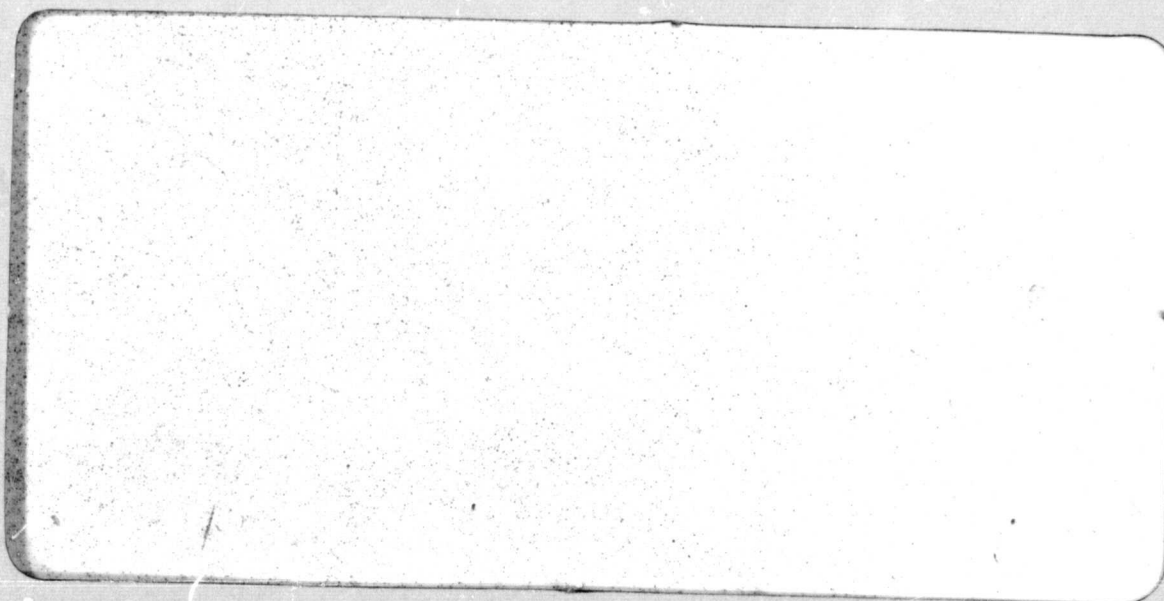


General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

ENGINEERING RESEARCH SERIES



FACILITY FORM 602

N71-303541
(ACCESSION NUMBER)

81
(PAGES)

C2-115075
(NASA CR OR TMX OR AD NUMBER)

A3
(THRU)

19
(CODE)

19
(CATEGORY)

Engineering Complex

TENNESSEE TECHNOLOGICAL UNIVERSITY
COOKEVILLE, TENNESSEE

CR-115075

FINAL PROJECT REPORT
Part D

FOURIER AND STATISTICAL ANALYSIS
AT RANDOM WAVEFORMS

by

Dennis L. Luckinbill, Project Director
Andrew Michael Forehand

Contract No. NAS 9-10385

NASA Manned Spacecraft Center
Houston, Texas

June 1971

ACKNOWLEDGEMENTS

This paper is based upon the masters thesis of the second author.

This research was partially supported by NASA Contract No. NAS 9-10385. The first author was supported by this NASA Contract. The support is gratefully acknowledged.

TABLE OF CONTENTS

	Page
LIST OF TABLES	v
LIST OF FIGURES	vi
Chapter	
I. INTRODUCTION	1
II. PROGRAM IMPLEMENTATION	7
III. MATHEMATICAL FORMULAS USED WITHIN THE PROGRAM . .	19
FILTER SIMULATION	19
FOURIER ANALYSIS	22
POWER SPECTRAL DENSITY	31
PROBABILITY MOMENTS	35
THE PROBABILITY DENSITY AND THE CHI-SQUARE TEST	37
AUTOCORRELATION FUNCTION	39
IV. EXAMPLES OF PROGRAM USAGE	42
EXAMPLE 1	42
EXAMPLE 2	52
V. ANALYSIS OF A DIGITAL FILTER	61
VI. CONCLUSIONS	73
REFERENCES	76

LIST OF TABLES

Table		Page
II-1	Definitions of Subroutines	10
II-2	Control Parameters	14
IV-1	Procedure for Digitizing Data	43
IV-2	Selected Control Parameters for Example 1 . .	47
IV-3	Selected Control Parameters for Example 2 . .	54
IV-4	Statistical Parameters for a Gaussian Probability Distribution	56
V-1	Selected Control Parameters for the Filter Problem	63

LIST OF FIGURES

Figure		Page
II-1	Computer Flow Chart	12
III-1	Original Sampled Signal	24
III-2a	Odd Numbered Samples	24
III-2b	Even Numbered Samples	24
III-3	Flow Graph for D.F.T. Reduction	27
III-4	Flow Graph for D.F.T. Reduction	27
III-5	Graph for the Fast Fourier Transform (Shuffled Input)	30
III-6	Graph for the Fast Fourier Transform (Ordered Input)	30
IV-1	Autocorrelation Function Versus Time Lags . .	50
IV-2	R.M.S. Amplitudes Versus Frequency	51
IV-3	Measured Probability Density	58
IV-4	Power Spectrum	60
V-1	Input Data for Filter (Random Gaussian Noise)	65
V-2	Filter Output Data	66
V-3	Autocorrelation Function Versus Time Lags (Random Gaussian Noise)	67
V-4	Autocorrelation Function Versus Time Lags (Filter Output)	68
V-5	R.M.S. Amplitudes Versus Frequency (Random Gaussian Noise)	70
V-6	R.M.S. Amplitudes Versus Frequency (Filter Output)	71

CHAPTER I

INTRODUCTION

Data processing is very important when analyzing physical problems. N.A.S.A. engineers contracted with Lockheed programmers to write a computer program which gives frequency information and statistical quantities for experimental data. The original N.A.S.A. program, which used the large storage facilities, was written for use on a Univac 1108 computer. The program discussed in the following material is the modified version of the original program. Since Tennessee Technological University's computer storage is limited, the amount of data which can be processed by the program has been reduced. Also, N.A.S.A.'s computer compiler was altered for their specific engineering problems. Thus, several routines had to be modified and rewritten in order to be compatible with Tennessee Technological University's IBM 360/40 compiler. The thorough investigation of the program and the N.A.S.A. program documentation [1967] lead to the development of this documentation so that other engineers can use the program with minimum effort.

The need for the analysis of random signals occurs in many complex problems of every field of study. Some examples of when this need may occur are: studying the temperatures

of unstable combustion processes, analyzing the economic potential of an average family in a small town, calculating the stress and strain of a bridge during rush hour traffic, or determining the effects of a family's income on each individual's accomplishments. It is easily seen that the data from each of the above categories will not be easily expressed by a mathematical formula. This kind of data is generally expressed by approximate mathematical methods. Many years have to be spent learning the detailed mathematical relations necessary to observe the varied qualities of this data. The same general analytic techniques will resolve the data from all major areas of study.

Over simplification of a complex process has lead many times to erroneous results. When data points are plotted, some type of a curve can generally be passed through these points. This graph will show the designer how the process varies at given points or at given times. In many reports small variations in data points are neglected, and these points are plotted as smooth curves. The loss of information often encountered by smoothing of the data curves will be cut to a minimum when the data is processed by the computer program described in the following material.

With exception of a few general tests of the data, it is not necessary to have any knowledge of the complex mathematics involved in the calculations. The first consideration is the meaning of a random signal. A signal is random if its value at one instant does not depend on the value at any

other instant. This type data is the most difficult of all data to analyze. Since the data is to be a single random signal, various assumptions must be made to obtain valid results. One assumption which should be made is that the process is ergodic. This means that the time averages for one set of data equal the averages for all other sets of data from that process. As an example of this, consider the speech analysis of the word "data." In order that this process be ergodic, every time the word is said it must be said identically to the previous statement of the word. This means that the data sample analyzed must be a good representation of every other data sample. Another equally important assumption is the stationarity of the process. Random data is stationary in the strict sense if its statistics are not varied by a shift in the independent variable's origin. This means that quantities such as the mean and standard deviation do not vary when looking at different sections of the process. Such assumptions are normally made without any depth of thought, but such assumptions must be mentioned when dealing with random processes.

The first operation performed by the computer program is to read the data. This is accomplished by reading the data from a magnetic tape or punched cards. Many times the input data has inherent noise superimposed on the true signal. Therefore, the next operation is often the application of a digital filter. The programmer must select the range of frequencies contained in the noise. This determines the

selection of a low-pass, band-pass, or high-pass filter. This filtering process is accomplished by selecting various weighting functions for the input signal which will free the data of the unwanted noise.

If frequency information is desired for a signal, the Fourier Analysis becomes an important tool. A Fourier Series is a sum of many sinusoidal functions of different magnitudes and frequencies which when added together in a series will give the original waveform. The method of obtaining the various frequency components is the most unique feature of this program. A computer algorithm devised by Cooley and Tukey [1965] is the formula applied to the data. This technique saves valuable computer time by cutting the total number of operations to a minimum. This technique is termed the "Fast Fourier Transform" and has been shown to be a major break through in the computation of frequency information.

Another important quantity needed, when frequency information is desired, is the power spectral density. This function gives the amount of power of the signal concentrated at any certain frequency. Notice that the term "power" in this case is somewhat ambiguous. As an example, consider the temperature of a flame in unstable combustion. Power in this case could be the number of B.T.U.'s released from the flame, or it could be just the square of the temperature. This term can have an engineering significance, or it can just be an abstract quantity giving only a weighted relationship with frequency. The density function is defined as the power per

unit cycle of the signal. The actual value of the function is the power within a small bandwidth about a given frequency. The traditional method of calculation involves the cosine transform of the autocorrelation function. This program bypasses the traditional method by altering the Fourier amplitudes. When the Fourier amplitudes are given, the power spectral density simply becomes the square of these coefficients. This function is recorded in a table and in a plot by an output subroutine.

Frequency information provides the specific behavior of a given function. Many times, however, general information about the data is needed. For instance, what is the average amplitude of the function over a particular time period? An example of this would be a speaker giving a lecture. If a need arises to analyze the speaker's voice, a frequency analysis might be required. If this need changed to the recording of his lecture on a tape recorder, the statistical values are more important. The operator of the tape recorder is interested in the average loudness of the lecture. It is easily seen that both frequency based and statistically based information are important, and their pertinence depends upon the specific application of this information.

In order to provide a general computer program, statistical quantities are calculated for the user's information. The expected value is the first calculation performed. This value is simply the average or mean value over a given region or a time period. A second quantity, the standard

deviation, gives numerical insight as to how the function varies. This calculation gives information concerning the average displacement from the mean value. Higher order statistics are also calculated to aid in the data reduction process.

When looking at a graph of data points, many times it is not obvious if a process has repeated itself. This is especially true when dealing with random data. A function which could relate to the programmer how well the process is repeated is called the autocorrelation function. An example of its importance is shown by the previous example of a lecture being taped on a recorder. Looking at a plot of the statistical quantities and frequency components of the man's voice would not show if he only said one sound 500 times or if he said 500 different sounds. The autocorrelation function will result in a numerical value between 0 and 1; where 1 corresponds to a perfectly correlated signal. If the man said one sound 500 times the function value would be approximately equal to 1, but if the man said 500 different sounds, the function value would be equal to 0. It can be concluded from this that if he said 250 different sounds and one sound 250 times, the function value would be $1/2$. This function is invaluable when evaluating the repeatability of a process. Each of the particular evaluations mentioned above have specific values when reducing data, and only the programmer can determine which is more useful to his application.

CHAPTER II

PROGRAM IMPLEMENTATION

Before using this program it is essential to know how to input your data, how to control the basic data processes, and how to interpret your results. Since the program is designed for general data processing, the input data may have various forms and meanings. Likewise, the results will be as varied as the input data. Also, the results required for an individual data reduction can use any or all of the available processing units. A general explanation will be given in the following material of how changing the input data and the control parameters will effect the results.

The first consideration is the method of handling the data. Data taken from real processes is generally in continuous form. Since the digital computer will handle only discrete quantities, the first operation required by the programmer is to sample the data at specific points or specific times. This may be accomplished by having an individual measure the function value at specific points or by using elaborate electronic sampling techniques. In either case a basic theorem must be used to determine where the function is to be evaluated. Shannon proved that in order to completely reconstruct a signal theorem II-1 must be applied.

Theorem II-1: The function must be sampled at least twice during the period of the highest frequency component (J. S. Tou [1959]).

When using Shannon's theorem, select the highest frequency component which is of interest in that particular application. This is not necessarily the highest frequency component contained in the data. A digital filter is provided to eliminate the unwanted higher frequency components. This will allow a slower sampling rate if the data is immediately processed by the filter. The Fourier series of a signal yields a value at a frequency of 0, and a series of numbers which correspond to frequencies of nf_0 , where $n = 1, 2, 3, \dots$ and f_0 = fundamental frequency. The fundamental frequency in a given set of data is dependent upon the number of data points provided. This dependence is shown by the formula $f_0 = \frac{1}{T}$ where T = total time involved in the data record. The total time involved is given by the sample rate multiplied by the number of samples. Since the number of samples which can be analyzed by this program is limited to 8192, it may become an advantage to use the minimum sample rate predicted by Shannon's Sampling Theorem. Now that the data has been placed in a usable form, it may be fed into the computer by punch cards or magnetic tape. This data is read by a subroutine called VREAD. The input data must have one data point followed by the corresponding time value then another data and time point, etc. Notice at this point that

the data record can be all data values, and VREAD will reformat the data such that a data value and a time value will be in series. A series of control cards tells VREAD the chosen sequence for the data (see Table II-2). The data record is now ready to be processed by the program.

The data analysis is accomplished by a series of subroutine programs. Each subroutine will be executed if and only if a main program calls it. The main program is named VIBN3. This section contains all the necessary CALL statements which are needed to execute the individual subroutines. These CALL statements are controlled by IF statements which simply compare a control parameter to a given quantity, and if they compare correctly the subroutine is then executed. Table II-1 gives the name of the subroutine and its individual function in the program.

Table II-1 shows that there are eight main sections to the program. These basic units are CARDS, VREAD, FILTER, DFILT, VIBA, PLTSTS, CONVO, and OUTF. The subroutine CARDS is always executed as the first operation. When the control parameters are read from the card reader by CARDS, they begin to guide the execution of VIBAN3. The various paths to be taken are shown in Figure II-1. This diagram illustrates the order in which each subroutine is called. The flow chart will give an idea of how each subroutine fits into the overall ensemble. The section called VIBA does the frequency analysis portion of the program. PLTSTS is a set of routines which will do a statistical analysis on the input data. The

TABLE II-1

Definitions of Subroutines

VIBAN3 is the name given to the entire group of subroutines which do the data analysis.

1. VIBN3 is the main program which controls the execution of the individual subroutine sections.
2. CARDS is a subroutine which reads and writes the control card values. These values control all options available to the program user.
3. VREAD reads and reformats the data into a full length record with alternating data points and time points. It then stores the data for further processing. VREAD reads the data in lengths of 128 data points.
- 4a. FILTER computes a set of digital filtering weights for filtering of the composite frequency data.
- b. DFILT applies the filtering weights to the data values computed in FILTER.
5. VIBA is a subroutine which controls two subroutines used in the frequency analysis. The Fourier series is computed when VIBA calls the subroutine RFORT. VIBA then converts the Fourier Series to the Power Spectral Density and then saves all this information on a magnetic disk.
 - a. RFORT is a one dimensional finite Fourier transform routine. It calls the complex transform routine FORT and makes the necessary adjustments for the complex numbers.
 - b. FORT computes the complex Fourier transform for the input data.
6. PLTSTS is a subroutine that controls three smaller subroutines used in the statistical analysis. It computes the complete statistics and the probability density. These are then plotted by a subroutine SPLOT.
 - a. SUMMIT computes the sums needed for the first, second, third, and fourth moments.
 - b. PROBL generates the probability density.

