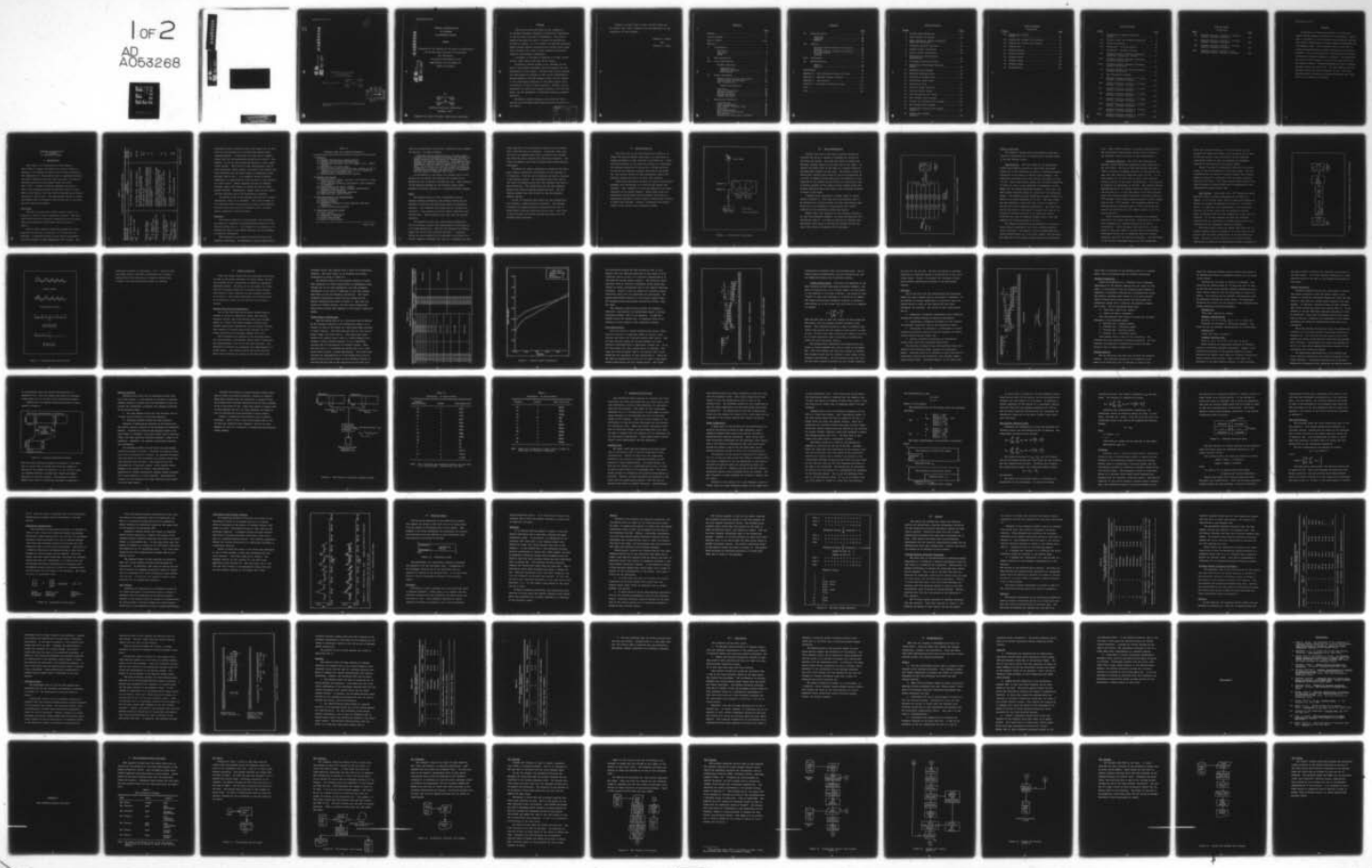
AD-A053 268 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 17/2
COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH.(U)
NOV 77 M F GUYOTE, P L SISSON

UNCLASSIFIED AFIT/GE/EE/77D-18 NL

1 OF 2
AD
A053268



AFIT/GE/EE/77D-18

(1)

COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH

Michael F. Cuyotte
Capt USAF
and
Patrick L. Sisson
Capt USAF

AFIT/GE/EE/77D-18

Approved for public release; distribution unlimited

AFIT/GE/EE/77D-18

COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH

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

Michael F. Guyote
Capt USAF
and
Patrick L. Sisson
Capt USAF

Graduate Electrical Engineering

November, 1977

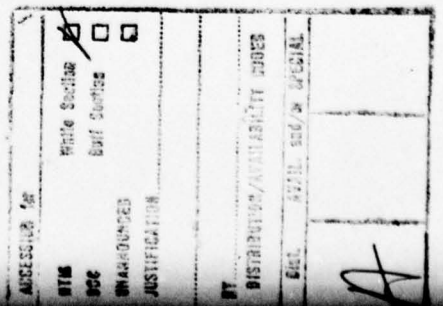
Preface

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. 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 D to help clarify certain terms used in the body of the thesis.

We extend a special thanks to Dr. Kabrisky for his advice and guidance throughout both the research for 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, 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 the thesis.



Finally, we would like to give a special thanks to our typist, Mrs. Dee D. Babiarz, for her dedication to the completion of this document.

Michael F. Guyote

and

Patrick L. Sisson

Contents

Contents

List of Figures

List of Figures

(Continued)

List of Tables

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, then digitally sampled, fast Fourier transformed and logarithmically compressed into 16 frequency bands. This processed data is then used in running crosscorrelation, phoneme recognition and location programs. Once the phonemes are located and/or recognized, a ranking of possible phonemes is selected. This procedure was used on four different speakers using both continuous and discrete speech. Phoneme averaging was used to improve previous results by nearly 28%. The rank ordering and new decision scheme improved recognition by 47%. The final improved phoneme location and recognition rates were 76.9% and 72.0% on dissimilar speakers.

COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH

I. Introduction

This paper is a continuation of work begun by Major Ralph W. Neyman and improved by Captain William R. Hensley on the problem of machine based speech recognition. The advantages of a free-speech input to computing machinery are widely recognized and have been the basis of extensive research projects by many groups around the world (Ref 1:319). Present literature expresses the opinion that a true continuous speech recognition system is still years in the future, and even then the systems may be highly restrictive (Ref 10:531). The preliminary results of Neyman and Hensley seem to contradict this belief and are the basis for this continued research.

Motivation

The past few years have brought various degrees of success in computer speech recognition systems. Some systems which are quite accurately recognizing limited vocabularies are presently on the market and are listed in Table I.

Some of these limited recognition systems have found practical use and one is available in kit form for the hobbiest. A practical system is that used by paraplegics for voice control of their wheelchairs (Ref 11:346). The

Table I

Performing 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, abritrary 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	90
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

(From Ref 12:34)

wheelchair device is only an eight word system, but it does point to the potential of an unrestricted speech understanding machine. In this case the only mode of communication left for the man/machine system was the voice. This, however, is not the only reason for desiring a voice recognition system. "Even with the best communication aids of high technology, speech remains unrivaled as the fastest and most convenient way for human beings to communicate interactively (Ref 13:40)." As an example, speech transfers information at approximately twice the rate of that possible by a good typist. Speech surpasses written or keyboard oriented inputs with respect to speed and ease of information transfer. Consequently, speech input is the natural and most desirable way of the man/machine interface.

To mention a few of the possible applications of speech recognition, Table II is included. This is by no means an all inclusive list but does point out some of the possible military tasks which could be automated were a reliable speech recognition system available.

Objective

The overall objective of this project was to improve and change as necessary the Neyman/Hensley recognition and location scheme (Ref 5). The essence of the program, as it existed and after it was highly modified, was the analysis of spectral information.

This approach to the speech recognition problem has inherent limitations. To completely recognize speech more

Table II

Military Tasks for Possible Automation

1) Security
   - 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) Command and Control
   - 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) Data Transmission and Communication
   - 3.1 Speech synthesis
   - 3.2 Vocoder systems
   - 3.3 Bandwidth reduction or, more general, bit-rate reduction
   - 3.4 Ciphering/coding/scrambling

4) Processing Distorted Speech
   - 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)

than just manipulation of spectral information will probably be required. 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 4:163).

In realization of the above, the overall goal of this project was to improve the location and recognition of phonemes by use of spectral information only. Rank ordering of the possible phonemes by the decision scheme makes possible the future addition of a linguistic/syntax program.

Scope

The desired result of this investigation was an improved phoneme recognition and location scheme for the analysis of discrete and continuous speech by dissimilar speakers. Four speakers were chosen and all recorded two sentences. Each sentence was first spoken discretely then continuously. These sentences were then used for program analysis.

To establish a base line, the original program was used to generate location and identification percentages of a 15 class phoneme set. This set was generated by speaker number one on his first discrete sentence. A learning scheme was then introduced to investigate the effects of a "fluid" template (phoneme) set; that is, a template set that

would represent not one particular speaker but the average from a selected group of speakers. To evaluate this idea, the first two speakers were used to generate the averaged set from both their discrete and continuous sentences. The other two speakers were also evaluated using these modified prototypes.

Following the initial performance evaluation and subsequent change evaluations, the phoneme set was expanded to 26 templates. This new set of phonemes was used to evaluate a new decision scheme. The averaging program was used on the first two speakers while all four speakers were used in data collection. The recognition rate was then compared against the original program recognition rate. The new decision scheme also rank orders the top five possible phoneme choices.

To aid in evaluating the results of the crosscorrelation program, a graph routine was written. This program displayed the phoneme/sentence crosscorrelation data in graph format, plotting amplitude against time. This gives a visual display of how each phoneme correlates with the sentence being evaluated.

II. Data Acquisition

The first step in the data acquisition scheme is to obtain the required speech input data in a form which is roughly analogous to that received by the human ear. Since the basic function of the outer ear system is to transform the pressure variations of sound into a format which can be used by the frequency analysis portions of the middle ear, the initial portion of data acquisition mimics this function through the use of an audio tape recorder.

Basic data acquisition consists of speaking the desired sentence into one channel of a reel-to-reel stereo tape recorder. Tone "markers" of 2 khz are spaced at ten second intervals on the second channel. These tones serve as a calibration system which allows personnel operating the preanalysis programs to have a means of knowing the location of each input sentence. Figure 1 illustrates the overall layout of the initial data acquisition scheme.

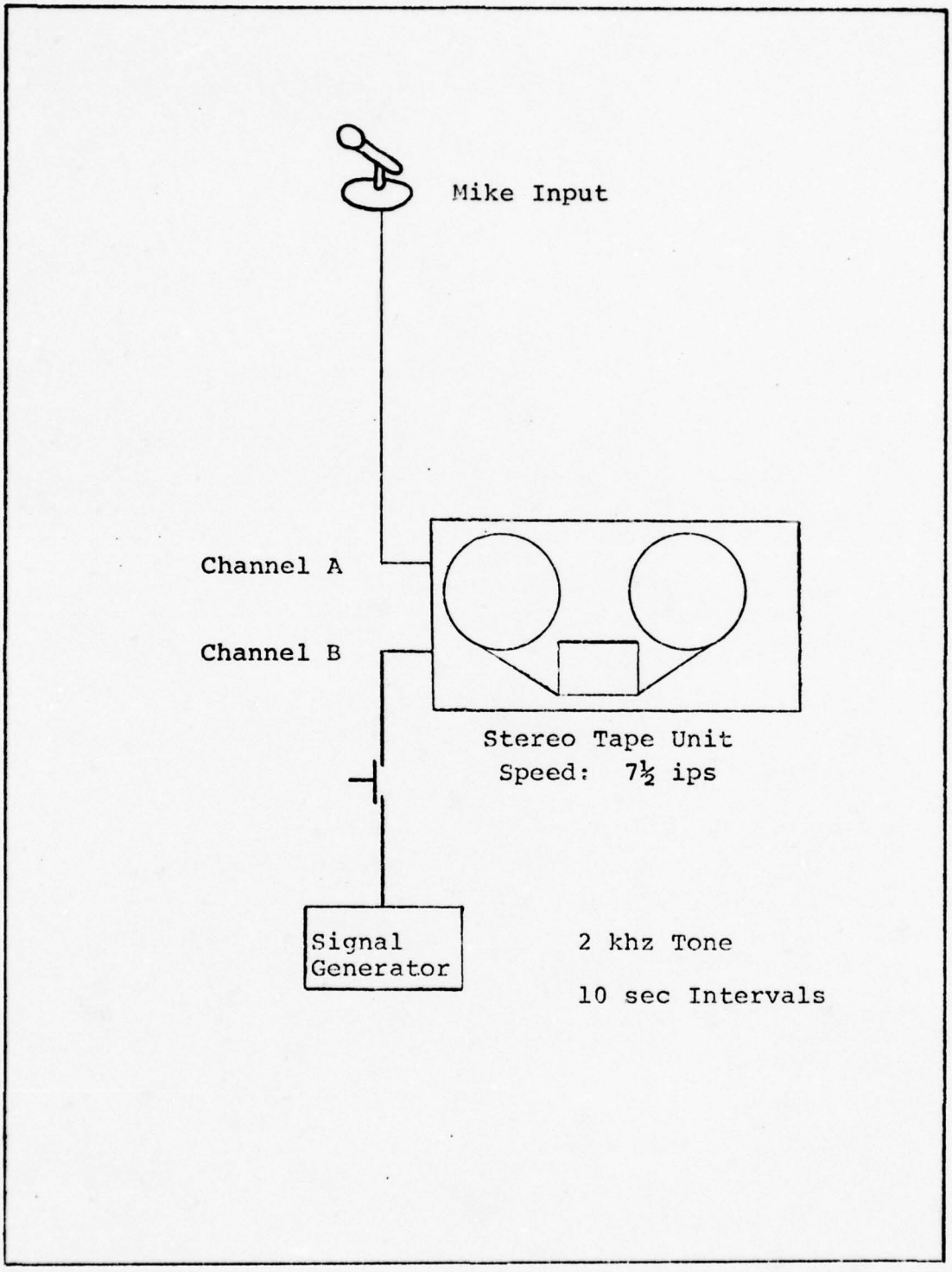


Figure 1. Initial Data Acquisition

III. Data Preprocessing

Initial work done in the area of speech recognition involved the use of a vocoder to simulate the actions of the inner ear system. The inner ear serves to accept the pressure changing inputs of the outer ear as its input. It outputs the speech data in the form of a running frequency and amplitude analysis of its input. The vocoder served to effectively simulate this action of the ear by operating as a series of matched filters which gave an indication of the amplitude of the output of each filter at a particular time. A running analysis of the various filter outputs would then perform functions which could mimic some operations believed to occur in the brain.

A simple block diagram of a vocoder layout is represented in Figure 2. Note that the analog outputs have to be converted to a digital format and then each digital word would have to be recorded in some sequence for further analysis by the information processing program.

Neyman found that operation of the vocoders available at that time presented problems with both accessibility and reliability (Ref 8). He chose to initiate an analysis program which would serve to imitate the vocoder through the use of Fast Fourier Transform (FFT) techniques.

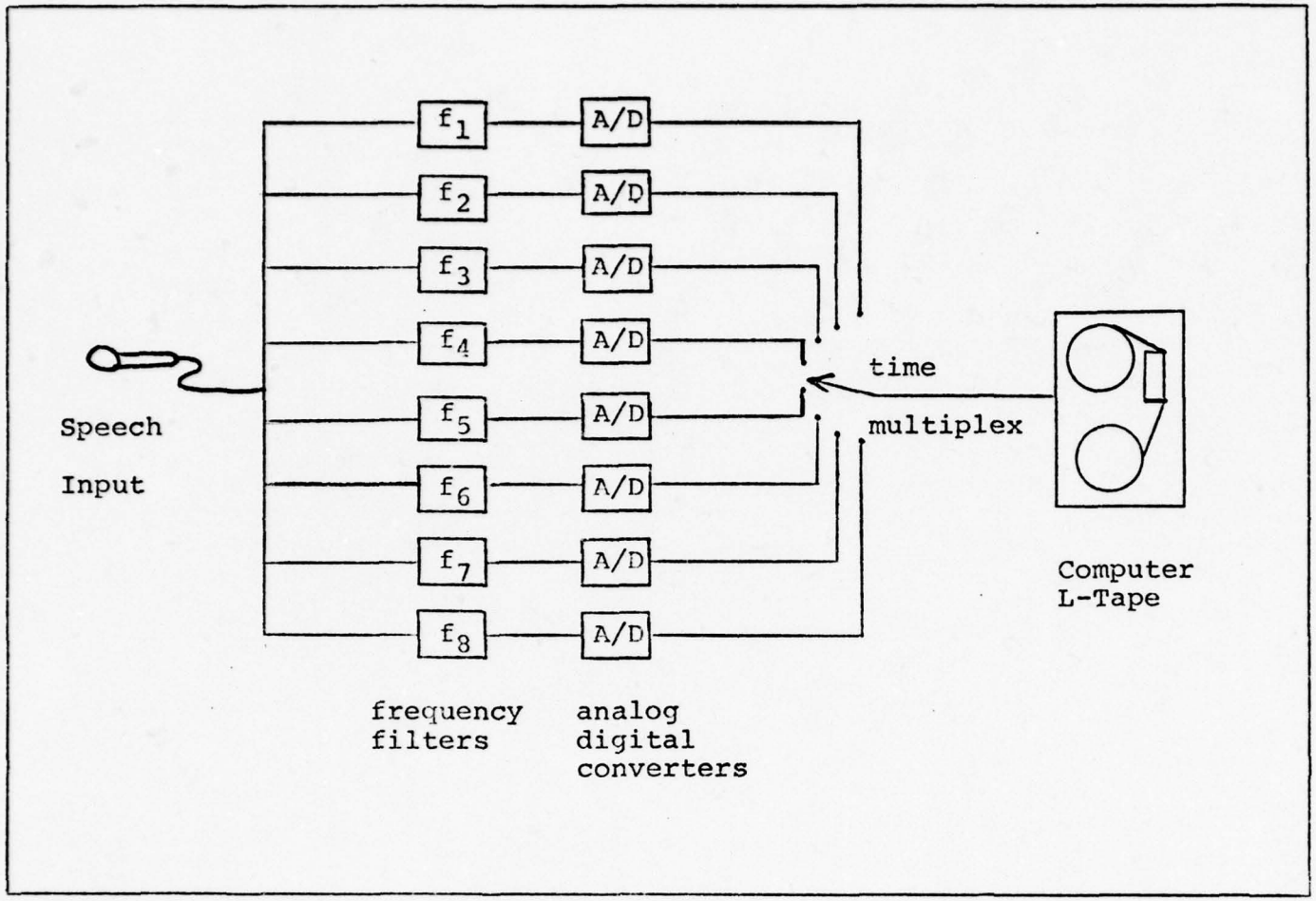


Figure 2. Block Diagram of Vocoder

Vocoder Simulation

The stages of speech signal preprocessing described below are accomplished by the Analog/Hybrid Systems Branch of the ASD Computer Center.

Amplification. The first stage of the preliminary signal preprocessing consists of amplifying the speech signals to a level which can be used by a digital-to-analog converters of the Comcor ci-5000/6 analog computer. The amplifiers contained within this machine are usable to only 2.5 khz. Since high quality speech data contains frequencies of close to 5 khz, it was necessary to compensate in some manner for the reduced amplifier response. To do this, the input speech tape was played at a speed of 3 3/4 inches per second. The resulting audio signal was low-pass filtered to insure a max input frequency of 2.5 khz. The input signal was then sampled at twice this rate (5 khz) in order to satisfy the Nyquist sampling requirements. Note that this overall procedure is equivalent to playing the tape at its originally recorded speed of 7 1/2 inches per second, filtering the tape to eliminate signals above 5 khz, and sampling the filtered output at 10 khz.

The sampled signals were then boosted to 100 volts to allow accurate sampling by the 11-bit analog-to-digital (A/D) converters. The output of the A/D converters was a binary representation of a four digit number, and described the amplitude of the analog voltage output at a particular

time. These numbers served as a digital representation of the time varying audio signal and were used as input to the frequency analysis portion of data preprocessing.

Frequency Analysis. Ever since the techniques of Discrete Fourier Transform computations were perfected in the late 1960's, the uses of the Fast Fourier Transform (FFT) to compute a frequency analysis of time-varying data have been well known and documented (Ref 2:41-52). It is this property of the FFT which is used in the frequency analysis portion of the signal preprocessing. The technique of analysis is implemented as follows: The incoming digital representations of the analog signals are taken in sets of 128 and used as a 1 x 128 input array to the FFT algorithm. Since each frequency sample represents the analog output at 10^{-4} seconds, 128 of these samples represent a total elapsed time of 12.8×10^{-3} seconds. The algorithm computes the Discrete Fourier Transform (DFT) of this time series and returns the amplitudes of each complex number in the frequency spectrum.

These frequency amplitudes represent the frequency spectrum of the input time signal. Each point in the FFT array represents an integral multiple of 78.125 hz (10 khz/128). Since the input time series x(t) is composed of only real numbers, the real part of the Fourier transformed series X(w) is symmetric about the folding frequency (one-half the sampling frequency). The magnitudes of the Fourier transformed series are also symmetrical

about this folding frequency. The final result is that, although 128 samples are taken, and a 128 point DFT is done on the time series, only the first 64 of the resulting transformed values are used to represent the frequency spectrum of that portion of the analog signal.

Figure 4 on page 15 illustrates pictorially the application of the FFT techniques to the input signal. The FFT transformations are done on a special purpose Xerox Digital computer operated in conjunction with the Comcor computer mentioned previously. The actual program which produces the desired results is called AMPSPC and is implemented by the Analog/Hybrid Systems Branch, ASD.

Data Storage. Since each 12.8×10^{-3} seconds of speech data will incur a storage requirement of 64 "sets" of numbers, it is obvious that a two or three second segment of speech can involve the generation of over 1.5×10^{4} data points. In addition, the requirements of the correlation programs for varied input data for subsequent analysis makes it necessary for the data preparation of more than one sentence. It is thus necessary to place the frequency analysis data in a form which is easily stored and is still accessible to the subsequent stages of analysis.

