NASA CONTRACTOR REPORT 166400

Ensemble Averaging of Acoustic Data

P. K. Stefanski

CONTRACT A88065B
September 1982

NASA

NASA CONTRACTOR REPORT 166400

Ensemble Averaging of Acoustic Data

P. K. Stefanski
Beam Engineering, Inc.
Sunnyvale, California

Prepared for
Ames Research Center
under Purchase Order No. A88065B

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

N82-34102#

ENSEMBLE AVERAGING OF ACOUSTIC DATA

Prepared for

NASA-Ames Research Center
Moffett Field, California

Under

Purchase Order No. A880658

by

Patricia K. Stefanski
Beam Engineering, Inc.

SUMMARY

This report documents a computer program called Ensemble Averaging of Acoustic Data. The program samples analog data, analyzes the data, and displays them in the time and frequency domains. Hard copies of the displays are the program's output. The documentation includes a description of the program and detailed user instructions for the program. This software was developed for use on the Ames 40x80-foot wind tunnel's Dynamic Analysis System consisting of a PDP-11/45 computer, two RK05 disk drives, a Tektronix 611 keyboard/display terminal, an FPE-4 Fourier Processing Element, and an analog-to-digital converter.

This software was developed by Beam Engineering, Inc. under purchase order no. A880658 to NASA-Ames Research Center with Marianne Mosher as contract monitor.

LIST OF SYMBOLS

$D(\omega_j)$	Array of data in frequency domain
$F(\omega_j)$	Array of power spectrum data after bandwidth has been corrected
$G(\omega_j)$	Array of data from single-sided transform with correct amplitude
$\bar{G}(\omega_j)$	Complex conjugate of array
$H(\omega_j)$	Array of power spectrum data
j	Index for frequency
K	Sample index
KMAX	Number of averages
N	Number of samples
n	Index for time array
P(n)	Array of data in pressure units of N/m^2
S(n)	Average sequence of sampled time data
$S_K(n)$	One sequence of sampled time data
SPL	Sound pressure level
VCAL	Calibration constant which converts volts to N/m^2
ω_j	Frequency

INTRODUCTION

The program "Ensemble Averaging of Acoustic Data" samples analog acoustic data, analyzes it, and displays it in both the time and frequency domains. Three graphs - a time history graph, a power spectrum graph, and a sound pressure level graph - are produced. Hard copies of these displays are the program's output.

This program was written for use on the Ames 40x80-foot wind tunnel's Dynamic Analysis System. The output is written to a Tektronix 611 keyboard/display terminal. The data is input from a playback tape recorder connected to the Dynamic Analysis System. This program can also use on-line data. For this report the descriptions are given as if data is played back through a tape recorder.

This report includes a description of the program which explains how the program analyzes the data and instructions for operating the program. The program is written in Fortran and uses Fortran subroutines for data gathering and processing. These subroutines are found in Reference 1. The report also contains a general flowchart for the program in Appendix A, a flowchart of the code in Appendix B, a listing of the program in Appendix C, and examples of typical running sequences in Appendix D.

There are two versions of this program: Time Frequency 2 (TF2) and Time Frequency 3 (TF3). They are identical except for the way the bandwidth is corrected for the power spectrum graph and they write the calibration data to different files. The differences in TF2 and TF3 will be explained fully in the Program Description section and the User Instruction section. This report is valid for both versions.

I would like to express my thanks to Marianne Mosher for her help and guidance in the development of this software.

PROGRAM DESCRIPTION

The following is an explanation of how the program "Ensemble Averaging of Acoustic Data" analyzes analog acoustic data. The flowchart in Appendix A illustrates the process. It would be beneficial to the reader to refer to Appendix A while reading this section.

The first step is to calibrate each channel of microphone data. This is accomplished by first sampling the analog data from a channel specified by Thumbwheel A. (Thumbwheel A is on the front panel of the Dynamic Analysis System) The analog data are digitized by the Dynamic Analysis System's converters and a sample length of 2048 data points are taken. The sampling frequency used is 1024 Hz with a low-pass cut-off frequency of 500 Hz. Data acquisition is started on receipt of an electronic trigger signal. One average is taken of the calibration data.

Next, the calibration data are converted from the time domain to the frequency domain by a single-sided Direct Fourier Transform. The output values are calculated by the following formula:

$$D(\omega_j) = 1/N \ \Sigma_{n=0}^{N-1} \ [P(n)\exp[-i2\pi\omega_j \ n/N] \times [\sqrt{8/3} \times 1/2 \times (1-\cos 2\pi n/N)]$$

where j - 0 to N/2. The Normalized Hanning operation is performed in the frequency domain after the direct transform is computed (Reference 1).

The correct amplitude from the single-sided transform is then computed by multiplying the data by a factor of 2

$$G(\omega_j) = (2) \times (D(\omega_j))$$

Then the transformed spectrum is converted to a power spectrum by multiplying the spectrum with its complex conjugate

$$H(\omega_j) = [G(\omega_j)] \times [\bar{G}(\omega_j)]$$

Next, the calibration constant is computed. The calibration data points 400 to 600, corresponding to the frequencies 400 to 600 Hz, are summed and the square root is taken of this sum. This value is then multiplied by the calibration factor which is input by the user. The calibration factor is determined by the type of calibrator and microphone used. The standard calibration factor is 1.0. The value, VCAL, is the calibration constant

$$VCAL = [\Sigma_{j=400}^{600} H(\omega_j)]^{1/2} \times [\text{calibration factor}]$$

VCAL is used to compute the pressure constant. This sequence continues until all the microphones have been calibrated. The program is then ready to sample the analog acoustic data.

The analog data are sampled as in the calibration sequence. The digitizing rate, sample frequency, gain, and the number of averages to be used are determined by user input. Anti-aliasing filters within the Dynamic Analysis System use a cut-off frequency that is determined by the sample frequency chosen. Data acquisition starts on receipt of an electronic trigger signal for each sample.

The data are then averaged in the time domain. The sampled waveforms are added up point by point and then divided by the number of averages

$$S(n) = \Sigma_{K=1}^{KMAX} \ S_K(n)/Kmax$$

The data are converted to pressure units of N/m^2 by multiplying the data by the computed pressure constant

$$P(n) = ((10^{(-gain/20)}) \times 31.7/VCAL) \times S(n)$$

The analyzed time history is then displayed on the Tektronix screen.

The data are converted from the time domain to the frequency domain, the amplitude is corrected, and the transformed spectrum is converted to a power spectrum by the same methods used in the calibration process. At this point the correct bandwidth is computed. The power spectrum data points are summed together by the following formula:

If TF2 is used, $F(\omega_j) = H(\omega_{2j-1}) + H(\omega_{2j})$ for $j = 1$ to $n/2$

If TF3 is used, $F(\omega_j) = H(\omega_{3j-2}) + H(\omega_{3j-1}) + H(\omega_{3j})$ for $j = 1$ to $n/3$

where n is the number of frequencies in the spectrum or half the number of points sampled. Summing the data points changes the bandwidth by a factor of two if TF2 is used and by a factor of three if TF3 is used. The power spectrum graph is then displayed on the Tektronix screen.

The sound pressure level graph is displayed on a log scale by the following formula:

$$SPL = 10\log_{10}[F(\omega_j)/(.00002)^2]$$

This sequence is repeated for all data to be analyzed until the user exits the program.

USERS' INSTRUCTIONS

The following is an explanation of how to operate the program "Ensemble Averaging of Acoustic Data" on the Dynamic Analysis System. Appendix D contains an example of a typical operating sequence. It would be beneficial to the reader to refer to Appendix D while reading this section.

The operator must load the disk pack labeled "TF2 and TF3" onto the RK05 drive and boot the Dynamic Analysis System. The operator then initializes the program by entering the following command:

.R TF2 <cr>

Once this command has been entered the program automatically reads the calibration data from the file VCAL2.DAT and stores it in memory. If the user wishes to use the TF3 version, R TF3 is entered and the calibration data are then read from the file VCAL3.DAT. The program will then begin prompting the operator for input parameters. The operator enters the desired values as described below.

ENTER TEST NUMBER NNN <cr>

The operator enters the test number of the analog acoustic data to be analyzed. The test number can range from 1 to 999. Entering a zero or a carriage return will cause the program to terminate.

ENTER RUN NUMBER NNN <cr>

The operator enters the run number of the data to be analyzed. The run number entered can range from 1 to 999. Entering a zero or a carriage return signals the end of a test to the program which then prompts for the next test number.

ENTER POINT NUMBER NNN <cr>

The operator enters the point number of the data to be analyzed. The point number can range from 1 to 999. Entering a zero or a carriage return will cause the program to prompt the next run number.

WHAT TYPE OF CALIBRATION? O OR N (NO <cr>)

The operator enters which type of calibration data are used, old or new, by entering an O or N respectively. No carriage return is necessary. If the operator enters an O, the old calibration data are used and the operator is then prompted for the voltage code. If an N is entered, the following two prompts appear on the screen.

ENTER NUMBER OF MICROPHONE TO CALIBRATE <cr>

The operator sets the appropriate channel on Thumbwheel A and then enters the microphone number. ***Before entering a carriage return the user turns on the tape recorder so the proper calibration signal is present.*** The microphone number can range from 1 to 99.

WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE XX (REAL INPUT) <cr>

The operator enters the calibration factor for the microphone just calibrated. The standard calibration factor is 1.0. The program then prompts for microphone, and calibration factor continues until the operator enters a zero or a carriage return in response to the prompt for the microphone number. Once this occurs it indicates to the program that the calibration process is complete. The new calibration data are then written to the appropriate file and the next prompt then appears on the screen.

ENTER VOLTAGE CODE <cr>

The voltage code specifies the maximum analog input signal level that is allowed while sampling the analog data. The voltage chosen should be higher then the maximum peak voltage of the signal. The voltage code can range from 1 to 7 and corresponds to the volts used as specified in the following table.

VOLTS	VOLTAGE CODE
± .125	1
± .25	2
± .5	3
±1.0	4
±2.0	5
±4.0	6
±8.0	7

Entering a zero or a carriage return will cause the program to prompt for the next point number.

ENTER SAMPLE FREQUENCY CODE <cr>

The sample frequency code entered specifies the sampling frequency of the analog to digital converter. The sample frequency code can range from 1 to 7. The sample frequency code also specifies the cut-off frequency to be used by the anti-alising filters according to the following table.

SAMPLE FREQUENCY	CUT-OFF FREQUENCY	SAMPLE FREQUENCY CODE
51.2	20 Hz	1
204.8	100 Hz	2
512.0	200 Hz	3
2048.0	1 KHz	4
5120.0	2 KHz	5
20480.0	10 KHz	6
51200.0	20 KHz	7

Entering a zero or a carriage return will cause the program to prompt for the voltage code.

ENTER SAMPLE CODE <cr>

The sample code entered specifies the number of analog data points to be acquired and stored per frame and per channel. The sample code can range from 1 to 4 and corresponds to the number of samples taken as specified in the following table.

NUMBER OF SAMPLES	SAMPLE CODE
128	1
512	2
1024	3
2048	4

Entering a zero or a carriage return will cause the program to prompt for the voltage code.

ENTER MICROPHONE NN <cr>

The microphone number entered is the microphone number of the data to be graphed. The microphone number can range from 1 to 99. Entering a zero or a carriage return will cause the operator to be prompted for the number of samples.

ENTER GAIN (REAL INPUT) <cr>

The gain must be entered as a real number. Entering a 99.0 will cause the program to prompt for the microphone number.

GAIN IS XXX Y OR N (NO <cr>)

The operator checks if the correct gain was entered. If the gain entered was not correct, the operator enters an N. The program then prompts for the gain. If the correct gain was entered, the operator enters a Y and the program continues with the following prompt.

DATA ACQUISITION STARTS WHEN <cr> IS ENTERED
ENTER NUMBER OF AVERAGES NNN <cr>

The number of averages entered is the number of averages to be taken of the waveforms. ***Before the carriage return is given the operator turns on the tape recorder so the proper signal is present.*** The number of averages can range from 1 to 999. Entering a zero or a carriage return will cause the program to prompt for the gain.

At this point the time history and spectra graphs are displayed on the Tektronix screen. The display will stay on the screen until a carriage return is entered. Before a carriage return is entered a hard copy of the display must be made since there is no way to back up to the previous display once the carriage return has been entered. After the sound pressure level graph is displayed and the operator has entered a carriage return the program prompts for the next microphone number. The program will continue until a zero or a carriage return is entered in response to the prompt for the test number.

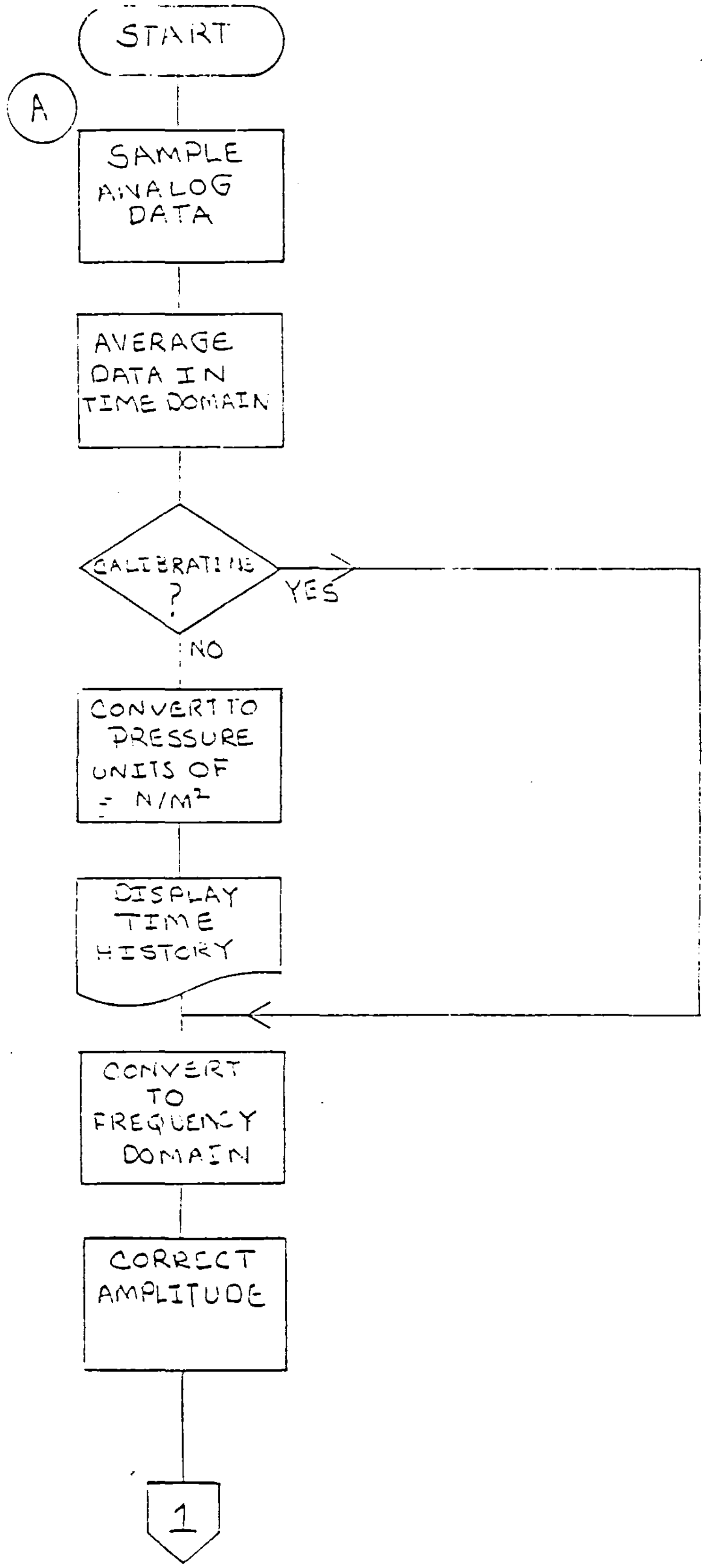
Various error messages may appear on the Tektronix screen while running this program. If this occurs, the user should refer to Reference 1.

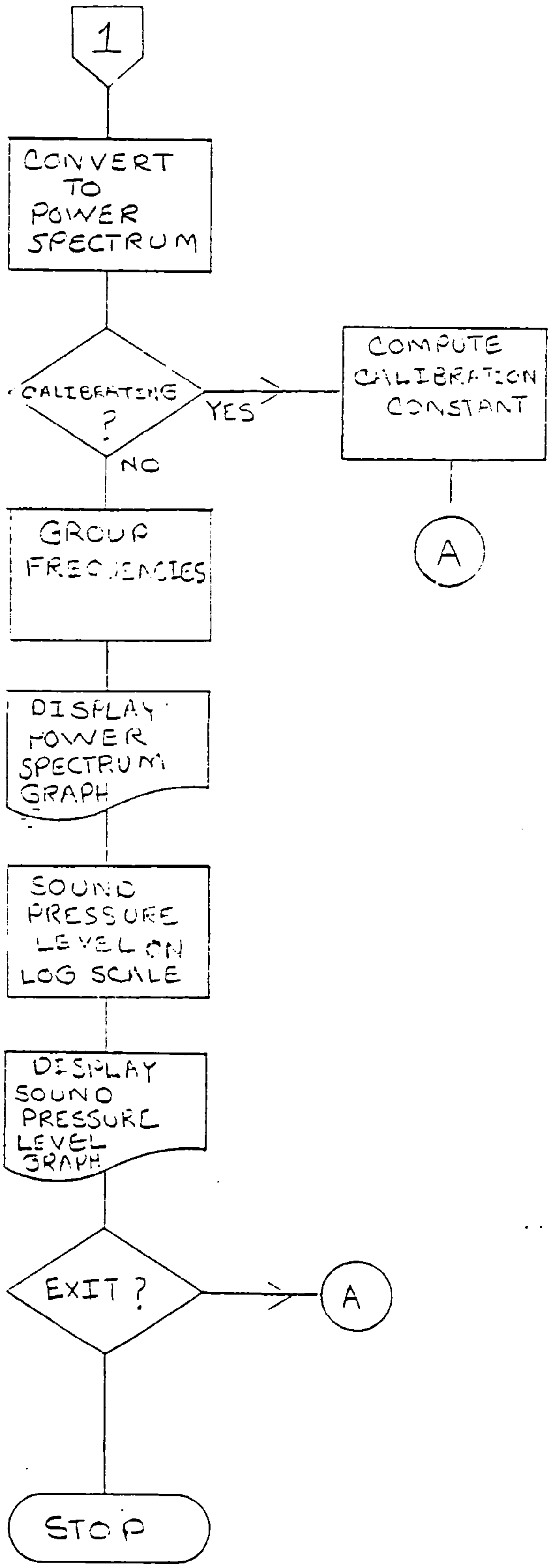
REFERENCE

1. Reference Manual TSALF Time Series Analysis Library - Fortran (for RT-11 Fortran System), GenRad Time/Data Division, 1976.

