. Each is orthogonal to the powers of z from z^{-1} to z^{-m}. Now, if $A_{m+1}(z)$, a polynomial of degree m+1, can be found such that: (1) the coefficients of z^{-1} are equal to 1, and (2) $A_{m+1}(z)$ is orthogonal to the powers of Z from z^{-1} to $z^{-(m+1)}$, then the recursive procedure will have been solved.

Any linear combination of $A_m(z)$ and $B_m(z)$ will be a polynomial of degree m+1 since $B_m(z)$ is of degree m+1. Any linear combination of $A_m(z)$ and $B_m(z)$ will be orthogonal to the powers of z from z^{-1} to z^{-m} as a consequence of equations (4-28) and (4-30). Since the coefficient of z^o is unity in $A_m(z)$ and zero in $B_m(z)$, the linear combination

$$A_{m+1}(z) = A_m(z) + k_m B_m(z) \tag{4-32}$$

where k_m is some, as of yet, unspecified constant, defines a polynomial $A_{m+1}(z)$ of degree m+1 satisfying the first requirement. To satisfy the

second requirement, it is only necessary to choose k_m so that $A_{m+1}(z)$ is orthogonal to $z^{-(m+1)}$; therefore, defining

$$\alpha_m = < z^{-(m+1)}, B_m(z)> \tag{4-33}$$

and

$$\beta_m = < z^{-(m+1)}, A_m(z)> \tag{4-34}$$

the inner product $< z^{-m-1}, A_{m+1}(z)> = 0$

immediately results in

$$k_m = -\alpha_m/\beta_m. \tag{4-35}$$

By applying equation (4-23), it can be readily noted that if $A_m(z)$ has only real coefficients, then α_m and β_m will be real. Therefore, k_m and finally the coefficients of $A_{m+1}(z)$ will be real.

Since from equation (4-27), $A_o = 1$ is the starting point, the assumption of real coefficients in the derivation was justified. Initial conditions in the recursion are obtained from equations (4-33) and (4-34) as

$$\alpha_o = <z^{-1}, B_o(z)> = <z^{-1}, z^{-1}> = r_o \tag{4-36a}$$

$$\beta_o = <z^{-1}, A_o(z)> = <z^{-1}, z^{o}> = r_1 \tag{4-36b}$$

and

$$a_{oo} = 1 \tag{4-36c}$$

The recursive solution of the autorrelation equations is completely specified at this point in terms of the above initial conditions and the parameters α_m, β_m, and k_m. At step m, knowing k_m, equation (4-32) is used to determine $A_{m+1}(z)$ in terms of $A_m(z)$ and $B_m(z)$. Computationally, from equations (4-27) and (4-31b)

$$a_{m+1,\ell} = \begin{cases} a_{m\ell} = 1 & , \ell = 0 \\ a_{m\ell} + k_m a_{m,m+1-\ell} & , \ell = 1,2,\ldots,m \\ k_m & , \ell = m+1 \end{cases} \qquad (4\text{-}37)$$

Based upon the inner product formulation, it is possible to obtain several different computational expressions for the parameters α_m and β_m in terms of the filter parameters a_{mk}, k = 0, 1, ..., m at recursion m.

First, by applying equations (4-29), equation (4-33) can be equivalently written as

$$\alpha_m = \langle z^{-(m+1)}, z^{-(m+1)} A_m(\tfrac{1}{z}) \rangle$$

$$= \langle A_m(z), 1 \rangle \qquad (4\text{-}38a)$$

By applying equation (4-23), the computational form is equivalent to

$$\alpha_m = \sum_{i=0}^{m} a_{mi}\, r_i \qquad (4\text{-}38b)$$

By applying the orthogonality relationship equation (4-20a), α_m can also be written as

$$\alpha_m = \langle A_m(z), A_m(z) \rangle \qquad (4\text{-}38c)$$

in addition, by applying equations (4-21), (4-22) and (4-29),

$$\alpha_m = \langle z^{-(m+1)} A(\tfrac{1}{z}),\ z^{-(m+1)} A(\tfrac{1}{z}) \rangle$$

$$\alpha_m = \langle B_m(z),\ B_m(z) \rangle \qquad (4\text{-}38d)$$

Computationally, from equation (4-23), equations (4-38c) or (4-38d) are equivalent to

$$\alpha_m = \sum_{i=0}^{m} \sum_{k=0}^{m} a_{mi}\ r_{i-k}\ a_{mk} \qquad (4\text{-}38e)$$

The coefficients α_m can also be calculated recursively. From equation (4-38a) and equation (4-32),

$$\alpha_{m+1} - \alpha_m = \langle A_{m+1}(z),\ 1 \rangle - \langle A_m(z),\ 1 \rangle$$

$$= \langle A_m(z) + k_m B_m(z),\ 1 \rangle - \langle A_m(z),\ 1 \rangle$$

$$= k_m \langle B_m(z),\ 1 \rangle \qquad (4\text{-}38f)$$

But from equation (4-29) and equation (4-22),

$$\alpha_{m+1} - \alpha_m = k_m \langle z^{-(m+1)} A_m(\tfrac{1}{z})\ 1 \rangle$$

$$= k_m \langle z^{-(m+1)},\ A_m(z) \rangle, \text{and hence with equation (4-34),}$$

the computational form is

$$\alpha_{m+1} = \alpha_m + k_m \beta_m, \qquad (4\text{-}38g)$$

or by applying (4-35),

$$\alpha_{m+1} = \alpha_m (1-k_m) \tag{4-38h}$$

By applying equation (4-19) and equation (4-21) to the definition equation (4-34), β_m is obtained as

$$\beta_m = \sum_{\ell=0}^{m} a_{m\ell} \; r_{m+1-\ell} \tag{4-39a}$$

By applying the orthogonality relationship (4-20a), β_m can also be written as

$$\beta_m = \langle B_m(z), A_m(z) \rangle \tag{4-39b}$$

Markel's Algorithm

The recursive solution that gives the predictor coefficients can be described in algorithmic terms as follows: The quantities r_o, r_1, ..., r_m are given. At the completion of steps m, the quantities a_{mo}, a_{m1}, ..., a_{mm}, α_m, β_m have been calculated. To obtain step m+1, k_m is obtained from equation (4-35), the coefficients $\{a_{m+1,k}\}$ are obtained from equation (4-37), and α_{m+1}, β_{m+1} are obtained from equations (4-38) and (4-39). As initial values for m = 0, $a_{oo} = 1$, $\alpha_o = r_o$ and $\beta_o = r_1$. This procedure is carried out until step M is obtained yielding the inverse filter

$$A(z) = 1 + \sum_{k=1}^{m} a_k z^{-k} \text{ where } a_k = a_{mk}, \text{ for } k = 0, 1, \ldots, M.$$

The gain term σ^2 is by inspection of equation (4-24) and (4-38a),

$$\sigma^2 = \alpha_m \tag{4-40}$$

Equation (4-38b) gives a physical interpretation of the α_m term, which is: α_m is precisely the least squares error or energy output from the inverse filter. Also α_m is equal to Q_m, the value of the quadratic from equation (4-11) which is normalized at step m. Thus, if $\{r_m\}$ is a normalized autocorrelation sequence, where $r_o = 1$, α_m will satisfy the relationship $0 < \alpha_m < \alpha_m - 1 <...< \alpha_o = 1$ (since each additional stage must decrease the squared error below that of the previous stage).

$|\beta_m|$ is bounded by the squared error α_m. The recursive gain term k_m has the important physical interpretation that if $|k_m| < 1$, the polynomial $A_{m+1}(z)$ corresponds to a stable filter and if $|k_m| > 1$, $A_{m+1}(z)$ corresponds to an unstable filter. Thus, by use of the recursion procedure for solving equation (4-10a), it is possible to test for stability at any step M without actually having to apply a polynomial root-solving program to see if any roots lie outside the unit circle. If the polynomial $A_m(z)$ is unstable at stage m, all further recursions remain unstable.

Markel also determined that M is not a strong function of the particular speech sound. However, it is a strong function of the system sampling rate. For $6 \leq F_S < 18$ KHz, the equation $M = F_S + \gamma$ where $\gamma = 4$ or 5 and F_S is truncated to a ones or tens decimal place, has been found generally sufficient for the analysis. The physical interpretation of this result is simply that independent of the sampling rate, roughly one complex pole pair is required to span every 700 Hz.

Lumped Linear Prediction

Having presented the above formulation, a radical departure is made from its normal application. Without great effort, normal speech can be produced with few variations; that is, the speech wave for a given utterance is relatively unchanged from one time to another. As a worse case, cerebral palsy victims can produce speech that is understandable to a listener after a "training" period, but their speech wave would contain a great variations for a repeated utterance. This is a function of the lack of motor control that can affect the glottis and the articulators as severely as the limbs. Note that physical or emotional stress has a compounding effect on their motor control and exacerbates speech production.

It was as a consequence of the above considerations, that prototype template matching was considered the least appropriate approach for pattern recognition of abnormal speech. Even though linear predictive coding gives what amounts to a spectral representation of speech, the spectral differences in abnormal speech might be so great as to make template matching futile. The pattern recognition approaches described in Chapter three appeared more suitable for the kinds of spectral variations that were expected.

The difficulty with the standard application of LPC with respect to pattern recognition stems from the fact that the size of the training sets is related (in a nonlinear fashion) to the length of the pattern vectors. Tou suggests that the size of the training sets per class (the number of vectors) should be ten times larger than the pattern

vector length (number of elements per vector). If we sample at 10 Kz and solve for 12 predictor coefficients for 50 segments (200 samples in length) of an utterance, a total of 600 coefficients would be required for that utterance. We could expect that as many as 4000 training vectors per class for pattern recognition would be needed. If this were the case, the solution to the problem would clearly not be practical.

As mentioned earlier, after the wave has been digitized, often pre-emphasized and multiplied by a Hamming window, segments of the speech are scanned one at a time either continuously or in an overlapping manner. These segments are within the range of 100 to 300 samples. A very accurate representation of the speech wave is obtainable from this procedure.

The approach taken here is to lump all spectral information of an utterance by solving for one set of predictor coefficients. That is, instead of solving for a set of predictor coefficients for speech wave segments, a set of coefficients for the entire speech wave is obtained without segmentation. It is not possible to recover spectral information from the inverse filter by doing this. However, one might expect that enough information about an utterance is contained in these lumped coefficients so that coding of the utterance for machine recognition could be accomplished. The major question is whether the utterance is uniquely coded: and if it is not, how large a vocabulary of words can be coded in this manner to make this technique viable. The justification for this approach has been given. The efficacy of the approach can be found through experimentation.

CHAPTER 5

EXPERIMENTAL RESULTS

Experiments

The primary objective of the required experiments was to determine which learning algorithm would perform best for pattern recognition using the lumped linear predictive coefficients described in Chapter 4. Best performance was measured by the rate of convergence of the algorithm and the subsequent recognition error obtained using a given algorithm. Recognition error is the number (percentage) of words that were not recognized during pattern recognition. Incorrect classification of words did not occur.

The experiments were performed as follows. Isolated words were recorded on analog tape and then digitized at approximately 6000 Hz over a two-second interval. A trigger signal was used to initiate analog to digital conversion on one track of the tape and speakers were required to say a word immediately after a trigger signal, thereby assuring that the entire utterance was sampled.