The format used to store all speech data after the frequency analysis stage is to write it in a form which is compatible with the input capabilities of the Cyber/6600 computer. The data is thus "written" on a computer library tape which is stored in the ASD Computer Center for access by

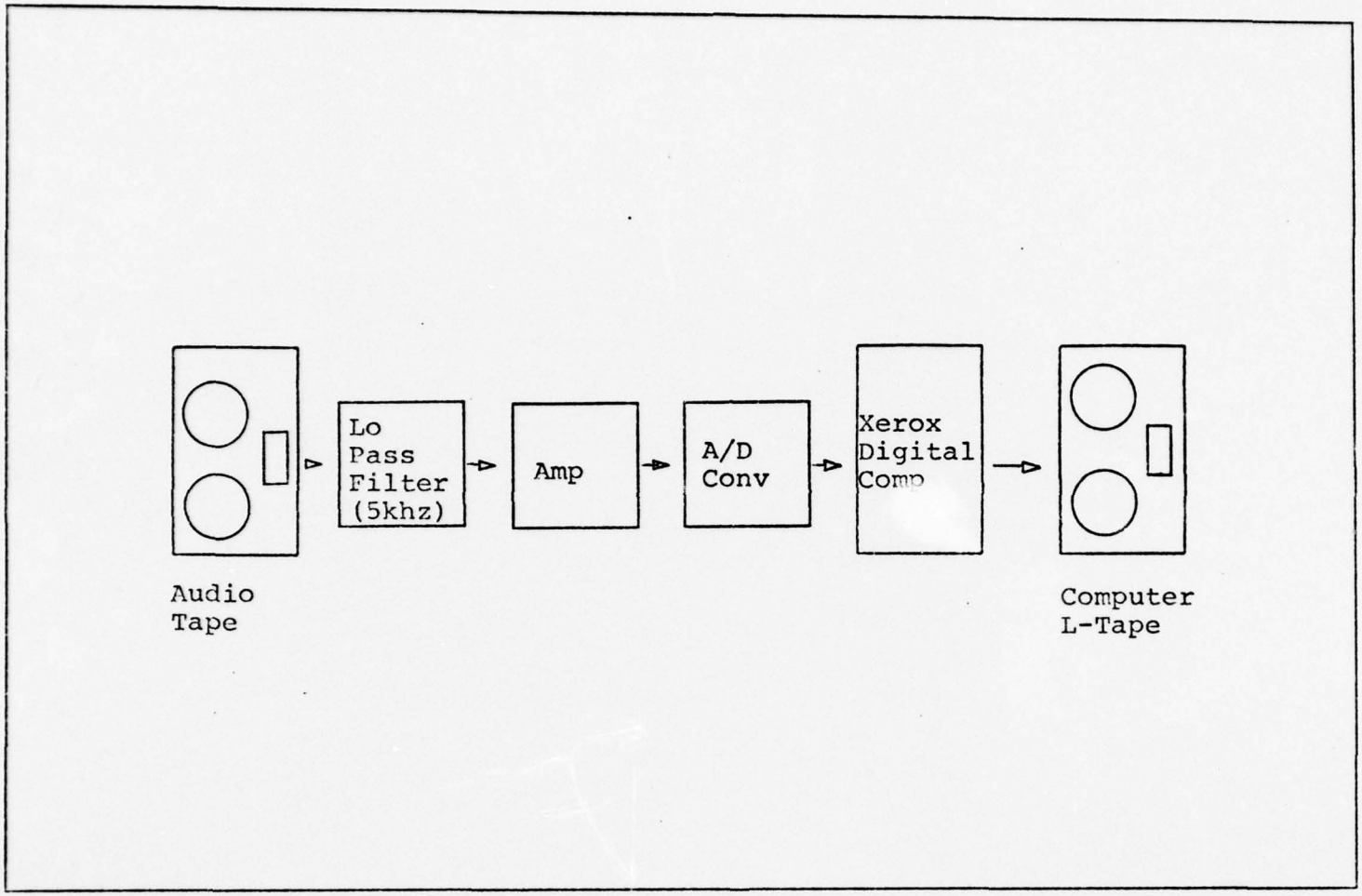


Figure 3. Block Diagram of Vocoder Simulation Using FFT Techniques

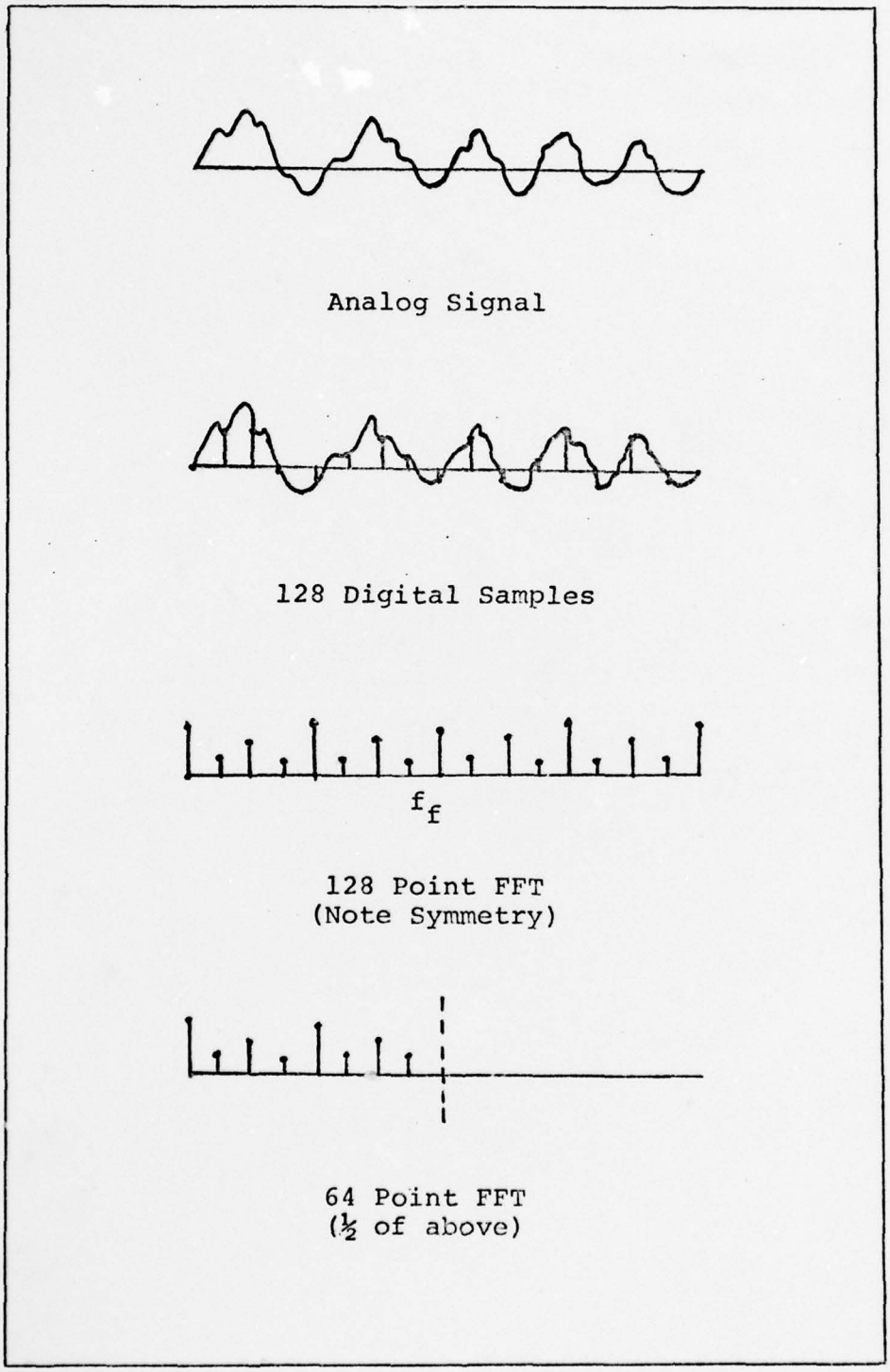


Figure 4. Frequency Spectrum Using FFT

subsequent programs (in particular: GS1). Once the original audio signals have been transformed into frequency analysis data and written onto a computer library tape (L-tape), the data preprocessing phase is complete.

IV. Signal Processing

After the analog signal has been digitized and written on tape in the manner described in section three, the digital records are in a form which is usable by subsequent processing stages. The data, as it now exists, is in the form of a digitized output of 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.

Channel Reduction and Equalization

Due to the fact that the ear-brain system seems to respond to ratios of frequencies rather than absolute frequency values, and since previous work with vocoders seemed to indicate that fewer frequency filters still yielded intelligible information, the 64 original channels were reduced in a manner which would simulate the ear's actual sensitivity to frequency changes (Ref 5:85).

The 64 channels are reduced in the following manner. The first channels, representing output center frequencies from approximately 78 to 470 hz are left unchanged. The remaining 58 channels are separated into approximately 1/3 octave groups. The channels within each group are then added (thus weighting the values at the high end of the

frequency scale) and combined into a total of 10 additional channels. The final result is 16 channels with center frequencies as given in Table III.

The overall effect of this channel reduction is somewhat analogous to a phono equalization or preemphasis curve. Through the use of this preemphasis, the high frequency information is not lost in comparison to the much greater amplitudes of the low frequency information. The actual frequency equalization curves for the vocoder and the digital simulation are shown in Figure 5. Note that the curves are roughly similar, with the higher frequencies being given slightly more emphasis on the digital simulation scheme.

Visual Output (Spectrogram)

Once the speech data is in a form which can be handled by the analysis portions of the recognition scheme, it is helpful to look at the output in a form which makes possible a visual analysis. Much work has been done in this area by Potter, Kopp, and Green (Ref 9). They found that there seemed to be enough visual clues in a time-frequency spectrogram to allow trained personnel to do a remarkably accurate job of interpreting the original speech. Thus, the next step is to transform our speech data into a form which would resemble a speech spectrogram. The method used involves the implementation of a two-dimensional printing scheme which plots the output of each frequency channel on one axis and the time of the occurrence on the other axis.

Table III

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	2812.500
390.625	390.625	2890.625	
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	1132.810	3515.625	3554.690
1093.750		3593.750	
1171.875		3671.875	
1250.000		3750.000	
1328.125	1445.310	3828.375	
1406.250		3906.250	
1448.375		3984.375	
1565.500		4062.500	
1640.625	1793.380	4140.625	
1718.750		4218.750	
1796.875		4296.875	
1875.000		4375.000	
1953.125		4453.125	
2031.250	2226.560	4531.250	4453.125
2109.375		4609.375	
2187.500		4687.500	
2265.625		4765.625	
2343.750		4843.750	
2421.875		4921.875	
2500.00		5000.000	

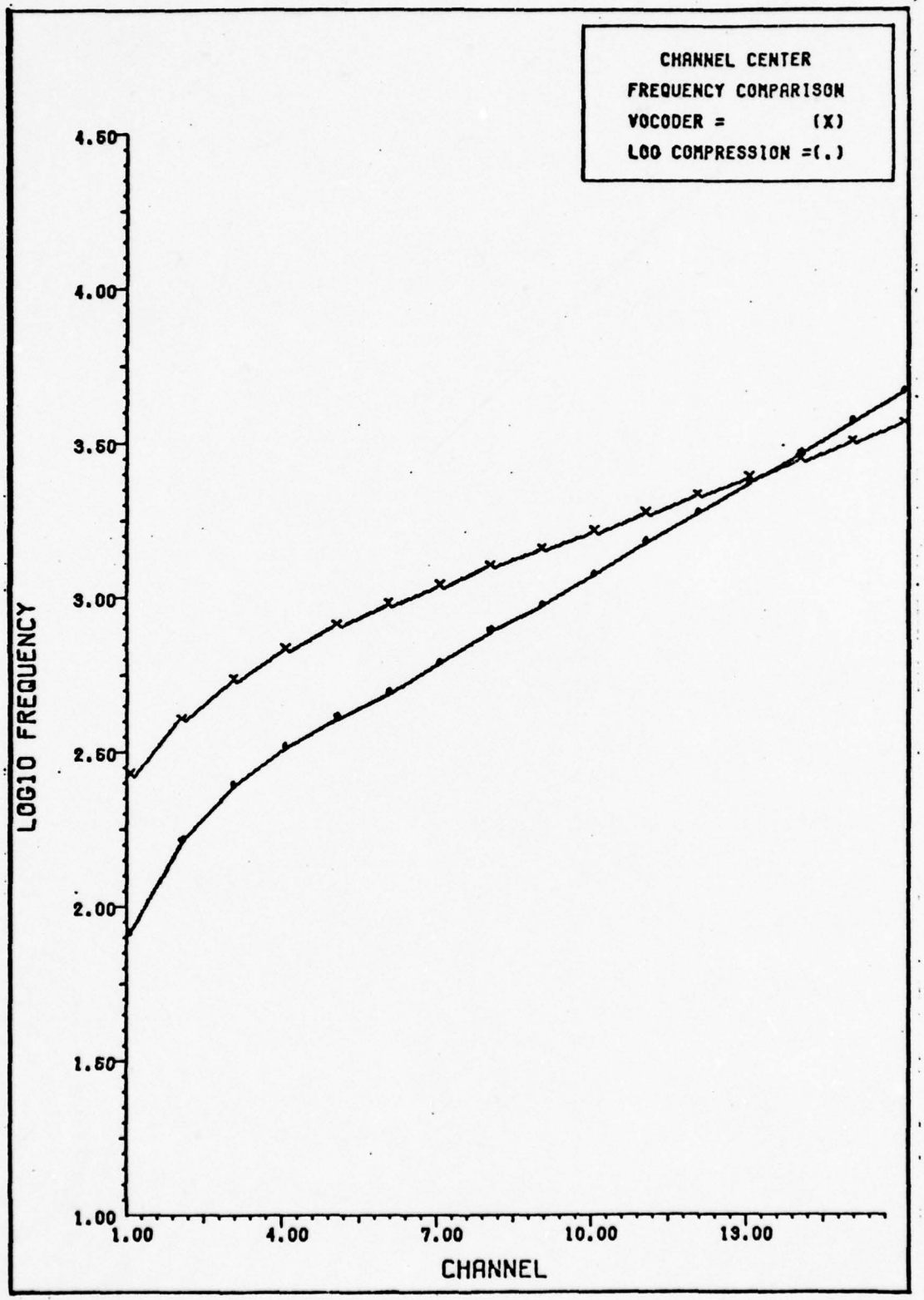


Figure 5. Channel Center Frequencies

The processing program has been devised so that it will present both the numerical magnitude of the output of each frequency channel as well as a pictorial representation of the combined sixteen channel outputs. The pictorial representation uses an overprint arrangement which causes each channel to become increasingly dark as the channel amplitude increases. Figure 6 gives an example of a specific speech occurrence along with its representative spectrogram. The speech spectrograms obtained in this manner closely mimic the frequency-time spectrograms mentioned by Potter, Kopp, and Green.

The computer program which performs the frequency reduction, equalization, and spectrogram output is listed as program OCTAVE1 (GS2) in the appendix. In addition, OCTAVE1 stores the reduced data on permanent files to be accessed by later stages of the recognition process.

Data Normalization

The very nature of speech gathering and digital representation assures a recognition scheme of having a time varying input which can fluctuate between wide limits. Even the same sentence spoken by the same speaker will have different representations in amplitudes, timing, etc. It is for this reason that data normalization is required. Previous work done by Neyman (Ref 8) and Hensley (Ref 5) has emphasized the importance of data normalization. There are two types of normalization which will be used in this paper: column normalization and unit normalization. The column

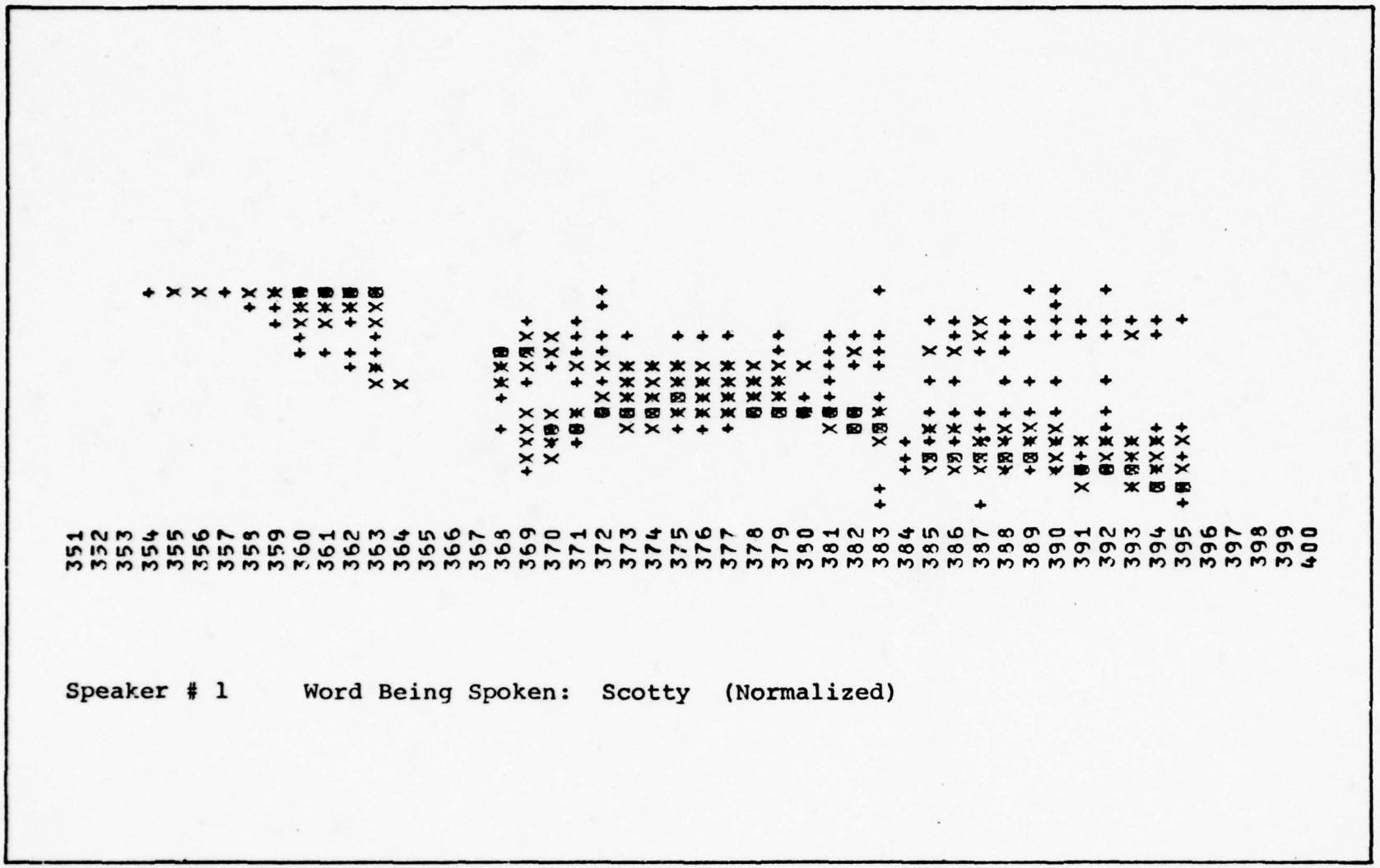


Figure 6. Digital Speech Spectrogram

normalization procedure shall be discussed below. Due to other program considerations, the unit normalization will be accomplished during the correlation section.

Column Normalization. Following the completion of the data reduction program mentioned previously (OCTAVE1), the data is available as a 16 x M array, where M is the length of the sentence in 12.8 msec intervals. The output of each "filter" at each time increment is a column of 16 numbers. The column normalization procedure consists of replacing each element e_i in the column with e_i/E where E is computed as follows:

$$E = \left(\sum_{i=1}^{16} e_i^2 \right)^{\frac{1}{2}}$$

What has been done is that the "energy" of each column has been found, and each element has been divided by that energy. This procedure serves as a type of automatic gain control and insures that the energy of each column is equal to one. The fact that each column has an energy of one will be of great importance later in arriving at normalizing values for the correlation vectors.

The program which implements this normalization procedure is called OCTAVE2 (GS3) and is listed in the appendix. Note that OCTAVE2 uses the files created by OCTAVE1. This program serves only to provide a visual output of the frequency spectrograms. The correlation program also will use the files created by OCTAVE1 and will column normalize

the data for its own use. The data was stored in unnormalized form to allow the option of normalization in the correlation phase. Figure 7 illustrates the increased clarity which column normalization provides in the spectrogram outputs.

Data Base

Due to the fact that the preprocessing and processing phases are quite lengthy and can take weeks to implement, it was decided to collect speech data in quantities large and varied enough to serve as both control and test data throughout the research. The thrust of this investigation was to:

1. Establish a "baseline" performance using unmodified recognition schemes devised by previous experiments;

2. Investigate a different procedure for the creation of prototype "templates" used in the correlation phase;

3. Investigate the reduction in performance of speech recognition programs caused by multiple speakers and widely varying spacing between words; and

4. Design a modified correlation and recognition scheme which would have increased versatility.

The decision was made to use test sentences which were spoken in two different modes, discrete and continuous speech. Discrete speech is a sentence in which each word is pronounced carefully and distinctly, with definite "dead" space between words. Continuous speech, on the other hand,

Figure 7. Normalized vs Non-normalized Spectrograms

could best be described as the sentence spoken in a natural manner with no allowances made for machine recognition.

Sentence Preparation

Since the production of a "complete" set of phonemes representive of the English language was not a goal of this paper, it was decided to use two sentences which seemed to represent a reasonable cross-section of phoneme-like sounds. The specific sentences were chosen so as to have combinations of sounds which would present the correlation and recognition phases with a realistic and hopefully complex set of data. The two sentences chosen were:

1. "Kirk here, beam me up, Scotty."
2. "Quoth the Raven, nevermore."

Four male speakers were chosen to recite the two above sentences in the following manner:

1. Sentence one - discrete speech
2. Sentence one - continuous speech
3. Sentence two - discrete speech
4. Sentence two - continuous speech

Each speaker spoke a total of four sentences. The combined data base consisted of sixteen sentences. All were recorded and processed as described previously and were stored in the 16 channel reduced form.

Phoneme Analysis

The two sentences used were then analyzed for phonemic content. The following analysis of the sentences is not meant to be definitive, but is designed to serve as the

basis for obtaining phonemic patterns which would serve as an adequate data base for subsequent portions of the recognition scheme.

Sentence one was found to contain 15 phonemes. The beginning and ending sounds of "K" from Kirk and "M" from me and beam were given separate phonemic representations to allow further research into the differences of these allophones. The sentence and its phonemic representation used by subsequent programs are listed below. The subscripts following the K and M are the results of the differentiation between beginning and ending phonemes.

Sentence One

"Kirk here, beam me up, Scotty."

Phonemic Representation

K_b UR K_e H I R B IE M_e M_b IE U P S K AH T IE

Sentence two contained 11 additional phonemes. Sentence two and its phonemic representation are listed below.

Sentence Two

"Quoth the raven, nevermore."

Phonemic Representation

QU OO T_e T_b E R A V EN N EE V ER M_b OO ER

These analyses are admittedly arbitrary in differentiation of individual phonemic characters. Indeed, a true professional analysis of two persons speaking the same sentence would stand a good chance of yielding slightly different results when analyzed. What is true about all sentences of the same reading, however, is that virtually

any human versed in English will interpret the sentence in the same manner. It is this relative nonvariance in interpretation which is the goal of the recognition portions of this paper.

Phoneme Extraction

Previous research dealing with this method of continuous speech recognition used the speech inputs of one speaker to establish a prototype phoneme set which was then used as a data base for further speech inputs by the same speaker. In addition, sentences spoken by other speakers were tried, always with quite poor results (Ref 5). It was decided to use the techniques employed previously in order to establish a "base line" performance rating which could then be used as a reference for performance of the various procedural and technical modifications which were to be introduced.

The first phoneme prototype set which was constructed consisted of the fifteen phoneme-like sounds contained in sentence one. The discrete sentence spoken by the first speaker was chosen as the sentence from which the phonemes were extracted. The phonemes were taken from segments of the N x 16 array which composed the stored results of the preanalysis and reduction programs.

The spectrogram representation of the sentence was first visually analyzed for the possible locations of the target phonemes. The phoneme positions were noted and a program was implemented which extracted the desired portions

of the sentence array and stored these portions in a permanent file. This file became the source of prototype arrays which were to be used by the correlation program.

Pictorially, the phoneme extraction process is represented in Figure 8.

