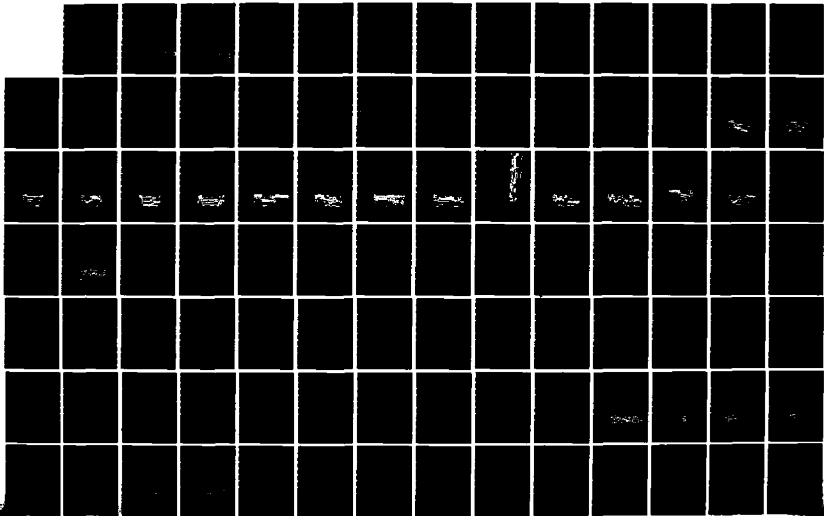
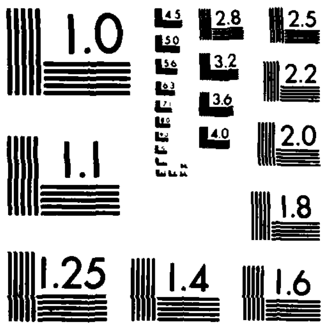
AD-A135 833 COMPUTER RECOGNITION OF PHONEMES IN THE PRESENCE OF COCKPIT INDUCED STRESS AND NOISE(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI..

1/2

UNCLASSIFIED K A BEACHY 20 DEC 82 AFIT/GE/EE/82D-20 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A135 833

COMPUTER RECOGNITION OF PHONEMES IN THE PRESENCE OF COCKPIT INDUCED STRESS AND NOISE

THESIS

AFIT/GE/EE/82D-20

Keith A. Beachy
Captain USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY (ATC)

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/EE/82D-20

COMPUTER RECOGNITION OF PHONEMES
IN THE PRESENCE OF COCKPIT INDUCED
STRESS AND NOISE

THESIS

AFIT/GE/EE/82D-20 Keith A. Beachy
Captain USAF

Approved for public release; distribution unlimited.

AFIT/GE/EE/82D-20

COMPUTER RECOGNITION OF PHONEMES
IN THE PRESENCE OF COCKPIT INDUCED
STRESS AND NOISE

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
Keith A. Beachy, BSEE
Captain USAF
Graduate Electrical Engineering

December 1982

Approved for public release; distribution unlimited.

Preface

This research was motivated by Dr. Matthew Kabrisky, Professor of Electrical Engineering, Air Force Institute of Technology (AFIT). This research effort uses normal and G-stressed speech to analyze algorithms associated with speech recognition systems. The algorithms analyzed in this thesis were developed by AFIT students for the study and recognition of speech.

I wish to thank Dr. Matthew Kabrisky and Major Larry Kizer for their guidance, and suggestions during this research. In addition I wish to thank Dr. Peter Maybeck for his assistance, and review of this report.

Finally, I wish to thank my wife Suzanne for her typing and support during this study.

Keith A. Beachy
Capt USAF

Contents

List of Figures

List of Tables

Abstract

Speech recognition algorithms were analyzed using normal and G-stressed speech as an input. Speech samples were recorded in centrifuge tests at the Air Force Medical Research Lab, Wright-Patterson AFB, Ohio. All speech was recorded using the MBU-12/P face mask. The algorithms studied are phoneme-based feature extractors which feed a recognition algorithm based on fuzzy set theory. Three feature extraction algorithm options were analyzed. One option used a phoneme length of 40 ms and the other options used a length of 8 ms. The recognition results for all three options using normal speech are above 90%, but the 40ms phoneme length give higher raw scores. For G-stressed speech the 40 ms phoneme length scored greater than 90% while the 8ms phoneme length options scored less than 60%.

I. INTRODUCTION

The cockpit tasks for a fighter pilot have increased significantly in the past 35 years. Present technology offers the pilot a multitude of system functions and displays, which have increased the pilot workload considerably. A speech recognition system can be used to decrease the pilot workload. Speech input would also be valuable on low-level missions or when flying wing, because speech input would enable the pilot to keep his eyes out of the cockpit. However, most speech recognition systems degrade considerably when exposed to G-stress speech associated with high performance aircraft.

This research project will use G-speech to analyze a feature extraction and speech recognition system. Solutions to the G-speech recognition problems associated with cockpit noise and stress will be helpful for speech recognition systems used in other applications, both military and civilian.

BACKGROUND

In 1981 at the Air Force Institute of Technology (AFIT), Carl Seelandt developed an extensive software package to extract features from speech (Ref 1). Seelandt's primary work used five-vector phoneme templates to extract features from input speech. Seelandt's work showed promising results because of the ability of his feature extraction system to resynthesize speech from independent speakers (Ref 1). The resynthesized speech was recognizable from many different speakers. However, a preliminary experiment performed by this study had recognition results of less than 20% using resynthesis techniques. These results are in Appendix G and were based on resynthesized speech using Seelandt's phonemes (extracted from Seelandt's speech), with an independent input speaker wearing a helmet and under G-stress. T poor recognition rate is attributed to wearing a mask ther phonemes used in this research were extracted from s ects wearing masks).

Further work in the feature extraction area was done by Martin in 1982 at AFIT (Ref 2). Martin's programs use the array processor and can be used for feature extraction. Software was developed during the course of this research so Martin's programs could be used as part of a feature extraction system. Both Seelandt's and Martin's feature extraction systems created input files for a word recognition algorithm.

The word recognition algorithm studied in this project is a new algorithm developed by Montgomery in 1982 also at AFIT (Ref 3). His algorithm is unique because it is based on fuzzy set theory. Montgomery demonstrated better than 50% recognition results for independent speakers using input data based on a feature extraction system developed by Seelandt (Ref 3:78).

PROBLEM

Three major items of the feature extraction system were investigated. The first item is the length of phonemes contained in a phoneme template. Five-vector length phonemes (40 ms) and one-vector length phonemes (8 ms) were studied. The phonemes are compared to input speech to find the distances between each phoneme and the speech.

The distance rule will also be studied. Seelandt's feature extraction system uses a distance rule called Minkowski one (M1). The M1 distance was chosen for its computational simplicity. This thesis project will study the difference between M1 and Minkowski two (M2) distance.

The third item studied is the averaging of phonemes in the phoneme template. The averaging of phonemes is an option with the feature extraction system software and has not been studied before. The averaging of phonemes will hopefully reduce the number of phoneme needed and make the phoneme template set more robust.

Approach

Five-vector and one-vector phoneme templates were developed from 15 speech files. The phoneme templates consisted of 70 sounds extracted from prerecorded speech of a vocabulary "zero" to "nine", "CCIP", "enter", "frequency", "step", and "threat". These two templates were used by the feature extraction system to create feature extraction files from 90 speech files. The feature extraction files were entered into the speech recognition program to conduct four experiments. The experiments are:

EXPERIMENT	PHONEME LENGTH	DISTANCE RULE
1	5-VECTOR	M1
2	1-VECTOR	M1
3	1-VECTOR	M2
4	1-VECTOR	M2

(fourth experiment is same as third but with different fuzzy variables and word representation used in recognition program)

The speech files used in the experiments consisted of normal (1g) speech and G-stressed speech. All speech was recorded with the subjects wearing a mask. The phoneme templates were extracted from normal (1g) speech with subjects wearing a mask.

Sequence of Presentation

Chapter II covers data acquisition and how the speech files were prepared before the feature extraction. The feature extraction system is discussed in Chapter III with emphasis on the phoneme templates. Chapter IV discusses the word recognition algorithm and how phoneme representations are picked. Results are in Chapter V, with conclusions and recommendations in Chapters VI and VII.

Speech files and computer programs are in Appendices A thru F. Appendix G contains other experiments which include:

1. Resynthesized speech experiment
2. Independent speaker experiment
3. 128-point DFT recognition experiment

II. Data Acquisition

The original data tapes for this research were generated by the Aerospace Medical Research Laboratory (AMRL), Wright-Patterson AFB, Ohio. Volunteers were subjected to different G-levels in the human centrifuge. A standard vocabulary was used which consisted of the words: zero, one, two, three, four, five, six, seven, eight, nine, CCIP, enter, frequency, step, and threat.

Subjects in the human centrifuge were seated in an F-16 seat, at a 30° bank angle with shoulder pads. Subjects were wearing an HGU 48P helmet with an MBU 12P mask connected to a CRU 66A Bendix regulator. In addition to repeating the standard vocabulary, the subjects were simultaneously doing a pitch axis tracking task with a side arm force stick.

The normal and G-stress speech utterances were recorded by AMRL on a small portable Nagra SN tape recorder operating at 3.75 IPS. These original recordings were transferred by AMRL to quarter inch tape using a Nagra IV-D at 7.5 IPS.

Analog to Digital Conversion

The audio equipment was connected as shown in Figure 1. The sampling rate of the input speech waveform for analog to digital (A/D) conversion was 8 kHz. The data was low-pass filtered at 3.6 kHz with -48db/octave slope above the 3.6 kHz break frequency which satisfies Nyquist's sampling criteria. (For more information on the analog to digital interface for the Nova 2 computer see reference 4).

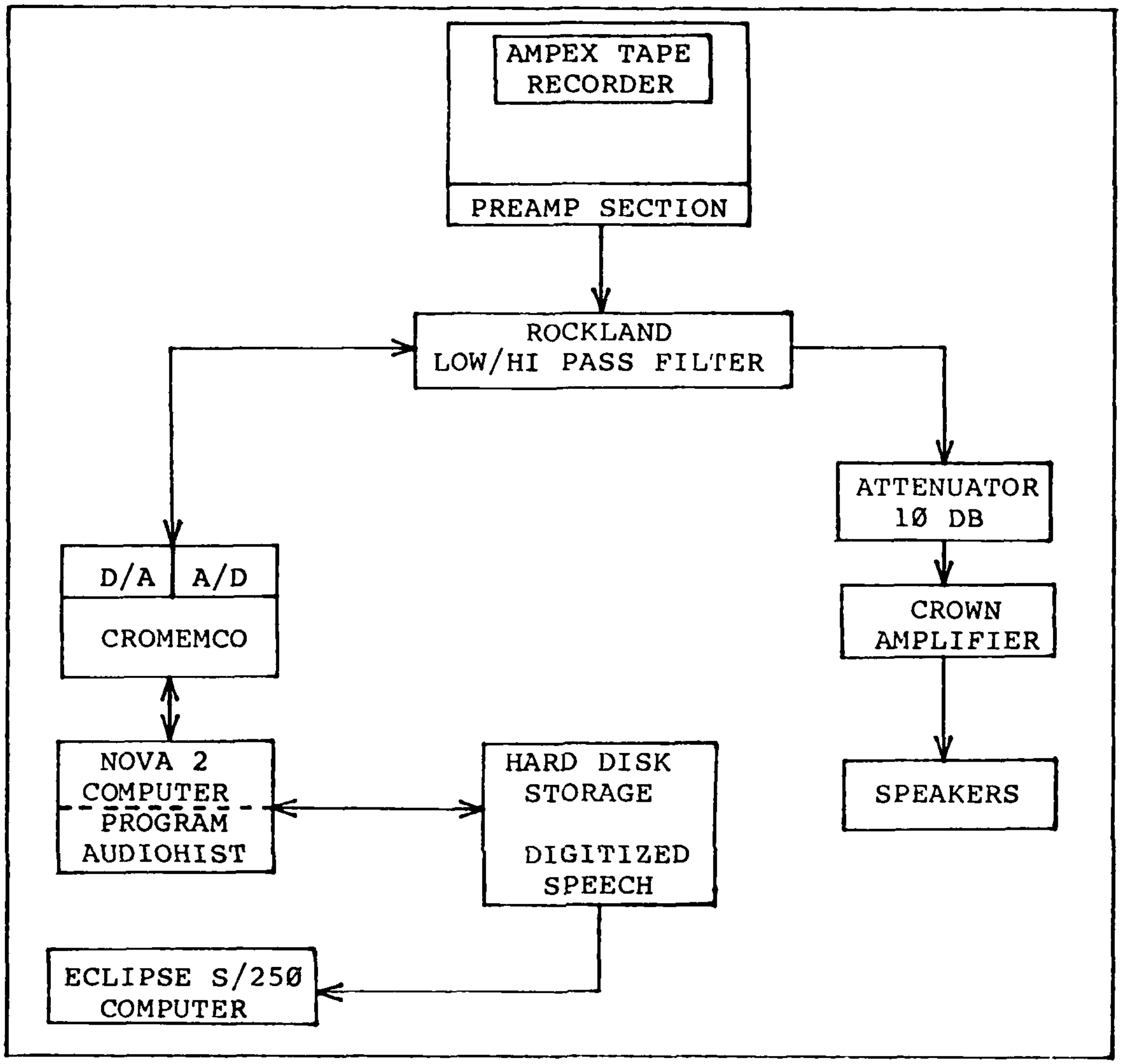


Figure 1. Equipment for Analog to Digital Conversion

The computer program used to digitize the recorded speech was "Audiohist" written by Paul Finkes and J. Hunter (Ref 5, 6). The speech utterances are digitized using "Audiohist" which produces 88 disk blocks (one file). The disk blocks are 256 16-bit integers, and the 88 disk blocks enable 2.816 seconds of data to be stored in each speech file. Program "Audiohist" also enables the user to play back the digitized speech. Digitized speech played back from "Audiohist" has no audible difference from the analog input. The "Audiohist" program was also used to edit each file from 88 blocks to a smaller block size to save file space. The files ranged in size from 15 to 40 blocks after editing. In some files breathing noises were kept in the file along with the word to be analyzed. These noises will be used in the analysis of the feature extraction algorithm and the recognition algorithm. Another computer program developed by Allen can also be used for digitizing data with similar results to "Audiohist" (Ref 7).

The Cromemco A/D converter has a voltage range of ±5 volts and a 12 bit word. Therefore +5 volts would be equal to 2047 stored in the computer.

The digitized speech files are stored as integers on disk and backed up on magnetic tape. The files used in this research project are listed in Appendix A. The file names take the general format of:

X X C X X . SP

- SP — used for all speech files; specifies files as speech
- 5th character (X) — utterance 0,...,9,C,E,F,S,T; 0=zero,...,C=CCIP,...,T=threat
- 4th character (X) — letter A - E,P,S; used to signify different control (16) utterances
- C — used to designate control or the no G-stress utterance ("normal")
- X X — one or two letters to specify the volunteer

or for G-stress utterances:

```
X X # X X . SP
│   │ │ │   └── same as above
│   │ │ └────── same as above
│   │ └──────── specifies the G-force in the Y axis
│   └────────── specifies the G-force in the Z axis
└────────────── same as above
```

Example: HCBE.SP is a speech file (SP) from the volunteer Capt Henwood (H), and is the second (B), control (C) utterance and the word spoken was "enter" (E). H30F.SP is the word "frequency" taken from Capt Henwood at three Gs.

The digitized speech files created and edited using "Audiohist" are processed by feature extraction algorithms. The resulting data from the feature extraction algorithms is then processed for recognition. This will be described in more detail in the next two chapters.

III. Feature Extraction

The feature extraction system used in this research was based on work done by Carl Seelandt (Ref 1). Seelandt's feature extraction system was based on finding distances between a phoneme template and speech utterances. Three different items were studied using the procedures and programs developed by Seelandt. The three items studied are: optimum phoneme length, which distance rule to use, and analysis of phoneme averaging. Phoneme lengths of five-vectors (40ms) and one-vector (8 ms) will be studied. The distance rule used by Seelandt was Minkowski one distance (M1) picked because of M1's computational advantage over other rules. Minkowski two distance will be used in the feature extraction process and compared to M1 distance. Seelandt also developed software to create phoneme templates which included the ability to average multiple source files into individual phonemes. The use of averaged phoneme templates used for feature extraction was also studied. The sequence to follow for feature extraction is depicted in Figure 2.

Discrete Fourier Transform

A discrete Fourier transform (DFT) was used to convert digitized speech files into frequency component files. The DFT process accepts N input samples from the digitized speech files, where N is some power of two. The DFT size used for this study set N equal to 64, which results in 32

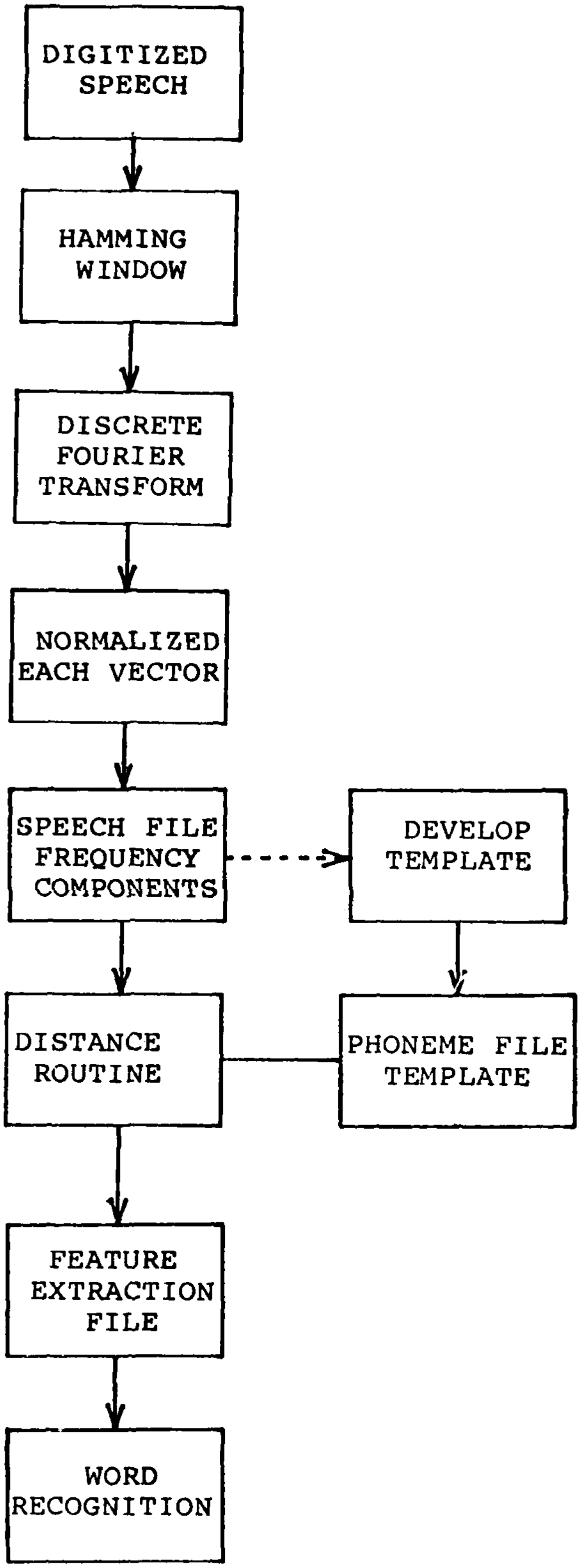


Figure 2. Feature Extraction Steps

components from dc through 3750 hz in 125 hz increments. Thus, frequency component files contain vectors of 32 components and each vector corresponds to 8 ms of original speech.

This research effort uses two different sets of programs to extract features from speech. The first set of feature extraction programs. developed by Karl Seelandt, used a 64 point DFT, normalized each vector, and used a 6 db per octave preemphasis with a corner frequency of 500 hz (Ref 1). The next set of feature extraction programs were developed by Martin (Ref 2). Martin's programs used a preemphasis above 500 hz of 10 db per octave, a deemphasis of 10 db per octave below 300 hz, replaced the dc component of each vector with the vector energy, and was used for single-vector phoneme analysis. The energy in each vector was added to the feature extraction output. Both sets of programs used a Hamming window as recommended by Finkes (Ref 5:22). The Hamming window also produced a cleaner spectrogram than the rectangular window did. The spectrograms were used as one tool to pick phoneme templates.

Phoneme Templates

Extracting features from speech frequency files was accomplished by comparing phoneme templates to the speech files. Producing phoneme templates involved using procedures and software tools developed by Seelandt (Ref 1) and extended to Martin's programs with new software tools developed by this author. Phoneme templates of five-vector and one-vector lengths were developed. In addition averaging different vectors into each phoneme of the phoneme set were investigated and used for the first time. In the work done by Seelandt, he did not have time to investigate the use of averaged phoneme templates.

Five-Vector Phonemes. To make the five-vector phonemes, techniques similar to those used by Seelandt were used. First a spectrogram was made using program TEKTALK, developed by Seelandt (Ref 1) and modified by Fletcher (Ref 8). Program TEKTALK presents a spectrogram of the input speech on a Tektronix Scope. A segment of speech represented by the spectrogram could be heard by placement of the Tektronix's cursors on the spectrogram. Vectors were picked to be in the phoneme template by listening to speech segments and looking at spectrograms generated from the speech files. This may seem rather ad hoc, but this method attempts to pick out the consistent components of speech to be used as a phoneme template. Figures 3 through 17 are spectrograms of the fifteen source files where all the phoneme templates were extracted from for this research. Table I lists, for five-vector and one-vector phoneme templates, the origin of each phoneme. In addition, Table I shows vectors that were used for speech synthesis. A speech synthesis experiment is discussed in Appendix G.

Seelandt tried to pick his phonemes to represent distinguishable sounds found in his speech utterances. In

Table I

Phoneme Template Source
(All phonemes from files with HCP prefix)

PHONEME NUMBER	WORD	5-VECTOR PHONEME START VECTOR	SINGLE VECTOR START / TIMES VECTOR/MODIFIED	VECTOR FOR SPEECH SYNTHESIS
1	noise	-	[zero] 1/5	-
2	zero	10,13	10/7	14
3	zero	18	18/5	20
4	zero	28	28/5	29
5	zero	35	35/5	38
6	zero	45	45/5	49
7	zero	52	52/5	54
8	one	10	10/3	12
9	one	16	16/7	18
10	one	23	23/5	25
11	one	29	29/5	30
12	one	34	35/6	36
13	one	40	39/5	40
13	seven	65	62/5	-
13	nine	64	64/5	-
13	enter	30	30/4	-
14	two	10	10/5	12
15	two	17,22	17/8	20
16	two	27,30	25/8	30
17	two	35,39	35/9	40
18	three	11	9/3	10
19	three	16,20	17/5	19
20	three	27	27/4	30
21	three	34,40	38/7	40
22	three	47	47/4	47
23	four	7	7/5	10
24	four	15,20,25,30	15/19	20
25	four	36	36/4	37
26	four	43	44/4	44
27	four	50	51/4	50
28	five	12	11/3	12
29	five	22,27,32,37	21/23	30
30	five	42,47	44/8	46
31	five	54,57	54/6	55
32	six	13,18,22	13/13	14
33	six	27,32	27/9	30
34	six	38,43	39/8	40
35	six	60,62	61/5	62
36	six	69,74,79	69/17	83
37	seven	11,16,21	11/16	16
38	seven	29	29/5	31
39	seven	35,40	35/10	38
40	seven	45,47	45/6	48
41	seven	52,55	53/7	56
42	seven	61	62/3	63

Table I (Continued)

Phoneme Template Source
(All phonemes from files with HCP prefix)

PHONEME NUMBER	WORD	5-VECTOR PHONEME START VECTOR	SINGLE VECTOR START / TIMES VECTOR/MODIFIED	VECTOR FOR SPEECH SYNTHESIS
43	eight	12,17,23	15/19	20
43	eight	28,33	-	-
44	eight	54,59	54/9	60
45	nine	22,27,32	22/15	24
46	nine	39,44	40/7	43
47	nine	51,56	51/10	55
48	CCIP	44,49	41/10	49
49	CCIP	55,60	55/10	60
50	CCIP	71,76	71/11	73
51	CCIP	82	82/4	83
52	CCIP	98	98/3	100
53	CCIP	104,109	104/19	110
53	CCIP	114,119	-	-
54	enter	19,24	21/8	22
55	enter	39	40/3	41
56	enter	48,53,58	48/13	48
57	frequency	19	19/3	20
58	frequency	25	25/4	26
59	frequency	32,34	32/7	34
60	frequency	46	47/3	47
61	frequency	50	51/4	52
62	frequency	55	56/3	57
63	frequency	69	68/7	70
64	frequency	78,83	78/10	83
65	step	15,20	15/11	23
66	step	39,41	39/7	42
67	step	46,49	46/8	50
68	threat	6,10	6/6	8
69	threat	27,32	28/7	30
70	threat	54,59	55/7	57
71	noise	-	[zero] 59/5	-
72	noise	-	[one] 48/5	-
73	noise	-	[one] 54/10	-
74	noise	-	[two] 1/6	-
75	noise	-	[three] 57/6	-
76	noise	-	[six] 55/5	-
77	noise	-	[eight] 43/10	-
78	noise	-	[nine] 1/10	-
79	noise	-	[nine] 90/7	-
80	noise	-	[step] 63/10	-
81	noise	-	[threat] 40/14	-

this research, phonemes were picked in a similar manner; however, distinct sounds were not the only input for picking a phoneme. Each speech file represented by the spectrograms in Figures 3 through 17 were looked at and listened to using program TEKTALK. Phonemes were picked not only according to sound, but according to how similar the spectrogram vectors were. An attempt was made to pick the vectors that had a consistent spectrographic pattern. The spectrogram for the word "eight" shows a very strong spectrographic pattern for the "a" sound in "eight". Because of that strong pattern for the "a" sound it was decided to average as many vectors as possible, without changing the overall pattern of the sound "a", into one phoneme. Initial results showed that averaging did work and extracted features as well as multiple phonemes for the "a" sound. When three phonemes were used for the "a" sound in "eight" all three phoneme sounds would come up as the top choice in the feature extraction system. When one phoneme was used for the "a" sound in "eight" it replaced all three sounds as the feature extraction choice. Thus, initial results show that an average phoneme could replace multiple phoneme sounds, therefore reducing the phonemes needed for each word. Phoneme templates were created interactively by bringing in speech that consisted of frequency components. The frequency components, which represent the original speech, were used as templates by picking out vectors to be phonemes. The program would take the beginning vecto. of

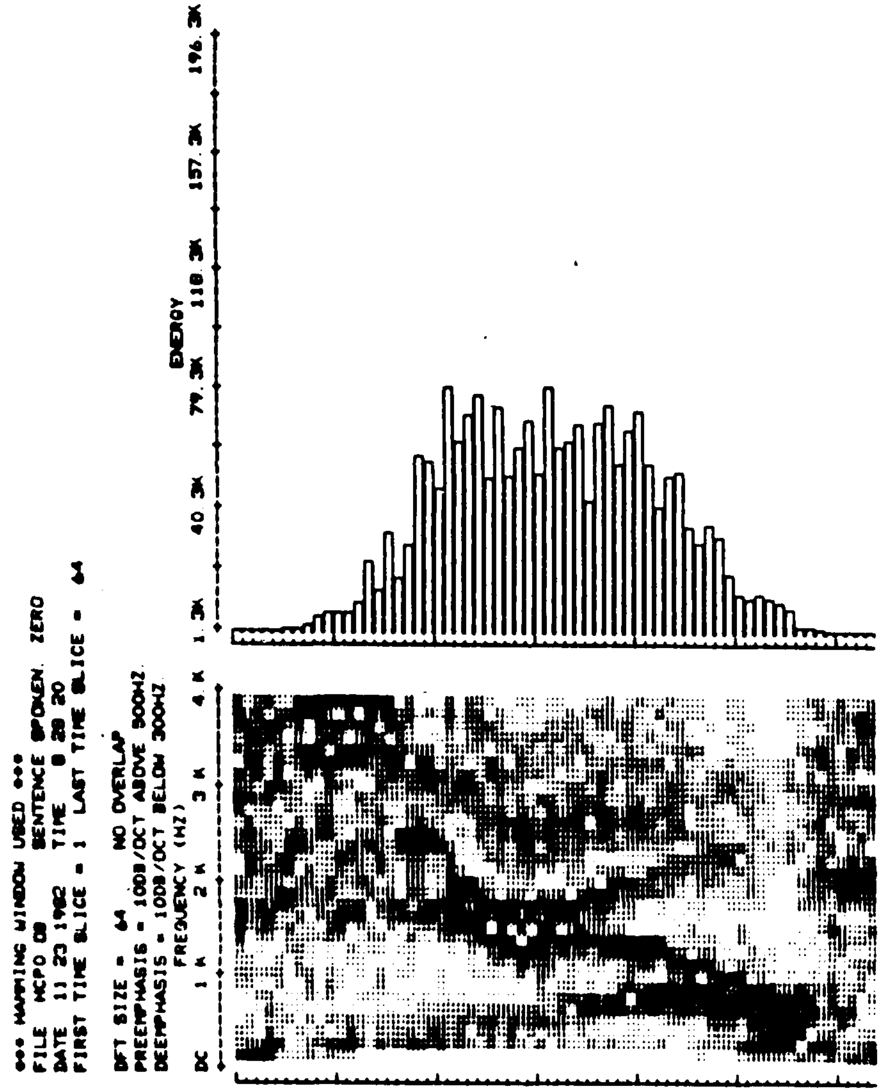


Figure 3. Spectrogram of Zero with Energy Source File for Phonemes 2-7

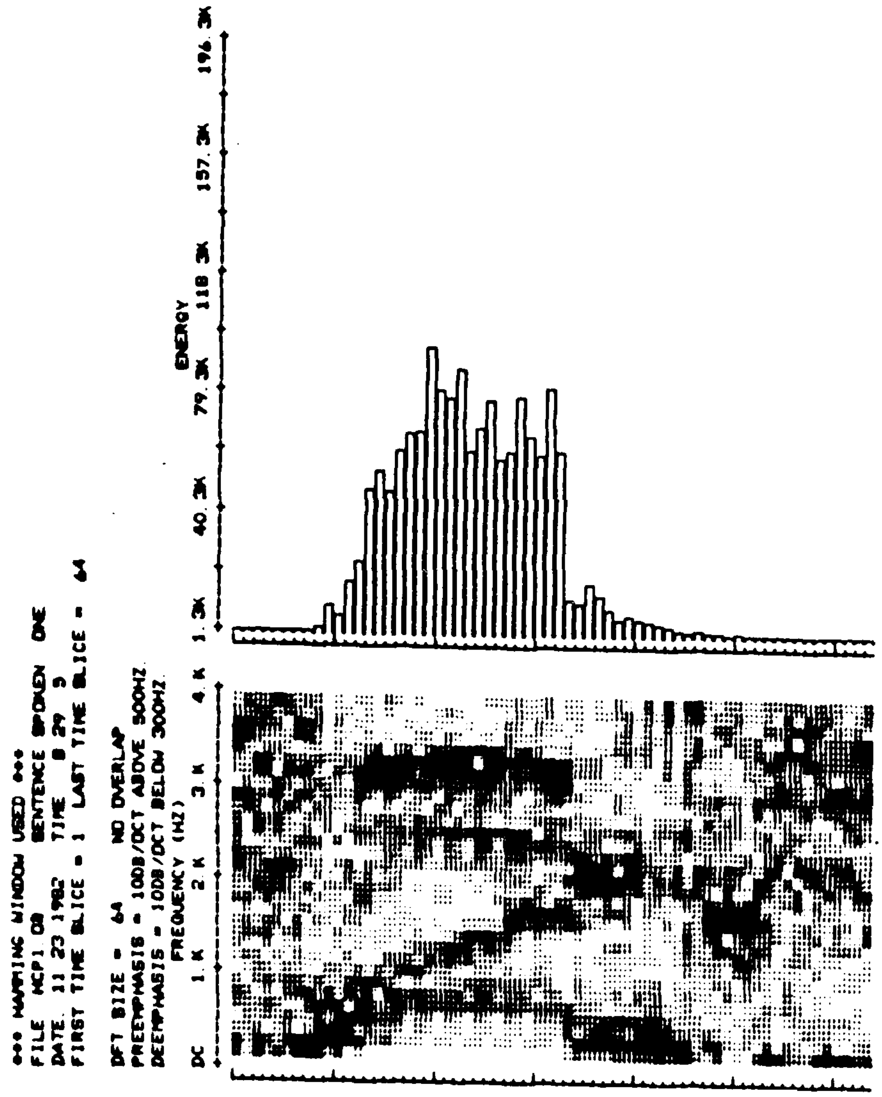


Figure 4. Spectrogram of "ONE"

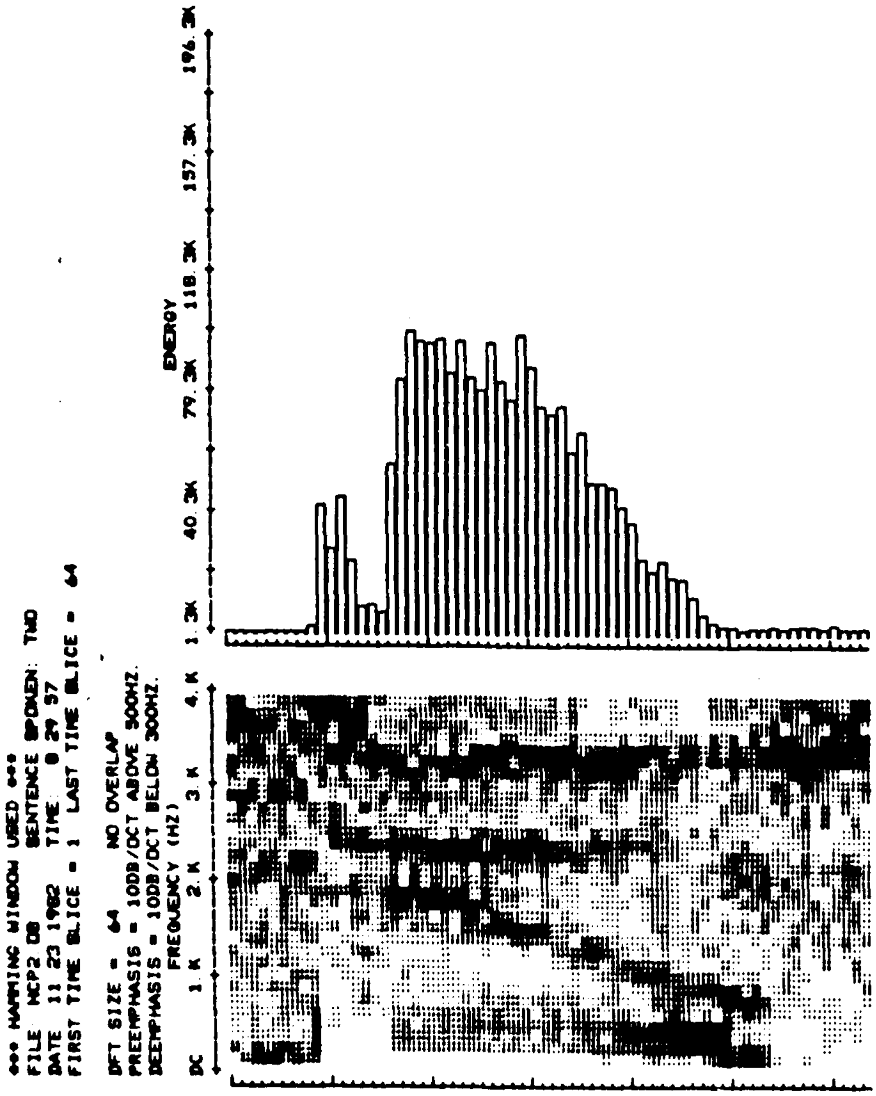


Figure 5. Spectrogram of "TWO"