Based on plots of the speech wave (see Figure 5-1) for various words an attempt to isolate the most significant parts of the word was done as follows. The beginning point was determined by searching for a difference between ten consecutive samples that was greater than a

Bracketed numbers indicate sample ranges

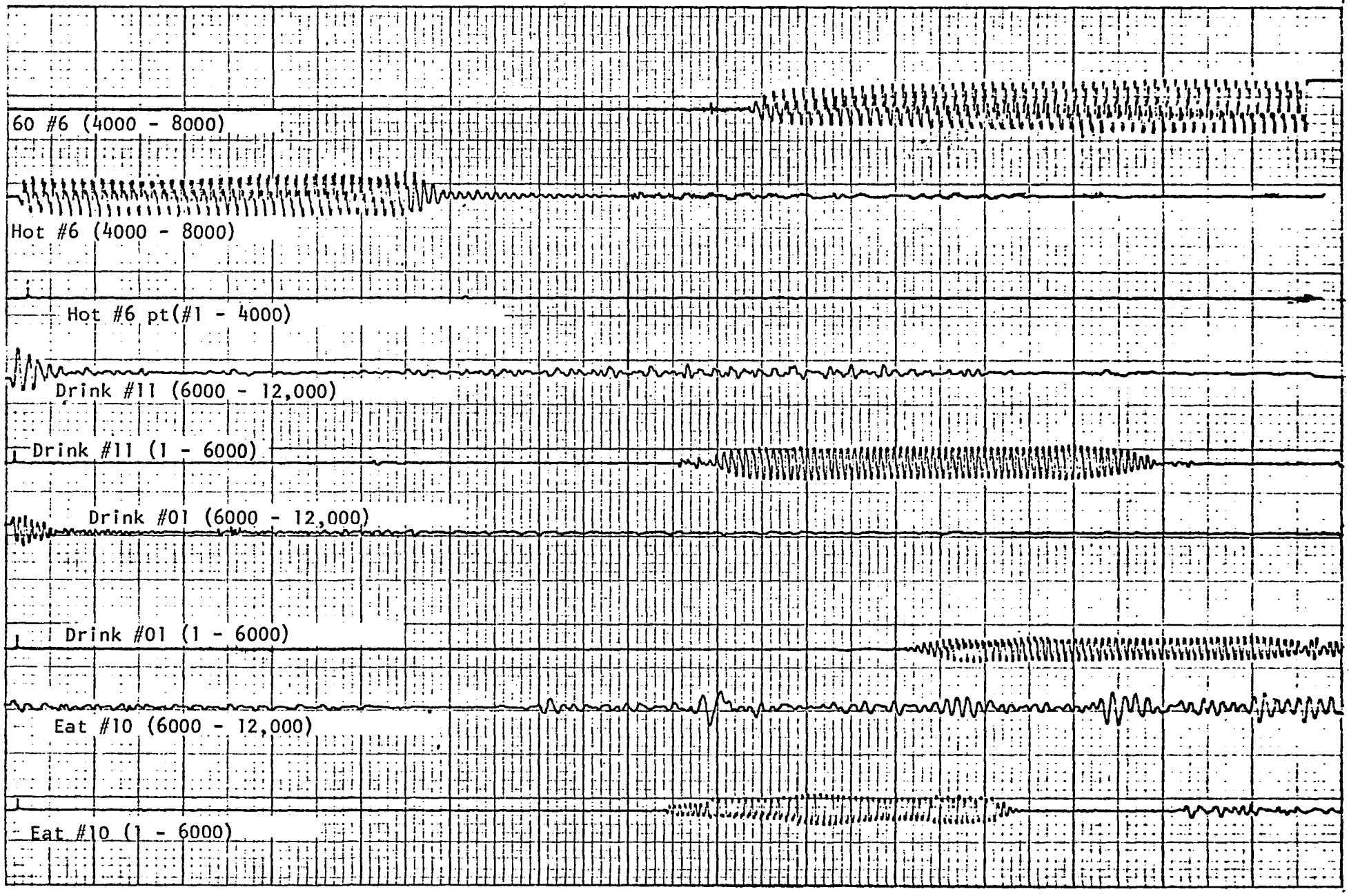


Figure 5-1

preset threshold. The threshold was decided upon after looking at samples whose differences were less than a preset threshold. This procedure does not take into account the following kinds of problems (26).

1. Weak fricatives (/f th, h/) at the beginning or end of an utterance.
2. Weak plosive bursts (/p, t,k/).
3. Final nasals.
4. Voiced fricatives at the ends of words which became devoiced.
5. Trailing off of certain voiced sounds; such as the final /i/ becomes unvoiced sometimes in words like "three" (/th-r-i-/) or "binary" (/b-al-n-e-r-i/).

The end point detection procedure described above was not expected to isolate all of the acoustic or spectral information contained in an utterance, but it was intended to and did isolate the portions of the utterances that would be adequate for these experiments. Again, by observing samples of the data, it was determined that out of the approximately twelve thousand samples per word that approximately nine thousand contained the utterances. The technique employed here isolated between 50% and 100% of these nine thousand samples.

The recordings were made in a closed but not sound-proofed room. Ambient noise came primarily from central air conditioning.

Fifteen enunciations of ten words were used. Ten of these were for training. The other five were for pattern recognition per se.

Eight predictive coefficients were calculated. This number of

coefficients was chosen arbitrarily.

In addition to the above, it should be mentioned that no normalization of the speech waves (see Chapter 2) was done. However, amplitude normalization was done on each set of predictive coefficients. The sets of coefficients were divided by the largest in magnitude of the coefficients in each set as an attempt to accelerate the rate of convergence during training. The relative values of the coefficients with respect to each other in a given set were unchanged, but significant magnitude differences between sets of coefficients for a given word were removed.

The experiments were performed twice. The first, with normal speech, was used to debug the various programs. The second, was a twenty-five year old female afflicted with cerebral palsy. Her speech was similar to that of a person under the influence of a sedative or another kind of depressant drug. Although her speech was somewhat slurred, it was intelligible. The results given below are with this subject's speech.

Words were selected from the training list of words and phrases used by the Cerebral Palsy Center in Norman, Oklahoma. The words are as follows:

hungry

sleep

eat

thirsty

happy

bed

play

go

T.V.

stop

Results

See Chapters 3 and 4 and the appendices for explanations and examples of the various techniques, flowcharts and examples that are applicable to the items below.

Linear Error Correction (LEC). This program was executed without pattern deletion (see below). Convergence to solution weight vectors using this algorithm was not achieved for any of several vocabulary sizes and different word groups. Initial weight vectors that were used included 1) zero-weight vectors, 2) the average of the training vectors for each class, and 3) selected weight-vectors from each class.

It is important to note that by allowing sufficient time convergence using this approach might have ultimately been achieved; for, as will be seen later, just two classes could require over 5000 iterations before convergence was achieved. For multiclass training sets, iterations of this size are for all practical purposes useless.

LEC with Pattern Deletion. This approach greatly accelerated the rate of convergence to solution weight vectors for the multiclass case. In this experiment, patterns that satisfy discriminant functions were deleted from the training classes during execution of the program.

Recalling from Chapter 3, the condition to be satisfied was that for a given pattern $\overline{x}$ belonging to class ω_i,

$$D_i(\overline{x}) > D_j(\overline{x})$$

for all $i, j = 1, 2, \ldots, M$, where $i \neq j$. If a given pattern satisfies this condition, then there is no reason to keep it in the training pat-

tern class, and the total number of training patterns would be reduced as convergence proceeds. The great improvement in the rate of convergence was unexpected. Convergence was achieved in twenty-four iterations for the words "go", "eat", "bed", "happy", "sleep", and "thirsty". The recognition error was 50%. The words that were not recognized were "eat", "bed", and "sleep". This recognition error is considered to result primarily from the size of the training classes.

Piecewise Linear Error Correction (PWL). This approach proved to be more successful. The pattern deletion technique was not applied in this program, but similar benefits are expected when it is applied. Solution weight vectors were sought for all distinct pairs of words in the vocabulary. Convergence was obtained for every word grouping, but with great differences in the rate of convergence for different pairs of words. For instance, "hot" and "go" converged in only nine iterations; whereas, "hot" and "eat" required 745 iterations for convergence. As many as 5,000 plus iterations were required for some of the pairs.

The recognition error varied with the vocabulary size. For eight words, "sleep", "eat", "thirsty", "happy", "bed", "play", "go" and "hungry", the recognition error was 12%. "Thirsty" was not recognized. When "T.V." and "stop" were added to this list, the error increased to as much as 37%. Failure occurred with "thirsty" and "play" in one instance and with "thirsty", "play" and "stop" in another test. The recognition error changed with different enunciations of the same word; for instance, a different enunciation of "happy" increased the error for a ten word vocabulary to 40%.

Fractional Error Correction (FEC). This approach gave results that were similar to those above when applied with PWL. When FEC was applied to LEC, convergence was achieved, but the magnitudes of the weight vectors were extremely large (on the order of 10^{42}). Numbers of this magnitude when applied to discriminant functions renders the functions meaningless.

The Potential Function Approach. Potential functions formed from Legendre polynomials and containing nine terms have the form

$$K(\bar{x},\bar{x}_k) = 1 + \sum_{i=1}^{M} x_i x_{ki},$$

The x_i are coefficients that are determined as training proceeds. Their initial values are arbitrary. The x_{ki} are features of the training patterns. Adjustments to the cumulative potential functions are made according to the same rules that apply to the perceptron algorithm. So for nine terms, there is no benefit in terms of rate of convergence or recognition error to be derived from this approach.

Using additional terms in the polynomials obtained by applying the techniques described in Chapter 3, can only decrease the rate of convergence. This is because the terms would look like $x_i x_{ki} x_j x_{kj}$ and $x_i^n x_{ki}^n$, where i, j = 1, ..., m, $i \neq j$ and n represents some power determined from the recursion relation that generates the Legendre polynomials. These additional terms cannot increase the rate of convergence because the predictive coefficients are between zero and one, so terms similar in magnitude to the original nine terms of the LEC approach would be added to the functions. At best, the rate of convergence of the LEC approach would be equalled, but it would not be increased by using

potential functions.

Increment Correction (Statistical). A program was written and executed, but this approach is considered unnecessary for this problem for the following reasons: adjustments in this and all statistical approaches considered in this study are based solely on a priori knowledge of the class membership of the training patterns. No discriminating conditions are invoked in order to determine whether an adjustment should be made and all weight vectors are continuously adjusted until convergence is achieved. Since the only condition for convergence is that $\overline{w}_{k+1}(\overline{x}) = \overline{w}_k(\overline{x})$, convergence is assured as a consequence of the fact that α_k approaches zero as k, the number of iterations, gets very large (see Chapter 3).

CHAPTER 6

DISCUSSIONS

The Effectiveness of Coding Words Using Lumped Linear Predictive Coefficients

Lumping the predictive coefficients so that one set of eight or more coefficients could code an entire word represents a new approach. Word production can be represented as a series of events in time and those events, using lumped linear prediction, are considered, in aggregate, unique for spectrally different words. This approach was chosen partly because it is computationally attractive and better suited for application in various learning algorithms; that is, the choice of eight or more coefficients for the coding of words offers considerable computational advantages over several hundred coefficients.

Coding of even a limited number of words using this approach, increases the likelihood of developing a useful machine for abnormal word recognition in the near future. Cerebral palsy victims often use a word board that contains useful words and phrases. The numbers zero through nine and the alphabet are also on this board. The user points to the appropriate letter to spell a word or to words and/or phrases to form a sentence. This is done when trying to communicate with someone who does not recognize cerebral palsy speech and when using new words.

This pointing process is tedious, slow and frustrating to both the user and to the interlocator. Using the lumped linear predictive approach of this study, these words can be coded for machine recognition.

Some of the words found on these word boards are the following:

Nouns	Verbs	Places-Things	Prepositions
daddy	go	school	in
mommy	come	home	of
sister	eat	bed	behind
brother	drink	book	through
you		car	beside
it		typewriter	between
I		hot	on
me		cold	over
			under
			below
			up
			down

Comparisons of Various Learning Algorithms

Piecewise Linear Error Correction. Although the best results (7 out of 8 words correctly recognized) were obtained using this algorithm, it is not the preferred learning algorithm. This is because separate weight vectors are generated for every distinct pair of words; that is, for ten words, there would be forty-five weight vectors as opposed to just ten using linear error correction. Clearly, the memory

size and time for pattern recognition would be considerable using this approach.

Linear Error Correction. With the deletion of patterns as training proceeds, this algorithm converges at an acceptable rate. The recognition error was high, but this can be greatly improved by increasing the number of patterns.

It should be noted that the errors using LEC and PWL varied with the test words. This is a further indication that too few training patterns were used. Indeed, if the recommendation of Tou and Gonzalez concerning the size of the training classes with respect to the number of elements in the training patterns had been followed, then eighty training patterns per word would have been used for the eight predictive coefficients. Based on the results of these experiments, it is felt that the recognition error can be greatly decreased without resorting to training classes that are quite this large.

Fractional Error Correction and Potential Functions. These algorithms are considered inappropriate for this study for the reasons given in Chapter 5.

Statistical Algorithms. These offer no real advantage for word recognition. These algorithms are slow to converge because adjustments on weight vectors are made at every step during training. Furthermore, once convergence is achieved, recognition errors can be expected to be higher than with deterministic algorithms because as the number of iterations increases, α_k (see Chapter 3) will become so small that incremental adjustments on the weight vectors will be negligible. As a consequence, the fact that deterministic algorithms converge to

solution weight vectors makes them more suitable for word recognition. It is only when deterministic algorithms fail to converge that statistical algorithms should be used for pattern recognition problems.

End-Point Detection

Extension of this study to a larger vocabulary will require better end-point detection so that all significant parts of the utterance are isolated. The technique that was employed in the experiments of Chapter 5 was adequate for a small and predetermined vocabulary, however, the extension of this study to vocabularies whose words will not be subjected to human scrutiny of the speech wave would require a more powerful algorithm.

