TUAN, Le-ba, 1939-
COMPUTER DESIGN AND EVALUATION OF OPTIMAL FILTERS
FOR A CLASS OF SPEECH SOUNDS BY CORRELATION
TECHNIQUES.

The Ohio State University, Ph.D., 1971
Engineering, electrical

University Microfilms, A XEROX Company, Ann Arbor, Michigan

© Copyright by
Le-ba Tuan
1971
COMPUTER DESIGN AND EVALUATION OF OPTIMAL FILTERS
FOR A CLASS OF SPEECH SOUNDS
BY CORRELATION TECHNIQUES

Dissertation

Presented in partial fulfillment of the requirement for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By
Le-ba Tuan, B.S.E.E., M.S.E.E.

The Ohio State University
1971

Approved by

[Signature]
Adviser
Department of Electrical Engineering
PLEASE NOTE:

Some pages may have
indistinct print.
Filmed as received.

University Microfilms, A Xerox Education Company
ACKNOWLEDGEMENTS

This research is dedicated to all of those people of my country who have fought courageously in the battle fields in order for me to have the free mind and the opportunity for this research. It is also dedicated to my family in VietNam and to my wife Crescencia for their unfailing support, encouragement and immeasurable sacrifices.

I am grateful to my advisor, Professor Herman R. Weed for his guidance and timely encouragements throughout my doctoral program, and for his careful review of this work. I would like to acknowledge the topic of research generated by the Bio-Medical Engineering Center of The Ohio State University. I would like also to express my most sincere appreciation to Professor Richard M. Campell and Professor Robert B. Lackey for their cheerful accepting to be on my Ph. D. committee and for their advice on the writing of the final work.

I am also grateful to Professor Frank E. Battocletti for his introducing me to the wonderful world of on-line computers, and to Professor Claude E. Warren for his valuable teaching and advising me about the Macro Language of the PDP-9, and for his FFT subroutines.

I am indebted to Professor R.W. Mann of MIT and to Professor J.M. Pickett of the Gallaudet School of Deaf in Washington D.C. for their providing me with valuable reviews of related works around the world, and to my office mates, Mssrs Roberto Marletti and Ram Engira for their suggestions and their reading of the zeroth draft.

My special thanks go to the ladies of the office of the Electrical
Engineering, and Mrs Sylvia Lewchuk for their numerous assistances at critical moments in typing and reproducing of this work. I am also indebted to Mssrs Tom Mc Cann, Dave Ossing and their associates for constant support in hardware building and in materials. Thanks are also due to Mr Henry Pagean for his fine photographic works.

The program ending with this research was supported by the Agency for International Development of the United States in cooperation with the Government of the Republic of VietNam.
VITA

May 5, 1939 Born - Phan-Thiet, VietNam

1962 B.S.E.E., The National Technical Center, Saigon, VietNam

1964 M.S.E.E., Bradley University, Peoria, Illinois U.S.A.


1967-1971 Doctoral program at The Ohio State University, Columbus, Ohio, U.S.A.

FIELDS OF STUDY

Major Field: Electrical Engineering

Studies in Control Theory. Professor J. Bacon, Professor J. Cowan

Studies in Computer Logical Design. Professor J. McMillan, Professor K.G. Breeding

Studies in Coding Theory. Professor R.B. Lackey

Studies in Industrial Electronics. Professor Herman R. Weed

Studies in Fields, Communication and Circuits. Professor R.C. Rudduck, Professor C.E. Warren, Professor E.M. Kennaugh

Studies in Quantum Mechanics. Professor R.B. Sanderson

Studies in Mathematics. Professor Stefen Drobot
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>VITA</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF ILLUSTRATIONS</td>
<td></td>
</tr>
<tr>
<td>Chapter I</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. Definition of problem</td>
<td>5</td>
</tr>
<tr>
<td>Chapter II</td>
<td></td>
</tr>
<tr>
<td>1. Theoretical Background</td>
<td>9</td>
</tr>
<tr>
<td>2. Correlation</td>
<td>12</td>
</tr>
<tr>
<td>3. Autocorrelation</td>
<td>14</td>
</tr>
<tr>
<td>4. Crosscorrelation</td>
<td>18</td>
</tr>
<tr>
<td>5. The Characteristic Feature Extraction</td>
<td>23</td>
</tr>
<tr>
<td>6. The Optimum Decision Problem</td>
<td>24</td>
</tr>
<tr>
<td>7. Review of Current Literature</td>
<td>26</td>
</tr>
<tr>
<td>Chapter III</td>
<td></td>
</tr>
<tr>
<td>1. Hardware and Software Involved</td>
<td>31</td>
</tr>
<tr>
<td>2. Summary on Hardware and Software</td>
<td>47</td>
</tr>
<tr>
<td>Chapter IV</td>
<td></td>
</tr>
<tr>
<td>1. Proposed Investigation</td>
<td>48</td>
</tr>
<tr>
<td>2. Report on Investigation and Discussion</td>
<td>49</td>
</tr>
<tr>
<td>Chapter V</td>
<td></td>
</tr>
<tr>
<td>Recommendation on:</td>
<td></td>
</tr>
<tr>
<td>1. Computer Program Refinements</td>
<td>96</td>
</tr>
<tr>
<td>2. Filter Implementation</td>
<td>97</td>
</tr>
</tbody>
</table>
3. Applications of the Research Results ... 97

Chapter VI

1. Reference ......................... 100
2. Appendices ....................... 102
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>36</td>
</tr>
<tr>
<td>5</td>
<td>59</td>
</tr>
<tr>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>7</td>
<td>61</td>
</tr>
<tr>
<td>8</td>
<td>62</td>
</tr>
<tr>
<td>9</td>
<td>63</td>
</tr>
<tr>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>11</td>
<td>65</td>
</tr>
<tr>
<td>12</td>
<td>66</td>
</tr>
<tr>
<td>13</td>
<td>67</td>
</tr>
<tr>
<td>14</td>
<td>68</td>
</tr>
<tr>
<td>15</td>
<td>69</td>
</tr>
<tr>
<td>16</td>
<td>70</td>
</tr>
<tr>
<td>17</td>
<td>71</td>
</tr>
<tr>
<td>18</td>
<td>72</td>
</tr>
<tr>
<td>19</td>
<td>73</td>
</tr>
<tr>
<td>20</td>
<td>74</td>
</tr>
<tr>
<td>21</td>
<td>75</td>
</tr>
<tr>
<td>22</td>
<td>76</td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>23</td>
<td>77</td>
</tr>
<tr>
<td>24</td>
<td>78</td>
</tr>
<tr>
<td>25</td>
<td>79</td>
</tr>
<tr>
<td>26</td>
<td>80</td>
</tr>
<tr>
<td>27</td>
<td>81</td>
</tr>
<tr>
<td>28</td>
<td>82</td>
</tr>
<tr>
<td>29</td>
<td>83</td>
</tr>
<tr>
<td>30</td>
<td>84</td>
</tr>
<tr>
<td>31</td>
<td>85</td>
</tr>
<tr>
<td>32</td>
<td>86</td>
</tr>
<tr>
<td>33</td>
<td>87</td>
</tr>
<tr>
<td>34</td>
<td>88</td>
</tr>
<tr>
<td>35</td>
<td>89</td>
</tr>
<tr>
<td>36</td>
<td>90</td>
</tr>
<tr>
<td>37</td>
<td>91</td>
</tr>
<tr>
<td>38</td>
<td>92</td>
</tr>
<tr>
<td>39</td>
<td>93</td>
</tr>
<tr>
<td>40</td>
<td>94</td>
</tr>
<tr>
<td>41</td>
<td>181</td>
</tr>
<tr>
<td>42</td>
<td>182</td>
</tr>
<tr>
<td>43</td>
<td>183</td>
</tr>
<tr>
<td>44</td>
<td>184</td>
</tr>
<tr>
<td>45</td>
<td>185</td>
</tr>
<tr>
<td>46</td>
<td>186</td>
</tr>
<tr>
<td>47</td>
<td>187</td>
</tr>
</tbody>
</table>
CHAPTER I

1. INTRODUCTION

The object of this research effort is to investigate the possibility of specifying the necessary characteristics of a set of optimal filters which could classify, within acceptable limits, a definite group of speech sounds. This is a special class of pattern recognition restricted to speech forms.

The main task resolves into the investigation of the specification of the characteristic features of the above mentioned group of speech sounds as spoken individually by a group of people. This sample of people should be as large as practically possible, and composed of people of different ages of both sexes. Furthermore, the methodology should be general enough to permit an attempt to generalize the results beyond the selected group of speech sounds in question. For initial testing and evaluation, the standard group of ten American pure vowels was chosen as the speech elements to be studied. Using the International Phonetic Association symbols, they are listed below:

<table>
<thead>
<tr>
<th>Vowel</th>
<th>Key Word</th>
<th>Vowel</th>
<th>Key Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>/i/</td>
<td>team</td>
<td>/o/</td>
<td>tone</td>
</tr>
<tr>
<td>/I/</td>
<td>tip</td>
<td>/U/</td>
<td>took</td>
</tr>
<tr>
<td>/e/</td>
<td>ten</td>
<td>/u/</td>
<td>tool</td>
</tr>
<tr>
<td>/æ/</td>
<td>tap</td>
<td>/ʌ/</td>
<td>ton</td>
</tr>
<tr>
<td>/ɔ/</td>
<td>talk</td>
<td>/a/</td>
<td>tar</td>
</tr>
</tbody>
</table>
If the investigation of the characteristic features of these pure vowels proves successful, the next step will be to investigate these vowels linked with ending consonants such as found in the key words given along with the vowels above. As it can be seen in the next few chapters, the evaluation of the method of feature extraction is done in the time domain, before these characteristic features are transferred to the frequency domain by classical links such as the Fourier Transforms. We will see then what can be done with the methodology developed applied to words starting with consonants, as these are more of a transient nature.

The problem at hand fits into a more general problem of speech recognition, which in turn, belongs to the general class of problems defined as pattern recognition by machines.

Through the ages, each generation tries to design (or proposes to design) machines for some kind of pattern recognition relying on available technology of the age. The standard against which people often would check those machines is, of course, a human being. Human recognition of patterns may be regarded as a psycho-phyziological problem involving a physical stimulus. When a person perceives a pattern, he is making an inductive inference and associating that perception with some general concept or clues which he has derived from his past experiences. Thus, human pattern recognition may be considered as the problem of discriminating the input data, not between individual patterns, but between populations via the search for characteristic features or invariants, among members of a population.

Our generation has seen the invention of one of the best tools for pattern recognition, the digital computer, with all the equipment available with it in an up-to-date installation.
The state-of-the-art dictates that we build on pattern recognition systems around this tool, at least at the investigation stage for best flexibility and speed.

In general, the design of pattern recognition systems involves several major problems areas. The first problem is concerned with the representation of input data which can be measured by the members of the class to be recognized. This is the sensing problem. The second one is the selection of characteristic features from the received input data. This is called the feature extraction, or selection problem. The third deals with the determination of optimum decision procedures needed in the process of identification. This is the optimum decision problem. In solving the feature extraction and the optimum decision problem, a set of parameters is generally to be estimated and optimized. This gives rise to the parameter estimation problem. By taking into account contextual information in the patterns, the feature selection and decision process may be improved. We face the problem of adaptation if we want our pattern recognition system to resist distortion, follow pattern deviations, or to be self-adjustable.

Among all of those problems, the characteristic feature extraction has been recognized as the most important problem in a pattern recognition system. If the complete set of discriminatory features for each pattern class can be determined from the measurements, the recognition and classification of the patterns present no major problem. However, in most pattern recognition systems in practice, except for some very simple cases, this determination of complete discriminatory features is extremely difficult if not impossible.

The problem at hand is not, per se, a problem of pure design of a
system for the purpose of pattern recognition in speech. Rather, it is one of designing a system of pattern recognition for the purpose of checking the feasibility in the design specification of a system of filters which ultimately will be used in the identification of a class of speech sounds within some definite criteria. These criteria and the dimension of the class of speech sounds for which the optimal filters will be built, are necessarily limited to some reasonable attainments that one can best achieve within the scope of the present state-of-the-art.

The general considerations in pattern recognition and our computer investigation point out the fact that unless the system which ultimately uses the optimal filters that this study helps design includes some kind of adaption process to take care of the variations in pronunciation coming from ethnic, subcultural, regional, dialetic differences, etc., the chance of success in recognizing completely all the sounds in the class, spoken by a large group of people coming from these sources of differences will be dim. The built-in learning capability of the computer simulation of this study has shown that minor adjustments to a new voice make the simulated set of filters accommodate a fairly large group* of people of both sexes, speaking at different pitches involuntarily or voluntarily on request. These people are of different ages, and nationalities. It is true that at the stage of testing the simulated filters for pure vowels everybody required some training and practice to pronounce clearly the subtle differences between pure vowels. It appeared easier for most people to pronounce the words at the stage of testing that simulated filter set for the sounds with vowels plus

*4 men, 2 women. (This group will be enlarged to include also children)
consonants at the end.

2. DEFINITION OF PROBLEM

The question of what parts of speech are most important in the transfer of information and recognition has been investigated through many approaches utilizing filters or other frequency-spectra methods.\(^\text{(12)}\) It appears that the general communication theory approach of correlation utilizing modern computer technology may be more productive. This research involves such a study and as a first phase, required the development of a fast and economical algorithm applicable to less expensive computers.\(^\text{(17)}\) The next phase involved the use of correlation functions in the time domain to extract the characteristics features of the group of speech sounds chosen for study. This phase included also the simulation on the computer of a set of filters derived from those correlation functions, for the purpose of testing and evaluating the system which will use such a set of filters. A learning process was built into the simulation for investigation of adaptivity of that system to new inputs. The last phase involved the use of fast algorithms for Fourier Transforms (Fast Fourier Transform, \(^\text{(8)}\) Cepstrum \(^\text{(8)}\)) to transfer the characteristic features of that group of sounds into the frequency domain in the form of power density spectrums.

It is hoped that this study will contribute to many useful applications in communication with the deaf and speech recognition.

A follow-up effort will be the hardware implementation of a set of filters from the above data and the construction of a maximum likelihood simple receiver; the simplest form would involve a very few syllables such as the long or short vowel sounds with a pair of consonants. The receiver would be a properly amplified microphone followed by three sets
of filters and a display. One of the set of filters is for beginning consonants, one for vowels and one for the ending consonants, Figure 1 depicts such a set up.

Fig. 1  Filter set A is for Beginning Consonants;
Filter set B is for Vowels;
Filter set C is for Ending Consonants;
D's are Displays (Characters, Scope, Meter, etc...)
Another type of setup (14) would combine selected filter characteristics of group A with selected filter characteristics of group B and C to produce only one column of combination filters which would recognize the whole word composed of a vowel between two consonants. Figure 2 shows such a system.

Fig. 2. Setup for Recognition of the whole Syllable.
Filters 1, 2, ..., n, are Combination Filters made up of selection of Filters Characteristics from the three Groups of Filters A, B and C of Fig. 1.
D's are Displays (Characters, Scope, Meters, etc...)
In either set up, the output indication from the filter most nearly matching the spectrum of the sound spoken will be of higher amplitude than that from the other filters, and thus permits a decision to be made regarding the sound spoken which would be reconstructed as a syllable in the set up of Figure 1 and directly displayed as a whole syllable in the set up of Figure 2.

Another application which is potentially important in continuous speech analysis will be discussed further in Chapter V.
CHAPTER II

1. THEORETICAL BACKGROUND

Review of literature covering current researches in speech shows that the determination of a set of completely discriminatory features has not yet been accomplished. However, it may be found frequently that some of the discriminatory features are sufficiently characteristic to direct the attention to possible fruitful direction of investigation.

Many researchers have pointed out the fact that human ears are insensitive to phase variations, within large limits, among the frequency components that make up the sound that the ears perceive. This fact has been checked also in the communication systems using the channel vocoder which essentially analyze the sounds to be transmitted in the frequency spectrum and then resynthesize these sounds at the receiving end by summing up the frequency components. In the process, the phase relationship among the component frequencies are completely ignored. Yet the speech received at the receiving end is almost as intelligible as that at the sending end. It is reported that even in the best systems using the channel vocoders, people still can detect a slight unnaturalness in the speech at the receiving end. (2) However, this unnaturalness is attributed to some other factors but not to the dropping of the phase relationships among components frequencies. As far as recognizing speech sounds for the purpose of teaching the deaf or inputting the computers is concerned, this unnaturalness—which is termed machined-like quality—is even desirable. The set of filters that we investigate the feasibility of,
is in fact, to treat ultimately the set of input data to be recognized as impersonal, free from the characteristic particularities of the group of people producing this set of input. Thus the road to frequency analysis is going to be followed in our work here. However, a heuristic examination of the possibility of using the techniques of averaging such as summation in the frequency domain shows that with the present technology the equipment required for such an investigation lacks either speed, flexibility, resolution or adaptivity. We must then find an equivalent way of doing the frequency analysis which is more desirable. The correlation techniques in the time domain provide precisely such an equivalent and permit us to simulate completely a set of filters on a medium-sized computer with all the speed, flexibility, resolution and adaptivity needed to carry out systematically and with reasonable speed, the investigation about the feasibility of such a set of filters. After the investigation has been done in the time domain we will use the time-frequency links which are the Fourier Transforms, to specify completely the characteristic amplitude-vs-frequency of such a set of filters. In recent years, the computer users have been overwhelmed by the introduction of a new short-cut algorithm that allows to compute the Fourier Transform coefficients with considerably higher speed than the brute force approach inspired by classical books. That algorithm has been called the Fast Fourier Transforms or FFT which approximates very well the Discrete Fourier Transforms which in turn are the digital computer approximations of the continuous Fourier Transforms. After the FFT another interesting entity called the Cepstrum \((8)\) was invented. This Cepstrum is obtained as follows:

a) Compute the spectrum \( F (w) \) (power density or amplitude spectrum)
of a given function of time \( f(t) \) by FFT;

b) Evaluate the db spectrum from \( F(w) \); and call it \( G(w) \).

c) Imagine \( G(w) \) as a fictive function of time \( g(t) \) with the x-axis representing time;

d) In as much as \( g(t) \) is a function of time now, we can take its Fourier Transform \( G_1(w_1) \) which would be the fictive frequency \( (w_1) \) spectrum of \( g(t) \).

e) This \( G_1(w_1) \) is defined as the Cepstrum of \( f(t) \).

One of the uses of the Cepstrum is the smoothering of the spectrum computed in part b/. It is done as follows:

f) Multiply the high frequency end of \( G_1(w_1) \) by a window function which tends to zero at a point within our control to produce a modified ceptrum \( G_2(w_1) \) which has no more of those high frequencies which were the cause of the rapid oscillations in its counterpart, \( g(t) \), which was actually a point-by-point replica of \( G(w) \).

g) Use FFT to inverse-transform \( G_2(w_1) \) to obtain \( g_1(t) \) which is a smoothened version of \( g(t) \).

h) Make the transformation inverse to that discussed in part c to get a smoothened version of \( G(w) \).

This feature will be built into the computer program to produce, as end products, physically realizable filters (not necessarily in the sense used in network synthesis, but in the broad sense of realization by some combinations of optical, mechanical or electronic devices).

In order to justify the time and frequency equivalence discussed above, we have to go back to some of the notions about statistical theory of communication dealing specially with what interests most here: The Correlation Techniques. (5)
2. CORRELATION

Let $f_1(t)$ and $f_2(t)$ have the same fundamental angular frequency $w_1 = 2\pi/T$. An expression of considerable importance and interest is:

$$\phi_{12}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} f_1(t) f_2(t + \tau) \, dt$$  \hspace{1cm} (II-2-1)

Where $\tau$ is a continuous time displacement in the range $( -\infty, +\infty )$, independent of $t$.

One important property of this expression is the fact that if Fourier Transform is:

$$\tilde{F}_1(w) \tilde{F}_2(w)$$  \hspace{1cm} (II-2-3)

if $f_1(t)$ has the complex spectrum $F_1(w)$, and, $f_2(t)$, $F_2(w)$. The bar on top of $F_1(w)$ in (2-2) indicates the conjugate (of the quantity with bar on it) should be taken. The property expressed in (2-2) can be seen through the steps below:

Let

$$f_1(t) = \sum_{n=-\infty}^{\infty} F_1(nw_1) e^{jnw_1t}$$  \hspace{1cm} (II-2-3)

and

$$f_2(t) = \sum_{n=-\infty}^{\infty} F_2(nw_1) e^{jnw_1t}$$  \hspace{1cm} (II-2-4)

where $w = nw_1$, with $w_1$ defined above, and where:

$$F_1(w) = \frac{1}{T} \int_{-T/2}^{T/2} f_1(t) e^{-jwt} \, dt$$  \hspace{1cm} (II-2-5)

and

$$F_2(w) = \frac{1}{T} \int_{-T/2}^{T/2} f_2(t) e^{-jwt} \, dt$$  \hspace{1cm} (II-2-6)

Substituting

(II-2-4) into (II-2-1) we have

$$\frac{1}{T} \int_{-T/2}^{T/2} f_1(t) f_2(t + \tau) \, dt = \frac{1}{T} \int_{-T/2}^{T/2} f_1(t) dt \sum_{n=-\infty}^{\infty} F_2(nw_1) e^{jnw_1t}$$  \hspace{1cm} (II-2-7)

A formal inversion of the order of summation and integration gives

$$\frac{1}{T} \int_{-T/2}^{T/2} f_1(t) f_2(t + \tau) \, dt = \sum_{n=-\infty}^{\infty} F_2(nw_1) e^{jnw_1} \frac{1}{T} \int_{-T/2}^{T/2} f_1(t) e^{jnw_1t} dt$$  \hspace{1cm} (II-2-8)
Where we also replace explicitly \( w \) by \( nw_1 \). By comparizing with the definition of \( F_1(w) \) and \( F_2(w) \), we see that the last integral in (II-2-8) is nothing else than the complex conjugate of \( F_1(w) \). Thus

\[
\frac{1}{T} \int_{-T/2}^{T/2} f_1(t) f_2(t + \tau) \, dt = [\bar{F}_1(nw_1) F_2(nw_1)] e^{jnwl} \tag{II-2-9}
\]

\[
= [F_1(w) F_2(w)] e^{jnwl} \tag{II-2-10}
\]

Recalling that the Fourier Transform pair for a periodic function of fundamental angular frequency \( w_1 \) is expressed as

\[
f(t) = \sum_{n=-\infty}^{\infty} F(nw_1) e^{jnwt} \tag{II-2-12}
\]

and

\[
F(nw_1) = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-jnwt} \, dt \tag{II-2-12}
\]

We can see that equation (II-2-9) expresses the fact that:

\[
\frac{1}{T} \int_{-T/2}^{T/2} f_1(t) f_2(t + \tau) \, dt
\]

has as transform the expression \([\bar{F}_1(nw_1) F_2(nw_1)]\).

To compute the value of \( \phi_{12}(\tau) \), 3 operations have to be performed:

1. One of the periodic functions concerned \( f_2(t) \), is displaced by a time \( \tau \);

2. The displaced function is multiplied by the other periodic function of the same fundamental frequency at every point of \( t \);

3. The product is averaged out with integration over the whole period. Each time these 3 above steps are performed, one value of \( \phi_{12}(\tau) \) is obtained. To obtain a curve \( \phi_{12}(\tau) \) vs \( \tau \), these steps are to be repeated for every value of \( \tau \) of interest in the interval of interest which could be from \( -\infty \) to \( +\infty \).

The operation performed by the steps above namely displacement, multiplication and integration, is defined as correlation. If \( f_1(t) \) and \( f_2(t) \) are different, the result is written as \( \phi_{12}(\tau) \), and called
cross-correlation between \( f_1(t) \) and \( f_2(t) \). If \( f_1(t) \) and \( f_2(t) \) are the same, the result is written as \( \phi_{11}(\tau) \) or \( \phi_{22}(\tau) \) and called auto-correlation. As \( \phi_{12}(\tau) \) and \( \phi_{11}(\tau) \) or \( \phi_{22}(\tau) \) are functions of \( \tau \) they are called also alternately cross-correlation functions and auto-correlation functions.

3. AUTOCORRELATION

In equation (II-2-9), if \( f_1(t) \) and \( f_2(t) \) are the same we obtain the form:

\[
\frac{1}{T} \int_{-T/2}^{T/2} f_1(t) f_1(t + \tau) \, dt = \sum_{n=-\infty}^{\infty} \frac{1}{F_1(nw_1)} e^{jnw_1 \tau} \quad (II-3-1)
\]

The left-hand member of equation (II-3-1) is defined as the autocorrelation function of the function \( f_1(t) \) and usually designated as \( \phi_{11}(\tau) \) thus;

\[
\phi_{11}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} f_1(t) f_1(t + \tau) \, dt \quad (II-3-2)
\]

The expression \( /F_1(nw_1)/^2 \) is denoted as the power spectrum, so that, if we use \( \phi(nw_1) \) to called it, we will have

\[
\phi_{11}(nw_1) = /F_1(nw_1)/^2 \quad (II-3-3)
\]

With the new symbols \( \phi_{11}(\tau) \) and \( \phi(nw_1) \) we can have equation (II-3-1) in the forms

\[
\phi_{11}(\tau) = \sum_{n=-\infty}^{\infty} \phi_{11}(nw_1)e^{jnw_1 \tau} \quad (II-3-4)
\]

and

\[
\phi_{11}(nw_1) = \frac{1}{T} \int_{-T/2}^{T/2} \phi_{11}(\tau) e^{-jnw_1 \tau} \, d\tau \quad (II-3-5)
\]

Identifying with the familiar Fourier Transform Pair, we can say from equation (II-3-4) and (II-3-5) that the autocorrelation function and the power spectrum of a periodic function are Fourier Transforms of each other, and given one, the other is uniquely determined.

One of the most important features that make the auto-correlation
functions attractive to our work is that the autocorrelation functions retain from the original function \( f(t) \) the characteristics of frequency and amplitude only, as all the phases between the different components are dropped completely since they are made equal to zero. This fact can be seen in the light of the Fourier Transform of the autocorrelation function, which is the power spectrum. From equation (II-3-3) we can see that it is the square of the amplitude spectrum. Thus, even if the amplitude spectrum contains an imaginary part along with the real one, the squaring of the two parts, real and imaginary of the spectrum giving the power spectrum makes this latter necessarily to have only real parts and discard thus all the phases by making them equal to zero.

What this means to us is that if \( f(t) \) are the speech sounds to be analyzed, all the sounds having the same harmonic content as far as frequency and amplitude are concerned, but different in phase arrangement between the different components, will have exactly the same power spectrum. As we have shown above that the power spectrum has its unique Fourier Transform, which is the autocorrelation function, this latter is also the same for all of those sounds. This is a valuable property of the autocorrelation for our work if we want to use this function for pattern recognition in the stage of evaluation in the design of our set of filters.

Practice has shown that sounds which sound the same to the ears and have the same frequency content can look vastly different in their primary form of representation by amplitude-vs-time, because of the different phase arrangements among different components of frequency. However, when these sounds are processed to give their autocorrelation functions, these functions emerge always the same, within practical
delays \( \tau \) that we can determine. If we use these autocorrelation functions as stored patterns and as input data, there is no problem of alignment before we can begin the recognition process by cross-correlation at zero delay or by Euclidean Metric that we will talk about later. Even with the most elaborate scheme of switching that we can build, on the other hand, it would be an impossible dream of pattern recognition if we tried to store as patterns and read in as input for recognition speech sounds represented in the form of amplitude-vs-time, without further processing before recognition takes place. This, simply, is because the phases would make these patterns greatly different even if they are for the same sound.

The dropping of phase in the autocorrelation can be seen also by examining the properties in the time domain of that function.

A simple change of variable in the equation (II-3-2) tells us that \( \phi_{11}(\tau) \) is an even function of \( \tau \). That is:

\[
\phi_{11}(-\tau) = \phi_{11}(\tau)
\]

Equations (II-3-4) is then completely definable by its cosine series expansion with zero initial phase angle. Thus equation (II-3-4) and (II-3-5) have as alternate forms

\[
\phi_{11}(\tau) = \sum_{n=-\infty}^{\infty} \phi_{11}(nw_1) \cos nw_1\tau
\] (II-3-6)

and

\[
\phi_{11}(nw_1) = \frac{1}{T} \int_{-T/2}^{T/2} \phi_{11}(\tau) \cos nw_1\tau \, d\tau
\] (II-3-7)

To get the autocorrelation function in terms of the coefficients of the Fourier expansion of the original \( f(t) \), we write

\[
\phi_{11}(\tau) = \phi_{11}(0) + 2 \sum_{n=-\infty}^{\infty} \phi_{11}(nw_1) \cos nw_1\tau
\] (II-3-8)
Usually \( f_1(t) \) expansion has the form

\[
f_1(t) = \frac{a_{10}}{2} + \sum_{n=-\infty}^{\infty} (a_{1n} \cos n\omega_1 t + b_{1n} \sin n\omega_1 t)
\]

and also, for \( F_1(n\omega_1) \),

\[
F_1(n\omega_1) = \frac{1}{2} (a_{1n} - jb_{1n})
\]

We then have

\[
\phi_{11}(n) = \frac{a_{1n}^2 + b_{1n}^2}{4} \quad \text{for } n = 0, \pm 1, \pm 2, \ldots \tag{II-3-11}
\]

and thus

\[
\phi_{11}(\tau) = \frac{a_{10}^2}{4} + \frac{1}{2} \sum_{n=-\infty}^{\infty} (a_{1n}^2 + b_{1n}^2) \cos n\omega_1 \tau \tag{II-3-12}
\]

From this last equation we see clearly that the autocorrelation function is a cosine series with zero phase for all harmonics present in the given function and with coefficients equal to the mean square values of the corresponding harmonics in the given function, and this applies, too, to the constant term (of frequency zero). This property also gives us the relatively complete freedom of beginning anywhere on a sample of speech sound entered into the computer memory via an A/D-converter to compute the autocorrelation function. This function will be the same as long as the sample is taken long enough. In practice the length of this sample is determined by an option in the computer programming called CORREC which permits to examine on the scope a sample of length of 512 points and then to call anywhere in that sample the time origin and to take any number of points up to 256 for correlation. A sample of 256 points taken at a sampling rate of 8196 samplings per second (about 1/32 second) is sufficient to guarantee consistent autocorrelation functions obtained with the same sound no matter where we begin to take in the data as long as the sound is sustained throughout the period of sampling (about 31 ms). This length of sample also has proved to be long enough to give meaningful results when a transformation into the
frequency domain is made. Yet, it is short enough to make direct

correlations in time domain still practical without making the different

schemes of investigation excessively tedious.

In order to be able to investigate the possibility of eliminat­
ing personal characteristics in a sound pronounced by several people by

repeatedly cross-correlating speech sound samples we will now look into

the cross-correlation function more closely to guide ourselves as to

how to cross-correlate the data when we get samples of the same sound
to design an optimal filter for, spoken by several people.

