Ensemble Averaging of Acoustic Data

P. K. Stefanski
Ensemble Averaging of Acoustic Data

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

Prepared for
Ames Research Center
under Purchase Order No. A88065B
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 RKOS disk drives, a Tektronix 511 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/m²
\( S(n) \)  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/m²
\( \omega_j \)  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:
\[ D(\omega_j) = \frac{1}{N} \sum_{n=0}^{N-1} [P(n)\exp[-i2\pi \omega_j n/N] \times \sqrt{8/3} \times 1/2 \times (1-\cos 2\pi n/N)] \]

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

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

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

\[ VCAL = \left[ \sum_{j=400}^{500} H(\omega_j) \right]^{1/2} \times [\text{calibration factor}] \]

\( VCAL \) 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

\[ S(n) = \frac{\sum_{K=1}^{K_{\max}} S_K(n)/K_{\max}}{} \]

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

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

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

If TF3 is used, \( F(\omega_j) = H(\omega_{j-2}) + H(\omega_{j-1}) + H(\omega_j) \) 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:

```plaintext
.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.

```plaintext
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? 0 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 than 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
\[\begin{align*}
\pm .125 & : 1 \\
\pm .25 & : 2 \\
\pm .5 & : 3 \\
\pm 1.0 & : 4 \\
\pm 2.0 & : 5 \\
\pm 4.0 & : 6 \\
\pm 8.0 & : 7
\end{align*}\]

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.

<table>
<thead>
<tr>
<th>SAMPLE FREQUENCY</th>
<th>CUT-OFF FREQUENCY</th>
<th>SAMPLE FREQUENCY CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>51.2</td>
<td>20 Hz</td>
<td>1</td>
</tr>
<tr>
<td>204.8</td>
<td>100 Hz</td>
<td>2</td>
</tr>
<tr>
<td>512.0</td>
<td>200 Hz</td>
<td>3</td>
</tr>
<tr>
<td>2048.0</td>
<td>1 KHz</td>
<td>4</td>
</tr>
<tr>
<td>5120.0</td>
<td>2 KHz</td>
<td>5</td>
</tr>
<tr>
<td>20480.0</td>
<td>10 KHz</td>
<td>6</td>
</tr>
<tr>
<td>51200.0</td>
<td>20 KHz</td>
<td>7</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>NUMBER OF SAMPLES</th>
<th>SAMPLE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>1</td>
</tr>
<tr>
<td>512</td>
<td>2</td>
</tr>
<tr>
<td>1024</td>
<td>3</td>
</tr>
<tr>
<td>2048</td>
<td>4</td>
</tr>
</tbody>
</table>

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

CONVERT TO POWER SPECTRUM

CALIBRATE?

GROUP FREQUENCIES

DISPLAY POWER SPECTRUM GRAPH

SOUND PRESSURE LEVEL ON LOG SCALE

DISPLAY SOUN LEVEL GRAPH

EXIT?

STOP

COPY

A-3
APPENDIX B

FLOWCHART OF CODE
START

DIMENSION, LOGICAL, AND DATA STATEMENTS

SET INTEGER COMPILE SWITCH TO ONE-WORD INTEGERS

READ CALIBRATION DATA FROM FILE INCAL.CAL

ENTER TEST NUMBER

TEST = 0?

YES

STOP

NO

A

ENTER RUN NUMBER

RUN = 0?

YES

NO

1
DISPLAY BANDWIDTH VOLTS, AND SAMPLE FREQUENCY

ENTER MICROPHONE

MICROPHONE=0?

ENTER GAIN (REAL INPUT)

GAIN=99.0?

WAS CORRECT GAIN INPUT?

ENTER NUMBER OF AVERAGES

B-5
CALL ANINIT (LONG FORM)

CALL ANINIT (SHORT FORM)

CALL Much

MUCh gets data channel number from the next cell A.
CALL MLCONR

MLCONR multiplies contents of TIME by PRESS.

CALL FIXFIN

FIXFIN converts data in TIME to normalized DECIR FORM. THE DATA ARE THEN STORED IN IARRAY.

CALL DFT

DFT performs DIRECT FOURIER TRANSFORMATION DATA IN IARRAY.

ERROR?

