NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report No. 32-993

A Power-Spectral-Density Computer Program for the Vibration Laboratory

C. P. Chapman

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

August 1, 1966

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report No. 32-993

A Power-Spectral-Density Computer Program for the Vibration Laboratory

C. P. Chapman

E. L. Sheldon, *Manager*
Environmental and Dynamic Testing Section

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

August 1, 1966

Prepared Under Contract No. NAS 7-100
National Aeronautics & Space Administration

CONTENTS

TABLES

FIGURES

ABSTRACT

This Report presents the considerations for, description of, results from, and summary of the computer program written by the Environmental Laboratory personnel at JPL.

The computer program calculates vibration-acceleration densities (g^2/cps) at particular frequencies along power-spectral density (PSD) plots, and root-mean-square (rms) acceleration (g_{rms}) across the PSD-frequency spectra of random-noise test specifications. The most significant advantage of this computer program is its complete generality. It can be used to calculate g^2/cps and g_{rms} over any PSD specification without modification; only the data cards have to be changed. The output data from the computer program are used to verify the specification test levels and provide the theoretical PSD to obtain a spectra ratio for purposes of analysis on completion of the vibration test. The results of the vibration test can then be displayed on a graph that illustrates in db the deviation from the specified g^2/cps at any frequency within the test-frequency spectrum.

I. INTRODUCTION

The excitation forces produced during a random-noise vibration test are not at discrete frequencies, but exist over a continuous band. For this reason, it is convenient and meaningful to think in terms of power concepts, such as rms and the concept of acceleration density, when discussing accelerations produced from random-noise excitation. In mathematical terms, the acceleration density is defined as

$$g_0 = \lim_{BW \to 0} \frac{g^2_{rms}}{BW} \tag{1}$$

where

g_0 = acceleration density
BW = bandwidth of the frequency spectrum in cps
g_{rms} = the rms value of the random acceleration

From this relationship, acceleration density is derived in units of g^2/cps.

As the bandwidth approaches zero cps, the acceleration density given by Eq. (1) approaches that of a single-frequency component. A plot of the acceleration density of each frequency component gives a locus of g^2/cps vs frequency over the bandwidth under investigation. This is the PSD curve. A summation of acceleration densities over the frequency spectrum of interest yields the mean-square value of acceleration (g^2_{rms}). In mathematical terms,

$$g^2_{rms} = \int_{f_1}^{f_2} g_0 \, df \tag{2}$$

where

f_1 = the lower-frequency limit of the frequency spectrum
f_2 = the upper-frequency limit of the frequency spectrum
g_0 = the acceleration density in units of g^2/cps

II. NEED FOR A COMPUTER PROGRAM TO CALCULATE THEORETICAL g^2/cps

When the Environmental Laboratory receives a random-noise test specification, it is sometimes illustrated in the form of a PSD sketch on log-log coordinates as in Fig. 1. An engineer will verify the overall acceleration level (g_{rms}) using the equations listed in Appendix A. This can be done on a slide rule. The test is performed and acceleration levels are recorded on tape. The taped data are analyzed and PSD plots are made for the test requester. There may be a large number of plots, depending on the type of test, the test specimen, and the number of accelerometers recorded during the vibration test.

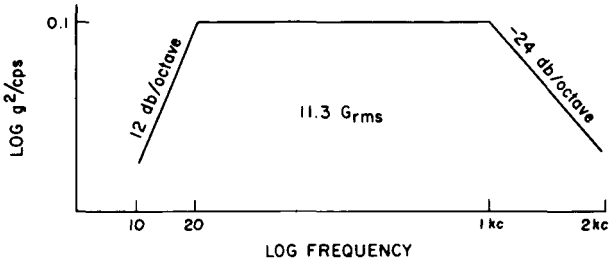

Fig. 1. Random-noise-vibration test specification

It was suggested that perhaps the recorded data could be displayed in such a manner that the test requester might better understand the results of testing his package. One method to simplify the interpretation of the test results would be to ratio the resulting PSD with the theoretical PSD over the entire frequency spectrum of the test specification. This is not a new concept, but no convenient method existed to obtain a large number of theoretical acceleration densities. Several hours or days of slide-rule calculations could be performed for complicated PSD specifications. Plotting the true PSD on

