AD-A064 058 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH.(U)
NOV 78 G L BROCK, E S KOLESAR

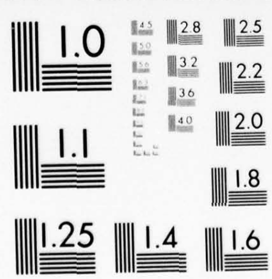
UNCLASSIFIED AFIT/GE/EE/78D-20 NL

1 OF 3
AD
A064058

ASSIFI

1 OF 3

AD A064058



MICROCOPY RESOLUTION TEST CHART

(1)

LEVEL II

COMPUTER IDENTIFICATION
OF PHONEMES
IN CONTINUOUS SPEECH

Gary L. Brock
Capt USAF
and
Edward S. Kolesar, Jr.
Capt USAF

AFIT/GE/EE/78D

Approved for public release:
distribution unlimited

19 Jan 79

JOSEPH P. HIPPS, Major, USAF
Director of Information

14 AFIT/GE/EE/78D-20

6

COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH.

9 Master's THESIS,

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

10 Gary L. Brock
Capt USAF
and
Edward S. Kolesar, Jr.
Capt USAF

Graduate Electrical Engineering

12 216p.

11 Nov[redacted]78

Approved for public release; distribution unlimited

012 225

mt

Preface

This work has been motivated by the research of Dr. Matthew Kabrisky, Professor of Electrical Engineering, at the Air Force Institute of Technology. The initial research was begun by Ralph W. Neyman and continued by William R. Hensley and the team of Michael F. Guyote and Patrick L. Sisson. It is an effort to identify continuous speech through phoneme identification without using higher level decision cues, such as that provided by syntactic, semantic, and prosodic information.

A glossary is included in Appendix F to help clarify certain terms used in the body of the thesis.

We are especially indebted to our advisors Dr. Kabrisky and Capt. Borky for their advice and guidance throughout the research and preparation of this report. We would also like to express our appreciation to William B. Hall of the Analog/Hybrid Systems Branch of the ASD Computer Center for his support in the preliminary processing of the analog speech data. In addition, we wish to thank William J. Bustard for his assistance in running the computer programs, and his wife, Molly Bustard, librarian at the Air Force Institute of Technology, for her assistance in obtaining necessary research materials.

We extend a special thanks to our wives for their patience and continued understanding during the writing of this thesis.

Finally, we would like to thank our typist, Evelyn Shaw, for her dedication to the completion of this document.

Gary L. Brock

and

Edward S. Kolesar Jr.

Contents

Contents

List of Figures

List of Figures
(Continued)

List of Tables

List of Tables
(Continued)

List of Tables
(Continued)

Abstract

An approach to computer recognition of continuous speech through phoneme identification is presented. Speech data is recorded on a tape recorder, digitally sampled, Fast Fourier Transformed and logarithmically compressed into 16 frequency channels. This digitized data is first processed by a crosscorrelation and then by a decision program. After the phonemes are located, a ranking of the selections is made. This procedure was used on both the discrete and continuous speech of five different speakers. Phoneme averaging was used to calculate a universal set of eight prototype phonemes. For the word groups analyzed, the final identification and location rates were 83.5 and 95.9 percent. For the verification sentences analyzed the final identification and location rates were 77.9 and 91.1 percent for discrete speech and 66.1 and 88.2 percent for continuous speech. For the test sentences analyzed the final identification and location rates were 62.3 and 77.9 for discrete speech and 48.6 and 66.1 percent for continuous speech.

COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH

I. Introduction

This research effort is a continuation of the work initiated by Major Ralph W. Neyman and continued by Captain(s) William R. Hensley, Michael F. Guyote, and Patrick L. Sisson on the problem of computer speech recognition. The long term goal of this research is to achieve the recognition of unrestricted, continuous speech by machine.

In various situations, such as the highly automated cockpit of today's aircraft, the restriction on man's ability to communicate to a computer or machine through the use of conventional input/output peripherals is becoming increasingly intolerable. The advantages of a spoken word input to a computer or machine have been recognized and techniques to solve this problem are being analyzed by many research groups throughout the world (Ref 1:319). Experiments comparing speech with other modes of communication, such as typing, have indicated that information is transferred almost twice as fast with speech (Ref 19:2). Thus, speech input will help optimize the man/machine interface.

Present literature expresses the opinion that a continuous speech recognition system is still years in the future, and even then, the system may be highly restrictive (Ref 22:531). However, the encouraging results presented in the Neyman, Hensley, Guyote, and Sisson theses seem to contradict this belief and are the basis for this continued research (Ref 11, 13, 19).

Motivation

All current systems of voice control that rely on the computer recognition of human speech are based on a highly constrained manner of speaking. A sample of the more accurate speech recognition systems is listed in Table I (Ref 8, 18, 22:531). These speech recognition systems perform only in response to an isolated-word or isolated-phrase input. Further, the general constraints that must be observed when using these machines include any one or combination of the following:

1. All commands must be separated by a long pause.
2. Vocabulary words are limited to a class size of approximately 560.
3. Commands must be spoken in a specified word order.
4. The speech recognition system must be programmed to the unique speaking characteristics of each user who must be very consistent in his speech.

The ultimate speech recognition system is one that would respond to a natural, unrestricted voice input. When a person speaks, a complex acoustic signal is generated. This signal is a function of the size and shape of the individual's vocal cavity and movements of the tongue, lips, and teeth. Also, the nature of the speech signal itself changes with the individual's rate of speaking, emotional state, and context of the utterance. Therefore, instead of

Table I

Claimed Performance of Speech Processing Systems

Facility and Investigator	System Capabilities	Percent Correct
Bolt, Beranak and Newman, Inc. D. G. Bobrow (1969)	109 isolated words, single speakers	91-94
SRI P. Vichens	54 isolated words, single speakers	98-100
	54 isolated words, 10 speakers, pooled data, arbitrary training order	79.4
	561 isolated words	91.4
Calgary University D. R. Hill (1969)	16 isolated words, 12 unknown speakers (system trained on different speakers)	78
IBM N. R. Dixon and C. C. Tappert (1971)	250-word vocabulary, continuous speech, several speakers	75
Threshold Technology, Inc. T. B. Martin (1971)	10 digits, pairs and triples, 170 male speakers (including 77-dB background noise, light labor for talkers), no adjustment from initial setting	99
Threshold Technology, Inc. M. B. Herscher and R. B. Cox (1972)	10 isolated digits, male and female speakers	99
Univac M. Medress (1972)	100 words, 5 speakers (one used for training)	94
Texas Instruments Doddington (1973)	10 digits, continuous speech	99

(Ref 8, 18, 22:531)

trying to recognize discrete words, of which there are literally tens of thousands in the English language, this research is concerned with identifying the fundamental elements of words. These elements are called phonemes and they are defined to be the smallest distinguishable units of speech.

Experiments indicate that one-fourth to one-half of the words in normal conversational speech are unintelligible when taken out of context and heard in isolation (Ref 31:41). This implies that a system for understanding continuous speech must use context related rules to identify the words in the sentence. A machine dedicated to recognizing isolated words would need an extremely large memory capacity to contain all the words in the English language in addition to the related context programs. However, a more versatile recognition system that relies upon phoneme detection would only require storage for approximately 100 phonemes and the related context programs.

Military applications for a speech recognition system include security, command and control, data transmission and communication, and the processing of distorted speech. Table II presents a representative listing of these potential applications (Ref 1:310).

Objective

The main objective of this research was to improve and change as necessary the speech recognition scheme previously

Table II

Military Tasks for Possible Automation

1) <u>Security</u>

   1.1 Speaker Verification (Authentication)
   1.2 Speaker Identification (Recognition)
   1.3 Determining emotional state of speaker (e.g., stress effects)
   1.4 Recognition of spoken codes
   1.5 Secure access voice identification, whether or not in combination with fingerprints, facial information, identity card, signature, etc.
   1.6 Surveillance of communication channels.

2) <u>Command and Control</u>

   2.1 System control (ships, aircraft, fire control, situation displays, etc.)
   2.2 Voice-operated computer input/output (each telephone a terminal)
   2.3 Data handling and record control
   2.4 Material handling (mail, baggage, publications, industrial applications)
   2.5 Remote control (dangerous material)
   2.6 Administrative record control

3) <u>Data Transmission and Communication</u>

   3.1 Speech synthesis
   3.2 Vocoder systems
   3.3 Bandwidth reduction or, more general, bit-rate reduction
   3.4 Ciphering/coding/scrambling

4) <u>Processing Distorted Speech</u>

   4.1 Diver speech
   4.2 Astronaut communication
   4.3 Underwater telephone
   4.4 Oxygen mask speech
   4.5 High "G" force speech

(Ref 1:310)

developed by Neyman, et al. (Ref 11, 13, 19). Also, a method was developed to identify phonemes from continuous speech so that an average or universal set of prototype phonemes could be calculated. This set of universal prototype phonemes was then used to locate and identify similar phonemes in the continuous speech of dissimilar speakers using pattern matching and crosscorrelation techniques.

Although analyzed spectral information can produce some recognition, it cannot do the entire job. To quote Flanagan:

> Automatic speech recognition--as the human accomplishes it--will probably be possible only through the proper analysis and application of grammatical, contextual, and semantic constraints. This approach also presumes an acoustic analysis which preserves the same information that the human transducer (i.e., the ear) does. It is clear, too, that for a given accuracy of recognition, a trade can be made between the necessary linguistic constraints, and complexity of vocabulary, and the number of speakers (Ref 9:163).

In recognition of the above, this research does not include linguistic or syntactic recognition schemes since the entire set of phonemes for the English language was not developed. However, the rank ordering of the identified phonemes by a decision scheme would permit the use of a higher-order linguistic/syntactic program.

Another objective was to identify the application of current microelectronic devices or the need for a special type of device to implement this speech recognition scheme.

Scope

The desired result of this research was to implement a technique to develop universal phonemes and to locate and identify these phonemes in discrete and continuous speech. A set of eight phonemes from Table III was used in this research. A representative phoneme set was chosen from word groups spoken by the authors to develop an average prototype phoneme set. This phoneme set was then correlated with discrete and continuous sentence samples spoken by the authors. This was done to verify that the phonemes could be located and identified in speech from which the average prototype phoneme set was calculated.

To verify that the phoneme set was universal and could be used to locate and identify phonemes in the speech of others, it was correlated with sentences spoken by three different speakers. In total, twelve sentences composed of words containing the eight prototype phonemes were analyzed.

A hardware modelling study was also done to implement this speech recognition scheme. A major goal of this study was to identify either the application of current microelectronic devices or the need for the development of special devices that could be used to make this speech recognition scheme a reality.

Table III

Fundamental Phoneme Set

Key Word	Phonetic Symbol	Computer Rep.	Key Word	Phonetic Symbol	Computer Rep.
Vowels			Glides (cont'd.)		
1. eve	/ i /	E	26. you	/ j /	Y
2. it	/ I /	I	27. read	/ r /	R
3. hate	/ e /	A	28. let	/ l /	L
4. met	/ ɛ /	>E	Combination Sounds		
5. at	/ æ /	&E	29. when	/ hw /	HW
6. ask	/ a /	&	30. church	/ tʃ /	CH
7. father	/ ɑ /	AE	31. judge	/ d_3 /	DZ
8. not	/ ɒ /	IO	Voiced Fricatives		
9. all	/ ɔ /	¢	32. he	/ h /	H
10. obey	/ o /	O	33. ahead	/ ḣ /	XH
11. put	/ U /	U	34. vote	/ v /	V
12. boot	/ u /	OO	35. then	/ § /	TH
13. up	/ ʌ /	-A	36. zoo	/ z /	Z
14. about	/ ə /	AUH	37. pleasure	/ 3 /	ZH
15. church	/ ɚ /	UR	Voiceless Fricatives		
Dipthongs			38. feel	/ f /	F
16. came	/ eI /	EI	39. thin	/ ϕ /	TS
17. I	/ aI /	&I	40. see	/ s /	S
18. boy	/ ɔI /	¢I	41. she	/ ʃ /	SH
19. out	/ ɑU /	AU	Voiceless Stops		
20. go	/ oU /	OU	42. pay	/ p /	P
21. new	/ IU /	IU	43. to	/ t /	T
Nasals			44. key	/ k /	K
22. me	/ m /	M	Voiced Stops		
23. no	/ n /	N	45. be	/ b /	B
24. sing	/ ŋ /	NG	46. day	/ d /	D
Glides			47. go	/ g /	G
25. we	/ w /	W			

(Ref 20)

II. Data Acquisition

The data acquisition scheme used to locate and identify prototype phonemes is based on the restriction that the speech data must be similar to that which is processed by the human ear. The basic function of the outer ear is to transform the acoustic pressure variations of sound energy so that it can be used by the frequency analysis portion of the middle ear and cerebral cortex to recognize speech (Ref 15). The data acquisition and processing scheme that best models the function of the human ear consists of the following elements: a speaker, a microphone, an audio tape recorder, an analog-to-digital computer, a Fast-Fourier Transform (FFT) computer algorithm, and a crosscorrelation/decision computer algorithm. An overview of the speech recognition process showing the parallels between human and machine recognition is shown in Figure 1.

The data acquisition process consists of reciting the desired words, phrases, or sentences into one channel of a reel-to-reel stereo tape recorder. Tone markers of 2kHz are recorded on the second channel to identify the beginning and end of each group of data, as well as the change of speakers. Thus, these tones identify discrete blocks of speech data and serve as a calibration reference point for the personnel who operate the preanalysis FFT computer algorithm. The scheme used to record the speech data is shown in Figure 2.

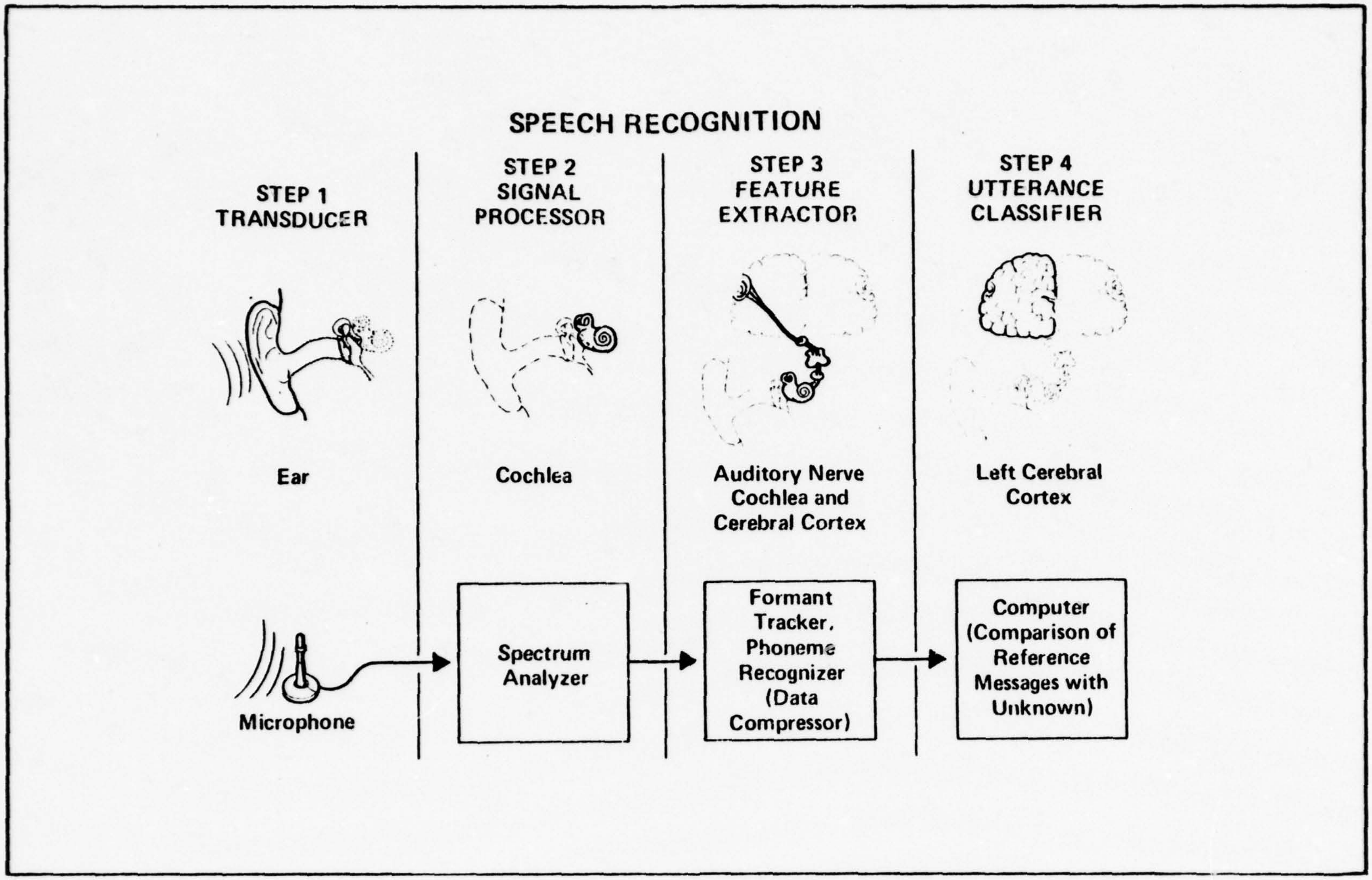


Figure 1. An Overview of Speech Recognition Showing Parallels Between Human and Machine Recognition (Ref 31)

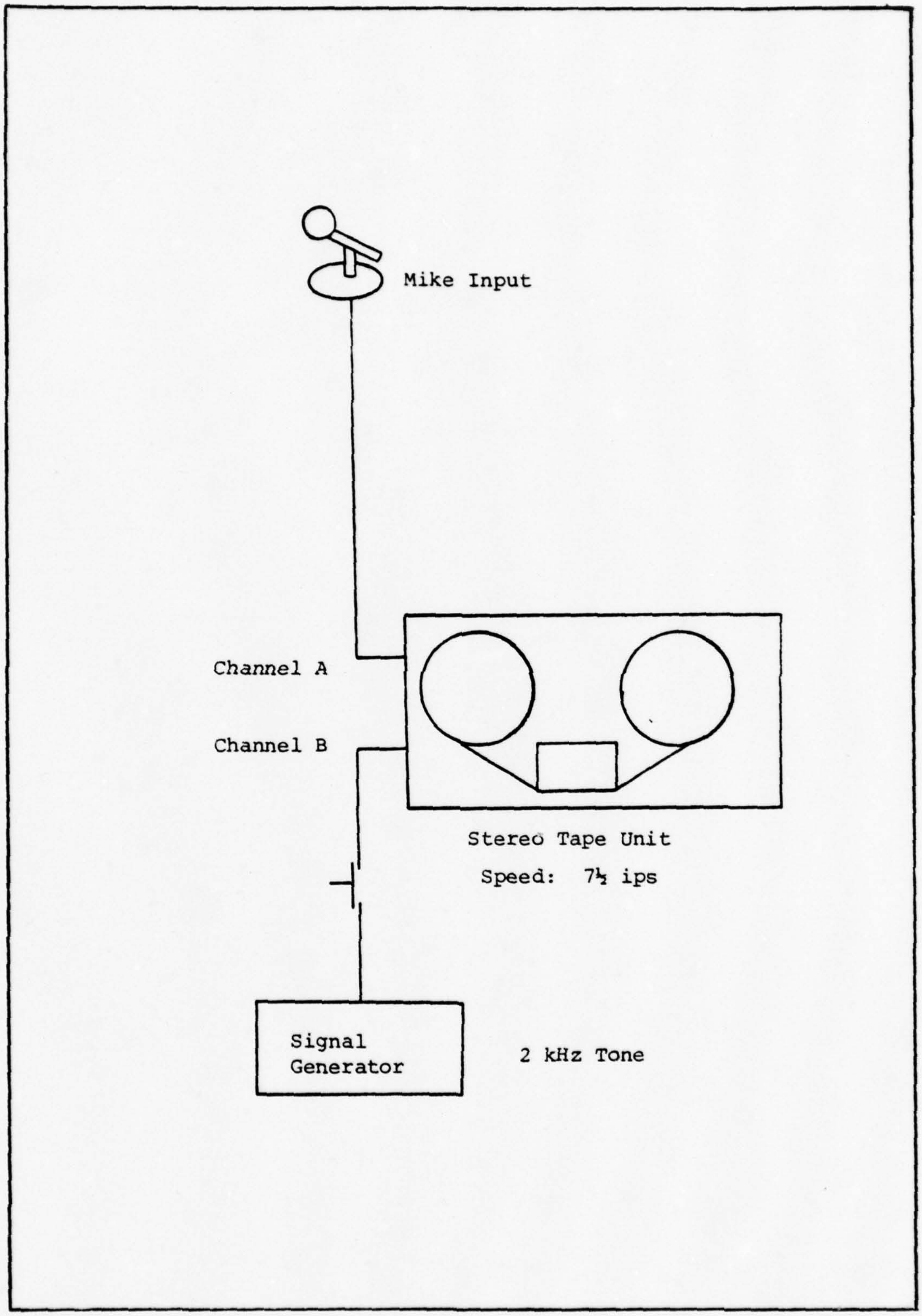


Figure 2. Data Acquisition Scheme

III. Data Preprocessing

The initial processing of the analog speech data was accomplished by the Analog/Hybrid Systems Branch of the ASD Computer Center. Figure 3 illustrates the hardware scheme used.

Analog-to-Digital Conversion

The COMCOR CI-5000/6 analog-to-digital computer was used to digitize the analog speech data. However, because its input amplifiers were limited to a bandwidth of 2.5kHz, it was necessary to modify the original speech data. Since normal speech contains important frequencies up to 5kHz, the bandwidth limitations of the computer's amplifiers were compensated for in the following manner. The original speech data was played at a speed of 3-3/4 inches-per-second, the resulting audio signal was low-pass filtered to 2.5kHz, and this signal was then sampled at twice this frequency (5kHz) in order to satisfy the Nyquist sampling requirements. This procedure, however, is equivalent to playing the tape at its originally recorded speed of 7-1/2 inches-per-second, low-pass filtering to 5kHz, and sampling the final output at 10kHz. In addition, before the analog speech data was digitized, the filtered signal was amplified to 100 volts to insure a signal of sufficient amplitude to permit accurate sampling by the 11-bit analog-to-digital converters of the COMCOR computer.

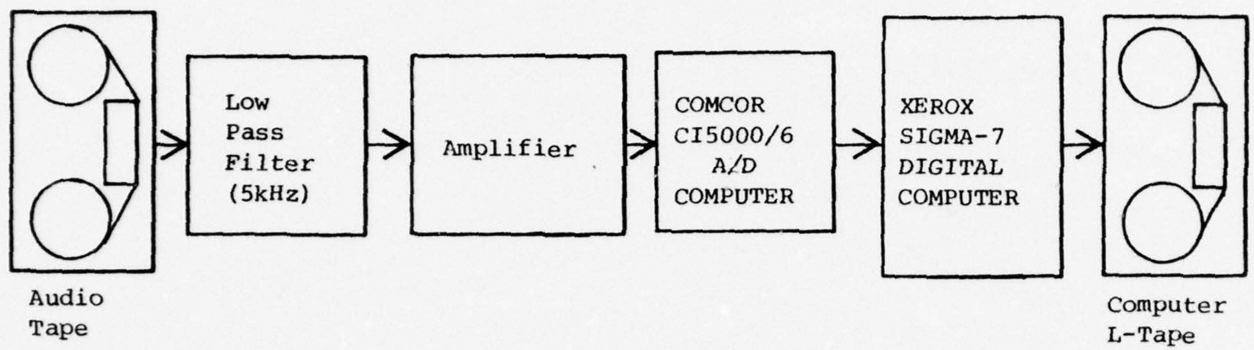


Figure 3. Data Preprocessing Scheme

The digital output from the COMCOR computer is an 11-bit binary representation of a four-digit decimal number that describes the amplitude of the analog speech data at a specific instant of time. This format of the original speech data was then used as the input signal to the frequency analysis segment of the data pre-processing sequence.

Frequency Analysis

In order to identify the frequency components of a particular phoneme, the digitized speech data was converted to an equivalent frequency representation. The properties of the Fast-Fourier Transform (FFT) that permit the computation of a frequency representation of a time-varying signal were used to accomplish this requirement (Ref 2:41-52).

The desired results were obtained using a Xerox Sigma-7 digital computer and the Analog/Hybrid Systems Branch AMPSPC FFT algorithm. The AMPSPC algorithm samples the digitized speech data of the COMCOR computer in sets of 128 samples and creates a (1 x 128) input array for the FFT algorithm. Since each frequency sample represents the analog output at each 10^{-4} (1/10 kHz) second time increment, 128 of the samples represent a net elapsed time of 12.8×10^{-3} seconds (12.8 ms). The AMPSPC algorithm then uses this input array to compute the Discrete-Fourier Transform (DFT) of the digitized time-varying speech signal. The result of this computation is the magnitude of each complex number in the frequency domain. Also, each point in the FFT array is an integral multiple of 78.125 Hz (10 kHz/128 samples).

Since the digitized input signal to the FFT is composed of real numbers, the real part of the FFT is symmetric about the folding frequency (one-half the sampling frequency). Also, the magnitudes of the FFT elements are symmetric about the folding frequency. Therefore, although 128 samples were used to calculate the 128-point DFT, the conjugate symmetry property of the FFT guarantees that only the first 64 transformed components are necessary to represent the frequency spectrum for each 12.8 ms time interval of the original analog speech signal. Figure 4 illustrates the application of the FFT technique to an analog speech signal.

Data Storage

The medium selected to store the FFT speech data was a magnetic library tape (L-tape) which is compatible with the input/output options of the Cyber/6600 computer. Since this L-tape was stored at the ASD Computer Center, access to it from the AFIT processing center was very convenient. One L-tape per speaker was created to avoid the confusion of having all five speakers on one or two L-tapes. This allowed ready access to a specific individual's words and/or sentences. The transfer of the FFT speech data to the L-tape completes the data preprocessing sequence.

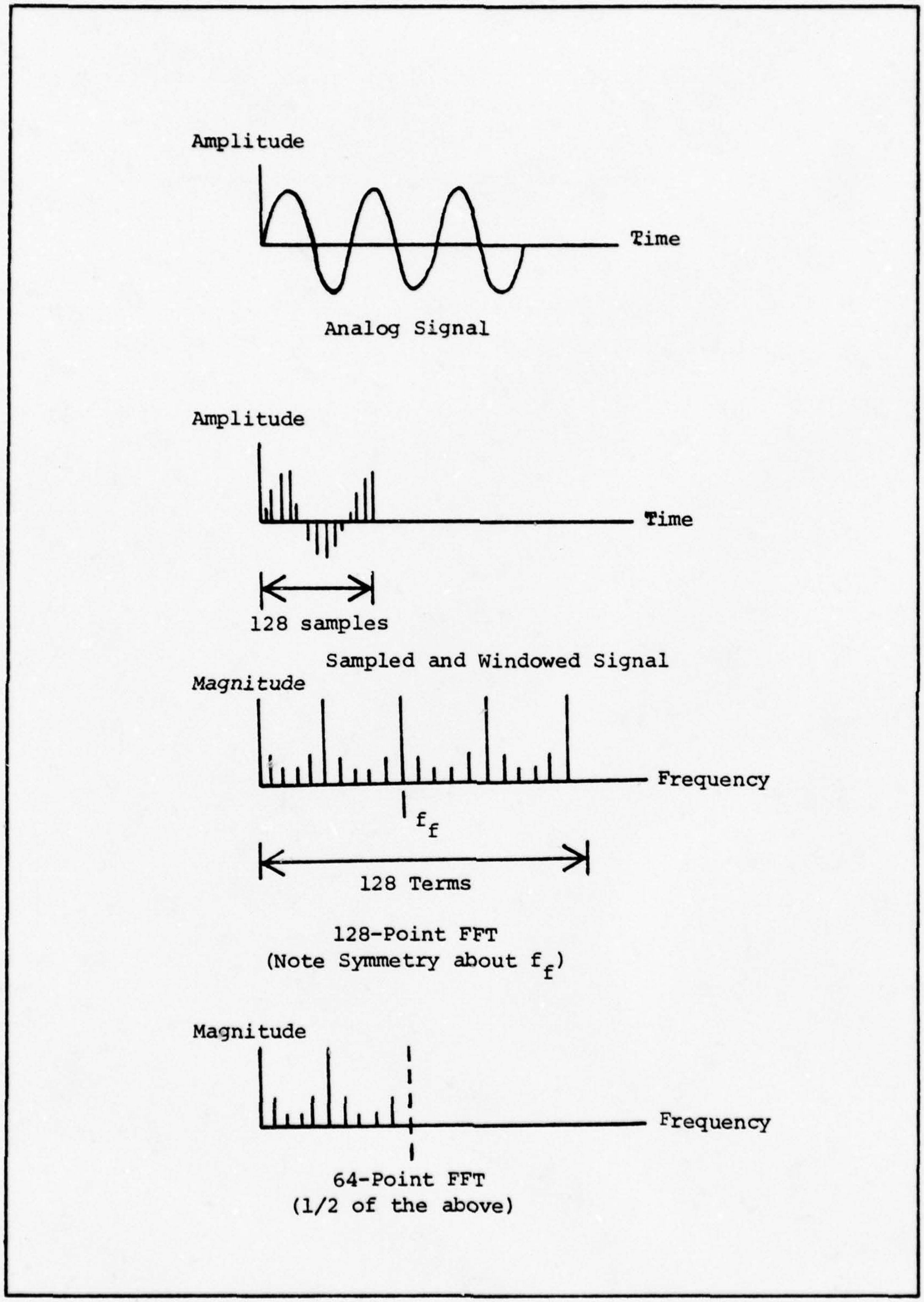


Figure 4. Waveform Sampling and the FFT

IV. Signal Processing

After the analog speech signal was digitized and written onto L-tapes as described in section three, it was readily accessible for subsequent processing. This data was in the form of a digitized output from 64 discrete audio filters each having a center frequency of some integral multiple of 78.125 Hz. Each number represents the averaged output of a particular filter over an interval of 12.8 milliseconds. Thus, each 12.8 millisecond sample of speech was represented by a frequency vector having 64 components.

Channel Compression

Due to the fact that the ear-brain system responds to ratios of frequencies rather than absolute frequency values, the original 64-component vectors were compressed to approximate the frequency response of the ear (Ref 16:85).

The compression was implemented in the following manner. The first six vector components with center frequencies from 78.125 Hz to 468.750 Hz were left unchanged. The remaining 58 vector components were separated into 1/3 octave groups. The magnitudes of the components in each group were added together, thus weighting the values at the high end of the frequency scale. This resulted in a 16-component frequency vector with the center frequencies shown in Table IV.

Another effect of this channel reduction is somewhat analogous to a phono equalization or preemphasis curve. The power density of the high frequency information is boosted

Table IV

Speech Frequencies

Center Frequency Original Data	Center Frequency Reduced Data	Center Frequency Original Data	Center Frequency Reduced Data
78.125	78.125	2578.125	
156.250	156.250	2656.250	
234.375	234.375	2734.375	
312.500	312.500	2812.500	
390.625	390.625	2890.625	2812.500
468.750	468.750	2968.750	
546.875	585.940	3046.875	
625.000		3125.000	
703.125	742.188	3203.125	
781.250		3281.250	
859.375	898.440	3359.375	
937.500		3437.500	
1015.625		3515.625	
1093.750	1132.810	3593.750	3554.690
1171.875		3671.875	
1250.000		3750.000	
1328.125		3828.375	
1406.250	1445.310	3906.250	
1448.375		3984.375	
1565.500		4062.500	
1640.625		4140.625	
1718.750		4218.750	
1796.875	1793.380	4296.875	
1875.000		4375.000	
1953.125		4453.125	
2031.250		4531.250	4453.125
2109.375		4609.375	
2187.500	2226.560	4687.500	
2265.625		4765.625	
2343.750		4843.750	
2421.875		4921.875	
2500.00		5000.000	

to approximately match the power density of the low frequency information.

Spectrogram Development

After the speech data was compressed, the frequency vectors were processed by the analysis portions of the recognition scheme. However, it is helpful to be able to look at the speech data in a format which allows a visual analysis. Much work has been done in visual speech analysis by Potter, Kopp and Green (Ref 20). They found that there were sufficient visual clues in a time-frequency spectrogram to allow trained personnel to do a remarkably accurate job of interpreting the original speech.

To transform the 16-component speech vectors into a form which would resemble a speech spectrogram, a two-dimensional printing scheme was used. The printing scheme adopted plots the numerical magnitudes of each component of the frequency vector on one axis and the time of the occurrence on the other axis. An overprint arrangement, which causes the representation of the frequency component to become increasingly dark as the component's magnitude increases, was used to produce the plots. Figure 5 shows an example of a speech sample along with its representative spectrogram. The speech spectrograms obtained by this process closely mimic the frequency-time spectrograms used by Potter, Kopp and Green. A complete description of the spectrogram overprint scheme is given in Appendix E.

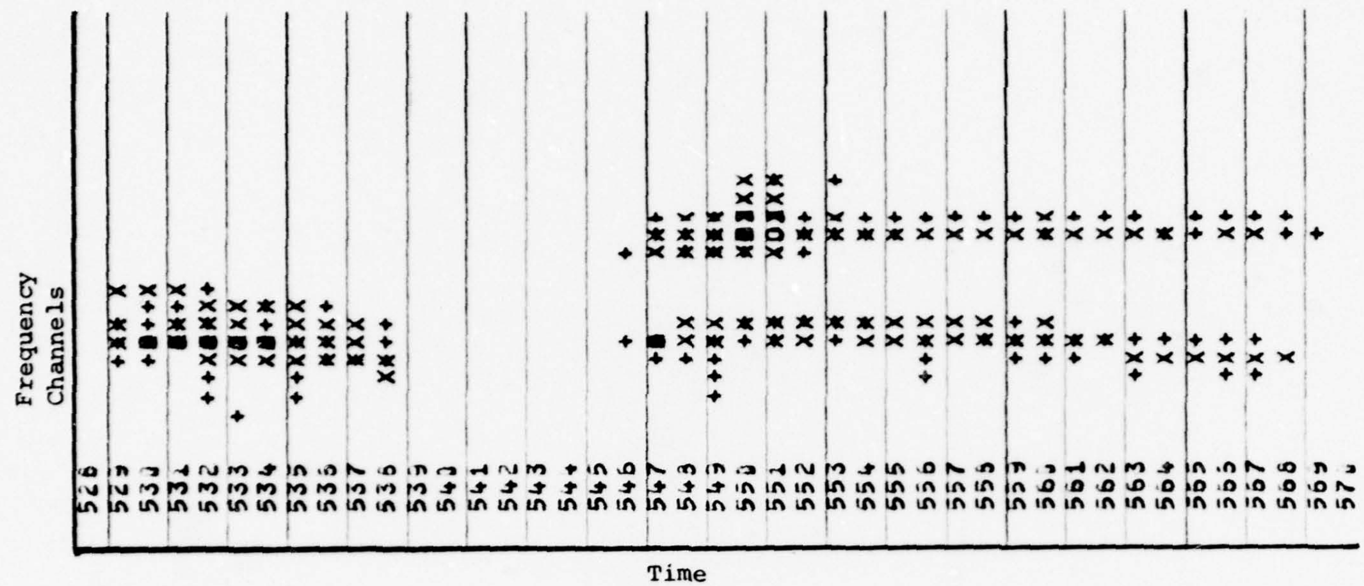


Figure 5. Spectrogram of the Word "Obey"

The computer program, which performs the frequency reduction, preemphasis, and spectrogram output, is listed in Appendix B as OCTAVE1. In addition, OCTAVE1 stores the reduced data onto a computer L-tape to be accessed by later stages of the recognition process.

Due to the very nature of speech, an utterance can contain phonemes with large amounts of energy next to phonemes containing less energy. These lower energy phonemes may not have enough energy to show up in the spectrogram previously described, and be missed even though they contain valuable information. To reconcile this problem, a normalization procedure was performed on each frequency vector. This is analogous to a conventional automatic gain control circuit. Previous work done by Neyman (Ref 19), Hensley (Ref 13), and Guyote and Sisson (Ref 11) emphasized the importance of data normalization.

To accomplish the normalization procedure, the 16-component frequency vectors produced by OCTAVE1 were manipulated vector by vector. Each frequency vector was normalized as follows. The magnitude of the vector was computed by:

$$G = \left(\sum_{i=1}^{16} s_i^{\,2} \right)^{\frac{1}{2}}$$

where the s_i's are the frequency vector components. The column was then normalized by replacing each vector component s_i by s_i^*, where $s_i^* = s_i/G$. This insured that the energy of each frequency vector was equal to one. To further emphasize the low energy phonemes, the new frequency components (s_i^*) were multiplied by 10 to ensure they would be depicted in the output of the normalized spectrogram.

To eliminate the effect of noise being amplified and overprinted in the spectrogram, the value of G was tested. If the value of G was less than a number calculated as the average magnitude of the noise level, the vector was not normalized and was assigned a magnitude of 1.0. Frequency vectors with magnitudes of 1.0 were too small to be represented by a character in the overprint scheme.

A comparison of the spectrogram produced by OCTAVE1 and its column normalized version is shown in Figure 6. As can be seen, the normalized spectrogram provides a more complete representation of the speech data. For example, the normalized spectrogram representation of the word "debt" in Figure 6 clearly shows the ending "t".

The program which implements this normalization procedure is called OCTAVE2 and its listing appears in Appendix B. This program produces a normalized spectrogram for use in the visual phoneme selection analysis, whereas the L-tapes produced by OCTAVE1 were used by the correlation program.