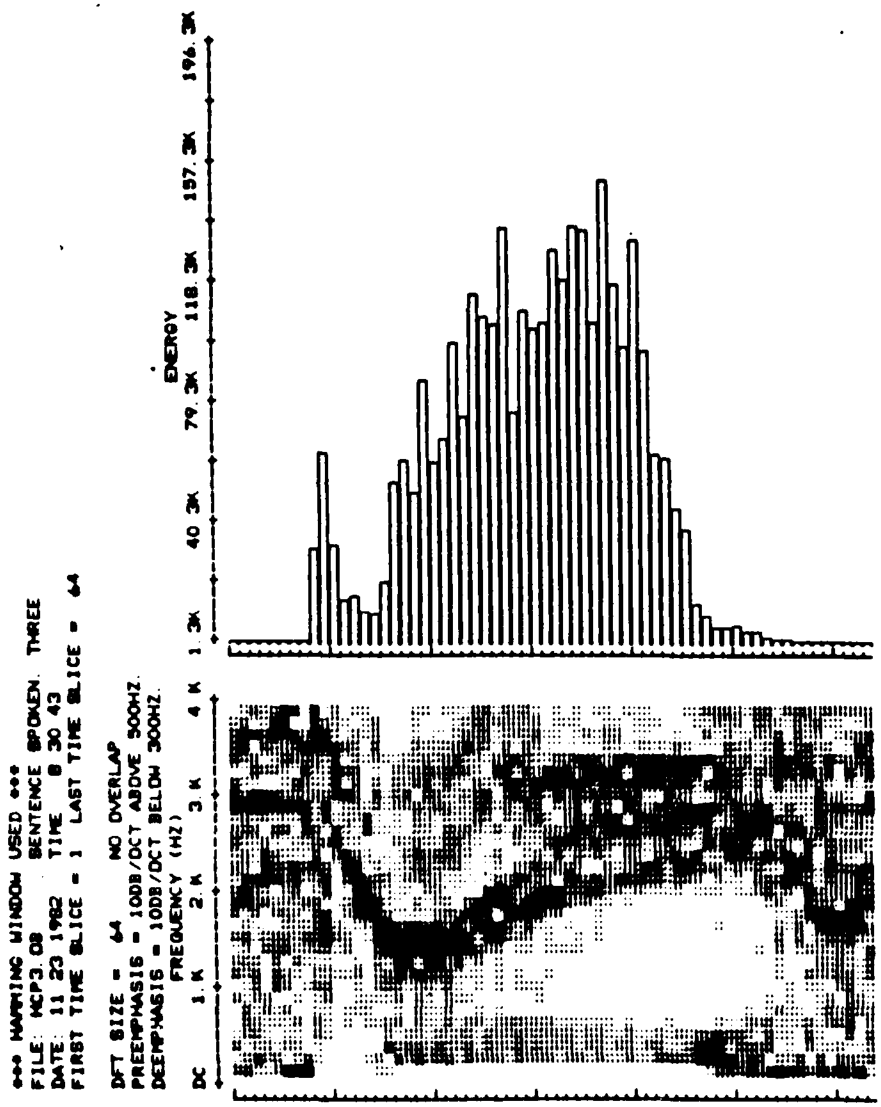


Figure 6. Spectrogram of "THREE"

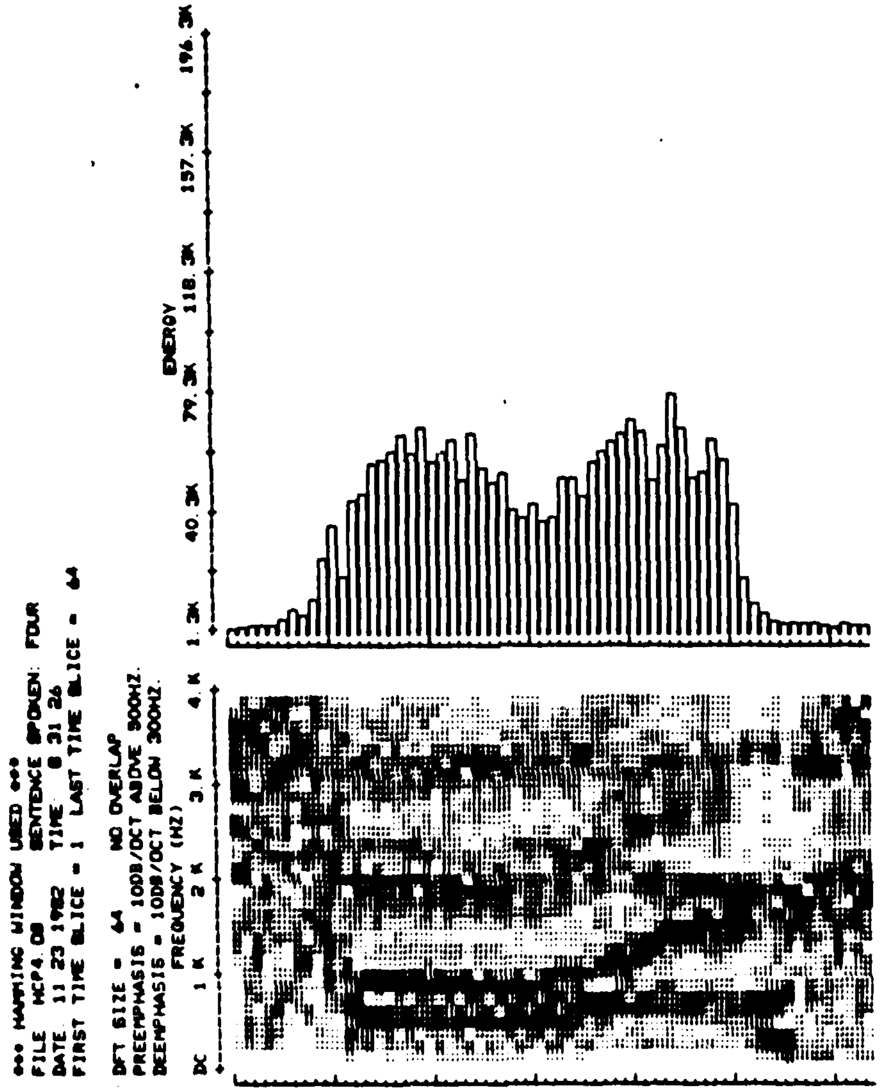


Figure 7. Spectrogram of "FOUR"

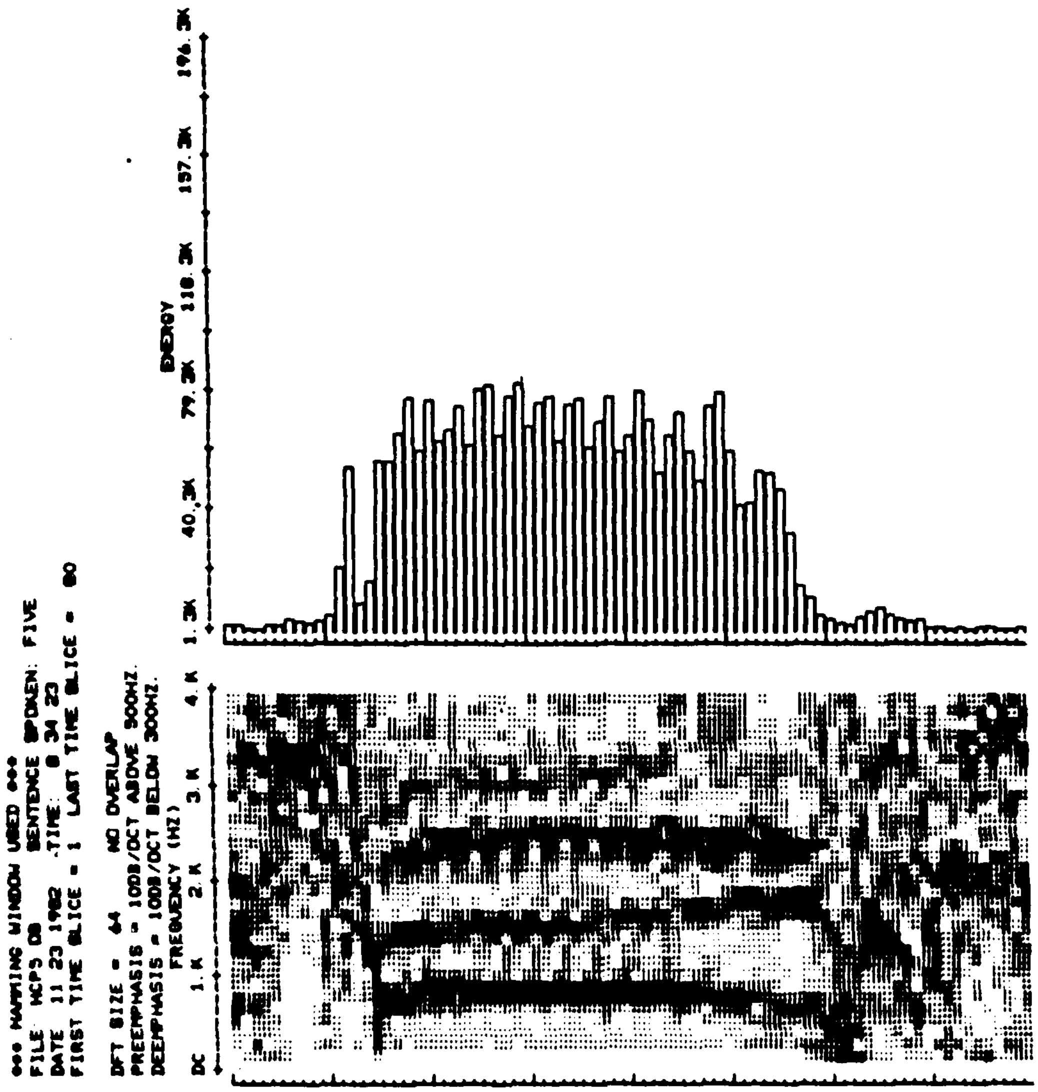


Figure 8. Spectrogram of "FIVE"

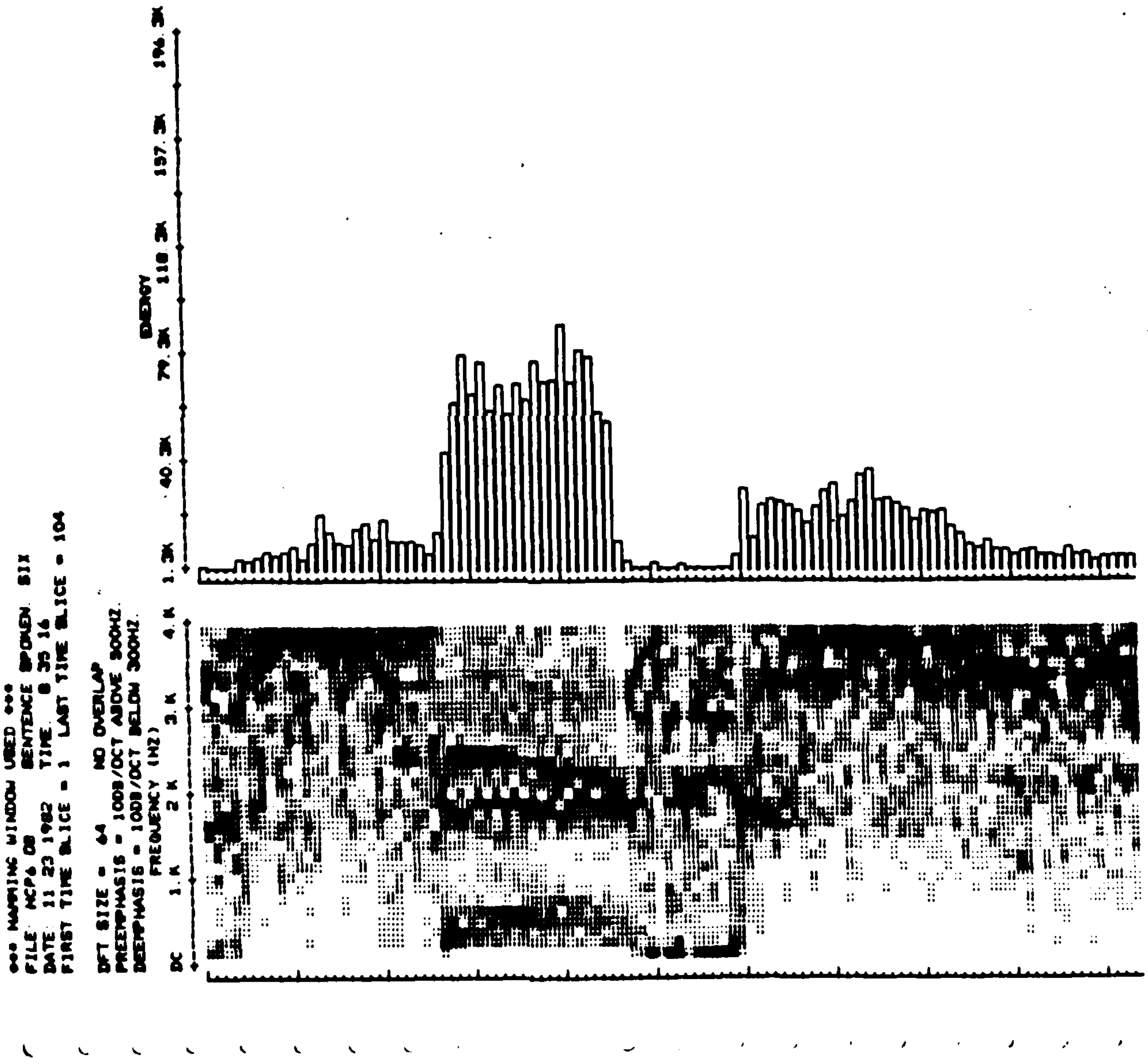


Figure 9. Spectrogram of "SIX"

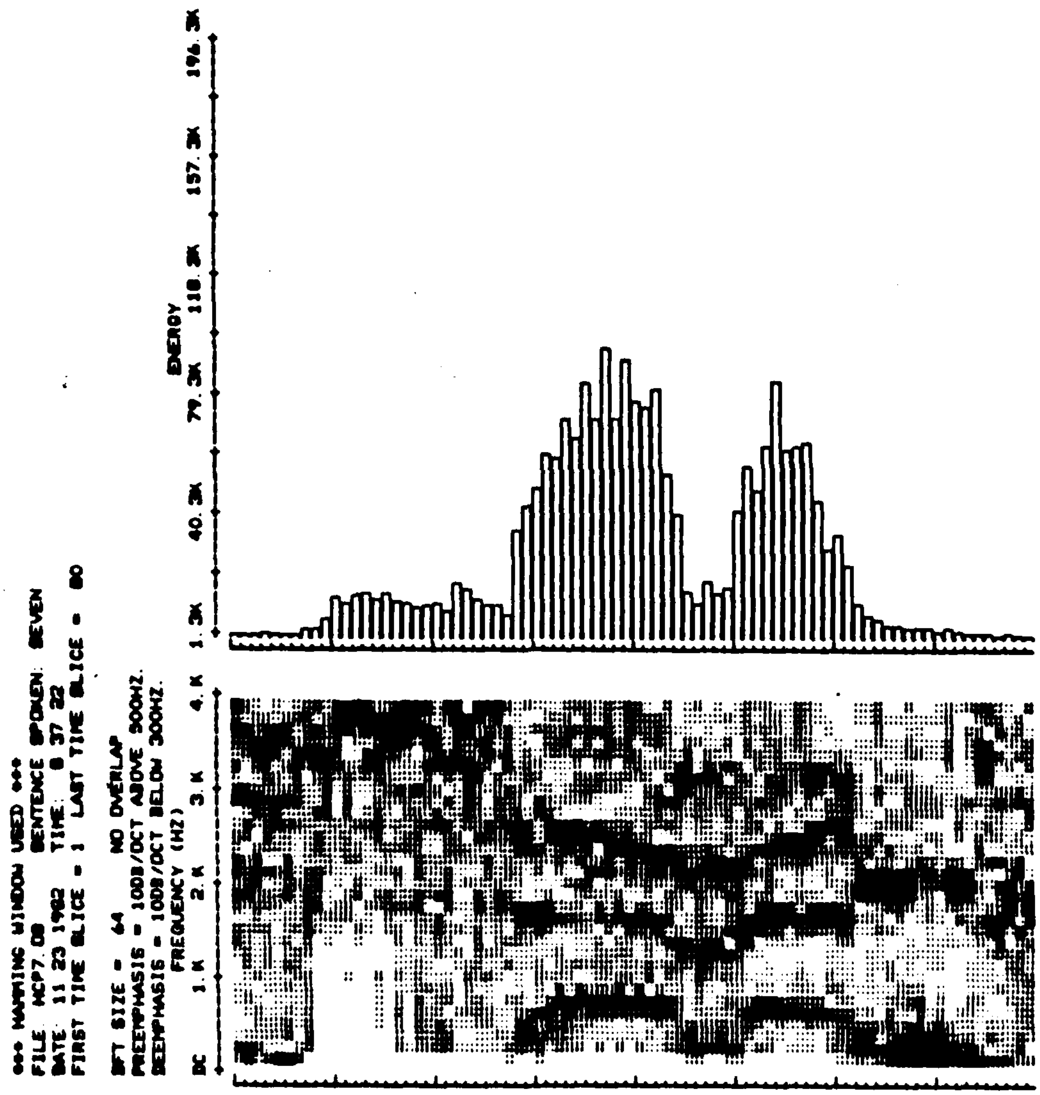


Figure 10. Spectrogram of "SEVEN"

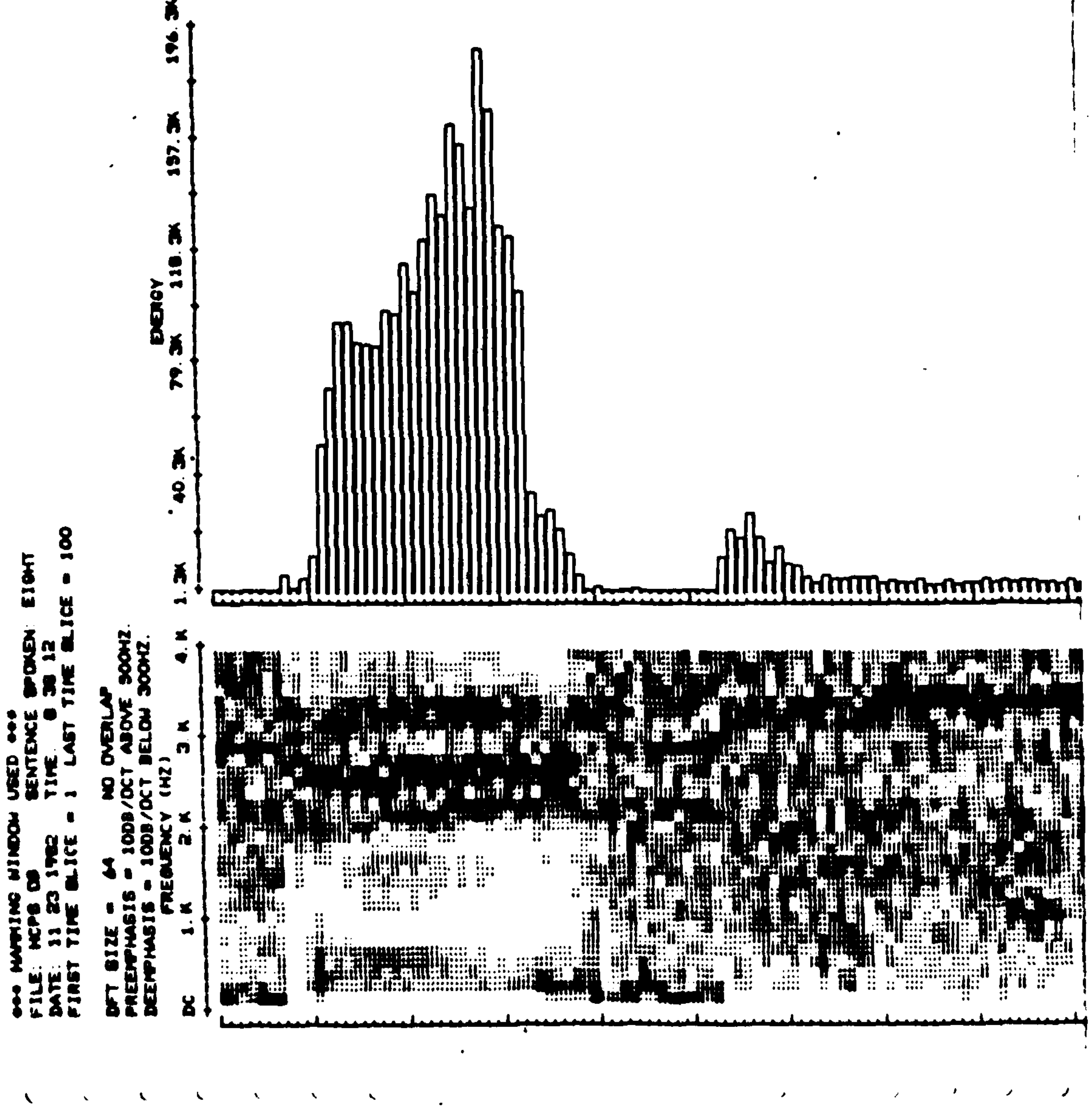


Figure 11. Spectrogram of "EIGHT"

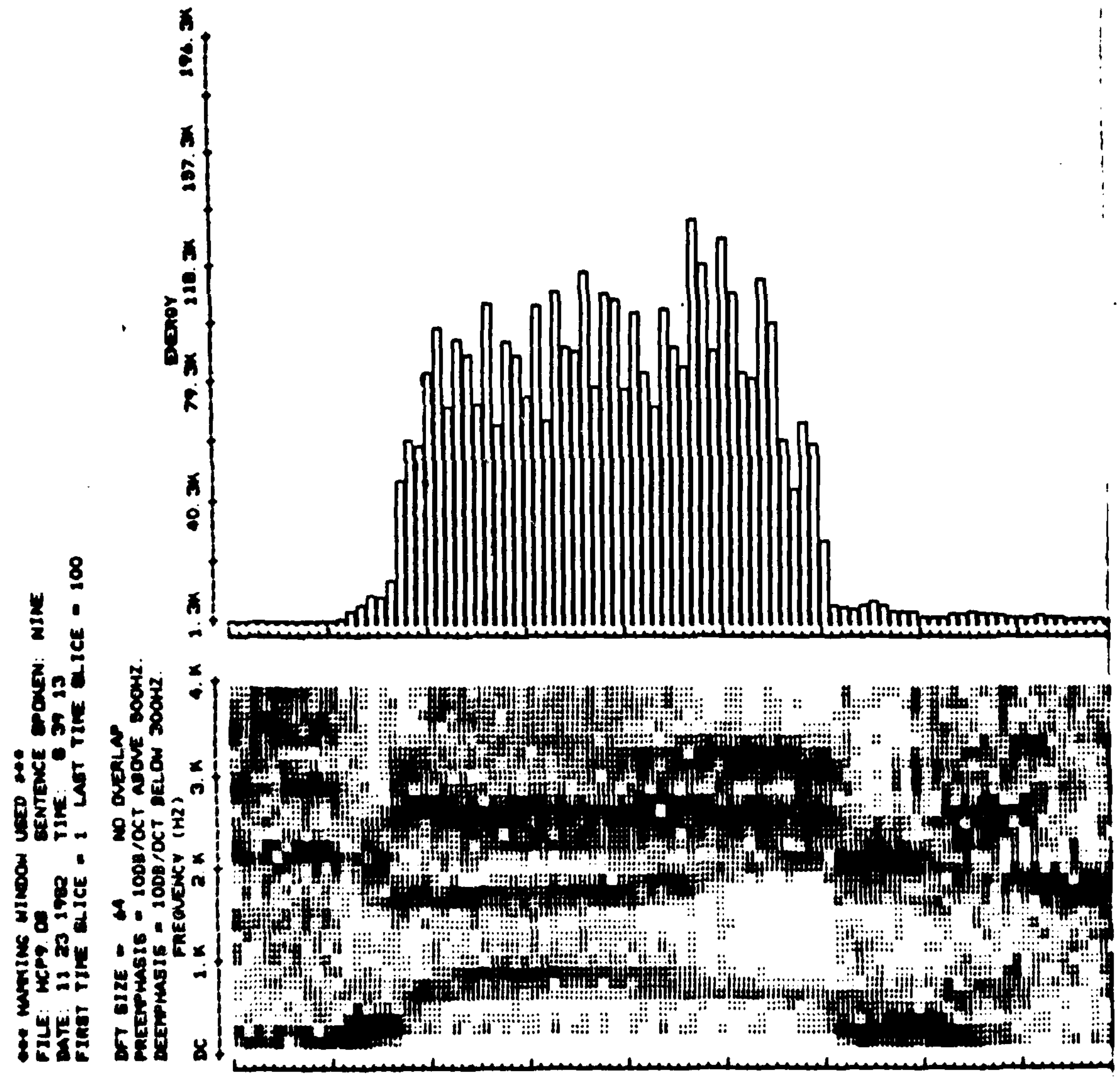


Figure 12. Spectrogram of "NINE"

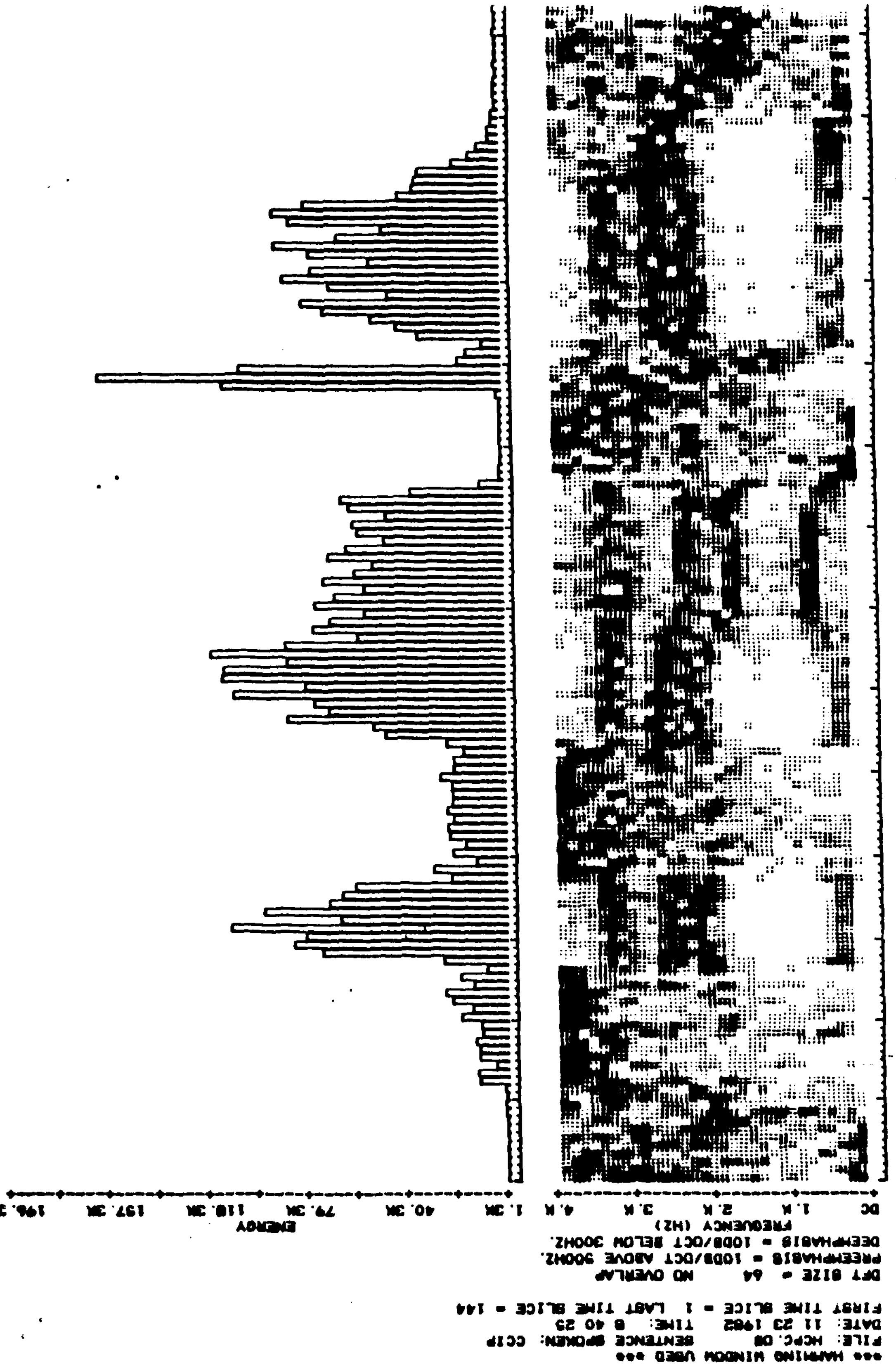


Figure 13. Spectrogram of "CCIP"

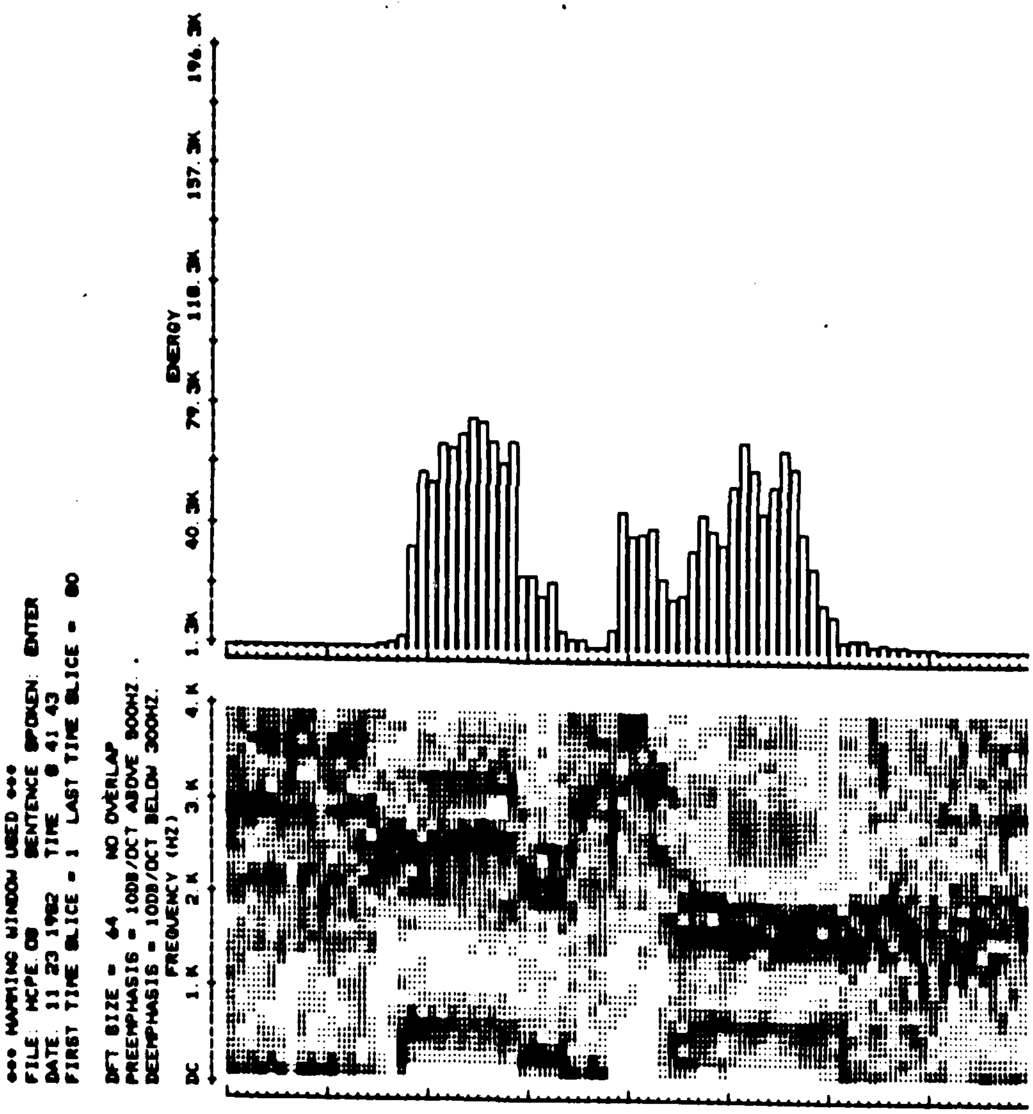


Figure 14. Spectrogram of "ENTER"

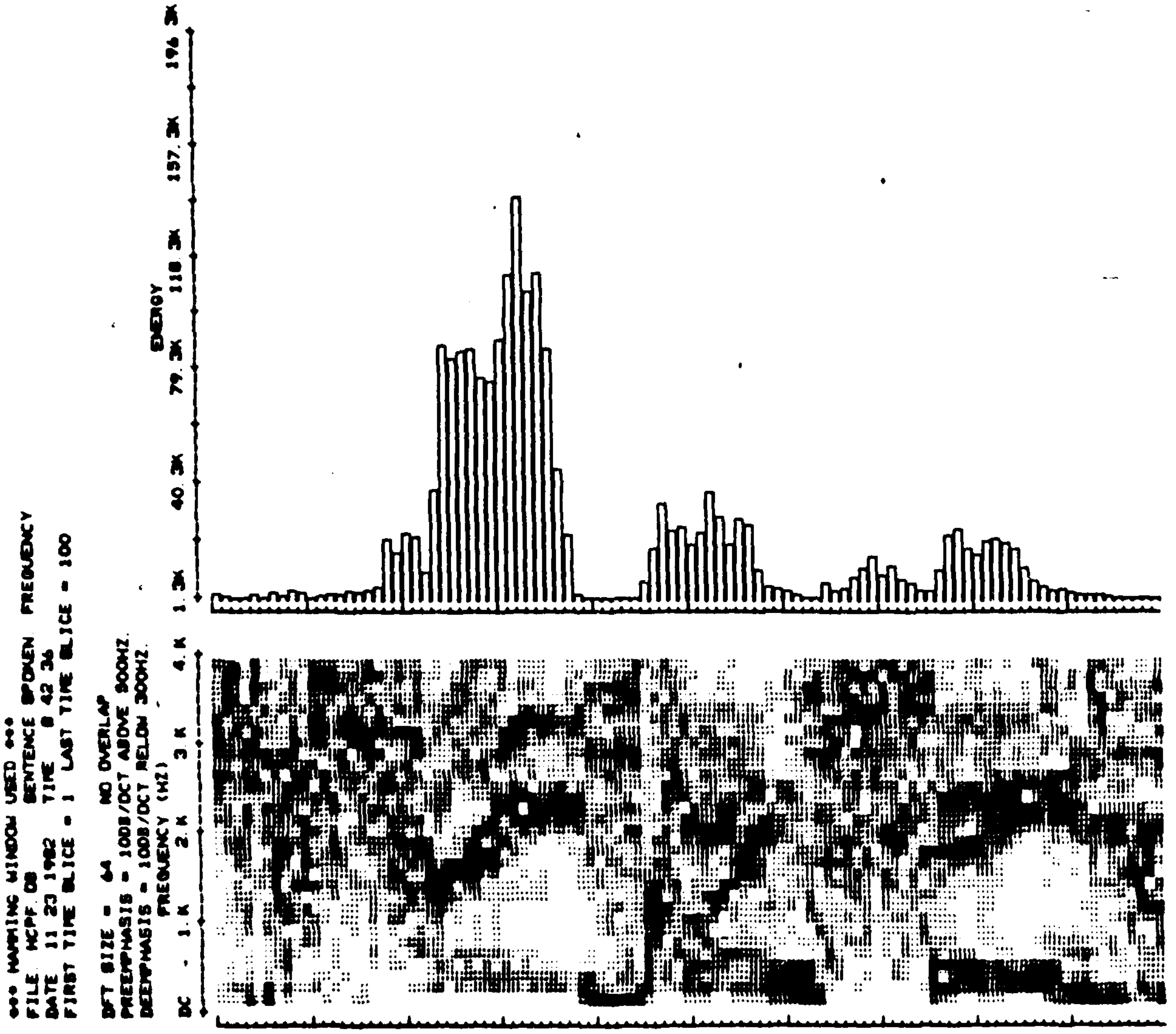


Figure 15. Spectrogram of "FREQUENCY"