APPENDIX A

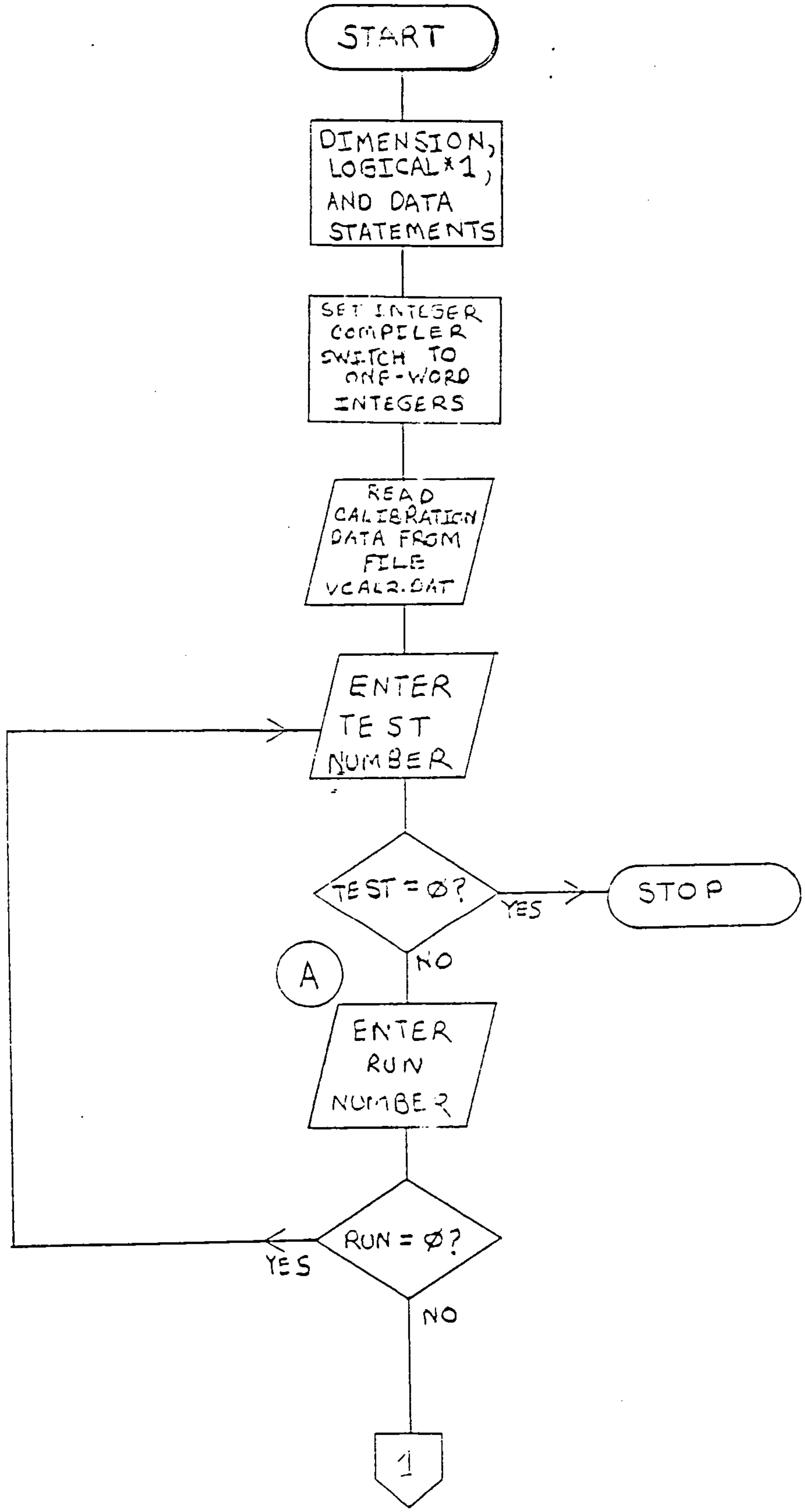
GENERAL FLOWCHART

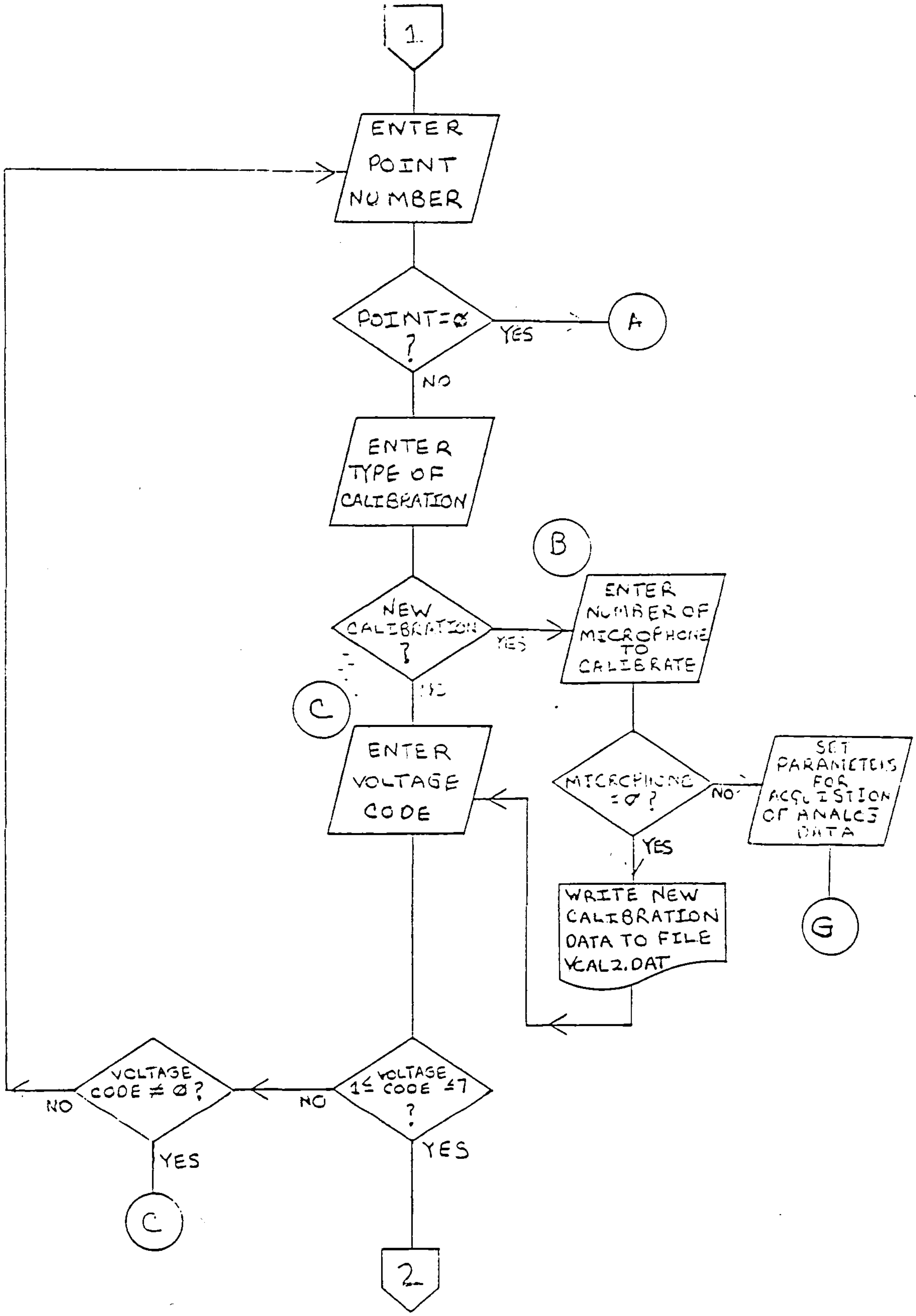

START
A
SAMPLE ANALOG DATA
AVERAGE DATA IN TIME DOMAIN
CALIBRATING ?
YES
NO
CONVERT TO PRESSURE UNITS OF N/M²
DISPLAY TIME HISTORY
CONVERT TO FREQUENCY DOMAIN
CORRECT AMPLITUDE
1

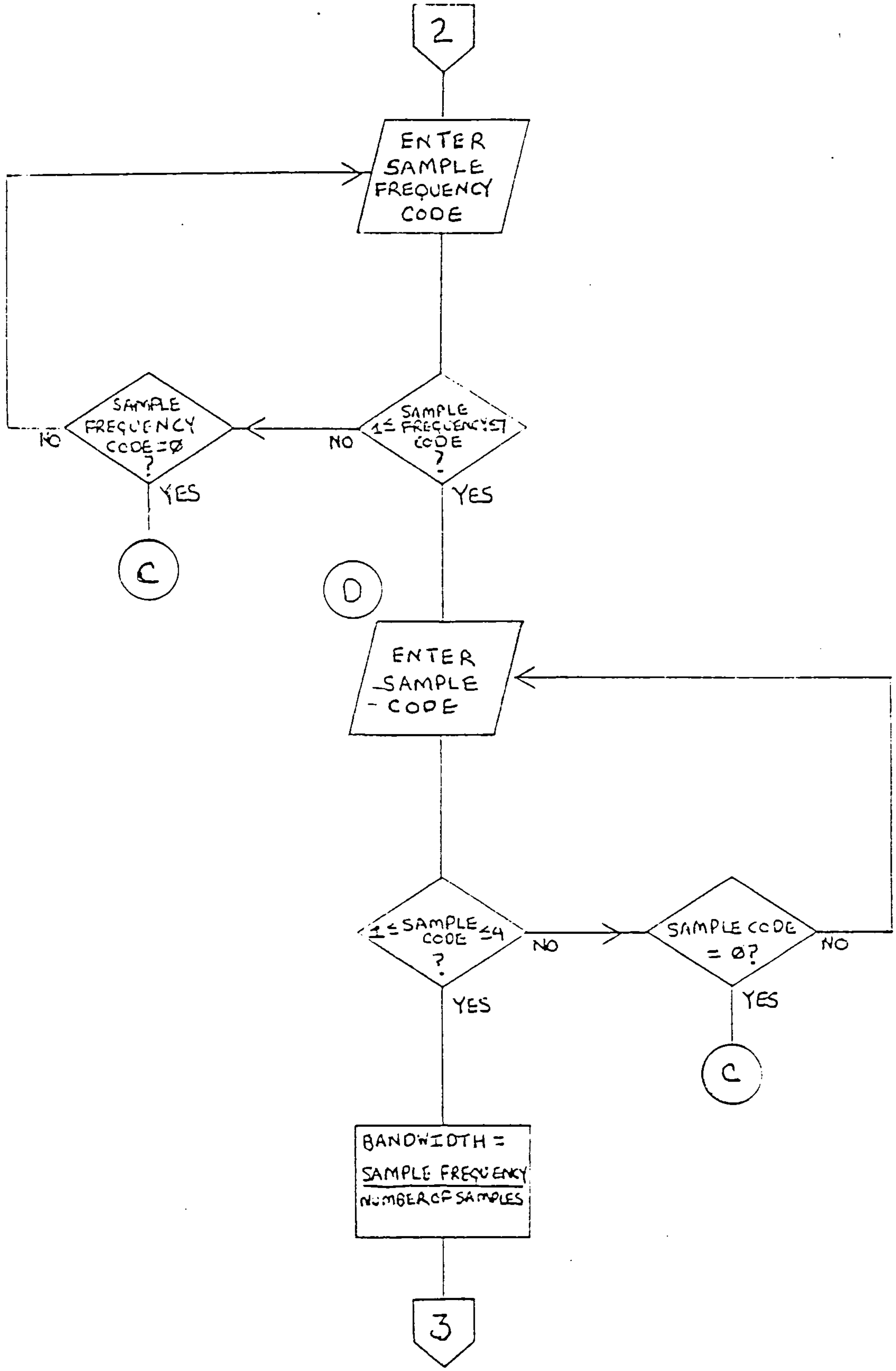

1
CONVERT TO POWER SPECTRUM
CALIBRATING ?
YES
NO
COMPUTE CALIBRATION CONSTANT
A
GROUP FREQUENCIES
DISPLAY POWER SPECTRUM GRAPH
SOUND PRESSURE LEVEL ON LOG SCALE
DISPLAY SOUND PRESSURE LEVEL GRAPH
EXIT ?
A
STOP