Data Base

Due to the fact that the preprocessing phase is quite lengthy and, at the Wright-Patterson computer facility, can take up to two weeks to obtain results, the investigation

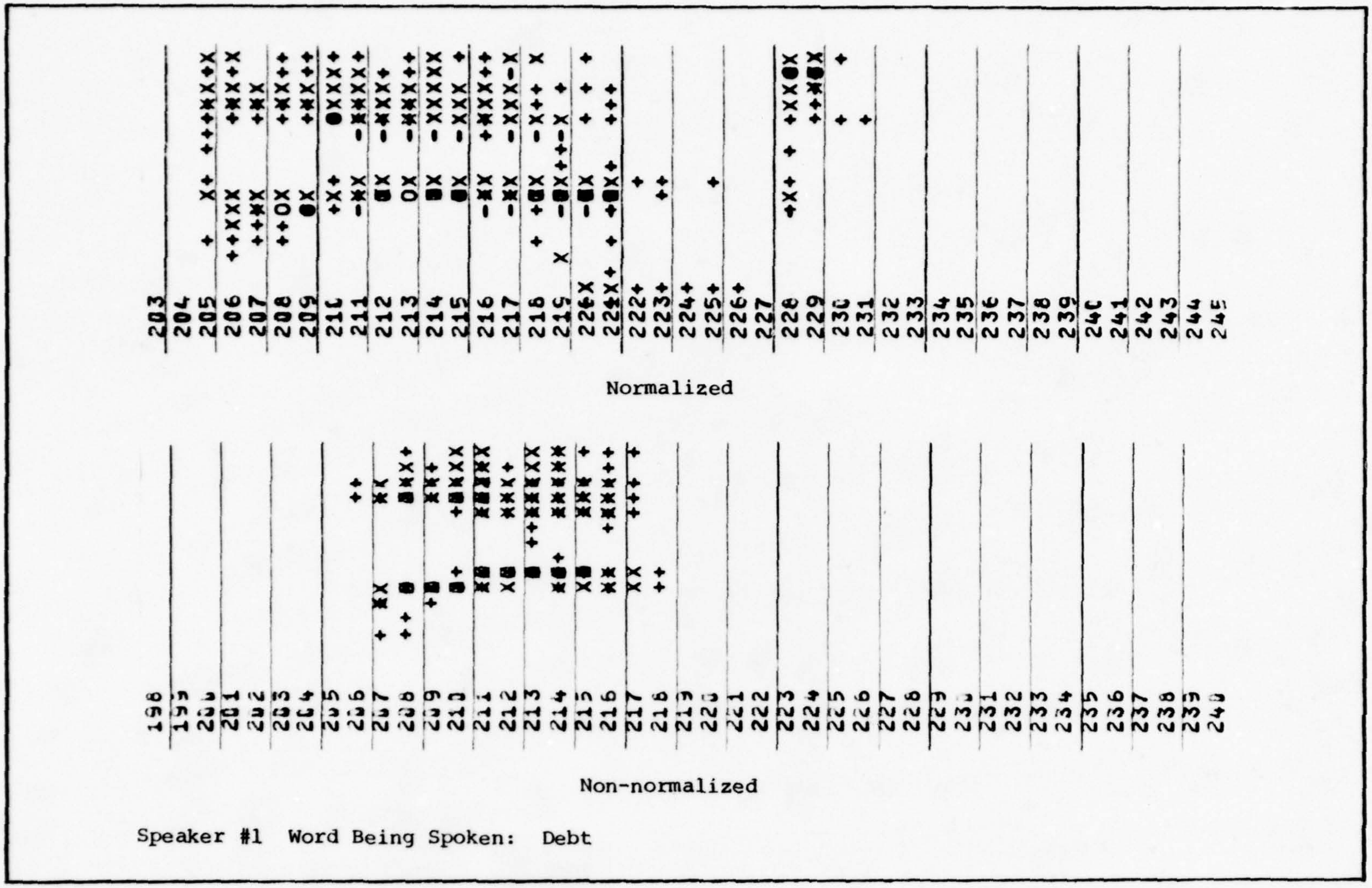


Figure 6. Normalized vs Non-normalized Spectrograms.

was done in two segments. First, a set of averaged prototype phonemes was selected from various word groups spoken by the authors. Then each of the averaged prototype phonemes was correlated with several sentence samples spoken by three different speakers. These sentence samples were composed to insure that the phonemes of interest were represented.

The first set of data is presented in Tables V and VI. Table V shows the various phoneme sounds that were selected for analysis in this research. It also lists the groups of 14 words containing the desired phoneme. From these words the specific phoneme was selected and averaged to give the prototype phoneme.

The sentences listed in Table VI were used for verifying the averaged prototype phonemes selected from the word groups. The first two sentences are composed of words from Table V. The last two sentences contain phoneme sounds like those listed in Table V. This test group of sentences was used to verify that the averaged prototype phonemes selected from the authors' word groups would identify similar phonemes appearing in their continuous and discrete speech.

To aid in the selection of the averaged prototype phonemes, each author recorded the words in Table V and sentences in Table VI according to the following format. The words in each word group were spoken discretely and a five-second 2kHz tone was recorded to mark the beginning and end of each word group. The speaker then recited each of sentences by first saying it discretely and then conti

Table V

Phoneme Word Groups

"B" Sound	"D" Sound	"R" Sound	"T" Sound
bay	debt	rat	taker
babble	debit	read	terminate
batter	ditto	ride	tide
be	donut	robe	tight
bench	dug	rut	toad
bitter	dust	rhino	tore
bite	drafted	rather	tub
boat	danger	rear	tube
bought	dagger	right	through
by	dread	resist	tither
butter	dead	rand	tribe
blend	dodge	rover	tip
bright	dude	rare	twist
bulb	day	rubber	trade
"A" Sound	**"AUH" Sound**	**"E" Sound**	**"O" Sound**
hate	among	leave	go
Abraham	about	each	so
hay	American	me	blow
range	topeka	see	obey
same	santa	even	omit
terminate	mascera	leech	over
wave	another	beat	note
shape	Caruso	meet	those
trace	appear	sleep	pose
angel	attempt	valley	rose
may	accumulate	reek	nose
ray	associate	key	most
say	approximate	egress	both
lay	against	ego	no

Table VI

Verification Sentence Groups

Abraham drafted a note.

See me wave at my associate.

A boy got out the back gate.

Joe was seen around the airplane.

Similarly, each sentence was separated from the other with a five-second 2kHz tone.

The other set of data consisted of the two groups of sentences listed in Table VII. The first six sentences contain words included in Table V. The last six sentences consist of words not appearing in Table V but having similar sounding phonemes. This data set was spoken by three different speakers. The purpose of this data was to investigate how well the averaged prototype phonemes could identify similar phonemes in speech from different speakers.

To keep the data between speakers separate, each would say a sentence discretely and then continuously until he had completed the sentences in Table VII. The five-second 2kHz tone was again used to mark the beginning and end of each sentence.

All the speakers involved were male and from different parts of the country so some dialect influences were present in their speech. Each data set was recorded and processed as described previously and stored in the 16 channel reduced form.

Phoneme Selection

Since the production of a complete set of phonemes was not a goal of this investigation, only the phonemes listed in Table V were pursued. The selection of the phoneme's sound was facilitated by the use of the normalized spectrogram and the pictorial representations of the phonemes from

Table VII

Test Sentence Groups

Abraham drafted a note.
See me wave at my associate.
The batter dug into the dust and made a rut the shape of his foot.
No note to terminate the leave of the American called Caruso was drafted this day.
The bright bulb formed a ray that made a trace of the rubber rat.
From the boat docked in the bay, we saw the rhino, leech, and toad as they lay dead along the tide.
Before the trip, the rabbit rested along the open field of the rancher.
A boy got out the back gate.
Does Dennis teach reading or does Dennis teach driving?
Joe was seen around the airplane.
Take a closer look at Eastman Kodak's bubbling reagents for photo-resist stripping.
Each person at Beckman sees his responsibility aimed toward fabricating better resistors, displays, and drugs.

Potter, Kopp and Green (Ref 20). The normalized spectrogram of each word group was compared with the pictorial representation of a particular phoneme during this process. Once found, the location of a phoneme was recorded by noting the time values printed on the spectrogram. These time values were used during the phoneme extraction process. This procedure was implemented for analyzing the spectrograms of the authors' speech.

The lengths of the phonemes were selected to minimize the transitions between phonemes. However, each specific phoneme was selected to be as long as possible within the above constraint. Also, the phonemes selected from each respective group of words were chosen to be of the same length so that they could be averaged together.

Phoneme Extraction and Averaging

During the analysis of the word groups, the locations of the target phonemes were recorded. This produced a time of occurrence listing for each of the 14 target phonemes in a particular word group. This list was then incorporated into a program called PUNCH, which produced a set of punched cards corresponding to the time of occurrence of the 14 target phonemes in each group of words. A listing of program PUNCH appears in Appendix B.

For each of the authors, this process resulted in a set of punched cards consisting of 14 target phonemes for each of the eight word groups. Thus, by combining the results for both authors, a set of punched cards for 28 target phonemes was produced for each of the desired phonemes.

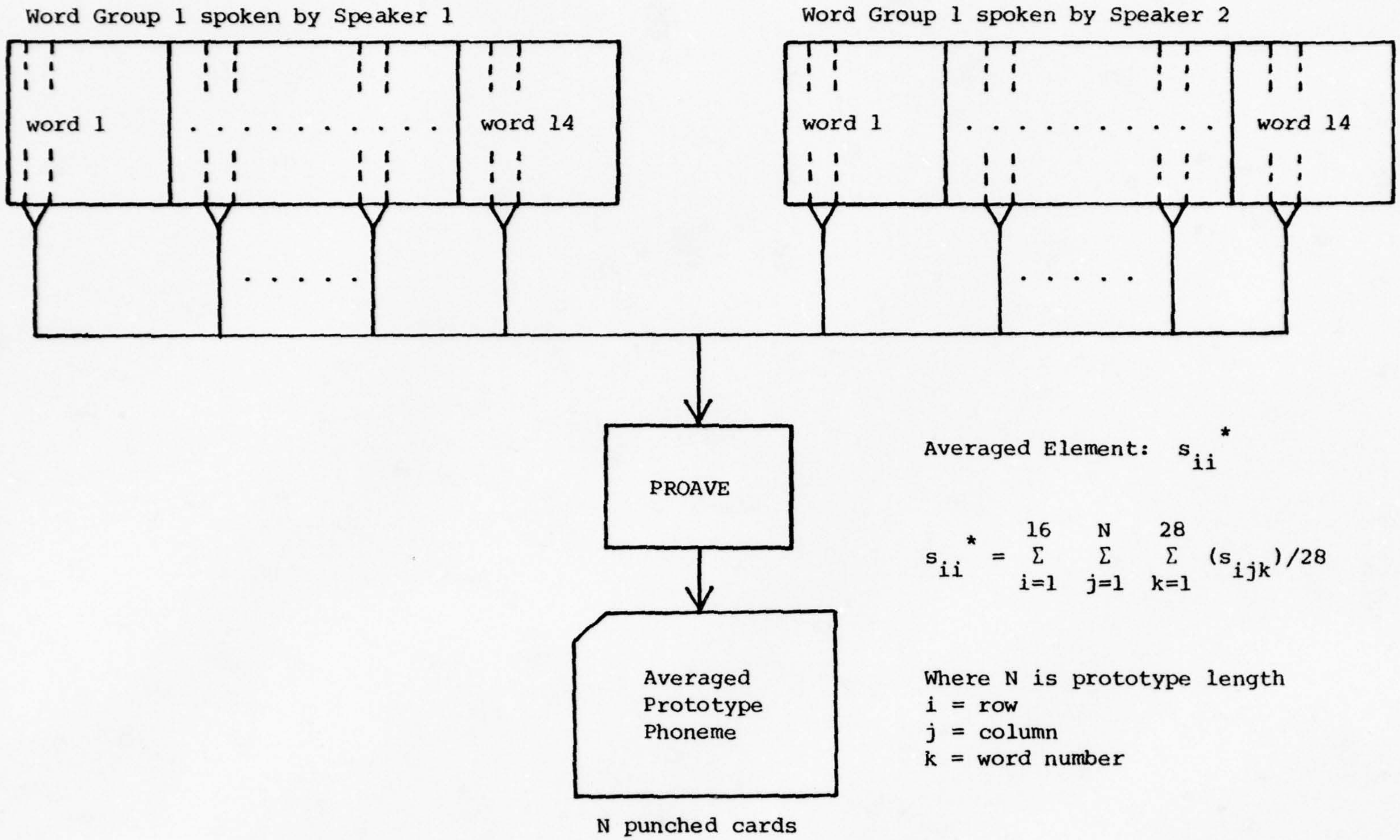
Since these 28 target phonemes were all of the same length, they could be averaged together. The program that performs this averaging process is called PROAVE and its listing appears in Appendix B.

For each frequency vector in a particular phoneme, the program sums up the 28 target phoneme components and divides by 28 to give an averaged value for the component. This process is illustrated in Figure 7. This results in an averaged prototype phoneme which will now be referred to as a prototype phoneme. For each of the desired phonemes, this averaging process was performed and resulted in a set of eight prototype phonemes.

Phoneme Analysis

Since these eight prototype phonemes were formed by averaging like phonemes from two speakers to yield an overall representation of the desired phoneme, they should be able to identify similar phonemes spoken by the same speakers. To facilitate the selection of a set of optimum phonemes to do this process, the averaged prototype phonemes were varied in length and correlated with the groups of words they were selected from. The correlation process is discussed in the next section.

Each prototype phoneme was varied in length to yield nine samples of the prototype. To help with this process,



Averaged Element: s_{ii}^*

$$s_{ii}^* = \sum_{i=1}^{16} \sum_{j=1}^{N} \sum_{k=1}^{28} (s_{ijk})/28$$

Where N is prototype length
i = row
j = column
k = word number

Figure 7. Averaging Scheme

OCTAVE2 was modified to accept punched cards and give a normalized spectrogram. From the results of the correlations of these nine samples of the prototypes with the words they came from, a subset of the three best prototype variations was selected.

These three variations were then correlated with the sentences in Table VI. This resulted in selecting an optimum prototype phoneme that yielded the best results in identifying the desired phoneme in both continuous and discrete speech. The final result of these correlations was a set of eight optimum prototype phonemes.

These eight prototype phonemes were then correlated with the sentences in Table VII. This stage of processing was done to determine whether the averaged prototype phoneme set was characteristic of similar phoneme sounds in the speech of others. The results of this analysis is discussed in a later section.

V. Recognition Processing

The recognition processing phase consists of performing a running crosscorrelation of each averaged prototype phoneme with the sentence samples. Each averaged prototype phoneme consists of an N x 16 array of frequency vector components, where N is the length of a particular phoneme. Similarly, each sentence sample consists of an M x 16 array of frequency vector components, where M is the length of a particular sentence. For this research, it was found that establishing an upper limit on M equal to 700 was adequate for all the sentences analyzed. This equates to an utterance 8.96 seconds long. The output of this correlation calculation is an M x R array where M, as defined above, is the length of the sentence sample and R is the number of phonemes contained in the prototype phoneme set. The value of each element in the M x R array is the result of the correlation of a particular phoneme with the sentence sample at a particular instant of time. In order to implement the correlation computation, it was necessary to prepare both the phonemes and sentence samples with the following operations: column normalization, array augmentation, and phoneme unit normalization. The program used to perform the correlation computations was called CRSCOR and its listing appears in Appendix B.

Column Normalization

An extremely important aspect of the recognition processing phase is the normalization of the data (Ref 11, 13, 19).

The purpose of normalization is to minimize the effect of speaker variation and provide a basis upon which a decision scheme could be devised. The prototype phonemes and the sentence samples were column normalized at each time increment. Each component of the 16-channel frequency vector was normalized according to the formula:

$$X_{N_j} = x_j / [\sum_{i=1}^{16} (x_i)^2]^{\frac{1}{2}}$$

From the spectrograms of the sentences, it was observed that the magnitude of the non-information bearing frequency vectors between the words was limited to an approximate value of 0.5. To prevent this information from entering the correlation calculation, the magnitude of each frequency vector of a particular sentence sample was tested by the following inequality prior to the column normalization calculation:

$$[\sum_{i=1}^{16} (x_i)^2]^{\frac{1}{2}} \leq 0.5$$

If the inequality was satisfied, the frequency vector was not column normalized, but instead was assigned a magnitude of 0.001 to insure that the correlation values for these components were very small numbers.

The column normalization calculation was the only normalization performed on the sentence samples. In addition to being column normalized, the phoneme arrays were unit normalized after their Discrete Fourier Transform (DFT) calculation.

Array Augmentation

The motivation of this research was directed toward the correlation of a two-dimensional averaged prototype phoneme with its "variations" that occur in everyday speech. Since real-time correlation calculations require enormous amounts of computations, even large-scale computers, such as the CDC Cyber 6600, would require excessive amounts of time to do the calculations. Therefore, the computer algorithm used in this research was based on the DFT.

The recent innovations in the past ten years for computing the DFT of matrices such as the Fast Fourier Transform (FFT) have made it possible to greatly reduce the amount of computations needed for correlation (Ref 1, 2, 12). However, the use of DFT theory requires that certain inherent problems be considered. The most critical problems are aliasing, leakage, and end-effect.

Aliasing is a term that refers to the fact that high-frequency components of a time function can impersonate low frequencies if the sampling rate is too low (Ref 2). This problem was avoided during the digitization process by using a 10 kHz sampling rate which was twice the highest speech frequency (5 kHz).

The problem of leakage is inherent in the Fourier analysis of any finite record of data. Furthermore, leakage is directly related to the method by which the digitized samples of an analog signal are selected or windowed. The ideal window function is one that would localize the contribution of a given frequency in a narrow main lobe while reducing the amount of "leakage" through the side lobes. It is well known that both of these criteria cannot be optimized simultaneously and that the selection of a window function is a compromise between leakage and the width of the main lobe. Neyman tested the Hanning and rectangular window functions and reported that the overall recognition results were not altered when either window function was used (Ref 19). The rectangular window function was used in this research since it was easiest to implement.

The problem referred to as end-effect occurs when two functions are correlated because of the periodicity imposed by the DFT. The correlation computations done in this research required that a buffer be included in the transformed functions so that the function which is being moved along the time axis does not encounter duplicates of the data being correlated. This problem was solved by augmenting the arrays in the following manner.

Let P_{ij} be the prototype phoneme array and S_{ij} be the sentence array. Let P be the number of points defining the length of P_{ij} and S the number of points defining the length of S_{ij}. Choose a V such that:

$$V \geq P + S - 1$$

and

$$V = 2^n$$

where n is an integer.

The augmented arrays S_{kb} and P_{kb} for S_{ij} and P_{ij} respectively, are defined as follows:

$$
S_{kb} = \begin{cases} 0 & k = 0, 1, 2, \ldots, V-S \\ & b = 0, 1, 2, \ldots, 15 \\ s_{ij} & k = V-S+1, V-S+2, \ldots, V-1 \\ & b = j = 0, 1, 2, \ldots, 15 \\ & i = 0, 1, 2, \ldots, S-1 \\ 0 & k = 16, 17, \ldots, V-1 \\ & b = 16, 17, \ldots, 31 \end{cases}
$$

$$
P_{kb} = \begin{cases} p_{ij} & k = i = 0, 1, 2, \ldots, P-1 \\ & b = j = 0, 1, 2, \ldots, 15 \\ 0 & k = P, P+1, \ldots, V-1 \\ & b = 0, 1, 2, \ldots, 15 \\ 0 & k = 0, 1, 2, \ldots, V-1 \\ & b = 16, 17, \ldots, 31 \end{cases}
$$

The array transformation is illustrated in Figure 8.

The augmented arrays serve to embed the prototype phoneme and sentence arrays in a sufficient buffer of zeroes to eliminate the end-effect problem. Furthermore, the augmented arrays, which are both 32 x 64 arrays, can be correlated to yield a 32 x 64 array.

The limiting values of V and S were determined by the AFIT FFT subroutine (FOURT) (Ref 12). For this research, V was limited to 64 and S to 48. Since the size of the sentence samples were fixed, it was necessary to limit the size of the prototype phonemes to affect a complete correlation. This situation was reconciled by incorporating an overlap variable, T, into the structure of the augmented sentence array. As each sequential sentence array was augmented,

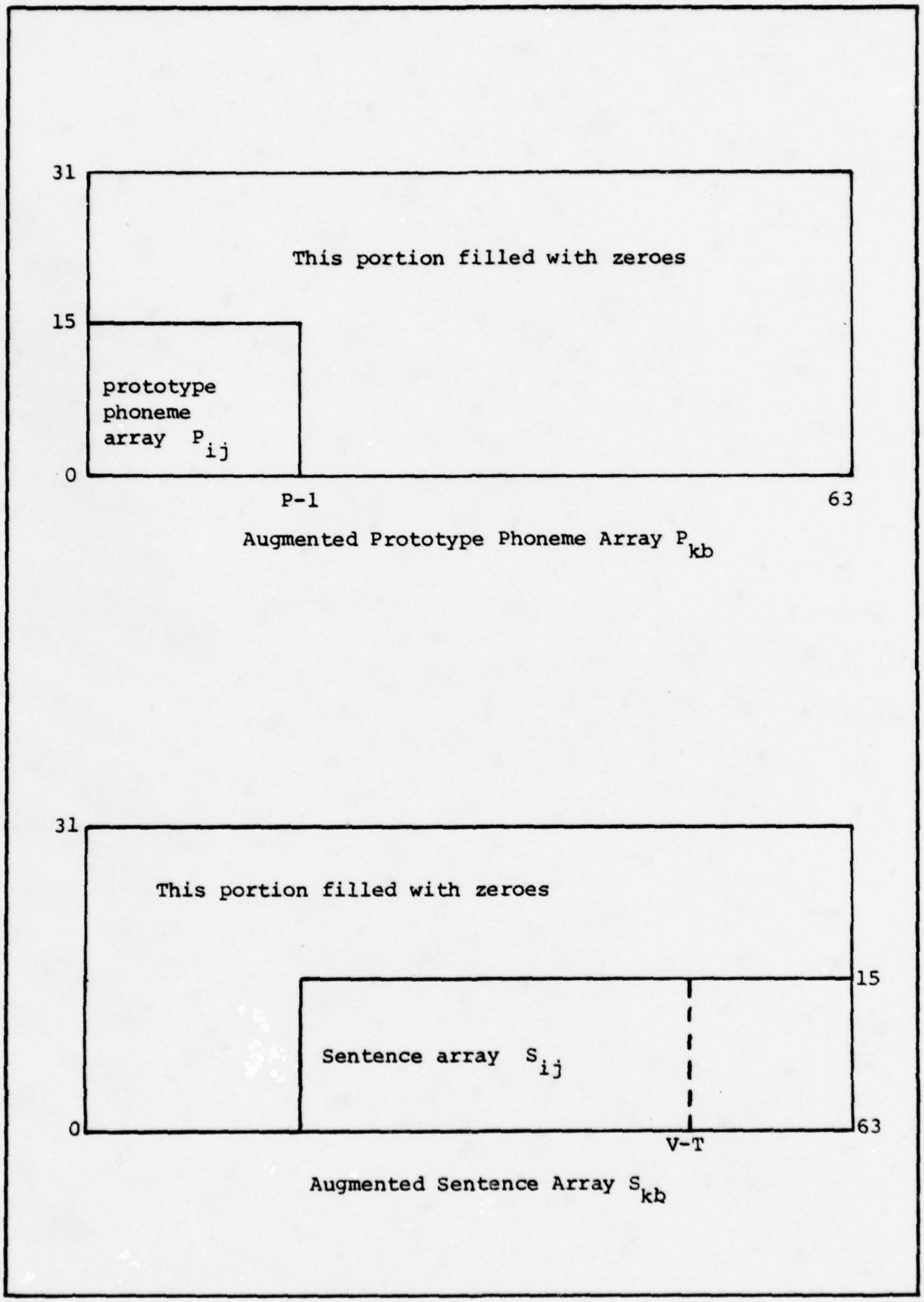


Figure 8. Augmented Arrays

only (S-T) new values were included in the array. The last T values of the previous sentence sample became the first T values of the new sentence sample. For this research, T was fixed at 8 and this meant that the length of the largest prototype phoneme was limited to 15 to insure that the overlap was greater than half the length of the largest prototype phoneme.

Fast Fourier Transform

Following the augmentation of both the prototype phoneme and sentence arrays, their two-dimensional DFT was computed using the FFT algorithm, FOURT (Ref 12). The transformed arrays were calculated as follows:

$$P_{rs} = \sum_{k=1}^{K} \sum_{b=1}^{L} p_{kb} \exp[(-2j\pi)(\frac{kr}{K} + \frac{bs}{L})]$$

$$S_{rs} = \sum_{k=1}^{K} \sum_{b=1}^{L} s_{kb} \exp[(-2j\pi)(\frac{kr}{K} + \frac{bs}{L})]$$

where $j = \sqrt{-1}$

The complex conjugate of the transformed prototype phoneme array was then formed as P^*_{rs}. The phoneme array was then ready for the unit normalization process.

Unit Normalization

The prototype phoneme array was column normalized before the DFT computation. The column normalization calculation, as discussed earlier, insured that the energy of each frequency vector was unity. However, the net energy of each prototype phoneme remained a direct function of its length. As a result, the correlation value for a perfect match between a prototype phoneme and a candidate phoneme *in a given* sentence sample could be compromised by a long prototype phoneme. Unit normalization was done to insure that each prototype, no matter what its length, had unit energy prior to the correlation computation.

Unit normalization was computed as follows:

$$P^*_{Nrs} = p^*_{rs} \,/\, (\text{Energy})^{1/2}$$

where

$$\text{Energy} = [\sum_{r=1}^{32} \sum_{s=1}^{64} (p^*_{rs})^2]$$

It is noted here that the energy computed above is the energy of the prototype phoneme after column normalization. Since there are N columns of unit energy, the net energy of a given column normalized prototype phoneme is N. Thus, to unit normalize a prototype phoneme, each element of the column normalized array is divided by $N^{1/2}$. The significance of this particular calculation with respect to correlation will be discussed in the next section.

Correlation Computations

After the unit normalization of the prototype phoneme array, the element-by-element product was computed as:

$$Z_{rs} = S_{rs} \cdot P^*_{Nrs}$$

The result of this multiplication is equivalent to correlation in the time domain. The desired correlation values were obtained by computing the inverse transform of Z_{rs}. The inverse transform was computed as follows:

$$z_{kb} = \frac{1}{KL} \sum_{k=1}^{K} \sum_{b=1}^{L} Z_{rs} \quad \exp[2j\pi(\frac{kr}{K} + \frac{bs}{L})]$$

Following the inverse transform computations, the correlation vector for a particular phoneme was formed by taking the first, or zero shift, row from the z_{kb} array. This row was transferred to the correlation array as follows:

$$C_i = z_{kb}$$

where

$k = S, S + 1, \ldots, V-T$

$b = 1$

$i = 1, 2, \ldots, R$ (The particular phoneme correlated with the sentence)

The first (S - 1) values were discarded to compensate for the end-effect. The last T values account for the overlap factor.

Before the correlation array could be used in a decision scheme a basis for comparing the correlation values over all time for all prototype phonemes had to be developed. Obviously a larger prototype phoneme will have a greater maximum correlation value when it encounters a large candidate phoneme than will a short prototype phoneme. It would be highly desirable to normalize the maximum correlation values to unity so that the performance of all prototype phonemes could be compared. Since the prototype phonemes were column and unit normalized and the sentence samples were column normalized, the maximum correlation obtainable by a prototype phoneme which encounters an exact replica of itself would be $N^{\frac{1}{2}}$, where N is the length of the prototype phoneme. A mathematical derivation of this fact is presented in Appendix D.

It was possible to ensure that the maximum correlation value for any prototype phoneme was unity by simply dividing the correlation values for each prototype phoneme by the square root of the length of the prototype phoneme ($N^{\frac{1}{2}}$). Since the computed correlation values for any prototype phoneme will be restricted between zero and unity, the relative performance of all prototype phonemes can be compared and evaluated in a decision scheme.

Data Storage

Following the completion of the correlation computations, the results were stored in permanent file in the form of an M x R array where M is the length of the particular sentence sample and R is the number of phonemes contained in the prototype phoneme set. In this form, each element in the correlation array represents the correlation of a particular prototype phoneme with the sentence sample at a particular instant of time. The structure of the array is illustrated in Figure 9.

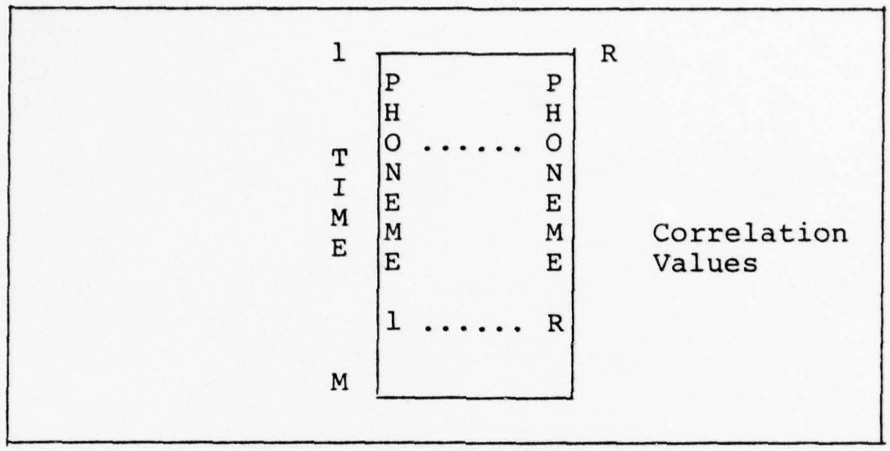


Figure 9. Correlation Array

In addition, this mode of storage provided optimum flexibility for exercising the decision scheme during its development.

Correlation Plot Output

Insight into the development of a decision scheme was enhanced from the analysis of the results of the correlation routine. A plotting routine was designed that permitted the selection of a prototype phoneme's correlation values to be sent to the Calcomp plotter for processing. This routine graphically depicts the running correlation of a particular sentence or word group.

Figure 10 shows the output of the "B" prototype as it was correlated with the words: "Bench, Bitter, and Bite". The word group started at time interval 20, and the "B" phoneme correlated with the beginning "B" of each word. The

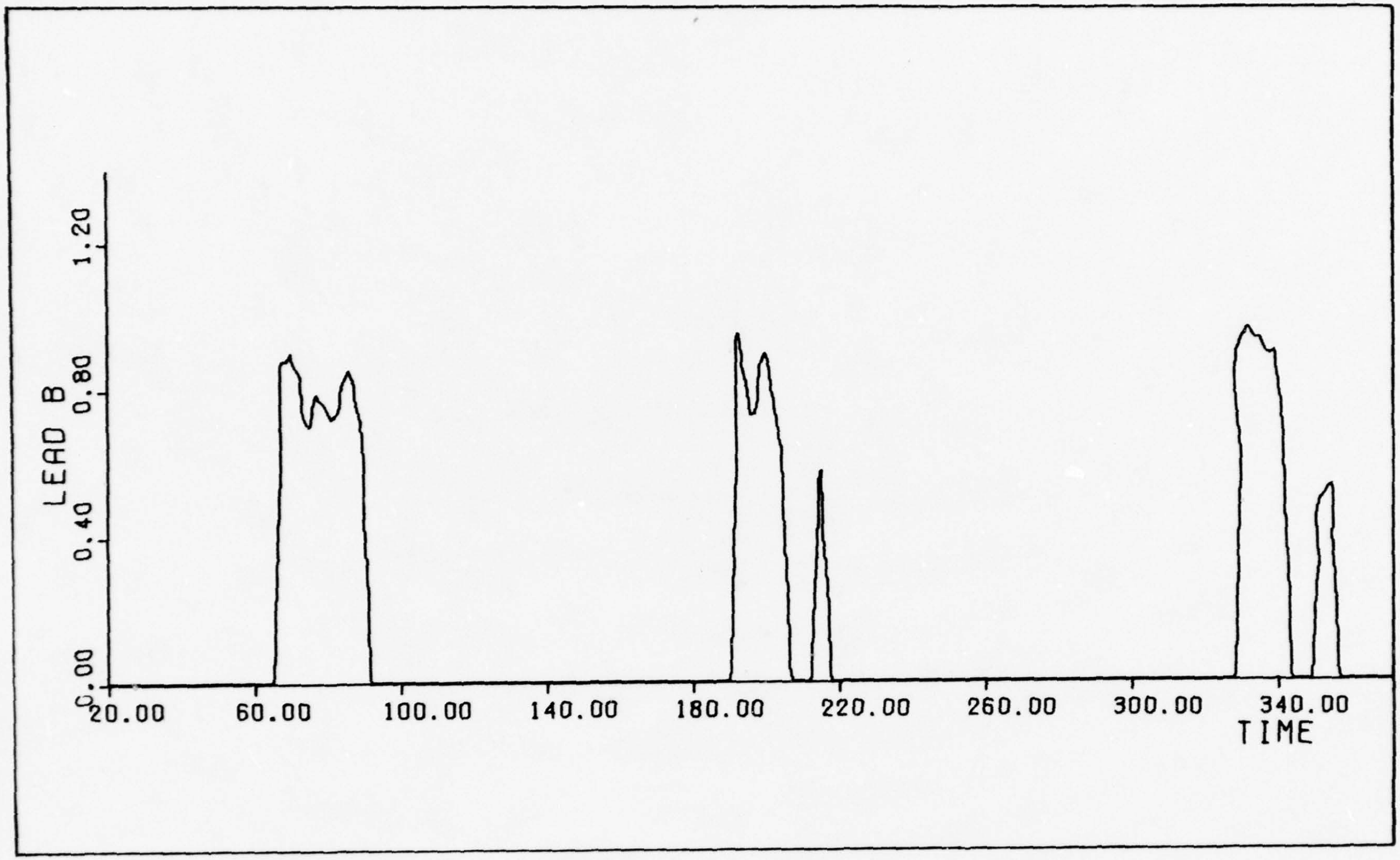


Figure 10. Correlation Plot Output

program that plots the correlation values was called FPLOT and its listing appears in Appendix B.

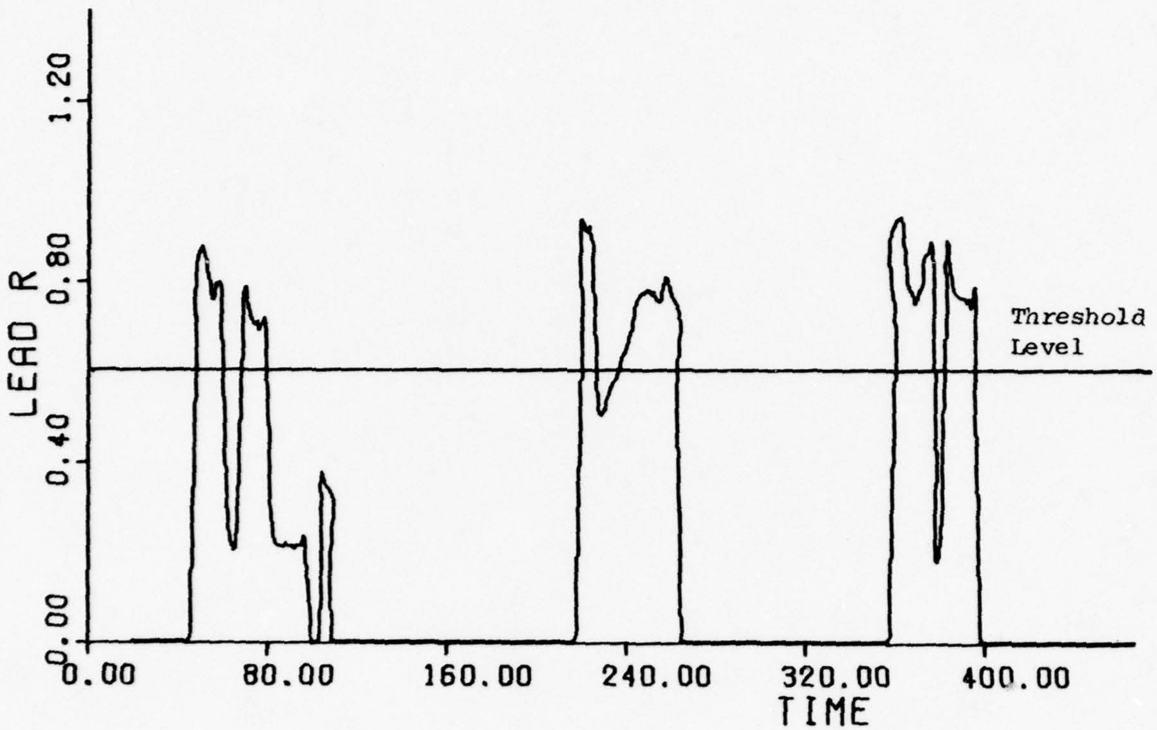
VI. Decision Scheme

The purpose of implementing a decision scheme was to locate the areas in a given sentence correlation array where a particular phoneme might have occurred. The organization of the correlation array facilitated a visual comparison of all prototype phoneme correlations by comparing their magnitudes in each row of the array. In addition, the dynamic performance of each prototype phoneme within a given sentence was readily analyzed from the correlation plot output. As a result, the final decision scheme tested the correlation array values against three criteria to arrive at a phoneme's location and identification. The three criteria used were: threshold, rate-of-change of correlation values, and endurance. They are defined as follows:

Threshold

The correlation array was first processed for magnitudes which were greater than or equal to a selected threshold. Amplitudes satisfying the threshold criteria were left unchanged; all others were set equal to zero. This threshold operation can be visualized by drawing a horizontal line on a given correlation plot as shown in Figure 11.

Thus, an appreciation for when a phoneme occurred can be gained by observing the peaks that lie above the threshold level. For this research, a threshold level of 0.6 was used. This threshold level was experimentally determined to be just above the average correlation value level for all the



Speaker 1: R phoneme correlated with: "Rat, Read, Ride"

Figure 11. Threshold Criteria Illustration

speech data processed. In addition, this level permitted the largest number of prototype phoneme correlation values to be selected for testing by the other criteria in the decision scheme.

Rate-of-Change of Correlation Values

An analysis of the correlation array data indicated that if a valid phoneme was located in a given sentence, the correlation magnitudes above the threshold level did not change dramatically with respect to time. On the other hand, when an invalid phoneme had correlation values above the threshold level, the rate-of-change between correlation values as time increased was more dramatic. Thus, for this research, correlation values with magnitudes above the threshold level were left unchanged if each preceeding and successive correlation value was within 61 percent of the value being tested. Otherwise, those correlation values above the threshold level not satisfying the above rate-of-change criteria were set to zero. The 61 percent rate-of-change value was experimentally determined such that it deleted only isolated correlation values that were surrounded by zeroes, but would preserve groupings of two or more correlation values.

Endurance

To further limit the opportunity for false hits, a time endurance criteria was incorporated into the decision scheme to eliminate the momentary correlation values that rise above the threshold level and satisfy the rate-of-change criteria. The endurance criteria was implemented by scanning the correlation array from beginning to end for a particular phoneme. When a correlation value above the threshold level was detected, a marker was set. When the correlation value fell below the threshold level, another marker was set. Then, the time increment between the two markers was compared to some specified percentage of the length of the prototype phoneme, for example, one-half its length. If the time increment of the "hit" was less than the "adjusted" length of the prototype phoneme, that portion of the correlation array between the markers was set to zero.

Thus, following the endurance processing, the correlation array for a particular prototype phoneme with a given sentence consisted of those values above a desired threshold level which did not change from value to value by more than 61 percent, and which stayed above this threshold for an amount of time dependent on the prototype phoneme's length.