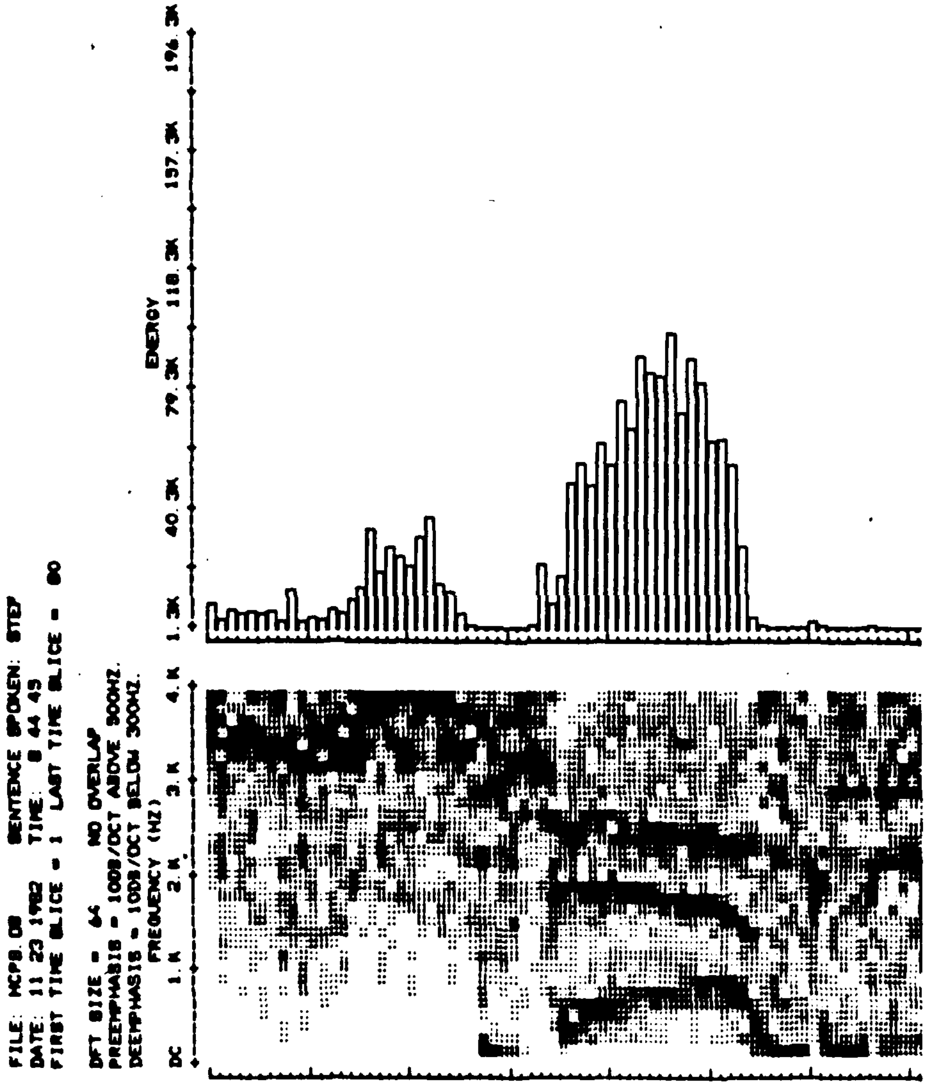


Figure 16. Spectrogram of "STEP"

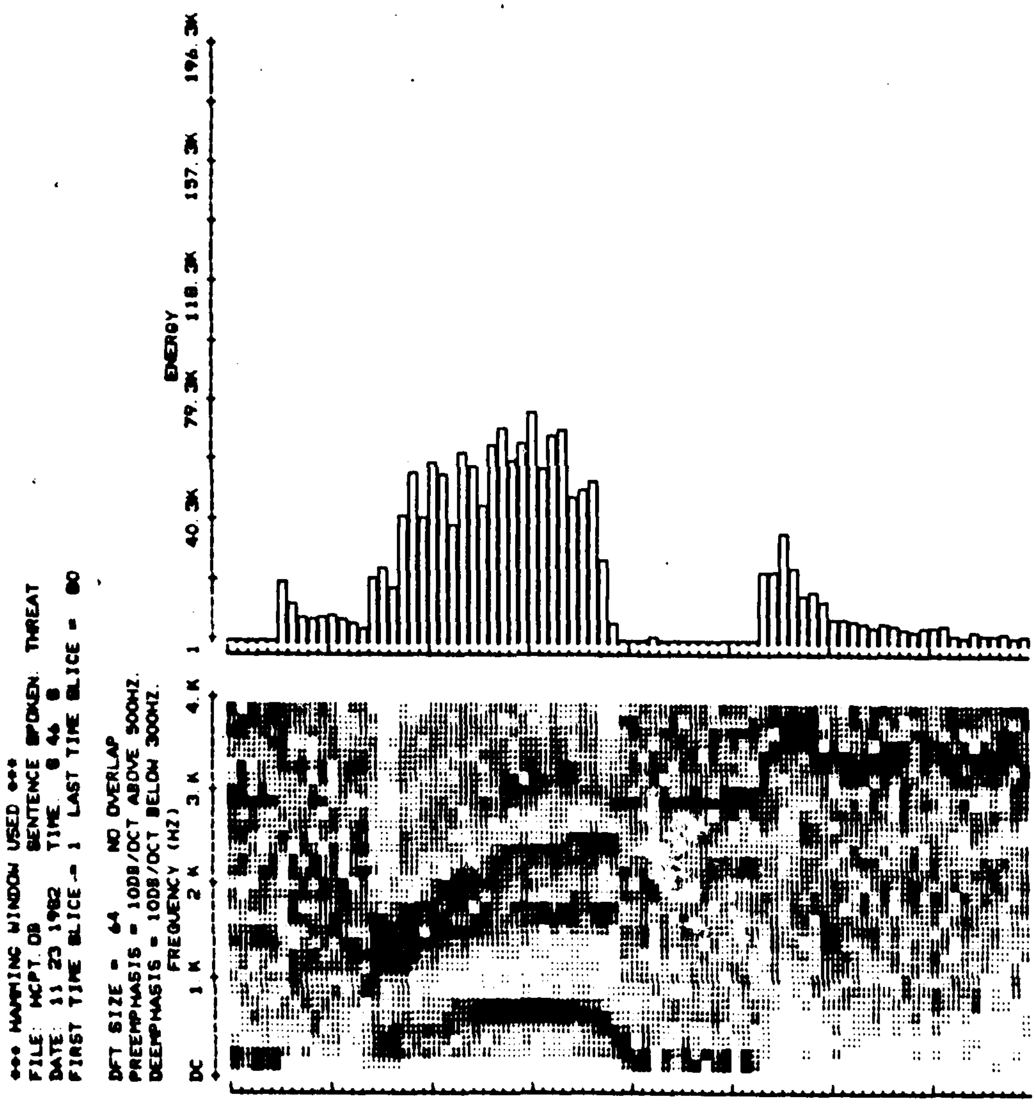


Figure 17. Spectrogram of "THREAT"

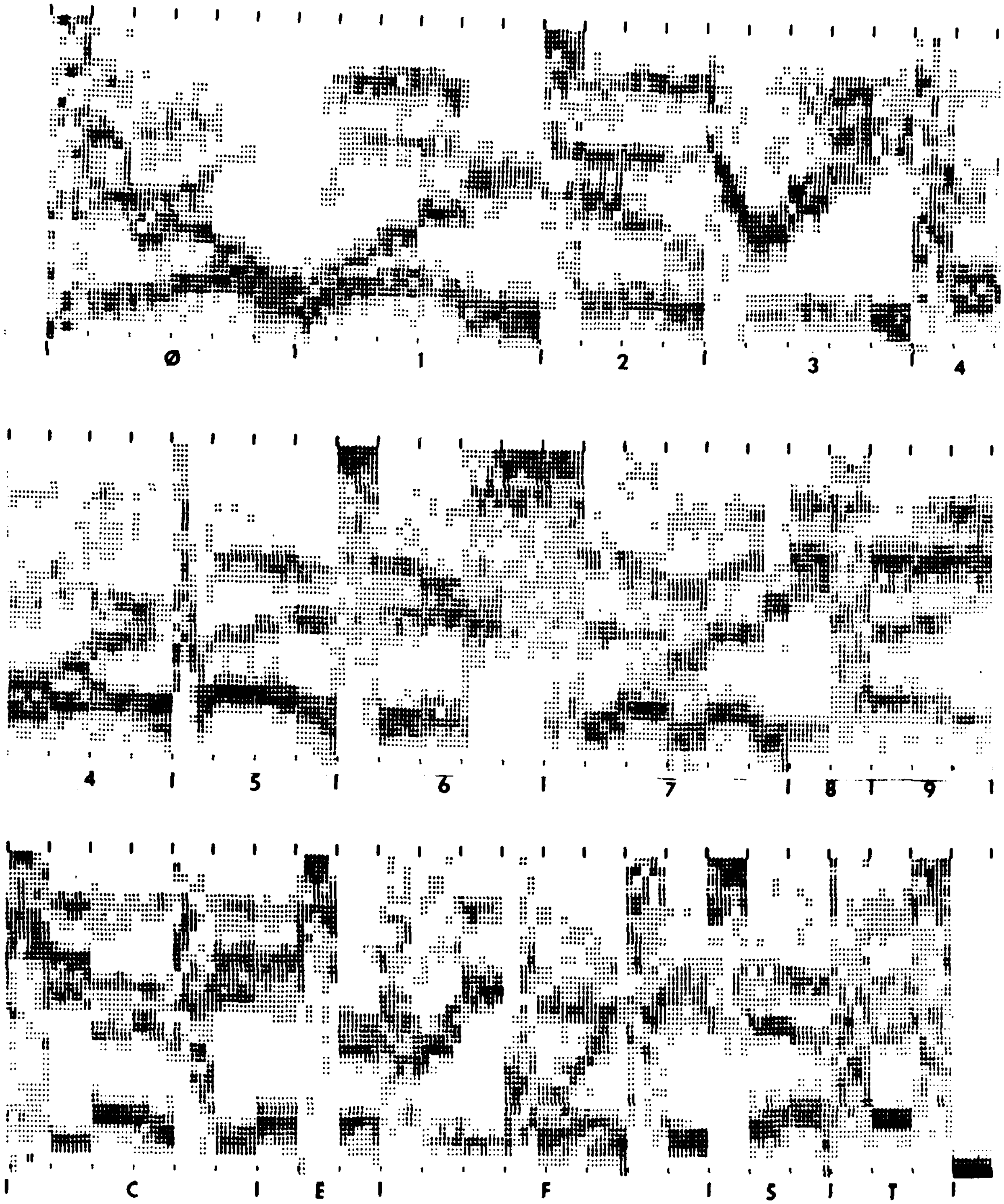


Figure 18. 5-vector Template

the phoneme to be added, find it in the speech, and add it to the template. The above procedure is done for several phoneme sounds in each word until the template is filled out from the various speech files. After the template is formed, it is available to the program that finds the distance between the input speech and the phoneme template.

One-Vector Phonemes. Templates of single-vector phonemes were studied using new programs developed by Martin. Martin's programs used the array processor and extended memory available in the Eclipse S/250 computer. His programs used one-vector phoneme templates and found the distance between these templates and input speech files. In addition, Martin's programs can be changed easily to study different size DFT's, change preemphasis, or deemphasis as needed for speech study.

Three programs were developed to interface and use Martin's programs in the word recognition cycle. Single-vector phonemes were developed using the same files as used for five-vector phonemes (Figures 3 thru 17). Phonemes were picked to be as close to the five-vector phoneme as possible. In Table I you can see that the one-vector phonemes came from almost the same vectors. In some cases, one or two vectors were left off from the one-vector phoneme in order to have the spectrogram characteristics more uniform. Figure 19, is a spectrogram of the single-vector phonemes. The single-vector phoneme template includes twelve noise vectors which were treated as one phoneme by

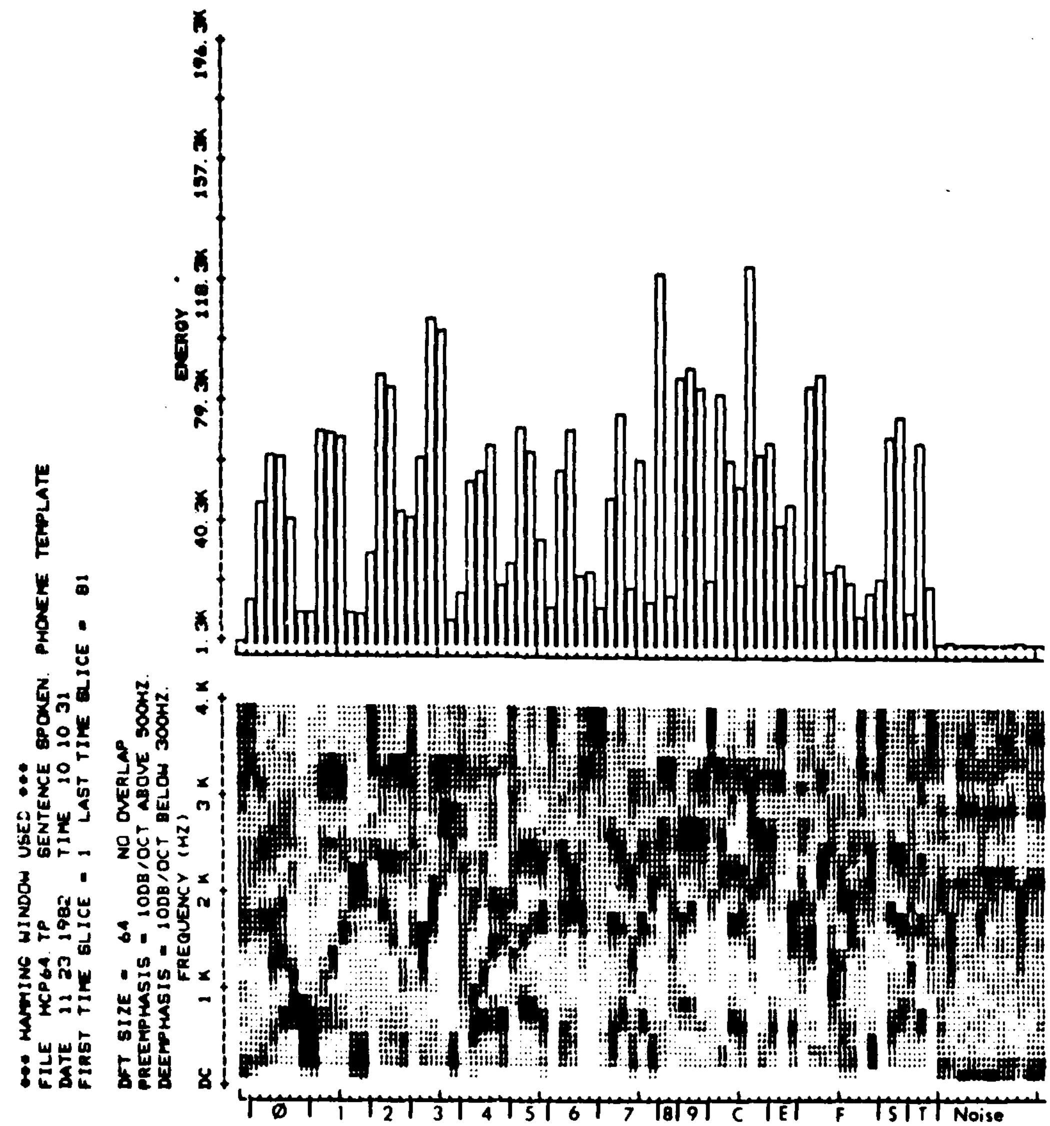


Figure 19. Single-Vector Phoneme Template

the feature extraction program. The five-vector phonemes had only two noise vectors.

Thresholding

Thresholding was used to find the beginning and ending of words for isolated word recognition. Seelandt's programs, which were used to study the five-vector phoneme templates, used a simple thresholding to find the beginning and end of words. When a threshold is set in Seelandt's program, any sound below that threshold is muted and represented by no frequency components in the output. Thus, when any distance routine was used, the phoneme that was most like the thresholded material was phoneme number one. Phoneme number one consisted of very few frequency components (the least amount in the phoneme template). Thus, if the thresholding worked properly, it looked as if Seelandt's program picked noise from the file very well. This simple threshold technique was effective in laboratory speech, but not effective for the speech used in this study.

For studying one-vector phoneme templates there is no thresholding incorporated to set the frequency components to zero. Thresholding for single-vector work was done after the distance routine and was used to find the beginning and end of words by using the energy in each vector. However, it was found that a simple thresholding technique did not do very well on speech which was recorded using a mask, because of breathing and exhaling. The threshold would sometimes set the beginning and end of words erroneously, when breathing or exhaling exceeded the threshold level set. Therefore, words could be represented by a feature extraction string much longer than the actual word should have been. In order to minimize this problem, a simple algorithm was devised to ignore short transients above the set threshold. The new thresholding algorithm would ignore transients shorter than five vectors (40ms). This algorithm worked better than simple thresholding and can be found in program TOP5 which prepares feature extraction files for the recognition routine program LEARN.

Distance Rule

After the phoneme template is formed the input speech can be entered into the program which finds the distance between each vector of speech and every phoneme in the template. The distance routines are seen in Table II below.

Table II. Minkowski Distance and Computational Load

M1 "CITY BLOCK" $\sum_{j=1}^{n} |\underline{X}_j - \underline{Y}_j|$

M2 EUCLIDEAN $[\sum_{j=1}^{n} (\underline{X}_j - \underline{Y}_j)^2]^{1/2}$

COMPUTATIONAL DIFFERENCES	M1	M2
ADD & SUB	2n	2n
MULTIPLIES	Ø	n
SQRT	Ø	1

The two different distance rules in Table II were studied by this research and are based on two cases of the Minkowski distance rule. Seelandt's programs, using the five-vector phonemes, used the M1 distance rule. The M1 and M2 distance rules were used for single-vector phonemes. The results of the single phonemes will be used to compare the M1 and M2 distances.

The distance routines were used to find the distance between a phoneme template and the speech files. The distance was found between the frequency components (32 x 5 array), of the five-vector phoneme template, and the equivalent number of components in the speech. The phoneme represented 40 ms of speech and distances were calculated at each 8 ms interval on the speech input. The single-vector phoneme templates represented 8 ms of speech. The distance was calculated for each vector of speech (32 frequency components) against the same number of components in the phoneme.

Five Top Choices

The end product of the feature extraction system is five top choices of phonemes for each 8 millisecond of input speech. In addition, each of these five choices will be scaled from 100 to zero. The top choice will be 100 and the last phoneme choice (not the fifth choice) would correspond to 0. Since only the top five choices will be seen the scale usually shows 100 for the top phoneme and 80 to 90 or even 50 for the fifth phoneme choice. In addition, there is a scale factor for each vector (8 ms) of speech. The scale factor is calculated as follows:

$$\text{SCALE FACTOR} = \frac{\text{Vector minimum phoneme distance}}{\text{Maximum }\{\text{minimum phoneme distance in file}\}}$$

The five top choices use the following formula to scale each of the five choices in a vector:

$$\text{SCALE} = \frac{\text{VECTOR MAXIMUM DISTANCE} - \text{CHOICE DISTANCE}}{\text{VECTOR MAXIMUM DISTANCE} - \text{VECTOR MINIMUM DISTANCE}}$$

This scale was used because program LEARN uses this scale in formation for word scoring (Ref 3).

The programs used by Seelandt to process each speech file in the feature extraction system consisted of programs called TRYDIST5 and LISTER4. TRYDIST5 and LISTER4 were modified to output the data as listed in Figure 22. TRYDIST5 and LISTER4 were modified by Montgomery, and renamed PHDIST and CHOICE5 respectively. To use Martin's programs, the program TOP5 was developed to link his program to the recognition algorithm program developed by Montgomery. In addition to listing the top five choices and the scale factor, program TOP5 also lists the energy for each vector in the speech file. Figure 20 shows output from the program TOP5.

After speech inputs are processed by the feature extraction system, the output from the feature extraction system is used for the recognition program described in the next chapter.

FOUR
H504.SP

THE DATE IS-- 11 17 1982
THE TIME IS-- 13 30 21

VECTOR NUMBER	FIRST CHOICE		SECOND CHOICE		THIRD CHOICE		FOURTH CHOICE		FIFTH CHOICE		SCALE FACTOR	VECTOR ENERGY
******	*******		*******		*******		*******		*******		*******	*******
15	44	100	63	98	35	97	52	93	48	92	.68725870	555
16	57	100	18	99	44	96	1	94	68	93	.79922780	840
17	25	100	39	82	1	80	61	73	67	71	.69884170	794
18	25	100	39	79	6	76	68	70	30	69	.64092660	939
19	39	100	25	97	6	85	68	82	30	81	.64092660	1004
20	63	100	48	99	39	96	44	96	14	95	.74131270	1074
21	14	100	63	96	48	96	70	92	37	91	.76447870	1184
22	39	100	70	100	14	96	63	92	18	90	.88416990	1203
23	39	100	18	84	30	84	6	83	23	81	.76061770	1156
24	39	100	18	92	45	91	46	91	23	91	.82625480	1221
25	39	100	18	93	23	92	45	91	46	91	.78764470	1240
26	39	100	70	96	37	96	63	94	48	92	.84942080	1101
27	39	100	45	83	54	81	30	81	46	79	.60617760	902
28	39	100	52	95	54	88	45	83	46	83	.76061770	1006
29	52	100	39	92	54	91	46	87	18	85	.71042470	1107
30	46	100	54	98	47	96	18	96	9	95	.83397680	1208
31	18	100	23	98	44	97	50	95	9	94	.83783780	1300
32	23	100	18	99	1	95	1	95	52	92	.74517370	1184
33	23	100	68	85	44	83	60	80	63	78	.59845550	1139
34	23	100	39	99	60	96	30	93	29	93	.82239380	1072
35	60	100	9	92	23	89	29	88	39	86	.79150580	1070
36	23	100	9	97	21	94	10	93	6	92	.86486480	1153
37	9	100	21	92	47	88	43	86	10	86	.68339770	1155
38	6	100	29	99	45	85	60	79	30	78	.70270270	903
39	45	100	23	88	60	87	6	86	68	85	.94594590	1164
40	23	100	44	97	68	93	63	93	60	89	.83011580	1194
41	60	100	23	86	68	83	18	82	28	82	.74131270	1047
42	10	100	18	94	44	91	70	91	23	91	.83011580	1301
43	18	100	29	95	60	93	28	83	23	83	.98841700	1034
44	10	100	11	99	23	91	28	91	29	90	.89961390	1084
45	29	100	19	87	11	86	60	78	16	77	.83397680	879
46	29	100	11	86	67	78	41	78	50	74	.67567560	681
47	29	100	50	90	19	87	41	86	11	86	.89575280	815
48	29	100	67	100	39	88	30	87	1	86	.84555980	611
49	26	100	50	86	69	83	5	82	29	79	.76061770	579
50	67	100	50	94	41	91	58	88	69	88	.76447870	679
51	1	100	56	95	67	88	29	86	62	86	.90733590	357
52	1	100	1	93	56	91	27	85	26	85	.81467180	224
53	27	100	24	92	62	90	56	87	1	84	.83011580	120
54	1	100	39	96	29	93	67	89	1	88	.90733590	52

Figure 20. Feature Extraction Output

IV. Word Recognition Algorithm

The word recognition algorithm used in this research was developed by Gerard Montgomery and uses fuzzy set theory for isolated word recognition (Ref 3). To use the word recognition program, files formatted in the exact form of Figure 20 or Figure 22 are stored in files to be used by program LEARN. Program LEARN prompts the user for a phoneme representation when each new word is encountered.

Phoneme Representation

Each word to be recognized by program LEARN must have a phoneme representation. This representation can be set according to the features extracted earlier. Phoneme #2 through phoneme #7 represent the word "zero" in the phoneme template. The logical phoneme representation for "zero" would be 2-3-4-5-6-7. However, when program LEARN scores a file against a phoneme representation it may delete some of the phonemes which are listed in the phoneme representation. These deletions can degrade the algorithm's preformance. Therefore, a phoneme representation was picked to minimize the number of deletions found when statistics are gathered by program LEARN.

The first step in picking a phoneme representation is to pick an initial phoneme representation for each word. These representations were used to gather statistics on 45 input files. The 45 inputs files consisted of the vocabulary "zero" through "nine", "CCIP", "enter", "frequency", "step", and "threat". After statistics are gathered, the phoneme representation can be changed by picking a representation that minimizes the number of deletions. Next program LEARN is used to recognize the same 45 training files. After the recognition results were obtained one could see how well the phoneme representation did. An example of minimizing the deletions follows: for the word "zero" a phoneme representation is picked to include 2-3-4-5-6-7 and it is found phoneme 5, 6 and 7 were deleted three times, it is possible to eliminate just the phoneme 5 and have zero deletions for the new phoneme representation 2-3-4-6-7. Phoneme representations were developed for all fifteen utterances based on the trial and error techniques discussed above. The trial and error techniques were only used for the training files and only for the five-vector phoneme templates. The single-vector phoneme template would use the same phoneme representations used for five-vector. The phoneme representation given in Table C was picked after several trials which consisted of changing the phoneme representation, collecting statistics, and running recognition results for the training set. Recognition results of 100% on the training file were obtained, then word recognition results were run on a new set of 45 speech files (non-training files) to find the actual performance of the recognition program.

TABLE III

PHONEME REPRESENTATION USED FOR VOCABULARY

IN WORD RECOGNITION ALGORITHM

WORD	PHONEME REPRESENTATION
ZERO	2 - 3 - 6 - 7
ONE	8 - 9 - 10 - 12 - 13
TWO	14 - 15 - 17 - 7
THREE	18 - 19 - 21 - 22
FOUR	23 - 24 - 25 - 27
FIVE	28 - 29 - 30 - 31
SIX	32 - 33 - 1 - 35 - 36
SEVEN	37 - 39 - 40 - 42 - 13
EIGHT	43 - 1 - 44
NINE	13 - 29 - 47 - 13
CCIP	37 - 49 - 36 - 49 - 53 - 22
ENTER	54 - 13 - 56
FREQUENCY	28 - 33 - 1 - 13 - 64 - 53 - 22
STEP	65 - 1 - 66 - 67
THREAT	68 - 19 - 69 - 70

Program LEARN

Program LEARN, the word recognition program, uses a set of fuzzy variables for recognition scoring. These fuzzy variables can be different for each word. The fuzzy variables used in this research are listed in Figure 21. The values listed in Figure 21 are variables which can be changed to improve the performance of program LEARN. The allowable limits for each variable and their meaning can be found in Montgomery's thesis (Ref 3). These values presented in Figure 21 would have to be used to duplicate the results in this report.

```
THE OVERALL FUZZY VARIABLES THAT WERE USED FOLLOW

STHR =  1.ØE+ØØ  SUBE =  1.ØE+ØØ  SUBF =  5.ØE-Ø1
INSE =  1.3E+ØØ  INSF =  5.ØE-Ø1
DELE =  1.ØE+ØØ  DELF =  8.ØE-Ø1  DELG =  1.ØE-Ø1
DCNE =  1.ØE+ØØ  DCNF =  1.2E+ØØ  DCNG =  5.ØE-Ø1
SFE  =  2.ØE+ØØ  SFF  =  2.ØE+ØØ
CHVE =  4.ØE+ØØ  CHVF =  2.5E-Ø1
STATE=  1.ØE+ØØ  STATF=  3.ØE+ØØ  STATG=  Ø.ØE+ØØ
THR1E=  1.ØE+ØØ  THR1F=  7.5E-Ø1
THR2E=  1.ØE+ØØ  THR2F=  5.ØE-Ø1

            THE WØRD FUZZY VARIABLES FØLLØW

WSTHR =  8.ØE-Ø1  WSUBE =  1.ØE+ØØ  WSUBF =  5.ØE-Ø1
WINSE =  1.3E+ØØ  WINSF =  5.ØE-Ø1
WDELE =  1.ØE+ØØ  WDELF =  8.ØE-Ø1  WDELG =  1.ØE-Ø1
WDCNE =  1.ØE+ØØ  WDCNF =  1.2E+ØØ  WDCNG =  5.ØE-Ø1
WSFE  =  2.ØE+ØØ  WSFF  =  2.ØE+ØØ
WCHVE =  4.ØE+ØØ  WCHVF =  2.5E-Ø1
WSTATE=  1.ØE+ØØ  WSTATF=  3.ØE+ØØ  WSTATG=  7.ØE-Ø1
WTHR1E=  1.ØE+ØØ  WTHR1F=  7.5E-Ø1
WTHR2E=  1.ØE+ØØ  WTHR2F=  5.ØE-Ø1
```

Figure 21. Fuzzy Variables Used For All Words

V. Results

The results of five-vector phoneme templates using M1 distance are listed in Table IV along with the results of one-vector phoneme for both M1 distance and M2 distance. The results are similar for five-vector M1 distance, one-vector M1, and M2 distances for normal speech (remember all speech was from subjects wearing a mask). Five-vector and one-vector phoneme templates have different results in feature extraction and recognition when G-speech is used. From Table IV it can be seen that G-speech has higher recognition scores for the five-vector phoneme template than the one-vector template.

The recognition files listed in Table IV are labeled C for control files, 3 for 3g files, and 5 for 5g files. In Tables V - VIII the following A, B, P, C, 3, and 5 represent different speech files. Files A, B, P, and C are speech files at no G-stress (control conditions). Files 3 and 5 are speech files at 3g and 5g respectively.

Table IV

RECOGNITION RESULTS

EXPERIMENT/LENGTH/ PHONEME/DISTANCE RULE	TRAINING FILES(45)	RECOGNITION FILES(15 ea.)		
		C	3	5
1 / 5 / M1	100%	93%	100%	80%
2 / 1 / M1	98%	93%	53%	33%
3 / 1 / M2	93%	93%	27%	27%
4 / 1 / M2	91%	93%	47%	33%

(experiment 4 used different fuzzy variables and phoneme representations for some words)

Tables V thru VIII have recognition scores listed for all the words and experiments. Scores were similar for five- vector and one-vector phoneme templates as seen by the recognition scores. Recognition results suffer when the words not to be recognized score higher than the actual word to be recognized.

Studying the input files to the recognition system (program LEARN), gave insight into why the scores increased for the words not wanted. For G-speech the five-vector phoneme templates gave a more consistent output than the output from single-vector phoneme templates.

Figure 22 (five-vector) and Figure 23 (one-vector) are the output files from the feature extraction system for the word "eight" at five G's. The first five-vectors in both the processes that correspond to the same time vectors, show the five-vector file to contain five different phonemes in the top five choices, whereas the single-vector phoneme template feature extraction listed in Figure 23 shows twelve different phonemes in the first five-vectors. In addition, the five-vector phonemes are more consistent in the representation.