DISPLAY ERROR MESSAGE

STOP

-- 00 I = 1, NUMBER OF SAMPLES

IARRAY(I) = IARRAY(I) * 2

CALIBRATING?

INITIALIZE DATA FOR TIME INFORMATION BLOCK

-- 00 I = 2, NUMBER OF SAMPLES

IARRAY(I) = IARRAY(I) * 2

CALIBRATING?

YES

INITIALIZE DATA FOR TIME INFORMATION BLOCK

NO

CALL MLCONR

MLCONR multiplies contents of TIME by PRESS.

CALL FIXFIN

FIXFIN converts data in TIME to normalized DECIR FORM. THE DATA ARE THEN STORED IN IARRAY.

CALL DFT

DFT performs DIRECT FOURIER TRANSFORMATION DATA IN IARRAY.

ERROR?

DISPLAY ERROR MESSAGE

STOP
CALL ZERO

ASPEC FENCIKMRS
SELF-CONJUGATE
MULTIPLY AND ADD
FOR ENERGED
ALTO SPECTRUM
FROM R27 Results
LINES ARE STORED
IN TIME ARRAY.

CALL ASPEC

CALL FLOTF

CALL MLCOR

MULTIPLY CONTENTS
OF TIME BY \( \frac{1}{2} \)
TO CORRECT VALUES
TO RMS PRESSURE
SQUARED.

B-10
A = 2.5 \times 10^9

\text{CALL MLCONR}

\begin{align*}
\text{DO } & I = 1, \text{NUMBER OF SAMPLES} \\
\text{TIME}(I) < 10 \quad & \text{YES} \\
\text{TIME}(I) = 10 \quad & \text{NO}
\end{align*}

\text{CALL LOG}

\text{LOG SERIES EVERY ELEMENT OF TIME IS \log \text{EASE} \cdot 10 \ \log \text{ARITHM}.}
CALL MLCONR

\[ \text{TIME}(I) = \text{TIME}(I) - 5\theta \]

\[ \text{TIME}(I) < \theta \]

\[ \text{TIME}(I) = \theta \]

DISPLAY BOARDERS FOR GRAPH ON SCREEN

\[ \text{SPECTRA GRAPHS} \]

\[ \frac{N}{2} \]

\[ \text{TIME HISTORY GRAPH} \]

\[ N = \text{NUMBER OF SAMPLES} \]

\[ N = 50\theta \]

\[ \text{TIME HISTORY GRAPH AND NASE} \]

11

12

[Image 0x0 to 615x784]
\[ \text{A} \max = \text{ABS} (\text{TIME}(i)) \]

\[ \text{DO } i = 2, N \]

\[ \text{TIME}(i) < 4 \times 10^{-6} \]

\[ \text{TIME}(i) = 4 \times 10^{-6} \]

\[ \text{TEST} = \log_{10} A \max \]

\[ \text{TIP} = \text{INTEGER PART OF TEST} \]

\[ \text{FP} = \text{TEST} - \text{TIP} \]

\[ \text{FACT} = 2 \]

13
C
C     FILE NAME: TF2.FOR
C     TO COMPILE: TF2=TF2/N
C     TO LINK: TF2=TF2,S,T/F
C
C
C
C
C
C
C
C
C
C
C
C
C

DIMENSION ARRAY(2048),CARRAY(2048),AIB1(6)
DIMENSION IRANGE(8),ISIZE(4),AIBC(6),ILEVEL(7)
DIMENSION DUN(1),SAN(7),SFREQ(8),TIME(2048)
DIMENSION UCAL(99),VOLT(7),AIB1(6),IFILTE(8)
LOGICAL I ACVAR(9)
LOGICAL I ACTMARK(7)
LOGICAL I ACHAR(60)
LOGICAL I ACHARM(12)
LOGICAL I ACMARK(6)
LOGICAL I 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 AHED/NHN/
DATA SAN/51,2,204,3,512,2048,5120,20480,51200/
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 compiler switch to 1-word integers ***
AIBI(6)=0.0

C*****************************
C read in calibration data from file VCAL2.DAT *
C*****************************

    Call assign2,'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 hnn ',@)
      accept 40,1test
40    format(i3)
      if (1test.eq.0) go to 999
50    type 60
60    format(1x,'enter run number hnn ',@)
      accept 40,irun
      if (irun.eq.0.000) go to 20
70 TYPE 80
80 FORMAT(1X, 'ENTER POINT NUMBER HPH ', I)
     ACCEPT 40, IPOINT
     IF (IPOINT.EQ.0.000) GO TO 50