4. CROSS-CORRELATION

In the correlation defined by equation (II-2-1), if \( f_1(t) \) and

\( f_2(t) \) are different periodic functions of the same fundamental frequency,

we define as cross-correlation function of \( f_1(t) \) and \( f_2(t) \) the

expression

\[
\phi_{12}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} f_1(t) f_2(t + \tau) \, dt
\]  

(II-4-1)

and as a cross-power spectrum the function

\[
\phi_{12}(nw_1) = \overline{F_1(nw_1)} F_2(nw_1)
\]  

(II-4-2)

The order of appearance of subscripts in \( \phi_{12}(\tau) \) follows that

appearing in the second member of equation (II-4-1).

Same way with \( \phi_{21}(\tau) \),

\[
\phi_{21}(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} f_2(t) f_1(t + \tau) \, dt
\]  

(II-4-3)

and

\[
\phi_{21}(nw_1) = \overline{F_2(nw_1)} F_1(nw_1)
\]  

(II-4-4)

In general, \( \phi_{12}(\tau) \) is different from \( \phi_{21}(\tau) \) and

\( \phi_{12}(nw_1) \) from \( \phi_{21}(nw_1) \), although they are related to each other in a

simple manner. By changing \( \tau \) into \( -\tau \) and letting \( x = t - \tau \)
in equation (II-4-1) it can be shown with a few mathematical steps that (5)

$$\phi_{12} (-\tau) = \phi_{21} (\tau)$$

(II-4-5)

and

$$\phi_{12} (nw_1) = \bar{\phi}_{21} (nw_1)$$

(II-4-6)

We have also the transform pairs in the fashion of equations (II-3-6) and (II-3-7) above:

$$\phi_{12} (\tau) = \sum_{n=-\infty}^{\infty} \phi_{12} (nw_1) e^{jnwl \tau}$$

(II-4-7)

and

$$\phi_{12} (nw_1) = \frac{1}{T} \int_{-T/2}^{T/2} \phi_{12} (\tau) e^{-jnwl \tau} d\tau$$

(II-4-8)

and similarly

$$\phi_{21} (\tau) = \sum_{n=-\infty}^{\infty} \phi_{21} (nw_1) e^{jnwl \tau}$$

(II-4-9)

and

$$\phi_{21} (nw_1) = \frac{1}{T} \int_{-T/2}^{T/2} \phi_{21} (\tau) e^{-jnwl \tau} d\tau$$

(II-4-10)

For a formula giving the cross correlation function in terms of the Fourier coefficients of the given periodic function $f_1 (t)$ and $f_2 (t)$, we consider

$$f_1 (t) = \frac{a_{10}}{2} + \sum_{n=-\infty}^{\infty} (a_{1n} \cos nw_1 t + b_{1n} \sin nw_1 t)$$

(II-4-11)

and

$$f_2 (t) = \frac{a_{20}}{2} + \sum_{n=-\infty}^{\infty} (a_{2n} \cos nw_2 t + b_{2n} \sin nw_1 t)$$

(II-4-12)

Rewriting equation (II-4-7) we get

$$\phi_{12} (\tau) = \sum_{n=-\infty}^{\infty} F_1 (nw_1) F_2 (nw_1) e^{jnwl \tau}$$

(II-4-13)

where

$$F_1 (nw_1) = \frac{a_{1n} - jb_{1n}}{2} ; \quad n=0, \pm 1, \pm 2...$$

and

$$F_2 (nw_1) = \frac{a_{2n} - jb_{2n}}{2} ; \quad n=0, \pm 1, \pm 2...$$
Recalling $a_n$ and $b_n$ are defined as

$$a_n = \frac{2}{\pi} \int_{-\pi}^{\pi} f(t) \cos nw_1 t \; dt; \; n=0, \pm 1, \pm 2, \ldots$$

$$b_n = \frac{2}{\pi} \int_{-\pi}^{\pi} f(t) \sin w_1 t \; dt; \; n=\pm 1, \pm 2, \ldots$$

we can see that

$$a_{-n} = a_n \quad \text{and} \quad b_{-n} = -b_n , \quad \text{so that}$$

$$\theta_{-n} = -\theta_n \quad (\text{II-4-15})$$

Proper manipulation of (II-4-13) with (II-4-14) and (II-4-15) will give us

$$\phi_{12}(\tau) = \sum_{n=-\infty}^{\infty} c_{1n} c_{2n} e^{j(nw_1 \tau + \theta_{1n} - \theta_{2n})} \quad (\text{II-4-16})$$

and similarly

$$\phi_{21}(\tau) = \sum_{n=-\infty}^{\infty} c_{2n} c_{1n} e^{j(nw_1 \tau + \theta_{1n} - \theta_{2n})} \quad (\text{II-4-17})$$

From (II-4-16) and (II-4-17) we can say that contrary to the autocorrelation functions, the cross-correlation functions retain the phase differences of the harmonics which are present in both periodic functions correlated. Also, the coefficients of $\phi_{12}(\tau)$ and $\phi_{21}(\tau)$ are products of corresponding coefficients of the given functions of $f_1(t)$ and $f_2(t)$. Thus, if any of the harmonics is absent in $f_1(t)$ or $f_2(t)$ that corresponding harmonic will be absent in the cross-correlation function.

The correlation technique does not limit itself to the periodic functions only. Theories have been promoted to extend this technique
into the domains of aperiodic functions and random functions. Within
the scope of our work we are generally interested in using the
correlation techniques with speech sounds which can be considered as
segments of periodic or quasi-periodic functions of time. Extension
of the above developments made for the auto correlation functions and
cross correlation functions of periodic \( f(t) \) into the domain of
aperiodic \( f(t) \) permits us to cover the treatment applicable for
periodic, quasi-periodic and aperiodic functions of time, as the latter
class of functions has also the properties of the other two classes.
By quasi-periodic we have in mind the kind of speech samples which are
taken when sound is in transition either in amplitude or in frequency.
Those samples of sound can be visualized easily by considering a vowel
pronounced with a sliding pitch or a vowel sound read into the computer
memory thru an amplifier which does not act linearly in time by one
reason or another.

The extension of the correlation techniques from the domain of
periodic functions into that of aperiodic functions follows practically
the steps taken in going from the Fourier Discrete Transforms to the
Fourier Integral Transforms, and we are not going to repeat them here.
Reference to many works on statistical communication can provide
these steps. (5) Suffice to say that equations such as (II-2-3) up
to (II-2-6) take the form of the familiar pair of Fourier Transforms.

\[
\begin{align*}
f(t) &= \int_{-\infty}^{\infty} F(w) e^{jwt} \, dw \quad \text{(II-4-18)} \\
F(w) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-jwt} \, dt \quad \text{(II-4-19)}
\end{align*}
\]

The autocorrelation function, generalized for all periodic or aperiodic
functions will take on the form

\[ \phi_{11}(\tau) = \int_{-\infty}^{\infty} f_1(t) f_1(t + \tau) \, dt \]  \hspace{1cm} (II-4-19)

and gives rise to the pair

\[ \phi_{11}(\tau) = \int_{-\infty}^{\infty} \phi_{11}(w) e^{j\omega \tau} \, dw \]  \hspace{1cm} (II-4-20)

and

\[ \phi_{11}(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi_{11}(\tau) e^{-j\omega \tau} \, d\tau \]  \hspace{1cm} (II-4-21)

These equations relate the autocorrelation function of a function of time to its spectrum. In the case of speech sounds, the Fourier Transform of generalized autocorrelation can be called appropriately the energy density spectrum. In general it may be another quantity.

The generalized cross-correlation function of two different functions \( f_1(t) \) and \( f_2(t) \) is defined similarly as

\[ \phi_{12}(\tau) = \int_{-\infty}^{\infty} f_1(t) f_2(t + \tau) \, dt \]  \hspace{1cm} (II-4-22)

which gives, in turn, the pairs

\[ \phi_{12}(\tau) = \int_{-\infty}^{\infty} \phi_{12}(w) e^{j\omega \tau} \, dw \]  \hspace{1cm} (II-4-23)

and

\[ \phi_{12}(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi_{12}(\tau) e^{-j\omega \tau} \, d\tau \]  \hspace{1cm} (II-4-24)

\[ \phi_{21}(\tau) = \int_{-\infty}^{\infty} \phi_{21}(w) e^{j\omega \tau} \, dw \]  \hspace{1cm} (II-4-25)

and

\[ \phi_{21}(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi_{21}(\tau) e^{-j\omega \tau} \, d\tau \]  \hspace{1cm} (II-4-26)

with the same properties

\[ \phi_{12}(-\tau) = \phi_{21}(\tau) \quad \text{and} \quad \phi_{12}(w) = \bar{\phi}_{21}(w) \]

All the above properties of autocorrelation and cross-correlation functions hold true in the generalized forms. These forms can
accommodate all types of functions, including noise which is of a random nature. This will be discussed further in the following sections.

5. THE CHARACTERISTIC FEATURE EXTRACTION PROBLEM

From the above review of correlation techniques, two approaches are seen as possible for characteristic feature extraction in speech recognition process leading to the optimal filters for the group of sounds we proposed to investigate.

a. The cross-correlation approach

Let \( f_1(t) \) be the amplitude-vs-time representation of a syllable or a sound spoken by speaker No. 1 who should be chosen for the best possible enunciation. Let \( \phi_{12}(\tau) = \int_{-\infty}^{\infty} f_1(t) f_n(t+\tau) \, dt \) represent the cross-correlation function of the same syllable spoken by speaker No. 1 and speaker No. n. Then the output of an optimal filter for that syllable should be derived from the average. \((14)\)

\[
\phi(\tau) = \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} \phi_{1k}(\tau) \quad (II-5-1)
\]

This approach may be likened to a logical .AND. operation for each speaker with speaker No. 1 before averaging.

With our computer program flexibility, we will investigate the merit of the approach proposed in \((II-5-1)\) compared to the same approach but with the operation of autocorrelation after each cross-correlation, to remove the extraneous phase information, before the operation of averaging. This can be expressed as

\[
\phi(\tau) = \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} \phi(1k)(1k)(\tau) \quad (II-5-2)
\]

where \( \phi(1k)(1k)(\tau) \) is defined as the cross-correlation of \( \phi_{1k}(\tau) \) with itself, this last function being defined as in
b. The autocorrelation approach

Preliminary investigations have shown that the autocorrelation functions of the same syllable spoken by different people have very similar appearances. Thus the output of an optimal filter for a syllable can be derived from the average

\[ \phi(\tau) = \lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} \phi_{kk}(\tau) \]  

Equation (II-5-3)

This approach may be likened to a logical .OR. operation of each speaker with the whole group of n speakers.

6. THE OPTIMUM DECISION PROBLEM

In investigating the problem of optimal filter design with the computer we can, in our particular case, simulate completely our set of filters on the computer, for the purpose of testing and evaluation. We have given two solutions to the problem of optimum decision to find ultimately the best decision rule for our recognition system. These two solutions are sometimes termed the correlation metric and the Euclidean metric.

a. Correlation metric

The idea comes from the classical notion of superposition integral which is stated as

\[ o(t) = \int_{-\infty}^{\infty} h(\lambda) i(\lambda-t) \, d\lambda \]  

Equation (II-6-1)

where \( h(t) \) is the unit impulse response of a system, if it exists, \( i(t) \) the input function and \( o(t) \) is the output response.

If we let \( \phi_{hh}(\tau) \) be the unit impulse autocorrelation function of the same linear system, \( \phi_{ii}(\tau) \), the autocorrelation of the input function and \( \phi_{oo}(\tau) \) the autocorrelation function
of the output response, we can write according to (II6-1):
\[ \phi_{00}(\tau) = \int_{-\infty}^{\infty} \phi_{hh}(\tau) \phi_{ii}(\tau - \lambda) \, d\lambda \]  (II-6-2)
changing \( \lambda \) to \(-t\) we have
\[ \phi_{00}(\tau) = \int_{-\infty}^{\infty} \phi_{hh}(-t) \phi_{ii}(\tau + t) \, dt \]
As \( \phi_{hh}(\tau) \) and \( \phi_{ii}(\tau) \) are even function changing the sign of the arrangement of these function will leave them unchanged. Thus
\[ \phi_{00}(\tau) = \int_{-\infty}^{\infty} \phi_{hh}(t) \phi_{ii}(t + \tau) \, dt \]  (II-6-3)
By relating this last equation to the definition equation of correlation, we see that the autocorrelation function of the output response of a linear system is equal to the crosscorrelation of the system unit-impulse response autocorrelation function and the autocorrelation of the input function. As this relation is true for all delays \( \tau \), it is true too for \( \tau = 0 \).
That is
\[ \phi_{00}(0) = \int_{-\infty}^{\infty} \phi_{hh}(t) \phi_{ii}(t) \, dt \]  (II-6-4)
Suppose now that we store a set of patterns made up by the autocorrelation functions of a group of sounds and that we make the computer accept an autocorrelation function of a sound as input for recognition, the operation of cross-correlating this input autocorrelation (which can be called now \( \phi_{ii}(\tau) \)) with the set of \( n \) stored patterns (which can be called \( \phi_{hhk}(\tau), k = 1, 2, 3, \ldots n \)) to produce a set of \( n \) outputs,
\[ \phi_{00k}(0) = \int_{-\infty}^{\infty} \phi_{hhk}(t) \phi_{ii}(t) \, dt \]  (II-6-5)
\( k = 1, 2, 3, \ldots n \)
is exactly like passing that actual sound thru the set of filters patterned after those \( \phi_{hhk}(\tau) \) and measuring the output of each
filter. A unique feature of our computer simulation compared to the actual set of filters is that a learning process has been built into the program to modify in a matter of seconds the characteristics of any filters of the set (or create new ones) to adapt, in incremental steps as small as necessary, to variations produced by new voices inputting the filters. As long as this adaptivity in the recognition of one sound does not hurt the recognition of the others of the set, the modified correlation functions would be considered optimal for the set and transferred to the frequency domain finally for hardware implementation.

b. The Euclidean Metric

This makes the computer pick out one stored pattern and make a template matching with the input by going along the curve of that stored pattern within a positive and a negative tolerances (set sometime before) on the vertical line passing thru each point in time, to see whether any point of the input pattern is encountered. The total number of points encountered is kept and compared with that produced by other stored patterns. The pattern producing the highest number is picked out as the most likely pattern recognized. This process can profit also from the learning capability as explained above. From the primary investigation it appeared that this scheme of recognition is in general inferior to that using correlation metric, and will not be used in the final investigation proposed in Chapter IV.

7. REVIEW OF CURRENT LITERATURE

Flannagan (2) in his book, Speech Analysis, Synthesis and Perception, mentioned about a device implementing the computation of the autocorrelation function of the function of time (page 133) but the delay \( \tau \) of this device is fixed for technical reason, thus only the
function $\phi(\tau,t)$ is produced here with $t_1$ a running variable varying with real time. Another device is mentioned (on the same page) which produces the short time correlation function weighing the product of the original signal and the signal delayed. The defining relation is
\[
\phi(\tau,t) = \int_{-\infty}^{t} f(x) f(x+\tau) k(t-x) \, dx
\]  
(II-7-1)
where $K(t) = 0$ if $t < 0$, is the weighting function. None of these two devices were used to produce any autocorrelation leading to optimal filter design. A correlation-vocoder using equation (II-7-1) also is mentioned (page 259), along with other similar devices, but all of these were for transmission of speech with one of the main aims common with the vocoder systems: how to transmit large-bandwidth information set thru a smaller bandwidth channel.

Cannon (4) reported about a method of speech analysis that has been shown to be capable of recognizing with high accuracy a set of seven voiced vowels spoken by twelve male talkers with various regional accents.

The wave forms used in the recognition are tapped from four points along a low dispersive delay line, which represents a model of the human cochlea. These four output signals are sampled for 4 ms at a rate of 25,000 points per second per output channel. Sampling time is synchronized later manually and visually, with the onset of a glottal pulse. The 4 ms samples are autocorrelated on a digital computer, then cross-correlated (at zero delay) with a set of stored prototype patterns to produce an array of cross-correlation coefficients. These coefficients are treated as components of a multidimensional vector that characterized the input sound. The final decision as to which sound was
spoken is made by a simple linear adaptive network that was trained to separate these multidimensional vectors into their proper classes. Successful training was attained in all cases, indicating a linear separability of the vowel sounds in the space described by the correlation operations. The recognition rates are reported as below:

<table>
<thead>
<tr>
<th></th>
<th># OF SPEAKERS</th>
<th>TOTAL # OF SOUNDS</th>
<th>TOTAL # OF ERRORS</th>
<th>TOTAL # OF TIES</th>
<th>RECOGNITION RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Training</td>
<td>8</td>
<td>356</td>
<td>11</td>
<td>20</td>
<td>92%</td>
</tr>
<tr>
<td>With Tie Breaking Logic</td>
<td>10</td>
<td>424</td>
<td>19</td>
<td></td>
<td>95%</td>
</tr>
<tr>
<td>With Adaptive Network,</td>
<td>5</td>
<td>82</td>
<td>5</td>
<td>6</td>
<td>Not Recognizable</td>
</tr>
<tr>
<td>After 1st Training</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>86%</td>
</tr>
<tr>
<td>With Adaptive Network</td>
<td>5</td>
<td>71</td>
<td>2</td>
<td>0</td>
<td>Not Recognizable</td>
</tr>
<tr>
<td>After 5th Training</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>97%</td>
</tr>
</tbody>
</table>

Although no attempt was given to the question of optimal filter design in this paper by Cannon, it is important to us as a direct means of checking the performance of our system about the rates of recognition. The hyperplane (11), if used in our system, would have only as many as one fourth of the dimensions he used in his scheme, because we use only one pattern for each sound compared to four that he used. If our system attains the same rate of recognition as his, our may be considered better.

A device called SCEPTRON (6) described by a group of researchers at Sperry Gyroscope Company comes close to being what we define as an optimal filter. A square array of optic fibers is arranged so that these fibers have different length, thus different resonant frequencies.
A photosensitive plate is placed in front of the fibers while light is directed on the other ends of the fibers and a sound to be recognized is spoken to activate the fiber driver. When plate is developed, black dots will appear where the fibers were stationary, and black lines, where fibers were in motion. This plate serves as memory pattern, when placed back in the same position as before. In the absence of any signal, a photocell placed behind the plate will be in the dark, since each fiber light spot will be blocked by a black spot on the mask. When a sound is received, the fibers will again vibrate in some pattern characteristic of that sound. If it is the same signal, all the light will still be blocked by the mask; if different, some light will get through because the mask pattern will not be masked. Thus the photocell can tell whether or not the signal is the same by virtue of the amount of light it sees.

Variations of this device were constructed on the same basic principle to recognize numbers. However, no performance as far as the number of persons whose voices were recognized, nor rates of recognition were reported.

Hyde, in a report on investigation about automatic speech recognition sponsored by the British government from 1967 to 1968 (12) gives a broad survey of the literature. His review of hardware covers about thirty complete recognition systems employing a variety of signal analysis and pattern classification techniques. The different approaches have resulted in devices with similar performance figures. A small vocabulary of words spoken one at a time by a few chosen talkers can be recognized with an accuracy of about 95%.
SUMMARY

 Practically all of the systems for speech sound recognition surveyed directly or indirectly in this review make use of spectrum analysis in the frequency domain. The equivalence between the correlation function and the power density spectrum as an element for speech sound recognition is barely touched upon. (13) The different accuracies obtained in the different systems of recognition will provide us with a good base for comparison of performance for the experiments proposed in Chapter IV. These experiments involve computer evaluation of the cross-correlation between the time function of a sound produced by a chosen speaker and that by a practical number n other chosen speakers. This cross-correlation will be repeated for the group of speech sounds chosen for study. The produced cross-correlations for each sound will be averaged and transferred to the frequency domain by Fourier Transforms. The evaluation of this approach can be done by simulating the filter set, which would be patterned after these cross-correlations, on the computer. A second approach also will be tried by taking the autocorrelation of each sound by each speaker and averaging it with the autocorrelations of the same sound by other speakers. This approach will also be evaluated by computer simulation of the filter sets which would be patterned after those autocorrelations.
CHAPTER III

1. HARDWARE AND SOFTWARE INVOLVED

a) The primary hardware involved in this research is composed of:

A PDP-9 Computer system with:

- Main CPU
- A core memory of 16K, 18-bit work with 1-us cycle time, 2-us add time, hardware multiply and divide.
- A disk system whose erasable scratch can have up to 1136 blocks of 256 words each, 18-bits long - The average access time is about 17 ms and average rate of transfer is 150,000 words /s. Any block on this disk system can be addressed randomly in MARCO language, (a language of the assembler language type). Another 568 blocks are reserved for system programs and subroutines. A group of 568 blocks are reserved for storage of source program and the last 568 blocks are saved for binary programs. This set up makes it easy for creating a program, editing it, compiling it into binary program, loading it and having the computer execute it in a matter of minutes. The system software is powerful and well debugged;
- 2 units of DEC Tape;
- A line printer of 300 lines/min;
- A teletype, type ASR-35;
- An A/D converter of 20 us-conversion/word of 11 bits plus sign;
- An A/D converter of 50 us-conversion of 8 bits plus sign;
- A multiplexer of 8 channels for A/D or D/A;
- A fast optical reader-punch for paper tape;
-A 7-channel analog FM tape recorder;
-A scope of 1024 x 1024-point accuracy and 2.5 us of conversion
time per point in vector display, and 80 us in point mode.
-An X-Y plotter slightly modified to work with the D/A and the
multiplexer;
-A portable line battery tape recorder adjusted to give response
with ±2db (in overall record-playback mode) from 100 to 5000 Hz.

After the primary investigation, a preconditioning unit for the
speech sound before read-in onto the computer disk via the A/D conver-
ter proved to be desirable. Such a unit has been proposed by many re-
search groups in speech including our team at Ohio State University. The
performance of some such units was reported, but design data was not
made available.(21) Simulation of a logarithmic amplifier on the computer
proved to help in the first attempts of recognition to save money for ex-
pensive logarithmic amplifiers, this unit continued to be simulated by the
software. However to gain in speed of processing, time was called out for
the design of a preconditioning unit which would have as high a compres-
sion ratio as possible to produce practically the same output amplitude to
be presented to the computer, from input sources which can vary in prati-
cal limits in distance, direction or amplitude. The compression in ampli-
tude is taken care of by two amplifiers in cascade built around a new in-
tegrated circuit made by Sylvania. The gain compression rate and speed of
gain recovery are made controllable in 5 positions for selection of the
best set of these parameters. The compression ration is a compromise
choice between stability, overshoot, gain recovery speed. The compression
is such that past a certain point in the input amplitude, this amplitude
can vary around 40 db (a 1-to-100 ratio) without causing the output to
vary in amplitude or produce any visible or perceivable distortion. A less technical description is that a tape recorder playing a tape recorded with a continuous sound of the same amplitude, whether brought to within 1 inch of the microphone feeding that amplifier or to around 6 or 7 yards from that microphone, does not cause the output of that amplifier to vary in magnitude or shape.

With such a high ratio of compression, noise amplitude generated at the input of the unit by electronics or by environmental sources will be amplified up to the point unbearable, whenever we stop feeding the desirable input into it. A switch is necessary to cut the noise out of the output when we stop feeding desirable input into it and to turn on immediately for desirable input. Immediately means a few micro seconds only if we don't want to lose the beginning of consonants when time comes for these to be investigated. An electronic switch built the way they are built in multiplexer circuits or fast A/D converters was designed and works well with switching time less than 1 micro second.

An amplitude limiter at the end of the automatic-gain controlled unit is also desirable to chop out the shot noise which can cause false gain compression. Practice in the field and preliminary investigation have shown that a pre-emphasis, beginning at 1 KHZ on the high frequency end at the rate of about 6 db/octave will help the recognition process. Without it, the confusion between sounds like /i/ and /u/ occurs too often. It is reported that the contour of pre-emphasis is not critical, as long as it raises approximately by 6 db/octave beginning at 0 KHZ. The process, it was suggested, could even be thought of as a differentiation in frequency. The pre-emphasis network in our system can be flat or raise up to around 18 db from 500 HZ to 3 KHZ, or somewhere between 0
and 18 db. We will have a good way here to ratify the above assertion. The question as to where to place the pre-emphasis network in our system is important. In order to get a constant amplitude at the output the only place to put that network is in front of the AGC unit. Such emplacement plus the high compression ratio of the AGC unit make the speech sounds at the output of our unit to be very unnatural. Heuristic remarks pointed to the fact that such a unit helps remove quite a bit of personal characteristics as it makes a person speaking thru it sound like just another one of the same sex and of about the same age group. The same unit that seems of so "low-fidelity" to the ears of the un-involved people turn out to help definitely to increase the rate of recognition never attained before with the same system without that unit, when such a unit was built and inserted properly between the speech sound source and the computer.

The presence of band pass filters in the unit helps cut the spectrum to be analyzed at the overall rate of 48 db/octave at both ends. The two points of band pass specification are 300 HZ and 3000 HZ. This band pass action proves necessary to get rid of the problem of leakage in using the sampling process to input data to the computer. Leakage is referred to as the problem of having the frequencies outside of the band of frequency of interest impersonate, by inadvertent sampling, some frequency bands which originally did not exist before sampling.

The synoptic diagram of the preconditioning unit is shown in Figure 3 and its photograph, in Figure 4. The complete schematic diagram appears on the appendix along with the pre-emphasis characteristic.

b) Description of the Software

The software designed for this research is peculiar only in the use
Figure 3. Synoptic Diagram of the Preconditioning Unit. A.G.C. is the Automatic-Gain-Controlled Amplifier.
Figure 4.- Photograph of the Preconditioning Unit.
of extremely short teletype messages. Anything else was written general
even and flexible enough that it can be used easily by any person later
in a follow-up effort or for some other kind of analysis requiring data
acquisition and processing by computer.

There are several reasons why the programming was done in MACRO9,
a sort of assembler language instead of in FORTRAN, for example, which is
available on the same system and understood by other people and systems
too. Among these reasons, some are:

i) On the present system available to us, random access to the
disks cannot be performed in FORTRAN;

ii) The assembler-type language is a lot more powerful and flexible
than FORTRAN. A large number of subroutines can be written directly into
the main program in MACRO9 to save a lot of core locations for other pur­
poses. After investing some time in learning MACRO9, this language can
be handled as easily as FORTRAN or any other high-level languages.

iii) A program for reading data into the computer disk via the A/D
and processing them for input to another program computing the FFT were
both written in MACRO9 by one of our faculty members, Professor C. E.
Warren of the Electrical Engineering Department. With the guidance of
Professor Warren and given those two above programs, the author tried to
learn to use MACRO9. Around these two programs many other subprograms
were designed to fit into the whole big program listed in one of the
appendices of this paper.

Good care was taken to optimize to the utmost parts of the program
necessitating speed, especially in the two subroutines SUBA and SUBB of
the calculation of the correlation functions. A subroutine was also
created to adapt an X-Y plotter for fast and inexpensive hard-copies
and at the same time to make full use of the speed in the scope display. In the following the author will try to describe as comprehensibly as possible the several options which have been built into the program so that other interested people can make use of this program without having to spend as much time as learning every detail of the program.

c) Program Options

Selection of options is initiated by typing an appropriate number after the computer types out GO: and waits for input.

At the beginning of the session, the computer will type out MEMO: and wait for input. This option allows us to enter a line of memo for the computer to keep. This memo maybe anything, a date for instance. When ever the line printer is ordered to print anything this line of memo will be printed along automatically for the record. Computer then types, after MEMO: has been entered,

SAVSTA: AVESTA: RECWRD: CPWR:1 CEPW:1 TOL1:

These are key numbers to help pre-select some of the automatic features which save tremendously time and nerve.

A number, say 400, for SAVSTA tells the computer about where to begin to save for later processing any number of blocks of data (a block is 256 words of 18 bits each) up to 68 at a time if SAVSTA is placed initially at 400, since block 468 heads the record read-in via the A/D converter and block 500, the record for AVESTA in general.

AVESTA is where to begin to save the averages when option AVERAG is called upon.

RECW RD is the length of the word to be used in the recognition
process.

**CPWR** is a switch which will save time by skipping the cumulative power calculation in the Fourier Transform operation. If we don't want cumulative power to be calculated 0 should be assigned to CPWR. 1 assigned to it will do the contrary.

**CEPSW** This is a switch that saves time if we don't wish the cepstrum to be computed, 1 will, and 0 will not make computer spend some extra time to process some data necessary only if a cepstrum is to be calculated.

**TOLL** is the tolerance preset value for recognition by Euclidean metric. After assigning six numbers properly chosen for these quantities (separated by one or more spaces and typed carriage return (CR), the computer will type:

```
DB:1 PT: DRE:1 MXSK:1 RSTA: LNC:
```

**DB** should be assigned the value of 0 if regular amplitude of the functions of time $f(t)$ read-in via A/D converter is wanted, 1 for logarithm of that amplitude and 2 if derivative of $f(t)$ is wanted. This option makes computer skip many questions in between and speeds up the investigation of recognition by simulation of a logarithmic amplifier, a linear amplifier or a differentiator to process the $f(t)$ read-in, before another action.

**PT** helps make automatic the choice of length of record read-in for processing, If assigned a number such as 256, this switch automatically make computer wait at a proper time to read in 256 points of data. If assigned 0, it gives free choices at will.

**DRE=1** directs computer automatically to recognition process after an autocorrelation function has been computed. If it is assigned 0,
recognition is for choice.

MXSK=1 will skip the teletype printout of the successive higher maximums for the record in the recognition process; MAXSK=0 will type out these maximums.

RSTA The number assigned to this word will direct computer to begin recognition process at that number. This option gives tremendous time saving in trying several schemes of recognition or several set of stored patterns.

LNC A number assigned to this words assigns a weigh to the stored patterns with which a new pattern (known to belong to a sound but is classified wrongly to another one) has to be averaged in order to "pull" computer toward a pattern next time of recognition. For instance, if a stored pattern is the average of the pattern of 10 persons, it is logical to assign a weight of 10 to LNC (For Learning Coefficient) so that if a new pattern is to be averaged with it the averaging process will multiply every amplitude of that stored pattern by 10, add to the corresponding amplitude of the new pattern, divide this sum by 11 and store the result back in the same place, automatically. If that is not enough for successful recognition the option LEARN will be called upon again by a few pushed buttons, to add more influence of the new pattern to the stored one. If the learning process does not converge quickly, we can examine (on scope, plotter or line printer) why.

After having assigned proper value to these items and typed CR, computer will type GO:. Here we have a number of choices and they are all safe as we never can make a wrong choice to harm the data read in before. Safety is built in here. From this point we can just type a number terminated by a CR to go automatically to options. The number in the
following will be that to be typed to get to the option listed along with that number. If a number is typed which does not figure among options, computer will type DO AGAIN, PLEASE and then, GO:, until option is correct.