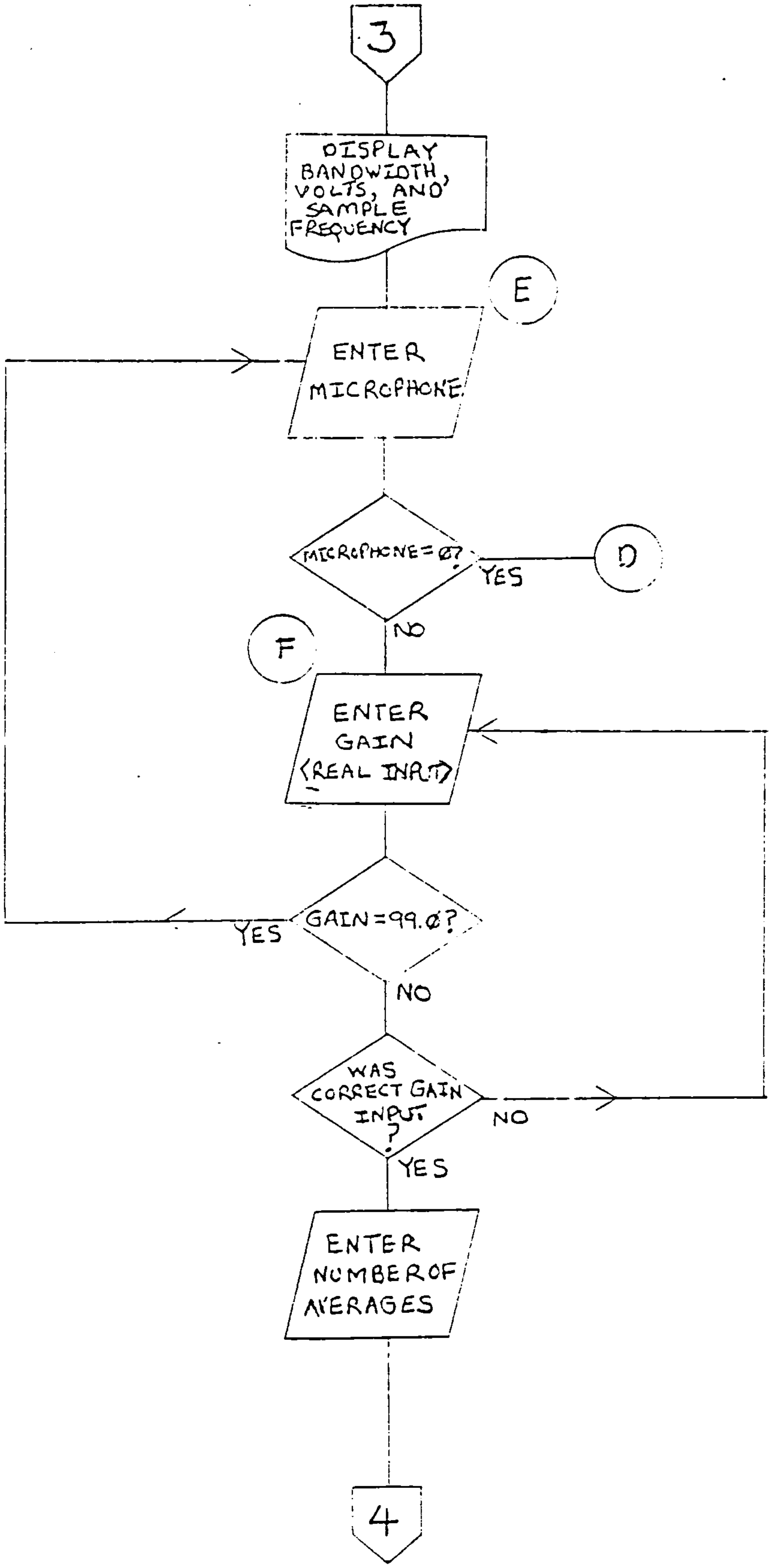
APPENDIX B

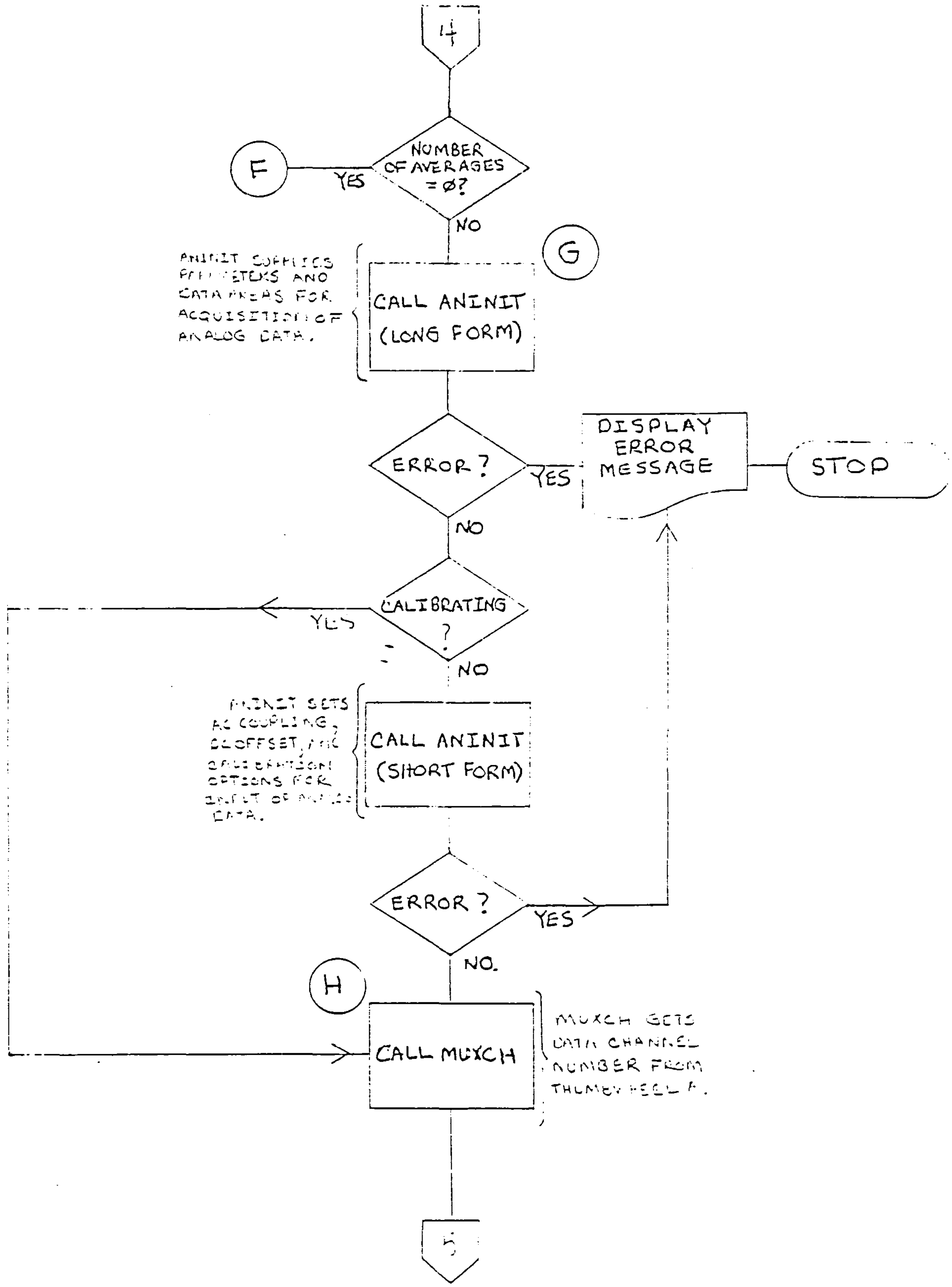
FLOWCHART OF CODE

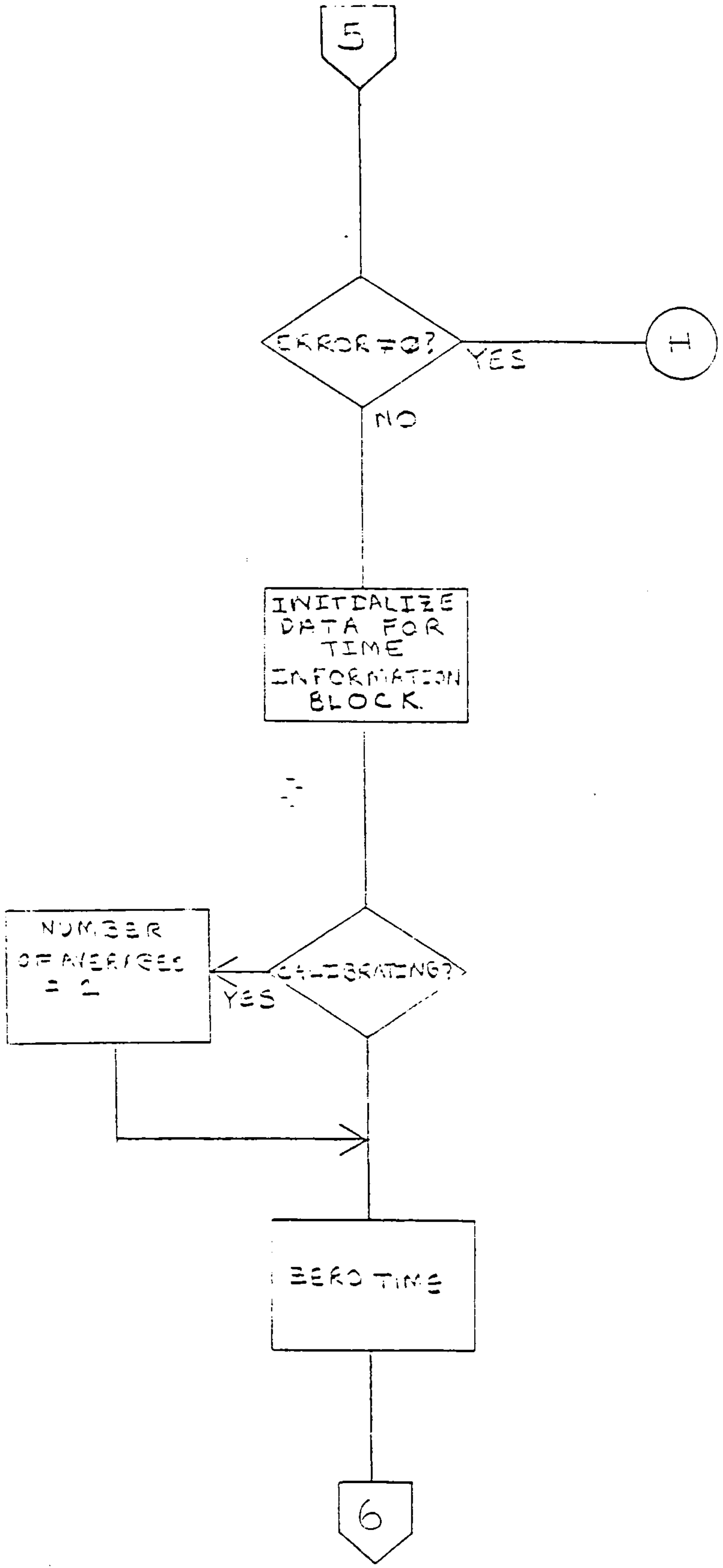

START
DIMENSION, LOGICAL*1, AND DATA STATEMENTS
SET INTEGER COMPILER SWITCH TO ONE-WORD INTEGERS
READ CALIBRATION DATA FROM FILE VCAL2.DAT
ENTER TEST NUMBER
TEST = Ø?
YES
STOP
NO
A
ENTER RUN NUMBER
RUN = Ø?
YES
NO
1

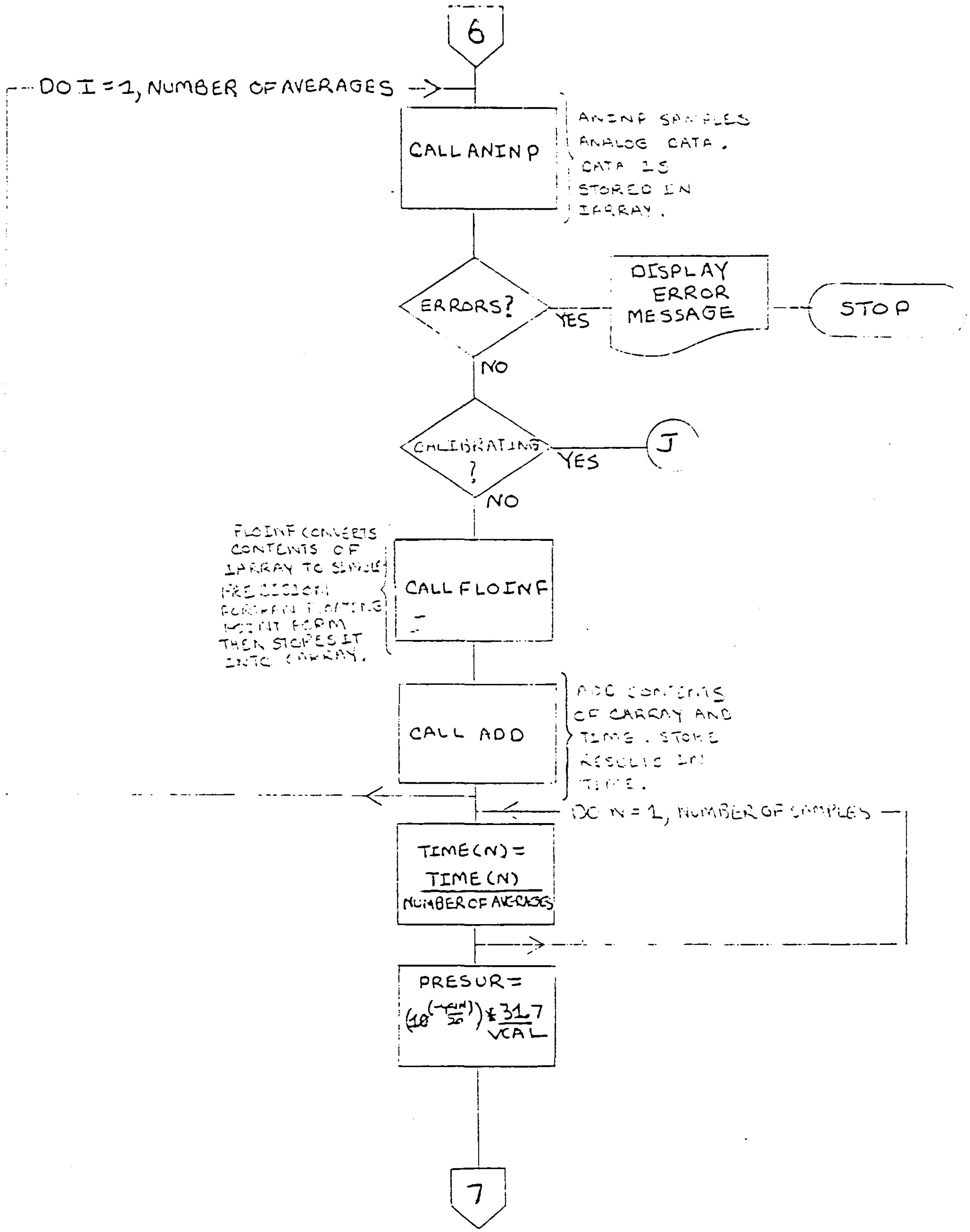

1
ENTER POINT NUMBER
POINT=Ø ?
YES
A
NO
ENTER TYPE OF CALIBRATION
B
NEW CALIBRATION ?
YES
C
ENTER NUMBER OF MICROPHONE TO CALIBRATE
ENTER VOLTAGE CODE
MICROPHONE =Ø ?
NO
SET PARAMETERS FOR ACQUISITION OF ANALOG DATA
YES
WRITE NEW CALIBRATION DATA TO FILE VCAL2.DAT
G
VOLTAGE CODE ≠ Ø ?
NO
1≤ VOLTAGE CODE ≤7 ?
NO
YES
YES
C
2

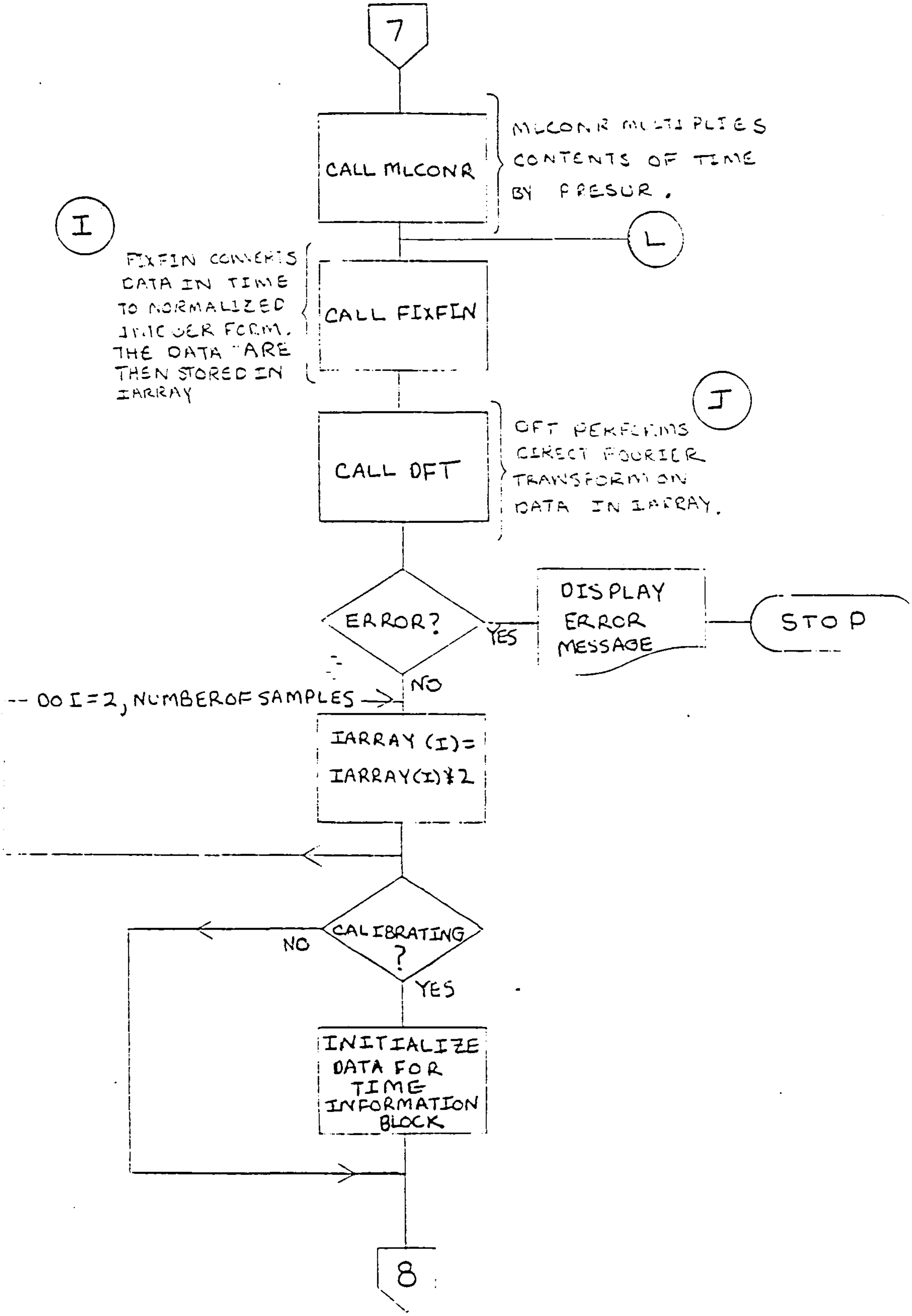

2
ENTER SAMPLE FREQUENCY CODE
1≤ FREQUENCY≤7 SAMPLE CODE ?
NO
SAMPLE FREQUENCY CODE=Ø ?
NO
YES
C
YES
D
ENTER SAMPLE CODE
1≤SAMPLE CODE≤4 ?
NO
SAMPLE CODE = Ø?
NO
YES
C
YES
BANDWIDTH = SAMPLE FREQUENCY / NUMBER OF SAMPLES
3

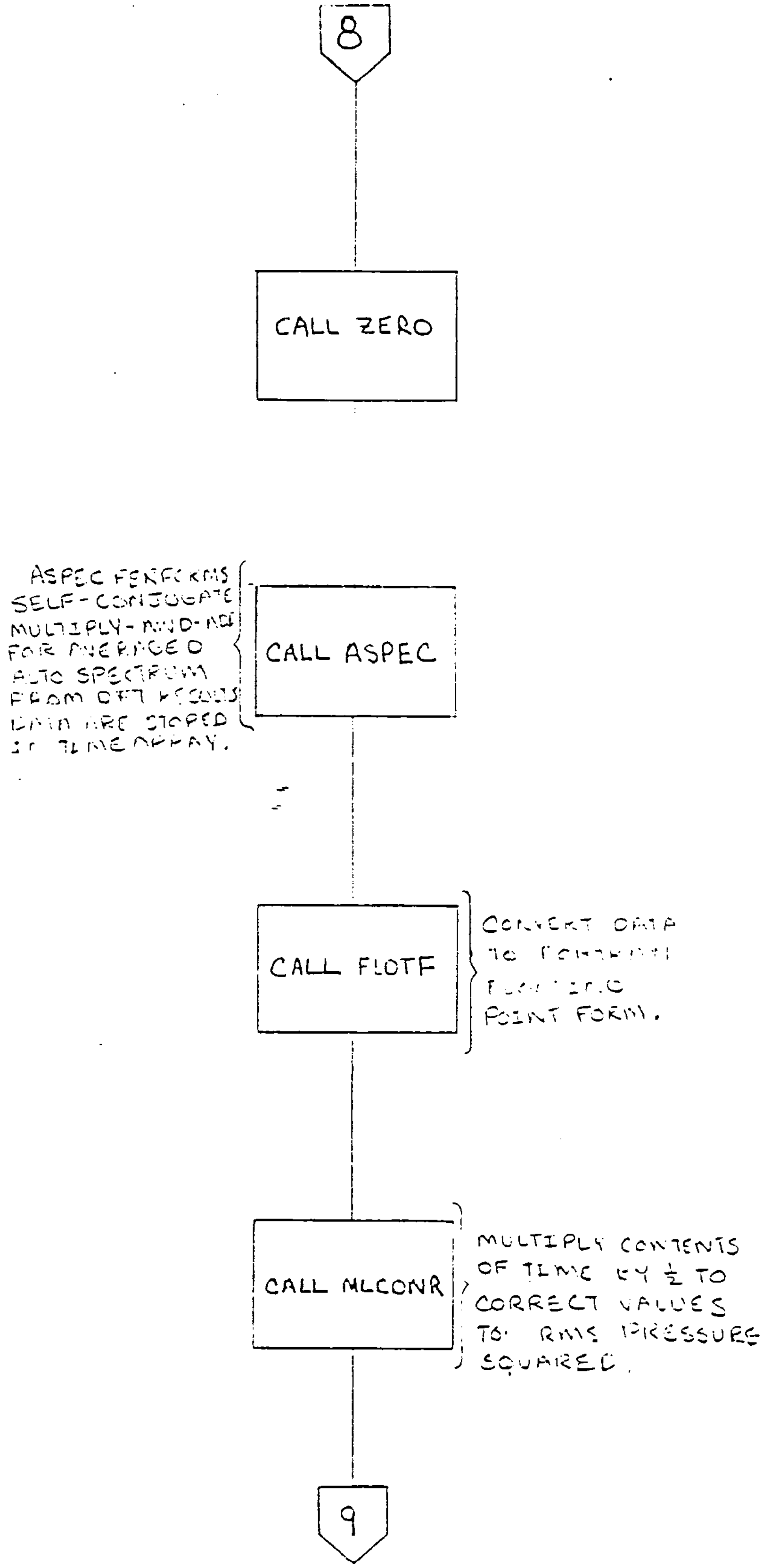

3
DISPLAY BANDWIDTH, VOLTS, AND SAMPLE FREQUENCY
E
ENTER MICROPHONE
MICROPHONE=Ø?
YES
D
NO
F
ENTER GAIN (REAL INPUT)
GAIN=99.Ø?
YES
NO
WAS CORRECT GAIN INPUT ?
NO
YES
ENTER NUMBER OF AVERAGES
4