TABLE II-1

(continued)

- c. CHI computes the chi-squared statistics and goodness of fit test for the probability density found by PROBL.
 - 7. CONVO is a subroutine which computes the autocorrelation function. This is accomplished by reading the Fourier transform from a magnetic disk, squaring the transform, and taking the inverse Fourier transform. The results are then plotted by the subroutine SPLOT.
 - 8. OUTF is a subroutine which prints and plots the results of the VIBA section. Using SPLOT, this routine plots the Power Spectral Density and the R.M.S. amplitude versus frequency.
- ** SPLOT is a subroutine which scales and plots two variables which are given as input. This routine also writes the plotted values in a table for convenience.

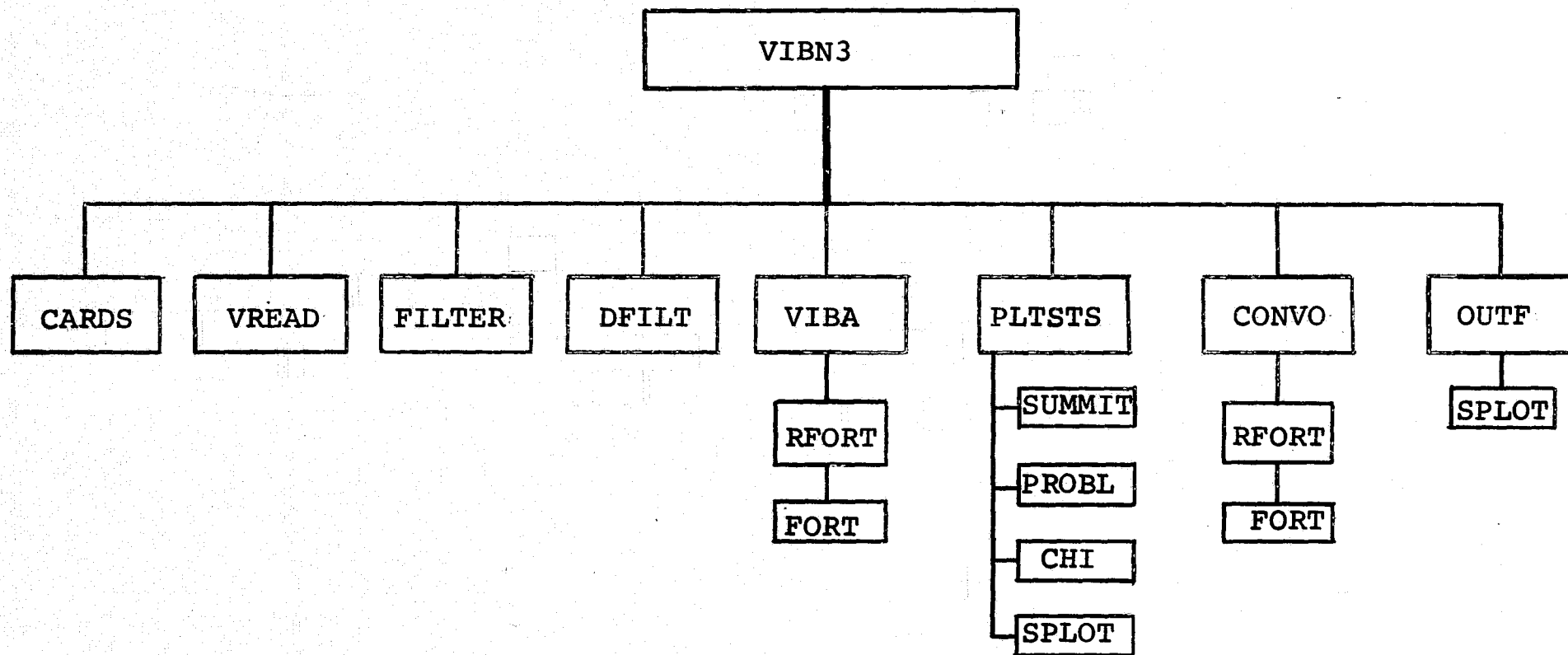


Figure II-1. Computer Flow Chart

autocorrelation function is calculated in the section called CONVO. The majority of the results is printed out by the subroutine called OUTF. At the end of the execution of OUTF the entire analysis has been accomplished.

The control parameters are probably the most important figures to be placed in the program by the user. If these values are given wrong or misinterpreted the analysis will be inaccurate. Included with these control parameters are groups of process information. An explanation of this information is given in Table II-2. As can be seen from the table, the required number of data cards for a specific input is thirteen. All cards must be present even if they are blank.

The amount of output will be dependent upon the values placed on the control cards. The maximum output will result in seven tables and five plots. The first table should contain all the control parameters discussed earlier. The next information will be a plot of the input data with each value presented in a table which follows the plot. All the calculated statistical quantities will be presented in a table which follows the input data plot. An individual plot and table of values will be given for the autocorrelation, the power spectral density, and the root mean square amplitudes. By varying the values on the first data card, the output can be changed from one or more of the above plots or tables to all of the plots and tables.

The plots of the power spectral density function and the root mean square amplitudes have a frequency axis with

TABLE II-2
Control Parameters**

Card Number	Variable Name	Identification	Format	Card Columns
1	IOP1	> 0, to plot input data	I1	1
1	IOP2	> 0, to compute and plot probability density and statistics	I1	2
1	IOP3	> 0, to compute and plot autocorrelation	I1	3
1	IOP4	> 0, to compute and plot P.S.D.	I1	4
1	IOP5	> 0, to compute and plot R.M.S. versus frequency	I1	5
1	IOP6	not used, use = 0	I1	6
1	IOP7	> 0, to plot Gaussian probability density	I1	7
1	IOP8	> 0, to plot Rayleigh probability density	I1	8
1	IOP9	> 0, to apply the digital filter	I1	9
1	IOP10	> 0, to plot P.S.D. in decibels referenced to variance	I1	10
1	IOP11	≠ 0 Fourier transform coefficients and P.S.D. will be output on Unit 12	I1	11
1	IOP12	not used, use = 0	I1	12
1	IOP13	> 0, delete the data listing from the plot	I1	13
1	IOP14	≤ 1, read input data from punch cards > 1, read data from magnetic tape Unit 10.	I1	14
1	IOP15	= 0, han (smooth) the data	I1	15
1	FC1	lower filter cut-off frequency (low pass)	F6.0	23-28
1	FC2	upper filter cut-off frequency (if band pass)	F6.0	29-34
1	FT1	lower filter terminating frequency (low pass)	F6.0	35-40
1	FT2	upper filter terminating frequency (if band pass)	F6.0	41-46
1	PERROR	desired percentage error	F6.0	47-52
1	IFORT	= 1 (used in VREAD)	I5	53-57
1	NTB	number of data points	I5	58-62

TABLE II-2
(continued)

Card Number	Variable Name	Identification	Format	Card Columns
1	NMB	number of pieces of information per time space. Note: Time space means one data value and associated time.	I5	63-67
1	NWP	word number of the data = 0, time value must be calculated = 1, data value preceeds the time value = 2, time value preceeds the data value	I5	68-72
2	ATITLE	title for input data plot	18A4	1-72
3	BTITLE	titles for probability density plots	18A4	1-72
4	CTITLE	title for autocorrelation plot	18A4	1-72
5	DTITLE	title for power spectral density plot	18A4	1-72
6	ETITLE	title for R.M.S. versus frequency plot	18A4	1-72
7	ID	any desired problem identification	12A4	1-48
8	IDATE	current date	2A4	1-8
8	ITAPE	tape number	2A4	10-17
8	IR	= 128 because VREAD reads data records in units of 128	I4	18-21
9	BW	desired bandwidth for power spectrum $BW \geq 2 \cdot$ (Fundamental Frequency)	F10.0	1-10
9	SAMPLE	sampling frequency	F10.0	11-20
9	CUTFRE	cut-off frequency to stop P.S.D. plot, IF blank CUTFRE=SAMPLE/2.0 is used	F10.0	21-30

TABLE II-2

(continued)

Card Number	Variable Name	Identification	Format	Card Columns
9	ANALOG	analog low-pass filter cut-off (highest frequency contained in the data)	F10.0	31-40
*	(note)	card 10 can be left blank and all values will be determined by the program		

10	FRQ(1)	lowest frequency for P.S.D. plots	F10.0	1-10
10	BWTH(1)	bandwidth for first frequency interval	F10.0	11-20
10	FRQ(2)	first frequency at which BWTH(1) will apply	F10.0	21-30
10	BWTH(2)	bandwidth for second frequency interval	F10.0	31-40
10	FRQ(3)	first frequency at which BWTH(2) will apply	F10.0	41-50
10	BWTH(3)	bandwidth for third frequency interval	F10.0	51-60
10	FRQ(4)	first frequency at which BWTH(3) will apply	F10.0	61-70
10	BWTH(4)	bandwidth for fourth frequency interval	F10.0	71-80
11	NTIME	number of time slices to be processed (number start and stop times) NTIME $\leq$ 16	I6	11-16
12	BTIME(1)	start time #1	F9.0	9-17
12	BTIME(2)	start time #2	F9.0	18-26
12	BTIME(3)	start time #3	F9.0	27-35
12	BTIME(4)	start time #4	F9.0	36-44
12	BTIME(5)	start time #5	F9.0	45-53
.	.	up to 16 times may be defined on cards like this.		
.	.	CARD 11 defines how many values to read. Continuation cards are 6 (F9.0),		
.	.	leave the last fields blank		

TABLE II-2
(continued)

Card Number	Variable Name	Identification	Format	Card Columns
13	ETIME(1)	stop time #1	F9.0	9-17
13	ETIME(2)	stop time #2	F9.0	18-16
	.	up to 16 times may be de-		
	.	finied use same format as		
	.	CARD 12		

** revised table from N.A.S.A. Report [1967]

discrete frequencies only. These points are multiples of the fundamental frequency. If the power spectral density is needed for a particular frequency, and that frequency occurs between two frequencies on the plot, an interpolation can be made to obtain the desired quantity. An alternate approach would be to increase or decrease the total time of the data record which would alter the fundamental frequency. By doing this the function could be found for any particular frequency. It is readily concluded that by altering the sampling rate, the total time, and the control cards almost any form of output can be found.

CHAPTER III

MATHEMATICAL FORMULAS USED WITHIN THE PROGRAM

The VIBAN3 program is a computer program which will reduce all forms of data into meaningful quantities. For many years mathematicians have been developing methods and various formulas which yield numbers that give meaning to large volumes of data. The VIBAN3 program uses many such formulas to obtain a few series of numbers which give clues to understand the complex data. The output of this program is a set of plots and tables giving information such as the frequency components and statistical quantities. Since the program must use these formulas to reduce the data, an outline of the mathematics used is presented in the remainder of this chapter.

FILTER SIMULATION

The first step in the resolution of the data is to filter out the unwanted portions. This is accomplished by a weighting function. The number of weights required is calculated from the relation below by solving for N.

$$N(\Delta t)(f_{t_1} - f_{c_1}) \geq 2 \quad \text{for 1.0\% accuracy}$$

$$N(\Delta t)(f_{t_1} - f_{c_1}) \geq 3 \quad \text{for 0.5\% accuracy}$$

where: Δt = time increment between samples

f_{t_1} = terminating frequency of the filtered data

f_{c_1} = cut-off frequency for high end of band

N = number of weights necessary

(Note: The difference in f_{t_1} and f_{c_1} determines the slope of the filter roll-off.)

A maximum value of $N = 600$ is allowed by program limitations.

The N filter weights are computed for a low-pass filter with

a cut-off frequency f_{c_1} . The weights in the time domain,

$h_1(n\Delta t)$, are

$$h_1(n\Delta t) = \frac{\pi}{2n\Delta t} \left[\frac{\sin(W_{t_1} n\Delta t) + \sin(W_{c_1} n\Delta t)}{\pi^2 - (W_{t_1} - W_{c_1})^2 (n\Delta t)^2} \right]$$

where: $W_{c_1} = 2\pi f_{c_1}$

$W_{t_1} = 2\pi f_{t_1}$

If a band-pass filter is required; that is $f_{c_2} \neq 0$, N filter weights are also computed for a low-pass filter with a cut-off

frequency of f_{c_2} . The weights in the time domain, $h_2(n\Delta t)$,

are

$$h_2(n\Delta t) = \frac{\pi}{2n\Delta t} \left[\frac{\sin(W_{t_2} n\Delta t) + \sin(W_{c_2} n\Delta t)}{\pi^2 - (W_{t_2} - W_{c_2})^2 (n\Delta t)^2} \right]$$

where: $W_{t_2} = 2\pi f_{t_2}$

$W_{c_2} = 2\pi f_{c_2}$

f_{t_2} = minimum frequency of the data

f_{c_2} = cut-off frequency of low-end of band

Now $2N + 1$ values of h are produced by taking $h(-n\Delta t) = h(n\Delta t)$ and $h(0) = (f_{c_1} + f_{t_2})$. Then the weights are normalized by the sum of all $2N + 1$ values; that is,

$$h(n\Delta t) = \frac{h(n\Delta t)}{\sum_{j=-N}^N h(j\Delta t)} \quad n = 1, 2, 3, \dots, 2N + 1$$

If $f_{c_2} \neq 0$, the band-pass filter weights are computed by subtracting the two sets of weights for low-pass filters with cutoff frequencies f_{c_1} and f_{c_2} . The filter bandwidth will be $(f_{c_1} - f_{c_2})$.

$$h(n\Delta t) = h_1(n\Delta t) - h_2(n\Delta t) \quad n = 1, 2, 3, \dots, 2N+1$$

The weights are applied by the equation:

$$d_i = \sum_{n=-N}^N h(n\Delta t) X_{n+i}$$

where: X_{n+i} = input data before weights are applied

d_i = adjusted data value

These techniques are applied by two subroutines called FILTER and DFILT.

FOURIER ANALYSIS

The frequency analysis is the most unique feature of this program. Since digital analysis is to be used, the continuous data must be digitized or sampled. This process is one operation which the program user must perform upon his data. The sampled data will completely represent the continuous data, if the data is sampled at least twice during the period of the highest frequency component. In the sub-routines RFORT and FORT, the Fourier analysis is applied to the filtered input data. Since digital techniques are to be used, the standard Fourier transform must be converted to digital form. This form is called the discrete Fourier transform (DFT). The discrete Fourier transform is totally analogous to the continuous Fourier transform. The discrete Fourier transform is given by the following equation.

$$A_r = \sum_{k=0}^{N-1} X_k \exp (-2\pi jrk/N)$$

where: A_r = the DFT coefficient

X_k = sampled data points

$r = 0, 1, 2, \dots, N-1$

$j = \sqrt{-1}$

(Note: The values of X_k can be complex, and the values of A_r are almost always complex.)

The inverse discrete Fourier transform is used for reconstructing the data if the Fourier coefficients are known. This

reconstruction is called the Fourier synthesis and is given in the following equation:

$$X_{\ell} = \frac{1}{N} \sum_{r=0}^{N-1} A_r \exp (2\pi j r \ell / N)$$

where: $\ell = 0, 1, 2, \dots, N-1$

Both forms of the above equations are programmed such that either can be used according to the desired result.

The fast Fourier transform (FFT) is a method for efficiently computing the discrete Fourier transform of a time series (discrete data samples). The efficiency of this method is such that many problems can now be obtained more economically than in the past. It may be useful to point out that the FFT not only reduces the calculation time, but it reduces round off error since the number of required calculations is cut substantially.