1-COR This option directs computer to read into core and ultimately reload disk 6 beginning from block 0. These data will be dug out along with those stored beginning from block 64 for cross-correlation. Reference 17 gives algorithm for this option.

2-COR Same as option 1, except loading from block 64.

3-BEGIN 1 This option tells computer to pick up data from block 0 and block 64 and process to take the correlation function, normalize it by dividing by \( \psi(0) \) if it was an autocorrelation function and by the maximum number that it had to figure out if it was a cross-correlation, and store the result beginning at block 192.

4-COR This option is like 1 or 2 except it combines both.

5-PRINTING This option picks up the data from the place specified by the last choice and automatically prints on the line printer the results of whatever was in question just before 5 was selected.

6-FastFourierTransform (8) - Depending upon when this number is typed the FFT will be computed for data stored beginning at block 64 or at block 192. Hanning or Hanning with trailing zeroes are also built in.

7-PARINI This makes computer go to top of program for all options to be chosen.

8-CORREC After this option has been selected computer will type:

DWRD: CWRD: PLACE:

To be assigned to DWRD (for Delected Word) is the number of leading points to be deleted in a record of 512 points to move the zero-time point to the right by such a number of points.
CWRD is the number of points to be used in the correlation process.

PLACE will assign the thus newly prepared record to either block 0, block 64 or both.

9-GTIOS  After having selected this option, allow computer to type out:

   FR: TO: #PTS
   FR: from which block?
   TO: to which block?
   #PTS: number of points?

Computer will compute automatically the number of blocks to be filled or fill in the only selected block if #PTS is less than or equal to 256.

10-SAVE  This saves beginning at block SAVSTA specified sometime before and then increment properly the counter to ready it for next SAVE, and print out the beginning block number of the record just filled in.

11-CEPSTR  This tells computer to go to block 64 and begin to read into core as much as specified by the number of points assigned just before going to this option, then use these data and perform FFT. The program reserves sequentially in time before 11 is selected the data of the db plot in the record beginning with block 64.

12-SMOOTH  This makes computer type #PTS:

These are the number of points to be multiplied by a half-cosine-bell function which make these points gradually smaller and finally zero. (See explanation about Cepstrum on page II- ). Computer will then take inverse FFT of this record to produce a smoothened spectrum for an originally "rough" spectrum.

13-AVERAG  Computer will type #BLK:  This is the number of blocks of 256 points (words) each to be averaged together. After assignment to
#BLK and CR, computer types:

LIST'EM

This is when to type sequentially the block numbers to be averaged.

At last block number, computer types:

W.BLK1: This is the weight assigned to block number one for the purpose of averaging the average of a previous group of data with a new record of data. Computer will pick up corresponding points on each block and add them to the corresponding point of block number one properly multiplied already by the weight coefficient and divide the whole thing by a number equal to weight coefficient plus one, then store result at a block whose number is specified by a counter which is printed out and everytime 13 is selected.

14-RECINI Initiate for recognition by specifying number of blocks containing the stored patterns to be compared with input, and then go to 15.

15-RECON This is the process of recognition by crosscorrelating at zero delay a pattern stored at block 192 successively in the patterns stored beginning at block RSTA specified at the very top of program until the number of pattern specified sometime before by RECINI has been reached. Depending upon the value given to MXSK at the very top of program, the computer prints out, for record, all the successive coefficients obtained in the crosscorrelation which it recognized as bigger and bigger and stops printing out at the biggest. Finally it prints out the coded number of the pattern it recognized as producing the biggest result by the above mentioned crosscorrelation process.

16-RECEU This option does the same things as 15 does, except the recognition is performed by, in a way, looking through a vertical slot (with preset upper and lower limits) and going along the curve of a stored
pattern and trying to see if a point of the input pattern is visible. One is added to the current score if such a point was visible.

23-AACOR  This option makes computer pick up data from block 192 (with length of record compatible with length used just before 23 was chosen) and put them into records beginning at 0 and 64, take the correlation function between these two records and put the result in a record beginning at 192. As block 192 usually heads the record containing the results of a correlation, this option, if performed after a correlation, gives quickly the auto correlation of either a cross correlation function or an autocorrelation stored in the record headed by block 192.

24-LEARN1  This option makes computer request for the code number it’s supposed to type out as that of the recognized pattern. It’s request is typed out as YES!. We then type in the code number we want it to produce as recognition result. It learns by averaging the new pattern, point by point, with the stored pattern it was directed to. A proper weight would be assigned to the stored pattern before averaging as explained in option AVERAG on page (III-10). After averaging the modified pattern, now with the influence of the new pattern, computer will put it back to the same location and go automatically through the recognition process again to show us whether it has "learned" enough to recognized the new pattern correctly. If not, we can renew the learning process for computer by typing 24 again. If the new pattern finally is recognized correctly we can go on to other options or renew the learning process once more to be sure. Each learning cycle like that takes about 2 seconds on the computer. As results have shown, since the influence of the new pattern can be added to the stored pattern in small increments, this stored pattern doesn't risk to be un usable for the voices that made it up
in the past. For some very particular new voices, we may have to readjust the pattern for some members of the past population after making this pattern adjust itself for those particular voices.

40-RECU This option makes computer go to record headed by block 468 which contains the original, unprocessed data read-in via the A/D converter. This record length will be requested by computer by its typing #PTS:. We will have the choice between any number from 1 to 256 or beyond that, any number multiple of 256, up to 8192 points. Depending upon the switches that provide automatic routing, the next shortest step for computer would be to load this record into two records headed by block 0 and block 64 and type out GO: for further order. We see that this recuperation process helps us in doing several things with the same record of data. This is particularly useful, for instance, when we want to have a look at the spectrums produced by a raw set of data, by its autocorrelation function, by the same data but with Hanning window or with Hanning window and trailing zeroes, etc. Another use of this recuperation is to determine the adequate length of record which will produce a consistent autocorrelation function, long enough to preserve information but short enough to give good speed to the autocorrelation routine.

60-RENT1 This option makes computer go directly to recuperate the record read-in previously by A/D converter and take FFT of it.

70-COR This option makes computer read in from A/D converter a record whose length is determined by our answer to computer by a number, after it types out #PTS:. After this, depending on the automatic skipping switches, it would go on and compute the FFT (direct or inverse) and get everything ready for plotting or printout.

100-START This makes computer go to the very beginning of program
for us to modify the content of memo (See page (III-6)) and the status of all the automatic skipping switches and values of the critical coefficients as explained on page (III-7) and (III-8).

**0-PL0TS** At any moment after GO: typed out by computer, a 0 followed by a CR directs computer to plot on the scope whatever was processed just before. The length of the X axis is, in most cases, automatically expanded or compressed to accommodate record length from 128 to 8192. Ordonate values on Y are scaled automatically to have ±250 as maximum.

**200-PL0TP** Same as in option 0, but instead of plotting on scope, computer will plot on the X-Y plotter. Scales in option 0 apply here also. All the necessary delays are built into our program, and so are the automatic pen-lift or pen-down actions. The special subroutine POINT modified from the system subroutine by the author, can be found in the appendix. The size of plots on the X-Y plotter can be adjusted with attenuators on this unit. An Y-axis and an X-axis are also drawn in this option.

**101-PL0TS** This option as well as 102 up to 203 below applies only after a FFT has been evaluated. Computer is directed to plot on scope the amplitude-vs-frequency spectrum.

**102-PL0TS** Computer plots on scope the db-amplitude-vs-frequency spectrum. Maximum of plot is scaled to represent 0 db. Around 50 db of dynamics is accommodated on the db axis.

**103-PL0TS** This produces the plot of cumulative power calculated from the amplitude spectrum. Maximum of this plot is automatically made equal 100%.

**201-PL0TP** Same as 101, but plot is on a X-Y plotter.
202-PL0TP Same as 102, but plot is on X-Y plotter.

203-PL0TP Same as 103, but plot is on X-Y plotter.

A separate program was also written to punch out or read in paper tape from and into the disk starting at any random block number. A copy is included in the Appendix.

2. SUMMARY DISCUSSION ON HARDWARE AND SOFTWARE

i) Primary investigation showed that the preconditioning unit is critical in producing consistent patterns for either recognition by the simulated filter set on the computer or by the systems which use the real filters when these are implemented with hardware.

ii) The A/D-converter sampling clock was set to have always the same rate of 8192 samples/sec which satisfies the Nyquist rate for the bandwidth limited between 300 and 3000 HZ received from the output of the preconditioning unit. The D/A-converter clock rate was not critical.

iii) The computer program was written with a view to future options since many likely repeated routines were written in subroutine forms into the main program and each can be called upon by a single statement. The whole program may be called out from the disk to begin to work in a matter of seconds. This speed applies also to the auxiliary program for paper tape punching and reading. Many combined operations which required lengthy dialog between computer teletype and experimenter were arranged by automatic switches in the main program so that they can be called upon by a single teletype input.
CHAPTER IV

1. PROPOSED INVESTIGATION

A. Data Collection No. 1
   (Tape recording of ten pure vowels)
   a. Ten speakers with the same age and sex.
   b. Ten speakers, both sexes, different ages and different nationalities.

B. Data Collection No. 2
   (Tape recording of ten (vowel + consonant) derived from page 1)
   a. Six speakers with the same age and sex.
   b. Six speakers, both sexes, different ages, and different nationalities.

C. Data Reduction and Experiments
   a. Computer calculations of $\mathcal{U}(\tau)$ as described by equation (II-5-1).
   b. Computer calculations of $\Phi(w)$, from part a.
   c. Computer calculations of $\mathcal{U}(\tau)$ as described by equation (II-5-2).
   d. Computer calculation of $\Phi(w)$ from part c and compare with part b.
   e. From part c, simulate on computer the filter set and test recognition.
   f. Compute calculations of $\mathcal{U}(\tau)$ as described by (II-5-3).
   g. From part e, simulate on computer the filter set and test recognition.
2. REPORT ON INVESTIGATION

Simulation on the computer of the filter set derived from equation (II-5-1) and (II-5-2) shows that recognition by such a set of filters is very poor compared to the recognition by the set of filters derived from equation (II-5-3) in every test involving speakers of the same or different age and sex. This comes from the fact that the dominant part of spectrum of a sound spoken by a person may fall out of the range of that of the spectrum of the same sound spoken by the chosen speaker No.1 and thus produces a cross-correlation function containing the frequencies common to these two spectrums, but is not characteristic enough to separate this cross-correlation function from others derived similarly from other sounds. In inspecting the spectrum of such cross-correlation functions, it is not rare to observe that such cross-correlation functions produces only a single narrow spectral peak, which is obviously not enough to characterize the sound. Another disadvantage of this cross-correlation approach compared to the approach based on equation (II-5-3) is that it cannot be made easily and speedily to benefit from the learning process so easily implemented for the other approach using the auto-correlation functions from the beginning.

The final stage of the investigation was consequently devoted completely to the optimization of results derived from the approach involving equation (II-5-3). Two advanced options of averaging were written into the computer program beside option 9 to take care automatically of the averaging of several blocks of data (see details on front pages of computer program in Appendix). In virtue of the fact that the preconditioning unit preceding the A/D converter makes the computer produce always about the same auto-correlation function for the same sound re-
corded on a magnetic tape which was subsequently played back at volume varying in fairly large limits (around 15 db), as long as the tone control is not modified, a single auto-correlation function generated by the sound played back at an average volume may be considered as the statistical average, as far as recognition is concerned, of the auto-correlation of that same sound played several times at different volumes; then the design and evaluation of a set of optimal filters for the 10 proposed vowels may be based on the following scheme:

The data collections were made according to the proposals A and B of Chapter IV. Each sound was spoken twice by each person. Suppose we are experimenting with the 10 vowels as spoken always in the same order as /i/, /I/, /æ/, /u/, /o/, /U/, /u/, /a/, /a/, by 10 speakers S₁, S₂, ...... S₁₀. Suppose further that each sound is spoken twice by each speaker. Each sound can then be played back on a tape recorder and its auto-correlation function calculated and normalized with respect to the value at delay Tau=0 and stored at a block on the disk memory. When all the 200 sounds spoken by 10 speakers are processed in that fashion, we can line up the auto-correlation functions so that the record of 20 auto-correlations functions of speakers S₂, which follow each other in the order /i/, /i/, /I/, /I/, .... /a/, /a/ follows that of speakers S₁ and precedes that of speaker S₂, and so on. The total record RT begins with /i/, /i/ of speaker S₁ and ends with /a/, /a/ of speaker S₁₀. The advanced averaging options can then be called upon to produce from that total record another record RP having only 10 blocks, each block being the arithmetical average of 20 blocks generated by 20 sounds of the same name spoken by 10 different speakers. This last record will be stored at a known location of the disk memory and each of its blocks can be accessed.
separately. The order of the blocks follows the same order as /i/, /I/... /u/, /a/. These blocks will be used as the prototypes for the recognition of the original sounds that made these prototypes up as they are played back again by the tape recorder.

Since we just showed above that each auto-correlation function in the record RT may be considered as the statistical average of several auto-correlation functions which would be generated by the same sound played back for recognition for several times, we may just as well use the 200 auto-correlation functions stored consecutively in an orderly fashion in record RT to evaluate the recognition by making the computer sort them out with the auto-correlation functions stored in record RP as prototypes.

An automatic recognition process called upon by the number 124 at the teletype will request some additional data concerning the location and length of records RT and RP and will direct the computer to go thru RT from the first to the last block, and after each block to print out the coded number of the recognized sound. One sub-option at this point directs the computer not to attempt any learning. Another sub-option directs the computer to check to see if recognition was correct and if not to make one pass of learning and try the recognition again; after that, to go on to the next block of RT. The checking information is "released" to the computer only at checking time and derived from the orderly lining-up of the blocks in RT. This last sub-option can be ordered to print out the results of recognition and learning or not to do so in order to speed up the process. At the end of record RT, computer will check to see whether it had to learn any time, or equivalently, whether there was an error in recognition, while it went thru all the
blocks of record RT. If result of this last checking is yes, it will go automatically back up to the first block of RT and goes thru the process again, until no error was encountered or the permissible number of recognition cycles (assigned by experimenter) has been elapsed, which ever came first. If no error was encountered throughout the recognition of the whole record RT, it will then print out "LEARNING DONE" and the number of cycles it went thru. If the permissible number of cycles was elapsed and there still was at least an error in the recognition, it will print out "CAN'T LEARN" and then stop and wait for further orders.

A good way to use this process with its two sub-options is to assign a learning coefficient equal to about 5 times the number of speakers involved and to make computer go thru a number of learning cycles without printing out result at each recognition. If "LEARNING DONE" is typed out before computer halting, it is the perfect learning and recognition now is 100% in the simulation and approaches this rate in recognition with tape recorder playback. If "CAN'T LEARN" is typed out before computer halting, we can expect that recognition is better than before learning or that recognition has reached saturation with recognition slightly better or worst than that before learning. A cycle of recognition without learning and with printing can be ordered at this point for evaluation. If recognition rate is judged acceptable within a set of criteria, then the auto-correlation functions in record RP, which have been adjusted for optimization during the learning process, can be duplicated (except for the point at Tau=0) for negative delays by an option called upon by 33 and transformed into the frequency domain with proper smoothing, for hardware implementation.

Prior to the automatic learning process, investigation showed that
the sound /a/ as in Tar consistently clustered around prototypes /ɔ/ or /o/ but rarely around prototype /a/; this comes from the fact that our scheme of recognition takes care of only around 31 msec at the very beginning of each sound, and the rolling effect of the letter r in that sound /a/ comes in general later than those 31 msec. This fact made up our mind about concentrating on the separability of the first 9 vowel sounds in the list. For this group of 9 sounds, success in training the computer to recognize at 100% for up to 5 persons of about the same age and of same sex was obtained after about 10 cycles of learning; at rate 100% for up to 2 persons of different age or sex, or both; at rate about 92% for a group of 10 of same sex and within the 18-to-30-year-old bracket; at rate about 90% for 10 speakers of different ages, sex, regional and subcultural origins; and at about 90% for any group of 6 speakers when these vowels were linked with ending consonants. Results listed below are for stored pattern recognition, which represent the upper limits for the recognition by actual tape recorder playback or live inputs to the computer in front of a microphone preceding the pre-conditioning unit.
Rate of recognition for a group of 10 male speakers of ages from 18 to 30. Learning coefficient was 50.

<table>
<thead>
<tr>
<th>Sound</th>
<th>/ɪ/</th>
<th>/ɪ/</th>
<th>/e/</th>
<th>/ae/</th>
<th>/</th>
<th>/o/</th>
<th>/u/</th>
<th>/u/</th>
<th>/</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Training</td>
<td>18/20</td>
<td>16/20</td>
<td>10/20</td>
<td>15/20</td>
<td>10/20</td>
<td>14/20</td>
<td>18/20</td>
<td>17/20</td>
<td>7/20</td>
</tr>
<tr>
<td>After 5 Training Cycles</td>
<td>18/20</td>
<td>18/20</td>
<td>15/20</td>
<td>19/20</td>
<td>15/20</td>
<td>15/20</td>
<td>18/20</td>
<td>18/20</td>
<td>14/20</td>
</tr>
<tr>
<td>After 10 Training Cycles</td>
<td>20/20</td>
<td>19/20</td>
<td>17/20</td>
<td>18/20</td>
<td>16/20</td>
<td>17/20</td>
<td>20/20</td>
<td>20/20</td>
<td>14/20</td>
</tr>
<tr>
<td>After 15 Training Cycles</td>
<td>20/20</td>
<td>19/20</td>
<td>17/20</td>
<td>20/20</td>
<td>18/20</td>
<td>17/20</td>
<td>19/20</td>
<td>20/20</td>
<td>17/20</td>
</tr>
</tbody>
</table>

| Overall Rate | 125/180 = 69.5% | 150/180 = 83% | 161/180 = 89.5% | 167/180 = 92.6% |

After 15 cycles of training, no gain was obtained in increasing the number of training cycles. Beyond 15 cycles of training, overall rate oscillate around 88%. This oscillation came mainly from sound /ɪ/ and /u/ being too close to each other in the pronunciation of some members of the group.
Rate of recognition for a group of 6 male speakers of ages from 18 to 30 and of 4 female speakers of ages from 12 to 45. Learning coefficient was 50.

<table>
<thead>
<tr>
<th>Sound</th>
<th>/i/</th>
<th>/I/</th>
<th>/E/</th>
<th>/ae/</th>
<th>/</th>
<th>/o/</th>
<th>/U/</th>
<th>/u/</th>
<th>/</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Training</td>
<td>15</td>
<td>12</td>
<td>5</td>
<td>11</td>
<td>13</td>
<td>11</td>
<td>19</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>Overall Rate</td>
<td>111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>180</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>After 5 Training Cycles</td>
<td>17</td>
<td>16</td>
<td>12</td>
<td>15</td>
<td>15</td>
<td>19</td>
<td>20</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>153</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>180</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>After 10 Training Cycles</td>
<td>17</td>
<td>15</td>
<td>10</td>
<td>17</td>
<td>14</td>
<td>16</td>
<td>18</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>145</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>180</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>180</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

After 10 cycles of training, no gain was obtained in increasing the number of training cycles. Beyond 10 cycles of training rate oscillated around 81% for this group of speakers.
Rate of recognition for a group of 6 male speakers of ages from 18 to 30. Recognition was actually made for the vowels as they are pronounced with an ending consonant. Learning coefficient was 50.

<table>
<thead>
<tr>
<th>Sound</th>
<th>/eam/</th>
<th>/ip/</th>
<th>/en/</th>
<th>/ap/</th>
<th>/alk/</th>
<th>/one/</th>
<th>/ook/</th>
<th>/ool/</th>
<th>/on/</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Training</td>
<td>11/12</td>
<td>11/12</td>
<td>7/12</td>
<td>11/12</td>
<td>2/12</td>
<td>0/12</td>
<td>0/12</td>
<td>6/12</td>
<td>6/12</td>
</tr>
<tr>
<td>After 10 Training Cycles</td>
<td>11/12</td>
<td>12/12</td>
<td>8/12</td>
<td>12/12</td>
<td>11/12</td>
<td>7/12</td>
<td>6/12</td>
<td>8/12</td>
<td>8/12</td>
</tr>
<tr>
<td>After 15 Training Cycles</td>
<td>11/12</td>
<td>12/12</td>
<td>12/12</td>
<td>12/12</td>
<td>11/12</td>
<td>7/12</td>
<td>9/12</td>
<td>9/12</td>
<td>10/12</td>
</tr>
<tr>
<td>After 20 Training Cycles</td>
<td>11/12</td>
<td>12/12</td>
<td>12/12</td>
<td>12/12</td>
<td>12/12</td>
<td>9/12</td>
<td>11/12</td>
<td>9/12</td>
<td>11/12</td>
</tr>
<tr>
<td>After 25 Training Cycles</td>
<td>11/12</td>
<td>12/12</td>
<td>12/12</td>
<td>12/12</td>
<td>12/12</td>
<td>9/12</td>
<td>12/12</td>
<td>9/12</td>
<td>11/12</td>
</tr>
</tbody>
</table>

After 25 training cycles, no gain was obtained in increasing the number of training cycles. Beyond 25 training cycles, overall rate oscillate around 90% for this group of speakers.
Rate of recognition for a group of 4 male speakers of ages from 18 to 30 and of 2 female speakers of ages 12 and 45. Recognition was actually made for the vowels as they are pronounced with an ending consonant. Learning coefficient was 50.

<table>
<thead>
<tr>
<th>Sound</th>
<th>/eam/</th>
<th>/ip/</th>
<th>/en/</th>
<th>/ap/</th>
<th>/alk/</th>
<th>/one/</th>
<th>/ook/</th>
<th>/ool/</th>
<th>/on/</th>
<th>Overall Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Training</td>
<td>11/12</td>
<td>11/12</td>
<td>7/12</td>
<td>11/12</td>
<td>2/12</td>
<td>0/12</td>
<td>0/12</td>
<td>4/12</td>
<td>6/12</td>
<td>52/108 = 48%</td>
</tr>
<tr>
<td>After 10 Training Cycles</td>
<td>11/12</td>
<td>12/12</td>
<td>8/12</td>
<td>12/12</td>
<td>11/12</td>
<td>7/12</td>
<td>6/12</td>
<td>8/12</td>
<td>8/12</td>
<td>83/108 = 62.5%</td>
</tr>
<tr>
<td>After 15 Training Cycles</td>
<td>11/12</td>
<td>11/12</td>
<td>12/12</td>
<td>12/12</td>
<td>11/12</td>
<td>7/12</td>
<td>9/12</td>
<td>9/12</td>
<td>10/12</td>
<td>92/108 = 85%</td>
</tr>
<tr>
<td>After 20 Training Cycles</td>
<td>11/12</td>
<td>12/12</td>
<td>11/12</td>
<td>12/12</td>
<td>11/12</td>
<td>9/12</td>
<td>11/12</td>
<td>9/12</td>
<td>11/12</td>
<td>97/108 = 89.9%</td>
</tr>
<tr>
<td>After 25 Training Cycles</td>
<td>11/12</td>
<td>12/12</td>
<td>12/12</td>
<td>12/12</td>
<td>9/12</td>
<td>10/12</td>
<td>9/12</td>
<td>11/12</td>
<td>96/108 = 91%</td>
<td></td>
</tr>
</tbody>
</table>

After 25 cycles of training, no gain was obtained in increasing the number of training cycles. Beyond 25 training cycles, rate or recognition oscillates around 90% for this group of speakers.

All the above rates are for members of the group of speakers making up the prototype, each one having the same number of sounds tested as the others.

Recognition of sounds by speakers who did not make up the prototypes attained about 60 to 70% if the speaker was of about the same age and sex and, about 30 to 40% if the speaker was not of the same age group.
or of the same sex, or both.

After training of computer reaching a maximum FL in the rate of recognition the prototype auto-correlation functions were transformed into the frequency domain and smoothened as shown in the Figures 5 to 41.
Figure 5.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /i/ for 10 male speakers of from 18 to 30 years old.

Unit of delay $\tau$ is $1/256$ sec.. Unit of frequency $f$ is 256 Hz.
Figure 6.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /I/ for 10 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hz.
Figure 7.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /E/, 10 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hz.
Figure 8.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /æ/ for 10 male speakers of from 18 to 30 years old. Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hz.
Figure 9. — Optimal Autocorrelation Function and Energy Density Spectrum of sound /σ/ for 10 male speakers of from 18 to 30 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency f is 256 Hz.
Figure 10. -- Optimal Autocorrelation Function and Energy Density Spectrum of sound /o/ for 10 male speakers of from 18 to 30 years old. Unit of delay Tau is $1/256$ sec.. Unit of frequency $f$ is 256 Hz.
Figure 11. --Optimal Autocorrelation Function and Energy Density Spectrum of sound /U/ for 10 male speakers of from 18 to 30 years old.  Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hz.
Figure 12.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /u/ for 10 male speakers of from 18 to 30 years old. Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hz.
Figure 13.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /æ/ for 10 male speakers of from 18 to 30 years old. Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hz.
Figure 14.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /i/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 15. -- Optimal Auto correlation Function and Energy Density Spectrum of sound /I/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 16.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /E/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 17.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /ae/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 18. -- Optimal Autocorrelation Function and Energy Density Spectrum of sound /9/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 19.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /o/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is 1/256 sec. Unit of frequency is 256 Hz.
Figure 20. — Optimal Autocorrelation Function and Energy Density Spectrum of sound /U/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is 1/256 sec. Unit of frequency is 256 Hz.
Figure 21.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /u/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is $1/256$ sec. Unit of frequency is 256 Hz.
Figure 22.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /ʌ/ for 6 male speakers of from 18 to 30 years old and of 4 female speakers of from 12 to 45 years old. Unit of delay Tau is 1/256 sec. Unit of frequency is 256 Hz.
Figure 23.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /i/ as it is pronounced in /eam/ by 6 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec.. Unit of frequency f is 256 Hz.
Figure 24.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /I/ as it is pronounced in /ip/ by 6 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hs.
Figure 25.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /E/ as it is pronounced in /en/ by 6 male speakers of from 18 to 30 years old. Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hz.
Figure 26.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /ae/ as it is pronounced in /ap/ by 6 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec. Unit of frequency f is 256 Hz.
Figure 27.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /a/ as it is pronounced in /alk/ by 6 male speakers of from 18 to 30 years old.
Unit of delay Tau is 1/256 sec.. Unit of frequency f is 256 Hz.
Figure 28.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /o/ as it is pronounced in /one/ (as in Tone) by 6 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec.. Unit of frequency f is 256 Hz.
Figure 29.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /ʊ/ as it is pronounced in /ook/ by 6 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec., Unit of frequency f is 256 Hz.
Figure 30.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /u/ as it is pronounced in /ool/ by 6 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec.. Unit of frequency f is 256 Hz.
Figure 31.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /\lambda/ as it is pronounced in /on/ (as in Ton) by 6 male speakers of from 18 to 30 years old.

Unit of delay Tau is 1/256 sec.. Unit of frequency f is 256 Hz.
Figure 32.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /i/ as it is pronounced in /eam/ by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is 1/256 sec. Unit of frequency is 256 Hz.
Figure 33.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /I/ as it is pronounced in /ip/ by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 34.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /E/ as it is pronounced in /en/ by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 35.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /ae/ as it is pronounced in /ap/ by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is 1/256 sec. Unit of frequency is 256 Hz.
Figure 36.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /ɔ/ as it is pronounced in /alk/ by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 37.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /o/ as it is pronounced in /one/ (as in Tone) by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is $1/256$ sec.. Unit of frequency is 256 Hz.
Figure 38.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /U/ as it is pronounced in /ook/ by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is 1/256 sec.. Unit of frequency is 256 Hz.
Figure 39.--Optimal Autocorrelation Function and Energy Density Spectrum of sound /u/ as it is pronounced in /ool/ by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is $1/256$ sec.. Unit of frequency is 256 Hz.
Figure 40.—Optimal Autocorrelation Function and Energy Density Spectrum of sound /ʌ/ as it is pronounced in /on/ (as in Ton) by 4 male speakers of from 18 to 30 years old and of 2 female speakers of 12 and 45 years old. Unit of delay Tau is 1/256 sec. Unit of frequency is 256 Hz.
DISCUSSION ON RESULTS

Investigation shows that for a group of around 5 speakers of the same age and sex our system of recognition can be trained to afford 100% of recognition easily as the sounds presented at the input of the system one by one by a tape recorder adjusted in tone very closely so that when this tape recorder was used to present the sound to train the system. Volume of tape recorder may vary around 15 db without changing recognition owing to the action of the Preconditioning Unit. Recognition higher then 90% for a group of 10 speakers of different ages and sexes and of same sex and approximate age shows that the investigation is on a good track to the solution of the characteristic feature problem. Inclusion of some other vowels such as /er/ as in Her and some sounds beginning with consonants such as Cat, Hat, Pat, etc., showed that separability of those 9 vowels that were singled out for evaluation of the methodology may be extended on the same way if more time was available. The computer program was completely debugged and goof-proofed at any moment and should be very easy in a follow-up effort after this research.

The merit of this method of optimal filter design for a certain class of chosen speech sounds is that it presents the possibility of a priori evaluation and optimization with speed, automation and continuous control and monitoring that only a computer could provide.
CHAPTER V

RECOMMENDATIONS

1. Computer Program Refinements

The capabilities developed into the computer program during this research has proven to save for the experimentor a great amount of time and nerve. As it is now, this program may be used to extend the investigation beyond the class of sounds and the number of speakers chosen in this research.

A follow-up effort would be to extend the number of speakers to a considerably larger number and have each sound (as vowels alone or as vowels linked with ending consonants) repeated several more times than twice at different pitches. They should be spoken with a 1 to 2 second pause in between to facilitate inputting to computer memory later on. Words with consonant should be given considerable attention, and if necessary another threshold switch can be built similarly to that of the Preconditioning Unit to channel properly vowels and consonants for different gains of amplification before prototype making and recognition.

The computer program may be refined one more step by including an automatic checking point to compute and report about the rate of recognition after each cycle of training and to stop the training should the rate of recognition continuously decreases with increasing the number of training cycles and to recover the prototypes which give the highest rate of recognition.
2. Optimal Filter Implementation

Using the spectrums of this research an effort can be directed toward the template type of filters using photographic processes and a bank of optic fiber rods of different resonance frequencies along with phototransistors and space/frequency encoder-decoders.

3. Application of the Research Results

The results of this research prove that the auto-correlation function of the time record of the speech sounds have enough characteristics to make each sound unique and thus separable from other sounds - at least within a limited class of sounds.