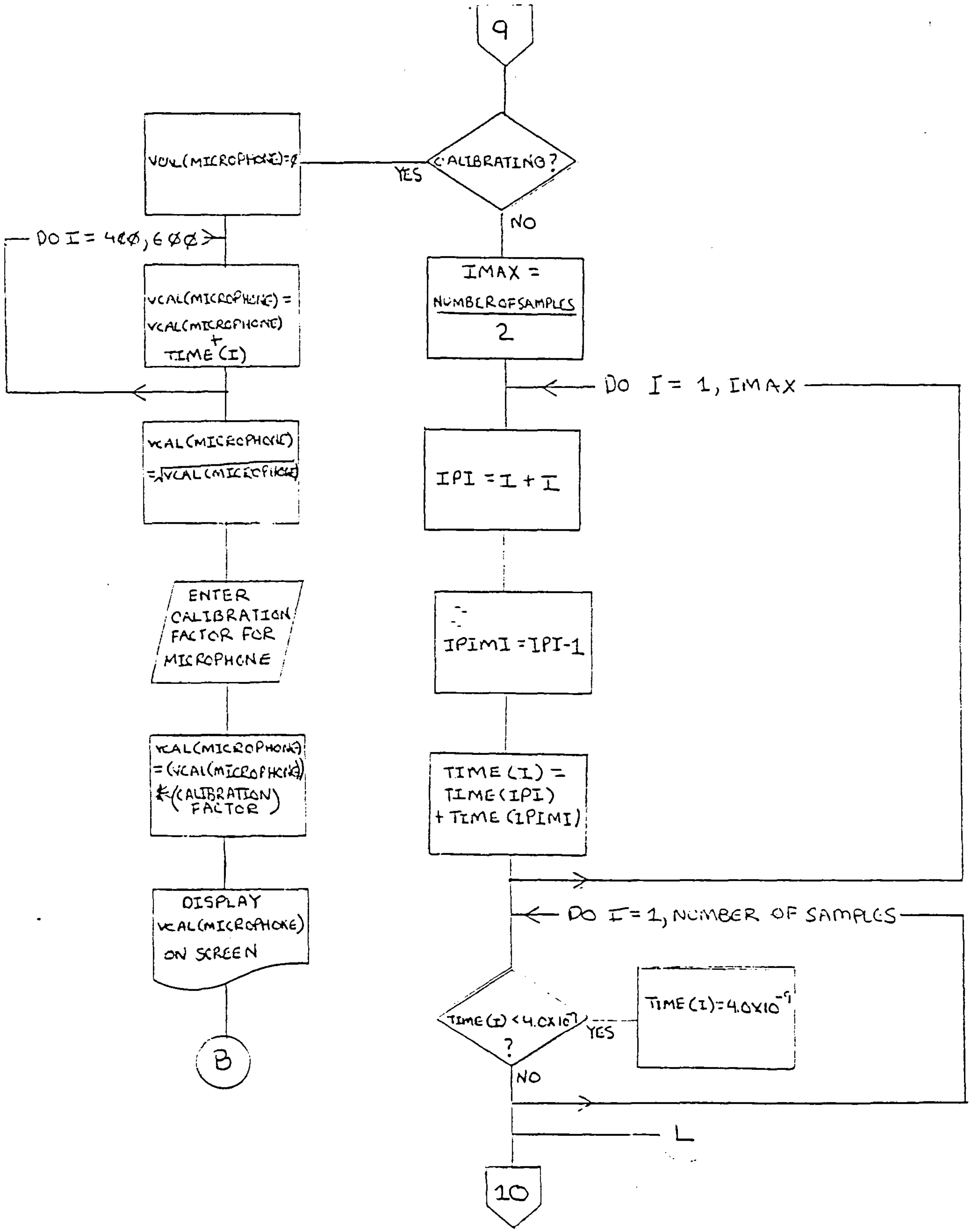

4
NUMBER OF AVERAGES = Ø?
F
YES
NO
G
ANINIT SUPPLIES
AND
FOR
ACQUISITION OF
ANALOG
CALL ANINIT (LONG FORM)
ERROR?
YES
DISPLAY ERROR MESSAGE
STOP
NO
CALIBRATING ?
YES
NO
SETS
COUPLING,
OFFSET,
OPTIONS FOR
DATA.
CALL ANINIT (SHORT FORM)
ERROR?
YES
NO.
H
CALL MUXCH
MUXCH GETS
CHANNEL
NUMBER FROM
5

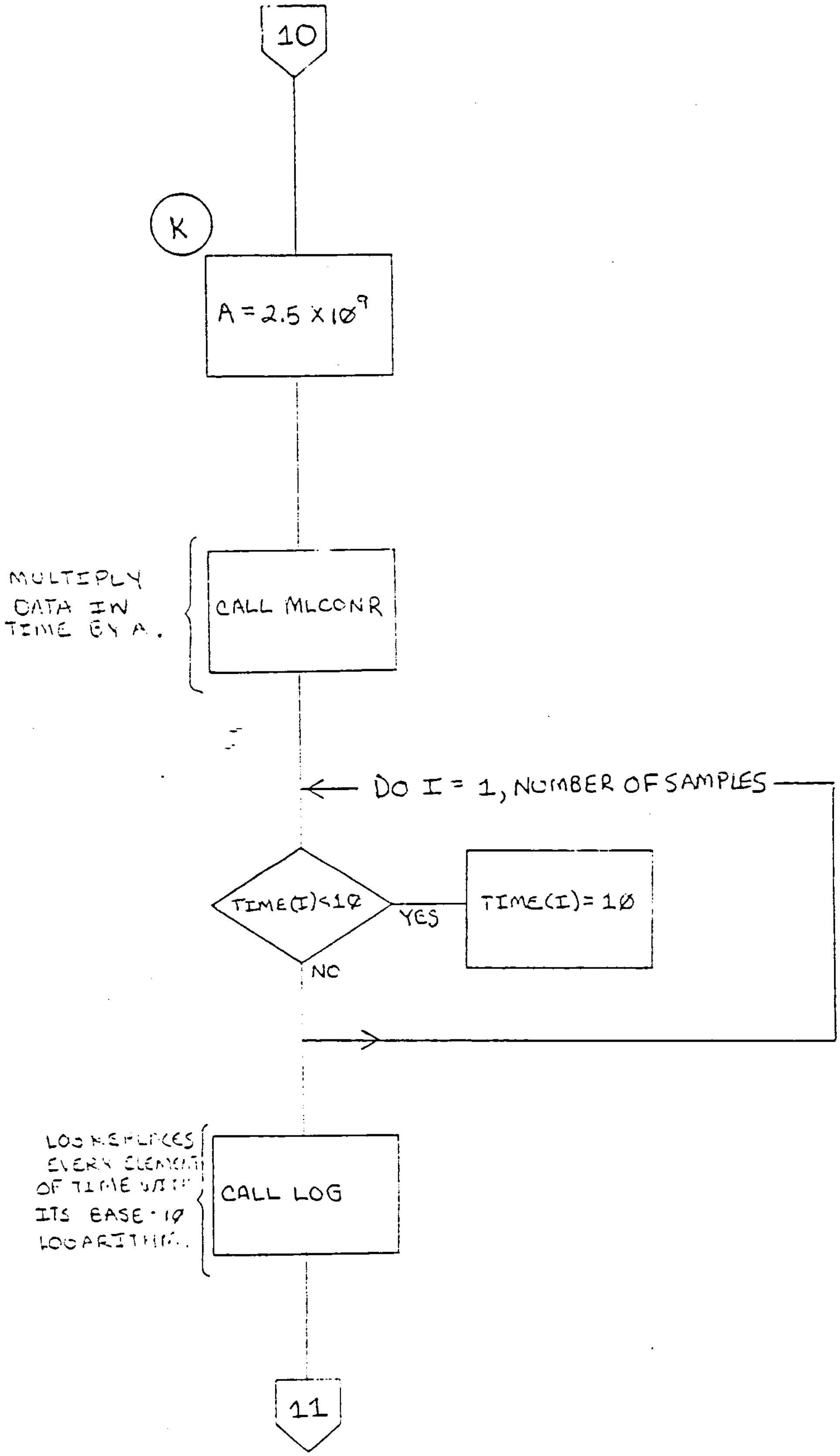

5
ERROR ≠ 0?
YES
H
NO
INITIALIZE DATA FOR TIME INFORMATION BLOCK
NUMBER OF AVERAGES = 2
YES
CALIBRATING?
ZERO TIME
6

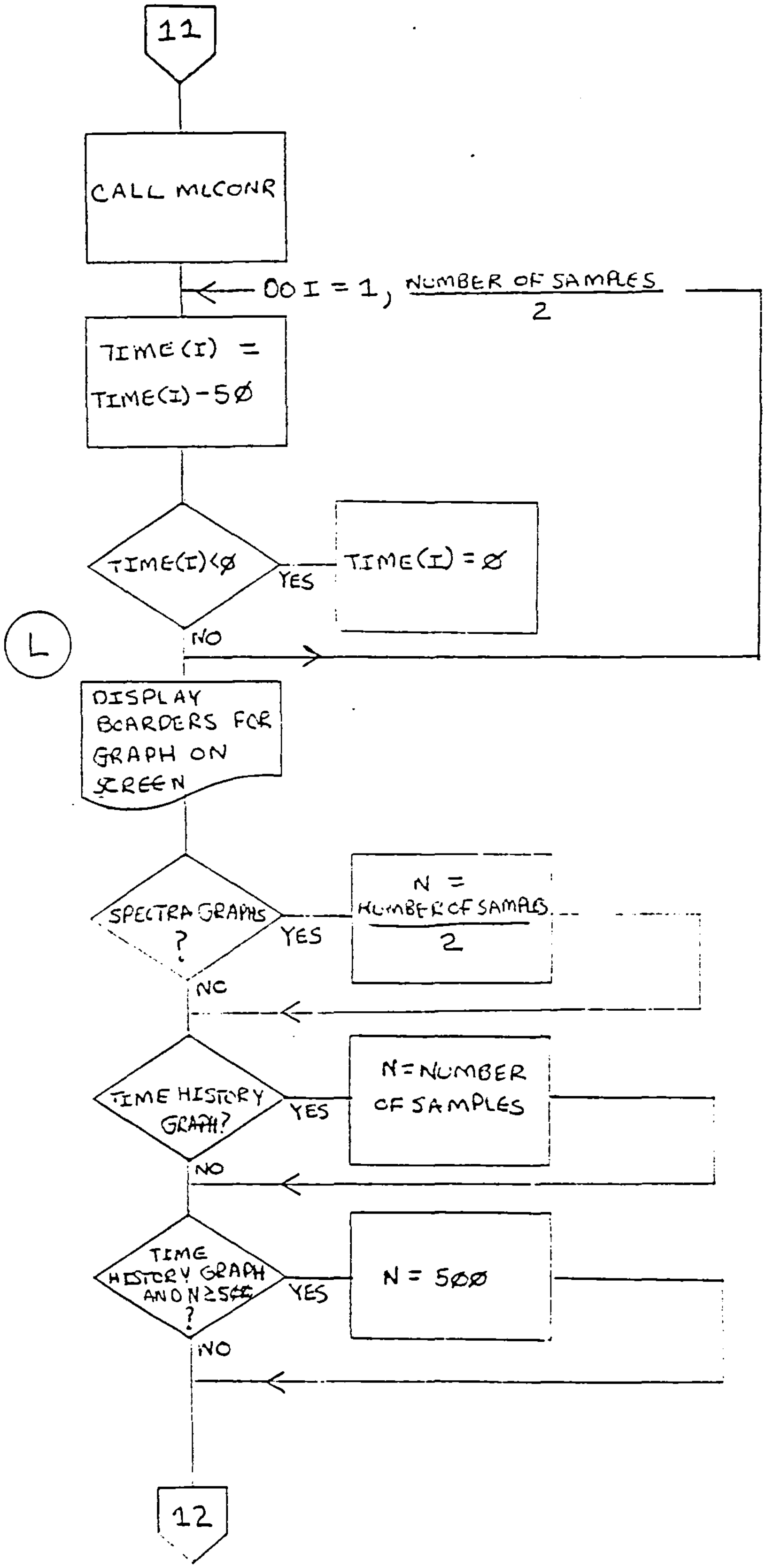

6
DO I = 1, NUMBER OF AVERAGES
CALL ANINP
ANINP SAMPLES ANALOG DATA. DATA IS STORED IN IARRAY.
ERRORS?
YES
DISPLAY ERROR MESSAGE
STOP
NO
CALIBRATING ?
YES
J
NO
FLOINF CONVERTS CONTENTS OF IARRAY TO SINGLE PRECISION FORTRAN FLOATING POINT FORM THEN STORES IT INTO CARRAY.
CALL FLOINF
CALL ADD
ADD CONTENTS OF CARRAY AND TIME. STORE RESULTS IN TIME.
DO N = 1, NUMBER OF SAMPLES
TIME(N) = TIME(N) / NUMBER OF AVERAGES
PRESUR =
VCAL
7

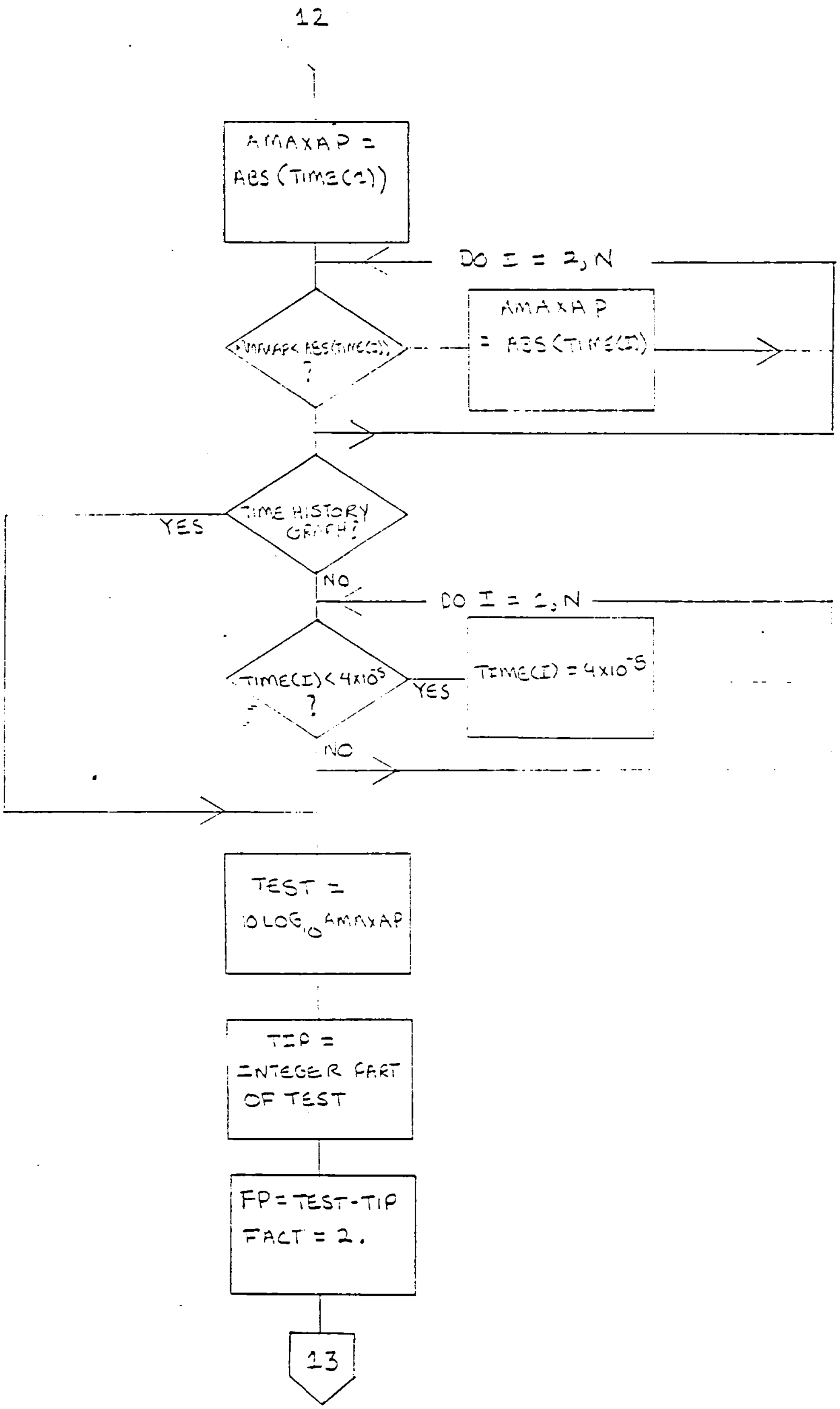

7
CALL MLCONR
MLCONR MULTIPLIES CONTENTS OF TIME BY PRESUR.
I
L
FIXFIN CONVERTS DATA IN TIME TO NORMALIZED INTEGER FORM. THE DATA ARE THEN STORED IN IARRAY
CALL FIXFIN
J
CALL DFT
DFT PERFORMS DIRECT FOURIER TRANSFORM ON DATA IN IARRAY.
ERROR?
YES
DISPLAY ERROR MESSAGE
STOP
NO
DO I=2, NUMBER OF SAMPLES
IARRAY (I)=
IARRAY(I)*2
CALIBRATING ?
NO
YES
INITIALIZE DATA FOR TIME INFORMATION BLOCK
8

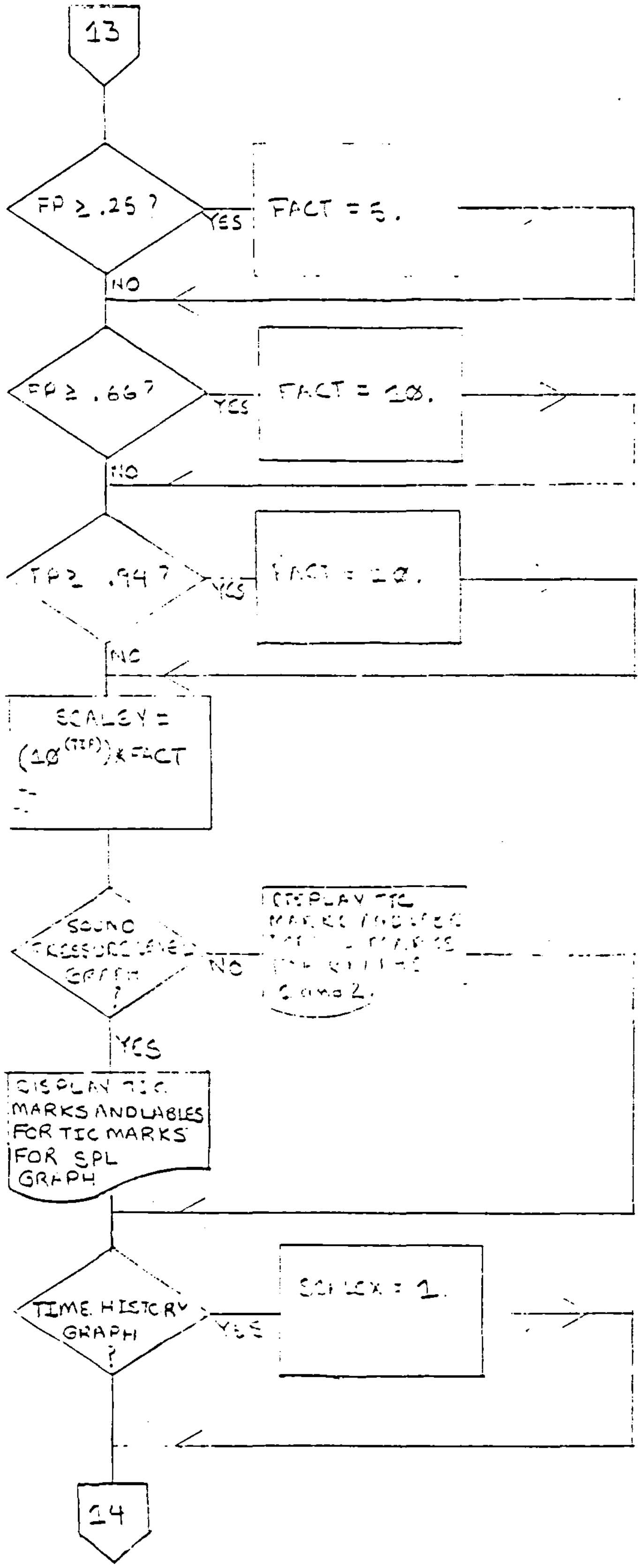

8
CALL ZERO
ASPEC PERFORMS
SELF-CONJUGATE
MULTIPLY-AND-ADD
FOR AVERAGED
AUTO SPECTRUM
FROM FFT RESULTS
DATA ARE STORED
IN TIME ARRAY.
CALL ASPEC
CALL FLOTF
CONVERT DATA
POINT FORM.
CALL MLCONR
MULTIPLY CONTENTS
OF TIME BY ½ TO
CORRECT VALUES
TO RMS PRESSURE
SQUARED.
9