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

   TYPE 90
90 FORMAT(1X, 'WHAT TYPE OF CALIBRATION? 0 OR N (NO <CR>)', I)
     CALL IPOSE('44', '00000')
100 ICAL = INT(I10)
     IF (ICAL.EQ.78 .OR. ICAI.EQ.79) CALL ECHO (ICAL)
     IF (ICAL.EQ.78 .OR. ICAI.EQ.79) CALL IPOSE('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 ', I)
     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) (ICAL(I),I=1,99)
     CALL CLOSE(2)
CALL ERASE('KB')

CHECK USER PROMPTS FOR VOLTAGE CODE AND SAMPLE FREQUENCY #1:

TYPE 150

150 FORMAT (3x,'VOLTS',8x,'ENTER',5x,'SAMPLE-FREQUENCY'),
     4x,'CUT-OFF FREQUENCY',5x,'ENTER',1x,'+',/2 0.125',7x,'1',13x,
     5',1.6x,'30 HZ',13x,'1',1x,'+',/2 0.25',7x,'2',12x,'204.8',
     15x,'100 HZ',13x,'2',1x,'+',/2 0.5',3x,'3',12x,'512.0',
     15x,'300 HZ',13x,'3',1x,'+',/2 1.0',3x,'4',11x,'2048.0',
     16x,'1 KHZ',13x,'4',1x,'+',/2 2.0',3x,'5',11x,'5120.0',
     16x,'2 KHZ',13x,'5',1x,'+',/2 4.0',3x,'6',10x,'20480.0',
     15x,'10 KHZ',13x,'6',1x,'+',/2 8.0',2x,'7',10x,'51200.0',
     15x,'20 KHZ',13x,'7')

160 TYPE 170

170 FORMAT (2x,23x,'ENTER VOLTAGE CODE ',#)

ACCEPT 180,INDEX

180 FORMAT (2x)

190 IF ((INDEX.GE.1) AND (INDEX.LE.7)) GO TO 220

IF (INDEX.LE.0) GO TO 200
CALL ERASE('KB')

GO TO 70

200 TYPE 210

210 FORMAT (2x,20x,'WHAT? VOLTAGE CODE IS 1-7!')

GO TO 160

220 TYPE 230

230 FORMAT (2x,19x,'ENTER SAMPLE-FREQUENCY CODE ',#)

ACCEPT 180,INDEX

IF ((INDEX.GE.1) AND (INDEX.LE.7)) GO TO 250

IF (INDEX.LE.0) GO TO 160
TYPE 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(/.,18X,'NUMBER OF SAMPLES',',,' ENTER',/.,T28,'128'
1 ,T43,'1',/.,T28,'512',T43,'2',/.,T27,'1024',T43,'3',/.,T27,
1 '2048',T43,'4')

C**** BANDWIDTH, VOLTS, AND SAMPLE FREQUENCY OUTPUT *****
310 BDW= SAMK INDEXZ/ISIZE(INDEXZ)
   TYPE 320,BDW
320 FORMAT (/.,19X,'BANDWIDTH = ',F10.3)
   TYPE 321,VOLT(INDEXZ)
321 FORMAT (/.,13X,'VOLTS = +/- ',F5.2)
   TYPE 322,SAMK INDEXZ)
322 FORMAT (/.,13X,'SAMPLE FREQUENCY = ',F8.2)

C**** USER PROMPT FOR MICROPHONE, GAIN, AND AVERAGES *****
323 TYPE 324
324 FORMAT(/.,19X,'ENTER MICROPHONE NN ',1)
   ACCEPT 180,IMIK
IF (IMIK.EQ.0) GO TO 250
325 TYPE 326
326 FORMAT ('+ ',/,'GAIN IS ',F5.1,' Y OR N (NO <CR>)',$)
   ACCEPT 327,GAIN
327 FORMAT (F5.1)
328 IF (GAIN.EQ.99.) GO TO 323
   TYPE 329,GAIN
329 FORMAT(19X,'ENTER GAIN (REAL INPUT) ',F5.1)
   CALL IPOKE("44,"10000)
330 IDU = IIN
   IF (IDU.EQ.78.OR.IDU.EQ.89) CALL ECHO (IDU)
   IF (IDU.EQ.78.OR.IDU.EQ.89) CALL IPOKE("44,"00000)
   IF (IDU.EQ.78) GO TO 325
   IF (IDU.EQ.89) GO TO 331
   GO TO 330
331 TYPE 332
332 FORMAT('+ ',/,'DATA ACQUISITION STARTS WHEN <CR> IS ENTERED',
   /,'ENTER NUMBER OF AVERAGES MNN ',$)
   ACCEPT 333,NOAVE
333 FORMAT(13)
   IF (NOAVE.EQ.0) GO TO 325
   GO TO 410