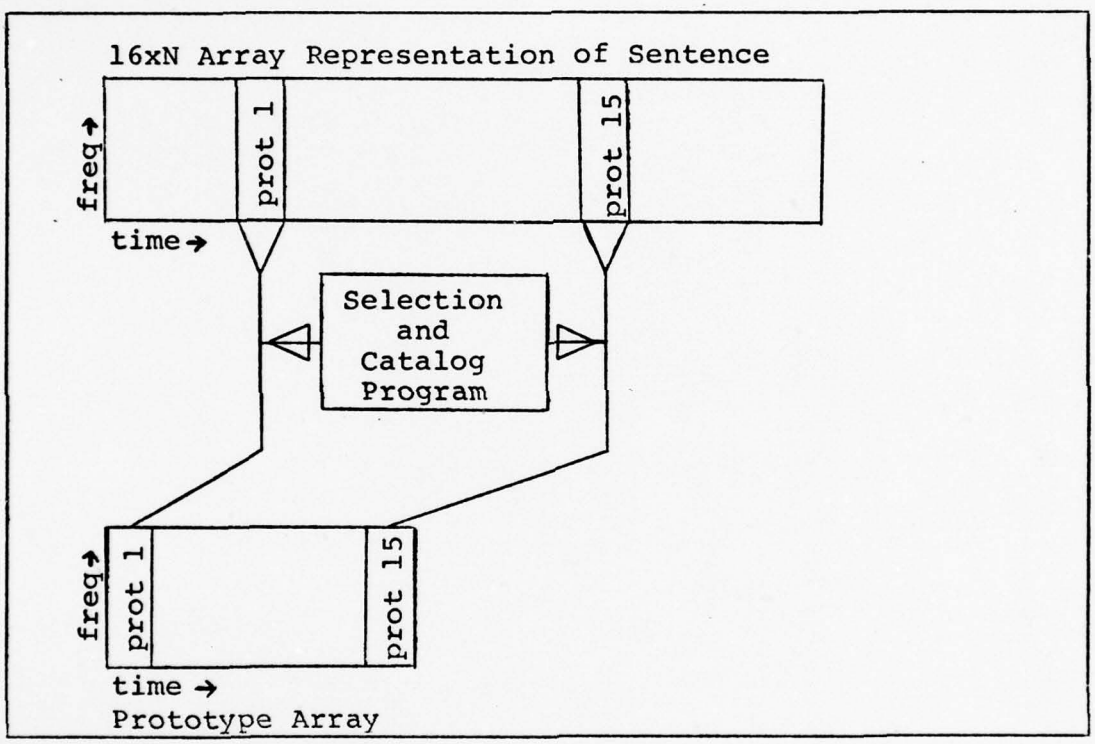


Figure 8. Prototype Extraction Process

Note that this selection program is universal in the sense that it can be used to extract and store any number of portions of any number of sentences. It is, in fact, used in a modified manner to produce the "fluid" prototypes mentioned in the next section. This program is called PROAVG (GS4) which is listed and discussed in Appendix A.

Phoneme Averaging

Following work done with the phonemes obtained from the first sentence, it was decided to introduce a modified phoneme extraction process which was designed to take into account the variability introduced into phonemic structure by two specific cases:

1. The same speaker saying the same sentence, but at a faster rate (discrete vs continuous speaking).

2. Different speakers saying the same sentence.

Analysis of spectrograms produced by GS3 showed that the overall spectral content of the sentences was remarkably similar. Although the discrete spectrograms proved to be much easier to interpret, the basic speech form of identical words, even when spoken by different speakers, seemed to be preserved. Therefore, the concept of prototype averaging was implemented.

The averaging method involved using the same phoneme extracting program as before. Following the phoneme extraction from the sentences of interest, the phonemes extracted from like words of both continuous and discrete speech were then averaged point by point. This required that all prototypes be of identical length. Since lengths varied somewhat with respect to speaker, this problem was approached by finding the shortest phoneme length available for a given phoneme in a given sentence. Representative portions of like phonemes from other sentences were chosen to be the same length.

Although the problem of varying phoneme lengths would seem at first to be quite difficult, analysis of spectrograph data revealed that the variations in length of both the discrete and continuous sentences with multiple speakers to be on the order of 20%. Since the largest prototype used in this research was 16 x 13, this variation in length is only 30 milliseconds which presented no great problem.

Figure 9 illustrates the procedure used in phoneme averaging. Table IV and V lists the phonemes used in the 15 class set (baseline data research), and the 26 class phoneme set used for evaluation of correlation and selection scheme changes.

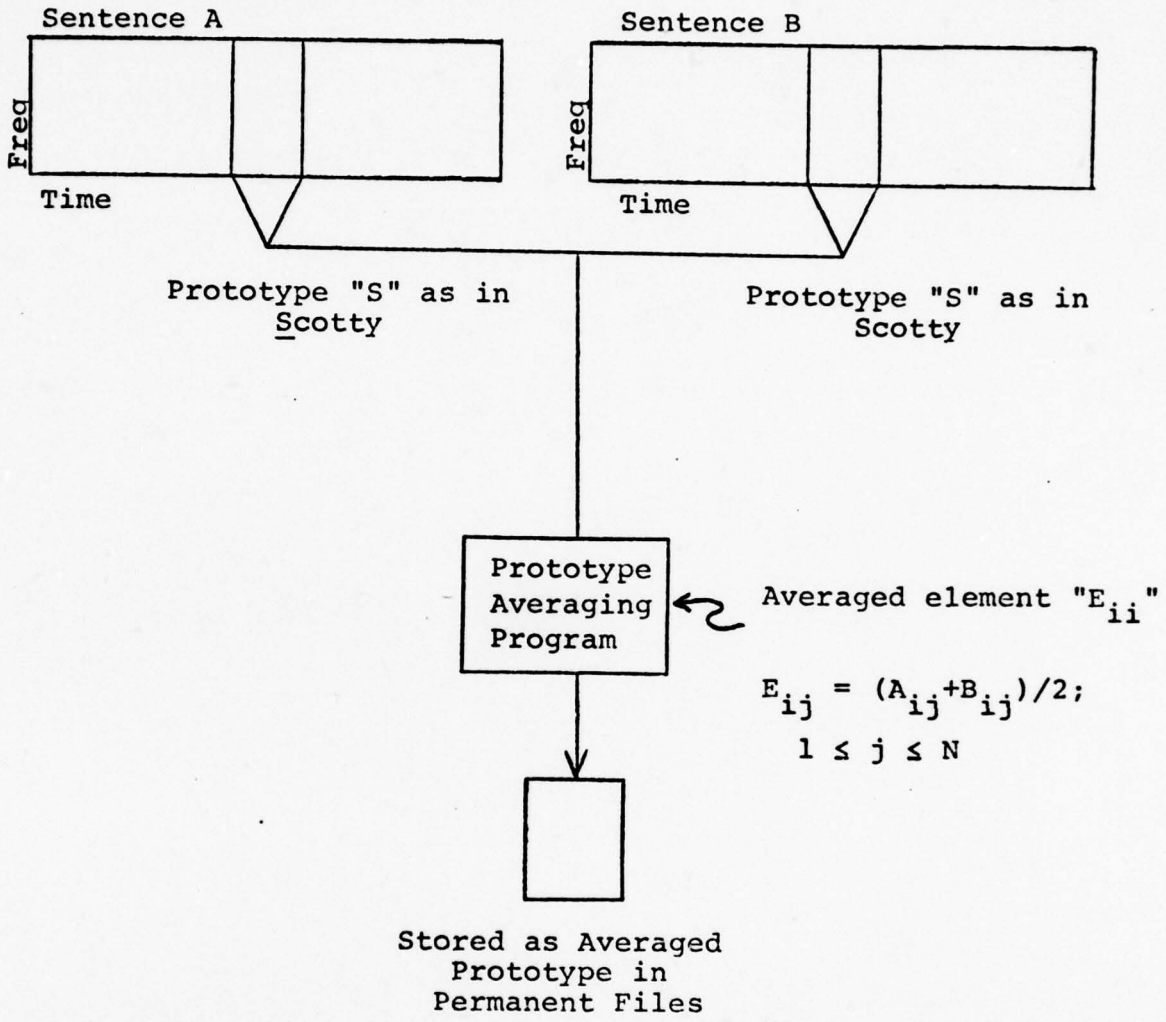


Figure 9. Flow Chart of Prototype Averaging Scheme

Table IV

Prototypes: 15 Class Problem

Phonome Representation	Length	Word Taken From
K_b	2	Kirk
UR	9	Kirk
K_e	2	Kirk
H	7	Here
I	8	Here
R	7	Here
B	3	Beam
IE	7	Beam
M_e	5	Beam
M_b	7	Me
U	7	Up
P	3	Up
S	6	Scotty
AH	13	Scotty
T	4	Scotty

NOTE: Both a discrete and averaged prototype set was used in 15 class problem. Lengths were the same.

Table V

Prototypes: 26 Class Problem

Phoneme	Length	Word Taken From
QU	7	Quoth
OO	8	Quoth
T_e	7	Quoth
T_b	10	The
E	7	The
A	6	Raven
VV	4	Raven
EN	6	Raven
N	6	Never
EE	7	Never
ER	6	Never

NOTE: These are in addition to those listed in Table IV. These were all averaged prototypes.

V. Recognition Processing

The correlation phase consists of attaching the files containing the stored prototype data and input speech data, then performing a running crosscorrelation of each prototype with the sentence. The output of this correlation procedure is an N x M array where N is the number of prototypes contained in the prototype set and M is the time length of the sentence. The value of each element is the correlation of that particular prototype with the sentence at a particular time. While the actual correlation procedure is relatively simple, various steps have to be taken in order to prepare both the sentence and the prototypes for the correlation computations. These steps include normalization, array augmentation, and DFT operations.

Normalization

One aspect which has been emphasized by both Neyman (Ref 8) and Hensley (Ref 5) was the importance of data normalization. Section IV dealt with the improvement attained in the clarity of spectrogram analysis by the process of column normalization. As stated before, the data was not stored in a normalized form so that it could be used in this section in an unchanged form. The correlation program is arranged so that column normalization is optional. However, all data processed and analyzed in this paper used the normalization option. This was done to minimize the effect of speaker variation. Both prototype

and sentence arrays are column normalized as they are read from the permanent files. This column normalization procedure is the only normalization that is done to the sentence data. The prototype arrays are unit normalized as well as column normalized, but this additional step occurs following DFT. The description of the unit normalization along with the reasons for doing it at a later stage will be discussed in the section dealing with Fourier filtering.

Array Augmentation

A major goal of this project was the modification of a previous correlation program to make possible a multi-segmented program which would allow greater latitude in postcorrelation analysis techniques. While actual real-time correlation techniques are not difficult, they require such a vast amount of computations that even large scale processing systems, such as the CDC Cyber 6600, would require excessive amounts of time to do the computations.

Improvements in past years of techniques of computing the DFT of matrices such as the Fast Fourier Transform (FFT) have made it possible to use the properties of the Fourier Transform to greatly reduce the computations needed for correlation (Ref 1). However, the use of the FFT requires safeguards against certain problems which are created. The most important of these problems are aliasing, leakage, and end effect.

Aliasing is the tendency of a high frequency signal to "mimic" that of a lower frequency signal if the sample rate

is not sufficiently high. This problem was solved during the digitization phase by insuring that the sampling rate (10 khz) was double the highest allowed voice signal (5 khz) and filtering to insure that no frequency above 5 khz was retained.

Leakage occurs due to the inherent properties of the DFT of a finite data record. This "rectangular window" of the time series causes the data to change in the DFT in a manner which can alter the overall results. While this alteration can seriously affect some types of data, Neyman found that overall results were not altered by the inclusion of various "window" functions which were designed to handle the leakage problem (Ref 8:34). The data used in this report were taken using a rectangular "window".

End effect occurs as a result of the periodicity imposed on a function by the use of the DFT. Although there are times when this duplication of the original function can be harmless, the very nature of the correlation calculations require that a "buffer" be included in the transformed functions so that the data which is being moved on the time axis does not run into repeated renditions of the data to be correlated. The problem can be eliminated by insuring that the arrays to be transformed are augmented in the following manner. Let P_{ij} be the prototype array and S_{ij} be the sentence array. If P_{ij} is of length P and S_{ij} is of length S, choose an N such that the following

two requirements are met:

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

$$N = 2^n$$

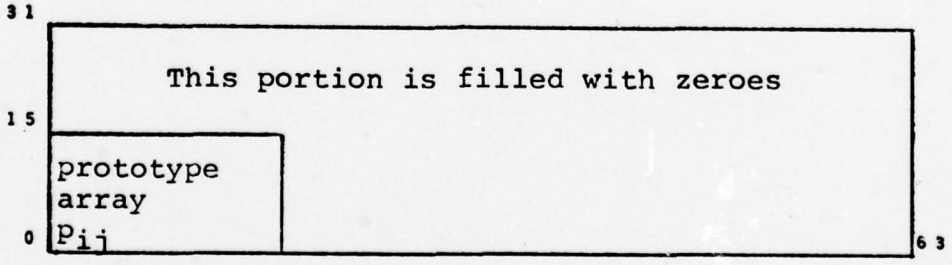
where n is an integer.

The augmented array is then formed using the following functions:

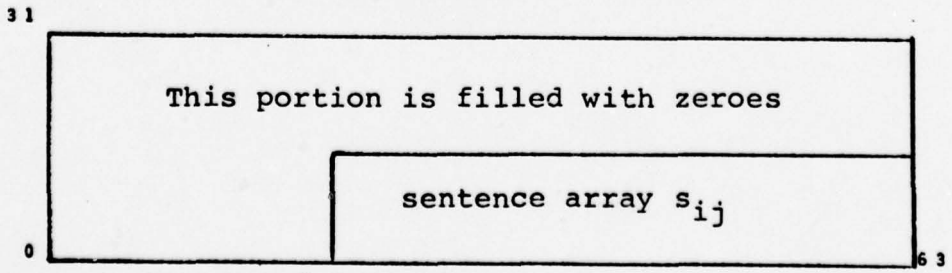
$$P_{kl} = \begin{cases} 0 & k=0,1,\ldots,N-P \\ & l=0,1,\ldots,15 \\ p_{ij} & k=N-P+1,N-P+2,\ldots,N-1 \\ & l=j=0,1,\ldots,15 \\ & i=0,1,\ldots,N-1 \\ & l=16,17,\ldots,31 \\ 0 & l=16,17,\ldots,31 \end{cases}$$

$$S_{kl} = \begin{cases} s_{ij} & k=i=0,1,\ldots,Q-1 \\ & l=j=0,1,\ldots,15 \\ 0 & k=Q,Q+1,\ldots,N-1 \\ & l=0,1,\ldots,N-1 \\ 0 & k=0,1,\ldots,N-1 \\ & l=16,17,\ldots,31 \end{cases}$$

The array transformation is pictorially illustrated below.



Augmented Array P_{kl}



Augmented Array S_{kl}

Figure 10. Augmented Array Diagram

As can be seen in Figure 10, the new augmented arrays, which are not both 32 x 64 arrays, can be multiplied point-by-point and yield another 32 x 64 array. In addition, a visual inspection of the results of a time-domain correlation will show that the imposed periodicity of a DFT on both of these truncated functions will not invalidate the correlation values due to the extra "buffer" space built into each array.

Fast Fourier Transform (FFT)

Following the augmentation of both the prototype and sentence arrays, the two-dimensional DFT is computed. The transformed functions are as follows:

$$S_{rs} = \sum_{k=1}^{K} \sum_{l=1}^{L} s_{kl} \exp -2j\pi \left(\frac{kr}{K} + \frac{ls}{L} \right)$$

$$P_{rs} = \sum_{k=1}^{K} \sum_{l=1}^{L} p_{kl} \exp -2j\pi \left(\frac{kr}{K} + \frac{ls}{L} \right)$$

The complex conjugate of P_{rs} (P^*_{rs}) was then formed, and the transformed arrays were then ready for the filtering and unit normalization process. Following the filtering and normalization stages, the element-by-element product

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

was computed.

The result of this multiplication is equivalent to correlation in the time domain. To obtain the actual

correlation values, the inverse transform of Z_{rs} was computed. The function is computed as follows:

$$z_{kl} = \frac{1}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} Z_{rs} \exp 2j\pi \left(\frac{kr}{K} + \frac{ls}{L} \right)$$

Following the inverse Fourier computation, the correlation vector is formed by taking the first, or zero shift, row from the z array. This row is written into the correlation array to be used by the graphing and decision schemes.

$$C_i = z_{kl}$$

where

$k = Q, Q+1, \ldots, N$

$l = 1$

(The first Q-1 values are not used due to end effect described on page 37.)

Filtering

Kabrisky (Ref 6), Daily and Sutton (Ref 3) found that filtering done in Fourier space helped to improve the performance of two-dimensional pattern recognition devices. Hensley chose to incorporate a filtering scheme into the correlation procedure by inserting a variable window filter into Fourier space. The rectangular filter as structured served as a low-pass filter whose cutoff frequency was controlled by two variables, width and length. Although the insertion of this filter seemed to improve overall performance, the rectangular nature of the filter seemed to have

the potential of introducing problems of leakage similar to those listed in an earlier section. It was decided to retain the filter as designed, but to alter the program so that only one program change would have to be made in order to vary the characteristics of the filter. The filter served to remove high frequency information in the Fourier transformed array as follows:

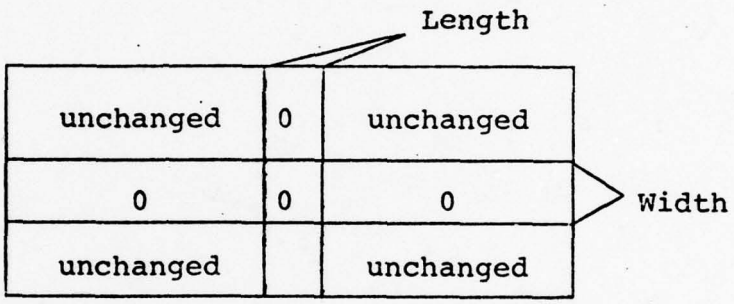


Figure 11. Modified Prototype Array

The zero section is a direct function of the variables width and length which are themselves functions of the single variable "IFILT".

The variables width and length are formed as follows:

$$\text{Width} = \text{Midth} = 32\text{-IFILT}/2$$

$$\text{Length} = \text{Mength} = 64\text{-IFILT}$$

where

$0 \leq \text{IFILT} \leq 64$; 0 removes filter from scheme

64 removes all Fourier information

Hensley had placed this filtering operation after prototype unit normalization. Since the filtering operation removed energy from the prototype, and since correlation

normalization relied upon a prototype array energy of one, this may have introduced inconsistencies in the resulting data. For this reason, the filtering operation was moved so that it was performed on the prototype array after the FFT, but before it was unit normalized and the energy then computed. The results of the filter routine are discussed in Appendix C.

Unit Normalization

The prototype array was column normalized prior to the DFT operation. This column normalization procedure, as discussed earlier, insured that the energy of the prototype was a direct function of the length since each column had an energy of one. Unit normalization was done to insure that each prototype, no matter what its length, had unit energy prior to the correlation computation.

The unit normalization was computed as follows:

$$P_{ij} = P_{ij}/(\text{Energy})^{\frac{1}{2}}$$

where

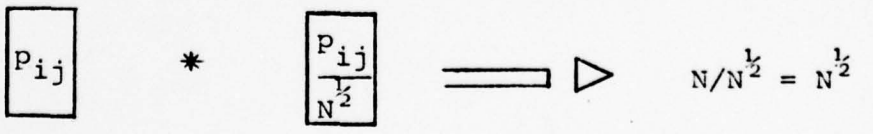
$$\text{Energy} = \left[\sum_{i=1}^{32} \sum_{j=1}^{64} (p_{ij})^2 \right]$$

Following these computations, the prototype vector had an energy of one. Since the prototype had previously been column normalized, the total energy value computed above would be a direct function of N, the length of the prototype. The total energy is, in fact, N, and each element is divided

by $N^{\frac{1}{2}}$. This will play an important part in the correlation normalization procedure which is mentioned in the next section.

Correlation Normalization

The correlation normalization procedure is required in order to have a basis for comparison between the maximum correlation values obtained over all time for all prototypes. Obviously larger prototypes will incur a greater maximum value when they encounter portions of sentence data like themselves than will the shorter prototypes. There must be a method of equalizing the maximum values so that the performance of the prototypes can be compared. Since all arrays have been column normalized, and since the prototype arrays have been unit normalized, it is easy to see that the maximum correlation obtainable by a prototype which encounters an exact replica of itself in a sentence would be $(N)^{\frac{1}{2}}$ where N is the length of the prototype. The reason for this is illustrated below:



Column Normalized Prototype | Column + Unit Normalized Prototype | Maximum Correlation Value

Figure 12. Correlation Normalization

Since the previous section established the fact that the energy of the prototype, when unit normalized, was also N, a correlation normalization can be computed by simply dividing the correlation values by the square root of the energy of the prototype ($N^{\frac{1}{2}}$).

Parseval's Theorem states that energy is conserved during Fourier operations. However, the nature of the discrete Fourier Transform algorithm used alters the actual energy in a predictable way. In this particular case, the energy is reduced by a factor of $(N \times M)^{\frac{1}{2}}$ where N and M are the dimensions of the transformed array. It is this value which was arrived at empirically by Neyman (Ref 8) and Hensley (Ref 5).

The computed energy of each prototype is stored as Good (JP) in the computer program which performs the correlation. As mentioned, this value is used by the program to divide the computed correlation values to insure that all prototypes will incur correlation values between zero and one. In this way, the relative values of each prototype can be compared and evaluated.

Data Storage

Following the computation of correlation values for all stored prototypes, the resulting array is stored in permanent file for evaluation by the decision scheme. Storage in this manner allows the access of correlated data by various modes of decision schemes and allows greater versatility in the analysis of overall program performance.

Correlation Graph Output (Calcomp)

An interesting method of analyzing the results of the correlation routine is to present the data in a manner which is analogous to the output of "matched filters" with respect to time. The matched filters in this case are the prototype templates. A graphing routine has been designed which allows selected prototype correlation values to be sent to a Calcomp Graphing Routine. This routine graphically depicts the running correlation of a particular prototype in a particular sentence.

Figure 13 shows the output of the first four prototypes in the 15 class problem as they were correlated with the first sentence: "Kirk here, beam me up, Scotty." The sentence started at time interval 20, the word "Kirk" appeared at time interval 78. Note the rapid rise in the "KB" and "KE" filters at the appropriate times along with the more gradual response of the longer "UR" filter.

