

AD-A047 240

ROME AIR DEVELOPMENT CENTER GRIFFISS AFB N Y
THE DEVELOPMENT OF A COMPUTER SPEECH PROCESSING SYSTEM AND ITS --ETC(U)
OCT 77 R J NIEDERJOHN, R A CURTIS
RADC-TR-77-310

F/G 17/2

UNCLASSIFIED

NL

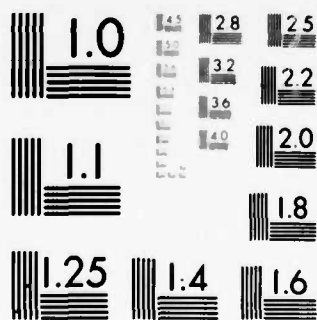
| 0f |

ADAO47 240



END
DATE
FILMED
1 - 78
DDC

A047



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

AD A047240

RADC-TR-77-310
In-house Report
October 1977

12



THE DEVELOPMENT OF A COMPUTER SPEECH PROCESSING SYSTEM AND ITS USE FOR
THE STUDY AND DEVELOPMENT OF PROCESSING METHODS FOR ENHANCING THE
INTELLIGIBILITY OF SPEECH IN NOISE

Russell J. Niederjohn
Marquette University

Robert A. Curtis
Rome Air Development Center

COPY AVAILABLE TO DDC DOES NOT
PERMIT FULLY LEGIBLE PRODUCTION

Approved for public release; distribution unlimited.

AD No. _____
DDC FILE COPY

✓ ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

DDC
RECEIVED
DEC 6 1977
D

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-77-318	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) THE DEVELOPMENT OF A COMPUTER SPEECH PROCESSING SYSTEM AND ITS USE FOR THE STUDY AND DEVELOPMENT OF PROCESSING METHODS FOR ENHANCING THE INTELLIGIBILITY OF SPEECH IN NOISE.	5. TYPE OF REPORT & PERIOD COVERED In-house Interim Report. 6 Jun 77 - 26 Aug 77	
7. AUTHOR(s) Russell J. Niederjohn Robert A. Curtis Marquette University (RADC/IRAD) Milwaukee, Wisconsin 53233 Griffiss AFB NY 13441	6. PERFORMING ORG. REPORT NUMBER N/A	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Rome Air Development Center (IRAD) Griffiss AFB NY 13441	8. CONTRACT OR GRANT NUMBER(s) N/A	
11. CONTROLLING OFFICE NAME AND ADDRESS Same	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 31011G 70530707	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same	12. REPORT DATE October 1977	
	13. NUMBER OF PAGES 102	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
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) Same		
18. SUPPLEMENTARY NOTES RADC Project Engineer: Captain Robert A. Curtis (IRAD)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Speech Enhancement Speech Intelligibility Computer Speech Processing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The enhancement of speech intelligibility in the presence of high noise levels is an important consideration in many practical communication situations. The work documented in this report describes the results of an examination of several methods for processing speech so as to enhance its intelligibility in the presence of wideband random noise. Four particular processing methods were considered: (a) INTEL, a method which involves processing in both the first and second order spectral domains; (b) spectral subtraction, a method which involves (over)		

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

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

309 050

LB

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20. (cont fr p 1)

3 → a simple subtraction of the average noise spectrum from the first-order spectrum; (a) minimum mean square error filtering, a method which involves filtering speech in such a way as to minimize the mean square error between a signal and its expected value in noise; and (a) methods based upon suppressing the frequency content of a speech plus noise signal between pitch harmonics of the speech signal.

4 → To carry out a study of methods to enhance speech intelligibility in noise, two general-purpose computer processing systems were implemented. The first, was a terminal interactive system for the generation, analysis, and graphic display of synthetic voiced speech sounds. Through the use of this system, a considerable insight into the effect of various processing algorithms upon speech and upon speech in noise has been effected.

Approved → The second computer processing system has been developed for the processing of real speech. This system involves the use of a DDP-116 data converter and a Honeywell 6000 Computer. Communication between these two computers is by means of seven track magnetic tape. In use, this system facilitates the input, process, and playback of real speech utterances. Through this system, the effect of numerous processing algorithms upon normal speech in noise has been studied.

While both of these processing systems have been developed for, and applied to, a study of processing techniques for enhancing the intelligibility of speech in noise, the computer programs generated have purposely been made general purpose so as to facilitate their future use at RADC for other speech processing and signal processing tasks.

The effect of several processing algorithms (based upon the four methods mentioned) has been studied for numerous synthetic voiced speech sounds and for two 12-second real speech utterances. These two speech utterances were generated by a male talker: one utterance in a signal-to-noise ratio of +6 dB, the other utterance in a signal-to-noise ratio of -6 dB. Overall results indicate that while the greatest speech enhancement success has been achieved with the INTEL and the minimum mean square error filtering methods, the four methods studied each offer a significant potential for speech intelligibility enhancement in noise.

ACCESSION for	
ATC	White Section ☒
DDC	Buff Section ☐
UNANNOUNCED	☐
JUSTIFICATION.....	
BY.....	
DISTRIBUTION/AVAILABILITY CODES	
Dist.	AVAIL. and/or SPECIAL
A	23 CP

DDC
RECEIVED
DEC 6 1977
D

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ACKNOWLEDGEMENTS

The authors of this report are grateful to many persons who contributed to make this Summer Faculty Research Participation Program at the Rome Air Development Center a very valuable experience.

The help of Dr. Bruno Beek, Mr. Richard Vonusa and Mr. Michael Heffron of the Speech Processing Group at the Rome Air Development Center is sincerely appreciated. Also, the assistance of Mr. Albert Proctor, Mr. David Clark and Mr. William Doane who helped with the preparation of the computer programs and the other work associated with the Honeywell 6000 Computer is appreciated.

To not acknowledge the administrative assistance of Mr. J. Fred O'Brien, Director of the "USAF/ASEE Summer Faculty Research Participation Program" would be a major oversight. The work of Mr. O'Brien resulted in the smooth operation of the program and, in a major way, contributed to the technical success achieved by the project. The authors of this report are very thankful for the help of Mr. O'Brien.

The authors of this report would also like to express their appreciation for the interest and help of several other persons at RADC who contributed to this project. Three of these persons include Mr. John Huss, who coordinated the program at the local level; Mrs. Emma Kimpton who typed this report and Mr. Kirk Starczewski who prepared the figures contained in this report.

Finally, the authors wish to thank the United States Air Force and the American Society for Engineering Education who sponsored and coordinated the activities of the summer research program.

TABLE OF CONTENTS

	PAGE
I. INTRODUCTION	1
A. SPEECH IN NOISE AT THE LISTENER	2
B. SPEECH IN NOISE AT THE SPEAKER	3
C. THIS REPORT	3
II. DISCUSSION	5
III. COMPUTER SYSTEM FOR SYNTHETIC SPEECH PROCESSING	8
A. INTRODUCTION	8
B. SYNTHETIC SPEECH GENERATION PROGRAM	9
C. SYNTHETIC SPEECH ANALYSIS PROGRAM	10
IV. COMPUTER SYSTEM FOR REAL SPEECH PROCESSING	11
A. INTRODUCTION	11
B. SPEECH INPUT/OUTPUT WITH THE DDP-116	12
C. DDP-116/HONEYWELL 6000 COMMUNICATIONS	12
D. HONEYWELL 6000 SPEECH PROCESSING PROGRAM	15
V. RESULTS	16
A. INTEL PROCESSING METHOD	20
1. Discussion	25
2. Tests With Real Speech	28
3. An INTEL Extension	29
B. SPECTRAL SUBTRACTION METHOD	30
1. Discussion	31
2. Tests With Real Speech	33
C. MINIMUM MEAN-SQUARE-FILTERING METHOD	34
1. Discussion	36
2. Tests With Real Speech	36
D. METHODS BASED UPON PITCH TRACKING	37
1. Discussion	39
2. Tests With Real Speech	40
VI. CONCLUSIONS	41
REFERENCES	43
APPENDIX	

LIST OF FIGURES

- Figure 1 - An illustration of the DDP-116/Honeywell 6000 magnetic-tape communication method.
- Figure 2 - An overview of the real speech program "PROCESS." N is the number of processing techniques implemented by the system (presently twelve); M is the number of subprocess processing steps implemented by the system (presently twenty-four).
- Figure 3 - Time waveform of the synthetic vowel /a/ as generated by "SPEECH"; (a) no noise, (b) signal-to-noise ratio of 0 db.
- Figure 4 - The INTEL process applied to the synthetically generated vowel /a/. Each waveform shown is after the corresponding step listed in the first paragraph of Section V-A; left is for a $S/N = \infty$, right is for a $S/N = 0$ dB.
- Figure 5 - The spectral subtraction process applied to the synthetically generated vowel /a/. Steps a-c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-B; left is for a $S/N = \infty$, right is for a $S/N = 0$ dB.
- Figure 6 - The minimum mean square error filtering process applied to the synthetically generated vowel /a/. Steps a-c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-C; left is for a $S/N = \infty$, right is for a $S/N = 0$ dB.
- Figure 7 - The process based upon pitch extraction applied to the synthetically generated vowel /a/. Steps a-c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-D; left is for a $S/N = \infty$, right is for a $S/N = 0$ dB.

I. INTRODUCTION

The understanding of speech contaminated by the presence of noise is an important consideration in many practical communication situations. Consider the airplane or helicopter pilot in a noisy cockpit attempting to communicate with ground based personnel; or consider the worker in the environment of noisy machinery attempting to communicate via telephone or other means with personnel outside the noise environment.

There are two cases for the speech in noise situation. First, there is the case where the noise is present at the speaker (1-6); second, there is the case where the noise is present at the listener (7-19). Each case presents a somewhat different situation. In the case of noise at the speaker, there is an opportunity to suppress the noise (relative to the speech) prior to its reception by the listener, thus resulting in an enhanced signal-to-noise ratio and hopefully more intelligible speech. In the case of noise at the listener, while there is no opportunity to suppress the noise, there is an opportunity to process the speech prior to its encounter with the noise. Conversely, in the first case, since the noise is already present with the speech there is no opportunity to process the uncorrupted speech signal. In the second case, since the noise is in the environment of the listener, there is no opportunity to suppress the noise level.

There are several types of noise which may contaminate a speech signal. These include: impulse noise, large amplitude sine-wave or sum of sine-wave noise, a conflicting speaker, and wideband random noise. The enhancement of speech intelligibility in the presence of each of these noise types is under study by the Rome Air Development Center (1-

3, 20). It is the purpose of the present study to consider the enhancement of speech intelligibility in the presence of wideband random noise.

A. SPEECH IN NOISE AT THE LISTENER

One obvious method for improving the intelligibility of speech in noise at the listener is to simply decrease the level of the noise. Although there exist methods for decreasing the noise levels produced by noisy equipment, these methods often do not reduce the noise levels sufficiently or are too expensive or too inconvenient to be practical.

Another obvious method for improving the intelligibility of speech in noise at the listener is to simply increase the power of the speech signal, thus resulting in a more favorable signal-to-noise ratio. Although such a technique may work in certain low-level noise situations; in high noise levels, the need to conform within a pre-established maximum sound level may prohibit the use of this simple technique. As a result, a method for enhancing the intelligibility of speech in high noise levels, without increasing the signal power, is desirable.

Several studies related to the intelligibility of speech in noise (at the listener) have been reported (7-19). In these studies, various techniques for processing speech (prior to reception by the listener) have been evaluated. Three particular techniques (8, 10, 11) have been recently reported and have been shown to offer intelligibility enhancement for speech in white noise at various signal-to-noise ratios and in one case in the environment of recorded power generating noise (9). These three techniques involve processing speech by: (a) high-pass filtering followed by infinite-amplitude clipping (8), (b) high-pass filtering (11), and (c) high-pass filtering followed by rapid amplitude

compression (8-9). At a signal-to-noise ratio of 0dB (noise at 90 dB (re. .0002 dy/cm²)) an intelligibility of greater than twice normal speech is obtained by processing speech by any of these three methods. At other signal-to-noise ratios a similar intelligibility enhancement is also achieved.

B. SPEECH IN NOISE AT THE SPEAKER

One technique which may sometimes be successfully applied for enhancing the intelligibility of speech in the environment of noise at the speaker makes use of a noise-canceling or close-speaking microphone. In some cases such a microphone may not be sufficient because the noise is at too high a level. In other cases it may not be possible to use such a microphone because it's too inconvenient for the speaker, because the speaker is unable to wear such a microphone, or because the speech is already added to the noise when it is available for listening. As a result, a processing technique for enhancing the intelligibility of speech contaminated with noise is desired. Several techniques have been investigated for the case of noise at the speaker. These techniques include high-pass filtering followed by infinite amplitude clipping (4), a technique known as "INTEL" which involves gating the second-order spectrum and the subsequent retransformation back into the time-domain (1-2), and methods based upon the use of linear prediction (5-6). While the techniques investigated for this case have displayed encouraging results, a significant enhancement in the intelligibility of speech in noise at the speaker has not been achieved.

C. THIS REPORT

While both cases for speech processing in noise have many

practical applications and while both cases are of considerable interest, greater application and interest exists within the Air Force for the later case (i.e., speech in noise at the speaker). As a result, the remainder of this report and the work performed in its preparation are directed specifically toward this later case.

The research work documented in the following sections of this report has included an analysis and study of methods for processing speech so as to result in an enhancement of its intelligibility in the presence of noise at the speaker. In this direction two specific approaches have been pursued. The first approach involves the development of interactive software for the analysis and study of processing techniques applied to synthetically generated voiced speech. For this study, interactive programs have been developed for the Honeywell 6000 accessed via a Tektronix 4002A CRT for the generation, processing, and graphic display of synthetically generated voiced speech sounds. This interactive system is described in Section III of this report and the results obtained with this system are described in Section V.

The second approach involves the development of a general purpose, batch-oriented, speech processing system for real speech and its use in the study of speech in noise processing techniques. This system involves magnetic-tape communication between the DDP-116 data converter and the Honeywell 6000 computer. Programs for analog to magnetic-tape conversion, previously developed by Captain Robert Curtis, are utilized. Programs for the input and output of this magnetic-tape to/from the Honeywell 6000 computer and general purpose programs for processing the data contained on these magnetic-tapes are described in

Section IV of this report. In Section V the results obtained using this system for processing speech contaminated with noise are described.

II. DISCUSSION

The most successful present technique for enhancing the intelligibility of speech in noise at the listener is the technique known as "INTEL" (an acronym for INTElligibility Enhancement by Liftering). This technique has been developed under the direction of the RADC (1-2). While only a small speech intelligibility gain has been achieved by this method, a significant enhancement of the listenability of speech in noise has been demonstrated.

It is one purpose of this work to investigate the operation of the INTEL technique through both an examination of the results of processing synthetic voiced sounds and through the processing of real speech. The results of this examination are presented in Section V.

One problem with the INTEL technique is that it requires four Fourier Transformations, two forward and two reverse. Since calculating a Fourier Transform is computationally time consuming, it is of interest to explore methods which require fewer transforms. As a result, three additional approaches to speech intelligibility enhancement are explored in this work with preliminary results documented in later sections of this report. These three approaches are: (a) spectral subtraction, (b) minimum mean square error (MSE) filtering (21), and (c) methods based upon pitch tracking.

Spectral subtraction is of interest since it is computationally and conceptually relatively simple. It involves the subtraction of the estimated noise spectrum from the transform of the speech plus noise

signal. After subtraction, the resulting spectrum is retransformed into the time-domain. Support for this technique can be gained from an intuitive analysis of speech plus noise and from its seeming similarity to "gating" which is performed on the second-order spectrum in the INTEL technique. One problem with subtraction is that it requires some determination of both the magnitude and the spectrum of the noise. In the INTEL technique, the magnitude of the noise is automatically accounted for (when gating to zero is used) and the spectrum of the noise is assumed flat. In the results presented in Section V for spectral subtraction it is assumed (as a first approximation) that the noise has a flat frequency spectrum (i.e., white) with a magnitude equal to the average first-order spectrum magnitude above 2.5 KHz.

A method for determining a filter which minimizes the mean square error between a signal embedded in noise and its estimated value, assuming the signal and noise are uncorrelated, can be shown to be given by (21):

$$H(j\omega) = \frac{S_{ss}(\omega)}{S_{ss}(\omega) + S_{nn}(\omega)} \quad (\text{eq. 1})$$

where:

$S_{ss}(\omega)$ = the estimated spectrum of the signal, and

$S_{nn}(\omega)$ = the estimated spectrum of the noise.

Using the longtime average speech spectrum for $S_{ss}(\omega)$ and the measured spectrum of the noise for $S_{nn}(\omega)$, the transfer function $H(j\omega)$ can be calculated. Then, multiplying $H(j\omega)$ by the spectrum of the input speech signal and performing an inverse transformation results in the processed speech output.

This method is attractive for two reasons. First, it is an optimum method (in the least mean square sense) for separating a signal from noise. Second, it is computationally simple to implement since it is a simple filtering process which can be performed in the frequency domain after transformation, or perhaps on the time-domain signal using digital filtering techniques. A complication of the method is that it requires an estimate of the noise spectrum. It is unclear at this point how the results of this method vary as a function of the error in estimating the noise spectrum. In the experiments described in Section V it is assumed that the noise spectrum is flat with an average magnitude equal to the magnitude above 2.5 KHz.

Speech in noise enhancement through methods based upon pitch extraction seem intuitively attractive. This is a result of the fact that the energy of a speech signal exists at harmonics of the pitch frequency while the energy of noise is distributed throughout the spectrum. From a knowledge of the pitch frequency, those lines in the spectrum not at harmonics of the pitch frequency can be suppressed leaving, hopefully, speech enhanced in the presence of noise. Several problems influence this method. There is the problem of accurately tracking the pitch frequency, even in a non-noise environment. There is, on the one hand, a desire to analyze speech over a long time segment to gain as much information about the signal as possible. There is, on the other hand, a desire to analyze a short interval so that changes in the pitch frequency during the analysis interval will not be significant. A recent report by Parsons and Weiss (3) suggests that the optimum segment size should be 40.8 msec. A segment of 51.2 msec is

used in this work.

Previous results obtained with methods based upon pitch tracking have not been particularly encouraging (1-2, 20). It has been suggested that even when the pitch frequency is accurately determined, for example from the original uncorrupted speech signal, such methods have not been shown capable of significant speech enhancement in noise. The use of comb filtering, for example, has not been found effective for improving the intelligibility of speech in noise. A discussion of the use of comb filtering for speech in noise enhancement is contained in a recent report by Weiss, Aschkenasy, and Parsons (1). Another report, by Weiss and Aschkenasy (2), discusses an experiment with pitch tracking in which good results were reported when the pitch frequency was found adequate. Details of this method are somewhat sketchy. However, pointed out in this report are the difficulties in determining pitch, particularly at low signal-to-noise ratios and the distortion produced by inaccurate pitch tracking and by analyzing unvoiced speech by a harmonic analysis.

An interesting method for pitch tracking and some interesting results obtained, from a relatively crude processing method based upon pitch tracking, are described in Section V.

III. COMPUTER SYSTEM FOR SYNTHETIC SPEECH PROCESSING

A. INTRODUCTION

In order to provide some insight into the speech in noise situation and into processing speech in noise, a terminal interactive speech processing system was developed. This system makes use of a Tektronix 4002A CRT character/graphics computer terminal connected via telephone modem with the Honeywell 6000 Computer System at RADC. A

program developed by Mr. David Clark of RADC which provides FFT, IFFT, data input/output, and graphic capability was used as a starting point. Added to this program is the capability for analyzing speech by the four methods (with several variations each) discussed in the last section. In addition, a program for generating synthetic voiced speech in a format acceptable to the analysis program was developed. These two programs are described in the next two subsections with program listings in Appendix A and Appendix B.