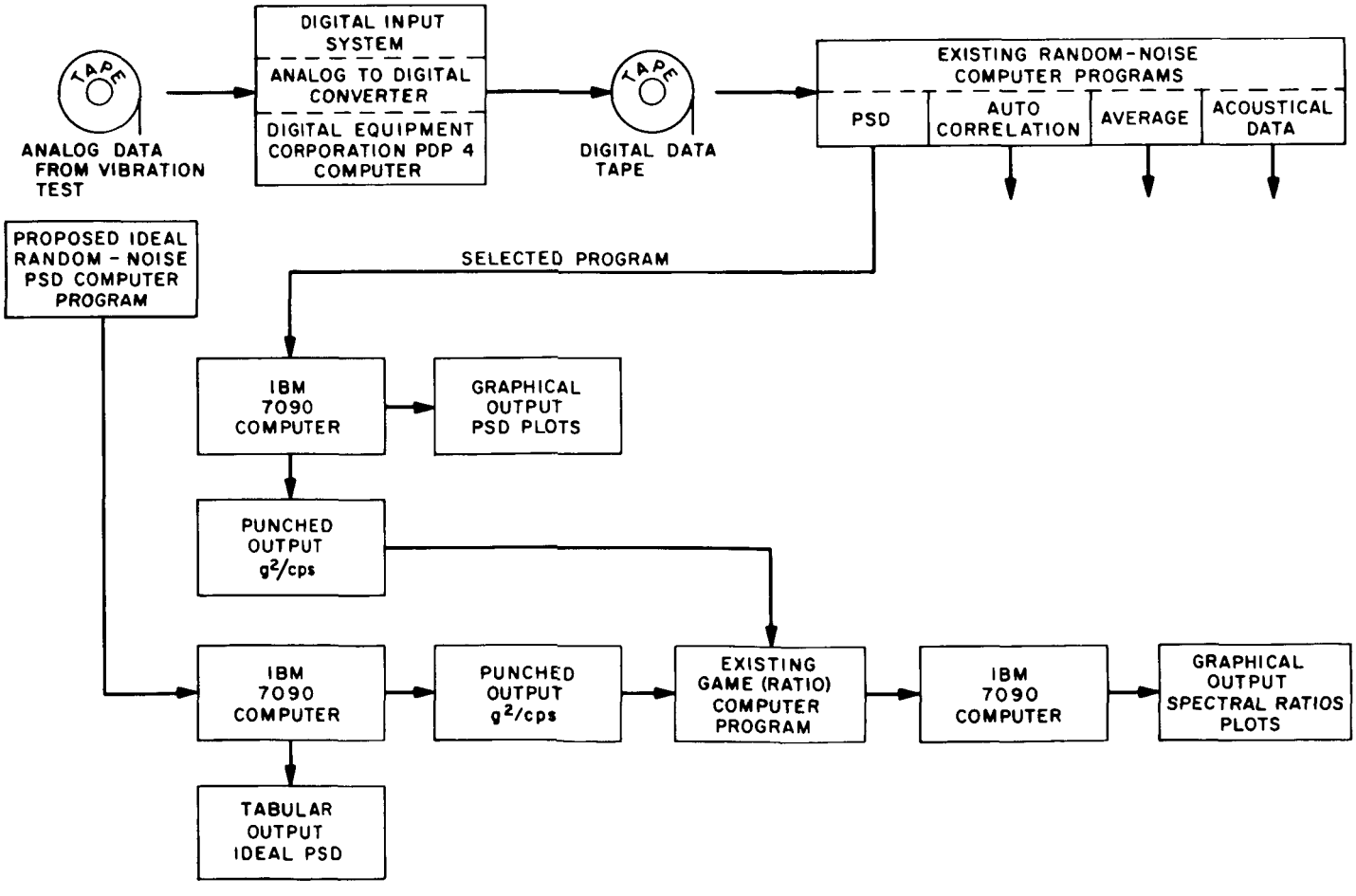

Fig. 2. Block diagram of random-noise-vibration data-analysis procedure

expanded log-log graph paper and reading g^2/cps from the graph paper is another means of obtaining the theoretical acceleration densities. An engineer with some digital-computer programming experience will realize that a modern digital computer can compute g^2/cps at 10-cycle intervals, or even quarter-cycle intervals if desired in 20 to 30 sec (a minimal cost). These acceleration densities can then be ratioed against the resulting vibration-test acceleration densities at the same frequencies to obtain a meaningful spectra ratio (test g^2/cps over ideal g^2/cps). A computer program already existed to analyze the data from the analog tapes.* Figure 2 illustrates, in block diagram form, the existing procedure for vibration data analysis, with the proposed procedure to simplify the understanding of random-noise vibration test results.

*Random-noise-vibration tests are analyzed by digital techniques rather than by analog techniques at JPL.

III. IMPORTANT CONSIDERATIONS

The most important consideration in writing this computer program was to make the program very general and easy to use. The program had to give accurate results for any random-noise test specification without requiring modification in any way, other than changing the data cards. The numbers to be put on the data cards had to be easy to understand and with very few calculations necessary.

Another consideration was the program output (calculations from the program). This output had to be in an easy-to-read form, and self-explanatory.

The last consideration, perhaps the hardest to implement, was to develop a computer program that could be used by people not familiar with either programming or computers, in a language common to digital computers.

IV. GENERAL DESCRIPTION

The program was written in Fortran IV, since this is the highest-order computer language for engineers and scientists. Most modern digital computers have Fortran IV compilers. The output data presented in this Report were obtained from an IBM 7090 computer.

The program was written in three segments. The first segment, labeled $IBFTC MAIN, reads the data cards into the computer and prints out the results of the calculations. The calculations for acceleration density are performed in the second segment, labeled $IBFTC SUBPSD. This segment is a subprogram called SUBROUTINE PWR. Two equations are available for solution:

$$PSD1 = \left[PSDU^2 \left(\frac{F1}{FU} \right)^{0.3322DB} \right]^{1/2} \qquad (3)$$

$$PSDU = \left[PSD1^2 \left(\frac{FU}{F1} \right)^{0.3322DB} \right]^{1/2} \qquad (4)$$

where

$PSD1$ = the acceleration density at the lower frequency $F1$

$PSDU$ = the acceleration density at the upper frequency FU

DB = the slope of the PSD curve (as defined on log-log coordinates) usually expressed in db/octave

The third segment, labeled $IBFTC RMS, calculates the area under each slope (g^2_{rms}), then sums each area and

takes the square root of the sum to obtain the total acceleration G_{rms} TOTAL. This is performed in a subprogram called SUBROUTINE GRMS. The equations solved in the third segment are for negative PSD slopes:

$$GSQ = \frac{PSD1 \times F1}{S}\left[1 - \frac{1}{(FU/F1)^S}\right] \qquad (5)$$

where

$GSQ = g^2_{rms}$, i.e., the mean-square value of the random acceleration as represented by the area under the negative slope