The recognition algorithm looks at the single-vector phoneme template feature extraction file, the word eight as represented by Figure 23, and tries to score each of the vocabulary words against this file (Figure 23). The single-vector files are more inconsistent and therefore the other words score higher than five-vector files based on what the

TABLE V

RECOGNITION SCORES FOR EXPERIMENT 1

Phoneme Length: 5 vector

Distance Rule: M1

WORDS TO BE RECOGNIZED	FILES: TRAINING SET			RECOGNITION SET		
	A	B	P	C	3	5
ZERO	.76	.83	.80	.80	.73	.67
ONE	.85	.83	.81	.84	.64	.62*
TWO	.78	.86	.88	.87	.71	.74
THREE	.86	.73	.86	.78	.82	.67
FOUR	.85	.74	.87	.66*	.72	.45*
FIVE	.83	.86	.87	.82	.68	.64
SIX	.84	.88	.88	.85	.76	.68
SEVEN	.84	.78	.85	.80	.68	.60*
EIGHT	.86	.89	.88	.86	.84	.83
NINE	.82	.83	.87	.82	.76	.76
CCIP	.82	.86	.80	.76	.76	.77
ENTER	.85	.84	.84	.84	.78	.75
FREQUENCY	.82	.77	.65	.82	.65	.69
STEP	.82	.88	.88	.85	.79	.69
THREAT	.82	.75	.83	.71	.73	.70
Percent Correct	100	100	100	93.3	100	.80
MEAN		.829		.805	.788	.684
STANDARD DEVIATION		.049		.058	.061	.089

*Word missed

TABLE VI

RECOGNITION SCORES FOR EXPERIMENT 2

Phoneme Length: 1 vector

Distance Rule: M1

WORDS TO BE RECOGNIZED	FILES: TRAINING SET			FILES: RECOGNITION SET		
	A	B	P	C	3	5
ZERO	.81	.86	.74	.82	.64*	.63*
ONE	.83	.86	.8Ø	.81	.64	.6Ø*
TWO	.81	.84	.86	.83	.65*	.63*
THREE	.86	.81	.8Ø	.76	.76	.67*
FOUR	.79	.83	.84	.68*	.69*	.54*
FIVE	.85	.88	.83	.85	.56*	.62*
SIX	.82	.84	.83	.78	.67*	.68*
SEVEN	.84	.79	.81	.77	.71*	.69*
EIGHT	.89	.89	.92	.87	.83	.79
NINE	.83	.83	.8Ø	.82	.68*	.65*
CCIP	.77	.81	.82	.8Ø	.75	.73
ENTER	.77	.78	.81	.76	.75	.7Ø
FREQUENCY	.78	.84	.77	.79	.75	.72
STEP	.84	.89	.79*	.81	.75	.69*
THREAT	.71	.81	.83	.75	.73	.74
Percent Correct	1ØØ	1ØØ	93.3	93.3	53.3	33.3
MEAN		.822		.793	.7Ø4	.672
STANDARD DEVIATION		.Ø41		.Ø46	.Ø67	.Ø63

*Word missed

TABLE VII

RECOGNITION SCORES FOR EXPERIMENT 3

Phoneme Length: 1 vector

Distance Rule: ·M2

WORDS TO BE RECOGNIZED	FILES					
	TRAINING SET			RECOGNITION SET		
	A	B	P	C	3	5
ZERO	.81	.85	.70*	.82	.65*	.64*
ONE	.83	.84	.80	.81	.65	.66
TWO	.81	.76	.85	.77	.58*	.66*
THREE	.86	.75*	.80	.74	.74*	.67*
FOUR	.82	.82	.84	.68*	.71*	.60*
FIVE	.81	.85	.88	.81	.69*	.62*
SIX	.85	.85	.86	.82	.72*	.69*
SEVEN	.83	.79	.80	.78	.65*	.69*
EIGHT	.87	.88	.92	.89	.84	.82
NINE	.77	.78	.77*	.78	.65*	.67*
CCIP	.76	.80	.82	.79	.78	.76
ENTER	.78	.77	.81	.76	.76	.72
FREQUENCY	.77	.81	.79	.80	.73*	.73*
STEP	.83	.84	.89	.82	.68*	.68*
THREAT	.80	.80	.78	.78	.75	.77
Percent Correct	100	93.3	86.7	93.3	26.7	26.7
MEAN		.816		.79	.705	.692
STANDARD DEVIATION		.042		.046	.065	.059

*Word missed

TABLE VIII

RECOGNITION SCORES FOR EXPERIMENT 4

Phoneme Length: 1 vector (8 ms)

Distance Rule: M2

WORDS TO BE RECOGNIZED	FILES					
	TRAINING SET			RECOGNITION SET		
	A	B	P	C	3	5
ZERO	.89	.88	.84	.86	.75	.71
ONE	.85	.87	.82	.85	.71*	.72*
TWO	.85	.84	.83	.81	.66*	.71*
THREE	.86	.75*	.80	.74*	.74*	.67*
FOUR	.88	.86	.84	.77	.71*	.61*
FIVE	.81	.89	.88	.85	.69	.61*
SIX	.87	.87	.74*	.84	.70*	.74*
SEVEN	.85	.86	.81	.82	.70*	.68*
EIGHT	.87	.88	.90	.88	.85	.81
NINE	.89	.88	.68*	.89	.63*	.64*
CCIP	.76*	.84	.89	.82	.82	.74
ENTER	.86	.85	.87	.82	.84	.72*
FREQUENCY	.83	.84	.82	.82	.80	.77
STEP	.86	.86	.86	.82	.76	.70
THREAT	.83	.80	.84	.77	.75*	.75*
Percent Correct	93.3	93.3	86.7	93.3	46.7	26.7
MEAN		.843		.824	.74	.705
STANDARD DEVIATION		.044		.041	.065	.056

*Word missed

recognition algorithm expects to see because of training (past statistics). This conclusion is supported by Montgomery's thesis when he discusses accuracy being higher when the acoustic analyzer output is more consistent (Ref 3:5). In Figures 24 thru 26 similar results can be seen for normal speech.

Distance Rule

Two distance rules were analyzed by this research, the M1 distance and the M2 distance. A comparison was made between the M1 distance and the M2 distance using single-vector phoneme templates feature extraction results. These results initially point to the M1 distance performing better than the M2 distance. However, the differences between the two are not as great as the distance seen between five-vector and one-vector phoneme templates. It is hard to distinguish between the M1 and M2 distances.

Figure 25 and Figure 26 are feature extraction files for M1 and M2 distance rules, respectively. The two files, in these figures, have only minor differences. In fact there are only one or two differences between the vectors shown in the top choice. The second, third, fourth and fifth choices have more differences; still no significant difference is found between the M1 and M2 distances when analyzing the feature extraction system.

EIGHT
H50

THE DATE IS-- 9 7 1982
THE TIME IS-- 18 12 10

VECTOR NUMBER	FIRST CHOICE	SECOND CHOICE	THIRD CHOICE	FOURTH CHOICE	FIFTH CHOICE	SCALE FACTOR
******	******	******	******	******	******	******
9	53 100	43 98	21 88	49 88	54 87	.97183881
10	53 100	43 99	21 91	49 91	47 88	.78375250
11	21 100	43 98	53 98	49 94	47 93	.88122890
12	21 100	43 97	53 97	49 93	47 87	.82865400
13	21 100	43 96	53 91	49 89	47 85	.80456350
14	21 100	43 99	53 88	49 87	47 84	.81374680
15	43 100	21 94	53 86	47 84	49 84	.82971850
16	43 100	21 85	47 84	49 84	53 84	.78037830
17	43 100	53 83	47 80	49 80	45 78	.78064980
18	43 100	53 88	45 82	49 81	47 80	.89740180
19	43 100	53 94	49 83	48 81	21 81	.79887450
20	53 100	43 99	21 94	49 89	37 88	.83948410
21	21 100	48 99	43 96	53 98	37 93	.88181570
22	43 100	21 98	48 98	37 94	53 93	.87674940
23	21 100	43 98	48 98	37 93	53 87	.88357551
24	21 100	43 95	48 95	37 89	55 84	.84875890
25	43 100	21 95	37 90	48 90	53 85	.89327700
26	43 100	21 91	48 91	37 90	53 88	.88821750
27	43 100	21 92	48 86	53 88	22 84	.86995100
28	43 100	21 91	53 89	43 89	22 83	.89017480
29	43 100	21 87	53 85	54 81	22 79	.84255110
30	43 100	53 88	21 86	49 89	47 79	.78217550
31	43 100	53 86	21 85	49 84	47 77	.83156270
32	01 100	43 89	53 83	49 79	59 75	.84302750
33	01 100	71 60	22 56	55 55	49 54	.62194230
34	01 100	71 65	07 45	22 41	59 41	.57457540
35	01 100	71 65	07 44	05 38	27 33	.12225580
36	01 100	71 64	07 41	05 35	27 31	.07057476
37	01 100	71 63	07 41	05 35	27 31	.05840125
38	01 100	71 64	07 42	05 35	08 33	.06365435
39	01 100	71 63	07 41	08 34	27 31	.05604802
40	01 100	71 62	07 40	05 34	08 30	.04565115
41	01 100	71 62	07 41	05 35	08 30	.04655514
42	01 100	71 61	07 40	08 34	05 30	.04299267
43	01 100	71 55	08 34	65 32	07 29	.34710840
44	01 100	71 50	36 45	65 49	59 48	.65799350
45	01 100	32 95	14 83	36 81	55 81	.94106024
46	36 100	14 93	23 99	44 95	43 95	.99259260
47	44 100	70 92	63 91	23 89	52 88	.90918840
48	44 100	70 98	63 97	23 90	45 86	.81884860
49	57 100	88 99	44 92	71 91	35 89	.88349850
50	44 100	52 93	68 92	23 87	57 85	.78112891
51	44 100	57 93	88 92	23 88	23 86	.82937850
52	44 100	88 93	71 91	23 95	52 85	.89328011
53	44 100	23 97	63 97	63 97	57 94	.81155131
54	44 100	88 90	57 89	70 95	23 84	.99036810
55	27 100	01 95	39 92	23 91	41 91	1.00000000

Figure 22. "Eight" (5g)
5-vector Phonemes
M1 distance

EIGHT
H508. 8P

THE DATE IS-- 11 17 1982
THE TIME IS-- 13 38 4

VECTOR NUMBER	FIRST CHOICE		SECOND CHOICE		THIRD CHOICE		FOURTH CHOICE		FIFTH CHOICE		SCALE FACTOR	VECTOR ENERGY
10	64	100	53	95	54	88	49	86	42	85	.55704690	52
11	53	100	43	99	54	95	47	95	49	92	.76174490	681
12	43	100	49	97	53	95	47	89	52	88	.68791940	1769
13	63	100	52	98	48	96	43	95	21	94	.69127510	1656
14	49	100	53	99	43	98	63	98	18	97	.71476510	1628
15	52	100	21	98	53	94	1	94	43	93	.89597310	1652
16	52	100	21	96	43	94	63	90	49	84	.76510070	1935
17	43	100	47	86	21	85	54	80	53	79	.71476510	2038
18	43	100	45	90	21	88	47	85	46	81	.73489930	2168
19	43	100	45	95	46	85	54	84	47	84	.65771810	2045
20	43	100	49	96	53	95	45	91	54	88	.66107380	1633
21	43	100	1	94	53	94	45	90	59	90	.81879190	1765
22	37	100	43	99	1	98	32	97	63	97	.81543620	1753
23	1	100	63	96	37	93	48	92	1	91	.68120800	1743
24	1	100	1	97	1	96	1	95	18	95	.64429530	1803
25	1	100	32	99	1	98	37	97	48	93	.81543620	1730
26	32	100	37	99	55	98	48	97	14	94	.84563760	1870
27	22	100	37	95	14	94	21	93	55	93	.73154360	1674
28	22	100	14	92	37	92	32	88	43	88	.72483220	1452
29	32	100	37	97	55	97	48	96	14	94	.83221470	1490
30	14	100	32	98	55	95	37	91	65	90	.74832210	1636
31	21	100	22	94	43	94	55	92	32	90	.86912750	1889
32	43	100	21	98	47	95	45	91	53	91	.76845630	1482
33	43	100	45	96	54	93	46	92	47	91	.69463090	1298
34	53	100	43	99	49	91	54	90	47	86	.62080530	471
35	2	100	32	93	37	93	48	88	14	83	.56040260	40
36	37	100	1	99	48	98	14	94	36	94	.65100670	12
37	1	100	1	96	1	95	1	94	1	93	.92617450	3
38	1	100	1	94	16	88	1	88	1	87	.91946300	4
39	1	100	1	96	1	90	1	88	1	86	.83557050	5
40	11	100	1	99	1	97	69	92	1	92	.82885900	4
41	1	100	1	94	11	93	1	84	1	84	.75167780	3
42	1	100	1	99	1	98	1	96	1	94	.76174490	4
43	1	100	1	92	1	92	1	91	2	91	.77181200	3
44	1	100	1	97	1	92	1	90	1	90	.71476510	3
45	1	100	1	96	1	95	37	90	1	87	.85570470	3
46	1	100	13	96	8	96	1	96	2	93	1.00000000	3
47	48	100	14	95	37	95	65	93	36	92	.37583890	122
48	48	100	14	95	37	94	65	93	55	92	.59395970	421
49	44	100	23	98	68	96	28	94	57	94	.65100670	772
50	57	100	1	98	18	90	58	88	52	87	.64429530	896
51	57	100	28	97	68	96	23	93	1	90	.56711410	820
52	15	100	35	99	28	89	44	88	57	86	.53355700	499
53	57	100	28	99	1	97	52	95	23	94	.80201340	296
54	23	100	44	98	63	87	52	87	9	83	.67785230	207
55	44	100	23	98	70	96	63	95	68	95	.57382550	82
56	68	100	66	99	35	98	1	95	15	92	.63758390	59
57	23	100	70	99	63	96	2	92	16	90	.79194630	66

Figure 23. "Eight" (5g)
1-vector Phonemes
M1 distance

Phoneme Averaging

Phoneme averaging was used extensively in this research project. Phonemes in the five-vector phoneme template were averaged when ever possible. The word "eight was represented by only two phonemes averaged from all the "a" sounds and all the "t" sounds respectively. The averaging used for "eight" was successful and is reflected in the 100% recognition across the board for the word "eight" by all the feature extraction processes in the body of this thesis. In addition, there was only one "n" sound used in this research. In previous research done by Seelandt he used an "n" sound for each word where an "n" sound occured throughout the vocabulary. In this research the "n" sound was averaged for each "n" sound in the vocabulary. The "n" sound performed well and was identified consistently throughout the feature extraction files. The one-vector phoneme templates were all averaged. The usual number of vectors averaged into the single-vector phoneme was five or more vectors. The single-vector phonemes also included twelve average noise templates.

EIGHT
HCF

THE DATE IS-- 9 3 1982
THE TIME IS-- 4 47 52

VECTOR NUMBER ******	FIRST CHOICE ******		SECOND CHOICE ******		THIRD CHOICE ******		FOURTH CHOICE ******		FIFTH CHOICE ******		SCALE FACTOR ******
9	47	100	01	99	54	94	9	93	53	93	.86700860
10	43	100	49	99	53	99	47	95	54	93	.75896120
11	43	100	21	96	53	96	54	95	47	94	.63405990
12	43	100	47	89	49	89	53	87	21	85	.52311840
13	43	100	21	91	47	90	54	86	53	84	.55234870
14	43	100	47	89	53	87	49	81	21	80	.56531110
15	43	100	54	90	47	87	53	86	21	79	.56298690
16	43	100	53	88	54	86	47	84	21	82	.59202590
17	43	100	53	83	54	81	21	76	47	75	.48049210
18	43	100	53	91	47	85	21	84	54	82	.57166930
19	43	100	53	87	21	80	49	78	54	77	.48710050
20	43	100	53	85	21	81	49	78	47	76	.45559020
21	43	100	49	87	53	86	21	79	47	75	.43542150
22	43	100	53	84	21	81	49	80	47	75	.39494030
23	43	100	49	89	53	89	21	84	47	79	.43938010
24	43	100	49	91	21	89	53	89	47	77	.45312820
25	43	100	21	95	53	91	47	89	49	87	.51561440
26	43	100	49	90	21	88	47	88	53	86	.51033290
27	43	100	49	90	21	88	53	83	47	77	.47718480
28	21	100	43	97	53	91	47	89	49	88	.55107870
29	43	100	49	94	21	90	53	90	47	89	.55405620
30	43	100	49	92	53	90	21	85	47	83	.57212520
31	53	100	49	99	43	97	21	94	47	80	.63441600
32	53	100	43	99	49	97	21	94	47	85	.63458700
33	43	100	53	98	49	96	22	89	47	84	.64819470
34	49	100	43	99	53	97	22	89	21	84	.69195870
35	53	100	43	91	49	90	21	78	22	76	.67771400
36	53	100	49	96	43	91	21	89	59	82	.80912200
37	01	100	49	85	53	85	59	80	22	77	.84878290
38	01	100	59	58	71	58	22	50	49	50	.59454550
39	01	100	71	57	07	44	08	35	64	35	.32789110
40	01	100	71	61	07	40	08	34	06	30	.04719339
41	01	100	71	62	07	40	08	34	27	31	.05180730
42	01	100	71	60	07	40	08	34	06	30	.03642284
43	01	100	71	60	07	40	08	34	06	30	.03669963
44	01	100	71	60	07	40	08	34	06	30	.03680750
45	01	100	71	60	07	40	08	34	06	30	.03091545
46	01	100	71	59	07	39	08	33	27	30	.02082468
47	01	100	71	59	07	39	08	33	06	30	.02171409
48	01	100	71	59	07	39	08	33	27	30	.02206822
49	01	100	71	59	07	39	08	33	27	30	.02153295
50	01	100	71	54	08	34	07	29	65	25	.34043840
51	01	100	71	45	08	40	20	35	32	33	.66956480
52	01	100	32	88	35	77	14	74	36	74	.98907680
53	44	100	52	91	70	91	63	90	23	88	.93931770
54	44	100	70	75	68	69	23	64	57	63	.38737940
55	44	100	63	91	70	86	35	80	23	78	.70782140
56	44	100	68	92	70	88	57	84	63	84	.74717910
57	69	100	44	98	23	91	57	86	70	86	.73809780
58	44	100	68	98	57	91	23	89	70	89	.77025070
59	44	100	68	76	70	67	46	62	23	60	.38947300
60	26	100	44	93	68	93	57	92	28	85	.87033210
61	68	100	44	97	35	94	26	89	70	88	.87007770
62	44	100	57	95	68	92	23	86	39	82	.85741560
63	68	100	23	97	39	95	44	94	29	90	.90153660
64	23	100	67	99	29	99	68	95	30	93	.91860820
65	68	100	44	99	26	97	32	95	67	95	.85876780
66	68	100	44	97	26	95	39	92	40	90	.87915090
67	26	100	44	96	50	93	57	93	39	91	.93255750
68	27	100	26	95	57	93	70	92	01	91	1.00000000

Figure 24. "Eight" no G-stress
5-vector Phonemes
M1 distance

EIGHT
HCP8.8P

THE DATE IS-- 11 17 1982
THE TIME IS-- 10 58 48

VECTOR NUMBER	FIRST CHOICE		SECOND CHOICE		THIRD CHOICE		FOURTH CHOICE		FIFTH CHOICE		SCALE FACTOR	VECTOR ENERGY
******	*******		*******		*******		*******		*******		*******	*******
11	64	100	42	98	54	92	53	86	49	82	.60869560	120
12	47	100	54	99	49	98	53	95	46	89	.57312250	494
13	47	100	49	88	53	86	43	86	21	86	.39130430	692
14	21	100	47	99	43	93	22	84	49	83	.57312250	921
15	43	100	47	98	49	96	21	90	53	87	.55335960	920
16	54	100	43	98	49	93	53	90	47	87	.66798420	849
17	43	100	54	89	49	83	47	79	53	76	.43873510	839
18	43	100	53	88	59	86	47	86	21	85	.64822130	832
19	43	100	54	88	49	85	47	85	53	80	.44268770	967
20	43	100	21	97	47	80	49	78	53	77	.52173910	949
21	43	100	53	90	54	84	49	83	47	78	.40711460	1130
22	43	100	21	84	53	83	49	82	47	76	.43478260	1028
23	43	100	49	91	53	88	47	86	54	80	.49802370	1209
24	43	100	49	91	53	84	21	78	47	77	.32015810	1370
25	43	100	49	95	53	95	21	90	47	89	.45849800	1299
26	43	100	49	98	47	91	53	90	21	87	.38339920	1614
27	43	100	21	96	49	86	53	82	47	79	.54150190	1545
28	21	100	43	99	53	93	47	91	49	91	.52569170	1325
29	43	100	49	91	21	90	47	88	53	84	.33596840	1877
30	43	100	21	95	49	89	47	83	53	82	.43873510	1666
31	49	100	43	98	21	97	59	97	53	96	.68379440	1256
32	21	100	43	93	47	88	49	84	53	83	.35177860	1216
33	43	100	53	96	49	95	21	90	47	89	.57312250	1032
34	49	100	21	97	43	94	53	93	22	76	.65217390	336
35	53	100	49	96	22	95	47	95	43	90	.81422920	253
36	53	100	47	99	49	95	21	95	43	95	.66403160	277
37	53	100	47	97	49	95	21	94	43	92	.70355730	214
38	.49	100	43	98	53	96	21	90	47	85	.53754940	128
39	17	100	43	86	49	86	21	86	9	85	.69960470	57
40	1	100	1	95	1	92	1	90	1	88	.81422920	3
41	16	100	36	99	65	99	43	97	17	96	.90118580	16
42	1	100	52	95	1	92	1	91	63	90	.58498020	2
43	1	100	1	97	1	96	1	93	1	92	.67588930	2
44	1	100	1	96	1	92	1	89	1	88	.74308300	4
45	13	100	12	96	42	93	64	84	61	72	.59683790	8
46	1	100	1	91	1	89	1	86	1	84	.49802370	2
47	1	100	1	92	1	89	1	86	1	83	.51778650	2
48	1	100	1	99	1	98	1	98	1	96	.69960470	2
49	1	100	1	92	1	92	1	90	1	87	.67588930	1
50	1	100	1	97	1	97	1	96	1	95	.60869560	2
51	1	100	1	94	1	92	1	90	1	85	.46640310	2
52	1	100	1	93	1	91	1	91	1	85	.54545450	2
53	1	100	1	100	1	87	1	87	1	86	.72332010	3
54	52	100	44	98	18	95	63	93	47	90	.66007900	123
55	44	100	57	98	35	96	28	96	68	92	.67193680	213
56	44	100	70	98	18	94	37	92	48	87	.68774700	185
57	44	100	23	92	63	88	70	85	28	82	.58498020	265
58	23	100	68	98	44	94	60	93	57	89	.60869560	185
59	35	100	1	96	44	95	63	92	46	89	.62450590	102
60	68	100	44	99	23	91	70	90	28	88	.74308300	152
61	44	100	35	86	23	86	15	83	68	83	.59683790	94
62	57	100	44	95	52	94	23	94	63	92	.70355730	89

Figure 25. "Eight" no G-stress
1-vector Phonemes
M1 distance

EIGHT
HCPB. 8P

THE DATE IS-- 11 13 1982
THE TIME IS-- 11 43 53

VECTOR NUMBER	FIRST CHOICE		SECOND CHOICE		THIRD CHOICE		FOURTH CHOICE		FIFTH CHOICE		SCALE FACTOR	VECTOR ENERGY
******	*******		*******		*******		*******		*******		*******	*******
11	42	100	64	97	54	91	53	88	1	79	.55649210	120
12	54	100	47	96	49	95	53	92	46	88	.51922510	494
13	47	100	43	83	54	82	21	82	49	82	.37341020	692
14	47	100	21	99	52	94	43	83	22	80	.60440700	921
15	47	100	43	95	49	93	21	87	54	85	.54466130	920
16	54	100	49	93	43	87	53	86	47	82	.61608990	849
17	43	100	54	91	47	86	49	86	53	77	.44513450	839
18	43	100	53	97	59	95	47	95	54	91	.76826380	832
19	43	100	54	88	49	86	47	84	53	80	.39692400	967
20	43	100	21	90	53	82	47	79	49	79	.53593610	949
21	43	100	53	95	49	89	54	87	47	82	.39884640	1130
22	43	100	53	85	49	84	21	83	47	78	.46790890	1028
23	43	100	53	96	49	96	54	91	47	89	.57453410	1209
24	43	100	49	93	53	86	47	77	54	75	.33850930	1370
25	43	100	49	92	53	86	47	83	21	82	.42383910	1299
26	43	100	49	96	47	90	53	87	21	82	.35906540	1614
27	43	100	21	86	49	86	53	80	47	74	.52750660	1545
28	21	100	43	94	53	91	47	90	49	85	.54037270	1325
29	43	100	49	85	21	83	47	82	53	80	.29503100	1877
30	43	100	21	89	49	86	47	80	53	79	.43019810	1666
31	43	100	49	99	53	97	47	90	59	87	.68870150	1256
32	21	100	43	96	47	88	53	81	49	81	.36291030	1216
33	43	100	53	90	49	88	47	85	21	81	.51390120	1032
34	49	100	43	96	53	90	21	89	22	79	.65439220	336
35	53	100	22	97	43	95	47	93	54	91	.85773440	253
36	53	100	47	97	43	96	49	95	54	94	.66903280	277
37	47	100	53	98	43	96	54	92	21	92	.73365870	214
38	43	100	49	99	53	98	21	88	47	81	.55323860	128
39	17	100	9	90	21	81	49	77	43	76	.63354030	57
40	1	100	1	94	1	90	1	89	1	88	.75288370	3
41	16	100	17	96	63	95	44	93	2	92	.76708070	16
42	1	100	52	92	1	91	1	91	1	90	.51981660	2
43	1	100	1	99	1	97	1	95	1	91	.60189290	2
44	1	100	1	96	1	92	1	91	1	90	.63102630	4
45	13	100	12	96	42	90	64	78	61	73	.55737940	8
46	1	100	1	90	1	89	1	86	1	83	.42102930	2
47	1	100	1	92	1	87	1	86	1	81	.45888790	2
48	1	100	1	99	1	97	1	96	1	95	.58207630	2
49	1	100	1	91	1	88	1	88	1	87	.58636490	1
50	1	100	1	97	1	97	1	96	63	95	.57202010	2
51	1	100	1	96	1	94	1	93	1	90	.45800050	2
52	1	100	1	92	1	90	1	85	1	84	.47811290	2
53	1	100	1	99	1	91	1	89	1	85	.61342790	3
54	52	100	44	97	18	89	47	87	21	86	.58429460	123
55	35	100	44	99	28	97	57	96	23	91	.58932260	213
56	44	100	18	84	14	82	63	82	70	81	.58680860	185
57	44	100	23	87	63	84	70	80	68	80	.49408450	265
58	23	100	68	98	60	97	44	96	57	86	.53578820	185
59	35	100	1	97	44	94	68	89	23	85	.54673170	102
60	44	100	68	94	70	91	23	88	28	88	.64566700	152
61	44	100	35	85	23	83	28	82	68	81	.48521140	94
62	23	100	44	96	57	94	68	91	52	88	.62688550	89

Figure 26. "Eight" no G-stress
1-vector Phonemes
M2 distance rule

VI. Conclusions and Recommendations

Conclusions

There are three main conclusions to be drawn from this research which involved five-vector and one-vector phoneme templates, distance rules and the averaging of phoneme templates. The five-vector phoneme template did significantly better in the recognition results for G-speech and was more consistent in the feature extraction process than the one-vector phoneme template. Single-vector phonemes do have a computational advantage over the five-vector templates but this advantage does not overcome the disadvantage of degraded recognition, discussed above.

The M1 and M2 distance rules studied showed little differences in feature extraction output. Even though results showed M1 distance to perform slightly better on normal speech and 40% better on one set of G-speech files conditions, adjusting the fuzzy variables and changing the phoneme representations (experiment 4) led to better results for the M2 distance. In addition the recognition scores for M1 and M2 distances showed little differences. Thus it seems that the M1 distance rule, which can have a 50% computational advantage in number of actual operations, can be used with results equal to or better than the M2 rule.

Phoneme averaging resulted in reducing the number of phonemes needed per word. This is the first research project based on Seelandt's techniques to use averaged phoneme templates. When averaged phonemes were used for the word "eight" only half the number of phonemes, compared to what Seelandt used, were needed. In addition the feature extraction based on the average phonemes for the word "eight" produced output more consistent than the multiple unaveraged phonemes.

Recommendations

The first recommendation to be made would cover data acquisition. This thesis used G-speech and normal speech to analyze the feature extraction and recognition algorithm used at the AFIT Signal Processing Laboratory. However, the G-speech obtained was not in sufficient quantities to establish meaningful baseline results for G-speech. There is a need for more G-speech or using G-speech already processed. In addition, actual aircraft speech should be obtained if possible in future projects since the noise level is significantly higher compared to speech obtained in the centrifuge.

Another study may want to investigate the use of G-speech templates. Different templates could be used that correspond to different G-levels. The G-speech templates could be implemented in a real aircraft by using the output of the G-meter to select the corresponding G-template. The only drawback is that different sets of templates would have to be made and stored.

It is also recommended that extensive use of the array processor be made for algorithms processing speech and the recognition results in the future. Efficient use of the array processor could lead to shorter turnarounds for results. In this study the recognition of 45 files could take up to 12 hours to run on the Data General Eclipse (using the recognition program LEARN). This does not include the run time for feature extraction on the same 45 files.

Software developed in this research and in the research done by Martin (Ref 2) makes the energy available to the recognition routine. However, the recognition routine did not use the energy in this research. Future researchers may find energy to be useful in the recognition of stops found in words and for thresholding.

This research concluded that five-vector phoneme template feature extraction system outperformed the single-vector feature extraction for G-speech. Thus it points to the need to study variable length phoneme templates to find the optimal length for feature extraction. Also, many of the differences found, even in the same person's speech, between the different phoneme sounds found in speech utterances can be attributed to minor frequency shifts which result in a degraded feature extraction performance. The need for a dynamic frequency sliding algorithm which would attempt to slide the phoneme template up and down the frequency components, within a certain tolerance, to find the best match may be effective in improving the feature extraction system.

Bibliography

1. Seelandt, Karl G. Computer Analysis and Recognition of Phoneme Sounds in Connected Speech. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1981. (AFIT/GE/EE/81D-53)

2. Martin, Daniel L. Parameter Analysis of the Speech Sound Analysis Machine. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology. (AFIT/GE/EE/82D-46)

3. Montgomery, Gerard J. Isolated Word Recognition Using Fuzzy Set Theory. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology. (AFIT/GE/EE/82D-74)

4. Beasley, George C., and Dan Fredal. An Analog Speech I/O Channel for the Nova 2 Computer Based on the S-100 Buss. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, March 1981. (AFIT/GE/EE/81M-2)

5. Finkes, Paul F. Jr. G-Stress Effect On Speech Recognition. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1981. (AFIT/GE/EE/81D-19)

6. Hunter, J. C. Time Axis Analysis of Gravity Distorted Speech, MS Thesis. Wright-Patterson Air Force Base, Ohio: School of Engineering, Air Force Institute of Technology, December 1981. (AFIT/GE/EE/81D-27)

7. Allen, Gordon R. Expansion of the NOVA/ECLIPSE Digital Signal Processing System. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology. (AFIT/GE/EE/82D-16)

8. Fletcher, James E. Performance Improvement of the Phoneme Recognition Algorithm. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology. (AFIT/GE/EE/82D-30)

Appendix A

Speech Files for Thesis

Speech files were created using AUDIOHIST on the NOVA. Digitized files are stored on magnetic tape (MTØ). Tape #1 contains the files listed in this appendix.

File Name Legend

HCAØ.SP
- H - Speaker's name
- C - control or static conditions
- A - # of utterance (A-E=1-5; P=prerun, S=postrun)
- Ø - word spoken ("zero")
- SP - speech file

H3ØS.SP
- H - speaker's name
- 3 - g level in z direction
- Ø - g level in y direction (A,C = +1.5g; B = -1.5g; G = Øg)
- S - word spoken (S-"step")
- SP - speech file

Captain Henwood 27 Apr 82 142Ø hrs

Digitized speech stored on MTØ:2. Conditions: Centrifuge test with F-16 seat, 3Ø degree bank angle with lateral shoulder pads, pitch axis tracking task.

FILE	MAX v	EDIT BLOCKS	COMMENTS
HCAØ.SP	2.89	3Ø	"ZERO"
HCBØ.SP	3.59	3Ø	
HCCØ.SP	4.Ø1	3Ø	
HCDØ.SP	4.33	3Ø	
HCA1.SP	4.6Ø	3Ø	"ONE"
HCB1.SP	4.39	3Ø	.39v noise (breathing)
HCC1.SP	4.54	3Ø	
HCD1.SP	4.Ø7	3Ø	
HCE1.SP	4.49	3Ø	.81v noise (breathing)
HCA2.SP	4.47	3Ø	
HCB2.SP	4.31	3Ø	
HCC2.SP	4.6Ø	3Ø	
HCD2.SP	4.17	3Ø	.Ø4v noise (typical)
HCE2.SP	4.1Ø	3Ø	
HCA3.SP	4.77	3Ø	
HCB3.SP	4.27	3Ø	
HCC3.SP	4.48	3Ø	.26v max noise
HCD3.SP	4.16	3Ø	.81v breathing noise
HCE3.SP	3.85	3Ø	
HCA4.SP	4.19	3Ø	
HCB4.SP	3.98	3Ø	noise

HCC4.SP	3.86	30	
HCD4.SP	3.95	30	
HCE4.SP	3.88	30	.25v noise
HCA5.SP	4.07	30	
HCB5.SP	3.57	30	
HCC5.SP	3.89	30	
HCD5.SP	3.76	30	
HCE5.SP	3.83	30	
HCA6.SP	3.79	30	
HCB6.SP	4.01	30	
HCC6.SP	3.99	30	
HCD6.SP	3.93	30	
HCE6.SP	4.02	30	
HCA7.SP	4.05	30	
HCB7.SP	3.61	30	
HCC7.SP	3.79	30	
HCD7.SP	4.06	30	
HCE7.SP	3.80	30	
HCA8.SP	3.76	30	
HCB8.SP	4.28	30	
HCC8.SP	3.91	30	
HCD8.SP	4.08	30	
HCE8.SP	3.86	30	
HCA9.SP	3.72	30	
HCB9.SP	3.50	30	
HCC9.SP	3.99	30	
HCD9.SP	3.93	30	
HCE9.SP	3.72	30	
HCAF.SP	3.95	30	
HCBF.SP	3.98	30	
HCCF.SP	3.87	30	
HCDF.SP	3.96	30	
HCEF.SP	4.09	30	
HCAE.SP	3.99	30	
HCBE.SP	3.87	30	
HCCE.SP	3.82	30	
HCDE.SP	3.97	30	
HCEE.SP	4.08	30	
HCAC.SP	3.75	40	
HCBC.SP	3.94	40	
HCCC.SP	3.55	40	
HCDC.SP	4.04	40	
HCEC.SP	3.95	40	
HCAT.SP	4.08	30	
HCBT.SP	3.72	30	
HCCT.SP	3.75	30	
HCDT.SP	4.16	30	
HCET.SP	4.11	30	
HCAS.SP	4.07	30	
HCBS.SP	4.07	30	
HCCS.SP	4.08	30	
HCDS.SP	4.08	30	
HCES.SP	3.59	30	

Capt Henwood, Pre-run Static List of Words

FILE	MAX v	EDIT BLOCKS	COMMENTS
HCPC.SP	3.85	40	"CCIP"
HCPE.SP	3.78	40	"ENTER"
HCPF.SP	3.86	40	"FREQUENCY"
HCPS.SP	3.94	40	"STEP"
HCPT.SP	3.96	40	"THREAT"
HCP0.SP	3.84	40	
HCP1.SP	4.00	40	
HCP2.SP	4.22	40	
HCP3.SP	4.13	40	
HCP4.SP	4.24	40	
HCP5.SP	4.06	40	
HCP6.SP	4.19	40	
HCP7.SP	4.28	40	
HCP8.SP	3.93	40	
HCP9.SP	4.08	40	

Captain Henwood G-Speech

EDIT FILE	MAX v	BLOCKS	COMMENTS
H300.SP	4.29	40	3GZ, 0GY
H301.SP	4.97	40	
H302.SP	4.55	40	
H303.SP	4.16	40	
H304.SP	4.68	40	
H305.SP	4.36	40	
H306.SP	4.19	40	
H307.SP	4.25	40	
H308.SP	4.60	40	
H309.SP	3.96	40	
H30C.SP	4.26	40	
H30E.SP	4.08	40	
H30F.SP	4.17	40	
H30S.SP	3.84	40	
H30T.SP	4.47	40	
H500.SP	4.39	40	5GZ, 0GY
H501.SP	4.71	30	
H502.SP	5(1)	30	
H503.SP	4.28	25	
H504.SP	4.81	40	
H505.SP	4.38	40	
H506.SP	4.41	40	
H507.SP	4.38	30	
H508.SP	4.93	40	
H509.SP	4.01	40	
H50C.SP	4.32	40	
H50E.SP	3.86	30	
H50F.SP	4.81	25	