If the refined scheme works reliably with words with consonants, the applications would be many, in both time and frequency domains. In frequency domain, the application will be to provide spectrums with enough resolution to help build teaching aids for the deaf. In the time domain, the scheme can be used as characteristic feature extraction elements for speech recognition. Investigation has shown that for sampling at same rate as maintained throughout this research but at number of points of samples limited to 128 recognition was still good to the point of successfully recognizing at 100% for 5 male speakers of about the same age at the time of simulation of the process. Each recognition at this number of samples would take less than 2 seconds on the PDP-9 from the time of inputting the sound to the time result was reported. Hardware correlator or faster computer should be able to recognize in real time. As Cannon (4) suggested in his work, if the delay time of each correlation function can be made short enough, at glottal pulses rate for instance, our scheme using the auto-correlation functions as
characteristic features for automatic recognition could be channelled into, at least, recognizing the parts of continuous speech containing vowels.
CHAPTER VI

1. REFERENCES

2. APPENDICES

A. Computer Program for Data Acquisition, and Data processing for cross correlation, Autocorrelations, Averaging, Block input--output Transfer, recognition.

B. Computer Program for Data Acquisition, and Data processing for FFT (direct and inverse), Cepstrum, Spectrum Smoothing, and curve plotting.

C. Computer Program for reading paper tape into computer disk 6, and punching paper tape for data from disk 6.

D. Computer Subroutine POINT for concurrent scope and X-Y plotter output.

E. Pre-emphasis characteristics.

F. Schematic Diagram of the Preconditioning Unit.

G. Details on Computer Program for Correlation.
REFERENCES


APPENDIX A
THIS PROGRAM WAS WRITTEN FOR A DIGITAL EQUIPMENT CORPORATION'S PDP-9 COMPUTER WITH LINE PRINTER,
DISK STORAGE AND A/D AND D/A CONVERTERS, THIS PROGRAM IS COMPOSED OF 5 DIFFERENT SUBROUTINES TO BE CHAINED TOGETHER, USING THE OVERLAY FEATURE OF THE PDP-9 SYSTEM,
 THESE SUBROUTINES ARE: "MASTER" AND "POINT" TO BE MADE THE RESIDENT CODE IN THE EXECUTE PROGRAM BUILT WITH THE SYSTEM CHAIN PROGRAM, "ACOR1" AND "PPR" TO BE COMBINED INTO LINK1, AND "ACOR2", TO BE MADE INTO LINK2, SUBROUTINE "POINT", FOR SCOPE AND X-Y PLOTTER IS MODIFIED BY THE AUTHOR FROM A VERSION AVAILABLE ON THE SYSTEM, THIS MODIFIED VERSION OF "POINT" ALLOWS THE CONCURRENT PLOTTING OF DATA ON THE SCOPE OR THE X-Y PLOTTER.
THIS PROGRAM ALLOWS TO INPUT INTO THE COMPUTER DISK 6 VIA THE A/D-CONVERTER, DATA FROM ANY SOURCE OF VOLTAGES PRODUCING FROM 1 TO A OR-10 VOLT MAX, (USE ATTENUATORS IF NECESSARY), IT ALLOWS FURTHER TO PROCESS THESE DATA IN SEVERAL WAYS, FOR A COMPLETE DESCRIPTION OF ALL THE OPTIONS AVAILABLE, REFER TO THE NEXT FEW PAGES AND REFERENCES THEREOF,
THE MAIN OPTIONS OF THIS PROGRAM ARE:
  1) AUTO- OR CROSS-CORRELATION FUNCTIONS OF DATA FROM 1 TO 256 POINTS, AND BEYOND 256 POINTS, ANY NUMBER OF MULTIPLE OF 256 UP TO 8192 POINTS, FOR AUTO- OR CROSS-CORRELATION OF 256 POINTS, IT WOULD TAKE ABOUT 2 SECONDS, THE TIME FOR CALCULATION INCREASES OR DECREASES PROPORTIONATELY WITH THE SQUARE OF THE NUMBER OF POINTS TO BE CORRELATED, RESULTS ARE NORMALIZED WITH RESPECT TO THE MAXIMUM VALUE AND ASSIGN THE NUMBER 250(10) TO THIS MAXIMUM VALUE FOR THE PURPOSE OF PLOTTING ON THE SCOPE OR THE X-Y PLOTTER.
  2) FAST-FOURIER TRANSFORM OF EITHER THE INPUT VIA THE A/D-CONVERTER OR OF THE CORRELATION FUNCTION JUST CALCULATED, RESULTS, READY TO BE PLOTTED ON THE SCOPE OR ON THE X-Y PLOTTER ARE NORMALIZED WITH RESPECT TO THE MAXIMUMS, CAN BE PLOTTED OUT ARE THE SPECTRUM IN LINEAR AMPLITUDE, THE SPECTRUM IN DECIBELS (MAX=0 DB.), AND THE CUMULATIVE POWER OF THE SPECTRUM (MAX=100%), THE PRINTOUT ON THE LINE PRINTER GIVES ALSO THE REAL AND IMAGINATIVE PARTS OF THE SPECTRUM, A CEPSTRUM CAN BE CALCULATED AFTER A FFT HAS BEEN DONE, BY JUST ONE TELETYPE INPUT, SMOOTHING WITH CEPSTRUM(SEE REFERENCE) CAN BE DONE ALSO BY JUST ANOTHER TELETYPE INPUT.
AFTER ONE FFT OPERATION, ANOTHER FFT(DIRECT OR INVERSE) CAN BE PERFORMED ON THE RESULTS OF THE LAST FFT BY JUST A FEW TELETYPE INPUTS.
BEFORE THE FFT OF AN AUTO-CORRELATION FUNCTION IS TAKEN, THE NEGATIVE-DELAY PART OF THIS FUNCTION IS CALCULATED AND PLACED ON THE LEFT OF THE PART OF THIS AUTO-CORRELATION FUNCTION WITH POSITIVE DELAY, THIS GIVES BETTER RESOLUTION IN THE SPECTRUM PRODUCED FROM THAT AUTO-CORRELATION FUNCTION. THIS OPTION, CALLED UPON BY 33, CAN OPERATE ON ANY RECORD TRANSFERRED TO THE RECORD HEADER BY BLOCK 192, BEFORE A FFT IS CALLED UPON, THIS OPTION PERMITS THE RECORDING ON SOUND TAPE RECORDERS SPEECH SOUNDS WHICH ARE REPRESENTED BY AUTO-CORRELATION FUNCTIONS FOR EVALUATION, ALL OTHER DATA CAN BE GIVEN A HANNING OR A HANNING WITH TRAILING ZEROES FOR BETTER RESOLUTION IN THE SPECTRUM PRODUCED BY FFT FROM THOSE DATA,

ANOTHER PROGRAM WRITTEN BY THE AUTHOR CALLED "PPPR" PERMITS THE INTERCHANGE OF DATA IN RANDOM BLOCKS BETWEEN DISK 6 AND DEC TAPe 1 AND BETWEEN THESE UNITS AND THE PAPER TAPE READER AND PUNCH, AND THE LINE PRINTER, AND GIVES THUS VIRTUALLY COMPLETE FREEDOM FOR INPUT/OUTPUT, SAVSTA IS WHERE TO BEGIN TO SAVE ANY NUMBER OF BLOCKS UP TO 68 AT A TIME IF SAVSTA IS PLACED INITIALLY AT 400(10), THE SAVING BLOCKS AUTOMATICALLY INCREMENT EVERY TIME THE OPTION SAVE IS CALLED UPON BY TYPING 10, BY USING OPTION 9 LATER ON, DATA CAN BE TRANSFERRED TO ANYWHERE ELSE FOR PROCESSING, SAVSTA CAN BE CONVENIENTLY PLACED AT 400(10), AVESTA IS WHERE TO BEGIN TO SAVE AVERAGES, THIS CAN BE CONVENIENTLY PLACED AT 500(10), AND IS AUTOMATICALLY INCREMENTED EVERY TIME AN AVERAGING IS CALLED UPON BY OPTION 10, BLOCKS FROM 400(10) TO 467(10) AND 500(10) TO 567(10) ARE UNTOUCHED BY THIS PROGRAM, EXCEPT FOR ABOVE PURPOSES, BLOCKS FROM 468 TO 499 ARE FOR AUTOMATIC SAVING OF DATA READ IN FROM A/D CONVERTER,

CONVSW CAN BE SET TO 0 FOR NORMAL AMPLITUDE CONVERSION, 1 FOR LOGARITHM OF AMPLITUDE, 2 FOR DERIVATIVE OF INPUT, AND ANYTHING ELSE, E.G., 4 FOR FREE CHOICE,

PTSW = 0 FOR FREE CHOICES AND ANYTHING ELSE, E.G., 256, FOR AUTOMATIC ASSIGNMENT OF NUMBER OF POINTS TO BE READ IN FROM THE A/D CONVERTER,

RECOSW = 1 FOR AUTOMATIC JUMP TO RECOGNITION, SKIPPING THE PRINTOUT OF EVERYTHING EXCEPT THE FINAL CODE NUMBER; RECOSW = ELSE, FOR PRINTOUT OF MAX. VALUE OF CORREL. FOR MONITORING OF ADEQUATE EJECT
INPUT TO THE A/D-CONVERTER,
// ADSH = 1 WILL MAKE COMPUTER WAIT AUTOMATICALLY TO READ
// IN FROM A/D-CONVERTER A NUMBER OF DATA POINTS SPECIFIED
// BY PTSH, THIS FEATURE SPEEDS UP CONSIDERABLY THE EVAL-
//UATION OF RECOGNITION PROCESSES,
// MAXSK = 1 FOR SKIPPING THE PRINTOUT OF THE CROSS-CORREL,
// FUNCTION VALUES (AT ZERO DELAY) IN THE RECOGNITION,
//
// RECSTA IS THE STARTING BLOCK WHERE TO BEGIN THE RECOGNI-
// TION, THIS OPTION GIVES EXTREME FLEXIBILITY IN COMPARING
// THE DIFFERENT PROCESSES OF RECOGNITION,
//
// FOR DETAILS OF OPTIONS PROVIDED IN THIS PROGRAM, SEE
// DISSERTATION "COMPUTER DESIGN AND EVALUATION
// OF OPTIMAL FILTERS FOR A CLASS OF SPEECH SOUNDS BY
// CORRELATION TECHNIQUES", BY LE-BA TUAN, OSU, 1971,
//
// OPTION NUMBER 19(10) IS FOR TAKING THE DIRECT OR INVERSE
// FFT OF A SPECTRUM JUST CALCULATED, IMMEDIATELY BEFORE
// THIS OPTION IS CALLED UPON, WHAT IT DOES IS TAKING THE
// RESULTS OF THE LAST FFT WHICH WERE STORED IN A RECORD
// HEADED BY BLOCK 256(10) AND READY THEM FOR INPUT OF
// ANOTHER DIRECT FFT OR INVERSE FFT AS WE
// WISH AFTER ANSWERING A FEW QUESTIONS TYPED OUT BY COMPUT-
// ER,
//
// THE X-AXIS IS ALWAYS EXPANDED OR COMPRESSED AUTOMATICALLY
// TO PLOT ON A LINE OF 511 POINTS ANY NUMBER OF DATA POINTS
// FROM 128 TO 256, AND BEYOND 256, ANY MULTIPLE OF 256, UP
// TO 8192 POINTS, IF NUMBER OF DATA POINTS TO BE PLOTTED IS
// UNDER 128 POINTS, THE X-AXIS IS EXPANDED TO THE MAXIMUM
// AS FOR 128 PTS, THE Y-AXIS IS ALWAYS SCALED TO HAVE MAX,= 
// + OR -250 IN LINEAR SCALE, OR 0 DB IN LOGARITHMIC SCALE,
// OPTION 500 OR 501 DRAWS A Y AXIS WITH ONLY THE
// POSITIVE Y, AND THIS AXIS IS DIVIDED INTO INTERVAL OF
// 16 DB EACH, BELOW THE X AXIS IT DRAWS ANOTHER AXIS
// WHICH IS DIVIDED INTO 16 INTERVALS, THIS OPTION OF AXES
// IS THOUGHT OF FOR THE DB-PLOTS,
//
// OTHER THAN THE AVERAGING OPTION CALLED UPON BY 13
// TWO MORE ADVANCED VERSIONS FOR AVERAGING CAN BE CALLED
// UPON WHERE AVAILABLE (SEE FRONT PAGE OF EACH SUBROUTINE),
// ONE OF THEM IS CALLED BY 113. THIS MAKES COMPUTER PRINT
// OUT:
// TOTAL #NB: 1ST BLK: NB TO BE GRID:
//
// TOTAL #NB: TOTAL NUMBER OF BLOCKS INVOLVED IN THE AVERAG-
// ING PROCESS,
// 1ST: FIRST BLOCK OF THE WHOLE RECORD CONTAINING TOTAL
// #NB,
//NB TO BE GRID: NUMBER OF BLOCKS TO BE GROUPED TOGETHER
//TO FORM AN AVERAGED BLOCK.
//WHEN ALL THESE INFORMATION BITS ARE FILLED IN, COMPUTER
//WILL TYPE:

//WT BK1:

//THIS IS THE WEIGHT ASSIGNED TO EACH FIRST BLOCK IN THIS
//AVERAGING PROCESS,
//AFTER THIS LAST INFORMATION HAS BEEN FILLED IN,
//COMPUTER WILL PICK UP THE FIRST BLOCK OF THE RECORD,
//MULTIPLY EACH WORD OF THE BLOCK BY THE
//WEIGHT, AND ADD TO IT THE CORRESPONDENT WORD OF THE NEXT
//SUCCESSIVE BLOCKS UP TO (NB TO BE GRID MINUS ONE). Suppose
//THE NUMBER OF BLOCKS TO BE GROUPED IS 4 AND THE TOTAL
//NUMBER OF BLOCKS OF THE RECORD IS 40, THEN THE FIRST 4
//BLOCKS WILL BE GROUPED TOGETHER AS WE JUST SAID,
//AND THEN BLOCKS 5 TO BLOCK 8 INCLUSIVE WILL BE GROUPED
//NEXT, AND THE AFTER THAT, BLOCKS 9 TO BLOCK 12 INCLUSIVE
// WILL BE GROUPED, AND SO ON UNTIL ALL 40 BLOCKS ARE AVE-
//RAGED INTO 10 BLOCKS WHICH WILL BE STORED AT LOCATIONS ON
//DISK 6 PRINTED OUT EVERY TIME AN AVERAGE HAS BEEN
//PERFORMED,
//ANOTHER OPTION OF AVERAGING INVOLVES SEVERAL RECORDS
//HAVING SEVERAL BLOCKS EACH, THIS OPTION IS CALLED UPON
//BY 213 AT THE TTY TYPE, COMPUTER WILL PRINT OUT THE SAME
//MESSAGE TO BE ANSWERED AS IN OPTION 113. HOWEVER, THE
//AVERAGING PROCESS NOW WILL GROUP BLOCK 1 OF RECORD 1
//WITH BLOCK 1 OF RECORD 2, AND THAT OF RECORD 3 ECT...
//UP TO BLOCK 1 OF THE RECORD NUMBER EQUAL
//TO THE TOTAL NUMBER OF BLOCKS DIVIDED BY THE NUMBER OF
//BLOCKS TO BE GROUPED, LOCATIONS OF SAVING ALSO WILL BE
//PRINTED OUT AS IN 113,
//OPTION 29 IS FOR TRANSFER OF DATA IN RANDOM BLOCKS
//BETWEEN DISK 6 AND THE TWO UNITS OF DECTAPE,
//DISK 6 IS HERE ASSIGN THE UNIT NUMBER 3 AND DT1 IS
//ASSIGNED UNIT NUMBER 8.

EJECT
.TITLE MASTER /LE-BA TUAN/8/7/71,

THIS PROGRAM CHAINED WITH THE SUBROUTINES
ACOR1, ACOR2, POINT AND PPPR GIVES COMPLETE CAPABILITY FOR
INTERCHANGE OF DATA AT RANDOM ACCESS BETWEEN CORES,
DISK 6, DECTAPE 1, PAPER READER, PAPER PUNCH, A/D CONVERT-
ER, D/A CONVERTER, SCOPE, X-Y PLOTTER AND LINE PRINTER,
SEE DETAILS ABOUT OPTIONS AVAILABLE ON THE HEADING
OF EACH SUBROUTINE.

.IODEV 4

.GLOBL ,FP,,FW,,FE,,FR,,FF
.GLOBL ACOR1,ACOR2,PPPR,ACCEPT
.DEFIN WRITED,IODEV,FMT,BUF,?Q

JMS*, FW

,DSA Q

,DSA FMT

JMS*, FE

,DSA BUF

JMS*, FF

JMP ,+2

Q IODEV

,ENDM

,DEFIN WRITMN,IODEV,FMT,COL,ROW,BUF,?C1,?C2,?P,?Q

LAC (-ROW

DAC C2

JMS*, FW

,DSA Q

,DSA FMT

LAC (-COL

DAC C1

JMS*, FE

P

,DSA BUF

ISZ P

ISZ P /SKIP 1 VALUE IN INTEGER PRINTING,

ISZ C1 /PUT ISZ P IN IF SKIP & REAL PRINTING IS

JMP P-1 /NEEDED

JMS*, FF

ISZ C2

JMP P-6

LAC (BUF

DAC P

JMP ,+4

Q IODEV

C1 0

C2 0

,ENDM

,EJECT
.DEFIN WRITEH, IODEV, FMT, ?Q
JMS*, FN
.DSA O
.DSA FMT
JMS*, FF
JMP, +2
Q IODEV
. ENDM
.DEFIN READ, IODEV, FMT, BUF, ?Q
JMS*, FR
.DSA O
.DSA FMT
JMS*, FE
.DSA BUF
JMS*, FF
JMP, +2
Q IODEV
. ENDM
PARIN1 WRITEH 4, FRMT1
JMS* ACCEPT; JMP, +3; DATS: PLACE
BEGIN0 LAC PLACE#
SAD (1750); JMS* ACOR1
//FOR CORRELATION PROGRAM AND OPTIONS,
SAD (3720); JMS* ACOR2 //FOR FFT AND OPTIONS,
SAD (5670); JMS* PPPR //FOR PPPR AND OPTIONS
JMS DOAGAN; JMP PARIN1
DOAGAN 0; WRITEH 4, FRMT75
JMP* DOAGAN
DATS 4
FRMT1 ,ASCII "(1X, 27HCOR: 1000 FFT: 2000 PPPR: 3000)"
FRMT75 ,ASCII "(1X, 8HDD AGAIN)"
.END PARIN1
OPTION AVAILABLE EVERY TIME "GO!" IS TYPE OUT BY THE COMPUTER:
0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 13, 14, 15, 16, 23, 24, 29, 33, 40, 100, 113
124, 213, 300, 301, 400, 401, 2000, 3000.

.EJECT
.TITLE ACOR1-COR/LE-BATUAN/8/1/71/1745.

ADACTV=701601
ADCLON=701624
ADCLOF=701644
CADAPI=701664

.GLOBL A, AA, AC, AD, AE, AF, AJ, AK, AL, AM, AN, AX, CO, ALOG10
.GLOBL ACOR1, ACOR2, PPR
.GLOBL FP, FW, FE, FR, FF, POINT, LINE
.DEFIN WRITED, IODEV, FMT, BUF, ?Q
JMS* , FW
.DSA Q
.DSA FMT
JMS* , FE
.DSA BUF
JMS* , FF
JMP , +2
Q
IODEV

.ENDM

LAC (-ROW
DAC C2
JMS* , FW
.DSA Q
.DSA FMT
LAC (-COL
DAC C1
JMS* , FE
.DSA BUF
ISZ P
ISZ P /SKIP 1 VALUE IN INTEGER PRINTING,
ISZ C1 /PUT ISZ P IN IF SKIP & REAL PRINTG IS
JMP P-1 /NEEDED
JMS* , FF
ISZ C2
JMP P-6
LAC (BUF
DAC P
JMP , +4
Q
IODEV

C1
0
C2
0

.ENDM

.DEFIN WRITEX, IODEV, FMT, ?Q
JMS* , FW
.DSA Q
.DSA FMT
.EJECT
JMSP, FF
JMP, +2
Q
IODEV
,ENDM
.DEFIN READ, IODEV, FMT, BUF, ?Q
JMSP, FR
,DSA Q
,DSA FMT
JMSP, FE
,DSA BUF
JMSP, FF
JMP, +2
Q
IODEV
,ENDM
DATE
, BLOCK 42 //
ACOR1 0 /CORRELATION PROGRAM
JMSP, FP
WRITEH 4, FRMT63
START
WRITEH 2, FRMT6
*INIT 4, 0, PARIN1
,READ 4, 2, DATE, 34
,WAIT 4
,CLOSE 4
WRITEH 4, FRMT9
JMSP ACCEPT; JMP, +6; DATS; SAVSTA; AVESTA; RECWRD
AOSW
WRITEH 4, FRMT91
JMSP ACCEPT; JMP, +10; DATS; CONVSW; PTSW; RECONSW
MAXSK; RECSTA; LNCOEF
LAC SAVSTA#; DAC TRANSA+2; LAC AVESTA#; DAC TRANAV+2
LAC TOL1; CLL; RAL; DAC TOL2#
PARIN1
WRITEH 4, FRMT1
JMSP ACCEPT; JMP, +3; DATS; PLACE
BEGIN0
LAC PLACE#
DAC DUMP /MAKES PLOTTING AXES INDEPENDENT,
JMSP ROUTIN; SAD (3); JMP BEGIN1
JMSP DOAGAN; JMP PARIN1
DOAGAN 0; WRITEH 4, FRMT75
JMP* DOAGAN
BEGIN1
WRITEH 4, FRMT6
JMSP ACCEPT; JMP BEGIN1
RECUPT
WRITEH 4, FRMT6
JMSP ACCEPT; LAC (4); DAC PLACE; JMP RENT1
//
RECIIN
JMSP WRITNB; JMSP ACSPNB; LAC SPNB; DAC RECC#
JMP DATINO
RECON DZM C; ISZ C; DZM KP; LAC RECC#; SNA; JMP RECIIN
JMSP INV; DAC CT
LAC (3), DAC MP; LAC RECSTA#; TAD (-100); DAC MQ
//BLOCS FOR RECOGNITION WITH CROSS-CORREL, START AT RECSTA
,EJECT
LAC RECD#; DAC WORD#; JMS INV; DAC MWORD#  
DAC FCT  
JMS REMAIN; DZM V#; JMS FINDP  
REC1 JMS FIND0; DZN L; LAC PLACF; SAD (20); JMP EUCLID  
JMS SUBBR; SMA  
REC3 JMS INV; DAC PR; TAD KP; SPA; JMP CHANGE  
REC2 ISZ MU; ISZ C; ISZ CT; JMP REC1; JMS WRITW  
DZM PLACF; LAC ADSW#; SAD (1); JMP ADINAU  
JMP DOUTU  
CHANGE LAC PR; JMS INV; DAC KP  
LAC MAXSK#; SAD (0); JMS WRITKQ; LAC C; DAC V  
JMP REC2  
EUCLID JMS COMPAR; LAC L; JMP REC3  
COMPAR 0; JMS PLOU1  
REC0 LAC* BUF1P; TAD TOL1; JMS INV; TAD* BUF2P  
SMA; JMP REJECT  
TAD TOL2; SPA; JMP REJECT  
OK1 ISZ L  
REJECT ISZ BUF1P; ISZ BUF2P  
ISZ FCT; JMP REC0; JMP* COMPAR  
//  
WRITKQ 0; WRITED 4,FRMT77,KP  
JMP* WRITKQ  
WRITW 0; LAC KP; TAD RECMIN; SMA; JMP OKWR; JMP NOOKWR  
OKWR WRITED 4,FRMT9,V  
JMP* WRITW  
NOOKWR WRITEN 4,FRMT10  
JMP* WRITW  
RECMIN -400 /256(10),  
//  
AACOR LAC (-2); DAC C; LAC SPOINT; DAC POINTS; LAC (0)  
AACOS DAC TOSTBL; LAC (300); DAC FRSTBL; JMS PRELIM  
JMS TIOS; LAC (190); ISZ C; JMP AACOS; JMP BEGIN1  
//  
//  
.LST  
LNSTA LAC RECCYr; SNA; JMP RECINI  
WRITEH 4,FRMT61  
JMS* ACCEPT; JMP .+6; DATS; INDBLS; NSS; TINDBL  
GTTLNB  
WRITEH 4,FRMT63  
JMS* ACCEPT; JMP .+6; DATS; WRITVS; WRSKFN; TRIAL  
LNK  
*EJECT
RECON0  LAC TRIAL#; TAD (1); JMS INV; DAC TRIALC#
      LAC WRSKFN#; TAD (1); JMS INV; DAC WRICT#
      LAC GTLNLB#; TAD INDBLS; DAC UBBLNO#
      LAC (-2); DAC ELNS#
      LAC LNSK#; SAD (1); JMP .+2; JMP NOSKIP
      LAC (JMP RECONT); DAC LNSKSW; JMP RECONP
NOSKIP  LAC (NOP); DAC LNSKSW
//
//MAX, NUMBER OF LEARNING TRIALS FOR EACH INDIVIDUAL SOUND
//IS LIMITED TO 1,
RECONP  LAC INDBLS#; DAC INDBL1#; DZN HLNS#
      ISZ TRIALC; JMP ,+2; JMP CANTDO
RECONQ  DZN NL#
RECONS  DZN C; ISZ C; DZN KP
      LAC RECCT; JMS INV; DAC CT
      LAC INDBL1; TAD NL; DAC MP; LAC RECSTA; TAD (-100)
      DAC MQ
      LAC RECORD; DAC WORD; JMS INV; DAC MWORD
      DAC FCT
      JMS REMAIN; DZN V; JMS FINDP
RECS1S  JMS FINDQ; DZN L
      JMS SUBBR; SHA
RECS3S  JMS INV; DAC PR; TAD KP; SPA; JMP CHANGS
RECS2S  ISZ MQ; ISZ C; ISZ CT; JMP REC1S; JMS WRTIZ
LNSKSW  NOP /OR JMP RECONT
      LAC NL; JMS* .AE; LAC NSS#; TAD (1); DAC RIGHTV#
      SAD V; JMP ,+2; JMP LEARS2
RECONT  LAC ELNS; DAC ELNSC#
      LAC NL; TAD (1); DAC NL; SAD TINDBL; JMP .+2
      JMP RECONS
      LAC INDBL1; TAD TINDBL#; DAC INDBL1
      SAD UBBLNO; JMP ,+2; JMP RECONQ
      LAC HLNS; SAD (1); JMP RECONP
      WRITEx 4,FMT62
      LAC TRIALC; TAD TRIAL; TAD (1); DAC V
      WRITED 4,FMT65,V
      JMP PARIN1
//
CHANGS  LAC PR; JMS INV; DAC KP
      LAC C; DAC V
      JMP RECS2S
LEARS2  ISZ ELNSC; JMP ,+2; JMP RECONT
      LAC (2); DAC SPNB; JMS AVEROU
      LAC RIGHTV; TAD RECSTA; TAD (-1)
      DAC SPKB; DAC LNPL
      DAC* FP; ISZ FP; LAC MP; DAC* FP
      LAC (JMP LEARN3); DAC LNSW; JMS ACC2S
      LAC LNPL; DAC TRAN+2; LAC (BUF2); DAC TRAN+3
      JMS DKOUT; LAC (1); DAC HLNS
      JMP RECONS
*EJECT
// WRITZ
0; LAC WRICT; TAD (1); DAC WRICT; SPA; JMP* WRITZ
LAC WRITVS#; SAD (1); JMP *, +2; JMP* WRITZ
JMS WRITW; JMP* WRITZ
//
CANT00
WRITEH 4,FMT64
JMP PARIN1
//
,"NOLST
//
SUBBR
0
DZM PR; DZM K
LAC (BUF2-1); DAC BUF2P
LAC (BUF1-1); DAC BUF1P
LAC MWORD; DAC END
LAC WORD; CLL; RAR; DAC HWORD#
T117
ISZ BUF1P; ISZ BUF2P
LAC# BUF1P; JMS# ,AD; LAC# BUF2P
JMS# ,AE; LAC WORD; DAC TAI1
LAC# ,CO; SMA; JMS INV
T217
TAD HWORD; SMA; JMP T118
LAC TAI; SPA; TAD (-2); TAD (1)
T119
TAD PR; DAC PR
ISZ END; JMP T117; JMP* SUBBR
//
T118
LAC TAI; JMP T119
//
,EJECT
FRMT1, ASCII "(1X,3HGO:)
FRMT11, ASCII "(1H+,27HCONVERTED DATA FOR MAX=250,)
FRMT2, ASCII "(28X,21HMULTIPLIER=+.122130,)
FRMT4, ASCII "(1X,8HMAX PH1=16)"
FRMT5, ASCII "(1X,6HPLACE:)
FRMT3, ASCII "(11)"
FRMT6, ASCII "(1X,5H#PTS:)
FRMT7, ASCII "(1H+,16X,15H,NUMBER OF BLK=I2)"
FRMT8, ASCII "(1X,5HMEM0:)
FRMT9, ASCII "(1X,24HSAVS: AVES: RECw: ADAU:1)"
FRMT91, ASCII "(1X,32HDB:1 PT: DRE:1 MXS:1 RSTA: LNC:)"
FRMT19, ASCII "(1X,9HDB:1DER:2)"
FRMT12, ASCII "(1X,5H#BLK:)
FRMT23, ASCII "(1X,1H:)
FRMT22, ASCII "(1X,7HLIST'EM)"
FRMT31, ASCII "(1X,12,8HB,#1,GO:)
FRMT32, ASCII "(1X,12,8HB,#4,GO:)
FRMT65, ASCII "(1X,12,8HB,#2,GO:)
FRMT67, ASCII "(1H+,18HDWRD: CWRD: PLACE:)"
FRMT35, ASCII "(1X,12HPLACE: #PTS:)"
FRMT63, ASCII "(1X,16(1X,I6))"
FRMT75, ASCII "(1X,16HD0 AGAIN,PLEASE.)"
FRMT76, ASCII "(1X,4HMAXP17)"
FRMT77, ASCII "(1X,I5)"
FRMT7, ASCII "(1X,1HAI4)"
FRMT8, ASCII "(1X,1HSI4)"
FRMT9, ASCII "(1X,I2)"
FRMT10, ASCII "(1X,11HNOT REC'BLE)"
FRMT22, ASCII "(1X,13HFR: TO: #PTS:)"
FRMT56, ASCII "(1X,7HWT BL1:)"
FRMT57, ASCII "(1X,4HYES!)"
FRMT60, ASCII "(1X,13HCOR & OPTIONS)"
//
FRMT61, ASCII "(1X,28HIN DBLS: NSS: TINDBL: GTTLNB:)"
FRMT62, ASCII "(1X,13THEARNING DONE)"
FRMT63, ASCII "(1X,30HWRIVS:1 WRSKFN: #TRIAL: LNSK:1)"
FRMT64, ASCII "(1X,12HCAN'T LEARN!)"
FRMT65, ASCII "(1X,8H#TRIALS:16)"
FRMT66, ASCII "(1X,15HNO CONVERGENCE!)"
//
EJECT
DTIN
  ,INIT 13,0,PARIN1
TTRAN1
  ,TRAN 10,0,0,BUF1,256
  ,WAIT 10
  ,CLOSE 10
  ISZ TTRAN1+2; JMP# DTIN
DTOUT
  ,INIT 10,1,PARIN1
TTRANO
  ,TRAN 10,1,0,BUF1,256
  ,WAIT 10
  ,CLOSE 10
  ISZ TTRAN0+2; JMP# DTOUT