<u>Ranking</u>

Following the threshold, rate-of-change, and endurance processing, the correlation array was ready for the final decision output. It might seem logical to merely select the largest correlation value at each time increment and use the corresponding prototype phoneme as the final selection. However, previous research efforts cite the conclusion that nominally correct phonemes are not always produced with a high frequency of occurrence in normal speech and that higher order decision schemes are used by the brain to determine the actual word content of a sentence (Ref 11, 13, 19). As a result, it was decided to implement a ranking algorithm which would simply print up to eight phoneme selections for each time increment by listing the selections from the highest to lowest correlation values. The advantages to this output format are:

1. It could be stored and used by additional decision schemes that take into account higher order levels of word structure such as syntax, grammar, and context.

2. The relative performance of each of the prototype phonemes could be readily analyzed. Invalid decisions could be noted so as to determine methods which would yield more correct results.

This was the final computer processing stage in this research. An illustration of the overall decision processing scheme is shown in Figure 12. The program which performs this decision scheme is called DECIS and its listing appears in Appendix B.

Threshold Process										
Proto 1 (Length 4)	.5	.5	.7	.8	.9	.9	.8	.7	.5	.4
Proto 2 (Length 6)	.5	.4	.7	.3	.5	.4	.8	.9	.9	.8
Proto 3 (Length 5)	.3	.8	.9	.6	.5	.5	.4	.5	.4	.3
Time	1	2	3	4	5	6	7	8	9	10

↓ Threshold Level = 0.6

Rate-of-Change Process										
Proto 1 (Length 4)	0	0	.7	.8	.9	.9	.8	.7	0	0
Proto 2 (Length 6)	0	0	.7	0	0	0	.8	.8	.9	.8
Proto 3 (Length 5)	0	.8	.9	.8	0	0	0	0	0	0
Time	1	2	3	4	5	6	7	8	9	10

↓ 61 Percent Rate-of-Change

Endurance Process										
Proto 1 (Length 4)	0	0	.7	.8	.9	.9	.8	.7	0	0
Proto 2 (Length 6)	0	0	0	0	0	0	.8	.8	.9	.8
Proto 3 (Length 5)	0	.8	.9	.8	0	0	0	0	0	0
Time	1	2	3	4	5	6	7	8	9	10

↓ Endurance = 0.5 x Proto Length

Ranking Process										
Proto 1 (Length 4)	0	0	.7	.8	.9	.9	.8	.7	0	0
Proto 2 (Length 6)	0	0	0	0	0	0	.8	.8	.9	.8
Proto 3 (Length 5)	0	.8	.9	.8	0	0	0	0	0	0
Time	1	2	3	4	5	6	7	8	9	10

Time	Phonemic Output	
1	--	
2	Proto 3	
3	Proto 3	Proto 1
4	Proto 3	Proto 1
5	Proto 1	
6	Proto 1	
7	Proto 1	
8	Proto 1	Proto 2
9	Proto 2	Proto 1
10	Proto 2	

Figure 12. Decision Scheme Process

VII. Results

The results obtained will be presented in three parts. The first part consists of the results of the prototype phonemes correlated with the authors' word groups used to select the prototype phonemes. The second part presents the results of the prototype phonemes correlated with the verification sentences spoken by the authors. The final part contains the results of the prototype phonemes correlated with sentences spoken by three different speakers.

Scoring Philosophy

As previously discussed, the decision scheme produced a ranking of the eight correlation values in descending order for each speech time segment. The scoring was accomplished by noting the relative position of the phoneme in the word being analyzed and its ranking in the decision program's output. If the phoneme was in the correct position within the word and was ranked within the first three choices in the decision program's listing, the phoneme was considered to be "located". If the phoneme was in the correct position and was the first choice in the decision program's listing, the phoneme was considered to be "identified". Otherwise, the phoneme was considered to be missed.

Only the eight phonemes under study were scored. If a word contained other phoneme sounds, they were not scored. For instance, in the word "the" only the "Auh" sound was scored, the "TH" sound was ignored since none of the eight phonemes being tested resembled it. The various symbols used in the scoring process are listed in Table XIX in Appendix C. The scoring charts for the words and sentences are presented in Tables XX thru XL in Appendix C.

Analysis and Calculations

Evaluation of the results obtained for various types of speech was based on the percent correct value, P_C; this same criterion was used by prior researchers (Ref 11, 13, 19):

$$P_C = (A/B) \times 100\%$$

where:

A is the quantity of correct phonemes "identified" or "located"

B is the total number of phoneme patterns considered

Thus, P_C is the percentage of the phoneme patterns correctly "identified" or "located".

The binomial distribution was used in developing the error rate calculations for the sentence phoneme analysis. The use of this distribution rather than some other distribution was determined by the manner in which the phonemes were located. Since the sentence phonemes were either "located" or "missed", the binary values one and zero can be used to represent these events. This binary representation of the sentence phoneme location implies a binomial distribution for the error rate.

The error rate for F misclassified events out of B possible random events is:

$$\hat{P}_E = \frac{F}{B} = 1 - P_C$$

But as the events are increased such that B approaches infinity the error rate, by definition, is:

$$\lim_{B\to\infty} [\hat{P}_E = \frac{F}{B}] = P_E$$

It was assumed that the number of random events B was large enough so that $\hat{P}_E = P_E$.

If the error rate calculated for F misclassified events out of B possible random events is P_E, then F has the binomial distribution (Ref 5:74).

$$P_E(F) = (\frac{B}{K})\ P_E{}^F\ (1-P_E)^{B-F}$$

The expected value of F is:

$$E\{F\} = BP_E$$

Thus the expected value of the error rate is:

$$E\{P_E\} = P_E \quad \text{(Ref 21:158)}$$

The variance of F for B random events is:

$$VAR\{F\} = BP_E\ (1-P_E)$$

Thus the variance of the error rate P_E is:

$$VAR\{P_E\} = \frac{P_E(1-P_E)}{B} \quad \text{(Ref 21:158)}$$

The 95 percent confidence intervals for error-rate estimates for the binomial distribution were determined from Figure 3.6 presented in the text written by Duda and Hart (Ref 5:75). Confidence intervals give statistical bounds for the certainty of an event. Thus, for this research the 95 percent confidence interval is the interval over which the error rate for a given sample size exists 95 percent of the time.

The summarized results for each of the three data groups are presented by giving the percent correct for each group of words or sentences for each speaker, and a combined score for the speakers. The total number of correct choices and the total number of events for each data group was determined so that the probability of being correct, probability of being in error, variance of the error, and the 95 percent confidence interval for the probability of error could be calculated. This information is presented in Tables VIII, X, and XII. In addition, the detailed scoring of each of the word groups or sentences is presented in Appendix C.

Word Groups

The results for the eight word groups are summarized in Table VIII. The percent correct for phoneme location and identification for each word group is tabulated for author 1, author 2, and the combination of the two. The first three columns present the results for the particular phoneme that was calculated from each of the word groups. For example,

Table VIII

Analysis of the Word Groups

Word Group	Scoring Descriptor	Desired Phoneme Author 1	Desired Phoneme Author 2	Desired Phoneme Combined	All Phonemes Author 1	All Phonemes Author 2	All Phonemes Combined
B	Located	93.3%	93.8%	93.5%	86.2%	90.0%	88.1%
	Identified	73.3%	87.5%	80.6%	65.5%	76.7%	71.2%
R	Located	100.0%	100.0%	100.0%	97.1%	100.0%	98.6%
	Identified	84.2%	100.0%	92.1%	80.0%	97.1%	88.6%
D	Located	94.4%	100.0%	97.2%	91.2%	100.0%	95.6%
	Identified	94.4%	83.3%	88.9%	91.2%	82.4%	86.8%
T	Located	100.0%	100.0%	100.0%	96.3%	100.0%	98.0%
	Identified	100.0%	78.5%	89.3%	85.1%	77.7%	81.5%
O	Located	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%
	Identified	100.0%	100.0%	100.0%	86.0%	100.0%	93.1%
E	Located	100.0%	100.0%	100.0%	95.0%	95.0%	95.0%
	Identified	92.8%	100.0%	96.4%	90.0%	95.0%	92.5%
A	Located	92.8%	100.0%	96.4%	91.6%	100.0%	95.8%
	Identified	71.4%	100.0%	85.7%	75.0%	95.8%	85.4%
Auh	Located	100.0%	100.0%	100.0%	93.3%	100.0%	96.7%
	Identified	66.6%	80.0%	73.3%	66.6%	80.6%	73.7%
Total	Located	$\frac{120}{123}$ =97.5%	$\frac{123}{124}$=99.1%	$\frac{243}{247}$=98.3%	$\frac{207}{221}$=93.6%	$\frac{219}{223}$=98.2%	$\frac{426}{444}$=95.9%
	Identified	$\frac{105}{123}$ =85.3%	$\frac{113}{124}$=91.1%	$\frac{218}{247}$=88.2%	$\frac{176}{221}$=79.6%	$\frac{195}{223}$=87.4%	$\frac{371}{444}$=83.5%

(continued)

Table VIII--continued

Analysis of the Word Groups

	Desired Phoneme		All Phonemes	
	Located	Identified	Located	Identified
Expected Value of P_E	.017	.118	.041	.165
Variance of P_E	6.8×10^{-5}	4.2×10^{-4}	8.9×10^{-5}	3.1×10^{-4}
95% Confidence Interval Around P_E	.009 to .03	.06 to .17	.035 to .1	.13 to .2

the B phoneme was scored with the B words. The last three columns are the results when all eight phonemes were scored with the words. The data used to make these calculations can be found in the corresponding tables in Appendix C.

As can be seen from Table VIII, the net probability of locating a particular phoneme for both authors is 0.983. The 95 percent confidence interval for the probability of error is 0.009 to 0.03. Thus, one can be 95 percent confident that for 247 trials the desired phoneme will be located at least 97 percent of the time. There was an 88.2 percent identification rate for these same trials, which produced a 95 percent confidence error interval of 0.06 to 0.17.

When all phonemes were scored, there were 444 trials. This yielded a 95.9 percent location rate that corresponds to a 95 percent confidence error interval of 0.035 to 0.10. Finally, the identification rate was 83.5 percent which yielded a 95 percent confidence error interval of 0.13 to 0.20.

Verification Sentences

The sentences used for verification of the phoneme set are listed in Table IX. The results for all of the located and identified phonemes are presented in Table X. As can be seen from Table X, the net probability of locating all the phonemes scored for discrete speech is 0.911 and the corresponding 95 percent confidence error interval is 0.0 to 0.06. For the 68 events scored, the net probability of identifying

Table IX

Verification Sentences

Sentence Number	Sentence
1.	Abraham drafted a note.
2.	See me wave at my associate.
3.	A boy got out the back gate.
4.	Joe was seen around the airplane.

Table X

Analysis of the Verification Sentences

Sentence	Scoring Descriptor	Discrete Author 1	Discrete Author 2	Discrete Combined	Continuous Author 1	Continuous Author 2	Continuous Combined
1.	Located	90.0%	100.0%	95.0%	80.0%	90.0%	85.0%
	Identified	70.0%	80.0%	75.0%	50.0%	70.0%	60.0%
2.	Located	85.7%	100.0%	92.8%	85.7%	100.0%	92.8%
	Identified	71.4%	85.7%	78.5%	85.7%	71.4%	78.5%
3.	Located	66.6%	100.0%	83.3%	77.7%	88.8%	83.3%
	Identified	55.5%	100.0%	77.7%	66.6%	66.6%	66.6%
4.	Located	100.0%	87.5%	93.7%	87.5%	100.0%	93.7%
	Identified	87.5%	75.0%	81.2%	37.5%	87.5%	62.5%
Total	Located	$\frac{29}{34}$=85.2%	$\frac{33}{34}$=97%	$\frac{62}{68}$=91.1%	$\frac{28}{34}$=82.3%	$\frac{32}{34}$=94.1%	$\frac{60}{68}$=88.2%
	Identified	$\frac{24}{34}$=70.5%	$\frac{29}{34}$=85.2%	$\frac{53}{68}$=77.9%	$\frac{20}{34}$=58.8%	$\frac{25}{34}$=73.5%	$\frac{45}{68}$=66.1%

	Discrete Located	Discrete Identified	Continuous Located	Continuous Identified
Expected Value of P_E	.009	.221	.118	.339
Variance of P_E	.00013	.0025	.0015	.0033
95% Confidence Interval Around P_E	0.0 to 0.06	.13 to .35	.055 to .21	.23 to .46

the phonemes was 0.779 and the corresponding 95 percent confidence error interval is 0.13 to 0.35. The continuous speech had a slightly lower net probability for location of 0.882 and the corresponding 95 percent confidence error interval was 0.055 to 0.21. Finally, the net probability of identifying the phonemes was 0.661 and the corresponding 95 percent confidence error interval was 0.23 to 0.46.

Test Sentences

The phonemes were further evaluated by scoring them with the sentences listed in Table XI that were spoken by three different speakers. Due to a preprocessing error, either in the recording equipment or the digitizing equipment, noise was introduced into the sentences. Thus, only the sentences presented in the Table XI were scored. Further, as can be seen from Table XII, not all speakers for these sentences were scored due to the noise problem. However, the sample size is still appreciable even with these losses.

As can be seen from Table XII, the net probability of locating all the phonemes scored for discrete speech is 0.779 and the corresponding 95 percent confidence error interval is 0.17 to 0.26. For the 377 events scored, the net probability of identifying the phonemes was 0.623 and the corresponding 95 percent confidence error interval is 0.33 to 0.42. The continuous speech had a slightly lower net probability for location of 0.661 and the corresponding 95 percent confidence error interval was 0.29 to 0.40. Finally, the net probability

Table XI

Test Sentence Group

Sentence Number	Sentence
1.	Abraham drafted a note.
2.	No note to terminate the leave of the American called Caruso was drafted this day.
3.	The bright bulb formed a ray that made a trace of the rubber rat.
4.	From the boat docked in the bay, we saw the rhino, leech, and toad as they lay dead along the tide.
5.	Before the trip, the rabbit rested along the open field of the rancher.
6.	Does Dennis teach reading, or does Dennis teach driving?
7.	Joe was seen around the airplane.
8.	Take a closer look at Eastman Kodak's bubbling reagents for photo-resist stripping.
9.	Each person at Beckman sees his responsibility aimed toward fabricating better resistors, displays, and drugs.

Table XII

Analysis of the Test Sentences

Sentence	Scoring Descriptor	Speaker 1 Discrete	Speaker 1 Cont.**	Speaker 2 Discrete	Speaker 2 Cont.	Speaker 3 Discrete	Speaker 3 Cont.	Combined Discrete	Combined Cont.
1.	Located	80.0%	60.0%	*	*	*	*	80.0%	60.0%
	Identified	60.0%	50.0%	*	*	*	*	60.0%	50.0%
2.	Located	63.6%	63.6%	81.0%	68.2%	72.7%	54.5%	72.3%	62.1%
	Identified	50.0%	54.5%	52.4%	54.5%	59.1%	45.5%	53.8%	51.5%
3.	Located	81.8%	50.0%	*	*	86.4%	72.7%	84.1%	61.4%
	Identified	63.6%	40.9%	*	*	63.6%	59.1%	63.6%	50.0%
4.	Located	75.0%	*	75.0%	67.9%	*	*	75.0%	67.9%
	Identified	67.9%	*	67.9%	57.1%	*	*	67.8%	57.1%
5.	Located	90.5%	81.0%	90.5%	85.7%	*	*	90.5%	83.3%
	Identified	76.2%	76.2%	81.0%	66.7%	*	*	78.6%	71.4%
6.	Located	*	*	80.0%	*	93.3%	86.6%	86.6%	86.6%
	Identified	*	*	60.0%	*	86.6%	26.6%	73.3%	26.6%
7.	Located	*	*	100.0%	*	*	*	100.0%	*
	Identified	*	*	75.0%	*	*	*	75.0%	*
8.	Located	70.8%	54.2%	79.2%	70.8%	70.8%	62.5%	72.2%	62.5%
	Identified	45.8%	25.0%	54.2%	50.0%	54.2%	33.3%	51.4%	36.1%
9.	Located	*	*	64.0%	64.0%	76.0%	56.0%	70.0%	60.0%
	Identified	*	*	52.0%	48.0%	68.0%	40.0%	60.0%	44.0%

(continued)

Table XII--continued

Analysis of the Test Sentences

Sentence	Scoring Descriptor	Speaker 1 Discrete	Speaker 1 Cont.	Speaker 2 Discrete	Speaker 2 Cont.	Speaker 3 Discrete	Speaker 3 Cont.	Combined Discrete	Combined Cont.
Total	Located	$\frac{97}{127}$=76.4%	$\frac{61}{99}$=61.6%	$\frac{112}{142}$=78.8%	$\frac{85}{120}$=70.8%	$\frac{85}{108}$=78.7%	$\frac{70}{108}$=64.8%	$\frac{294}{377}$=77.9%	$\frac{216}{327}$=66.1%
	Identified	$\frac{77}{127}$=60.6%	$\frac{48}{99}$=48.5%	$\frac{88}{142}$=61.9%	$\frac{66}{120}$=55%	$\frac{70}{108}$=64.8%	$\frac{45}{108}$=41.6%	$\frac{235}{377}$=62.3%	$\frac{159}{327}$=48.6%

	Discrete Located	Discrete Identified	Continuous Located	Continuous Identified
Expected Value of P_E	.221	.377	.339	.514
Variance of P_E	4.6×10^{-4}	6.2×10^{-4}	6.9×10^{-4}	7.6×10^{-4}
95% Confidence Interval Around P_E	0.17 to 0.26	0.33 to 0.42	0.29 to 0.40	0.46 to 0.56

*Not scored due to irreversible preprocessing error.

**Continuous (Cont.)

Figure 12. Decision Scheme Process

of identifying the phonemes was 0.486 and the corresponding 95 percent confidence error interval was 0.446 to 0.56.

scored, the "TH" sound was ignored since none of the eight

VIII. Hardware Modelling Analysis

The purpose of the hardware modelling analysis was to investigate the feasibility of implementing the speech recognition scheme with available semiconductor technology. Since the speech recognition scheme is computationally oriented, it was decided to base this hardware modelling study around a microprocessor. Once the preliminary design was completed, memory technologies and sizes were selected along with compatible peripheral support elements, and then an overall time-delay analysis was accomplished. It was from the time-delay analysis that the limitations of the hardware model were recognized. Finally, projections of future semiconductor technologies were appropriately applied to reconcile the limitations of the hardware model to give it near real-time capability.

Microprocessor Selection

The general requirement for a microprocessor to have writable control-store and multitasking software was recognized as a prerequisite for implementing a signal processing scheme such as the speech recognition routine developed in this thesis. These characteristics permit the scheduling of several programs in main memory at once for execution either simultaneously or at different times. A microprocessor possessing these characteristics and selected for this hardware design was the Texas Instruments TMS 9900. Several of distribution for the error rate.

the important features of this microprocessor are listed in Table XIII (Ref 27).

Microprogramming and Hardwire Multiply

The conventional division of functions between hardware and software in a computer system severely limits signal processing calculations to data rates of a few kilohertz in real time. But the speed of the complex computations associated with signal processing can be increased well into the megahertz range if microprogramming and hardwire multiplication are implemented. The advantage in signal processing time using microprogramming and hardwire multiply to calculate the Cooley-Tukey DFT is shown in Table XIV (Ref 17).

Speech Recognition Flow Chart

The first step taken to model the speech recognition scheme was to construct a flow chart linking the vital processing computations. The flow chart used for this hardware modelling analysis is shown in Figure 13.

All limitations in the programs and processing of the speech recognition scheme were included in the flow chart and subsequent hardware model. For example, all input speech data was low-pass filtered to 5 kHz and sampled at a 10 kHz rate in the analog-to-digital conversion process. These limitations influenced the selection of specific peripherals to complement the TMS 9900 microprocessor and in performing the all important time-delay analysis. The overall control of the sequence of operations depicted in the flow chart

Table XIII

Texas Instruments TMS 9900 Microprocessor

Technology	NMOS
Word Size	16-bit
Speed	3.33 mHz
Access Time	333 nsec
Clocks	4 phase/dynamic
Interrupts	16 levels
Directly Accessible Memory	65K words
I/O System	Separate/Serial/TTL compatible
Package	64-pin
Power	1 watt

(Ref 27)

$$VAR\{P_E\} = \frac{P_E(1-P_E)}{B} \quad \text{(Ref 21:158)}$$

was calculated from each of the word groups. For example,

Table XIV

Times for Calculating Different Versions of a 128-Point DFT

Version	Execution Time (μ sec)
Assembly Language	94
Microcode	35
Microcode plus Hardwire Multiply	11.3

(Ref 17)

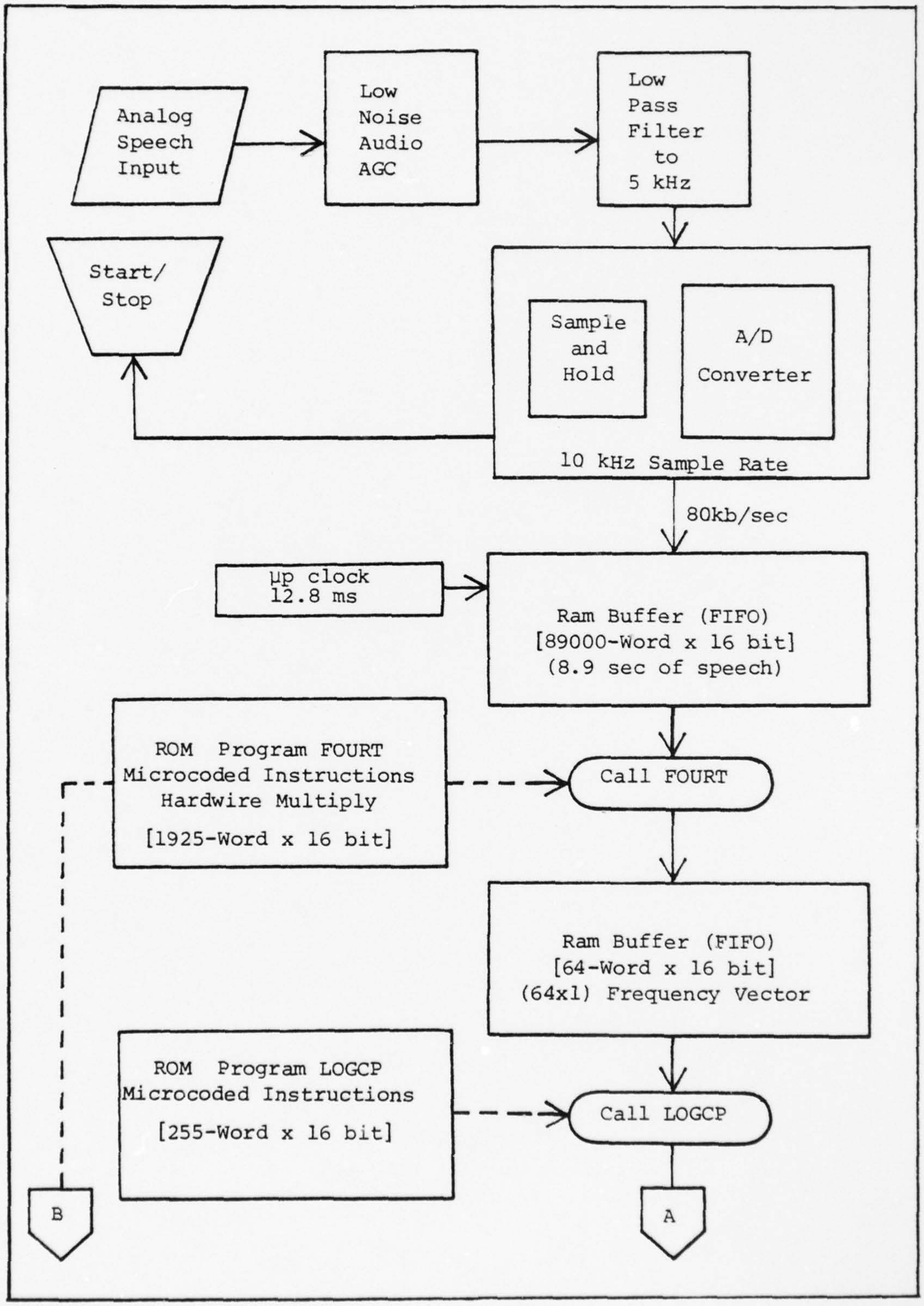


Figure 13. Speech Recognition Chart
(Plate 1)

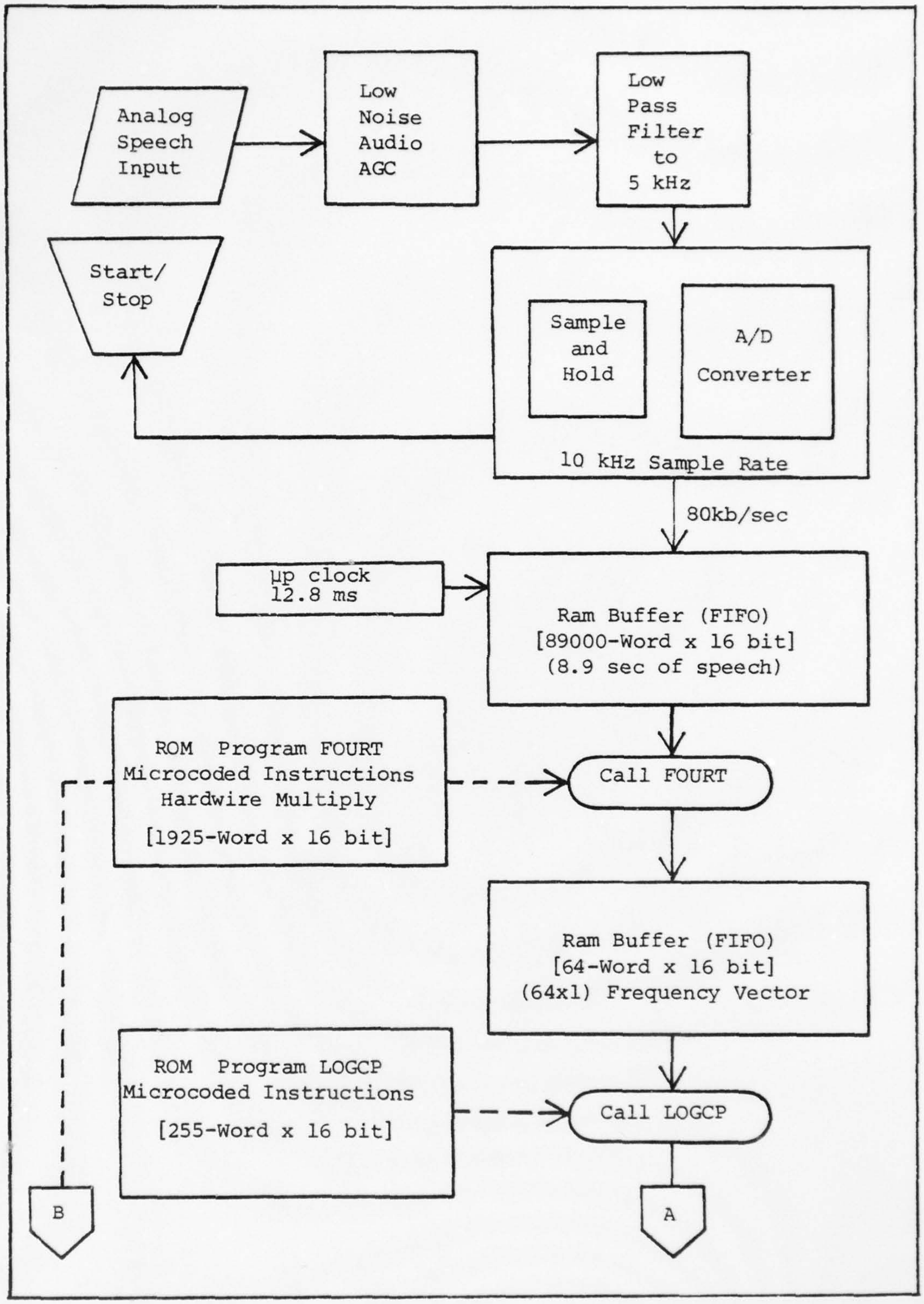


Figure 13. Speech Recognition Chart (Plate 1)

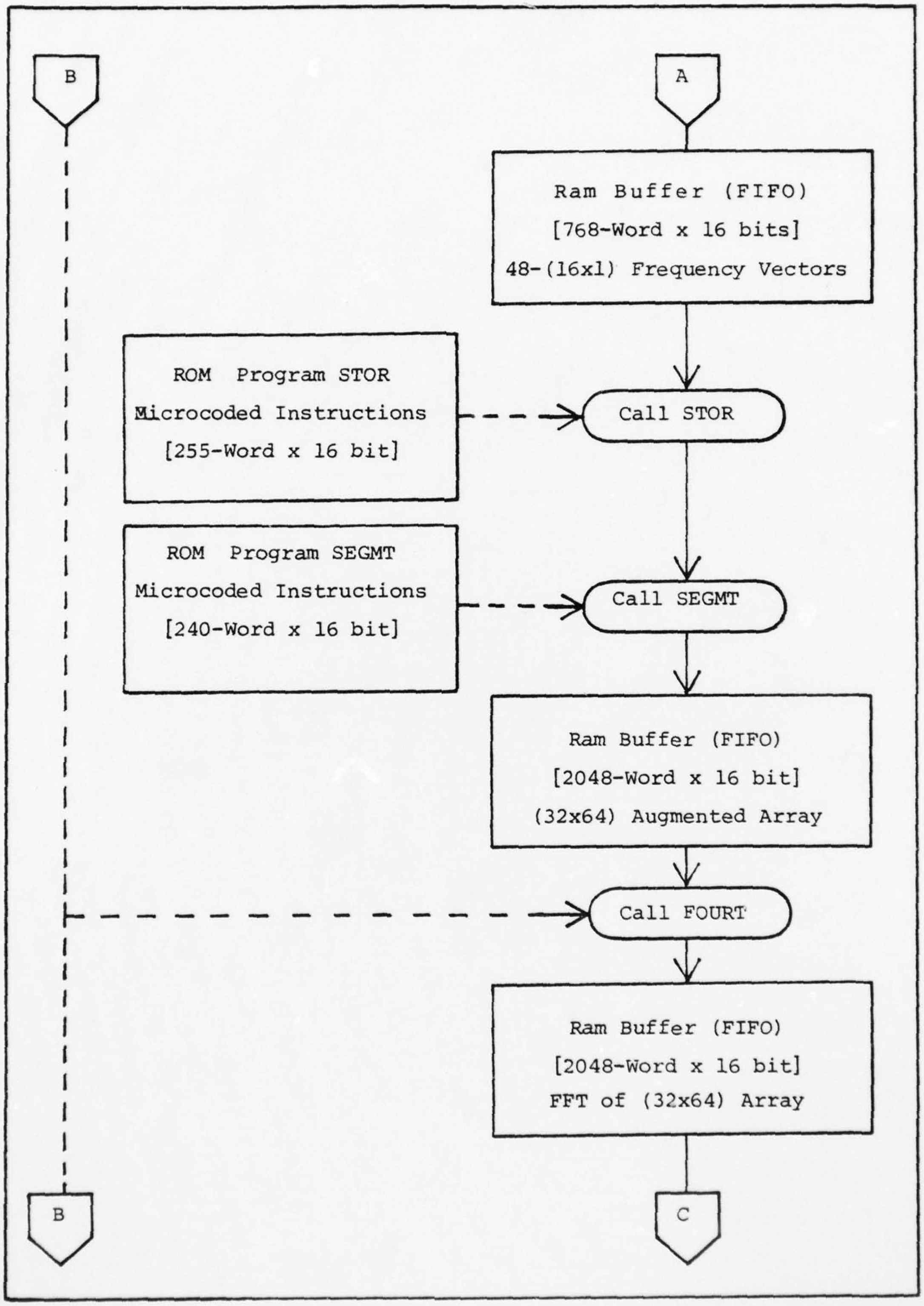


Figure 13. Speech Recognition Flow Chart
(Plate 2)

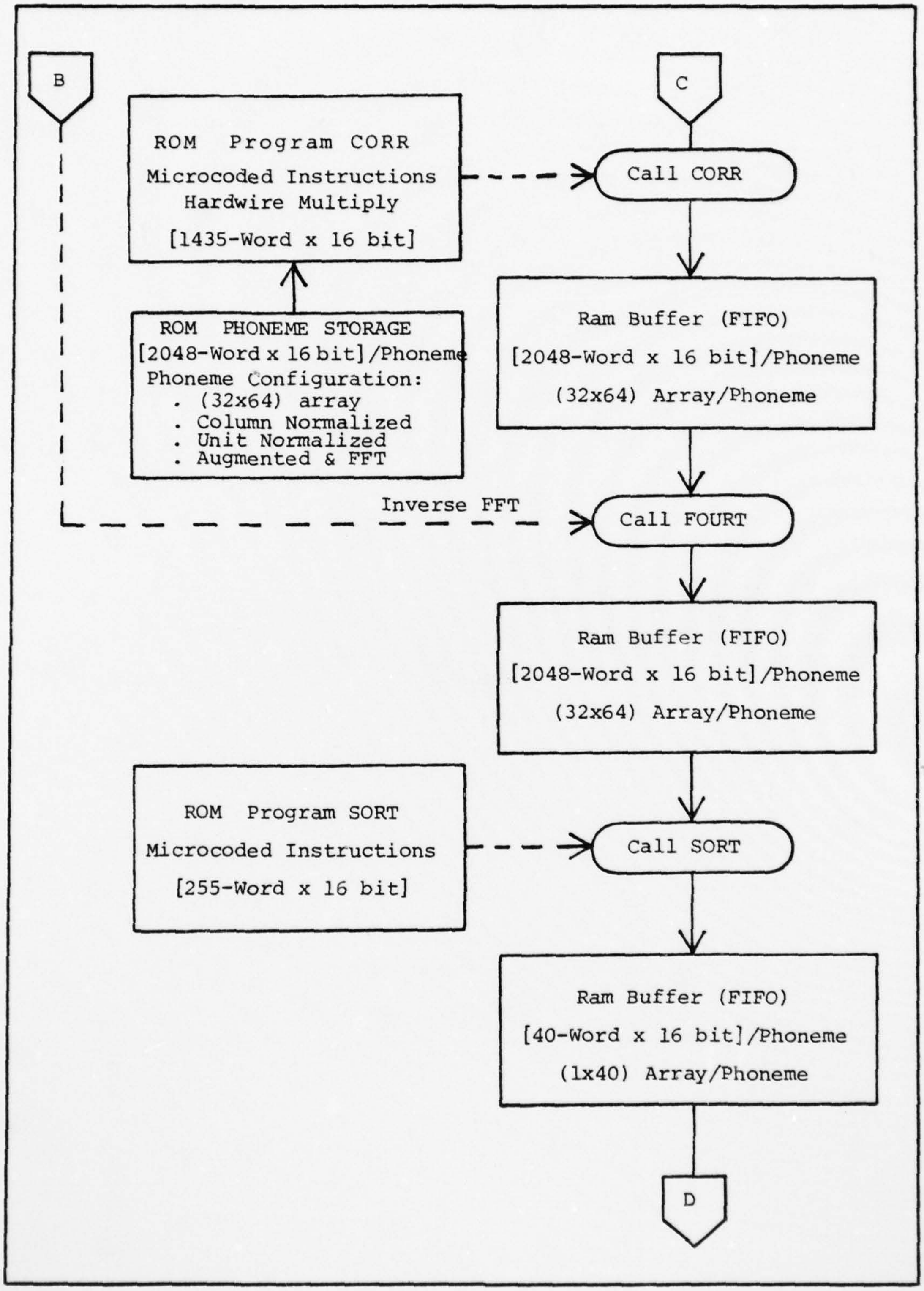


Figure 13. Speech Recognition Flow Chart (Plate 3)

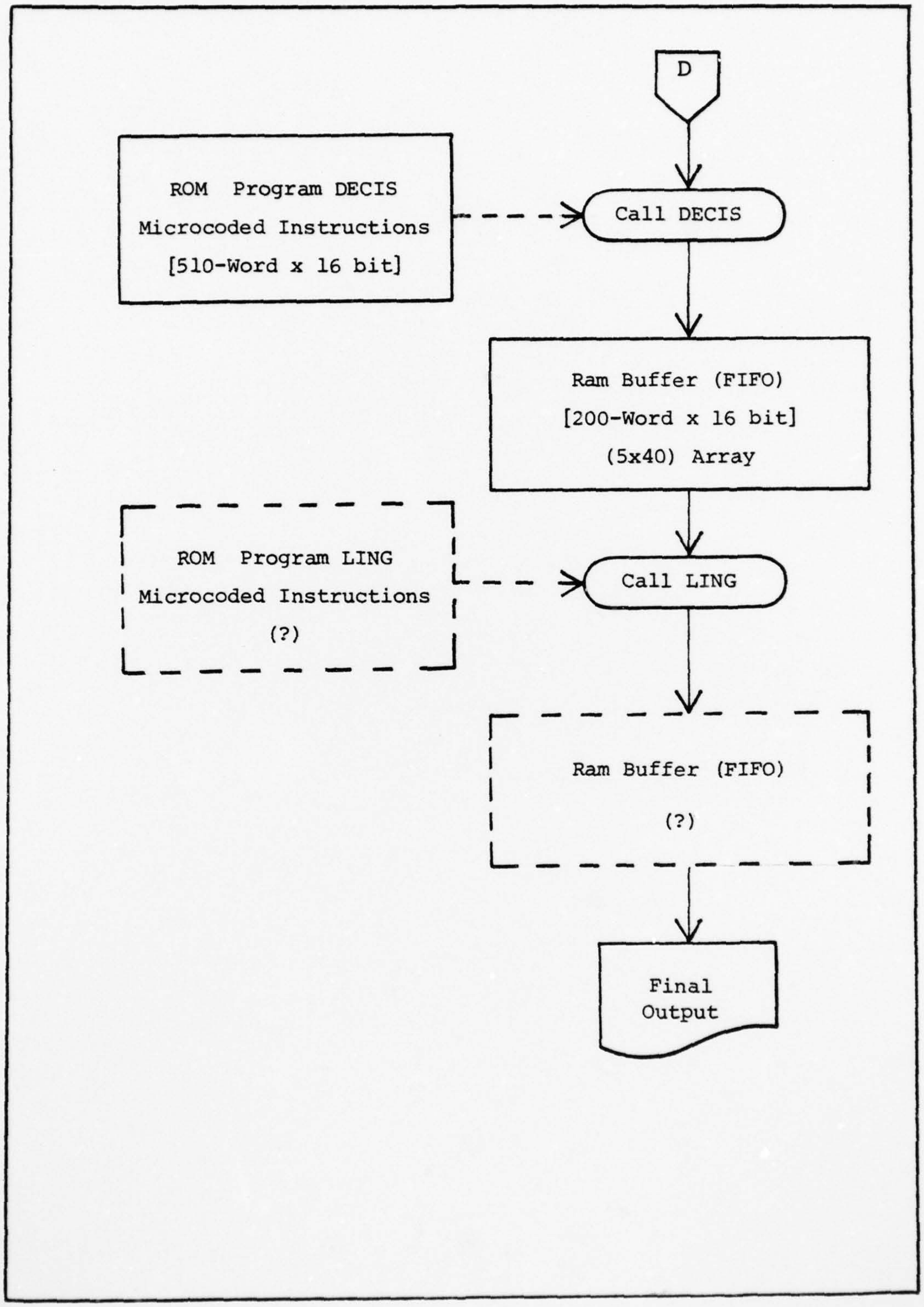


Figure 13. Speech Recognition Flow Chart (Plate 4)

would be accomplished by an executive program in the TMS 9900 microprocessor.