EDIT FILE	MAX v	BLOCKS	COMMENTS
H50S.SP	4.74	40	
H50T.SP	4.65	40	

Capt C. St Sauver

FILE	MAX v	EDIT BLOCK	COMMENTS
SCA0.SP	3.17	30	
SCB0.SP	3.38	30	
SCC0.SP	3.36	30	
SCD0.SP	4.00	30	
SCE0.SP	3.76	30	
SCA1.SP	4.44	30	
SCB1.SP	4.22	30	
SCC1.SP	3.90	30	
SCD1.SP	4.43	30	
SCE1.SP	4.18	30	
SCA2.SP	4.26	30	
SCB2.SP	4.26	30	
SCC2.SP	3.67	30	
SCD2.SP	3.67	30	
SCE2.SP	3.99	30	
SCA3.SP	4.00	30	
SCB3.SP	4.54	30	
SCC3.SP	4.11	30	
SCD3.SP	4.07	30	
SCE3.SP	4.08	30	
SCA4.SP	4.07	30	
SCB4.SP	3.74	30	
SCC4.SP	3.83	30	.14v
SCD4.SP	3.93	30	
SCE4.SP	4.00	30	.12v
SCA5.SP	4.02	30	
SCB5.SP	4.53	30	
SCC5.SP	4.40	30	
SCD5.SP	4.71	30	
SCE5.SP	4.47	30	
SCA6.SP	4.88	30	.47v pre-noise
SCB6.SP	3.67	30	
SCC6.SP	4.77	30	
SCD6.SP	4.93	30	
SCE6.SP	5.00(2)	30	
SCA7.SP	5.00(4)	30	.18v noise (MAX)
SCB7.SP	4.31	30	
SCC7.SP	4.61	30	
SCD7.SP	4.45	30	
SCE7.SP	4.70	30	
SCA8.SP	3.98	30	
SCB8.SP	4.13	30	

EDIT FILE	MAX v	BLOCKS	COMMENTS
SCC8.SP	4.11	30	
SCD8.SP	4.03	30	
SCE8.SP	3.83	30	
SCA9.SP	4.16	30	
SCB9.SP	4.35	30	
SCC9.SP	3.95	30	
SCD9.SP	4.08	30	
SCE9.SP	3.93	30	
SCAC.SP	4.23	40	
SCBC.SP	4.20	40	
SCCC.SP	4.11	40	
SCDC.SP	4.19	40	
SCEC.SP	4.21	40	
SCAE.SP	4.47	30	
SCBE.SP	4.49	30	
SCCE.SP	4.44	30	
SCDE.SP	4.31	30	
SCEE.SP	4.32	30	
SCAF.SP	3.89	30	
SCBF.SP	3.85	30	
SCCF.SP	3.78	30	
SCDF.SP	3.69	30	
SCEF.SP	3.65	30	
SCAS.SP	4.71	30	
SCBS.SP	4.58	30	
SCCS.SP	4.44	30	
SCDS.SP	4.48	30	
SCES.SP	4.54	30	
SCAT.SP	4.65	30	
SCBT.SP	4.43	30	
SCCT.SP	4.57	30	
SCDT.SP	4.50	30	
SCET.SP	3.76	30	
S30E.SP	4.46	30	
S305.SP	4.72	30	
S30T.SP	4.54	30	
S309.SP	4.62	30	
S304.SP	4.81	30	
S301.SP	4.41	30	
S300.SP	4.50	30	
S30F.SP	4.56	30	
S306.SP	4.70	30	
S30S.SP	4.64	30	
S302.SP	4.80	30	
S307.SP	4.74	30	
S502.SP	4.99	30	
S503.SP	4.56	30	
S50S.SP	4.64	30	
S504.SP	4.88	30	
S508.SP	4.62	30	
S50C.SP	4.50	40	

EDIT FILE	MAX v	BLOCKS	COMMENTS
S501.SP	4.81	30	
S50T.SP	4.68	30	
S505.SP	4.22	30	
S509.SP	3.99	30	
S50E.SP	4.54	30	
S50F.SP	4.41	30	
S507.SP	4.70	30	
S500.SP	4.55	30	
S506.SP	4.41	30	
S3B3.SP	4.29	30	
S3B8.SP	4.24	30	
S3B6.SP	4.43	30	
S3B4.SP	4.24	30	
S3B0.SP	4.43	30	
S3B5.SP	4.21	30	
S3B2.SP	4.31	30	
S3B7.SP	4.41	30	
S3BC.SP	3.90	40	
S3BF.SP	4.45	40	
S3BT.SP	4.20	30	
S3BE.SP	4.21	30	
S3B9.SP	4.17	30	
S3B1.SP	4.39	30	
S3BS.SP	4.51	30	
S5A1.SP	4.17	40	Rename S5A-.- S5C-.-
S5A2.SP	4.32	30	
S5AS.SP	4.28	30	
S5AG.SP	4.24	30	
S5AT.SP	4.13	30	
S5A4.SP	4.25	30	
S5AE.SP	4.40	30	
S5C6.SP	4.59	30	
S5AC.SP	4.10	40	
S5CT.SP	4.22	30	
S5A0.SP	4.45	30	
S5CE.SP	4.52	30	
S5CS.SP	4.10	30	
S5A8.SP	3.97	30	
S5C2.SP	4.30	30	
S5B4.SP	4.24	30	
S5B1.SP	4.05	30	
S5A7.SP	4.54	30	
S5A3.SP	4.07	30	
S5A5.SP	3.88	30	
S5AF.SP	3.92	30	
S5A9.SP	4.01	30	
S3AC.SP	3.84	40	
S3AF.SP	3.74	30	
S3A3.SP	3.56	30	
S3AS.SP	4.16	30	
S3A1.SP	3.84	30	

EDIT FILE	MAX v	BLOCKS	COMMENTS
S3AT.SP	3.97	30	
S3A2.SP	4.40	30	
S3A5.SP	4.04	30	
S3A9.SP	4.11	30	
S3A4.SP	4.07	30	
S3A7.SP	4.49	30	
S3A6.SP	4.50	30	
S3AE.SP	4.33	30	
S3A8.SP	4.01	30	
S3A0.SP	4.57	30	
S5B2.SP	4.31	30	
S5B9.SP	4.32	30	
S5BC.SP	4.22	40	
S5B7.SP	4.23	30	
S5BF.SP	4.38	30	
S5B4.SP	4.20	30	
S5BE.SP	4.29	30	
S5B8.SP	3.84	30	
S5B5.SP	4.24	30	noise up to .97
S5BT.SP	4.23	30	
S5B0.SP	4.24	30	
S5B3.SP	3.92	30	
S5BS.SP	4.55	30	
S5B1.SP	3.99	30	
S5B6.SP	4.39	30	.72v noise
S3G1.SP	4.16	30	
S3G6.SP	4.43	30	
S3G8.SP	4.27	30	

EDIT FILE	MAX v	BLOCKS
S3G7.SP	4.05	30
S3GS.SP	4.02	30
S3GF.SP	4.20	30
S3GC.SP	3.91	40
S3GT.SP	4.28	30
S3G0.SP	4.14	30
S3G3.SP	3.95	30
S3G2.SP	3.80	30
S3G9.SP	4.04	30
S3G5.SP	4.09	30
S3G4.SP	4.07	30
S3GE.SP	4.41	30
SCSC.SP	3.83	40
SCS5.SP	4.06	30

EDIT FILE	MAX v	BLOCKS
SCS5.SP	4.28	30
SCS6.SP	4.62	30
SCS4.SP	4.11	30
SCS8.SP	3.27	30
SCS3.SP	3.68	30
SCS0.SP	4.17	30
SCS7.SP	4.52	30
SCS1.SP	4.08	30
SCST.SP	4.31	30
SCS9.SP	3.82	30
SCSE.SP	3.62	30
SCSF.SP	3.25	30
SCS2.SP	3.20	30

APPENDIX B

PROGRAM SPENPLOT

Program SPENPLOT creates spectrograms of speech and was developed from programs by Seelandt (Ref 1:198) and Finkes (Ref 5:167). Before SPENPLOT is used, speech files must be processed by program DRVR (Ref 2). Program DRVR outputs frequency component files from digital speech inputs. The files from DRVR are entered into program SPENPLOT, and SPENPLOT makes a spectrogram as in Figure B-2. The steps to create a spectrogram are listed in Figure B-1.

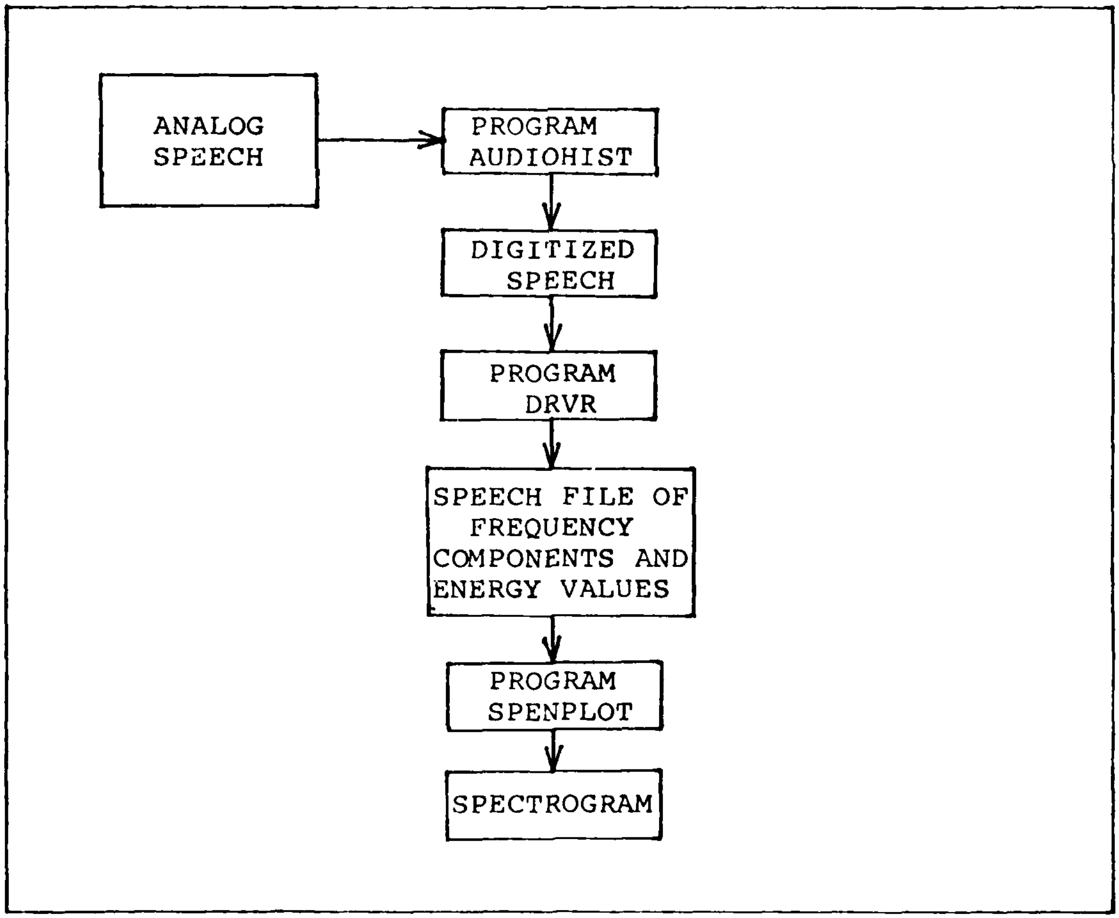


Figure B-1. Steps for Spectrogram Generation

Martin's program does a DFT of the digitized speech and the size of the DFT can be specified in the program. However, program SPENPLOT will only print up to a 256 point DFT (128 frequency components) because of the 132 character limit on the Printronix model P300 printer. SPENPLOT will send the necessary symbols to the Printronix model P300 printer to create a spectrogram as seen in Figure B-2 thru B-11. SPENPLOT accepts input files which consist of a header (block 0 with 256 integers) followed by data blocks which contain 128 real numbers per each block of data. A 64 point DFT has four vectors (8 milliseconds per vector) in each data block. SPENPLOT checks the header with values listed in Table B-I.

The values listed in Table B-I help prepare the spectrograms. The header of the spectrogram is filled out by reading the header (block 0) of the data input files. In program DRVR the dc component of the spectrum was replaced by the energy per vector before normalization. SPENPLOT can accept information that has been normalized or not normalized by DRVR. The program SPENPLOT listed in this appendix will only give a scale for 64 point and 128 point DFTs. However, the spectrum will be created for lower DFT sizes and up to 256 point DFTs. DFT sizes greater than 256 will cause erroneous output from SPENPLOT.

The source code that follows will allow regular interactive use when compiled using the FORTRAN/X statement. This program will be loaded with the relocatable binary for

SPENPLOT and subroutines BYTEOUT, IOFT5 and the FORTRAN library. Program SPENPLOT was developed from spectrogram programs found in Seelandt's thesis (Ref A). See the Printronix manual for how to use the plot mode as used in SPENPLOT for the spectrogram plot.

Table B-I

Header Values used for Program DRVR and SPENPLOT

ELEMENT	CONTENTS
1-13	Observation file name(channel 4)
14-26	Speech file name(channel 5)
27	Switch: 1=preemphasize Ø=don't preemphasize
28	Preemphasis slope
29	Preemphasis corner frequency
3Ø	Number of time points per FFT
31	Switch: 1=Hamming window Ø=rectangular window
32	**Normalization: 1 = normalize to unity 2 = no normalization Ø = divide by vector energy
33	Switch: 1=create test file Ø=don't create
34-53	not used
54	*Vector length of phonemes
55	Number of first time slice in file
56	Number of last time slice in file
57	Number of points per time slice in file
58	Switch: 1=overlapping Ø=non-overlapping
59	Number of disk blocks in observation file
6Ø	Switch: 1=deemphasis Ø=no deemphasis
61	Deemphasis slope
62	Deemphasis corner frequency
*63	*Switch: 1=phoneme file Ø=not phoneme file
64-256	*Used to store times phoneme has been modified (Can only store 193 modification numbers.)

* Added for use by program MKPHON, which makes phoneme templates.