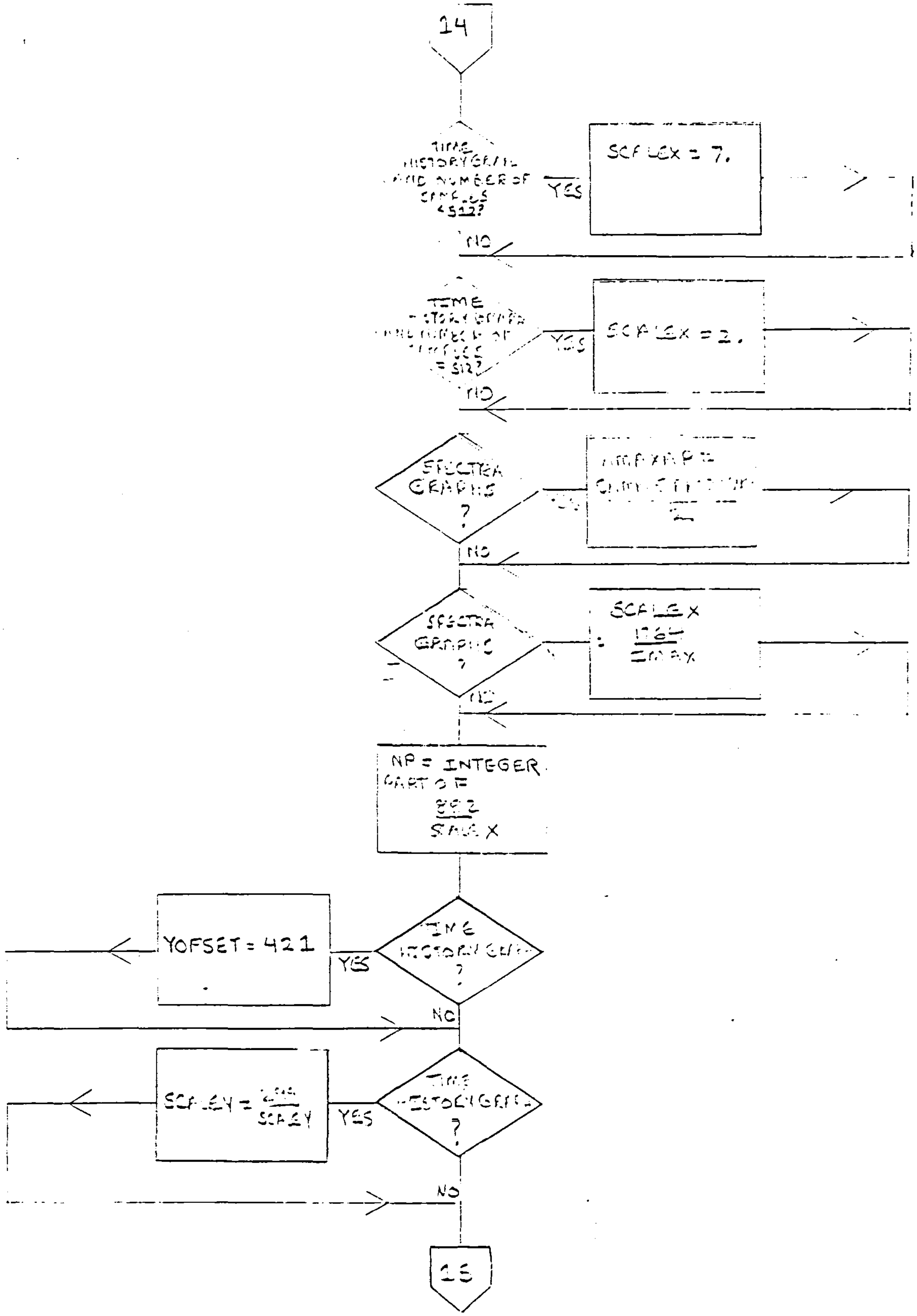

9
CALIBRATING?
YES
NO
VCAL(MICROPHONE)=Ø
DO I = 4ØØ, 6ØØ
VCAL(MICROPHONE) = VCAL(MICROPHONE) + TIME (I)
VCAL(MICROPHONE) = √VCAL(MICROPHONE)
ENTER CALIBRATION FACTOR FOR MICROPHONE
VCAL(MICROPHONE) = (VCAL(MICROPHONE)) * (CALIBRATION FACTOR)
DISPLAY VCAL(MICROPHONE) ON SCREEN
B
IMAX = NUMBER OF SAMPLES / 2
DO I = 1, IMAX
IPI = I + I
IPIMI = IPI-1
TIME (I) = TIME (IPI) + TIME (IPIMI)
DO I = 1, NUMBER OF SAMPLES
TIME (I) < 4.0X10⁻⁹ ?
YES
TIME (I)=4.0X10⁻⁹
NO
L
10


10
K
A = 2.5 X 10^9
MULTIPLY DATA IN TIME BY A.
CALL MLCONR
DO I = 1, NUMBER OF SAMPLES
TIME(I)<10
YES
TIME(I)= 10
NO
LOG REPLACES EVERY ELEMENT OF TIME WITH ITS BASE-10 LOGARITHM.
CALL LOG
11

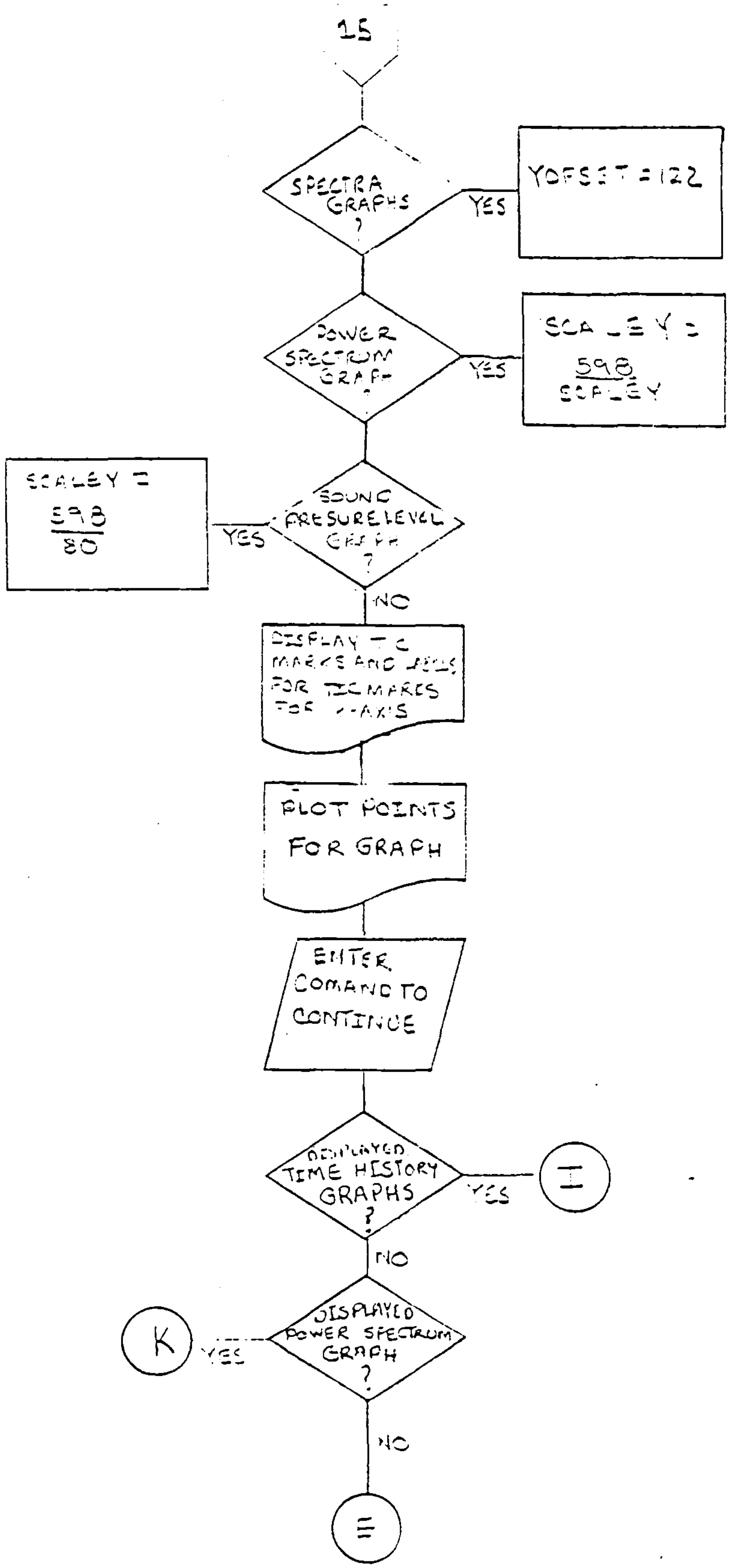

11
CALL MLCONR
DO I = 1, NUMBER OF SAMPLES / 2
TIME(I) = TIME(I) - 5Ø
TIME(I)<Ø
YES
TIME(I) = Ø
NO
L
DISPLAY BOARDERS FOR GRAPH ON SCREEN
SPECTRA GRAPHS ?
YES
N = NUMBER OF SAMPLES / 2
NO
TIME HISTORY GRAPH?
YES
N = NUMBER OF SAMPLES
NO
TIME HISTORY GRAPH AND N ≥ 5ØØ ?
YES
N = 5ØØ
NO
12


12
AMAXAP =
ABS (TIME(1))
DO I = 2, N
AMAXAP < ABS(TIME(I)) ?
AMAXAP
= ABS (TIME(I))
TIME HISTORY GRAPH?
YES
NO
DO I = 1, N
TIME(I) < 4x10-5 ?
YES
TIME(I) = 4x10-5
NO
TEST =
10 LOG10 AMAXAP
TIP =
INTEGER PART
OF TEST
FP = TEST-TIP
FACT = 2.
13


13
FP ≥ .25 ?
YES
FACT = 5.
NO
FP ≥ .66 ?
YES
FACT = 1Ø.
NO
YES
NO
SCALEY =
NO
YES
MARKS AND LABLES
FOR TIC MARKS
FOR SPL
GRAPH
TIME HISTORY
GRAPH
?
14



14
SCALEX = 7.
YES
NO
SCALEX = 2.
YES
NO
SPECTRA GRAPHS ?
NO
SPECTRA GRAPHS ?
SCALEX = 1764 / IMAX
NO
NP = INTEGER PART OF 892 / SCALEX
YOFSET = 421
YES
TIME HISTORY GRAPH ?
NO
SCALEY = / SCALEY
YES
TIME HISTORY GRAPH ?
NO
15


15
SPECTRA GRAPHS ?
YES
YOFSET = 122
POWER SPECTRUM GRAPH ?
YES
SCALEY = 598 / SCALEY
SOUND PRESURE LEVEL GRAPH ?
YES
SCALEY = 598 / 80
NO
DISPLAY TIC MARKS AND LABELS FOR TIC MARKS FOR X-AXIS
PLOT POINTS FOR GRAPH
ENTER COMAND TO CONTINUE
DISPLAYED TIME HISTORY GRAPHS ?
YES
I
NO
DISPLAYED POWER SPECTRUM GRAPH ?
YES
K
NO
E