Hardware Implementation

The flow chart served as the basis for developing a hardware version of the speech recognition process. The system design is shown in Figure 14. The specific peripherals selected are listed in Table XV (Ref 3, 6, 26, 27, 29, 32). Each RAM buffer was sized according to the following division calculation:

$$\text{(Number of RAM chips needed)} = \frac{\text{(Number of 16-bit words to be stored)}}{\text{(Number of 16-bit words per chip)}}$$

Similarly, each ROM program buffer was sized according to the following multiplication and division calculations:

$$\begin{array}{rl} & \text{(Number of executable Fortran statements)} \\ \times & \underline{\text{(5 microcode statements per executable Fortran statement)}} \\ & \text{(Number of 16-bit microcode statements)} \end{array}$$

$$\text{(Number of ROM chips needed)} = \frac{\text{(Number of 16-bit microcode statements)}}{\text{(Number of 16-bit words per chip)}}$$

The execution time to either load or read a RAM buffer was calculated using the following formula (Ref 26):

$$T = t_C + (M)(t_A)$$

where

T = Total execution time

t_C = Microprocessor access time

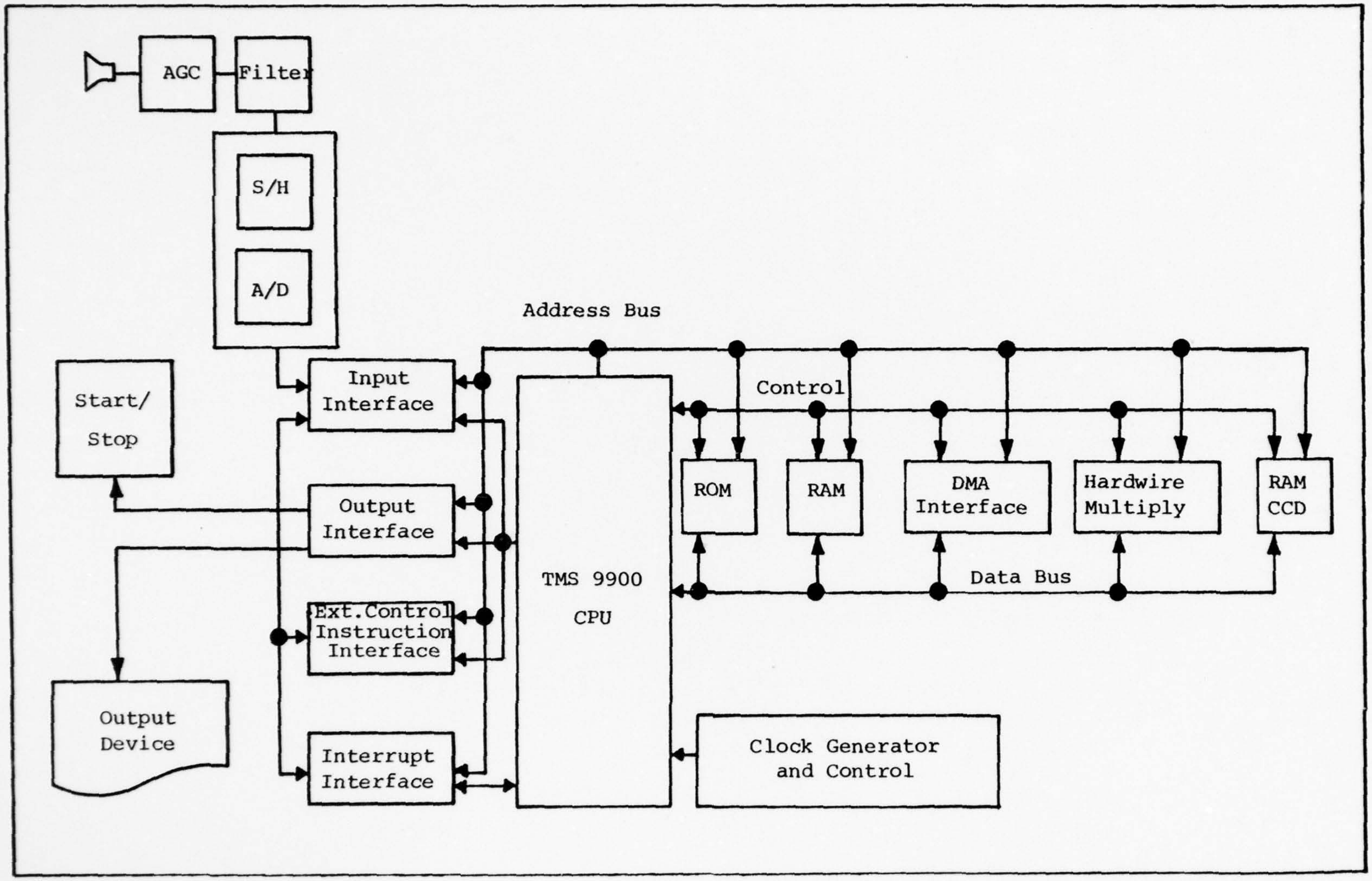


Figure 14. Speech Recognition System Design

Table XV

Peripherals Selected to Implement the Hardware Model

Peripheral	Specific Component
Microprocessor	Texas Instruments TMS 9900
Input Interface	Texas Instruments TIM 9905
Output Interface	Texas Instruments TIM 9906
Output Device	Texas Instruments Video Display Terminal Model 913A or Texas Instruments Line Printer Model 306
External Control Instruction Interface	Texas Instruments TIM 9914
Interrupt Interface	Texas Instruments TIM 9907
DMA Interface	Texas Instruments TIM 9911
Clock Generator and Control	Texas Instruments TIM 9904
Hardwire Multiply	TRW 16-bit Multiplier (TDC 1010J)
START/STOP	User Indicator Light Panel
Data Acquisition Network	Burr-Brown SHC85/ADC 60
Filter	Audio Low Pass (5kHz)
AGC	Audio Low Noise FET Design
Microphone	Audio (Directional)

(continued)

Table XV--continued

Peripherals Selected to Implement the Hardware Model

Peripheral	Specific Component
ROM	
Program FOURT	Texas Instruments SN74S371
Program LOGCP	Texas Instruments SN74S371
Program STOR	Texas Instruments SN74S371
Program SEGMT	Texas Instruments SN74S371
Program CORR	Texas Instruments SN74S371
Program SORT	Texas Instruments SN74S371
Program DECIS	Texas Instruments SN74S371
Program LING	(?)
Phoneme Storage	*Mostek MK 36000*
RAM	
Buffer for Digitized Speech Samples	Texas Instruments TMS 3064
Buffer for Results of FOURT	Texas Instruments SN74S200A
Buffer for Results of LOGCP	Texas Instruments SN74S200A
Buffer for Results of STOR and SEGMT	Fairchild 100470
Buffer for Results of FOURT	Fairchild 100470
Buffer for Results of CORR	Texas Instruments TMS 3064
Buffer for Results of FOURT	Texas Instruments TMS 3064
Buffer for Results of SORT	Texas Instruments SN74S214A
Buffer for Results of DECIS	Fairchild 10074
Buffer for Results of LING	(?)

(Ref 3, 6, 26, 27, 29, 32)

M = Number of memory accesses

t_A = Maximum memory access time

Similarly, the execution time for each ROM program was calculated by extrapolating the results documented for the microcoded 128-point DFT program (Ref 17). Specifically, the extrapolation was accomplished using the following division and multiplication calculations:

$$\frac{\text{(Time to execute microcoded DFT)}}{\text{(Number of executable microcoded instructions in the DFT)}} = \text{(Average execution time per microcoded statement)}$$

$$\begin{array}{ll} & \text{(Number of microcoded instructions per ROM Program)} \\ \times & \underline{\text{(Average execution time per microcoded statement)}} \\ & \text{(Total execution time per microcoded ROM program)} \end{array}$$

Finally, the time to execute the hardware multiplications was estimated from:

$$T_t = (T_{MA})(H)$$

where

T_t = Total time for hardware multiplications

T_{MA} = Time required for each multiplication and accumulation

H = Number of calculations

It is noted here that the memory sizes and time of execution calculations were not done for program LING or the RAM buffer that holds the results of LING. This particular program was not a part of this thesis, but would be necessary for a complete speech recognition system as it involves the higher-order syntactic and contextual logic necessary to construct words from phonemes.

The results for the sizes of the memories used in the hardware model are listed in Table XVI. Similarly, the results of the time-delay analysis are presented in Table XVII. From these results, the net time-delay to completely process an utterance 8.9 seconds long for one phoneme is 10.2 seconds. To process one-hundred phonemes, the time-delay is 71.1 seconds.

Limitations of the Hardware Model

The fundamental limitation of the hardware model is the long time-delay required to process a set of one-hundred phonemes (71.1 seconds). From Table XVII the operations that contribute most significantly to the long time-delay are numbers 3, 7, 8, and 9. The most significant factor that contributes to this long time-delay is the 100 μsec access time associated with the Texas Instruments CCD RAM buffer.

The most promising memory technology for the near future that could reduce the net delay time is a Fairchild 64K RAM with an access time of 25 μsec. Fairchild has developed a walled-emitter isoplanar fabrication process that, with 2 μm design rules, will permit a density of 64K memory cells on a 17300 square-mil die (Ref 7). For this hardware model, the application of this technology means that the CCD RAM buffers can be replaced chip-for-chip and result in a net time-delay

Table XVI

Memory Size Analysis

Memory Element	Number 16-bit Words	Number of Chips	Technology	Specific Component
ROM				
Program FOURT	1925	16	Schottky-TTL	TI SN74S371
Program LOGCP	255	2	Schottky-TTL	TI SN74S371
Program STOR	255	2	Schottky-TTL	TI SN74S371
Program SEGMT	240	2	Schottky-TTL	TI SN74S371
Program CORR	1435	12	Schottky-TTL	TI SN74S371
Program SORT	255	2	Schottky-TTL	TI SN74S371
Program DECIS	510	4	Schottky-TTL	TI SN74S371
Program LING	(?)	(?)	(?)	(?)
Phoneme Storage	2048/phoneme	0.5/phoneme	NMOS	Mostek MK 36000
RAM				
Buffer for Digitized Speech Samples	89000	24	CCD	TI TMS 3064
Buffer for Results of FOURT	64	4	Schottky-TTL	TI SN74S200A
Buffer for Results of LOGCP	768	48	Schottky-TTL	TI SN74S200A
Buffer for Results of STOR and SEGMT	2048	8	TTL	Fairchild 100470
Buffer for Results of FOURT	2048	8	TTL	Fairchild 100470
Buffer for Results of CORR	2048/phoneme	0.5/phoneme	CCD	TI TMS 3064
Buffer for Results of FOURT	2048/phoneme	0.5/phoneme	CCD	TI TMS 3064
Buffer for Results of SORT	40	1/phoneme	Schottky-TTL	TI SN74S214A
Buffer for Results of DECIS	200	1	TTL	Fairchild 10074
Buffer for Results of LING	(?)	(?)		(?)

Table XVII

Time-Delay Analysis

Operation	Description	Execution Time (sec)
1.	Load CCD RAM buffer with all digitized speech data	8.9 (maximum)
2.	Load the first group of 128 samples of digitized speech into the CCD RAM buffer	12.8×10^{-3}
3.	Read the first 48 groups of 128 samples from the CCD RAM buffer; call and execute FOURT: load the (64x1) frequency vectors into RAM	0.611
4.	Read the first 48 groups of (64x1) frequency vectors from the RAM buffer; call and execute LOGCP; load the (16x1) frequency vectors into RAM	4.1×10^{-4}
5.	Read the first 48 groups of (16x1) frequency vectors from the RAM buffer; call and execute STOR and SEGMT; load the (32x64) augmented array into RAM	1.39×10^{-4}
6.	Read the (32x64) augmented array from the RAM buffer; call and execute FOURT; load the (32x64) FFT array into RAM	1.55×10^{-4}
7.	Read the (32x64) FFT array from the RAM buffer; call and execute CORR; read each phoneme from ROM; perform hardware multiplications; load the (32x64) array/phoneme into RAM	0.205/phoneme
8.	Read each (32x64) array/phoneme from the RAM buffer; call and execute FOURT; load the inverse FFT (32x64) array/phoneme into RAM	0.41/phoneme
9.	Read each (32x64) array/phoneme from the RAM buffer; call and execute SORT; load results (1x40) array/phoneme into RAM	0.205/phoneme

(continued)

Table XVII--continued

Time-Delay Analysis

Operation	Description	Execution Time (sec)
10.	Read each (1x40) array/phoneme from the RAM buffer; call and execute DECIS; load results (5x40) array into RAM	2.8×10^{-5}

Summary of System Time-Delays

Time-delay to process sentence 8.9 seconds long for one phoneme = 10.2 seconds

Time-delay to process sentence 8.9 seconds long for 100 phonemes = 71.1 seconds for operations 3, 7, 8, and 9 of 3.66 seconds. Translating this advantage for the entire system means that the time-delay to process an entire 8.9 second utterance for one phoneme will be 9.24 seconds and for 100 phonemes, it will be 12.87 seconds. Additionally, new 16-bit microprocessors are being developed such as the Intel 8086 that has an operating speed of 8 mHz compared to 3 mHz for the Texas Instruments TMS 9900 (Ref 14). The application of this new and faster microprocessor would also contribute toward the objective of recognizing speech in near real-time. In summary, digital processing components either currently available or projected in the near future will support a near real-time realization of what has been to date exercised as an offline, non real-time speech recognition process.

AD-A064 058 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH.(U)
NOV 78 G L BROCK, E S KOLESAR
UNCLASSIFIED AFIT/GE/EE/78D-20 NL

2 OF 3
AD
A064058

OF 3

4058

IX. Conclusions

This research effort had three primary goals:

1. Develop a method to select and calculate universal prototype phonemes so as to improve the performance of the existing method of multiple-speaker and continuous speech recognition.

2. Modify the existing speech recognition programs so that their outputs can be readily analyzed and used as the input to a higher-order syntactic decision scheme.

3. Model the speech recognition scheme developed in this research with existing and/or projected solid-state technology to permit the speech recognition process to be done in near real-time.

The results obtained in this research indicate that these goals have been accomplished.

The results obtained for the word groups indicate a minimum 95 percent confidence level of 0.90 for the location of any phoneme and a 95 percent confidence level of 0.80 for the identification of any phoneme. These figures demonstrate that the phonemes could be located and identified satisfactorily for the words the phonemes were taken from. They also show that averaging the phonemes from the two speakers will drop the results from a perfect autocorrelation but yield high enough values to warrant averaging the phonemes to develop a universal phoneme. Since the scores were not perfect, this may indicate that averaging accentuates the slight differences in speech patterns between speakers, such as, dialect or regional speech characteristics. With this assumption a prototype set may have to be developed for different regions and dialects.

The verification sentences show a slight decrease in the scores compared to the word groups. For discrete speech, the location score was 91.1 percent. The identification score was 77.9 percent. Due to the way the previous researchers determined their location and identification scores, a direct comparison with their results was not possible. However, one set of data from the Guyote and Sisson thesis could be compared (Ref 11). They used two speakers and their phonemes were produced by averaging two phoneme samples to yield an averaged prototype phoneme. In contrast, the method used in this thesis averaged 28 sample phonemes from two speakers to yield a prototype phoneme. The Guyote and Sisson score for location was 97 percent and for identification it was 78.6 percent. Since they only scored two sentences for each speaker, a detailed comparison between the scores is not warranted. However, a very simple comparison of the raw scores shows them to be comparable.

The continuous speech shows a further drop in the scores with 88.2 percent for location and 66.1 percent for identification. Only one sentence was scored in the Guyote and Sisson thesis producing 83.3 percent for both location and identification (Ref 11). The results still show some consistency between the two scores with the present results having higher location but lower identification.

The test sentences show a further decrease in the scores obtained. The location of phonemes was 66.1 percent and the identification was 48.6 percent. There are no scores from the previous research efforts that can be used as a basis for comparison. The significant drop in the scores seem to indicate that the phonemes are not representative of those of other speakers.

The differences in the scores between the discrete and continuous speech may be reconciled since continuous speech merges phonemes to such an extent that the phonemes at the beginning of some words are smeared together and may be overshadowed by the phoneme at the end of another word. Also, the phonemes of some words may be missed entirely but the brain can still comprehend the word. For example, in Sentence 3 of the Test sentences, the words "The Rubber" can be analyzed. In the discrete sentence, the E sound in "The" is identified while the R in "Rubber" is located. But when the sentence is spoken continuously, the R is missed entirely and is engulfed by the E sound in "The". Even so, the missing R does not impede the brain from identifying "Rubber" from the context of the sentence.

However, with the present correlation scheme, if the phoneme was smeared, it is displayed as a miss. As a result, the scores show the drastic reduction as noted. Since the location scores are still greater than the identification scores, this indicates that the phonemes are still there but they are not necessarily the first choice in the decision program and some type of higher-order decision scheme must be used to select the correct phoneme. The use of a higher-order decision scheme to help identify the located phonemes was also noted by Neyman (Ref 19:71).

Another factor, that was observed in the sentences, was that the vowels seemed to overshadow the leading consonants. It is inferred from Fletcher that combinations of consonants and vowels significantly influence their frequency density functions (Ref 10:59). Since the consonant lengths were generally shorter than the vowel lengths, the consonant could be missed and the vowel could still be identified. This implies that the basic units of speech should not be phonemes but combined sounds such as the consonant-vowel sounds covering all possible combinations as presented in Fletcher (Ref 10:60-61).

The speech recognition programs were modified to permit a larger amount of data to be processed per execution. This was necessary because of the large number of calculations and tests that had to be done. These programs can now process any length sentence. Finally, each of the programs were fully documented to allow for easy interpretation and rapid modification.

The section on the modelling of the speech recognition scheme showed that with present technology, the processing of an 8.9 second segment of speech would take as long as 71.1 seconds to receive an initial response from the processor. It was shown that by changing the 64K CCD RAM's to a faster future technology, the delay-time could be reduced to about 13 seconds. Also, by using a proposed faster 16-bit microprocessor, the processing will approach real-time.

X. Recommendations

Two classes of recommendations for continuation of this research are listed below. Class I deals with methods for phoneme preparation, analysis, and correlation. Class II deals with other modifications which would give the user greater insight into the correlation performance.

Class I

1. Investigate the performance of the recognition scheme by using a larger phoneme set where the phonemes are consonant-vowel combinations. A recommended set would be the consonant-vowel combinations listed in Fletcher (Ref 10:60-61).

2. A larger population of speakers and sample phonemes should be used to calculate the averaged prototype phonemes. At least 10 speakers should be used. For each desired phoneme sound, each speaker should say at least 12 sample words. This should produce a more universal set of averaged prototype phonemes.

3. The speakers used to form the phoneme set should be from a common region and have similar dialects. Otherwise, the phonemes produced might accentuate the differences in the speech patterns and reduce the correlation values.

Class II

1. Have the Analog/Hybrid Systems Branch reconstruct analog speech from the digitized L-tapes. This should be

done after the initial digitization (64-channels) and after the logarithmic compression (16-channels). This will verify that the speech has not been significantly altered from normal speech and that noise has not been introduced into the speech.

2. Modify the correlation routine to accept 32-component frequency vectors. Compare the results of a 32-component frequency vector correlation with a 16-component frequency vector correlation. The added information contained in the 32-component frequency vector calculation may warrant a permanent modification to the correlation program.

BIBLIOGRAPHY

Bibliography

1. Beck B., et al. "An Assessment of the Technology of Automatic Speech Recognition for Military Applications," IEEE Transactions on Acoustics, Speech and Signal Processing, ASSP-25:310-322 (August 1977).

2. Bergland, G. D. "A Guided Tour of the Fast Fourier Transform," IEEE Spectrum 6:41-52 (July 1969).

3. Burr-Brown Research Corporation. Instrumentation for Data Acquisition and Control. Burr-Brown Research Corporation, 1977.

4. Daily, Keith G. and Franker, Sutton S. An Automatic Speech Recognition System Using a Vocoder Input. M.S. Thesis GE/GGC/EE/72-18. Wright-Patterson AFB, Ohio: Air Force Institute of Technology (1972).

5. Duda, Richard O. and Peter E. Hart. Pattern Classification and Scene Analysis. New York: John Wiley and Sons, Inc., 1973.

6. Fairchild Incorporated. "Fairchild's 4K Static RAMs are the Fastest Ever Made." Electronic Design 15: 124-25 (19 June 1978).

7. Fairchild Incorporated. "World's Fastest 4K Static RAM Speeds Up Cache Designs." Electronic Design 15: 17 (19 June 1978).

8. Fink, Daniel F. Intelligent Voice Data Entry System. Interstate Electronics Corporation, Anaheim, California, 1978.

9. Flanagan, James L. Speech Analysis Synthesis and Perception. New York: Academic Press, Inc., 1965.

10. Fletcher, Harvey. Speech and Hearing in Communication. New York: D. Van Nostrand Co., Inc., 1953.

11. Guyote, Michael F. and Sisson, Patrick L. Computer Identification of Phonemes in Continuous Speech. M.S. Thesis GE/EE/77D-18. Wright-Patterson AFB, Ohio: Air Force Institute of Technology (1977).

12. Haller, Mark. The Cooley-Tukey Fast Fourier Transform in USASI Basic Fortran. A Computer Program for the CDC 6600. Wright-Patterson Air Force Base, Ohio: ASD Computer Center, 1972.

13. Hensley, William R. Computer Identification of Phonemes in Continuous Speech. M.S. Thesis GE/EE/76-24. Wright-Patterson AFB, Ohio: Air Force Institute of Technology (1976).

14. Intel Corporation. "Microcomputer Prototyping Kit Helps Design System with the 8086." Electronic Design 15:126 (19 June 1978).

15. Kabrisky, Matthew. A Proposed Model for Visual Information Processing in the Human Brain. Uroan, Illinois: University of Illinois Press, 1966.

16. Laefoged, Peter. Elements of Acoustic Phonetics. Chicago, Illinois: The University of Chicago Press, 1962.

17. Mulrooney, Timothy. "Microprogramming a Minicomputer for Fast Signal Processing," Electronics 6:136-141 (16 March 1978).

18. NaKagawa, Sei-Ichi. A Machine Understanding System for Spoken Japanese Sentences. Department of Information Science, University, Japan, October 1976.

19. Neyman, Ralph W. Computer Identification of Phonemes in Continuous Speech. M.S. Thesis GE/EE/76-10. Wright-Patterson AFB, Ohio: Air Force Institute of Technology (1976).

20. Potter, Ralph K., et al. Visible Speech. D. Van Nostrand Co., Inc., 1947.

21. Papoulis, Athanasios. Probability, Random Variables, and Stochastic Processes. New York: McGraw-Hill Book Co., 1965.

22. Reddy, D. Raj. "Speech Recognition by Machine: A Review," Proceedings of the IEEE, 64:501-531 (April 1976).

23. Spitznoghe, Frank. Texas Instruments 990 Computer Systems Handbook. Texas Instruments Incorporated, 1975.

24. "Talking to Your Wheelchair," Science News, 111(22):346 (May 1977).

25. Texas Instruments Incorporated. Bipolar Microcomputer Components Data Book. Texas Instruments Incorporated, 1977.

26. Texas Instruments Incorporated. Memory System Design Utilizing 4K Dynamic RAMs. Texas Instruments Incorporated, 1976.

27. Texas Instruments Incorporated. TMS 9900 System Development Manual. Texas Instruments Incorporated, 1976.

28. Toombs, Dean. "An Update: CCD and Bubble Memories," IEEE Spectrum 4:22-30 (April 1978).

29. TRW Incorporated. "Multiply and Accumulate in 70 μsec," Electronics 15:71 (20 June 1978).

30. Turn, R., et al. Military Applications of Speech Understanding Systems. Rand Report, R-1434-ARPA, June 1974.

31. White, George M. "Speech Recognition a Tutorial Overview," Computer 9:40-53 (May 1976).

32. Wilson, Dennis R. "Cell Layout Boosts Speed of Low-Power 64K ROM," Electronics 7:96-99 (30 March 1978).

APPENDIX A

DATA PROCESSING CHARTS AND NOTES

A. Data Processing Charts and Notes

This appendix contains the flow charts of the seven programs used in the speech recognition process. These flow charts outline the operation of each program.

Also included are notes which clarify the important operating points of each program. Table XVIII lists the seven programs along with their associated inputs and outputs.

Table XVIII

Data Processing Programs

Name	Input	Output
EK1 (OCTAVE1)	L-tape #1	L-tape #2 Spectrogram
EK2 (OCTAVE2)	L-tape #2	Normalized Spectrogram
EK3 (PUNCH)	L-tape #2	Punched Cards #1 (Target Phonemes)
EK4 (PROAVE)	Punched Cards #1	Punched Cards #2 Prototype Phonemes)
EK5 (CRSCOR)	L-tape #2 Punched Cards #2	PF (Correlation arrays) Calcomp Graphs
EK6 (FPLOT)	PF	Calcomp Graphs
EK7 (DECIS)	PF	Phonemic Output

Note: PF refers to Permanent Files on the CDC system.

EK1 (OCTAVE1)

EK1 (OCTAVE1) processes the 64-component vectors on L-tape #1 as input and assigns it as a local file name called Tape 1. L-tape #1 is the digitized data produced by the ASD Computer Center. Each line of data on L-tape #1 consists of two leading numbers (NCHAN and NDIM) followed by the 64 components of each frequency vector. NCHAN and NDIM are produced by the subroutine used to calculate the DFT. EK1 reads each line of data but only processes the 64 components.

The program logarithmically compresses the data from 64 components per frequency vector to 16 components per frequency vector. This process causes the higher frequencies to be emphasized. The results of this compression are stored in a local file called Tape 2 and written on L-tape #2 for use in subsequent programs. EK1 also produces a non-normalized spectrogram of the speech data.

The two variables which are important in EK1 are NREC and NN2. NREC represents the number of files to be read. A file is defined to be the digitized speech between the 2 kHz tones on the recording tape. NN2 is set to be one more than the number of records contained in the largest file. The number of files and the number of records contained in each file is available on the printout received from the ASD Computer Center.

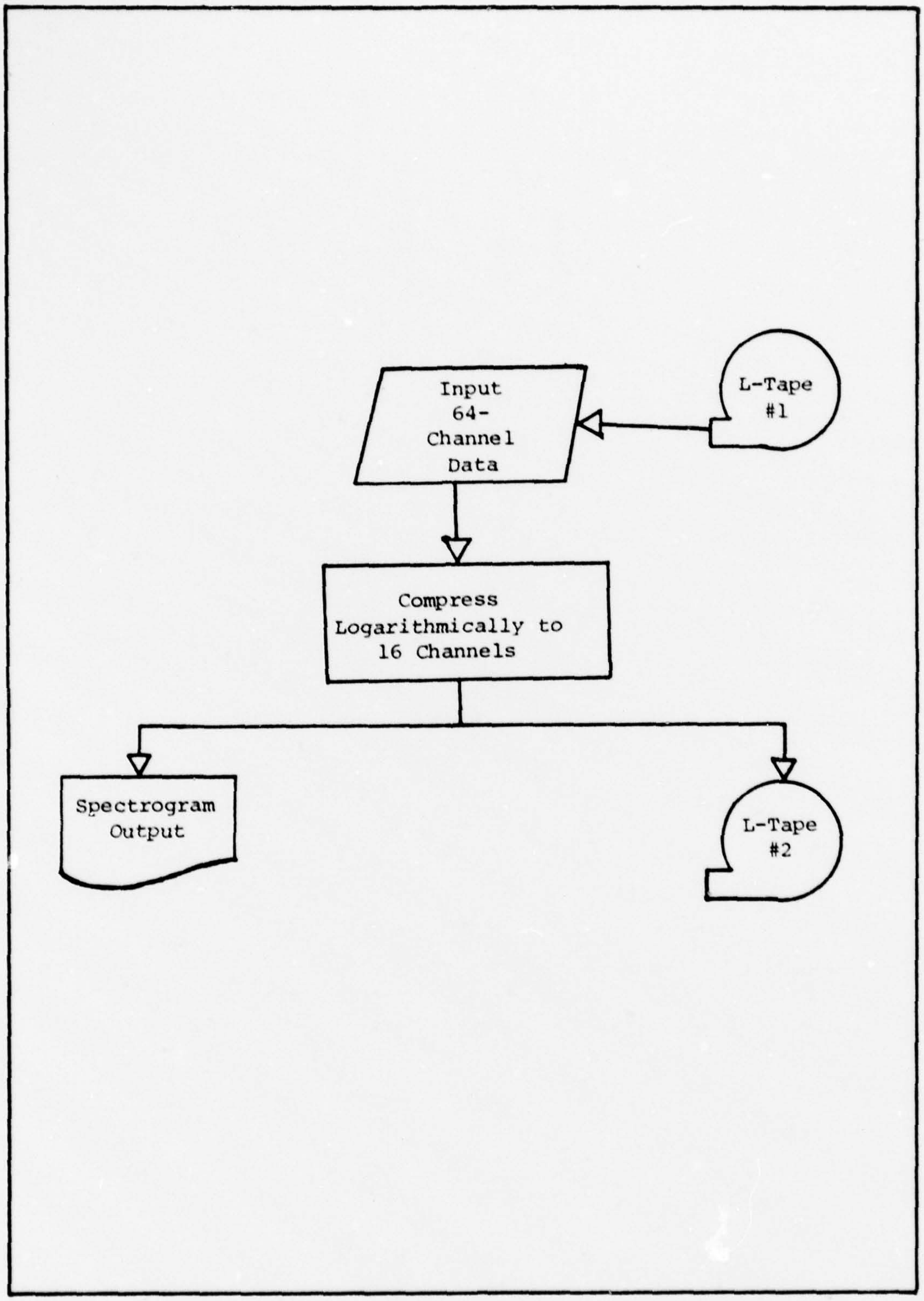


Figure 15. EK1 (OCTAVE 1) Flow Diagram

EK2 (OCTAVE2)

The input to EK2 (OCTAVE2) is the compressed data on L-tape #2 created by EK1. EK2 only produces a normalized spectrogram of the speech data. This presents the speech data in a more easily interpreted form than EK1. Although it is necessary to read the entire sentence record to produce the spectrogram, the two variables NSTART and NSTOP allow the user to select portions of the speech file of interest. The entire file will be read and stored on a local file called Tape 1. However, only the desired portions of the speech data (between NSTART and NSTOP) will be output in the normalized spectrogram. The total number of speech files to be read is set by NREC.

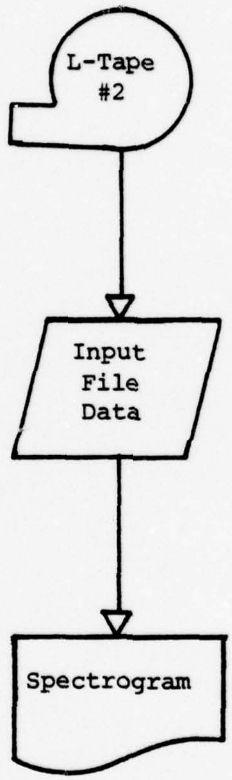


Figure 16. EK2 (OCTAVE 2) Flow Diagram

EK3 (PUNCH)

EK3 (PUNCH) uses the data on L-tape #2 as input. EK3 is used to select the phonemes from the word groups and it produces a set of punched cards for each target phoneme (punched cards #1).

During the phoneme selection process the locations of the target phonemes were noted by recording the values of the time increments given on the normalized spectrogram. The length of a target phoneme was the same within a word group.

The program can only be used to process one word group from one speaker at a time. The beginning values of the target phonemes were put into the IBGN data statement. The end values of the target phonemes were put into the IEND data statement. The program reads in the entire word group and selects the target phonemes according to the data statements.

The program generates 14 sets of punched cards for each word group. A printout of these values is also produced. The 16 components for each frequency vector are contained on two punched cards.

This process must be repeated for each speaker's group of words. This results in 28 sets of target phonemes for each word group, assuming there are two speakers.

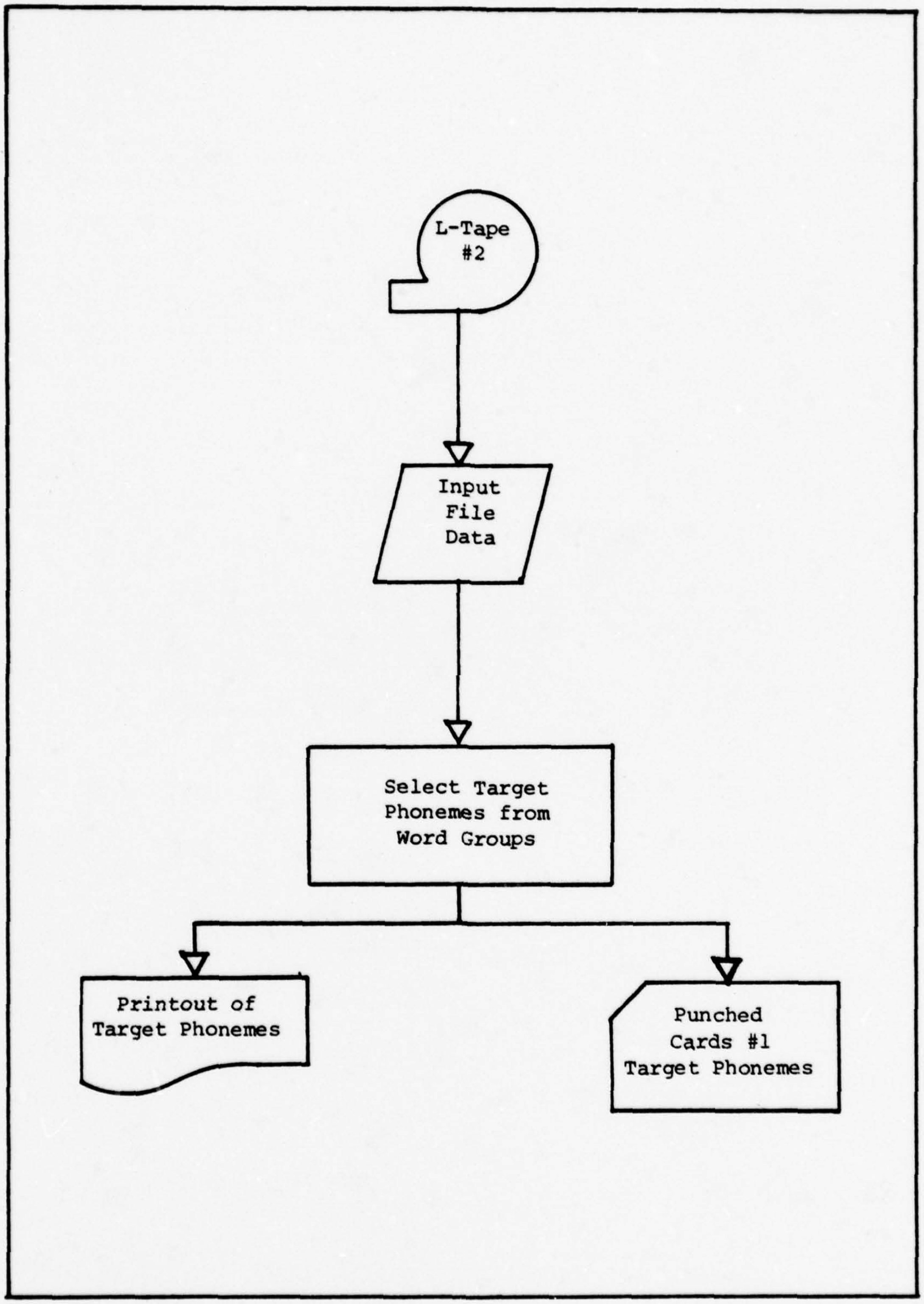


Figure 17. EK3 (PUNCH) Flow Diagram

EK4 (PROAVE)

EK4 (PROAVE) uses the punched cards of EK3 as input and averages the 28 target phonemes of a particular word group to yield an averaged prototype phoneme. EK4 then produces a set of punched cards for the averaged prototype phoneme (punched cards #2).

The program does the averaging by summing all 28 values of a specific frequency vector component and then dividing by 28 to give an average value for the component. When all 16 components of a frequency vector have been averaged the program produces a punched card output of the vector.

The variables that control this process are JE, JI, DIV, KARD. KARD is the number of lines of input data (half the number of cards). DIV is the number of target phonemes that are to be averaged. JI is the length of the target phonemes. JE is (1 + KARD - JI), which gives the first line of data of the last target phoneme.

This program must be run for each prototype. The end product of this process is a set of eight averaged prototype phonemes.