** Entry 32 is used by program SPENPLOT, so proper spectrogram is plotted.

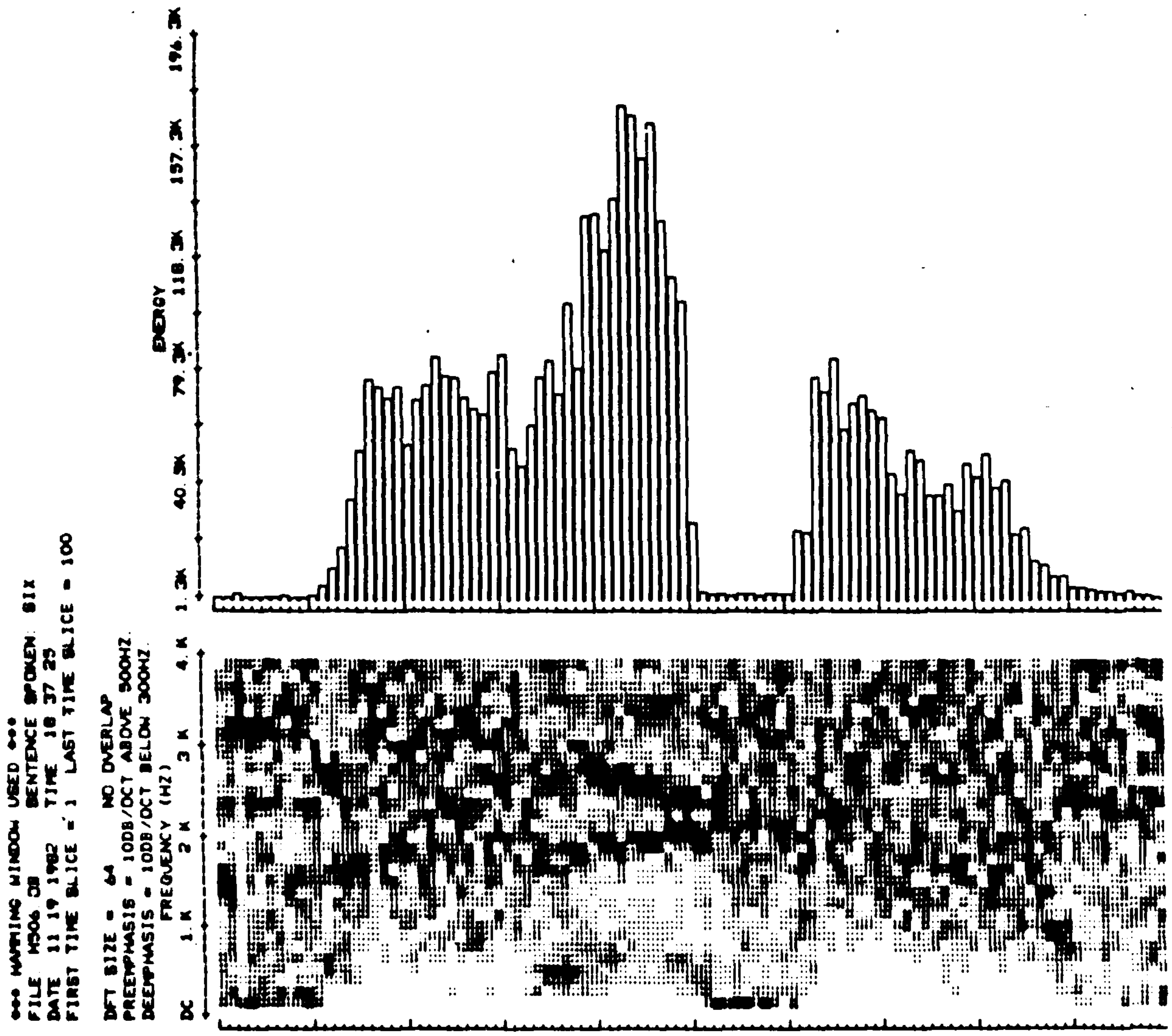


Figure B-2. Spectrogram of IX

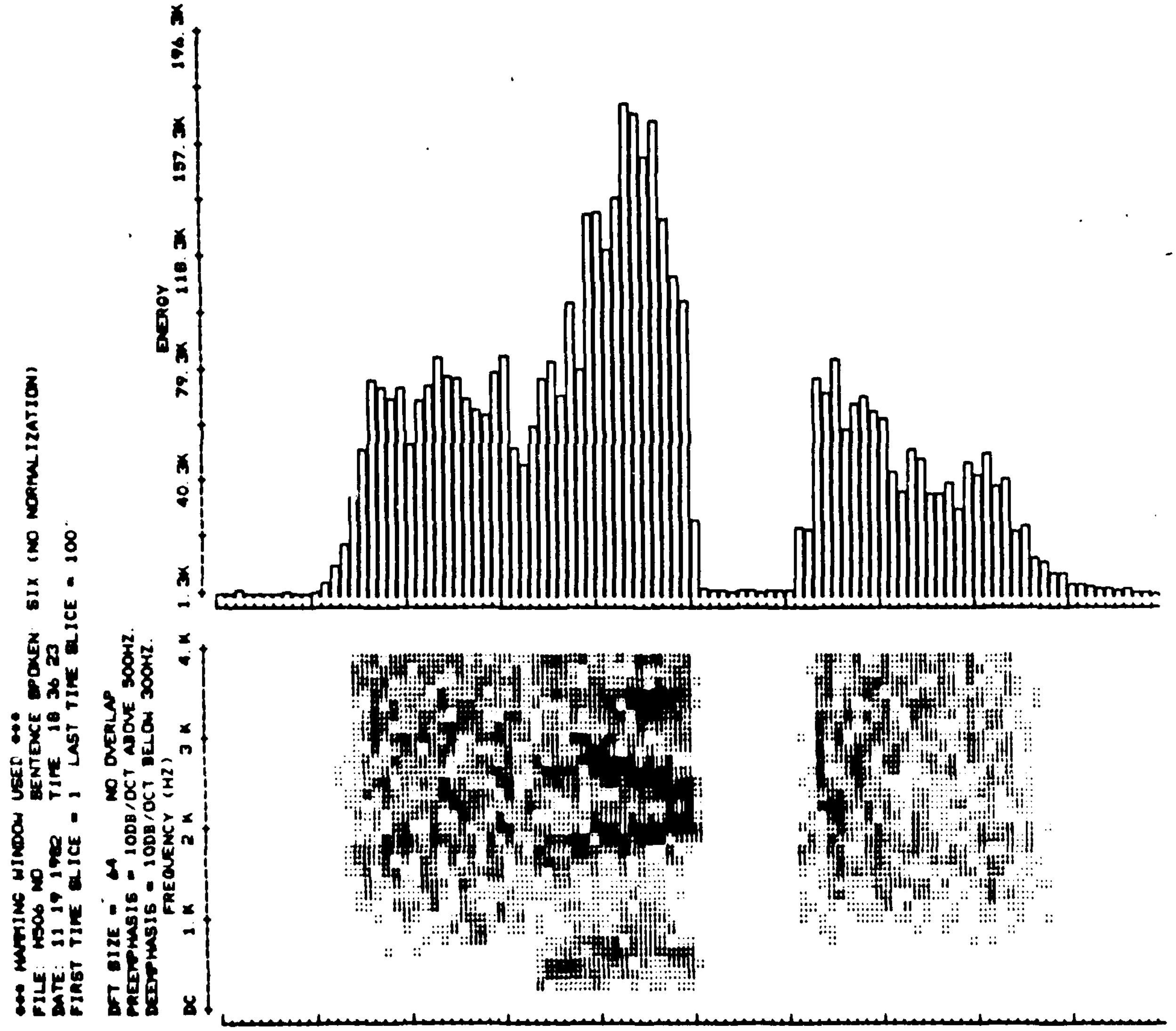


Figure B-3. Spectrogram of SIX
No Normalization

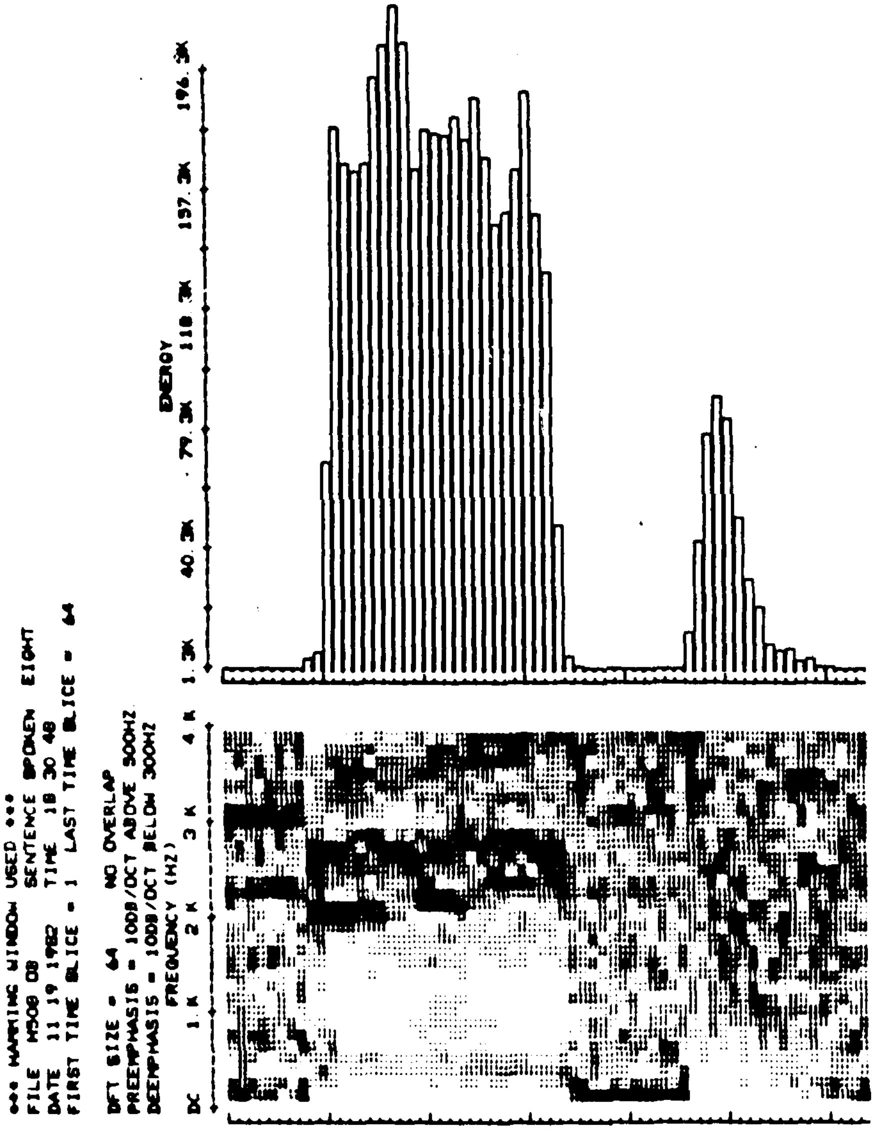


Figure B-4. Spectrogram of EIGHT at 5Gs

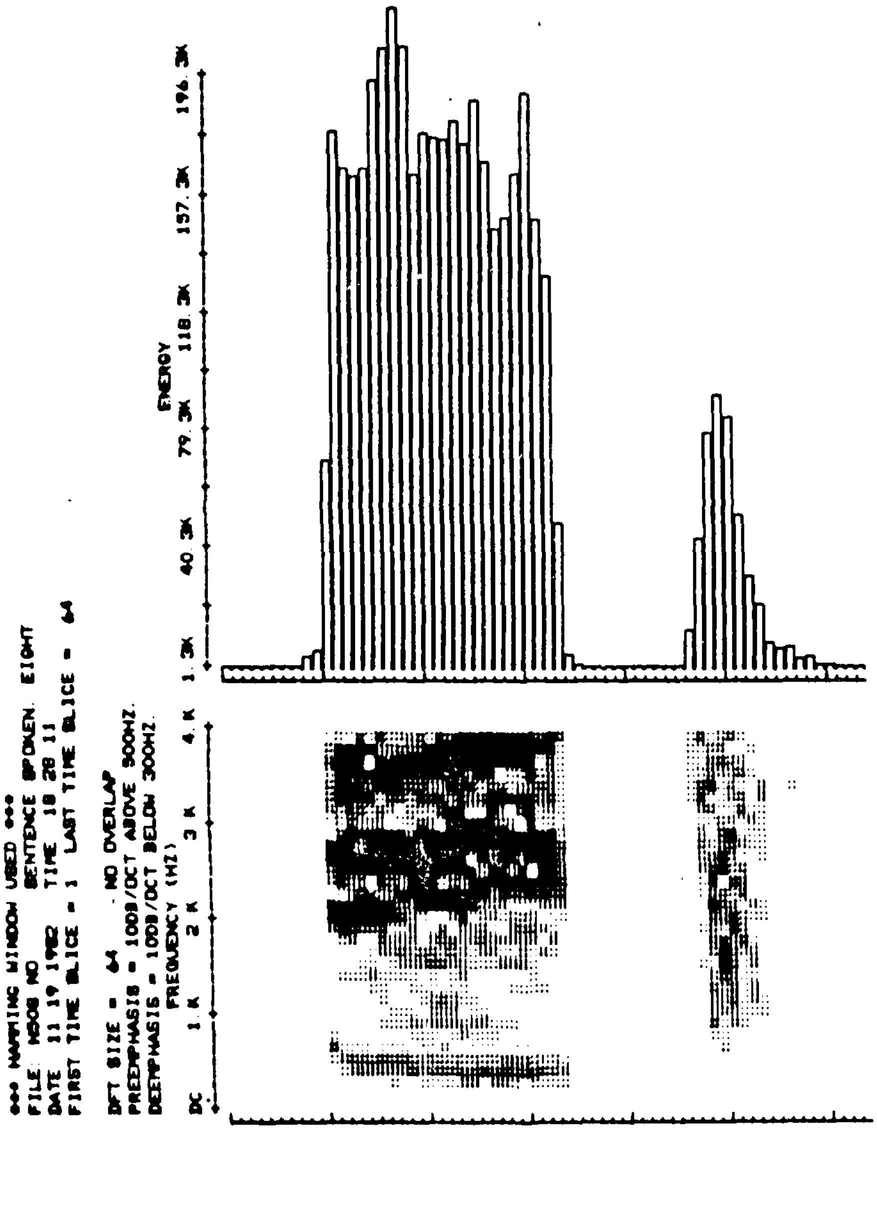


Figure B-5. Spectrogram of EIGHT at 5Gs
No Normalization

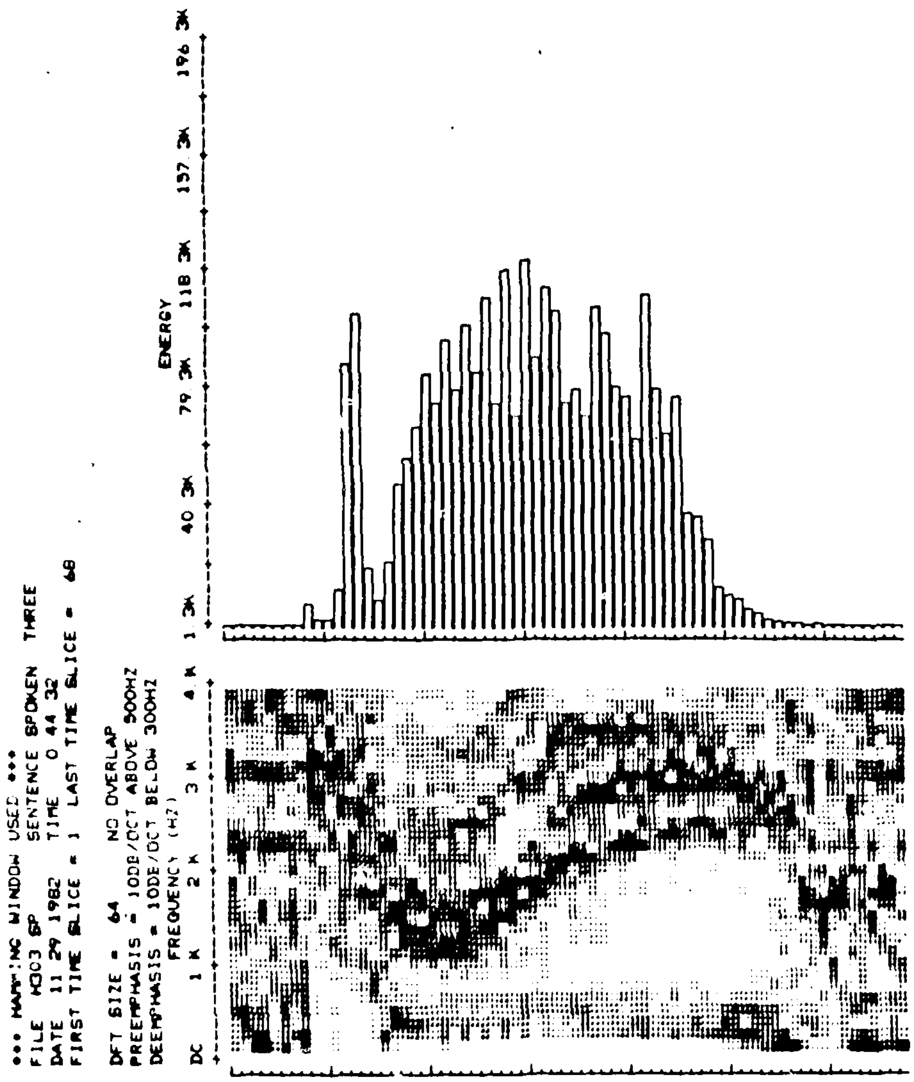


Figure B-6. Spectrogram of THREE at 3Gs

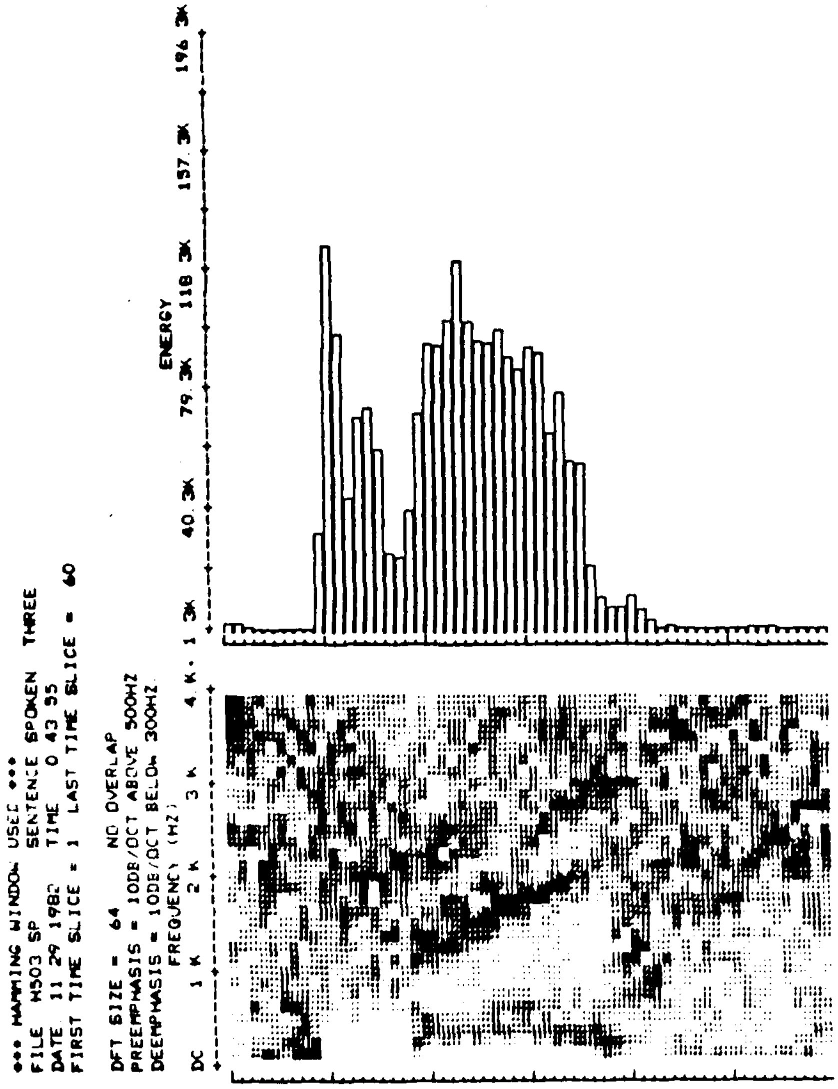


Figure B-7. Spectrogram of THREE at 5Gs

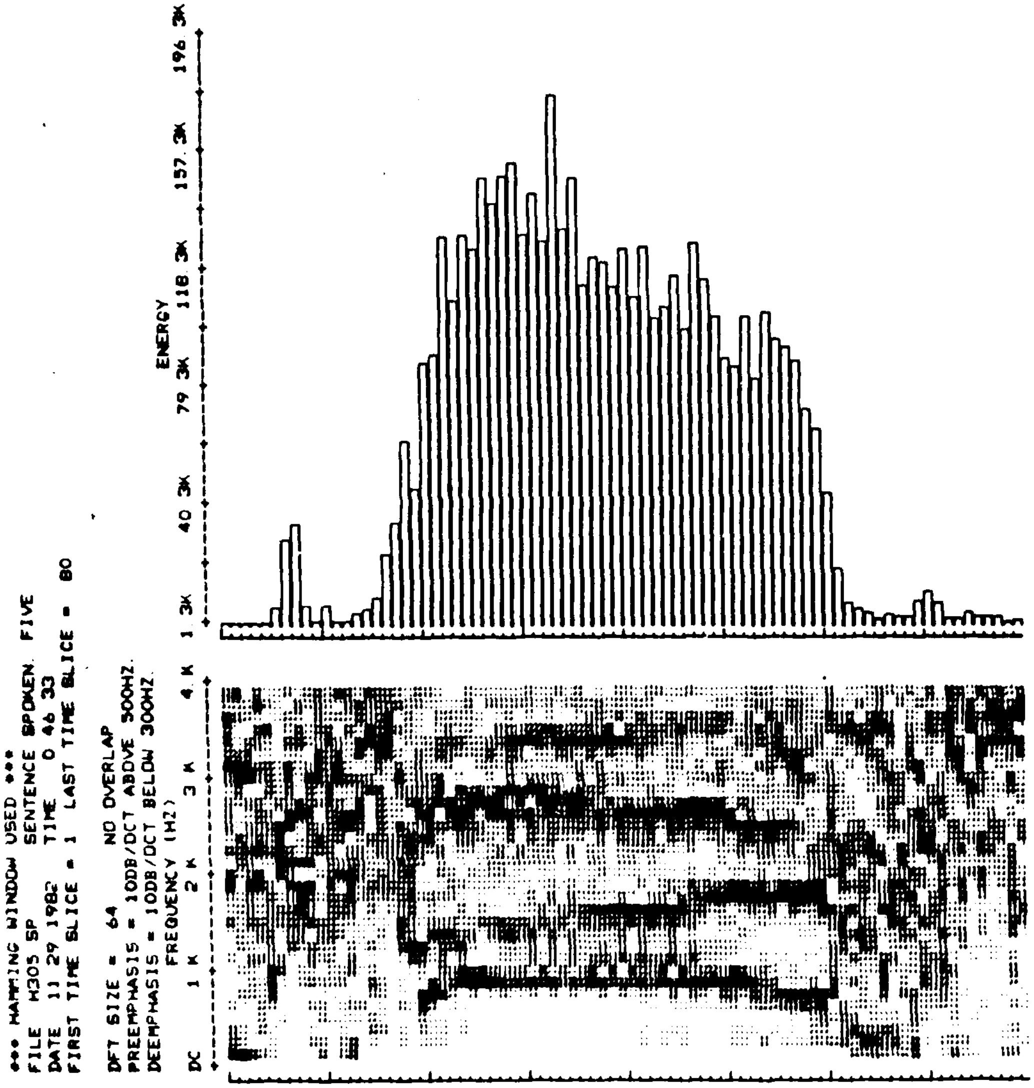


Figure B-8. Spectrogram of FIVE at 3Gs

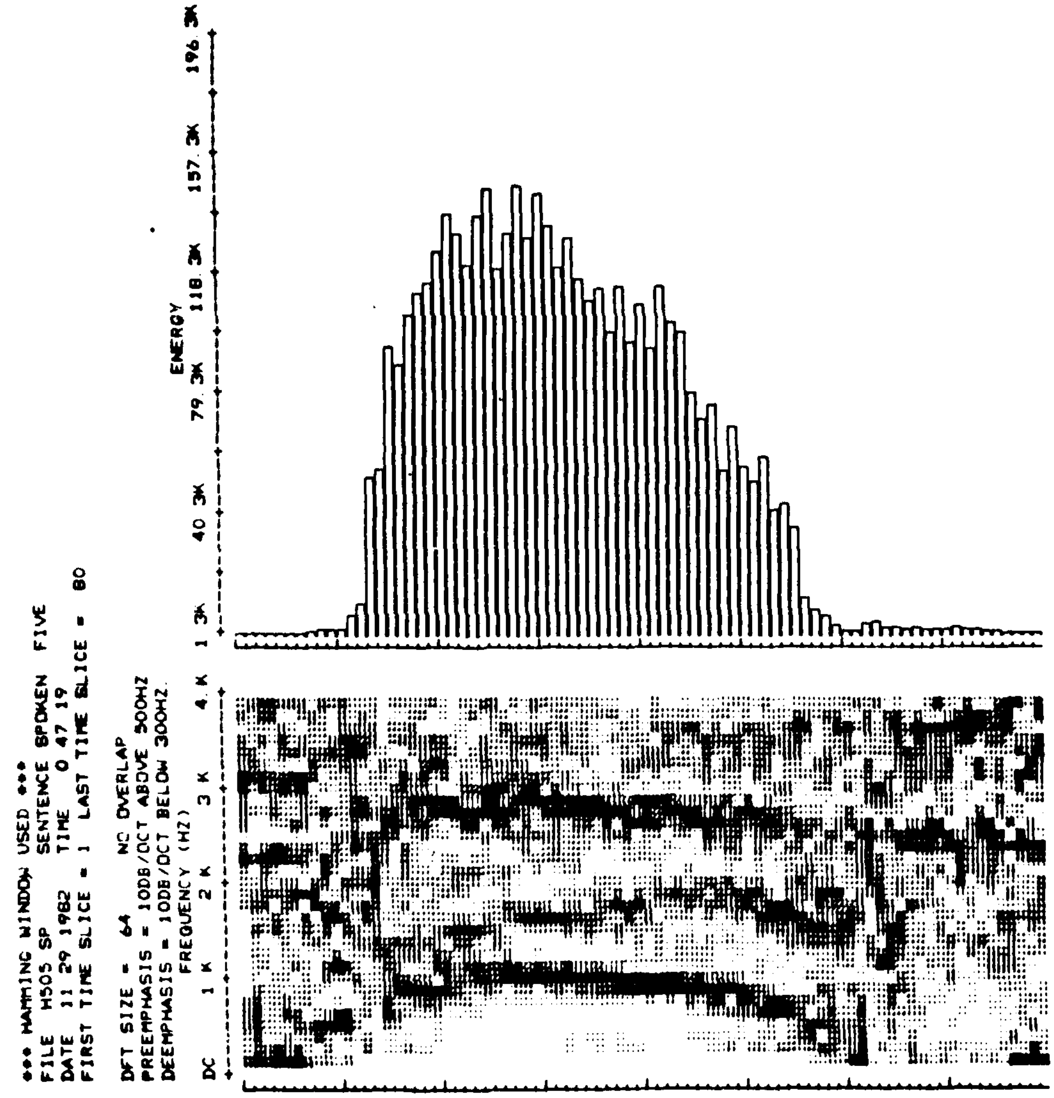


Figure B-9. Spectrogram of FIVE at 5Gs

```
C       PROGRAM:        SPENPLOT
C       LANGUAGE:       FORTRAN5
C       DATE:           7 OCT 82
C       AUTHOR:         K. BEACHY
C       SUBJECT:        SPEECH, PLOTS SPECTRUM AND ENERGY
C
C       COMPILE:        FORTRAN/X SPENPLOT  (FOR REGULAR VERSION)
C                       FORTRAN SPENPLOT (FOR AUTO VERSION)
C                       (RENAME THE AUTO PROGRAM)
C       LOAD:           RLDR SPENPLOT BYTEOUT IOFT5 @FLIB@
C                       RLDR (AUTONAME) BYTEOUT IOFT5 @FLIB@
C
C
C ------THIS PROGRAM CREATES A SPECTROGRAM FROM THE
C OBSERVATION FILES CREATED BY DRVR.
C       These observation files contain frequency components from
C   the speech files that have been DFT by DRVR.
C       The input files consist of a HEADER block (256 integers),
C   followed by data blocks which contain 128 reals per block.
C   For a 64 point DFT there would be 4 vectors(8ms in each vector)
C   in each block of data corresponding to the original speech file.
C
C INITIALIZE VARIABLES
C
        INTEGER FILENM(13),SAID(50)
        INTEGER HEADER(256),STRBLK,NUMVEC,NUMCMP,BLKRD,VECBLK,SFAC
        DIMENSION ISYMBOL1(10),ISYMBOL6(10),IDATE(3)
        DIMENSION IBI(128),DSP(512),IENSYM(6),ISDSYM(6)
        DIMENSION ISYMBOL2(10),ISYMBOL3(10),ISYMBOL4(10),ISYMBOL5(10)
        COMMON/BLK/ISYMBOL1,ISYMBOL2,ISYMBOL3,ISYMBOL4,ISYMBOL5
        COMMON/BLK/ISYMBOL6,IENSYM,ISDSYM
        REAL TFAC
C
C PLOTTER CHARACTERS.
C
        ICHAN = 3
        IPLOT=005K  ;PLOT COMMAND
        ILF=012K    ;PRINT LINE OF DATA JUST SENT.
        INON=101K   ;line components of time axis
        IZER = 100K   ;send nothing to printer
        IONE = 107K   ;separates single vectors on time axis
        ITENT = 137K   ;sends tens markers for time axis
        ITEN=177K  ;DASH USED FOR SCALE ON SGRAM
        IBYTE = 999
        IBLANK=0    ;EMPTY CHARACTER.
        ICOUNT = 0
        ESCALE = 1500.0 ;energy scale increment
        SFAC = 3500     ;changes plot intensity
        TFAC = 10.0   ;scale to set symbols
C       TFAC is later set to 0.1 for those file not normalized
C
C SPECTROGRAM SYMBOLS.
C
        DATA ISYMBOL1/100K,100K,100K,122K,122K,122K,122K,166K,177K,177K/
```

```
      DATA ISYMBOL2/100K,122K,166K,166K,177K,177K,177K,177K,177K,177K/
      DATA ISYMBOL3/100K,100K,100K,100K,100K,122K,133K,133K,133K,177K/
      DATA ISYMBOL4/100K,100K,100K,122K,122K,122K,122K,166K,177K,177K/
      DATA ISYMBOL5/100K,122K,166K,166K,177K,177K,177K,177K,177K,177K/
      DATA ISYMBOL6/100K,100K,100K,100K,100K,122K,133K,133K,133K,177K/
      DATA IENSYM/101K,102K,104K,110K,120K,140K/
      DATA ISDSYM/101K,103K,107K,117K,137K,177K/
C
C INPUT CONTROL VARIABLES, OPEN FILES, & PRINT HEADING ON SGRAM.
C
C FOR SPECTROGRAPH GENERATION
X     IF(ICOUNT.EQ.0) GO TO 3   ;dummy to compile and skip 2
      CALL IOFT5(2,M1,FILENM,SAID,I1,M2,I2,I3,I4)
      CALL OPEN(2,"FILE2",1,IER1)
      IF(IER1.NE.1) TYPE"ERROR ON OPEN, IER1=",IER1
X3    ACCEPT"ENTER FILE WHICH CONTAINS THE SPECTRAL COMPONENTS
X    / <15>     FILENAME = "
X     READ(11,1) FILENM(1)
X1    FORMAT(S13)
X     ACCEPT"<15>   WORD(OR SENTENCE) SPOKEN = "
X     READ(11,2) SAID(1)
X2    FORMAT(S50)
X     CALL OPEN(2,FILENM,1,IER1)
X     IF(IER1.NE.1)TYPE"ERROR ON OPEN, IER1=",IER1
C
C     Read header and set up program to make proper spectrogram
C
      CALL RDBLK(2,0,HEADER,1,IER2)
      IF(IER2.NE.1)TYPE"ERROR ON RDBLK, IER2=",IER2
      NUMVEC = (HEADER(56)-HEADER(55))+1
      IF(HEADER(58).EQ.0.OR.HEADER(63).EQ.1) GO TO 400
      NUMVEC = NUMVEC-1  ;then skip last vector
 400  CONTINUE
      NUMCMP = HEADER(57)
      VECBLK = 128/NUMCMP
      BLKRD = INT((1/VECBLK)-0.1)+1
C   If no normalization was used reset scale
      IF(HEADER(32).EQ.2) TFAC=0.1
C
C     Symbols file give only symbols, no header on output
C
        ;CHECK TO SEE IF SYMBOLS FILE IS DESIRED.
X     ACCEPT "SEND SGRAM TO PRINTER? (Y=1,N=2):",IREPLY
X     IF (IREPLY.NE.1) GO TO 580
      CALL FOPEN(3,"$LPT")
X     GO TO 590
        ;
X580  CALL FOPEN(3,"SYMBOLS")   ;XFER SYMBOLS $LPT,for plot
X590  CONTINUE
X     TYPE"ENTER SCALE FACTOR (3500 ok, lower values darken plot)"
X     ACCEPT "SCALE FACTOR TO SET SPECTRAL INTENSITY = ",SFAC
C
C     Now set up header as needed
C
```

```
      CALL DATE(IDATE,IER)
      IF(IER.NE.1) TYPE "ERROR ON DATE,IER=",IER
      CALL FGTIME(IHOUR,IMIN,ISEC)
      IF(HEADER(31).EQ.1) WRITE(3,581)
 581  FORMAT(1X,"*** HAMMING WINDOW USED ***")
      IF(HEADER(31).EQ.0) WRITE(3,585)
 585  FORMAT(1X,"*** RECTANGULAR WINDOW USED ***")
      WRITE(3,589) FILENM(1),SAID(1)
 589  FORMAT(1X,"FILE: ",S13,"SENTENCE SPOKEN: ",S50)
      WRITE(3,592) IDATE,IHOUR,IMIN,ISEC
 592  FORMAT(1X,"DATE:",2I3,I5,3X,"TIME:",3I3)
      WRITE(3,594) HEADER(55),HEADER(56)
 594  FORMAT(1X,"FIRST TIME SLICE =",I2,"  LAST TIME SLICE = ",I3 /)
      IF(HEADER(58).EQ.1) WRITE(3,583) HEADER(30)
 583  FORMAT(1X,"DFT SIZE = ",I3,5X,"50% OVERLAP")
      IF(HEADER(58).EQ.0) WRITE(3,596) HEADER(30)
 596  FORMAT(1X,"DFT SIZE = ",I3,5X,"NO OVERLAP")
      IF(HEADER(27).EQ.1) WRITE(3,597) HEADER(28),HEADER(29)
 597  FORMAT(1X,"PREEMPHASIS = ",I2,"db/OCT ABOVE ",I3,"hz.")
      IF(HEADER(60).EQ.1) WRITE(3,598) HEADER(61),HEADER(62)
 598  FORMAT(1X,"DEEMPHASIS = ",I2,"db/OCT BELOW ",I3,"hz.")
      IF(NUMCMP.EQ.64) GO TO 602
      IF(NUMCMP.EQ.32) GO TO 610
      GO TO 620
C
C     DFT size = 128 points.  Send scales for 128pt DFT
C
 602  WRITE(3,604)
 604  FORMAT(30X,"FREQUENCY (HZ)",50X,"ENERGY")
      WRITE(3,606)
 606  FORMAT(1X,"DC",13X,"1.K",13X,"2.K",13X,"3.K",13X,"4.K"
     / ,3X,"3K",7X,"93K",6X,"183K",6X,"273K",6X,"363K",6X,"453K"
     / ,6X,"543K")
      WRITE(3,608)
 608  FORMAT(1X,"+-------+-------+-------+-------+---"
     / "----+-------+-------+",4X,"+----+----+----+----+----"
     / "+----+----+----+----+----+----+----+")
      GO TO 620   ;finished with 32 component graph label
C
C     DFT = 128 points, send proper scales
C
 610  WRITE(3,612)
 612  FORMAT(10X,"FREQUENCY (HZ)",36X,"ENERGY")
      WRITE(3,614)
 614  FORMAT(1X,"DC",5X,"1.K",5X,"2.K",5X,"3.K",5X,"4.K",2X,"1.3K"
     / ,5X,"40.3K",5X,"79.3K",4X,"118.3K",4X,"157.3K",4X,"196.3K")
      WRITE(3,616)
 616  FORMAT(1X,"+-------+-------+-------+-------+",4X,"+----+"
     / "----+----+----+----+----+----+----+----+----+")
 620  CONTINUE
        ;
C
C     Reset ESCALE for 64 point DFT.  When ESCALE is changed
C   you must change the scale on the ENERGY scale formats.
```

```
C
      IF(HEADER(30).EQ.64) ESCALE = 650
        ;
        ;
        ;MAIN SECTION OF PROGRAM
        ;
        ;
      CALL BYTEOUT(ICHAN,IBYTE)
      DO 1000 IM=1,NUMVEC
      STRBLK = INT((IM-1)/VECBLK)+1
      CALL RDBLK(2,STRBLK,DSP,BLKRD,IER3)
      IF(IER3.NE.1)TYPE"ERROR ON RDBLK, IER3=",IER3
C
C     The dc component has been replaced by the energy present
C   in the vector before the vector was normalized.
C
      IBI(1) = 1      ;ignore dc component
      IOFF = MOD((IM-1),VECBLK)*NUMCMP
C
C     Set up energy for plot
C
      ENERGY = DSP(1+IOFF)
      ENMAX = 359.0*ESCALE
      IF(ENERGY.GT.ENMAX) ENERGY=ENMAX
C
C     Set up frequency components for plot
C
      DO 245 I=2,NUMCMP
      IBI(I)=INT(DSP(I+IOFF)/SFAC*TFAC)+1
      IF (IBI(I).LE.0) IBI(I)=1
      IF(IBI(I).GT.10) IBI(I)=10
 245  CONTINUE
C
C SAVE SYMBOLS THAT WILL CONSTRUCT THE SPECTROGRAM
C
      IF(ICOUNT.EQ.10.OR.ICOUNT.EQ.0) GO TO 248
      CALL BYTEOUT(ICHAN,IPLOT)
      CALL BYTEOUT(ICHAN,IONE)
      GO TO 249
 248  CALL BYTEOUT(ICHAN,IPLOT)
      CALL BYTEOUT(ICHAN,ITEN)
 249  DO 250 JJ=1,NUMCMP
      JS=IBI(JJ)
      CALL BYTEOUT(ICHAN,ISYMBOL1(JS))
 250  CONTINUE
      DO 251 I=1,4
 251  CALL BYTEOUT(ICHAN,IZER)
      IF(ICOUNT.EQ.10.OR.ICOUNT.EQ.0) GO TO 238
      CALL BYTEOUT(ICHAN,IONE)
      GO TO 239
 238  CALL BYTEOUT(ICHAN,ITENT)
 239  CONTINUE
      DO 254 IX=1,60
      DO 252 IY=1,6
```

```
      IF(ENERGY.GT.FLOAT((6*IX+IY-7)*ESCALE)) GO TO 252
      CALL BYTEOUT(ICHAN,ISDSYM(IY))
      GO TO 256
252   CONTINUE
      CALL BYTEOUT(ICHAN,ITEN)
254   CONTINUE
256   CALL BYTEOUT(ICHAN,ILF)
      CALL BYTEOUT(ICHAN,IPLOT)
      CALL BYTEOUT(ICHAN,INON)
      DO 260 JJ=1,NUMCMP
      JS=IBI(JJ)
      CALL BYTEOUT(ICHAN,ISYMBOL2(JS))
260   CONTINUE
      DO 261 I=1,4
261   CALL BYTEOUT(ICHAN,IZER)
      CALL BYTEOUT(ICHAN,INON)
      DO 264 IX=1,60
      DO 262 IY=1,6
      IF(ENERGY.GT.FLOAT((6*IX+IY-7)*ESCALE)) GO TO 262
      CALL BYTEOUT(ICHAN,IENSYM(IY))
      GO TO 266
262   CONTINUE
      CALL BYTEOUT(ICHAN,IZER)
264   CONTINUE
266   CALL BYTEOUT(ICHAN,ILF)
      CALL BYTEOUT(ICHAN,IPLOT)
      CALL BYTEOUT(ICHAN,INON)
      DO 270 JJ=1,NUMCMP
      JS=IBI(JJ)
      CALL BYTEOUT(ICHAN,ISYMBOL3(JS))
270   CONTINUE
      DO 271 I=1,4
271   CALL BYTEOUT(ICHAN,IZER)
      CALL BYTEOUT(ICHAN,INON)
      DO 274 IX=1,60
      DO 272 IY=1,6
      IF(ENERGY.GT.FLOAT((6*IX+IY-7)*ESCALE)) GO TO 272
      CALL BYTEOUT(ICHAN,IENSYM(IY))
      GO TO 276
272   CONTINUE
      CALL BYTEOUT(ICHAN,IZER)
274   CONTINUE
276   CALL BYTEOUT(ICHAN,ILF)
      CALL BYTEOUT(ICHAN,IPLOT)
      CALL BYTEOUT(ICHAN,INON)
      DO 280 JJ=1,NUMCMP
      JS=IBI(JJ)
      CALL BYTEOUT(ICHAN,ISYMBOL4(JS))
280   CONTINUE
      DO 281 I=1,4
281   CALL BYTEOUT(ICHAN,IZER)
      CALL BYTEOUT(ICHAN,INON)
      DO 284 IX=1,60
      DO 282 IY=1,6
```

```
      IF(ENERGY.GT.FLOAT((6*IX+IY-7)*ESCALE)) GO TO 282
      CALL BYTEOUT(ICHAN,IENSYM(IY))
      GO TO 286
 282  CONTINUE
      CALL BYTEOUT(ICHAN,IZER)
 284  CONTINUE
 286  CALL BYTEOUT(ICHAN,ILF)
      CALL BYTEOUT(ICHAN,IPLOT)
      CALL BYTEOUT(ICHAN,INON)
      DO 290 JJ=1,NUMCMP
      JS=IBI(JJ)
      CALL BYTEOUT(ICHAN,ISYMBOL5(JS))
 290  CONTINUE
      DO 291 I=1,4
 291  CALL BYTEOUT(ICHAN,IZER)
      CALL BYTEOUT(ICHAN,INON)
      DO 294 IX=1,60
      DO 292 IY=1,6
      IF(ENERGY.GT.FLOAT((6*IX+IY-7)*ESCALE)) GO TO 292
      CALL BYTEOUT(ICHAN,IENSYM(IY))
      GO TO 296
 292  CONTINUE
      CALL BYTEOUT(ICHAN,IZER)
 294  CONTINUE
 296  CALL BYTEOUT(ICHAN,ILF)
      CALL BYTEOUT(ICHAN,IPLOT)
      CALL BYTEOUT(ICHAN,INON)
      DO 300 JJ=1,NUMCMP
      JS=IBI(JJ)
      CALL BYTEOUT(ICHAN,ISYMBOL6(JS))
 300  CONTINUE
      DO 301 I=1,4
 301  CALL BYTEOUT(ICHAN,IZER)
      CALL BYTEOUT(ICHAN,INON)
      DO 304 IX=1,60
      DO 302 IY=1,6
      IF(ENERGY.GT.FLOAT((6*IX+IY-7)*ESCALE)) GO TO 302
      CALL BYTEOUT(ICHAN,ISDSYM(IY))
      GO TO 306
 302  CONTINUE
      CALL BYTEOUT(ICHAN,ITEN)
 304  CONTINUE
 306  IF(ICOUNT.NE.10) GO TO 310
      ICOUNT = 0
 310  CALL BYTEOUT(ICHAN,ILF)
      CALL BYTEOUT(ICHAN,IBLANK)
C
C     Keep track of 10 vectors to be marked off.
C
      ICOUNT = ICOUNT+1
C
C END OF SGRAM CONSTRUCTION
C
 1000 CONTINUE
```

```
CALL RESET
STOP
END
```

APPENDIX C

SUBROUTINE IOFT5 and BYTEOUT

Subroutine IOFT5

Subroutine IOFT5 was written by Lt Simmons. The version presented here has been changed slightly. The size of F1 and F2 arrays were increased. Subroutine IOFT5 was used so automatic programs could be run using macrofiles. Subroutine IOFT5 was used to pass information to the automatic programs. In one case, use of IOFT5 saved hours of editing by passing needed information to the main program from a macrofile. Subroutine IOFT5 can be used for any program to send the needed switch information or ASCII strings to the main program. Subroutine IOFT5 was used in program SPENPLOT and program TOP5 to aid automatic program execution using a macrofile.

```
      SUBROUTINE IOFT5(N,MAIN,F1,F2,F3,MS,S1,S2,S3)
C
C     Written by Lt. Simmons         10 Sep 1981
C     Version 2
C
C     This FORTRAN 5 subroutine will read from the file
C     COM.CM (FCOM.CM in the foreground) the program name,
C     any global switches, and up to three local file
C     names and corresponding local switches.
C
C     Calling arguments:
C
C     N is the number of local files and switches to be
C     read from (F)COM.CM.  N must be 1, 2, or 3.
C
C     MAIN is an ASCII array for the main program file name.
C
C     F1, F2, and F3 are the three ASCII arrays to return
C     the local file names.
C
C     MS is a two-word integer array that holds any global
C     switches.
C
C     S1, S2, and S3 are two-word integer arrays that
C     hold the local switches corresponding to F1 through
C     F3 respectively.
C
C
C     Dimension the arrays.
C
      DIMENSION MAIN(13),MS(2)
      INTEGER F1(13),F2(50),F3(7),S1(2),S2(2),S3(2)
C
C     Check the bounds on N.
C
      IF(N.LT.1.OR.N.GT.3)STOP "N out of bounds in IOF."
C
C     Process the data in (F)COM.CM
C
      CALL GROUND(I)  ;Find out which ground program is in
      IF(I.EQ.0)OPEN 0,"COM.CM"       ;Open ch. 0 to COM.CM
      IF(I.EQ.1)OPEN 0,"FCOM.CM"      ;Open ch. 0 to FCOM.CM
      CALL COMARG(0,MAIN,MS,IER)      ;Read from (F)COM.CM
      IF(IER.NE.1)TYPE" COMARG error:",IER
      WRITE(10,1)MAIN(1)              ;Type program name
    1 FORMAT(' Program ',S13,'running.')
      CALL COMARG(0,F1,S1,JER)        ;Read from (F)COM.CM
      IF(JER.NE.1)TYPE" COMARG error (F1):",JER
      IF(N.EQ.1)GO TO 2               ;Test N
      CALL COMARG(0,F2,S2,KER)        ;Read from (F)COM.CM
      IF(KER.NE.1)TYPE" COMARG error (F2):",KER
      IF(N.EQ.2)GO TO 2               ;Test N
      CALL COMARG(0,F3,S3,LER)        ;Read from (F)COM.CM
      IF(LER.NE.1)TYPE" COMARG error (F3):",LER
```

```
2    CLOSE 0
     RETURN
     END
```

AD-A135 833 COMPUTER RECOGNITION OF PHONEMES IN THE PRESENCE OF COCKPIT INDUCED STRESS AND NOISE(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGI.. 2/2

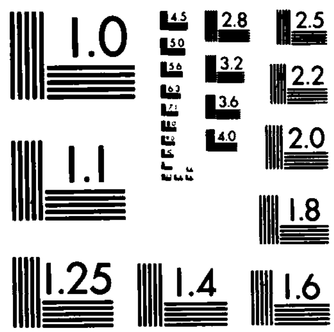
UNCLASSIFIED K A BEACHY 20 DEC 82 AFIT/GE/EE/82D-20 F/G 9/2 NL

END

FILMED

1-84

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Subroutine BYTEOUT

Subroutine BYTEOUT is similar to a subroutine program called BYTEPAC pack by Lt Carl Seelandt. BYTEOUT is used with program SPENPLOT and packs two bytes of information into one memory word. This information is then transferred to the output device, the printer in this case. It was necessary to use this version of BYTEOUT instead of byte pack. This version of BYTEOUT does not send extraneous dots or push up any dots in a line when you are plotting with the Printronix model P3ØØ printer.

```
      SUBROUTINE BYTEOUT(ICHAN,IBYTE)
      IF(IBYTE.EQ.999) GO TO 100
      MASK = 177400K
      IF(IFLAG.EQ.1) GO TO 50
      IOUT = IBYTE
      IOUT = ISHFT(IOUT,8)
      IOUT = IAND(IOUT,MASK)
      IFLAG = 1
      RETURN
50    IOUT = IOR(IOUT,IBYTE)
      WRITE BINARY(ICHAN) IOUT
100   IFLAG = 0
      RETURN
      END
```

APPENDIX D

PROGRAM MKPHON

This program allows the user to develop phoneme templates. These templates can be used by program DRVR to find the distance between the template and an input speech file or any other template or itself. Program MKPHON runs interactively and uses input speech files that consist of frequency components (speech files after DFT and only in the form of files from program DRVR) to develop a phoneme template. For example, if you have an input file and you want to have phoneme #22, you select the vector from the speech file you want to be phoneme 22 and how many consecutive vectors from that input vector are to be included in the new or modified phoneme 22 (if you specify 3 consecutive vectors to be averaged in, and the phonemes are 5-vectors in length, then the next 3 consecutive five-vector groupings will be averaged into the specified five-vector phoneme). When program MKPHON is started it requests the input name of the phoneme file. If this phoneme file is a new file, initialization procedures will begin. The program will ask what values to set for the new phoneme template's characteristics (DFT size, etc). Program MKPHON is now limited to constant length phoneme template. The information requested during initialization is used to fill in the template header (block Ø in the file, see Table D-1).

Table D-I

Header Values used for Program DRVR and MKPHON

ELEMENT	CONTENTS
1-13	Observation file name(channel 4)
14-26	Speech file name(channel 5)
27	Switch: 1=preemphasize Ø=don't preemphasize
28	Preemphasis slope
29	Preemphasis corner frequency
3Ø	Number of time points per FFT
31	Switch: 1=Hamming window Ø=rectangular window
32	**Normalization: 1 = normalize to unity 2 = no normalization Ø = divide by vector energy
33	Switch: 1=create test file Ø=don't create
34-53	not used
54	*Vector length of phonemes
55	Number of first time slice in file
56	Number of last time slice in file
57	Number of points per time slice in file
58	Switch: 1=overlapping Ø=non-overlapping
59	Number of disk blocks in observation file
6Ø	Switch: 1=deemphasis Ø=no deemphasis
61	Deemphasis slope
62	Deemphasis corner frequency
*63	*Switch: 1=phoneme file Ø=not phoneme file
64-256	*Used to store times phoneme has been modified (Can only store 193 modification numbers.)

* Added for use by program MKPHON, which makes phoneme templates.

** Entry 32 is used by program SPENPLOT, so proper spectrogram is plotted.

Table D-I list the values found in a phoneme template file header (block Ø). It is important to initialize the template properly, because only matching input files can be used to modify a phoneme template.

MKPHON averages in each vector requested into the phoneme template. Phoneme averaging for each vector is done by the following equation:

$$P^*_j = \frac{(P_j * n) + V_j}{n + 1}$$

P^* - new phoneme value
P - old phoneme value
V - input value to be averaged
n - times phoneme has been modified
j - phoneme vector component number

Each new modification to the phoneme template is equally weighted with previous vectors. When each phoneme is averaged (modified) it is also renormalized. The dc component has been replaced by energy and is not normalized. Care must be taken not to enter a phoneme from a speech file which has been made from an overlapping 128 point DFT, which may contain erroneous information because not enough extended memory was used. This type of error can be seen in Appendix M where an overlapping 128 point DFT phoneme template was used for recognition results.

The normal operation of program MKPHON is to enter a speech file which consist of frequency components from program DRVR (or files in the same format). The format of the input files as well as the phoneme template format is:

Block Ø - a header of 256 integers which contains important file information (see Table D-1).

Remaining Blocks - multiple data entries consisting of an energy value and the appropriate number of frequency components for each vector.