Rabiner and Sambur (26) reported the results of their end-point detection algorithm in February, 1975. This algorithm is based on two measures of the speech signal, zero (level) crossing rate and energy. Rabiner and Sambur reported that the algorithm is capable of performing correctly in any reasonable acoustic environment in which the signal-to-noise ratio is on the order of 30 dB or higher.

An important assumption is that during the first 100 ms of the recording (also sampling) interval, there is no speech present. Thus, during this interval, the statistics of the background noise are measured. These measurements include the average and standard deviation of the zero crossing rate and the average energy. The zero (level) crossing rate of the speech is defined as the number of zero (level) crossings per 10 ms interval. The energy, E(n), is defined as the sum of the magnitudes of 10 ms (for a 10 KHz sampling rate) of speech

centered on the measurement interval; that is,

$$E(n) = \sum_{i=-50}^{50} |s(n + i)|,$$

where s(n) are the speech samples. The choice of a 10 ms window for computing the energy and the use of a magnitude function rather than a squared magnitude function were dictated by the desire to perform the computations in integer arithmetic and, thus, to increase computation speech.

Using these definitions, background and signal measurements are made that allow for very accurate isolation of the utterance using Rabiner and Sambur's algorithm. See appendix B for flowcharts.

Other Considerations

The purpose of normalization is to make repetitions of the same word more uniform. As discussed in Chapter 2, there are various normalization techniques. The one that is most readily applicable is amplitude normalization, for it can be accomplished by searching for the largest (magnitude) sample and dividing the speech samples by it. It should be mentioned that this is not necessarily preferred over amplitude normalization of the predictive coefficients, because the latter is considerably faster. Other variational effects can be offset by the size of the training sets.

The 6 KHz sampling rate, which is considered low, was used because of limitations of the equipment. Greater resolution of the speech wave by increasing the sampling rate is required for extending the vocabulary size.

Finally, cerebral palsy victims often move their heads in an uncontrolled fashion; therefore, an attached close speaking microphone would be required in order to minimize signal level variations due to this movement.

Pattern Deletion During Training

During the early stages of this research, it was thought that the time required for convergence to solution weight vectors using linear error correction was excessive. There is no way to predict the required number of iterations for convergence, and there was always the possibility that the patterns were not seperable. As was mentioned previously, different initial weight vectors were used, but there was no improvement in achieving convergence. Close scrutiny of the weight vectors showed that after a number of iterations (this varied with class sizes and different groups of classes) that the weight vectors were fluctuating--apparently around solution weight vectors. Indeed this fluctuation was almost periodic during some computer runs. It should be noted that the test for convergence was done in integer arithmetic. Furthermore, all pattern classes converged to solution weight vectors when the piece-wise linear error correction algorithm was applied. Although this does not imply that as larger groups the classes would be seperable, success with PWL and the fluctuation of the weight vectors using LEC strongly suggested that the classes were seperable. Also, attempts at pattern recognition using weight vectors that were not solution weight vectors always resulted in recognition of some of the wrods. These latter weight vectors were obtained by stopping training after an

arbitrarily selected number of iterations.

The cost and time of allowing training to continue until convergence was achieved would have been prohibitive. Clearly, 5000 or more iterations could not be permitted, and, if there was to ever be practical application of this research (meaning with larger vocabularies) this dilemma had to be resolved. The solution was unexpectedly simple and effective. Various authors suggest removing patterns from the training classes as these patterns demonstrate that discriminant conditions are satisfied. The training classes would be reduced in size and hence training would be accelerated. Indeed training is accelerated, for convergence had never been achieved before for more than three words when LEC was applied. Now convergence for six classes was achieved in twenty-four iterations.

The reason for applying these kinds of learning algorithms was that prototypical templates of words were not being used. For each word, there are significant variations between different enunciations. Now it is to be expected that for every word class there will be come patterns having greater spectral similarity to each other than to the other members of the class, and the codes of these words will bear greater similarity. Hence, as training proceeds, some of the training patterns will satisfy discriminant conditions.

Even with a small vocabulary consisting of only ten training patterns per word, the number of calculations that are performed during a number of iterations is impressive. For every pattern, the discriminant function value must be calculated for the particular pattern with every weight vector. These functions are then compared and weight vector

adjustments are made accordingly. By deleting patterns, the total number of computations is dramatically reduced.

Indeed as the solution weight vectors were approached, the number of calculations per iteration approached numbers like twenty, twelve, ten, etc. as opposed to several hundred. So deletion of patterns during training is a very important procedure for successful application of the algorithms that were discussed in this dissertation.

CHAPTER 7

CONCLUSION

Normal speech production, a series of glottal epochs with articulatory shaping, is produced with normal motor control and can be made in a highly repetitive fashion. Normal speech usually conforms to a given dialect of a given language so that it is readily understood. In contrast, the motor control problems of cerebral palsy victims are often sufficiently severe to inhibit the movements of the glottis and of the articulators, but the afflicted persons can develop the ability to make utterances that approximate those of normal speech through long and intense training. These approximations or attempts at speech production are often so far removed from readily recognizable patterns that communication represents a tremendous problem for cerebral palsy victims. The important consideration is that these attempts at speech production are often consistent enough to allow others to learn to recognize the speech of the persons with cerebral palsy. In other words, if the ability to make reasonable approximations of the words in their vocabulary on a repetitive basis did not exist, then cerebral palsy victims would not be able to communicate verbally.

The most important assumption that was made in undertaking this study was that if normal speech can be coded and recognized by machines,

then the speech of the cerebral palsy victims who are capable of approximating normal speech can also be coded and recognized by machines if their utterances are unique for different words.

Segmentation of the speech wave, although necessary for spectral analysis of speech, is not computationally attractive for machine recognition of speech. Because of variational considerations in abnormal speech and of the computation time required for convergence of learning algorithms, the several glottal events required for speech production were lumped into one spectral event per word in this dissertation by calculating a small set of linear predictive coefficients that would code words for machine recognition. This is lumped linear predictive coding of the speech wave. Limitations in vocabulary size are accepted because of the gains made in memory utilization and computational speed.

Ten words that were spoken by a person with a slight voice handicap were coded and recognized by a computer with a significant degree of success: 50% to 89% recognized. The approaches of this study, lumped linear prediction of the speech wave used with the perceptron algorithm or its variations, may be extended to larger vocabularies by using the techniques described in Chapter 6.

An IBM 370, Model 158, computer was used for this study. If a machine that can be used by the voice handicapped is to be realized, then the approaches used in this study should be applied to a microprocessor. The realizations of such a machine can be the goal of further efforts.

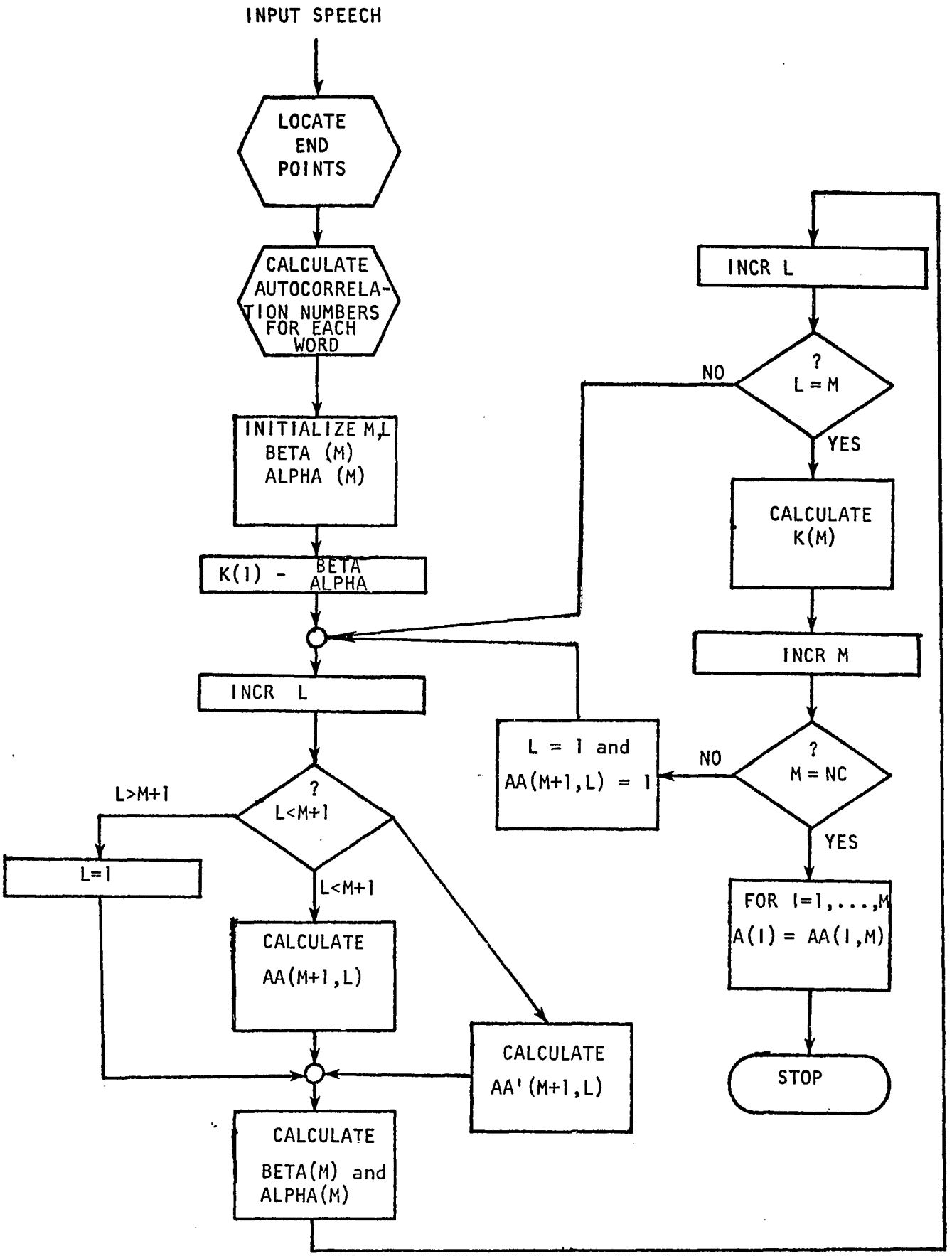
APPENDIX A

FLOWCHART AND PROGRAM FOR LUMPED LINEAR PREDICTION

Explanation of program variables and parameters:

(See Chapter 4 for equations and explanations)

1. The subroutine for calculating the autocorrelation numbers is straightforward; therefore it is not shown on the flowchart.
2. BETA = β
3. ALPHA = α
4. AA(M + 1, L) = $a_{m+1,\ \ell}$
5. AA'(M + 1,L) = $a_{m+1,\ \ell} = k(m)$ when L = m + 1
6. A(I) = a_i, the coefficients predictor
7. NC is the maximum number of coefficients
8. M,L, and I are array indices


INPUT SPEECH
LOCATE END POINTS
CALCULATE AUTOCORRELATION NUMBERS FOR EACH WORD
INITIALIZE M,L
BETA (M)
ALPHA (M)
K(1) - BETA ALPHA
INCR L
? L<M+1
L>M+1
L=1
L<M+1
CALCULATE AA(M+1,L)
CALCULATE AA'(M+1,L)
CALCULATE BETA(M) and ALPHA(M)
L = 1 and AA(M+1,L) = 1
? M = NC
NO
YES
INCR M
CALCULATE K(M)
? L = M
NO
YES
INCR L
FOR I=1,...,M
A(I) = AA(I,M)
STOP