B. SYNTHETIC SPEECH GENERATION PROGRAM

A program for the generation of synthetic voiced speech called "SPEECH," is listed in Appendix A. This program assumes a sum of three damped sine-waves model for voiced speech. It requests the following input: (a) amplitude, frequency (Hz), and damping rate (Hz) for three formants; (b) pitch period (msec); and (c) RMS signal to RMS noise ratio. The program generates 51.2 msec of the speech signal (non-pitch synchronous, with sample values spaced at 100 us), adds random noise of the specified signal-to-noise ratio, and provides the capability for viewing any generated sample values. Finally, the program outputs the generated speech plus noise samples into a (previously defined) random file, with name specified in the RUN statement (program line number 0010 in program listing, Appendix A). The format of this file is acceptable as input to the synthetic speech processing program. The program runs under GCOS FORT.

A sample run of this program is as follows:

```

*RUN
INPUT: AMPL,FREQ,ALPHA, FOR ALL 3 FORMANTS
=1.,730.,27.
=.5,1090.,28.5
=.04,2240.,46.5
INPUT PITCH PERIOD IN MSEC
=9.
NOISE?(1-YES, 0-NO)
=1
INPUT S(RMS)/N(RMS) IN DB
=0.
VIEW ANY SAMPLES?(1-YES, 0-NO)
=0

```

C. SYNTHETIC SPEECH ANALYSIS PROGRAM.

A very flexible program for the analysis and display of general time-waveforms has been developed by Mr. David Clark of RADC. This program provides the capability for the input and output of data files, for FFT and IFFT analysis, for the display of data, for data generation under cursor control, and for several mathematical operations between two data files. This program has been modified to include the processing of speech by several methods based upon (a) INTEL, (b) spectral subtraction, (c) minimum mean square error filtering, and (d) pitch extraction. In addition, the updated program permits the capability for examining (plotting) the results of any intermediate subprocess within each processing algorithm. The program runs under GCOS TFORT, is named M2 (following an appellation M1 for the previous program), and contains instructions upon initiating a run of the program. A listing of M2 is contained in Appendix B.

IV. COMPUTER SYSTEM FOR REAL SPEECH PROCESSING

A. INTRODUCTION

While the use of synthetic voiced speech is very useful for providing an understanding of both the general speech in noise situation as well as specific speech processing methods; the real proof of a speech in noise enhancement technique is how well it performs for real speech. In an effort to study speech processing methods on real speech a set of programs were developed for the Honeywell 6000. These programs run under the GCOS operating system to facilitate the input of data, the processing of that data, and the output of results. Since the Honeywell 6000 does not provide A/D or D/A capability, the DDP-116 data converter is used. Using the DDP-116, input speech waveforms are sampled (A/D converter) and these samples are written onto magnetic tape. Having a magnetic tape of samples to be output to the DDP-116, a time signal can be constructed using the D/A converter. Programs for the DDP-116 for A/D conversion to magnetic tape and for magnetic tape to D/A conversion, written by Captain Robert Curtis of RADC, were used. Making use of these magnetic tapes of speech data, programs have been written for the Honeywell 6000 Computer System to read-in these magnetic tapes, process the data they contain, and output the results on a second magnetic tape in a format suitable for input (and subsequent D/A output) by the DDP-116.

The software system developed for the Honeywell 6000 has been purposely written in a very flexible, very general, and well commented way so as to facilitate its future use at the RADC for any general speech or signal processing task.

B. SPEECH INPUT/OUTPUT WITH THE DDP-116

Input signals to the DDP-116 are sampled at a 10 KHz rate (10-bit sign-magnitude) and written to magnetic tape in records of 1024 samples (102.4 ms/record). The DDP-116 generates three tape words per sample in a tape format established by the hardware of the system.

For output, the software of the DDP-116 requires magnetic tapes of 1024 samples/record written in the proper sign-magnitude format. The output software created for the DDP-116 can read these magnetic tapes and generate a time-waveform from the sample values. The D/A converter is 10-bit, sign-magnitude.

For both input and output the DDP-116 requires a one word zero record at the beginning of any magnetic tape.

C. DDP-116/HONEYWELL 6000 COMMUNICATIONS

The media for communication between the DDP-116 data converter and the Honeywell 6000 computer is 7-track magnetic tape. The Honeywell 6000 has a memory word size of 36-bits, utilizes two's complement arithmetic, and when communicating with a magnetic tape unit utilizes six tape-words/CPU word. The DDP-116 has a 16-bit word, utilizes sign/magnitude arithmetic, and communicates with a magnetic tape unit with three tape-words/CPU word. A method for tape to data-word and data-word to tape translation was developed for the Honeywell 6000. The essence of the processing necessary for this magnetic tape communication situation is illustrated in Figure 1. Each successive word read from magnetic tape by the Honeywell 6000 consists of two successive words written by the DDP-116 (6 tape-words/CPU word for the Honeywell 6000 and 3 tape-words/CPU word for the DDP-116). Figure 1 illustrates (right) the

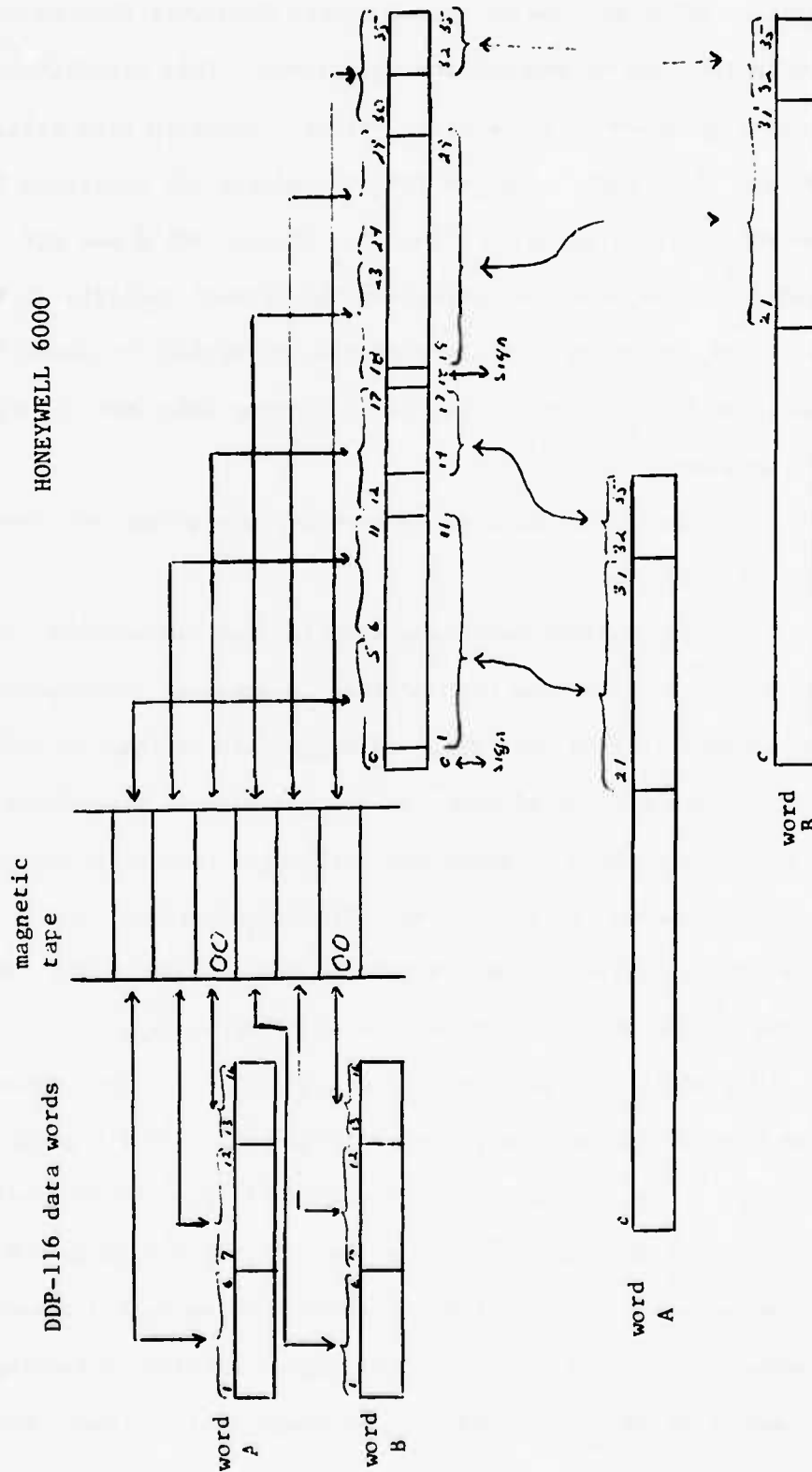


FIGURE 1. An illustration of the DDP-116/Honeywell 6000 magnetic-tape communication method,

translation which must be performed by the Honeywell 6000 to map a tape-read word into two data-words and vice-versa. This translation process requires a sequence of two events. First, tape-word bits 1-11, 14-17, 19-29, and 32-35 must be mapped into the proper bit positions in the two data-words as illustrated in Figure 1. Second, bit 0 and bit 18 of the tape-word must be tested to ascertain the correct polarity of the data-word. If the data-word is to be positive, no action is taken; if the data-word is to be negative, the corresponding data-word is negated (two's complement sense).

To translate a pair of data-words into a tape-word requires the reverse of this process.

The subroutines which are used to read and write a tape record by the Honeywell 6000 have been written in assembly language (GMAP). These programs (called RTB, WTB, and WTBZ) were written in consultation with Mr. Albert Proctor of RADC and are included in Appendices C, D, and E. The processes used to translate data-words to tape-words and tape-words to data-words are written as FORTRAN subroutines (called DATTAP and TAPDAT respectively) and are part of the program called PROCESS described in the next section and listed in Appendix H.

One additional problem exists with magnetic tape communication between the DDP-116 and the Honeywell 6000. The DDP-116 outputs a parity word at the end of each record which is inconsistent with that expected by the Honeywell 6000. As a result, when reading DDP-116 generated magnetic tapes on the Honeywell 6000, a parity error signal to the operator's console results. This signal requires acknowledgement by the computer operator in order for the program to proceed. When reading

a large number of records by this process, the operator tends to prefer to abort the job rather than acknowledge every parity error (as many as 2,000 during a long run).

There exists a software system on the Honeywell 6000, called UTILITY, which may be used to read magnetic tapes ignoring parity errors. Using this system a magnetic tape can be copied onto another magnetic tape (with proper parity for reading) or can be written directly into a data file. In the present work a copy to a second magnetic tape is utilized. This second magnetic tape can then be left as part of the Honeywell 6000 system and the program "UTILITY" need not be used again.

The program UTILITY runs under GCOS CARDIN. A listing of a batchjob for a tape-to-tape copy using utility is shown in Appendix F. A second batch program to read a "UTILITY" generated magnetic tape into a data file is shown in Appendix G.

D. HONEYWELL 6000 SPEECH PROCESSING PROGRAM

The main program for processing real speech on the Honeywell 6000 computer system is called PROCESS and is listed in Appendix H. This program runs under GCOS CARDIN. It reads-in speech data from a file (defined in the batchjob), processes that data, and outputs the results to a magnetic tape (also defined in the batchjob) in a format acceptable for input by the DDP-116. The processing program assumes that some method of overlapping time-windows (of 1024 data samples) is to be applied to the input data. In the present implementation of the program only a triangular time-window is implemented. This processing program is purposely written very flexibly to permit its future use for speech in noise work, for other speech processing work, and for other

signal processing work.

The program has four hierarchal levels of subroutine program depth as illustrated in Figure 2. At the highest level, the main program performs data input, data output, overlapping time windowing, output amplitude normalization, and calls to the various processing techniques. At the second level, the program includes the processing techniques. At the third level, the program includes the subprocessing steps. The fourth level (not illustrated in Figure 2) includes subroutines called by the subprocessing steps. The particular processing techniques to be called may be inserted with appropriate subroutine calls at program lines 310 to 436. A program run may include the performance of any number of processes (defined by NPROC, line 102, maximum value of 20), any number of records to be processed by each process (defined by NREC, line 105), starting with any record (defined by NFREC, line 107). The program creates a magnetic tape of the processed output with each successive processed speech output separated by ten zero records which result in a 1 sec silence gap for use when listening or recording. In addition, the program prints a listing showing each processing technique executed, the numbers of the first and last tape records created, and the CPU execution time required for each processing method.

In its present version, the program includes an implementation of eleven processing techniques. It also includes twenty-three third level subroutines for call by each processing technique. The program is well commented to permit ease in modification and future usage.

V. RESULTS

An analysis of several speech in noise enhancement methods has been

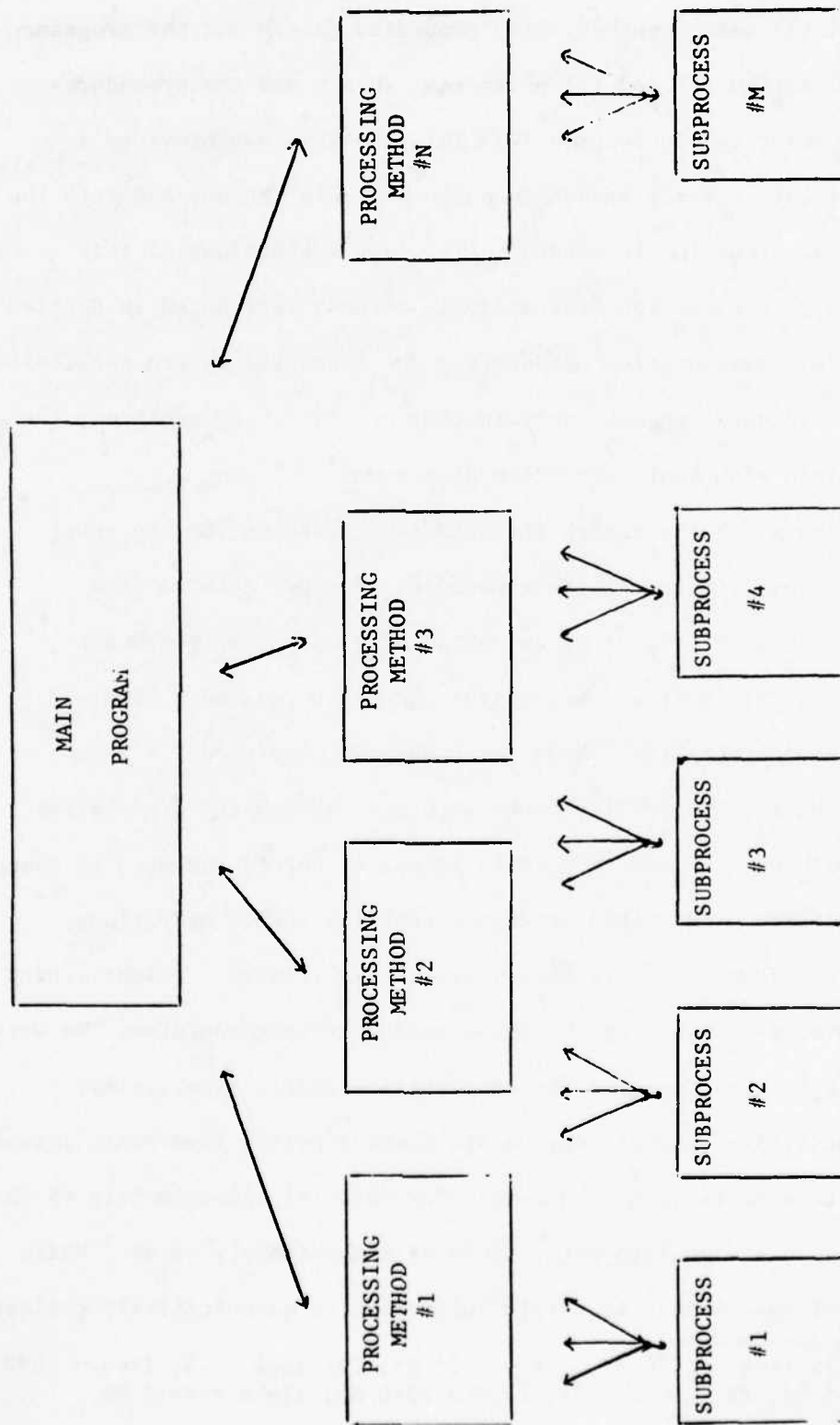


FIGURE 2. An overview of the real speech processing program "PROCESS." N is the number of processing techniques implemented by the system (presently twelve), M is the number of subprocessing steps implemented by the system (presently twenty-four).

carried out: (1) using synthetically generated speech and the programs discussed in Section III and (2) using real speech and the procedures and programs described in Section IV. This analysis has provided a basic insight into several speech in noise analysis methods and into the speech in noise situation in general. The four subsections of this section describe each of the four analysis methods introduced in Section II and contain representative output from the synthetic speech generation (SPEECH) and synthetic speech analysis (M2) programs. In addition, the results obtained with real speech are discussed.

For the synthetic speech illustrations, results for the vowel /a/ have been selected for all four methods. Formant data is from Peterson and Barney (22)*. A pitch period of 9 ms was selected and signal-to-noise ratios of ∞ (no noise) and 0dB were used. Figure 3 illustrates the output from SPEECH for these two conditions.

For the intelligibility tests with real speech two utterances are used. Both were spoken by Captain Robert A. Curtis and each is about 12 sec in duration. The first utterance contains speech as follows: "Testing...one...two...three...four...five...six...seven... eight...nine... ten... we were away a year ago." The second utterance contains, "We were away a year ago...testing...one...two...three...four...five...six... seven...eight...nine...ten... may we all learn a yellow lion roar...Hawaii." The first utterance is in a signal-to-noise ratio of approximately +6 dB; the second is in a signal-to-noise ratio of approximately -6 dB. While this corpus of speech data is hardly sufficient to quantitatively evaluate

*F1: ampl.= 1, freq. = 730 Hz, alpha = 27 Hz; F2: ampl.= .5, freq.= 1090 Hz, alpha = 28.5 Hz; F3: ampl.= .04, freq.= 2240 Hz, alpha = 46.5 Hz

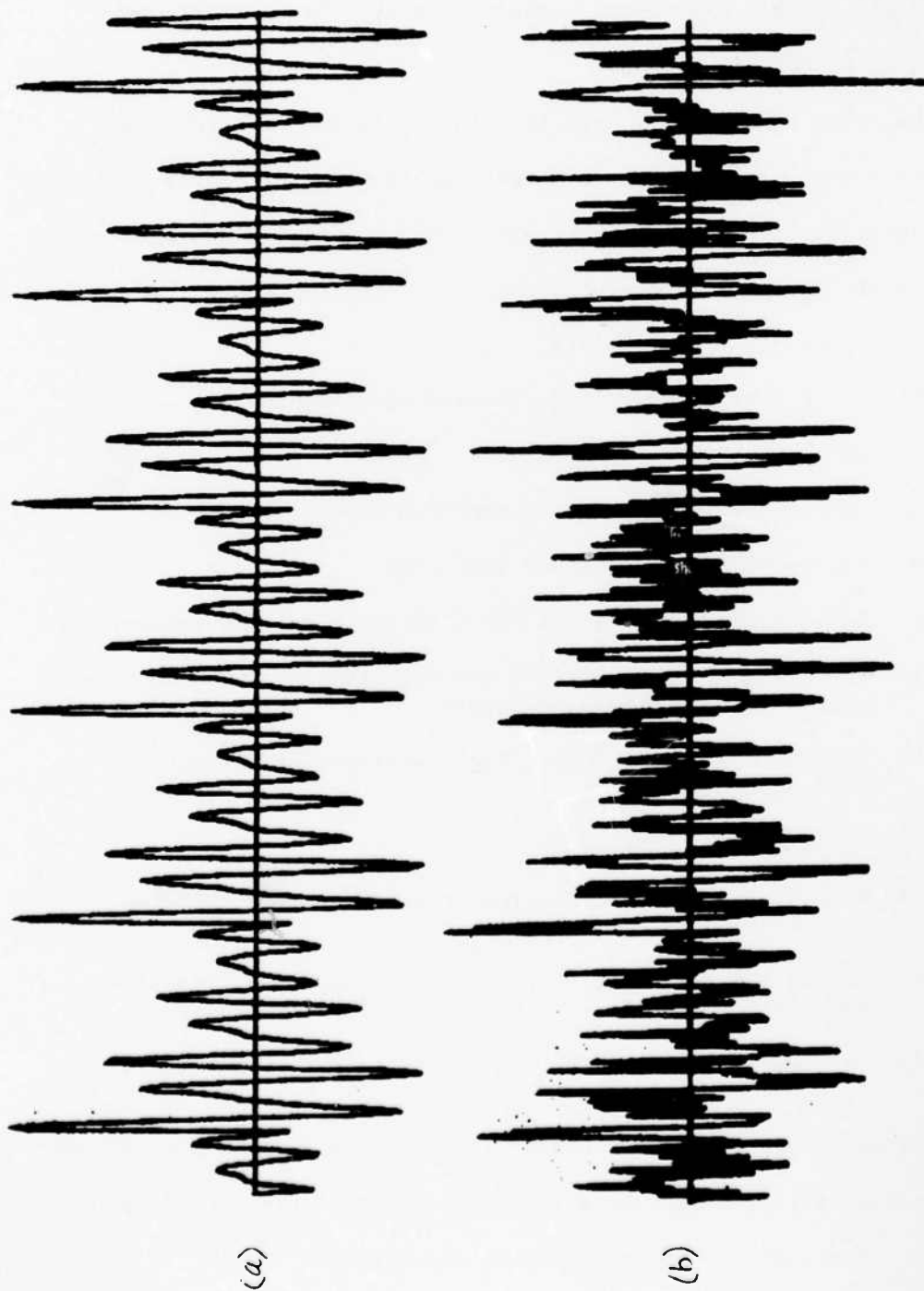


FIGURE 3. Time waveform of the synthetic vowel /a/ as generated by "SPEECH"; (a) no noise, (b) signal-to-noise ratio of 0dB.

speech intelligibility, it is sufficient to make a preliminary study of the processing methods and to make several qualitative observations.