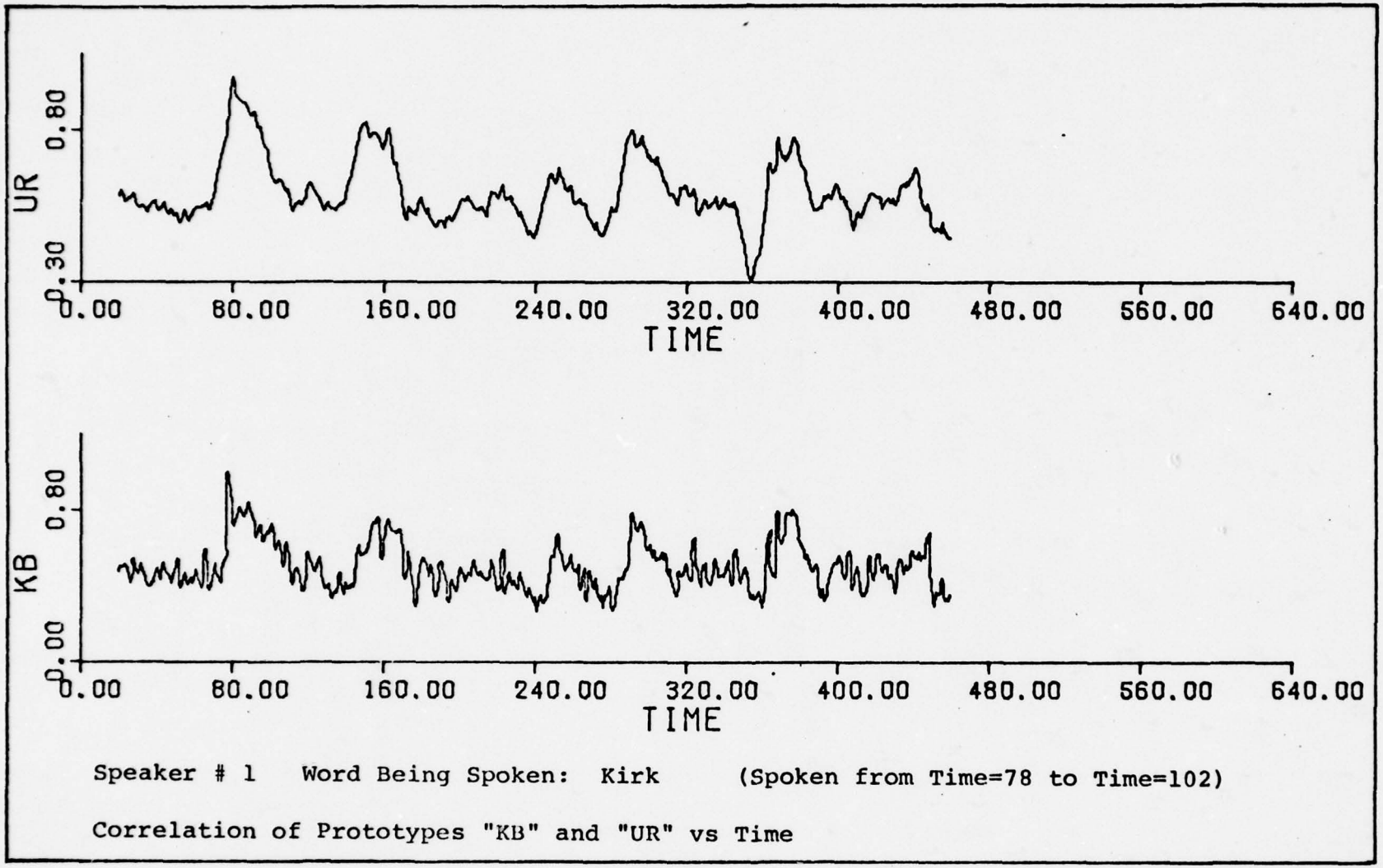


Figure 13. Correlation Graph Output

VI. Decision Scheme

Following the completion of the correlation program, the results are stored in the form of an M x N array where M is the number of prototypes and N is the length. Each element in the array represents the correlation of a particular prototype with the sentence at that particular time. The array can be pictured as follows:

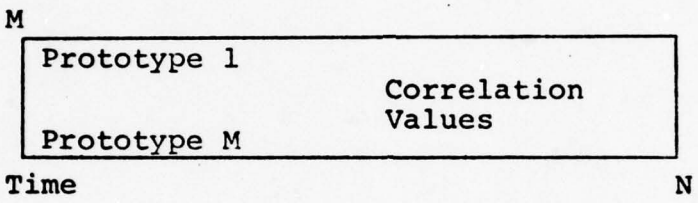


Figure 14. Correlation Array

The performance of a particular prototype throughout the sentence is on the horizontal axis. A comparison of all prototype correlations at one particular time is obtained by looking at the vertical axis. It is this array which is used for subsequent portions of the decision scheme.

Threshold

The array is first processed for values which are above a selected threshold. Values equal to or greater than the selected threshold are left unchanged; all other values are set equal to zero. This threshold operation is easily imagined by drawing a horizontal line on the correlation

graphs mentioned earlier. It is quite easy to gain a preliminary idea of when the phoneme occurred at a given time by observing its peaks.

Endurance

It is obvious from the graphs in Figure 13 that the shorter prototypes have a much more irregular and peaky correlation graph. In particular, an examination of the prototype "KB" shows that it has numerous peaks, even though the actual "KB" sound occurred only once in the sentence. It was decided that a time endurance criteria would be established to insure that "false alarms" incurred by momentary correlation values above threshold would not serve to make the final decision scheme any more complicated than it already was. The endurance criteria involved scanning the correlation array along the time axis. When a correlation value above threshold was found, a marker was set. When the correlation value fell below threshold, the time the threshold was exceeded was checked. If this time was below some specified percentage of the time value of the prototype, e.g., 1/2 the length, that portion of the array was set to zero.

Following endurance processing, the correlation array consists of values above the desired threshold which stayed above this threshold for an amount dependent on a function of the prototype length.

Ranking

Following the threshold and endurance processing, the correlation array is ready for the final decision output. At first, it seemed quite logical to simply pick the highest correlation value at each time increment and use this as the prototype selected. However, preliminary experimentation with this process showed remarkable results when the prototypes were autocorrelated (used to judge the sentence from which they came), but poor results when the prototypes were used with sentences from different speakers.

These results, coupled with similar results from other research efforts led to the conclusion that there may be higher order decision schemes which are used by the brain to determine actual word content of sentences following a less than perfect recognition process. It was decided to design a final decision scheme which would simply list a number of choices as the representation of the speech. This type of output had two advantages:

1. It would allow the data to be stored for further processing by decision schemes which would take into account higher order levels of structure such as syntax, grammer, and context.

2. It would allow an easily understandable representation of the relative performance of each of the prototype arrays. Incorrect decisions could be scanned for the position of the correct choice so as to determine methods of producing more correct results.

The ranking program, as used in this paper, searched the processed correlation arrays at each time segment for the five highest correlation values. The phonemes which incurred these values were then printed in the order of their correlation values, from highest to lowest. This was the final computer processing stage in the recognition process. However, it is quite possible to store this final decision array to be used as data for some future syntax or grammer processing routine. The overall processing stages of the decision scheme are shown in Figure 15. The program which performs the described operations is called DECIS (GS6) and is listed in the appendix.

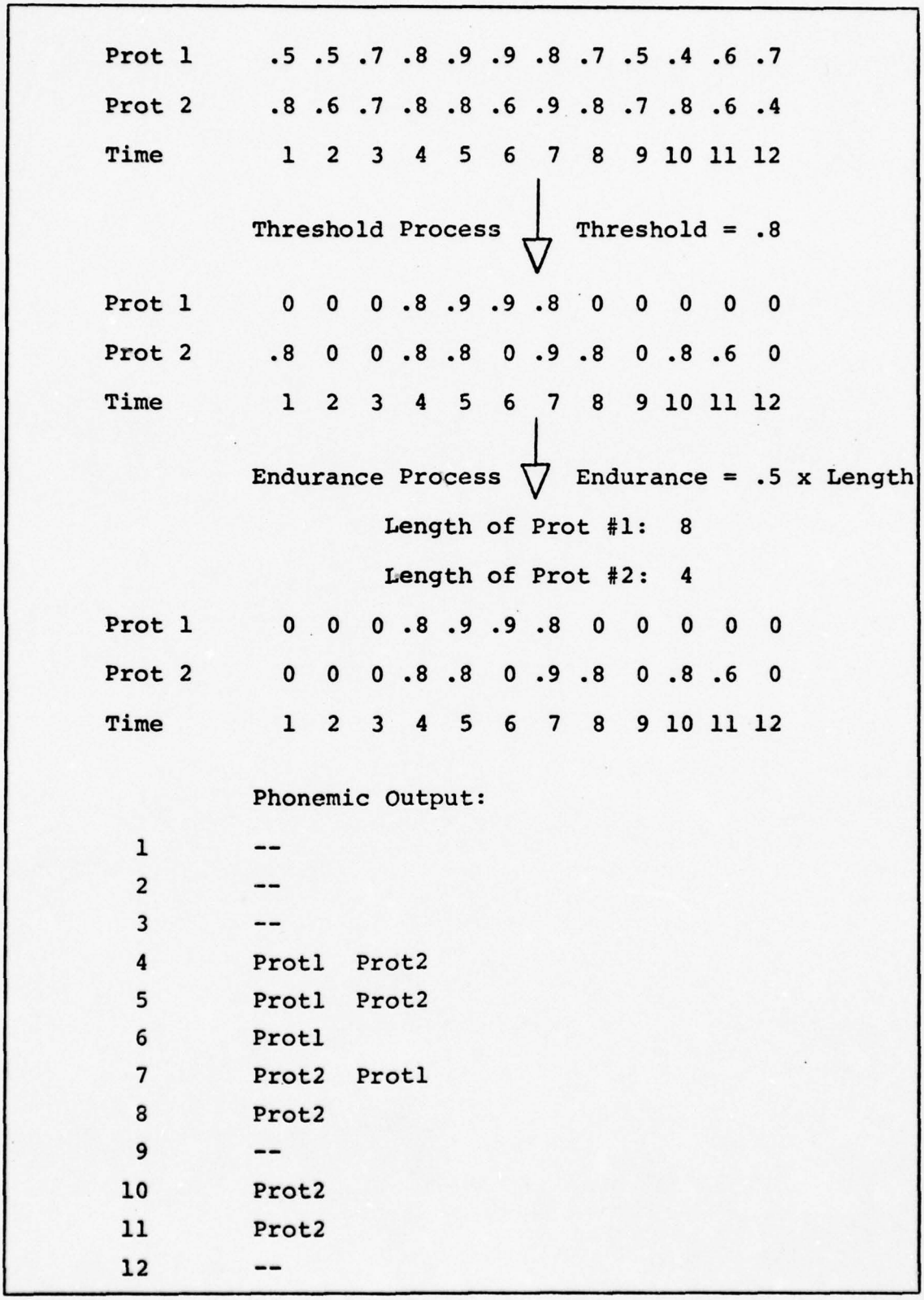


Figure 15. Decision Scheme Operation

VII. Results

The results are divided into three main sections. Section one establishes a baseline performance evaluation for the program as originally designed by Neyman and modified by Hensley. Section two deals with the 15-class problem when prototypes were used from an averaged set of data. This section also deals with the difficulties encountered with the existing classification scheme and the modifications where were incorporated. Section three lists the results of an expanded 26-class problem.

15-Class Problem (Discrete Prototypes)

The first task in the modification of existing computer algorithms is to establish a baseline performance rating so that there is a guideline for comparison. Additionally, the problem differences in scoring the individual data outputs can be lessened. This is due to the fact that both the baseline and subsequent results can be scored using the same data base along with the same scoring technique. Ideally, it would have been best to use the actual prototype and sentence data from previous efforts in order to see what improvements could be gained by revised methods. However, previous data sets were unavailable at the beginning of this research.

The 15-class problem consisted of phonemes extracted from the sentence: "Kirk here, beam me up, Scotty." The sentence was spoken by test subject one and was spoken

in a way as to insure that each word was spoken clearly, distinctly, and was not connected with the words surrounding it.

Analysis of this sentence yielded a set of 15 phoneme-like sounds which were chosen to represent the entire sentence. These phonemes were extracted and cataloged according to the phoneme extraction program as described in Section IV. The prototypes were then used in the correlation and decision programs as listed in the Hensley paper (Ref 5). Scoring was done in the following manner:

1. A phoneme was "located" if it fulfilled the specifications listed for location in the Hensley paper.

2. A phoneme was "identified" if it was printed as a correct selection in the phonemic representation of the sentence.

The scoring of the sentence was as follows. The score for both location and identification consisted of a percentage number which was derived by dividing the number of correct choices by the total number of phonemic elements believed to be in the sentence.

The actual program performance is listed in Table VI. The individual sentence scores are listed in Appendix C.

Analysis

The overall performance of the recognition programs as used by previous researchers proved to be essentially perfect when used to autocorrelate the phonemic data. The 92% score on sentence one, speaker one, was due to an

Table VI

15-Class Problem, Discrete Prototypes,
Unmodified Recognition Program

Sentence	Speaker	Type of Speech	Location	Identification
1	A	Discrete	92.3%	92.3%
3	A	Continuous	61.1%	44.4%
5	B	Discrete	46.2%	23.1%
7	B	Continuous	26.7%	20.0%
9	C	Discrete	61.5%	53.8%
11	C	Continuous	38.9%	27.8%
13	D	Discrete	38.9%	33.3%
15	D	Continuous	22.2%	22.2%

Overall: Same Speaker: Location-74.2%, Identification-64.5% (Sentences 1 & 3 Avg)

All Speakers: Location-46.8%, Identification-38.1% (Sentences 1-15 Avg)

Sentence Spoken: "Kirk here, beam me up, Scotty."

incorrect phoneme length given to the recognition program. When the error is taken into account, the location and identification rate becomes 100%.

The performance degrades noticeably when the same speaker recites the sentence at a different rate. The performance degrades even more when different speakers were tested. The overall identification attained by the same speaker was 64.5%, while the identification attained for all four speakers was 38.1%.

These results are essentially consistent with established data in that the recognition program performs in a competent manner only if it is "trained" with a specific speaker for specific words. This type of performance is inadequate for any generalized speech recognition system.

15-Class Problem (Averaged Prototypes)

The prototypes used in the first portion of the results were then modified in the manner described in Section IV. Representative sections of sentences spoken by speaker A and speaker B were chosen to contain like phonemes. Both the discrete and continuous sentences were used. Scoring was accomplished in the same manner as described previously. The results are listed in Table VII with individual sentence performance listed in Appendix C.

Analysis

As was expected, the averaging scheme lowered the performance on sentence one. This was as expected since the

Table VII

15-Class Problem, Averaged Prototypes,
Unmodified Recognition

Sentence	Speaker	Type of Speech	Location	% Improve	Identification	% Improve
1	A	D	78.8%	-15.5	50.0%	-45.8
3	A	C	83.3%	+36.4	72.2%	+62.6
5	B	D	72.2%	+56.3	38.9%	+68.4
7	B	C	100%	+274	77.8%	+289
9	C	D	80.0%	+30.1	53.3%	-0.9
11	C	C	66.7%	+71.5	27.8%	0
13	D	D	55.6%	+42.9	33.3%	0
15	D	C	77.8%	+250.5	38.9%	+75.2

Overall Performance: Same Speaker: Not Applicable

All Speakers: Location-76.6%, Identification-48.9%

prototypes were no longer formed by this sentence. However, the location and identification scores show a remarkable improvement. In the case of sentence 7, the location rate jumped from 26.7% to 100%. Although the identification scores also improved in a similar manner, the overall scores were much too poor to be considered acceptable. In addition, the location performance was extremely difficult to tabulate due to the extremely complex manner in which the location was tabulated in the unmodified programs. It was at this point that the modifications listed in Section V were introduced. These modifications necessitated an amended scoring system which is discussed in the next section.

26-Class Problem

The prototypes used for the 26-class problem were extracted from the two sentences and averaged as described in Section IV. The prototypes as used are listed in Table V.

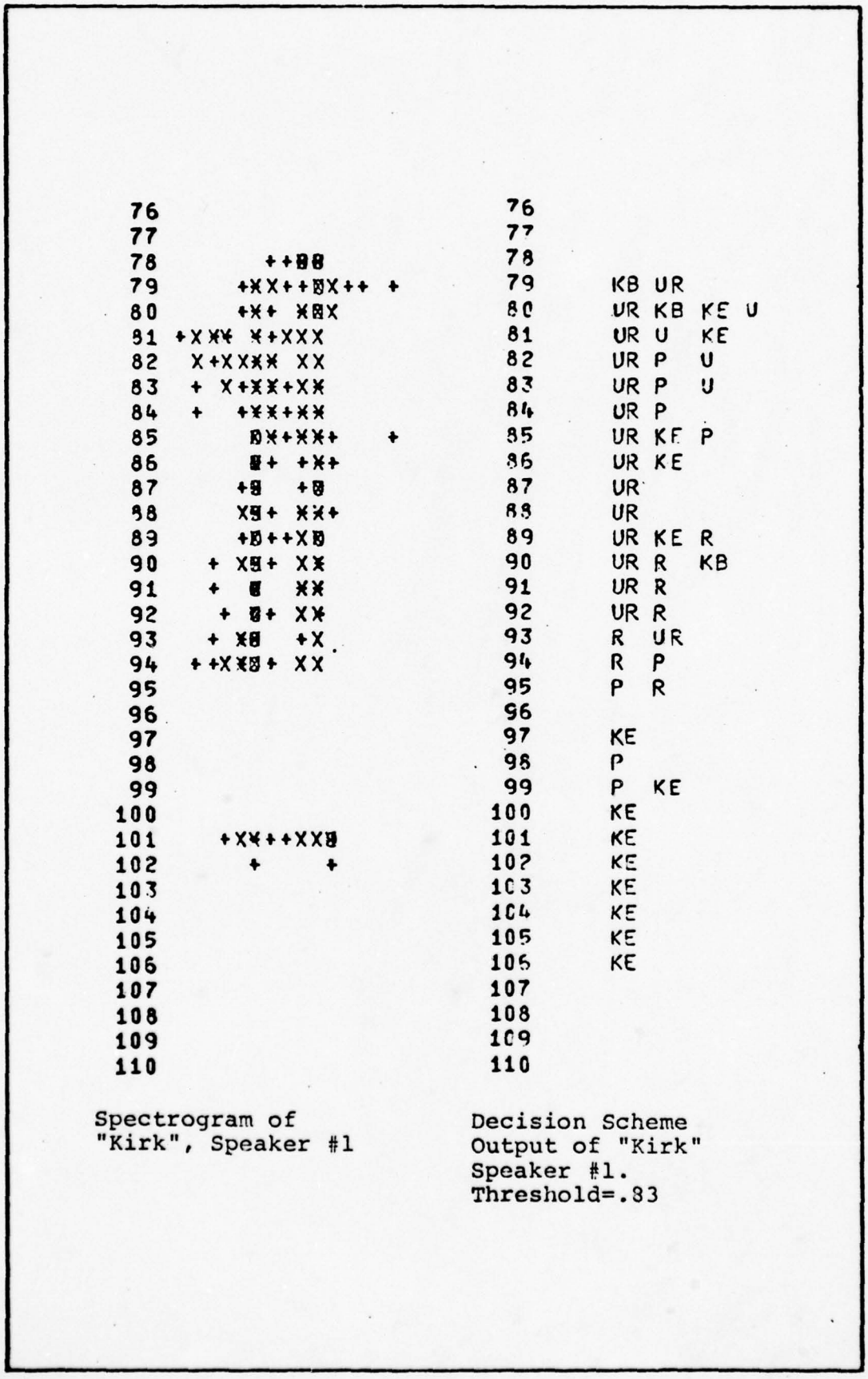
Preliminary scoring with the extended prototype set revealed that the existing program decision scheme functioned in an extremely poor manner. The existing program, prior to modification, yielded scores which were consistently below 30% for all sentences. However, analysis of actual prototype correlation values showed that the overall correlation scheme was still functioning in an accurate manner. It was decided to change the methods of presenting the

correlation data so that analysis and scoring would be made easier. Section V deals with the revised decision scheme which was used in this section.

When the decision scheme was revised, it became necessary to change the methods of scoring phonemic selections.

As has been noted in Section VI, the output of the final decision scheme is in the form of a ranked phonemic output of the test sentence. Each time increment contains the top five choices of the correlation program which fulfilled the requirements of the decision scheme. See Figure 16 for an example of the decision scheme output.

The rules governing location and identification were modified to fit this revised decision scheme as follows. A phoneme was considered to be located if it ranked within the top three choices of the phonemic output. It was labeled as identified if it was ranked as the first choice. In this manner, there was a close similarity between the revised ranking and the original decision output. There is obviously still a discrepancy between the performance of the 26-class problem when compared to the two 15-class problems. However, the goal of this extended look into the phoneme averaging process was to insure that the preliminary results obtained were not just a function of the particular data used. In addition, the improved and more



Spectrogram of "Kirk", Speaker #1

Decision Scheme Output of "Kirk" Speaker #1. Threshold=.83

Figure 16. Decision Scheme Output

versatile decision scheme which came about because of the problems encountered in this phase of the research will be useful in subsequent inquiries into the nature of phonemic speech recognition.

The results of the 26-class problem are listed in Table VIII and IX.

Analysis

The results of the 26-class problem are somewhat difficult to interpret due to the limited amount of data collected along with the lack of a suitable baseline with which to compare the revised identification and location techniques. However, the following facts can be noted.

1. The location score for both 15-class and the 26-class problems are identical. This means that the correlation performance of the prototypes remained such that the correct prototypes still scored within the top three highest ratings. In essence, the increased decision space accorded by the increased prototype size does not seem to degrade the overall location performance.

2. The identification scores cannot be compared directly to the previous scores due to the revised methods for identification. All four sentences scored showed notable improvements, but it must be remembered that an identification meant only scoring the highest of any prototypes located. The previous identifications were the result of a much more restrictive scheme (Ref 5).

Table VIII

26-Class Problem, Averaged Prototypes,
Modified Recognition

Sentence	Speaker	Type of Speech	Location	Identification
1	A	Discrete	77.8%	55.5%
3	A	Continuous	83.3%	83.3%
9	C	Discrete	80.0%	76.9%
11	C	Continuous	72.2%	66.7%

Note: Only four sentences scored due to computer time available.

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentence Spoken: "Kirk here, bring me up, Scotty."

Table IX

26-Class Problem, Averaged Prototypes,
Modified Recognition

Sentence	Speaker	Type of Speech	Location	Identification
2	A	D	87.5%	68.8%
14	D	D	56.3%	43.8%

Note: Only two sentences scored due to computer time available.

Sentence Spoken: "Quoth the Raven, nevermore."

3. Two test sentences from the second sentence data set were also scored. Although this is a very small data set, the results seem to indicate that the correlation performance remains consistent with different sentences.

VIII. Conclusions

This research had two basic goals:

1. To evaluate existing methods of phoneme recognition and implement improvements in the system with respect to multiple speaker and continuous speech performance; and

2. Modify the existing recognition programs so that they might be more versatile and serve as input to a more sophisticated recognition system.

It is felt that both goals have been obtained.

The original recognition system was acceptable when it had as its input sentences spoken by the same person who created the prototypes. The introduction of multiple speakers and varying speech speeds showed that the system was highly unreliable. The phoneme prototype averaging idea was an attempt to move the prototype vectors closer to some imaginary center of a hypothetical hypersphere so that they would serve at a more universal prototype set. The improvements were apparent in both the 15 and the 26-class problem.

Obviously, even the 26-class prototype set is not a complete set. It serves, however, to illustrate that it is possible to still achieve acceptable recognition and location scores even though the decision space has been almost doubled. This relative insensitivity to mal-effects of an increased decision space supports the belief that there is

possibly a universal speech recognition machine which would have as its first step a correlation-based phoneme recognizer.

The modifications to the decision scheme are ones which greatly inhance the versatility of the program. The final phonemic output can be stored for futher processing, and the output as is printed is suitable for side-by-side analysis with the spectrogram data. In addition, the graph option allows future researchers to have a pictoral representation of the performance of each phoneme prototype. This will allow insight into the reasons for the poor performance of certain prototypes along with a basis for improving the entire prototype set.

The speech recognition program, as it now exists, is a versatile, easily changed, speech phoneme analysis system. This program can serve as the first portion of a multi-segmented speech recognition system involving grammer, syntax, and context programs.

IX. Recommendations

There are two classes of recommendations which are listed below. Class one deals with methods for phoneme preparation, analysis, and correlation. Class two deals with other modifications which would allow the user to have greater insight into program performance.

Class I

1. Use the spectrogram program (GS3) to produce visual outputs of the prepared prototypes. This feedback process will enable researchers to examine the effects of prototype averaging to help form prototype sets which are more clearly distinct.

2. Make future prototype lengths as short as possible. Although shorter prototypes have more "false alarms", there seem to be problems obtaining consistent performance when longer prototypes are used.