Multiple files will most likely be entered to make one phoneme template. The phoneme templates can be used by programs developed by Martin to find the distance between the template and other speech files. When the template has been made it can be changed later in part or whole as necessary. Every time the program is used it should be terminated from the main menu in order to properly fill out the header (block 0) for any changes made. If variable length phoneme templates are required major modifications will have to be made in MKPHON.

```
C       PROGRAM:        MKPHON
C       LANGUAGE:       FORTRAN 5
C       DATE:           10 SEP 82
C       AUTHOR:         K. BEACHY
C       SUBJECT:        SPEECH, PHONEME GENERATION
C       LAST REVISION:  26 NOV 82
C
C
C       COMPILE:        FORTRAN MKPHON
C       LOAD:           RLDR MKPHON @FLIB@
C
C       This program allows the user to store speech in
C       multiples of 8ms time slices.  These slices can
C       then be used as templates for the program "DRVR".
C
C       VLENGTH----Length in vectors of phoneme
C
C       INUM-------Real components per phoneme vector
C
C       VECBLK-----Number of phoneme vectors per 128 component block
C
C       ISIZE------Total size of one phoneme
C
C       BLKSRD-----Block needed to read or write
C
C       STRBLK-----Block to start read or write for phoneme
C
C       PHST-------Array position to start at
C
C       START------Block to start read on input file
C
C       VECST------Array position to start at
C
C       IPT--------Storage pointer for modification storage
C
C       STATUS(9)--Is the number of the last block in file
C       for CALL STAT(file,..)
C
C
C       Start program
C
        INTEGER FILEN(13),VLENGTH,VECBLK,ISIZE,BLKSRD,OPTION
        INTEGER FLAG,PHNUM,IPH,STRBLK,LASTPH,STP,START,VECST
        INTEGER IPT,STATUS(18),PHST,HEADER(256),HEADER2(256)
        INTEGER DIFF,TPLATE(13),VECLFT,STRB,MAXPHON
        DIMENSION AAR(512),PHAR(512)
C
C       Enter name of new or exisiting phoneme template
C       This template can be used with program DRVR to find the
C       distance between the phonemes and each vector in a speech file
C
        ACCEPT"ENTER FILENAME FOR PHONEME TEMPLATE.<15>
     *     TEMPLATE FILE = "
        READ(11,1) TPLATE(1)
```

```
      CALL FOPEN(4,TPLATE)
      CALL RDBLK(4,0,HEADER,1,IER5)
      IF(IER5.EQ.9) GO TO 116
      IF(IER5.NE.1)TYPE"ERROR ON RDBLK, IER5=",IER5
C
C     Check template header to see if template is new
C     if new initialize-if not skip initialization
C
      IF(HEADER(63).EQ.1) GO TO 111
C
C     INITIALIZE THE PHONEME TEMPLATE HEADER(BLOCK 0)
C
 116  DO 115 J1=1,256 ;Clear header
      HEADER(J1) = 0
 115  CONTINUE
 112  TYPE"INITIALIZE NEW PHONEME PROTOTYPES<15>"
      ACCEPT"Max number of phonemes for TEMPLATE = ",MAXPHON
      ACCEPT"Enter 0 for NO preemphisis<15>      1 for
     / preemphisis<15>      Preemphisis = ",HEADER(27)
      ACCEPT"Preemphisis slope(db) = ",HEADER(28)
      ACCEPT"Preemphisis corner frequency(hz) = ",HEADER(29)
      ACCEPT"Enter 0 for NO preemphisis,<15>      1 for
     / Deemphisis.<15>      Deemphisis = ",HEADER(60)
      ACCEPT"Deemphisis slope(db) = ",HEADER(61)
      ACCEPT"Deemphisis corner frequency(hz) = ",HEADER(62)
      ACCEPT"Enter 0 for rectangular window,<15>      1 for
     / haming window.<15>      WINDOW = ",HEADER(31)
      ACCEPT"Enter 0 for non-overlapping,<15>      1 for
     / overlapping.<15>      OVERLAP = ",HEADER(58)
      ACCEPT"Length of phonemes(in vectors) = ",HEADER(54)
      TYPE"      Enter number of components per vector.<15>
     / Note: A 128 point FFT = 64 components per vector."
      ACCEPT"Number of components per vector = ",HEADER(57)
      HEADER(30) = HEADER(57)*2
      HEADER(55) = 1  ;First time slice or phoneme
      HEADER(63) = 1   ;value set to show TPLATE has been init
C   £OF TIME SLICES=MAXPHON*VECTOR PER PHONEME
      HEADER(56) = MAXPHON*HEADER(54)
C
C     Now print out values and check initialization
C
      TYPE"<15><15>      TEMPLATE INIALIZATION VALUES"
      TYPE"<15>Number of phonemes in template =",MAXPHON
      TYPE"Vectors(time slices) per phoneme =",HEADER(54)
      TYPE"Total vectors in template =",HEADER(56)
      TYPE"Preemphisis(0=no, 1=yes) =",HEADER(27)
      TYPE"Slope(db) =",HEADER(28)
      TYPE"Corner frequency(hz) =",HEADER(29)
      TYPE"Deemphisis(0=no, 1=yes) =",HEADER(60)
      TYPE"Slope(db) =",HEADER(61)
      TYPE"Corner frequency(hz) =",HEADER(62)
      TYPE"Points per time slice =",HEADER(57)
      TYPE"Points per FFT =",HEADER(30)
      TYPE"Window(0=rectangular, 1=haming) =",HEADER(31)
```

```
      TYPE"Overlap(0=no overlap, 1=overlap) =",HEADER(58)
C
 114  TYPE"<15><15>MAIN >OPTIONS:"
      TYPE" 1 = CONTINUE<15> 2 = RE-INITIALIZE"
      ACCEPT"<15> OPTION = ",IOPT
      IF(IOPT.EQ.2) GO TO 112
C
C     Set values for program operations
C

 111  VLENGTH = HEADER(54)
      MAXPHON = HEADER(56)/VLENGTH
      INUM = HEADER(57)
      VECBLK = 128/INUM   ;128 reals per block
      ISIZE = INUM*VLENGTH
      BLKSRD = INT((VLENGTH-1)/VECBLK)+1
      GO TO 35
C
C     Be sure to close chan 2 at proper times!!!
C
 30   CALL CLOSE(2,IER6)
      IF(IER6.NE.1)TYPE"ERROR ON CLOSE, IER6=",IER6
      GO TO 35
 34   TYPE"TRY AGAIN, ERROR ON OPEN IER3 =",IER3
C
C     Enter source file for vectors to add to phoneme template
C     This file will be checked for compatibility with template
C
 35   TYPE"ENTER FILENAME WHICH CONTAINS FREQUENCY"
      TYPE"     COMPONENTS.(FROM MARTIN'S PROGRAM)"
      ACCEPT"<15>      FILENAME = "
      READ(11,1) FILEN(1)
 1    FORMAT(S13)
      CALL OPEN(2,FILEN,1,IER3)
      IF(IER3.NE.1) GO TO 34
C
C
C
 50   WRITE(10,52) FILEN(1)
 52   FORMAT("<15> PRESENT INPUT FILE: ",S13)
C
C
C     CHECK SOURCE FILE FOR COMPATIBLE VALUES
C
      CALL RDBLK(2,0,HEADER2,1,IER12)
      IF(IER12.NE.1)TYPE"ERROR ON RDBLK, IER12=",IER12
      DIFF = 0
      IF(HEADER(27).NE.HEADER2(27)) DIFF = DIFF+1
      IF(HEADER(28).NE.HEADER2(28)) DIFF = DIFF+1
      IF(HEADER(29).NE.HEADER2(29)) DIFF = DIFF+1
      IF(HEADER(30).NE.HEADER2(30)) DIFF = DIFF+1
      IF(HEADER(31).NE.HEADER2(31)) DIFF = DIFF+1
      IF(HEADER(57).NE.HEADER2(57)) DIFF = DIFF+1
      IF(HEADER(58).NE.HEADER2(58)) DIFF = DIFF+1
```

```
      IF(HEADER(60).NE.HEADER2(60)) DIFF = DIFF+1
      IF(HEADER(61).NE.HEADER2(61)) DIFF = DIFF+1
      IF(HEADER(62).NE.HEADER2(62)) DIFF = DIFF+1
      IF(DIFF.EQ.0) GO TO 150
      TYPE"THE INPUT FILE AND THE PHONEME FILE ARE NOT
     /  COMPATIBLE!"
C
C     Tell user about problem and show values
C
      TYPE"Number of difference(s) = ",DIFF
      WRITE(10,56) FILEN(1)
 56   FORMAT("VALUES FOR TPLATE AND ",S13," RESPECTIVELY")
      TYPE"Preemphisis(0=no, 1=yes) =",
     / HEADER(27),HEADER2(27)
      TYPE"Slope(db) =",HEADER(28),HEADER2(28)
      TYPE"Corner frequency(hz) =",HEADER(29),HEADER2(29)
      TYPE"Deemphisis(0=no, 1=yes) =",
     / HEADER(60),HEADER2(60)
      TYPE"Slope(db) =",HEADER(61),HEADER2(61)
      TYPE"Corner frequency(hz) =",HEADER(62),HEADER2(62)
      TYPE"Points per time slice =",HEADER(57),HEADER2(57)
      TYPE"Points per FFT =",HEADER(30),HEADER2(30)
      TYPE"Window(0=rectangular, 1=haming) =",
     / HEADER(31),HEADER(31)
      TYPE"Overlap(0=no overlap, 1=overlap) =",
     / HEADER(58),HEADER2(58)
 120  TYPE"<15>INPUT FILE NOT COMPATIBLE WITH PHONEME
     / TEMPLATE<15>"
C
C     Give user option on mistake
C
      TYPE">OPTIONS:"
      TYPE" 1 = READ IN NEW FILE"
      TYPE" 2 = TERMINATE PROGRAM"
      TYPE" 3 = RE-INITIALIZE TEMPLATE"
      ACCEPT"<15> OPTION = ",IOP
      IF(IOP.EQ.1) GO TO 30
      IF(IOP.EQ.2) GO TO 40
      CALL CLOSE(2,IER16)
      IF(IER16.NE.1)TYPE"ERROR ON CLOSE, IER16=",IER16
      IF(IOP.EQ.3) GO TO 112
      GO TO 120
C
C     Main option list(should always terminate program here)
C
 150  TYPE"<15>>MAIN OPTIONS:"
      TYPE" 1 = MAKE PHONEME TEMPLATE"
      TYPE" 2 = READ IN NEW FILE"
      TYPE" 3 = CHANGE MAX NUMBER OF PHONEMES"
      TYPE" 4 = TERMINATE PROGRAM"
      ACCEPT"<15> OPTION = ",IOPTION
C
      IF(IOPTION.EQ.1) GO TO 25
      IF(IOPTION.EQ.2) GO TO 30
```

```
      IF(IOPTION.EQ.3) GO TO 160
      IF(IOPTION.EQ.4) GO TO 40
      GO TO 150
C
 160  WRITE(10,8) MAXPHON,HEADER(54)
 8    FORMAT(1X,I3," IS THE MAX NUMBER OF PHONEMES IN TEMPLATE "
    /"WITH",I2," VECTORS PER PHONEME.")
      ACCEPT"ENTER, NEW MAXIMUM = ",MAXPHON
      HEADER(56) = MAXPHON*HEADER(54)
      GO TO 150
C
 170  WRITE(10,9) MAXPHON
 9    FORMAT("TRY AGAIN, Template's max phoneme number set at "
    /,I3,)
 25   WRITE(10,52)FILEN(1)
      TYPE"<15>>OPTIONS: 0 = FORM NEW PHONEME"
      TYPE"          1 = AVERAGE IN PHONEMES"
      ACCEPT"          2 = return to main options    >OPTION = ",
    /IOPTION
      IF(IOPTION.EQ.0) GO TO 10
      IF(IOPTION.EQ.1) GO TO 20
      GO TO 150
C
 20   WRITE(10,6) HEADER(54)
 6    FORMAT(/"AVERAGE IN PHONEMES.",I3," VECTORS PER PHONEME.")
      IFLAG = 1
      GO TO 15
C
 10   WRITE(10,7) HEADER(54)
 7    FORMAT(/"FORM NEW PHONEME.",I3," VECTORS PER PHONEME.")
      IFLAG = 0
C
C
C     READ IN PHONEME
C
 15   ACCEPT"<15>     PHONEME NUMBER = ",PHNUM
      IF(PHNUM.GT.MAXPHON) GO TO 170
      IPH = PHNUM-1
      STRBLK = INT(((IPH*VLENGTH)+VECBLK)/VECBLK)
      PHST = MOD(IPH,VECBLK)*INUM
      CALL RDBLK(4,STRBLK,PHAR,BLKSRD,IER2)
      IF(IER2.NE.1.AND.IER2.NE.9)TYPE"ERROR ON RDBLK, IER2=",IER2
C
C
C     READ IN VECTOR
C
      GO TO 220
 230  TYPE"    TRY AGAIN, LAST AVAILABLE VECTOR =",HEADER2(56)
 220  ACCEPT"ENTER, FIRST VECTOR OF PHONEME = ",STR
      GO TO 210
 200  TYPE"     TRY AGAIN, CONSECUTIVE VECTORS LEFT =",VECLFT
 210  ACCEPT"ENTER, TOTAL CONSECUTIVE PHONEMES TO AVERAGE = ",
    /NUMVEC
      VECLFT = HEADER2(56)-STR+1
```

```
      IF(STR.GT.(HEADER2(56)-(HEADER(54)-1))) GO TO 230
      IF(NUMVEC*HEADER(54).GT.VECLFT) GO TO 200
      DO 100 JX=1,NUMVEC
      STRB = STR+JX-2
      START = INT((STRB+VECBLK)/VECBLK)        ;SET BLOCK FROM -.OB
      VECST = MOD(STRB,VECBLK)*INUM   ;0,1,...,LAST VECTOR
      CALL RDBLK(2,START,AAR,BLKSRD,IER4)
      IF(IER4.NE.1)TYPE"ERROR ON RDBLK, IER4=",IER4
C
C
C     AVERAGE VECTOR WITH PHONEME
C
C     IPT, IS STORAGE FOR MOD NUMBER IN TEMPLATE HEADER
      IPT = PHNUM+63  ;CAN STORE 193 PHONEMES
C
      IF(JX.GT.1) IFLAG=1
      IF(IFLAG.NE.0) GO TO 60
      HEADER(IPT) = 0
C
 60   DO 70 J=1,ISIZE ;AVERAGE PHONEMES
      PHAR(J+PHST)=((PHAR(J+PHST)*HEADER(IPT))
     / +AAR(J+VECST))/(HEADER(IPT)+1)
 70   CONTINUE
C
      HEADER(IPT) = HEADER(IPT)+1
C
      WRITE(10,54) PHNUM,HEADER(IPT)
 54   FORMAT("PHONEME ",I3,", HAS BEEN MODIFIED ",I2," TIME(S).")
C
C
C
C     NORMALIZE EACH VECTOR
C
      JOFF = 1+PHST
      ICOMPS = INUM-1
      DO 90 IIN=1,VLENGTH
      SUME = 0.0
      DO 80 J=1,ICOMPS
 80   SUME = SUME+(PHAR(J+JOFF)**2)
      ENERGY = SQRT(SUME)
      DO 85 J=1,ICOMPS
 85   PHAR(J+JOFF) = (10000)*(PHAR(J+JOFF))/ENERGY
      JOFF = (JOFF+ICOMPS)+1
 90   CONTINUE
C
C
C
      CALL WRBLK(4,STRBLK,PHAR,BLKSRD,IER7)
      IF(IER7.NE.1)TYPE"ERROR ON WRBLK, IER7=",IER7
 100  CONTINUE
C
      GO TO 25   ;return to mod or add to phonemes
C
 40   CALL STAT(TPLATE,STATUS,IER10)
```

```
IF(IER10.NE.1)TYPE"ERROR ON STAT, IER10=",IER10
HEADER(59) = STATUS(9)+1
CALL WRBLK(4,0,HEADER,1,IER8)
IF(IER8.NE.1)TYPE"ERROR ON WRBLK, IER8=",IER8
CALL RESET
STOP
END
```

APPENDIX E

Program TOP5

Program TOP5 takes distance files from DRVR and prepares that data for use by program LEARN. TOP5 output can be seen in Figure E-1. TOP5 also decides on the beginning and end point of speech based on the energy present in each vector of speech (8 ms for 64 point DFT and 8k Hz sample on original speech). In addition, TOP5 creates a listing of the top phoneme choice for each vector and for use in resynthesis of speech.

TABLE E-I

Distance File Header

ELEMENT	CONTENTS
1-13	Distance file name
14-26	Observation file name
27-39	Phonet file name
4Ø	Number of first observation time slice to do
41	Number of last observation time slice to do
42	Number of first phonet time slice to do
43	Number of last phonet time slice to do
44	Number of disk block that holds first observation
45	Number of disk block that holds first phonet
46	Switch: 4=observation and phonet files identical
47	not used
48	Number of observation time slices to do
49	Number of phonet time slices to do
5Ø	Number of elements per time slice
51	Number of extended memory blocks used
52-57	not used
58	*Switch: 1=overlapping Ø=non-overlapping
59-256	not used

*-Added to subroutine DSTN of program DRVR, the value here will act as a switch for program TOP5.
Distance files are created by option 3of DRVR.

Figures E-1, E-2, E-3, E-5, and E-7 are examples of output from program TOP5 and E-4, E-6, and E-8 are output from CHOICE5.

FOUR
HCA4. SP

THE DATE IS-- 11 13 1982
THE TIME IS-- 12 6 10

VECTOR NUMBER ******	FIRST CHOICE *******	SECOND CHOICE *******	THIRD CHOICE *******	FOURTH CHOICE *******	FIFTH CHOICE *******	SCALE FACTOR *******	VECTOR ENERGY *******
66	24 100	7 87	8 83	27 74	25 69	.67982320	217
67	24 100	61 70	50 66	1 66	34 64	.52316360	478
68	24 100	61 59	25 51	1 51	51 45	.42029860	530
69	24 100	6 79	29 76	30 70	39 68	.70054860	461
70	24 100	61 72	1 63	34 63	50 62	.50106670	561
71	24 100	6 68	60 66	25 66	61 62	.58762570	496
72	24 100	61 74	1 69	25 65	68 60	.51447730	644
73	24 100	25 86	61 81	60 76	1 74	.65757390	543
74	24 100	61 73	1 72	25 70	68 65	.54708930	633
75	24 100	25 84	61 81	1 71	60 70	.62298080	522
76	24 100	61 78	1 73	25 71	68 64	.54785130	588
77	24 100	25 95	61 83	60 76	68 74	.67007010	538
78	24 100	25 80	61 80	6 77	1 76	.66260280	634
79	25 100	24 86	6 82	39 76	60 76	.75038090	532
80	24 100	61 72	25 68	6 63	1 63	.56263330	595
81	25 100	24 98	6 84	39 79	50 78	.68790000	479
82	24 100	39 94	6 91	25 89	61 89	.77613530	472
83	25 100	6 71	39 68	60 64	61 60	.49146600	547
84	6 100	39 92	24 88	30 88	25 87	.76958240	490
85	66 100	27 96	31 94	25 87	69 84	.71319720	491
86	5 100	39 88	30 87	60 84	61 82	.76211520	440
87	5 100	4 83	60 79	25 78	66 76	.59646450	484
88	5 100	27 99	56 96	4 93	26 93	.83846380	336
89	5 100	4 98	60 93	56 92	26 91	.78619320	340
90	27 100	56 92	5 83	7 82	4 78	.86162750	192
91	7 100	8 88	27 68	24 61	38 52	.57360560	113

FOUR
HCA4. SP

THE DATE IS-- 11 17 1982
THE TIME IS-- 11 18 53

VECTOR NUMBER ******	FIRST CHOICE *******	SECOND CHOICE *******	THIRD CHOICE *******	FOURTH CHOICE *******	FIFTH CHOICE *******	SCALE FACTOR *******	VECTOR ENERGY *******
66	24 100	7 93	8 92	27 75	64 63	.68401490	217
67	24 100	34 76	61 73	51 68	50 64	.48327140	478
68	24 100	61 56	51 53	34 49	25 49	.44609660	530
69	24 100	6 79	29 74	30 73	7 69	.71375460	461
70	24 100	61 68	34 66	51 64	50 59	.49442370	561
71	24 100	25 70	6 68	61 63	31 60	.62081780	496
72	24 100	61 76	25 73	1 71	51 63	.54646840	644
73	24 100	25 84	61 81	1 79	60 72	.68401490	543
74	24 100	1 75	25 74	61 72	68 63	.57620820	633
75	24 100	25 81	61 77	1 75	1 70	.65799250	522
76	24 100	61 78	1 76	25 74	1 65	.57992560	588
77	24 100	25 93	1 81	61 79	68 75	.71003720	538
78	24 100	25 91	1 87	61 86	39 82	.73977690	634
79	25 100	24 76	39 75	6 74	66 72	.72118960	532
80	24 100	25 78	61 76	6 73	1 72	.64684010	595
81	24 100	25 88	27 76	6 74	39 73	.64684010	479
82	24 100	61 97	30 91	39 91	6 90	.81784390	472
83	25 100	6 80	39 69	31 68	30 66	.52044610	547
84	25 100	6 99	24 91	30 90	60 85	.79925650	490
85	66 100	27 92	31 90	33 89	25 85	.70260220	491
86	5 100	30 92	61 90	4 88	39 88	.82156130	440
87	5 100	4 92	66 91	38 83	61 82	.66914490	484
88	56 100	5 97	4 97	27 96	19 91	.86617100	336
89	4 100	5 98	38 93	19 90	56 89	.78438660	340
90	27 100	7 85	56 79	8 79	19 74	.74721190	192
91	7 100	8 89	27 83	24 59	29 55	.62825270	113

Figure E-1. Feature Extraction for FOUR
M2 (top) and M1 (bottom) Distance, Single-Vector

ONE

HCP1.SP

THE DATE IS-- 11 13 1982
THE TIME IS-- 11 32 40

VECTOR NUMBER	FIRST CHOICE	SECOND CHOICE	THIRD CHOICE	FOURTH CHOICE	FIFTH CHOICE	SCALE FACTOR	VECTOR ENERGY
******	*******	*******	*******	*******	*******	*******	*******
10	8 100	7 79	27 54	13 50	38 45	.41288820	84
11	8 100	7 66	13 57	22 50	42 43	.51829710	52
12	8 100	7 82	13 64	24 60	42 59	.56480530	166
13	24 100	7 88	8 82	38 80	66 73	.72561600	228
14	24 100	9 77	50 71	69 68	51 68	.58711180	467
15	9 100	17 85	10 82	11 63	21 63	.61277150	521
16	9 100	10 64	17 63	54 61	24 60	.51115320	460
17	9 100	17 68	10 65	11 54	69 54	.35063420	585
18	9 100	17 71	10 70	47 54	21 52	.36608830	644
19	9 100	10 69	17 67	47 64	54 64	.46654030	654
20	9 100	10 75	17 66	47 61	21 56	.30908290	934
21	9 100	10 84	17 78	47 66	21 65	.48403560	792
22	9 100	10 98	17 76	44 64	23 63	.48447290	761
23	10 100	9 84	17 66	47 55	44 54	.31156140	859
24	10 100	9 77	17 67	44 56	40 48	.39043590	586
25	10 100	60 72	5 70	52 67	23 66	.65344800	666
26	10 100	9 65	17 56	44 53	23 50	.25776350	757
27	10 100	9 80	17 79	44 65	16 61	.47251780	562
28	16 100	41 92	11 91	67 82	60 82	.59979590	589
29	11 100	16 90	41 84	69 78	67 73	.40953490	771
30	11 100	41 94	16 75	35 74	67 74	.51698490	637
31	11 100	41 75	58 69	1 64	50 64	.46974770	575
32	11 100	9 72	52 68	41 66	3 66	.44204690	798
33	11 100	69 72	41 71	52 68	50 68	.56407640	587
34	62 100	11 98	20 93	42 91	3 88	.70301790	111
35	12 100	13 89	42 87	34 76	61 74	.44146380	103
36	12 100	42 88	13 85	34 69	61 68	.31214460	165
37	12 100	13 87	42 87	64 63	61 63	.27598770	129
38	12 100	13 91	42 83	64 69	34 68	.45268990	79

Figure E-2. Feature Extraction for ONE
M2 Distance, Single-Vector Template

ONE
HCP1.SP

THE DATE IS-- 11 17 1982
THE TIME IS-- 10 48 19

VECTOR NUMBER	FIRST CHOICE		SECOND CHOICE		THIRD CHOICE		FOURTH CHOICE		FIFTH CHOICE		SCALE FACTOR	VECTOR ENERGY
******	*******		*******		*******		*******		*******		*******	*******
10	8	100	7	87	27	59	13	57	42	45	.43388430	84
11	8	100	7	76	13	61	38	52	17	50	.53305780	52
12	8	100	7	82	13	59	24	55	64	55	.56198350	166
13	24	100	8	100	7	94	26	80	9	79	.94214870	228
14	24	100	51	78	9	76	69	74	50	71	.72314050	467
15	9	100	17	84	10	80	21	67	11	66	.71074380	521
16	9	100	17	75	10	71	6	66	11	64	.61570250	460
17	9	100	10	69	17	68	11	67	47	59	.40495870	585
18	9	100	10	77	17	72	47	56	11	55	.47933880	644
19	9	100	10	74	17	69	47	63	6	61	.43388430	654
20	9	100	10	80	17	67	47	66	11	62	.36363630	934
21	9	100	10	92	17	79	47	67	21	64	.59917350	792
22	9	100	10	99	17	76	11	70	47	69	.54132230	761
23	10	100	9	85	17	63	47	62	21	59	.34297520	859
24	10	100	9	79	17	66	47	54	44	48	.43801650	586
25	10	100	9	82	47	63	11	62	29	62	.57024790	666
26	10	100	9	74	11	61	47	59	17	59	.30165280	757
27	10	100	9	80	17	78	11	61	16	59	.53305780	562
28	41	100	16	100	11	98	29	95	67	94	.71900820	589
29	11	100	16	91	41	86	69	84	67	81	.52066110	771
30	11	100	41	91	16	75	67	71	3	69	.53305780	637
31	11	100	41	80	50	72	58	69	20	68	.52066110	575
32	11	100	41	74	16	69	9	68	3	66	.41322310	798
33	11	100	41	74	69	69	9	65	3	65	.64049580	587
34	11	100	42	94	62	92	3	90	20	86	.78925620	111
35	12	100	13	97	42	91	34	82	61	74	.52892560	103
36	12	100	42	89	13	88	34	72	64	71	.39669420	165
37	12	100	13	91	42	88	64	68	34	63	.36363630	129
38	12	100	13	92	42	82	64	71	34	71	.50000000	79

Figure E-3. Feature Extraction for ONE
M1 Distance, Single-Vector Template

ONE
HCP

THE DATE IS-- 9 9 1982
THE TIME IS-- 9 42 54.

VECTOR NUMBER	FIRST CHOICE		SECOND CHOICE		THIRD CHOICE		FOURTH CHOICE		FIFTH CHOICE		SCALE FACTOR
******	******		******		******		******		******		******
9	07	100	08	98	01	76	13	76	27	70	.77520730
10	08	100	07	61	24	51	38	48	13	43	.00030663
11	08	100	24	89	07	65	38	76	54	70	.74336790
12	08	100	24	99	07	89	09	82	54	82	.81813620
13	09	100	24	96	54	80	08	79	06	78	.74314500
14	09	100	24	68	17	74	54	72	06	70	.69664710
15	09	100	25	73	24	70	06	69	17	68	.64803630
16	09	100	25	49	06	48	10	48	17	45	.00027789
17	09	100	10	84	25	71	06	64	17	63	.60434800
18	09	100	10	81	25	75	17	67	06	66	.65814080
19	09	100	10	67	25	73	06	67	17	66	.61813490
20	10	100	09	60	25	66	06	60	17	58	.57194840
21	10	100	09	84	06	70	25	68	17	64	.65434260
22	10	100	09	83	06	67	05	63	17	61	.61997290
23	10	100	09	49	05	43	26	42	08	40	.00024874
24	10	100	26	77	09	72	29	72	05	67	.71060850
25	10	100	09	85	29	85	11	82	67	82	.80592440
26	10	100	41	90	11	68	29	81	50	79	.78916450
27	41	100	11	97	69	91	50	90	29	87	.88856750
28	11	100	41	88	50	82	69	81	67	78	.65036960
29	11	100	41	83	50	54	65	51	16	47	.00022758
30	11	100	41	87	50	78	69	78	67	72	.72028900
31	50	100	31	99	03	98	11	98	62	98	1.00000000
32	42	100	62	83	12	79	03	74	31	74	.77346090
33	12	100	42	99	62	81	64	78	13	77	.65090520
34	12	100	42	63	13	62	64	59	62	56	.00028986
35	12	100	13	99	42	83	64	86	62	76	.60653560
36	13	100	12	91	64	87	42	80	61	70	.69472660
37	13	100	12	88	64	64	42	73	62	71	.64065800
38	13	100	12	86	64	74	62	69	42	67	.56834790
39	13	100	12	81	64	75	07	65	42	65	.53113600
40	13	100	12	70	64	69	07	59	42	59	.59725180
41	13	100	08	75	12	74	64	74	07	72	.66633650

Figure E-4. Feature Extraction for ONE
M1 Distance, Five-Vector Template

ONE
H301.SP

THE DATE IS-- 11 14 1982
THE TIME IS-- 19 44 47

VECTOR NUMBER ******	FIRST CHOICE *******	SECOND CHOICE *******	THIRD CHOICE *******	FOURTH CHOICE *******	FIFTH CHOICE *******	SCALE FACTOR *******	VECTOR ENERGY *******
14	8 100	7 86	27 53	13 48	38 47	.58571880	83
15	8 100	7 68	22 60	13 58	64 51	.46445420	61
16	8 100	17 79	7 72	22 71	9 67	.76612140	158
17	17 100	9 90	22 85	8 82	10 73	.76376220	166
18	17 100	9 78	22 68	10 67	21 64	.70242210	246
19	17 100	16 73	10 65	23 64	44 61	.60050330	375
20	17 100	9 84	10 74	21 69	44 64	.68748030	524
21	9 100	17 99	10 79	44 75	16 67	.74756210	739
22	9 100	17 98	10 82	21 74	16 71	.75605540	1108
23	9 100	17 89	10 81	11 79	44 78	.77492920	1124
24	9 100	17 91	10 85	11 80	21 72	.75369610	1187
25	9 100	17 99	44 90	51 78	24 76	.86741110	1074
26	9 100	10 77	50 74	17 72	69 71	.75794270	1145
27	9 100	10 95	21 93	17 89	47 78	.79584770	1489
28	9 100	17 96	10 91	21 86	44 68	.79285940	1939
29	17 100	10 87	9 82	21 81	43 78	.84429060	1685
30	17 100	9 84	10 81	21 80	16 70	.90028310	2293
31	17 100	10 97	9 94	21 85	43 78	.80843030	1517
32	10 100	43 86	23 84	44 83	47 79	.74551740	1331
33	23 100	16 97	63 94	44 93	17 91	1.00000000	1184
34	23 100	16 99	44 98	18 94	63 92	.82793330	1325
35	45 100	18 94	38 91	41 90	1 87	.92371810	1138
36	45 100	66 89	41 85	35 84	46 83	.78279330	1102
37	35 100	41 97	45 95	1 89	18 87	.90893360	1210
38	45 100	1 96	35 93	46 93	43 90	.94385020	819
39	54 100	43 98	16 97	1 93	23 92	.74221450	208
40	16 100	18 93	3 91	43 88	35 86	.75542620	187
41	16 100	35 97	18 88	23 86	1 85	.88266750	110
42	15 100	33 92	1 90	34 90	64 89	.62236550	91
43	62 100	33 99	64 96	66 94	3 92	.70179300	91
44	64 100	33 92	13 86	62 86	42 83	.64941800	62

Figure E-5. Feature Extraction for ONE at 3Gs
M2 Distance, Single-Vector Template

ONE
H30

THE DATE IS-- 9 3 1982
THE TIME IS-- 17 19 18

VECTOR NUMBER	FIRST CHOICE		SECOND CHOICE		THIRD CHOICE		FOURTH CHOICE		FIFTH CHOICE		SCALE FACTOR
******	******		******		******		******		******		******
12	07	100	08	95	01	86	13	77	64	74	.63598940
13	08	100	07	97	64	77	15	76	17	70	.55226770
14	08	100	07	94	64	76	17	74	22	74	.56067990
15	08	100	07	96	17	87	64	79	35	78	.67335410
16	08	100	17	99	07	90	54	86	38	78	.73760150
17	17	100	08	86	38	81	54	76	09	77	.71085040
18	17	100	09	85	54	81	08	79	16	78	.73690070
19	17	100	16	87	09	84	38	83	54	81	.76387530
20	17	100	09	97	16	96	54	87	35	85	.84923450
21	17	100	09	98	16	94	54	94	24	90	.89253960
22	50	100	09	94	69	94	24	90	16	89	.86437150
23	50	100	09	96	69	92	66	84	16	82	.85682090
24	09	100	10	90	50	84	24	81	69	79	.87524150
25	09	100	10	87	17	78	50	72	16	71	.88009860
26	09	100	10	79	17	70	47	66	21	62	.94140080
27	10	100	09	94	17	89	55	77	21	75	.92263150
28	10	100	09	90	17	89	55	80	47	77	.94309850
29	10	100	09	77	17	76	44	74	48	70	.88194410
30	10	100	44	95	70	88	14	84	17	81	.97769460
31	70	100	44	96	10	94	45	94	48	89	1.00000000
32	45	100	46	81	70	81	25	79	44	77	.68359550
33	45	100	46	91	41	90	11	89	70	80	.50672270
34	45	100	41	88	11	82	48	81	35	69	.79606800
35	45	100	48	91	41	87	11	84	38	79	.68019450
36	48	100	11	95	45	97	41	91	54	90	.94595800
37	16	100	41	99	15	98	49	94	11	93	.96245160
38	15	100	03	93	33	95	43	92	31	91	.89926810
39	33	100	03	98	64	95	62	93	31	92	.84073500
40	62	100	12	95	64	97	31	93	13	92	.92786210
41	12	100	13	92	64	96	42	94	31	93	.75262150
42	13	100	64	92	12	83	62	82	31	76	.70183930
43	13	100	64	90	01	82	12	81	38	74	.76012830

Figure E-6. Feature Extraction for ONE at 3Gs
M1 Distance, Five-Vector Template

ONE
H501.SP

THE DATE IS-- 11 14 1982
THE TIME IS-- 20 15 6

VECTOR NUMBER ******	FIRST CHOICE *******		SECOND CHOICE *******		THIRD CHOICE *******		FOURTH CHOICE *******		FIFTH CHOICE *******		SCALE FACTOR *******	VECTOR ENERGY *******
19	8	100	7	77	17	73	24	68	9	64	.57320300	64
20	9	100	17	95	10	81	21	71	47	62	.76734120	752
21	9	100	10	97	21	87	17	82	11	77	.92909970	554
22	9	100	17	99	10	95	21	90	47	74	.77976270	656
23	9	100	17	90	10	89	21	84	11	76	.82358680	721
24	9	100	17	95	10	88	21	84	11	77	.80209350	779
25	9	100	17	86	11	85	10	85	21	82	.94975570	745
26	9	100	11	88	21	88	52	85	10	84	1.00000000	859
27	11	100	9	99	52	99	40	92	28	87	.96161900	893
28	1	100	52	97	1	96	1	94	9	93	.93649690	763
29	5	100	60	90	1	87	6	86	39	86	.91374740	672
30	69	100	50	98	39	83	25	80	1	78	.70607110	698
31	1	100	50	93	6	90	39	88	1	87	.70928120	807
32	17	100	9	97	10	96	21	84	44	81	.75868800	1707
33	17	100	10	81	9	81	16	77	21	76	.82581990	2336
34	17	100	10	78	9	76	16	75	21	72	.86099090	2656
35	17	100	63	84	10	82	1	81	16	80	.91946960	2686
36	9	100	17	97	18	97	10	90	1	88	.94947660	2229
37	23	100	63	99	17	97	18	97	44	96	.87355200	2036
38	70	100	63	96	18	94	1	93	23	92	.79162590	1853
39	70	100	23	99	1	98	63	98	18	93	.83349620	1692
40	1	100	1	91	18	91	1	87	1	86	.77725050	1876
41	1	100	1	95	18	89	1	88	1	87	.87048150	1910
42	18	100	23	99	1	97	1	96	63	95	.90272150	1693
43	18	100	70	97	1	95	63	94	23	90	.84773200	1939
44	45	100	35	97	18	94	1	93	23	92	.91584090	1837
45	35	100	44	90	1	89	23	86	18	84	.72533140	1552
46	21	100	35	98	11	91	44	88	52	87	.85289600	1308
47	62	100	38	94	1	88	31	84	42	83	.64159100	263
48	62	100	42	96	13	91	12	87	64	85	.52170270	208
49	64	100	33	97	3	91	42	87	62	86	.65652470	159
50	64	100	13	97	42	92	1	87	62	87	.71263080	175
51	13	100	1	98	33	96	1	96	1	95	.74584780	101
52	13	100	1	99	42	90	33	87	61	87	.73342630	89
53	1	100	13	94	1	93	1	89	1	88	.70802510	64
54	33	100	64	88	15	88	66	86	1	86	.56092110	74
55	1	100	1	86	15	83	35	83	1	83	.68011160	81

Figure E-7. Feature Extraction for ONE at 5Gs
M2 Distance, Single-Vector Template

ONE
H50

THE DATE IS-- 9 7 1982
THE TIME IS-- 17 27 51

VECTOR NUMBER	FIRST CHOICE	SECOND CHOICE	THIRD CHOICE	FOURTH CHOICE	FIFTH CHOICE	SCALE FACTOR
******	******	******	******	******	******	******
18	08 100	17 96	01 91	05 90	07 84	.86678300
19	17 100	09 84	08 81	16 70	07 67	.73904350
20	17 100	21 77	09 75	16 67	43 65	.73776950
21	17 100	21 74	09 71	16 69	22 69	.74047990
22	17 100	21 78	22 75	47 70	09 65	.76430080
23	17 100	08 76	22 72	21 70	51 67	.77668620
24	17 100	08 84	22 73	33 73	54 73	.79075190
25	08 100	17 93	38 93	33 90	54 84	.85891380
26	08 100	69 91	09 90	24 90	33 90	.83852360
27	50 100	24 99	69 97	39 92	38 85	.79372530
28	50 100	09 98	24 96	29 96	25 93	.82399950
29	50 100	09 99	39 93	06 92	10 91	.86558530
30	09 100	10 98	50 88	69 84	08 83	.90566720
31	09 100	10 95	55 91	36 85	17 83	.97117440
32	55 100	10 98	36 95	65 92	09 91	1.00000000
33	36 100	65 96	55 95	10 94	14 91	.97198820
34	36 100	65 94	10 88	55 86	14 85	.93038000
35	36 100	65 97	70 97	10 91	14 89	.93637720
36	70 100	38 90	44 90	65 90	10 88	.89505360
37	70 100	44 89	23 88	36 87	69 86	.90645880
38	70 100	48 97	23 93	44 93	37 91	.97042100
39	70 100	48 98	32 98	23 87	37 87	.97727550
40	70 100	48 94	45 92	63 88	32 85	.96842350
41	70 100	45 97	48 92	35 89	48 89	.95276590
42	45 100	46 99	35 93	11 92	70 90	.96345570
43	45 100	11 97	46 92	35 88	58 88	.96427980
44	31 100	03 97	43 94	62 94	48 90	.95260520
45	42 100	62 89	31 82	12 81	13 79	.80977060
46	13 100	42 95	12 93	62 91	64 91	.75259040
47	13 100	64 86	12 82	42 79	62 75	.58009630
48	13 100	64 85	41 78	12 77	33 76	.62960760
49	13 100	64 77	12 68	42 68	62 65	.61693980
50	13 100	64 79	12 72	33 71	42 64	.66938120
51	13 100	12 67	64 65	33 64	62 63	.70416060
52	13 100	12 70	64 69	31 66	33 66	.75276590
53	13 100	12 70	62 69	64 69	31 63	.71392170
54	13 100	31 73	12 72	64 72	33 71	.72357980
55	13 100	12 77	64 70	31 68	42 65	.71209070
56	13 100	12 75	62 73	64 68	31 66	.61907020
57	13 100	12 75	62 71	64 64	71 62	.65020120
58	13 100	01 79	12 78	64 75	71 73	.72078230

