A Computer Program for Processing Impedance Cardiographic Data: Improving Accuracy Through User-Interactive Software

Patricia S. Cowings, Karen Naifeh, and Chet Thrasher

October 1988
A Computer Program for Processing Impedance Cardiographic Data: Improving Accuracy Through User-Interactive Software

Patricia S. Cowings, Ames Research Center, Moffett Field, California
Karen Naifeh
Chet Thrasher, University of California, San Francisco, California

October 1988
A COMPUTER PROGRAM FOR PROCESSING IMPEDANCE CARDIOGRAPHIC DATA:

IMPROVING ACCURACY THROUGH USER-INTERACTIVE SOFTWARE

Patricia S. Cowings, Karen Naifeh,* and Chet Thrasher*

Ames Research Center

SUMMARY

This report contains the source code and documentation for a computer program used to process impedance cardiography data. The cardiodynamic measures derived from impedance cardiography are ventricular stroke volume, cardiac output, cardiac index and Heathen index. The program digitizes data collected from the Minnesota Impedance Cardiograph, electrocardiography (ECG), and respiratory cycles and then stores these data on hard disk. It computes the cardiodynamic functions using interactive graphics and stores the means and standard deviations of each 15-second data epoch on floppy disk. This software was designed on a Digital PRO380 microcomputer and used version 2.0 of P/OS, with (minimally) a 4-channel 16 bit analog/digital (A/D) converter. Applications software is written in Fortran 77, and uses Digital's Pro-tool Kit Real Time Interface Library (PRTIL), CORE Graphic Library (CGL), and laboratory routines. Source code can be readily modified to accommodate alternative detection, A/D conversion and interactive graphics. The object code utilizing overlays and multitasking has a maximum of 50 Kbytes.

INTRODUCTION

The Psychophysiological Research Laboratory of the Neurosciences Branch at NASA Ames Research Center has been engaged in a series of ground-based investigations of human autonomic responses to motion sickness. With the final goal of developing a treatment for the motion sickness-like symptoms which affect astronauts exposed to the microgravity environment of space, our group uses noninvasive electrophysiological measures to document changes in physiological activity levels in different subject populations. In the course of this research, we have found that measures of cardiovascular function are very sensitive indices of the malaise levels experienced by test participants. In previous studies (ref. 1), it was observed that high-susceptibility subjects tended to produce more labile and larger magnitude changes in heart rate and blood volume pulse (a relative measure of peripheral resistance), when exposed to motion sickness stressors than low-susceptibility subjects. We decided to investigate this result further.

---

*University of California, San Francisco, CA.
Although electrocardiography (ECG) provides excellent information regarding the electrical function of the heart muscle, it gives no definitive information regarding physical function. Heart rate alone is insufficient for determining the level of sympathetic influence on cardiac muscle. An increase in heart rate can be caused by either an increase in sympathetic tone or a decrease in parasympathetic activation. Finger pulse volume, although easy to monitor, is not as reliable an index of sympathetic tone of the vasomusculature as total peripheral resistance, which is derived from both blood pressure and cardiac output (ref. 2).

We needed a noninvasive, atraumatic means of examining human cardiodynamics in a motion-sickness inducing environment. Our assessment of impedance cardiography is that it provides relatively accurate data on a wide range of dynamic cardiovascular variables and appears even to be sensitive to rapid changes in these variables. Impedance cardiography techniques enable the measurement of myocardial contractility, (heater index) which is directly related to the level of sympathetic innervation of the heart. Even when there is no apparent change in heart rate, changes in stroke volume or myocardial contractility may be occurring which reflect significant differences in autonomic function (refs. 2,3). Other valuable information obtained from this device include cardiac output and central volume (transthoracic impedance).