A. INTEL PROCESSING METHOD

The INTEL algorithm is described in two recent reports (1-2). This author's interpretation of the algorithm from these reports indicates that the analysis of a single time-frame of 512 points (51.2 ms) consists of the following sequence of steps:

- (a) Input 512 time samples;
- (b) Apply a triangular window to the time samples;
- (c) Perform a 512 point FFT;
- (d) Set to zero the magnitude spectrum above 2.5 KHz;
- (e) Square root the magnitude spectrum;
- (f) Reverse the signs of all odd numbered magnitude harmonics;
- (g) Perform a 512 point FFT on the magnitude spectrum as a real signal with zero imaginary part;
- (h) Set the magnitude of the five low frequency harmonics to zero (gating);
- (i) Perform a 512 point IFFT;
- (j) Reverse the signs of the real part of all odd numbered harmonics;
- (k) Square the real part of the spectrum, make it a magnitude, and restore the phase from the original time waveform; and
- (l) Perform a 512 point IFFT to result in the output time signal.

The input time waveform and the result after each processing step listed in the previous paragraph are illustrated in Figure 4 for: (1) no noise (left) and (2) a signal-to-noise ratio of 0dB (right).

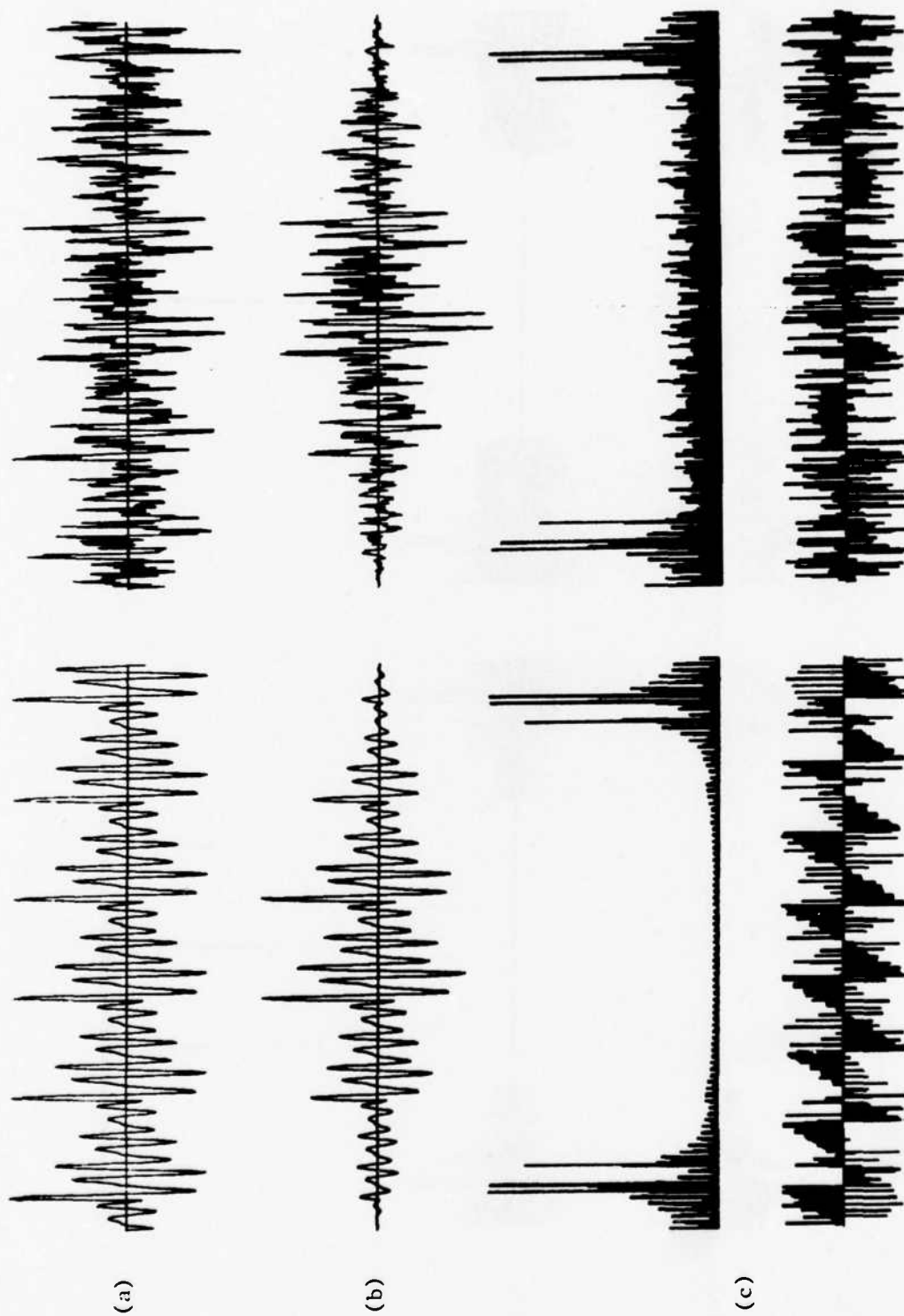


FIGURE 4. The Intel process applied to the synthetically generated vowel /a/. Each waveform shown is after the corresponding step listed in the first paragraph of Section V-A; left is for a $S/N = \infty$, right is for a $S/N = 0\text{dB}$.

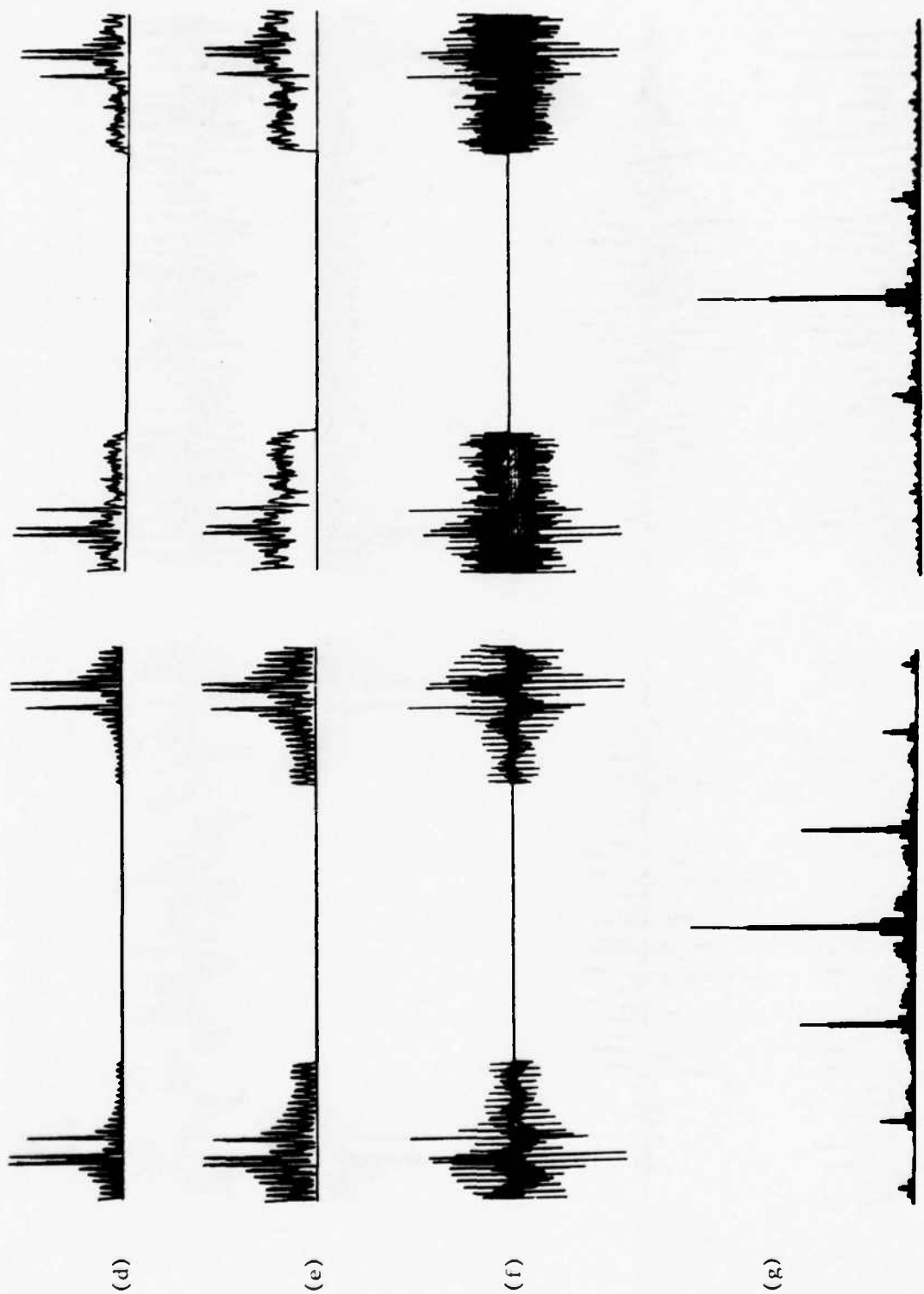


FIGURE 4. Continued



(h)



(i)



(j)

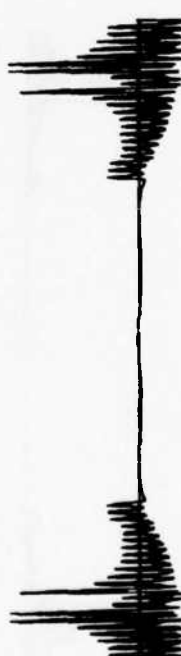
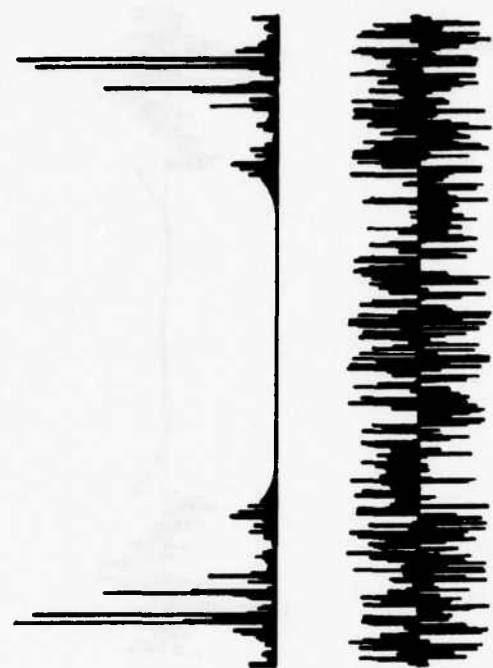


FIGURE 4. Continued



(k)



(l)

FIGURE 4. Continued

1. Discussion

A discussion of the overall INTEL speech processing method can be found in two recent reports (1-2). The discussion here assumes general familiarity with the INTEL method.

From Figures 4a - 4c the effect of noise on both the time waveform as well as on the magnitude and phase of the frequency spectrum can be observed. From Figure 4c, it is noticed that the effect of noise on the frequency spectrum is to add a randomized baseline shift to the magnitude of the spectrum and a seeming randomization of the phase characteristic. The phase of the spectrum of the speech plus noise signal is apparently not without information, however, since greater intelligibility has been achieved with INTEL by restoring the input phase rather than discarding it in producing the processed output (20).

In step e of the INTEL process a square root of the magnitude spectrum is taken. In a recent report (2), Weiss and Aschkenesy go to great lengths to demonstrate that a justification for this square root operation centers around the fact that such an operation causes an increased fraction of the noise energy (relative to the speech energy) to be concentrated in the region near the origin in the second order spectrum (Figure 4g) than would otherwise be present. It is therefore argued (2) that such a square root operation makes it possible to more completely remove the noise. What is unclear to the present author is whether the "extra noise" energy found near the origin in the second order spectrum is due to the original noise being moved or, more simply, the first order spectrum flattening which is caused by the square root operation. Following this latter argument, it is then not a desirable

feature of square rooting which causes increased energy near the origin of the second order spectrum, but rather a consequence of square rooting.

Steps f and j of the INTEL process both involve reversing the signs of all odd numbered harmonics. The only effect of this pair of sign reversals is to reverse the scale of the axis of the second-order spectrum. A computational advantage has been suggested as a justification in a recent report (2).

The essence of the INTEL technique is observed in Figure 4g which shows the second-order spectrum for the synthetic vowel /a/ in signal-to-noise ratios of ∞ and 0dB. Note the large peak at low "frequency"* in the second order spectrum for the signal plus noise. As shown in Figure 4h, this peak is removed by "gating" in the INTEL process.

Figure 4i illustrates the first-order spectrum after "gating" and Figure 4j after the sign reversals are "reversed". Comparing Figure 4j with Figure 4e (the spectrum of the signal before gating), the effect of the gating operation upon the spectrum is observed. In the absence of noise, the effect appears to be primarily a shift in the amplitude of all magnitude harmonics. In the presence of noise, the effect appears

*The use of the term "frequency" in describing the second-order spectrum may be confusing. While it is true that the units of the horizontal axis for the second-order spectrum are not Hz when carried from the original time waveform; the units are Hz if we consider the first-order spectral signal like a time waveform as seems to be a convenient way for viewing the INTEL Technique. In any case, a clear distinction of which spectrum is being discussed when using the term "frequency" will be made.

to be a removal of the baseline (Figure 4e) by a downward shift plus an added distortion in the frequency range above 2.5 KHz. A more careful observation of Figure 4j, however, reveals a decrease in spectral energy at about 1.5 KHz and a marked increase above about 2 KHz. This distortion was observed in a recent report (2) and can be theoretically shown to be a consequence of the gating operation in the second order spectrum. A method for automatically compensating for this distortion is described in a recent report (2). Time did not permit the addition of this compensation algorithm to the INTEL procedure programmed in the present work.

In step k of the INTEL procedure, the spectrum of the signal is squared prior to retransformation back into the time-domain. The justification for this step is an attempt to compensate for the square-root operation in step e(2). This author wonders why it is thought necessary to compensate for the square-root operation. There is little reason to suspect that attempts to preserve the original frequency spectrum are necessarily beneficial, or even desirable. The now classic work of Licklider and Pollack (27), as well as work of others, clearly shows that speech can withstand severe frequency as well as amplitude distortion without a significant loss of intelligibility.

To test the need for the square operation, a comparison of speech processed by the INTEL procedure, both with and without the square operation were examined by listening. Speech processed without squaring sounded very much like the unprocessed original. This indicates that apparently the square operation is essential to the INTEL process.

This surprising result (surprising to this author anyway) leaves open several unanswered questions regarding the use of the square root and square operations in the INTEL process.

One question remaining is why square rooting (and subsequent squaring) has been observed to improve the intelligibility of speech in noise relative to not square rooting. One argument is because the result of these two operations is to modify the output frequency spectrum in such a way as to emphasize lower frequency components relative to higher frequency components. Since lower frequency components tend to best survive the noise, speech square rooted (followed by squaring) would be expected to sound better (enhanced listenability) in noise. Whether this process results in enhanced intelligibility remains to be demonstrated.

The argument of the previous paragraph would suggest that the root used in the rooting operation (step e) should be dependent upon the signal-to-noise ratio. This is, in fact, what has been observed with the INTEL Process (2, 20), root factors of one-third and one-half being found best dependent upon the signal-to-noise ratio.

One final observation about the INTEL technique can be made from Figure 4e. From this figure it is observed that the INTEL process does not preserve the triangular time weighting applied in step b. This may or may not result in any significant distortion depending upon how successive triangularly weighted windows sum. This particular problem has been previously observed and some work at compensating for this distortion has been pursued (20).

2. Tests with Real Speech

From listening to the two utterances (described at the beginning of this section) processed by INTEL, the advantages of processing speech in noise (by INTEL) are apparent. Clearly, speech processed by the INTEL technique "sounds better." This has been described as enhanced "listenability." While there is some question as to whether there is an enhanced intelligibility through the use of the INTEL technique, there seems no question as to a perceived improvement in the signal-to-noise ratio.

A recent communication (28) in evaluating the INTEL technique in signal-to-noise ratios of -5 dB, 0 dB, +5 dB, and no-noise, found little intelligibility gain for INTEL processed versus unprocessed speech.

3. An INTEL Extension

Figure 4g, the second-order spectrum for speech in no noise and at a signal-to-noise ratio of 0 dB, clearly reveals an increased amplitude near the origin which results from the additive noise. In the INTEL technique described with Figure 4 and Section V-A-1, a suppression of this second-order spectral peak to zero was implemented as described in two recent reports (1,2). Looking at Figure 4g (no noise), however, reveals that even for normal speech this peak is non-zero. In fact, from an observation of several synthetically generated speech sounds without noise, it appears that the peak in the second-order spectrum near the origin is usually approximately twice the amplitude of the peak due to the formant frequency envelope.

Pursuing this observation, an experiment was carried out in which the INTEL technique (step h) was changed to suppress this low frequency peak (and other low frequency harmonics) by a factor which

causes the zero frequency peak to be twice the amplitude of the formant peak in the processed output. This is accomplished by calculating a factor in the second-order spectrum given by:

$$x = \frac{\text{amplitude of the zero frequency component}}{\text{amplitude of the maximum formant envelope component}}$$

and dividing each of the ten low frequency, second-order harmonics by half this factor.

Upon listening to the result in comparison with INTEL a small improvement in intelligibility, listenability, and naturalness seemed apparent.

The use of this modification to the INTEL technique has advantages other than (perhaps) enhanced intelligibility. First, it should result in less distortion of the type shown in Figure 4j (and described earlier) since the "gate" is less severe. (This fact was not verified with synthetic speech.) Second, it results in an overall system automatically compensated for a changing signal-to-noise ratio.

A method of suppressing the zero frequency peak in the second-order spectrum by fixed factors has been previously investigated with INTEL (20). Results have indicated that a suppression of the peak by about one-third worked best at a signal-to-noise ratio of 0 dB. From Figure 4g, it can be noted that this is in very close agreement with causing the zero-frequency peak to be twice the formant peak as described in the previous paragraph.