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

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

400 INDEX=8
   INDEXZ=4
   INDEXL=5
PARAMETERS AND DATA FOR ACQUISITION OF ANALOG DATA

CALL ANINIT(IARRAY,AIBL,DUN,DUN,ISIZE(INDEXZ),1,
  1 SRE(DK,INDEXS),IRANGE(INDEXZ),TLEVEL(INDEXZ),IER,0,6,
  1 IFILTER(INDEXS),1)
  IF (IER.EQ.0) GO TO 430
  TYPE 420,IER

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

CALL ANINIT(IARRAY,AIBL,DUN,DUN,ISIZE(INDEXZ),1,
  1 SRE(DK,INDEXS),IRANGE(INDEXZ),TLEVEL(INDEXZ),IER,0,6,
  1 IFILTER(INDEXS),1)
  IF (IER.EQ.0) GO TO 430
  TYPE 420,IER

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

CALL ZER0(TIME,AIBT)
  DO 460 I=1,NOAVE
CALL AINHP(IF RAME, IER)
IF (((IF RAME .GE. 0).AND.(IER.GE.0)) GO TO 462
TYPE 461,IER,IF RAME

461 FORMAT (/ / 5X, 'ERROR AINHP'. I5, I5X, 'FRAME CODE'. I5)
STOP

462 IF (ICAL. E0.78) GO TO 500
AIBC(2)=AIBK(2)
CALL FLOHP(IF RAME,AIBC,AARRAY,AIBC)
CALL ADD(CARRAY,AIBC,TIME,AIBT)

CONTINUE
GO 470 N=1,ISIZE(INDEXZ)

470 TIME(N)=TIME(N)/HOAVE

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

480 BETA=10.0*(-(GAIN)/20.) "'
PRESUR=(BETA*(31.7*ICAL(IMIK)))
CALL MLCOR(PRESUR,TIME,AIBT)
LG=1

C*GRAPH TIME HISTORY **
GO TO 630

C* DATA FOR POWER SPECTRUM GRAPH *

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',15)
 STOP

C** CORRECT FOR ONE-SIDED DFT **

520  DO 521 I=2,1,SIZE(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)
**CONVERT DATA TO FORTRAN FLOATING POINT FORM**

    CALL FLOTF(TIME,AIBT)

**CORRECT VALUES TO RHS PRESSURE SQUARED**

    CALL MLCONR(.5,TIME,AIBT)

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

    IF (ICAL.EQ.78) GO TO 540

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

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

**SET LOWER LIMIT FOR POWER SPECTRA**

    DO 530 I=1,ISIZE(INEXZ)
    530 IF (TIME(I).LT.(4.0E+10.0*E(-9.))) TIME(I)=4.0E(10.0*E(-9.))

**GRAPH POWER SPECTRUM**

    LG = 2
    GO TO 630

**SUM MAGNITUDES SQUARED FOR CALIBRATED DATA**

    540 UCAL(IMIK)=0
DO 550 I=400,800
  VCAL(IMIK)=VCAL(IMIK)+TIME(I)
  VCAL(IMIK)=SQRT(VCAL(IMIK))
  TYPE 560,IMIK
560  FORMAT(2X,'WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE ',1X,13,' (REAL INPUT)')
      ACCEPT 570, CALFAC
570  FORMAT (618.3)
      VCAL(IMIK)=VCAL(IMIK)*CALFAC
      TYPE 580,IMIK,VCAL(IMIK)
580  FORMAT(2X,'VCAL(',.12,'') = ',F8.6)