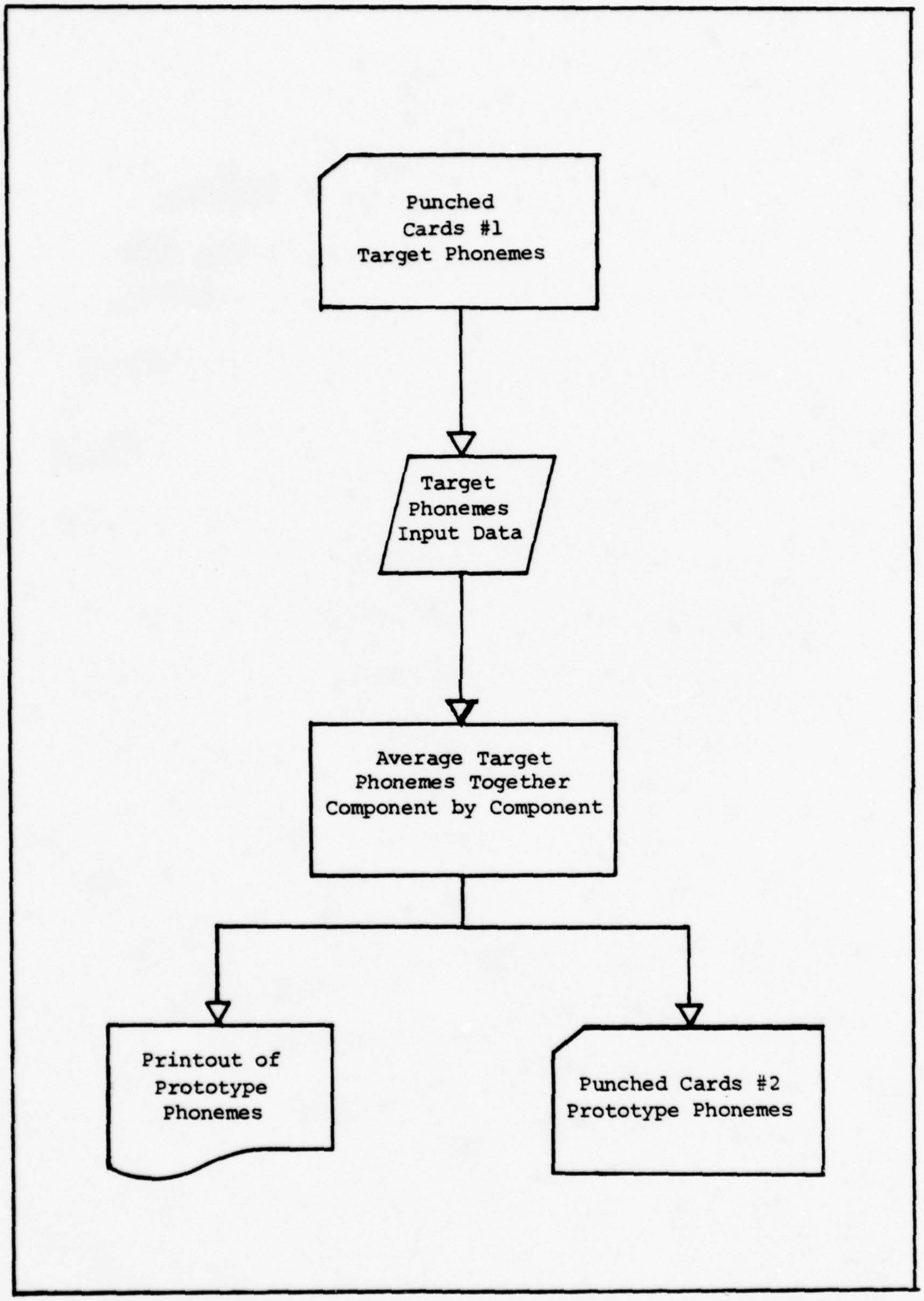


Figure 18. EK4 (PROAVE) Flow Diagram

EK5 (CRSCOR)

This program comprises the main body of the research. EK5 (CRSCOR) consists of a main program (CRSCOR) which inputs selected variables that are established using the comments in the program. Following the initialization of the desired variables, the main program calls a subroutine (XCORR) which handles the correlation computations.

The sentence data is attached on Tape 3 and is read from L-tape #2. The SKIPF control card is used to skip down to the file on L-tape #2 containing the desired speech data. The value put into the SKIPF card is one less than the number of the file desired. The prototype data is attached as Tape 1 and is read from punched cards (Punched Cards #2) produced by EK4. Tape 2 is a local file used to store the prototypes for each speech segment so they only need to be read in once for each block of speech.

The main program is organized to accept data in blocks of 700 frequency vectors. If the data is longer than this, the utterance must be segmented and the value of IRUN adjusted accordingly. The block of comments at the beginning of the program listing (Appendix B) gives the statement numbers of the variables that must be changed for each run.

Subroutine XCORR calculates the correlation values of each prototype with the sentence and produces an array containing this information. The printout includes a listing of the sentence data, prototype values, and correlation computations. The program also prints information concerning the subdivisions of the sentence, number of zeros required to augment the data arrays, and prototype lengths.

The subroutine XCORR calls the plot routine which produces a correlation versus time graph for each prototype. Since the correlation values for up to nine prototypes can be plotted, the DISPOSE command was used to load several buffers and permit the Calcomp plotter to output the results in groups of nine graphs.

Finally, the subroutine XCORR writes the correlation values into a permanent file called Tape 4. An end-of-file is then placed after the last correlation value.

Each phase of the correlation processing is well documented to make it easy to follow. This fact allows rapid changes or revisions to be made to the program.

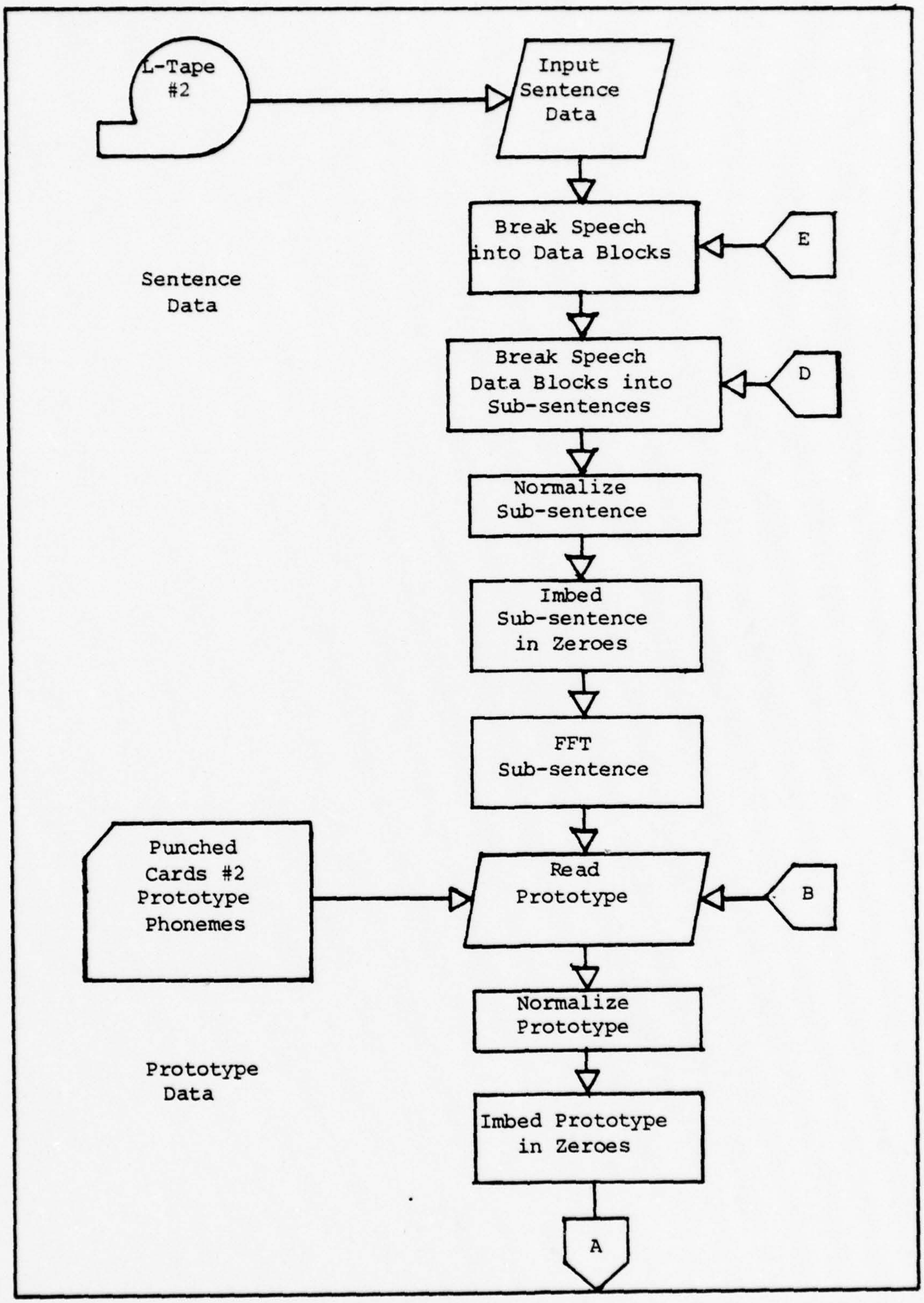


Figure 19. EK5 (CRSCOR) Flow Diagram
(Plate 1)

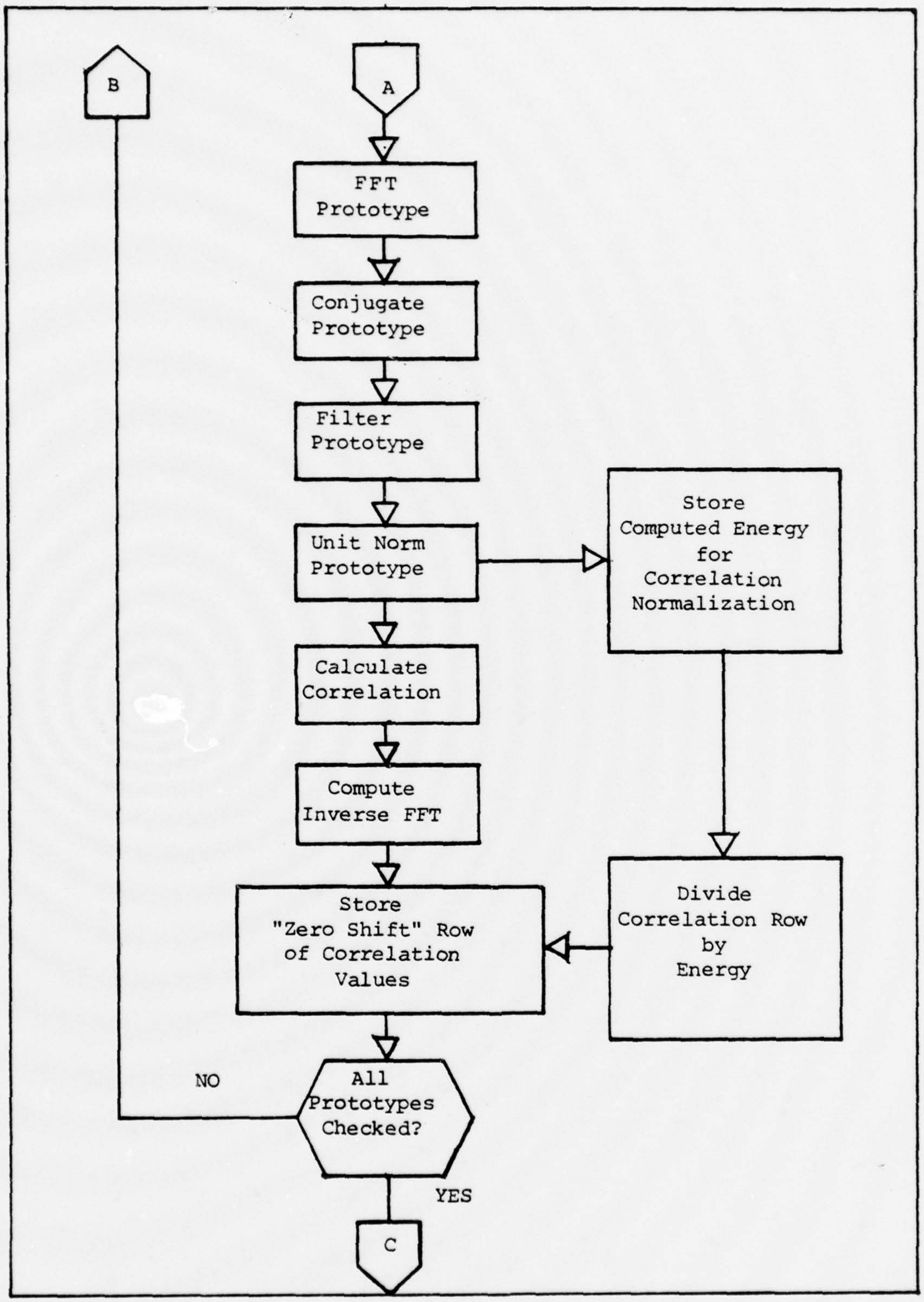


Figure 19. EK5 (CRSCOR) Flow Diagram (Plate 2)

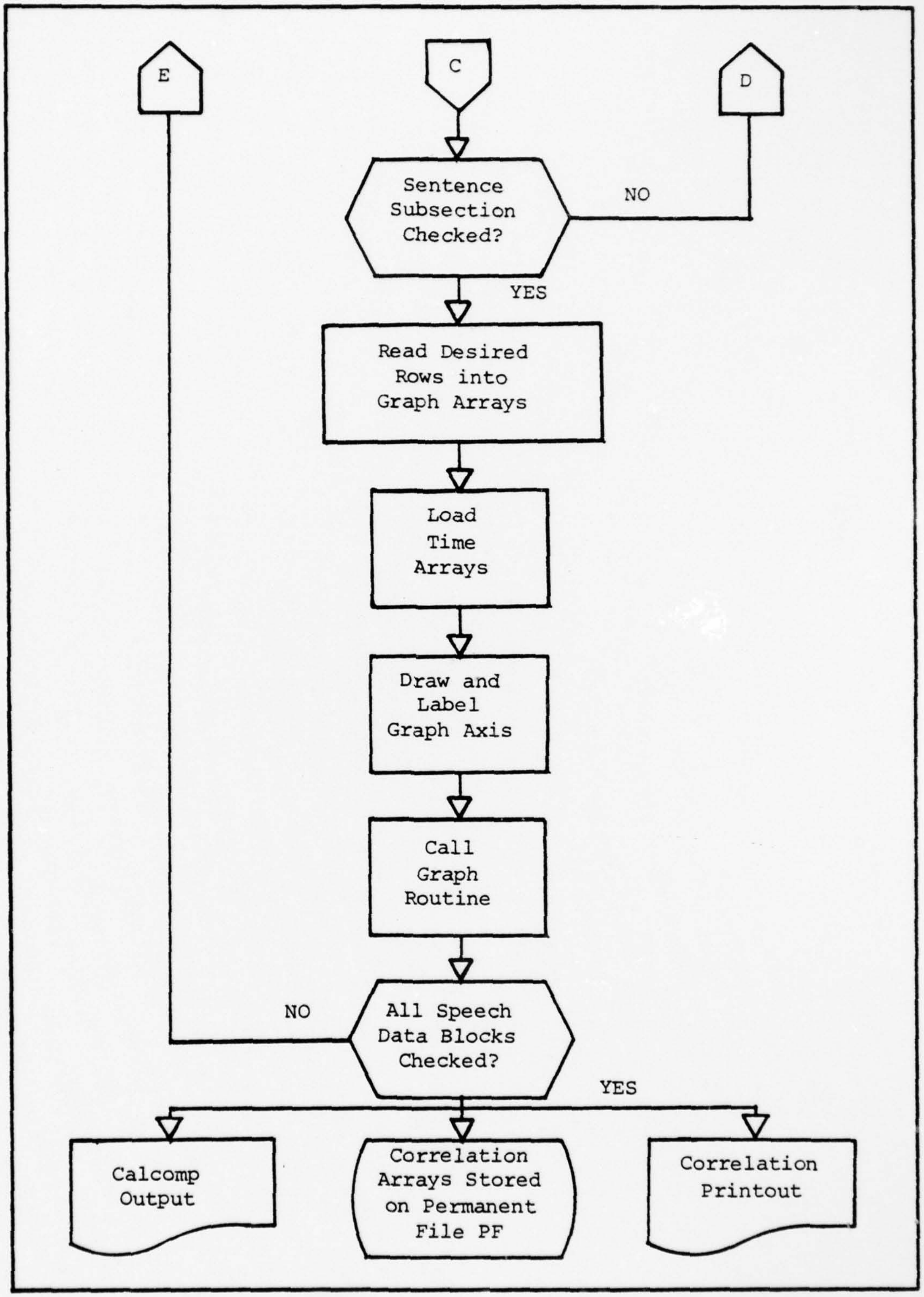


Figure 19. EK5 (CRSCOR) Flow Diagram (Plate 3)

EK6 (FPLOT)

EK6 (FPLOT) attaches the PF generated in CRSCOR as local file Tape 1. It reads selected portions of the correlation arrays into an array called SAMPLE. These values are then sent to special graphing routines which have been attached to the program through the control cards. Following the graph calls, the resulting data is sent to the Calcomp Plotter through the use of the CALL PLOTE (N) instruction.

The output can be adjusted to provide the same plot as was produced with the correlation routine. However, this plot routine is more versatile and can be used to plot any of the prototypes at any point in the correlation array by varying the IBGN1 and IEND1 data statements and NPRO. The labels of the axis on the graphs must be changed according to the prototypes being plotted. The other variables that must be changed for each run are listed in the comments at the beginning of the program (Appendix B).

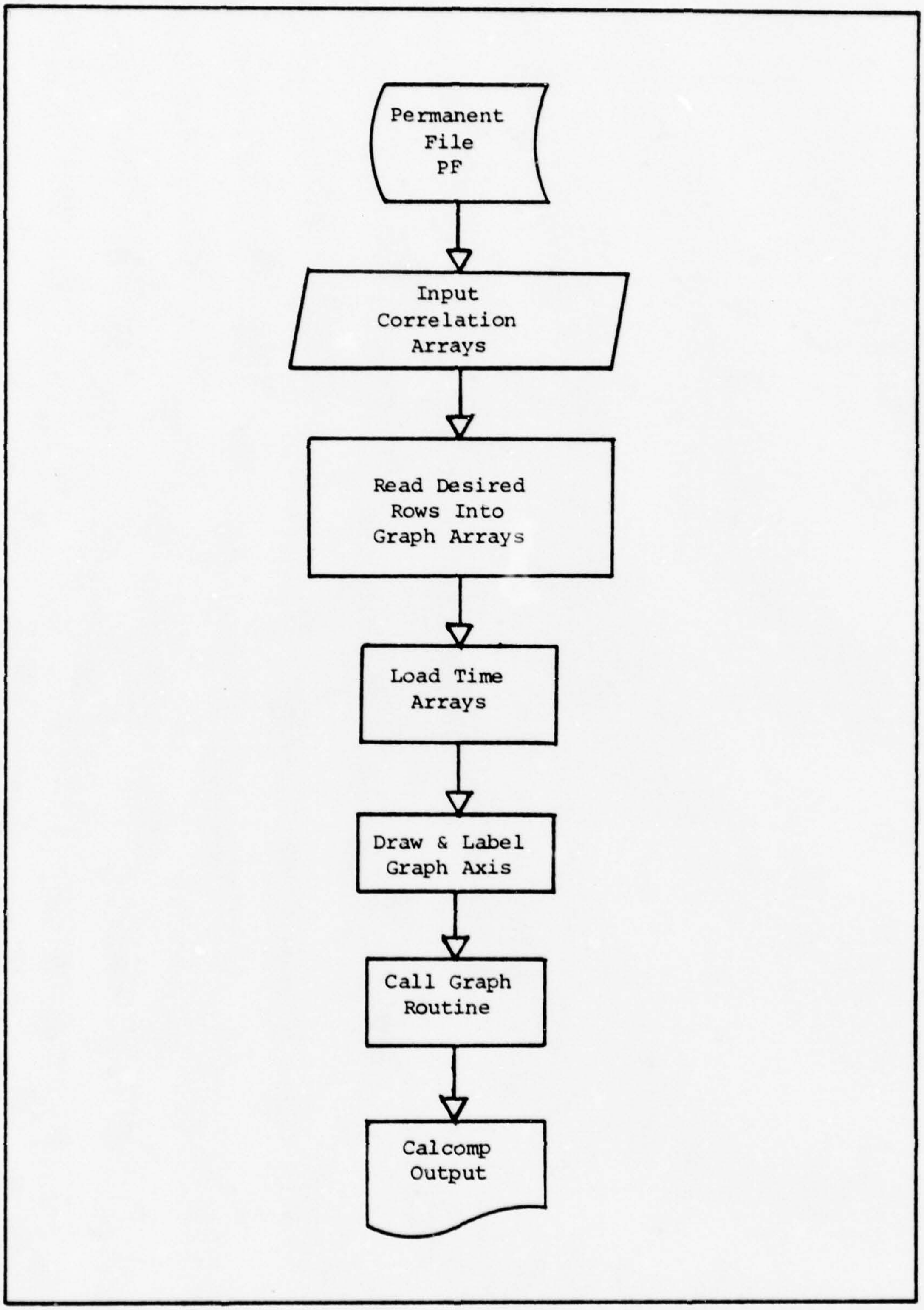


Figure 20. EK6 (FPLOT) Flow Diagram

EK7 (DECIS)

EK7 (DECIS) attaches the PF generated in CRSCOR as local file Tape 1 and processes the correlation arrays according to the methods described in Section VI. The input arrays which contain information concerning the phonemes and their lengths must be adjusted for each set of phonemes. The variables ENDUR, DELTA and THRHLD are the endurance (time), rate-of-change criterion, and correlation threshold values, respectively. This program generates a list of the phonemes identified in a sentence. The list gives the eight phonemes with the highest correlation values versus time. It is possible to store these results in permanent files in order to preserve these processed arrays for a higher-order decision scheme.

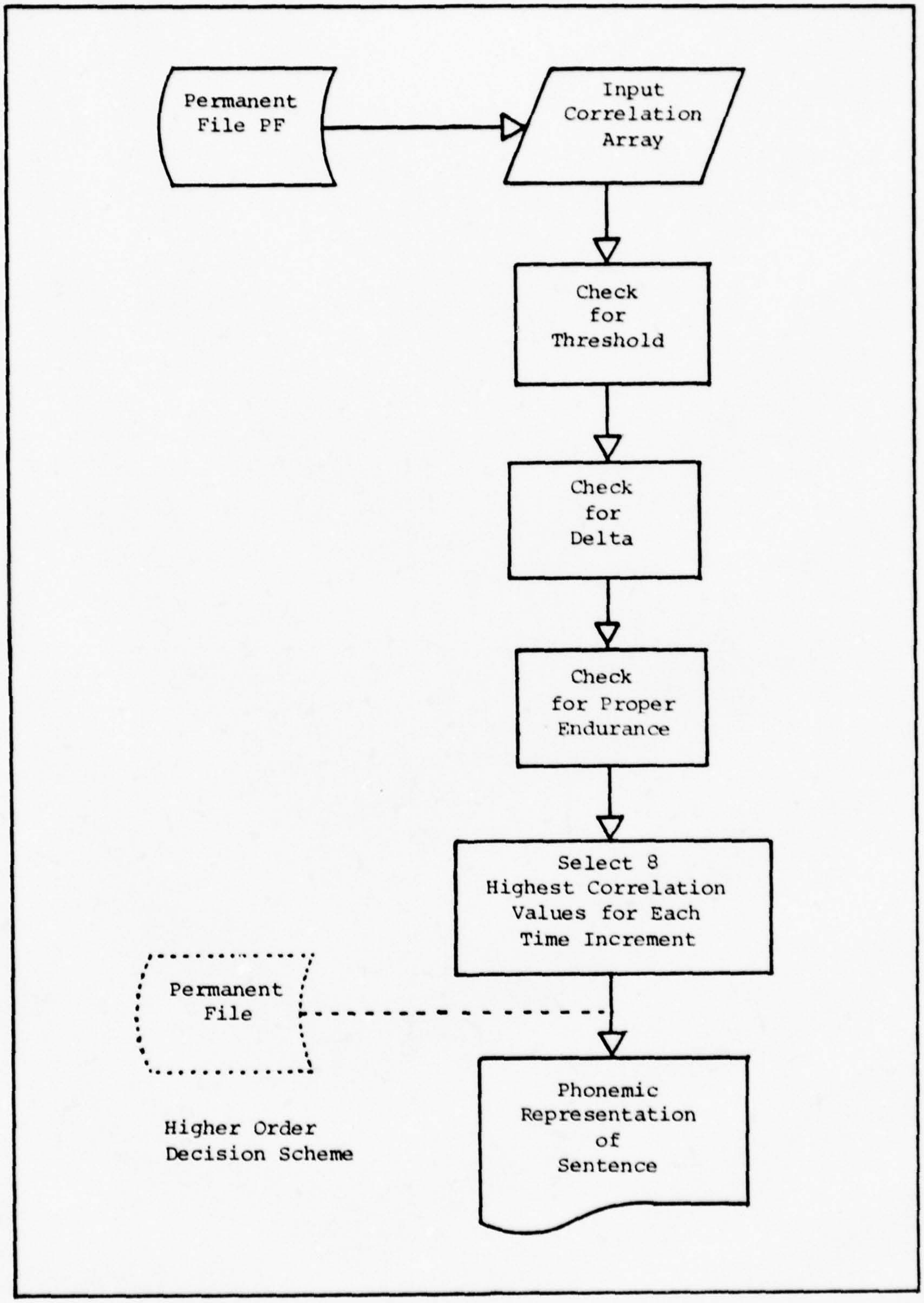


Figure 21. EK7 (DECIS) Flow Diagram

APPENDIX B

COMPUTER PROGRAM LISTINGS

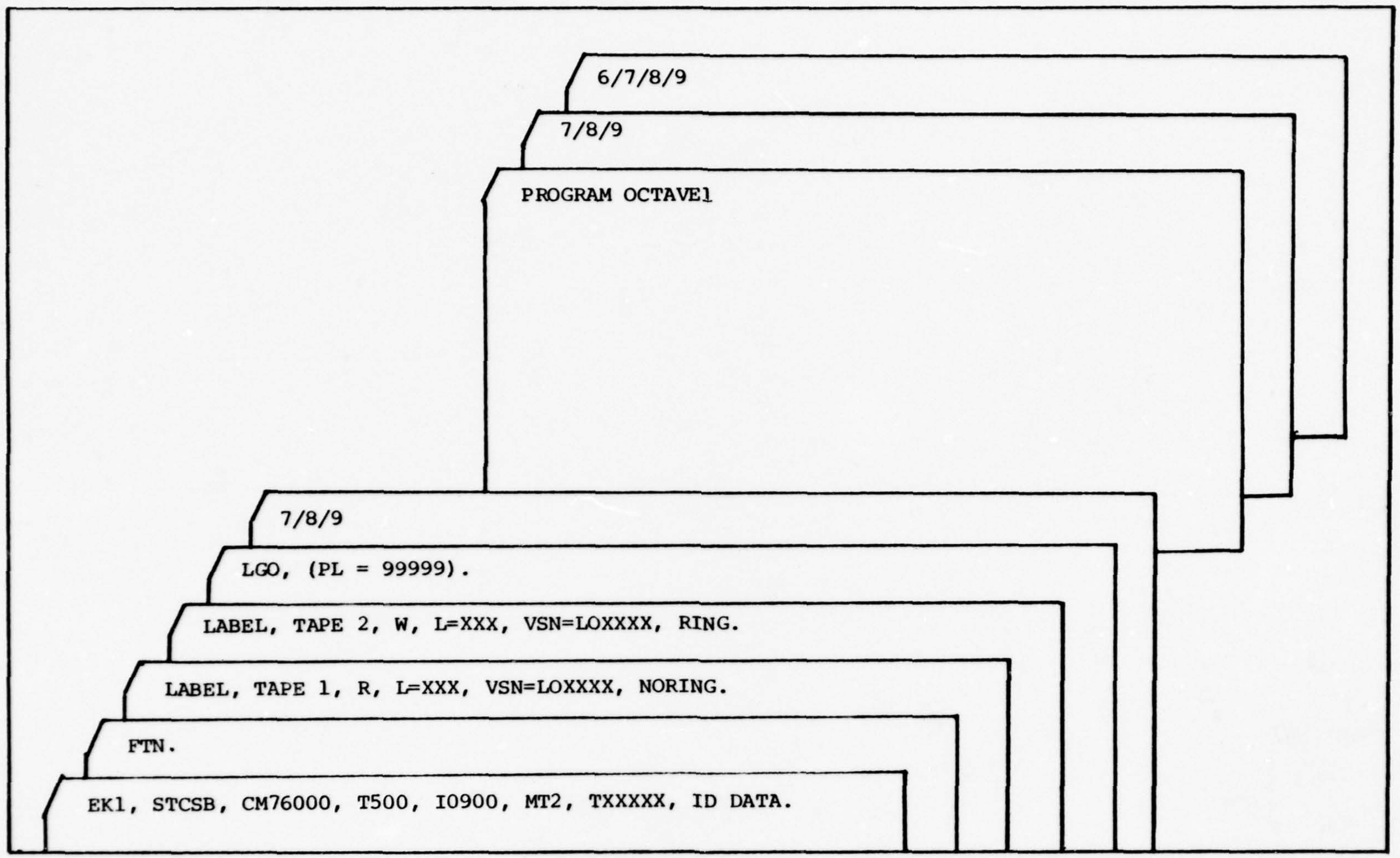


Figure 22. Program OCTAVE1

```
C***************************************************************************
C***************************************************************************
C**    THIS PROGRAM REDUCES 64-CHANNELS OF DIGITIZED SPEECH DATA TO 16      **
C**    CHANNELS. THE ORIGINAL 64 CHANNELS ARE COMBINED BY ADDING THE ENERGY **
C**    CONTRIBUTIONS OF EACH ELEMENT WITHIN A 1/3 OCTAVE GROUP. THE OUTPUT  **
C**    IS THE COMPRESSED ARRAY AND AN ACCOMPANYING SPEECH SPECTROGRAM.      **
C**    THE RESULTS ARE STORED ON MAGNETIC TAPE FOR FUTURE                   **
C**    USE BY LATER STAGES OF THE RECOGNITION PROGRAM                       **
C***************************************************************************
C***************************************************************************
C---------------------------------------------------------------------------
C---------------------------------------------------------------------------
C--                                                                         --
C--     THE FOLLOWING FORTRAN STATEMENTS MUST BE ADJUSTED FOR               --
C--     EACH RUN                                                            --
C--                                                                         --
C--       PROGRAM OCTAVE1:       36        38                               --
C--                                                                         --
C---------------------------------------------------------------------------
C---------------------------------------------------------------------------
      PROGRAM OCTAVE1(INPUT,OUTPUT,TAPE1,TAPE2,TAPE6=OUTPUT)
      DIMENSION SYMBOL2(10),SYMBOL3(10),SYMBOL4(10),SYMBOL5(10)
      DIMENSION A(64),B(19),SYMBOL1(10),BI(19),IBI(19)
C---------------------------------------------------------------------------
C--                      SPECTROGRAM OVERPRINT SYMBOLS                      --
C---------------------------------------------------------------------------
      DATA SYMBOL1/1H ,1H ,1H+,1HX,1HX,1HX,1HX,1HX,1HX,1HX/
      DATA SYMBOL2/1H ,1H ,1H ,1H ,1H-,1H+,1HO,1HO,1HO,1HO/
      DATA SYMBOL3/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H-,1H-,1H /
      DATA SYMBOL4/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H+,1H+/
      DATA SYMBOL5/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H*/
C---------------------------------------------------------------------------
C--                            PROGRAM VARIBLES                             --
C---------------------------------------------------------------------------
C     NUMBER OF RECORDS TO BE READ
      NREC=12
```

```
C      A NUMBER ONE MORE THAN THE MAXIMUM RECORD LENGTH
       NN2=2229
C-----------------------------------------------------------------------------
C--                  INPUT ARRAY LOGARITHMICALLY COMPRESSED                 --
C-----------------------------------------------------------------------------
       NN1=64
 1     CONTINUE
       DO 305 I=1,NN2
 5     READ(1)(NCHAN,NDIM,(A(K),K=1,64))
       IF(EOF(1))310,30
 30    CONTINUE
       JJ=1
       DO 40 J=1,6
       B(JJ)=A(J)
       JJ=JJ+1
 40    CONTINUE
       DO 50 J=7,11,2
       B(JJ)=(A(J)+A(J+1))
       JJ=JJ+1
 50    CONTINUE
       DO 60 J=13,17,4
       B(JJ)=(A(J)+A(J+1)+A(J+2)+A(J+3))
       JJ=JJ+1
 60    CONTINUE
       SUM1=0
       DO 70 J=21,25
       SUM1=(SUM1+A(J))
 70    CONTINUE
       B(JJ)=SUM1
       SUM2=0
       DO 80 J=26,31
       SUM2=(SUM2+A(J))
 80    CONTINUE
       JJ=JJ+1
       B(JJ)=SUM2
```

```
      SUM3=0
      DO 90  J=32,40
      SUM3=(SUM3+A(J))
 90   CONTINUE
      JJ=JJ+1
      B(JJ)=SUM3
      SUM4=0
      DO 100  J=41,50
      SUM4=(SUM4+A(J))
100   CONTINUE
      JJ=JJ+1
      B(JJ)=SUM4
      SUM5=0
      DO 110  J=51,64
      SUM5=(SUM5+A(J))
110   CONTINUE
      JJ=JJ+1
      B(JJ)=SUM5
C------------------------------------------------------------------------------------
C--                        ARRAY VALUES CONVERTED TO INTEGER FORM                    --
C------------------------------------------------------------------------------------
      DO 240  JJ=1,16
      BI(JJ)=(B(JJ)+.5)
      IBI(JJ)=IFIX(BI(JJ))
240   CONTINUE
      IF(I.GT.1) GO TO 295
C------------------------------------------------------------------------------------
C--                 COMPRESSED ARRAY AND ASSOCIATED SPECTROGRAM OUTPUT               --
C------------------------------------------------------------------------------------
      PRINT 250
250   FORMAT(1H1,///,87X,*SYMBOLS REPRESENT INTEGER VALUES AS FOLLOWS:*)
      PRINT 260
260   FORMAT(83X,"0=BLANK",2X,"1=( )",2X,"2=(+)",2X,"3=(X)",
     12X,"4=(X)")
      PRINT 266
266   FORMAT("+",112X,"    - ")
```

```
      PRINT 261
261   FORMAT(83X,"5=(X)",2X,"6=(X)",2X,"7=(X)",2X,"8=(X)",2X,
     1"9=(X)")
      PRINT 262
262   FORMAT("+",82X,"   + "2X,"   0 ",2X,"   0 ",2X,"   0 ",2X,"   0 ")
      PRINT 263
263   FORMAT("+",96X,"   - ",2X,"   - ",2X,"   - ")
      PRINT 264
264   FORMAT("+",103X,"   + ",2X,"   + ")
      PRINT 265
265   FORMAT("+",110X,"   * ")
      PRINT 270
270   FORMAT(92X,*0000000001111111*)
      PRINT 280
280   FORMAT(92X,*1234567890123456*)
      PRINT 290
290   FORMAT(89X,*-------------------*)
295   CONTINUE
      PRINT 210,(B(JJ),JJ=1,16),I,(SYMBOL1(IBI(JJ)+1),JJ=1,16)
210   FORMAT(1X,16F5.2,7X,I4,16A1)
      PRINT 211,(SYMBOL2(IBI(JJ)+1),JJ=1,16)
      PRINT 211,(SYMBOL3(IBI(JJ)+1),JJ=1,16)
      PRINT 211,(SYMBOL4(IBI(JJ)+1),JJ=1,16)
      PRINT 211,(SYMBOL5(IBI(JJ)+1),JJ=1,16)
211   FORMAT("+",91X,16A1)
C-------------------------------------------------------------------------------
C--     COMPRESSED ARRAY WRITTEN TO TAPE2 TO ALLOW DATA TO BE TRANSFERRED      --
C--   TO PERMANENT FILE UPON COMPLETION OF PROGRAM.                            --
C-------------------------------------------------------------------------------
500   CONTINUE
      WRITE(2,315)(B(JJ),JJ=1,16)
315   FORMAT(16F6.3)
305   CONTINUE
310   CONTINUE
      ENDFILE2
      WRITE (6,511) (NREC,NN1,NN2,I,K)
```

```
511   FORMAT(5X,"NREC= ",I3,5X,"NN1= ",I3,5X,"NN2= ",I5,5X,"I= ",I5,5X,
     C"K= ",I3)
      WRITE (6,998)
  998 FORMAT (1H1,//)
      NREC=NREC-1
      IF(NREC.GT.0) GO TO 1
      STOP
      END
```