Suppose a time series has N samples such as X_k as shown in Figure III-1 (G-AE Subcommittee [1967]). If this series is broken into two separate series such as Y_k and Z_k , they would appear as shown in Figures III-2a and III-2b (G-AE Subcommittee [1967]). Y_k is composed of the even numbered points ($X_0, X_2, X_4, \dots$), and Z_k is composed of the odd numbered points ($X_1, X_3, X_5, \dots$). These functions can be written as

$$\begin{aligned} Y_k &= X_{2k} & k &= 0, 1, 2, \dots, \frac{N}{2} - 1 \\ Z_k &= X_{2k+1} \end{aligned}$$

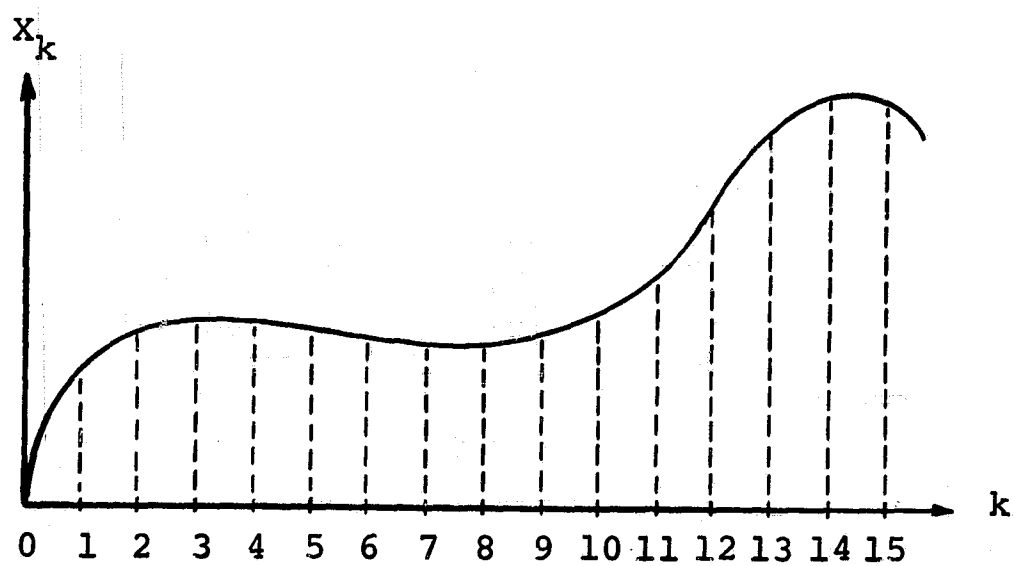


Figure III-1. Original Sampled Signal

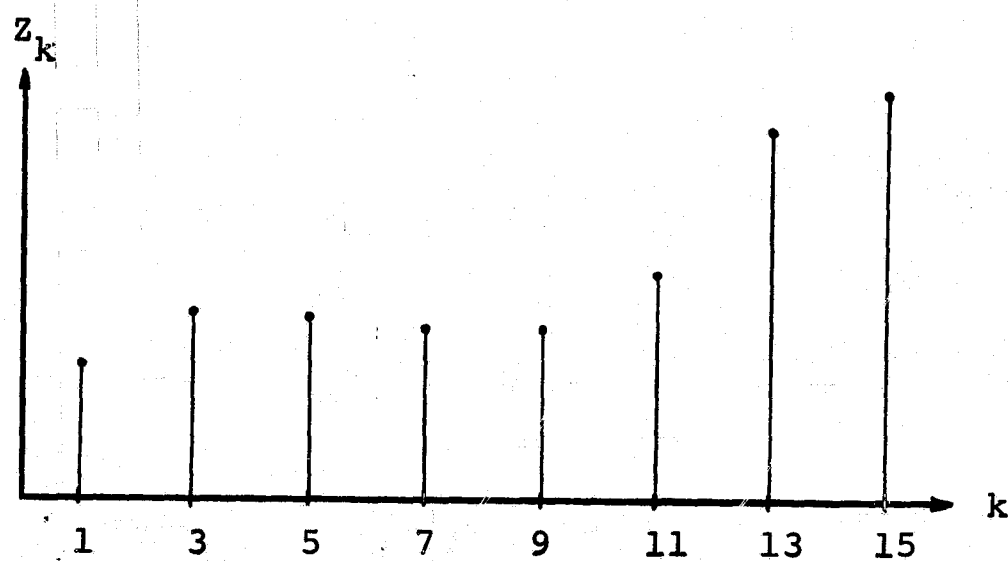


Figure III-2a. Odd Numbered Samples

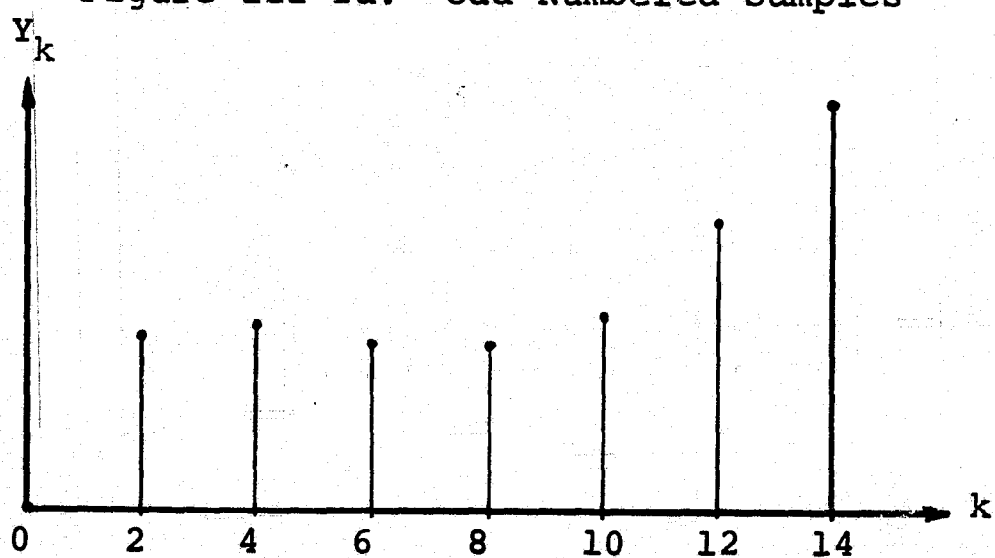


Figure III-2b. Even Numbered Samples

Since Y_k and Z_k are sequence of $\frac{N}{2}$ points each, they each have a discrete Fourier transform. Their transform is given by

$$B_r = \sum_{m=0}^{\frac{N}{2}-1} Y_m \exp [-2\pi jrm/(N/2)]$$

$$C_r = \sum_{m=0}^{\frac{N}{2}-1} Z_m \exp [-2\pi jrm/(N/2)]$$

where: $r = 0, 1, 2, \dots, (\frac{N}{2} - 1)$

The DFT of the sum of two functions is the sum of the DFT of the two functions. Therefore, the discrete Fourier transform of the original data is given in the following equation in terms of odd and even numbered points.

$$A_r = \sum_{k=0}^{\frac{N}{2}-1} [Y_k \exp (-2\pi jr/N)(2k) + Z_k \exp (-2\pi jr/N)(2k+1)]$$

where: $r = 0, 1, 2, \dots, \frac{N}{2}$

or

$$A_r = \sum_{k=0}^{\frac{N}{2}-1} [Y_k \exp (-4\pi jrk/N) + \exp (-2\pi jr/N) \sum_{k=0}^{\frac{N}{2}-1} [Z_k \exp (-4\pi jrk/N)]]$$

by substitution

$$A_r = B_r + \exp (-2\pi jr/N) C_r \quad \text{where } 0 \leq r \leq \frac{N}{2}$$

When values of r become greater than $\frac{N}{2}$, the DFT of B_r and C_r repeat periodically the values taken on when $r < \frac{N}{2}$. Therefore, a substitution of $r + \frac{N}{2}$ for r will yield the following results.

$$A_{r+\frac{N}{2}} = B_r + \exp [-2\pi j(r + \frac{N}{2})/N]C_r$$

where: $0 \leq r < \frac{N}{2}$

which reduces to

$$A_{r+\frac{N}{2}} = B_r - \exp (-2\pi jr/N)C_r$$

where: $0 \leq r < \frac{N}{2}$

The discrete Fourier transform for N points is given by

$$A_r = B_r + \exp (-2\pi jr/N)C_r$$

$$A_{r+\frac{N}{2}} = B_r - \exp (-2\pi jr/N)C_r$$

where: $0 \leq r < \frac{N}{2}$

The above property allows the program to calculate only A_r and not the remaining $\frac{N}{2}$ points. This reduces the output to only $\frac{N}{2}$ complex numbers for a transform of N points.

The following signal flow chart in Figure III-3 (G-AE Subcommittee [1967]) shows the method of solving for the Fourier coefficients for eight input data points. The signal flow chart shows that the outputs of the discrete Fourier transform sections are either B_r or C_r . These coefficients are then added as the equations indicate. The coefficient A_0

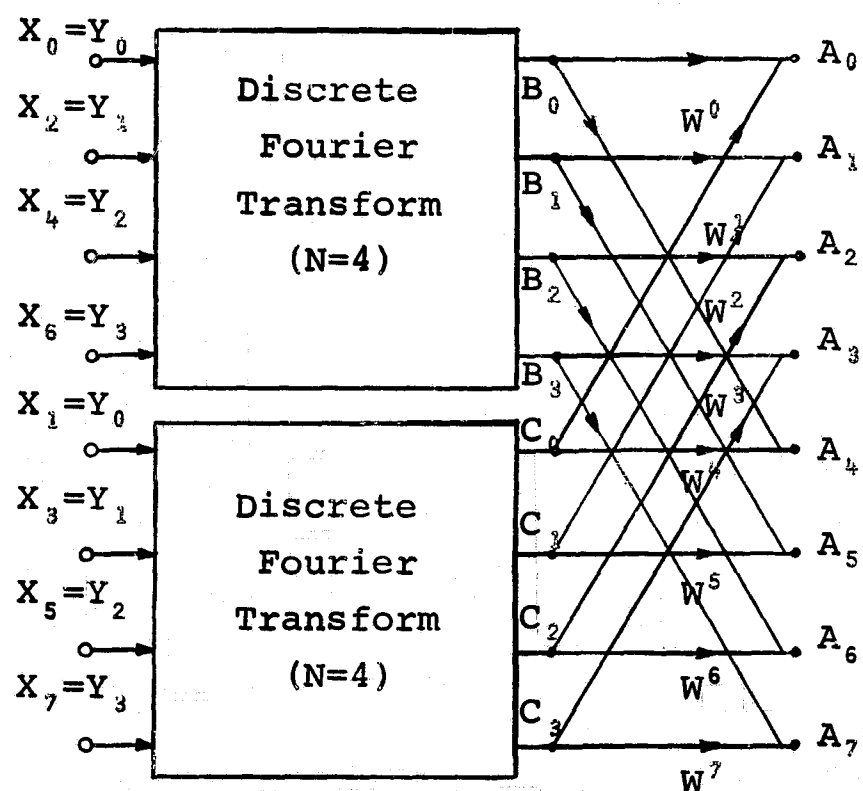


Figure III-3. Flow Graph for D.F.T. Reduction

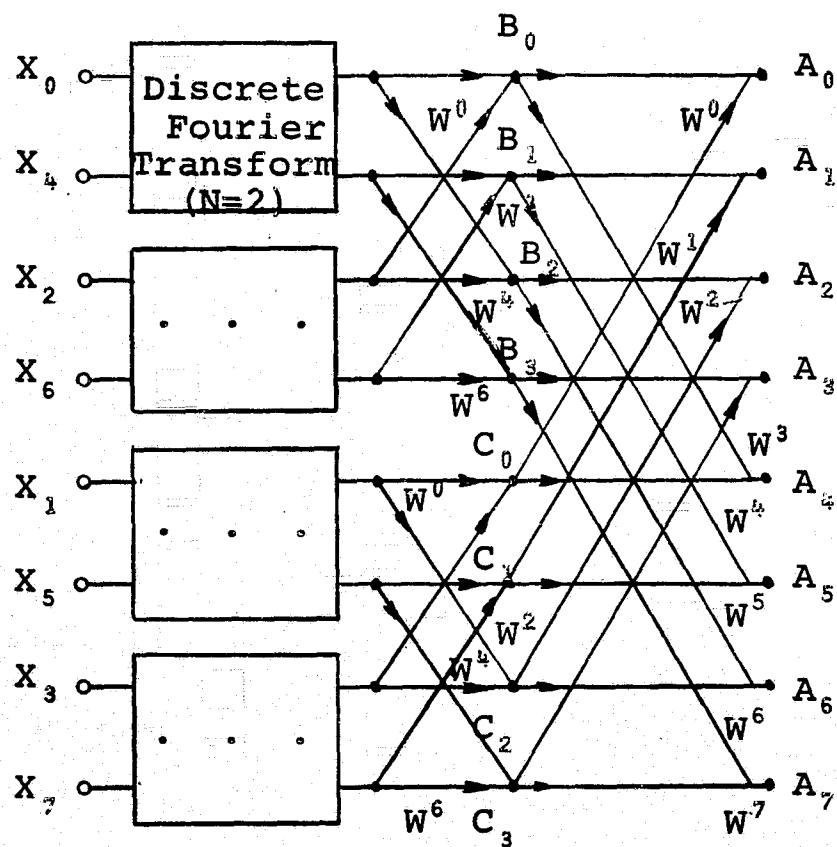


Figure III-4. Flow Graph for D.F.T. Reduction

is composed of $B_0 + \exp(-2\pi j[0]/N)C_0$. This reduces to $A_0 = B_0 + C_0$ which would be the first term of the Fourier coefficient. For simplicity of notation $\exp(-2\pi j/N)r$ will be denoted as W^r . With this simplification, an example would be the coefficient A_7 is equal $B_3 + W^7C_3$.

The next step in the fast Fourier transform is the reduction of each of the two DFT sections. Since the original N point data set was divided into two $\frac{N}{2}$ point sets, it is easily seen that each of these smaller sets can be subdivided into two $\frac{N}{4}$ point sets. This applies theorem III-1.

Theorem III-1. The DFT of the sum of two functions is the sum of the DFT of the two functions.

The signal flow graph shown in figure III-4 (G-AE Subcommittee [1967]) illustrates this principle.

The fast Fourier transform continues this process until there are no groups of data points left to be broken down. At this point theorem III-2 is applied.

Theorem III-2. The discrete Fourier transform of a one-point function is simply the function value itself.

With this version the length of the input data must be some integer power of 2. The program accomplishes this by appending zeroes at the end of the data if it is not of proper length. There exist other forms of the FFT, but this program uses the base 2 algorithm. For an example of the complete Fast Fourier Transform, a data set of 8 points, $N = 2^3 = 8$,

is Fourier transformed in the signal flow graph shown in Figure III-5 (G-AE Subcommittee [1967]). It is easily seen that all operations have been reduced to complex additions and complex multiplications. The number of multiplications and additions per step (breakdown) is equal to N , and the number of steps (breakdowns) taken is equal to $N \log_2 N$. This is a substantial reduction from the DFT analysis which requires N^2 operations.

A small problem has occurred in the order of the input data. Due to the method of breaking the points down, the data has been shuffled. Certain variations can be made in the signal flow graphs which will result in proper orientation of the input-output information. If all the nodes on the horizontal level with A_1 in Figure III-5 are interchanged with all the horizontal nodes of A_4 , and all the nodes on the level of A_3 are interchanged with A_6 , then the flow graph is altered as shown in Figure III-6 (G-AE Subcommittee [1967]). This flow graph is Cooley's [1965] original description of the algorithm. There are certain other variations of the signal flow graph which result in better input-output forms, but these variations require larger memory units; therefore, they were not considered in this evaluation.