B. SPECTRAL SUBTRACTION METHOD

As previously discussed, one problem with the INTEL technique is that it requires four Fourier Transformations. A process which is intuitively appealing, appears similar in function to "gating" in the

INTEL technique, and requires only two Fourier Transformations is simple spectral subtraction.

The basic method implemented for spectral subtraction consists of the following sequence of five steps:

- (a) Input 512 time samples;
- (b) Apply a triangular window to the time samples;
- (c) Perform a 512 point FFT;
- (d) Estimate the average noise level from the magnitude of the spectrum above 2.5 KHz, subtract this level from the magnitude spectrum and zero all frequency components above 3 KHz; and
- (e) Perform a 512 point IFFT to result in the output time signal.

The input time waveform and the result from steps b and c above are the same as for the INTEL technique and are illustrated in Figure 4. The results for steps d and e are illustrated in Figure 5 for: (1) no noise (left) and (2) a signal-to-noise ratio of 0dB (right).

1. Discussion

One problem which results when subtracting from the magnitude spectrum involves the action to be taken when a difference results in a magnitude of less than zero. The method employed in this work for the subtraction of a level from the magnitude of the spectrum (step d previous paragraph) is slightly more complex than simple subtraction. If the value of the magnitude of a component is greater than the noise level, the noise level is subtracted from the magnitude value. If the value of the magnitude of a component is less than or equal to the noise level, that magnitude is divided by two.

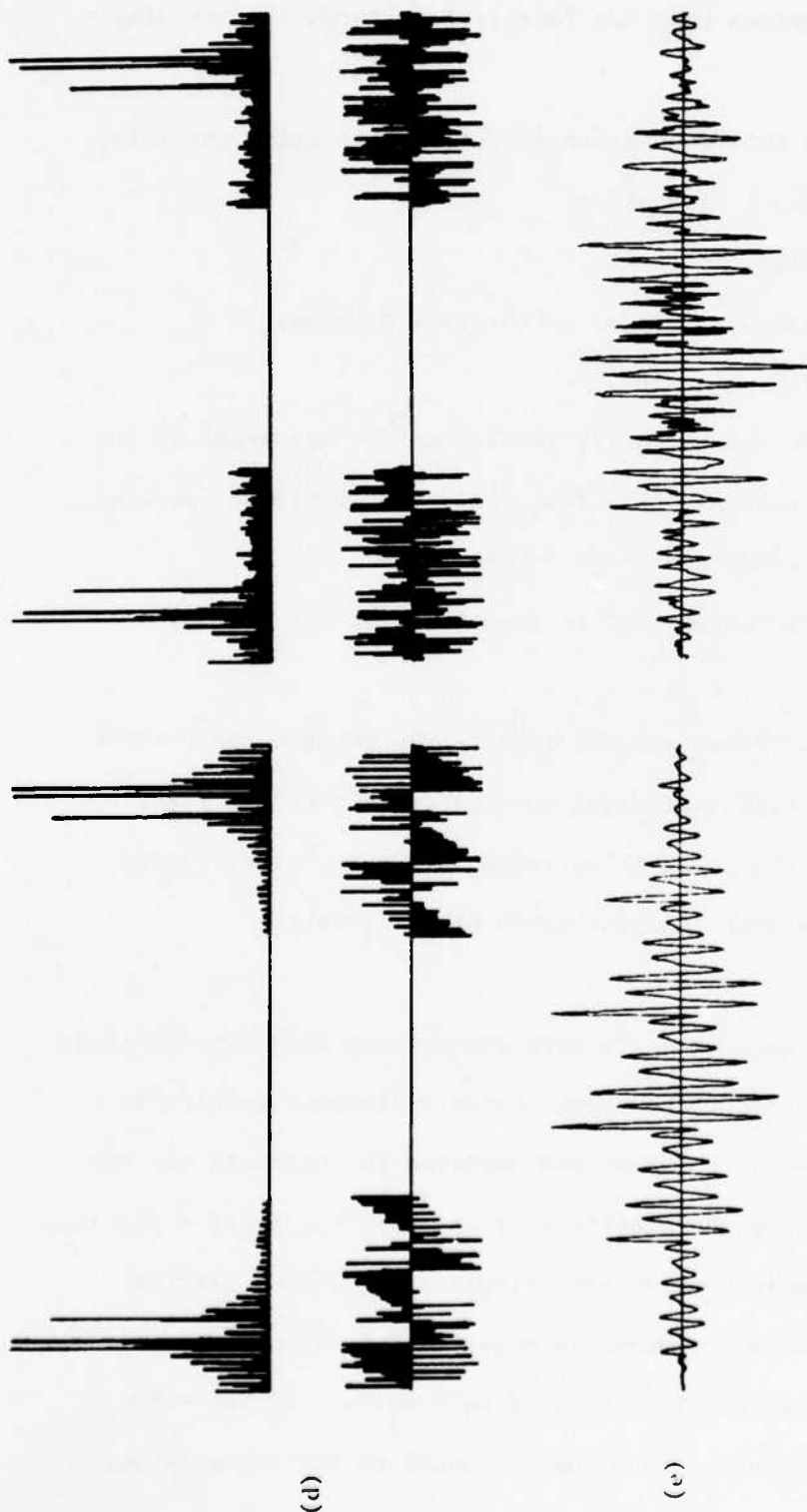


FIGURE 5. The spectral subtraction process applied to the synthetically generated vowel /a/. Steps a - c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-B; left is for a $S/N = \infty$, right is for a $S/N = 0$ dB.

From Figure 5d, it appears that the simple method employed to estimate and subtract the noise is not totally effective. Comparing Figure 5d with Figure 4k, the output spectrum from INTEL, reveals that INTEL appears to do a significantly better job at reducing noise than the subtractive method. Examining the time waveforms for the output from the spectral subtraction method (Figure 5e) and INTEL (Figure 4e) reveals that the spectral subtraction technique does not modify the time waveform as severely in the absence of noise. Probably a greater amount of subtraction (greater noise suppression) would be beneficial.

2. Tests With Real Speech

The results obtained for the processing of the real speech utterances described in the first paragraph of Section V confirm the expectations of the previous paragraph. The signal-to-noise ratio does not sound particularly enhanced over that obtained by simply zeroing the frequency range above 3 KHz and the intelligibility does not appear improved.

As a second experiment with subtractive noise cancellation, an additional processing step was added between steps d and e of the subtractive method. This additional process involves the use of a filter to emphasize the second formant frequency range. The characteristic of this filter was chosen to have a rising slope of 18 dB/octave below 1 KHz, a passband from 1 KHz to 2 KHz, and a falling slope of 12 dB/octave above 2 KHz. At a signal-to-noise ratio of -6 dB, the speech with high frequency emphasis sounds less intelligible than that with subtraction alone. This is probably due to the fact that enhancing the second formant frequency range is detrimental since (at this low signal-to-

noise ratio) the second formant range is so heavily obscured by noise. At a signal-to-noise ratio of +6 dB it is not clear whether the high frequency emphasis is helpful or not. Certainly, such an emphasis causes the speech to be less natural sounding in both cases.

C. MINIMUM MEAN-SQUARE-ERROR FILTERING

As indicated in Section II of this report, the method of minimum mean square error filtering is an attractive technique for processing speech in noise for two main reasons. First, it is an optimum method (in the least mean square error sense) for filtering a signal in noise; and second, it can be implemented in a computationally efficient manner (relative to other techniques).

The implementation used for this method is based upon an analysis from Papoulis (21) as described in Section II of this report. The implementation consists of the following sequence of five steps:

- (a). Input 512 time samples;
- (b). Apply a triangular window to the samples;
- (c). Perform a 512 point FFT;
- (d). Estimate the expected noise level from the magnitude of the spectrum above 2.5 KHz, estimate the expected signal from the long-time average for normal speech, and modify the magnitude spectrum using the filter $H(j\omega)$ as given by Eq. 1 (Section II); and
- (e). Perform a 512 point IFFT to result in the output time signal.

The input time waveform and the results from steps b and c above are the same as those illustrated in Figure 4. The results for steps d and e are illustrated in Figure 6 for: (a) no noise (left) and

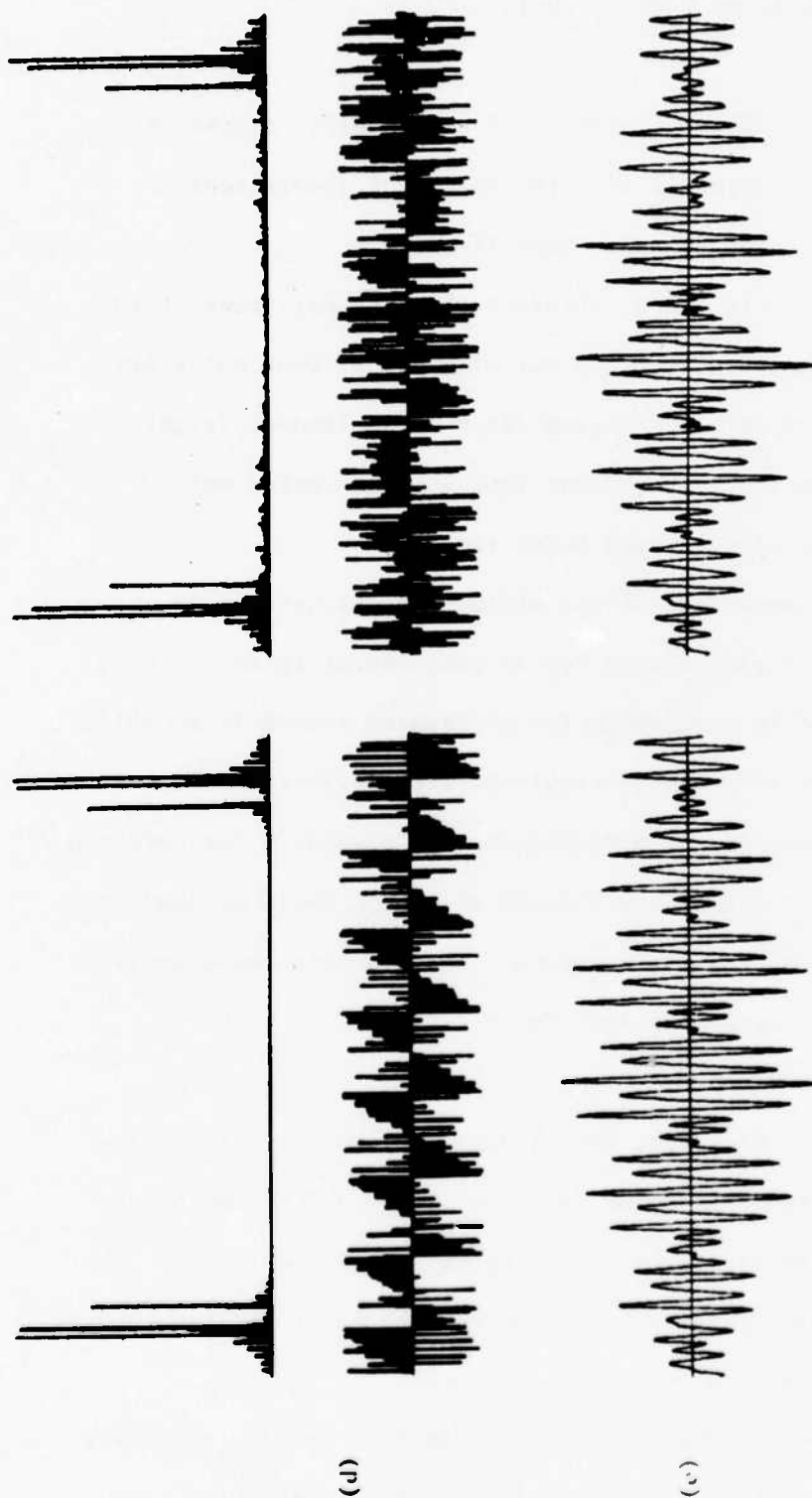


FIGURE 6. The minimum mean square error filtering process applied to the synthetically generated vowel /a/. Steps a - c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-C; left is for a $S/N = \infty$, right is for a $S/N = 0$ dB.

(2) a signal-to-noise ratio of 0 dB (right).

1. Discussion

A discussion of the minimum mean square error filtering process is contained in Section II of this report. A theoretical justification of Eq. 1 can be found in Papoulis (21).

From Figure 6 it can be observed that the magnitude of the frequency spectrum of the output (Figure 6d) with and without noise are surprisingly similar. The output spectrum after noise removal (right) appears very free from noise. The output time waveform after noise removal (Figure 6e) also appears very noise free.

One additional advantage of the minimum mean square error filtering method of processing speech for an enhancement in noise is that the method is directly extendible for processing speech in nonwhite noise environments. The only change required is a modification of $S_{nn}(w)$ in Eq. 1. In addition, if a technique were developed for detecting the presence of speech in noise, the absence of speech could be used to continually update a running noise estimate. This running noise estimate could then be readily incorporated into Eq. 1.

2. Tests With Real Speech

Like Figure 6 displays, the results achieved for processing real speech in noise (signal-to-noise ratios of +6 dB and -6 dB) by the method of minimum mean square error filtering are very encouraging. The signal-to-noise ratio is significantly enhanced, the naturalness unchanged, and the intelligibility sounds improved.

Two experiments were carried out with real speech, one using the procedure outlined in the second paragraph of this subsection (and

illustrated in Figure 6), the other with the addition of a zeroing of all magnitude components above 2.5 KHz between steps d and e. In both cases the results were very good with little observable difference between them. An examination of Figure 6d indicates that there is very little energy above 2.5 KHz such that little change would be expected.

For many reasons, the method of minimum mean square error filtering appears to offer a great potential for speech in noise intelligibility enhancement.

D. METHODS BASED UPON PITCH TRACKING

As described in Section II of this report, methods of enhancing the intelligibility of speech in noise based upon pitch analysis are intuitively attractive. However, as also indicated in Section II, such methods have, in general, produced discouraging results (20).

In order to experiment with methods based upon pitch tracking, a technique was implemented which consists of the following sequence of steps:

- (a) Input 512 time samples;
- (b) Apply a triangular window to the samples;
- (c) Perform a 512 point FFT;
- (d) Determine the pitch frequency using a method to be described and zero all magnitude components between pitch harmonics; and
- (e) Perform a 512 point IFFT to result in the output time signal.

The input time waveform and the results from steps b and c above are the same as for the INTEL technique and are illustrated in Figure 4. The results after steps d and e are illustrated in Figure 7

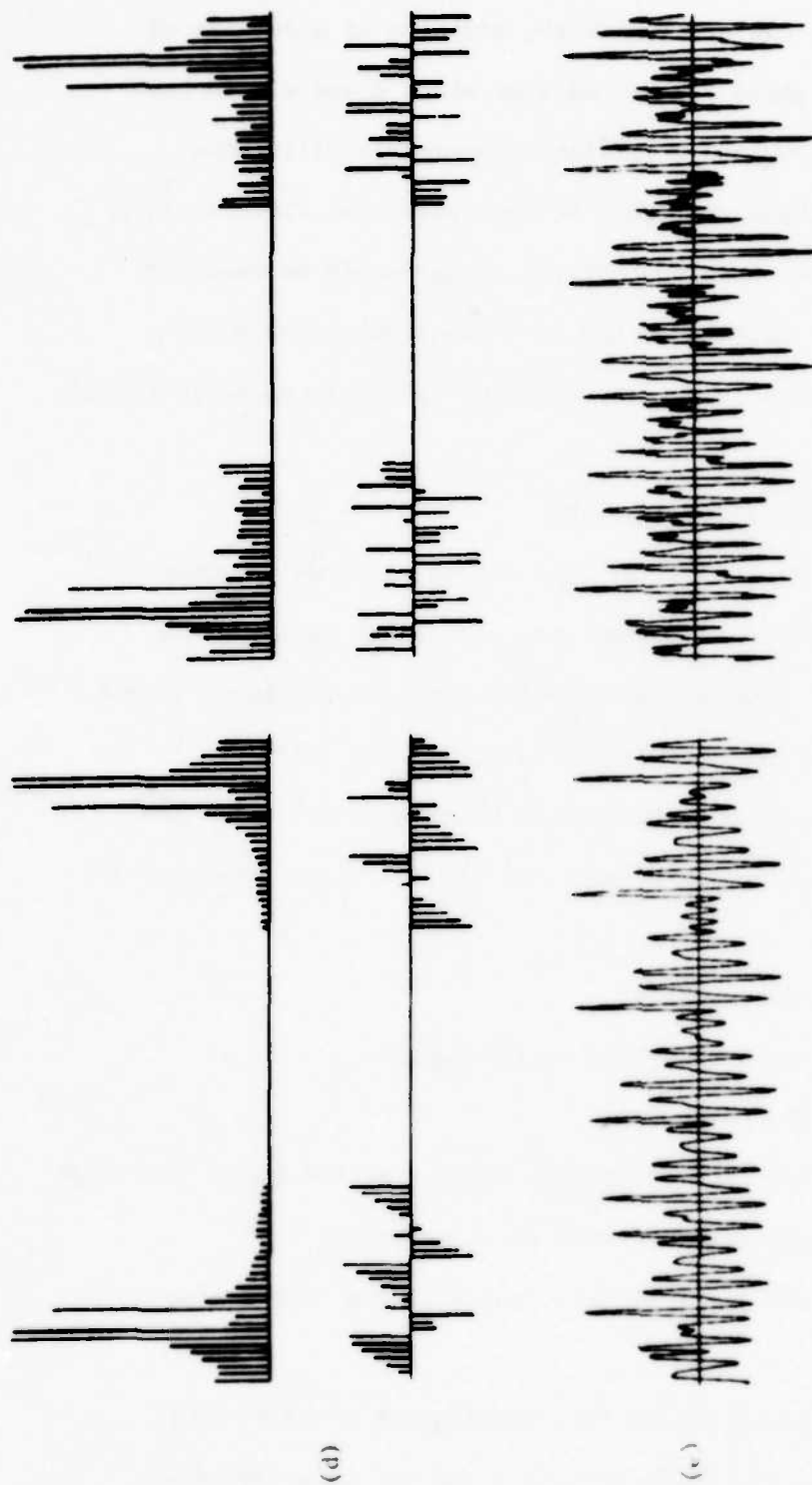


FIGURE 7. The process based upon pitch extraction applied to the synthetically generated vowel /a/. Steps a - c are the same as Figure 4. Steps d and e are illustrated after the corresponding step listed in the second paragraph of Section V-D; left is for a $S/N = 0$ dB, right is for a $S/N = 0$ dB.

for: (1) no noise (left) and (2) a signal-to-noise ratio of 0 dB (right).

1. Discussion

A discussion of the use of pitch extraction methods for speech intelligibility enhancement in noise is contained in Section II of this report.

The method used for estimating the pitch frequency from the spectrum of speech plus noise consists of determining that integer frequency (F_0) between 80 Hz and 250 Hz which maximizes the following function:

$$\frac{1}{N} \sum_{i=1}^N F(i \cdot F_0) \quad (\text{Eq. 2})$$

where:

N = the greatest integer such that $N \cdot F_0 \leq 3000$ Hz

$F(i \cdot F_0)$ is that line of the magnitude spectrum closest to the frequency $i \cdot F_0$.

The results of Figure 7d display the expected spectrum of the output. With several tests on synthetic voiced sounds, at several signal-to-noise ratios from +6 dB to -6 dB, the pitch frequency was determined quite accurately (to within a few percent) using the method.

The output time waveform (Figure 7e) indicates the appearance of considerable improvement for the enhancement of speech in noise by using this method based upon pitch tracking.

2. Tests With Real Speech

Three experiments based upon pitch extraction and the method outlined earlier in this subsection were performed with real speech. These involve three degrees of suppression of non-pitch magnitude components in step d. These three degrees of suppression are: (1) suppression to zero, (2) suppression by a factor of two, and (3) suppression by a factor of four.