3. Investigate the use of Fourier-Space filtering to help the decision process. Although the filter has been designed and tested, no actual data was obtained using filtered results due to time limitations and problems with the correlation normalization factor. (See item 4 in the Class II recommendations.)

4. Investigate the possibilities of reducing the frequency response of the input data set. It may not be necessary to use all frequencies from 100 to 5 khz to

initiate correct recognition. The speech bandwidth may be able to be further restricted without hindering overall results.

Class II

1. Investigate the possibilities of constructing and using a device which could reconstruct actual sounds from the phonemic renditions of the decision scheme. The use of this device would allow the researcher to obtain an audio feedback of what the correlation program has chosen as the phonemic output of the sentence. Such a device is currently being designed by these researchers and shows great promise.

2. Expand the FFT subroutine in the correlation program (GS5) so that the entire sentence can be transformed at one time. This would greatly reduce the computer time required by reducing the number of computations required. It would also lessen the problems which are incurred by incorrect correlation values near the edges of the present sentence arrays. This problem was accepted as a necessary evil which was offset by the advantages to be gained by having an entire sentence correlation array available for the final decision program.

3. Install a threshold routine which would zero elements of the sentence array when there is no speech present. This would serve to remove many "false alarms" which occur when prototypes correlate with "noise". Neyman (Ref 8) used a threshold technique similar to the

one mentioned above. It was removed by Hensley (Ref 5) and all data in this paper was obtained without any thresholding whatsoever. Although the overall problem did not seem to be serious, the correlation technique is not at a stage where small improvements can increase accuracy.

4. Investigate a revised correlation normalization procedure which could be used when Fourier-Space filtering is utilized. Preliminary research with the filter indicates that it does indeed function in the desired manner. However, the maximum correlation values are not being limited to a maximum of one as they should be. It will be necessary to develop an algorithm which will recompute the correlation normalization factor as some function of the percentage of energy removed by the filter.

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. Daily, Keith G. and Franker S. Sutton. 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).

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

5. 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).

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

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

8. 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).

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

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

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

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

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

APPENDIX A

Data Processing Charts and Notes

A. Data Processing Charts and Notes

This appendix contains the flow charts which give an overview of the operation of the seven main segments of the speech recognition system. Also included are notes which clarify important operating points of each program. Listed below are the seven programs along with the associated inputs and outputs. Subsequent pages contain the flow chart for each program along with the notes concerning its operation.

Table X

Data Processing Programs

Name	Input	Output
GS1 (Main)	L-Tape	PF#1
GS2 (Octave1)	PF#1	PF#2 Spectrogram
GS3 (Octave2)	PF#2	Averaged Spectrogram
GS4 (Proavg)	PF#2	PF#3 (Averaged prototypes)
GS5 (Crscor)	PF#2 PF#3	PF#4 (Correlation arrays)
GS6 (Corgph)	PF#4	Calcomp Graphs
GS7 (Decis)	PF#4	Phonemic Output

Note: PF refers to Permanent Files on the CDC system. These files can be output as cards if the user so desires.

GS1 (Main)

Program GS1 (Main) is used to read data from the L-Tape which was produced by the ASD Computer Center and write it on a permanent file (PF). This PF is used in subsequent processing. The program attaches the L-Tape under the name of Tape1. It reads the tape and transfers it to a program file called Tape2. Following completion of the transfer, the system catalogs the Tape file and gives it a new name of SENT1. The new name is entirely the choice of the user. The storage space required by this program is quite large. If there is insufficient space to store the results, program GS2 can be modified to use the L-Tape as its input.

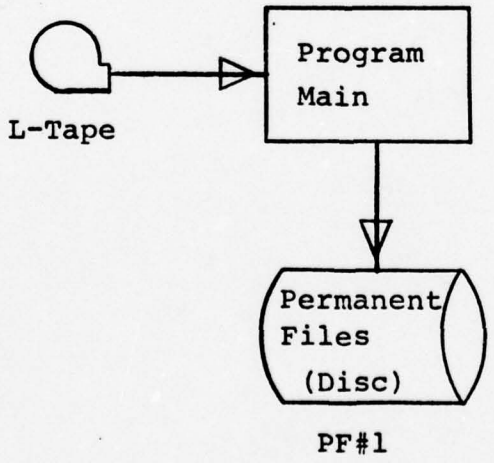


Figure 17. Flow Diagram for GS1 (Main)

GS2 (Octavel)

GS2 (Octavel) uses the results of GS1 as input data. The program attaches PF#1 created by GS1 and gives it a local file name of Tape1. It then reads this file and logarithmically compresses the data from 64 to 16 channels. The reduced data is stored on a local file called Tape2 and is stored for use as PF#2 by subsequent portions of the program. Two variables which are important in this program are NREC and NN2. NREC represents the number of files to be read. A file is one entire speech segment. NN2 should be set as a number which is a value one more than the number of records in that particular set. For example, if ASD Center states that the sentence data had 480 records, set NREC to 481. This will insure that the data is handled correctly. Tape2 is stored as PF#2 under the name SENT2 and SENT3.

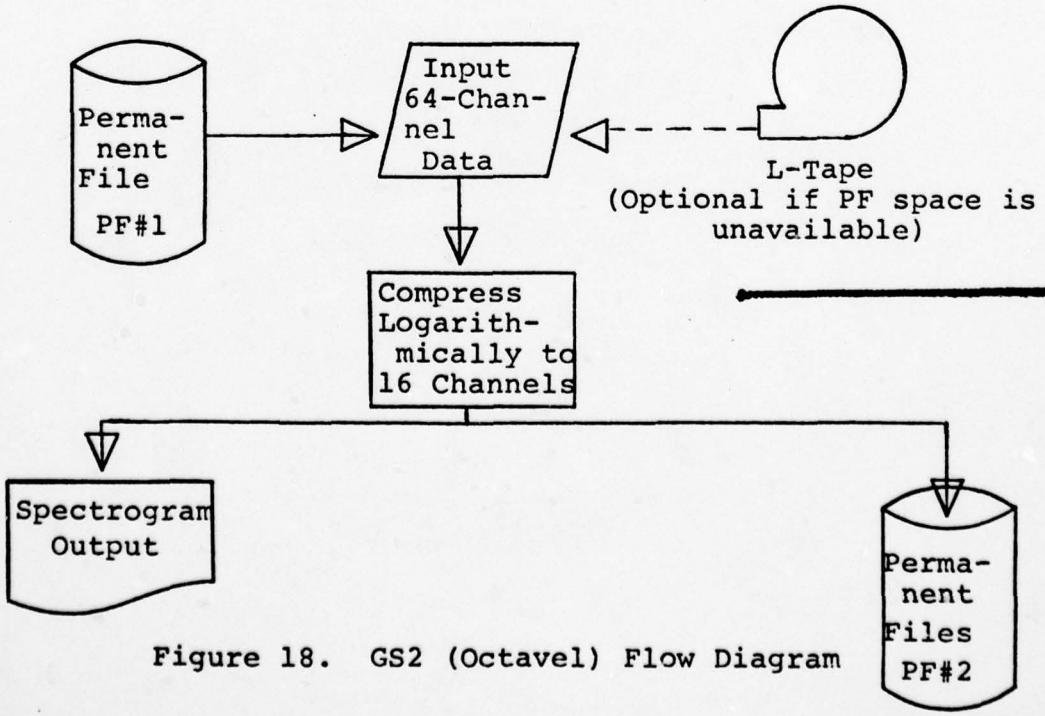


Figure 18. GS2 (Octavel) Flow Diagram

GS3 (Octave2)

GS3 (Octave2) uses as its input the data stored by GS2 (PF#2) and produces a normalized spectrogram. This program does not create any permanent files. Its only purpose is to produce a spectrogram which is more easily interpreted than is the one produced by GS2 (Octavel). 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 only those portions of the sentence record which are desired. The entire sentence will be read, but only the desired portions will be output as spectrograms.

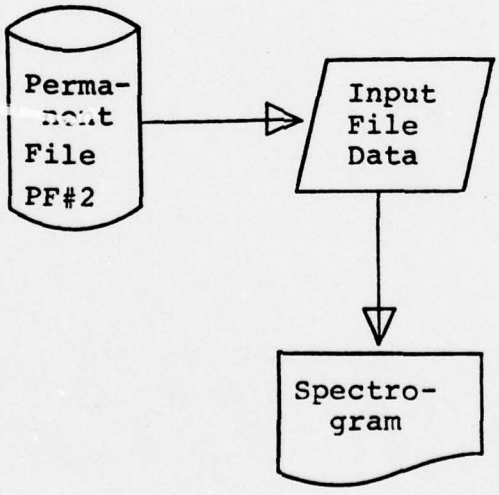


Figure 19. Program GS3 (Octave2) Flow Diagram

GS4 (Proavg)

Program GS4 (Proavg) is used to create a permanent file (PF#3) of averaged phonemes. This PF is attached by GS5 and used to correlate with the input sentence data.

To run the program, the sentences from which the phonemes are being made must be visually analyzed and the locations of the desired phonemes noted. The program then uses these notations along with the attached sentence data to produce the prototypes. The location of each phoneme is to be placed on data cards which will be read into the IBEGIN and IEND arrays.

The manner in which data was collected required that every other sentence be read. This is the reason for the skip functions within the program. Data PROREC and PRORET are used to determine which sentence is being analyzed on each run through the averaging portion of the program. Data NUPROC and NUPROT are used to tell the length of each set of prototypes being averaged. A "set" is a collection of prototypes of the same sound.

The program first reads the IBEGIN and IEND data. The first sentence to be used is selected. The phonemes are read and written on Tape2 based on the values of IBEGIN and IEND. Phonemes from like sentences are successively selected based on PROREC and PRORET and written on Tape2. Thus, selected groups of like phonemes are then written together on Tape2.

Tape2 is then read and each set of phonemes to be averaged is summed and then divided by the number of sentences (in this case, four). The averaged prototypes are written on Tape4 and followed by an end of file statement (EOF).

To complete the prototype set, the initial conditions are reset. Tapel and Tape2 are rewound and the averaging scheme is repeated with the next group of prototypes being written on Tape4 following the previous prototypes. Tape4 is then stored as PF#3 under the name PROAVG.

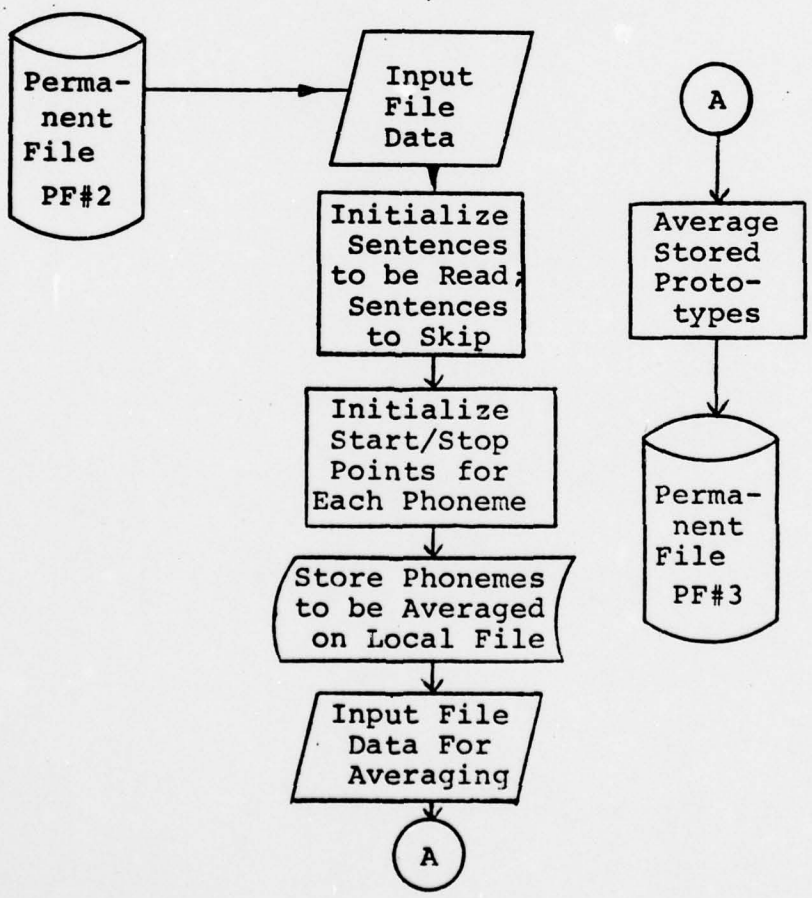


Figure 20. GS4 (Proavg) Flow Diagram

GS5 (Crscor)

This program comprises the main body of the research. GS5 (Crscor) consists of a main program (Crscor) which inputs the necessary variables for correlation, such as normalization desired (NORM), filtering (IFILT), prototype lengths (ITYP), etc. Following the initialization of desired variables, the main program calls a subroutine (SCORR) which handles the correlation computations. The variables are clearly documented in the program listing found in Appendix B.[1] This program has as its output PF#4 which consists of correlation values of all prototypes over a selected length of input data. PF#4 is named CORR. The program list will consist of prototype values as read in, along with the normalized values if desired. The program will also print out information on the subdivision of the sentence, number of zeroes required to augment the data arrays, and prototype length. Each phase of data processing is clearly labeled so as to make it easier to insert changes and revisions.

[1]The sentence Data (PF#2) is attached as Tapel, while the prototype Data (PF#3) is attached as Tape6.

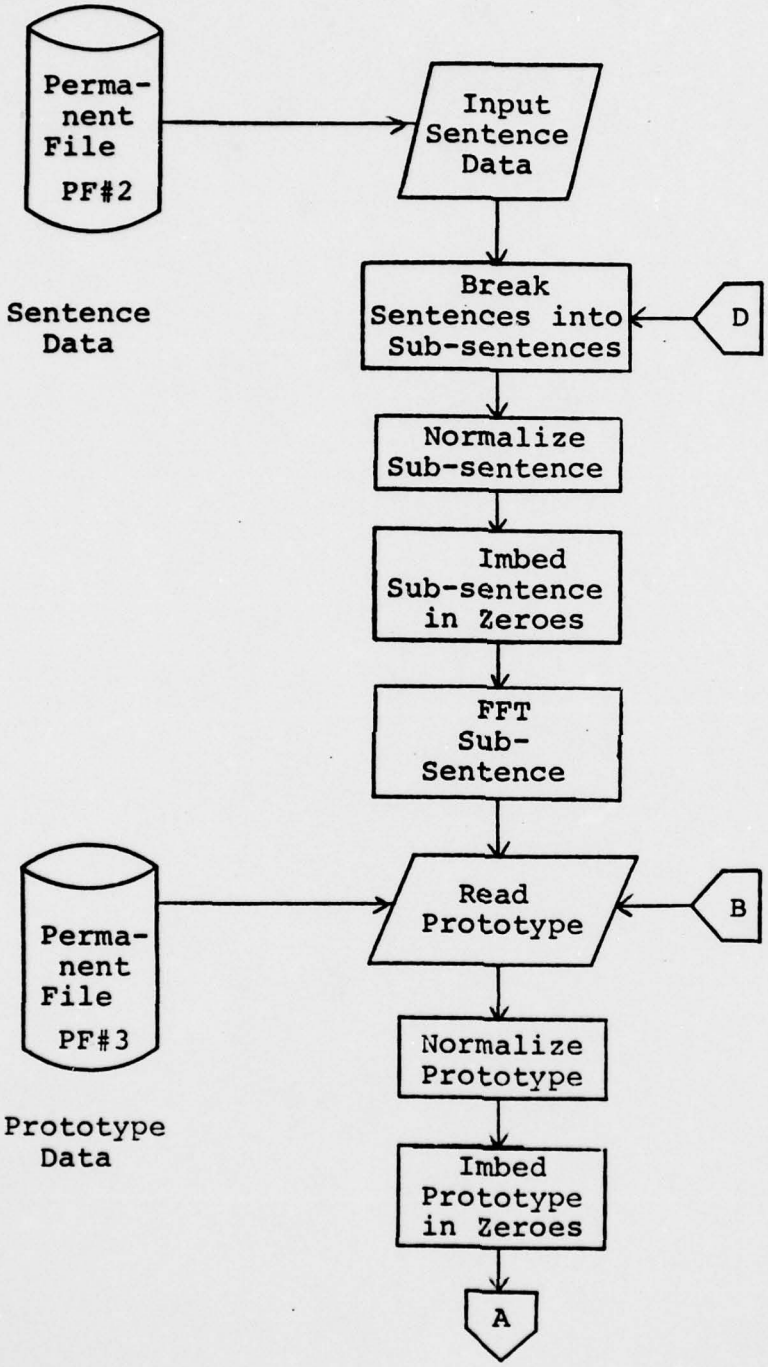


Figure 21. Program GS5 (Crscor) Flow Diagram
(Plate 1)

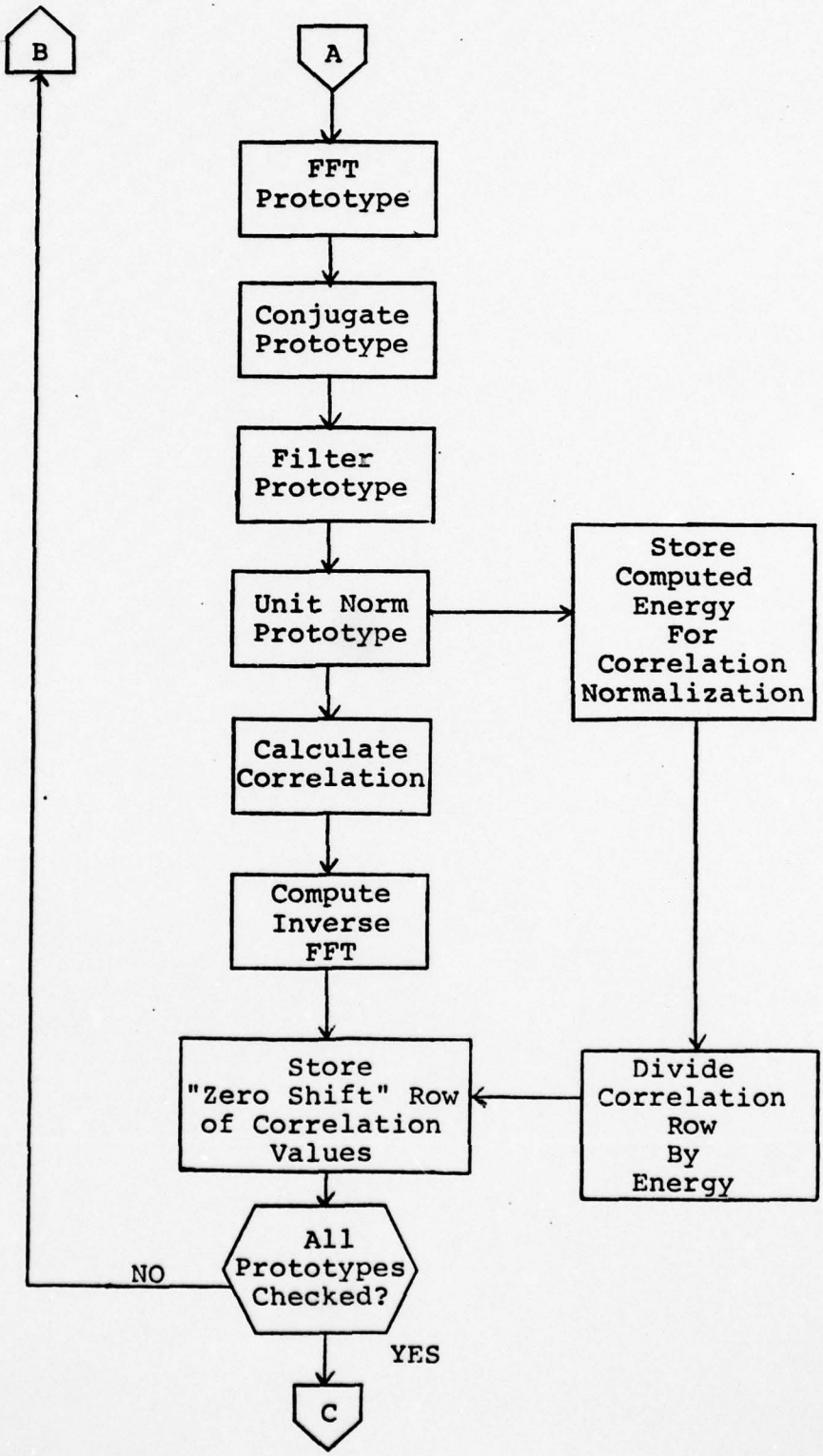
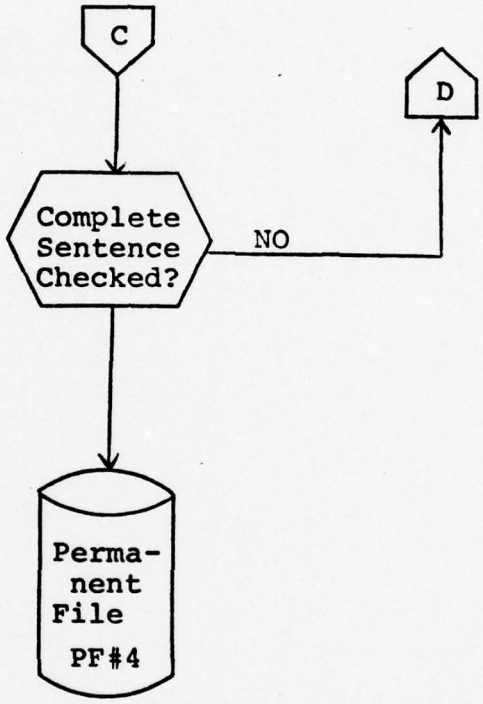


Figure 22. Program GS4 (Crscor)
(Plate 2)



Stored Correlation Arrays

Figure 23. Program GS5 (Crscor) (Plate 3)

GS6 (Corgph)

GS6 (Corgph) uses PF#4 as its input. It reads selected (by user) 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 programs 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 is a page listing of four correlation outputs for the entire length of one sentence. The labels of the axis of the graphs are controlled within GS6 and should be changed according to what prototypes are output.