C### PROMPT FOR NEXT MICROPHONE C###

GO TO 110

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

C### CONVERT TO DB SCALE C###

600  A = 2.5E10,49.
    CALL MLCONR(A,TIME,AIMB)
    DO 610 I=1,ISIZE(INDEXZ)
610  IF (TIME(I).LT.10.) TIME(I)=10.
    CALL LOG(TIME,AIMB)
    CALL MLCONR(10.,TIME,AIMB)
    DO 602 I=1,ISIZE(INDEXZ)/2
      TIME(I) = TIME(I)-50.
602  IF (TIME(I).LT.0.) TIME(I)= 0.
    LG=3
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',469,760,12,'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,ACHAR) NOAVE
FORMAT(13,12,'AVERAGES')
CALL ALPHA('KB',460,729,12,ACHAR)
IF (LG.EQ.1) CALL ALPHA('KB',1,500,9,'AMPLITUDE')
IF (LG.EQ.1) CALL ALPHA('KB',1,461,7,'(N/MIN)')
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/MIN)')
IF (LG.EQ.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) ITST,IRUN,IPINT,IMIK
FORMAT('TEST = ',I3.4X,'RUN = ',I3.4X,'POINT = ',I3,14X,'MICROPHONE = ',I2,3X)
CALL ALPHA('KB',145,20,60,ACHAR)
**Y-AXIS SCALE**

```plaintext
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
```

**FIND MAXIMUM AMPLITUDE**

```plaintext
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
```

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

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

**COMPUTE SCALING FACTOR**

```plaintext
680 TEST=ALOG10(AMAXAP)
TIP=AINT(TEST)
FP=TEST-TIP
FACT = 2.
IF (FP.GE..25) FACT = 5.
IF (FP.GE..40) FACT = 10.
IF (FP.GE.90) FACT = 20.
SCALEY=(10.*TIP)*FACT
ISCLY=INT(SCALEY)
IF (LG.EQ.3) GO TO 720
```
**Tic Marks for Y Axis Graphs 1 and 2**

ITICY = 122
DO 605 I = 1, 10
CALL BEAMP('KB',141,ITICY)
CALL VECTOR('KB',125,ITICY)
605 ITICY = ITICY + 60
CALL BEAMP('KB',141,721)
CALL VECTOR('KB',125,721)

**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(5,620,ACMAX) ISCLY
690 FORMAT (I4, ' ')
IF (LG.NE.2) CALL ALPHA('KB',20,720,5,ACMAX)
IF (LG.EQ.2) ENCODE(9,700,ACMAX) ISCLY
700 FORMAT (I8, ' ')
IF (LG.EQ.2) CALL ALPHA('KB',1,720,9,ACMAX)
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

**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)
CALL ALPHA('K8', 40, 720, 4, '130.')  
CALL ALPHA('K9', 55, 122, 3, '50.')  

CALL TIC  
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 = SAMK(INDEXZ)/2  
IF (LG.NE.1) SCALEX = 1764./IMAX  
NP = INT(832./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./26.

CALL TIC FOR X AXIS

IF (LG.EQ.1) AMAXAP=NP/(BOW*ISIZE(INDEXZ))  
TEST=Aalog10(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.**K* #IP))
DELMARK=VALKEY/10.
NMARKS=1+INT(AMAXAP/DELMARK)
DO 750 N=1,NMARKS
NHL=N-1
IXMARK = 141+INT(NHL*DELMARK*32/AMAXAP)
CALL BEAMP('KB',IXMARK,122)
CALL VECTOR('KB',IXMARK,110)
IF (N.EQ.1) CALL ALPHAK('KB',IXMARK,75,2,'0.')
IF (N.NE.1) GO TO 750
CALL BEAMP('KB',IXMARK,122)
CALL VECTOR('KB',IXMARK,100)
IF (LG.EQ.1) ENCODE(7,742,ACTMAR) VALKEY
742 FORMAT(F7.5)
IF (LG.EQ.1) CALL ALPHAK('KB',IXMARK-49,75,7,ACTMAR)
IF (LG.NE.1) ENCODE(6,744,ACMAX) VALKEY
744 FORMAT(ES.0)
IF (LG.NE.1) CALL ALPHAK('KB',IXMARK-49,75,6,ACMAX)
750 CONTINUE