$S = (|DB|/3) - 1$, therefore, $|DB| \neq 3$

$PSD1 =$ the acceleration density at the *first* frequency $F1$ beginning the slope (break-frequency value)

$FU =$ the upper (or last) frequency of the negative slope

$|DB| =$ absolute value of the slope in db per octave. This is always a positive number even though the slope is negative

For zero slope:

$$GSQ = PSD1(FU - F1) \qquad (6)$$

where

$GSQ = g^2_{rms}$ — the mean-square value of the random acceleration as represented by the area under the zero slope section of the PSD curve

$PSD1 =$ the acceleration density at the first frequency $F1$ beginning at the zero slope section of the PSD curve

$FU =$ the last frequency where zero slope occurs

For positive slopes:

$$GSQ = \frac{PSDU \times FU}{T}\left[1 - \left(\frac{F1}{FU}\right)^T\right] \qquad (7)$$

where

$GSQ = g^2_{rms}$ — the mean-square value of the random acceleration as represented by the area under the positive slope of the PSD curve

$T = (DB/3) + 1$

$PSDU =$ the acceleration density at the *upper* frequency FU ending the positive slope

$F1 =$ the beginning frequency of the positive slope

$DB =$ slope in db per octave. This is a positive number

Equations (5), (6), and (7) are particular solutions to Eq. (2). To determine the total rms acceleration as represented by the area under the PSD curve, it is necessary to sum the squared values of acceleration under each slope and take the square root of the sum:

$$G_{rms} = [GSQ(1) + GSQ(2) + \cdots + GSQ(n)]^{1/2} \qquad (8)$$

where

$GSQ(n) = g^2_{rms}$ as represented by the area under the n^{th} slope

Note that if a negative 3 db/octave slope is used in Eq. (5), GSQ (i.e., g^2_{rms}) goes to zero. This will not affect the calculations performed in segment 2 of the computer program (acceleration-density calculations). Should the number 3 be used for the variable DB in the equation involving S, Eq. (5), the program is arranged to write a message on the output sheet to inform the user that a negative 3-db slope will not compute; the computer will set that particular area (g^2_{rms}) equal to zero so the total G_{rms} value will not be accurate. Again, this will not affect the values of g^2/cps. Negative 3-db slopes are practically nonexistent in vibration testing.

V. THE VARIABLES

To use the program, it is only necessary to understand what the variables are and how to represent them by numbers on an IBM card or code form sheet. There is one data card (IBM card) for each slope of the PSD specification, including zero slopes. For example, Fig. 1 would have three data cards. Table 1 describes the variables. Some of the variables are defined in computer language as INTEGERS which must *not* have

Table 1. Definitions of data card variables, referencing Fig. 3 and 4

Variable	Definition	Figure 3			Figure 4			
		Data cards			Data cards			
		1	2	3	1	2	3	4
K	Adjustable dimension information for SUBROUTINE GRMS. This is an *integer* number equal to the number of slopes (including zero slopes) that make up the total PSD curve.	3	3	3	4	4	4	4
KI	Subscript information for SUBROUTINE GRMS. This is an *integer* number and can be thought of as a sequence number for each data card. KI goes from 1 to K.	1	2	3	1	2	3	4
L	Subscript information for SUBROUTINE PWR. This is an *integer* number. It is an index number and depends upon the variables DELTA and N. For any PSD, $L = 1$ for the first data card and thereafter, $L_{i+1} = N_i + 1$ where i = the number of the data card, i.e., first data card, second data card, etc. Refer to the definitions of N and DELTA.	1	2	100	1	2	8	128
M	Adjustable dimensions information for SUBROUTINE PWR. M is an *integer* number and represents the total number of discrete frequency points where g^2/cps is calculated. M should be made larger than what is necessary. M = 3000 should be adequate for any vibration analysis.	3000	3000	3000	3000	3000	3000	3000
N	Subscript information for SUBROUTINE PWR. This is an *integer* number; it is an index number and depends upon the variables *DELTA, FU, FI.* $$N_i = \frac{FU - FI}{DELTA_i} + N_{i-1}$$ where i = the number of the data card, i.e., first data card, second, etc. For the last data card only, i.e., when $i = K$: $$N_K = \frac{FU - FI}{DELTA_K} + N_{K-1} + 1$$	1	99 (*DELTA* = 10 cps for all slopes)	250	1	7 (*DELTA* = 10 cps for all slopes)	127	200
PSDI	g^2/cps at the lower break frequency FI of the slope. This is a *real* variable. If g^2/cps is not known (which is usually the case for the first positive slope), read in 0.0. PSDI must have a decimal point.	0.0	0.1	0.1	0.0	0.04	0.04	0.07
PSDU	g^2/cps at the upper break frequency FU of the slope. This is a *real* variable and must have a decimal. If g^2/cps is not known (which is usually the case for the last negative slope), read in 0.0.	0.1	0.1	0.0	0.04	0.04	0.07	0.07
DB	This is a *real* variable (and therefore must contain a decimal point) and represents the slope in db/octave. Observe signs. Positive and zero slopes do not need a sign, but a negative slope must contain a *minus* sign. For zero slopes, read in 0.0.	24.0	0.0	−24.0	24.0	0.0	0.61	0.0
DELTA	This is a *real* variable representing the frequency incrementing over the slope (in cps). Different slopes along the PSD curve can have different incrementing.	10.0	10.0	10.0	10.0	10.0	10.0	10.0
FI	Lower break frequency of the slope (cps). This is a *real* variable.	10.0	20.0	1000.0	10.0	20.0	80.0	1280.0
FU	Upper break frequency of the slope. This is a *real* variable.	20.0	1000.0	2500.0	20.0	80.0	1280.0	2000.0