//
//
DTIO
  ,INIT 10,1,PARIN1
  ,TRAN 10,1,0,BUF1,256
  ,WAIT 10
  ,CLOSE 10
  ISZ TTRAN0+2; JMP# DTIO

//
//
PRR
  ,INIT 10,1,PARIN1
  ,TRAN 10,1,0,BUF1,256
  ,WAIT 10
  ,CLOSE 10
  ISZ TTRAN0+2; JMP# PRR

//
//
DOIT40
  ,INIT 10,1,PARIN1
  ,TRAN 10,1,0,BUF1,256
  ,WAIT 10
  ,CLOSE 10
  ISZ CT; JMP# DOIT40
  JMP# DTIO
  ,EJECT
//
TDKIN  JMS DKin; JMP DOIT41  
//
TDKOUT JMS DKOUT; JMP DOIT42
TDOUT  JMS DTOUT; JMP DOIT42
TDIN   JMS DTIN; JMP DOIT41
//
FRMT38 'ASCII "(1X,23HTO UNIT#: STABLK: #PTS:"
FRMT37 'ASCII "(1X,17HFR UNIT#: STABLK:)"
DKDTIO JMS DTIO; LAC (35); DAC PLACE
       JMP DATING
       IODEV 10
//
DATS  4
TOL1 12 / TOLERANCE FOR EUCLIDEAN METRIC.
//
ADINAU LAC (4); DAC PLACE; JMP COR
//
COR   LAC PTSW#: SZA; JMP STC
//OPTION PT: GIVES FREEDOM FOR CHOICE OF #PTS
//IF PT:0, OR AUTOMATICALLY FIX #PT EQUAL TO
//ANY NUMBER OTHER THAN 0,
       WRITE 4,FRTM6
STA  NOP; JMS ACCPT; JMP STB
ACCT  0; JMS* ACCEPT
       JMP ,+3
       DSA DATS
       DSA POINTS#; JMS PRELIM; JMS PREAC; JMP* ACCPT
PRELIM 0; LAC POINTS; DAC SPOINT#: JMS* ,AE; LAC (400)
       JMS INV; DAC PNBC; DAC CTC#: SAU (0); LAC (-1)
       DAC NB#: JMS INV; DAC NB#: LAC* ,CO
       SAO (0); JMP ,+2; JMP ,+2; LAC (400); DAC WORD
       JMS INV; DAC MWORD
       LAC NB; JMS* ,AD; LAC WORD; DAC NB#: JMP* PREAC
PREAC 0; LAC NB; DAC NF#: LAC WORD; DAC WORF#
       JMP* PREAC
//
STC   DAC POINTS; JMS PRELIM; JMS PREAC
STB   LAC ADINBL /STARTING BLOCK #BER HERE
       DAC TRANC+2
       JMP LD&F11
//
ADINBL 00724 /468(10),
//THIS IS WHERE TO BEGIN TO SAVE THE DATA READ IN
//FROM THE A-D CONVERTER FOR ANY LATER PROCESSING
//BEFORE A NEW READ-IN FROM THE A-D CONVERTER, THE
//OPTIONS "CORREC", "RECUP", ETC...DIG
   .EJECT
//UP THE INPUT DATA FROM THIS BLOCK ON.
BUF1    ,BLOCK 400
BUF2    ,BLOCK 400
F       ,BLOCK 1000
DUMP    ,BLOCK 2
TWO     2
FOUR    4
RENT6   LAC WORF; DAC WORD; JMS INV; DAC MWORD
         LAC NF; DAC NB; JMS INV; DAC NBC; JMP RENT1
LDBF0   LAC MWORD
         DAC* (32)
         LAC (BUF1-1)
         DAC* (33)
         / LAC NBC
         / DAC BLKCT#
INP     ADACTV
         / ADCLON
         IORS
         AND (4)
         SZA
         JMP .-3
         ISZ BLKCT /DOUBLE BUFFERING
         JMP LDBF2
OUT1    LAC (BUF1)
         JMP DKOOUTF
LDBF1B  0; LAC (-1); DAC* (32); DAC FCT
         LAC (BUF1-1); DAC* (33); ADACTV; JMP* LDBF1B
LDBF1I  LAC NBC; DAC BLKCT#; JMS LDBF1B; ADCLON
LDBF1C  IORS; AND (4); SZA; JMP .-3
TEST    LAC (BUF1); DAC BUF1P
TESI    LAC* BUF1P; SPA; JMS INV
         TAD MTWRD; SPA; JMP TESS; JMP LDBF0
TESS    ISZ FCT; JMP TESI; JMS LDBF1B; JMP LDBF1C
         //
         /MTWRD -146 (=102(10)), AROUND 26 DB BELOW MAX,
         MTWRD -20
         //THIS MAKES COMPUTER WAIT TILL INPUT EXCEEDS THIS
         //THRESHOLD TO BEGIN TO READ IN VIA A/D-CONVERTER,
LAC (BUF2)
JMP DKOUTF

LDBF1 LAC MWORD
DAC* (32)
LAC (BUF1-1
DAC* (33)
ADACTV
LAC (BUF2
JMS DKOUA
IORS
AND (4
SIZE
JMP .-3
ISZ BLKCT
JMP LDBF2
LAC (BUF1
JMP DKOUTF

DKOUA 2: DAC TRAN+3; JMS DKOUT; JMP* DKOUA
DKOUTF JMS DKOUA
ADCOF
JMP RENT1

//
DUM ,BLOCK 2
MULT ,BLOCK 2
R2047 000013; 377700
R12213 454775; 372076 /250/2047.
INCOEF 216007; 227003 /,755032E+02=250/ALOG10(2048),

//
RENT1 LAC CONVSW#; SAD (1); JMP DB1; SAD (2); JMP DER0
SAD (3); JMP DB2
WRITE 1,FRMT19
READ 4,FRMT3,DUMP
LAC DUMP; SAD (1); JMP DB1; SAD (2); JMP DER0
JMP Db2

DB1 LAC (JMP INDB); DAC DBSW; LAC (JMP SIGNF)
DAC SIGNW
LAC (NOP); DAC DERSW; JMP RENS1

DB2 LAC (NOP); DAC DBSW; DAC SIGNSW; DAC DERSW
JMP RENS1

DER0 LAC (JMP DER1); DAC DERSW; LAC (NOP); DAC DBSW
DAC SIGNW
JMP RENS1

RENS1 LAC ADINBL; DAC TRANI+2
INI LAC (1W2); DAC TRANO2+2
LAC (0); DAC TRAN01+2
LAC NBC
DAC ADBLCT#
LAC MWORD
DAC BUF1CT#
,EJECT
LAC (BUF1)
DAC BUF1P#
DAC TRAM1+3
LAC (BUF2)
DAC BUF2P#; DAC TRAN1+3; DAC TRAN02+3

CONVRT
JMS DK1N
LAC* BUF1P
ISZ BUF1P

DERSW
JMP DER1 /OR NOP

SIGNSW
JMP SIGNF /OR NOP

RENS3
JMS* .AX

DBSW
JMP IN0B /OR NOP.
JMS* .AX

.DSA R12213 /250/2047(10), MAKES P&Q MAX = 250,

ININ
JMS* .AX

INIO
DAC* BUF2P
ISZ BUF2P
ISZ BUF1CT
JMP CONVRT+1

LAC MW0RD
DAC BUF1CT

DK0
LAC PLACE
SAO (1
JMP DK01
SAD (2
JMP DK02
SAD (74); JMP DK02
SAD (106); JMP DK02/74 IS 60(1); 106 IS 70(10),
JMP DK012

DK01
JMS DKOUT1
JMP EX

DK02
JMS DKOUT2
JMP EX

DK012
JMS DKOUT1
JMS DKOUT2

EX
LAC (BUF2
DAC BUF2P
ISZ ABLC1T
JMP GOAD
JMP FINALE

GOAD
LAC (BUF1
DAC BUF1P
JMP CONVRT

DER1
JMS INV; TAD* BUF1P; JMP RENS3

SMA; JMP +2; JMP DER1

SZA1 JMP +2; JMP DER2

/DERP
LAC (401); JMP INIO

/DERN
LAC (-400); JMP INIO

/DER2
LAC (0); JMP INIO

.EJECT
//
SIGNF
SMA; JMP POSKP; JMS INV; DAC TA1
LAC (JMS* ,BA); DAC SW10
LAC TA1; JMP RENS3

POSKP
SNA; LAC (1); DAC TA1; LAC (NOP); DAC SW10
LAC TA1; JMP RENS3

INDB
JMS* ,AH; TA1; JMS* ALOG10; JMP +2; TA1
JMS* ,AK; INCOEF

SW10
JMS* ,BA /OR NOP
JMP ININ

//
DKOUT1 0
.INIT 3,1,PARIN1
TRAN01 .TRAN 3,1,0,BUF2,256
.WAIT 3
.CLOSE 3
ISZ TRAN01+2
JMP* DKOUT1

DKOUT2 1
.INIT 3,1,PARIN1
TRAN02 .TRAN 3,1,100,BUF2,256
.WAIT 3
.CLOSE 3
ISZ TRAN02+2
JMP* DKOUT2

FINALE
LAC RECCSW#; SAD (1); JMP BEGIN1
LAC PLACE
SAD (1)
JMP W1
SAD (2)
JMP W2
SAD (4)
JMP W12
JMP PARIN1

WRN6 0; WRITED 4,FRMT65,NB
JMP* WRN6

W2 JMS WRN6; JMP PT
W12 WRITED 4,FRMT32,NB
JMP PT
W1 WRITED 4,FRMT31,NB
JMP PT

SAV1 JMS SAVE; JMP DATINO
SAVE 0; LAC NBC; DAC PNBC; JMS PLROU2
LAC TRNSA+2; DAC V
SAVF JMS DKIN; JMS DKSAY; ISZ CT; JMP SAVF; JMS WRITV
JMP* SAVF

//
WRITV 0; WRITED 4,FMT8,V
JMP* WRITV
.EJECT
BEGIN1 LAC (-2) /THIS CONVERTS NEG,BERS TO 1'S COMPL,
DAC G# /NECESSARY FOR THE EAE COMMANDS,
LAC (0)

X3 DAC TRANI+2
DAC TRANO+2
LAC (BUF1); DAC TRANI+3; DAC TRANO+3
LAC NBC
DAC BLKCT

X2 LAC (BUF1
DAC BUF1P
LAC MWORD
DAC CT#
JMP DKIN

X1 LAC< BUF1P
SMA
JMP #+2

X0 TAO (-1
DAC< BUF1P
ISZ BUF1P
ISZ CT
JMP X1
JMP DKOUT
ISZ BLKCT
JMP X2
ISZ C
JMP #+2
JMP BEGIN2
LAC (ADD) M2 /SEE DECUS AD SUBROUTINE,
DAC X0
LAC (100
JMP X3

M2 -2
DKIN 0

,INIT 3,0,PARIN1
TRANI ,TRAN 3,0,0,BUF1,256
,WAIT 3
,CLOSE 3
ISZ TRANI+2
JMP* DKIN

DKOUT 0

,INIT 3,1,PARIN1
TRANSO ,TRAN 3,1,0,BUF1,256
,WAIT 3
,CLOSE 3
ISZ TRANO+2
JMP* DKOUT

DKSAV 0

,INIT 3,1,PARIN1
TRANSA ,TRAN 3,1,0,BUF1,256
,EJECT
.WAIT 3
.CLOSE 3

ISZ TRANSA+2; JMP* DKSAV

DKPRIN
JMS DKin; JMP D02

DTDKIN
LAC TOUTIT; SAD (10); JMP DTTIN; JMS DKin; JMP D02

DTTN
JMS DTIN; JMP D02

ACCPA
0; JMS* ACCEPT; JMP ,+3; DATS; DUMP; JMP* ACCPA

PRINTR
LAC CTG; DAC PNBC; JMS PLOU1; JMS PLOU2

DOIT2
LAC PLACE; SAD (35); JMP DTTKIN; JMP DKPRIN

D02
WRITEH 6,FRT2
WRITEH 6,FRT11

JMS PRINTH; ISZ CT; JMP DOIT2

DATINO
WRITEH 4,FRT1

PT
JMS ACCPA
LAC DUMP; JMS ROUTIN
SAD (3); JMP BEGIN1
SAD (41); JMP DACOCA /33(10)
SAD (0); JMP PLOTIS
SAD (310); JMP PLOTIP /200(10)
SAD (5); JMP PRINTR
SAD (12); JMP SAV1; JMS DOAGAN; JMP DATINO

ROUTIN
0; SAD (7); JMP PARIN1
SAD (174); JMP LNSTA /124(10)
SAD (3720); JMS* ACO02 /2000(10), TO GO TO FFT ,
SAD (5470); JMS* PPRR /3020(10), TO GO TO PPRR,
SAD (325); JMP ADVAVB /213(10), AVERAGES

//BETWEEN TO RECORDS OF BLOCKS,
SAD (1); JMP BACK1
SAD (2); JMP BACK1
SAD (4); JMP BACK1
SAD (27); JMP AACOR /23(10)
SAD (32); JMP LEARN1 /24(10)
SAD (10); JMP CORREC
SAD (161); JMP ADVAVE /ADVANCED AVERAGES,113(10)
SAD (144); JMP START; SAD (35); JMP DKDT10 /29(10)
SAD (11); JMP GTIO
SAD (15); JMP AVERAG; SAD (50); JMP RECUP
SAD (16); JMP RECINI; SAD (17); JMP RECON
SAD (454); JMS DRAWAS /300(10)
SAD (455); JMS DRAWAP /301(10)

//THIS OPTION DRAWS X & Y AXES AND PRINTS OUT 'DO AGAIN
//PLEASE', THEN 'GO' FOR PLOTTING OF DATA PROCESSED
//JUST BEFORE THIS OPTION WAS CALLED UPON,
SAD (620); JMS DRAWAS
SAD (621); JMS DRAWAP

//OPTION 400, ONLY X AND Y AXES ARE DRAWN, NO MARKINGS,
//SAME WITH OPTION 401, FOR PLOTTER,
SAD (20); JMP RECEU; JMP* ROUTIN

RECEU
DAC PLACF#; JMP RECON

LEARN1
DAC PLACE; JMP LEARN2

WPINI
0; LAC (1); DAC* WPOINT; JMP* WPINI

.EJECT
WPOINT 0
/DAC# WPOINT WITH 0 OR 1 FOR SCOPE
/OR XY- PLOTTER, RESPECTIVELY, WPOINT SHOULD
/BE LOADED INDIRECTLY IN MAIN PROGRAM, AND DIRECTLY
/IN A SUBROUTINE OTHER THAN SUBR, POINT,
//
PRINTH 0
/ WRITE 6, 2, DATE, 52
/ WRITE 5, FRMT63, 20, 10, BUF1
/ JMP# PRINTH
DACOCA JMS ACOCA; JMP DATINO
//
CORREC WRITEH 4, FRMT67 / REMOVE'/* FOR OPTION CORREC,
/ JMS* ACCEPT; JMP , + 5; DATS; DWRD#; CWRD#
PLACE
LAC DWRD; DAC POINTS; JMS PRELIM
LAC (F); TAD DWRD; DAC BUFC#
LAC BUFC; DAC TRANO+3
LAC (483); DAC TRANO+2; DAC TRAN1+2
LAC (724); DAC TRAN1+2
LAC NB; JMS* .AD; LAC WORD; DAC NBB
JMS DKINN; JMS DKOUT; JMP INI
//
BACK1 DAC PLACE; JMP COR
PLOTIP JMS WPINI; JMP , + 2
PLOTIS D2M* WPOINT; LAC CTC; DAC PNBC#; JMS PLOTG
JMP DATINO
PLOTG 0; JMS PLROU1; JMS PLROU2; JMS MDONE; JMS DKIN
LAC# BUF1P; DAC Y0#
JMS DELAY01; JMS DELAY1; JMS PLENT0; JMP# PLOTG
DRAWAS 0; D2M* WPOINT; JMS AX; LAC (454); JMP# DRAWAS
DRAWAP 0; JMS WPINI; JMS AX; LAC (455); JMP# DRAWAP
//
MDONE 0; LAC PNBC; SAD (0); JMP MOD0; SAD (-1); JMP , + 3
LAC ONE; JMP , + 2; LAC TWO; DAC XINC; JMP* MDONE
AX 0
JMS* POINT; JMP , + 4; ZERO; P250; ZERO; JMS DELAYG
JMS* POINT; JMP , + 4; ZERO; P250; ONE; JMS DELAYG
LAC DUMP; SAD (454); JMP MARK22; SAD (455)
JMP MARK22
JMS* LINE; JMP , + 3; ZERO; M250; JMS DELAYG
AX1 JMS* POINT; JMP , + 4; ZERO; M250; ZERO; JMS DELAYG
LAC DUMP; SAD (454); JMS MARK00
SAD (455); JMS MARK00
JMS* POINT; JMP , + 4; ZERO; ZERO; ZERO; JMS DELAYG
JMS DELAYG
JMS* POINT; JMP , + 4; ZERO; ONE; JMS DELAYG
JMS* LINE; JMP , + 3; P777; ZERO; JMS DELAYG
JMS* POINT; JMP , + 4; P777; ZERO; ZERO; JMS DELAYG
JMP* AX
EJECT
MARKYY
0; LAC P250; DAC YE

MARK2
LAC YE; TAD M50; DAC YE; SAD (-454); JMP* MARKYY
JMS* LINE; JMP +3; ZERO; YE; JMS DELAYS
JMS* MARK3
JMP MARK2

//
//
P250 372; M250 -372; P12 14; M50 -62

MARK3
0
JMS* LINE; JMP +3; P12; YE; JMS DELAYS
JMS* LINE; JMP +3; ZERO; YE; JMS DELAYS
JMP* MARK3

//
//
DELA YS
0; LAC (-100); DAC NBBC
JMS* ALOG10; JMP +2; ONE; ISZ NBBC
JMP -4; JMP* DELAYS

//
//
MARK2
JMS* MARKYY; JMP AX1

//
//
MARK00
0; DTM XE
JMS* POINT; JMP +4; ZERO; M250; ONE; JMS DELAYG

MARK0
LAC XE; TAD P64; DAC XE; SAD (1000); JMP +3
.EJECT
JMS MARK1; JMP MARK0
TAD (-1); DAC XE
JMS* LINE; JMP +3; XE; M250; JMS DELAYG
JMS* POINT; JMP +4; XE; M250; ZERO; JMS DELAYG
JMP* MARK00

//
M377 -377; P777 777; P64 100; M356 -356

//
MARK1
✓
JMS* LINE; JMP +3; XE; M250; JMS DELAYS
JMS* LINE; JMP +3; XE; M356; JMS DELAYS
JMS* LINE; JMP +3; XE; M250; JMS DELAYS
JMP* MARK1

ACOCA
✓; LAC NB; TAD (300); DAC TOSTBL
LAC (30? ?); DAC FRSTBL; JMS TIOS

//THIS OPTION PRODUCES THE PART OF THE AUTOCORRELATION
//FUNCTION FOR NEGATIVE TAU BEFORE TAKING THE FFT.
//THIS OPTION IS VALID ONLY FOR #PTS=256,
//BUT CAN BE ARRANGED LATER TO ACCOMODATE OTHER #PTS#,
//THIS CAN BE USED FOR CONVOLUTION IF STATEMENT
//LAC (BUF2+400) BELOW IS MODIFIED TO LAC (BUF2+401) TO
//PRODUCE A COMPLETE MIRROR IMAGE, INCLUDING THE POINT
//AT ORIGIN, OF THE FUNCTION ON THE RIGHT OF THE Y-AXIS.
JMS PLROU2; LAC (BUF2); DAC TRAN0+3
ACOFFW
JMS PLROU1; DZM BUF2; ISZ FCT; ISZ BUF1P
JMS DKIN; LAC (BUF2+400); DAC BUF2P
ACOFF1
LAC BUF2P; TAD (-1); DAC BUF2P
LAC* BUF1P; DAC* BUF2P
ISZ BUF1P; ISZ FCT; JMP ACOFF1
LAC TOSTBL; TAD (-1); DAC TOSTBL; DAC TRAN0+2
JMS DKOUT
LAC (BUF1); DAC TRAN1+3; DAC TRAN0+3
LAC (303); DAC TRAN1+2; LAC (100); DAC TRAN0+2
JMS DKIN; JMS DKOUT; JMS DKIN; JMS DKOUT
LAC MWORD; JMS INV; JMS* .AD; LAC NB
CLL; RAL; DAC POINTS; JMS PRELIM
LAC (3); DAC PLACE
JMP* ACOCA

MOD0
LAC WORD; TAO (-200); SNAISZA; JMP ,+3; LAC FOUR
JMP ,+2
LAC TWC; DAC XINC; JMP* MODONE

PLENTW
✓; DZM V
PLENT1
LAC V; DAC XE#; LAC* BUF1P; DAC YE#
JMS* LINE; JMP ,+3; DSA XE; DSA YE
ISZ BUF1P; ISZ NB2C; JMP DONTX
DOXX
LAC V; TAD XINC; DAC V; LAC NB2#; DAC NB2C
DONTX
ISZ FCT; JMP PLENT1; ISZ CT

//
THIS ACTION IS FOR NB=1 & 2 TO HAVE FULL SCALE,
JMP ,+2; JMP ,+4; JMS PLROU1; JMS DKIN; JMP PLENT1
LAC (1); DAC XINC
JMS DELAY2; JMS DELAYE; JMS DELAY0; JMP* PLENT0
.EJECT
PLROU0 0; LAC MWORD; DAC FCT
  LAC (BUF1); DAC BUF1P; LAC (BUF2); DAC BUF2P
  JMP* PLROU0

//
PLROU1 0; LAC MWORD; DAC FCT
  LAC (BUF1); DAC BUF1P
  LAC (BUF2); DAC BUF2P
  LAC (F); DAC FP; JMP* PLROU1

//
PLROU2 0; LAC PNBC; SAD (0); LAC (-1); DAC CT; SAD (-1)
  JMP ROU2EN
  JMS* ,AE; LAC (2)
ROU2EN DAC NB2; DAC NB2C#
  LAC PLACE; SAD (3); JMP PLOT03; SAD (1); JMP PLOT01
  SAD (15); JMP PLOT0B
  SAD (30); JMP PLOTLN; SAD (35); JMP PLOTKT
PLOT02 LAC (100); JMP PLOTN
PLOT01 LAC (0); JMP PLOTN
PLOT0B LAC AVECT; JMP PLOTN
PLOTLN LAC LNPL; JMP PLOTN
PLOTKT LAC STABLO; JMP PLOTN
PLOT03 LAC (300)
PLOTN DAC TRAN1+2; DAC TTRAN1+2; LAC (BUF1); DAC TTRAN1+3
  DAC TTRAN1+3
  JMP* PLROU2

//
DELAYG 0; LAC (-200); DAC NBBC#
  D1  JMS* ALOG10; JMP .+2; ONE; ISZ NBBC
  JMP D1; JMP* DELAYG

//
DELAY0 0; JMS* POINT; JMP ,+4; ZERO; Y0; ZERO
  JMS DELAYG; JMS DELAYG; JMP* DELAY0
DELAY1 0; JMS* POINT; JMP ,+4; ZERO; Y0; ONE
  JMS DELAYG; JMP* DELAY1
DELAY2 0; JMS* POINT; JMP ,+4; XE; YE; ONE
  JMS DELAYG; JMP* DELAY2

//
DELAYE 0; JMS* POINT; JMP ,+4; XE; YE; ZERO
  JMS DELAYG; JMP* DELAYE

//
BREAK1 0 /FOR PAPER TAPE SEGMENTATION PURPOSE,
BUFOUT ,BLOCK 17
TWENTY ,DSA 000005
 ,DSA 240000
ONE ,DSA 000001; ,DSA 200000
/NUMBER ,DSA 364755; ,DSA 206157 /,1E-5
/NUMBER ,DSA 463775; ,DSA 314631 /0.1.

/////
//
NREAL ,BLOCK 2
,EJECT
DEGRAD, BLOCK 2
BLOKOT, BLOCK 1
BLOKIN, BLOCK 1
ZERO, 0; 0
ON 377
R250, DSA 00010; DSA 372000
RFIVE, 00003; 240000
R8188, 061004; 203004
M8188, BLOCK 2
XV, BLOCK 2
LL, BLOCK 2
NBB, BLOCK 2

//FRMT78, ASCII "{(X,4HTAU;)
//BEGIN2 WRITE 4,FRMT78
//JMS* ACCEPT; JMP *,+3; DATS; MWORC
//LAC MWORC; JMS INV; DAC MWORC
//TAD (407); SMA; JMP BEGIN3; LAC (-400); DAC MWORC
//REMOVE THE SLASHES (/) IN FRONT OF STATEMENTS STARTING
//WITH FRMT78, IF VARIABLE DELAY TIME TAU IS WANTED.
//ALSO A SLASH SHOULD BE PUT IN FRONT OF BEGIN2
//IN THE STATEMENT BELOW, SEE ALSO REMARK BELOW,
BEGIN2 LAC WORD; JMS INV; DAC MWORC
BEGIN3 D2M IT:
LAC (200);
DAC TRANSF+2; LAC (F); DAC TRANSF+3; JMS REMAIN
JMP T102
REMAIN 0; LAC MWOR; SAD (-400); JMP *,+2; JMP REMANC
LAC NBB; SAD (1); LAC (2); JMS*, AF; LAC (-1000)
REMANC DAC DCF#;
LAC NBB; JMS* , AD; LAC WORD; DAC NBB; DAC NBBB
DAC NBBBP; DAC NBBAP /FOR RECUP, FR, REMAINERS,
JMP* REMAIN
T102 D2M L# /L=0 TO 255
LAC (F)
DAC FP#
LAC MWORC
DAC FCT#

//THIS PERMITS TO VARY DELAY TIME TAU AT WILL
//IF ABOVE STATEMENTS ASSOCIATED WITH
//FRMT78 ARE TAKEN INTO PROGRAMMING,
T100 D2M* FP
IS2 FP
IS2 FCT
JMP T102
COMP0 LAC (F)
DAC FP;
,EJECT
LAC MWORD /SAME REMARK,
DAC FCT

COMP1
LAC NB
TAD (-1
DAC MP# /MP=NB-1
LAC IT
CMA
TAD NB /MQ=NB-IT-1
DAC MQ
LAC IT
DAC MPP# /MPP=IT
JMS FINDQ

T4
JMS FINDP
T23
JMS SUBB
TAD* FP /PR IS RETURNED IN AC BY SUBB.
DAC* FP

//IF (L,EQ,WORD)GOTO T211
ISZ FCT
JMP OUT2

T211
LAC MQ#
SAD (0
JMP ,+2
JMP CC3
LAC MP
SAD MPP
JMP T41

CC3
DZM L
LAC (F
DAC FP
LAC MWORD
DAC FCT
LAC MQ
TAD (-1
DAC MQ
JMS FINDQ

T222
JMS SUBA
TAD* FP
DAC* FP /PR IS RETURNED IN AC BY SUBB.