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

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

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

ACCEPT 770
770 FORMAT(1A1)
****** IF DONE WITH TIME HISTORY THEN GO AND CALCULATE DATA FOR
****** POWER SPECTRUM GRAPH ****

IF (LG.EQ.1) GO TO 490

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

IF (LG.EQ.2) GO TO 600

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

CALL ERASE('KB')
GO TO 323

**$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
** PROGRAM ENDS *
**$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

999 STOP
END

**$$$$$$$$$$$$$$$
C
C ECHO SUBROUTINE, TO ECHO CHARACTERS ON SCREEN
C
SUBROUTINE ECHO (ICAL)
CALL IPOKE ("44,"00000)
I=ITTO(ICAL)
CALL IPOKE ("44,"10000)
RETURN
END
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? 0 OR H. (NO, (CR)) H
ENTER NUMBER OF MICROPHONE TO CALIBRATE 01
WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE 1. 1 (REAL INPUT)
 \[ \text{VCAL}(1) = 0.579954 \]
ENTER NUMBER OF MICROPHONE TO CALIBRATE 05
WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE 5. 1 (REAL INPUT)
 \[ \text{VCAL}(5) = 0.579961 \]
ENTER NUMBER OF MICROPHONE TO CALIBRATE 07
WHAT IS THE CALIBRATION FACTOR FOR MICROPHONE 7. 1 (REAL INPUT)
 \[ \text{VCAL}(7) = 0.580003 \]
ENTER NUMBER OF MICROPHONE TO CALIBRATE 00
<table>
<thead>
<tr>
<th>VOLTS</th>
<th>ENTER</th>
<th>SAMPLE-FREQUENCY</th>
<th>CUT-OFF FREQUENCY</th>
<th>ENTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>+/- 0.125</td>
<td>1</td>
<td>51.2</td>
<td>20 Hz</td>
<td>1</td>
</tr>
<tr>
<td>+/- 0.25</td>
<td>2</td>
<td>204.8</td>
<td>100 Hz</td>
<td>2</td>
</tr>
<tr>
<td>+/- 0.5</td>
<td>3</td>
<td>512.0</td>
<td>200 Hz</td>
<td>3</td>
</tr>
<tr>
<td>+/- 1.0</td>
<td>4</td>
<td>2048.0</td>
<td>1 kHz</td>
<td>4</td>
</tr>
<tr>
<td>+/- 2.0</td>
<td>5</td>
<td>5120.0</td>
<td>2 kHz</td>
<td>5</td>
</tr>
<tr>
<td>+/- 4.0</td>
<td>6</td>
<td>20480.0</td>
<td>10 kHz</td>
<td>6</td>
</tr>
<tr>
<td>+/- 8.0</td>
<td>7</td>
<td>51200.0</td>
<td>20 kHz</td>
<td>7</td>
</tr>
</tbody>
</table>

ENTER VOLTAGE CODE 6

ENTER SAMPLE-FREQUENCY CODE 5
NUMBER OF SAMPLES

128
512
1024
2048

ENTER SAMPLE CODE, 2

BANDWIDTH = 10.000

VOLTS = +/- 4.00

SAMPLE FREQUENCY = 5120.00

ENTER MICROPHONE H1 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 H10 001
Power Spectrum
1 Averages

Pressure Squared (N^2/m^2)

0.0

0.0

Frequency (Hz)

0.0

2000.

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 NNH 010
<table>
<thead>
<tr>
<th>NUMBER OF SAMPLES</th>
<th>ENTER CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>1</td>
</tr>
<tr>
<td>512</td>
<td>2</td>
</tr>
<tr>
<td>1024</td>
<td>3</td>
</tr>
<tr>
<td>2048</td>
<td>4</td>
</tr>
</tbody>
</table>

**Enter Sample Code:**
- 1: $10,000$
- 2: $4,000$

**Sample Frequency:** $520.60$

**Enter Microphone:** In 87

**Enter Gain (Real Input):** 0.

**Gain is 0, Y or N (CR)?**

**Data Acquisition Starts When (CR)?**

**Enter Number of Averages Min 010**
POWER SPECTRUM
10 AVERAGES

PRESSURE SQUARED (N/MM)

FREQUENCY (HZ)

TEST = 1  RUN = 1  POINT = 1  MICROPHONE = 7
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.
End of Document