The purpose of the fast Fourier transform is the economical evaluation of the discrete Fourier transform. As was shown earlier the calculations are cut from N^2 to $N \log_2 N$ operations where N is the number of data points. This reduction in the number of calculations results in the saving of

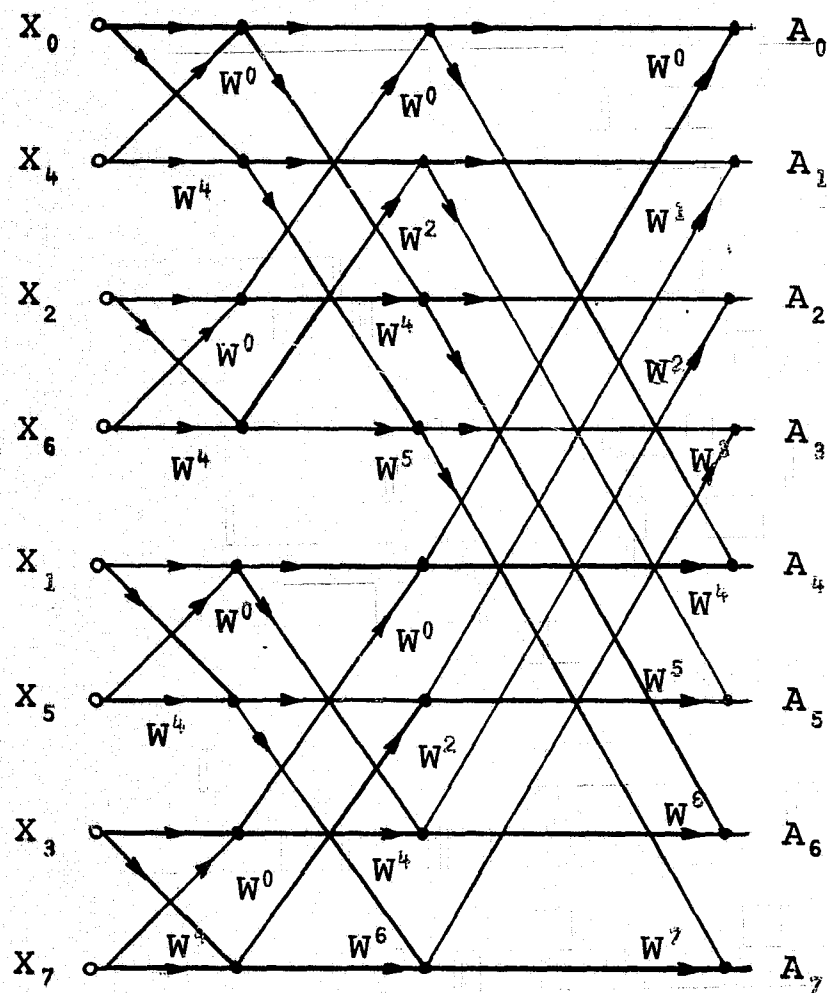


Figure III-5. Graph for the Fast Fourier Transform (Shuffled Input)

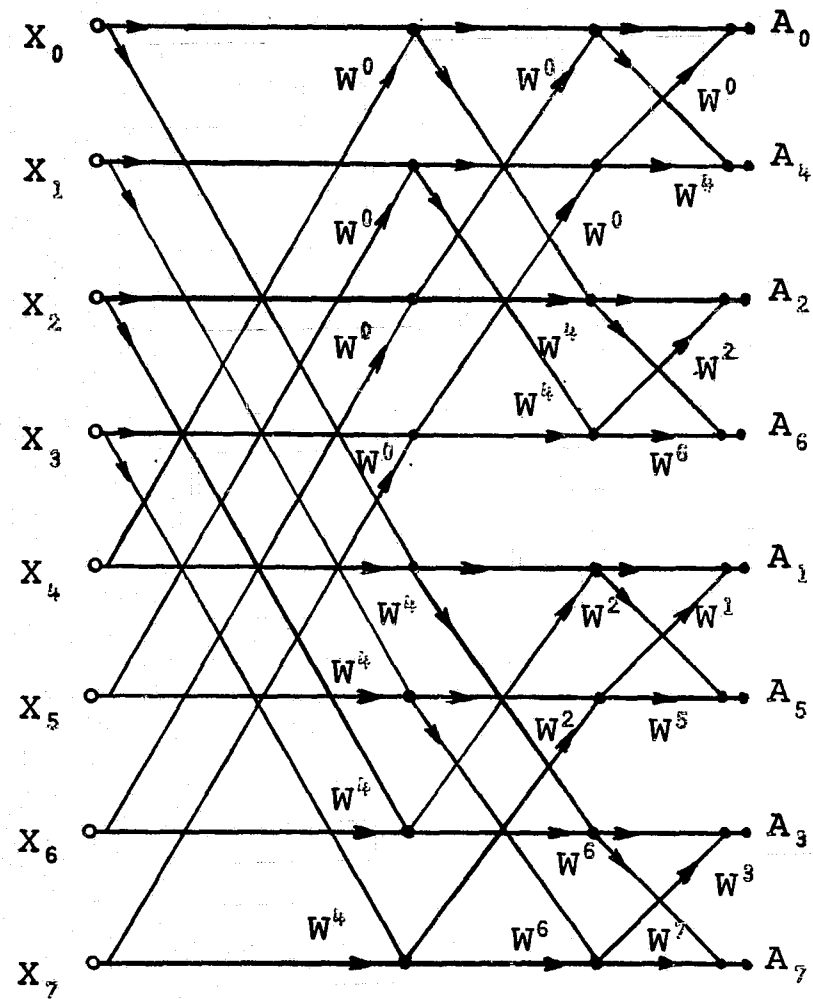


Figure III-6. Graph for the Fast Fourier Transform (Ordered Input)

expensive computer time. The G-AE Subcommittee report [1967] compared the DFT and the FFT by programming them on the IBM 7094 computer. The program was to evaluate 8192 data points, $N = 2^{13} = 8192$. It was reported that the DFT took approximately half of an hour to compute, whereas, the FFT took about five minutes. The chart below shows the ratio of the standard evaluation to the FFT evaluation.

N points	$N^2/N \log_2 N$
$2^5 = 32$	6.4
$2^6 = 64$	10.6
$2^8 = 256$	32.0
$2^{10} = 1024$	102.4
$2^{12} = 4096$	341.0
$2^{13} = 8192$	630.0

Since the number of operations have decreased, the round-off error becomes smaller. The ratio of the resulting errors is simply $\frac{\epsilon N^2}{\epsilon N \log_2 N}$. With a quick reference to the evaluation chart, one can readily see why the error is much smaller. The fast Fourier transform is well worth the effort spent in its derivation.

POWER SPECTRAL DENSITY

The power spectral density function is defined as the power per unit cycle in a signal. This means the power in a frequency band about a discrete frequency f_i . The standard method of calculating the power spectra is the use of the cosine transform of the autocorrelation function. A "raw"

estimate, $G_x(f)$, of a true power spectral density function is defined for an arbitrary f in the range $0 \leq f \leq f_c$ by

$$G_x(f) = 2h[\hat{R}_0 + 2 \sum_{r=1}^{m-1} \hat{R}_r \cos\left(\frac{\pi r f}{f_c}\right) + \hat{R}_m \cos\left(\frac{\pi m f}{f_c}\right)]$$

where: h = interval between samples

$\hat{R}_r$ = estimate of autocorrelation function at lag r

m = maximum lag number

$f_c = (1/2h)$ is the cutoff frequency

This complicated equation is normally used since the Fourier transform is very hard to evaluate. However, this computer program evaluates the Fourier transform easily. The power spectral density is now determined by taking the square of the Fourier coefficients and then dividing it by the bandwidth, BW.

$$P_{f_i}(BW) = \int_{f=f_i - \frac{BW}{2}}^{f_i + \frac{BW}{2}} A^2(f)$$

where: $P_{f_i}(BW)$ = power in band of BW centered at f_i

$A^2(f)$ = squared Fourier amplitude for frequency f

$$G_i = \sum_{i=0}^m \frac{P_{f_i}(BW)}{BW}$$

where: BW = bandwidth

m = maximum lag number

It is also noted that the technique of hamming or hanning is used to smooth the power spectral density function. The smoothing technique used in this program is given by:

$$\hat{G}_0 = \frac{1}{2}(G_0 + G_1)$$

$$\hat{G}_k = \frac{1}{4}G_{k-1} + \frac{1}{2}G_k + \frac{1}{4}G_{k+1}$$

$$\hat{G}_m = \frac{1}{2}(G_{m-1} + G_m)$$

where: $k = 0, 1, 2, \dots, m-1$

G_k = raw power spectral density function

m = maximum lag number

One helpful result of the power spectral density is the ability to plot the root-mean-square amplitude versus frequency. This plot will give us the information as to what the main frequency components are. The following equation is used in determining the values for such a plot.

$$AMP_k = \sqrt{\hat{G}_{k,x}} BW$$

where: $k = 0, 1, 2, \dots, m$

$\hat{G}_{k,x}$ = power spectral density function

BW = bandwidth

m = maximum lag number

This plot is most valuable when making a frequency analysis of a particular signal.

References have been made to bandwidth and maximum

lag numbers. Some explanation is needed about these various quantities. The frequencies which can be analyzed by the Fourier analysis are totally specified by the length of the data values and the sampling rate. These values are selected before this program is used since the program accepts only digital data. An external program or machine must convert the analog data into digitized quantities. The maximum frequency which can be represented is limited to $\frac{2}{\Delta t}$ where Δt is the time between samples. The lowest frequency or fundamental frequency which can be recognized is equal to $\frac{1}{T}$ where T is the total time interval for all the data points. These quantities must be known to determine the bandwidth of the power spectral density function. The narrowest bandwidth which can be used is

$$BW_n = 2f_0$$

where: f_0 = the fundamental frequency.

Values of BW can be varied but must be some multiple of the narrowest possible bandwidth. The program will adjust the bandwidth such that the power spectral density can be given in 4096 points or less. The maximum lag number, m , is the number of points used in the power spectral density. This value is varied between 0 and 4096 and should be calculated by the following equation.

$$m = \frac{2f_c}{BW}$$

where: f_c = maximum frequency analyzed by the power spectral density

Each of these values are dependent on the length of the data segment and the sample rate of the digital converter.

PROBABILITY MOMENTS

If information dealing with time averages and other functions of time are desired, statistical quantities are used. Such quantities, which are termed probability moments, are calculated in this program. These moments tell how the process varies as a function of time.

The first moment is called the expected value. This quantity gives the average or mean value of the function and is given in the following equation.

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$$

where: $\bar{X}$ = average value

N = number of data points

X_i = raw data values

Once the average value has been found, the average variation away from the mean becomes a meaningful quantity. The second moment will tell us this information and is termed the variance. This quantity is given by

$$S^2 = \frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})^2$$

The form most often seen is the positive square root of the above equation. The positive square root is called the standard deviation and is given in the following equation.

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (X_i - \bar{X})^2}$$

This equation has been altered for the computer analysis. The altered equation used in the program is given by

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^N [X_i^2 - N\bar{X}^2]}$$

The third moment called the skewness coefficient is found by the following equation.

$$SK = \frac{1}{NS^3} \sum_{i=1}^N [X_i - \bar{X}]^3$$

This equation can be expanded into the equation below which yields computational advantages for computer calculations.

$$SK = \frac{\sum_{i=1}^N X_i^3}{NS^3} - 3\bar{X} \frac{\sum_{i=1}^N [X_i^2 + 2N\bar{X}^3]}{NS^3}$$

The fourth moment called the Kurtosis coefficient is determined by the following equation

$$K = \frac{1}{NS^4} \sum_{i=1}^N [X_i - \bar{X}]^4$$

Again for computational advantages, the above equation is

expanded into the form shown below

$$K = \sum_{i=1}^N X_i^4 - 4\bar{X} \sum_{i=1}^N X_i^3 + 6\bar{X}^2 \sum_{i=1}^N X_i^2 - 3N\bar{X}^4$$

The quantities listed above reveal detailed information about the variations of the data. A small value for the variance coefficient would tell us that the data was not too oscillatory, but a large value for this quantity would lead to a conclusion that the data varies rapidly. Each of the remaining coefficients will lead to similar conclusions.

THE PROBABILITY DENSITY AND THE CHI-SQUARE TEST

The chi-square goodness-of-fit test compares the probability density calculated for the given data values to a standard Rayleigh and Gaussian density. This comparison is made so that the program user will know if his process is nearer a Gaussian type process or a Rayleigh process. The first calculation needed is to compute the probability density for the given data. The program approximates $f(x)$, the probability density, as the number of observations and the percentage of data in each of K class intervals. The number of class intervals K is chosen such that $K = 3 \log N$, where N equals the number of data points and K truncates the value to an integer. The Gaussian (normal) probability density function is calculated using the following equation.

$$\phi(X_i) = \frac{\Delta X}{S\sqrt{2\pi}} e^{[-X_i^2/2S^2]}$$

where: $i = 1, 2, 3, \dots, K$

S = standard deviation

X_i = midpoint of i^{th} class interval

ΔX = class interval width, $X_{i+1} - X_i$

K = number of class intervals

The Rayleigh density function is computed by the following equation.

$$C(X_i) = \frac{\Delta Y}{2Z^2} e^{[-Y_i^2/2Z^2]}$$

where: $i = 1, 2, 3, \dots, K$

K = number of class intervals

ΔY = width of class interval $Y_{i+1} - Y_i$

Y_i = midpoint of i^{th} class interval

Z^2 = variance estimator, $\frac{1}{N} \sum_{i=1}^N X_i^2$

N = number of data values

X_i = data values

The chi-square goodness-of-fit test can now be applied. By placing the calculated density function and one of the theoretical density functions into the test, it can be determined if the data is a close fit to the theoretical density. This is accomplished in the following formula.

$$\chi^2 = \sum_{i=1}^K \frac{[P(X_i) - f(X_i)]^2}{f(X_i)}$$

where: K = number of class intervals

X_i = midpoint of i^{th} class interval

$P(X_i)$ = function fitted at X_i as Gaussian or Rayleigh probability density curves

$f(X_i)$ = probability density of data values

The region of acceptance is

$$X^2 \leq \chi^2(n, \alpha)$$

where n = number of degrees of freedom

α = percent confidence level

$\chi^2(n, \alpha)$ is read in as tabulated values by the subroutine CHI. With use of all the formulas mentioned, a large amount of engineering information is determined about a set of data values.

AUTOCORRELATION FUNCTION

The autocorrelation function produces numerical values which relate the repeatability of the process. The autocorrelation is defined as the average lagged products of the data values and is calculated for a displacement rh by the formula shown below.

$$\hat{R}_r = \hat{R}_x(rh) = \frac{1}{N-r} \sum_{n=1}^{N-r} X_n X_{n+r}$$

and

$$r = 0, 1, 2, \dots, m$$

where: r = the lag number
 m = the maximum lag number
 X_n = the n^{th} data value

X_{n+r} = the $(n+r)$ data value

N = the number of data values

The above equation shows that to find one lagged product, $(N-r)$ products and summations must be performed. For large values of N it can be easily seen that many calculations must be made. In an effort to by-pass the many calculations involved in such an evaluation, mathematicians have applied the convolution theorem to the formula. This theorem is shown in the equation below.

$$\sum_{j=0}^{N-1} X_j X_{j+l} = N \sum_{k=0}^{N-1} A_k e^{[\frac{2\pi j l k}{N}]}$$

where: A_k = the Fourier transform of X_k

$j = \sqrt{-1}$

l = lag number

To find each R_r the factor of N can be removed and replaced by $\frac{1}{N-r}$. The values of A_k^2 are simply the power spectral density values. The autocorrelation function simply becomes the inverse Fourier transform of the power spectral density function. The formula used in the program to determine the autocorrelation is given below.

$$R_r = Z_r [NN/(N-r)]$$

where: $r = 0, 1, 2, 3, \dots, MP-1$

Z_r is the Fourier inverse of the r^{th} squared Fourier coefficient of the data

NN = the first power of two greater than or equal to
(N+MP)

N = number of data points

MP = number of autocorrelation lags

The autocorrelation function is normalized by dividing by R_0 . This function is maximum when the shift is equal to zero, therefore the normalized function value will be $0 \leq R_r \leq 1$. The meaning of this is if $R_r = 1$, the data repeated itself exactly after time equal to r . Likewise if $R_r = 0$, it can be concluded that the data is totally uncorrelated after time equal to r . This function gives valuable information when dealing with complex processes.

CHAPTER IV

EXAMPLES OF PROGRAM USAGE

In order to make this documentation more effective, two examples for which the results can be analytically verified will be considered. The first example will be the sum of two sine waves of different frequencies. The second example will be the analysis of a square wave. Hopefully, these two examples will provide some basis for setting up a similar problem and interpreting its result.

EXAMPLE 1

The mathematical representation of the sum of two sine waves of different frequencies is:

$$X(t) = 75.0 \sin 2\pi[25]t + 75.0 \sin 2\pi[100]t$$

Obviously, the expression means $X(t)$ is composed of a sine wave with a frequency of 25 cycles per second and an amplitude of 75 units plus a sine wave with a frequency of 100 cycles per second and an amplitude of 75 units. The first step in analyzing $X(t)$ is to digitize it, which simply means to evaluate the function at a series of specified intervals. Since the signal $X(t)$ has not been digitized, the sampling frequency and number of data points are arbitrary. Table IV-1

TABLE IV-1

Procedure for Digitizing Data

1. Decide upon a fundamental frequency which will give the desired plot.
2. The fundamental frequency determines the total amount of time for the data.

$$\text{Total Time} = 1/f_{\text{zero}}$$

where: f_{zero} = fundamental frequency

3. Apply Shannon's Sampling Theorem to find the minimum sampling frequency.

$$f_s = 2f_h$$

where: f_h = highest frequency in data

4. Convert the determined sampling frequency to a sampling period.

$$T_s = 1/f_s$$

where: T_s = sampling period (seconds)

f_s = sampling frequency (c.p.s.)