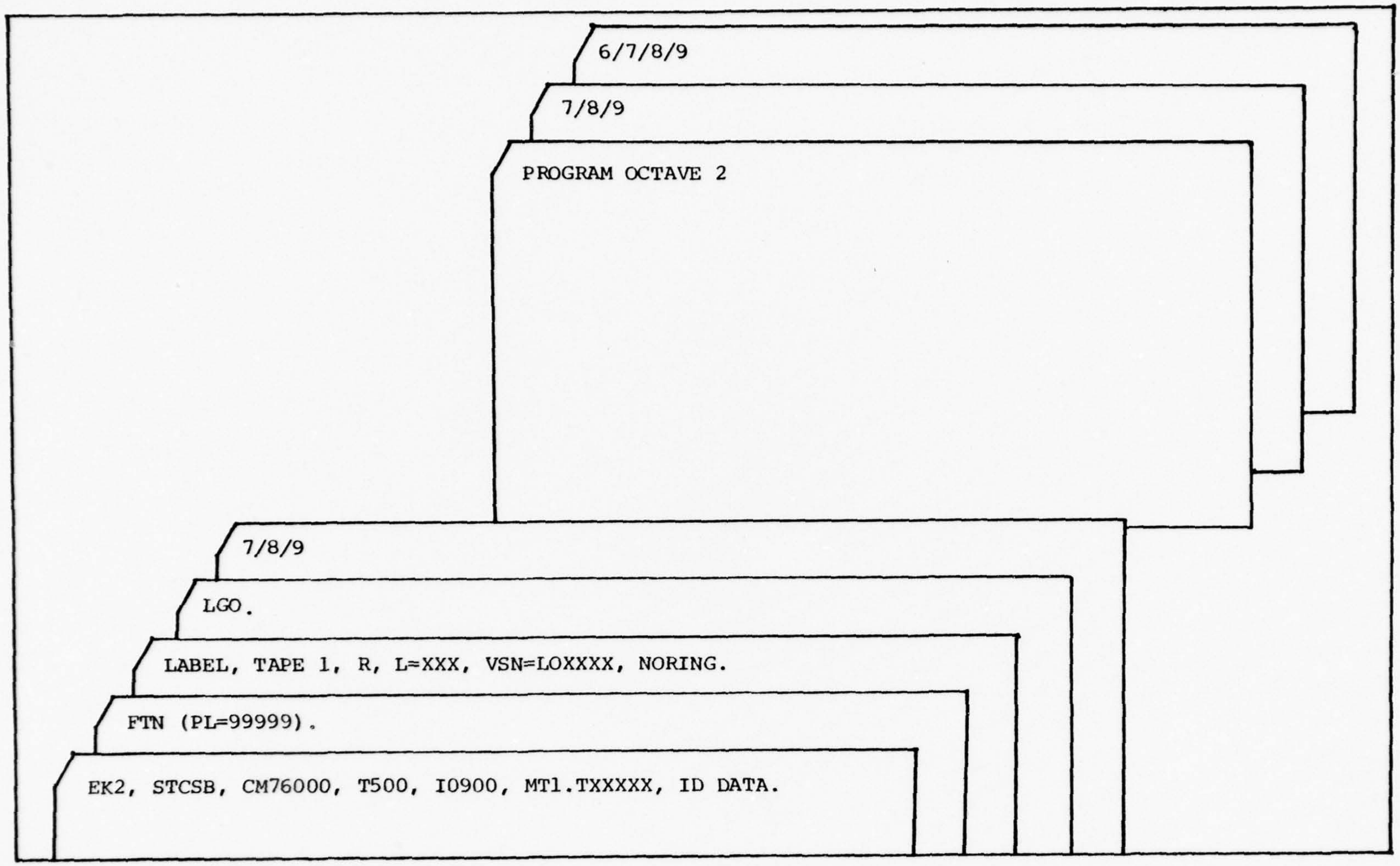


Figure 23. Program OCTAVE2

```
C*******************************************************************************
C*******************************************************************************
C**     THIS PROGRAM ATTACHES THE MAGNETIC TAPE CONTAINING THE               **
C**     16-CHANNELS OF DIGITIZED SPEECH DATA AND PRODUCES A NORMALIZED       **
C**     VERSION OF THE SPECTROGPAM.                                          **
C*******************************************************************************
C*******************************************************************************
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
C--                                                                          --
C--     THE FOLLOWING FORTRAN STATEMENTS MUST BE ADJUSTED FOR                --
C--     EACH RUN                                                             --
C--                                                                          --
C--       PROGRAM OCTAVE2:  26    27    26    29                             --
C--                                                                          --
C-------------------------------------------------------------------------------
C-------------------------------------------------------------------------------
      PROGRAM OCTAVE2(INPUT,OUTPUT,TAPE1,TAPE6=OUTPUT)
      DIMENSION SYMBOL2(10),SYMBOL3(10),SYMBOL4(10),SYMBOL5(10)
      DIMENSION B(16),SYMBOL1(10),BI(16),ISI(16),A(16)
      DATA SYMBOL1/1H ,1H ,1H+,1HX,1HX,1HX,1HX,1HX,1HX,1HX/
      DATA SYMBOL2/1H ,1H ,1H ,1H ,1H-,1H+,1HO,1HO,1HO,1HO/
      DATA SYMBOL3/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H-,1H-,1H#/
      DATA SYMBOL4/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H+,1H+/
      DATA SYMBOL5/1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H*/
      NREC=12
      NN1=16
      NSTART=1
      NSTOP=2230
      L=0
 1    CONTINUE
      DO 305 I=1,NSTOP
 5    READ(1,10)(B(J),J=1,NN1)
 10   FORMAT(16F6.3)
      IF(EOF(1))310,30
 30   CONTINUE
```

```
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CXXXX                              NORMALIZATION  ROUTINE                           XXXXX
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      SUME=0.0
      DO 33  J=1,16
      SUME=SUME + (B(J))**2
   33 CONTINUE
      DO 34  J =1,16
      ENERGY=SQRT(SUME)
      IF(ENERGY.GT.0.50) GO TO 32
      ENERGY=1.0
   32 CONTINUE
      B(J) = (B(J)/ENERGY)*10.
   34 CONTINUE
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CXXXX                                END    NORMALIZATION                            XXXX
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      IF(I.LT.NSTART) GO TO 305
      L=L+1
      DO 240 JJ=1,16
      IF(B(JJ).LE.9.0) GO TO 31
       B(JJ)=9.0
   31  CONTINUE
       BI(JJ)=(B(JJ)+.5)
       IBI(JJ)=IFIX(BI(JJ))
  240  CONTINUE
       IF(L.GT.1) GO TO 295
       PRINT*," "
       PRINT*," "
       PRINT*," "
       PRINT*," THIS IS A LISTING OF THE PROTOTYPE VALUES AS USED "
       PRINT*," ON THE CORRELATION PROGRAM.  THESE COEFFICIENTS "
       PRINT*," ARE TO BE USED WITH THE SPEECH PRODUCING ALGORITHM "
       PRINT*," WHICH IS NOW BEING DESIGNED. "
       PRINT*," "
       PRINT*," "
```

```
      PRINT 250
250   FORMAT(///,87X,*SYMBOLS REPRESENT INTEGER VALUES AS FOLLOWS:*)
      PRINT 260
260   FORMAT(83X,"0=BLANK",2X,"1=( )",2X,"2=(+)",2X,"3=(X)",
     12X,"4=(X)")
      PRINT 266
266   FORMAT("+",112X,"    - ")
      PRINT 261
261   FORMAT(83X,"5=(X)",2X,"6=(X)",2X,"7=(X)",2X,"8=(X)",2X,
     1"9=(X)")
      PRINT 262
262   FORMAT("+",92X,"    + "2X,"    0 ",2X,"    0 ",2X,"    0 ",2X,"    0 ")
      PRINT 253
263   FORMAT("+",96X,"    - ",2X,"    - ",2X,"    - ")
      PRINT 264
264   FORMAT("+",103X,"    + ",2X,"    + ")
      PRINT 265
265   FORMAT("+",110X,"    * ")
      PRINT 270
270   FORMAT(92X,*0000000001111111*)
      PRINT 280
280   FORMAT(92X,*1234567890123456*)
      PRINT 290
290   FORMAT(89X,*--------------------*)
295   CONTINUE
      PRINT 210,(B(JJ),JJ=1,16),I,(SYMBOL1(IBI(JJ)+1),JJ=1,16)
210   FORMAT(1X,16F5.2,8X,I4,16A1)
      PRINT 211,(SYMBOL2(IBI(JJ)+1),JJ=1,16)
      PRINT 211,(SYMBOL3(IBI(JJ)+1),JJ=1,16)
      PRINT 211,(SYMBOL4(IBI(JJ)+1),JJ=1,16)
      PRINT 211,(SYMBOL5(IBI(JJ)+1),JJ=1,16)
211   FORMAT("+",91X,16A1)
305   CONTINUE
      DO 306 I=1,110
      READ(1,10)(B(J),J=1,NN1)
      IF(EOF(1)) 310,306
```

```
  306 CONTINUE
 310  CONTINUE
      PRINT*," "
      PRINT*," "
      NREC=NREC-1
      IF(NREC.GT.0) GO TO 1
      STOP
      END
```

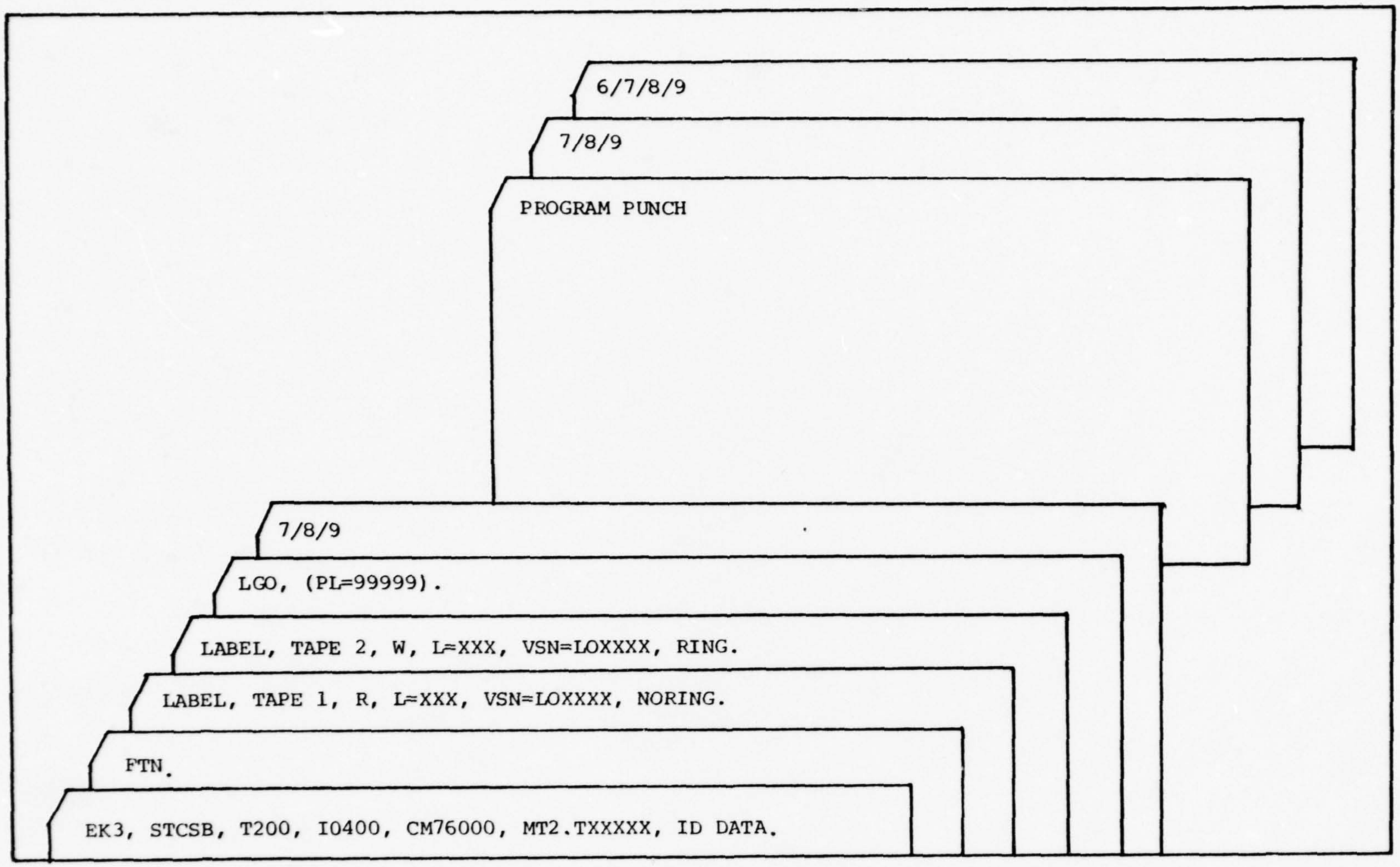


Figure 24. Program PUNCH

```
C------------------------------------------------------------------------
C------------------------------------------------------------------------
C--                                                                    --
C--    THIS PROGRAM ATTACHES THE MAGNETIC TAPE CONTAINING THE           --
C--    16-CHANNELS OF DIGITIZED SPEECH DATA AND SELECTIVELY             --
C--    PUNCHES A SET OF CARDS FOR THE PROTOTYPE PHONEME                 --
C--    FREQUENCY VECTORS OF INTEREST.                                   --
C--                                                                    --
C------------------------------------------------------------------------
C------------------------------------------------------------------------
C--                                                                    --
C--    THE FOLLOWING FORTRAN STATEMENTS MUST BE ADJUSTED FOR EACH RUN   --
C--                                                                    --
C--    PROGRAM PUNCH:      21-25       27       32                      --
C--                                                                    --
C------------------------------------------------------------------------
C------------------------------------------------------------------------
      PROGRAM PUNCH(INPUT,OUTPUT,TAPE1,PUNCH,TAPE6=OUTPUT)
      DIMENSION E(16)
      DIMENSION A(16)
      DIMENSION IBGN1(14),IEND1(14)
      DATA IBGN1/28,144,262,383,514,645,786,915,1032,1166,1325,1441,
     C1552,1679/
      DATA IEND1/39,155,273,394,525,656,797,926,1043,1177,1336,1452,
     C1573,1690/
C
C     GARY'S  'O' PROTOTYPE OF LENGTH 10
C
C
C
C NUMBER OF RECORDS TO SKIP
      IREC=7
      DO 10 K=1,IREC
      DO 5 L=1,2400
      READ(1,100) (E(J),J=1,16)
      IF(EOF(1)) 10,3
```

```
    3 CONTINUE
    5 CONTINUE
   10 CONTINUE
  100 FORMAT (16F6.3)
  200 FORMAT (8F9.3)
  300 FORMAT(5X,16F7.3)
  500 FORMAT(5X//////////)
      K=1
      DO 1 I=1,1718
      READ (1,100) (A(J),J=1,16)
      IF (EOF(1)) 13,26
   26 CONTINUE
      IF(K .GT. 14) GO TO 13
      B=IBGN1(K)
      C=IEND1(K)
      IF (I .LT. B) GO TO 1
      IF(I .GT. C) GO TO 40
      PUNCH 200, (A(J),J=1,16)
      WRITE (6,300) (A(J),J=1,16)
      GO TO 1
   40 K=K+1
      WRITE (6,500)
    1 CONTINUE
   13 CONTINUE
      STOP
      END
```

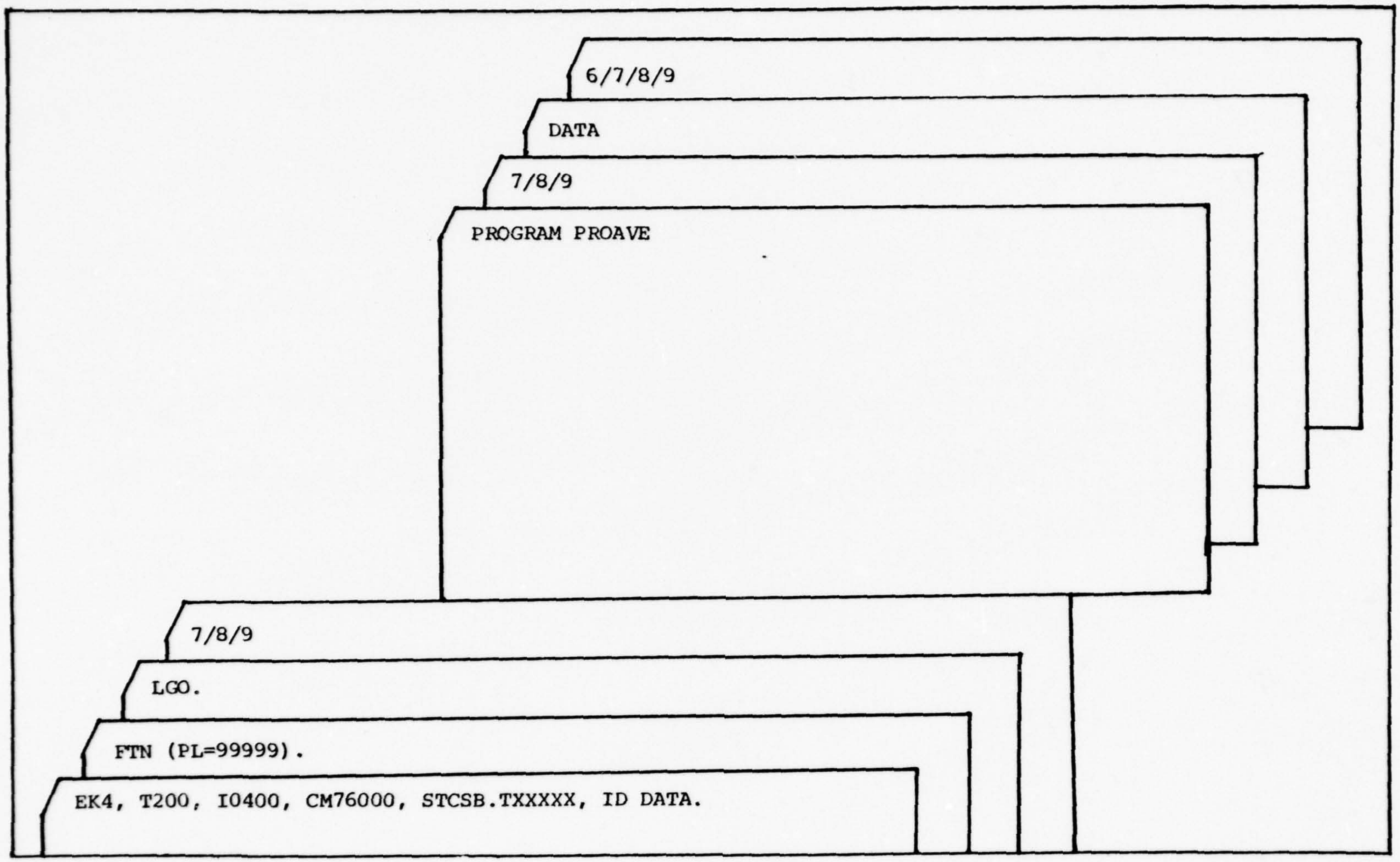


Figure 25. Program PROAVE

```
C------------------------------------------------------------------------
C------------------------------------------------------------------------
C--                                                                    --
C--     THIS PROGRAM CALCULATES AN AVERAGED PROTOTYPE PHONEME          --
C--     USING THE PUNCHED CARD SETS PRODUCED BY PROGRAM PUNCH.  THE    --
C--     OUTPUT IS A SET OF PUNCHED CARDS THAT REPRESENTS AN            --
C--     AVERAGED PROTOTYPE PHONEME.                                    --
C--                                                                    --
C------------------------------------------------------------------------
C------------------------------------------------------------------------
C------------------------------------------------------------------------
C------------------------------------------------------------------------
C--                                                                    --
C--     THE FOLLOWING FORTRAN STATEMENTS MUST BE ADJUSTED FOR EACH RUN --
C--     PROGRAM PROAVE:     23       25       27       29              --
C--                         30       32       33       61              --
C--                                                                    --
C------------------------------------------------------------------------
C------------------------------------------------------------------------
      PROGRAM PROAVE(INPUT,OUTPUT,PUNCH,TAPE6=OUTPUT)
      DIMENSION A(250,16),B(20,16)
C TOTAL NUMBER OF LINES OF INPUT DATA (HALF THE NUMBER OF DATA CARDS)
      KARD=234
C       BEGINNING VALUE OF PROTOTYPE FOR ROW ONE
      JB=1
C     END VALUE OF PROTOTYPE ROW ONE
      JE=226
C     LENGTH OF PROTOTYPE
      JI=9
C NUMBER OF PROTOTYPES TO BE AVERAGED
      DIV=25.
C               THIS IS THE COMPUTATION OF THE
C             "AUH" PROTOTYPE
C         USING 25 VALUES
      DO 1 I=1,JI
      DO 2 J=1,16
```

```
      B(I,J)=0.
    2 CONTINUE
    1 CONTINUE
      PRINT*,"INPUT VALUES ARE"
      DO 3 I=1,KARD
      READ 100,(A(I,J),J=1,16)
  100 FORMAT (8F9.3)
      WRITE(6,300)(A(I,J),J=1,16)
  300 FORMAT(16F7.3)
    3 CONTINUE
      DO 10 KK=1,JI
      DO 20 I=1,16
      DO 30 J=JB,JE,JI
      B(KK,I)=A(J,I)+B(KK,I)
   30 CONTINUE
      B(KK,I)=B(KK,I)/DIV
   20 CONTINUE
      PUNCH 500,(B(KK,K),K=1,16)
  500 FORMAT(8F9.3)
      JB=JB+1
      JE=JE+1
      IF(JE.GT.KARD) GO TO 110
   10 CONTINUE
  110 CONTINUE
      DO 101 I=1,JI
      PRINT*,"AVERAGED ENERGY FOR ROW ",I," FOR THE LETTER B IS:"
      WRITE (6,200)(B(I,J),J=1,16)
  200 FORMAT(16F8.3)
  101 CONTINUE
      STOP
      END
```

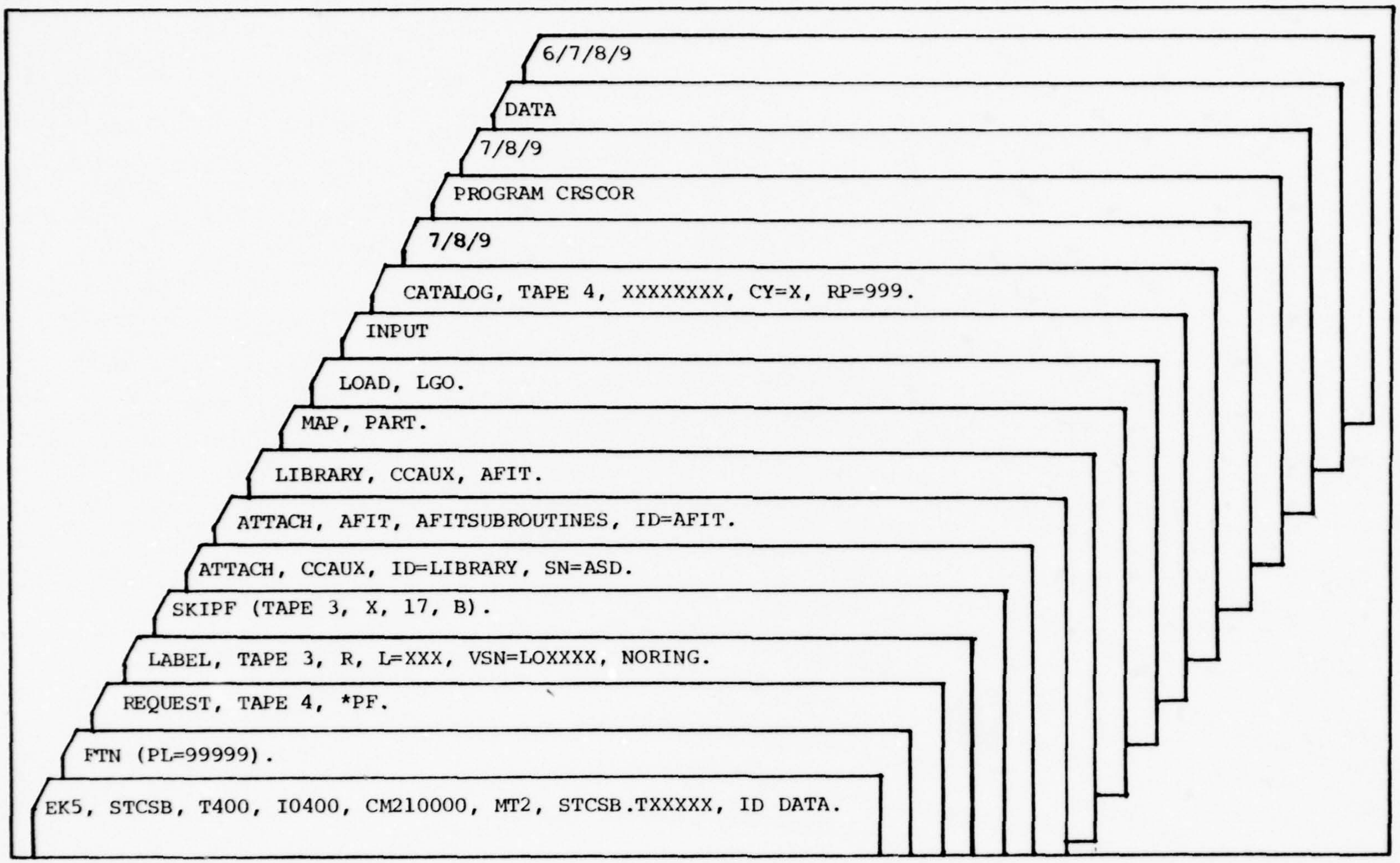


Figure 26. Program CRSCOR

```
C*************************************************************************
C*************************************************************************
C**    THIS PROGRAM IS A SPEECH PHONEME RECOGNITION SCHEME BASED ON      **
C**    PROTOTYPE MATCHING. THE SPEECH DATA IS READ (ONE SENTENCE AT A TIME)  **
C**    FROM AN INPUT FILE CALLED TAPE3. THE DATA MUST BE IN AN ARRAY 16XB,  **
C**    WHERE B<701. UP TO 30 PROTOTYPES OF SIZE 15XN, WHERE N<16, CAN BE   **
C**    ATTACHED FROM PF OR READ FROM CARDS. THE PROGRAM VARIABLES ARE SET   **
C**    IN THE MAIN PROGRAM AND FED THROUGH COMMON TO THE SUBROUTINE XCORR   **
C**    WHERE ALL THE ANALYSIS TAKES PLACE.                                 **
C*************************************************************************
C*************************************************************************
C-----------------------------------------------------------------------C
C-----------------------------------------------------------------------C
C                                                                       C
C      THE FOLLOWING FORTRAN STATEMENTS MUST BE ADJUSTED PRIOR          C
C      TO EACH RUN.                                                     C
C                                                                       C
C      PROGRAM CRSCOR                                                   C
C                      STATEMENT NUMBER(S):     36     41     42     44       C
C                                               51     59     66     84       C
C                                               86     88     91     93       C
C                                              102    107    113    117-145   C
C                                              152    153                     C
C                                                                       C
C      PROGRAM XCORR                                                    C
C                      STATEMENT NUMBER(S):    24-27  29-31   45    378       C
C                                              421    429    438    446       C
C                                              387    395    404    412       C
C                                                                       C
C-----------------------------------------------------------------------C
C-----------------------------------------------------------------------C
      PROGRAM CRSCOR(INPUT,OUTPUT,TAPE3,TAPE4,TAPE9=OUTPUT,PLOT,TAPE1=IN
     CPUT,TAPE2)
      DIMENSION GOOD(64),ITYP(64)
      DIMENSION M(16,15)
      DIMENSION IBGN1(2),IEND1(2)
```

```
      DIMENSION B(700,15)
      COMMON NSTART,NN2,NN3,NN5,ISUBLN,IOVLAP,NORMAL,NORMAR,ATOL,BTOL,
     1INHIP,LOOK,IDECID,GOOD,ITYP,ILIM,ILIN,I7SEL,IFILT,NSNORM
      COMMON B,INEND
      DATA IBGN1/1,673/
      DATA IEND1/672,1076/
C     TO ENABLE NOISE REDUCTION IN NORM ROUTINE SET NSNORM EQUAL TO ONE
      NSNORM=1
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
C
C    THIS RUN HAS ALL EIGHT PROTOTYPES
C                               CORRELATED WITH
C JOHN'S  SENTENCE:  BEFORE THE TRIP, THE RABBIT RESTED ALONG THE
C                    OPEN FIELD OF THE RANCHER.
C
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C
C
C-----------------------------------------------------------------------
C--                        TITLE OF WORD/SENTENCE BEING READ           --
C JOHN'S  SENTENCE:  BEFORE THE TRIP, THE RABBIT RESTED ALONG THE
C                    OPEN FIELD OF THE RANCHER.
C-----------------------------------------------------------------------
C#######################################################################
C##                          VARIBLES USED BY PROGRAM                  ##
C##             (*) MUST BE SET FOR EACH SENTENCE/WORD CHANGE          ##
C#######################################################################
C NUMBER OF ELEMENTS IN /DATA/ STATEMENT
      IRUN=2
      DO 5 KB=1,IRUN
      JNEW=IBGN1(KB)
      KK=1
      PRINT 26
```

```
   26 FORMAT(////,1X,"THE ACTUAL SENTENCE DATA IS",84X,"TIME"//)
      JNEWS=IEND1(KB)
      MIT=0
      DO 20 I=JNEW,JNEWS
      MIT=MIT+1
      READ (3,12) (B(MIT,J),J=1,16)
   12 FORMAT (16F6.3)
      WRITE (9,999) (B(MIT,J),J=1,16),I
  999 FORMAT(1X,16F6.3,15X,I4)
      KK=KK+1
   20 CONTINUE
      INEND=KK-1
C    *POSITION OF SENTENCE INFORMATION IN INPUT ARRAY
      NSTART=20
C    *NUMBER OF THE SENTENCE/WORD BEING READ
      NN5=1
C     SIZE OF LARGEST PROTOTYPE + ONE
      NN3=8
C     LENGTH OF SUB-SENTENCE(THIS ESTABLISHES THE SIZE SECTIONS
C        THE SENTENCE IS BROKEN INTO)
      ISUBLN=48
C     DESIRED SUB-SENTENCE OVERLAP
      IOVLAP=8
C
C ---------------------------------------------------------------------------
C ---      FILTERING DESIRED IN FOURIER SPACE                       ---------
C ---      FILTER RANGE:  0 TO 64                                   ---------
C ---      F=0 REMOVES FILTER FROM PROGRAM
C ---      F=64 REMOVES ALL FOURIER INFORMATION                     ---------
C ---------------------------------------------------------------------------
C----------------------------------------------------------------------------
      IFILT=0
C--                         ENERGY NORMALIZATION                          --
C----------------------------------------------------------------------------
C     IF ENERGY NORMALIZATION OF DATA IS DESIRED,SET "NORMAL" TO "1"
C          OTHERWISE SET "NORMAL" TO "0"
```

```
      NORMAL=1
C----------------------------------------------------------------------
C--                              PRINTOUTS                          --
C----------------------------------------------------------------------
C     TO INHIBIT PRINTOUT OF PROTOTYPSET "INHIB" TO 0,OTHERWISE
C        SET "INHIB" TO 1.
      INHIB=1
C----------------------------------------------------------------------
C--                     PROTOTYPE SIZE = ITYP(X)                    --
C----------------------------------------------------------------------
      ITYP(1)=5
      ITYP(2)=7
      ITYP(3)=5
      ITYP(4)=6
      ITYP(5)=3
      ITYP(6)=2
      ITYP(7)=2
      ITYP(8)=4
      ITYP(9)=1
      ITYP(10)=1
      ITYP(11)=1
      ITYP(12)=1
      ITYP(13)=1
      ITYP(14)=1
      ITYP(15)=1
      ITYP(16)=1
      ITYP(17)=1
      ITYP(18)=1
      ITYP(19)=1
      ITYP(20)=1
      ITYP(21)=1
      ITYP(22)=1
      ITYP(23)=1
      ITYP(24)=1
      ITYP(25)=1
      ITYP(26)=1
```

```
      ITYP(27)=1
      ITYP(28)=1
      ITYP(29)=1
      ITYP(30)=1
C-----------------------------------------------------------------------------
C--                    OUTPUT SENTENCE/WORD TITLE                           --
C-----------------------------------------------------------------------------
      PRINT 3
 3    FORMAT(////,1X,"THE WORD/SENTENCE BEING ANALYZED IS :")
      PRINT*,"BEFORE THE TRIP, THE RABBIT RESTED ALONG THE OPEN FIELD"
      PRINT*,"OF THE RANCHER."
      PRINT*,"SPOKEN BY: JOHN"
C-----------------------------------------------------------------------------
C--                   TRANSFER CONTROL TO SUBROUTINE                        --
C-----------------------------------------------------------------------------
      CALL XCORR
    5 CONTINUE
      ENDFILE4
      STOP
      END
C*****************************************************************************
C*****************************************************************************
C**     THIS SUBROUTINE USES FFT TECHNIQUES TO CROSSCORRELATE PROTOTYPES    **
C**    WITH SPEECH DATA.THE OUTPUT IS A PHONEMIC REPRESENTATION OF THE INPUT. **
C**    ALSO INCLUDED AS AN OUTPUT IS ALL THE CORRELATION COEFFICIENTS FOR   **
C**    EACH PROTOTYPE  BY RANK,IN TIME OCCURRANCE ORDER.                    **
C*****************************************************************************
C*****************************************************************************
      SUBROUTINE XCORR
      COMPLEX SENT(64,32),CPPOTO(64,32),CONPRO(64,32),CORR(64,32)
      REAL MARR
      DIMENSION SYMBOL1(30),SYMBOL2(30),SUMM(64),EPROTO(15,16)
      DIMENSION PRO(700,26),B(700,16),SAMPLE(700),TIME(700)
      DIMENSION NN(2),PROTO(15,16),C(64,16),D(64,16)
      DIMENSION GOOD(64),ITYP(64)
      DIMENSION SYMBOL3(1),SYMBOL4(1),SYMBOL5(1),SYMBOL6(1),SYMBOL7(1)
```

```
      DIMENSION M(16,16)
      COMMON NSTART,NN2,NN3,NN5,ISUBLN,IOVLAP,NORMAL,NORMAR,ATOL,BTOL,
     1INHIB,LOOK,IDECID,GOOD,ITYP,ILIM,ILIN,IZSEL,IFILT,NSNORM
      COMMON B,INEND
      EQUIVALENCE (CPROTO,CORR)
C     PHONEME SYMBOL SET
      DATA SYMBOL1/7H  AUH   ,7HLONG A ,7HLONG E ,7HLONG O ,7HLEAD B ,
     17HLEAD T ,7HLEAD D ,7HLEAD R ,1H ,
     11H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     C1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
C     PHONEME-WORD SET
      DATA SYMBOL2/4HAUH ,4HTAKE,4HSEE ,4HOVER,4HBITE,4HTIDE,4HDIP ,
     14HRIDE,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     C1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
C-------------------------------------------------------------------------------
C--                       VARIABLES PERTINENT TO TRANSFORM                    --
C-------------------------------------------------------------------------------
C     THE MAXIMUM ARRAY SIZE THAT CAN BE TRANSFORMED IS
      NN(1)=64
      NN(2)=32
C     SIZE OF REDUCED ARRAY
      NN4=16
C     SIZE OF EXPANDED ARRAY
      NN10=32
C     STARTING POINT IN J DIRECTION TO IMBED ARRAY IN ZEROS
      NN11=NN4+1
C     NUMBER OF PROTOTYPES
      NPRO=8
C     A VALUE ONE MORE THAN THE NUMBER OF SYMBOLS PROVIDED
      NZERO=16
C     LENGTH OF ARRAY TO BE SORTED
      N=64
C-------------------------------------------------------------------------------
C--              READ COMPRESSED SENTENCE DATA FROM PERMANENT FILE             --
C-------------------------------------------------------------------------------
      PRINT 22,NN5,INEND
```

```
 22   FORMAT(/,1X,"THE LENGTH OF THE SENTENCE #",I2,1X,"IS",I4)
C
C
C
C--------------------------------------------------------------------------------
C--                REDUCE SENTENCE TO SUB-SENTENCES OF LENGTH "ISUBLN"         --
C--------------------------------------------------------------------------------
      ISCLIM=((INEND-NSTART)/(ISUBLN-IOVLAP))+1
      PRINT 25,ISCLIM
 25   FORMAT(/,1X,"THE NUMBER OF SUB-SENTENCES REQUIRED IS",I3)
      K=1
      MSTART=0
      MSTOP=0
      DO 800 ISECTN=1,ISCLIM
      IF(MSTOP.GE.INEND) GO TO 706
      IF(ISECTN.EQ.1) GO TO 28
      REWIND 2
 28   CONTINUE
      IEND =0
      IF(ISECTN.NE.1) GO TO 31
      MSTART=NSTART
      GO TO 32
 31   CONTINUE
      MSTART=(MSTOP+1)-IOVLAP
 32   CONTINUE
      MSTOP=MSTART+(ISJBLN-1)
      IF(MSTOP.LE.INEND) GO TO 37
      MSTOP=INEND
 37   CONTINUE
      I=1
      DO 35 K=MSTART,MSTOP
      DO 34 J=1,NN4
      C(I,J)=B(K,J)
 34   CONTINUE
      I=I+1
 35   CONTINUE
```

```
      LEN=I-1
      PRINT 33,ISECTN,LEN
 33   FORMAT(////,1X,"THE LENGTH OF SUB-SENTENCE #",I2,1X,"IS",I4)
      IF(LEN.LT.22) GO TO 706
      IF(NORMAL.NE.1) GO TO 123
C
C-----------------------------------------------------------------------
C--                         ENERGY NORMALIZE SENTENCE                  --
C-----------------------------------------------------------------------
      IASIZE=64
      CALL NORM(C,D,LEN,NN4,IASIZE)
      GO TO 128
 123  CONTINUE
      DO 127 II=1,LEN
      DO 127 JJ=1,NN4
      D(II,JJ)=C(II,JJ)
 127  CONTINUE
 128  CONTINUE
C-----------------------------------------------------------------------
C--                MAKE SENTENCE COMPLEX AND APPEND TO ZEROS           --
C-----------------------------------------------------------------------
      IP=N-LEN
      PRINT 183 ,IP
 183  FORMAT(/,1X,"THE NUMBER OF ZEROS ADDED TO THE SUB-SENTENCE",
     1I4,/)
      DO 210 NK=1,IP
      DO 210 JJ=1,NN10
      SENT(NK,JJ)=(0.,0.)
 210  CONTINUE
      IP1=IP+1
      II=1
      DO 220 NK=IP1,N
      DO 215 JJ=1,NN4
      SENT(NK,JJ)=D(II,JJ)
 215  CONTINUE
      II=II+1
```

```
  220   CONTINUE
        DO 211 NK=IP1,N
        DO 211 JJ=NN11,NN10
        SENT(NK,JJ)=(0.,0.)
  211   CONTINUE
C-------------------------------------------------------------------------------
C--                                  FFT SENTENCE                              --
C-------------------------------------------------------------------------------
        CALL FOURT(SENT,NN,2,-1,0,0)
C
C
C###############################################################################
C##                            CROSSCORRELATION SEQUENCE                       ##
C###############################################################################
        DO 400 JP=1,NPRO
C-------------------------------------------------------------------------------
C--                    READ PROTOTYPE FROM CARDS/PERMANENT FILE                --
C-------------------------------------------------------------------------------
        IF(ISECTN.GT.1) GO TO 870
        DO 150 K=1,NN3
        READ 140,(PROTO(K,L),L=1,NN4)
        IF(EOF(1).NE.0)GO TO 151
  140   FORMAT(8F9.3)
  150   CONTINUE
  151   CONTINUE
        GO TO 875
  870   CONTINUE
        DO 874 K=1,NN3
        READ(2,871)(PROTO(K,L),L=1,NN4)
  871   FORMAT(16F6.3)
        IF(EOF(2).NE.0) GO TO 875
  874   CONTINUE
  875   CONTINUE
        NUM=K-1
        IF(INHIB.EQ.0) GO TO 147
        PRINT 153,JP,NUM
```

```
153   FORMAT(//,1X,"THE LENGTH OF PROTOTYPE #",I2,1X,"IS",I3)
      PRINT 144,SYMBOL1(JP),SYMBOL2(JP)
144   FORMAT(/,1X,"THE PROTOTYPE REPRESENTS",1X,A7,1X,"AS IN(",A6,")")
147   CONTINUE
      IF(ISECTN.GT.1) GO TO 149
      DO 152 K=1,NUM
      IF(INHIB.EQ.0)GO TO 148
      WRITE(9,146)(PROTO(K,L),L=1,NN4)
146   FORMAT(1X,16F6.3)
148   CONTINUE
       IF(ISECTN.GT.1) GO TO 152
       WRITE(2,145)(PROTO(K,L),L=1,NN4)
145    FORMAT(16F6.3)
152    CONTINUE
       IF(ISECTN.GT.1) GO TO 149
       ENDFILE2
       IF(NORMAL.NE.1) GO TO 159
C-----------------------------------------------------------------------------------------
C--                           ENERGY NORMALIZE PROTOTYPE                               --
C-----------------------------------------------------------------------------------------
149    CONTINUE
       IASIZE=15
       CALL NORM(PROTO,EPROTO,NUM,NN4,IASIZE)
       IF(INHIB.EQ.0) GO TO 969
       IF(ISECTN.GT.1) GO TO 969
       PRINT 965
965   FORMAT(/,1X,"VECTOR NORMALIZED PROTOTYPE")
      DO 967 K=1,NUM
      WRITE(9,155)(EPROTO(K,L),L=1,NN4)
155   FORMAT(1X,16F6.3)
967   CONTINUE
969   CONTINUE
154   CONTINUE
801   CONTINUE
      GO TO 161
159   CONTINUE
```

```
      DO 157 II=1,NUM
      DO 157 JJ=1,NN4
      EPROTO(II,JJ)=PROTO(II,JJ)
 157  CONTINUE
 161  CONTINUE
C------------------------------------------------------------------------------
C--            DETERMINE NUMBER OF ZEROS REQUIRED TO PREVENT "END EFFECT"    --
C------------------------------------------------------------------------------
      IZ=1
      ZEROS=NUM+LEN
      MARR=ZEROS
 160  MARR=MARR/2
      IF(MARR.LT.2) GO TO 170
      IZ=IZ+1
      GO TO 160
 170  IZ=IZ+1
      IDIN=2**IZ
      IF(INHIB.EQ.0) GO TO 171
      PRINT 173,IDIN
 171  CONTINUE
      IF(IDIN.GT.N) GO TO 704
 173  FORMAT(/,1X,"THE LENGTH OF SUPPLEMENTED PROTOTYPE & SENTENCE VECTO,
     1RS ARE",I4)
      IF (IDIN.GT.64) GO TO 702
C------------------------------------------------------------------------------
C--           MAKE PROTOTYPE COMPLEX AND APPEND NECESSARY ZEROS              --
C------------------------------------------------------------------------------
      DO 176 K=1,NJM
      DO 176 L=1,NN4
      CPROTO(K,L)=EPROTO(K,L)
 176  CONTINUE
      DO 177 K=1,NJM
      DO 177 L=NN11,NN10
      CPROTO(K,L)=(0.,0.)
 177  CONTINUE
      NUM1=NUM+1
```

```
      DO 180 K=NUM1,IDIN
      DO 180 L=1,NN10
      CPROTO(K,L)=(0.,0.)
 180  CONTINUE
C-----------------------------------------------------------------------
C--                              FFT PROTOTYPE                        --
C-----------------------------------------------------------------------
      CALL FOURT(CPROTO,NN,2,-1,0,0)
C-----------------------------------------------------------------------
C--                 FIND  COMPLEX CONJUGATE OF PROTOTYPE              --
C-----------------------------------------------------------------------
      DO 200 K=1,IDIN
      DO 200 L=1,NN10
      CONPRO(K,L)=CONJG(CPROTO(K,L))
 200  CONTINUE
 201  CONTINUE
C-----------------------------------------------------------------------
      SUME=0.0
      DO 994 I=1,64
      DO 994 J=1,32
      E=REAL(CONPRO(I,J))
      F=AIMAG(CONPRO(I,J))
      G=E**2+F**2
      SUME = SUME + G
994   CONTINUE
      ENERGY1=SQRT(SUME)
      PRINT*," THE ENERGY1 VALUE BEFORE FILTERING IS: ",ENERGY1
C-----------------------------------------------------------------------
C         FREQUENCY SELECTION FILTER  WITH VARIABLES MIDTH AND MENGTH
C-----------------------------------------------------------------------
      MIDTH=32-IFILT/2
      MENGTH=64-IFILT
      MM=MIDTH/2+1
      N4=NN10-MIDTH/2
      J=IDIN-(MENGTH/2-1)
      II=MENGTH/2+1
```

```
      DO 990 K1=1,IDIN
      DO 990 K2=1,NN10
      IF(K2.GT.MM.AND.K2.LE.N4) CONPRO(K1,K2)=(0.0,0.0)
      IF(K1.GT.II.AND.K1.LT.J) CONPRO(K1,K2) =(0.0,0.0)
  990 CONTINUE
C-----------------------------------------------------------------------------
C--                  PROTOTYPE  UNIT  NORMALIZATION
C-----------------------------------------------------------------------------
      SUME=0.0
      DO 996 I=1,64
      DO 996 J=1,32
      E=REAL(CONPRO(I,J))
      F=AIMAG(CONPRO(I,J))
      G=E**2+F**2
      SUME = SUME + G
  996 CONTINUE
      ENERGY = SQRT(SUME)
      PRINT*," THE ENERGY VALUE AFTER FILTERING IS: ",ENERGY
      ENGREM=1.0 - (ENERGY/ENERGY1)
      PRINT*," THE PERCENTAGE OF ENERGY REMOVED IS: ",ENGREM
      DO 997 I=1,64
      DO 997 J=1,32
      CONPRO(I,J) = CONPRO(I,J)/ENERGY
  997 CONTINUE
      GOOD(JP)=ENERGY
C-----------------------------------------------------------------------------
C--                  CACULATE CORRELATION IN FREQUENCY DOMAIN              --
C-----------------------------------------------------------------------------
      DO 250 K=1,IDIN
      DO 250 L=1,NN10
      CORR(K,L)=CONPRO(K,L)*SENT(K,L)
  250 CONTINUE
C-----------------------------------------------------------------------------
C--                            TAKE INVERSE TRANSFORM                      --
C-----------------------------------------------------------------------------
      CALL FOURT(CORR,NN,2,+1,+1,0)
```

```
C
      DO 290 IK=1,IDIN
      SUMM(IK)=CORR(IK,1)
  290 CONTINUE
C-----------------------------------------------------------------------
C    STORE THE CORRELATION VECTOR IN PROTOTYPE ARRAY (PRO)
C-----------------------------------------------------------------------
      IDEN=IP+1
      KSEC=ISUBLN-IOVLAP
      IOFSET=(ISECTN-1)*KSEC
      LAP=IDIN-IOVLAP
      PRINT 27
   27 FORMAT (////,1X,"THE TIME DOMAIN CORRELATION VALUES ARE")
      DO 300 KK=IDEN,LAP
      LP=KK+IOFSET-(IDEN-1)
      PRO(LP,JP)=SUMM(KK)/GOOD(JP)
 300  CONTINUE
C  PRINT CORR VALUES
      K1=IDEN+IOFSET-(IDEN-1)
      WRITE(9,998)(PRO(KK,JP),KK=K1,LP)
  998 FORMAT(1X," THIS"/(((15(2X,F6.3)))))
  400 CONTINUE
  800 CONTINUE
 706  PRINT*," REST OF DATA INSUFFICIENT LENGTH,SENTENCE TRUNCATED."
      IEND=(ISCLIM-1)*KSEC
      NP1=IEND+1       $       NP2=IEND+2
      PRINT*," THE LENGTH OF PROTOTYPE ARRAY IS ",IEND," TIME UNITS."
      PRINT*," FILTER USED IN THIS RUN: FILTER = ",IFILT
C-----------------------------------------------------------------------
C--                      OUTPUT CORRELATION DATA                     --
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C--                   OUTPUT CORRELATION COEFFICIENTS                --
C ---   WRITE THE CORRELATION DATA ON PERMANANT FILE FOR FUTURE USE   -------
C-----------------------------------------------------------------------
C
```

```
      NSENT=NN5    $    NREC=IFND
      WRITE(9,1290)NSENT,NREC,NPRO,NSTART
1290  FORMAT(4I3)
      PRINT*, "NSTART= ",NSTART
      PRINT*, "NSENT= ",NSENT
      PRINT*, "NREC= ",NREC
      PRINT*, "NPRO= ",NPRO
      PRINT*, "IEND= ",IEND
      PRINT*, "NP1= ",NP1
      PRINT*, "NP2= ",NP2
      DO 1310 I=1,IEND
      WRITE(9,1320)(PRO(I,J),J=1,26)
1320  FORMAT(13F6.3)
      WRITE (4,991) (PRO(I,J),J=1,26)
 991  FORMAT (13F6.3)
1310  CONTINUE
C*************************************************************************
C***********GRAPH ROUTINE **********
C*************************************************************************
      DO 701 J=1,NPRO
      DO 1150 I=1,IEND
      SAMPLE(I)=PRO(I,J)       $       TIME(I)=I*(NSTART-1)
1150  CONTINUE
      IF(J .EQ. 1) GO TO 1
      IF(J .EQ. 2) GO TO 2
      IF(J .EQ. 3) GO TO 3
      IF(J .EQ. 4) GO TO 4
      IF(J .EQ. 5) GO TO 5
      IF(J .EQ. 6) GO TO 6
      IF(J .EQ. 7) GO TO 7
      IF(J .EQ. 8) GO TO 8
      IF(J .EQ. 9) GO TO 9
   1  CALL DSP(2HBB,0)
      CALL PLOT(0.,1.,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7H  AUH  ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
```

```
      CALL  SCALE(TIME,8.25,IEND,1)
      CALL  AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
      CALL  FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL  PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.1) GO TO 701
    2 CALL  PLOT(0.,5.5,-3)
      CALL  SCALE(SAMPLE,3.5,IEND,1)
      CALL  AXIS(0.,0.,7HLONG A ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL  SCALE(TIME,8.25,IEND,1)
      CALL  AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
      CALL  FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL  PLOT(0.,-5.5,-3)
      GO TO 11
    3 CALL  PLOT(0.0,1.0,-3)
      CALL  SCALE(SAMPLE,3.5,IEND,1)
      CALL  AXIS(0.,0.,7HLONG E ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL  SCALE(TIME,8.25,IEND,1)
      CALL  AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
      CALL  FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL  PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.3) GO TO 701
    4 CALL  PLOT(0.,5.5,-3)
      CALL  SCALE(SAMPLE,3.5,IEND,1)
      CALL  AXIS(0.,0.,7HLONG O ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL  SCALE(TIME,8.25,IEND,1)
      CALL  AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
      CALL  FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL  PLOT(0.,-5.5,-3)
      GO TO 11
    5 CALL  PLOT(0.0,1.0,-3)
      CALL  SCALE(SAMPLE,3.5,IEND,1)
      CALL  AXIS(0.,0.,7HLEAD B ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL  SCALE(TIME,8.25,IEND,1)
      CALL  AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
```

```
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.5) GO TO 701
    5 CALL PLOT(0.,5.5,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLEAD T ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,8.25,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-5.5,-3)
      GO TO 11
    7 CALL PLOT(0.0,1.0,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLEAD D ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,8.25,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.7) GO TO 701
    8 CALL PLOT(0.,5.5,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLEAD R ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,8.25,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-5.5,-3)
      GO TO 11
    9 CALL PLOT(0.0,1.0,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,5HBPRO1,5,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,8.25,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,8.25,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-1.,-3)
```

```
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.9) GO TO 701
   11 CALL PLOT(3.0,0.0,-3)
      CALL PLOTE(N)
  701 CONTINUE
      RETURN
  702 STOP"ARRAY EXCEEDS DIMENSIONS"
  703 STOP"ID EXCEEDS LIMIT"
  704 STOP"IDIN NOT EQUAL TO N"
  705 STOP
      END
C*****************************************************************************
C          SUBROUTINE USED TO NORMALIZE DATA AT EACH TIME INCREMENT
C*****************************************************************************
      SUBROUTINE NORM(DATA,RDATA,IX,IY,IZ)
      DIMENSION GOOD(64),ITYP(64)
      DIMENSION B(700,16)
      DIMENSION DATA(IZ,16),RDATA(IZ,15)
      DIMENSION M(16,15)
      COMMON NSTART,NN2,NN3,NN5,ISUBLN,IOVLAP,NORMAL,NORMAR,ATOL,BTOL,
     1INHIB,LOOK,IDECID,GOOD,ITYP,ILIM,ILIN,IZSEL,IFILT,NSNORM
      COMMON B,INEND
      DO 25 II=1,IX
      SUME=0
      DO 20 JJ=1,IY
      SUME=SUME+DATA(II,JJ)**2
 20   CONTINUE
      ENERGY=SQRT(SUME)
      IF (NSNORM .EQ. 0) GO TO 30
      IF(ENERGY .LE. 0.5) GO TO 40
 30   CONTINUE
      DO 31 JJ=1,IY
      RDATA(II,JJ)=DATA(II,JJ)/ENERGY
   31 CONTINUE
      GO TO 25
```

```
   40 CONTINUE
      DO 50 KZ=1,IY
      RDATA(II,KZ)=0.001
   50 CONTINUE
 25   CONTINUE
      RETURN
      END
```