Table 2. Code-form sheet for the vibration test specifications of Fig. 3 and 4

IBM

FORTRAN CODING FORM

DATA FROM TABLE 1 | Punching Instructions | Page of

Program | Graphic | Card Form # * | Identification (73–80)

Programmer | Date | Punch

C FOR COMMENT

STATEMENT NUMBER (1–5)	Cont. (6)	FORTRAN STATEMENT (7–72)
$DATA		
3		1 1 3000 1 0.0 0.1 24.0 10.0 10.0 20.0
3		2 2 3000 99 0.1 0.1 0.0 10.0 20.0 1000.
3		3 100 3000 250 0.1 0.0 -24.0 10.0 10000. 2500.
K	K1	L M N PSD1 PSDU DB DELTA F1 FU
$DATA		
4		1 1 3000 1 0.0 0.04 24.0 10.0 10.0 20.0
4		2 2 3000 7 0.04 0.04 0.0 10.0 20.0 80.
4		3 8 3000 127 0.04 0.07 0.61 10.0 80.0 1280.
4		4 128 3000 200 0.07 0.07 0.0 10.0 1280. 2000.

* A standard card form, IBM electro 888157, is available for punching source statements from this form.

decimal points. The rest of the variables are defined as REAL and *must* have decimal points. Figures 3 and 4 will serve as examples. These Figures represent actual test requirements for a random-noise vibration test performed by the Environmental Laboratory at JPL. Both specifications were analyzed at 10-cycle increments starting at 10 cps. Every variable listed must appear as a number (integer or real) on *each* data card *in the order listed in Table 1.* Except for the variables L and N, the numbers associated with the variables can be written directly from the specification sheets (Fig. 3 and 4).

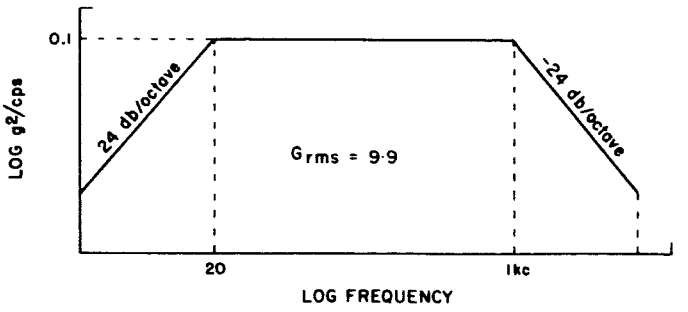

Fig. 3. Actual PSD test specifications used for analysis 1

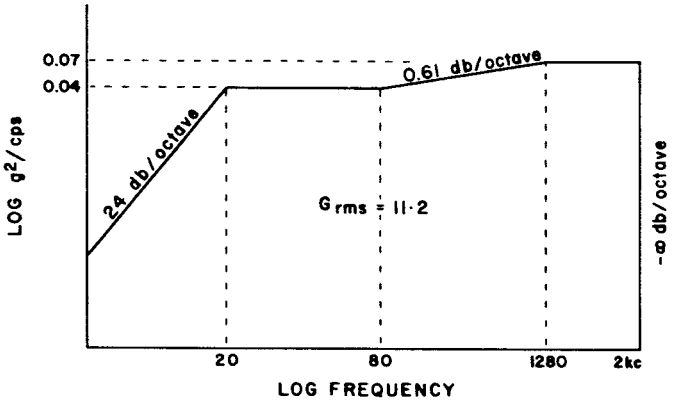

Fig. 4. Actual PSD test specifications used for analysis 2

Once the variables have been converted to numbers for a particular PSD analysis, it is necessary to punch them in the correct columns of the data cards. Usually the numbers are printed on a code-form sheet first, such as illustrated in Table 2. This form is given to a keypunch operator for punching. It is important that the numbers appear in the correct fields or columns of the code-form sheet. The first three rows of numbers (Table 2) represent the data cards for the PSD specification of Fig. 3; the second set of numbers is the data for the specifications of Fig. 4. In between the sets of data, columns representing the variables associated with the numbers are blocked off for reference purposes only. The numbers *must* start on the right hand side of each field. For example, the number representing K must *start* in column 4; K1 must start in column 9, etc. Note that the variables, K, K1, L, M, and N do not have decimals. In the computer program, these variables are defined as INTEGER type. Variables PSDl, PSDU, DB, DELTA, Fl, and FU *must* have decimals because they have been defined as REAL type variables in the sense of computer language. Notice that the variable K is in a field of four columns, but the variables K1, L, M, and N are in a field of five columns each. The other variables (the REAL's) are each contained within a field of eight columns.