When the non-pitch components are suppressed to zero, a strong, low frequency distortion, probably a result of the time analysis window, is apparent. As the degree of suppression is lessened the strong window distortion diminishes, however, at the expense of increased noise. At a high suppression of non-pitch harmonics (suppression to zero), there is a noticable loss of intelligibility, particularly at a signal-to-noise ratio of -6 dB. This is probably due to the inaccuracy in pitch tracking which is greater at lower signal-to-noise ratios. With a suppression of non-pitch harmonics to other than zero, the method is able to tolerate greater pitch tracking errors without as serious a degradation of speech intelligibility.

To determine some measure of the accuracy of the pitch tracking algorithm applied to real speech, two simple experiments were performed. First, a listing was created of the pitch values determined for the utterance at a signal-to-noise ratio of +6 dB. While no standard of comparison for the determined pitch values was available, the printed values seemed reasonable for a male speaker. The printed values were generally fairly continuous with fundamental frequency values between 110 Hz and 120 Hz during voiced speech intervals.

As a second experiment, a constant frequency of 10 Hz was subtracted from each pitch measurement and the result was used to generate the output waveform in the usual manner. Upon listening to the result it was found unintelligible.

Overall, the intelligibility results using this pitch tracking method sound of lesser intelligibility than either the INTEL or the minimum mean square error filtering methods. There are, however, a number of improvements which could be added to the basic method and which might substantially improve this first attempt. It is this author's feeling that methods based upon pitch tracking offer the potential to result in an enhancement of speech intelligibility in noise and that such methods should not be overlooked because of past discouraging results.

VI CONCLUSIONS

This study has explored several methods for the enhancement of speech intelligibility in the presence of wideband random noise at the speaker. This exploration has involved a study of the effect of the methods upon synthetically generated voiced sounds in noise as well as on real speech in white noise at two signal-to-noise ratios.

Four basic methods have been investigated: (a) INTEL, (b) spectral subtraction, (c) minimum mean square error filtering, and (d) methods based upon pitch tracking. Several variations in each basic method have been tested and numerous experiments with speech in noise have been performed. The experiments with synthetic speech have provided a substantial insight into not only the four methods, but also the speech in noise situation in general. Through the experiments with real speech, some qualitative results for the speech processing methods have been presented.

It appears from the experiments with synthetic and real speech that all four methods have the potential to result in an enhancement of the intelligibility of speech in noise. While the qualitative intelligibility results obtained during this work have indicated that, of the methods tested, the INTEL method and the method based upon minimum mean square error filtering seem to result in the greatest enhancement of speech in noise, the other methods should not be discarded. There are several reasons to suspect that substantial improvement in the intelligibility of speech in noise can be obtained through the use of each of the four basic techniques.

REFERENCES

1. M.R. Weiss, E. Aschkenasy, and T. W. Parsons, "Study and development of the INTEL Technique for Improving Speech Intelligibility," Technical Report No. RADC-TR-75-108, Rome Air Development Center, Griffiss Air Force Base, New York, April 1975. (AD# A011709)
2. M. R. Weiss and E. Aschkenasy, "Automatic Detection and Enhancement of Speech Signals," Technical Report No. RADC-TR-75-77, Rome Air Development Center, Griffiss Air Force Base, New York, March 1975. (AD# B004521L)
3. T.W. Parsons and M.R. Weiss, "Enhancing/Intelligibility of Speech in Noisy or Multi-talker Environments," Technical Report No. RADC-TR-75-155, Rome Air Development Center, Griffiss Air Force Base, New York, June 1975. (AD# A013767)
4. I.B. Thomas and A. Ravindran, "Intelligibility Enhancement of Already Noisy Speech Signals," J. Aud. Eng. Soc., 22, 234-236, 1974.
5. M.R. Sambur and N.S. Jayant, "LPC Analysis/Synthesis from Speech Inputs Containing Quantizing Noise or Additive White Noise," IEEE Trans. Acoustics, Speech, and Signal Processing, ASSP-24, 488-494, 1976.
6. S.F. Boll, "Improving Linear Prediction Analysis of Noisy Speech by Predictive Noise Cancellation," 1977 IEEE International Conf. on Acoustics, Speech, and Signal Processing, 10-12, 1977.
7. R.J. Niederjohn, "A Comparison of Three Recently Reported Methods of Processing Speech for the Enhancement of Speech Intelligibility in High Noise Levels," Proceedings of the Midwest Symposium on Circuits and System, 325-331, 1976.
8. R.J. Niederjohn and J. H. Grotelueschen, "The Enhancement of Speech Intelligibility in High Noise Levels by High-pass Filtering Followed by Rapid Amplitude Compression," IEEE Trans. Acoust., Speech, and Sig. Proc., ASSP-24, 277-282, 1976.
9. R.J. Niederjohn and J. H. Grotelueschen, "Speech Intelligibility Enhancement in a Power Generating Noise Environment," IEEE Trans. Acoustics, Speech, and Signal Processing, accepted for publication.
10. I.B. Thomas and R.J. Niederjohn, "The Intelligibility of Filtered Clipped Speech in Noise," J. Aud. Eng. Soc., 18, 299-303, 1970.
11. I.B. Thomas and W.J. Ohley, "Intelligibility Enhancement Through Spectral Weighting," 1972 Conference on Speech Communication and Processing, 360-363, 1972.
12. J.P. Egan and F.M. Wiener, "On the Intelligibility of Bands of Speech in Noise," J. Acoust. Soc. Am., 18, 435-441, 1946.

13. I. Pollack and J.M. Pickett, "Masking of Speech by Noise at High Sound Levels," J. Acoust. Soc. Am., 30, 127-130, 1958.
14. I. Pollack and J.M. Pickett, "Intelligibility of Peak-Clipped Speech at High Noise Levels," J. Acoust. Soc., Am., 31, 14-16, 1959.
15. J.C.R. Licklider, "Effects of Amplitude Distortion on the Intelligibility of Speech," J. Acoust. Soc., Am., 18, 429-434, 1946.
16. G.A. Miller and S. Mitchell, "Effects of Distortion on the Intelligibility of Speech at High Altitudes," J. Acoust. Soc. Am., 19, 120-125, 1950.
17. W. Wathen-Dunn and D.W. Lipke, "On the Power Gained by Clipping Speech in the Audio Band," J. Acoust. Soc. Am., 30, 36-40, 1958
18. E.A. Kretsinger and N.B. Young, "The Use of Fast Limiting to Improve the Intelligibility of Speech in Noise," Speech Monogr., 27, 63-69, 1960.
19. H. Drucker, "Speech Processing in a High Ambient Noise Environment," IEEE Trans. Audio and Electroacoust., AU-16, 165-168, 1968.
20. Personal communication with Dr. Bruno Beek and Captain Robert Curtis, Rome Air Development Center, Griffiss Air Force Base, New York.
21. A. Papoulis, "Probability, Random Variables, and Stochastic Processes," McGraw-Hill Book Company, New York, 1965, pp 403-406.
22. G.E. Peterson and H.L. Barney, "Control Methods Used in a Study of Vowels," J. Acoust. Soc. Am. 24, 175-184. 1952.
23. N.R. French and J.C. Steinberg, "Factors Governing the Intelligibility of Speech Sounds," J. Acoust. Soc. Am., 19, 90-119, 1947.
24. H.K. Dunn and S.D. White, "Statistical Measurements on Conversational Speech," J. Acoust. Soc. Am., 11, 278-288, 1940.
25. I.B. Thomas, "The Second Formant and Speech Intelligibility," Proc. Nat. Elect. Conf., 23, 544-548, 1967.
26. I.B. Thomas, "The Influence of the First and Second Formants on the Intelligibility of Clipped Speech,," J. Audio Eng. Soc., 16, 182-185, 1968.
27. J.C.R. Licklider and I. Pollack, "Effects of Differentiation, Integration, and Infinite Peak Clipping Upon the Intelligibility of Speech," J. Acoust. Soc. Am., 20, 42-51, 1948.
28. Personal Communication from J.S. Lim, A. V. Oppenheim and L.D. Braida.

APPENDIX A

Source listing of the program "SPEECH", synthetic voiced speech generation program.

*LIST SPEECH

```

0010**#RUN#DATA"01"
0020  DIMENSION A(1024),VAR(3,3),W(3),AL(3)
0030  PI=3.1416
0040  PRINT,"INPUT: AMPL,FREQ,ALPHA, FOR ALL 3 FORMANTS"
0050  DO 20 I=1,3
0060  READ,(VAR(I,J),J=1,3)
0070  DO 30 J=1,3
0080  W(I)=2.*PI*VAR(I,2)
0090  AL(I)=2.*PI*VAR(I,3)
0100  DO 40 I=1,1024
0110  A(I)=0.
0120  T=I
0130  DO 40 J=1,3
0140  A(I)=A(I)+VAR(J,1)*EXP(-AL(J)*T/1.E4)*SIN(W(J)*T/1.E4)
0150  PRINT,"INPUT PITCH PERIOD IN "SEC"
0160  READ,F0
0170  ISHIFT=F0*10.
0180  DO 60 I=1,ISHIFT
0190  DO 60 II=I,1024,ISHIFT
0200  A(I)=A(I)+A(II)
0210  DO 65 I=ISHIFT+1,1024
0220  A(I)=A(I-ISHIFT)
0230  DO 70 I=1,512
0240  A(I)=A(I+512)
0250  PRINT,"NOISE?(1=YES, 0=NO)"
0260  READ,K
0270  IF(K)210,85,210
0280  PRINT,"INPUT S(RMS)/N(P'S) IN DB"
0290  READ,SN
0300  SPAMPPL=0.
0310  DO 220 I=1,512

```

```

0320 220 SPAMPL=SPAMPL+A(I)**2
0330 SPAMPL=SQRT(SPAMPL/512.)
0340 XNAMPL=SPAMPL/10.**(SN/20.)
0350 DO 230 I=1,512
0360 230 A(I)=A(I)+XNAMPL*XNORM(I.)
0370 35 PRINT,"VIEW ANY SAMPLES?(1=YES, 0=NO)"
0380 READ,K
0390 IF(K)90,110,90
0400 40 WRITE(6,91)
0410 91 FORMAT('SAMPLE NOS: FIRST, LAST')
0420 READ,K,L
0430 DO 100 I=K,L
0440 100 WRITE(6,92)I,A(I)
0450 99 FORMAT(1X,14,4X,E17.10)
0460 GO TO 95
0470 110 CALL RANSIZ(1,2,1)
0480 N=512;AA=0.
0490 WRITE(1)N,AA
0500 B=0.
0510 DO 120 J=2,513
0520 120 WRITE(1,J)A(J),B
0530 CALL DETACH(01,ISTAT,)
0540 STOP
0550 END
0560 FUNCTION XNORM(X)
0570 X1=Rand(1.)
0580 X2=Rand(1.)
0590 Y=SQRT(-2.*ALOG(X1))*(COS(6.2831853*X2))
0600 XNORM=Y
0610 RETURN
0620 END

```

ready

APPENDIX B

Source listing of the program "M2", synthetic speech processor.
Only additions to the processor of this program "M1" (created by Mr.
David Clark of RADC) are listed.

```

:
:
60 GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400,1500),M

```

```

:
:

```

```

2350-----CALL TO INTEL PROCESSOR ON "11"
240 1100 CALL INTEL;GO TO 10
0240C-----CALL TO RJN PROCESSOR ON "12"
0250 1200 CALL RJN(IMAX);GO TO 10
0251C-----CALL TO SPECTRAL SUBTRACTION PROCESSOR
0252 1300 CALL SSP(XNOISE);GO TO 10

```

```

:
:

```

```

0590C-----INTEL PROCESSING ALGORITHM(PROG. BY R. NIEDERJOHN)
0600 SUBROUTINE INTEL
0610 COMMON APRAY(63),DT,N,F(512,2),G(512,2)
0615 DIMENSION A(512,2)
0620 PRINT:"ENTER LAST STEP TO COMPLETE"
0630 READ*LAST
0631 PRINT:"ENTER STARTING POINT NUMBER"
0632 READ*NSTART;IF(NSTART.EQ.0)GO TO 1
0633 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16),NSTART

```

```

0340C-----TRIANGULAR TIME WINDOW(1)
0350 1 DO 19 J=1,256
0360 XJ=J
0370 F(J,1)=F(J,1)*XJ/256.
0380 19 F(J+256,1)=F(J+256,1)*(257.-XJ)/256.
0390 IF(LAST.EQ.1)RETURN
0700C-----FOURIER TRANSFORM(2)
0710 2 CALL CHAIN("LINK5")
0720 IF(LAST.EQ.2)RETURN
0730C-----SAVE F IN A SO THAT PHASE MAY BE RESTORED LATER(3)
0740 3 DO 20 I=1,512
0750 A(I,1)=F(I,1)
0760 20 A(I,2)=F(I,2)
0770 IF(LAST.EQ.3)RETURN
0780C-----TAKE RE(F) EQUAL TO PRESENT MAGNITUDE(4)
0790 4 DO 30 I=1,512
0800 30 F(I,1)=SQRT(F(I,1)**2+F(I,2)**2)
0810 IF(LAST.EQ.4)RETURN
0820C-----SET I4(F) EQUAL TO ZERO(5)
0830 5 DO 40 I=1,512
0840 40 F(I,2)=0.
0850 IF(LAST.EQ.5)RETURN
0860C-----ZERO UPPER HALF OF ARRAY(6)
0870 6 DO 50 I=129,385
0880 50 F(I,1)=0.
0890 IF(LAST.EQ.6)RETURN
0900C-----TAKE SQRT OF ARRAY(7)
0910 7 DO 60 I=1,512
0920 60 F(I,1)=SQRT(F(I,1))
0930 IF(LAST.EQ.7)RETURN
0940C-----REVERSE SIGN OF ALL ODD ARRAY ELEMENTS(8)
0950 8 DO 70 I=1,511,2
0960 70 F(I,1)=-F(I,1)
0970 IF(LAST.EQ.8)RETURN
0980C-----TAKE SECOND FOURIER TRANSFORM(9)
0990 9 CALL CHAIN("LINK5")
1000 IF(LAST.EQ.9)RETURN
1090C-----SET LOW 5 ELEMENTS EQUAL TO ZERO(10)

```

```

1100 10 DO 100 I=253,261
1110 F(I,1)=0.
1115 100 F(I,2)=0.
1120 IF(LAST.EQ.10)RETURN
1130C-----TAKE INVERSE FOURIER TRANSFORM(11)
1140 11 CALL CHAIN("LINK10")
1150 IF(LAST.EQ.11)RETURN
1160C-----DISCARD IM(F) (12)
1170 12 DO 110 I=1,512
1180 110 F(I,2)=0.
1190 IF(LAST.EQ.12)RETURN
1200C-----REVERSE SIGN OF ALL ODD ELEMENTS(13)
1210 13 DO 130 I=1,511,2
1220 130 F(I,1)=-F(I,1)
1230 IF(LAST.EQ.13)RETURN
1240C-----SQUARE ARRAY(14)
1250 14 DO 140 I=1,512
1260 140 F(I,1)=F(I,1)**2
1270 IF(LAST.EQ.14)RETURN
1300C-----RESTORE PHASE(15)
1340 15 DO 160 I=1,512
1350 F(I,2)=F(I,1)*SIN(ATAN2(A(I,2),A(I,1)))
1360 160 F(I,1)=F(I,1)*COS(ATAN2(A(I,2),A(I,1)))
1370 IF(LAST.EQ.15)RETURN
1380C-----TAKE SECOND INVERSE FOURIER TRANSFORM(16)
1390 16 CALL CHAIN("LINK10")
1400 RETURN
1410 END
2000C-----SPEECH IN NOISE PROCESSING ALGORITHM
2010 SUBROUTINE RJN(IMAX)
2020 COMMON ARRAY(63),DI,N,F(512,2),G(512,2)
2030 DIMENSION A(512,2)
2040 PRINT:"ENTER LAST STEP TO COMPLETE"
2050 READ:LAST
2060 PRINT:"ENTER STARTING POINT NUMBER"
2070 READ:NSTART:IF(NSTART.EQ.0)GO TO 1
2080 GO TO (1,2,3,4,5,6,7,9),NSTART
2090C-----TRIANGULAR TIME WINDOW(1)

```

```

2100 1 DO 19 J=1,256
2110 100000 F(J,1)=F(J,1)*XJ/256.
2130 19 F(J+256,1)=F(J+256,1)*(257.-XJ)/256.
2140 IF(LAST.EQ.1)RETURN
2150C-----FOURIER TRANSFORM(2)
2160 2 CALL CHAIN("LINK5")
2170 IF(LAST.EQ.2)RETURN
2180C-----SAVE F IN A FOR LATER RECONSTRUCTION(3)
2190 3 DO 20 I=1,512
2200 A(I,1)=F(I,1)
2210 20 A(I,2)=F(I,2)
2220 IF(LAST.EQ.3)RETURN
2230C-----MAKE RE(F) EQUAL TO PRESENT MAGNITUDE(4)
2240 4 DO 30 I=1,512
2250 30 F(I,1)=SORT(F(I,1)**2+F(I,2)**2)
2260 IF(LAST.EQ.4)RETURN
2270C-----DETERMINE PITCH(5)
2280 5 SUMMAX=0.
2290 DO 40 I=30,250
2300 XNUM=3000/I
2310 SUM=0.
2320 DO 50 J=1,3000,I
2330 XJ=J
2340 LNUM=XJ/19.53125+.5
2350 LNUM=LNUM+1
2360 50 SUM=SUM+F(LNUM,1)
2370 SUM'=SUM/XNUM
2380 IF(SUM'.LE.SUM'MAX)GO TO 40
2381 IF(IABS(IMAX-I/2).GE.5)GO TO 45
2382 IF(SUM'.LE.1.33*SUM'MAX)GO TO 40
2390 45 SUM'MAX=SUM'
2400 IMAX=I
2410 40 CONTINUE
2420 XIMAX=IMAX
2430 FO=1000./XIMAX
2440 PRINT,"PITCH FREQ.=" ,FO,"MSEC"
2450 IF(LAST.EQ.5)RETURN
2460C-----CONSTRUCT NEW RE(F) *IM(F) KEEPING ONLY HARMONICS AT PITCH FREQ.(6)

```

```

2470 6 DO 60 I=2.512
2480 F(I,1)=0.
2490 F(I,2)=0
2500 DO 70 I=IMAX,3000,IMAX
2510 XI=I
2520 LNUM=XI/19.53125+.5
2530 LNUM=LNUM+1
2540 F(LNUM,1)=A(LNUM,1)
2550 F(LNUM,2)=A(LNUM,2)
2551 F(514-LNUM,1)=A(514-LNUM,1)
2552 70 F(514-LNUM,2)=A(514-LNUM,2)
2560 IF(LAST.EQ.6)RETURN
2570C----INVERSE FOURIER TRANSFORM(7)
2580 7 CALL CHAIN("LINK10")
2590 RETURN
2600C-----*****
2610C-----CONSTRUCT NEW F KEEPING HARMONICS AT PITCH FREQ AND SUPPRESSING ALL
2620C OTHERS BY 1/2.
2630 8 DO 80 I=1.512
2640 F(I,1)=A(I,1)/2.
2650 80 F(I,2)=A(I,2)/2.
2651 DO 85 I=154.360
2652 F(I,1)=0.
2653 85 F(I,2)=0.
2660 DO 90 I=IMAX,3000,IMAX
2670 XI=I
2680 LNUM=XI/19.53125+.5
2690 LNUM=LNUM+1
2700 F(LNUM,1)=A(LNUM,1)
2710 F(LNUM,2)=A(LNUM,2)
2711 F(514-LNUM,1)=A(514-LNUM,1)
2712 90 F(514-LNUM,2)=A(514-LNUM,2)
2720 IF(LAST.EQ.8)RETURN
2730C----INVERSE FOURIER TRANSFORM
2740 CALL CHAIN("LINK10")
2750 RETURN
2760 END
3000C-----SPECTRAL NOISE SUBTRACTION PROCESSOR