Program

for

Lumped Linear Prediction

```
//GO.FT08F001 DD DSN=LEARNA,DISP=OLD,VOL=SER=003580,LABEL=(1,,,IN),
//  UNIT=2400
//GO.SYSIN DD *
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CTHE FIRST WORD IS HAPPY          C
C THE SECOND WORD IS EAT          C
C THE THIRD WORD IS HUNGRY        C
C THE FOURTH WORD IS DRINK        C
C THE FIFTH WORD IS THIRSTY       C
C THE SIXTH WORD IS BEDRSTY       C
C THE SIXTH WORD IS BED           C
C THE SEVENTH WORD IS SLEEP       C
C THE EIGHTH WORD IS SAD          C
C THE NINTH WORD IS HOT           C
C THE TENTH WORD IS GO            C
C THE ELEVENTH WORD IS PLAY
C THE TWELFTH WORD IS T.V.        C
C THE THIRTEENTH WORD IS STOP     C
C THE FOURTHEENTH WORD IS COLD    C
C THE FIFTEENTH WORD IS DADDY     C
C THE SIXTEENTH WORD IS MOMMY     C
C THE SEVENTEENTH WORD IS BROTHER
C THE EIGHTEENTH WORD IS SISTER   C
C THE NINTEENTH WORD IS LETTER    C
C THE TWENTIETH WORD IS I LOVE YOU
C 'N' IS USED TO INDICATE THE NUMBER OF PARAMETERS
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      INTEGER INDEX1,INDEX2,G,H,X(9000),T,U,F,SAV1,SAV2,DIFF
      INTEGER SAV3, CNT
      REAL K(9)
      DIMENSION R(9), BETA(9), ALPHA(9), AA(9,9), A(8),CARD(20)
      SAV1 = 1; SAV2 = 9000
      T = 1
      U = 1
   50 READ(8,20)CARD
   20 FORMAT(20A4)
      READ(8,21)X
   21 FORMAT(18I4)
      GO TO 2000
      SAV3 = 180
      CNT = 1
   22 DO 200 I = SAV3,8999
      J = I + 1
      DIFF = X(J) - X(I)
      DIFF = IABS(DIFF)
      SAV1 = I
      IF(DIFF.GT.5)GO TO 999
  200 CONTINUE
  999 CNT = CNT + 1
      IF(CNT-3)23,24,24
   23 SAV3 = SAV1
      GO TO 22
   24 I = SAV1
 1000 N = 1
      I = I + 1
      J = I + 1
      DIFF = X(J) - X(I)
      DIFF = IABS(DIFF)
 1001 IF(DIFF-2)1002,1002,1000
 1002 N = N + 1
      I = I + 1
      J = I + 1
      DIFF = X(J) - X(I)
      DIFF = IABS(DIFF)
```

```
1005 IF(N-1000)1001,1003,1003
1003 SAV2 = I
     GO TO 2000
1006 SAV2 = 9000
2000 N = 1
     R(N) = 0.0
     DO 31 M = SAV1,SAV2
     R(N) = (X(M) * X(M)) + R(N)
  31 CONTINUE
     N = 2
   1 H = SAV2-N-1
     R(N) = 0.0
     DO 2 M = SAV1,H
     L = M + N
   2 R(N) = (X(M)*X(L)) + R(N)
     N = N + 1
     IF(N-9)1,1,4
   4 N = 1
     L = 1
     I = 1
     M = 1
     AA(1,1) = 1.0
     BETA(1) =  R(2)
     ALPHA(1) = R(1)
     K(1) = -(BETA(1)/ALPHA(1))
   5 G = M + 1
     AA(G,L) = 1.0
   6 L = L + 1
     INDEX2 = M + 1 - L
     IF(L - G)7,8,9
   7 AA(G,L) = AA(M,L) + (K(M)*AA(M,INDEX2))
     GO TO 6
   8 AA(G,L) = K(M)
     I = I + 1
     GO TO 6
   9 L = 1
     M = M + 1
     INDEX2 = M + 1 - L
     BETA(M) = 0.0
     ALPHA(M) = 0.0
  10 BETA(M) = (AA(M,L)*R(INDEX2)) + BETA(M)
     ALPHA(M) = (AA(M,L)*R(L)) + ALPHA(M)
     L = L + 1
     INDEX2 = M + 1 - L
     IF(L-M)10,10,11
  11 K(M) = -(BETA(M)/ALPHA(M))
  42 IF(M - 8)12,12,14
  12 L = 1
     GO TO 5
  14 WRITE(7,15)CARD
  15 FORMAT(20A4)
     DO 16 I=1,8
     A(I) = AA(8,I)
  16 WRITE(7,17)A(I)
  17 FORMAT(8F9.2)
CCCC
```

```
100 STOP
    END
```

APPENDIX B

FLOWCHARTS OF RABINER AND SAMBUR'S END-POINT LOCATION ALGORITHM

Explanation of program variables

1. A zero crossing threshold, IZCT, for unvoiced speech is chosen as the minimum of a fixed threshold, IF (25 crossings per 10 ms), and the sum of the mean zero crossing rate during silence, $\overline{IZC}$, plus twice the standard deviation of the zero crossing rate during silence; that is,

$$IZCT = MIN(IF, \overline{IZC} + 2\sigma_{IZC})$$

Peak energy, IMX, and silence energy, IMN, are used to calculate the following:

2. I1 is a level that is 3 percent of IMX, (adjusted for the silence energy); that is,

$$I1 = 0.03*(IMX - IMN) + IMN$$

3. I2 is a level that is set to four times the silence energy; that is,

$$I2 = 4*IMN$$

4. ITL, the lower threshold, is the minimum of I1 and I2; that is,

$$ITL = MIN(I1, I2)$$

5. ITU, the upper threshold, is five times the lower threshold; that is,

ITU = 5*ITL

The algorithm for a first approximation of the beginning point location is shown in Figure B-2. The algorithm begins by searching from the beginning of the interval until ITL is exceeded. If the energy falls below ITL before it rises above ITU, a new beginning point is obtained by finding the first point at which the energy exceeds ITL, and then exceeds ITU before falling below ITL; eventually such a beginning point must exist. The ending point is determined in a similar manner and is shown in Figure B-3. The beginning and ending points are labeled N1 and N2, respectively.

N1 and N2 are initial estimates. The algorithm proceeds to examine the interval from N1 and N1-25, a 250-ms interval preceeding the initial beginning point, and counts the number of intervals where the zero crossing rate exceeds the threshold IZCT. If the number of times the threshold was exceeded was three or more, the starting point is set back to the first point (in time) at which the threshold was exceeded. Otherwise, the beginning point is kept at N1. Rabiner and Sambur's rationale behind this strategy was that for all cases of interest, exceeding a tight threshold of zero (level) crossing rate is a strong indication of unvoiced energy.

A similar search procedure is used on the ending point of the utterance to determine if there is unvoiced energy in the interval from N2 to N2 + 25.

PLEASE NOTE:

Some pages have small
and indistinct print.
Best copies available.
Filmed as received.

UNIVERSITY MICROFILMS

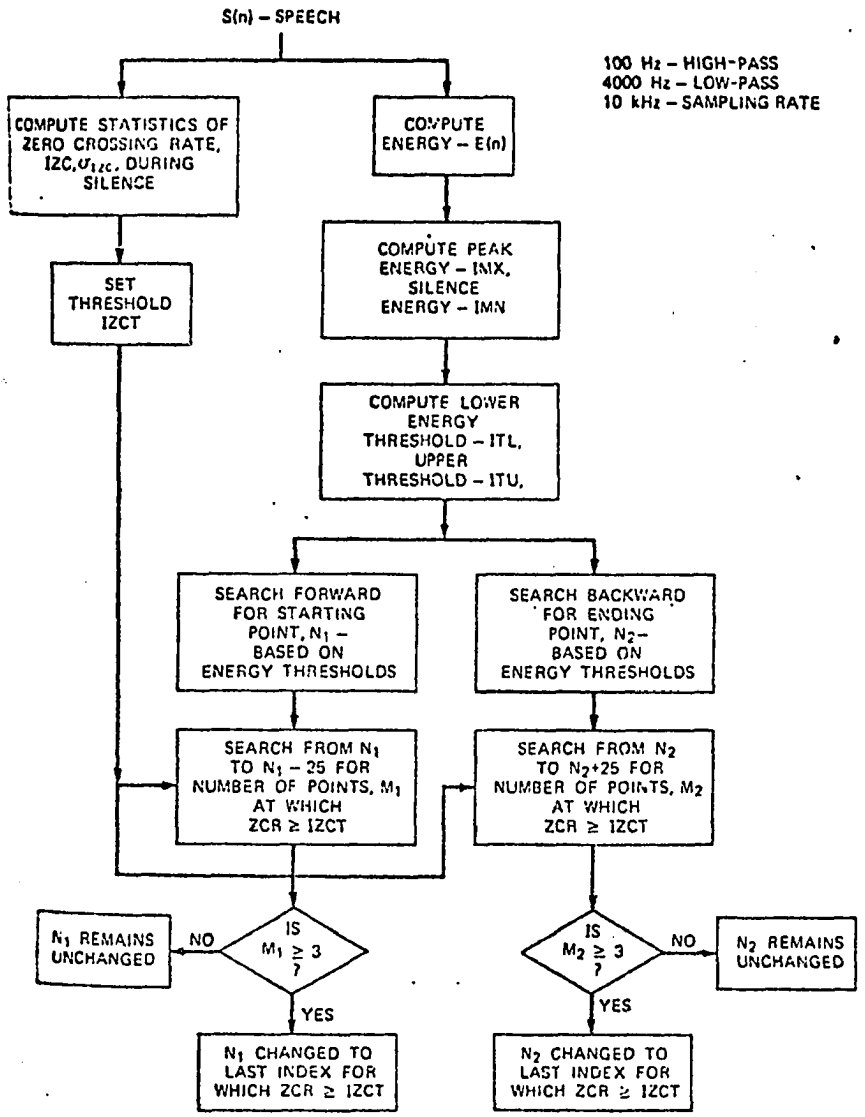


Figure B-1. Flowchart for the Endpoint Algorithm.

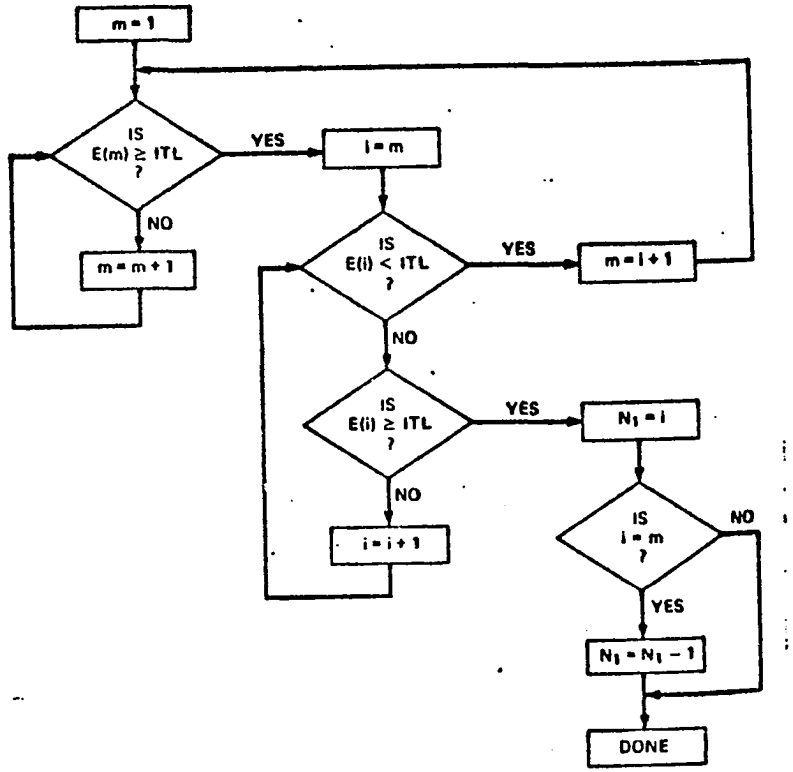


Figure B-2. Flowchart For the Beginning Point Initial Estimate Based on Energy Considerations.

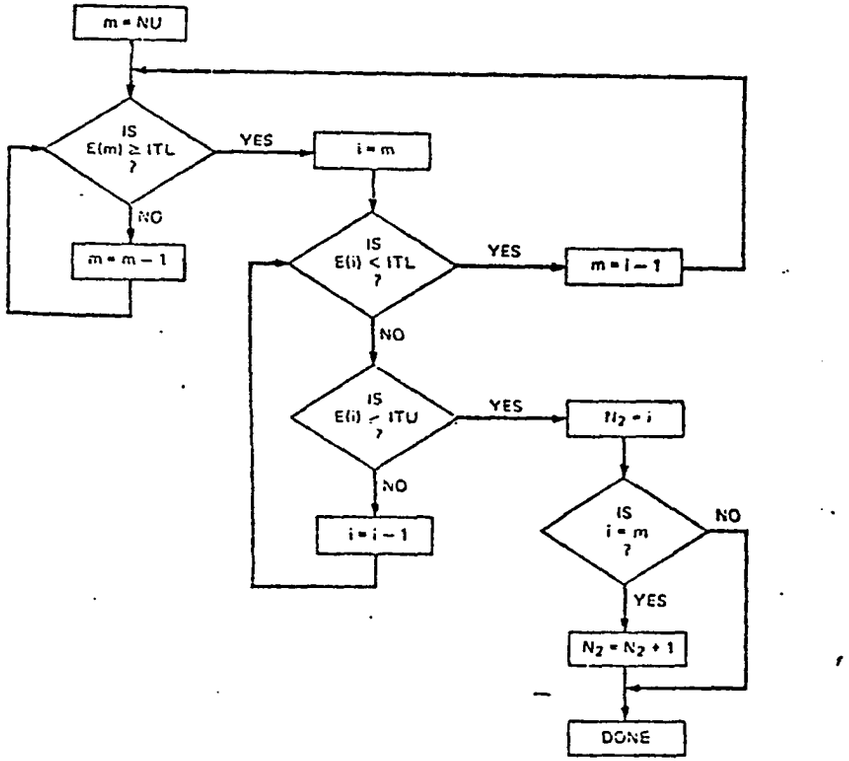


Figure B-3. Flowchart for the Ending Point Initial Estimate Based on Energy Considerations.