//IF(L,EQ,WORD)GOTO T212
ISZ FCT
JMP OUT3

T212
DZM L
LAC (F
DAC FP
LAC MWORD
DAC FCT
LAC MP
TAD (-1
DAC MP
*EJECT
OUT2
JMP T4
ISZ L
ISZ FP
JMP T23
OUT3
ISZ FP
ISZ L
JMP T222
T41
JMS DKOF
JMP T703
DKOF
°
TRANOF
,INIT 3,1,PARIN1
,TRAN 3,1,200,F,256
,WAIT 3
,CLOSE 3
ISZ TRANOF+2
JMP* DKOF
,/*IF(IT,EQ,(NB-1))GOTO T199
T703
LAC NB
TAD (-1
SAO IT
JMP T299
CC5
ISZ IT
JMP T102
TA1
,BLOCK 2
KK
,BLOCK 2
FINDP
°
LAC MP
DAC TRANP+2 /BLK START AT 0,MP STARTS AT 0
,INIT 3,0,PARIN1
TRANP
,TRAN 3,0,0,BUF1,256
,WAIT 3
,CLOSE 3
JMP* FINDP
FINDQ
°
LAC MQ
TAD (100 /TO TAKE CARE OF LOCATION OF 1RST ADDR
,/*OF FILE 0,
DAC TRANQ+2
,INIT 3,0,PARIN1
TRANQ
,TRAN 3,0,0,BUF2,256
,WAIT 3
,CLOSE 3
BRK
JMP* FINDQ
SUBB
°
D2M PR#
D2M K#; LAC DCF; DAC DCT#
LAC (BUF2-1 /-1 IS TO TAKE CARE OF IS# AT 117,
DAC BUF2P /K=L TO (WORD-1),KK=0 TO (WORD-1-L)
LAC L /L = 0 TO (WORD-1),L IS TAU IN THIS PROG,
,EJECT
TAD (BUF1-1
DAC BUF1P
LAC L
TAD MWORD /TO TAKE CARE OF ISZ ENO
DAC END# /BEFORE CHECKING FOR EXIT.
T17
ISZ BUF1P
ISZ BUF2P
LAC* BUF1P
GSM
DAC MULTB
LAC* BUF2P
MULS

MULTB
Ø
CLL
LACQ /IN 1'S COMPL,NEEDED FOR IDIVS.
IDIVS

NBBB
Ø
SPA; TAD (1); TAD K; DAC K /REMAINDER,
ISZ DCT; JMP LACQ; LACQ; DAC TA1
CLL; LAC K; SPA; TAD (-1); IDIVS

NBBBP
Ø
SPA; TAD (1); TAD K; DAC K
LACQ; SPA; TAD (1); TAD PR; DAC PR; LAC DCF
DAC DCT; LAC TA1; JMP +2

LACQB
LACQ
SPA
TAD (1
TAD PR
DAC PR
ISZ END
JMP T17
JMP* SUBB

SUBA
Ø
DZM PR
DZM K; LAC DCF; DAC DCT
LAC L /L=Ø TO (WORD-1)
SAD (Ø
JMP* SUBA
LAC (BUF1-1 /K=Ø TO L-1, KK=WORD-L+K
DAC BUF1P
LAC L
CMA
TAD (1
DAC END
TAD WORD
TAD (BUF2-1
DAC BUF2P
ISZ BUF1P
ISZ BUF2P
,EJECT
LAC* BUF1P
GSM
DAC MULTA
LAC* BUF2P
MULS

MULTA
0
CLL
LACQ
IDIVS

NBBA
0
SPA; TAD (1); TAD K; DAC K /REMAINDER
ISZ DCT; JMP LACQA; LACQ; DAC TAI
CLL; LAC K; SPA; TAD (-1); IDIVS

NBBAP
0
SPA; TAD (1); TAD K; DAC K
LACQ; SPA; TAD (1); TAD PR; DAC PR; LAC DCF
DAC DCT; LAC TAI; JMP +2

LACQA
LACQ
SPA
TAD (1)
TAD PR
DAC PR
ISZ ENO
JMP T18
JMP* SUBA

//

T299
LAC (200); DAC TRAN1+2; LAC NBC; DAC CT; JMS PLROU1
LAC (300); DAC TRANO+2
LAC MWORC#; DAC MWORD; LAC (BUF1); DAC TRANI+3
DAC TRANO+3
LAC PLACE; SAD (4); JMP ,+2; JMP CROSCO
DAC FFTPL#; JMS DKIN
LAC* BUF1P; DAC KP#; JMP SCALG

CROSCO
DAC FFTPL; DZM KP
DOIT3
JMS DKIN
T19
JMS PLROU1
T20
LAC* BUF1P; SPA /POSITIVE?
JMP T22

T21
JMS INV
T22
DAC DUM /MAKES DUM ALW, NEG,
TAD KP; SZA; NOP; SPA; JMP T30
T25
ISZ BUF1P; ISZ FCT; JMP T20; ISZ CT; JMP DOIT3

JMP SCALE
T30
LAC DUM; TAD (-1); CMA; DAC KP; JMP T25
SCALE
JMS PLROU1; LAC NBC; DAC CT
LAC (200); DAC TRAN1+2; LAC (300); DAC TRANO+2
JMS DKIN
SCALG
JMS* FLOAT; JMP ,+2; ,DSA KP
JMS*, AL; ,DSA R250; JMS*, AH; ,DSA KK
,EJECT
T40
JMS* FLOAT; JMP ,+2; ,DSA BUF1P+400000
JMS* ,AL; ,DSA KK
JMS* ,AX
DAC* BUF1P; ISZ BUF1P
ISZ FCT; JMP T40; JMS DKOUT; JMS PLROU1; ISZ CT
JMP ,+2; JMP ,+3; JMS DKIN; JMP T40
LAC (3); DAC PLACE
JMP OUTPUT

//
DKINN
0
,INIT 3,0,PARIN1
TRANNI
,TRAN 3,0,0,F,512
,WAIT 3
,CLOSE 3
ISZ TRANNI+2; ISZ TRANNI+2
JMP* DKINN

//
WRITKP
0; WRITED 4,FRMT76,KP
JMP* WRITKP
OUTPUT
LAC RECONSW; SAD (1); JMP RECON; JMS WRITKP
DATOUT
WRITEH 4,FRMT1
JMS ACCPA
LAC MWORC; DAC MWORD
LAC DUMP; JMS ROUTIN
SAD (5)
JMP PRINT3
SAD (3); JMP BEGIN5
SAD (12); JMP SAV2
SAD (0); JMP PLOT3S
SAD (310); JMP PLOT3P
JMS DOAGAIN; JMP DATOUT
SAV2
JMS SAVE; JMP DATOUT
PLOT3P
JMS WPINI; JMP ,+2
PLOT3S
D2MS* WPOINT; LAC CTC; DAC PNBC; JMS PLOTG
JMP DATOUT
BEGIN5
DAC PLACE; JMP BEGIN2
PRINT3
JMS PRINTG; JMP DATOUT
PRINTG
0; LAC NBC; DAC PNBC; JMS PLROU1; JMS PLOKU2
DOIT5
JMS DKIN
WRITED 6,FRMT4,KP
WRITED 6,FRMT7,N8
JMS PRINTH
ISZ CT
JMP DOIT5; JMP* PRINTG
GTIO
WRITEH 4,FMT22
JMS* ACCEPT; JMP ,+5; DATS; FRSTBL#; TOSTBL#; POINTS
JMS PRELIM; JMS TIOS; JMP DATINO
TIOS
0; LAC FRSTBL; DAC TRANNI+2; LAC TOSTBL; DAC TRAN0+2
LAC NBC; DAC CT; LAC (BUF1); DAC TRANNI+3 ,EJECT
DAC TRAN0+3
DOIT14 JMS DNKIN; JMS DNKOUT; ISZ CT; JMP DOIT14
LAC TOSTBYL; SAD (0); JMP PLO1
SAD (100); JMP PLO2; DAC AVECT; LAC (15); JMP GTIP
PL01 LAC (1); JMP GTIP
PL02 LAC (2)
GTIP DAC PLACE; JMP* TIOS
//
//
//
ADVAC 0; WRITEH 4,FMT58
JMS* ACCEPT; JMP .*5; DATS; TNB; FBK; GRPN
LAC TRANAV+2; DAC ADVACT#
LAC (NOP); DAC LNSW
ADV1 LAC GRPN#; DAC SPNB; JMS AVEROU
LAC TNB#; JMS* ,AE; LAC GRPN; DAC TBNBYN#
JMS INV; DAC TNBC#
JMP* ADVAC
ADVAVE JMS ADVAC
WRITEH 4,FMT56
JMS ACSPNB; LAC SPNB; DAC WEIGHT
AV2 JMS PRLOU1; LAC SPNBC; DAC CT
AV22 LAC FBK#; DAC* FP; ISZ FP; ISZ FBK
ISZ CT; JMP AV22
JMS ACC2S; JMS AV3; ISZ TNBC; JMP AV2; JMP ADVOUT
//
ADVAVB JMS ADVAC
WRITEH 4,FMT56
JMS ACSPNB; LAC SPNB; DAC WEIGHT
AVB2 JMS PRLOU1; LAC SPNBC; DAC CT; LAC FBK; DAC FBKC#
AVB3 LAC FBKC; DAC* FP; ISZ FP; TAD TBNBYN; DAC FBKC
ISZ CT; JMP AVB3
JMS ACC2S; JMS AV3; ISZ TNBC; JMP AVB4; JMP ADVOUT
AVB4 ISZ FBK; JMP AVB2
//
ADVOUT LAC ADVACT; DAC AVECT; LAC (15); DAC PLACE
LAC TBNBYN; JMS* .AD; LAC (400); DAC POINTS
JMS PRELIM; JMP DATINO
AV3 0; LAC TRANAV+2; DAC AVECT; DAC V; JMS DKAVE
JMS WRITU; JMP* AV3
WRITNB 0; WRITEH 4,FRT12
JMP* WRITNB
AVERAG LAC (NOP); DAC LNSW
JMS WRITNB; JMS ACSPNB; JMS AVEROU; JMP ACC0
AVEROU 0; LAC (-400); DAC MWORD; LAC (1)
DAC NB; LAC (-1); DAC NBC
LAC SPNB#; DAC SPNC#; JMS INV
DAC SPNBC#; DAC CT
, EJECT
JMS PLROU1 /VALID ONLY FOR NB=1 & WORD=256, JMS* AVEROU
ACC0 WRITEH 4,FRTM22
ACC1 WRITEH 4,FRTM23
JMS ACSNPB
LAC SPNB; DAC* FP; ISZ FP; ISZ CT; JMP ACC1
WR156 WRITEH 4,FRTM56
JMS ACSNPB; LAC SPNB
DAC WEIGH#
JMS ACC2S; JMP ACC6
ACC2S 0
ACC2 JMS PLROU1; LAC SPNBC; DAC CT
LNSW JMP LEARN3 /OR NOP, DEPENDING ON OPT, LEARN2
// OR AVERAG WAS CHOSEN, THIS MAKES LEARNING OF THE
// MACHINE AUTOMATICALLY FAST, SKIPPING STEPS
// IN BETWEEN.
ACC7 JMS ACCB2; ISZ CT
ACC3 JMS ACCB1; JMS PLROU0
ACC4 LAC* BUF1P; TAD* BUF2P
DAC* BUF2P; ISZ BUF1P; ISZ BUF2P
ISZ FCT; JMP ACC4; JMS PLROU0; ISZ CT; JMP ACC3
JMS PLROU1
LAC SPNC; TAD WEIGH; TAD (-1); DAC DIV#
ACC5 LAC* BUF2P; JMS* ,AE; LAC DIV
DAC* BUF2P; SMA; JMP POSA; JMP Nega
NEGA LAC*,C0; JMS INV; CLL; RAL; TAD DIV; SNA; JMP ,+4
SPA; JMP +2
JMP NOROU; LAC (-1); JMP ADDON
POSA LAC*,C0; JMS INV; CLL; RAL
TAD SPNC; SNA; JMP ,+4; SPA; JMP ,+2; JMP NOROU
LAC (1); JMP ADDON
ADDON TAD* BUF2P; DAC* BUF2P
NOROU ISZ BUF2P; ISZ FCT; JMP ACC5
JMP* ACC2S
ACC6 LAC (15); DAC PLACE
LAC TRANAV+2; DAC AVECT#; DAC V; JMS DKAVE
JMS WRITU; LAC (400); DAC POINTS; JMS PRELIM
JMP DATIN0
// LEARN2 LAC (2); DAC SPNB; JMS AVEROU
WRITEH 4,FRTM57
JMS ACSNPB; LAC SPNB
TAD RECSRA; TAD (-1)
DAC SPNB; DAC LNPL#
DAC* FP; ISZ FP; LAC (300); DAC* FP
LAC (JMP LEARN3); DAC LNNSW; JMS ACC2S
LAC LNPL; DAC TRANAV+2; LAC (BUF2); DAC TRANAV+3
JMS DKOUT; JMP RECON
//
EJECT
LEARN3 LAC LNCOEF#; DAC WEIGH: JMP ACC7
FMT58 ASCII "(1X,34HTOTAL #NB: 1ST BLK: NB TO BE GR'D:)

//
WRITU 0; WRITED 4,FMT7,V
JMP* WRITU

//
INV 0; CMA; TAD (1); JMP* INV

//
DKAVE 0
.INIT 3,1,PARIN1
TRANAV .TRAN 3,1,0,BUF2,256
.WAIT 3
.CLOSE 3
ISZ TRANAV+2; JMP* DKAVE

//
ACSPNB 0; JMS* ACCEPT; JMP .+3; DATS; SPNB; JMP* ACSPNB
ACCB1 0; LAC* FP; DAC TRANI+2; LAC (BUF1); DAC TRANI+3
.DAC BUF1P
.ISZ FP; JMS DKIN; JMP* ACCB1
ACCB2 0; LAC* FP; DAC TRANI+2; LAC (BUF2); DAC TRANI+3
.DAC BUF2P; LAC (-400); DAC FCT
.ISZ FP; JMS DKIN
ACCC2 LAC* BUF2P; JMS* ,AD; LAC WEIGH; DAC* BUF2P
.ISZ BUF2P; ISZ FCT; JMP ACCC2; JMP* ACCB2

//
//
NBBR  .BLOCK 2
HALF  .DSA 000000; .DSA 200000
PICOEF  .BLOCK 2
COEFF  000010

//
DKOUTT 0
.INIT 3,1,PARIN1
TRANTT .TRAN 3,1,0,F,512
.WAIT 3
.CLOSE 3
ISZ TRANTT+2
ISZ TRANTT+2
JMP* DKOUTT

//
SAVECT 0
.END
.TITLE ACOR2-FFT/8/1/71.

//
//
///OPTIONS AVAILABLE EVERY TIME "GO:" IS TYPED OUT BY
//THE COMPUTER:
//
//0,5,6,7,9,10,11,12,19,29,33,60,70,100,101,102,103,200,
//201,202,203,300,301,400,401,500,501,1000,3000,
//
///FOR COMPLETE DETAILS ON EACH OPTION, SEE "COMPUTER
//DESIGN AND EVALUATION OF OPTIMAL FILTERS FOR A CLASS
//OF SPEECH SOUNDS", PH.D. DISSERTATION BY LE-BA TUAN,
//
//
//
//

.EJECT

ADACTV=701601
ADCLON=701624
ADCLOF=701644
CADAPI=701664

.GLOBL .AA,.AB,.AC,.AD,.AE,.AF,.AJ,.AY,.BA,ACCEPT
.GLOBL FLOAT,.AH,.AG,.AL,.AN,.ALOG10,.AM,.AX,.CO
.IODEV 6,4,3
.GLOBL ACOR2,ACOR1,PPPR
.GLOBL ,FP,.FW,.FE,.FR,.FF,POINT,LINE
.DEFIN WRITED,IODEV,FMT,BUF,?Q
JMS*,FW,
,DSA Q
,DSA FMT
JMS*,FE
,DSA BUF
JMS*,FF
JMP ,+2
.IODEV
,ENDM
,DEFIN WRITMN,IODEV,FMT,COL,ROW,BUF,?C1,?C2,?P,?Q
.LAC (-ROW
.DAC C2
.JMS*,FW,
,DSA Q
,DSA FMT
.LAC (-COL
.DAC C1
.JMS*,FE
,DSA BUF
.ISZ P
.ISZ P /SKIP 1 VALUE IN INTEGER PRINTING.
.ISZ C1 /PUT ISZ P IN IF SKIP & REAL PRINTG IS
JMP P-1 /NEEDED
.JMS*,FF
.ISZ C2
.JMP P-6
.LAC (BUF
.DAC P
.JMP ,+4
.Q
.IODEV
.C1 0
.C2 0
,ENDM
,DEFIN WRITEH,IODEV,FMT,?Q
JMS*,FW,
,DSA Q
,DSA FMT
.JMS*,FF
,EJECT
JMP +2
Q IODEV
.ENDM
.DEFIN READ, IODEV, FMT, BUF, ?Q
JMS*, FR
.DSA Q
.DSA FNT
JMS*, FE
.DSA BUF
JMS*, FF
JMP +2
Q IODEV
.ENDM
DATE .BLOCK 42

//
ACOR2 0 /FFT PROGRAM,
JMS*, FP
WRITEH 4, FRMT4
START WRITEH 4, FRMT8
.INIT 4, 0, PARIN1
.READ 4, 2, DATE, 34
.WAIT 4
.CLOSE 4
WRITEH 4, FRMT9
JMS* ACCEPT; JMP +5; DATS; SAVSTA; CPWR; CEPSW
ZM CONVSW /THIS COMES FR, ACOR1
LAC SAVSTA#; DAC TRASA+2
LAC AVESTA#; DAC TRANAV+2
PARIN1 WRITEH 4, FRMT1
JMS* ACCEPT; JMP +3; DATS; PLACE
BEGIN LAC PLACE#
DAC DUMP /MAKES PLOTTING AXES INDEPENDANT,
JMS ROUTIN
JMS DOAGAN; JMP PARIN1
DOAGAN @; WRITEH 4, FRMT75
JMP* DOAGAN
RECPUP WRITEH 4, FRMT6
JMS ACCPT; LAC (4); DAC PLACE; JMP RENT1

//

L .BLOCK 2
VI:MWORD; FCT; FP
DKOF @
.INIT 3, 1, PARIN1
TRANOF .TRAN 3, 1, 200, F, 256
.WAIT 3
.CLOSE 3
ISZ TRANOF+2; JMP* DKOF
WRITK0 @; WRITED 4, FRMT77, KP
JMP* WRITK0
.EJECT
FRMT 1  .ASCII "(1X,3HG0:)
FRMT 11 .ASCII "(1H+,25HCONVERTED DATA FOR INPUT,)"
FRMT 2  .ASCII "(28X,21HMULTIPLIER=+0.122130,)
FRMT 4  .ASCII "(1X,13HFFT & OPTIONS)"
FRMT 5  .ASCII "(1X,6HPLACE:)"
FRMT 3  .ASCII "((1)"
FRMT 6  .ASCII "(1X,5H#PTS")"
FRMT 7  .ASCII "((1H+,16X,15H,NUMBER OF BLK=12)"
FRMT 9  .ASCII "((1X,17HSAYS: CP:1 CEPS:1)"
FRMT 12  .ASCII "((1X,5H#BLK:)"
FRMT 10  .ASCII "((1X,5HADFFT)"
FRMT 30  .ASCII "((1X,12HPLACE: #PTS:)"
FRMT 83  .ASCII "((1X,16(1X,16))"
FRMT 75  .ASCII "((1X,16HDU AGAIN,PLEASE,)
FRMT 78  .ASCII "((1X,15)"
FRMT 87  .ASCII "((1X,1HS,14)"
FRMT 92  .ASCII "((1X,12)"
FRMT 26  .ASCII "((1X,3HFFT)
FRMT 25  .ASCII "((1X,4HFTIO)"
FRMT 22  .ASCII "((1X,13HFR: TO: #PTS:)
FRMT 23  .ASCII "((1X,12H#PTS=2#M;M;)"
FRMT 24  .ASCII "((1X,12HEXP7GN NEG:1)"
FRMT 30  .ASCII "((8X,15,3(4X,E11,4)4(4X,G10,3))"
FRMT 36  .ASCII "((50X,16HTRANSFORM OUTPUT)"
FRMT 36  .ASCII "((45X,20HNO OF DATA POINTS N=15)"
FRMT 37  .ASCII "((45X,31HTRANSFORM EXPONENT WAS NEGATIVE)"
FRMT 40  .ASCII "((45X,31HTRANSFORM EXPONENT WAS POSITIVE)"
FRMT 45  .ASCII "((1X,8HCOMMENT:)
FRMT 43  .ASCII "((OX,8HHARMONIC4X,4HREAL12X,4HIMAG12X,3HAMP)"
FRMT 44  .ASCII "((1H+,63X,4HAMPSN9X,5HAMPDB8X,6HAMPDBN8X)"
FRMT 44  .ASCII "((1H+,104X,8HCUF,PR.)"
FRMT 46  .ASCII "((1X,12H#PTS SKIP'D)"
FRMT 50  .ASCII "((110X,7THE END)"
FRMT 51  .ASCII "((1X,9H#PTS OUT;)"
FRMT 52  .ASCII "((1X,9H11,12,GO;)"
FRMT 55  .ASCII "((1X,6HH1H02:)"

EJECT
DATS 4

//
//
COR LAC PTSX#; SPA; JMP STC

//OPTION PT: GIVES FREEDOM FOR CHOICE OF #PTS
//IF PT: 0, OR AUTOMATICALLY FIX #PT EQUAL TO
//ANY NUMBER OTHER THAN 0,
WRITEH 4,FRMT6

STA NOP; JMS ACCEPT; JMP STB

ACCEPT 0; JMS* ACCEPT

JMP *3

,DSA DATS

,DSA POINTS#; JMS PRELIM; JMS PREAC; JMP* ACCEPT

PRELIM 0; LAC POINTS; DAC SPOINT#; JMS* ,AE; LAC (400)

JMS INV; DAC PNBC; DAC CTC; SAD (0); LAC (-1)

DAC NBC#; JMS INV; DAC NB#; LAC* ,CO

SAD (3); JMP *2; JMP *2; LAC (400); DAC WORD

JMS INV; DAC HWORD

LAC NB; JMS* ,AD; LAC WORD; DAC NBB; JMP* PRELIM

PREAC 0; LAC NB; DAC NF#; LAC WORD; DAC WORF#

JMP* PREAC

//

STC DAC POINTS; JMS PRELIM; JMS PREAC

STB LAC ADINBL /STARTING BLOCK #BER HERE

DAC TRACG+2

JMP LDHF11

,EJECT
ADINBL 000724 /468(10),
// THIS IS WHERE TO BEGIN TO SAVE THE DATA READ IN
// FROM THE A-D CONVERTER FOR ANY LATER PROCESSING
// BEFORE A NEW READ-IN FROM THE A-D CONVERTER, THE
// OPTIONS "CORREC", "RECUP", ETC... DIG
// UP THE INPUT DATA FROM THIS BLOCK ON.
BUF1 .BLOCK 400
BUF2 .BLOCK 420
F .BLOCK 1000
RENT6 LAC WRF; DAC WORD; JMS INV; DAC MWORD
      LAC NF; DAC NB; JMS INV; DAC NBC; JMP RENT1
LDBF0 LAC MWORD
       DAC* (32)
       LAC (BUF1-1)
       DAC* (33)
       LAC NBC
       DAC BLKCT#
INP  ADACTV
    / ADCLON
    IORS
    AND (4
    SZA
    JMP .-3
    ISZ BLKCT /DOUBLE BUFFERING
    JMP LDBF2
OUT1 LAC (BUF1
       JMP DKOUTF
LDBF1B 0; LAC (-1); DAC* (32); DAC FCT
       LAC (BUF1-1); DAC* (33); ADACTV; JMP LDBF1B
LDBF1I LAC NBC; DAC BLKCT#; JMS LDBF1B; ADCLON
LDBF1C IORS; AND (4); SZA; JMP .-3
   ,EJECT
TEST
LAC (BUF1); DAC BUF1P
TESI
LAC BUF1P; SPA; JMS INV
TAD MTWRD; SPA; JMP TESS; JMP LDBF0
TESS
ISZ FCT; JMP TESI; JMS LDBF1B; JMP LDBF1C

//
/MTWRD  -146/102 (10), AROUND 26 DB BELOW MAX.
MTWRD    -24/2.8 (10)
//THIS MAKES COMPUTER WAIT TILL INPUT EXCEEDS THIS
//THRESHOLD TO BEGIN TO READ IN VIA A/D-CONVERTER.
LDBF2 LAC MWORD
DAC# (32
LAC (BUF2-1
DAC# (33
ADACTV
LAC (BUF1
JMS OKUA
IORS
AND (4
SZA; JMP , -3
ISZ BLKCT
JMP LDBF1
LAC (BUF2
JMP OKUUF
LDBF1 LAC MWORD
DAC# (32
LAC (BUF1-1
DAC# (33
ADACTV
LAC (BUF2
JMS OKUA
IORS
AND (4
SZA
JMP , -3
ISZ BLKCT
JMP LDBF2
LAC (BUF1
JMP OKUUF
OKUUF OKOUA
; DAC TRAND+3; JMS OKOUT; JMP* OKOUA
OKOUA JMS OKOUA
ADCLOF
JMP RENT1
//
DUM .BLOCK 2
MULTF .BLOCK 2
R2047 000013; 377700
R12213 454775; 372076 /250/2047.
INCOEF 216007; 227003 /.755032E+02=250/ALOG10(2048).
//
RENT1 LAC CONVSW#; SAD (1); JMP DB1; SAD (2); JMP DER0
; EJECT
SAD (0); JMP DB2
JMP DB2

DB1
LAC (JMP INDB); DAC DBSW; LAC (JMP SIGNF)
DAC SIGNSW
LAC (NOP); DAC DERSW; JMP RENS1

DB2
LAC (NOP); DAC DBSW; DAC SIGNSW; DAC DERSW
JMP RENS1

DER0
LAC (JMP DERI); DAC DERSW; LAC (NOP); DAC DBSW
DAC SIGNSW
JMP RENS1

RENS1
LAC AD1NBL; DAC TRAN1+2
INI
LAC (102); DAC TRAN1+2
LAC (0); DAC TRAN1+2
LAC NUC
DAC ADDBLCT#
LAC MWORD
DAC BUF1CT#
LAC (BUF1)
DAC HUF1P#
DAC TRAN1+3
LAC (BUF2)
DAC BUF2P#; DAC TRAN1+3; DAC TRAN2+3

CONVRT
JMS 0K1M
LAC BUF1P
ISZ BUF1P

DERSW
JMP DERI /OR NOP

SIGNSW
JMP SIGNF /OR NOP

RENS3
JMS 0K1M

DBSW
JMP INDB /OR NOP

JMS 0K1M

.ININ
JMS 0K1M

.IN10
DAC BUF2P
ISZ BUF2P
ISZ BUF1CT
JMP CONVRT+1

LAC MWORD
DAC BUF1CT

.DK0
LAC PLACE
SAD (1
JMP DK01
SAD (2
JMP DK02
SAD (74); JMP DK02
SAD (186); JMP DK02 /74 IS 60(1); 186 IS 70(10),
JMP DK012

.DK01
JMS DK0UT1
JMP EX

.DK02
JMS DK0UT2
.EJECT
JMP EX

DK012
JMS DKOUT1
JMS DKOUT2

EX
LAC (BUF2)
DAC BUF2P
ISZ ADDBLCT
JMP GOAD
JMP FINALE

GOAD
LAC (BUF1)
DAC BUF1P
JMP CONVRT

DERI
JMS INV; TAU* BUF1P; JMP RENS3
/
SMA; JMP .+2; JMP DERN
/
SZA; JMP .+2; JMP DERZ
/DERP
LAC (4:*); JMP IN10
/DERN
LAC (-4:*); JMP IN10
/DERZ
LAC (3); JMP IN10

SIGNF
SMA; JMP POSKP; JMS INV; DAC TA1
LAC (JMS* ,BA); DAC SW10
LAC TA1; JMP RENS3

POSKP
SMA; LAC (1); DAC TA1; LAC (NOP); DAC SW10
LAC TA1; JMP RENS3

INDB
JMS* ,AH; TA1; JMS* ALOG10; JMP .+2; TA1
JMS* ,AK; INCGEF

SW10
JMS* ,BA /UR NOP
JMP IN1N

//

DKOUT1
.INIT 3,1,PARIN1
TRAN01
.TRAN 3,1,0,BUF2,256
.WAIT 3
.CLOSE 3
.ISZ TRAN01+2
JMP* DKOUT1

DKOUT2
.INIT 3,1,PARIN1
TRAN02
.TRAN 3,1,100,BUF2,256
.WAIT 3
.CLOSE 3
.ISZ TRAN02+2
JMP* DKOUT2

FINALE
LAC PLACE
SAD (74); JMP ADFFT
SAD (106); JMP ADFFT
JMP PARIN1

WRNB
.WRITED 4,FRMT65,N8
JMP* WRNB

W2
JMS WRNB; JMP PT
.EJECT
SAVE  JMS SAVE; JMP DATINO
SAVE  JMP LAC_NBC; DAC_PDBC; JMS PLROU2
SAVF  JMP LAC TRANS+A2; DAC V
SAVF  JMP LAC_OKIN; JMS DSAV; ISZ CT; JMP SAVF; JMS WRITV
JMP* SAVE
//
WRITV  JMP* WRITED IV,FMT0,V
JMP* WRITV
ADFFT  JMP LAC WR16
JMP PT
ADFFA  JMP LAC (1,IV); JMP FFTCON+1 /DO NOT CHANGE,
M2 =-2
//
//
DKIN  JMP
.INIT 3,0,PARIN1
TRAN  .TRAN 3,0,0,BUF1,256
.WAIT 3
.CLOSE 3
ISZ TRANS+2
JMP DKIN
DKOUT  JMP
.INIT 3,1,PARIN1
TRANS  .TRAN 3,1,0,BUF1,256
.WAIT 3
.CLOSE 3
ISZ TRANS+2
JMP DKOUT
DKSAV  JMP
.INIT 3,1,PARIN1
TRANS  .TRAN 3,1,4,BUF1,256
.WAIT 3
.CLOSE 3
ISZ TRANS+A2; JMP DSAV
ACCPA  JMP* JMS ACCPA; JMP *,3; CATS; DUMP; JMP* ACCPA
PRINTR  LAC CTC; DAC_PDBC; JMS PLROU1; JMS PLROU2
DOIT2  JMS DKIN
WRITEH 6,FRTM2
WRITEH 6,FRTM11
JMS PRINTH
ISZ CT
JMP DOIT2
DATINO  WRITEH 4,FRTM1
PT  JMS ACCPA
LAC DUMP; JMS ROUTIN
SAD (41); JMP DACOCA /33(10)
SAD (0); JMP PLOTIS
SAD (3143); JMP PLOTIP /280(10)
SAD (5); JMP PRINTR
.EJECT
SAD (6); JMP ADFFA
SAD (12); JMP SAV1; JMS DOAGAN; JMP DATINO
WPINI 0; LAC (1); DAC* WPOINT; JMP* WPINI
WPOINT 0
//DAC* WPOINT WITH 0 OR 1 FOR SCOPE
//OR XY-PlotTER, RESPECTIVELY, WPOINT SHOULD
//BE LOADED INDIRECTLY IN MAIN PROGRAM, AND DIRECTLY
//IN A SUBROUTINE OTHER THAN SUBR. POINT,
// DTIN 0
  .INIT 1®, 0, PARIN1
TTRANI .TRAN 1®, 2, 0, BUF1, 256
         , WAIT 1®
         , CLOSE 10
         ISZ TTRANI+2; JMP* DTIN
DTOUT 0
  .INIT 1®, 1, PARIN1
TTRANO .TRAN 1®, 1; 0, BUF1, 256
         , WAIT 1®
         , CLOSE 10
         ISZ TTRANO+2; JMP* DTOUT
// //
DTIO 0; WRITEH 4, FRMT37
   JMS* ACCEPt; JMP .+4; DATS; FRUNIT; STABLI
   WRITEH 4, FRMT39
   JMP DTIO1
P RR 0; JMS* ACCEPt; JMP .+5; DATS; TOUNIT; STABLO
   POINTS
   JMS PRELIM; LAC NBC; DAC PNBC; JMS PLROU1
   JMS PLROU2; JMP* PRR
DTIO1 JMS PRR; JMS DGTIO; JMP* DTIO
DGTIO 0
     LAC STABL#; DAC TRAN1+2; DAC TTRAN1+2
     LAC STABLO#; DAC TRANO+2; DAC TTRANO+2
     LAC (BUF1); DAC TRAN1+3; DAC TRANO+3
     DAC TTRAN1+3
DOIT40 LAC FRUNIT#; SAD (3); JMP TDKIN
   SAD (10); JMP TDTIN; JMS DOAGAN; JMP PARIN1
DOIT41 LAC TOUNIT#; SAD (3); JMP TDKOUT
   JMP TDTOUT; JMS DOAGAN; JMP PARIN1
DOIT42 ISZ CT; JMP DOIT40
         JMP* DGTIO
// //
TDKIN JMS DKIN; JMP DOIT41
//
TDKOUT JMS DKOUT; JMP DOIT42
TDTOUT JMS DTOUT; JMP DOIT42
TDTIN JMS DTIN; JMP DOIT41
,EJECT
//
// FRMT38 .ASCII "(1X,23HT0 UNIT#: STABLK! #PTS:)
// FRMT37 .ASCII "(1X,17HFR UNIT#: STABLK:)
DKOTIO JMS DTIO; LAC (35); DAC PLACE
JMP DATINO
,IODEV 1W
//
// ROUTIN 0; SAD (7); JMP PARIN1
SAD (1750); JMS* ACOR1 /1000(10), TO GO TO CORREL,
SAD (74); JMP RENT6; SAD (106); JMP COR
SAD (144); JMP START; SAD (35); JMP DKOTIO /29(10)
SAD (11); JMP GTIO; SAD (13); JMP CEPSTR
//CEPSTR ASSUMES STABLK 64(10) HEADS FILE OF AMPL, IN DB,
SAD (954); JMS DRAWAS /300(10), AXES ON SCOPE,
SAD (455); JMS DRAWAP /301(10), AXES ON PLOTTER,
SAD (621); JMS DRAWAP /401(10), AXES ON PLOTTER,
//THIS OPTION DRAWS X & Y AXES AND PRINTS OUT 'DO AGAIN'
//PLEASE!', THEN 'GO!' FOR PLOTTING OF DATA PROCESSED
//JUST BEFORE THIS OPTION WAS CALLED UPON,
SAD (622); JMS DRAWAS
SAD (5670); JMS* PPRR /3000(10), GO TO PPRR
//OPTION 400, ONLY X AND Y AXES ARE DRAWN, NO MARKINGS,
//SAME WITH OPTION 401 FOR PLOTTER,
SAD (764); JMS DRAWAS /500(10)
SAD (765); JMS DRAWAP /501(10)
//THESE OPTIONS DRAWS AN AXIS WITH MARKINGS AT X=-50.
//THIS IS BETTER FOR FFT PLOTS.
JMP* ROUTIN
//
PRINTH 0
.WRITE 6,2,DATE,52
WRITMN 6,FRMT63,20,10,BUF1
JMP* PRINTH
//
// INV 0; CMA; TAD (1); JMP* INV
// DACOCA JMS DACOCA; JMP DATINO
//
// BACK1 DAC PLACE; JMP COR
PLOTIP JMS WPINI; JMP ,+2
PLOTIS D2M* WPOUN1; LAC CTC; DAC PNB#; JMS PLOTG
JMP DATINO
PLOTG 0; JMS PLROU1; JMS PLROU2; JMS MODONE; JMS DKIN
LAC* BUF1P; DAC Y0#
JMS DELAY0; JMS DELAY1; JMS PLENT0; JMP* PLOTG
,EJECT
DRAWP