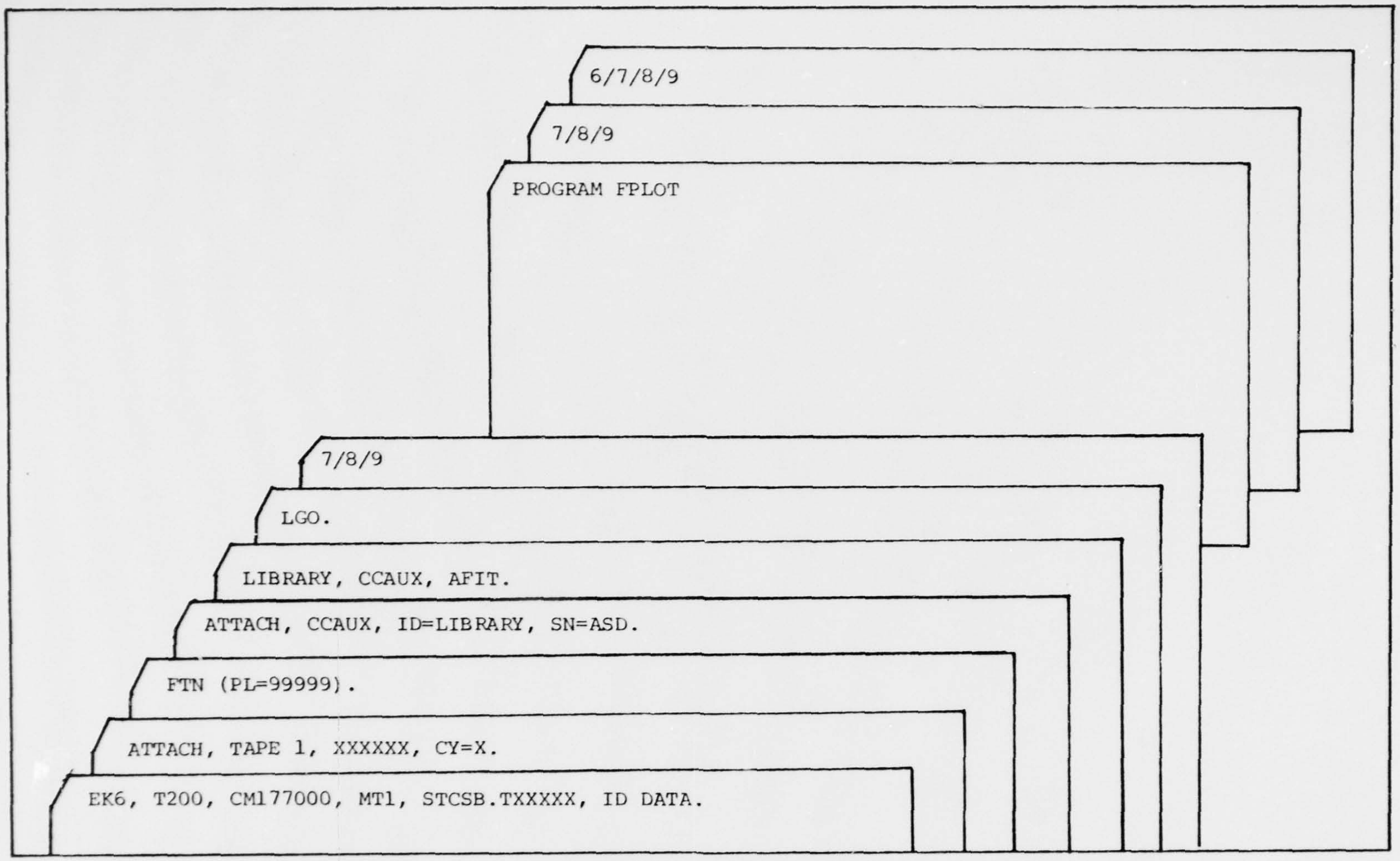


Figure 27. Program FPLOT

```
C-----------------------------------------------------------------------C
C-----------------------------------------------------------------------C
C                                                                       C
C       THIS PROGRAM ATTACHES THE PERMANENT FILE OF                     C
C       CORRELATION VALUES PRODUCED BY PROGRAM CPSCOP                   C
C       AND GENERATES A PLOT OF THE CORRELATION VALUES VERSUS           C
C       TIME FOR EACH PHONEME.                                          C
C                                                                       C
C-----------------------------------------------------------------------C
C-----------------------------------------------------------------------C
C-----------------------------------------------------------------------C
C-----------------------------------------------------------------------C
C-----------------------------------------------------------------------C
C                                                                       C
C                                                                       C
C      THE FOLLOWING FORTRAN STATEMENTS MUST BE ESTABLISHED BEFORE      C
C      EXECUTING THE PROGRAM:                                           C
C                                                                       C
C                           DIMENSION  IEND2                            C
C                           DATA       IEND2                            C
C                           NPRO                                        C
C                           IEND                                        C
C                           ISKIP                                       C
C                           IRUN                                        C
C                           ALL AXIS LABELS IN PLOT CALLS               C
C                                                                       C
C-----------------------------------------------------------------------C
C-----------------------------------------------------------------------C
      PROGRAM FPLOT(INPUT,OUTPUT,TAPE1,TAPE6=OUTPUT,PLOT)
      DIMENSION PRO(700,26),SAMPLE(700),TIME(700)
      DIMENSION IEND2(3)
C THE LENGTH OF EACH OF THE CORRELATION ARRAYS IS FOUND BY REFERRING TO THE
C CORRELATION PROGRAM'S OUTPUT AND NOTING THE VALUE OF IEND FOR EACH SUBDIVISION
C THE VALUES PUT INTO THE DATA STATEMENT ARE THESE IEND VALUES.
      DATA IEND2/540,600,519/
```

```
C NUMBER OF PROTOTYPES TO BE PLOTTED:  NPRO
      NPRO=8
C NUMBER OF SUBDIVISIONS CONTAINED IN THE DATA STATEMENT TO BE SKIPPED.
C TO INCLUDE ALL OF THE SUBDIVISIONS "ISKIP" IS SET TO ZERO.
      ISKIP=1
C NUMBER OF SENTENCE SUBDIVISIONS IN DATA STATEMENT:  IRUN
      IRUN=3
      NSTART=23
      DO 702 KP=1,IRUN
      ILAST=IEND2(KP)
      IEND=ILAST
      READ(1,25) ((PRO(M,KT),KT=1,26),M=1,ILAST)
 25   FORMAT(13F6.3)
      IF(EOF(1) .NE. 0) STOP  "JOB FINISHED"
      IF(KP .LE. ISKIP) GO TO 702
      NP1=ILAST + 1
      NP2=ILAST + 2
      DO 701 J=1,NPRO
      PRINT*," "
      PRINT*," "
      PRINT*," "
      PRINT*," "
      PRINT*," "
      PRINT*,"DATA FOR PROTOTYPE NUMBER:   ",J
      PRINT*,"NPRO=  ",NPRO
      PRINT*,"IEND= ",ILAST
      PRINT*," "
      PRINT*,"CORRELATION VALUES READ IN FROM TAPE        CORRELATION VA
     CLUES TO BE PLOTTED          TIME"
      PRINT*," "
      PRINT*," "
      DO 709 I=1,ILAST
      SAMPLE(I)=PRO(I,J)
      TIME(I)=I + (NSTART - 1)
      WRITE(6,15) (PRO(I,J)),SAMPLE(I),TIME(I)
 15   FORMAT(19X,F6.3,32X,F6.3,22X,F6.1)
```

```
709   CONTINUE
      IF(J .EQ. 1) GO TO 1
      IF(J .EQ. 2) GO TO 2
      IF(J .EQ. 3) GO TO 3
      IF(J .EQ. 4) GO TO 4
      IF(J .EQ. 5) GO TO 5
      IF(J .EQ. 6) GO TO 6
      IF(J .EQ. 7) GO TO 7
      IF(J .EQ. 8) GO TO 8
      IF(J .EQ. 9) GO TO 9
  1   CALL DSP(2HBB,0)
      CALL PLOT(0.,1.,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7H  AUH  ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.1) GO TO 701
   2  CALL PLOT(0.,5.5,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLONG A ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-5.5,-3)
      GO TO 11
    3 CALL PLOT(0.0,1.0,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLONG F ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
```

```
      IF(J.EQ.3) GO TO 701
    4 CALL PLOT(0.,5.5,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLONG O ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-5.5,-3)
      GO TO 11
    5 CALL PLOT(0.0,1.0,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLEAD P ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.5) GO TO 701
    6 CALL PLOT(0.,5.5,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLEAD T ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-5.5,-3)
      GO TO 11
    7 CALL PLOT(0.0,1.0,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,0.,7HLEAD O ,7,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.7) GO TO 701
    8 CALL PLOT(0.,5.5,-3)
```

```
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,  .,7HLEAD P ,7,3.5,90.  ,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-5.5,-3)
      GO TO 11
    9 CALL PLOT(0.0,1.0,-3)
      CALL SCALE(SAMPLE,3.5,IEND,1)
      CALL AXIS(0.,  .,5HBPRO1,5,3.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      CALL SCALE(TIME,16.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,16.0,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
      CALL PLOT(0.,-1.,-3)
      IF(J.EQ.NPRO) GO TO 11
      IF(J.EQ.9) GO TO 701
   11 CALL PLOT(13.0,0.0,-3)
      CALL PLOTE(N)
  701 CONTINUE
  702 CONTINUE
      PRINT*," "
      PRINT*," "
      PRINT*," "
      PRINT*,"JOB FINISHED -- PICK UP PLOTS"
      STOP
      END
```

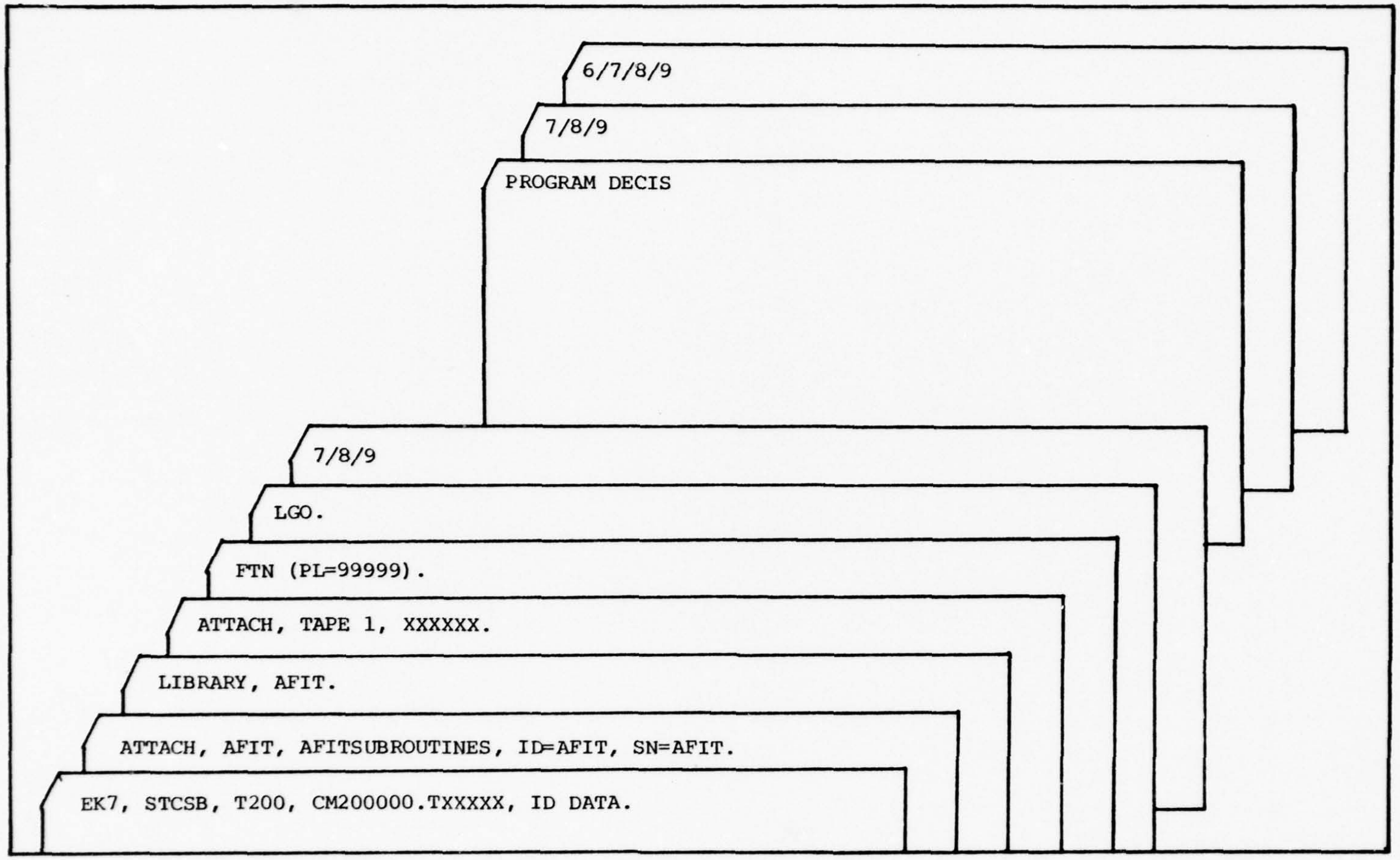


Figure 28. Program DECIS

```
C****************************************************************************
C*****   THIS PROGRAM ATTACHES A PERMANANT FILE (TAPE1) AND PROCESSES         ******
C*****   THE FILE ACCORDING TO THE FOLLOWING SCHEME:                          ******
C*****   1) ARRAY VALUES LESS THAN A GIVEN THRESHOLD ARE GIVEN A VALUE OF     ******
C*****      ZERO.                                                             ******
C*****   2) ARRAY VALUES NOT SATISFYING THE RATE-OF-CHANGE CRITERIA ARE       ******
C*****      GIVEN A VALUE OF ZERO.                                            ******
C*****   3) ARRAY VALUES ALONG TIME AXIS ARE CHECKED FOR AN ENDURANCE         ******
C*****      GREATER THAN A PERCENTAGE OF THE SPECIFIED PROTOTYPE LENGTH.      ******
C*****   4) THE RESULTING ARRAY IS RANKED IN DESCENDING ORDER AND PRINTED     ******
C*****      FOR EACH TIME INCREMENT.                                          ******
C*****   5) IF A SERIES OF XXXXXXX'S APPEAR IN A LINE OF THE FINAL            ******
C*****      OUTPUT, THERE ARE AT LEAST TWO EQUAL CORRELATION VALUES FOR       ******
C*****      THAT PARTICULAR TIME INCREMENT.                                   ******
C****************************************************************************
C---------------------------------------------------------------------------C
C---------------------------------------------------------------------------C
C--  THE FOLLOWING FORTRAN STATEMENTS MUST BE ESTABLISHED FOR EACH RUN:     --C
C--      DIMENSION NREC( )               DATA SYMBOL1/ /                    --C
C--      DATA ITYP/ /                    DATA NREC/ /                       --C
C--      NSENT          NPRO             NSTART          THRHLD             --C
C--      IAS            ENDUR1           ENDUR2          IAD                --C
C--      DELTA                                                              --C
C---------------------------------------------------------------------------C
C---------------------------------------------------------------------------C
      PROGRAM DECIS(INPUT,OUTPUT,TAPE1,TAPE2=OUTPUT)
      DIMENSION BPRO(1,26),PFO(700,8),IPHON(700,9),SART(8),SART1(8)
      DIMENSION ITYP(30),SYMBOL1(30)
      DIMENSION NREC(2)
C----------------------------------------------------------------------------
C-----                      PHONEME SYMBOL SET                          ------
C----------------------------------------------------------------------------
      DATA SYMBOL1/7H  AUH  ,7HLONG A ,7HLONG E ,7HLONG O ,7HLEAD B ,
     17HLEAD T ,7HLEAD D ,7HLEAD R ,7HXXXXXXX,
     11H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,
     A1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H ,1H /
```

```
      DATA ITYP/5,7,5,6,3,2,2,4,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
     A1,1,1,1,1,1,1,1/
      DATA NREC/640,280/
C ---------------------------------------------------------------------------------
C -----     SET THRESHOLD, DELTA, AND TIME ENDURANCE VALUES                  ------
C ---------------------------------------------------------------------------------
      NSENT=1
      NPRO=8
      NSTART=20
      IAD=1
      IAS=1
      THRHLD=0.6
      DELTA=0.61
      ENDUR1=0.5
      ENDUR2=0.5
      DO 56 IBG=1,2
      WRITE(2,998)
      PRINT*," TIME                CORRELATION VALUES READ IN AS DATA"
      PRINT*," "
      ILAST=NREC(IBG)
      DO 10 I=1,ILAST
      READ(1,5)(BPRO(1,J),J=1,26)
5     FORMAT(13F6.3)
      IF(EOF(1)) 20,30
   30 CONTINUE
      DO 6 K=1,8
      PRO(I,K)=BPRO(1,K)
    6 CONTINUE
      IF((IAS .EQ. 1) .AND. (IAD .EQ. 1)) GO TO 29
      GO TO 31
   29 WRITE(2,50)(PRO(I,KK),KK=1,8)
   50 FORMAT(1X,8F6.3)
   31 CONTINUE
10    CONTINUE
20    CONTINUE
C ---------------------------------------------------------------------------------
```

```
C -----    LOAD PRO ARRAY WITH VALUES GREATER THAN THRESHOLD (THRHLD)      -------
C ---------------------------------------------------------------------------------
      DO 1095 I=1,NPRO
      DO 1090 J=1,ILAST
      IF(PRO(J,I).GT.THRHLD) GO TO 1090
      PRO(J,I)=0.
1090  CONTINUE
1095  CONTINUE
C ---------------------------------------------------------------------------------
C -----      TEST RATE-OF-CHANGE CRITERIA                                   ------
C ---------------------------------------------------------------------------------
      DO 917 IT=1,NPRO
      ILSST=ILAST-2
      DO 918 IZ=1,ILSST
      KZ=IZ+1
      KB=IZ+2
      DIF1=ABS(PRO(IZ,IT)-PRO(KZ,IT))
      DIF2=ABS(PRO(KZ,IT)-PRO(KB,IT))
      IF((DIF1 .GE. DELTA) .AND. (DIF2 .GE. DELTA)) PRO(KZ,IT)=0.0
  918 CONTINUE
  917 CONTINUE
C ---------------------------------------------------------------------------------
C -----     CHECK THRESHOLD ARRAY FOR PROPER TIME ENDURANCE                 ------
C ---------------------------------------------------------------------------------
      DO 1296 I=1,NPRO
      IFLAG=0        $       ICOUNT=0
      DO 1295 J=1,ILAST
      IF(PRO(J,I).EQ.0.) GO TO 1285
      IF(IFLAG.EQ.1) GO TO 1280
      IFLAG=1    $    MARK=J    $    ICOUNT=1
      GO TO 1295
1280  ICOUNT=ICOUNT+1
      GO TO 1295
1285  IF(IFLAG.EQ.0) GO TO 1295
      IF(I.NE.4) GO TO 71
      ENDUR=ENDUR2
```

```
      GO TO 73
   71 ENDUR=ENDUR1
   73 CONTINUE
      IFLAG=0        $       ITIME=ENDUR*ITYP(T)
      IF(ICOUNT.GE.ITIME) GO TO 1295
      MCCNT=MARK+ICOUNT
      DO 1290 JJ=MARK,MCCNT
      PRO(JJ,I)=0.
 1290 CONTINUE
 1295 CONTINUE
      MARK=0    $    ICOUNT=0
 1296 CONTINUE
      WRITE(2,998)
      PRINT*,"DATA AFTER DELTA AND ENDURANCE TESTING"
      PRINT*," "
      PRINT*,"TIME              DATA AFTER DELTA AND ENDURANCE TESTING"
      PRINT*," "
      PRINT*," "
      PRINT*,"            AUH        LONG A     LONG E     LONG O     LEAD B
     Z  LEAD T     LEAD D     LEAD R"
      PRINT*," "
      DO 1283 JR=1,ILAST
      JJ=JR+ NSTART
      PRINT 410,JJ,(PRO(JR,JT),JT=1,8)
  410 FORMAT(1X,I4,5X,8(F6.3,4X))
 1283 CONTINUE
C -----------------------------------------------------------------------------
C -----    USE SUBROUTINE TO SORT THE PRO ARRAY (SORT)                   ------
C -----------------------------------------------------------------------------
      DO 1500 ICOL=1,ILAST
      IFHON(ICOL,9)=30
      DO 1310 IROW=1,NPRO
      SAPT(IROW)=PRO(ICOL,IROW)
      SAFT1(IROW)=SAPT(IROW)
 1310 CONTINUE
      CALL SORT(NPRO,SAFT1)
```

```
      DO 1350 IB=1,8
      IIB=NPRO+1-IB
      DO 1320 IA=1,NPRO
      IF(SART1(IA).EQ.SART1(IIB)) GO TO 1330
1320  CONTINUE
      GO TO 1340
1330  IF(SART1(IIB).EQ.0.) GO TO 1340
      IPHON(ICOL,IB)=IA
      GO TO 1350
1340  IPHON(ICOL,IB)=30
1350  CONTINUE
1500  CONTINUE
      DO 1284 ICOL=1,ILAST
      DO 21 KL=1,7
      IF(IPHON(ICOL,KL).EQ.30) GO TO 1284
      N=KL+1
      DO 22 KM=N,8
      IF(IPHON(ICOL,KL).EQ.IPHON(ICOL,KM)) GO TO 23
   22 CONTINUE
   21 CONTINUE
      GO TO 1284
   23 IPHON(ICOL,9)=9
 1284 CONTINUE
      WRITE(2,998)
  998 FORMAT(1H1,//)
      PRINT*," DECISION SCHEME FOR SENTENCE NUMBER   ",NSENT
      PRINT*," NUMBER OF PROTOTYPES IN DECISION SCHEME  ",NPRO
      PRINT*,"PARAMETERS FOR THIS ITERATION:"
      PRINT*,"        THRESHOLD=  ",THRHLD
      PRINT*,"        RATE OF CHANGE CONSTANT=  ",DELTA
      PRINT*,"        ENDURANCE FOR VOWELS=  ",ENDUR1
      PRINT*,"        ENDURANCE FOR CONSTANTS=  ",ENDUR2
      PRINT*,"THE DECISION SCHEME OUTPUT RANKED FROM 1 TO 8 IS "
      DO 401 J=1,ILAST
      JJ=J+NSTART
      PRINT 411,JJ,(SYMBOL1(IPHON(J,N)),N=1,9)
```

```
  411 FORMAT(1X,I4,5X,9(A7,6X))
 401   CONTINUE
  56   CONTINUE
       REWIND 1
  58   CONTINUE
  72   CONTINUE
       STOP
       END
```

APPENDIX C

DATA RESULTS

Table XIX

Scoring Symbol Set

Phoneme	Symbol
Lead B	B
Lead D	D
Lead R	R
Lead T	T
Long A	A
Long O	O
Long E	E
Auh	@
Scoring Descriptors	**Symbol**
Located	L
Identified	I
Missed	O
Not Evaluated	-

Table XX

B-Word Group Analysis

Word	Phonemic Rendition	Author 1 B Phoneme	Author 1 All Phonemes	Author 2 B Phoneme	Author 2 All Phonemes
Bay	BA	*-	*I	I-	II
Babble	B-B-	I-I-	I-I-	I-I-	I-I-
Batter	B-T-R	I----	I-I-L	I----	I-L-I
Be	BE	O-	OI	O-	OI
Bench	B---	I---	I---	I---	I---
Bitter	B-T-R	I----	I-I-I	I----	I-O-L
Bite	B-T	I--	I-O	I--	I-I
Boat	BOT	L--	LIO	I--	III
Bought	B@-	I--	II-	L--	LI-
Buy	B-	I-	I-	I-	I-
Butter	B@T-R	I----	ILI-L	I----	IIO-L
Blend	B----	L----	L----	I----	I----
Bright	B--T	I---	I--O	I---	I--I
Bulb	B--B	L--I	L--I	I--I	I--I

Overall Performance

Speaker(s)	B-Phoneme Score Located	B-Phoneme Score Identified	All Phoneme Score Located	All Phoneme Score Identified
Author 1	$\frac{14}{15}$=93.3%	$\frac{11}{15}$=73.3%	$\frac{25}{29}$=86.2%	$\frac{19}{29}$=65.5%
Author 2	$\frac{15}{16}$=93.8%	$\frac{14}{16}$=87.5%	$\frac{27}{30}$=90%	$\frac{23}{30}$=76.7%
Combined	$\frac{29}{31}$=93.5%	$\frac{25}{31}$=80.6%	$\frac{52}{59}$=88.1%	$\frac{42}{59}$=71.2%

*Not scored due to irreversible data preprocessing malfunction.