APPENDIX C

FLOWCHART AND EXAMPLE PROGRAMS

OF PATTERN RECOGNITION ALGORITHMS

The flowchart that is given on the next page is basically applicable to all of the pattern recognition techniques that were discussed in Chapter 3. For the deterministic algorithms, only the subroutine has to be changed for weight adjustments. For the statistical algorithms, the adjustments of weight vectors are made solely on the basis of a priori knowledge of class membership and so no comparison of discriminant functions is needed in statistical programs.

Explanation of program variables and parameters:
(See Chapter 3 for equations and explanations)

1. Y(I,J,Z) = I classes containing J patterns with Z components in each pattern.
2. W(I,Z) = I weight vectors corresponding to I classes with Z components in each vector.
3. S(I) = I discriminant functions.

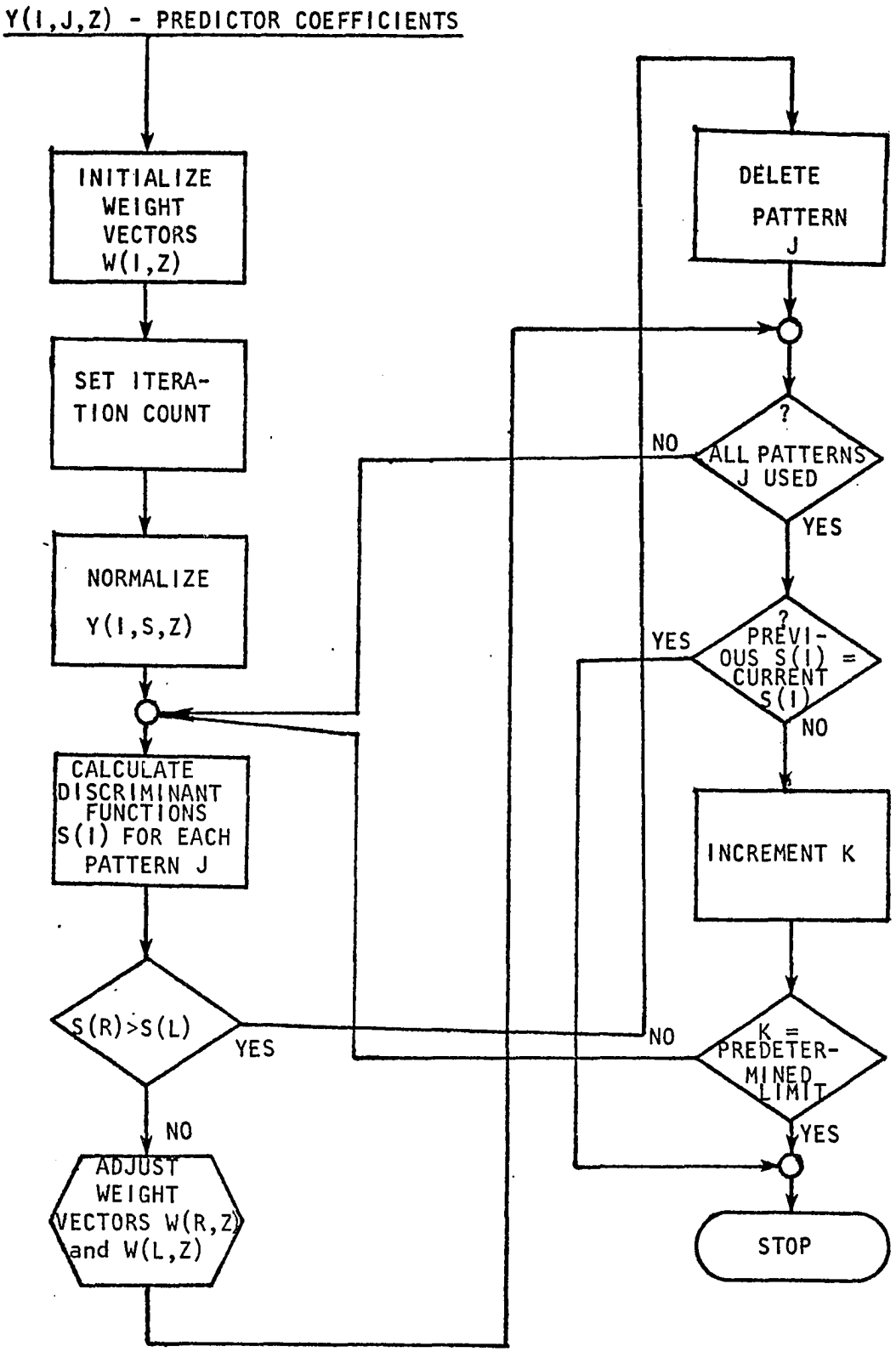


Figure C-1. Flowchart for Linear Error Correction

Program

for

Linear Error

Correction

```
C###################################################################
C   THE FIRST WORD IS HOT      ***************************************
C   THE SECOND WORD IS GO      ***************************************
C   THE THIRD WORD IS EAT      ***************************************
C   THE FOURTH IS SLEEP      *****************************************
C   THE FIFTH WORD IS DRINK      ***************************************
C   THE SIXTH WORD IS SAD      *****************************************
C###################################################################
CEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
C   THESE CALCULATIONS ARE DISTINQUISHED BY THE FACT THAT THE
C   INTEGER VALUES OF THE WEIGHT VECTORS THAT ARE COMPARED ARE
C   MULTIPLIED BY TEN.
      INTEGER F,E1,AY
      INTEGER R,T,U,Z,X,B,TEMP1,TEMP2,TEMP3,Q,E
      INTEGER X12,X11,X13,C,X2,X5,X3,CNT,EO
      DIMENSION Y(20,10,9),S(20),SAV(20,9),SAV1(20,9),IW(20,9)
      DIMENSION ISAV(20,9), ISTORJ(20)
      DIMENSION IDIFF(20,9),D(20,10,9),SUM(20,9),W(20,9),DIM(20,10,9)
      REAL LARG1,LARG2,LARG3
CEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
C  E REPRESENTS THE NUMBER OF CLASSES
      COMMON Y,TEMP1,TEMP2,TEMP3,S,I,J,Z,E,CNT,W,ISTORJ
C   STEPS THROUGH 3996 SET THE INITIAL NUMBERS OF PATTERNS FOR THE
C VARYING CLASS SIZES THAT RESULT FROM DELETING PATTERNS THAT
C   SATISFY DISCRIMINANT CONDITIONS.
      DO 3996 J = 1,10
 3996 ISTORJ(J) = 10
      E = 7
      CNT = 1
      DO 1 X = 1,E
      DO 1 U=1,9
    1 SAV(X,U)=0.0
      READ(5,2)(((Y(I,J,Z),Z=1,8),J=1,10),I=1,E)
    2 FORMAT(F9.2)
C   THIS SETS THE LASTS ELEMENT IN EACH TRAINING PATTERN TO 1.0
      DO 900 I = 1,E
      DO 900 J = 1,10
  900 Y(I,J,9) = 1.0
C
C*******************************************************************
C   THIS ACCOMPLISHES NORMALIZATION
      DO 398 I = 1,E
      DO 398 J = 1,10
  449 K = 2
  450 K1 = K + 1
  397 LARG1 = ABS(Y(I,J,K))
      LARG2 = ABS(Y(I,J,K1))
      IF(LARG2.GT.LARG1)GO TO 451
  452 K1 = K1 + 1
      IND = K
      IF(K1-8)397,397,453
  451 IND = K1
      K = K + 1
      IF(K-8)397,397,453
  453 LARG3 = ABS(Y(I,J,IND))
      DO 455 K = 2,8
      Y(I,J,K) = Y(I,J,K)/LARG3
  455 CONTINUE
  398 CONTINUE
      A = 1000
      GO TO 1000
  999 A = 100
 1000 DO 2000 I = 1,E
```

```
      X1 = Y(I,J,Z)
      IF(X1)151,1002,150
C   THESE STEPS THROUGH 2000 ROUND OFF THE PATTERN VECTORS
  151 X1 = ABS(Y(I,J,Z))*A + 0.00001
      T = X1 *100.0 + 1.0
      T = T + 2
      X1 = T/100.0
      GO TO 1001
  150 X1 = Y(I,J,Z)*A + 0.00001
      T = X1 *100.0 + 1.0
      T = T + 2
      X1 = T/100.0
      GO TO 1003
 1001 X2 = ABS(X1)
      X1 = ABS(X1)
      X3 = (X1 - X2)*10
      IU = X3
 1030 IF(IU-5)1004,1005,1007
 1004 Y(I,J,Z) = -X2/A
      GO TO 2000
 1005 X4 = (X2/A)*10.0
      X5 = X4
       IX6 = (X4-X5)*10
      N = 1
      C = 2*N
 1008 IF(C-IX6)1006,1007,1006
 1006 N = N + 1
      IF(N-4)1008,1008,1009
 1007 X9 = X2/10.0
      X11 = X9
      X12 = (X9-X11)*10 + 1
 1016 X12 = X12 + 1
 1017 X13 = X11* 10 + X12
      IX20 = X13 - X2
      IF(IX20-2)1018,1050,1050
 1050 X13 = X13 - 1
 1018 Y(I,J,Z) = -X13/A
      GO TO 2000
 1009 X8 = X2/A
      Y(I,J,Z)= -X8
      GO TO 2000
 1002 Y(I,J,Z) = Y(I,J,Z)
      GO TO 2000
 1003 X2 = X1
      X3 = (X1 - X2)*10
      IU = X3
 1040 IF(IU-5)1010,1011,1014
 1010 Y(I,J,Z) = X2/A
      GO TO 2000
 1011 X4 = (X2/A)*10.0
      X5 = X4
       IX6 = (X4-X5)*10
      N = 1
      C = 2*N
 1012 IF(C-IX6)1013,1014,1013
 1013 N = N + 1
      IF(N-4)1012,1012,1015
 1014 X9 = X2/10.0
      X11 = X9
      X12 = (X9-X11)*10 + 1
 1021 X12 = X12 + 1
 1020 X13 = X11* 10 + X12
      IX20 = X13 - X2
      IF(IX20-2)1070,1060,1060
```

```
      GO TO 2000
1015  Y(I,J,Z) = X2/A
2000  CONTINUE
      IF(A.EQ.1000)GO TO 999
      DO 8000 K = 1,9
      DO 8000 I = 1,E
8000  W(I,K) = SAV(I,K)
      K = 1
 400  I=1;J=1;R=1;Z=1
 91   DO 700 R=1,E
 700  S(R)=0.0
      DO 6 R = 1,E
      DO 6 Z=1,9
 6    S(R) = S(R) + Y(I,J,Z)*W(R,Z)
      IF(I.EQ.1)GC TC 705
      GO TO 499
 705  R = 1
      L = 2
      CNT = 1
   7  IF(S(R)-S(L))601,601,10
 601  TEMP1=L
      TEMP2=R
      CALL ADJUST
 10   L = L + 1
      IF(L-E)7,7,11
  11  IF(CNT.EQ.1)GO TO 807
      J = J + 1
      GO TO 808
 807  CALL DELETE
      WRITE(6,104)I,J
      WRITE(6,104)ISTORJ(I)
      IF(ISTORJ(I).EQ.1)GO TO 41
 808  IF(J-ISTORJ(I))91,91,41
 499  IF(I-E)15,500,500
 15   R = I; L = R + 1
      CNT = 1
  17  IF(S(R)-S(L))602,602,21
 602  TEMP1=L
      TEMP2=R
      CALL ADJUST
 21   L=L+1
      IF(L-E)17,17,22
 22   L=1
  24  IF(S(R)-S(L))603,603,27
 603  TEMP1=L
      TEMP2=R
      CALL ADJUST
 27   L=L+1
      IF(L-R)24,40,40
  40  IF(CNT.EQ.1)GC TO 809
      J = J + 1
      GO TO 810
 809  CALL DELETE
      WRITE(6,104)I,J
      WRITE(6,104)ISTORJ(I)
      IF(ISTORJ(I).EQ.1)GO TO 41
 810  IF(J-ISTORJ(I))91,91,41
 500  L=1;R=I
      CNT = 1
 699  IF(S(R)-S(L))703,703,701
 703  TEMP1= L
      TEMP2 = R
      CALL ADJUST
 701  L = L + 1
```

```
      J = J + 1
      GO TO 812
  811 CALL DELETE
      WRITE(6,104)I,J
      WRITE(6,104)ISTORJ(I)
      IF(ISTORJ(I).EQ.1)GO TO 41
  812 IF(J-ISTORJ(I))91,91,41
   41 I=I+1
      IF(I-E)42,42,6001
   42 J=1
      GO TO 91
 6001 R = 1
 5000 DO 5001 Z = 1,9
      IW(R,Z) = W(R,Z)*100.0
      ISAV(R,Z) = SAV(R,Z)*100.0
 5001 CONTINUE
      Z = 1
   53 IF(IW(R,Z)-ISAV(R,Z))610,60,610
C    THE STEPS BELOW (5000 THRU 804) MARKED WITH A 'C' REPRESENT AN
C    ATTEMPT TO FORCE CONVERGENCE TO A DIFFERENCE BETWEEN WEIGHT VECTORS
C    OF SUCCESSIVE ITERATIONS OF 0.05.
C 604 IDIFF(R,Z) = IW(R,Z) - ISAV(R,Z)
C 608 IF(IDIFF(R,Z)-5)60,804,804
C 804 IW(R,Z) = IW(R,Z) - IDIFF(R,Z)/2
C     W(R,Z) = IW(R,Z)/100.0
  610 DO 605 R = 1,E
      DO 605 Z = 1,9
      SAV(R,Z) = W(R,Z)
  605 CONTINUE
   57 WRITE(6,58) K
   58 FORMAT('0THE WEIGHT VECTORS FOR THE',1X,'NUMBER',1X,I4,1X,
     +'ITERATION ARE:')
   59 WRITE(6,61)(R,(W(R,Z),Z=1,9),R=1,E)
   61 FORMAT(1X,I3,9E12.3)
      WRITE(6,102)CNT
   62 K=K+1
  720 IF(K-2000)400,400,100
   60 Z=Z+1
      IF(Z-9)53,53,63
   63 R = R + 1; Z = 1; U = R
      IF(R-E)5000,5000,70
   70 R = 1
      WRITE(6,108)
  108 FORMAT('1')
      WRITE(6,80)(R,(W(R,Z),Z=1,9),R=1,E)
  101 WRITE(6,102)K
  102 FORMAT(1X,I4,1X,'ITERATIONS WERE REQUIRED FOR CONVERGENCE')
   80 FORMAT(1X,'WORD',1X,I3,1X,9E12.3)
      GO TO 100
C    THESE STEPS WERE USED TO CHECK FOR SATISFACTION FO THE ALGORITHM
C    WITHOUT ABSOLUTE CONVERGENCE BY CHECKING TO SEE WHETEER FOR R=1
C    G(1) WAS THE LARGEST DISCRIMINANT FUNCTION.  NOTE THATH THE INEEGER
C    FORM OF THE WEIGHT VECTORS WAS USED.
C7000 DO 7001 R = 1,E
C7001 S(R) = 0.0
C     DO 7002 R = 1,E
C     DO 7002 Z = 1,9
C     W(R,Z) = SAV1(R,Z)
C7002 S(R) = S(R) + Y(1,1,Z)*W(R,Z)
C     R = 1; L = 2
C7003 IF(S(R)-S(L))7004,7004,7006
C7004 DO 7005 R = 1,E
C     DO 7005 Z = 1,9
C7005 W(R,Z) = SAV1(R,Z)
```

```
C7006 L = L + 1
C       IF(L-E)7003,7007,7007
C7007 IF(S(R)-S(L))7004,7004,70
  104 FORMAT(1X,5I4)
  100 STOP
      END
      SUBROUTINE ADJUST
      INTEGER CNT
      INTEGER Q,TEMP1,TEMP2,TEMP3, Z,E
      DIMENSION Y(20,10,9),D(20,9),W(20,9),S(20),ISTORJ(20)
      COMMON Y,TEMP1,TEMP2,TEMP3,S,I,J,Z,E,CNT,W,ISTORJ
      C = 1.
      CNT = CNT + 1
      Q = 1
  200 IF(Q-TEMP1)206,202,206
  206 IF(Q-TEMP2)201,205,201
  201 Q = Q + 1
      IF(Q-E)200,200,210
  202 DO 211 Z = 1,9
  211 W(Q,Z) = W(Q,Z) - C*Y(I,J,Z)
      Q = Q + 1
      GO TO 200
  205 DO 212 Z = 1,9
  212 W(Q,Z) = W(Q,Z) + C*Y(I,J,Z)
      Q = Q + 1
      GO TO 200
  210 RETURN
      END
      SUBROUTINE DELETE
C   THIS SUBROUTINE DELETES PATTERNS THAT SATISFY DISCRIMINANT
C   CONDITIONS
      DIMENSION ISTORJ(20), Y(20,10,9),S(20),W(20,9)
      INTEGER CNT, Z
      COMMON Y,TEMP1,TEMP2,TEMP3,S,I,J,Z,E,CNT,W,ISTORJ
      CNT = 1
      IF(J.GE.ISTORJ(I))GO TO 3999
      J2 = ISTORJ(I) - 1
      DO 4000 J1 = J,J2
      J4 = J1 + 1
      DO 4000 Z = 1,9
 4000 Y(I,J1,Z) = Y(I,J4,Z)
 3999 IF(ISTORJ(I).EQ.1)GO TO 4001
      ISTORJ(I) = ISTORJ(I) - 1
 4001 RETURN
      END
```