```

```

3010 SUBROUTINE SSP(XNOISE)
3020 COMMON ARRAY(63),DT,N,F(512,2),G(512,2)
3025 DIMENSION A(512)
3030 PRINT:"ENTER LAST STEP TO COMPLETE"
3040 READ:LAST
3050 PRINT:"ENTER STARTING POINT NUMBER"
3060 READ:NSTART;IF(NSTART.EQ.0)GO TO 1
3070 GO TO (1,2,3,4,5,6,7,8),NSTART
3080C-----TRIANGULAR TIME WINDOW(1)
3090 1 DO 19 J=1,256
3100 XJ=J
3110 F(J,1)=F(J,1)*XJ/256.
3120 19 F(J+256,1)=F(J+256,1)*(257.-XJ)/256.
3130 IF(LAST.EQ.1)RETURN
3140C-----FOURIER TRANSFORM(2)
3150 2 CALL CHAIN("LINK5")
3160 IF(LAST.EQ.2)RETURN
3170C-----ESTIMATE AVERAGE NOISE LEVEL(3)
3180 3 SUM=0.
3190 DO 20 I=154,257
3200 20 SUM=SUM+SQR(F(I,1)**2+F(I,2)**2)
3210 XNOISE=SUM/104.
3220 IF(LAST.EQ.3)RETURN
3230C-----MODIFY RE(F) AND I(F) BY SUBTRACTING NOISE(4)
3240 4 DO 30 I=1,512
3250 XMAG=SQR(F(I,1)**2+F(I,2)**2)
3260 IF(XMAG.GT.2.*XNOISE)GO TO 40
3270 F(I,1)=F(I,1)/2.
3280 F(I,2)=F(I,2)/2.
3290 GO TO 30
3300 40 F1=(XMAG-XNOISE)*COS(ATAN2(F(I,2),F(I,1)))
3310 F(I,2)=(XMAG-XNOISE)*SIN(ATAN2(F(I,2),F(I,1)))
3320 F(I,1)=F1
3330 30 CONTINUE
3340 50 DO 50 I=154,360
3350 F(I,1)=0.
3360 F(I,2)=0.
3370 IF(LAST.EQ.4)RETURN
3380C-----INVERSE FOURIER TRANSFORM(5)

```

```

33350 F(I,2)=0.
3340 IF(LAST.EQ.4)RETURN
3350C-----INVERSE FOURIER TRANSFORM(5)
33605 CALL CHAIN("LINK10")
3370 RETURN
3380C*****OPTIMUM METHOD(6)*****
3390C-----MODIFY POWER SPECTRUM ACCORDING TO CURTIS' OPTIMUM METHOD(6)
34006 DO 60 I=1,512
3410 RE=F(I,1)
3420 XI'=F(I,2)
3430 XMAG2=F(I,1)**2+F(I,2)**2
3440 XMAG=SQRT(XMAG2**2/(XMAG2+XNOISE**2))
3450 F(I,1)=XMAG*COS(ATAN2(XI',RE))
3460 F(I,2)=XMAG*SIN(ATAN2(XI',RE))
3470 IF(LAST.EQ.6)RETURN
3480C-----INVERSE FOURIER TRANSFORM
3490 CALL CHAIN("LINK10")
3500 RETURN
3510C*****MODIFY POWER SPECTRUM AS IN 6 PLUS GATE OUT HIGH FREQ TERMS(7)*****
3520C-----MODIFY POWER SPECTRUM AS IN 6 PLUS GATE OUT HIGH FREQ TERMS(7)
35307 DO 70 I=1,512
3540 RE=F(I,1)
3550 XI'=F(I,2)
3560 XMAG2=F(I,1)**2+F(I,2)**2
3570 XMAG=SQRT(XMAG2**2/(XMAG2+XNOISE**2))
3580 F(I,1)=XMAG*COS(ATAN2(XI',RE))
3590 F(I,2)=XMAG*SIN(ATAN2(XI',RE))
3600 DO 80 I=154,360
3610 F(I,1)=0.
3620 F(I,2)=0.
3630 IF(LAST.EQ.7)RETURN
3640C-----INVERSE FOURIER TRANSFORM
3650 CALL CHAIN("LINK10")
3660 RETURN
3670C*****MODIFY POWER SPECTRUM AS IN 6 WITH WEIGHTING LIKE THAT FOR LONG TERM
3680C-----MODIFY POWER SPECTRUM AS IN 6 WITH WEIGHTING LIKE THAT FOR LONG TERM
3690C AVERAGE OF NORMAL SPEECH
37003 A(1)=0.1A(2)=1./2**8

```

```

3590 A(3)=1./2.**5
3690 A(4)=1./12.*A(5)=.5
3700 DO 90 I=6,27
3710 90 A(I)=1.
3720 DO 100 I=28,257
3730 XI=I
3740 FREQ=(XI-1.)*19.53125
3750 100 A(I)=(500./FREQ)**2
3760 SUM=0.
3770 DO 110 I=1,257
3780 110 SUM=SUM+A(I)**2
3790 SUM=SQRT(SUM)/257.
3800 RNOISE=0.
3810 DO 120 I=154,257
3820 120 RNOISE=RNOISE+SQRT(F(I,1)**2+F(I,2)**2)
3830 RNOISE=RNOISE/104.
3840 DO 130 I=1,257
3850 130 A(I)=A(I)*RNOISE/SUM
3860 DO 140 I=2,256
3870 140 A(514-I)=A(I)
3880 DO 150 I=1,512
3890 RE=F(I,1)
3900 XI=F(I,2)
3910 XWAG2=F(I,1)**2+F(I,2)**2
3920 XWAG=(A(I)**2*XWAG2)/(A(I)**2+RNOISE**2)
3930 F(I,1)=XWAG*COS(ATAN2(XIM,RE))
3940 150 F(I,2)=XWAG*SIN(ATAN2(XIM,RE))
3950 IF(LAST.EQ.8)RETURN
3960 ---- INVERSE FOURIER TRANSFORM
3970 CALL CHAIN("LINK10")
3980 RETURN
3990 END

```

APPENDIX C

Source listing of the program "RTB", magnetic tape read program.
This program reads a record of length 512 words from tape with assigned number "10".

```

*LIST RTBFILE
010 SYMDEF RTB
020 BLOCK IO 512
030IBUF BSS 1
040IEOF BSS 1
050IBORI BSS 1
060IPARTY BSS 1
065 BSS 3328
070 USE PREVIOUS
080PTB SAVE
090 LDA =-2,DL
100 STA STR
110 ME GEINOS
120 RTB
130 ZERO FC.DCW
140 ZERO STR
150 ME GEROAD
160 LDA STR
170 ANA =00700000000000
180 CMPA =00300000000000
190 TZE ABORT
200 CMPA =00400000000000
210 TZE EOF
220 LDA =0,DL
230 TRA RETURN
240ABORT =1,DL
250 STA IPARTY
260 TRA RETURN+1
270EOF LDA =1,DL
280RETURN STA IEOF

```

290
300-C
310-G
320-TR
330

PETUPH
BCI
IOM
RSS
END

PTB
1.000010
IBUF, 512
2

ready

APPENDIX D

Source listing of the program "WTB", magnetic tape write program.
This program writes a record of length 512 words onto tape with assigned number 20.

```
*LIST WTBFILE
0010      SYMDEF      WTB
0020      BLOCK      10
0030      BSS        512
0040      BSS        1
0050      BSS        1
0060      BSS        1
0070      BSS        3328
0080      USE        PREVIOUS
0090      SAVE
0100      WME        GEINOS
0110      WTB        FC.DCW
0120      ZERO       STR
0130      ZERO       GERDAD
0140      WME        WTB
0150      RETURN     1,000020
0160      BCI        IBUF,512
0170      IOTD
0180      BSS        2
0190      END

ready
```

APPENDIX E

Source listing of the program "WTBZ", magnetic tape write program.
This program writes a record of length 1 word onto tape with assigned
number 20

```
*LIST WTBZFILE
0010 SYMDEF WTBZ
0020 BLOCK IO
0030 BSS 512
0040 BSS 1
0050 BSS 1
0060 BSS 1
0070 BSS 3328
0080 USE PREVIOUS
0090 SAVE
0100 WTBZ
0110 WTBZ
0120 ZERO
0130 ZERO
0140 WTBZ
0150 RETURN
0160 BCI
0170 IOTD
0180 BSS
0190 END

FC.DCW
STR
GEROAD
WTBZ
1.000020
IBUF.1
2

GEINOS
```

ready

APPENDIX F

Batchjob listing of the sequence of operations to be performed in copying one magnetic tape (assigned name "IN") onto another magnetic tape (assigned name "OT") which may be read without parity errors.

```
*LIST UTILITY
0020s:IDENT:AIAD0005,NIEDERJOH,40270001RADC
0022s:USERID:AIAD0005,WORLD
0025s:MSG:NICK_D0 A "KILL" AFTER EQJ-NIEDERJOHN
0030s:UTILITY
0040s:QUTIL:IGNORE/145
0050s:FFILE:IN,PHYREC
0060s:FUTIL:IN,OT,RWD/IN,OT,COPY/144R/
0070s:TAPE7:IN,AID,,S0621,,INPUT,,DEN5
0080s:TAPE7:OT,X2D,,S0622,,OUTPUT,,DEN5
0090s:LIMITS:99,20K,,10000
0100s:ENDJOB
```

ready

APPENDIX G

Batchjob and source listing of a program (RTAPE) to copy a magnetic tape to a data file. Note that program "RTBFILE" (Appendix C) is used.

*LIST BATCHJOB

```
0020s:IDENI:AIAD0005,NIEDERJOH,40270001RADC
0030s:USERID:AIAD0005$WORLD
0040s:OPTION:FORTRAN
0050s:FORTY:NDECK
0060s:SELECTA:AIAD0005/RTAPE
0070s:GAP:NDECK
0080s:SELECTA:AIAD0005/RTBFILE
0090s:EXECUTE
0100s:LIMITS:25,25K
0110s:PRMFL:12,R/A,L,AIAD0005/TAPDAT2
0120s:TAPE7:10,XID,,50623,,SPCH + NOISE,,DEN5
0130s:FFILE:10,NSIDLB,BUFSIZ/512,NOSRLS
0140s:ENDJOB
```

ready

*LIST RTAPE

```
0010 COMMON /IO/ IBUF(512),IEOF,IBORT,IPARTY,IDATA(2048),JODATA(1280)
0030 CALL RTB:CALL RTB:CALL RTB
0050 DO 1 I=1,150
0060 CALL RTB
0065 IF(IEOF.NE.0)GO TO 2
0070 WRITE(12)IBUF
0080 1 CONTINUE
0100 2 CONTINUE
0110 STOP
0120 END
```

ready

APPENDIX H

Batchjob (FITA25) and source listing for a program (PROCESS) to process real speech. Input data is from a data file (TAPDAT2) and output data is directed to tape with label "S0625". Note that programs "WTBFILE" (Appendix D) and "WTBZFILE" (Appendix E) are called during execution of this job.

*LIST FITA25

```
0020s:IDENT:AIAD0005,NIEDERJON,40270001RADC
0030s:USERID:AIAD0005SWORLD
0040s:OPTION:FORTRAN
0050s:FORIY:VDECK
0055s:LIMITS:,30K
0060s:SELECT:AIAD0005/PROCESS
0070s:GRAP:VDECK
0082s:SELECT:AIAD0005/WTBFILE
0083s:GRAP:VDECK
0084s:SELECT:AIAD0005/WTBZFILE
0090s:EXECUTE
0100s:LIMITS:100,25K
0120s:PRMFL:10,P/W,L,AIAD0005/TAPDAT2
0131s:TAPE7:20,X20,,S0625,,OUTPUT,,DEV5
0132s:FFILE:20,MSIDLB,BUFSIZ/512,NOSRLS
0140s:ENDJOB
```

Ready

*LIST PROCESS

```
0010C:----ESTABLISH ARRAYS USED BY THE VARIOUS PROCESSING ALGORITHMS:
0020C IOBUF(512)-FOR I/O FROM/TO TAPE
0030C IEOF,IBORI,IPARTY-READ ERROR FLAGS
0040C IDATA(2048)-PAST AND PRESENT DATA RECORDS
0045C JODATA(1280)-OUTPUT DATA RECORD
0050C F(512,2)-PROCESSING BUFFER
```

```

010600 G(512,2)-TEMP BUFFER USED BY FFT AND IFFT
010700 A(512,2)-TEMP BUFFER USED IN THE ALGORITHMS
010700 S(512)-SIN/COS TABLE USED BY FFT & IFFT
010800 COMMON /IO/ IOBUF(512),IEOF,IBORT,IPARTY,IDATA(2048),JODATA(1280)
010900 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
010900 COMMON /SCTAB/ S(512)
011000-----NPROC SHOULD EQUAL THE NUMBER OF ENHANCEMENT PROCESSES TO BE
011000 PERFORMED
011000 NPROC=4
011100-----NREC SHOULD EQUAL THE NUMBER OF RECORDS TO BE PROCESSED BY EACH
011100 ENHANCEMENT PROCESS, NREC+NPREC MUST BE .LE. TOT. NU. REC. INPUT
011100 NREC=120
011200-----NPREC SHOULD EQUAL THE FIRST RECORD TO BE PROCESSED
011200 NPREC=1
011300 REWIND 10;REWIND 20;CALL TABLE(512)
011300-----WRITE A ONE WORD RECORD FOR THE ODP-116
011400 DO 50 I=1,1230
011500 JODATA(I)=0
011600 CALL DATTAP
011700 CALL J12Z
011800-----FIRST JUST COPY INPUT TO OUTPUT
011900 DO 62 I=1,NPREC-1
012000 READ(10)IOBUF
012100 DO 51 I=1,NREC
012200 CALL PTIME(A1);READ(10)IOBUF
012300 CALL WTB;CALL PTIME(B)
012400-----WRITE TEN ZERO RECORDS
012500 DO 52 I=1,1280
012600 JODATA(I)=0
012700 CALL DATTAP;DO 55 I=1,10
012800 CALL WTB;REWIND 10
012900 JRI=1;JR2=NPREC
013000 WRITE(06,53)JRI, JR2
013100 FORMAT(IX,"INPUT SPEECH,
013200 REC. NOS. ",I5," TO ",I5)
013300 II=0-A1
013400 WRITE(06,54)II
013500 FORMAT(IX,"PROCESSING TIME= ",F10.6," HOURS")
013600-----BEGIN PROCESSING TECHNIQUES

```



```

0135 20 CONTINUE
0500 49 CONTINUE
0500 C-----NORMALIZE OUTPUT
0500 I=1;JJJ=JJ
0507 CALL NORM1(II, JJJ, SFAC1)
0511 79 DO 30 II=N,N+511
0520 FJ0=F(II-II+1,1)*SFAC1
0521 J0=SIGN(1.0,FJ0)*(ABS(FJ0)+.5)
0522 80 J0DATA(II)=J0DATA(II)+J0
0530 200 CONTINUE
0530 C-----OUTPUT PROCESSED RECORD
0540 GO TO 202
0551 C-----TEMP TEST
0552 IF(JJ.GT.2)GO TO 202
0553 WRITE(66,201)J0DATA
0554 201 FORMAT(1X,10(13,2X))
0555 203 WRITE(66,203)J, JJ
0557 202 FORMAT(20X,"PROCESS ",I3,". RECORD ",I3)
0557 202 CONTINUE
055 C-----TEMP TEST
0557 CALL DATAP
0560 CALL WTB
0561 C-----REINITIALIZE J0DATA & IDATA
0562 DO 210 I=1,256
0570 210 J0DATA(I)=J0DATA(I+1024)
0580 DO 220 I=257,1280
0590 220 J0DATA(I)=0
0500 DO 230 I=1,1024
0510 230 IDATA(I)=IDATA(I+1024)
0520 300 CONTINUE
0525 C-----OUTPUT TEN ZERO RECORDS
0527 CALL PTIME(B)
0530 DO 390 I=1,1024
0540 390 J0DATA(I)=0
0547 CALL DATAP
0550 DO 391 I=1,10
0560 391 CALL WTB
0570 REWIND 10

```

```

0571 JR1=J*(NREC+10)+1
0572 JR2=JR1+NREC-1
0573 WRITE(06,399)J,JR1,JR2
0574 399 FORMAT(1X,"PROCESS",2X,13,2X,"COMPLETE, REC. NOS. ",15," TO ",15)
0575 TI=3-A1
0576 WRITE(06,54)TI
0577 400 CONTINUE
0580 STOP
0581 END
0600*****
0610C---PROCESSING ALGORITHMS
0620C INPUT/OUTPUT FOR ALGORITHMS IS F
0630C*****
0640 SUBROUTINE INTEL
0650C-----INTEL PROCESSOR
0660 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
0670 CALL ITW
0680 CALL FFT
0690 CALL FTOA
0700 CALL ZIMF
0710 CALL ZUHALF
0720 CALL SORTF
0730 CALL REVSIGN
0740 CALL FFT
0750 CALL ZLOW
0760 CALL IFFT
0770 CALL ZIMF
0780 CALL REVSIGN
0790 CALL SQUARE
0800 CALL RESTPH
0810 CALL IFFT
0820 RETURN
0830 END
0840C-----
0850 SUBROUTINE SUBT1
0860 SUBTRACT AVERAGE NOISE FROM FFT(F)
0870 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
0880 CALL ITW

```

```

1300 CALL FFT
1310 CALL ESTNOISE(XNOISE)
1320 CALL NOISESUB(XNOISE)
1330 CALL IFFT
1340 RETURN
1350 END

```

```

1360-----
1370 SUBROUTINE SUBT2
1380-----SUBTRACT AVERAGE NOISE, THEN WEIGHT TO EMPHASIZE F2
1400 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
1410 CALL TIM
1420 CALL FFT
1430 CALL ESTNOISE(XNOISE)
1440 CALL NOISESUB(XNOISE)
1450 CALL WEIGHT3
1460 CALL IFFT
1470 RETURN
1480 END

```

```

1490-----
1500 SUBROUTINE PAP
1510-----POWER SPECTRAL OPTIMIZATION WITH ACTUAL SPEECH AS EST. SPEECH
1511C SPECTRUM
1520 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
1530 CALL TIM
1540 CALL FFT
1550 CALL ESTNOISE(XNOISE)
1560 CALL WEIGHT(XNOISE)
1570 CALL IFFT
1580 RETURN
1590 END

```

```

1600-----
1610 SUBROUTINE MODPAP
1620-----SAME AS PAP WITH UPPER HALF OF FFT SET TO ZERO
1620 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
1630 CALL TIM
1640 CALL FFT
1650 CALL ESTNOISE(XNOISE)
1660 CALL WEIGHT(XNOISE)

```

```

1670 CALL ZUH1F
1680 CALL IFFT
1690 RETURN
1700 END
1710C-----
1720 SUBROUTINE SPPAP
1730C-----POWER SPECTRAL OPTIMIZATION WITH LONG TERM AVERAGE SPECTRUM USED AS
1740C ESTIMATED SPEECH SPECTRUM
1750 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
1760 CALL TTH
1770 CALL FFT
1780 CALL WEIGH2
1790 CALL IFFT
1800 RETURN
1810 END
1820C-----
1830 SUBROUTINE MODSPPAP
1840C-----SAVE AS SPPAP WITH UPPER HALF OF FFT SET TO ZERO
1850 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
1860 CALL TTH
1870 CALL FFT
1880 CALL WEIGH2
1890 CALL ZUH1F
1900 CALL IFFT
1910 RETURN
1920 END
1930C-----
1940 SUBROUTINE RJH1
1950C-----LEAVE SPECTRAL LINES AT PITCH FREQ. ONLY
1960 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
1970 CALL TTH
1980 CALL FFT
1990 CALL FTOA
2000 CALL MAG
2010 CALL PITCH(X)
2020 CALL HARM(X,O.)
2030 CALL IFFT
2040 RETURN