Table XXI

D-Word Group Analysis

Word	Phonemic Rendition	Author 1 D Phoneme	Author 1 All Phonemes	Author 2 D Phoneme	Author 2 All Phonemes
Day	DA	I-	II	L-	LI
Debt	D--	I--	I--	I--	I--
Debit	D-B-T	I----	I-I-O	I----	I-I-I
Ditto	D-TO	I---	I-OI	I---	I-II
Donut	DO-@-	I----	II-I-	I----	IL-I-
Dug	D@-	I--	II-	I--	II-
Dust	D@-T	I---	II-I	I---	II-I
Drafted	D--T-D	I----I	I--I-I	I----I	I--I-I
Danger	DA--R	I----	II--I	I----	II--L
Dagger	D--R	I---	I--I	I---	I--L
Dread	DR-D	I--I	II-I	I--I	II-I
Dead	D-D	I-O	I-O	I-I	I-I
Dodge	D@-	I--	II-	I--	II-
Dude	D-D	I-I	I-I	L-L	L-L

Overall Performance

Speaker(s)	D-Phoneme Score Located	D-Phoneme Score Identified	All Phoneme Score Located	All Phoneme Score Identified
Author 1	$\frac{17}{18}$=94.4%	$\frac{17}{18}$=94.4%	$\frac{31}{34}$=91.2%	$\frac{31}{34}$=91.2%
Author 2	$\frac{18}{18}$=100%	$\frac{15}{18}$=83.3%	$\frac{34}{34}$=100%	$\frac{28}{34}$=82.4%
Combined	$\frac{35}{36}$=97.2%	$\frac{32}{36}$=88.9%	$\frac{65}{68}$=95.6%	$\frac{59}{69}$=86.8%

Table XXII

R-Word Group Analysis

Word	Phonemic Rendition	Author 1 R Phoneme	Author 1 All Phonemes	Author 2 R Phoneme	Author 2 All Phonemes
Rat	R-T	I--	I-I	I--	I-I
Read	RED	I--	III	I--	III
Ride	R-D	I--	I-I	I--	I-I
Robe	ROB	I--	III	I--	III
Rut	R@T	I--	IIO	I--	III
Rhino	R--O	I---	I--I	I---	I--L
Rather	R--R	I--L	I--L	I--I	I--I
Rear	R-R	I-L	I-L	I-I	I-I
Right	R-T	I--	I-L	I--	I-I
Resist	RE--T	I----	II--I	I----	II--I
Rand	R--D	L---	L--L	I---	I--I
Rover	RO--R	I---I	II--I	I---I	II--I
Rare	R-R	I-I	I-I	I-I	I-I
Rubber	R@B-R	I---I	IIL-I	I---I	III-I

Overall Performance

Speaker(s)	R-Phoneme Score Located	R-Phoneme Score Identified	All Phoneme Score Located	All Phoneme Score Identified
Author 1	$\frac{19}{19}$=100%	$\frac{16}{19}$=84.2%	$\frac{34}{35}$=97.1%	$\frac{28}{35}$=80%
Author 2	$\frac{19}{19}$=100%	$\frac{19}{19}$=100%	$\frac{35}{35}$=100%	$\frac{34}{35}$=97.1%
Combined	$\frac{38}{38}$=100%	$\frac{35}{38}$=92.1%	$\frac{69}{70}$=98.6%	$\frac{62}{70}$=88.6%

Table XXIII

T-Word Group Analysis

Word	Phonemic Rendition	Author 1 T Phoneme	Author 1 All Phonemes	Author 2 T Phoneme	Author 2 All Phonemes
Taker	TA--R	I----	II--I	L----	LI--L
Terminate	TR--A-	I-----	II--O-	I-----	II--I-
Tide	T-D	I--	I-L	I--	I-I
Tight	T-T	I-I	I-I	I-I	I-I
Toad	TO-	I--	II-	L--	LI-
Tore	T--	I--	I--	L--	L--
Tub	T@B	I--	III	I--	III
Tube	T-B	I--	I-L	I--	I-L
Through	*				
Tither	*				
Tribe	T--B	I---	I--I	I---	I--I
Tip	T--	I--	I--	I--	I--
Twist	T--T	I--I	I--I	I--I	I--I
Trade	TRAD	I---	ILII	I---	ILII

Overall Performance

Speaker(s)	T-Phoneme Score Located	T-Phoneme Score Identified	All Phoneme Score Located	All Phoneme Score Identified
Author 1	$\frac{14}{14}$=100%	$\frac{14}{14}$=100%	$\frac{26}{27}$=96.3%	$\frac{23}{27}$=85.1%
Author 2	$\frac{14}{14}$=100%	$\frac{11}{14}$=78.5%	$\frac{27}{27}$=100%	$\frac{21}{27}$=77.7%
Combined	$\frac{28}{28}$=100%	$\frac{25}{28}$=89.3%	$\frac{53}{54}$=98%	$\frac{44}{54}$=81.5%

*Not scored due to irreversible data preprocessing malfunction.

Table XXIV

A-Word Group Analysis

Word	Phonemic Rendition	Author 1 A Phoneme	Author 1 All Phonemes	Author 2 A Phoneme	Author 2 All Phonemes
Hate	-AT	-I-	-II	-I-	-II
Abraham	ABR@--	I-----	IIIL--	I-----	IIIL--
Hay	-A	-I	-I	-I	-I
Range	RA--	-I--	II--	-I--	II--
Same	-A-	-I-	-I-	-I-	-I-
Terminate	T-R--AT	-----I-	I-I--IO	-----I-	I-I--II
Wave	-A-	-I-	-I-	-I-	-I-
Shape	-A-	-I-	-I-	-I-	-I-
Trace	T-A-	--L-	I-L-	--I-	I-I-
Angel	A---	I---	I---	I---	I---
May	-A	-I	-I	-I	-I
Ray	RA	-L	IL	-I	II
Say	-A	-L	-L	-I	-I
Lay	-A	-O	-O	-I	-I

Overall Performance

Speaker(s)	A-Phoneme Score Located	A-Phoneme Score Identified	All Phoneme Score Located	All Phoneme Score Identified
Author 1	$\frac{13}{14}$=92.8%	$\frac{10}{14}$=71.4%	$\frac{22}{24}$=91.6%	$\frac{18}{24}$=75%
Author 2	$\frac{14}{14}$=100%	$\frac{14}{14}$=100%	$\frac{24}{24}$=100%	$\frac{23}{24}$=95.8%
Combined	$\frac{27}{28}$=96.4%	$\frac{24}{28}$=85.7%	$\frac{46}{48}$=95.8%	$\frac{41}{48}$=85.4%

Table XXV

AUH (@)-Word Group Analysis

Word	Phonemic Rendition	Author 1 AUH Phoneme	Author 1 All Phonemes	Author 2 AUH Phoneme	Author 2 All Phonemes
Among	@---	I---	I---	I---	I---
About	@B-T	I---	II-O	I---	II-I
American	@-----	L-----	L-----	L-----	L-----
Topeka	TO-E-@	*----I	*I-I-I	-----I	LL-I-I
Santa	---T@	----I	---II	----I	---II
Mascara	-----@	-----I	-----I	-----I	-----I
Another	@---R	I----	I---L	I----	I---I
Caruso	-@R--O	-I----	-II--I	-I----	-II--I
Appear	@--R	L---	L--L	I---	I--I
Attempt	@T---	L----	LI---	L----	LI---
Accumulate	@-----AT	L-------	L-----LO	I-------	I-----LI
Associate	@-O-E-	I-----	I-I-I-	L-----	L-I-I-
Approximate	@--@---	I--I---	I--I---	I--I---	I--I---
Against	@---T	L----	L---I	I----	I---I

Overall Performance

Speaker(s)	AUH-Phoneme Score Located	AUH-Phoneme Score Identified	All Phoneme Score Located	All Phoneme Score Identified
Author 1	$\frac{15}{15}$=100%	$\frac{10}{15}$=66.6%	$\frac{28}{30}$=93.3%	$\frac{20}{30}$=66.6%
Author 2	$\frac{15}{15}$=100%	$\frac{12}{15}$=80%	$\frac{31}{31}$=100%	$\frac{25}{31}$=80.6%
Combined	$\frac{30}{30}$=100%	$\frac{22}{30}$=73.3%	$\frac{59}{61}$=96.7%	$\frac{45}{61}$=73.7%

*Not processed due to preprocessing error.

Table XXVI

E-Word Group Analysis

Word	Phonemic Rendition	Author 1 E Phoneme	Author 1 All Phonemes	Author 2 E Phoneme	Author 2 All Phonemes
Leave	-E-	-I-	-I-	-I-	-I-
Each	E-	I-	I-	I-	I-
Me	-E	-I	-I	-I	-I
See	-E	-I	-I	-I	-I
Even	E--	I--	I--	I--	I--
Leach	-E-	-I-	-I-	-I-	-I-
Beat	BET	-I-	OII	-I-	OII
Meet	-ET	-I-	-II	-I-	-II
Sleep	--E-	--I-	--I-	--I-	--I-
Valley	---E	---L	---L	---I	---I
Reek	RE-	-I-	II-	-I-	II-
Key	-E	-I	-I	-I	-I
Egress	E-R-	I---	I-I-	I---	I-I-
Ego	E-O	I--	I-I	I--	I-I

Overall Performance

Speaker(s)	E-Phoneme Score Located	E-Phoneme Score Identified	All Phoneme Score Located	All Phoneme Score Identified
Author 1	$\frac{14}{14}$=100%	$\frac{13}{14}$=92.8%	$\frac{19}{20}$=95%	$\frac{18}{20}$=90%
Author 2	$\frac{14}{14}$=100%	$\frac{14}{14}$=100%	$\frac{19}{20}$=95%	$\frac{19}{20}$=95%
Combined	$\frac{28}{28}$=100%	$\frac{27}{28}$=96.4%	$\frac{38}{40}$=95%	$\frac{37}{40}$=92.5%

Table XXVII

O-Word Group Analysis

Word	Phonemic Rendition	Author 1 O Phoneme	Author 1 All Phonemes	Author 2 O Phoneme	Author 2 All Phonemes
Go	-O	-I	-I	-I	-I
So	-O	-I	-I	-I	-I
Blow	B-O	--I	L-I	--I	I-I
Obey	OBA	I--	IIL	I--	III
Omit	O--	I--	I--	I--	I--
Over	O-R	I--	I-I	I--	I-I
Note	-OT	-I-	-II	-I-	-II
Those	-O-	-I-	-I-	-I-	-I-
Pose	-O-	-I-	-I-	-I-	-I-
Rose	-O-	-I-	II-	-I-	II-
Nose	-O-	-I-	-I-	-I-	-I-
Most	-O-T	-I--	-I-I	-I--	-I-I
Both	BO-	-I-	LI-	-I-	II-
No	-O	-I	-I	-I	-I

Overall Performance

Speaker(s)	O-Phoneme Score Located	O-Phoneme Score Identified	All Phoneme Score Located	All Phoneme Score Identified
Author 1	$\frac{14}{14}$=100%	$\frac{14}{14}$=100%	$\frac{22}{22}$=100%	$\frac{19}{22}$=86%
Author 2	$\frac{14}{14}$=100%	$\frac{14}{14}$=100%	$\frac{22}{22}$=100%	$\frac{22}{22}$=100%
Combined	$\frac{28}{28}$=100%	$\frac{28}{28}$=100%	$\frac{44}{44}$=100%	$\frac{41}{44}$=93.1%

Table XXVIII

Verification Sentence Analysis

"Abraham drafted a note."

Word	Phonemic Rendition	Author 1 Discrete	Author 1 Continuous	Author 2 Discrete	Author 2 Continuous
Abraham	ABR@--	ILIL--	ILIL--	ILIL--	IOIL--
Drafted	D--T-D	I--I-I	I--O-I	I--I-I	I--L-I
A	@ or A	I	L	I	I
Note	-OT	-IO	-IO	-II	-II

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Author 1	$\frac{9}{10}$=90%	$\frac{7}{10}$=70%	$\frac{8}{10}$=80%	$\frac{5}{10}$=50%
Author 2	$\frac{10}{10}$=100%	$\frac{8}{10}$=80%	$\frac{9}{10}$=90%	$\frac{7}{10}$=70%
Combined	$\frac{19}{20}$=95%	$\frac{15}{20}$=75%	$\frac{17}{20}$=85%	$\frac{12}{20}$=60%

Table XXIX

Verification Sentence Analysis

"See me wave at my associate."

Word	Phonemic Rendition	Author 1 Discrete	Author 1 Continuous	Author 2 Discrete	Author 2 Continuous
See	-E	-I	-I	-I	-I
Me	-E	-I	-I	-I	-I
Wave	-A-	-I-	-I-	-I-	-I-
At	-T	-O	-O	-I	-L
My	--	--	--	--	--
Associate	@-O-E-	L-I-I-	I-I-I-	I-I-L-	L-I-I-

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Author 1	$\frac{6}{7}$=85.7%	$\frac{5}{7}$=71.4%	$\frac{6}{7}$=85.7%	$\frac{6}{7}$=85.7%
Author 2	$\frac{7}{7}$=100%	$\frac{6}{7}$=85.7%	$\frac{7}{7}$=100%	$\frac{5}{7}$=71.4%
Combined	$\frac{13}{14}$=92.8%	$\frac{11}{14}$=78.5%	$\frac{13}{14}$=92.8%	$\frac{11}{14}$=78.5%

Table XXX

Verification Sentence Analysis

"A boy got out the back gate."

Word	Phonemic Rendition	Author 1 Discrete	Author 1 Continuous	Author 2 Discrete	Author 2 Continuous
A	@ or A	I	I	I	I
Boy	B-	L-	I-	I-	I-
Got	-@T	-IO	-IO	-II	-IL
Out	-T	-O	-O	-I	-O
The	-E or -@	-I	-I	-I	-L
Back	B--	I--	I--	I--	I--
Gate	-AT	-IO	-IL	-II	-II

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Author 1	$\frac{6}{9}$=66.6%	$\frac{5}{9}$=55.5%	$\frac{7}{9}$=77.7%	$\frac{6}{9}$=66.6%
Author 2	$\frac{9}{9}$=100%	$\frac{9}{9}$=100%	$\frac{8}{9}$=88.8%	$\frac{6}{9}$=66.6%
Combined	$\frac{15}{18}$=83.3%	$\frac{14}{18}$=77.7%	$\frac{15}{18}$=83.3%	$\frac{12}{18}$=66.6%

Table XXXI

Verification Sentence Analysis

"Joe was seen around the airplane."

		Author 1		Author 2	
Word	Phonemic Rendition	Discrete	Continuous	Discrete	Continuous
Joe	-O	-I	-I	-I	-I
Was	-@-	-I-	-L-	-I-	-I-
Seen	-E-	-I-	-L-	-I-	-I-
Around	@R--D	II--I	LO--I	II--O	II--L
The	-E or -@	-I	-I	-I	-I
Airplane	-R----	-L----	-L----	-L----	-I----

Overall Performance

	Discrete Score		Continuous Score	
Speaker(s)	Located	Identified	Located	Identified
Author 1	$\frac{8}{8}$=100%	$\frac{7}{8}$=87.5%	$\frac{7}{8}$=87.5%	$\frac{3}{8}$=37.5%
Author 2	$\frac{7}{8}$=87.5%	$\frac{6}{8}$=75%	$\frac{8}{8}$=100%	$\frac{7}{8}$=87.5%
Combined	$\frac{15}{16}$=93.7%	$\frac{13}{16}$=81.2%	$\frac{15}{16}$=93.7%	$\frac{10}{16}$=62.5%

Table XXXII

Test Sentence Analysis

"Abraham drafted a note."

Word	Phonemic Rendition	Speaker 1* Discrete	Speaker 1* Continuous
Abraham	ABR@--	OILI--	OIOI--
Drafted	D--T-D	I--L-I	O--L-I
A	@ or A	I	I
Note	-OT	-IO	-IO

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Speaker 1	$\frac{8}{10}$=80%	$\frac{6}{10}$=60%	$\frac{6}{10}$=60%	$\frac{5}{10}$=50%

*Speaker 2 and Speaker 3 not scored due to irreversible data preprocessing malfunction.

AD-A064 058 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 9/2
COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH.(U)
NOV 78 G L BROCK, E S KOLESAR
UNCLASSIFIED AFIT/GE/EE/78D-20 NL

3 OF 3
AD
A064058

END
DATE
FILMED
3-79
DDC

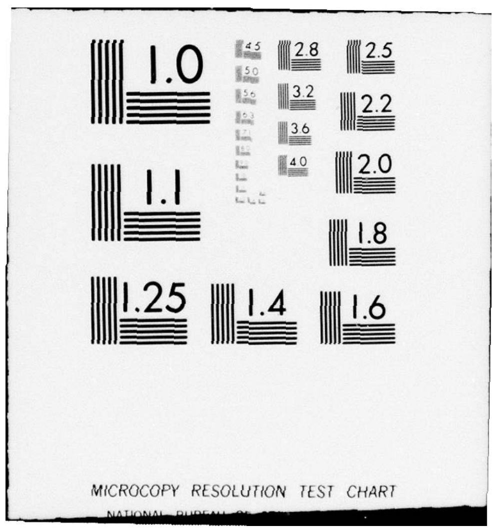

1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART

Table XXXIII

Test Sentence Analysis

"No note to terminate the leave of the American called Caruso was drafted this day."

	Phonemic	Speaker 1		Speaker 2		Speaker 3	
Word	Rendition	Discrete	Continuous	Discrete	Continuous	Discrete	Continuous
No	-O	-L	-O	-I	-I	-I	-I
Note	-OT	-LO	-OO	-II	-LO	-IL	-LO
To	T-	O-	O-	I-	I-	I-	O-
Terminate	TR--A-	LO--O-	OI--O-	IO--O-	IO--O-	IO--O-	IO--O-
The	-E or -@	-I	-I	-L	-I	-I	-I
Leave	-E-	-I-	-O-	-L-	-L-	-O-	-O-
Of	@-	I-	I-	*	I-	I-	I-
The	-E or -@	-I	-I	-I	-I	-I	-I
American	@-----	I-----	I-----	L-----	I-----	I-----	I-----
Called	---D	---I	---L	---I	---I	---I	---O
Caruso	-@R--O	-IO--I	-II--I	-IO--I	-II--I	-II--L	-IO--I
Was	-@-	-I-	-I-	-I-	-L-	-I-	-I-
Drafted	D--T-D	I--O-I	I--L-I	L--L-L	I--O-O	L--O-I	I--O-L

(continued)

Table XXXIII--continued

Test Sentence Analysis

"No note to terminate the leave of the American called Caruso was drafted this day."

Word	Phonemic Rendition	Speaker 1 Discrete	Speaker 1 Continuous	Speaker 2 Discrete	Speaker 2 Continuous	Speaker 3 Discrete	Speaker 3 Continuous
This	--	--	--	--	--	--	--
Day	DA	OO	IO	IO	OO	OO	OO

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Speaker 1	$\frac{14}{22}$=63.6%	$\frac{11}{22}$=50%	$\frac{14}{22}$=63.6%	$\frac{12}{22}$=54.5%
Speaker 2	$\frac{17}{21}$=81%	$\frac{11}{21}$=52.4%	$\frac{15}{22}$=68.2%	$\frac{12}{22}$=54.5%
Speaker 3	$\frac{16}{22}$=72.7%	$\frac{13}{22}$=59.1%	$\frac{12}{22}$=54.5%	$\frac{10}{22}$=45.5%
Combined	$\frac{47}{65}$=72.3%	$\frac{35}{65}$=53.8%	$\frac{41}{66}$=62.1%	$\frac{34}{66}$=51.5%

*Not scored due to irreversible data preprocessing malfunction.

Table XXXIV

Test Sentence Analysis

"The bright bulb formed a ray that made a trace of the rubber rat."

Word	Phonemic Rendition	Speaker 1* Discrete	Speaker 1* Continuous	Speaker 3 Discrete	Speaker 3 Continuous
The	-A or -@	-I	-L	-I	-I
Bright	B--T	I--L	L--O	I--O	I--O
Bulb	B--B	I--I	I--I	I--I	I--I
Formed	---D	---I	---O	---I	---O
A	@ or A	I	I	I	I
Ray	RA	LO	OO	IO	IO
That	--	--	--	--	--
Made	-AD-	-OI	-OO	-OI	-OI
A	@ or A	I	I	I	L
Trace	T-A-	I-I-	I-O-	I-L-	I-L-
Of	@-	I-	I-	L-	I-
The	-E or -@	-I	-I	-I	-I
Rubber	R@B-R	LII-L	OII-O	LII-I	OII-L
Rat	R-T	O-O	O-O	L-L	I-O

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Speaker 1	$\frac{18}{22}$=81.8%	$\frac{14}{22}$=63.6%	$\frac{11}{22}$=50%	$\frac{9}{22}$=40.9%
Speaker 3	$\frac{19}{22}$=86.4%	$\frac{14}{22}$=63.6%	$\frac{16}{22}$=72.7%	$\frac{13}{22}$=59.1%
Combined	$\frac{37}{44}$=84.1%	$\frac{28}{44}$=63.6%	$\frac{27}{44}$=61.4%	$\frac{22}{44}$=50%

*Speaker 2 not scored due to irreversible data preprocessing malfunction.

Table XXXV

Test Sentence Analysis

"From the boat docked in the bay, we saw the rhino, leech, and toad as they lay dead along the tide."

Word	Phonemic Rendition	Speaker 1		Speaker 2*	
		Discrete	Continuous	Discrete	Continuous
From	--@-	--O-	*	--I-	--I-
The	-A or -@	-I	*	-I	-I
Boat	BOT	IIO	*	III	ILO
Docked	D@-D	II-I	*	IL-I	II-I
In	--	--	*	--	--
The	-A or -@	-I	*	-I	-L
Bay	BA	IO	*	IO	IO
We	-E	-L	*	-O	-O
Saw	-@	-I	*	-I	-I
The	-A or -@	-I	*	-I	-I
Rhino	R--O	I--I	*	O--I	O--I
Leech	-E-	-L-	*	-O-	-O-
And	--D	--I	*	--I	--O
Toad	TO-	II-	*	II-	II-
As	--	--	*	--	--
They	-A	-O	*	-O	-O
Lay	-A	-O	*	-O	-O
Dead	D-D	O-I	*	O-L	O-I
Along	@--	I--	*	I--	I--

(continued)

Table XXXV--continued

Test Sentence Analysis

"From the boat docked in the bay, we saw the rhino, leech, and toad as they lay dead along the tide."

Word	Phonemic Rendition	Speaker 1 Discrete	Speaker 1 Continuous	Speaker 2* Discrete	Speaker 2* Continuous
The	-A or -@	-I	*	-I	-I
Tide	T-D	O-I	*	I-I	L-I

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Speaker 1	$\frac{21}{28}$=75%	$\frac{19}{28}$=67.9%	*	*
Speaker 2	$\frac{21}{28}$=75%	$\frac{19}{28}$=67.9%	$\frac{19}{28}$=67.9%	$\frac{16}{28}$=57.1%
Combined	$\frac{42}{56}$=75%	$\frac{38}{56}$=67.8%	$\frac{19}{28}$=67.9%	$\frac{16}{28}$=57.1%

*Speaker 3's test sentence and Speaker 1's continuous speech was not processed due to an irreversible preprocessing malfunction.

Table XXXVI

Test Sentence Analysis

"Before the trip, the rabbit rested along the open field of the rancher."

Word	Phonemic Rendition	Speaker 1 Discrete	Speaker 1 Continuous	Speaker 2* Discrete	Speaker 2* Continuous
Before	BE-R	II-I	II-I	II-I	IL-O
The	-E or -@	-I	-I	-I	-I
Trip	T---	I---	I---	I---	I---
The	-E or -@	-I	-I	-I	-I
Rabbit	R-B-T	L-I-O	I-I-O	L-I-L	I-I-I
Rested	R-TD	I-OI	I-OO	O-II	I-OI
Along	@--	I--	I--	I--	I--
The	-E or -@	-I	-I	-I	-I
Open	O--	I--	I--	I--	L--
Field	-E-D	-I-I	-O-L	-O-I	-O-I
Of	@-	I-	I-	I-	I-
The	-E or -@	-I	-I	-I	-I
Rancher	R---R	L---L	I---I	I---I	L---L

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Speaker 1	$\frac{19}{21}$=90.5%	$\frac{16}{21}$=76.2%	$\frac{17}{21}$=81%	$\frac{16}{21}$=76.2%
Speaker 2	$\frac{19}{21}$=90.5%	$\frac{17}{21}$=81%	$\frac{18}{21}$=85.7%	$\frac{14}{21}$=66.7%
Combined	$\frac{38}{42}$=90.5%	$\frac{33}{42}$=78.6%	$\frac{35}{42}$=83.3%	$\frac{30}{42}$=71.4%

*Speaker 3 not scored due to irreversible data preprocessing malfunction.

Table XXXVII

Test Sentence Analysis

"Does Dennis teach reading or does Dennis teach driving?"

Word	Phonemic Rendition	Speaker 2* Discrete	Speaker 2* Continuous	Speaker 3 Discrete	Speaker 3 Continuous
Does	D@-	II-	*	II-	LI-
Dennis	D---	O---	*	I---	L---
Teach	TE-	IL-	*	IL-	LO-
Reading	RED-	IIO-	*	III-	ILO-
Or	-R	-L	*	-I	-I
Does	D@-	LI-	*	II-	LL-
Dennis	D---	O---	*	O---	L---
Teach	TE-	II-	*	II-	LI-
Driving	D----	I----	*	I----	L----

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Speaker 2	$\frac{12}{15}$=80%	$\frac{9}{15}$=60%	*	*
Speaker 3	$\frac{14}{15}$=93.3%	$\frac{13}{15}$=86.6%	$\frac{13}{15}$=86.6%	$\frac{4}{15}$=26.6%
Combined	$\frac{26}{30}$=86.6%	$\frac{22}{30}$=73.3%	$\frac{13}{15}$=86.6%	$\frac{4}{15}$=26.6%

*Speaker 1's test sentence and Speaker 2's continuous speech was not processed due to an irreversible preprocessing malfunction.

Table XXXVIII

Test Sentence Analysis

"Joe was seen around the airplane."

Word	Phonemic Rendition	Speaker 2* Discrete	Speaker 2* Continuous
Joe	-O	-I	*
Was	-@-	-I-	*
Seen	-E-	-L-	*
Around	@R--D	II--I	*
The	-E or -@	-I	*
Airplane	-R----	-L----	*

Overall Performance

Speaker(s)	Discrete Score Located	Discrete Score Identified	Continuous Score Located	Continuous Score Identified
Speaker 2	$\frac{8}{8}$=100%	$\frac{6}{8}$=75%	*	*

*Speaker 1 and Speaker 3's test sentence and Speaker 2's continuous speech not scored due to irreversible preprocessing malfunction.

Table XXXIX

Test Sentence Analysis

"Take a closer look at Eastman Kodak's bubbling reagents for photo-resist stripping."

	Phonemic	Speaker 1		Speaker 2		Speaker 3	
Word	Rendition	Discrete	Continuous	Discrete	Continuous	Discrete	Continuous
Take	TA-	OO-	OO-	IO-	OO-	IO-	OL-
A	@ or A	I	I	L	I	I	I
Closer	--OR	--LL	--LO	--LI	--II	--II	--LO
Look	---	---	---	---	---	---	---
At	-T	-O	-L	-O	-I	-L	-I
Eastman	E-T---	I-O---	O-O---	O-I---	I-O---	I-I---	O-O---
Kodak's	-OD---	-IO---	-IO---	-II---	-LO---	-LI---	-IO---
Bubbling	B@B--	III--	III--	III--	III--	III--	III--
Reagents	REA---	III---	LOO---	IOO---	ILO---	OOL---	LIL---
For	-R	-L	-O	-I	-L	-I	-O
Photo-	-OTO	-LLO	-LLL	-LIL	-IOI	-IOL	-LOL
Resist	RE-T	II-O	OL-I	IL-I	OL-L	OO-I	OO-I
Stripping	-R---	-L---	-O---	-L---	-I---	-O---	-L---

(continued)

Table XXXIX--continued

Test Sentence Analysis

"Take a closer look at Eastman Kodak's bubbling reagents for photo-resist stripping."

Overall Performance

Speaker(s)	Discrete Score		Continuous Score	
	Located	Identified	Located	Identified
Speaker 1	$\frac{17}{24}$=70.8%	$\frac{11}{24}$=45.8%	$\frac{13}{24}$=54.2%	$\frac{6}{24}$=25%
Speaker 2	$\frac{19}{24}$=79.2%	$\frac{13}{24}$=54.2%	$\frac{17}{24}$=70.8%	$\frac{12}{24}$=50%
Speaker 3	$\frac{17}{24}$=70.8%	$\frac{13}{24}$=54.2%	$\frac{15}{24}$=62.5%	$\frac{8}{24}$=33.3%
Combined	$\frac{53}{72}$=72.2%	$\frac{37}{72}$=51.4%	$\frac{45}{72}$=62.5%	$\frac{26}{72}$=36.1%

Table XL

Test Sentence Analysis

"Each person at Beckman sees his responsibility aimed toward fabricating better resistors, displays and drugs."

Word	Phonemic Rendition	Speaker 2* Discrete	Speaker 2* Continuous	Speaker 3 Discrete	Speaker 3 Continuous
Each	E-	L-	O-	I-	O-
Person	----	----	----	----	----
At	-T	-I	-O	-O	-O
Beckman	B-----	I-----	I-----	I-----	I-----
Sees	-E-	-O-	-O-	-O-	-O-
His	--	--	--	--	--
Responsibility	R-----B---	O-----I---	L-----I---	I-----I---	O-----I---
Aimed	A-D	O-I	O-I	O-I	O-I
Toward	T---D	I---I	I---I	I---I	L---L
Fabricating	--B-@-AT-	--I-I-OO-	--I-I-OO-	--I-I-OO-	--I-I-OO-
Better	B-TR	I-OI	I-OI	I-OI	I-IO
Resistors	RE-T--	LO-I--	IO-O--	II-I--	OL-I--
Displays	D---A-	L---O-	L---L-	I---L-	O---O-
And	--D	--I	--I	--I	--L
Drugs	D-@-	O-I-	L-I-	L-I-	I-I-

(continued)

Table XL--continued

Test Sentence Analysis

"Each person at Beckman sees his responsibility aimed toward fabricating better resistors, displays and drugs."

Overall Performance

Speaker(s)	Discrete Score: Located	Discrete Score: Identified	Continuous Score: Located	Continuous Score: Identified
Speaker 2	$\frac{16}{25}$=64%	$\frac{13}{25}$=52%	$\frac{16}{25}$=64%	$\frac{12}{25}$=48%
Speaker 3	$\frac{19}{25}$=76%	$\frac{17}{25}$=68%	$\frac{14}{25}$=56%	$\frac{10}{25}$=40%
Combined	$\frac{35}{50}$=70%	$\frac{30}{50}$=60%	$\frac{30}{50}$=60%	$\frac{22}{50}$=44%

*Speaker 1 not scored due to irreversible data preprocessing malfunction.

APPENDIX D

CORRELATION DEPENDENCY ON PROTOTYPE PHONEME LENGTH

D. Correlation Dependency on Prototype Phoneme Length

This appendix contains a mathematical analysis of the dependency of a correlation's magnitude when a column normalized prototype phoneme is correlated with a column plus unit normalized prototype phoneme. This analysis demonstrates that the maximum correlation of these two "processed" arrays is a function of the square root of the length of a particular prototype phoneme. In addition, the analysis shows that the maximum correlation magnitude can be limited to unity for different length phonemes.

Definition of Terms:

1. Let a prototype phoneme consist of a matrix

$$\tilde{p} = (\overline{P}_j \ldots .)_j$$

where

$$\overline{P}_j = \begin{bmatrix} \vdots \\ p_i \\ \vdots \end{bmatrix}_i$$

2. Energy of $\tilde{P}$ = E $(\tilde{P}) = \sum^{i} \sum^{j} (p_{ij})^2$

3. $||\overline{P}_j|| = [\sum^{i} (p_i)^2]^{1/2}$

4. Energy of $\overline{P}_j = E(\overline{P}_j) = \sum^{i} (p_i)^2$

5. Unit Vector $\hat{P}_j = \begin{bmatrix} \vdots \\ p'_i \\ \vdots \end{bmatrix}_i$

where

$$p'_i = p_i/[\sum^{i} (p_i)^2]^{1/2}$$

The Column Normalization of $\tilde{P}$ is:

$$\hat{P}_j = \bar{P}_j/|| \bar{P}_j ||$$

The Column plus Unit Normalization of $\tilde{P}$ is:

$$\overset{*}{P} = (\bar{K}_j \ldots)_j$$

where

$$\bar{K}_j = \hat{P}_j/||\tilde{\hat{P}}||$$

and where

$$||\tilde{\hat{P}}|| = [\sum^{j} |\hat{P}_j|^2]^{1/2}$$

Since $||\hat{P}_j|| = 1$, $||\hat{P}_j||^2 = 1$

so $\sum^{j} 1 = j$

Therefore, $||\tilde{\hat{P}}|| = \sqrt{j}$

and $\overset{*}{P} = (\frac{1}{j})\ (\tilde{\hat{P}})$

Assuming that somewhere in the sentence sample there is an exact replica of the prototype phoneme $\tilde{P}$, this means that the correlation computation will be performed between $\overset{*}{P}$ and $\tilde{\hat{P}}$.

Correlation implies:

$$\{\overset{*}{P} \cdot \tilde{\hat{P}}\} = [\sum^{j} (\bar{K}_j \cdot \hat{P}_j)]$$

Maximum correlation occurs when two identical elements are correlated.

Maximum Correlation:

$$\{\dot{P} \cdot \tilde{\hat{P}}\} = [\sum^{j} (\hat{P}_j/\sqrt{j}\,) \cdot (\hat{P}_j)]$$

$$= [\sum^{j} (|\hat{P}_j|^2/\sqrt{j})]$$

$$= [\sum^{j} (1.0/\sqrt{j})]$$

$$= [\sum^{j} (\sqrt{j}\ /j)]$$

$= \sqrt{j}$ which is the square root of the length of the prototype phoneme.

In order to insure that the correlation amplitudes be limited to a maximum value of unity, the correlation values for each phoneme must be divided by the square root of the length of the prototype phoneme.

APPENDIX E

SPECTROGRAM OVERPRINT SCHEME

E. Spectrogram Overprint Scheme

This appendix contains an explanation of the spectrogram overprint scheme used in this research. The two programs, OCTAVE1 and OCTAVE2, produced spectrograms of the 16 component frequency vectors according to the following procedure. Each component had a threshold to select the proper number of overprints; a round-up procedure was used to form integer values and these integer values correspond to the overprint level of darkness shown in Table XLI.

Table XLI

Spectrogram Overprint Scheme

Channel Component Magnitude	Level of Darkness	Number of Overprints	Spectrogram Character Components	Spectrogram Character
0	0	0	blank	blank
1	1	0	blank	blank
2	2	1	+	+
3	3	1	X	X
4	4	2	X,-	[illegible]
5	5	2	X,+	[illegible]
6	6	2	X,O	[illegible]
7	7	3	X,O,-	[illegible]
8	8	4	X,O,-,+	[illegible]
9	9	5	X,O,+,#,*	[illegible]

APPENDIX F

GLOSSARY

F. Glossary of Technical Terms

1. Aliasing: The term "aliasing" refers to the fact that high-frequency components of a time function can impersonate low frequencies if the sampling rate is too low.

2. Allophone: The variant forms of a phoneme as conditioned by position or adjoining sounds.

3. Autocorrelation: The discrete convolution of the function x(n) with x(-n). Compute X(k), the DFT of x(n), and multiply by X*(k). The inverse DFT of $X(k)X^*(k) = X(k)^2$ corresponds to the circular convolution of x(n) with x(-n), i.e., a circular correlation.

4. Crosscorrelation: The discrete convolution of the function x(n) with the function y(-n). Note above and that the DFT of y(-n) is Y*(k).

5. Dipthong: A combination of two vowels in the same syllable, in which the speaker glides continuously from one vowel to another.

6. Discrete Fourier Transform (DFT): The Discrete Fourier Transform (DFT) is defined as

$$F(k) = \sum_{n=0}^{(N-1)} f(nT)e^{-j(\frac{2\pi}{N})nk}$$

where f(nT) corresponds to equally spaced samples of an analog time function f(t). Assuming that the sampling has been done at a rate equal to or higher than the NyQuist rate ($2f_m$, where f_m is the highest frequency in the analog time function), then the magnitude of the k^{th} spectral point $|F(k)|$ corresponds to the magnitude that would be obtained at a time t = (N-1)T if the sample of the analog function f(t) were processed by an analog filter with a frequency response H(w) given by:

$$H(w) = \frac{\sin\frac{NT}{2}(w - \frac{2\pi k}{NT})}{(w - \frac{2k}{NT})}$$

7. End-Effect: The effect on computational results caused by the periodicity imposed on a function by use of the DFT.

8. Fricatives: Sounds produced by partial constriction along the vocal tract which results in turbulence. The sounds can be further subdivided into voiced and unvoiced categories. The voiceless fricatives are produced as a result of frictional modulation. The voiced fricatives combine frictional with vocal cord and cavity modulation.

9. Leakage: The term "leakage" refers to the discrepancy between the continuous and discrete Fourier transforms caused by the required time domain truncation.

10. Morpheme: Any of the minimum meaningful elements in a language, not further divisible into smaller meaningful elements, usually recurring in various contexts with relatively constant meaning, such as a word.

11. Nasals: Sounds that are produced by allowing the air to flow through the nasal cavities. Coupling the nasal cavities to the resonance system of the vocal tract results in nasalized vowels. If the air flow is restricted to only flowing through the nasal cavities, nasal consonants are produced.

12. Phone: An individual speech sound.

13. Phoneme: The smallest distinctive group or class of phones in a language. In a very general sense, the phonemes that make up a speech sound can be compared to the letters that make up a written word.

14. Pitch: The pitch of a sound with a periodic wave form--i.e., a voiced sound--is determined by its fundamental frequency, or rate of repetition of the cycles of air pressure.

15. Plosives: Sounds that are produced by a sudden release of built up air pressure. The sounds can be further distinguished by the presence of absence of voicing. A voiceless stop occurs when the stop is combined with fricative modulation. A voiced stop occurs when vocal cord modulation is combined with stop and fricative modulation.

16. Template: The phoneme employed for matching in the correlation program.

17. Vowels: Sounds whose source of excitation is the glottis. During vowel production, the vocal tract is relatively open and the air flows over the center of the tongue, causing a minimum of turbulence. The phonetic value of the vowel is determined by the resonances of the vocal tract, which are in turn determined by the shape and position of the tongue and lips.

VITA

Gary Lee Brock was born on 5 July 1951 in Denver, Colorado. He graduated from Central High School in Aurora, Colorado and attended the University of Colorado in Boulder, Colorado, from which he received the degree of Bachelor of Electrical Engineering in December, 1973. Upon graduation, he received a commission in the USAF through the ROTC program. In February, 1974, he was initially assigned to the Foreign Technology Division at Wright-Patterson AFB, Ohio. During August, 1974, he was transferred to the Aeronautical Systems Division, where he worked in the Electro-Magnetic Test and Checkout Branch of the Directorate of Equipment Engineering, until he entered the Air Force Institute of Technology in June, 1977.

Permanent Address: 1065 Fulton Street
Aurora, Colorado 80010

VITA

Edward S. Kolesar, Jr. was born on 24 June 1950 in Canton, Ohio. He graduated from Central Catholic High School in 1968, entered the University of Akron that same year, and graduated in June, 1973, with a Bachelor's Degree in Electrical Engineering. He entered the Air Force September, 1973, and served as a scientific and technical intelligence analyst with the Directorate of Intelligence, Electronic Systems Division, Hanscom AFB, Massachusetts through 1977. Upon completion of a Master of Business Administration degree and SOS in residence, he entered the Air Force Institute of Technology in June, 1977.

Permanent Address: 1145 Clarendon Ave. S.W.
Canton, Ohio 44710

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER GE/EE/78-D-20	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH		5. TYPE OF REPORT & PERIOD COVERED MS thesis
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gary L. Brock Captain Edward S. Kolesar Jr. Captain		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology(AFIT/EN) Wright-Patterson AFB OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AMRL/BB Wright-Patterson AFB OH 45433		12. REPORT DATE November, 1978
		13. NUMBER OF PAGES 202
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for pubic release; IAW AFR 190-17
Joseph S. Hipps, Major, USAF
Director of Information
1-19-79

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Identification
Phonemes
Continuous Speech

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An approach to computer recognition of continuous speech through phoneme identification is presented. Speech data is digitally processed through correlation, recognition, and location programs. Methods of phoneme prototype production were explored using multiple speaker discrete and averaged prototypes. The identification process presents a rank ordering of probable phonemic occurances at each time period. The method is used to attain an average recognition rate of 81% for discrete speech and 61% for continuous speech spoken by dissimilar speakers.