VI. THE PROGRAM

Each of the three segments of the computer program is presented in Appendix B. This Report will not involve itself with the language of FORTRAN, and therefore, no attempt will be made to explain the FORTRAN statements. An attempt to understand someone else's computer program is an unpleasant task, even for the experienced programmer. To successfully use the program, it is only necessary to understand the variables listed in Table 1 and where to put the numbers associated with these variables on the code-form sheet. However, a word must be said about one set of statements in the MAIN part of the program ($IBFCT MAIN, Appendix B-1). Statements 91 and 92 (these are the sequence numbers at the extreme right) have to do with how much data will be printed on each page of the output. The data will appear single spaced, 50 lines to a page. Table 3 is a portion of the output from the specifications of Fig. 4. If the users computer installation off-line printer

Table 3. Computer output as calculated from the second segment of the computer program, referencing Fig. 4 test specifications

FREQUENCY	POWER SPECTRAL DENSITY
510.0	5.809E-02
520.0	5.832E-02
530.0	5.855E-02
540.0	5.877E-02
550.0	5.899E-02
560.0	5.920E-02
570.0	5.942E-02
580.0	5.963E-02
590.0	5.983E-02
600.0	6.004E-02
610.0	6.024E-02
620.0	6.044E-02
630.0	6.063E-02
640.0	6.083E-02
650.0	6.102E-02
660.0	6.121E-02
670.0	6.139E-02
680.0	6.158E-02
690.0	6.176E-02
700.0	6.194E-02
710.0	6.212E-02
720.0	6.230E-02
730.0	6.247E-02
740.0	6.264E-02
750.0	6.281E-02
760.0	6.298E-02
770.0	6.315E-02
780.0	6.331E-02
790.0	6.348E-02
800.0	6.364E-02
810.0	6.380E-02
820.0	6.396E-02
830.0	6.412E-02
840.0	6.427E-02
850.0	6.443E-02
860.0	6.458E-02
870.0	6.473E-02
880.0	6.488E-02
890.0	6.503E-02
900.0	6.518E-02
910.0	6.532E-02
920.0	6.547E-02
930.0	6.561E-02
940.0	6.575E-02
950.0	6.590E-02
960.0	6.604E-02
970.0	6.617E-02
980.0	6.631E-02
990.0	6.645E-02
1000.0	6.658E-02

will not print 50 single-spaced lines on a page, the number 50 must be reduced to whatever number of lines can be printed (single spaced), and the number 49 must be reduced to one less number than the number chosen in statement 91.

The MAIN part of the program is basic. That is, it will give the user the desired data in tabular form (Tables 3 and 4). It will print out what is read into the computer from the data cards (Table 5) so the user can check for human errors. The requirements of vibration laboratories vary. One might want the data punched on cards and plotted as well as listed in tabular form. In this case, the appropriate statements must be added to the MAIN program. In the case of the Environmental Laboratory, all PSD analyses are performed by digital techniques rather than by analog techniques, so cards are punched in the MAIN program and later used with another computer program to obtain spectra ratios.

Table 4. Computer output as calculated from the third segment of the computer program, referencing Fig. 4 test specifications (areas are g^2_{rms}); GRMS = total broadband acceleration

```
G SQUARED RMS

AREA( 1) =   8.87152770E-02

AREA( 2) =   2.40000000E 00

AREA( 3) =   7.18115510E 01

AREA( 4) =   5.04000000E 01

TOTAL RMS ACCELERATION

GRMS =   1.11669270E 01
```

Table 5. Input data printout, referencing Fig. 4 test specifications

K	K1	L	M	N	PSD1	PSDU	DB	DELTA	F1	FU
4	1	1	3000	1	0.00	0.04	24.00	10.00	10.0	20.0
4	2	2	3000	7	0.04	0.04	0.00	10.00	20.0	80.0
4	3	8	3000	127	0.04	0.07	0.61	10.00	80.0	1280.0
4	4	128	3000	200	0.07	0.07	0.00	10.00	1280.0	2000.0

VII. RESULTS

Figures 5 and 6 display the computer output for the test specifications of Fig. 3 and 4, respectively. The plotting routine to obtain Fig. 5 and 6 was inserted in the MAIN segment to verify the computer program.

The main concepts of this Report are illustrated in Fig. 7 and 8. These Figures represent the results of two separate vibration tests using the specifications of Fig. 5. The first test was controlled by a single accelerometer although acceleration data from three other accelerometers were recorded. An automatic equalizer was used to equalize the vibration systems. The second test, using the same test specimen, accelerometers, etc., was performed immediately after the first test, but used all four accelerometers for control. Again, an automatic equalizer was utilized along with a special accelerometer scanning device designed by the author. By merely looking at Fig. 7 and 8, certain important conclusions can be reached without having to analyze separate power-spectral-density plots.