; JMS WPINI; JMS AX; LAC (455); JMP* DRAWP

DRAWS

; DMS WPINT; JMS AX; LAC (454); JMP* DRAWS

//

MODONE

; LAC PNBC; SAD (0); JMP MOD0; SAD (-1); JMP ,+3

LAC ONE; JMP ,+2; LAC TWO; DAC XINC; JMP* MODONE

//

AX

; JMS* POINT; JMP ,+4; ZERO; P250; ZERO; JMS DELAYG

JMS* POINT; JMP ,+4; ZERO; P250; ONE; JMS DELAYG

LAC DUMP; SAD (454); JMP MARK22; SAD (764)

JMP MARK33; SAD (765); JMP MARK33

SAD (455); JMP MARK22

JMS* LINE; JMP ,+3; ZERO; M250; JMS DELAYG

JMS* POINT; JMP ,+4; ZERO; M250; ZERO; JMS DELAYG

LAC DUMP; SAD (454); JMS MARK00

SAD (455); JMS MARK00

AX1

; JMS* POINT; JMP ,+4; ZERO; ZERO; ZERO; JMS DELAYG

JMS DELAYG

JMS* POINT; JMP ,+4; ZERO; ZERO; ONE; JMS DELAYG

JMS* LINE; JMP ,+3; P777; ZERO; JMS DELAYG

JMS* POINT; JMP ,+4; P777; ZERO; ZERO; JMS DELAYG

JMP* AX

MARK22

; D2M XE

JMS* POINT; JMP ,+4; ZERO; M250; ONE; JMS DELAYG

MARK3

LAC XE; TAD 64; DAC XE; SAD (1000); JMP ,+3

JMS MARK1; JMP MARK0

TAD (-1); DAC XE

JMS* LINE; JMP ,+3; XE; M250; JMS DELAYG

JMS* POINT; JMP ,+4; XE; M250; ZERO; JMS DELAYG

JMP* MARK00

MARKYY

; LAC P250; DAC YE; JMS MARK3

MARK2

LAC YE; TAD 50; DAC YE; SAD (-454); JMP* MARKYY

JMS* LINE; JMP ,+3; ZERO; YE; JMS DELAYS

JMS MARK3

JMP MARK2

//

P250 372; M50 -62; P12 14; M250 -372; M34 -44; P32 40; M83 -123

MARK3

; JMS* LINE; JMP ,+3; P12; YE; JMS DELAYS

JMS* LINE; JMP ,+3; ZERO; YE; JMS DELAYS

JMP* MARK3

//

MARK22

JMS MARKYY; JMP AX1

MARK33

JMS MARKY1; JMP AX2

AX2

; JMS* POINT; JMP ,+4; ZERO; M50; ZERO; JMS DELAYS

JMS MARK11; JMP AXE

EJECT
MARK11 0; OZH XE
JMS* POINT; JMP ,+4; ZERO; M50; ONE; JMS DELAYG
JMS MARK4
MARK01 LAC XE; TAD P32; DAC XE; SAD (1000); JMP ,+3
JMS MARK4; JMP MARK01
TAD (-1); DAC XE
JMS* LINE; JMP ,+3; XE; M50; JMS DELAYG
JMS* POINT; JMP ,+4; XE; M50; ZERO; JMS DELAYG
JMP* MARK11
MARKY1 0; LAC P230; DAC YE; JMS MARK3
MARK21 LAC YE; TAU M33; DAC YE; SAD (-122); JMP* MARKY1
//THIS MARKS Y-AXIS WITH INTERVALS OF 16 DB EACH,
JMS* LINE; JMP ,+3; YE; JMS DELAYS
JMS MARK3; JMP MARK21
//
M377 -377; P777 777; P64 10U; M356 -356
C; CT; KP; WORD
//
AVECT; SPNF; WEIGH
//
MARK1 0
JMS* LINE; JMP ,+3; XE; M250; JMS DELAYS
JMS* LINE; JMP ,+3; XE; M356; JMS DELAYS
JMS* LINE; JMP ,+3; XE; M250; JMS DELAYS
JMP* MARK1
//
MARK4 0
JMS* LINE; JMP ,+3; XE; M50; JMS DELAYS
JMS* LINE; JMP ,+3; XE; M34; JMS DELAYS
JMS* LINE; JMP ,+3; XE; M50; JMS DELAYS
JMP* MARK4
ACOCA 0; LAC NB; TAD (300); DAC TOSTBL
//TO USE THIS OPTION, LOAD DATA WITH OPTION 9, INTO
//BLOCK 192(10), BEFORE TYPING 33.
//CONVERTED DATA WILL BE RETURNED AT BLOCK 64(10), READY
//FOR OTHER OPTIONS, INCLUDING FFT.
//FFTCONVERSION PICKS UP DATA AT 64(10),
LAC (300); DAC FRSTBL; JMS TIOS
//THIS OPTION PRODUCES THE PART OF THE AUTOCORRELATION
//FUNCTION FOR NEGATIVE TAU BEFORE TAKING THE FFT,
//UNNORMALIZED VALUES OF AUTOCORREL. FCT. ARE USED
//FOR BETTER PRECISION, THIS OPTION VALID ONLY FOR #PTS=256,
//BUT CAN BE ARRANGED LATER TO ACCOMODATE OTHER PTS#,
JMS PLROU2; LAC (BUF2); DAC TRANO+3
ACOFF0 JMS PLGU1; OZH BUF2; ISZ FCT; ISZ BUF1P
+EJECT
ACOFF1

JMS DKin; LAC (BUF2+400); DAC BUF2P
LAC BUF2P; TAD (-1); DAC BUF2P
LAC* BUF1P; DAC* BUF2P
ISZ BUF1P; ISZ FCT; JMP ACOFF1
LAC TOSTBL; TAD (-1); DAC TOSTBL; DAC TRAN0+2
JMS DKOUT
LAC (120); DAC TRAN0+2; LAC (300)
DAC TRAN1+2; LAC (BUF1); DAC TRAN1+3; DAC TRAN0+3
JMS DKIN; JMS DKOUT; JMS DKin; JMS DKOUT
LAC MWORD; JMS INV; JMS* AD; LAC NB
CALL; RAL; DAC POINTS; JMS PRELIM
LAC (2); DAC PLACE
JMP* AC0CA

MOD0

LAC WORD; TAD (*200); SMAXSZA; JMP *3; LAC FOUR
JMP *2
LAC TWO; DAC XINC; JMP* MODONE

PLENT0

0; D2M V

PLENT1

LAC V; DAC XE#; LAC* BUF1P; DAC YE#
JMS* LINE; JMP *3; ESA XE; ESA YE
ISZ BUF1P; ISZ NB2C; JMP DONTX

DOXX

LAC V; TAD XINC#; DAC V; LAC NB2#; DAC NB2C
DONTX
ISZ FCT; JMP PLENT1; ISZ CT

//
THIS ACTION IS FOR NB=1 & 2 TO HAVE FULL SCALE.
JMP *2; JMP *4; JMS PLROU1; JMS DKin; JMP PLENT1
LAC (1); DAC XINC
JMS DELAY2; JMS DELAYE; JMS DELAY0; JMP* PLENT0

//

PLROU0

0; LAC MWORD; DAC FCT
LAC (BUF1); DAC BUF1P; LAC (BUF2); DAC BUF2P
JMP* PLROU0

//

PLROU1

0; LAC MWORD; DAC FCT
LAC (BUF1); DAC BUF1P
LAC (BUF2); DAC BUF2P
LAC (F); DAC FP; JMP* PLROU1

//

PLROU2

0; LAC PNBC; SAD (0); LAC (-1); DAC CT; SAD (-1)
JMP ROU2EN

JMS* AE; LAC (2)

ROU2EN

DAC NB2; DAC NB2C#
LAC PLACE; SAD (3); JMP PLOTD3; SAD (1); JMP PLOTD1
SAD (15); JMP PLOTD8
SAD (35); JMP PLOTKT

PLOTD2

LAC (14C); JMP PLOTN
PLOTD1

LAC (0); JMP PLOTN
PLOTD8

LAC AVEIT; JMP PLOTN
PLOTKT

LAC STABLE; JMP PLOTN
PLOTD3

LAC (2C); PLOTN
DAC TRAN1+2; DAC TTRAN1+2; LAC (BUF1); DAC TRAN1+3

.EJECT
DAC TTRANI+3
JMP* PLRQOU2

//
DELAYG 0; LAC (-200); DAC NBC#
D1 JMS* ALOG10; JMP ,*2; ONE; ISZ NBC
JMP D1; JMP* DELAYG

DELYS 0; LAC (-100); DAC NBC
D2 JMS* ALOG10; JMP ,*2; ONE; ISZ NBC
JMP D2; JMP* DELAYS

//
DELAYG 0; JMS* POINT; JMP ,+4; ZERO; Y0; ZERO
JMS DELAYG; JMS DELAYG; JMP* DELAY0

DELAY1 0; JMS* POINT; JMP ,+4; ZERO; Y0; ONE
JMS DELAYG; JMP* DELAY1

DELAY2 0; JMS* POINT; JMP ,+4; XE; YE; ONE
JMS DELAYG; JMP* DELAY2

//
DELAYE 0; JMS* POINT; JMP ,+4; XE; YE; ZERO
JMS DELAYG; JMP* DELAYE

//
BREAK1 0 /FOR PAPER TAPE SEGMENTATION PURPOSE,
BUFFOUT ,BLOCK 17
DATOUN LAC EXSIGN; SAD (1); JMP ,*2; JMS DIVBYN
WRITE 4,FMT51
JMS* ACCEPT; JMP ,+3; DATS; POINTS
LAC POINTS; JMS* ,AE; LAC (400); DAC BLKOUT
LAC ,CO; SAD (0); LAC (400); DAC WORD
JMS INV
DAC MWORD /*FOR PLOT,
LAC BLKOUT#; JMS* ,AD; LAC (4); DAC LS#
LAC CPWR#; SAD (1); JMP ,*2; JMP ,+3
JMS CPACTV; JMP ,+2; JMS CPDEAC
LAC CEPSW#; SAD (1); JMP ,+2; JMP ,+3
JMS CEAONLY; JMP ,+2; JMS CEDEAC

DATOUF JMS PR1; JMS ROUTIF
DATOU1 DUM TA1; DUM TA1+1 /*DON'T MOVE THIS,
FINMAX JMS DKin
F1 JMS AMP; JMS CUMPWR
JMS MAXFF
ISZ FCT; JMP F1
LAC (-100); DAC FCT; LAC (F); DAC FP
ISZ CT; JMP FINMAX
JMS MAXC

SCALFF JMS ROUTIF
SC1 JMS DKin; JMP SC2 /*OR NOP.
LAC CTC; SAD (0); JMP ,*2; JMP SC2
LAC C; SAD (-2); JMP SC3
SC2 JMS AMP; JMS CUMPWR; JMS AMPLOG
//
LEAVE ABOVE IN IF ONLY NB/2 WANTED,
,EJECT
152

SC3

OATFFT

JMS DOUT1; LAC (BUF1); DAC B1P
JMS DOUT2; LAC (BUF2); DAC B2P
JMS DOUTF; LAC (F+400); DAC B3P
ISZ CT; JMP SC1; LAC (BUF1); DAC TRANI+3

//

AMP

JMS* .AG; .USA FP+400000
JMS* .AK; .DSA FP+400000
JMS* .AH; .DSA XV
ISZ FP; ISZ FP
JMS* .AG; .USA FP+400000
JMS* .AK; .DSA FP+400000
ISZ FP; ISZ FP
JMS* .AI; .DSA XV
JMS* .AH; .USA XV; JMP* AMP

CUMPWR

JMP POWER1

POWER1 JMS* .AG; .USA POWERG; JMS* .AI; .DSA XV
JMS* .AL; .USA HALF; JMS* .AI; .DSA SUM
JMS* .AH; .USA SUM

POWER0 ISZ CT3; LAC CT3; SAD (2); JMS POWERA
JMS* .AG; XV; JMS* .AH; POWERG; JMP* CUMPWR

POWERA JMS* .AG; SUM; JMS* .AH; POWERB; JMP* POWERB

POWERB .BLOCK 2

/// ///

AMPLOG

JMS* SQRT; JMP ,+2; .DSA XV
JMS* .AH; .USA BUFOUT+5
JMS* .AL; .USA KK
JMS* .AH; .USA BUFOUT+7
JMS* .AX; DAC* B1P; ISZ B1P
JMS* .AG; BUFOUT+5; JMS* .AJ; ONE
,EJECT
LAC A; SPA! SNA! CLA; CLC; SNA; JMP A22; JMP A23
A23
LAC A; SPA; ONE; JMS A; AH; BUFOUT+5
A22
JMS A; AH; DSA BUFOUT+5
JMS A; ALOG10; JMP +2; DSA BUFOUT+5
JMS A; AK; DSA TWENTY; JMS A; AJ; DSA MAXLOG
JMS A; AH; DSA BUFOUT+11
JMS A; AI; MAXLOG; JMS A; AK; RFIVE
JMS A; AX
TAD (372); TAD MIMAXL; SPA; CLA; DAC A; B2P; ISZ B2P
// GIVES ABOUT 50 DB IN DYNAMICS,
JMS A; AH; JMS A; AH; BUFOUT+13
CPSW
NOP / OR JMS A; AMPL0G
JMS A; AG; DSA SUM; JMS A; AL; DSA POWERM
JMS A; AH; DSA BUFOUT+15; JMS A; AK; DSA R250
JMS A; AX; DAC A; B3P; ISZ B3P
JMP A; AMPL0G
TWENTY
,DSA 660005
,DSA 243000
ONE
,DSA 631001; DSA 200000
/NUMBER
,DSA 664755; DSA 206157 / 1E-5
/NUMBER
,DSA 463775; DSA 314631 / 0.1.
// //
PRINIT
JMS CPACTV; JMS ROUTIF; JMS HEADER
JMS PR1; JMP BLRD6
// PR1
J
LAC (4VC)
DAC BLRD6+6
LAC LS; SAU (0); LAC (2)
PR2
CH
TAD (1
- DAC CTC# /KEEP THIS CONT.
DAC OUTCT1#
JMS A; AE; LAC (4)
DAC CTC# /KEEP THIS CONT.
DZK BUFOUT
LAC (1
DAC NUMBLK#
LAC SKIPNO
DAC SKIPCT#; JMP A; PR1
///
BLRD6
.INIT 3, 0, PARIN1
.TRAN 3, 0, 0, F, 256
.WAIT 3
.CLOSE 3
ISZ BLRD6+6
LAC IF
DAC BF1PT
LAC (-180
DAC OUTCT2#
.EJECT
LAC (1)
SAD NUMBLK
JMP DAT4GO
ISZ SKIPCT
JMP SKIP4
LAC SKIPNO
DAC SKIPCT
JMP DAT4GO

MAXFF 0 /INPUT FOR SORTING IS XV,

SORT
JMS*,AJ; ,DSA TAI /ASSUME ONLY + VALUES,
LAC*,AB; SMA!SZA!CLA; CLC; SNA
JMP* MAXFF
JMS*,AG; ,DSA XV; JMS*,AH; ,DSA TAI; JMP* MAXFF

////////

MAXC 0

JMS* SORT; JMP +2; ,DSA TAI
JMS*,AH; ,DSA TAI
JMS*,AL; ,DSA R250
JMS*,AH; ,DSA KK
JMS* ALG10; JMP +2; ,DSA TAI
JMS* AK; TWENTY; JMS* AH; MAXLOG; JMS* AK; Rfive
JMS* AX; DAC IMAXL#; JMS INV; DAC MIMAXL#
LAC CTC2; SAD BLOFC; JMP A24
JMS* AG; SUM; JMS* A1; POWERB; JMS* AK; HALF
JMS* AH; POWERM
JMP* MAXC

A24 JMS* AG; SUM; JMS* A1; POWERB; JMS* AH; POWERM
JMP* MAXC

MAXLOG 0; 2

ROUTIF 0

LAC (BUF1); DAC B1P#; DAC TRAN01+3
LAC (F); DAC FP; LAC (F+400); DAC B3P#

// THIS MAKES USE OF LOWER HALF OF F-BUFFER,
DAC TRANOF+3; LAC (300); DAC TRANOF+2
LAC (F); DAC TRANI+3; LAC (BUF2); DAC TRAN02+3
DAC B2P#
LAC (440); DAC TRANI+2; LAC (100); DAC TRAN02+2
LAC (-4); DAC C; LAC (-100); DAC CFT
LAC (2); DAC TRAN01+2; LAC BLOFC; DAC CT
DZ M SUM; DZ M SUM+1; DZ M CT3#
JMP* ROUTIF

////////

DALDAA SAD (145); JMP PLS1 /101(10)
SAD (146); JMP PLS1+2 /102(10)
SAD (147); JMP PLS1+4 /103(10)
SAD (311); JMP PLP1 /201(10)
SAD (312); JMP PLP1+2 /202(10)
SAD (313); JMP PLP1+4 /203(10)
,EJECT
JMS DOAGAN; JMP DATFFT

PLS1
LAC (1); JMP PLOTOK
LAC (2); JMP PLOTOK
LAC (3); JMP PLOTOK

PLP1
LAC (1); JMP PLTG
LAC (2); JMP PLTG
LAC (3); JMP PLTG

PLTG
DAC PLACE; JMS WPINI; JMP PLOTOK+2

PLOTOK
DAC PLACE; DZM* WPPOINT
LAC CTC; DAC PNBC; JMS PLOTG; JMP DATFFT

//
TWO
000002; FOUR
000004

BLRDGO
ISZ NUMBLK
JMP BLRD6

DAT4GO
JMS DAT4
JMS POLAR
JMP PR1N:

SKIP4
ISZ BF1PT
ISZ BF1PT
ISZ BF1PT
ISZ BF1PT
ISZ BUFOUT
ISZ OUTCT2
JMP SK
ISZ OUTCT1
JMP BLRDGO
JMP FINAL

SK
ISZ SKIPCT
JMP SKIP4
LAC SKIPNO
DAC SKIPCT
JMP DAT4GO

//
DAT4
LAC (BUFOUT+1
DAC BFOUTP#
LAC (-4
DAC DAT4CT#

DAT4LD
LAC* BF1PT
ISZ BF1PT
DAC* BFOUTP
ISZ BUFOUTP
ISZ DAT4CT
JMP DAT4LD
JMP* DAT4

//
POWERG
.BLOCK 2

POWERM
.BLOCK 2

SUM
.BLOCK 2
.EJECT
DUMP

HEADER

\begin{verbatim}
;BLOCK 42

;INIT 6,1,PARIN1
WRITEH 6,FMT35
WRITED 6,FMT36,NFULL
LAC ESSIGN
SAD (1)
JMP NEG; JMP POS

CONTIN

WRITEH 4,FMT41
;WRITE 6,2,DATE,52
;INIT 4,0,PARIN1
;READ 4,2,DUMP,34
;WAIT 4
;WRITE 6,2,DUMP,52
;WAIT 6
;CLOSE 4
;CLOSE 6
WRITEH 6,FMT43
WRITEH 6,FMT44
WRITEH 6,FMT441

WN

WRITEH 4,FMT46
JMS* ACCEPT
JMP ,+3
;DSA DATS
;DSA DUMP
LAC DUMP
CMA \NOTE THIS ISN'T 2'S COMP.
DAC SKIPNO#
JMP* HEADER

TIO

WRITEH 4,FMT21
LAC BLOFCC /GIVES NB=256 PTS.
DAC OUTCT1
LAC (BUF1); DAC TRANI+3; DAC TRAN0+3
LAC (0); DAC TRAN0+2

TRANIO

JMS DKKIN; JMS DKOUP
DONE

ISZ OUTCT1
JMP TRANIO
WRITEH 4,FMT20
JMP REENT2

NEG

WRITEH 6,FMT37
JMP CONTIN

POS

WRITEH 6,FMT40
JMP CONTIN

PRIN

JMS* ,FW
;DSA IODEVU; ,DSA FMT30

PRI3

LAC (-7)
DAC PRINC
JMS* ,FE
;EJECT
\end{verbatim}
BUFLOC
.DSA BUFOUT
JMS* ,FE
.DSA BUFOUT+1
ISZ BUFLOC
ISZ BUFLOC
ISZ PRIMCT
JMP BUFLOC+1
JMS* ,FF
LAC (BUFOUT+1
DAC BUFLOC
ISZ BUFOUT
ISZ OUTCT2
JMP ,+4
ISZ OUTCT1
JMP BLROGO
JMP FINAL
ISZ SKIPCT
JMP SKIP4
LAC SKIPNO
DAC SKIPCT
JMP DATORGO

IDDEVQ 6
TEMPR .BLOCK 2
POLAR 0; JMS* ,AG
.DSA BUFOUT+1
JMS* ,AK
.DSA BUFOUT+1
JMS* ,AH
.DSA TEMPR
JMS* ,AG
.DSA BUFOUT+3
JMS* ,AK
.DSA BUFOUT+3
JMS* ,AI
.DSA TEMPR
JMS* ,AH; ,DSA XV

POLAS JMS CUMPFR; JMS AMPLOG; JMP* POLAR
//
NREAL .BLOCK 2
//
DIVBYN 0; LAC BLROC; DAC CT
LAC (400000); DAC TRAN0+2; DAC TRAN1+2
DIVIN LAC (BUF1); DAC TRAN0+3; DAC TRAN1+3
DAC POINT1#
LAC (-200
DAC DIVCT#
JMS OKEON
DIVGO JMS* ,AG
.DSA POINT1+400000
,EJECT
JMS° .AL
.JSA NREAL
JMS° .AH
.JSA POINT1+400000
ISZ POINT1
ISZ POINT1
ISZ DIVCT
JMP DIVGO
JMS DKOUT
ISZ CT; JMP DIVIN
JMP* DIVBYN

//
FINAL WRITEH 6,FMT50
JMP DATFFT
DEG
.JSA 02010
.JSA 2640000
PI
.JSA 552002
.JSA 311037
DEGRAD
.BLOCK 2
BLOKOT
.BLOCK 1
BLOKIN
.BLOCK 1
ZERO 0; 0
ON 377
R250
.JSA 02016; .JSA 372000
RFIVE 000003; 240000
R81880
061004; 203004
M81880
.BLOCK 2
XV
.BLOCK 2
LL
.BLOCK 2
NBB
.BLOCK 2

KK
.BLOCK 2
TA1
.BLOCK 2

PRINTG 0; LAC NBC; DAC PNBC; JMS PLROU1; JMS PLROU2
DOIT5
JMS DKIN
WRITED 6,FMT4,KP
WRITED 6,FMT7,KB
JMS PRINTH
ISZ CT
JMP DOIT5; JMP* PRINTG

GTIO WRITEH 4,FMT22
JMS* ACCEPT; JMP *,+5; DATS; FRSTBL#; TOSTBL#; POINTS
JMS PRELIM; JMS TIOS; JMP DATINO
TIOS 0; LAC FRSTBL; DAC TRANI+2; LAC TOSTBL; DAC TRANO+2
LAC NBC; DAC CT; LAC (BUF1); DAC TRANI+3
DAC TRANO+3
DOIT14
JMS DKIN; JMS DKOUT; ISZ CT; JMP DOIT14
,EJECT
LAC TO STBL; SAD (0); JMP PLO1
SAD (13); JMP PLO2; DAC AVECT; LAC (15); JMP GTIP
PLO1 LAC (1); JMP GTIP
PLO2 LAC (2)
GTIP DAC PLACE; JMP* TIOS

S A V 4 LAC S A V E C T / FOR S A V E O F FFT R E S U L T S ,
SAD (0); JMP SA1
SAD (1); JMP SA2
SAD (2); JMP SA3; JMS DOAGAN; JMP DATFFT
SA1 LAC (1); JMP SAG
SA2 LAC (2); JMP SAG
SA3 LAC (3); JMP SAG
SAG DAC PLACE; JMS SAVE; ISZ SAVEC; JMP DATFFT

C E P S T R LAC (13); DAC HAHASH#; LAC (40); DAC WORD
JMS INV; DAC MWORD
JMP ADFFA

S M O O T H WRITEH 4,FRMT6
JMS* ACCEPT; JMP ,+3; DATS; SPNBG; DZM EXSIGN
LAC (1); DAC COEFF; LAC NBB; DAC NBBSA#
LAC SPNBBS#; DAC NBB
JMS HANNIN; LAC (-1); DAC CT1; DAC CT2
LAC (10); DAC COEFF; LAC NBBSA; DAC NBB
JMS* ,AG; XV; JMS* ,AI; ONE; JMS* ,AH; XV
LAC (40); DAC TRANS1+2; LAC (0); DAC TRANS0+2
LAC (BUF1); DAC TRANS1+3; DAC TRANS0+3
LAC BLOFC; DAC CT; JMS SM6; JMP SM3

SM6 0; JMS DKIN; LAC (-100); DAC FCT; LAC (BUF1)
DAC BUF1P; JMP* SM6
SM3 JMS FINMUL; LAC HCT; SAD SPNBG; JMP SM5
JMS* ,AG; ,DSA BUF1P+400000
JMS* ,AK; MULTF; JMS* ,AH; ,DSA BUF1P+400000
ISZ BUF1P; ISZ BUF1P; JMS* ,AG; ,DSA BUF1P+400000
JMS* ,AK; MULTF
JMS* ,AH; ,DSA BUF1P+400000; ISZ BUF1P; ISZ BUF1P
JMP SM7
SM5 DZM* BUF1P; ISZ BUF1P; DZM* BUF1P; ISZ BUF1P
DZM* BUF1P; ISZ BUF1P; DZM* BUF1P; ISZ BUF1P
ISZ FCT; JMP SM5; JMS SM4; ISZ CT; JMP SM5
JMP REENTJ
SM7 ISZ FCT; JMP SM3; JMS SM4; ISZ CT; JMP SM3
JMP REENTJ
SM4 0; JMS OKOUT; JMS SM6; JMP* SM4

D K A V E 0
,INIT 3,1,PARIN1
,EJECT
TRANAV
.TRAN 3,1,0,BUF2,256
.WAIT 3
.CLOSE 3
ISZ TRANAV+2; JMP* DKAVE

//
ACSPNB
0; JMS* ACCEPT; JMP .+3; DATS; SPNB; JMP* ACSPNB
ACCB1
0; LAC* FP; DAC TRANI+2; LAC (BUF1); DAC TRANI+3
ISZ FP; JMS DKN; JMP* ACCB1
ACCB2
0; LAC* FP; DAC TRANI+2; LAC (BUF2); DAC TRANI+3
DAC BUF2P; LAC (-400); DAC FCT
ISZ FP; JMS DKN
ACCC2
LAC* BUF2P; JMS* ,AD; LAC WEIGH; DAC* BUF2P
ISZ BUF2P; ISZ FCT; JMP ACCC2; JMP* ACCB2

//
FFTCON
LAC (12)
DAC TRANI+2
LAC WORD; SAD (400); JMP FI; SAD (200); JMP ,+3
JMS DOAGAN; JMP PARIN1

// THIS ALLOWS FFT OF DOWN TO 128(1) PTS.
FI
LAC (F); DAC FP; DAC TRANI+3
LAC (0)
DAC TRANI+2 /TO INIT, DKOHTT,
LAC (BUF1)
DAC BUF1P; DAC TRANI+3
LAC (-2:0); DAC FCT
LAC (-2); DAC C
LAC NBB
JMS* ,AD
DAC (2)
DAC ADDBLCT
LAC WORD; SAD (200); JMP ,+2; JMP W18; LAC (-1)
DAC ADDBLCT
W18
LAC HAHASK; SAD (13); JMP W19; JMP W20
W19
DZM DUMP; JMP W21
W20
WRITEH 4,FMT55
READ 4,FMT3,DUMP
W21
LAC DUMP; SAD (1); JMP YESHAN; SAD (2); JMP HANANO
LAC (NOP); DAC FFTR; DZM HCT
LAC (JMP FSW); DAC FSV
JMS FFTB; JMP OTOFGO
YESHAN
JMS HANNIN; JMS FFTB; JMP OTOFGO

//
HANNIN
0; LAC (NOP); DAC FSV
DZM HCT#; LAC NBB; JMS* ,AE; LAC COEFF; DAC CT1#
LAC NBB; JMS* ,AY; LAC CT1; DAC CT2#
LAC (JMS FINMUL); DAC FFTR
JMS* FLOAT; JMP ,+2; ,OSA NBB; JMS* ,AH; ,OSA NBBR
JMS* ,AJ; ,OSA ONE; JMS* ,AH; ,OSA XV
JMS* FLOAT; JMP ,+2; ,OSA COEFF; JMS* ,AK; ,OSA PI
,EJECT
FFT B
JMS D K I N

FFT R
JMS F I N M U L
JMS F L O A T
JMP +2
,DSA BU F 1 P +400000
ISZ BU F 1 P

FS V
NOP / OR J M P F S W
JMS , AH; , DSA M U L T F

FS W
JMS , AH
, DSA FP +400000
ISZ FP
ISZ FP
D Z M , FP
ISZ FP
D Z M , FP
ISZ FP
ISZ F C T
JMP F F T R
L A C (-200
D A C F C T
JMS D K O U T T
L A C (F
D A C FP
ISZ A D B L C T
JMP G O A 0 1
JMP F F T B