```

```

2040      END
2050C-----
2060      SUBROUTINE RJN2
2070C-----LEAVE SPECTRAL LINES AT PITCH FREQ. WITH ALL OTHERS SUPPRESSED
2071C      BY 1/2.
2080      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
2090      CALL TTH
2100      CALL FFT
2110      CALL FTOA
2120      CALL MAG
2130      CALL PITCH(X)
2140      X=X-10.
2150      CALL HARM(X,.5)
2160      CALL IFFT
2170      RETURN
2180C-----
2190      SUBROUTINE RJN3
2200C-----LEAVE SPECTRAL LINES AT PITCH FREQ. WITH ALL OTHERS SUPPRESSED
2201C      BY 1/4.
2210      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
2220      CALL TTH
2230      CALL FFT
2240      CALL FTOA
2250      CALL MAG
2260      CALL PITCH(X)
2270      CALL HARM(X,.25)
2280      CALL IFFT
2290      RETURN
2300      END
2310C-----
2320      SUBROUTINE RJN4
2330C-----SAME AS RJN1 WITH WEIGHTING ON F LIKE THAT CONSIDERING THE
2340C      IMPORTANCE OF EACH LINE TO NORMAL SPEECH
2350      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
2360      CALL TTH
2370      CALL FFT
2380      CALL FTOA

```

```

2390 CALL MAG
2400 CALL PITCH(X)
2410 CALL HARP(X,O.)
2420 CALL HEIGHT3
2430 CALL IFFT
2440 RETURN
2450 END
2460C-----
2470 SUBROUTINE NOPROC
2480C-----NO PROCESS,JUST TEST
2490 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
2500 CALL TIM
2510 RETURN
2520 END
2530C-----
2540 SUBROUTINE INTEL2
2550C-----MODIFIED INTEL PROCESS
2560 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
2570 CALL TIM
2580 CALL FFT
2590 CALL FTOA
2600 CALL MAG
2610 CALL ZIMF
2620 CALL ZURHALF
2630 CALL SORTF
2640 CALL REVSIGN
2650 CALL FFT
2660 CALL ZLOW2
2670 CALL IFFT
2680 CALL ZIMF
2690 CALL REVSIGN
2700 CALL SQUARE
2710 CALL RESIPH
2720 CALL IFFT
2730 RETURN
2740 END
2750C-----
2760 SUBROUTINE INTEL3

```

```

2770C-----INTEL WITHOUT SQUARE
2780  COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
2790  CALL TLW
2800  CALL FFT
2810  CALL FTOA
2820  CALL WAG
2830  CALL ZINF
2840  CALL ZUHALF
2850  CALL SORTF
2860  CALL REVSIGN
2870  CALL FFT
2880  CALL ZLOW
2890  CALL IFFT
2900  CALL ZINF
2910  CALL REVSIGN
2920  CALL ABSVF
2930  CALL RESTPH
2940  CALL IFFT
2950  RETURN
2960  END
2970C-----
2980  SUBROUTINE SUBT3
2990C-----SUBTRACT NOISE FROM FFT
3000  COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
3010  CALL TLW
3020  CALL FFT
3030  CALL ESTNOISE(XNOISE)
3040  CALL NOISUR2(XNOISE)
3050  CALL IFFT
3060  RETURN
3070  END
3080C-----
3090C-----SUBROUTINES USED BY THE VARIOUS PROCESSING ALGORITHMS
3100C-----
3110C-----
3120  SUBROUTINE TLW
3130C-----TRIANGULAR TIME "WINDOW" ON F
3140  COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
3150  DO 10 I=1,256

```

```

5080      XI=I-1
5090      F(I,1)=F(I,1)*XI/255.
5100      F(I+256,1)=F(I+256,1)*(255.-XI)/255.
5110      RETURN
5111      END
5112-----
5113      SUBROUTINE FFT
5114      SUPPLIED TRANSFORM
5117C-----
5119      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
5120      COMMON /SCTAB/ S(512),N=512
5121      DO 100 I=1,N;N=N+1;II=I
5122      CALL FILE(N,II,J)
5123      G(J,1)=F(II,1)
5124      G(J,2)=F(II,2)
5125      CONTINUE
5126      DO 100 I=1,N
5127      CALL FORTIER(N,G)
5128      DO 200 I=1,N
5129      F(I,1)=G(I,1)
5130      F(I,2)=G(I,2)
5140      CONTINUE
5150      RETURN
5160      END
5170-----
5171      SUBROUTINE FILE(N,J,I)
5172      SUBROUTINE USED BY FFT AND IFFT
5173C-----
5174      DIMENSION L(12),LA(12)
5175      I=1
5176      IF(J.NE.1)GO TO 103
5177      DO 100 K=1,12
5178      LA(K)=0
5179      NA=2**K
5180      L(K)=NA/NA
5181      IF(NA.EQ.N)GO TO 101
5182      CONTINUE
5183      I=K+1
5184      IF(K.EQ.12)GO TO 110
5185      DO 102 K=I,12
5186      LA(K)=0
5187      CONTINUE
5188      GO TO 101
5189      END

```

```

5337 L(K)=0
5340 102 CONTINUE
5345 110 CONTINUE
5350 RETURN
5360 103 DO 104 K=1,12
5370 IF(LA(K).NE.0)GO TO 104
5380 LA(K)=1
5390 GO TO 105
5400 104 LA(K)=0
5410 105 DO 106 K=1,12
5420 106 I=1+LA(K)*L(K)
5430 RETURN
5431 END
5440 -----
5450 SUBROUTINE FORIER(N,F)
5460 -----SUBROUTINE USED BY FFT
5465 COMMON /SCTAB/ S(512)
5470 DIMENSION F(512,2)
5480 DO 100 I=1,10
5490 K=2*I
5500 100 IF(MOD(K)30 TO 101
5510 STOP
5520 -----ERROR THAT N IS NOT MOD 2 OR N IS GREATER THAN 1024 OR LESS THAN 2
5530 101 K=1
5540 K1=2*(K-2)
5550 K2=2*K1
5560 KSTEP=2*K1-1
5570 DO 102 I=1,K
5580 L=2*(I-1)
5590 I1=2*L
5600 DO 104 I2=1,N,I1
5610 DO 103 J=1,L
5620 K=I2+J-1
5630 AP=F(K,1)
5640 AI=F(K,2)
5650 BR=F(I1+L,1)
5660 BI=F(I1+L,2)
5670 A=(J-1)*K2/L+1

```

```

5652 CI=-S(MA)
5654 IF(MA-KI)107,107,105
5656 107 KK=MA+KI
5658 CR=S(KK)
5660 30 TO 106
5662 105 KK=MA-KI
5664 CR=-S(KK)
5680 106 DR=BP*CR-BI*CI
5690 DI=BR*CI+BI*CR
5700 F(M,1)=AR+DR
5710 F(M,2)=AI+DI
5720 F(M+L,1)=AR-DR
5730 F(M+L,2)=AI-DI
5740 103 CONTINUE
5750 104 CONTINUE
5760 102 CONTINUE
5770 RETURN
5771 END
5780C-----
5790 SUBROUTINE FTOA
5800C-----SAVE F IN A
5820 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
5830 DO 20 I=1,512
5840 A(I,1)=F(I,1)
5850 20 A(I,2)=F(I,2)
5860 RETURN
5861 END
5870C-----
5880 SUBROUTINE MAG
5890C-----MAKE RE(F) EQUAL TO PRESENT MAGNITUDE
5892 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
5900 DO 30 I=1,512
5910 30 F(I,1)=SORT(F(I,1)**2+F(I,2)**2)
5920 RETURN
5921 END
5930C-----
5940 SUBROUTINE ZIMP
5950C-----SET IM(F)=C

```

```

5251 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
5260 DO 40 I=1,512
5270 40 F(I,2)=0.
5280 RETURN
5290 END
-----
5290C SUBROUTINE ZUHALF
6000 ZERO UPPER HALF OF F
6010C-----
6030 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
6040 DO 50 I=129,395
6050 F(I,1)=0.
6051 50 F(I,2)=0.
6060 RETURN
6065 END
-----
6070C SUBROUTINE SQPTF
6080 TAKE SORT OF F
6090C-----
6110 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
6120 DO 60 I=1,512
6130 60 F(I,1)=SORT(F(I,1))
6140 RETURN
6145 END
-----
6150C SUBROUTINE REVSIGN
6160 REVERSE SIGNS OF ALL ODD ELEMENTS OF F
6170C-----
6190 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
6200 DO 70 I=1,511,2
6210 70 F(I,1)=-F(I,1)
6220 RETURN
6225 END
-----
6230C SUBROUTINE ZLOW
6240 ZERO LOW FIVE ELEMENTS OF F
6250C-----
6270 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
6280 DO 100 I=253,261
6290 100 F(I,1)=0.
6300 100 F(I,2)=0.
6310 RETURN
6315 END

```

```

6320C-----
6330      SUBROUTINE IFFT
6340C-----INVERSE FOURIER TRANSFORM
6350      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
6360      COMMON /SCTAB/ S(512)
6370      N=512
6380      DO 100 I=1,N;NM=N;II=I
6390      CALL FILE(NM,II,J)
6400      G(J,1)=F(II,1)
6410      G(J,2)=F(II,2)
6420      CONTINUE
6430      CALL IFOIER(N,G)
6440      DO 200 I=1,N
6450      F(I,1)=G(I,1)
6460      F(I,2)=G(I,2)
6470      CONTINUE
6480      RETURN
6490      END
6500C-----
6510      SUBROUTINE IFOIER(N,F)
6520C-----SUBROUTINE USED BY IFFT
6530      COMMON /SCTAB/ S(512)
6540      DIMENSION F(512,2)
6550      DO 100 I=1,10
6560      K=2**I
6570      IF(N.EQ.K)GO TO 101
6580C-----STOP
6590      101      K=I
6600      K1=2*(K-2)
6610      K2=2*K1
6620      KSTP=2*K1-1
6630      DO 102 I=1,K
6640      L=2*(I-1)
6650      II=2*L
6660      DO 104 I2=1,II,II
6670      DO 103 J=1,L
6680      M=I2+J-1

```

6690 IF(N.EQ.K)GO TO 101

6700 STOP

6710 IF(N.EQ.K)GO TO 101

6720 STOP

6730 IF(N.EQ.K)GO TO 101

6740 STOP

6750 IF(N.EQ.K)GO TO 101

6760 STOP

6770 IF(N.EQ.K)GO TO 101

6780 STOP

6790 IF(N.EQ.K)GO TO 101

6800 STOP

6810 IF(N.EQ.K)GO TO 101

6820 STOP

6830 IF(N.EQ.K)GO TO 101

6840 STOP

6850 IF(N.EQ.K)GO TO 101

6860 STOP

6870 IF(N.EQ.K)GO TO 101

6880 STOP

6890 IF(N.EQ.K)GO TO 101

6900 STOP

6910 IF(N.EQ.K)GO TO 101

6920 STOP

6930 IF(N.EQ.K)GO TO 101

6940 STOP

6950 IF(N.EQ.K)GO TO 101

6960 STOP

6970 IF(N.EQ.K)GO TO 101

6980 STOP

6990 IF(N.EQ.K)GO TO 101

7000 STOP

7010 IF(N.EQ.K)GO TO 101

7020 STOP

7030 IF(N.EQ.K)GO TO 101

7040 STOP

7050 IF(N.EQ.K)GO TO 101

7060 STOP

7070 IF(N.EQ.K)GO TO 101

7080 STOP

7090 IF(N.EQ.K)GO TO 101

7100 STOP

7110 IF(N.EQ.K)GO TO 101

7120 STOP

7130 IF(N.EQ.K)GO TO 101

7140 STOP

7150 IF(N.EQ.K)GO TO 101

7160 STOP

7170 IF(N.EQ.K)GO TO 101

7180 STOP

7190 IF(N.EQ.K)GO TO 101

7200 STOP

7210 IF(N.EQ.K)GO TO 101

7220 STOP

7230 IF(N.EQ.K)GO TO 101

7240 STOP

7250 IF(N.EQ.K)GO TO 101

7260 STOP

7270 IF(N.EQ.K)GO TO 101

7280 STOP

7290 IF(N.EQ.K)GO TO 101

7300 STOP

7310 IF(N.EQ.K)GO TO 101

7320 STOP

7330 IF(N.EQ.K)GO TO 101

7340 STOP

7350 IF(N.EQ.K)GO TO 101

7360 STOP

7370 IF(N.EQ.K)GO TO 101

7380 STOP

7390 IF(N.EQ.K)GO TO 101

7400 STOP

7410 IF(N.EQ.K)GO TO 101

7420 STOP

7430 IF(N.EQ.K)GO TO 101

7440 STOP

7450 IF(N.EQ.K)GO TO 101

7460 STOP

7470 IF(N.EQ.K)GO TO 101

7480 STOP

7490 IF(N.EQ.K)GO TO 101

7500 STOP

7510 IF(N.EQ.K)GO TO 101

7520 STOP

7530 IF(N.EQ.K)GO TO 101

7540 STOP

7550 IF(N.EQ.K)GO TO 101

7560 STOP

7570 IF(N.EQ.K)GO TO 101

7580 STOP

7590 IF(N.EQ.K)GO TO 101

7600 STOP

7610 IF(N.EQ.K)GO TO 101

7620 STOP

7630 IF(N.EQ.K)GO TO 101

7640 STOP

7650 IF(N.EQ.K)GO TO 101

7660 STOP

7670 IF(N.EQ.K)GO TO 101

7680 STOP

7690 IF(N.EQ.K)GO TO 101

7700 STOP

7710 IF(N.EQ.K)GO TO 101

7720 STOP

7730 IF(N.EQ.K)GO TO 101

7740 STOP

7750 IF(N.EQ.K)GO TO 101

7760 STOP

7770 IF(N.EQ.K)GO TO 101

7780 STOP

7790 IF(N.EQ.K)GO TO 101

7800 STOP

7810 IF(N.EQ.K)GO TO 101

7820 STOP

7830 IF(N.EQ.K)GO TO 101

7840 STOP

7850 IF(N.EQ.K)GO TO 101

7860 STOP

7870 IF(N.EQ.K)GO TO 101

7880 STOP

7890 IF(N.EQ.K)GO TO 101

7900 STOP

7910 IF(N.EQ.K)GO TO 101

7920 STOP

7930 IF(N.EQ.K)GO TO 101

7940 STOP

7950 IF(N.EQ.K)GO TO 101

7960 STOP

7970 IF(N.EQ.K)GO TO 101

7980 STOP

7990 IF(N.EQ.K)GO TO 101

8000 STOP

8010 IF(N.EQ.K)GO TO 101

8020 STOP

8030 IF(N.EQ.K)GO TO 101

8040 STOP

8050 IF(N.EQ.K)GO TO 101

8060 STOP

8070 IF(N.EQ.K)GO TO 101

8080 STOP

8090 IF(N.EQ.K)GO TO 101

8100 STOP

8110 IF(N.EQ.K)GO TO 101

8120 STOP

8130 IF(N.EQ.K)GO TO 101

8140 STOP

8150 IF(N.EQ.K)GO TO 101

8160 STOP

8170 IF(N.EQ.K)GO TO 101

8180 STOP

8190 IF(N.EQ.K)GO TO 101

8200 STOP

8210 IF(N.EQ.K)GO TO 101

8220 STOP

8230 IF(N.EQ.K)GO TO 101

8240 STOP

8250 IF(N.EQ.K)GO TO 101

8260 STOP

8270 IF(N.EQ.K)GO TO 101

8280 STOP

8290 IF(N.EQ.K)GO TO 101

8300 STOP

8310 IF(N.EQ.K)GO TO 101

8320 STOP

8330 IF(N.EQ.K)GO TO 101

8340 STOP

8350 IF(N.EQ.K)GO TO 101

8360 STOP

8370 IF(N.EQ.K)GO TO 101

8380 STOP

8390 IF(N.EQ.K)GO TO 101

8400 STOP

8410 IF(N.EQ.K)GO TO 101

8420 STOP

8430 IF(N.EQ.K)GO TO 101

8440 STOP

8450 IF(N.EQ.K)GO TO 101

8460 STOP

8470 IF(N.EQ.K)GO TO 101

8480 STOP

8490 IF(N.EQ.K)GO TO 101

8500 STOP

8510 IF(N.EQ.K)GO TO 101

8520 STOP

8530 IF(N.EQ.K)GO TO 101

8540 STOP

8550 IF(N.EQ.K)GO TO 101

8560 STOP

8570 IF(N.EQ.K)GO TO 101

8580 STOP

8590 IF(N.EQ.K)GO TO 101

8600 STOP

8610 IF(N.EQ.K)GO TO 101

8620 STOP

8630 IF(N.EQ.K)GO TO 101

8640 STOP

8650 IF(N.EQ.K)GO TO 101

8660 STOP

8670 IF(N.EQ.K)GO TO 101

8680 STOP

8690 IF(N.EQ.K)GO TO 101

8700 STOP

8710 IF(N.EQ.K)GO TO 101

8720 STOP

8730 IF(N.EQ.K)GO TO 101

8740 STOP

8750 IF(N.EQ.K)GO TO 101

8760 STOP

8770 IF(N.EQ.K)GO TO 101

8780 STOP

8790 IF(N.EQ.K)GO TO 101

8800 STOP

8810 IF(N.EQ.K)GO TO 101

8820 STOP

8830 IF(N.EQ.K)GO TO 101

8840 STOP

8850 IF(N.EQ.K)GO TO 101

8860 STOP

8870 IF(N.EQ.K)GO TO 101

8880 STOP

8890 IF(N.EQ.K)GO TO 101

8900 STOP

8910 IF(N.EQ.K)GO TO 101

8920 STOP

8930 IF(N.EQ.K)GO TO 101

8940 STOP

8950 IF(N.EQ.K)GO TO 101

8960 STOP

8970 IF(N.EQ.K)GO TO 101

8980 STOP

8990 IF(N.EQ.K)GO TO 101

9000 STOP

9010 IF(N.EQ.K)GO TO 101

9020 STOP

9030 IF(N.EQ.K)GO TO 101

9040 STOP

9050 IF(N.EQ.K)GO TO 101

9060 STOP

9070 IF(N.EQ.K)GO TO 101

9080 STOP

9090 IF(N.EQ.K)GO TO 101

9100 STOP

9110 IF(N.EQ.K)GO TO 101

9120

```

6570 AR=F(M,1)
6580 AI=F(M,2)
6590 BP=F(M+L,1)
6600 BI=F(M+L,2)
6710 MA=(J-1)*K2/L+1
6711 CI=S(MA)
6712 IF(MA-K1)107,107,105
6713 KK=MA+K1
6714 CR=S(KK)
6720 GO TO 106
6721 KK=MA-K1
6722 CR=-S(KK)
6730 DR=BP*CR-BI*CI
6731 DI=BP*CI+BI*CR
6740 F(M,1)=A+DR
6750 F(M,2)=AI+DI
6760 F(M+L,1)=AP-DR
6770 F(M+L,2)=AI-DI
6780
6790
6800
6810
6820
6830
6840
6850-----
6860 SUBROUTINE SQUARE
6870-----SQUARE RE(F)
6880 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
6890 DO 100 I=1,512
6900 F(I,1)=F(I,1)**2
6910 RETURN
6920 END
6930-----
6940 SUBROUTINE RESTPH
6950-----RESTORE PHASE OF F FROM A
6960 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
6970 DO 10 I=1,512
6980 F(I,2)=F(I,1)*SIN(ATAN2(A(I,2),A(I,1)))
6990 G(I,1)=F(I,1)*COS(ATAN2(A(I,2),A(I,1)))