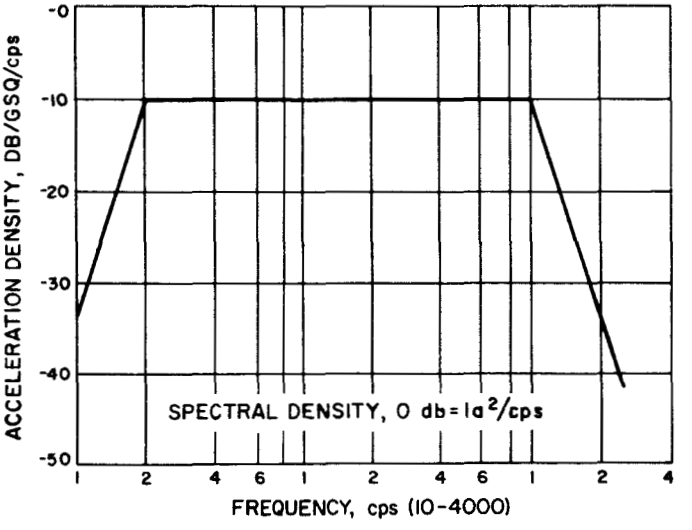

Fig. 5. Computer output for the test specifications of Fig. 3

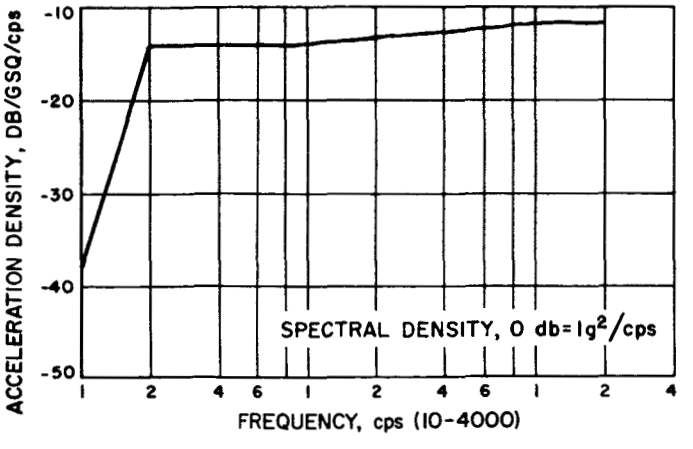

Fig. 6. Computer output for the test specifications of Fig. 4

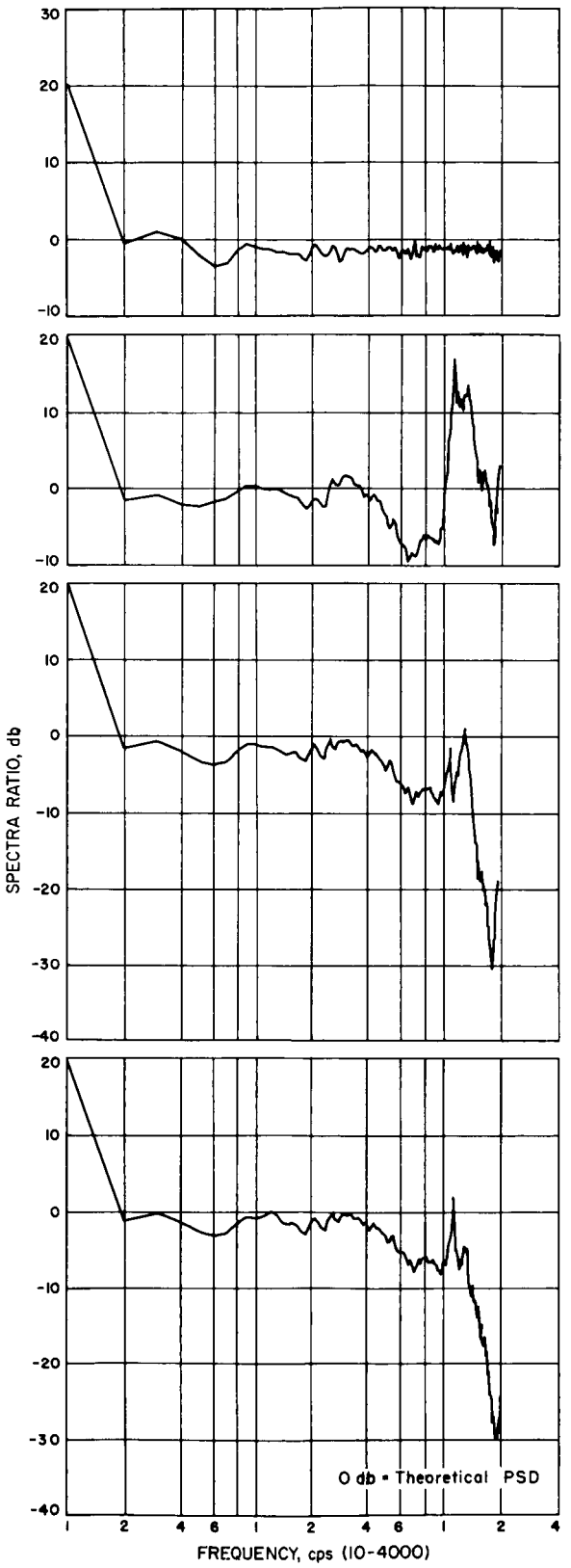

Fig. 7. Single-pt. spectra-ratio control; Ch-1, control accelerometer; 0 db, theoretical test spec., Fig. 4