5. Determine the number of data points to satisfy the values above.

$$\# \text{ samples} = \frac{\text{Total Time}}{T_s}$$

6. Adjust the number of samples to equal the next higher power of 2.
7. Using the adjusted # of samples, recompute the new sampling period.

$$T_s = \frac{\text{Total Time}}{\# \text{ samples}}$$

8. Convert the sampling period into new sampling frequency.

$$f_s = 1/T_s$$

shows a method which can be used to select these variables. The values calculated from this method can be used to fill out the control cards.

Step 1: This step involves selecting a fundamental frequency. The frequency axes for the power spectral density function and root mean square amplitudes are composed of integer multiples of the fundamental frequency. If particular frequencies are of interest, then select the fundamental frequency to be an even divisor of those particular frequencies. The frequencies 25 and 100 are important in this example, therefore, a fundamental frequency of 5 cycles per second is selected.

Step 2: The fundamental frequency, f_{zero} , determines the total time span of the data and is found by:

$$\text{Total Time} = \frac{1}{f_{\text{zero}}}$$

$$\text{Total Time} = 0.2 \text{ second}$$

Step 3: Shannon's sampling theorem states that the sampling frequency must be twice the highest frequency component contained in the data.

$$f_s = 2f_h \quad \text{but } f_h = 100$$

$$f_s = 200 \text{ samples/second}$$

To obtain a more thorough set of data points, increase the sampling frequency.

Let $f_s = 1000$ samples/second

Step 4: The sampling period is defined to be the inverse of the sampling frequency.

$$T_s = \frac{1}{f_s}$$

$$T_s = 0.001 \text{ second/sample}$$

Step 5: The number of data points to be evaluated is determined by the following formula.

$$\# \text{ samples} = \frac{\text{Total Time}}{T_s}$$

$$\# \text{ samples} = 200 \text{ samples}$$

Step 6: The number of data points should be equal to some integer power of 2, therefore, the number of samples should be adjusted to agree with this.

Let $\# \text{ samples} = 2^8 = 256$

Step 7: Since step 6 changed the sampling period, its value must be recomputed.

$$T_s = \frac{\text{Total Time}}{\# \text{ samples}}$$

$$T_s = 0.0007812 \text{ second/sample}$$

Step 8: The sampling period determines the sampling frequency by the following formula.

$$f_s = \frac{1}{T_s}$$

$$f_s = 1280.08 \text{ samples/second}$$

This procedure provides all the necessary information to process the data. It can be easily seen from step 7 that the function $X(t)$ is to be evaluated every 0.0007812 seconds. This evaluation was made by a Fortran computer program which was placed in VIBAN3 to provide the program with direct data. The control parameters must now be defined to tell the program what to do with the data (see Table II-2 to fill out Table IV-2). These parameters are selected as indicated in Table IV-2 and placed on punch cards. The control cards are then placed between the last two delimiter cards. The program can now be executed.

The output of the program will be exactly as requested by the control parameters. This will consist of a plot of the data, measured probability density, autocorrelation function, power spectral density, and root mean square amplitude. Because some of these plots are quite lengthy, only the most important pieces of information are revealed in this report. This will include a portion of the autocorrelation plot and the root mean square plot. Some of the other examples will present other pieces of information because they are more meaningful to those examples. It is noted at this point that these plots are not from the computer but are condensed hand-drawn versions of the original computer plots.

TABLE IV-2

Selected Control Parameters for Example 1

Card 1

Name	Value	Reason for Its Selection
IOP1	5	Needed the data values plotted
IOP2	5	Needed the statistical analysis
IOP3	5	Needed to see where data repeated itself
IOP4	5	Needed to see where the power is concentrated
IOP5	5	Needed R.M.S. values at particular frequencies
IOP6	0	Not used in program anymore
IOP7	0	Was not interested in the probability density
IOP8	0	Was not interested in the probability density
IOP9	0	Do not apply the filter
IOP10	0	Was not interested in logarithmic P.S.D. plot
IOP11	0	No need to output values on Unit 12
IOP12	0	Not used in program anymore
IOP13	5	Did not want a listing of the data after the plot
IOP14	0	Read data from punch cards
IOP15	5	Do not han (smooth) the data (sine waves are smooth)
FC1	Blank	Leave blank because the filter is not used
FC2	Blank	Leave blank because the filter is not used
FT1	Blank	Leave blank because the filter is not used
FT2	Blank	Leave blank because the filter is not used
PERROR	0.0	Wanted 0% error
IFORT	1	Needed to organize the data values
NTB	256	Number of data values used in this example
NMB	2	Amount of information per time space
NWP	2	The data value followed the time value

Card 2

ATITLE $X(t) = 75 \sin 2\pi(25)t + 75 \sin 2\pi(100)t$

Card 3

BTITLE $X(t) = 75 \sin 2\pi(25)t + 75 \sin 2\pi(100)t$

TABLE IV-2 (continued)

Name	Value	Reason for Its Selection
<u>Card 4</u>		
CTITLE	$X(t) = 75 \sin 2\pi(25)t + 75 \sin 2\pi(100)t$	
<u>Card 5</u>		
DTITLE	$X(t) = 75 \sin 2\pi(25)t + 75 \sin 2\pi(100)t$	
<u>Card 6</u>		
ETITLE	$X(t) = 75 \sin 2\pi(25)t + 75 \sin 2\pi(100)t$	
<u>Card 7</u>		
ID	Mike Forehand --- Example 1	
<u>Card 8</u>		
IDATE	4-29-71	The date which the program ran
ITAPE	No Tape	The tape reader was not used
IR	128	Because VREAD reads data in increments of 128 sections
<u>Card 9</u>		
BW	10.0	$BW \geq 2.0$ (fundamental frequency)
SAMPLE	1280.08	Sampling Frequency
CUTFRE	Blank	The program will find the maximum frequency for the P.S.D. plot
ANALOG	Blank	Not needed in this evaluation
<u>Card 10</u>		
↓	Blank	When card 10 is blank the program finds the correct bandwidths
<u>Card 11</u>		
NTIME	1	Only processing one set of data
<u>Card 12</u>		
BTIME	0.0	Starting time for data set
<u>Card 13</u>		
ETIME	0.19922	End time for data set (Note: 256 samples only goes to time 0.19922 not 0.2 as discussed earlier)

The autocorrelation function shows how the data repeats itself. If repetition occurs, the autocorrelation value is equal to 1, if the data does not repeat itself, then the value is 0. By looking at Figure IV-1 it can be seen that the process $X(t)$ repeated itself at a time of 0.03969 seconds. This almost corresponds to the period of the 25 cycle per second sine wave which should be expected. The time did not correspond exactly to the period of the 25 cycle per second signal because the plot is only a digital representation of the continuous plot. If 0.04 seconds had appeared on the time axis, the value of the autocorrelation function would have been exactly 1.0 instead of the value 0.995752. The computer plot repeats itself every 0.03969 seconds. Because of space limitations, the plot was truncated at 0.07315 seconds. Common knowledge of the cyclic nature of sine waves reduces the splendor of this plot. However, if the process had been considerably more complex, the autocorrelation function might become magnificent by showing the process was repeating itself.

Any original data set can be approximated by summing several sine waves of various frequencies and amplitudes. The root mean square plot shows the frequencies and root mean square amplitudes of each component. Figure IV-2 shows that two major frequencies make up the data. These frequencies are 25.099 cycles per second and 100.398 cycles per second which is exactly as was anticipated. The amplitudes of the waves are 48.99423 and 48.09558, respectively. The root mean square

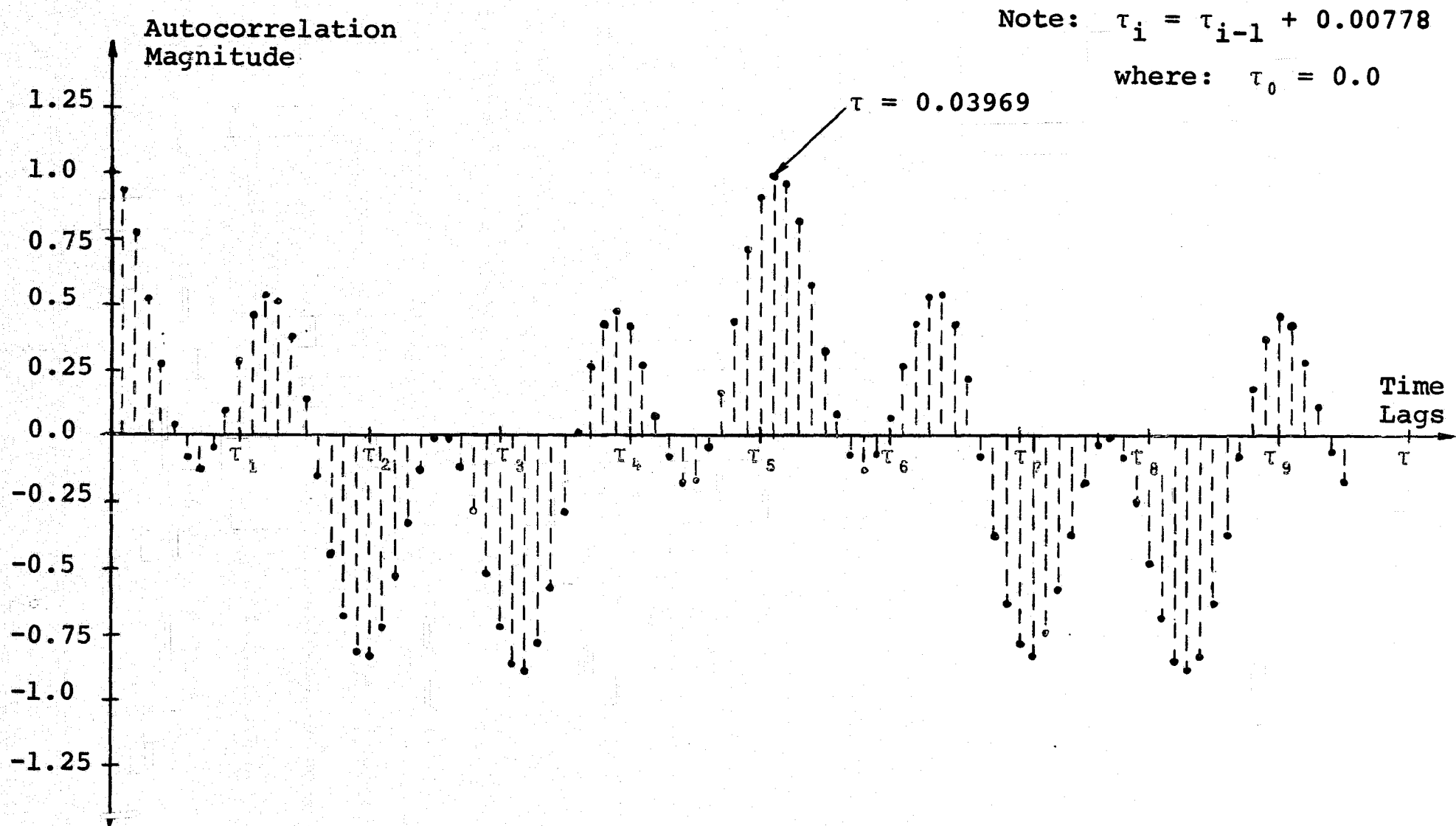


Figure IV-1. Autocorrelation Function Versus Time Lags

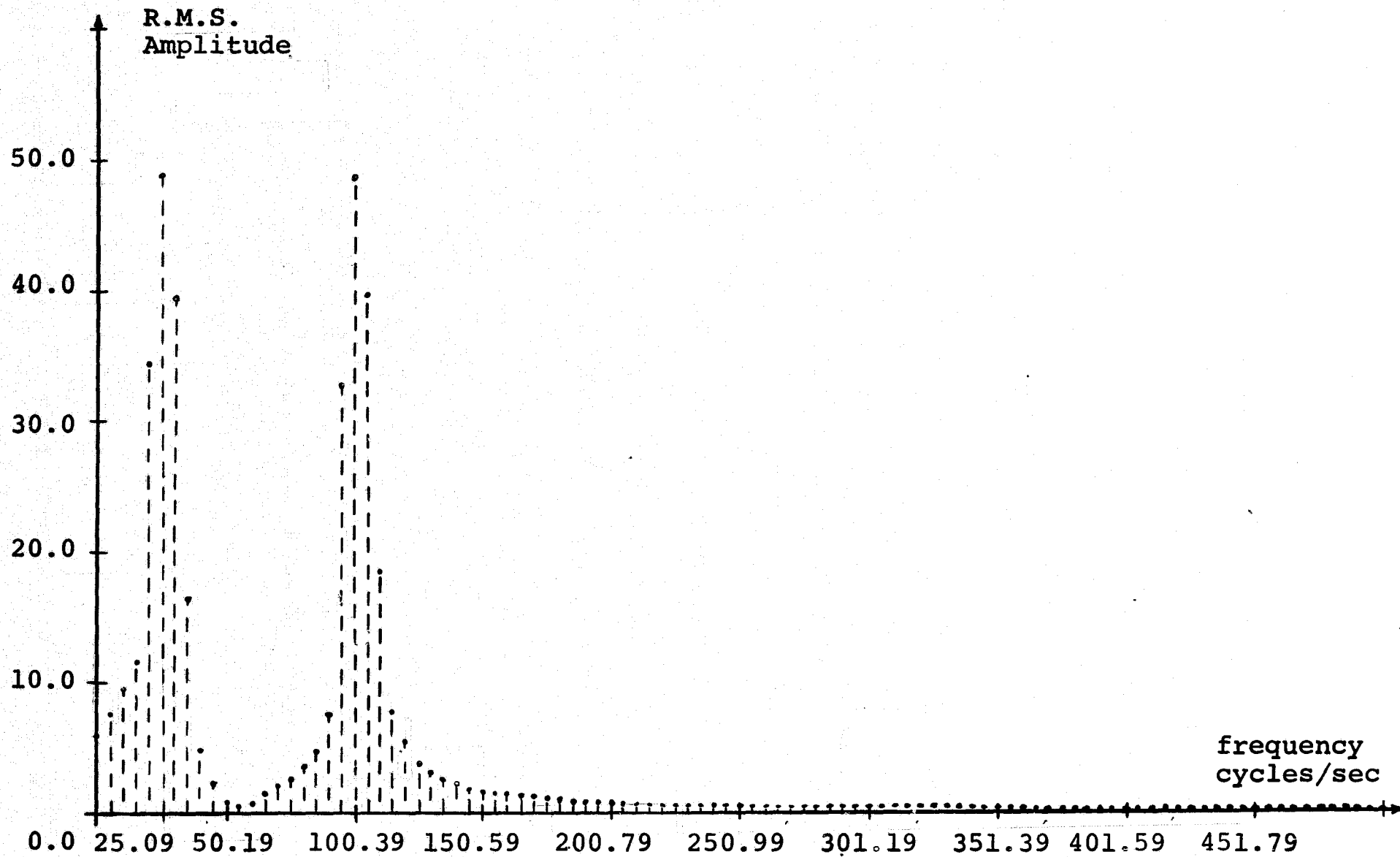


Figure IV-2. R.M.S. Amplitudes Versus Frequency

amplitude of a sine wave is defined to be $(\sqrt{2}/2)$ multiplied by the magnitude of the wave. Each of the sine waves used in the input had a magnitude of 75 which corresponds to a root mean square amplitude of 53.025. This corresponds closely to the computer results when considering that the plot is a digital representation of a continuous plot. These two plots would have defined the process $X(t)$ if it had been unknown. This program will aid many engineers by defining specific parameters in their areas of study.