Program

for

Pattern Recognition

Based on

Linear Error Correction

```
C<*************************************************@*********************
C   THE FIRST WORD IS GO
C   THE SECOND WORD IS EAT
C   THE THIRD WORD IS BED
C   THE FOURTH IS HAPPY
C   THE FIFTH WORD IS SLEEP
C   THE SIXTH WORD IS THIRSTY
C   THE SEVENTH WORD IS HUNGRY
C<*************************************************************************
      INTEGER B,C
      REAL LARG1,LARG2,LARG3
      DIMENSION Y(20,1,9), W (20,9), S(20)
C   THE LETTER C REPRESENTS THE NUMBER OF CLASSES
      C = 7
      READ(5,1)((Y(I,1,K),K=1,9),I=1,C)
    1 FORMAT(F9.2)
      READ(5,2)((W(I,K),K=1,9),I=1,C)
    2 FORMAT(E12.3)
C NORMALIZATION
      I = 1
      J = 1
  449 K = 2
  450 K1 = K + 1
  397 LARG1 = ABS(Y(I,J,K))
      LARG2 = ABS(Y(I,J,K1))
      IF(LARG2.GT.LARG1)GO TO 451
  452 K1 = K1 + 1
      IND = K
      IF(K1-8)397,397,453
  451 IND = K1
      K = K + 1
      IF(K-8)397,397,453
  453 LARG3 = ABS(Y(I,J,IND))
      DO 454 K = 2,8
      Y(I,J,K) = Y(I,J,K)/LARG3
  454 CONTINUE
      I = I + 1
      IF(I.LE.C)GO TO 449
C
C CALCULATION OF THE DISCRIMINANT FUNCTIONS' VALUES FOR THE CASE I=1.
      I = 1
      DO 20 M = 1,C
   20 S(M) = 0.0
      DO 4 M =1,C
      DO 4 N = 1,9
    4 S(M) = S(M) + Y(1,1,N)*W(M,N)
      L = 2
C PATTERN RECOGNITION FOR THE CASE I = 1.
    5 IF(S(1)-S(L))6,6,7
    6 WRITE(6,90)L
      GO TO 9
    7 L = L + 1
      IF(L-C)5,5,8
    8 WRITE(6,91)
    9 I = I + 1
      IF(I-C)21,200,200
C CALCULATION OF THE DISCRIMINANT FUNCTIONS' VALUES FOR THE CASES
C I GREATER THAN ONE
   21 DO 10 M =1,C
   10 S(M) =0.0
      DO 11 M = 1,C
      DO 11 N = 1,9
   11 S(M) = S(M) + Y(I,1,N)*W(M,N)
```

```
C PATTERN RECOGNITION FOR THE CASE I GREATER THAN ONE.
   12 IF(S(I)-S(L))13,13,14
   13 WRITE(6,92)I,L
      GO TO 9
   14 L = L + 1
      IF(L-C)12,12,15
   15 L = 1
   16 IF(S(I)-S(L))17,17,18
   17 WRITE(6,92)I,L
      GO TO 9
   18 L = L + 1
      B = I - 1
      IF(L-B)16,16,19
   19 WRITE(6,93)I
      GO TO 9
  200 DO 80 M = 1,C
   80 S(M) = 0.0
      DO 81 M = 1,C
      DO 81 N = 1,9
   81 S(M) = S(M) + Y(I,1,N)*W(M,N)
      L = 1; I = C ; B = C - 1
   29 IF(S(I)-S(L))30,30,31
   30 WRITE(6,92)I,L
      GO TO 100
   31 L = L + 1
      IF(L-B)29,29,39
   39 WRITE(6,93)I
      GO TO 100
   90 FORMAT('0THE ATTEMPT WAS FOR WORD NUMBER ONE.',1X,
     +'THE FAILURE OCCURED WITH VECTOR NUMBER',1X,I2,'.')
   91 FORMAT('0THE WORD SPOKEN  WAS NUMBER 1')
   92 FORMAT('0THE ATTEMPT WAS FOR WORD NUMBER',1X,I2,1X,'.',1X,
     +'THE FAILURE OCCURRED WITH WEIGHT VECTOR',1X,I2)
   93 FORMAT('0THE WORD SPOKEN WAS NUMBER',1X,I2)
  100 STOP
      END
$EXEC
```

Excerpt from Program for Piece-wise Linear approach with

Fractional Error Correction of the Weight Vectors (Deterministic)

```
      I = 1; E1 = 2
  400 AY = I ; J = 1
   91 DO 700 R = 1,2
  700 S(R) = 0.
      DO 6 R = 1,2
      DO 6 Z = 1,9
    6 S(R) = S(R) + Y(AY,J,Z)*W(R,Z)
      R = 1 ; L = 2
      IF(S(R)-S(L))601,601,11
  601 TEMP1 = L ; TEMP2 = R
      CALL ADJUST
   11 J = J + 1
      IF(J-10)91,91,41
   41 AY = E1
      J = 1
   93 DO 701 R = 1,2
  701 S(R) = 0.
      DO 702 R = 1,2
      DO 702 Z = 1,9
  702 S(R) = S(R) + Y(AY,J,Z)*W(R,Z)
      R = 2 ; L = 1
      IF(S(R)-S(L))703,703,704
  703 TEMP1 = L ; TEMP2 = R
      CALL ADJUST
  704 J = J + 1
      IF(J-10)93,93,6001
 6001 R = 1
 5000 DO 5001 Z = 1,9
      IW(R,Z) = W(R,Z)*100.
      ISAV(R,Z) = SAV(R,Z)*100.
 5001 CONTINUE
      Z = 1
   53 IF(IW(R,Z)-ISAV(R,Z))610,60,610
  610 DO 605 R = 1,2
      DO 605 Z = 1,9
      SAV(R,Z) = W(R,Z)
  605 CONTINUE
      IF(K-1990)62,62,57
   57 WRITE(6,58)K
   58  FORMAT('0THE WEIGHT VECTORS FOR THE',1X,'NUMBER',1X,I4,1X,
     +'ITERATION ARE:')
      WRITE(6,61)1,(W(1,Z),Z=1,9)
      WRITE(6,61)E1,(W(E1,Z),Z=1,9)
   61 FORMAT(1X,I3,9E12.3)
      WRITE(6,102)CNT
   62 K = K + 1
      IF(K-2000)400,400,92
   60 Z = Z + 1
      IF(Z-9)53,53,63
```

```
 63 R = R + 1 ; Z = 1
    IF(R-2)5000,5000,70
 70 WRITE(6,80)I,(W(I,Z),Z=1,9)
    WRITE(6,80)E1,(W(E1,Z),Z=1,9)
    WRITE(6,102)K
102 FORMAT(1X,I4,1X,'ITERATIONS WERE REQUIRED FOR CONVERGENCE')
 80 FORMAT(1X,'WORD',1X,I3,1X,9E12.3)
 92 E1 = E1 + 1
    IF(E1-F)401,401,30
401 K = 1 ; CNT = 1
    GO TO 400
 30 I = I + 1
    IF(I.EQ.F)GO TO 100
    K = 1 ; CNT = 1
    E1 = I + 1
    GO TO 400
100 STOP
    END
    SUBROUTINE ADJUST
    INTEGER CNT,AY
    INTEGER Q,TEMP1,TEMP2,TEMP3,Z,E
    DIMENSION Y(20,10,9),D(20,9),W(20,9),S(20)
    COMMON Y,TEMP1,TEMP2,TEMP3,S,I,J,Z,E,CNT,W,AY
    CNT = CNT + 1
    Q = 1
200 IF(Q-TEMP1)206,4999,206
206 IF(Q-TEMP2)201,5999,201
201 Q = Q + 1
    IF(Q-2)200,200,210
4999 DOTWY = 0.
    DO 5000 Z = 1,9
    DOTWY = DOTWY + Y(AY,J,Z)*W(Q,Z)
5000 CONTINUE
    DOTWY = ABS(DOTWY)*2.
    DOTYY = 0.
    DO 5001 Z = 1,9
    DOTYY = DOTYY + Y(AY,J,Z)*Y(AY,J,Z)
5001 CONTINUE
    C = DOTWY/DOTYY
202 DO 211 Z = 1,9
211 W(Q,Z) = W(Q,Z) - C*Y(AY,J,Z)
    Q = Q + 1
    GO TO 200
5999 DOTWY = 0.
    DO 6000 Z = 1,9
    DOTWY = DOTWY + Y(AY,J,Z)*W(Q,Z)
6000 CONTINUE
    DOTWY = ABS(DOTWY)*2.
    DOTYY = 0.
```

```
      DO 6001  Z = 1,9
      DOTYY = DOTYY + Y(AY,J,Z)*Y(AY,J,Z)
 6001 CONTINUE
      C = DOTWY/DOTYY
  205 DO 212 Z = 1,9
  212 W(Q,Z) = W(Q,Z) + C*Y(AY,J,Z)
      Q = Q + 1
      GO TO 200
  210 RETURN
      END
```