G O A D 1
ISZ C
JMP F F T R
L A C (-2
D A C C
L A C (B U F 1
D A C BU F 1 P
JMP F F T R +1

F I N M U L
0
JMS F L O A T; JMP +2; ,DSA H C T; JMS , AH; , DSA LL
L A C H C T; JMS , AY; L A C C T 1; S P A; J M P H A N 1; L A C H C T
JMS , AY; L A C C T 2; S M A; J M P H A N 2
L A C O N E ; D A C M U L T F; L A C O N E +1; D A C M U L T F +1; ISZ H C T
JMP F I N M U L

H A N 1
JMS , AG; , DSA LL; JMS , AL; , DSA N B B R
JMS , AK; , DSA P I C D E F; JMS , AH; , DSA T A 1
JMS C O S; JMP +2; , DSA T A 1
JMS , AM; , DSA O N E; JMS , AK; , DSA H A L F
JMS , AH; , DSA M U L T F; ISZ H C T; J M P F I N M U L
,EJECT
HAN2    JMS* ,AG; ,DSA XV; JMS* ,AJ; ,DSA LL; JMP HAN1+2
//
HANAN0 JMS HANNIN; JMS FFTB
            LAC NB; CLL; RAL; DAC NB
// THIS MAKES FFT ACCOMMODATE 2NB WITH 1NB
// FILLED WITH 0'S ALL OVER. THIS ACTION IN TURN
// GIVES A SPECTRUM WITH BETTER SEPARATED FREQUENCIES,
            LAC NBC; CLL; RAL; DAC NBC; DAC CT
DOIT11   LAC (-1%00); DAC FCT; LAC (F); DAC FP
DOIT12   DZH* FP; ISZ FP; ISZ FCT
            JMP DOIT12; JMS DKOUTT
            ISZ CT; JMP DOIT11
            JMP OTOFGo /JMP TOFFT IF WANT PRINTOUT.
//
NBBBR    ,BLOCK 2
HALF     ,DSA 0%0000; ,DSA 200000
PICOEF   ,BLOCK 2
COEFF    000010
//
DKOUTT   ISZ
TRANTT   .INIT 3,1,PARINIT
            ,TRAN 3,1,0,F,512
            ,WAIT 3
            ,CLOSE 3
            ISZ TRANTT+2
            ISZ TRANTT+2
            JMP* DKOUTT
//
BREAK2   @ /FOR TAPE SEGMENTATION PURPOSE, THIS IS THE LAST,
            ,GLOBL FLOAT, ,AV, ,AH, ,AS, ,SQRT, ,DCOS, ,DB, ,AD, ,CF, ,COS
            ,GLOBL ,AA, ,AB, ,AC, ,AW, ,AP, ,AG, ,CC, ,CH, ,AK, ,BA, ,CA
            ,GLOBL ,AL, ,AI
FFTSTA   WRITEH 4,FMT20
REENT1   WRITEH 4,FMT23
            DZH MVAL
            JMS* ACCEPT
            JMP ,+3
            ,DSA DATS
            ,DSA MVAL
REENT2   WRITEH 4,FMT24
            READ 4,FRMT3,EXSIGN
REENT3   DZH BLOKCT#
            JMP TRM0D
/T OFFT  WRITEH 4,FMT19 /LEAVE IN FOR PRINTOUT OF ADFFT,
/FMT19   ,ASCII "(1X,7HLPPFFT:)
/FRMT79  ,ASCII "(1X,8(1X,E13.6)"
/READ 4,FRMT3,DUM
/LAC DVU; SAU (5); JMP:PRINFF; JMP OTOFGo
/PRINFF  LAC NBC; JMS* ,AD; LAC (4); DAC CT; LAC (0)
         ,EJECT
DAC TRANS2; LAC (BUF1); DAC TRANS+3
/Doit10 JMS DKin; WRITMN 6,FRMT79,10,20,BUF1
/ISZ CT; JMP DOIT10
/
OTOFFT WRITEH 4,FMT20
OTOFG0 LAC NU; JMS*,AD; LAC MWORD; DAC C
//THIS ALLOWS FFT WITH 128(10) POINTS.
DZM MVAl; DZM CT; LAC (1)
R1 JMS*,AD; LAC (2); DAC CT
ISZ MVAl; TAD C
SNAI; JMP R2
LAC CT; JMP R1
R2 DZM BLOKCT; LAC (1)/NEG.
DAC EXSIGN; JMP TRMOD
EXSIGN ,BLOCK 2
MVAl ,BLOCK 1
AA ,BLOCK 1
AB ,BLOCK 1
AC ,BLOCK 1
TRMOD JMS SETUP; DZM SAVECT#; DZM HAHASK
COMPRI LAC NHALF
JMS*,AV/N2 CONV TO FP
JMS*,AV/PI DIVIDED BY N2
,DSA PIAR
JMS*,AP/STORE
,DSA TWPIN
DZM SINE
DZM SINEl+1
DZM SINE+2
LAC (1
DAC PASS#
DAC COSINE
LAC (200000
DAC COSINE+1
DZM COSINE+2
DZM CNTR#
DZM NEWSC#
JMS INIT
DZM RI#
LAC EXSIGN
SAD (1)
JMP ,+3
DZM XSIGNB
JMP ,+3
LAC (400000
DAC XSIGNB
LAC (1
DAC PASCT3#
LAC MVAl; SAD (0); JMP PARIN1
,EJECT
CMAC
   TAD (1
   DAC PASCT1#
   JMS RESET1
   JMP INIT1

RESET1
   0
   LAC (JMP COMPR
   DAC DK62
   DAC DK52
   LAC (JMP MASTR2
   DAC DK5
   DAC DK6
   JMP* RESET1

SETUP
   0
   LAC (643700
   TAD MVAL
   DAC LSHIFT#
   CLL
   LAC (1
   XCT LSHIFT
   DAC NFULL#
   RAR
   DAC NHALF#
   LRS 5
   DAC BLOCF#
   RAR
   DAC BLOCH#
   ALS 11
   DAC WCDKBF#
   JMS* FLOAT
   JMP .+2
   .DSA NFULL
   JMS* .AH
   .DSA NREAL
   LAC BLOCH; CLL; RAR; DAC NB; CMA; TAD (1); DAC NBC
   LAC BLOCF; JMS INV; DAC BLOFC# /KEEP THIS CONSTANT!
   JMP* SETUP

INIT1
   CLA
   DAC DK6IN1+6
   LAC BLOCH
   DAC DK6IN2+2
   LAC BLOCH
   CMA
   TAD (1
   DAC OUTCT#
   LAC (400
   DAC DK6OUT+6
   LAC (JMP DK6IN1
   DAC DK6-1
   .EJECT
ODINIT
JMP DK6IN1
LAC (200)
DAC DK5IN1+6
LAC BLOCH
TAD (200)
DAC DK5IN2+2
LAC BLOCH
CMA
TAD (1)
DAC OUTCT
LAC (400)
DAC DK6OUT+6
JMP DK5IN1

EVINIT
LAC (400)
DAC DK6IN1+6
LAC BLOCH
TAD (400)
DAC DK5IN2+2
LAC BLOCH
CMA
TAD (1)
DAC OUTCT
LAC (200)
DAC DK5OUT+6
LAC (JMP DK5IN1)
DAC DK6-1
JMP DK6IN1

DK5IN1
.INIT 3,0,PARIN1
.TRAN 3,0,0,BUF1,256
.WAIT 3

DK5IN2
.TRAN 3,0,0,BUF2,256
.WAIT 3
.CLOSE 3
.ISZ DK5IN1+6
.ISZ DK5IN2+2

DK52
JMP COMPR

DK6IN1
.INIT 3,0,PARIN1
.TRAN 3,0,0,BUF1,256
.WAIT 3

DK6IN2
.TRAN 3,0,0,BUF2,256
.WAIT 3
.CLOSE 3
.ISZ DK6IN1+6
.ISZ DK6IN2+2

DK62
JMP COMPR

DK5OUT
.INIT 3,1,PARIN1
.TRAN 3,1,0,F,512
.WAIT 3
.CLOSE 3
.EJECT
DK5
JMP MASTR2

DK6OUT
.INIT 3,1,PARIN1
.TRAN 3,1,0,F,512
.WAIT 3
.CLOSE 3
.ISZ DK6OUT+6
.ISZ DK6OUT+6
.ISZ OUTCT
JMP DK6IN1

DK6
JMP MASTR2

MASTR1
LAC (1
SAD PASCT3
JMP DK6OUT
JMP MASTR3

MASTR2
ISZ PASCT3
LAC PASS
RCL
DAC PASS
DZN CNTR
DZN NEASC
DZN SINE
DZN SINE+1
DZN SINE+2
LAC (1
DAC COSINE
LAC (200000
DAC COSINE+1
DZN COSINE+2
ISZ PASCT1
JMP +2
JMP BITREV
LAC PASCT3
AND (1
SZA
JMP DDINIT
JMP EVINIT

MASTR3
LAC PASCT3
AND (1
SZA
JMP DK6OUT
JMP DK5OUT

COMP
LAC (-1#0
DAC ARCNT#

ARITH
JMS,AG
/DNGL LOAD
.DSA BF2PT+400000
.EJECT
JMS*.AP /STORE
JMS*.AA
JMS*.CF /HOLD R2
JMS*.AG /SNGL LOAD
JMS*.BF1PT+400000
JMS*.CC /ADD
JMS*.DSA 42
JMS*.CH /ROUND AND SIGN
JMS*.DSA 1
JMS*.DSA 777776
JMS*.CF /HOLD
LAC*.AB
AND (377777
DAC*.AB
JMS*.CH /ROUND AND SIGN
JMS*.DSA 42
JMS*.DSA 777000
JMS*.AH /SNGL STORE
JMS*.BF3PT+400000
ISZ BF3PT
ISZ BF3PT
ISZ BF1PT
ISZ BF1PT
ISZ BF2PT
ISZ BF2PT
LAC AB /LOAD HELD ACC, WITH -2R2 OR -2I2
SZA
JMP .+5
D2M*.AA
D2M*.AB
D2M*.AC
JMP .+12
XOR (4%0000
DAC*.AB
LAC AC
DAC*.AC
LAC AA
TAD (1
DAC*.AA /ADD (R1+R2)-2R2 OR (I1+I2)-2I2
JMS*.CC /ADD (R1+R2)-2R2 OR (I1+I2)-2I2
JMS*.CH /ROUND AND SIGN
JMS*.DSA 1
JMS*.DSA 777776
JMS*.AP /STORE R1-R2
JMS*.DSA AA
LAC R1 /Ø IF REAL , 1 IF IMAGINARY
SZA
JMP IMAG
.EJECT
LAC XSIGNB /ACC, MINUS IF NEG, EXP.
SPA
JMS*  ,BA /NEGATE
JMS*  ,AP
.DSA RSIN
ISZ RI
JMP ARITH
JMS*  ,AS /MULT
.DSA COSINE /DBL STORE (R1-R2)COS
JMS*  ,AP /DBL STORE
.DSA RCOS /LOAD R1-R2
JMS*  ,AO /DBL MULT
.DSA AA
JMS*  ,AS /DBL MULT
.DSA SINE
LAC XSIGNB /ACC, MINUS IF NEG, EXP.
SMA
JMS*  ,BA /NEGATE
JMS*  ,CF /HOLD ISIN
JMS*  ,AO /DBL LOAD
.DSA RCOS /LOAD I1-I2
JMS*  ,CC /ADD
.DSA 42
JMS*  ,CH /ROUND AND SIGN
.DSA 400
.DSA 777000 /STORE (R1-R2)COS+(11-12)SIN
JMS*  ,AH /DBL LOAD (11-12)COS
.DSA BF3PT+400000
ISZ BF3PT
ISZ BF3PT
JMS*  ,AO /DBL LOAD (11-12)COS
.DSA ICOS /HOLD
JMS*  ,CF /DBL LOAD
.DSA AO /DBL LOAD (11-12)COS+(R1-R2)SIN
.DSA RSIN
JMS*  ,CC
.DSA 42
JMS*  ,CH
.DSA 400
.DSA 777000
JMS*  ,AH /SNGL STORE
.EJECT
,DSA BF3PT+400000
ISZ BF3PT
ISZ BF3PT
DZM RI

SIN COS
ISZ CN1R
LAC CNTR
AND PASS
SAD NEWSC
JMP TARCNT
DAC NEWSC
LAC CNTR
JMS* .AW
JMS* .AS
,DSA TWOPIN
JMS* .AP
,DSA ARG
JMS* .DB
JMS* .AP
,DSA SINE
JMS* D*COS
JMP +2
,DSA ARG
JMS* .AP
,DSA COSINE

TARCNT
ISZ ARCNT
JMP ARITH
JMS INIT
JMP MASTR1

INIT
0
LAC (BUF1
DAC BF1PT
LAC (BUF2
DAC BF2PT
LAC (F
DAC BF3PT
JMP* INIT

SINE
,C BLOCK 3
COSINE
,C BLOCK 3
RCOS
,C BLOCK 3
RSIN
,C BLOCK 3
ICOS
,C BLOCK 3
XSIGNB 0
TWOPIN
,C BLOCK 3
ARG
,C BLOCK 3
PIAR 000002
311037
552421

BITREV
LAC (1
DAC KCONT#
,EJECT
LAC MV AL
CMA
TAD (2
DAC PASCT2#
JMS RESET2
JMS RESET3
LAC MV AL
AND (1
SZA
JMP EVINIT
JMP ODDM1
RESET2
Ø
LAC (JMP BRRT
DAC DK62
DAC DK52
LAC (JMP MASTR5
DAC DK5
DAC DK6
JMP* RESET2
RESET3
Ø
LAC (2kΩ
DAC DK5IN1+6
TAD BLOCH
DAC DK5IN2+2
LAC (4kΩ
DAC DK6IN1+6
TAD BLOCH
DAC DK6IN2+2
LAC BLOCH
CMA
TAD (1
DAC OUTCT
LAC (4kΩ
DAC DK6OUT+6
LAC (2kΩ
DAC DK5OUT+6
JMP* RESET3
MASTR4
LAC MV AL
AND (1
SZA
JMP ODDM1
LAC KCONT
AND (1
SZA
JMP DK6OUT
JMP DK5OUT
ODDM1
LAC KCONT
AND (1
SZA
.EJECT
JMP DK5OUT
JMP DK6OUT

MASTR5
ISZ KCONT
ISZ PASCT2
JMP .+2

AAA
JMP DATOUF
LAC KCONT
TAD (-7
SPA
JMP .+2
JMP BLSORT
LAC MVAL
AND (1
SZA
JMP ODDM2
LAC KCONT
AND (1
SZA
JMP EVINIT
JMP ODINIT

ODDM2
LAC KCONT
AND (1
SZA
JMP EVINIT
JMP ODINIT

BRRT
LAC (-220
DAC BF3CT#
JMS BFPTI
JMS COPY
JMP READ1

BFPTI
Ø
LAC (BUF1
DAC BF1PT#
LAC (BUF2
DAC BF2PT#
LAC (F
DAC BF3PT#
LAC (-4
DAC FOUR1#
DAC FOUR2#
JMP* BFPT1

COPY
Ø
LAC KCONT
CMA
TAD (1
DAC COUNT#
LAC (1

CHECK
ISZ COUNT
JMP ROTAT
,EJECT
CMA
  TAD (1
  DAC COPYCT#
  DAC STCOPY#
  JMP* COPY

ROTA
  RCL
  JMP CHECK

READ1
  LAC* BF1PT
  ISZ BF1PT
  DAC* BF3PT
  ISZ BF3PT
  ISZ FOUR1
  JMP READ1
  LAC (-4
  DAC FOUR1
  JMP CONTR1

READ2
  LAC* BF2PT
  ISZ BF2PT
  DAC* BF3PT
  ISZ BF3PT
  ISZ FOUR2
  JMP READ2
  LAC (-4
  DAC FOUR2
  JMP CONTR2

CONTR1
  ISZ BF3CT
  ISZ COPYCT
  JMP READ1
  LAC STCOPY
  DAC COPYCT
  JMP READ2

CONTR2
  ISZ BF3CT
  JMP *+2
  JMP MASTR4
  ISZ COPYCT
  JMP READ2
  LAC STCOPY
  DAC COPYCT
  JMP READ1

BLSORT
  LAC (402
  DAC BLRD6T+6
  DAC BLWR6+7
  TAD BLOCH
  DAC BLRD6M+6
  LAC (202
  DAC BLRD5T+6
  DAC BLWR5+7
  TAD BLOCH
  DAC BLRD5M+6
  *EJECT
LAC BLOCF
CMA
TAD (1
DAC BLCNT#
JMS GROUP
LAC MVAL
AND (1
SZA
JMP MODD
LAC KCONT
AND (1
SZA
JMP BLRD5T
JMP BLRD6T
MODD
LAC KCONT
AND (1
SZA
JMP BLRD6T
JMP BLRD5T
GROUP
LAC KCONT
TAD (-6
CMA
TAD (1
DAC COUN2#
LAC (1
CHECK2
ISZ COUN2
JMP ROTAT2
CMA
TAD (1
DAC GRUPCT#
DAC GRSTOR#
JMP> GROUP
ROTAT2
RCL
JMP CHECK2
BLRD5T
,INIT 3,?,PARIN1
,TRAN 3,3,0,BUF1,256
,WAIT 3
,CLOSE 3
ISZ BLRD5T+6
JMS BLWR6
JMP CONTR3
BLWR6
0
,INIT 3,1,PARIN1
,TRAN 3,1,0,BUF1,256
,WAIT 3
,CLOSE 3
ISZ BLWR6+7
JMP> BLWR6
.EJECT
BLRD6T .INIT 3,0,PARIN1 .TRAN 3,0,0,BUF1,256 .WAIT 3 .CLOSE 3 ISZ BLRD6T+6 JMS BLWR5 JMP CONTR4

BLWR5 0 .INIT 3,1,PARIN1 .TRAN 3,1,0,BUF1,256 .WAIT 3 .CLOSE 3 ISZ BLWR5+7 JMP* BLWR5

BLRD6M .INIT 3,0,PARIN1 .TRAN 3,0,0,BUF1,256 .WAIT 3 .CLOSE 3 ISZ BLRD6M+6 JMS BLWR5 ISZ BLCNT ISZ BLCNT JMP CONTR6 JMP MASTR5

BLRD5M .INIT 3,0,PARIN1 .TRAN 3,0,0,BUF1,256 .WAIT 3 .CLOSE 3 ISZ BLRD5M+6 JMS BLWR6 ISZ BLCNT ISZ BLCNT JMP CONTR5 JMP MASTR5

CONTR3 ISZ GRUPCT JMP BLRD5T LAC GRSTOR DAC GRUPCT JMP BLRD5M

CONTR4 ISZ GRUPCT JMP BLRD6T LAC GRSTOR DAC GRUPCT JMP BLRD6M

CONTR5 ISZ GRUPCT JMP BLRD5M LAC GRSTOR DAC GRUPCT JMP BLRD5T .EJECT
CONTR6

isz grupct
jmp blrd6m
lac grstor
dac grupct
jmp blrd6t
.END
.TITLE PPPR/LE-BA Tuan/8/1/71.

//PROGRAM FOR TRANSFER OF DATA BETWEEN DISK 6 AND PAPER
//READER AND PAPER PUNCH, TYPE 5 FOR PRINTOUT,
//8 FOR PAPER PUNCHING FROM DISK 6, 9 FOR PAPER READING
//INTO DISK 6, 1000 TO GO BACK TO CORRELATION
//PROGRAM AND 2000 TO GO BACK TO FFT PROGRAM,

.GLOBAL .AA,.AB,.AC,.AD,.AE,.AF,.AJ,.AY,.BA,.ACCEPT
.GLOBAL .AH,.AG,.AL,.AN,.AM,.AX,.CO,.FG,.FJ,.FX
.GLOBAL ACOR1,ACOR2,PPPR

.IODEV 4,3,7,5,6

.GLOBAL .FP,.FW,.FE,.FR,.FF
.DEFIN WRITED,IODEV,FMT,BUF,?Q

.JMP* .FW

.DSA Q

.JMS* .FE

.DSA BUF

.JMS* .FF

.JMP +2

.Q

.IODEV

.ENDM

.DEFIN WRITM,IODEV,FMT,COL,ROW,BUF,?C1,?C2,?P,?Q

.LAC (-ROW

.DAC C2

.JMS* .FM

.DSA Q

.DSA FMT

.LAC (-COL

.DAC C1

.JMS* .FE

.P

.DSA BUF

.ISZ P

.ISZ P /SKIP 1 VAL IN INTG; OK FOR REAL PRINTG,

.ISZ C1 /PUT IN IF SKIP & REAL PRINTG ARE NEEDED,

.JMP P-1

.JMS* .FF

.ISZ C2

.JMP P-6

.LAC (BUF

.DAC P

.JMP +4

.Q

.IODEV

.C1 0

.C2 0

.ENDM

.EJECT
,DEFSIN WRITEH, IODEV, FMT, ?O
JMS* ,FW
.OIA Q
.DSA FMT
JMS* ,FF
JMP ,+2

Q
IDB
.DEFSIN READ, IODEV, FMT, BUF, ?O
JMS* ,FI
.OIA Q
.DSA FMT
JMS* ,FE
.OIA BUF
JMS* ,FF
JMP ,+2

Q
IDB

START

JMS* ,FP
JMP PARIN5

DIN


INDEX 3, 0, PARIN5
.trans 3, 0, 0, BUF1, 256
.wait 3
.close 3
.isz TRAN1+2; JMP# DKN

DKOUT1


INDEX 3, 1, PARIN5
.trans 3, 1, 0, BUF1, 256
.wait 3
.close 3
.isz TRAN01+2; JMP# DKOUT1

DKOUT2


INDEX 3, 1, PARIN5
.trans 3, 1, 0, BUF1, 256
.wait 3
.close 3
.isz TRAN02+2; JMP# DKOUT2

//

PPPR


WRITEH 4, FMT36

PARIN5

WRITEH 4, FRT1

JMS* ACCEPT; JMP ,+3; DATS; PLACE
LAC (40); DAC WORD#; LAC (777400); DAC MWORD#

BEGIN

LAC PLACE#
SAD (1750); JMS* ACOR1
SAD (3720); JMS* ACOR2
SAD (5); JMP PRNTR
SAD (11); JMP READPR
.EJECT
READPR

SAD (10); JMP PUNCH1; JMP PARIN5

LAC (BUF1); DAC TRAN01+3; DAC TRAN02+3; JMS PR

DAC STABLK; DAC TRAN01+2; JMP READP2

PR

0; WRITE 4,FRT35

JMS; ACCEPT; JMP *,+4; DATS; STABLK; POINTS

JMS PRELIM; LAC NBC; DAC PNBC; JMS PLROU1

JMS PLROU2; JMP* PR

READP2

JMS; ACCEPT; JMP *,+3; DAT5; DSA XV#

READP3

LAC XV; DAC* BUF1P#; ISZ BUF1P

ISZ FCT#; JMP READP2

JMS OKOUT1; LAC MWORD; DAC FCT; LAC (BUF1)

DAC BUF1P; ISZ CT

JMP READP2; JMP PARIN5

DATS

4

DAT5

5

ACCEPT

0; JMS* ACCEPT

JMP *,+3

,DSA DATS

,DSA POINTS#; JMS PRELIM; JMP* ACCEPT

PRELIM

0; LAC POINTS; JMS*,AE; LAC (400); CMA; TAD (1)

DAC PNBC

SAD (0); LAC (-1); DAC NBC#; CMA; TAD (1)

DAC NB#; DAC NF#; LAC*,CO

SAD (0); JMP *,+2; JMP *,+2; LAC (400); DAC WORD

DAC WORF#; CMA; TAD (1); DAC MWORD

JMP* PRELIM

BUF1 ,BLOCK 400

PUNCH1 JMS PR

PUNCH2 JMS OKIN

PUNCH3

LAC* BUF1P; DAC XV; ISZ BUF1P

WRTED 7,FMT8,XV

ISZ FCT; JMP PUNCH3; JMS PLROU1

ISZ CT; JMP PUNCH2; JMP PARIN5

PLROU1

0; LAC MWORD; DAC FCT;

LAC (BUF1); DAC BUF1P

JMP* PLROU1

//

PLROU2

0; LAC PNBC#; SAD (0); LAC (-1); DAC CT#; SAD (-1)

JMP ROU2EN

JMS*,AE; LAC (2); ROU2EN DAC NB2#; DAC NB2C#

DAC STAHLK#

PLOTN

DAC TRAN1+2; LAC (BUF1); DAC TRAN1+3

JMP* PLROU2

PRINTR

JMS PR

LAC NBC; DAC PNBC; JMS PLROU1; JMS PLROU2

DOIT2 JMS OKIN

WRTMN 6,FRT63,20,20,6,BUF1

ISZ CT; JMP DOIT2; JMP PARIN5

//

,EJECT
<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRT1</td>
<td>ASCII &quot;(1X,10HLP5PPBPR9:)&quot;</td>
</tr>
<tr>
<td>FRMT63</td>
<td>ASCII &quot;(1X,16(1X,I6))&quot;</td>
</tr>
<tr>
<td>FMT8</td>
<td>ASCII &quot;(1X,14)&quot;</td>
</tr>
<tr>
<td>FRT35</td>
<td>ASCII &quot;(1X,13HSTABLK: #PTS:)&quot;</td>
</tr>
<tr>
<td>FMT36</td>
<td>ASCII &quot;(1X,14HPPPR &amp; OPTIONS)&quot;</td>
</tr>
</tbody>
</table>

*END*
.TITLE POINT/LE-BA TUAN/7/10/71.
/POINT -- SUBROUTINE TO PLOT A POINT
/R.J. ROBBINS -- JUNE 1969
/REVISED FOR 611:
/JIM MOONEY -- JULY 1970
/REVISED FOR XY-PLotte:
/CALLING SEQUENCE:
/CALL POINT(X,Y,N)
/X -- COORDINATE BETWEEN 0 AND +511
/Y -- COORDINATE BETWEEN -256 AND +255
/N -- 1 FOR VISIBLE, 0 FOR INVISIBLE
/WPOIN -- 0 FOR SCOPE, 1 FOR PLOTTIER,
,.GLOBL POINT,XPOINT,YPONT,WPOINT,.,DA
XPOINT
0
YP0INT
0
BUFOUT , ,BLOCK 6  /OUTPUT BUFFER
XV
0
Y
0
N
0
OFF -400
ON 377
DACLON=701522
DACTV=701521
DACLOF=701542
DAINIT=701541
DASQCL=701561
DPLG=705245
DPRDS=705072
PM=0
WPOINT 0

//
POINT
0
//
JMS* , ,DA  /GET ARGUMENTS
JMP+4
XTEMP
0  /X COORDINATE
YTEMP
0  /Y COORDINATE
NTEMP
0  /N
LAC* XTEMP; DAC XPOINT
LAC* YTEMP; DAC YPOINT
LAC* NTEMP; DAC N  /STORE X,Y & N
LAC* WPOINT; SAD (0); JMP POINTS

POINTP
LAC XPOINT
TAU (-400) /TRANSLATE COORDINATE.
DAC XV; DAC* (BUFOUT); DAC* (BUFOUT+3)
LAC(-11) /INITIALIZE WORD COUNT
DAC* (36
LAC (BUFOUT-1) /INITIALIZE DATA ADDRESS
DAC* (37
DASQCL /RESET D/A SEQUENCE GENERATOR
DAINIT /INITIALIZE CHANNEL 1 OF D/A CONVERTER
EJECT
LAC YPOINT; DAC Y
DACx (BUFOUT+1
DACx (BUFOUT+4
DAINIT /INITIALIZE CHANNEL 2 OF D/A CONVERTER
LAC N
SNA /IS N DIFFERENT FROM 0?
JMP .+3 /YES -- TURN BEAM ON
LAC ON /NO -- LEAVE BEAM OFF
SKP
LAC OFF
DACx (BUFOUT+2); DACx (BUFOUT+5); DAC N
DAINIT /INITIALIZE CHANNEL 3 OF D/A CONVERTER
NOP
NOP
DAACTV /SET ACTIVE FLIP-FLOP OF D/A CONV.
DACON /TURN CONVERTER CLOCK ON
TORS /WAIT FOR COMPLETION OF TRANSFER
AND (2); SZA; JMP ,.-3
DACLAOF /TURN CONVERTER CLOCK OFF
JMP* POINT /EXIT

POINTS
LAC YPOINT
RCL
TAD (1000)
AND (1777
XOR (140000)
DAC FILE+1
LAC XPOINT
RCL
AND (1777
XOR (100000)
DAC FILE+2
LAC N /CHECK INTENSIFICATION
SNA
JMP .+4 /NOT INTENSIFIED
LAC FILE+2
XOR (20000) /INTENSIFIED, SET BIT
DAC FILE+2
LAC (FILE
DPLG /START DISPLAY
DPRDS
AND (1 /WAIT FOR COMPLETION
SZA
JMP ,.-3 /RETURN

FILE
PM 0 /DISPLAY FILE
XX /(Y COORDINATE)
XX /(X COORDINATE)
PM 1 /HALT
-END
Figure 41.—Pre-emphasis Characteristics of the Preconditioning Unit.
Curves show 3 particular positions of a continuous control.
Figure 42.— Input part of the Preconditioning Unit. 399 are Fairchild's Op Amp 399.
Figure 43.— First stage of AGC of the Preconditioning Unit. 370 is Sylvania's AGC370, 399 is Fairchild's Op Amp 399.
Figure 44.— Second stage of AGC of the Preconditioning Unit. 741 is Op Amp 741
Figure 45.— Switch Control of the Preconditioning Unit.
Figure 46.-- Output part of the Preconditioning Unit.
DEFINITION OF SYMBOLS:

- **N**: TOTAL NUMBER OF DATA POINTS TO BE CORRELATED.
- **NB**: TOTAL NUMBER OF BLOCKS OF DATA, EACH BLOCK HAS α POINTS OF DATA.
- **W**: TOTAL NUMBER OF POINTS TO BE CORRELATED IF LESS THAN α, AND EQUAL TO α IF GREATER THAN α.
- **IT**: THIS RELATES TO τ by the relation τ = IT * W + L.
- **F**: BUFFER IN CORES FOR THE RESULTS OF CORRELATIONS.
- **P**: BUFFER IN CORES FOR FIRST FUNCTION TO BE CORRELATED.
- **Q**: BUFFER IN CORES FOR SECOND FUNCTION TO BE CORRELATED.
- **PR**: PARTIAL SUM USED IN SUBROUTINE SURA AND SUBB.
- **L**: DUMMY DELAY NUMBER (INTEGER), RELATED TO τ AND IT AS ABOVE.

Figure 47.—Detail of Correlation Subroutines.