EXAMPLE 2

The problem to be analyzed is described mathematically by $Y(t) = 1$ for $n \leq t < \frac{n+1}{2}$ and $Y(t) = -1$ for $\frac{n+1}{2} \leq t < n + 1$ where $n = 0, 1, 2, \dots$. This problem is nothing more than a square wave with a period of 1 second and an amplitude of 1 unit. The parameters for digitizing the data can be found by using the method outlined in Table IV-1.

Step 1: Pick the fundamental frequency to be 0.25 cycles per second.

Step 2: Find the total time of the data set.

$$\text{Total Time} = \frac{1}{f_{\text{zero}}} = 4.0 \text{ seconds}$$

Step 3: Since the highest frequency component contained in the data is unknown, Shannon's sampling theorem cannot be applied.

Step 4: Because step 3 could not be applied, the sampling period is arbitrary.

Step 5 and 6: Find the number of data points contained in the record. Since all previous values have been arbitrary, the number of samples can be chosen arbitrarily. Let:

sample = 1024.

Step 7: The final sampling period can now be found by the following formula.

$$T_s = \frac{\text{Total Time}}{\# \text{ samples}}$$

$$T_s = 0.003906 \text{ second}$$

Step 8: The sampling frequency can now be found using the result of step 7.

$$f_s = 256 \text{ samples/second}$$

A process for digitizing the data must be selected. In this case a computer subroutine generated the data and punched out a set of data cards. Table IV-3 lists the control parameters, and why each value was selected. These values are punched on cards and then placed in front of the data cards. The control parameters and the data should be placed between the last two delimiter cards. As in Example 1, the entire computer output is not shown in this document because of its large size. However, the printed statistics, the measured probability density, and the power spectral density function have been

TABLE IV-3

Selected Control Parameter for Example 2

Name	Value	Reason for Value Selected
<u>Card 1</u>		
IOP1	5	Needed a plot of the input data
IOP2	5	Needed the probability density and statistics
IOP3	5	Curious about the autocorrelation
IOP4	5	Needed the Power Spectral Density
IOP5	5	Curious about R.M.S. plot
IOP6	0	Not used in program at this time
IOP7	0	Did not need a Gaussian plot
IOP8	0	Did not need a Rayleigh plot
IOP9	0	Did not need a filter
IOP10	0	Was not interested in logarithmic P.S.D.
IOP11	0	No need to output information on Unit 12
IOP12	0	Not used in program at this time
IOP13	5	Delete data values from plot output
IOP14	0	Read input data from punched cards
IOP15	0	Han (smooth) the data
FC1	Blank	The filter was not used
FC2	Blank	The filter was not used
FT1	Blank	The filter was not used
FT2	Blank	The filter was not used
PERROR	0.0	0% error desired
IFORT	1	Controls some data shuffling
NTB	1024	Number of data points
NMB	2	Number of words stored per time space
NWP	2	Data value is the second word in the time space
<u>Card 2</u>		
ATITLE	Square Wave With a Period of 1.0 cycle/second	
<u>Card 3</u>		
BTITLE	Square Wave With a Period of 1.0 cycle/second	
<u>Card 4</u>		
CTITLE	Square Wave With a Period of 1.0 cycle/second	
<u>Card 5</u>		
DTITLE	Square Wave With a Period of 1.0 cycle/second	
<u>Card 6</u>		
ETITLE	Square Wave With a Period of 1.0 cycle/second	
<u>Card 7</u>		
ID	Mike Forehand --- Example 2	

TABLE IV-3 (continued)

Name	Value	Reason for Value Selected
<u>Card 8</u>		
IDATE	5-3-71	Current date of the computer run
ITAPE	No Tape	The data was read on punched cards
IR	128	VREAD reads records of 128 words long
<u>Card 9</u>		
BW	0.5	2 (Fundamental Frequency)
SAMPLE	256.0	Sampling frequency
CUTFRE	Blank	If left blank CUTFRE=SAMPLE/2.0
ANALOG	Blank	Not used in this evaluation
<u>Card 10</u>		
	Blank	If card 10 is left blank the bandwidths are selected by the program
<u>Card 11</u>		
NTIME	1	Only one data set processed
<u>Card 12</u>		
BTIME	0.0	The process starts at 0 seconds
<u>Card 13</u>		
ETIME	3.999	The process ends at 4 seconds

hand plotted using the information from the computer plots.

Statistical values give information about how the input data values vary with respect to time. These values are printed in Table IV-4.

TABLE IV-4

Statistical Parameters for a Gaussian Probability Distribution

PROBLEM -- Mike Forehand -- Example 2

TIME INTERVAL -- 0.0 to 4.000

DATE PROCESSED -- 5-3-71

TAPE NUMBER -- NO TAPE

NUMBER OF DATA VALUES -- 1024

Maximum Data Value .	1.00000	Minimum Data Value. .	1.00000
Mean	0.00000	Standard Deviation. .	1.00049
Skew	0.00000	Kurtosis	0.99805
2.5% Confidence Level	30.19099	5.0% Confidence Level	27.58710
CHI Square	18674.52734	Degree of Freedom . .	17
Coefficient of Variation . . .		1.000488	

The above table shows that the maximum data value is 1.0, the minimum data value is -1.0, and the average data value, the mean, is 0.00000. The standard deviation is 1.00049. This implies that the signal varies on the average of 1.00049 from the mean which should be anticipated because the process $Y(t)$ is a square wave. The skew and kurtosis are the third and fourth probability moments. These quantities are mathematical terms which have very little physical meaning. The remaining parameters are quantities comparing the measured probability density to that of a Gaussian probability density. The number

of degrees of freedom determines which value to pick out of a chi-square table for the specified percent confidence levels. The 2.5 percent and 5.0 percent confidence levels are tabulated quantities found in the chi-square tables. These numbers should be compared to the quantity CHI-Square. If CHI-Square is less than the 2.5 percent confidence level, then the process has a Gaussian distribution within that percentage. A similar analysis can be applied for the 5.0 percent confidence level. Therefore, it is quite obvious that this square wave does not have a Gaussian distribution. Another table exists in the program output which compares the process to a Rayleigh distribution. The procedure to determine if the measured probability has a Rayleigh distribution is identical to the above procedure for the Gaussian distribution. The coefficient of variation is a number found by dividing the mean into the standard deviation. A large number would show that the data is very erratic, but a small number would mean the data had only a few rapid changes. These statistical quantities relate useful information about the variations of the process being analyzed.

The probability density function determines the possibility of obtaining a particular value of the original process. The Figure IV-3 shows the measured probability of a particular value occurring for the process. It is obvious from the plot that only two values occurred during the complete set of data. These two values are 0.9500 and -0.9500, and the corresponding probability of each is 5.0000095 and 5.0000095, respectively. Each value of the

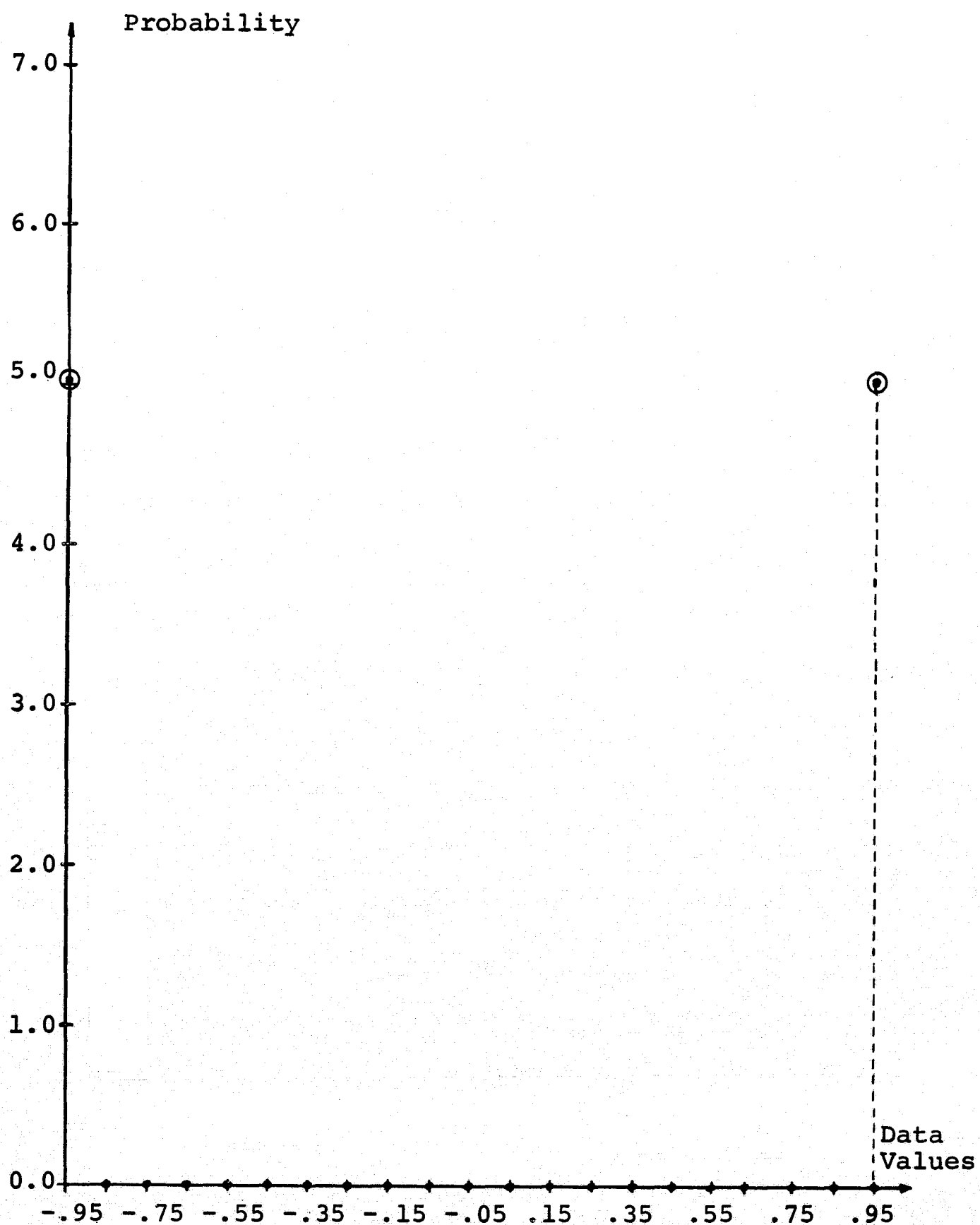


Figure IV-3. Measured Probability Density

abscissas has a range, plus and minus a constant, associated with it; therefore, the value 0.9500 also contains the probability of the value 1.0. The probability density function gives a good indication of the randomness of the process.

The power spectral density function provides information about the amount of power concentrated at a given frequency. As can be seen from Figure IV-4, the majority of the power is contained in the low frequency components; however, some power remains at the higher frequencies. The Fourier analysis of a square wave indicates that an infinite sum of sine waves of decreasing amplitude and increasing frequency will result in a waveform exactly like the square wave. The mathematical expression of the Fourier series for a square wave is

$$F(t) = \frac{4}{\pi} \sum_{n=1,3,5,\dots\infty} \frac{1}{n} \sin \frac{n\pi t}{0.5}$$

The amplitudes of this expression are not equal to the power spectral density amplitudes, but it is easily seen that the amplitudes of the power spectrum do decrease similar to those of the Fourier series. The power spectrum peaks occur at the same frequencies as indicated in the Fourier series. This plot is very important in determining the frequency components of an input signal.

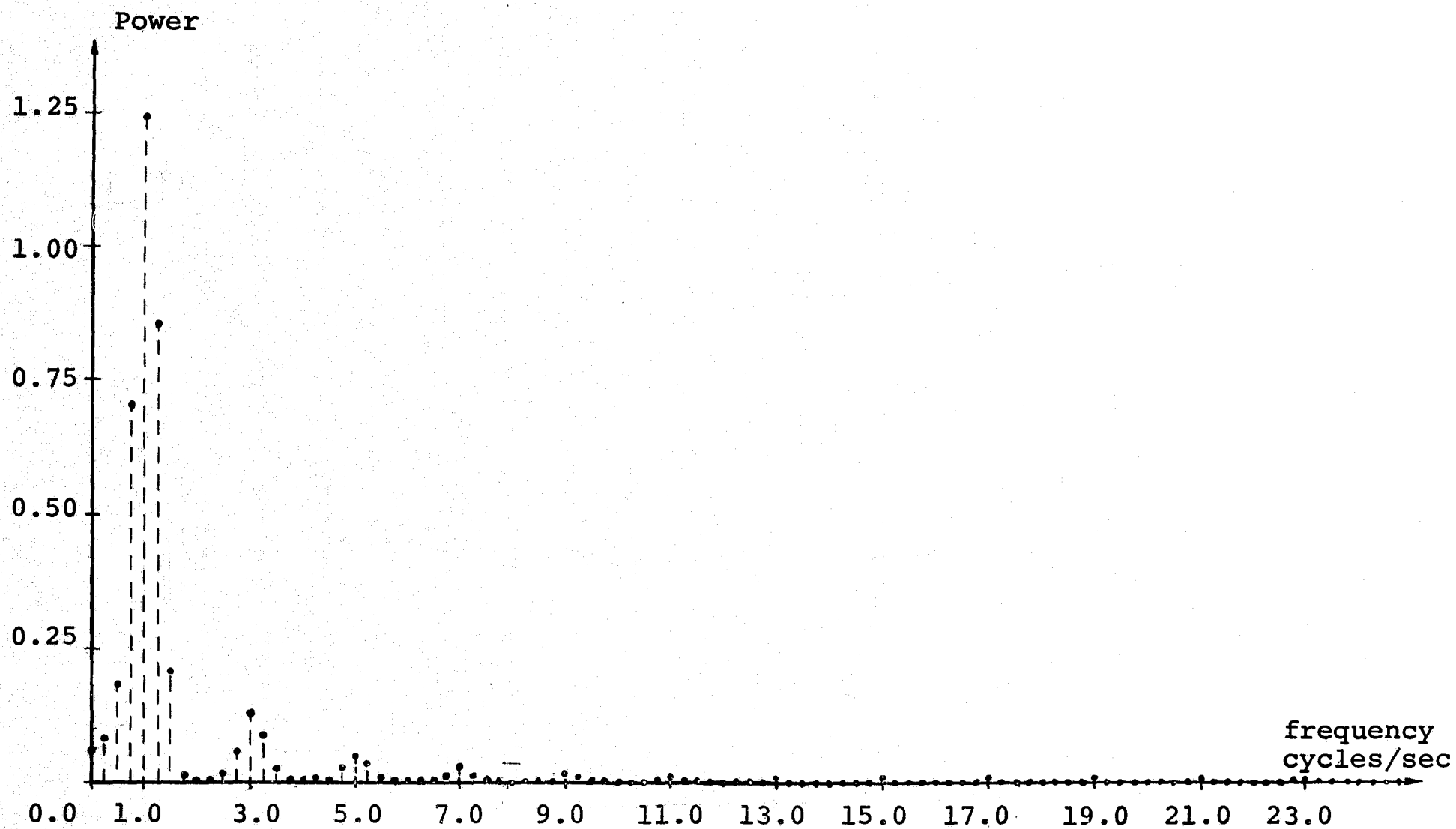


Figure IV-4. Power Spectrum

CHAPTER V

ANALYSIS OF A DIGITAL FILTER

The application of this data analysis program to a real problem will illustrate the potential value of this piece of computer software. More and more of these canned programs will appear in the years to come. The ease of using this program makes data processing become a simple task. The problem posed in this chapter is how effective will a particular digital filter be when it must filter random Gaussian noise. Without this program many long hours would be required to calculate mathematical values and then make statements about the input signal and the output signal. With the aid of this program, only fifteen or twenty minutes is needed to prepare the program for processing. Approximately ten minutes are required for the program to be run on an IBM 360/40 computer, then the answers are ready for the programmer to make his conclusions. The results of the program allow definite statements to be made about the data characteristics.