APPENDIX C

PROGRAM LISTING

```
C*******
C
C     FILE NAME:   TF2.FOR
C     TO COMPILE:  TF2=TF2/W
C     TO LINK:     TF2=TF2,S,T/F
C
C*******

C*********************************************
C*  DIMENSION AND DATA STATEMENTS            *
C*********************************************

        DIMENSION IARRAY(2048),CARRAY(2048),AIBI(6)
        DIMENSION IRANGE(8),ISIZE(4),AIBC(6),ILEVEL(7)
        DIMENSION DUM(1),SAM(7),SFREQ(8),TIME(2048)
        DIMENSION VCAL(99),VOLT(7),AIBT(6),IFILTE(8)
        LOGICAL*1 AC2MAX(9)
        LOGICAL*1 ACTMAR(7)
        LOGICAL*1 ACHAR(60)
        LOGICAL*1 ACHAV(12)
        LOGICAL*1 ACMIN(6)
        LOGICAL*1 ACMAX(5)
        DATA IFILTE/2,4,5,7,8,10,11,6/
        DATA ISIZE/128,512,1024,2048/
        DATA ILEVEL/ 11,22,33,44,55,66,77/
        DATA IRANGE/ 3,2,2,1,1,0,0,2/
        DATA ANEW/2HN /
        DATA SAM/ 51.2,204.8,512.,2048.,5120.,20480.,51200./
        DATA SFREQ/ 7.0,2.0,7.0,2.0,7.0,2.0,7.0,10.0/
        DATA VOLT/ .125,.25,.5,1.0,2.0,4.0,8.0/
```

```
C****************************************
C* PROGRAM BEGINS                        *
C****************************************


C** SET INTEGER COMPLIER SWITCH TO 1-WORD INTEGERS ***

        AIBI(6)=0.0

C************************************************
C* READ IN CALIBRATION DATA FROM FILE VCAL2.DAT *
C************************************************

        CALL ASSIGN(2,'VCAL2.DAT',9,'OLD')
        READ (2,1) (VCAL(J),J=1,99)
1       FORMAT(F8.6)
        CALL CLOSE(2)


C****************************************************
C* PROMPT OPERATOR FOR TEST, RUN, AND POINT NUMBERS *
C****************************************************

10      CALL ERASE('KB')
20      TYPE 30
30      FORMAT(1X,'ENTER TEST NUMBER NNN ',$)
        ACCEPT 40,ITEST
40      FORMAT (I3)
        IF(ITEST.EQ.0) GO TO 999
50      TYPE 60
60      FORMAT (1X,'ENTER RUN NUMBER NNN ',$)
        ACCEPT 40,IRUN
        IF (IRUN.EQ.000) GO TO 20
```

```
70        TYPE 80
80        FORMAT (1X,'ENTER POINT NUMBER NNN ',$)
          ACCEPT 40,IPOINT
          IF (IPOINT.EQ.000) GO TO 50

C**************************
C* NEW OR OLD CALIBRATION *
C**************************

          TYPE 90
90        FORMAT(1X,'WHAT TYPE OF CALIBRATION ? O OR N (NO <CR>) ',$)
          CALL IPOKE("44,"10000)
100       ICAL = ITTINR()
          IF(ICAL.EQ.78 .OR. ICAL.EQ.79) CALL ECHO (ICAL)
          IF(ICAL.EQ.78 .OR. ICAL.EQ.79) CALL IPOKE("44,"00000)
          IF(ICAL .EQ. 78) GO TO 110
          IF(ICAL .EQ. 79) GO TO 140
          GO TO 100

C*** PROMPT OPERATOR FOR NUMBER OF NEXT MICROPHONE TO CALIBRATE ***

110       TYPE 120
120       FORMAT('+ ',/,' ENTER NUMBER OF MICROPHONE TO CALIBRATE  ',$)
          ACCEPT 130,IMIK
130       FORMAT(I2)
          IF (IMIK.NE.0) GO TO 400
          ICAL = 79

C* WHEN DONE WITH CALIBRATION WRITE CALIBRATION DATA TO FILE VCAL2.DAT *

          CALL ASSIGN(2,'VCAL2.DAT',9,'NEW')
          WRITE(2,1) ( VCAL(I),I=1,99)
          CALL CLOSE(2)
```

```
C***********************************
C          M E N U                 *
C***********************************

140     CALL ERASE('KB')

C**** USER PROMPTS FOR VOLTAGE CODE AND SAMPLE FREQUENCY ***

        TYPE 150
150     FORMAT(3X,'VOLTS',8X,'ENTER',5X,'SAMPLE-FREQUENCY',
     1  4X,'CUT-OFF FREQUENCY',5X,'ENTER',//,1X,'+/-  0.125',7X,'1',13X,
     1  '51.2',16X,'20 HZ',13X,'1',//,1X,'+/-  0.25',8X,'2',12X,'204.8'
     1  ,15X,'100 HZ',13X,'2',//,1X,'+/-  0.5',9X,'3',12X,'512.0',
     1  15X,'200 HZ',13X,'3',//,1X,'+/-  1.0',9X,'4',11X,'2048.0',
     1  16X,'1 KHZ',13X,'4',//,1X,'+/-  2.0',9X,'5',11X,'5120.0',
     1  16X,'2 KHZ',13X,'5',//,1X,'+/-  4.0',9X,'6',10X,'20480.0',
     1  15X,'10 KHZ',13X,'6',//,1X,'+/-  8.0',9X,'7',10X,'51200.0',
     1   15X,'20 KHZ',13X,'7')

160     TYPE 170
170     FORMAT (//,23X,'ENTER VOLTAGE CODE ',$)
        ACCEPT 180,INDEXL
180     FORMAT(I2)
190     IF ((INDEXL.GE.1).AND.(INDEXL.LE.7)) GO TO 220
        IF (INDEXL.NE.0) GO TO 200
        CALL ERASE ('KB')
        GO TO 70
200     TYPE 210
210     FORMAT (/,20X,'WHAT?  VOLTAGE CODE IS 1-7!')
        GO TO 160
220     TYPE 230
230     FORMAT (//,19X,'ENTER SAMPLE-FREQUENCY CODE ',$)
        ACCEPT 180,INDEXS
        IF ((INDEXS.GE.1).AND.(INDEXS.LE.7)) GO TO 250
        IF (INDEXS.EQ.0) GO TO 160
```

```
      TYPE 240
240   FORMAT (/,18X,'WHAT?  SAMPLE FREQUENCY IS 1-7!')
      GO TO 220
250   CALL ERASE('KB')

C**** USER PROMPT FOR NUMBER OF SAMPLES  ****

260   TYPE 270
270   FORMAT(///,T20,'NUMBER OF SAMPLES ',',    ENTER',//,T28,'128'
    1 ,T43,'1',//,T28,'512',T43,'2',//,T27,'1024',T43,'3',//,T27,
    1 '2048',T43,'4')
280   TYPE 290
290   FORMAT (//,19X,'ENTER SAMPLE CODE ',$)
      ACCEPT 180,INDEXZ
      IF ((INDEXZ.GE.1).AND.(INDEXZ.LE.4)) GO TO 310
      IF (INDEXZ.EQ.0) GO TO 140
      TYPE 300
300   FORMAT (18X,'WHAT?  SAMPLE NUMBER IS 1-4!')
      GO TO 280

C**** BANDWIDTH, VOLTS, AND SAMPLE FREQUENCY OUTPUT ***

310   BDW= SAM(INDEXS)/ISIZE(INDEXZ)
      TYPE 320,BDW
320   FORMAT (19X,'BANDWIDTH = ',F10.3)
      TYPE 321,VOLT(INDEXL)
321   FORMAT (/,19X,'VOLTS = +/- ',F5.2)
      TYPE 322,SAM(INDEXS)
322   FORMAT (/,19X,'SAMPLE FREQUENCY = ',F8.2)

C***** USER PROMPT FOR MICROPHONE, GAIN, AND AVERAGES *******

323   TYPE 324
324   FORMAT(/,19X,'ENTER MICROPHONE NN ',$)
      ACCEPT 180,IMIK
```

```
      IF (IMIK.EQ.0) GO TO 250
325   TYPE 326
326   FORMAT ('+ ',/,19X,'ENTER GAIN (REAL INPUT) ',$)
      ACCEPT 327,GAIN
327   FORMAT (F5.1)
328   IF (GAIN.EQ.99.) GO TO 323
      TYPE 329,GAIN
329   FORMAT(19X,'GAIN IS ',F5.1,'    Y OR N (NO <CR>) ',$)
      CALL IPOKE("44,"10000)
330   IDU = ITTINR()
      IF (IDU.EQ.78.OR.IDU.EQ.89) CALL ECHO (IDU)
      IF (IDU.EQ.78.OR.IDU.EQ.89) CALL IPOKE("44,"00000)
      IF (IDU.EQ.78) GO TO 325
      IF (IDU.EQ.89) GO TO 331
      GO TO 330
331   TYPE 332
332   FORMAT('+ ',//,19X,'DATA ACQUISTION STARTS WHEN <CR> IS ENTERED',
     1/,19X,'ENTER NUMBER OF AVERAGES NNN ',$)
      ACCEPT 333,NOAVE
333   FORMAT(I3)
      IF (NOAVE.EQ.0) GO TO 325
      GO TO 410

C****************************************
C*      DATA                            *
C****************************************

C***** CALIBRATION PARAMETERS FOR ACQUISITION OF ANALOG DATA *****

400   INDEXS=8
      INDEXZ=4
      INDEXL=5
```

```
C**** PARAMETERS AND DATA FOR ACQUISITION OF ANALOG DATA ***

410      CALL ANINIT(IARRAY,AIBI,DUM,DUM,ISIZE(INDEXZ),1,
     1   SFREQ(INDEXS),IRANGE(INDEXS),ILEVEL(INDEXL),IER,0,6,
     1   IFILTE(INDEXS),1)
         IF (IER.EQ.0) GO TO 430
         TYPE 420,IER
420      FORMAT (//,5X,'ERROR ANINIT',I5)
         STOP
430      IF (ICAL.EQ.78) GO TO 440

C*** AC COUPLING, DC OFFSET, AND CALIBRATION OPTIONS FOR
C*** INPUT OF ANALOG DATA *****

         CALL ANINIT (2,IER)
         IF (IER.EQ.0) GO TO 440
         TYPE 420,IER
         STOP
440      CALL MUXCH (IER)
         IF (IER.NE.0) GO TO 440

C**** INITIALIZE DATA FOR TIME INFORMATION BLOCK ****

         AIBT(1) = 0.0
         DO 450 I=2,6
450      AIBT(I) = AIBI(I)

C** IF CALIBRATING SET NUMBER OF AVERAGES TO 1 **

         IF (ICAL.EQ.78) NOAVE = 1

C*** START SAMPLING DATA POINTS *****

         CALL ZERO(TIME,AIBT)
         DO 460 I=1,NOAVE
```

```
        CALL ANINP (IFRAME,IER)
        IF ((IFRAME.GE.0).AND.(IER.GE.0)) GO TO 462
        TYPE 461,IER,IFRAME
461     FORMAT (//5X,'ERROR ANINP',I5,15X,'FRAME CODE ',I5)
        STOP
462     IF (ICAL.EQ.78) GO TO 500
        AIBC(2)=AIBI(2)
        CALL FLOINF(IARRAY,AIBI,CARRAY,AIBC)
        CALL ADD(CARRAY,AIBC,TIME,AIBT)
460     CONTINUE
        DO 470 N=1,ISIZE(INDEXZ)
470     TIME(N)=TIME(N)/NOAVE

C** CONVERT TO PRESSURE UNITS OF N/(M*M) **

480     BETA=10.0**(-(GAIN)/20.)
        PRESUR= (BETA*(31.7/VCAL(IMIK)))
        CALL MLCONR(PRESUR,TIME,AIBT)
        LG=1

C**** GRAPH TIME HISTORY  *******

        GO TO 630

C************************************
C*  DATA FOR POWER SPECTRUM GRAPH  *
C************************************

C** CONVERT DATA TO NORMALIZED INTEGER FORM **

490     AIBI(2)=AIBT(2)
        CALL FIXFIN(TIME,AIBT,IARRAY,AIBI)
```

```
C** DIRECT FOURIER TRANSFORM **

500     CALL DFT(IARRAY,AIBI,IER,1)
        IF (IER.EQ.0) GO TO 520
        TYPE 510,IER
510     FORMAT (//,5X,'ERROR DFT',I5)
        STOP

C** CORRECT FOR ONE-SIDED DFT **

520     DO 521 I=2,ISIZE(INDEXZ)
521     IARRAY(I)=IARRAY(I)*2


C** OLD CALIBRATION ? THEN NO NEED TO INITIALIZE DATA FOR
C** TIME INFORMATION BLOCK **

        IF (ICAL.EQ.79) GO TO 526

C** CALIBRATING SO INITIALIZE DATA FOR TIME INFORMATION BLOCK **

        AIBT(1) = 0.0
        AIBT(2)=AIBI(2)
        DO 525 I=3,6
525     AIBT(I)=AIBI(I)
526     CALL ZERO (TIME,AIBT)
        AIBT(5)=4.0

C** PERFORM SELF-CONJUGATE MULTIPLY-AND-ADD FOR AVERAGED
C** AUTO SPECTRUM FROM DFT RESULTS ***

        CALL ASPEC(IARRAY,AIBI,TIME,AIBT)
```

```
C** CONVERT DATA TO FORTRAN FLOATING POINT FORM **

        CALL FLOTF(TIME,AIBT)

C*** CORRECT VALUES TO RMS PRESSURE SQUARED ****

        CALL MLCONR(.5,TIME,AIBT)

C** IF CALIBRATING THEN SUM MAGNITUDES SQUARED
C** FOR CALIBRATED DATA ***

        IF (ICAL.EQ.78) GO TO 540

C** IF NOT CALIBRATING THEN GROUP POWER SPECTRA IN 2'S **

        IMAX = ISIZE(INDEXZ)/2
        DO 528 I = 1,IMAX
        IPI = I+I
        IPIMI = IPI - 1
528     TIME(I) = TIME(IPI) + TIME(IPIMI)

C** SET LOWER LIMIT FOR POWER SPECTRA **

        DO 530 I=1,ISIZE(INDEXZ)
530     IF (TIME(I).LT.(4.0*(10.**(-9.)))) TIME(I)=4.0*(10.**(-9.))

C** GRAPH POWER SPECTRUM ***

        LG = 2
        GO TO 630

C**** SUM MAGNITUDES SQUARED FOR CALIBRATED DATA ****

540     VCAL(IMIK)=0
```

```
      DO 550 I=400,600
550   VCAL(IMIK)=VCAL(IMIK)+TIME(I)
      VCAL(IMIK)=SQRT(VCAL(IMIK))
      TYPE 560,IMIK
560   FORMAT(2X,'WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE ',
     1 I3,' (REAL INPUT)')
      ACCEPT 570, CALFAC
570   FORMAT (G18.3)
      VCAL(IMIK)=VCAL(IMIK)*CALFAC
      TYPE 580,IMIK,VCAL(IMIK)
580   FORMAT(2X,'VCAL(',I2,') = ',F8.6)

C**** PROMPT FOR NEXT MICROPHONE *******

      GO TO 110


C*************************************
C* DATA FOR SOUND PRESSURE LEVEL GRAPH *
C*************************************


C** CONVERT TO DB SCALE ****

600   A = 2.5*(10.**9.)
      CALL MLCONR(A,TIME,AIBT)
      DO 610 I=1,ISIZE(INDEXZ)
610   IF (TIME(I).LT.10.) TIME(I)=10.
      CALL LOG(TIME,AIBT)
      CALL MLCONR(10.,TIME,AIBT)
      DO 602 I=1,ISIZE(INDEXZ)/2
      TIME(I) = TIME(I)-50.
602   IF (TIME(I).LT.0.) TIME(I)= 0.
      LG=3
```

```
C******************************
C*  GRAPHS                    *
C******************************

C**** DRAW BOARDERS *****

630     CALL ERASE('KB')
        CALL BEAMP('KB',141,122)
        CALL VECTOR('KB',1023,122)
        CALL VECTOR('KB',1023,720)
        CALL VECTOR('KB',141,720)
        CALL VECTOR('KB',141,122)
        IF (LG.EQ.1) CALL BEAMP('KB',141,421)
        IF (LG.EQ.1) CALL VECTOR('KB',1023,421)
        IF (LG.EQ.1) CALL ALPHA('KB',460,760,12,'TIME HISTORY')
        IF (LG.EQ.2) CALL ALPHA('KB',442,760,14,'POWER SPECTRUM')
        IF (LG.EQ.3) CALL ALPHA('KB',399,760,20,'SOUND PRESSURE LEVEL')
        ENCODE(12,640,ACHAV) NOAVE
640     FORMAT(I3,1X,'AVERAGES')
        CALL ALPHA('KB',460,729,12,ACHAV)
        IF (LG.EQ.1) CALL ALPHA('KB',1,500,9,'AMPLITUDE')
        IF (LG.EQ.1) CALL ALPHA('KB',1,461,7,'(N/M*M)')
        IF (LG.EQ.3) CALL ALPHA('KB',35,407,2,'DB')
        IF (LG.EQ.3) CALL ALPHA('KB',30,368,3,'REF')
        IF (LG.EQ.3) CALL ALPHA('KB',5,329,6,'.00002')
        IF (LG.EQ.2) CALL ALPHA('KB',1,500,8,'PRESSURE')
        IF (LG.EQ.2) CALL ALPHA('KB',1,461,7,'SQUARED')
        IF (LG.EQ.2) CALL ALPHA('KB',97,422,1,'2')
        IF (LG.EQ.2) CALL ALPHA('KB',1,399,7,'(N/M*M)')
        IF (LG.GT.1) CALL ALPHA('KB',471,50,14,'FREQUENCY (HZ)')
        IF (LG.EQ.1) CALL ALPHA('KB',500,50,10,'TIME (SEC)')
        ENCODE(60,650,ACHAR) ITEST,IRUN,IPOINT,IMIK
650     FORMAT('TEST = ',I3,4X,'RUN = ',I3,4X,'POINT = ',I3,
     14X,'MICROPHONE = ',I2,3X)
        CALL ALPHA('KB',145,20,60,ACHAR)
```

```
C**** Y-AXIS SCALE *****

          IF (LG.NE.1) N = ISIZE(INDEXZ)/2
          IF (LG.EQ.1) N = ISIZE(INDEXZ)
          IF ((LG.EQ.1).AND.(N.GE.500)) N = 500

C*** FIND MAXIMUM AMPLITUDE *****

          AMAXAP = ABS(TIME(1))
          DO 660 I=2,N
660       IF (AMAXAP.LT.ABS(TIME(I))) AMAXAP = ABS(TIME(I))
          IF (LG.EQ.1) GO TO 680

C***** FOR SPECTRA PLOTS DON'T GRAPH ANYTHING LESS THEN 50 DB **

          DO 670 I=1,N
670       IF(TIME(I).LT..00004) TIME(I) = .00004


C****** COMPUTE SCALING FACTOR *******

680       TEST=ALOG10(AMAXAP)
          TIP=AINT(TEST)
          FP=TEST-TIP
          FACT = 2.
          IF (FP.GE..25) FACT = 5.
          IF (FP.GE..66) FACT = 10.
          IF (FP.GE..94) FACT = 20.
          SCALEY=(10.**TIP)*FACT
          ISCLY=INT(SCALEY)
          IF (LG.EQ.3) GO TO 720
```

```
C******* TIC MARKS FOR Y AXIS GRAPHS 1 AND 2 ******

         ITICY = 122
         DO 685 I=1,10
         CALL BEAMP('KB',141,ITICY)
         CALL VECTOR('KB',125,ITICY)
685      ITICY = ITICY + 60
         CALL BEAMP('KB',141,721)
         CALL VECTOR('KB',125,721)

C******* LABELS FOR Y AXIS TIC MARKS FOR GRAPHS 1 AND 2 *****

         IF (LG.EQ.2) CALL ALPHA('KB',60,122,2,'0.')
         IF (LG.NE.2) ENCODE(5,690,ACMAX) ISCLY
690      FORMAT (I4,'.')
         IF (LG.NE.2) CALL ALPHA('KB',20,720,5,ACMAX)
         IF (LG.EQ.2) ENCODE(9,700,AC2MAX) ISCLY
700      FORMAT(I8,'.')
         IF (LG.EQ.2) CALL ALPHA('KB',1,720,9,AC2MAX)
         IF (LG.EQ.1) CALL ALPHA('KB',60,421,2,'0.')
         IF (LG.EQ.1) ENCODE(6,710,ACMIN) ISCLY
710      FORMAT('-',I4,'.')
         IF (LG.EQ.1) CALL ALPHA('KB',20,122,6,ACMIN)
         GO TO 740

C***** TIC MARKS FOR Y AXIS GRAPH 3 ******

720      ITICY=122
         DO 730 I=1,8
         CALL BEAMP('KB',141,ITICY)
         CALL VECTOR('KB',125,ITICY)
730      ITICY=ITICY+75
         CALL BEAMP('KB',141,721)
         CALL VECTOR('KB',125,721)
```

```
C**** LABELS FOR Y AXIS TIC MARKS GRAPH 3 ****

        CALL ALPHA('KB',40,720,4,'130.')
        CALL ALPHA('KB',55,122,3,'50.')


C*** COMPUTE X AND Y AXIS SCALING FACTOR FOR GRAPHS ****

740     IF (LG.EQ.1) SCALEX = 1.
        IF ((LG.EQ.1).AND.(ISIZE(INDEXZ).LT.512)) SCALEX = 7.
        IF ((LG.EQ.1).AND.(ISIZE(INDEXZ).EQ.512)) SCALEX = 2.
        IF (LG.NE.1) AMAXAP = SAM(INDEXS)/2
        IF (LG.NE.1) SCALEX = 1764./IMAX
        NP = INT(882/SCALEX)
        IF (LG.EQ.1) YOFSET = 421.
        IF (LG.EQ.1) SCALEY = 299./SCALEY
        IF (LG.GT.1) YOFSET = 122.
        IF (LG.EQ.2) SCALEY = 598./SCALEY
        IF (LG.EQ.3) SCALEY = 598./80.


C*** TIC MARKS FOR X AXIS **********

        IF(LG.EQ.1) AMAXAP=NP/(BOW*ISIZE(INDEXZ))
        TEST=ALOG10(AMAXAP)
        TIP=AINT(TEST)
        FP=TEST-TIP
        IF (FP.LT.0) TIP=TIP-1
        IF (TIP.LT.0) FP=FP+1
        FACT=1.
        IF (FP.GT.(0.30102)) FACT = 2.
        IF (FP.GT.(0.69897)) FACT = 5.
```

```
      VALKEY = FACT * (10.**(TIP))
      DELMRK=VALKEY/10.
      NMARKS=1+INT(AMAXAP/DELMRK)
      DO 750  N=1,NMARKS
      NML=N-1
      IXMRK = 141+INT(NML*DELMRK*882/AMAXAP)
      CALL BEAMP('KB',IXMRK,122)
      CALL VECTOR('KB',IXMRK,110)
      IF (N.EQ.1) CALL ALPHA('KB',IXMRK,75,2,'0.')
      IF (N.NE.11) GO TO 750
      CALL BEAMP('KB',IXMRK,122)
      CALL VECTOR('KB',IXMRK,100)
      IF (LG.EQ.1) ENCODE (7,742,ACTMAR) VALKEY
742   FORMAT(F7.5)
      IF (LG.EQ.1) CALL ALPHA('KB',IXMRK-49,75,7,ACTMAR)
      IF (LG.NE.1) ENCODE(6,744,ACMAX) VALKEY
744   FORMAT(F6.0)
      IF (LG.NE.1) CALL ALPHA('KB',IXMRK-49,75,6,ACMAX)
750   CONTINUE

C*** PLOT POINTS FOR ALL 3 GRAPHS ******

      IY = INT (YOFSET + TIME(1)*SCALEY)
      IX = 141.
      CALL BEAMP('KB',IX,IY)
      DO 760 I=2,NP
      IX= INT(141. + (I-1)*SCALEX)
      IY= INT(YOFSET+TIME(I)*SCALEY)
760   CALL VECTOR('KB',IX,IY)

C***** DONE WITH PLOTTING NOW WAIT FOR COMMAND TO CONTINUE ****

      ACCEPT 770
770   FORMAT(1A1)
```

```
C**** IF DONE WITH TIME HISTORY THEN GO AND CALCULATE DATA FOR
C*** POWER SPECTRUM GRAPH ****

        IF (LG.EQ.1) GO TO 490


C*** IF DONE WITH POWER SPECTRUM GRAPH THEN GO AND CALCULATE
C*** DATA FOR SOUND PRESSURE LEVEL GRAPH ****

        IF (LG.EQ.2) GO TO 600

C** IF DONE WITH SOUND PRESSURE LEVEL GRAPH CLEAR
C*** SCREEN AND START OVER ***

        CALL ERASE('KB')
        GO TO 323

C*****************
C*  PROGRAM ENDS   *
C*****************

999     STOP
        END

C*******
C
C   ECHO SUBROUTINE, TO ECHO CHARACTERS ON SCREEN
C
        SUBROUTINE ECHO (ICAL)
        CALL IPOKE ("44,"00000)
        I=ITTOUR(ICAL)
        CALL IPOKE ("44,"10000)
        RETURN
        END
```