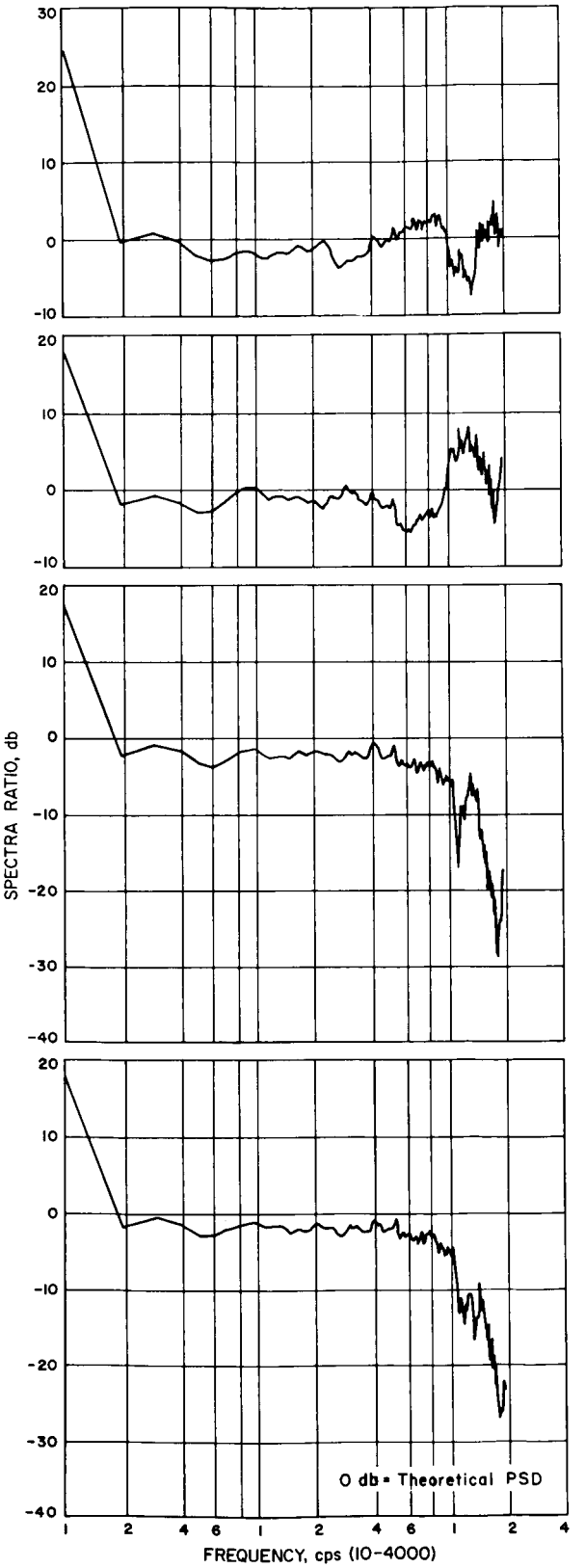

Fig. 8. Multiple-pt. spectra-ratio control (scan technique); 0 db, theoretical test spec., Fig. 4

VIII. CONCLUSION

A computer program has been presented with the idea that, perhaps, random-noise vibration test results can be simplified and made more meaningful for the test requester. The program can also be used to verify PSD test specifications, and setup automatic equalization equipment. An exact acceleration level for the shaped random noise can be calculated prior to the test.

Spectra ratios appear to simplify the interpretation of the test results; however, to synthesize a meaningful spectra ratio, it is necessary to have the exact acceleration densities of the ideal PSD spectrum for the vibration test at small frequency increments across the PSD spectrum. Spectra ratios are obtained when the ideal g^2/cps is ratioed against the vibration test g^2/cps. The results can be expressed in decibels as: $10 \log_{10} \times$ ratio.

Some vibration laboratories may not have elaborate computer programs to analyze random-noise vibration test data. The analyses can be performed using analog techniques. To obtain the spectra ratios, the results of the completed analog analysis can be ratioed against results obtained from the digital-computer program presented in this Report.

APPENDIX A

Equations

The following equations can be used to analyze power-spectral density curves. The use of these equations is valid for log-log plots and not for semi-logarithmic plots.

1. Positive Slopes

$$\text{Rolloff in db/octave} = 10 \log \frac{\text{value of PSD level at 2f}}{\text{value of PSD level at f}} \quad \text{(A-1)}$$

where PSD level is in units of g^2/cps and f is a frequency in cps somewhere in the interval over which the positive slope occurs. The frequency, 2f, may or may not occur over the slope interval. If 2f is outside the slope interval, the slope is extended to 2f so a PSD-level reading can be taken.

Acceleration density is expressed as:

$$g^2rms = \left[\frac{G_0 f_2}{k_1} 1 - \left(\frac{f_1}{f_2}\right)^{k_1}\right] \quad \text{(A-2)}$$

where

G_0 = PSD level at f_2 in g^2/cps

f_1 = low-frequency limit of the slope in cps

f_2 = high-frequency limit of the slope in cps

$k_1 = D/3 + 1$

where

D = rolloff in db/octave from Eq. (A-1)

2. No Slope

$$\text{Rolloff in db/octave} = 0$$

Acceleration density is expressed as:

$$g^2 = G_0(f_2 - f_1) \quad \text{(A-3)}$$

where

G_0 = PSD level at f_1 in g^2/cps

f_1 = low-frequency limit in cps over the interval

f_2 = high-frequency limit in cps over the interval

3. Negative Slopes

$$\text{Rolloff in db/octave} = 10 \log \frac{\text{value of PSD level at f}}{\text{value of PSD level at 2f}} \quad \text{(A-4)}$$

Acceleration density is expressed as:

$$g^2rms = \frac{G_0 f_1}{k_2}\left[1 - \frac{1}{\left(\frac{f_2}{f_1}\right)^{k_2}}\right] \quad \text{(A-5)}$$

where

G_0 = PSD level at f_1 in g^2/cps

f_1 = low-frequency limit in cps over the interval

f_2 = high-frequency limit in cps over the interval

$k_2 = D/3 - 1$

where

D = rolloff in db/octave from Eq. (A-4)

4. Total rms Acceleration

$$g_{rms}\ \text{TOTAL} = (g_1^2\ rms + g_2^2\ rms + g_3^2\ rms + \cdots + g_n^2\ rms)^{1/2} \tag{A-6}$$

where g_n^2 is the mean-square value of acceleration as represented by the area under the nth slope.