Allen R. Atkins [1971] developed a mathematical model of an existing bandpass filter. This mathematical model was programmed for the computer and subjected to random Gaussian noise, and the output was recorded on punch cards. It is

desired to analyze the input and output signals and draw a conclusion about the filter and its effectiveness. The data was already in the digital form. The sampling period was 0.002 seconds which gives a sampling frequency of 500 samples per second. In order that the lengthy computer output could be redrawn and included in this document, the total number of data points was selected to be 256. This makes the final time equal to 0.512 seconds. If space had not have been a factor, the number of data points chosen would have been approximately 4096 or 8192 values. This would probably give a stationary data set. If 256 data points gives a stationary record, then inclusive statements can be made about the filter. Table V-1 lists the control parameters for the input and the output signals since both signals were to be processed identically. The resulting plots included in this document are the input and output data plots, the input and output autocorrelation plots, and the input and output root mean square amplitude plots.

The first set of plots are the filter input data shown in Figure V-1, and the filter output data shown in Figure V-2. Figure V-1 shows that the signal is definitely a random process. However, the filter output in Figure V-2 is shown to be a smooth sinusoidal waveform with a decreasing amplitude. Since the output is a smooth process, a conclusion can be made that the filter is working.

The autocorrelation function of the input and the output is recorded in Figure V-3 and Figure V-4, respectively.

TABLE V-1

Selected Control Parameters for the Filter Problem

Name	Value	Reason for Value Selected
<u>Card 1</u>		
IOP1	5	Needed a plot of data values
IOP2	0	Statistics and Probability not needed
IOP3	5	Needed autocorrelation function
IOP4	5	Needed power spectrum
IOP5	5	Needed R.M.S. amplitudes
IOP6	0	Not used in the program anymore
IOP7	0	Gaussian probability not needed
IOP8	0	Rayleigh probability not needed
IOP9	0	Do not filter the data
IOP10	0	Log P.S.D. plot not needed
IOP11	0	No need to output values on Unit 12
IOP12	0	Not used in the program anymore
IOP13	5	Did not want a listing of the data after the plot
IOP14	0	Read the data from the punch cards
IOP15	0	Ham (smooth) the data
FC1	Blank	Leave blank because internal filter is not used
FC2	Blank	Leave blank because internal filter is not used
FT1	Blank	Leave blank because internal filter is not used
FT2	Blank	Leave blank because internal filter is not used
PERROR	0.0	Wanted 0% error
IFORT	1	Needed to organize the data values
NTB	256	Number of data values used in this example
NMB	1	Amount of information per time space
NWP	0	Time value was not with the data values
<u>Card 2</u>		
ATITLE	Analysis of a Digital Filter	
<u>Card 3</u>		
BTITLE	Analysis of a Digital Filter	
<u>Card 4</u>		
CTITLE	Analysis of a Digital Filter	
<u>Card 5</u>		
DTITLE	Analysis of a Digital Filter	

TABLE V-1 (continued)

Name	Value	Reason for Value Selected
<u>Card 6</u>		
ETITLE	Analysis of a Digital Filter	
<u>Card 7</u>		
ID	Mike Forehand --- Chapter V	
<u>Card 8</u>		
IDATE	5-4-71	The date which the program ran
ITAPE	No Tape	The tape reader was not used
IR	128	Because VREAD reads data in increments of 128 sections
<u>Card 9</u>		
BW	1.95306	2 (Fundamental Frequency)
SAMPLE	500.0	Sampling frequency
CUTFRE	Blank	The program will find the correct value
ANALOG	Blank	Not used in this evaluation
<u>Card 10</u>		
↓	Blank	When card 10 is blank the program finds the correct bandwidths
<u>Card 11</u>		
NTIME	1	Only one data set processed
<u>Card 12</u>		
BTIME	0.0	Starting time for the data set
<u>Card 13</u>		
ETIME	0.512	End time for data set

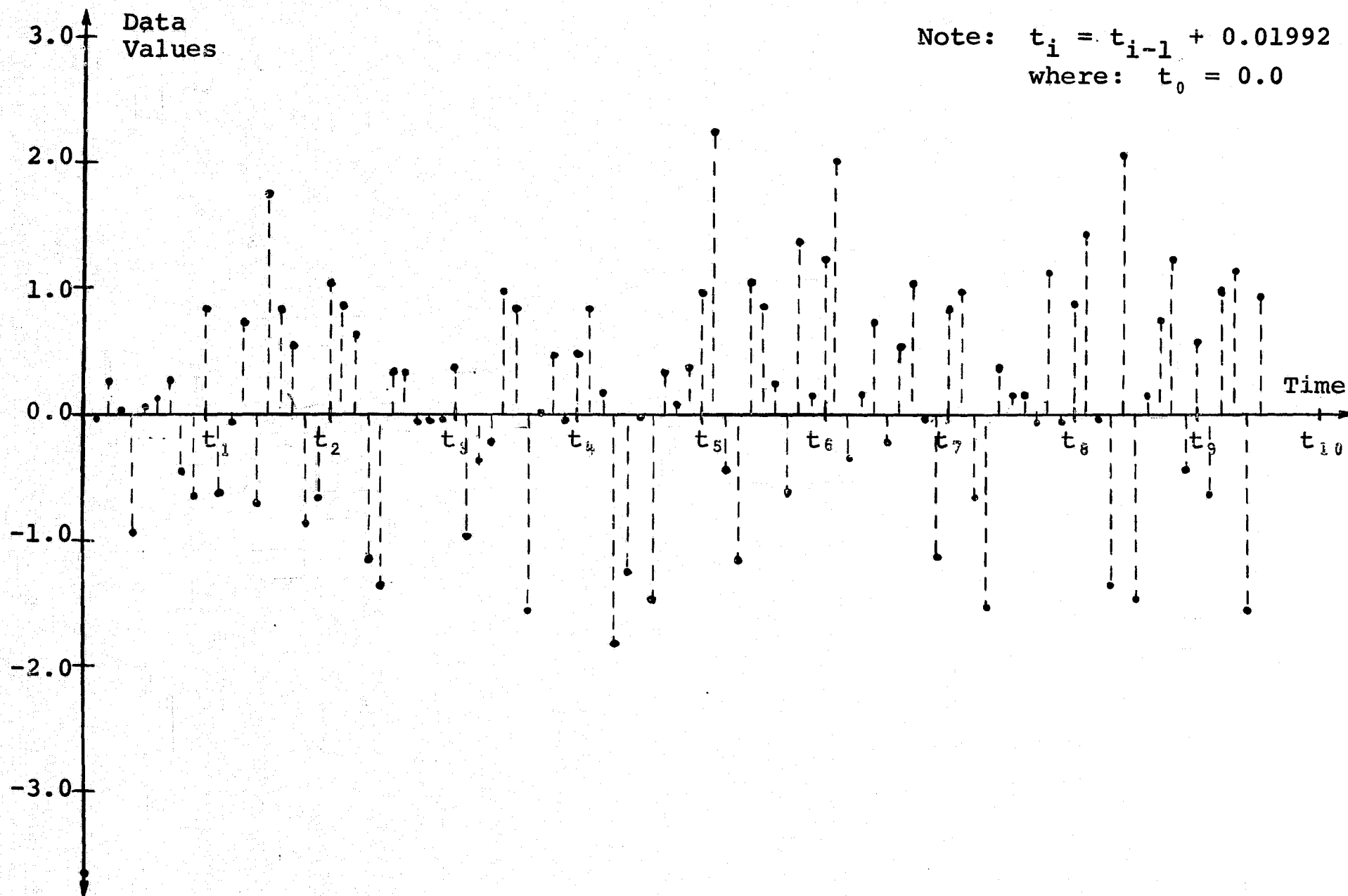


Figure V-1. Input Data for Filter (Random Gaussian Noise)

Data Values

Note: $t_i = t_{i-1} + 0.01992$

where: $t_0 = 0.0$

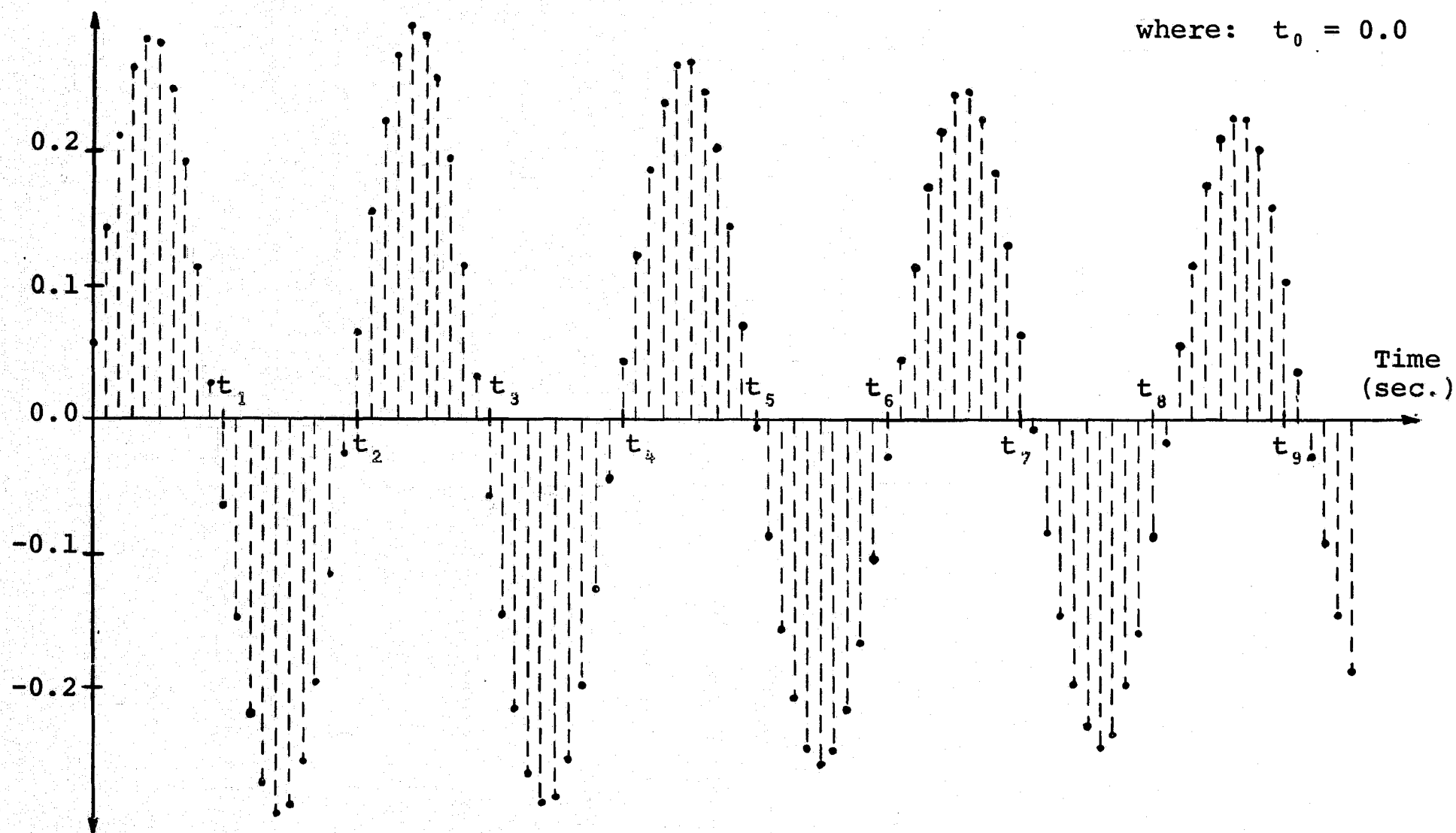


Figure V-2. Filter Output Data

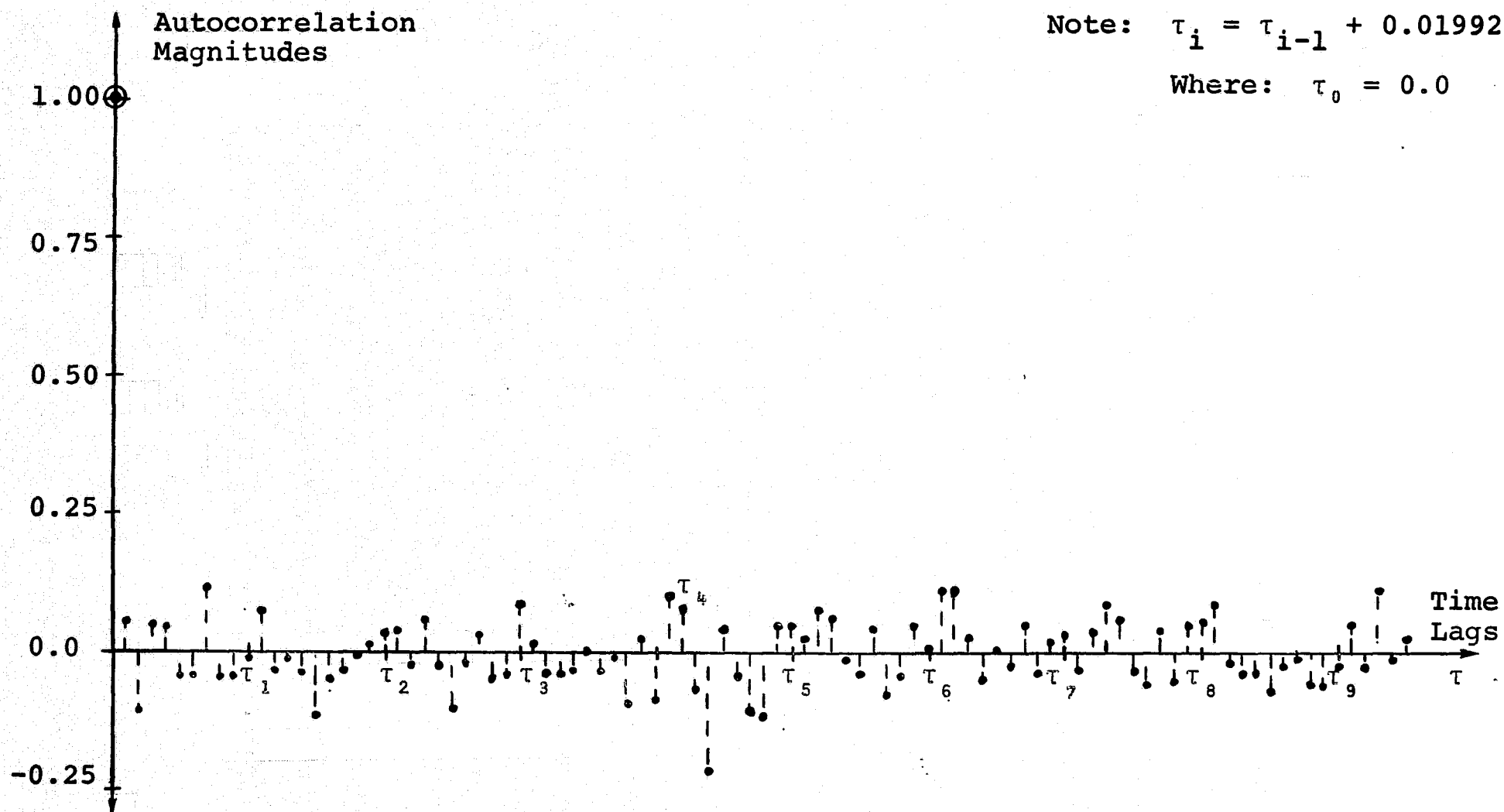


Figure V-3. Autocorrelation Function Versus Time Lags (Random Gaussian Noise)