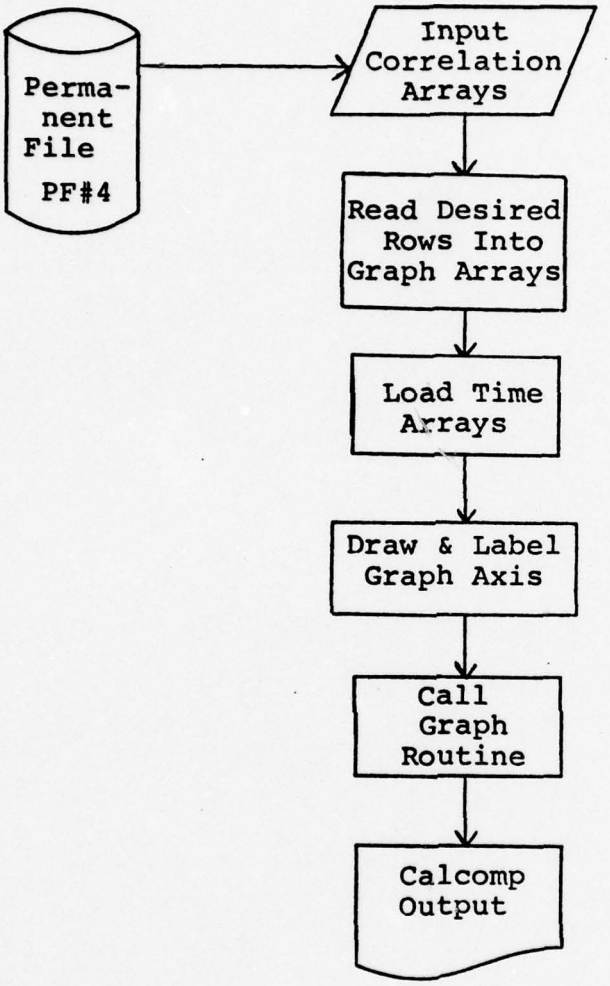


Figure 24. Program GS6 (Corgph) Flow Diagram

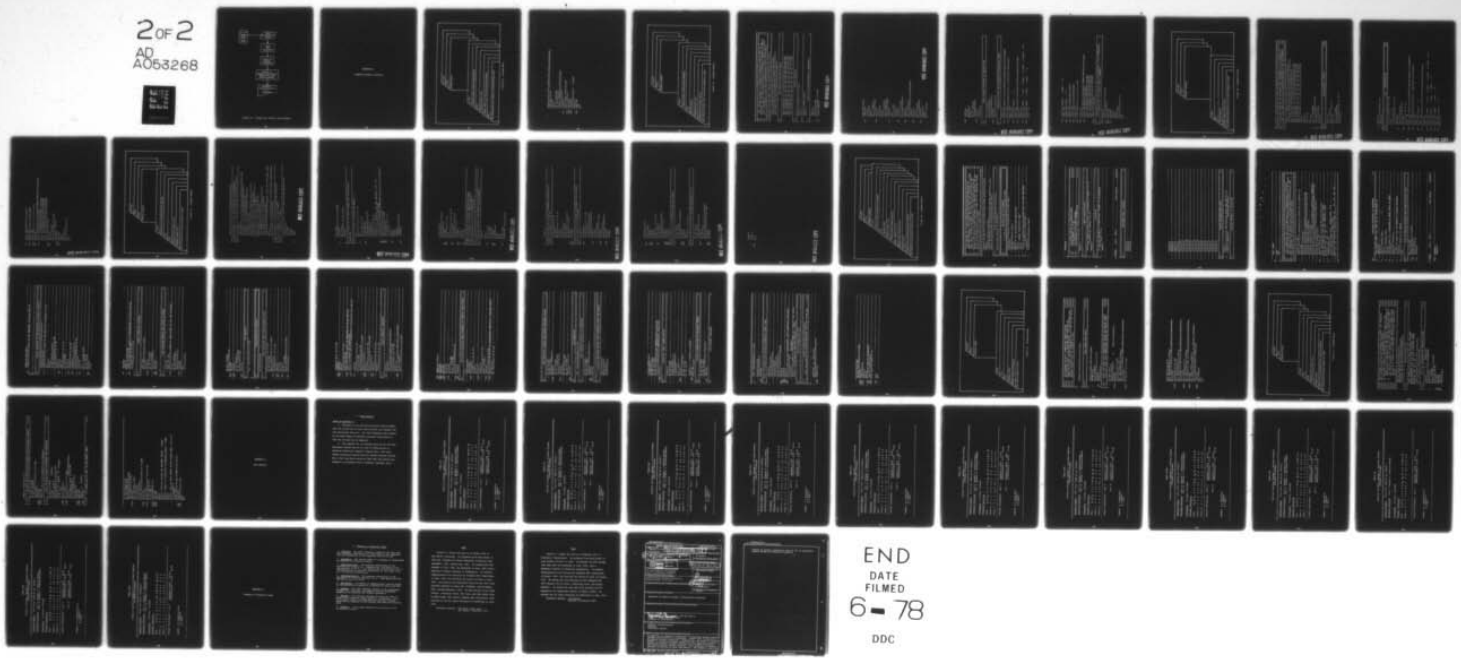
GS7 (Decis)

GS7 (Decis) attaches PF#4 and processes the correlation arrays according to the methods described in Section VI. The input arrays which contain information on the phoneme names and length must be altered for each new set of phonemes. The variables ENDUR and THRHLD are the endurance (time) and correlation threshold values, respectively. This program has as an output the list of the phonemic representation of the sentence. It is possible to store these results in permanent files if desired in order to present these processed arrays to a future higher-order decision scheme.

AD-A053 268 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 17/2
COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH.(U)
NOV 77 M F GUYOTE, P L SISSON

UNCLASSIFIED AFIT/GE/EE/77D-18 NL

2 OF 2
AD
A053268



END
DATE
FILMED
6-78
DDC

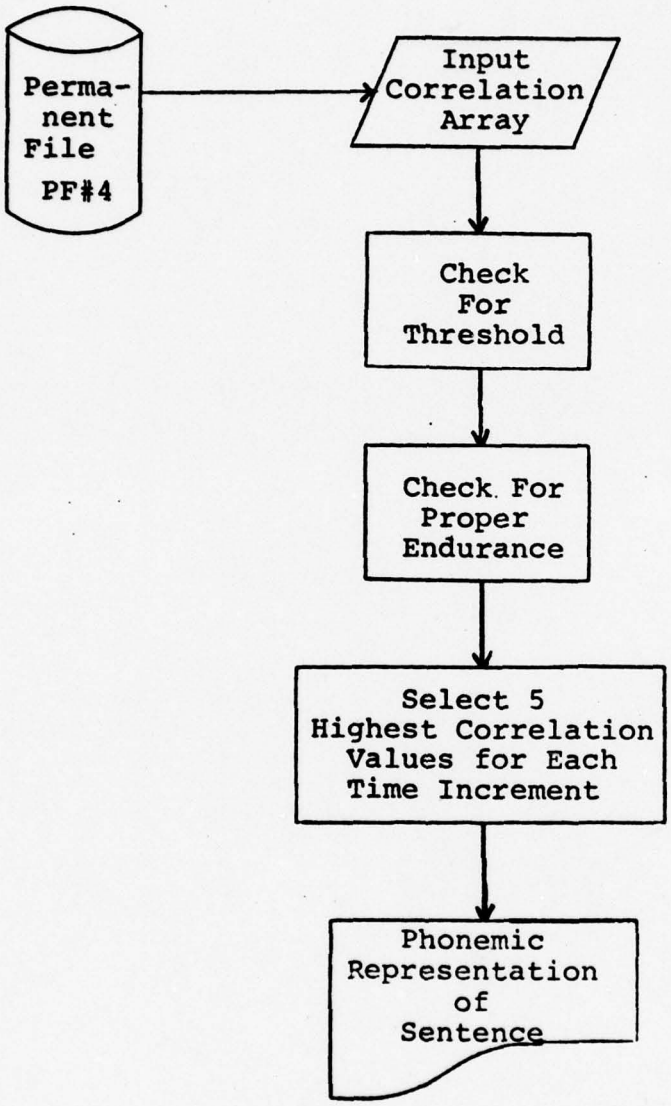


Figure 25. Program GS7 (Decis) Flow Diagram

APPENDIX B

Computer Program Listings

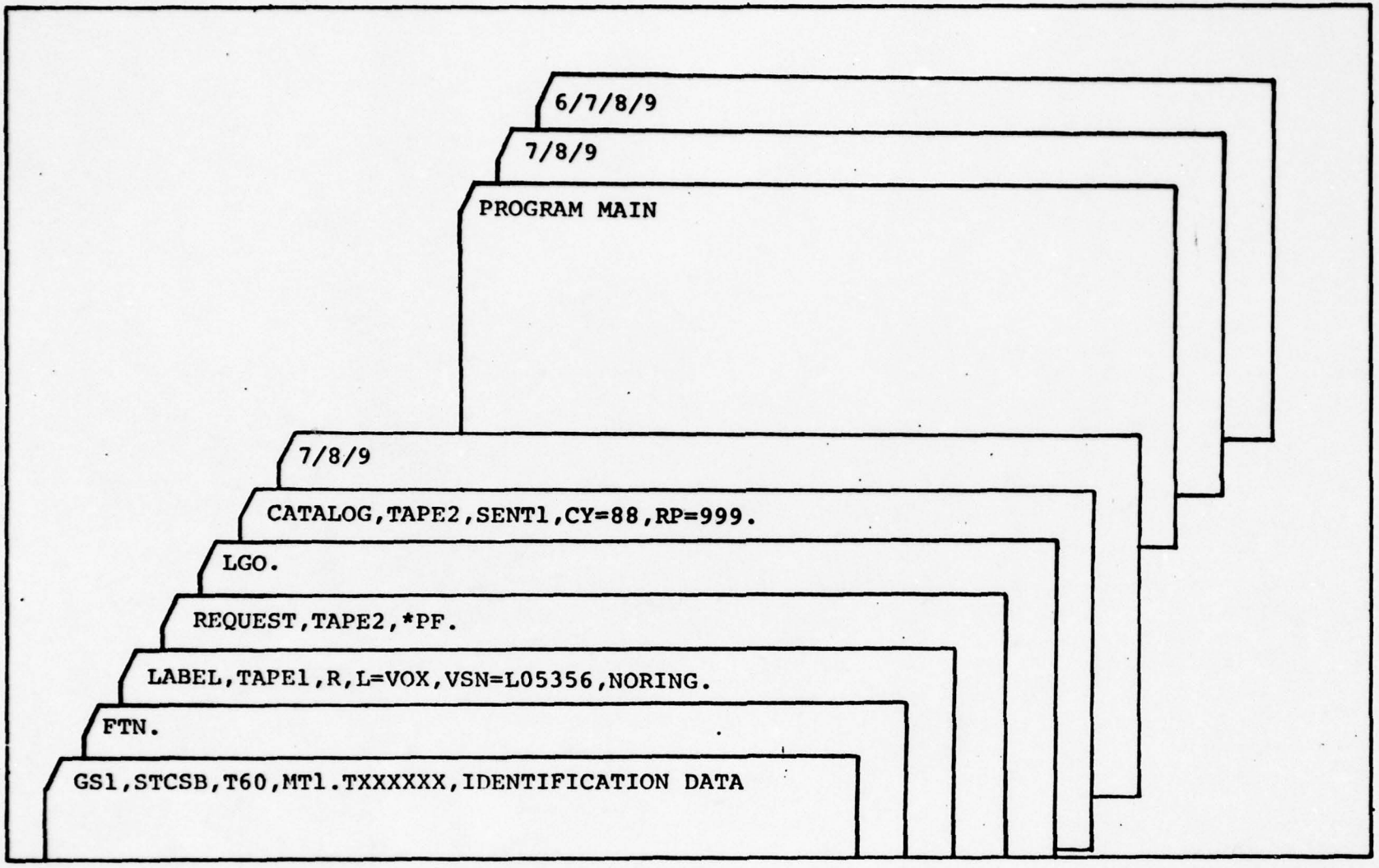


Figure 26. Program Main

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE1,TAPE2,TAPE6=OUTPUT)
      DIMENSION A(70)
      IREC=16
      JK=0
      DO 500 KK=1,IREC
      DO 100 IK=1,481
      READ(1) (NCHAN,NDIM,(A(K),K=1,64))
      IF(EOF(1))200,15
 15   CONTINUE
      WRITE(2,20) (A(K),K=1,64)
 20   FORMAT(22F6.3)
 100  CONTINUE
 200  CONTINUE
      PRINT*,"EOF:IK,KK=    ",IK,KK
      ENDFILE2
 500  CONTINUE
      STOP
      END
```

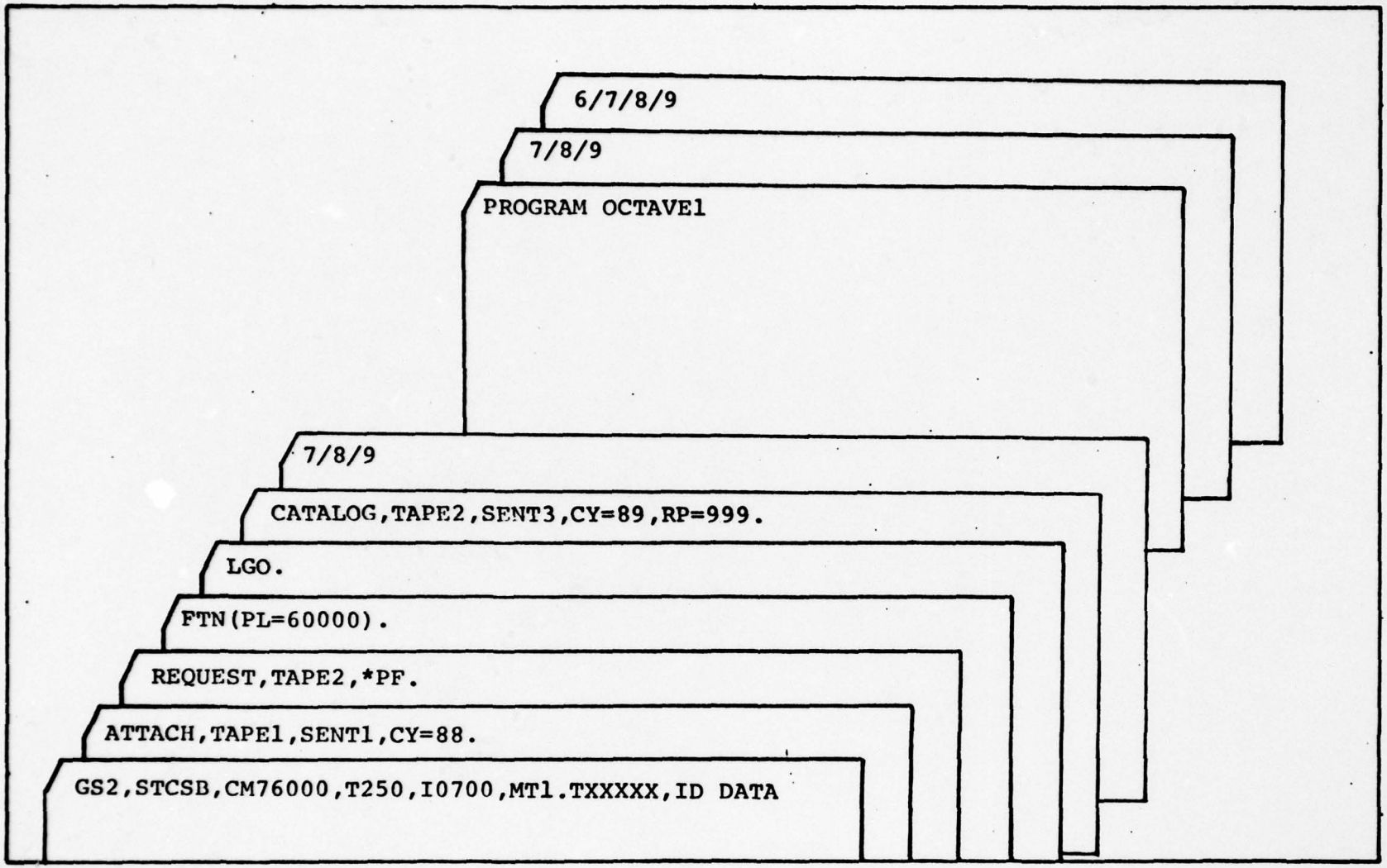


Figure 27. Program Octave1

```
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**********************************************************************
C THIS RUN READS THE FIRST 8 OF 16 RECORDS CREATED FOR THESIS.
C**********************************************************************
      PROGRAM OCTAVE(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=8
C----------------------------------------------------------------------
C     MAXIMUM RECORD LENGTH
      NN2=181
C--                INPUT ARRAY LOGARITHMICALLY COMPRESSED            --
C----------------------------------------------------------------------
      NN1=64
 1    CONTINUE
      DO 305 I=1,NN2
 5    READ(1,10)(A(J),J=1,NN1)
 10   FORMAT(22F5.3)
```

```
      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))
      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
```

```
  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(T.GT.1) GO TO 295
C-------------------------------------------------------------------------------
C--            COMPRESSED ARRAY AND ASSOCIATED SPECTROGRAM OUTPUT             --
C-------------------------------------------------------------------------------
       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("+",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,16F5.2,8X,I3,16A1)
      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 TRANSFERED       --
C--   TO PERMANENT FILE UPON COMPLETION OF PROGRAM.                           --
C--------------------------------------------------------------------------------
500   CONTINUE
      WRITE(2,315)(B(JJ),JJ=1,16)
315   FORMAT(16F5.3)
305   CONTINUE
310   CONTINUE
      ENDFILE2
      PRINT*
      PRINT*
      NREC=NREC-1
      IF(NREC.GT.0) GO TO 1
      STOP
      END
```

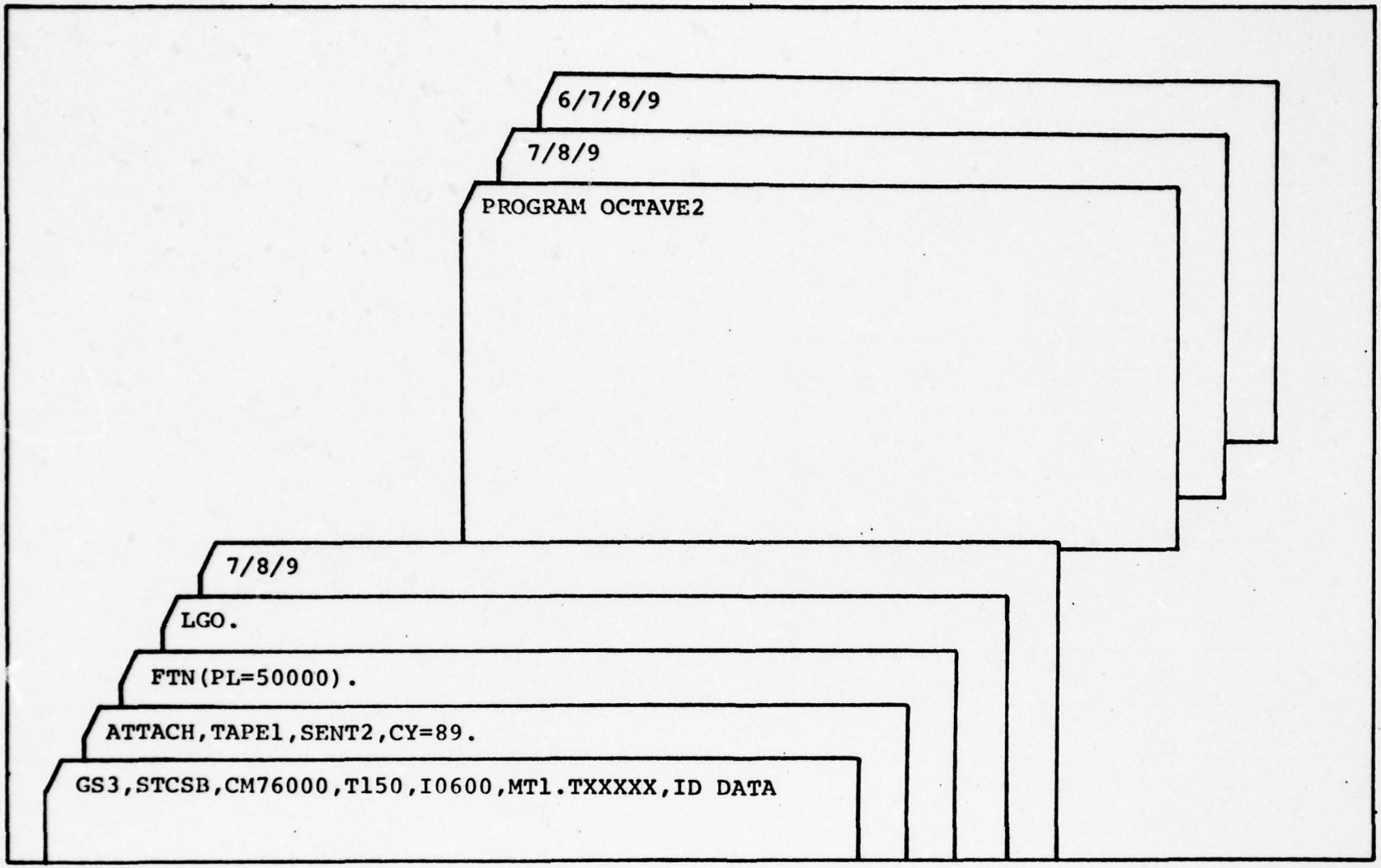


Figure 28. Program Octave2

```
C*********************************************************************************
C*********************************************************************************
C**THIS PROGRAM ATTACHES THE PERMANENT FILE CONTANING THE 16 CHANNELS OF      ****
C**    DIGITIZED DATA AND GIVES A NORMALIZED VERSION OF THE SPECTROGRAM.      ****
C*********************************************************************************
C*********************************************************************************
      PROGRAM OCTAVE(INPUT,OUTPUT,TAPE1,TAPE2,TAPE6=OUTPUT)
      DIMENSION SYMBOL2(10),SYMBOL3(10),SYMBOL4(10),SYMBOL5(10)
      DIMENSION B(16),SYMBOL1(10),BI(16),IBI(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=8
      NN1=16
      NSTART=1
      NSTOP=470
      L=0
 1    CONTINUE
      DO 305 I=1,NSTOP
 5    READ(1,10)(B(J),J=1,NN1)
 10   FORMAT(16F5.3)
      IF(EOF(1))310,30
 30   CONTINUE
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CXXXX                         NORMALIZATION  ROUTINE                       XXXXX
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      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
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CXXXX                                    END    NORMALIZATION                  XXXX
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      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 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 263
 263  FORMAT("+",96X,"   - ",2X,"   - ",2X,"   - ")
      PRINT 264
 264  FORMAT("+",103X,"   + ",2X,"   + ")
      PRINT 265
 265  FORMAT("+",110X,"   * ")
      PRINT 270
```

```
270   FORMAT(92X,*0000000011111111*)
      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,I3,16A1)
      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
```

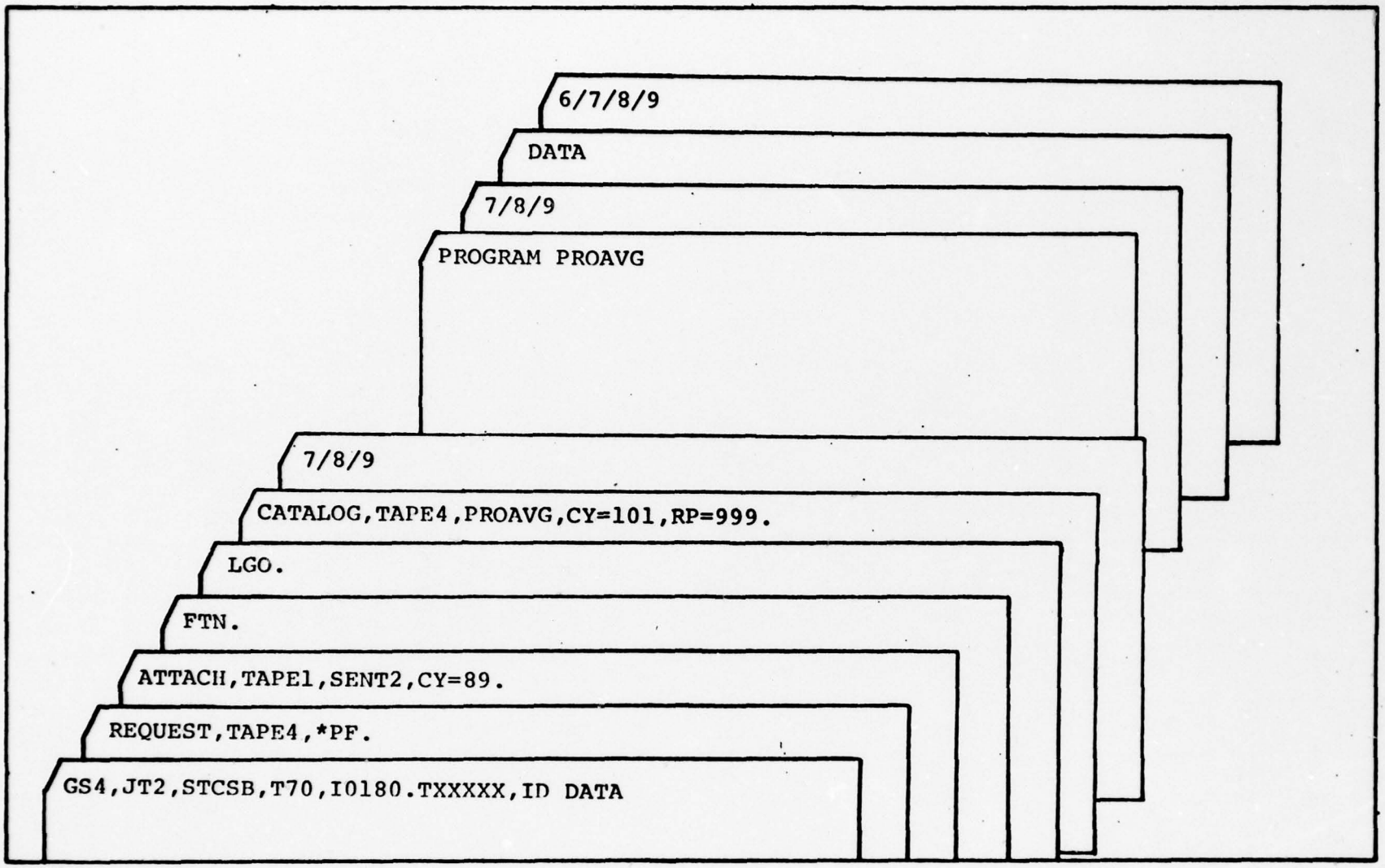


Figure 29. Program Proavg