Program

for

Pattern Recognition Based on

Piece-wise Linear Error

```
      INTEGER E1,C,C1,CNT,CNT1,Z,F,SAV
      INTEGER WD,WD1,SAV1,D1,SAV2
      REAL LARG1,LARG2,LARG3
      DIMENSION WORD(80),Y(9),W(20,38,9),S(20,38)
C     C REPRESENTS THE NUMBER OF WORDS
      C = 10
      E1 = C - 1; C1 = 2*C - 2; F = E1
      READ(5,2)(WORD(I),I=1,40)
    2 FORMAT(4A4)
    3 FORMAT(E12.3)
      WRITE(6,80)
   80 FORMAT(1X,'THE INPUT LIST IS:')
      WRITE(6,81)(WORD(I),I=1,40)
      WRITE(6,600)
  600 FORMAT(1X,//,1X,'THE OUTPUT IS:')
   81 FORMAT(1X,4A4)
      READ(5,3)((W(1,J,Z),Z=1,9),J=1,18)
      READ(5,3)((W(2,J,Z),Z=1,9),J=1,16)
      READ(5,3)((W(3,J,Z),Z=1,9),J=1,14)
      READ(5,3)((W(4,J,Z),Z=1,9),J=1,12)
      READ(5,3)((W(5,J,Z),Z=1,9),J=1,10)
      READ(5,3)((W(6,J,Z),Z=1,9),J=1,8)
      READ(5,3)((W(7,J,Z),Z=1,9),J=1,6)
      READ(5,3)((W(8,J,Z),Z=1,9),J=1,4)
      READ(5,3)((W(9,J,Z),Z=1,9),J=1,2)
      K = 1
    1 I = 1; SAV = 1
      SAV2 = 1
      SAV1 = K
      READ(5,70,END=100)(Y(Z),Z=1,9)
C
C     NORMALIZATION
  449 K = 2
  450 K1 = K + 1
  397 LARG1 = ABS(Y(K))
      LARG2 = ABS(Y(K1))
      IF(LARG2.GT.LARG1)GO TO 451
  452 K1 = K1 + 1
      IND = K
      IF(K1-8)397,397,453
  451 IND = K1
      K = K + 1
      IF(K-8)397,397,453
  453 LARG3 = ABS(Y(IND))
      DO 454 K =2,8
      Y(K) = Y(K)/LARG3
  454 CONTINUE
C
      K = SAV1
      DO 4 N = 1,C1
      S(1,N) = 0.
    4 CONTINUE
      DO 9 N = 1,C1
      DO 9 Z =1,9
      S(1,N) = S(1,N) + Y(Z)*W(1,N,Z)
    9 CONTINUE
      CNT1 = 1
      KNT = 1
  150 I = 1
      N = 1; N1 = 2; KNT = 1
      D1 = C1 - 2
    5 IF(S(I,N)-S(I,N1))6,6,7
    6 IF(N1-D1)61,400,400
```

```
      KNT = KNT + 1
      SAV2 = N1; SAV = (N1/2) + 1; I = SAV; N = 1; N1 = 2
      IF(I-1)10,10,63
   63 LIM = (C-SAV)*2
      GO TO 11
   10 LIM = 2*C-2
   11 IF(KNT-C1)12,12,200
   12 IF(CNT1-2)99,150,200
   99 LIM1 = LIM - 1
      IF(I-LIM1)64,400,400
   64 DO 13 N = 1,LIM
      S(I,N) = 0.
   13 CONTINUE
      DO 14 N = 1,LIM
      DO 14 Z = 1,9
      S(I,N) = S(I,N) + Y(Z)*W(I,N,Z)
   14 CONTINUE
      CNT1 = CNT1 + 1
      N = 1; N1 = 2
      GO TO 5
C
C   THIS SECTION FOR LAST 2 WORDS
  400 S(F,1) = 0.
      S(F,2) = 0.
      WD = 4*K
      WD1 = WD - 3
      DO 401 Z = 1,9
      S(F,1) = S(F,1) + Y(Z) * W(F,1,Z)
      S(F,2) = S(F,2) + Y(Z)*W(F,2,Z)
  401 CONTINUE
      IF(S(F,1)-S(F,2))402,402,403
  402 WRITE(6,92)(WORD(L),L=WD1,WD)
      GO TO 60
  403 WRITE(6,91)(WORD(L),L=WD1,WD)
      GO TO 60
C
C   THIS INDICATES FAILURE TO RECGONIZE WORD(K)
  200 WD = 4*K
      WD1 = WD -3
      WRITE(6,92)(WORD(L),L=WD1,WD)
      GO TO 60
C
C   THIS SECTION SETS I TO THE NUMBER CORRESPONDING TO THE
C   DISCRIMINANT FUNCTION WHOSE VALUE IS THE LARGEST FOR THE KTH WORD.
C   THIS IS USED ONLY WHEN THE SAME VALUE OF SAV IS ENCOUNTERED ON A
C   CONSECUTIVE RUN.
  300 SAV = (N1/2) + 1
      DO 301 N = SAV,C1
      S(1,N) = 0.
  301 CONTINUE
      D1 = C1 - 2
      IF(N1-D1)65,400,400
   65 DO 302 N = SAV,C1
      DO 302 Z = 1,9
      S(1,N) = S(1,N) + Y(Z)*W(1,N,Z)
  302 CONTINUE
      N = N1 + 1; N1 = N1 + 2; I = 1
      CNT1 = 1
      GO TO 5
C
    7 KNT = KNT + 1
      IF(I-1)20,20,30
   30 LIM = (C-SAV)*2
      GO TO 21
   20 LIM = 2*C-2
```

```
      IF(KNT-C1)8,8,200
    8 IF(N1-LIM)5,5,22
   22 WD = 4*K
      WD1 = WD - 3
      WRITE(6,91)(WORD(L),L=WD1,WD)
   60 K = K + 1
      IF(K-C)1,1,100
   91 FORMAT(1X,4A4)
   92 FORMAT(1X,'ERROR WITH',1X,4A4)
   70 FORMAT(F9.2)
  100 STOP
      END
```