```

```

7000 RETURN
7010 END
7120C-----
7130 SUBROUTINE ESTNOISE(XNOISE)
7140C-----ESTIMATE AVERAGE NOISE LEVEL
7145 C0=0.0 /BUFFER2/ F(512,2),G(512,2),A(512,2)
7150 SUM=0.
7160 DO 20 I=154,257
7170 SUM=SUM+SORT(F(I,1)**2+F(I,2)**2)
7180 XNOISE=SUM/104.
7190 RETURN
7195 END
7110C-----
7120 SUBROUTINE NOISESUB(XNOISE)
7130C-----MODIFY F BY SUBTRACTING NOISE AND SET F(I,1 AND 2)=0., I=154,360
7140 C0=0.0 /BUFFER2/ F(512,2),G(512,2),A(512,2)
7150 DO 30 I=1,512
7160 XNAG=SQRT(F(I,1)**2+F(I,2)**2)
7170 IF(XNAG.GT.2.*XNOISE)GO TO 40
7180 F(I,1)=F(I,1)/2.
7190 F(I,2)=F(I,2)/2.
7200 GO TO 30
7210 40 F1=(XNAG-XNOISE)*COS(ATAN2(F(I,2),F(I,1)))
7220 F(I,2)=(XNAG-XNOISE)*SIN(ATAN2(F(I,2),F(I,1)))
7230 F(I,1)=F1
7240 30 CONTINUE
7250 DO 50 I=154,360
7260 F(I,1)=0.
7270 F(I,2)=0.
7280 RETURN
7290 END
7300C-----
7310 SUBROUTINE WEIGH(XNOISE)
7320C-----MODIFY POWER SPECTRUM ACCORDING TO CURTIS' OPTIMUM METHOD
7325 C0=0.0 /BUFFER2/ F(512,2),G(512,2),A(512,2)
7330 DO 60 I=1,512
7340 RE=F(I,1)
7350 XI=F(I,2)

```

```

7350 IF((XIM.EQ.0.).AND.(RE.EQ.0.))30 TO 60
7360 XMAG2=F(I,1)**2+F(I,2)**2
7370 XMAG=SQRT(XMAG2**2/(XMAG2+XNOISE**2))
7380 F(I,1)=XMAG*COS(ATAN2(XIM,RE))
7390 F(I,2)=XMAG*SIN(ATAN2(XIM,RE))
7395 60 CONTINUE
7400 RETURN
7410 END
7420-----
7430 SUBROUTINE WEIGH2
7440C-----MODIFY POWER SPECTRUM AS IN WEIGHT1 BUT WITH WEIGHTING LIKE THAT FOR
7450C LONG TERM AVERAGE OF NORMAL SPEECH
7455 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
7459C-----CALCULATE ARRAY G, THE LONG TERM AV. FOR NORMAL SPEECH
7460 G(I,1)=0.;G(2,1)=1./2.**8;G(3,1)=1./2.**5;G(4,1)=1./12.;G(5,1)=.5
7470 DO 90 I=6,27
7480 90 G(I,1)=1.
7490 DO 100 I=28,257
7500 XI=I
7510 FREQ=(XI-1.)*19.53125
7520 100 G(I,1)=(500./FREQ)**2
7530 SUM=0.
7540 DO 110 I=1,257
7550 110 SUM=SUM+G(I,1)
7560 SUM=SUM/257.
7561C-----CALCULATE AVERAGE OF MAG(F)
7562 FSUM=0.
7563 DO 115 I=1,257
7564 115 FSUM=FSUM+SQRT(F(I,1)**2+F(I,2)**2)
7565 FSUM=FSUM/257.
7566C-----CALCULATE NOISE AVERAGE
7570 RNOISE=0.
7580 DO 120 I=154,257
7590 120 RNOISE=RNOISE+SQRT(F(I,1)**2+F(I,2)**2)
7600 RNOISE=RNOISE/104.
7610 DO 130 I=1,257
7620 130 G(I,1)=G(I,1)*FSUM/SUM
7630 DO 140 I=2,256
7640 140 G(514-I,1)=G(I,1)

```

```

7650      DO 150 I=1.512
7660      RE=F(I,1)
7670      XIM=F(I,2)
7680      XMAG2=F(I,1)**2+F(I,2)**2
7690      XMAG=(G(I,1)**2*XMAG2)/(G(I,1)**2+RN(I)*SE**2)
7700      F(I,1)=XMAG*COS(ATAN2(XIM,RE))
7710      F(I,2)=XMAG*SIN(ATAN2(XIM,RE))
7720      RETURN
7730      END
7740C-----
7750      SUBROUTINE PITCH(X)
7760C-----CALCULATE PITCH AS HARMONIC NUMBER = X
7770      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
7780      SUMMAX=0.
7790      DO 40 I=80,250
7800      XNUM=3000/I
7810      SUM=0.
7820      DO 50 J=1,3000.I
7830      XJ=J
7840      LNUM=XJ/19.53125+.5
7850      LNUM=LNUM+1
7860      SUM=SUM+F(LNUM,1)
7870      SUM=SUM/XNUM
7880      IF(SUM.LE.SUMMAX)GO TO 40
7890      IF(1ABS(IMAX-1/2).GE.5)GO TO 45
7900      IF(SUM.LE.1.33*SUMMAX)GO TO 40
7910      SUMMAX=SUM
7920      IMAX=I
7930      CONTINUE
7940      X=IMAX
7950      RETURN
7960C-----
7970      SUBROUTINE HARM(X,XF)
7980C-----CALCULATE NEW SPECTRUM(F) FROM OLD SPECTRUM(IN A), PITCH(FROM X).
7990C      AND FACTOR BY WHICH NON-HARMONIC LINES ARE TO BE MULTIPLIED (XF)
8000      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
8010      DO 80 I=1.512
8020      F(I,1)=A(I,1)*XF

```

```

8030 80 F(I,2)=A(I,2)*XF
8040 D0 85 I=154,360
8050 F(I,1)=0.
8060 85 F(I,2)=0.
8070 IMAX=X+.5
8080 D0 90 I=IMAX,3000,IMAX
8090 XI=I
8100 LNUM=XI/19.53125+.5
8110 LNUM=LNUM+1
8120 F(LNUM,1)=A(LNUM,1)
8130 F(LNUM,2)=A(LNUM,2)
8140 F(514-LNUM,1)=A(514-LNUM,1)
8150 90 F(514-LNUM,2)=A(514-LNUM,2)
8160 RETURN
8170 END
8180C-----
8190 SUBROUTINE TAPDAT(J)
8200C-----TRANSLATE IOBUF(I),I=1,512 INTO IDATA(I), 1ST OF 2ND HALF
8210C IOBUF IS IN TAPE BINARY FORMAT, IDATA IS IN NUMERICAL FORMAT
8220 COMMON /IO/ IOBUF(512),IEOF,IBORT,IPARTY,IDATA(2048),JODATA(1280)
8225 K=(J-1)*1024
8230 D0 20 I=1,512
8240 IDATA(2*I-1+K)=0
8250 IDATA(2*I+K)=0
8260 FLD(21,1,1,IDATA(2*I-1+K))=FLD(1,11,IOBUF(I))
8270 FLD(32,4,IDATA(2*I-1+K))=FLD(14,4,IOBUF(I))
8280 FLD(21,1,1,IDATA(2*I+K))=FLD(19,11,IOBUF(I))
8290 FLD(32,4,IDATA(2*I+K))=FLD(32,4,IOBUF(I))
8300 IF(FLD(0,1,IOBUF(I)).EQ.0)GO TO 10
8310 IDATA(2*I-1+K)=-IDATA(2*I-1+K)
8320 10 IF(FLD(18,1,IOBUF(I)).EQ.0)GO TO 20
8330 IDATA(2*I+K)=-IDATA(2*I+K)
8340 20 CONTINUE
8350 RETURN
8360 END
8370C-----
8380 SUBROUTINE DATAP
8390C-----TRANSLATE JODATA(I) INTO IOBUF(I)

```

```

8400 COMMON /IO/ IOBUF(512), IE0F, IB0RT, IPARTY, IDATA(2048), JODATA(1280)
8410 DO 100 I=1,512
8420 IOBUF(I)=0
8430 IF(JODATA(2*I-1).GE.0)GO TO 10
8440 JODATA(2*I-1)=-JODATA(2*I-1)
8450 FLD(0,1,IOBUF(I))=1
8460 10 FLD(1,1,IOBUF(I))=FLD(21,11,JODATA(2*I-1))
8470 FLD(14,4,IOBUF(I))=FLD(32,4,JODATA(2*I-1))
8480 IF(JODATA(2*I).GE.0)GO TO 20
8490 JODATA(2*I)=-JODATA(2*I)
8500 FLD(18,1,IOBUF(I))=1
8510 20 FLD(19,11,IOBUF(I))=FLD(21,11,JODATA(2*I))
8520 100 FLD(32,4,IOBUF(I))=FLD(32,4,JODATA(2*I))
8530 RETURN
8540 END
8550C-----
8560 SUBROUTINE WEIGHT3
8570C-----MODIFY F BY CURVE OF APPROXIMATE RELATIVE IMPORTANCE FOR INTELLIGIBILITY
8580 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
8590 DO 10 I=1,51
8600 XI=I
8610 F(I,1)=F(I,1)*(XI*19.53125)**3/1.E09
8620 10 F(I,2)=F(I,2)*(XI*19.53125)**3/1.E09
8630 DO 20 I=102,257
8640 XI=I
8650 F(I,1)=F(I,1)*4.E06/(XI*19.53125)**2
8660 20 F(I,2)=F(I,2)*4.E06/(XI*19.53125)**2
8670 DO 30 I=2,256
8680 F(514-I,1)=F(I,1)
8690 30 F(514-I,2)=F(I,2)
8700 RETURN
8710 END
8720C-----
8730 SUBROUTINE NORM1(I,JJ,SFACT)
8740C-----CALCULATE SFACT- THE NORMALIZATION FACTOR BASED UPON 40 $ 380
8750 COMMON /IO/ IOBUF(512), IE0F, IB0RT, IPARTY, IDATA(2048), JODATA(1280)
8760 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
8770 OMAX=1.E-37
8780 DO 10 II=1,512

```

```

8810 10 OMAX=AMAX1(OMAX,ABS(F(II,1)))
8820 IF((I.EQ.1).AND.(JJ.FJ.1))GO TO 20
8830 IF(OMAX*SFACT.GT.380.)GO TO 30
8840 IF(OMAX*SFACT.LT.40.)GO TO 40
8850 RETURN
8860 20 SFACT=40./OMAX
8870 RETURN
8880 30 SFACT=SFACT*.6
8890 RETURN
8900 40 SFACT=SFACT*1.2
8910 RETURN
8920 END
9000-----
9010 SUBROUTINE NORW2(I,JJ,N,SFACT)
9020-----CALCULATE SFACT BY UPDATING PREVIOUS VALUE BY .5*(RATIO-SFACT)
9030 COMMON /IO/ IORUF(512),IEOF,IPORT,IPARTY,IData(2048),J0DATA(1280)
9040 COMMON /BUFFER2/ F(512.2),G(512.2),A(512.2)
9050 OMAX=1.F-37
9060 DO 10 II=1,256
9070 XII=II
9080 VAL=(AMAX1(F(II,1),F(513-II,1)))*((257.-XII)/XII)
9090 10 OMAX=AMAX1(OMAX,VAL)
9100 INMAX=0
9110 DO 20 II=N,N+511
9120 20 INMAX=MAX0(INMAX,IData(II))
9130 XI=MAX=INMAX
9140 RATIO=XINMAX/OMAX
9150 IF((I.EQ.1).AND.(JJ.EQ.1))GO TO 30
9160 SFACT=SFACT+.5*(RATIO-SFACT)
9170 RETURN
9180 30 SFACT=RATIO
9190 RETURN
9200 END
9210-----
9220-----TEMP TEST
9230 SUBROUTINE TEST1
9240 COMMON /RUFFER2/ F(512.2),G(512.2),A(512.2)
9250 CALL TTW

```

```

9260 CALL FFT
9270 CALL IFFT
9280 RETURN
9290 END
9300 SUBROUTINE TEST2
9310 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
9320 CALL TTW
9330 CALL FFT
9340 CALL IFFT
9350 CALL IFFT
9360 CALL IFFT
9370 RETURN
9380 END
9400C-----
9410 SUBROUTINE TABLE(N)
9420C-----GENERATE SIN AND COS TABLE FOR FFT & IFFT
9430 COMMON /SCTAB/ S(512)
9440 DO 100 I=1,10
9450 K=2**I
9460 100 IF(N.EQ.K)GO TO 101
9470 STOP
9480C-----ERROR THAT N IS NOT MOD 2 OR N IS GREATER THAN 1024 OR LESS THAN 2
9490 101 K=I
9500 L=2**(K-1)
9510 PI=3.14159
9520 DO 103 J=1,L
9530 P=PI*FLOAT(J-1)/FLOAT(L)
9540 S(J)=SIN(P)
9550 103 CONTINUE
9560 RETURN
9570 END
9580C-----
9590 SUBROUTINE ZLOW2
9600C-----SUPPRESS LOW TEN ELEMENTS TO EQUIL TWICE FORMANT PEAK
9610 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
9620 FMAX=SQRT(F(1,1)**2+F(1,2)**2)
9630 DO 100 I=2,221
9640 100 FMAX=AMAX1(FMAX,SQRT(F(I,1)**2+F(I,2)**2))

```

```

9650 FACT=SQRT(F(257,1)**2+F(257,2)**2)/FMAX
9660 DO 110 I=248,266
9670 F(I,1)=F(I,1)*2/FACT
9680 110 F(I,2)=F(I,2)*2/FACT
9690 RETURN
9700 END
9710C-----
9720 SUBROUTINE ARSVF
9730C-----TAKF ABSOLUTE VALUE OF ARRAY F
9740 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
9750 DO 100 I=1,512
9760 100 F(I,1)=ABS(F(I,1))
9770 RETURN
9780 END
9790C-----
9800 SUBROUTINE NOISUR2(XNOISE)
9810C-----SUBTRACT 3*XNOISE FROM MAG(F)
9815 COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
9820 XNOISE=XNOISE*3.
9830 DO 30 I=1,512
9840 XMAG=SQRT(F(I,1)**2+F(I,2)**2)
9850 XMAG=XMAG-XNOISE
9860 IF(XMAG.GE.O.)GO TO 40
9870 F(I,1)=0.
9880 F(I,2)=0.
9890 GO TO 30
9900 40 F1=XMAG*COS(ATAN2(F(I,2),F(I,1)))
9910 F(I,2)=XMAG*SIN(ATAN2(F(I,2),F(I,1)))
9915 F(I,1)=F1
9920 30 CONTINUE
9930 RETURN
9940 END

```

```

3080C-----
3090      SUBROUTINE NPAP
3100C-----POWER SPECTRAL OPTIMIZATION WITH PAST NOISE WEIGHTING AND SPEECH/
3110C      NO-SPEECH TEST
3120      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
3130      COMMON /START/ INIT(16),RUNWGT(32)
3140      CALL TTW
3150      CALL FFT
3160      CALL ESINQ2(XNOISE)
3170      CALL WEIGH3(XNOISE)
3180      CALL IFFT
3190      RETURN
3200      END
9950C-----
9960      SUBROUTINE WEIGH3(XNOISE)
9970C-----MODIFY POWER SPECTRUM AS IN WEIGH2 BUT WITH ESTIMATED NOISE FROM
9980C      ESTNOISE2
9990      COMMON /BUFFER2/ F(512,2),G(512,2),A(512,2)
00010000C-----CALCULATE ARRAY G, THE LONG TERM AVERAGE FOR NORM. SPEECH
00010010      G(1,1)=0.;G(2,1)=1./2.**8;G(3,1)=1./2.**5
00010020      G(4,1)=1./12.;G(5,1)=.5
00010030      DO 90 I=6,27
00010040  90      G(I,1)=1.
00010050      DO 100 I=28,257
00010060      XI=I
00010070      FREQ=(XI-1.)*19.53125
00010080      100      G(I,1)=(500./FREQ)**2
00010090      SUM=0.
00010100      DO 110 I=1,257
00010110      110      SUM=SUM+G(I,1)
00010120      SUM=SUM/257.
00010130C-----CALCULATE AVERAGE OF MAG(F)
00010140      FSUM=0.
00010150      DO 115 I=1,257
00010160      115      FSUM=FSUM+SQRT(F(I,1)**2+F(I,2)**2)
00010170      FSUM=FSUM/257.
00010180      DO 130 I=1,257
00010190      130      G(I,1)=G(I,1)*FSUM/SUM

```

```

00010200      DO 140 I=2,256
00010210      140      G(514-I,1)=G(I,1)
00010220      DO 150 I=1,512
00010230      RE=F(I,1)
00010240      XIM=F(I,2)
00010250      XMAG2=F(I,1)**2+F(I,2)**2
00010260      XMAG=(G(I,1)**2+XMAG2)/(G(I,1)**2+XNOISE**2)
00010270      F(I,1)=XMAG*COS(ATAN2(XIM,RE))
00010280      150      F(I,2)=XMAG*SIN(ATAN2(XIM,RE))
00010290      RETURN
00010300      END
00010310-----
00010320      SUBROUTINE ESTNO2(XNOISE)
00010325C-----ESTIMATE AVERAGE NOISE LEVEL FROM 50 PAST SAMPLES
00010330      COMMON /BUFFER2/ F(512,2),3(512,2),A(512,2)
00010340      COMMON /START/ INIT(16),RUNWGT(32)
00010350C-----CALCULATE AVERAGE SPECTRUM FOR EACH FRAME
00010360      SUMA=0.
00010370      DO 10 I=2,257
00010380      10      SUMA=SUMA+SQRT(F(I,1)**2+F(I,2)**2)
00010390      AVER=SUMA/256.
00010400C-----CALCULATE VARIANCE FOR EACH FRAME
00010410      SUMV=0.
00010420      DO 11 I=2,257
00010430      11      SUMV=SUMV+(SQRT(F(I,1)**2+F(I,2)**2)-SUMA)**2
00010440      VAR=SQRT(SUMV)/256.
00010450C-----INIT(1) IS FRAME COUNT TO 50
00010460      IF(INIT(1).NE.1)GO TO 15
00010465      INIT(1)=INIT(1)+1
00010470C-----RUNWGT(1) IS AVERAGE SUMA FOR SPEECH
00010480C-----RUNWGT(2) IS AVERAGE SUMA FOR NOISE
00010490C-----INIT(2) IS 1-SPEECH FRAME, 0-NOISE FRAME
00010500      RUNWGT(1)=SUMA
00010510      RUNWGT(2)=SUMA-.1*SUMA
00010520      INIT(2)=1
00010530      XNOISE=RUNWGT(2)
00010540C-----RUNWGT(3) IS AVERAGE SPEECH VARIANCE
00010550C-----RUNWGT(4) IS AVERAGE NOISE VARIANCE

```

```

00010560C-----RUNWGT(5) IS THRESHOLD FOR NOISE-SPEECH
00010570  RUNWGT(3)=SUMV
00010580  RUNWGT(4)=SUMV
00010590  RUNWGT(5)=.92*SUMA
00010600  RETURN
00010610C-----UPDATE RUNNING AVERAGES FOR THRESHOLD
00010620 15 IF (INIT(1).GT.50)GO TO 12
00010630  INIT(1)=INIT(1)+1
00010650 12 IF (SUMA.GT.RUNWGT(5))GO TO 13
00010660  K1=2
00010670  K2=4
00010680  INIT(2)=0
00010690  GO TO 14
00010700 13 K1=1
00010710  K2=3
00010720  INIT(2)=1
00010725 14 FINIT=FLOAT(INIT(1))
00010730  RUNWGT(K1)=(FINIT-1.)*RUNWGT(K1)/FINIT+SUMA/FINIT
00010740  RUNWGT(K2)=(FINIT-1.)*RUNWGT(K2)/FINIT+SUMV/FINIT
00010750  RATIO=RUNWGT(1)/(RUNWGT(1)+RUNWGT(2))
00010760  RUNWGT(5)=RUNWGT(1)-1.2*(RUNWGT(1)-RUNWGT(2))*RATIO
00010770  XNOISE=RUNWGT(2)
00010780  RETURN
00010790  END

```

ready

**ATE
LMED**