```
      PROGRAM PROAVG(INPUT,OUTPUT,TAPE1,TAPE2,TAPE4)
C-----------------------------------------------------------------------
C PROGRAM PROAVG CREATES AN AVERAGED SET OF PROTOTYPES
C ON A SEPERATE TAPE FOR ACCESS BY THE CORRELATION PROGRAM.
C-----------------------------------------------------------------------
      REAL NEWPRO
      DIMENSION IBEGIN(4,15),IEND(4,15),SYMBOL1(15),SYMBOL2(15)
      DIMENSION ARR(50,16),SUM(50,16),NEWPRO(50,16)
      DIMENSION PROREC(8),NUPROT(15),A(16)
      DIMENSION PRORET(8),SYMBOL3(11),SYMBOL4(11)
      DIMENSION NUPROC(11)
      DATA SYMBOL1/2HKB,2HUP,2HKE,2HH ,2HI ,2HR ,2HB ,
     A2HIF,2HE,2HM3,2HU ,2HP ,2HS ,2HSH,2HT /
      DATA SYMBOL2/4HKICK,4HUP  ,4HKICK,4HEM ,4HTT  ,
     A4H2UN ,4HREG ,4HEEL ,4HMEN ,4HMEN ,
     A5HUNDER,5HBABE3,4HSEND,4HODD ,4HTOE /
      DATA PROREC/1,0,7,0,5,0,7,0/
      DATA NUPROT/8,9,2,7, 8,7,3,7,5,5,7,3,6,13,4/
      DATA NUPROC/7,8,7,10,7,6,4,6,6,7,6/
      DATA PRORET/0,2,0,4,0,6,0,8/
      DATA SYMBOL3/2HOU,2HOO,2HTR,2HTE,2HF ,2HA ,2HV ,2HEN,2HN ,2HEE,
     A2HER/
      DATA SYMBOL4/5HOUGTH,4HOPE,4HTHIN,4HTHEN,4HMAKE,4HRAVE,3HVOW,
     A6HSUDDEN,4HNOSE,3HRED,5HNEVER/
C-----------------------------------------------------------------------
C SET THE NUMBER OF RECORDS NOT CONTAINING PROTOTYPES
C TO ZERO IN DATA PROREC AND PRORET.
C N IS THE NUMBER OF RECORDS.
C-----------------------------------------------------------------------
      PRINT*," PROTOTYPE VECTORS SELECTED AND AVERAGED FOR
     ATHESIS USE."
      LNGTH=15
      IFIN=0
      ETH=16
      MM=1
1     NH=8
```

```
      DO 3 I=1,4
      READ*,(IBEGIN(I,J),J=1,LNGTH)
3     CONTINUE
      DO 4 I=1,4
      READ*,(IEND(I,J),J=1,LNGTH)
4     CONTINUE
C-----------------------------------------------------------------------
C L IS THE NUMBER OF THE RECORD BEING READ SEQUENTIALLY
C NUMBERED.
C-----------------------------------------------------------------------
C MM IS THE PROTOTYPE LOCATION ON RECORD L.
      L=1
      IF(PROREC(1).EQ.0) GO TO 55
5     K=1
      LIM=IEND(MM,LNGTH)
      DO 40 I=1,LIM
10    FORMAT(16F5.2)
      READ(1,10)(A(J),J=1,16)
      B=IBEGIN(MM,K)
      C=IEND(MM,K)
      IF(I.LT.B) GO TO 40
      IF(I.NE.B) GO TO 20
      PRINT*," THE PROTOTYPE DATA FOR PROTOTYPE ",K," IS: "
      PRINT 15,SYMBOL1(K),SYMBOL2(K)
15    FORMAT(1X,"THE PROTOTYPE REPRESENTS",1X,A2,1X,"AS IN(",A5,")")
20    WRITE(2,10)(A(J),J=1,16)
28    PRINT 29,(A(J),J=1,16)
29    FORMAT(16F5.2)
      IF (I.EQ.C) GO TO 30
      GO TO 40
30    K=K+1
      PRINT*
      PRINT*
      ENDFILE2
      PRINT*,"EOF WRITTEN"
40    CONTINUE
```

```
      DO 45 I=1,700
      READ(1,10)(A(J),J=1,16)
      IF(EOF(1))50,45
   45 CONTINUE
   50 IF(L.EQ.NN)GO TO 75
      L=L+1
      IF(PROREC(L).NE.0)GO TO 70
   55 DO 60 N=1,481
      READ(1,10)(A(J),J=1,16)
      IF(EOF(1)) 60,60
   60 CONTINUE
   70 MM=MM+1
      GO TO 5
   75 CONTINUE
C-----------------------------------------------------------------------
C TAPE2 NOW HAS THE PROTOTYPES WRITTEN IN ORDER
C FROM EACH RECORD OF INTEREST WITH AN EOF AFTER
C EACH.  THE NEXT PART OF THE PROGRAM AVERAGES
C THE PROTOTYPES AND WRITES THEM ON TAPE4.
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C L EQUALS THE NUMBER OF PROTOTYPES TO BE AVERAGED.
C-----------------------------------------------------------------------
      REWIND2
      L=4
      LNG=1
   76 DO 79 I=1,50
      DO 79 J=1,16
      SUM(I,J)=0
   79 CONTINUE
   77 K=NUPROT(LNG)
      M=K+1
      KK=1
      LL=0
   78 DO 85 I=1,M
      READ(2,10) (A(J),J=1,16)
```

```
      ENDFILE4
C-----------------------------------------------------------------------
C TAPE4 NOW HAS THE AVERAGED PROTOTYPES WRITTEN ON IT
C WITH EACH FOLLOWED BY AN EOF.
C-----------------------------------------------------------------------
      PRINT*,"EOF WRITTEN"
      LNG=LNG+1
      IF(LNG.EQ.FIN) GO TO 135
      REWIND2
      KK=LNG-1
      DO 134 MM=1,KK
      DO 133 NN=1,50
      READ(2,10)(A(J),J=1,16)
      IF(EOF(2)) 134,133
133   CONTINUE
134   CONTINUE
C-----------------------------------------------------------------------
C THIS SECTION REINITIALIZES THE PROGRAM FOR ANALYSIS
C OF THE SECOND SET OF SENTENCES.
C-----------------------------------------------------------------------
      GO TO 75
135   IFIN=IFIN+1
      IF(IFIN.EQ.2) GO TO 136
      DO 140 MI=1,11
      SYMBOL1(MI)=SYMBOL3(MI)
      SYMBOL2(MI)=SYMBOL4(MI)
140   CONTINUE
      LNGTH=11
      DO 150 MN=1,8
      PROBEG(MN)=PROBET(MN)
150   CONTINUE
      DO 160 MK=1,11
      NUPROT(MK)=NUPROC(MK)
160   CONTINUE
      FIN=12
      REWIND2
```

```
      IF(EOF(2)) 90,80
80    DO 81 J=1,16
      ARR(I,J)=A(J)
81    CONTINUE
85    CONTINUE
      LL=LL+1
      IF(LL.EQ.6) GO TO 106
90    DO 100 II=1,14
      DO 105 I=1,50
      READ(2,10)(A(J),J=1,16)
      IF(EOF(2)) 100,105
105   CONTINUE
100   CONTINUE
106   CONTINUE
C-----------------------------------------------------------------------
C THE ELEMENTS OF EACH PROTOTYPE ARE NOW SUMMED.
C-----------------------------------------------------------------------
      DO 120 I=1,K
      DO 115 J=1,16
      SUM(I,J)=SUM(I,J)+ARR(I,J)
115   CONTINUE
120   CONTINUE
      KK=KK+1
      IF(KK.LE.4) GO TO 78
C-----------------------------------------------------------------------
C THE SUM OF THE PROTOTYPE ELEMENTS IS DIVIDED BY L.
C-----------------------------------------------------------------------
      DO 125 I=1,K
      DO 125 J=1,16
      NEWPRO(I,J)=SUM(I,J)/L
125   CONTINUE
      DO 129 I=1,K
      WRITE(4,10)(NEWPRO(I,J),J=1,16)
      PRINT 29,(NEWPRO(I,J),J=1,16)
129   CONTINUE
130   CONTINUE
```

```
      REWIND1
      MM=0
      GO TO 1
136   STOP
      END
```

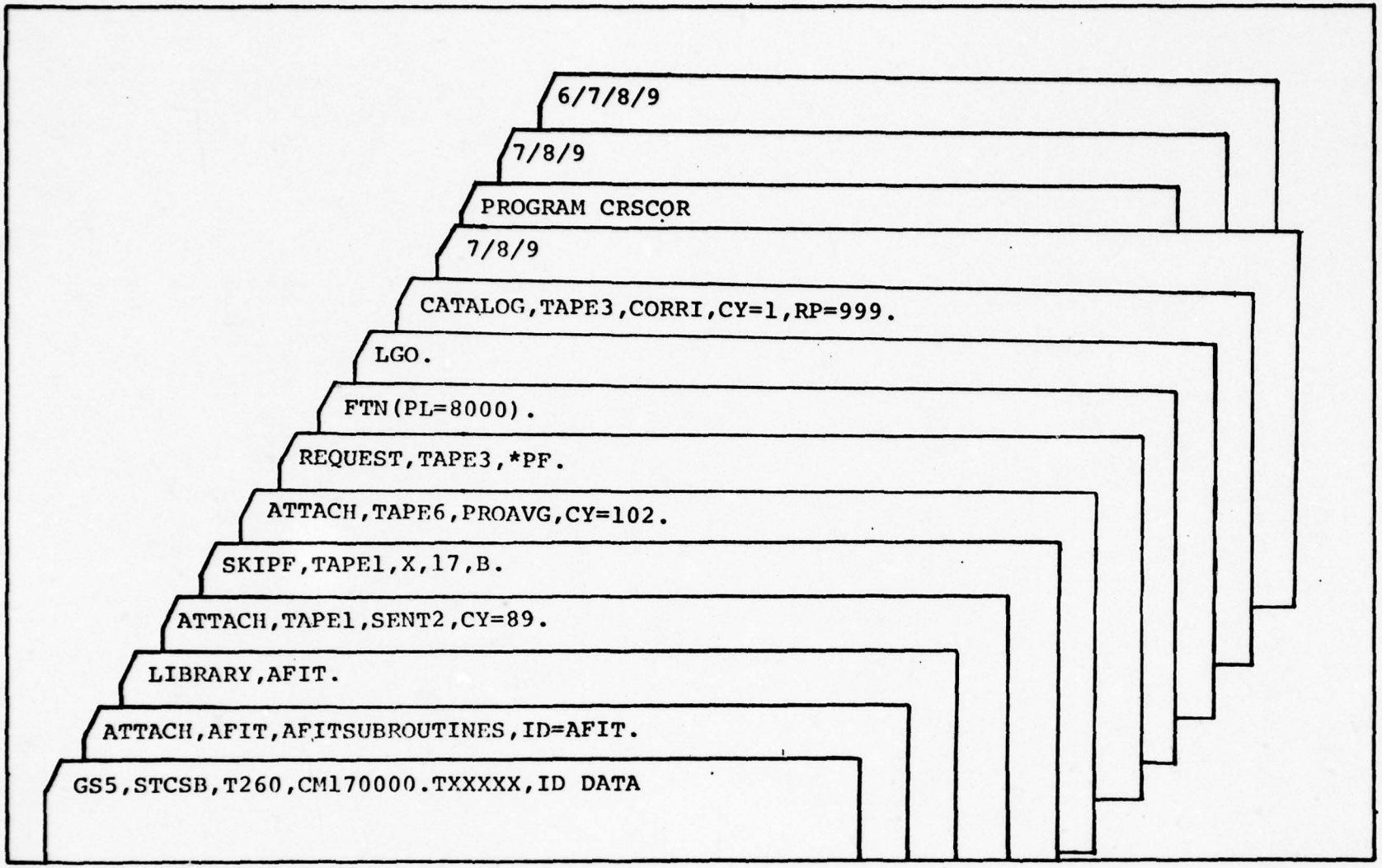


Figure 30. Program Crscor

```
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 A FILE ATTACHED AS TAPE1. THE DATA MUST BE IN AN ARRAY 16XM,       **
C**    WHERE M<501. UP TO 61 PROTOTYPES OF SIZE 16XN,WHERE N<16,CAN BE         **
C**    ATTACHED AS TAPE6 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*****************************************************************************
      PROGRAM CRSCOR(INPUT,OUTPUT,TAPE1,TAPE2,TAPE6,TAPE9=OUTPUT,PLOT,
     ATAPE3)
      DIMENSION GOOD(64),ITYP(64)
      COMMON NSTART,NN2,NN3,NN5,ISUBLN,IOVLAP,NORMAL,NORMAR,ATOL,BTOL,
     1INHIB,LOOK,IDECID,GOOD,ITYP,ILIM,ILIN,IZSEL,IFILT
C-----------------------------------------------------------------------------
C--                    TITLE OF WORD/SENTENCE BEING READ                    --
C    "KIRK, HERE, BEAM ME UP, SCOTTY.  "
C-----------------------------------------------------------------------------
C#############################################################################
C##                         VARIBLES USED BY PROGRAM                        ##
C##              (*) MUST BE SET FOR EACH SENTENCE/WORD CHANGE              ##
C#############################################################################
C    *POSITION OF SENTENCE INFORMATION IN INPUT ARRAY
      NSTART=20
C    *MAXIMUM SENTENCE LENGTH
      NN2=480
C    *NUMBER OF THE SENTENCE/WORD BEING READ
      NN5=16
C     SIZE OF LARGEST PROTOTYPE + ONE
      NN3=16
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 ------------------------------------------------------------------------
      IFILT=0
C-------------------------------------------------------------------------
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 1,OTHERWISE
C       SET "INHIB" TO 0.
      INHIB=1
```



```
1 CRSCOR     74/74   OPT=1                    FTN 4.5+414        10/31/77

C-------------------------------------------------------------------------
C--                      PROTOTYPE SIZE = ITYP(X)                       --
C-------------------------------------------------------------------------
      ITYP(1)=2
      ITYP(2)=9
      ITYP(3)=2
```

```
      ITYP(4)=7
      ITYP(5)=8
      ITYP(6)=7
      ITYP(7)=3
      ITYP(8)=7
      ITYP(9)=5
      ITYP(10)=5
      ITYP(11)=7
      ITYP(12)=3
      ITYP(13)=6
      ITYP(14)=13
      ITYP(15)=4
      ITYP(16)=7
      ITYP(17)=8
      ITYP(18)=7
      ITYP(19)=10
      ITYP(20)=7
      ITYP(21)=6
      ITYP(22)=4
      ITYP(23)=6
      ITYP(24)=6
      ITYP(25)=7
      ITYP(26)=6
      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*," SPOKEN BY            . CONTINUOUS SPEECH. "
C-----------------------------------------------------------------------
C--                  TRANSFER CONTROL TO SUBROUTINE                   --
C-----------------------------------------------------------------------
```

```
      CALL XCORR
C
      STOP
      END
C
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),CPROTO(64,32),CONPRO(64,32),CORR(64,32)
      REAL MARR
      DIMENSION PRO(500,26)
      DIMENSION NN(2),B(500,16),PROTO(15,16),C(64,16),D(64,16)
      DIMENSION SYMBOL1(30),SYMBOL2(30),SUMM(64),EPROTO(15,16)
      DIMENSION SAMPLE(500),TIME(500)
      DIMENSION GOOD(64),ITYP(64)
      DIMENSION SYMBOL3(1),SYMBOL4(1),SYMBOL5(1),SYMBOL6(1),SYMBOL7(1)
      COMMON NSTART,NN2,NN3,NN5,ISUBLN,IOVLAP,NORMAL,NORMAR,ATOL,BTOL,
     1INHIB,LOOK,IDECID,GOOD,ITYP,ILIM,ILIN,IZSEL,IFILT
      EQUIVALENCE (CPROTO,CORR)
C     PHONEME SYMBOL SET
      DATA SYMBOL1/2HKB,2HUR,2HKE,2HH ,2HI ,2HR ,2HB ,
     A2HIE,2HME,2HMB,2HU ,2HP ,2HS ,2HAH,2HT ,2HQU,
     A2HOO,2HT3,2HTE,2HE ,2HA ,2HV ,2HEN,2HN ,2HEE,
     A2HER,2H  ,2H  ,2H  ,2H  /
C     PHONEME-WORD SET
      DATA SYMBOL2/4HKICK,4HUR  ,4HKICK,4HHEM ,4HIT  ,
     A4HRUN ,4HBEG ,4HEEL ,4HMEN ,4HMEN ,
     A5HUNDER,5HPAPER,4HSEND,4HODD ,4HTOE ,4H    ,2H  ,2H  ,2H  ,2H  ,
     A2H  ,2H  ,2H  ,2H  ,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=26
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-----------------------------------------------------------------------
      DO 20 I=1,NN2
    5 READ(1,10)(B(I,J),J=1,NN4)
   10 FORMAT(16F6.3)
      IF(EOF(1))30,20
   20 CONTINUE

E XCORR      74/74   OPT=1                    FTN 4.5+414      10/31/77

   30 CONTINUE
      INEND=I-1
```

```
      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+(ISUBLN-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(6,140)(PROTO(K,L),L=1,NN4)
  140 FORMAT(16F6.3)
      IF(EOF(6).NE.0) GO TO 151
  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,A4,1X,"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 966
  966 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,NUM
      DO 176 L=1,NN4
      CPROTO(K,L)=EPROTO(K,L)
  176 CONTINUE
      DO 177 K=1,NUM
      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-----------------------------------------------------------------------
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)
      DO 997 I=1,54
      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
      DO 300 KK=IDEN,LAP
      LP=KK+IOFSET-(IDEN-1)
      PRO(LP,JP)=SUMM(KK)/GOOD(JP)
  300 CONTINUE
  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
C
      NSENT=NN5    $    NREC=IEND
      WRITE(3,1290)NSENT,NREC,NPRO,NSTART
 1290 FORMAT(4I3)
```

```
      DO 1310 I=1,IEND
      WRITE(3,1320)(PRO(I,J),J=1,26)
1310  CONTINUE
1320  FORMAT(13F6.3)
      ENDFILE3
      RETURN
 702  STOP"ARRAY EXCEEDS DIMENSIONS"
 703  STOP"ID EXCEEDS LIMIT"
 704  STOP"IDIN NOT EQUAL TO N"

 705  STOP
      END
```

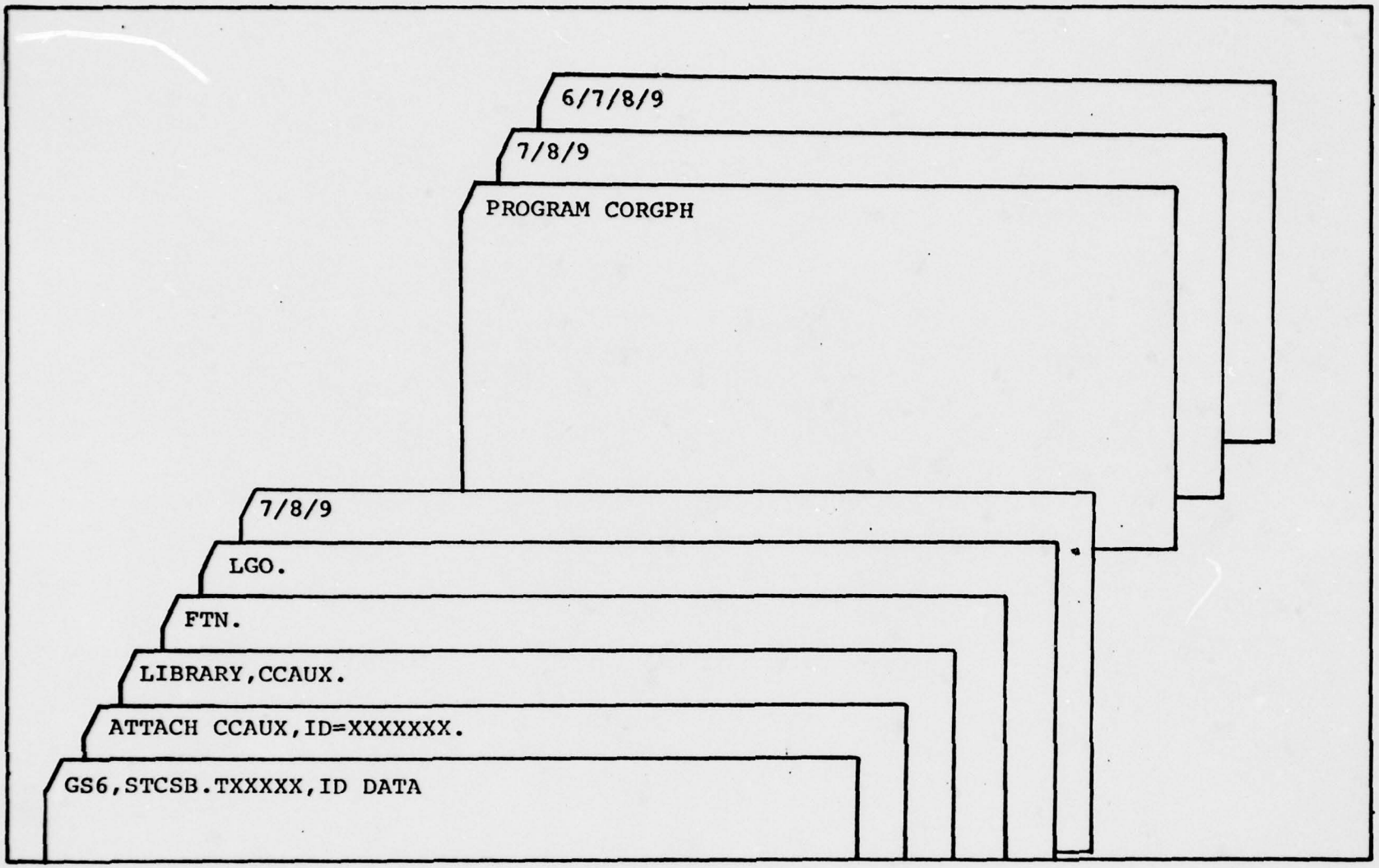


Figure 31. Program Corgph

```
C*************************************************************************
C*****   THIS PROGRAM ATTACHES CORRELATION ARRAYS STORED                 *********
C*****   IN PERMANANT FILES.  IT THEN USES CCAUX ROUTINES TO             ********
C*****   GRAPH THE DESIRED CORRELATION OUTPUTS.  THE INFORMATION CON-    *********
C*****   CERNING THE GRAPH OUTPUT PROCEDURE CAN BE FOUND IN THE YELLOW   *********
C*****   CALCOMP BOOKLET WHICH IS PRINTED BY ASD.                        *********
C*************************************************************************
C
C
      PROGRAM CORGPH(INPUT,OUTPUT,TAPE1)
C
C*************************************************************************
C*****                        INPUT CORRELATION ARRAY                    ********
C*************************************************************************
C
      READ(1,2)NSENT,NREC,NPRO,NSTART
 2    FORMAT(4I3)
      DO 10 I=1,NREC
      READ(1,5)(PRO(I,J),J=1,26)
 10   CONTINUE
 5    FORMAT(13F6.3)
C *************************************************************************
C ***   OUTPUT THE FIRST FOUR CORRELATION VALUES IN CALCOMP FORMAT       ********
C ***    (ONLY THE FIRST FOUR CORRELATION GRAPHS ARE OUTPUT HERE)         *******
C *************************************************************************
      DO 1250 J=1,4
      DO 1150 I=1,IEND
      SAMPLE(I)=PRO(I,J)       $       TIME(I)=I+(NSTART-1)
1150  CONTINUE
      IF(J.GT.1) GO TO 1115
      CALL PLOT(0.,1.,-3)
      CALL SCALE(SAMPLE,1.5,IEND,1)
      CALL AXIS(0.,0.,2HKB,4,1.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      GO TO 1240
1115  IF(J.GT.2) GO TO 1120
```