APPENDIX B

Computer Program Segments

1. The main segment of the computer program

```
$IBFTC MAIN
C       COMPUTER PROGRAM BY P. CHAPMAN, ENVIRONMENTAL AND DYNAMIC TESTING      001
C       LABORATORY, JET PROPULSION LABORATORY.                                 0*2
        DIMENSION PSD(3000),F(3000),GSQ(9),PAREA(969)                          005
        WRITE (6,29)                                                             6
   29 FORMAT (1H1,7X,1HK,10X,2HK1,8X,1HL,10X,1HM,10X,1HN,9X,4HPSDI,6X,         007
     1    4HPSDU,7X,2HDB,6X,5HDELTA,7X,2HF1,8X,2HFU////)                        008
    1 READ (5,30) K,K1,L,M,N,PSD1,PSDU,DB,DELTA,F1,FU                          010
   30 FORMAT (I4,4I5,6F8.0)                                                    015
C                                                                              20
        WRITE (6,35) K,K1,L,M,N,PSD1,PSDU,DB,DELTA,F1,FU                       025
   35 FORMAT (1H0,3X,5(I6,5X),4(F7.2,3X),2(F8.1,2X)/////)                      030
C                                                                              35
        CALL PWR (L,M,N,PSDU,F1,FU,DB,DELTA,F,PSD)                             040
        CALL GRMS (K,K1,F1,FU,PSD1,PSDU,DB,GSQ)                                045
C                                                                              50
        IF (K1.LT.K) GO TO 1                                                   55
        SUM = 0.0                                                              60
C                                                                              65
        DO 4 J= 1,K                                                            70
    4 SUM = SUM + GSQ(J)                                                       75
C                                                                              80
        GRM = SQRT(SUM)                                                        85
C                                                                              90
        DO 100 IBEG =1,N,50                                                    91
        IEND=IBEG+49                                                           92
        IF(IEND.GT.N) IEND=N                                                  093
  100 WRITE (6,40)(F(I),PSD(I),I=IBEG,IEND)                                   095
   40 FORMAT (1H1,26X,9HFREQUENCY,21X,22HPOWER SPECTRAL DENSITY///(1H ,        100
     1    26X,0PF9.1,28X,1PE10.3))                                            101
C                                                                             105
        WRITE (6,45) (K1,GSQ(K1),K1=1,K)                                      110
   45 FORMAT(1H1,20X,13HG SQUARED RMS///(1H0,20X,5HAREA(,I2,4H) =  ,          115
     1    1PE16.8)///)                                                        116
C                                                                             120
        WRITE(6,50) GRM                                                       125
   50 FORMAT(1H0,20X,22HTOTAL RMS ACCELERATION///1H0,20X,7HGRMS = ,           130
     1    1PE16.8)                                                            131
C                                                                             135
        STOP                                                                  140
        END                                                                   141
```

2. The second segment of the computer program

```
$IBFTC SUBPSD
      SUBROUTINE PWR (L,M,N,PSDU,F1,FU,DB,DELTA,F,PSD)                  004
      DIMENSION PSD(M),F(M)                                               8
      F(L)= F1                                                           12
      IF(DB) 10,5,1                                                      16
    1 ARG1=(F(L)/FU)**(0.3322*DB)                                        020
      PSD(L)=PSDU*ARG1                                                   24
      DO 2 I=L,N                                                         28
      ARG2=F(I)+DELTA                                                    32
      F(I+1)=ARG2                                                        36
      ARG3=PSD(I)*((F(I+1)/F(I))**(0.3322*DB))                           040
    2 PSD(I+1)=ARG3                                                      44
      RETURN                                                             48
    5 DO 6 I=L,N                                                         52
      ARG4=F(I)+DELTA                                                    56
      F(I+1)=ARG4                                                        60
      ARG5=PSD(I)                                                        64
    6 PSD(I+1)=ARG5                                                      68
      RETURN                                                             72
   10 DO 11 I=L,N                                                        76
      ARG6=F(I)+DELTA                                                    80
      F(I+1)=ARG6                                                        84
      ARG7=PSD(I)*(F(I+1)/F(I))**(0.3322*DB)                             088
   11 PSD(I+1)=ARG7                                                      92
      RETURN                                                             96
      END                                                                64
```

3. The third segment of the computer program

```
$IBFTC RMS
      SUBROUTINE GRMS (K,K1,F1,FU,PSD1,PSDU,DB,GSQ)                     003
      DIMENSION GSQ(K)                                                    6
      IF (DB)  100,200,300                                                9
  100 IF(DB .EQ.3.) GO TO 400                                            12
      S=(ABS(DB)/3.)-1.                                                  15
      GSQ(K1) = (PSD1 * F1)/S * (1.-(1./((FU/F1)**S)))                   018
      RETURN                                                             21
  200 GSQ(K1)= PSD1 *(FU - F1)                                           24
      RETURN                                                             27
  300 T= (ABS(DB)/3.) + 1.                                               30
      GSQ(K1)= ((PSDU *FU)/T) * (1.-((F1/FU)**T))                        033
      RETURN                                                             36
  400 GSQ(K1) =0.0                                                       39
      WRITE(6,10) K1                                                     42
   10 FORMAT(1H0,36HNEGATIVE 3DB SLOPE WILL NOT COMPUTE.,3X,             045
     1    4HGSQ(I3,6H) =0.0)                                            *48
      RETURN                                                             51
      END                                                                54
```