C-13

APPENDIX D

EXAMPLES OF TYPICAL RUNNING SEQUENCES

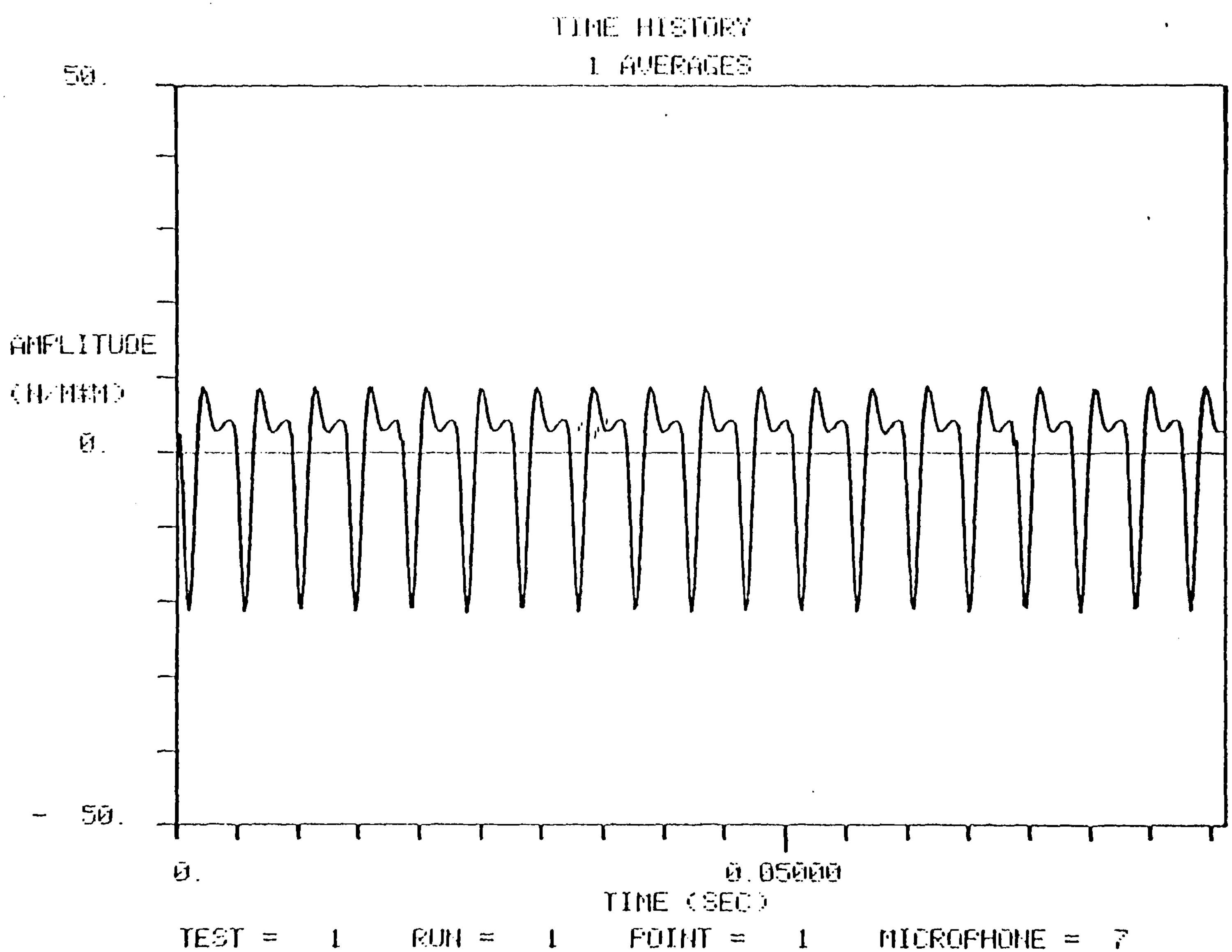
```
ENTER TEST NUMBER NNN 1

ENTER RUN NUMBER NNN 1

ENTER POINT NUMBER NNN 1

WHAT TYPE OF CALIBRATION ? O OR N (NO <CR>) N
ENTER NUMBER OF MICROPHONE TO CALIBRATE  01

 WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE    1 (REAL INPUT)
1.

 VCAL( 1) = 0.579954
ENTER NUMBER OF MICROPHONE TO CALIBRATE  05

 WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE    5 (REAL INPUT)
1.

 VCAL( 5) = 0.579961
ENTER NUMBER OF MICROPHONE TO CALIBRATE  07

 WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE    7 (REAL INPUT)
1.

 VCAL( 7) = 0.580003
ENTER NUMBER OF MICROPHONE TO CALIBRATE  00
```

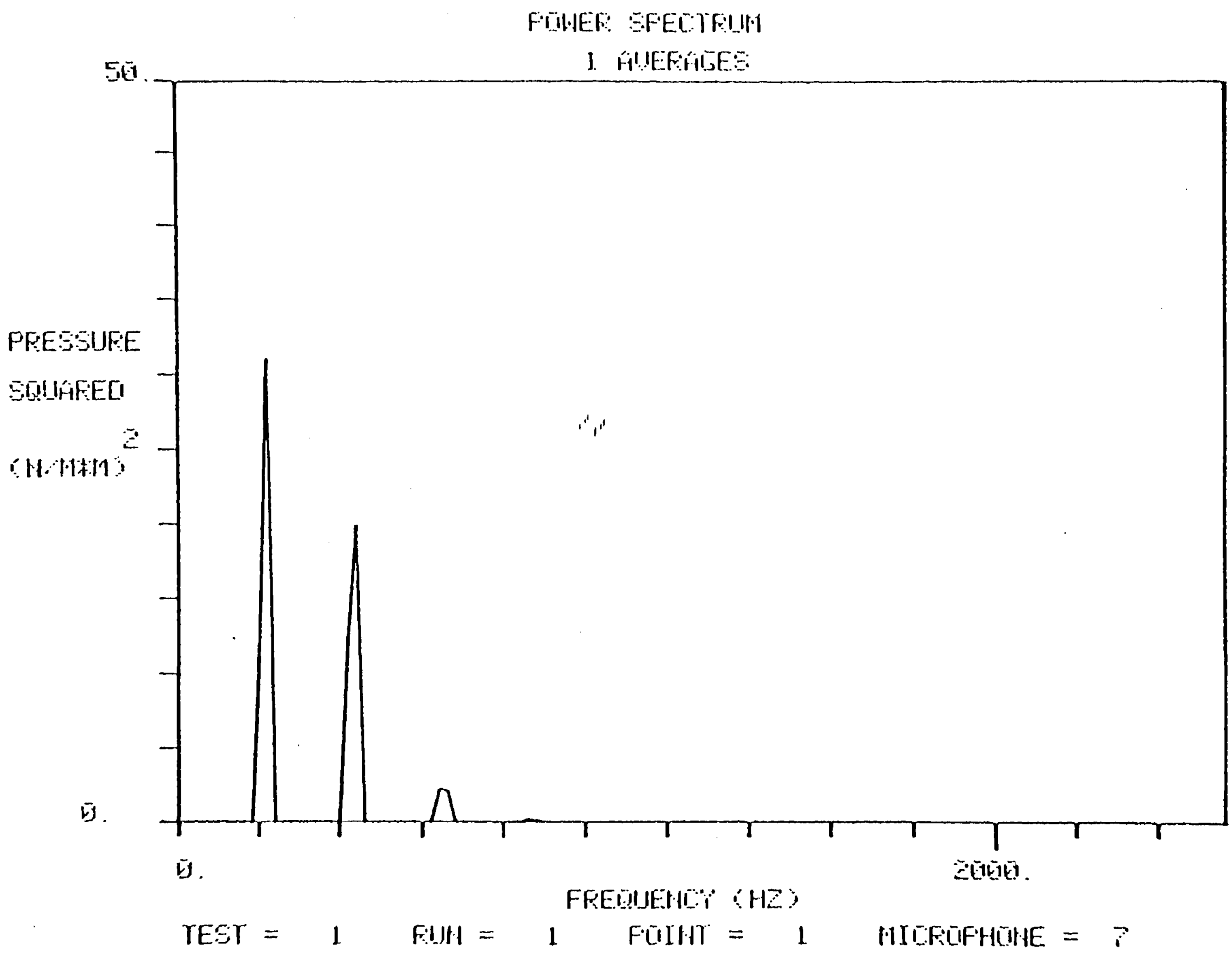
VOLTS	ENTER	SAMPLE-FREQUENCY	CUT-OFF FREQUENCY	ENTER
+/- 0.125	1	51.2	20 HZ	1
+/- 0.25	2	204.8	100 HZ	2
+/- 0.5	3	512.0	200 HZ	3
+/- 1.0	4	2048.0	1 KHZ	4
+/- 2.0	5	5120.0	2 KHZ	5
+/- 4.0	6	20480.0	10 KHZ	6
+/- 8.0	7	51200.0	20 KHZ	7

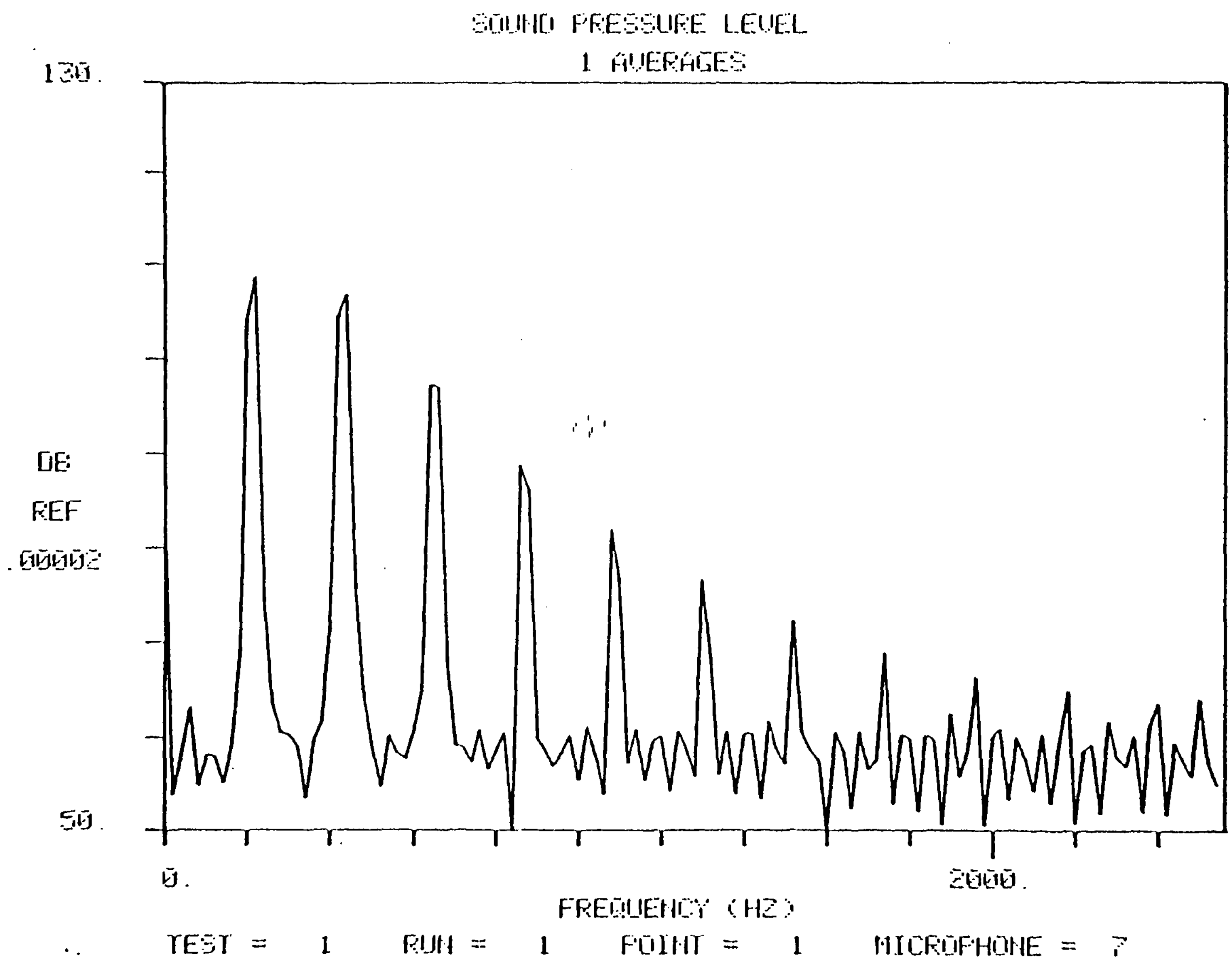
ENTER VOLTAGE CODE 6

ENTER SAMPLE-FREQUENCY CODE 5

```
NUMBER OF SAMRLES    ENTER

        128              1

        512              2

       1024              3

       2048              4

ENTER SAMPLE CODE,2

BANDWIDTH =     10.000

VOLTS = +/-  4.00

SAMPLE FREQUENCY =  5120.00

ENTER MICROPHONE NN 07

ENTER GAIN (REAL INPUT) 0.

GAIN IS   0.0   Y OR N (NO <CR>) Y

DATA ACQUISTION STARTS WHEN <CR> IS ENTERED
ENTER NUMBER OF AVERAGES NNN 001
```


TIME HISTORY
1 AVERAGES
50.
AMPLITUDE
(N/M*M)
0.
- 50.
0.
0.05000
TIME (SEC)
TEST = 1 RUN = 1 POINT = 1 MICROPHONE = 7


POWER SPECTRUM
1 AVERAGES
50.
PRESSURE
SQUARED
2
(N/M*M)
0.
0.
2000.
FREQUENCY (HZ)
TEST = 1 RUN = 1 POINT = 1 MICROPHONE = 7


SOUND PRESSURE LEVEL
1 AVERAGES
130.
DB
REF
.00002
50.
0.
2000.
FREQUENCY (HZ)
TEST = 1 RUN = 1 POINT = 1 MICROPHONE = 7

NUMBER OF SAMPLES	ENTER
128	1
512	2
1024	3
2048	4

ENTER SAMPLE CODE,2

BANDWIDTH = 10.000

VOLTS = +/- 4.00

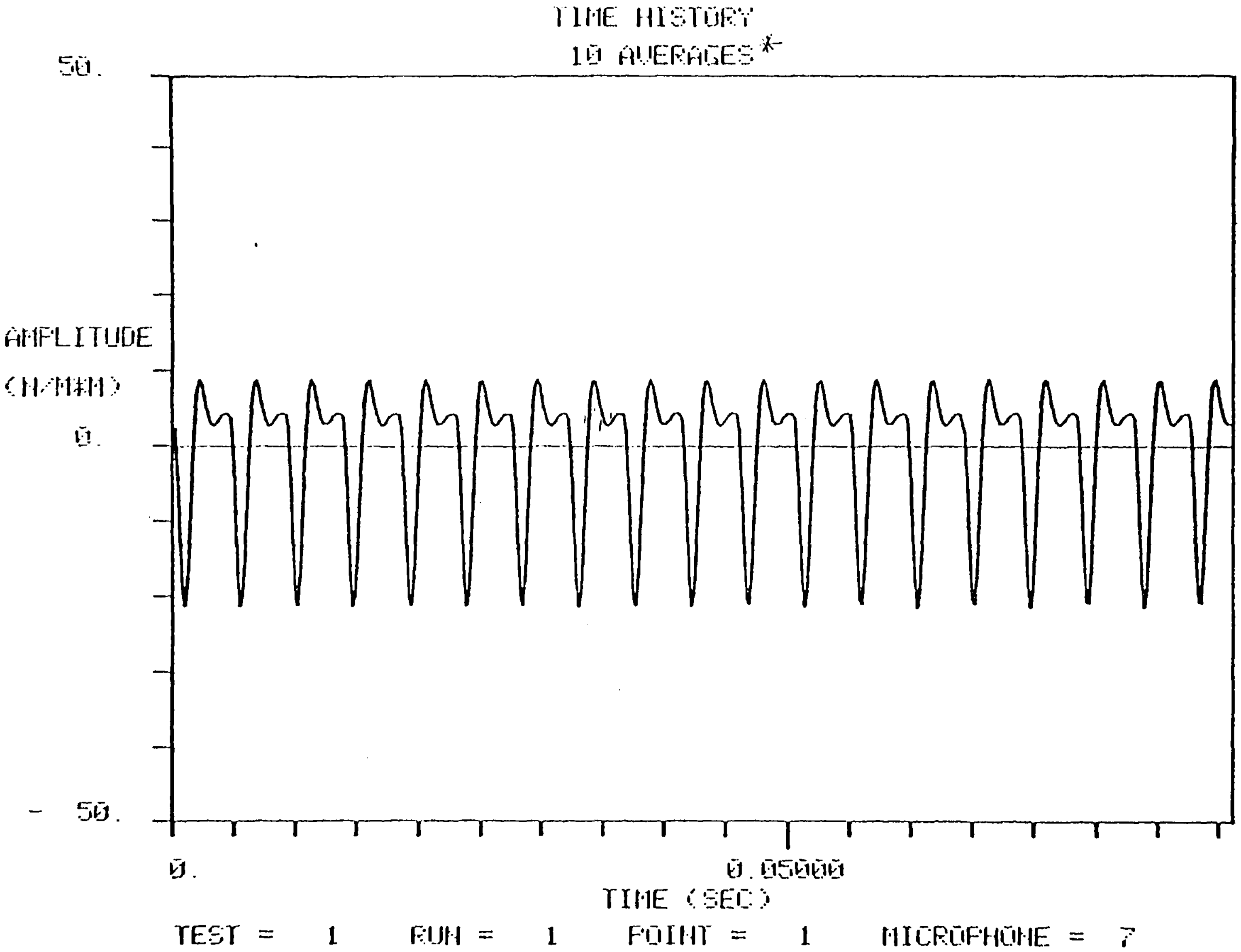
SAMPLE FREQUENCY = 5120.00

ENTER MICROPHONE NN 07

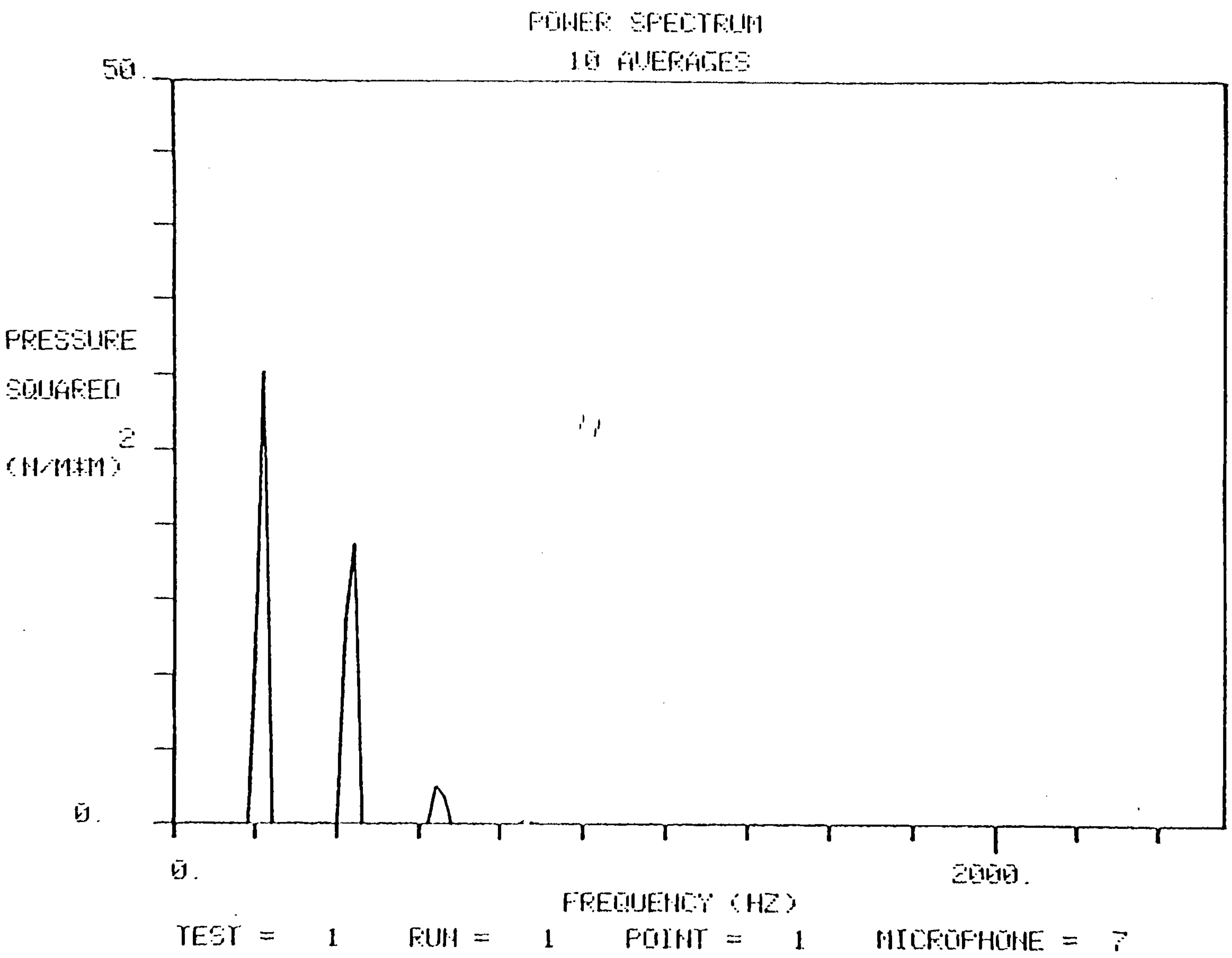
ENTER GAIN (REAL INPUT) 0.