```
      CALL PLOT(0.,2.5,-3)
      CALL SCALE(SAMPLE,1.5,IEND,1)
      CALL AXIS(0.,0.,2HUR,4,1.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      GO TO 1240
 1120 IF(J.GT.3) GO TO 1130
      CALL PLOT(0.,2.5,-3)
      CALL SCALE(SAMPLE,1.5,IEND,1)
      CALL AXIS(0.,0.,2HKE,4,1.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
      GO TO 1240
 1130 CALL PLOT(0.,2.5,-3)
      CALL SCALE(SAMPLE,1.5,IEND,1)
      CALL AXIS(0.,0.,2HH ,4,1.5,90.0,SAMPLE(NP1),SAMPLE(NP2))
 1240 CONTINUE
      CALL SCALE(TIME,8.,IEND,1)
      CALL AXIS(0.,0.,4HTIME,-4,8.,0.,TIME(NP1),TIME(NP2))
      CALL FLINE(TIME,SAMPLE,-IEND,1,0,0)
 1250  CONTINUE
      CALL PLOTE(N)
      STOP
      END
```

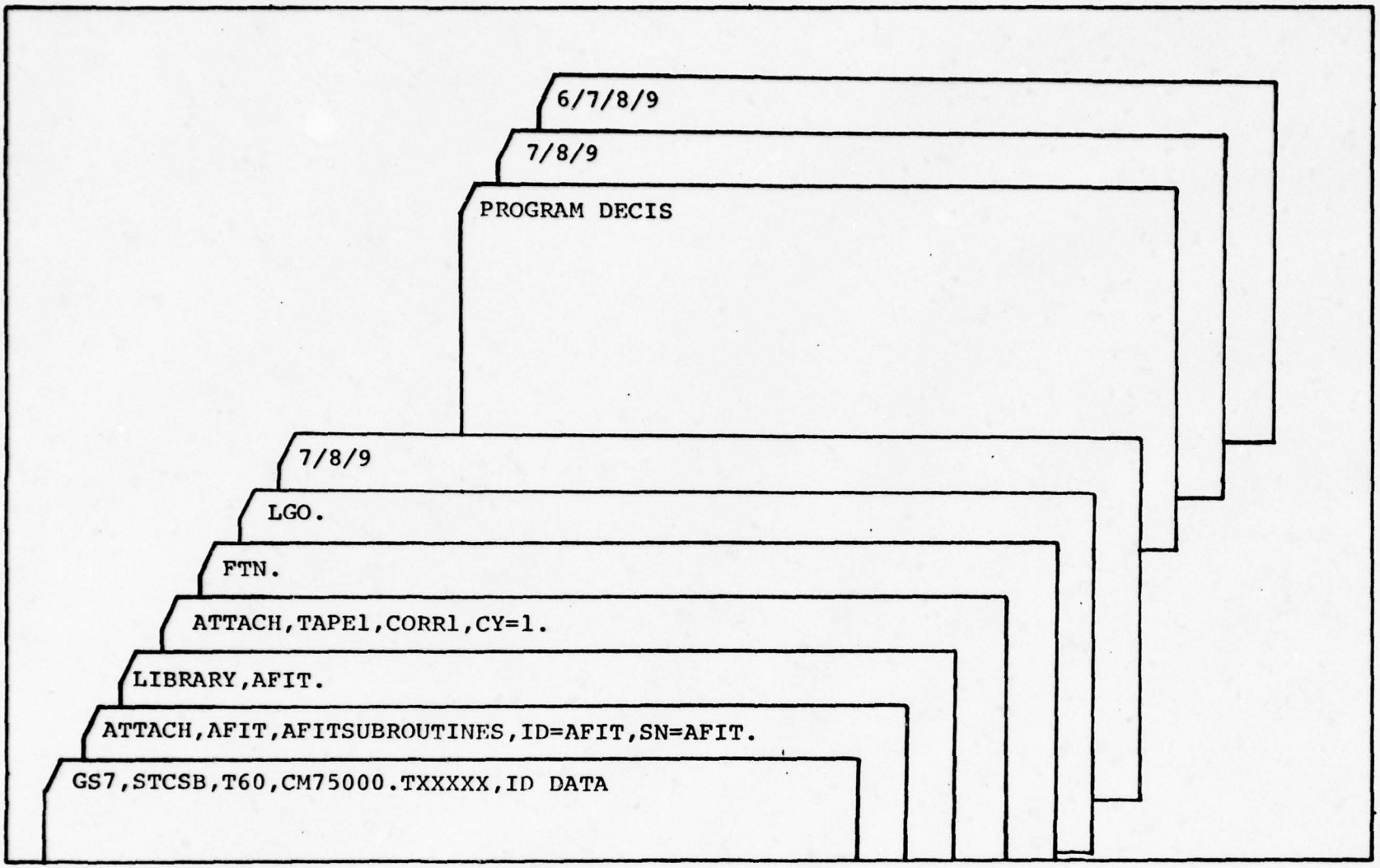


Figure 32. Program Decis

```
C*****
C*****  THIS PROGRAM ATTACHES A PERMANANT FILE (TAPE1) AND PROCESSES      ******
C*****  THE FILE ACCORDING TO THE FOLLOWING SCHEME:                       ******
C*****  1) ALL ARRAY VALUES LT THRESHOLD ARE GIVEN A VALUE OF ZERO        ******
C*****  2)  ARRAY VALUES ALONG TIME AXIS ARE CHECKED  FOR AN ENDURANCE OF ******
C*****      GREATER THAN A PERCENTAGE OF THE SPECIFIED PROTOTYPE LENGTH.  ******
C*****  3)  THE RESULTING ARRAY IS RANKED FOR THE TOP FIVE VALUES FOR     ******
C*****      EACH TIME INCREMENT.  THE RESULTING RANKING IS PRINTED USING  ******
C*****      THE PROTOTYPE VALUES WHICH INCURRED THE RANKING CORRELATION   ******
C*****      VALUES.                                                       ******
C*****
      PROGRAM DECIS(INPUT,OUTPUT,TAPE1)
      DIMENSION PRO(500,26),SART(26),SART1(26)
      DIMENSION IPHON(500,5),ITYP(30),SYMBOL1(30)
C -------------------------------------------------------------------------
C-----                    PHONEME SYMBOL SET                             ------
C -------------------------------------------------------------------------
      DATA SYMBOL1/2HK ,2HUR,2HKE,2HH ,2HI ,2HR ,2HB ,
     A2HIE,2HME,2HM ,2HU ,2HP ,2HS ,2HAH,2HT ,2HOU,
     A2HOO,2HTR,2HTE,2HE ,2HA ,2HV ,2HEN,2HN ,2HEE,
     A2HER,2H  ,2H  ,2H  ,2H  /
      DATA ITYP/2,9, 2,7,8,7,3,7,5,5,7,3,6,13,4,
     A7,8,7,10,7,6,9,6,6,7,6,1,1,1,1/
C -------------------------------------------------------------------------
C -----        SET TIME ENDURANCE AND THRESHOLD VALUES                    -----
C -------------------------------------------------------------------------
      ENDUR=.5       $       THRHLD=.83
      READ(1,2)NSENT,NREC,NPRO,NSTART
2     FORMAT(4I3)
      DO 10 I=1,NREC
      READ(1,5)(PRO(I,J),J=1,26)
      IF(EOF(1)) 20,30
30    CONTINUE
10    CONTINUE
20    CONTINUE
5     FORMAT(13F6.3)
```

```
      PRINT*," DATA READ IN. "
C -----------------------------------------------------------------------
C -----   LOAD PRO ARRAY WITH VALUES GREATER THAN THRESHOLD (THRHLD)   -------
C -----------------------------------------------------------------------
      DO 1095 I=1,NPRO
      DO 1090 J=1,NPREC
      IF(PRO(J,I).GT.THRHLD) GO TO 1090
      PRO(J,I)=0.
1090  CONTINUE
1095  CONTINUE
      PRINT*," DATA ARRAY PROCESSED FOR THRESHOLD. "
C -----------------------------------------------------------------------
C -----    CHECK THRESHOLD ARRAY FOR PROPER TIME ENDURANCE             -----
C -----------------------------------------------------------------------
      DO 1296 I=1,NPRO
      IFLAG=0       $       ICOUNT=0
      DO 1295 J=1,NPREC
      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
      IFLAG=0       $       ITIME=ENDUR*ITYP(I)
      IF(ICOUNT.GE.ITIME) GO TO 1295
      MCONT=MARK+ICOUNT
      DO 1290 JJ=MARK,MCONT
      PRO(JJ,I)=0.
1290  CONTINUE
1295  CONTINUE
      MARK=0    $   ICOUNT=0
1296  CONTINUE
C -----------------------------------------------------------------------
C -----   USE SUBROUTINE TO SORT THE PRO ARRAY (SORT)                  -----
```

```
C ----------------------------------------------------------------------
      DO 1500 ICOL=1,NREC
      DO 1310 IROW=1,NPRO
      SART(IROW)=PRO(ICOL,IROW)
      SART1(IROW)=SART(IROW)
1310  CONTINUE
      CALL SORT(NPRO,SART1)
      DO 1350 IB=1,5
      IIB=NPRO+1-IB
      DO 1320 IA=1,NPRO
      IF(SART(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
      PRINT*
      PRINT*
      PRINT*," DECISION SCHEME FOR SENTENCE NUMBER   ",NSENT
      PRINT*
      PRINT*," NUMBER OF PROTOTYPES IN DECISION SCHEME  ",NPRO
      PRINT*
      PRINT*
      PRINT*," THE DECISION SCHEME OUTPUT RANKED FROM 1 TO 5 IS "
      DO 400 J=1,NREC
      JJ=J+NSTART
      PRINT 410,JJ,(SYMBOL1(IPHON(J,I)),I=1,5)
400   CONTINUE
410   FORMAT(1X,I3,5X,5A3)
      STOP
      END
```

APPENDIX C

Data Results

C. Data Results

Notes on Appendix C

1. Sections in the scoring row which contain dashes were not scored due to time limits within the computer for that particular data run. All like sentences were limited to the same number of possible prototype occurrences so that the scoring may be compared.

2. The complete set of sentence data on the 26-Class prototype problem was not run due to difficulties in obtaining sufficient computer running time. The final tables containing results from the second sentence reflect only a few runs which served to show that the overall performance is consistent with a different sentence input.

Table XI

Sentence Analysis, Speaker 1, 15-Class Problem,
Discrete Prototype

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 1, Speaker 1, Discrete Speech.
Sent 3, Speaker 1, Continuous Speech.

Prototypes: 15-Class, Discrete

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	SKK			AH	T	IE
Score 1:	X	X	X	X	X	X	X	X	X	X	O	X	X	---			---	---	---
Score 3:	X	X	X	O	L	X	O	X	L	X	O	O	O	X	X	X	O	O	

Overall Performance: Sent 1 Location = 12/13 92.3%
Identification = 12/13 92.3%

Sent 3 Location = 11/18 61.1%
Identification = 8/18 44.4%

Legend: X = Identified
L = Located
O = Miss

Table XII

Sentence Analysis, Speaker 2, 15-Class Problem
Discrete Prototype

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 5, Speaker 2, Discrete Speech.
Sent 7, Speaker 2, Continuous Speech.

Prototypes: 15-Class, Discrete

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE
Score 5:	X	O	X	L	O	O	L	O	L	X	L	O	O	---	---	---	---	---
Score 7:	---	---	---	O	O	X	O	O	O	L	O	O	O	X	O	X	O	O

Overall Performance: Sent 5 Location = 6/13 46.2%
Identification = 3/13 23.1%

Sent 7 Location = 4/15 26.7%
Identification = 3/15 20.0%

Legend: X = Identified
L = Located
O = Miss

Table XIII

Sentence Analysis, Speaker 3, 15-Class Problem,
Discrete Prototype

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 9, Speaker 3, Discrete Speech.
Sent 11, Speaker 3, Continuous Speech.

Prototypes: 15-Class, Discrete

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE	
Score 9:	X	X	O	O	O	X	O	O	X	X	L	X	X	----------					
Score 11:	X	L	X	X	L	O	O	O	O	X	O	O	O	X	O	O	O	O	

Overall Performance: Sent 9 Location = 8/13 61.5%
Identification = 7/13 53.8%

Sent 11 Location = 7/18 38.9%
Identification = 5/18 27.8%

Legend: X = Identified
L = Located
O = Miss

Table XIV

Sentence Analysis, Speaker 4, 15-Class Problem, Discrete Prototype

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 13, Speaker 4, Discrete Speech.
Sent 15, Speaker 4, Continuous Speech.

Prototypes: 15-Class, Discrete

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE
Score 13:	X	X	O	X	O	X	L	O	O	O	O	O	O	X	O	O	O	O
Score 15:	X	X	O	O	O	X	O	O	O	O	O	O	O	O	X	O	O	O

Overall Performance:

Sent 13	Location = 7/18	38.9%
	Identification = 6/18	33.3%
Sent 15	Location = 4/18	22.2%
	Identification = 4/18	22.2%

Legend: X = Identified
L = Located
O = Miss

Table XV

Sentence Analysis, Speaker 1, 15-Class Problem,
Averaged Prototypes

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 1, Speaker 1, Discrete Speech.
Sent 3, Speaker 1, Continuous Speech.

Prototypes: 15-Class, Averaged

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE	
Score 1:	X	X	X	L	X	X	L	O	X	X	O	L	L	X	O	X	L	O	
Score 3:	X	X	L	X	L	L	X	O	X	X	O	X	X	X	X	X	X	X	

Overall Performance: Sent 1 Location = 14/18 78.8%
Identification = 9/18 50.0%

Sent 3 Location = 15/18 83.3%
Identification = 13/18 72.2%

Legend: X = Identified
L = Located
O = Miss

Table XVI

Sentence Analysis, Speaker 2, 15-Class Problem, Averaged Prototypes

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 5, Speaker 2, Discrete Speech.
Sent 7, Speaker 2, Continuous Speech.

Prototypes: 15-Class, Averaged

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE
Score 5:	X	L	X	L	O	X	L	O	L	X	O	X	L	X	L	X	O	O
Score 7:	X	L	X	X	X	X	X	X	X	X	L	X	L	X	X	X	X	L

Overall Performance: Sent 5 Location = 13/18 72.2%
Identification = 7/18 38.9%

Sent 7 Location = 18/18 100%
Identification = 14/18 77.8%

Legend: X = Identified
L = Located
O = Miss

Table XVII

Sentence Analysis, Speaker 3, 15-Class Problem, Averaged Prototypes

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 9, Speaker 3, Discrete Speech.
Sent 11, Speaker 3, Continuous Speech.

Prototypes: 15-Class, Averaged

Phonemic Rendition:

	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE
Sent:																		
Score 9:	X	X	X	O	O	X	L	O	L	X	L	L	X	X	X	-------		
Score 11:	O	L	L	L	L	O	O	O	X	X	O	L	X	X	O	X	L	L

Overall Performance: Sent 9 Location = 12/15 80.0%
Identification = 8/15 53.3%

Sent 11 Location = 12/18 66.7%
Identification = 5/18 27.8%

Legend: X = Identified
L = Located
O = Miss

Table XVIII

Sentence Analysis, Speaker 4, 15-Class Problem, Averaged Prototypes

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 13, Speaker 4, Discrete Speech.
Sent 15, Speaker 4, Continuous Speech.

Prototypes: 15-Class, Averaged

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE
Score 13:	X	X	L	L	L	L	L	L	L	L	L	X	O	X	X	L	X	O
Score 15:	X	L	L	O	O	X	L	L	L	L	X	X	X	X	L	O	X	O

Overall Performance:

Sent 13 Location = 10/18 55.6%
Identification = 6/18 33.3%

Sent 15 Location = 14/18 77.8%
Identification = 7/18 38.9%

Legend: X = Identified
L = Located
O = Miss

Table XIX

Sentence Analysis, Speaker 1, 26-Class Problem, Averaged Prototypes

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 1, Speaker 1, Discrete Speech.
Sent 3, Speaker 1, Continuous Speech.

Prototypes: 26-Class, Averaged

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE
Score 1:	X	X	X	X	O	X	L	O	L	X	O	X	X	X	O	X	L	O
Score 3:	X	X	X	X	X	X	X	X	O	X	O	X	X	X	X	X	O	O

Overall Performance:

Sent 1	Location = 14/18	77.8%
	Identification = 10/18	55.6%
Sent 3	Location = 15/18	83.3%
	Identification = 15/18	83.3%

Legend: X = Identified
L = Located
O = Miss

Table XX

Sentence Analysis, Speaker 2, 26-Class Problem,
Averaged Prototypes

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 5, Speaker 2, Discrete Speech.

Prototypes: 26-Class, Averaged

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE
Score 5:	X	L	X	X	X	X	X	X	O	X	O	X	X	X	O	X	O	O

Overall Performance: Sent 5 Location = 13/18 72.2%
Identification = 12/18 66.6%

Legend: X = Identification
L = Location
O = Miss

Table XXI

Sentence Analysis, Speaker 3, 26-Class Problem,
Averaged Prototypes

Sentence Spoken: "Kirk here, beam me up, Scotty."

Sentences Analyzed: Sent 9, Speaker 3, Discrete Speech.
Sent 11, Speaker 3, Continuous Speech.

Prototypes: 26-Class, Averaged

Phonemic Rendition:

Sent:	K_b	UR	K_e	H	I	R	B	IE	M_e	M_b	IE	U	P	S	K	AH	T	IE
Score 9:	X	X	X	X	X	X	L	O	X	X	L	X	X	---	---	---	---	---
Score 11:	X	X	X	X	X	X	O	O	X	X	X	X	X	X	O	O	O	X

Overall Performance: Sent 9 Location = 12/13 92.3%
Identification = 10/13 76.9%

Sent 11 Location = 13/18 72.2%
Identification = 13/18 72.2%

Legend: X = Identification
L = Location
O = Miss

Table XXII

Sentence Analysis, Speaker 1 & 4, 26-Class Problem,
Averaged Prototypes

Sentence Spoken: "Quoth the Raven, nevermore."

Sentences Analyzed: Sent 2, Speaker 1, Discrete Speech.
Sent 14, Speaker 4, Discrete Speech.

Prototypes: 26-Class, Averaged

Phonemic Rendition:

Sent:	QU	OO	T_e	T_b	E	R	A	V	EN	N	EE	V	ER	M_b	OO	ER	
Score 2:	X	X	O	X	L	X	L	X	X	L	O	X	X	X	X	X	
Score 14:	O	X	O	L	O	X	L	X	O	O	O	L	X	X	X	O	

Overall Performance: Sent 2 Location = 14/16 87.5%
Identification = 11/16 68.8%

Legend: X = Identification
L = Location
O = Miss

APPENDIX D

Glossary of Technical Terms

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

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

7. Phoneme: The smallest distinctive group or class of phones (an individual speech sound) 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.

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

Vita

Michael F. Guyote was born on 31 August 1946 in New Iberia, Louisiana. He graduated from high school in 1964 and attended the George Washington University from September, 1964, through May, 1965. He entered the USAF Academy in June, 1965, and graduated in June, 1969, with a Bachelor's Degree, majoring in mathematics. He entered Undergraduate Pilot Training at Columbus AFB, Mississippi, in July, 1969, and received the rating of Pilot in July, 1970. He served as an Instructor Pilot for the 71st Pilot Training Squadron at Vance AFT, Oklahoma, from November, 1970, through February, 1974. He then served as Air Staff Officer, Instructor Pilot, FCF Pilot, and Test Chase Pilot at USAF Flight Test Center, Edwards AFB, California, until arriving at the Air Force Institute of Technology in June, 1976.

Permanent Address: 508 Weeks Island Road
New Iberia, Louisiana 70560

Vita

Patrick L. Sisson was born on 16 February 1947 in Allentown, Pennsylvania. He graduated from high school in Casa Grande, Arizona, in 1965. He entered the USAF Academy that same year and graduated in June, 1969, with a Bachelor's Degree in Electrical Engineering. He entered Undergraduate Pilot Training at Columbus AFB, Mississippi, in August, 1969, and received the rating of pilot in August, 1970. He served with the Military Airlift Command from 1970 through 1976 as pilot, Instructor Pilot, and Flight Examiner. He served his last year with Military Airlift Command as Air Operations Officer in Adana, Turkey. He entered the Air Force Institute of Technology in June, 1976.

Permanent Address: 610 Georgia
Vallejo, California 94590

~~UNCLASSIFIED~~
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER GE/EE/77-D-18 (14) AFIT/GE/EE/77D-18	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) COMPUTER IDENTIFICATION OF PHONEMES IN CONTINUOUS SPEECH.	5. TYPE OF REPORT & PERIOD COVERED MS Thesis
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Michael F. Guyote [redacted] Patrick L. Sisson [redacted]	8. CONTRACT OR GRANT NUMBER(s) (9) Master's thesis,
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology School of Engineering (AFIT/ENG) Wright-Patterson AFB OH 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (11)
11. CONTROLLING OFFICE NAME AND ADDRESS Aerospace Medical Research Laboratory/EM Wright-Patterson AFB OH 45433	12. REPORT DATE Nov[redacted]77 (12D) 13. NUMBER OF PAGES 156 p.
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 public release; IAW AFR 190-17
JERRAL F. GUESS, Captain, USAF
Director of Information

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 including single and 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

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE UNCLASSIFIED

24 2 3 3 5 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

attain an average recognition rate of 72% on continuous speech spoken by dissimilar speakers.