Program

for

Incremental Error

Correction (Statistical)

```
// EXEC WATFIV
      INTEGER E,R,Z,SQN,F,T,U,X,B,X12,X11,X13,C,X2,X5,X3
      DIMENSION Y(20,10,9), D(20), W(20,9),ISTORJ(20)
      REAL ITER
      REAL LARG1, LARG2, LARG3
C   E REPRESENTS THE NUMBER OF CLASSES, M REPRESENTS THE NUMMBER OF
C   PATTERNS
C    THE COUNTER,ICNT, CONTROLS WHETHER A PATTERN SHOULD BE DELETED ANC
C    RESET EVERY TIME A PATTERN IS DELETED SO THAT FOR EVERY PATTERN TH
C    COUNT IS STARTED AT 1 AND INCREMENTED TO A MAXIMUM OF (E-1).
C    ICNVRG IS INCREMENTED FOR ALL COMPARISONS ONLY IF DISCRIMINANT
C    CONDITIONS ARE SATISFIED.  AT THE END OF EACH ITERATION ICNVRG IS
C    COMPARED TO THE LIMIT (LIM) WHICH IS THE MAXIMUM NUMBER OBTAINED
C    IF CONVERGENT CONDITIONS ARE SATISFIED.
      COMMON Y,ISTORJ, ICNT, J,I
      E = 3; M = 10
      READ(5,1)(((Y(I,J,Z),Z=1,8),J=1,M),I=1,E)
    1 FORMAT(F9.2)
C   STEPS THROUGH 2 SET THE LAST ELEMENT IN Y(I,J,Z) TO 1.0
      DO 2 I = 1,E
      DO 2 J = 1,M
    2 Y(I,J,9) = 1.0
C   THE 400 SERIES ACCOMPLISHES AMPLITUDE NORMALIZATION.
      DO 398 I = 1,E
      DO 398 J = 1,M
  449 K = 2
  450 K1 = K + 1
  397 LARG1 = ABS(Y(I,J,K))
      LARG2 = ABS(Y(I,J,K1))
      IF(LARG2.GT.LARG1)GO TO 451
  452 K1 = K1 + 1
      IND = K
      IF(K1-8)397,397,453
  451 IND = K1
      K = K + 1
      IF(K-8)397,397,453
  453 LARG3 = ABS(Y(I,J,IND))
      DO 455 K = 2,8
      Y(I,J,K) = Y(I,J,K)/LARG3
  455 CONTINUE
  398 CONTINUE
C   THE 1000 SERIES ROUNDS OFF THE PATTERNS ELEMENTS TO THE SECOND DECI
C   PLACE
      A = 1000
      GO TO 1000
  999 A = 100
 1000 DO 2000 I = 1,E
      DO 2000 J = 1,10
      DO 2000 Z = 1,8
```

```
      X1 = Y(I,J,Z)
      IF(X1)151,1002,150
  151 X1 = ABS(Y(I,J,Z))*A + 0.00001
      T = X1 *100.0 + 1.0
      T = T + 2
      X1 = T/100.0
      GO TO 1001
  150 X1 = Y(I,J,Z)*A + 0.00001
      T = X1 *100.0 + 1.0
      T = T + 2
      X1 = T/100.0
      GO TO 1003
 1001 X2 = ABS(X1)
      X1 = ABS(X1)
      X3 = (X1 - X2)*10
      IU = X3
 1030 IF(IU-5)1004,1005,1007
      GO TO 2000
 1004 Y(I,J,Z) = -X2/A
 1005 X4 = (X2/A)*10.0
      X5 = X4
       IX6 = (X4-X5)*10
      N = 1
      C = 2*N
 1008 IF(C-IX6)1006,1007,1006
 1006 N = N + 1
      IF(N-4)1008,1008,1009
 1007 X9 = X2/10.0
      X11 = X9
      X12 = (X9-X11)*10 + 1
 1016 X12 = X12 + 1
 1017 X13 = X11* 10 + X12
      IX20 = X13 - X2
      IF(IX20-2)1018,1050,1050
 1050 X13 = X13 - 1
 1018 Y(I,J,Z) = -X13/A
      GO TO 2000
 1009 X8 = X2/A
      Y(I,J,Z)= -X8
      GO TO 2000
 1002 Y(I,J,Z) = Y(I,J,Z)
      GO TO 2000
 1003 X2 = X1
      X3 = (X1 - X2)*10
      IU = X3
 1040 IF(IU-5)1010,1011,1014
 1010 Y(I,J,Z) = X2/A
      GO TO 2000
 1011 X4 = (X2/A)*10.0
```

```
 12 DO 13 R = 1,E
 13 D(R) = 0.0
    DO 14 K = 1,9
    DO 14 R = 1,E
 14 D(R) = D(R) + Y(I,J,K)*W(R,K)
    IF(I.EQ.1)GO TO 103
    GO TO 20
103 R = 1; L = 2
 15 IF(D(R).GT.D(L))GO TO 16
    GO TO 17
 16 ICNT = ICNT + 1
    ICNVRG = ICNVRG + 1
 17 L = L + 1
    IF(L.LE.E)GO TO 15
    F = E-1
    IF(ICNT.EQ.F)GO TO 18
    J = J + 1; ICNT = 0
    GO TO 19
 18 CALL DELETE
 19 IF(ISTORJ(I).EQ.1)GO TO 30
    IF(J.LE.ISTORJ(I))GO TO 12
    GO TO 30
 20 IF(I.GE.E)GO TO 31
    R = I; L = R + 1
 21 IF(D(R).GT.D(L))GO TO 22
    GO TO 23
 22 ICNT = ICNT + 2
    ICNVRG = ICNVRG + 1
 23 L = L + 1
    IF(L.LE.E)GO TO 21
    L = 1
 24 IF(D(R).GT.D(L))GO TO 25
    GO TO 26
 25 ICNT = ICNT + 1
    ICNVRG = ICNVRG + 1
 26 L = L + 1
    IF(L.LT.R)GO TO 24
    F = E -1
    IF(ICNT.EQ.F)GO TO 27
    GO TO 28
 27 CALL DELETE
    GO TO 29
 28 J = J + 1; ICNT = 0
 29 IF(ISTORJ(I).EQ.1)GO TO 30
    IF(J.LE.ISTORJ(I))GO TO 12
 30 I = I + 1
    IF(I.GT.E)GO TO 37
    J = 1; ICNT = 0
    GO TO 12
```

```
      X5 = X4
       IX6 = (X4-X5)*10
      N = 1
      C = 2*N
 1012 IF(C-IX6)1013,1014,1013
 1013 N = N + 1
      IF(N-4)1012,1012,1015
 1014 X9 = X2/10.0
      X11 = X9
      X12 = (X9-X11)*10 + 1
 1021 X12 = X12 + 1
 1020 X13 = X11* 10 + X12
      IX20 = X13 - X2
      IF(IX20-2)1070,1060,1060
 1060 X13 = X13 - 1
 1070 Y(I,J,Z) = X13/A
      GO TO 2000
 1015 Y(I,J,Z) = X2/A
 2000 CONTINUE
      IF(A.EQ.1000)GO TO 999
C    STEPS THROUGH 3996 SET THE INITIAL NUMBERS OF PATTERNS FOR THE
C    VARYING L  CLASS SIZES THAT RESULT FROM DELETING PATTERNS THAT
C    SATISFY DISCRIMUNANT CINDITIONS.
      DO 3996 J = 1,M
 3996 ISTORJ(J) = M
C    STEPS THROUGH (10) STATISTICALLY DETERMINE CLASS WEIGHT VECTORS
      ITER = 1.0; J = 1; ICNT = 1; ALPHA = 1.0
      DO 3 K = 1,9
      DO 3 I = 1,E
    3 W(I,K) = 0.0
    4 I = 1; J = 1
    5 DO 6 K = 1,9
    6 FCN = 1.0 - w(I,K)*Y(I,J,K)
      IF(FCN.GT.0.0)GO TO 7
      SQN = -1
      GO TO 8
    7 SQN = 1
    8 DO 9 K = 1,9
    9 W(I,K) = W(I,K) + ALPHA*SQN*Y(I,J,K)
  301 J = J + 1
      IF(J.LE.ISTORJ(I))GO TO 5
      I = I + 1
      IF(I.LE.E)GO TO 10
      GO TO 11
   10 J = 1
      GO TO 5
C    STEPS THROUGH 36 COPARE DISCRIMINANT FUNCTIONS  FOR AL PATTERNS TC
C    FOR CONVERGENCE
   11 I = 1; ICNVRG = 0; J = 1; ICNT = 0
```

```
   31 L = 1; R = I
   32 IF(D(R).GT.D(L))GO TO 33
      GO TO 34
   33 ICNT = ICNT + 1
      ICNVRG = ICNVRG + 1
   34 L = L + 1
      IF(L.LT.E)GO TO 33
      F = E - 1
      IF(ICNT.EQ.F)GO TO 35
      J = J + 1; ICNT = 0
      GO TO 36
   35 CALL DELETE
   36 IF(ISTORJ(I).EQ.1)GO TO 30
      IF(J.LE.ISTORJ(I))GO TO 12
      GO TO 30
C    STEPS THROUGH 38 CHECK FOR CONVERGENCE
   37 LIM = 0; I = 1; F = E - 1
   42 J1 = ISTORJ(1)
   38 LIM = LIM + F*J1
      I = I + 1
      IF(I.LE.E)GO TO 42
      IF(ICNVRG.GE.LIM)GO TO 41
      ITER = ITER + 1.0
      ALPHA = 1/ITER
C    ITER IS OFF BY ONE (GREATER THAN IT SHOULD BE)
      IF(ITER.LE.150.0)GO TO 40
   39 WRITE(6,58)ITER
      WRITE(6,61)(R,(W(R,Z),Z=1,9),R=1,E)
      WRITE(6,200)ICNVRG
      WRITE(6,200)LIM
   40 IF(ITER.LE.2000.0)GO TO 4
      GO TO 100
   41 WRITE(6,80)(R,(W(R,Z),Z=1,9),R=1,E)
      WRITE(6,102)ITER
      WRITE(6,200)ICNVRG
      WRITE(6,200)LIM
      GO TO 100
   58 FORMAT('0THE WEIGHT VECTORS FOR THE ',1X,'NUMBER',1X,F6.1,1X,
     +'ITERATION ARE:')
   61 FORMAT(1X,I3,9E12.3)
   80 FORMAT(1X,'WORD',1X,I3,1X,9E12.3)
  102 FORMAT(1X,F6.1,'ITERATIONS WERE REQUIRED FOR CONVERGENCE')
  200 FORMAT(1X,I5)
  100 STOP
      END
      SUBROUTINE DELETE
C    THIS SUBROUTINE DELETES PATTERNS THAT SATISFY DISCRIMINANTS CONDITION
      DIMENSION ISTORJ(20),Y(20,10,9)
      INTEGER Z
```

```
      COMMON Y,ISTORJ, ICNT, J, I
      ICNT = 0
      IF(J.GE.ISTORJ(I))GO TO 3999
      J2 = ISTORJ(I) - 1
      DO 4000 J1 = J,J2
      J4 = J1 + 1
      DO 4000 Z = 1,9
4000  Y(I,J1,Z) = Y(I,J4,Z)
3999  IF(ISTORJ(I).EQ.1)GO TO 4001
      ISTORJ(I) = ISTORJ(I) - 1
4001  RETURN
      END
```

BIBLIOGRAPHY

1. Andrews, D.F., "Plots of High Dimensiona Data," Biometrics, Vol. 28, March 1972, pp. 125-136.

2. Atal, B.S., and Hanauer, S., "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Acoustic Society of America, Vol. 50, 1971, pg. 637.

3. Blackman, R.B., and Tukey, J.W., The Measurement of Power Spectra, Dover, New York, 1958.

4. Brown, William, M., and Palermo, Carmen, J., Random Processes, Communications and Radar, New York, McGraw-Hill, 1969.

5. Chien, Y.T., "Interactive Pattern Recognition: Techniques and Systems," Computer, May 1976, pp. 11-25.

6. Dorland's Illustrated Medical Dictionary, Twenty-fifth edition, Saunders, Philadelphia, 1974.

7. Drucker and Preusse, "Real-Time Recognition of Ten Vowel-like Sounds in Continous Speech," IEEE Transactions on Acoustics, Speech, and Signal Processing, Feb. 1974, pg. 74.

8. Dubnowski, John J., Schafer, Ronald W., Rabiner, Lawrence, R., "Real-time Digital Hardware Pitch Detector," IEEE Transactions on Acoustics, speech, and signal processing, Vol. ASSP-24, No. 1, pp. 2-8, February, 1976.

9. Flanagan, J.L. Speech Analysis, Synthesis and Perception, New York, Academic Press, 1965.

10. Fu. K.S., Sequential Methods in Pattern Recognition and Machine Learning, New York, Academic Press, 1968.

11. Gray Jr., A.H. and Markel, J.D., "COSH Measure for Speech Processing," 90th Mtg. Acoustical Society of America, Vol. 58 (1), p. 547, Fall 1975.

12. Itakura, F., "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Transactions Acoustics, Speech, and Signal Processing, Vol. ASSP-23 (1), pp. 67-72, Feb. 1975.

13. Lea, W.A., "An Approach to Syntactic Recognition Without Phonemics,"

IEEE Transactions Audio and Electroacoustics. Vol. AU-21, pp. 249-258, June 1973.

14. Makhoul, John, "Spectral Analysis of Speech by Linear Prediction," IEEE Transactions on Audio and Electroacoustics, Vol. AU-21 (3) pp. 140-148, June 1973.

15. Markel, John D., and Grah, A.H., "On Autocorrelation Equations as Applied to Speech Analysis," IEEE Transactions on Audio and Electroacoustics, Vol. AU-21 No. 2, March 1973, pp. 69-79.

16. Markel, J.D., and Gray Jr., A.H., "Cepstral Distance and the Frequency Domain," 90th Mtg., Acoustical Society of America, Vol. 58, p. 597, Fall 1975.

17. Markel, John D., "Digital Inverse Filtering, A New Tool for Formant Trajectory Estimation," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. AU-20, June 1972, pp. 129-137.

18. Mathis, J.H., and Hurtley, H.O., "Stochastic Compartmental Analysis: Model and Least Squares Estimation from Time Series Data," Biometrics, Vol. 27, pp. 77-102, March 1971.

19. Mermelstein, Paul, "Automatic Segmentation of Speech into Syllabic Units," J. Acoustical Society of America, Vol. 58 (4), pp. 880-883, October 1975.

20. Mix, Dwight F., Random Signal Analysis, Reading, Mass., Addison-Wesley, 1969.

21. Mizoguichi, Riichiro and Shimura, Masamichi, "Non-parametric Learning Without a Teacher Based on Mode Estimation," IEEE Transactions on Computers, Vol. C-25, No. 11, November 1976, pp. 1109-1117.

22. Newell, A., Barnett, J.; Fogie, J.; Green, C.; Klatt, D.; Licklider, J.C.R.; Munson, J.; Reddy, R.; and Woods, W.; Speech Understanding Systems: Final Report of a Study Group, North-Holland Publishing Co., Amsterdam/American Elsevier Publishing Co., New York, 1973.

23. Nilsson, Nils J., Problem Solving Methods in Artificial Intelligence, New York, McGraw-Hill, 1971.

24. Nilsson, Nils J., Learning Machines, New York, McGraw-Hill, 1965.

25. Orth, Harper, Norman, OK., Oklahoma Cerebral Palsy Center, Personal communication.

26. Pols, L.C.W.; Tromp, H.R.C.; and Plomp, R.; "Frequency Analysis of Dutch Vowels from 50 Male Speakers," J. Acoustical Society of America, Vol. 53 (4), pp. 1093-1101, 1973.

27. Rabiner, L.R. and Sambur, M.R., "An Algorithm for Determining the End-points of Isolated Utterances," The Bell System Technical Journal, Vol. 54, (2), pg. 297.

28. Schafer, R.N. and Rabiner, L.R., "Parametric Representations of Speech--Speech Recognition," Invited Papers of the IEEE Symposium, New York, Academic Press, 1975.

29. Thomas, John B., An Introduction to Statistical Communications, New York, Wiley and Sons, 1969.

30. Tabu, Herbert and Schilling, Donald L., Principles of Communications, New York, McGraw-Hill, 1971.

31. Taube, Mortimer, Computers and Common Sense, New York, Columbia University Press, 1961.

32. Tou, J.T. and Gonzalez, R.C., Pattern Recognition Principles, Reading, Massachusetts, Addison-Wesley, 1974.

33. Wakita, H., "Direct Estimation of the Vocal Tract Shape by Inverse Filtering of Acoustic Speech Waveforms," IEEE Transactions on Audio and Electroacoustics, Vol. AU-21, pp. 417-427, October 1973.

34. Wiener, Norbert, Extrapolation, Interpolation and Smoothing of Stationary Time Series, Cambridge, Massachusetts, Technology Press of M.I.T., 1949.

35. White, George M., "Speech Recognition: A Tutorial Overview," Computer, May 1976, pp. 40-53.

36. White, G.M. and Neely, R.B., "Speech REcognition Experiments with Linear Prediction, Bandpass Filtering and Dynamic Programming," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, April 1976, pp. 183-188.