GAIN IS 0.0 Y OR N (NO <CR>) Y

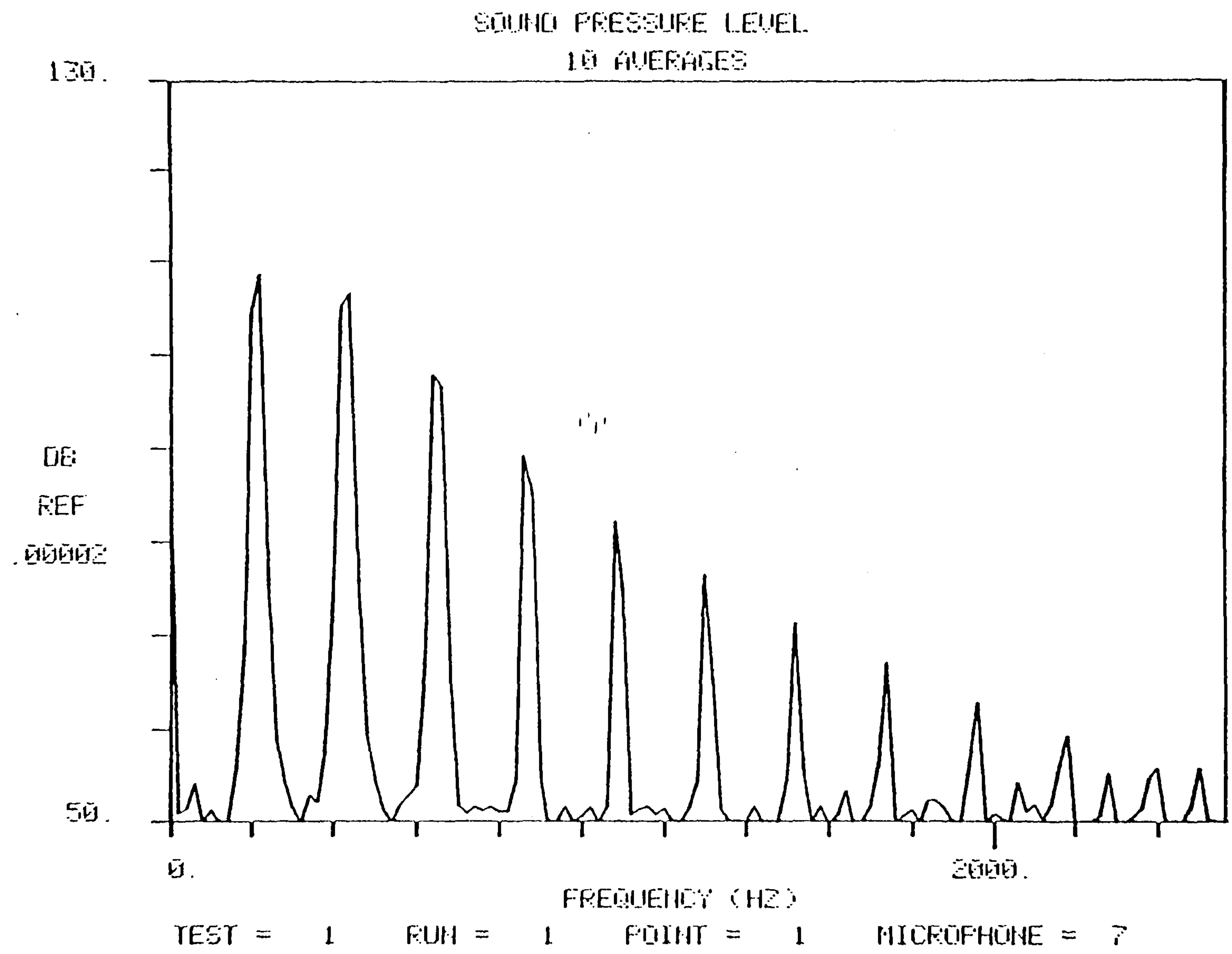
DATA ACQUISITION STARTS WHEN <CR> IS ENTERED
ENTER NUMBER OF AVERAGES NNN 010



* Synchronized with trigger signal.

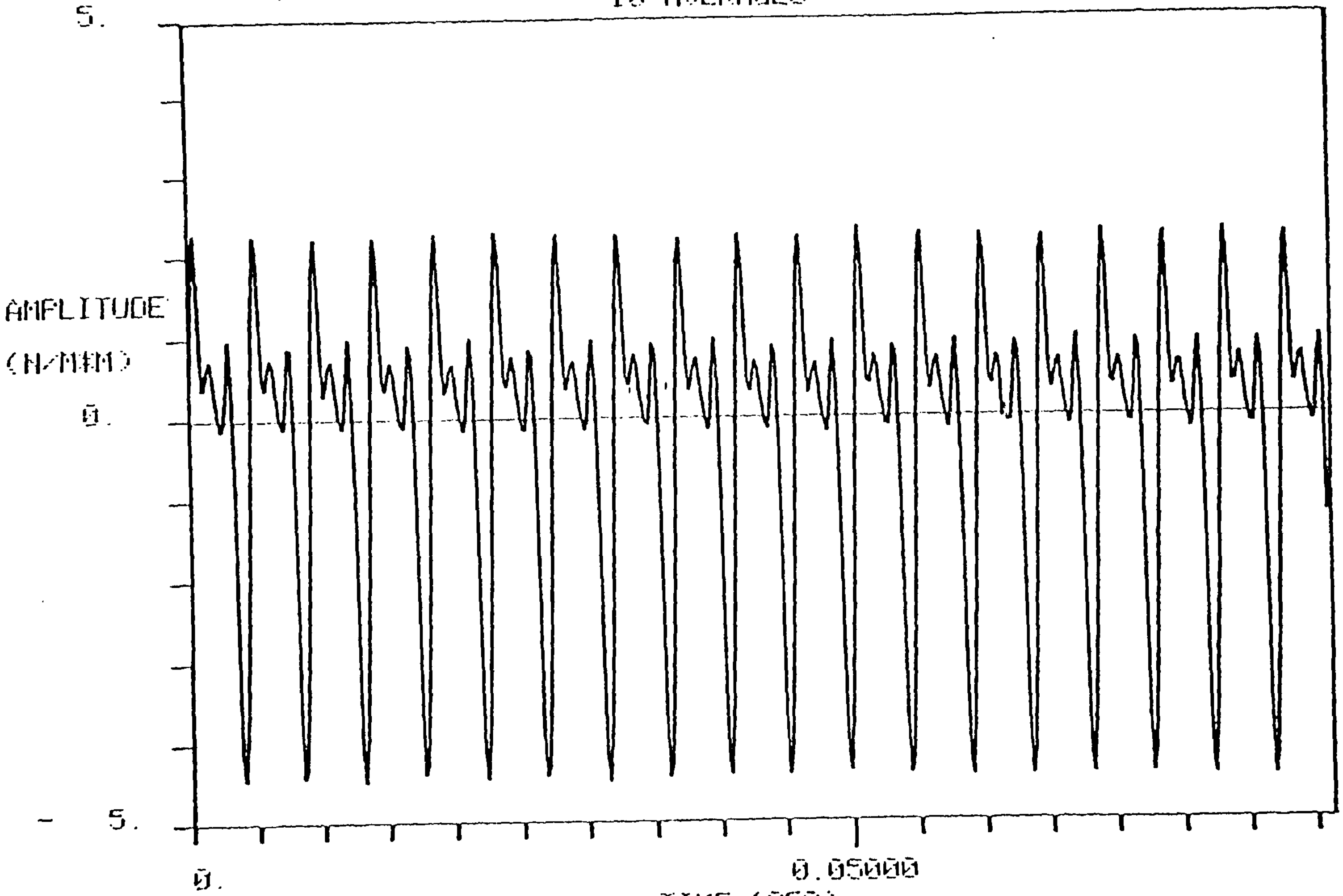


POWER SPECTRUM
10 AVERAGES
50.
PRESSURE
SQUARED
2
(N/M*M)
0.
0.
2000.
FREQUENCY (HZ)
TEST = 1 RUN = 1 POINT = 1 MICROPHONE = 7


SOUND PRESSURE LEVEL
10 AVERAGES
130.
DB
REF
.00002
50.
0.
2000.
FREQUENCY (HZ)
TEST = 1 RUN = 1 POINT = 1 MICROPHONE = 7

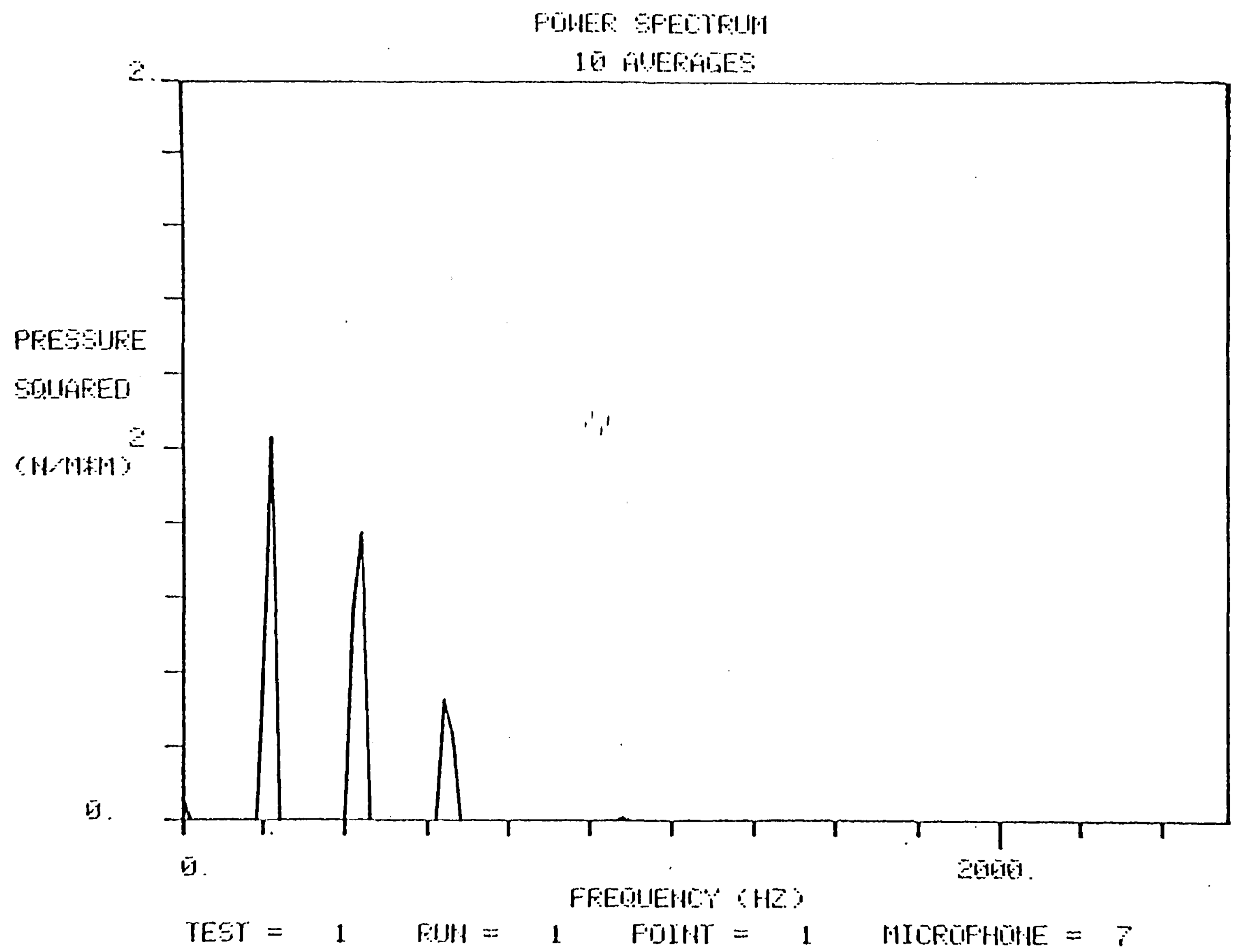
```
NUMBER OF SAMPLES     ENTER
       128              1
       512              2
      1024              3
      2048              4

ENTER SAMPLE CODE,2

BANDWIDTH =      10.000

VOLTS = +/-  4.00

SAMPLE FREQUENCY =  5120.00

ENTER MICROPHONE NN 07

ENTER GAIN (REAL INPUT) 0.

GAIN IS   0.0   Y OR N (NO <CR>) Y

DATA ACQUISTION STARTS WHEN <CR> IS ENTERED
ENTER NUMBER OF AVERAGES NNN 010
```

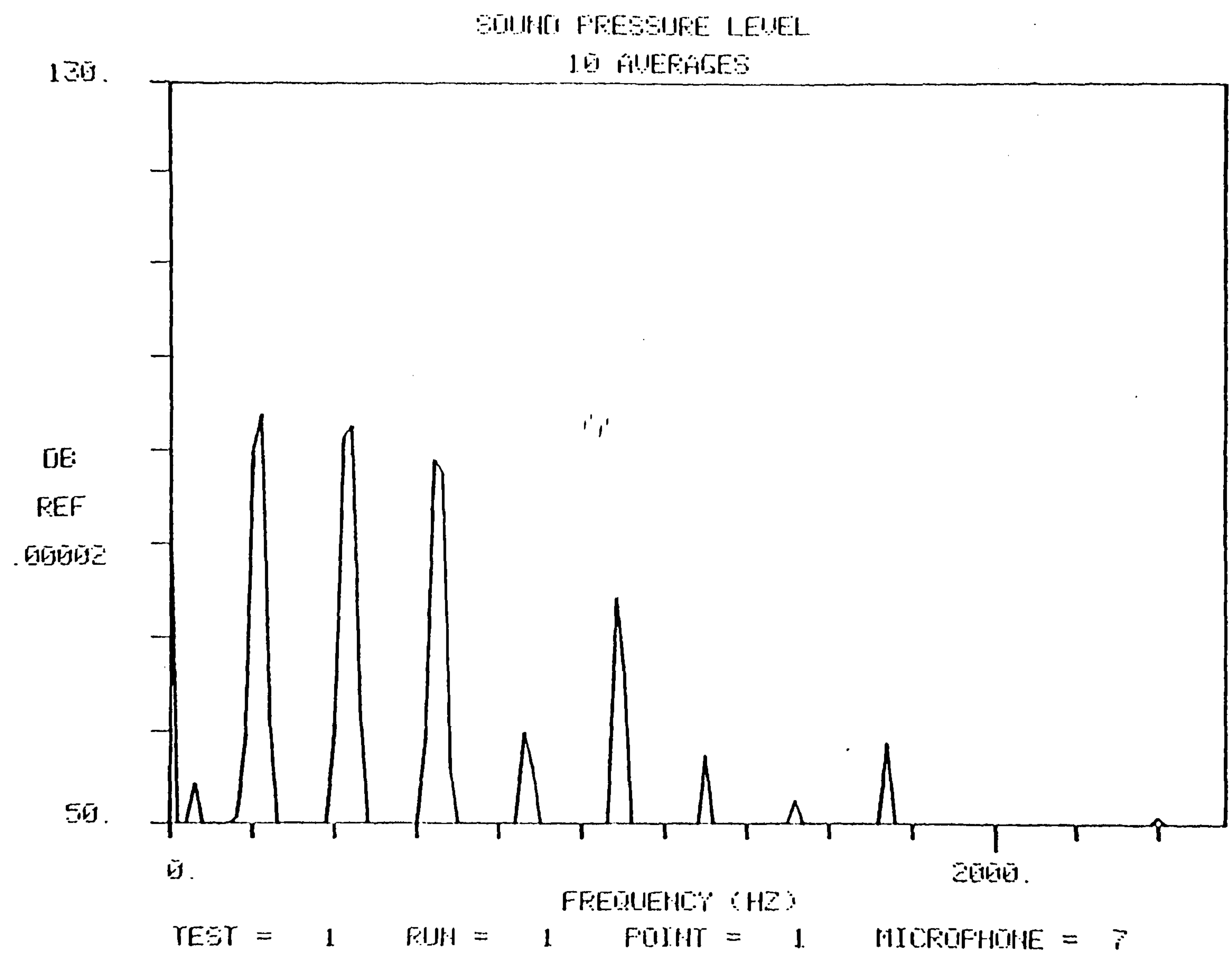
TIME HISTORY
10 AVERAGES*



TEST = 1 RUN = 1 POINT = 1 MICROPHONE = 7

* Not synchronized with trigger signal.


POWER SPECTRUM
10 AVERAGES
2.
PRESSURE
SQUARED
2
(N/M*M)
0.
0.
2000.
FREQUENCY (HZ)
TEST = 1 RUN = 1 POINT = 1 MICROPHONE = 7


SOUND PRESSURE LEVEL
10 AVERAGES
130.
DB
REF
.00002
50.
0.
2000.
FREQUENCY (HZ)
TEST = 1 RUN = 1 POINT = 1 MICROPHONE = 7

1. Report No. CR 166400	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Ensemble Averaging of Acoustic Data		5. Report Date September 1982 6. Performing Organization Code
7. Author(s) Patricia K. Stefanski		8. Performing Organization Report No.
9. Performing Organization Name and Address Beam Engineering, Inc. P. O. Box 60745 Sunnyvale, California 94086		10. Work Unit No. T5486 11. Contract or Grant No. A88065B
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Contractor Report 14. Sponsoring Agency Code 505-42-21

15. Supplementary Notes
Point of Contact: Technical Monitor, Marianne Mosher
Ames Research Center
Moffett Field, California 94035

16. Abstract

This report documents a computer program called Ensemble Averaging of Acoustic Data. The program samples analog data, analyzes the data, and displays them in the time and frequency domains. Hard copies of the displays are the program's output. The documentation includes a description of the program and detailed user instructions for the program. This software was developed for use on the Ames 40- by 80-Foot Wind Tunnel's Dynamic Analysis System consisting of a PDP-11/45 computer, two RK05 disk drives, a tektronix 611 keyboard/display terminal, an FPE-4 Fourier Processing Element, and an analog-to-digital converter.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Acoustic Data processing	Unlimited Subject Category 61

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	61	

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document