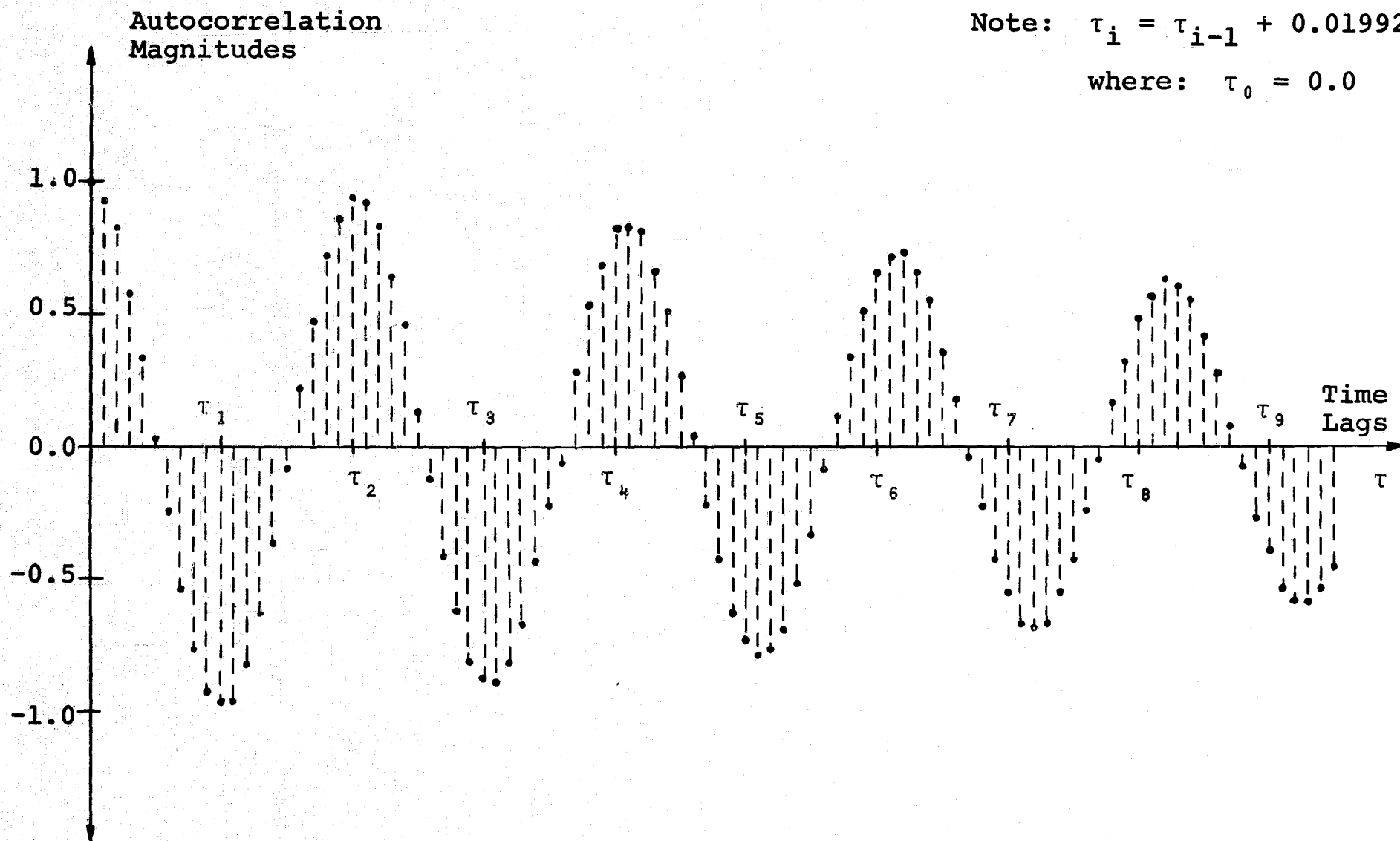


Figure V-4. Autocorrelation Function Versus Time Lags (Filter Output)

This function gives information about the repeatability of a signal. Obviously Figure V-3 shows that the input data never repeats itself. The first large peak shows that the input signal is only perfectly correlated when the time shift, τ , equals zero. The smaller peaks show that the signal is slightly similar at various lag times. The autocorrelation plot of the output signal, Figure V-4, is considerably different. This plot shows that the output is very cyclic in nature and repeats itself at regular intervals. These intervals of time are 0.03984 seconds which corresponds closely to the period of a sine wave with 25 cycles per second frequency. The regular repetition of the autocorrelation plot fosters the conjecture that the output of the filter consists primarily of a sinusoidal wave with a frequency of 25 cycles per second.

The root mean square amplitude plot will reveal any frequency components contained in the data set. Figure V-5 reveals the frequency components of the filter input, and Figure V-6 shows the frequencies contained in the filter output. The filter input is random Gaussian noise which is composed of many frequencies of varying amplitudes which can be confirmed by Figure V-5. Both the amplitudes and the frequencies of these components are continually changing. For this reason Figure V-2 appeared to be a sinusoidal wave of varying amplitude. The magnitude of the 25 cycle per second component varied exactly as the output of the filter. It is noted that the root mean square plots will only show

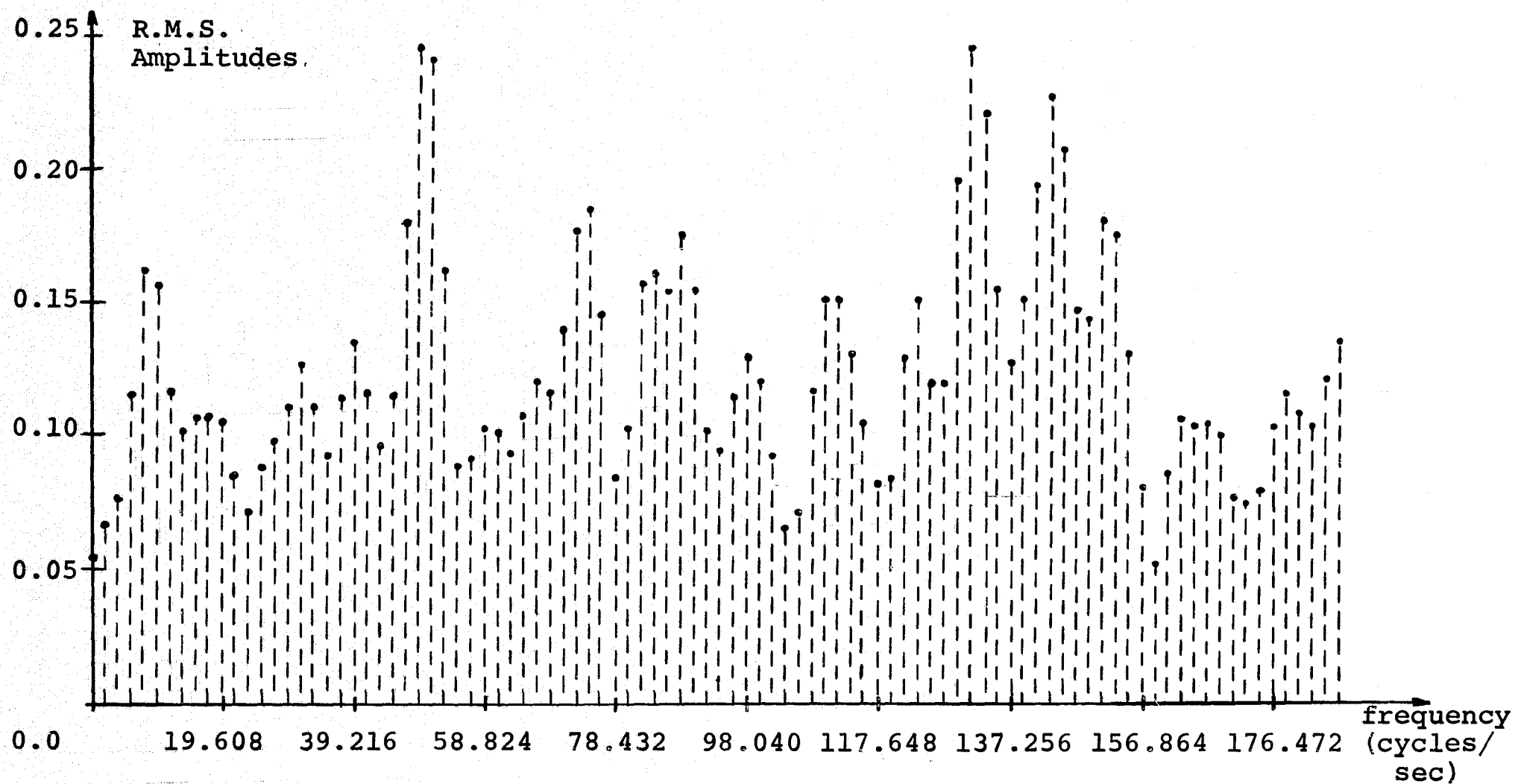


Figure V-5. R.M.S. Amplitudes Versus Frequency (Random Gaussian Noise)

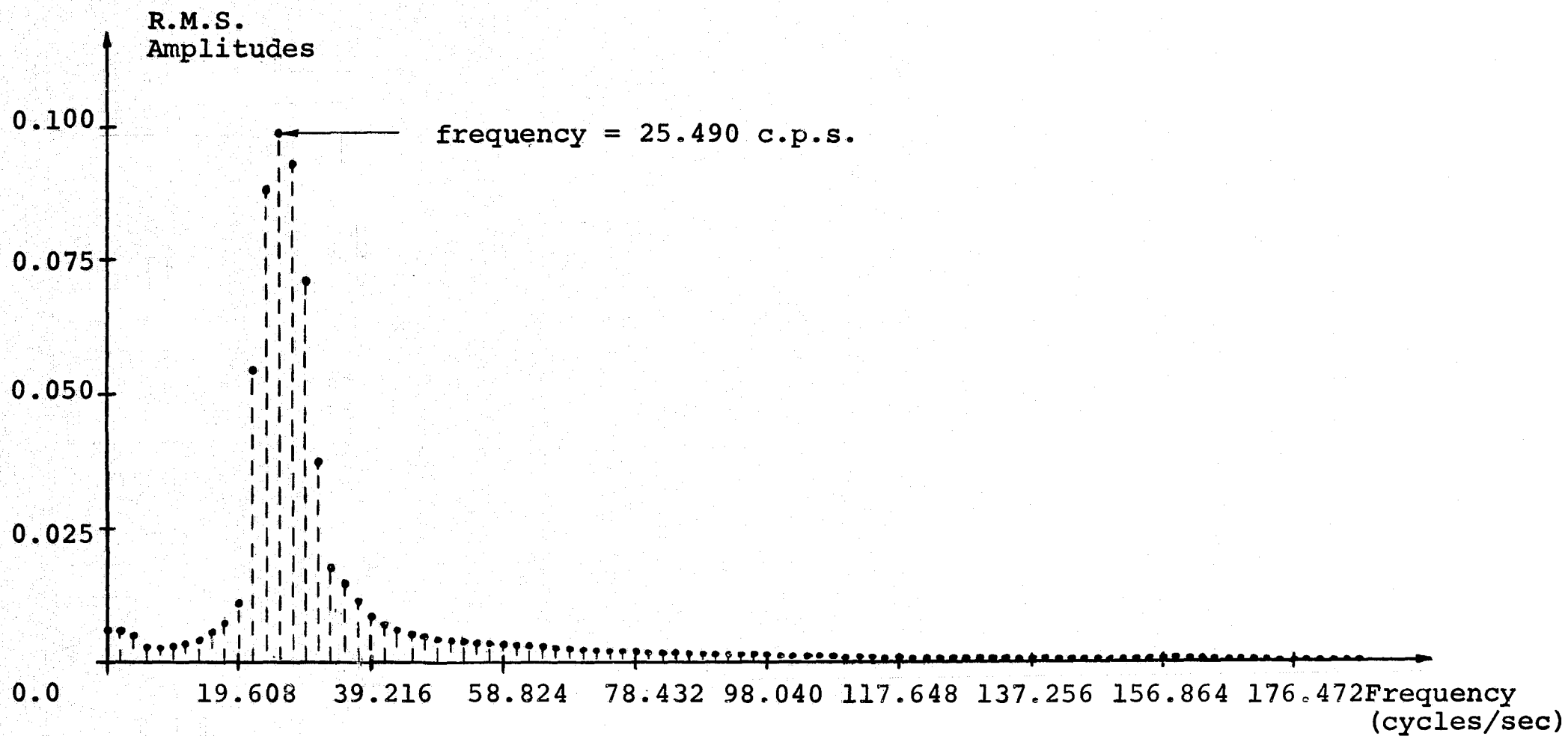


Figure V-6. R.M.S. Amplitudes Versus Frequency (Filter Output)

the average magnitude during the time span of the data. Figure V-6 shows that the filter output consisted of the sum of sinusoidal waves with frequencies of 23.52972 to 27.45132. An educated guess based on the plot of Figure V-2 would be that only one sine wave existed, and the bell shaped curve in Figure V-6 is due to the hanning (smoothing) process. If this guess is true, the filter is definitely a narrow band pass unit. It would be very interesting to see Figure V-6 if the data were not hanned.

CHAPTER VI

CONCLUSIONS

The modified version of the VIBAN3 program offers an economical method of analyzing random sets of data. This program will analyze 16 sets of data where each set contains 8192 points. The computer output includes plots and tables of the input data, the general statistics, the probability density, the autocorrelation function, the power spectral density, and the root mean square frequency components. Standard mathematical techniques are employed to determine the general statistics and the probability density. The standard procedure for finding the basic frequency components is to find the autocorrelation function then take the inverse Fourier transform. This results in the power spectral density which is then converted into the Fourier frequency components. This program saves time in evaluating the discrete Fourier transform by the fast Fourier transform method. This method is so fast that the total evaluation time is decreased if the Fourier frequency components are found first. The power spectrum can be evaluated by squaring these coefficients, and the autocorrelation is found by taking the inverse discrete Fourier transform of the power spectrum. This procedure has shortened the number of numerical operations by factors of

several hundred. Typical execution times of the program on the IBM 360/40 computer are eight minutes for 256 data points and fifteen minutes for 1024 data points. These unique digital techniques are proven to be effective in data analysis.

The accuracy of this procedure is dependent upon the amount of data analyzed. It was found that even though the data had been digitized properly, errors could be detected when the entire number of data points was relatively low. For instance, in Example 2 of Chapter IV the total number of data points was selected to be 256, which were evenly distributed in one cycle of the square wave. The autocorrelation plot was found to be in error. This plot should vary linearly between +1 and -1 in 0.5 seconds. The autocorrelation plot was found to have a curved shape, and the power spectral density had similar erroneous values. The problem was due to the small number of data values. Therefore, it was decided to increase the number of data points to 1024 by increasing the overall time span of the data and maintain the same sampling frequency. This corrected the errors in Example 2. It is noted that the increase in the total time span of the data lead to a finer grid for the power spectrum. This would not occur if the number of data values were increased by sampling faster since the fundamental frequency is not changed. Therefore, when increasing the number of data values to obtain better results, two methods are available. The first is the increase of the sampling frequency which will change the autocorrelation plot but will not alter the

power spectrum. The second method increases the total time span of the data and should alter both plots. The method chosen will depend upon the physical problem being analyzed.

A recommendation for future work would be to obtain a binary card deck which would alleviate the long program compilation time. Other points to consider for future evaluations will depend upon the current computer size. With the IBM 360/40 computer, this program has evolved close to its maximum capability. If Tennessee Technological University obtains a new computer, several program modifications may aid in data analysis. For instance, the addition of a subroutine to generate a one-third octave plot would lend additional information about the frequency characteristics of the input data. Also, a subroutine which performs the cross correlation would prove valuable when processing several sets of data consecutively. Analog plots would make the program complete. At present the program must make digital plots using the computer printer, however, a continuous plot would present the output in a neater form. These modifications would require additional computer equipment. This evaluation shows that digital processing of numerical data can be accomplished accurately and rapidly.

REFERENCES

- Atkins, Allen R. 1971. "An Optimal Controller Based on Linear Approximation of an Acoustical Test Facility." Unpublished Master's Thesis, Tennessee Technological University.
- Bendat, Julius S. and Allan G. Piersol. 1967. Measurement and Analysis of Random Data. John Wiley and Sons, Inc., New York.
- Cooley, James W. and John W. Tukey. 1965. "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation, Vol. 19, pp. 297-301.
- G-AE Subcommittee. 1967. "What is the Fast Fourier Transform," I.E.E.E. Transactions on Audio and Electroacoustics, Vol. AU-15, pp. 45-55.
- Gold, Bernard and Charles M. Rader. 1969. Digital Processing of Signals. McGraw-Hill Book Company, New York.
- Papoulis, Athanasios. 1965. Probability, Random Variables, and Stochastic Processes. McGraw-Hill Book Company, New York.
- National Aeronautics and Space Administration. 1967. Program Documentation M.S.C. Statistical and Wave Analysis Program (VIBAN3). Program Number 0467, Project Number 4587.
- Tou, Julius S. 1959. Digital and Sampled-Data Control Systems. McGraw-Hill Book Company, New York.