Figure E-8. Feature Extraction for ONE at 5Gs
M1 Distance, Five-Vector Template

```
C      PROGRAM:        TOP5 AND ATOP5
C      LANGUAGE:       FORTRAN5
C      DATE:           18 SEP 82
C      AUTHOR:         K. BEACHY
C      SUBJECT:        SPEECH, LIST TOP PHONEMES
C      LAST REVISION:  15 OCT 82
C
C    Regular interactive operation.
C
C      COMPILE:        FORTRAN/X TOP5
C      LOAD:           RLDR TOP5 IOFT5 @FLIB@
C
C    For auto execution rename source file to ATOP5.FR and
C  compile without the /X option.
C
C      COMPILE:        FORTRAN ATOP5
C      LOAD:           RLDR ATOP5 IOFT5 @FLIB@
C      TO USE:         Name input file "FILE3" and enter
C                      ATOP5 speaker(or file) word
C                      The speaker and the word are passed to the
C                      main program by subroutine IOFT5
C                      Example: ATOP5 HCPO.SP ENTER
C
C
C    This program takes blocks of data prepared by program
C  DRVR (written by Martin, see his Thesis DEC 1982).
C  The information in the data blocks is put in a form to be
C  used by program LEARN.
C  The first block is a header block of pertinent file
C   information.  The header is a 256 integer array.
C    The data is in the remaining blocks and arranged as follows:
C  14 integer elements for each time slice of original speech.
C    The 14 elements are:
C  1  Time slice number of file.
C  2  Energy of slice(useful since each time slice is normalized.
C  3  Phoneme number with maximum distance to time slice
C  4  Maximum distance
C  5  Phoneme number that has the minimum distance to time slice
C  6  Minimum distance
C  7  Phoneme number that has the next minimum distance
C  8  Next minimum distance
C  .
C  .
C  13 Phoneme number with the 5th minimum distance
C  14 The 5th minimum distance
C    The data is stored 14 elements followed by the next 14
C  for all the data.  The total number of time slices
C  is found in the header(1st Block) HEADER(48)
C
C    VARIABLES AND VALUES
C
C    FILENM(13)---Holds input filename
C    FILE3---Is the auto program filename
C
```

```
C    HEADER(256)---Holds header information from file
C
C    STATUS(18)---Receives information on the file of
C      interest.This program uses CALL STAT() to find out
C      how many Blocks should be read from the
C      input file FILENM or FILE3.
C
C      STORE(5888)---Input file is stored in this array.
C      This array can handle 23 blocks of input which is
C      equal to 100 blocks of original speech (2.3sec).
C      This is based on distance choices every 8 ms.
C
C      SPEAKER(13)---When available the speaker is read
C      from the header.
C
C      SAID(50)---Up to 50 characters may be specified for
C      what the speaker said.
C
C      MAXMIN---The maximum minimum distance (of the top choice distance)
C
C      VALUE(5)---Holds the values for each time slice. This
C      value is the relative score for each phoneme for that time slice
C
C      SET---is the offset required to read the correct value from
C      the STORE array
C
C      SKIP---a dummy value used for auto execution and to help
C      avoid compiler error("next statement can not be reached")
C
C      START PROGRAM
C
       INTEGER FILENM(13),HEADER(256),STATUS(18),BSET
       INTEGER STORE(5888),SPEAKER(13),SAID(50),MAXMIN
       INTEGER VALUE(5),SET,IDATE(3),SKIP
       INTEGER THRESH,SETOT,SPEC,NUMPHX,NOISE
       REAL SFACTOR
C
C      Use SKIP for interactive use
C
C
X      SKIP = 1   ;set for interactive program
X      IF(SKIP.EQ.1) GO TO 5
C
C    The next 5 lines are for auto program execution.
C      (which are skipped over for interactive use)
C
       CALL IOFT5(2,M1,SPEAKER,SAID,I1,M2,I2,I3,I4)
       CALL STAT("FILE3",STATUS,IER)
       IF(IER.NE.1) TYPE"ERROR ON STAT, IER=",IER
       CALL OPEN(1,"FILE3",1,IER2)
       IF(IER2.NE.1) TYPE"ERROR ON OPEN, IER2=",IER2
C
C      Ask for interactive information
C
```

```
X5      TYPE"ENTER FILENAME WHICH CONTAINS DISTANCE
X     / <15>      INFORMATION. (FROM DRVR PROGRAM)"
X       ACCEPT"<15>      FILENAME = "
X       READ(11,1) FILENM(1)
X1      FORMAT(S13)
C
X       ACCEPT"<15>     WORD SPOKEN = "
X       READ(11,2) SAID(1)
X2      FORMAT(S50)
C
C       Set blocks to be read in
C
X       CALL STAT(FILENM,STATUS,IER)
X       IF(IER.NE.1)TYPE"ERROR ON STAT, IER=",IER
        BLOCKS = STATUS(9)+1
C
C
X       CALL OPEN(1,FILENM,1,IER2)
X       IF(IER2.NE.1)TYPE"ERROR ON OPEN, IER2=",IER2
C
        CALL RDBLK(1,0,HEADER,1,IER3)
        IF(IER3.NE.1)TYPE"ERROR ON RDBLK, IER3=",IER3
C
        CALL RDBLK(1,1,STORE,STATUS(9),IER4)
        IF(IER4.NE.1)TYPE"ERROR ON RDBLK, IER4=",IER4
C
        CALL CLOSE(1,IER5)
        IF(IER5.NE.1)TYPE"ERROR ON CLOSE, IER5=",IER5
C
C       Set up output file that contain all vectors
C
        CALL DFILW("OUT5",IER6)
        IF(IER6.NE.1)TYPE"ERROR ON DFILW, IER6=",IER6
        OPEN 4,"OUT5"
C
C       Set file with no noise, to give to recognition PROGRAM
C
        CALL DFILW("OUT7",IER7)
        IF(IER7.NE.1)TYPE"ERROR ON DFILW, IER7=",IER7
        OPEN 5,"OUT7"
C
C       OPEN file for list of top choice, to be used for
C       speech generation
C
        CALL DFILW("OUTT",IERA)
        IF(IERA.NE.1)TYPE"ERROR ON DFILW, IERA=",IERA
        OPEN 3,"OUTT"
C
        CALL FGTIME(IHOUR,IMIN,ISEC)
        CALL DATE(IDATE,IER8)
        CALL CHECK(IER8)
C
C
X       DO 30 I=1, 13
```

```
X       SPEAKER(I) = HEADER(I+13)
X30     CONTINUE
X       WRITE(10,3) SPEAKER(1)
X3      FORMAT("<15>     SPECTRAL FILE: ",S13)
C
C       send header to proper files
C
        WRITE(4,230) SAID(1)
        WRITE(5,235) SAID(1)
 230    FORMAT(15X,"SENTENCE SPOKEN : ",S50 /)
 235    FORMAT(S50)
C
        WRITE(4,240) SPEAKER(1)
        WRITE(5,245) SPEAKER(1)
 240    FORMAT(19X,"SPEAKER WAS : ",S13 /)
 245    FORMAT(S13)
C
        WRITE(4,247) IDATE
        WRITE(4,249) IHOUR,IMIN,ISEC
        WRITE(5,247) IDATE
        WRITE(5,249) IHOUR,IMIN,ISEC
 247    FORMAT(15X,"THE DATE IS--"2I3,I5)
 249    FORMAT(15X,"THE TIME IS--",3I3 //)
C
        WRITE(4,250)
        WRITE(4,260)
        WRITE(4,270)
        WRITE(5,250)
        WRITE(5,260)
        WRITE(5,270)
C
 250    FORMAT(4X,"VECTOR",4X,"FIRST",3X,"SECOND",4X,"THIRD",
     /  3X,"FOURTH",4X,"FIFTH",7X,"SCALE",6X,"VECTOR")
 260    FORMAT(4X,"NUMBER",3X,"CHOICE",3X,"CHOICE",3X,"CHOICE",
     /  3X,"CHOICE",3X,"CHOICE",6X,"FACTOR",6X,"ENERGY")
 270    FORMAT(4X,"******",2X,"*******",2X,"*******",2X,"*******",
     / 2X,"*******",2X,"*******",5X,"*******",5X,"*******" /)
C
C
C   Find the maximum of the minumum distance
C   Also do one vector less than the total number, this will
C  eliminate any anomalies caused by the overlapping window.
C
        MAXMIN = 1
        IDONE = HEADER(48)-1  ;list all but last vector
C
C   If the files are the same or are non-overlapping, do all
C
        IF(HEADER(46).EQ.4.OR.HEADER(58).EQ.1) IDONE=IDONE+1
        DO 40 I=1,IDONE
           SET = (I-1)*14
           IF(MAXMIN.GT.STORE(SET+6)) GO TO 40
           MAXMIN = STORE(SET+6)
 40     CONTINUE
```

```
C
C      Initialize first loop
C
       IBEGIN = 1
       IEND = IDONE
       BSET = 0
C
C      Set threshold for 64 or 32 components
C      HEADER(50):number of components in FFT
C
       IF(HEADER(50).EQ.128) THRESH=100
       IF(HEADER(50).EQ.64) THRESH=50
C
C      FIRST LOOP---this loop sends data to OUT5 and also
C   set the beginning and end vectors for OUT7 file.
C   hopefully this will eliminate noise present before and after word.
C
       DO 55 I=1,IDONE
          SET = (I-1)*14
          DO 50 J=1,5  ;5=NUMBER OF TOP CHOICES
             VALUE(J) = INT(100*FLOAT(STORE(SET+4)-STORE(SET+6+((J-1)*2
     /       FLOAT(STORE(SET+4)-STORE(SET+6)))
 50       CONTINUE
          SFACTOR = FLOAT(STORE(SET+6))/FLOAT(MAXMIN)
C
C      Set begin and end vectors for OUT7. Ignore short term
C      peaks(less than 5 vectors above THRESHOLD)
C
       IF(BSET.EQ.1) GO TO 310
       IF(STORE(SET+2).GT.THRESH) GO TO 300
       COUNT = 0
       GO TO 350
 300   CONTINUE
       COUNT = COUNT+1
       IF(COUNT.GT.4) GO TO 305
       GO TO 350
 305   IBEGIN = I-4
       BSET = 1
       GO TO 350
 310   CONTINUE
       IF(STORE(SET+2).GT.THRESH) GO TO 320
       COUNT = 0
       GO TO 350
 320   CONTINUE
       COUNT = COUNT+1
       IF(COUNT.GT.4) GO TO 330
       GO TO 350
 330   CONTINUE
       IEND = I
 350   CONTINUE
C
C      Send data to OUT5
C
          WRITE(4,280) I,STORE(SET+5),VALUE(1),STORE(SET+7),
```

```
     /    VALUE(2),STORE(SET+9),VALUE(3),STORE(SET+11),VALUE(4),
     /    STORE(SET+13),VALUE(5),SFACTOR,STORE(SET+2)
 55     CONTINUE
C
C       The next section, up to statement 75, is used to generate
C       the file needed for program LEARN (recognition algorithm)
C       First two statements used to adjust beginning or end as desired
C
        IBEGIN = IBEGIN-0
        IEND = IEND+0
        IF(IBEGIN.LE.0) IBEGIN = 1
        IF(IEND.GT.IDONE) IEND = IDONE
C
        SETOT = 1   ;set value for counter used to make OT files

C
        DO 75 I=IBEGIN, IEND
          SET = (I-1)*14
          DO 70 J=1, 5
            VALUE(J) = INT(100*FLOAT(STORE(SET+4)-STORE(SET+6+((J-1)*2)))/
     /      FLOAT(STORE(SET+4)-STORE(SET+6)))
 70       CONTINUE
          SFACTOR = FLOAT(STORE(SET+6))/FLOAT(MAXMIN)
C
C   Set all noise phonemes to 1,or to noise phoneme number
C       Set SPEC, NUMPHX, and NOISE for special phoneme templates.
C   For special templates use proper equations between 73 and 72.
C
C
        SPEC = 126    ;special template
        NUMPHX = 5    ;number of vectors in phoneme
        NOISE = 24    ;number of the noise phoneme
C
        DO 72 IX=1,5
        ISET = SET+3+(2*IX)
C
C   For special templates
C
        IF(HEADER(43).EQ.SPEC) GO TO 73
        IF(STORE(ISET).GE.71) STORE(ISET)=1   ;for 71 + noise phonemes
        GO TO 72
C
C   Special equations to reduce template number into proper phonemes
C
 73     CONTINUE
        STORE(ISET) = INT((STORE(ISET)-1)/NUMPHX)+1
        IF(STORE(ISET).GT.NOISE) STORE(ISET)=NOISE
 72     CONTINUE
          WRITE(5,280) I,STORE(SET+5),VALUE(1),STORE(SET+7),
     /    VALUE(2),STORE(SET+9),VALUE(3),STORE(SET+11),
     /    VALUE(4),STORE(SET+13),VALUE(5),SFACTOR,STORE(SET+2)
C
C       Set up for speech syn.
C
```

```
      WRITE BINARY(3) SETOT,STORE(SET+5)
      SETOT = SETOT+1
C
 75   CONTINUE
C
C
 280  FORMAT(5X,I4,3X,I3,1X,I3,2X,I3,1X,I3,2X,I3,1X,
     / I3,2X,I3,1X,I3,2X,I3,1X,I3,3X,F11.8,4X,I6)
C
      STOP
      END
```

APPENDIX F

AUTOMATIC PROGRAMS

This research needed to use as many automatic programs as possible (programs to process many files at one time automatically). Programs were adjusted to run automatically using a macrofile. When a program runs automatically it is not practical to enter values interactively. IOFT5 can pass information to programs that need input data. The programs listed in this appendix are written in PASCAL, and based upon a program developed by Montgomery (Ref G). These programs are used to make macrofiles interactively. The macrofiles are used to run the programs SPENPLOT, MKPHON, TOP5, and DRVR (including special versions of DRVR). Using the macrofiles enables auto operation of programs.

```
PROGRAM AUTOTOP5 (INPUT,OUTPUT,AOUT );

VAR
   I:INTEGER;
   DIRECTORY:STRING[20];
   TPLATE:STRING[20];
   FILENAME:STRING[20];
   WORD:STRING[30];
   FLAG:BOOLEAN;
   AOUT :TEXT;

BEGIN(*PROGRAM AUTOTOP5*)

   FLAG:=FALSE;
   REWRITE(AOUT );
   DIRECTORY:='                    ';
   WRITELN('INPUT NAME OF DIRECTORY WHERE PROGRAM WILL RUN.');
   WRITE('DIRECTORY = ');
   READLN(DIRECTORY);
   TPLATE:='                    ';
   WRITELN('INPUT NAME OF PHONEME TEMPLATE');
   WRITE('PHONEME TEMPLATE = ');
   READLN(TPLATE);
   REPEAT
      FILENAME:='                    ';
      WRITELN('INPUT NAME OF SPEECHFILE TO BE PROCESSED.');
      WRITE('FILENAME = ');
      READLN(FILENAME);
      WORD:='                              ';
      WRITELN('INPUT WORD(S) SPOKEN.');
      WRITE('WORD(S) = ');
      READLN(WORD);
      WRITELN;
      WRITELN(AOUT ,'DELETE/V FILE1 ');
      WRITELN(AOUT ,'MOVE/V ',DIRECTORY,' ',FILENAME,'/S FILE1');
      WRITELN(AOUT ,'DRVS');
      WRITELN(AOUT ,'DELETE/V FILE1 ');
      WRITELN(AOUT ,'MOVE/V',DIRECTORY,' ',TPLATE,'/S FILE1');
      WRITELN(AOUT ,'DRVD');
      WRITELN(AOUT ,'ATOP5',' ',FILENAME,' ',WORD);

      I:=0;
      REPEAT
         I:=I+1;
      UNTIL FILENAME[I]='.';
      I:=I+1;
      FILENAME[I]:='0';
      FILENAME[I+1]:='7';
      WRITELN(AOUT ,'RENAME OUT7 ',FILENAME);

      I:=0;
      REPEAT
```

```
         I:=I+1;
      UNTIL FILENAME[I]='.';
      I:=I+1;
      FILENAME[I]:='O';
      FILENAME[I+1]:='T';
      WRITELN(AOUT ,'RENAME OUTT ',FILENAME);

   UNTIL FLAG;

END(*PROGRAM AUTOTOP5*).
```

```
PROGRAM AUTOSPEN (INPUT,OUTPUT,AOUT );


VAR
   I:INTEGER;
   DIRECTORY:STRING[20];
   FILENAME:STRING[20];
   WORD:STRING[20];
   FLAG:BOOLEAN;
   AOUT :TEXT;


BEGIN(*PROGRAM AUTOSPEN*)

   FLAG:=FALSE;
   REWRITE(AOUT );
   DIRECTORY:='                    ';
   WRITELN('INPUT NAME OF DIRECTORY WHERE PROGRAM WILL RUN.');
   WRITE('DIRECTORY = ');
   READLN(DIRECTORY);
   REPEAT
      FILENAME:='                    ';
      WRITELN('INPUT NAME OF SPEECHFILE TO BE PROCESSED.');
      WRITE('FILENAME = ');
      READLN(FILENAME);
      WORD:='                    ';
      WRITELN('INPUT WORD(S) SPOKEN.');
      WRITE('WORD(S) = ');
      READLN(WORD);
      WRITELN;
      WRITELN(AOUT ,'DELETE/V FILE1 ');
      WRITELN(AOUT ,'MOVE/V ',DIRECTORY,' ',FILENAME,'/S FILE1');
      WRITELN(AOUT ,'DRVS');
      WRITELN(AOUT ,'ASPENPLT',' ',FILENAME,' ',WORD);

      I:=0;
      REPEAT
         I:=I+1;
      UNTIL FILENAME[I]='.';
      I:=I+1;
      FILENAME[I]:='O';
      FILENAME[I+1]:='B';
      WRITELN(AOUT ,'RENAME FILE2 ',FILENAME);

   UNTIL FLAG;

END(*PROGRAM AUTOSPEN*).
```

```
PROGRAM AUTOSPEC (INPUT,OUTPUT,AOUT );


VAR
   I:INTEGER;
   DIRECTORY:STRING[20];
   FILENAME:STRING[20];
   FLAG:BOOLEAN;
   AOUT :TEXT;


BEGIN(*PROGRAM AUTOSPEC*)

   FLAG:=FALSE;
   REWRITE(AOUT );
   DIRECTORY:='                    ';
   WRITELN('INPUT NAME OF DIRECTORY WHERE PROGRAM WILL RUN.');
   WRITE('DIRECTORY = ');
   READLN(DIRECTORY);
   REPEAT
      FILENAME:='                    ';
      WRITELN('INPUT NAME OF SPEECHFILE TO BE PROCESSED.');
      WRITE('FILENAME = ');
      READLN(FILENAME);
      WRITELN;
      WRITELN(AOUT ,'DELETE/V FILE1 ');
      WRITELN(AOUT ,'MOVE/V ',DIRECTORY,' ',FILENAME,'/S FILE1');
      WRITELN(AOUT ,'DRVS');

      I:=0;
      REPEAT
         I:=I+1;
      UNTIL FILENAME[I]='.';
      I:=I+1;
      FILENAME[I]:='O';
      FILENAME[I+1]:='B';
      WRITELN(AOUT ,'RENAME FILE2 ',FILENAME);

   UNTIL FLAG;

END(*PROGRAM AUTOSPEC*).
```

```
PROGRAM AUTODIST (INPUT,OUTPUT,AOUT2 );


VAR
   I:INTEGER;
   DIRECTORY:STRING[20];
   FILENAME:STRING[20];
   FLAG:BOOLEAN;
   AOUT2 :TEXT;


BEGIN(*PROGRAM AUTODIST*)

   FLAG:=FALSE;
   REWRITE(AOUT2 );
   DIRECTORY:='                    ';
   WRITELN('INPUT NAME OF CURRENT DIRECTORY.');
   WRITE('DIRECTORY = ');
   READLN(DIRECTORY);
   REPEAT
      FILENAME:='                    ';
      WRITELN('INPUT NAME OF SPEECHFILE TO BE PROCESSED.');
      WRITE('FILENAME = ');
      READLN(FILENAME);
      WRITELN;
      WRITELN(AOUT2 ,'DELETE/V SPCHFILE ');
      WRITELN(AOUT2 ,'MOVE/V ',DIRECTORY,' ',FILENAME,'/S SPCHFILE');
      WRITELN(AOUT2 ,'ADISTSP');
      WRITELN(AOUT2 ,'ALIST5');

      I:=0;
      REPEAT
         I:=I+1;
      UNTIL FILENAME[I]='.';
      I:=I+1;
      FILENAME[I]:='0';
      FILENAME[I+1]:='2';
      WRITELN(AOUT2 ,'RENAME OUT2 ',FILENAME);
      I:=0;
      REPEAT
         I:=I+1;
      UNTIL FILENAME[I]='.';
      I:=I+1;
      FILENAME[I]:='0';
      FILENAME[I+1]:='3';
      WRITELN(AOUT2 ,'RENAME OUT3 ',FILENAME);

   UNTIL FLAG;

END(*PROGRAM AUTODIST*).
```

APPENDIX G

OTHER EXPERIMENTS

Resynthesized Speech

An experiment was made using Seelandt's phoneme templates to resynthesize normal and G-speech with a mask. The speech files were digitized using Program AUDIOHIST. The digitized speech files were processed by TRYDIST5 and the resultant files from TRYDIST5 were processed by LISTER4 (Ref 1). A product of LISTER4 is the file OUT3. OUT3 consists of the top choice phoneme sound for each time slice (8ms), when the input speech file is compared to the phoneme template. OUT3 is used by program TALK to form resynthesized speech files (Ref 1). The resynthesized speech files are formed from digitized speech samples of Seelandt's phonemes. The resynthesized speech files can be heard by using Program AUDIOHIST.

Forty-five files (HCPØ.SP, HCP1.SP, to HCPT.SP, H30Ø.SP, H3Ø1.SP, to, H3ØT.SP, and H5ØØ.SP, H5Ø1.SP, to, H5ØT.SP) were processed for resynthesis and the same files were processed by program LEARN for recognition results. Steps in obtaining recognition results from LEARN:

1. Digitize speech (AUDIOHIST analog in digitized speech out)

2. Extract features (PHDIST developed from TRYDIST5, compares digitized speech to phoneme template)

3. Output features (CHOICE5 developed from LISTER4, uses ouputs from PHDIST to prepare for input to LEARN with top 5 choices)

4. Recognition (LEARN uses phonme representations and fuzzy variables to score each file against vocabulary. Input file from CHOICE5)

Phoneme representations, listed in Table G-II, were chosen from the control files (C) and used by program LEARN. The overall fuzzy variables and word fuzzy variables were both the same and are listed in Table G-III.

Three people were asked to listen to the resynthesized speech and try to recognize the utterances, given knowledge of the 15 word vocabulary. The results of these listening tests and the recognition results of program LEARN are in Table G-I. The files are represented by C, 3, and 5 which stand for control (1G), 3Gs and 5Gs respectively.

TABLE G-I

RECOGNITION RESULTS

	PEOPLE LISTENING 1ST			PEOPLE LISTENING 2ND			PEOPLE LISTENING 3RD			PROGRAM LEARN		
	C	3	5	C	3	5	C	3	5	C	3	5
ZERO	-	R	-	-	R	R	-	-	-	-	R	-
ONE	R	R	R	R	R	R	R	R	R	R	-	R
TWO	-	-	-	-	-	-	-	-	-	R	-	R
THREE	R	R	-	R	R	-	R	R	-	R	-	R
FOUR	-	-	-	-	-	-	-	-	-	-	R	-
FIVE	-	-	-	-	-	-	-	-	-	-	-	-
SIX	-	-	-	-	-	-	-	-	-	R	R	R
SEVEN	-	-	-	-	-	-	-	-	-	R	-	-
EIGHT	-	-	-	-	-	-	-	R	-	R	-	R
NINE	-	-	-	-	-	-	-	-	-	R	R	R
CCIP	R	-	R	R	-	R	-	-	-	R	R	R
ENTER	-	R	-	-	-	-	R	R	-	R	R	R
FREQUENCY	-	-	-	-	-	-	-	-	-	R	-	R
STEP	-	-	-	-	-	-	-	-	-	R	-	-
THREAT	-	-	-	-	-	-	-	-	-	-	-	-
RECOGNITION	20%			20%			17.8%			60%		

R=CORRECTLY RECOGNIZED

TABLE G-II

PHONEME REPRESENTATION USED

WORD	PHONEME REPRESENTATION
ZERO	5-39-10-38
ONE	13-14-15-67-70
TWO	36-69-22-17-23-10-33
THREE	22-5-47-28-13
FOUR	67-39-67-41
FIVE	36-52-39-47-48
SIX	36-47-28-1-36
SEVEN	19-36-39-17-70
EIGHT	29-30-1-67-40
NINE	70-17-29-57-70
CCIP	36-29-36-29-1-29-30
ENTER	14-57-70-17-47
FREQUENCY	64-47-69-17-70-28-30
STEP	19-3-36-28-17-39
THREAT	67-64-5-47-64-67

TABLE G-III

FUZZY VARIABLES

STHR = 1.0E+00	SUBE = 1.0E+00	SUBF = 1.0E+00
INSE = 1.5E+0	INSF = 1.0E+00	
DELE = 1.0E+00	DELF = 8.0E-01	DELG = 1.5E-01
DCNE = 1.0E+00	DCNF = 1.2E+00	DCNG = 5.0E-01
SFE = 2.0E+00	SFF = 2.0E+00	
CHVE = 4.0E+00	CHVF = 2.5E-01	
STATE= 1.0E+00	STATF= 1.2E+00	STATG= 0.0E+00
THR1E= 1.2E+00	THR1F= 1.0E+00	
THR2E= 1.5E+00	THR2F= 1.0E+00	

Independent Speaker

The main body of this thesis used speech files from one subject. This section used speech files from an independent speaker, and the phoneme template of Figure 18 was used to extract features. The phoneme representation used in this section is identical to that used on page 41. For an idea of how an independent speaker would affect the overall system, Tables G-IV and G-V can be compared to Table V in the main body. This experiment is identical to the one in Table V, page 45 except the speaker is different.

Ninety-Nine files were used with the prefixs of SCA, SCB, SCC, S3Ø, S3A, S5Ø, and S5A. These files are listed in Appendix A. Files for this experiment were processed just like the files whose results are depicted in Table V, page 45.

No training in Table G-IV means, program LEARN was set up for the speaker used in the main body of this Thesis and no statistics were gathered on the independent speakers files. Table G-V used files with no G-stress to train program LEARN for the independent speaker. Files A, B, and C are without G-stress, while files 3, 3A, 5, and 5A have various levels of G-stress (3=3Gz, 3A=3Gz and 1.5Gy, 5=5Gz, 5A=5Gz and 1.5Gy).

TABLE G-IV

RECOGNITION SCORES FOR INDEPENDENT SPEAKER
NO TRAINING

Phoneme Length: 5 vector

Distance Rule: M1

WORDS TO BE RECOGNIZED	FILES NO TRAINING SET A	B	C	RECOGNITION SET 3	3A	5	5A
ZERO	.67	.71	.66	.69	.73	-	-
ONE	.66	.72	.72	.78	.65*	.72*	.54*
TWO	.75	.72	.81	.76*	.81	.72	.65*
THREE	.63*	.60*	.64*	.66*	-	.64*	.73
FOUR	.66	.56*	.65	.65*	-	.63*	.67*
FIVE	.66	.75	.72	.67	.73	.71	.66*
SIX	.80	.86	.74	.79	.83	.84	.80
SEVEN	.58*	.74	.74	.79	.68	.76	.70
EIGHT	.74	.75	.77	.77	-	.74	.77
NINE	.67	.70	.68	.73	.71	.74	.69
CCIP	.68	.67	.72	.75	-	.68	.68*
ENTER	.79	.65*	.71	.81	.78	.51*	.65*
FREQUENCY	.66	.69	.63	.69	.66*	.65*	.68
STEP	.68*	.70*	.81	.70	.75	.83	.72
THREAT	.73	.68	.74	.69	.70*	.70	.73
Percent Correct	80	73.3	93.3	80	63.6	64.3	57.1
MEAN		.702		.729	.730	.705	.691
STANDARD DEVIATION		.061		.053	.059	.084	.062

*Word missed

TABLE G-V

RECOGNITION SCORES FOR INDEPENDENT SPEAKER WITH TRAINING

Phoneme Length: 5 vector

Distance Rule: M1

WORDS TO BE RECOGNIZED	FILES: TRAINING SET			RECOGNITION SET			
	A	B	C	3	3A	5	5A
ZERO	.87	.90	.72*	.75	.83	-	-
ONE	.79	.85	.83	.81	.84	.82	.72
TWO	.77*	.78*	.78*	.74*	.74*	.73*	.71*
THREE	.79	.79	.78	.78	-	.76	.76*
FOUR	.83	.82	.86	.77*	-	.75*	.76
FIVE	.81	.86	.85	.76	.83	.77	.77
SIX	.82	.85	.85	.80	.78	.78	.81
SEVEN	.74	.86	.84	.83	.76	.84	.79
EIGHT	.82	.84	.84	.77*	-	.71*	.76*
NINE	.85	.85	.83	.82	.80	.78	.79
CCIP	.77	.77	.79	.76	-	.75	.67*
ENTER	.88	.81	.85	.83	.83	.66*	.82
FREQUENCY	.78	.78	.80	.74*	.76	.72*	.70*
STEP	.81	.81	.87	.80	.79	.87	.80
THREAT	.73	.73	.75	.66*	.68*	.67*	.68*
Percent Correct	93.3	93.3	86.7	66.7	81.8	57.1	57.1
MEAN		.813		.775	.785	.758	.753
STANDARD DEVIATION		.043		.044	.049	.060	.049

*Word missed

128-point DFT Recognition

This section contains the results of recognition work using a 128-point DFT instead of a 64-point DFT. The 128-point results used a 1-vector template to extract features from the following speech files: HCPØ.SP, HCP1.SP, to HCPT.SP, H3ØØ.SP, H3Ø1.SP, to, H3ØT.SP, and H5ØØ.SP, H5Ø1.SP, to, H5ØT.SP. The results of recognition using program LEARN can be found in Table G-VI. The phoneme representations used in program LEARN for this experiment are listed in Table G-VII. Phoneme representations were chosen from the characteristics of the P files (15 utterances at 1G).

TABLE G-VI

RECOGNITION SCORES FOR 128 POINT DFT

Phoneme Length: 1 vector

Distance Rule: M2

WORDS TO BE RECOGNIZED	FILES TRAINING SET			RECOGNITION SET		
	A	B	P	C	3	5
ZERO	.83	.86	.80	.83	.70	.60*
ONE	.73*	.84	.83	.80	.70	.60*
TWO	.81	.84	.82	.81	.65*	.70*
THREE	.85	.72*	.86	.74	.70*	.57*
FOUR	.81	.85	.75	.80	.67*	.66*
FIVE	.86	.86	.84	.84	.60*	.66*
SIX	.81	.83	.79	.79*	.73	.72*
SEVEN	.77	.83	.83	.81	.70*	.65*
EIGHT	.88	.88	.92	.89	.88	.83
NINE	.81	.81	.78	.80	.64*	.70
CCIP	.78	.83	.79	.81	.80	.73
ENTER	.83	.79	.84	.77	.80	.68
FREQUENCY	.83	.85	.72	.82	.74	.75
STEP	.83	.83	.85	.75*	.77	.71*
THREAT	.83	.79	.83	.82	.78	.76
Percent Correct	93.3	93.3	100	86.7	60	40
MEAN		.820		.805	.724	.688
STANDARD DEVIATION		.041		.036	.073	.068

*Word missed

TABLE G-VII

PHONEME REPRESENTATION USED FOR 128-POINT DFT

WORD	PHONEME REPRESENTATION
ZERO	2-6-7
ONE	8-10-12
TWO	15-17-7
THREE	18-19-21-22
FOUR	23-24-27
FIVE	28-29-31
SIX	32-1-36
SEVEN	37-39-42-13
EIGHT	43-1-44
NINE	13-45-47
CCIP	37-49-22
ENTER	54-1-56
FREQUENCY	28-49-1-22
STEP	65-1-67
THREAT	68-1-70

Vita

Keith A. Beachy, was born on 30 May 1952 in Philadelphia, Pennsylvania. He graduated from Cross Keys High School in Atlanta, Georgia, 1970. In 1974, he graduated from Clemson University with the degree of Bachelor of Science in Electrical Engineering with Honor. He was Commissioned in 1974 in the Air Force and attended Undergraduate Pilot Training (UPT) at Moody AFB, Valdosta, Georgia. He graduated from UPT in 1975 and was assigned to Kadena Air Base, Japan. At Kadena Air Base he served as a KC-135 pilot with the 376 Strategic Wing until entering the School of Engineering, Air Force Institute of Technology in June 1981.

Permanent Address: 2720 Division Street
Burlington, Iowa 52601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/82D-20	2. GOVT ACCESSION NO. AD-A135 833	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER RECOGNITION OF PHONEMES IN THE PRESENCE OF COCKPIT INDUCED STRESS AND NOISE		5. TYPE OF REPORT & PERIOD COVERED MS THESIS
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Keith A. Beachy Captain USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Electrical Engineering Department Air Force Institute of Technology Wright-Patterson AFB, Ohio 45433		12. REPORT DATE 20 December 1982
		13. NUMBER OF PAGES 141
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		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

12 SEP 1983

Approved for public release: IAW AFR 190-17.
LYNN E. WOLAVER
Dean for Research and Professional Development
Air Force Institute of Technology (ATC)
Wright-Patterson AFB OH 45433

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

- -phoneme recognition by prototype matching
- -g-speech analysis
- -phoneme recognition by average magnitude differencing
- -isolated speech recognition

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

Speech recognition algorithms were analyzed using normal and G-stressed speech as an input. Speech samples were recorded in centrifuge tests at the Air Force Medical Research Lab, Wright-Patterson AFB, Ohio. All speech was recorded using the MBU-12/P face mask. The algorithms studied are phoneme-based feature extractors which feed a recognition algorithm based on fuzzy set theory. Three feature extraction

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) algorithm options were analyzed. One option used a phoneme length of 40ms and the other options used a length of 8 ms. The recognition results for all three options using normal speech are above 90%, but the 40ms phoneme length give higher raw scores. For G-stressed speech the 40 ms phoneme length scored greater than 90% while the 8ms phoneme length options scored less than 60%.

FILM

1-8