Miller and Horvath (ref. 3) describe the advantages and drawbacks of impedance cardiography and compare the accuracy of this technique to other invasive and noninvasive measures of cardiodynamics. The principal disadvantage is that erroneous results may be produced if the user is unfamiliar with the effects of artifact on the computation of specific data epochs and is not careful to select cardiac cycles (dZ/dT peaks) that occur only during the expiratory pause between breaths. The Minnesota Impedance Cardiograph, manufactured by Instrumentation for Medicine, Inc., Greenwich, Conn., is available with a (firmware only) microprocessor which is designed to automate this process. However, we found that with the microprocessor, one can obtain data from one heartbeat at a minimum of 8-sec intervals. Thus much data are lost from the intervening beats, especially when rapid changes are occurring in the cardiovascular system.

This paper contains the source code and operator's instructions for a user-interactive program written in support of our research. The program provides more accurate calculations of cardiac parameters based on a greater quantity of data. The sampling rate for digitizing data is 200 samples/sec; the data can be sampled at twice real-time speed from analog tape. The collected data consist of the first derivative of pulsatile thoracic impedance change (dZ/dt), basal impedance, electrocardiogram, and respiration waveforms. During an interactive graphics session, data for dZ/dt, ECG and chest respiration are displayed on a monitor in 15-sec epochs. The user can select from the screen the dZ/dt peaks that are used to calculate cardiodynamic functions. The program computes and stores the means and standard deviations for the cardiac measures for each 15-sec epoch.
CALCULATIONS PERFORMED:

Stroke Volume = \( \frac{PL^2 T(dZ/dt)_{min}}{Z_0^2} \)

\( p = 53.2 \ e^{(0.002)\text{(Hematocrit)}} \)

Expressed in ohms-cm

\( e = 2.7183 \) the natural exponent

Hematocrit expressed in units of percent

\( L = \) distance between two inner electrodes in cm

\( Z_0 = \) Transthoracic Impedance expressed in ohms

Cardiac Output = (Stroke Volume)(Pulse rate)/1000

Expressed in liters per min

Pulse rate is expressed in beats per min

Cardiac Index = Cardiac Output/Body Surface Area

Expressed in liters per min

Body surface area = \( H \times W \times 0.007184 \)

Expressed in m\(^2\)

\( H = \) height in cm

\( W = \) weight in Kg
Heather Index = \((dZ/dt)_{\text{min}}/R-Z\)

Expressed in ohms per sec\(^2\)

\(R-Z\) = interval between the R-wave of the ECG and the peak of \(dZ/dt\) expressed in sec

---

REQUARED HARDWARE AND SUBJECT INFORMATION

Hardware for this research includes a Minnesota Impedance Cardiograph, a respiration transducer (e.g., a piezoelectric or mercury strain gage) and a preamplifier capable of producing an analog output signal of respiratory responses, electrocardiography equipment (either the Minnesota impedance cardiograph for direct measurement or an ECG amplifier for external measurement), a Digital PRO380 microcomputer which uses version 2.0 of P/OS, and a four-channel A/D converter (16-bit). Applications software, written in Fortran 77, uses Digital's Pro-tool Kit Real Time Interface Library (PRTIL), CORE Graphic Library (CGL), and laboratory routines. Source code can be readily modified to accommodate alternative routines for peak detection, A/D conversion and interactive graphics. The object code uses overlays and multitasking and has a maximum of 50 Kbytes.

Additional subject information required to implement this software is (a) hematocrit count; (b) weight in Kg; (c) height in cm; and (d) the distance between the two inner impedance cardiography electrodes (tapes), measured both in front and back and then averaged.

This software is installed on a DEC PRO 380 computer by following applications installation instructions in Professional Developer's Toolkit Reference Manual, Chapters 3 and 6.
OPERATOR INSTRUCTIONS

Three options will appear on the screen:

*************************************************************************
*
* SELECT FROM THE FOLLOWING
*
* 1. DIGITIZE A DATA FILE FROM TAPE
*
* 2. DATA REDUCTION OF DIGITIZED FILE
*
* 3. EXIT
*
*************************************************************************

[ENTER 1, 2, OR 3]:

Procedure for digitizing a tape data file--SELECTION 1. If the user selects option 1, the following displays appear on the screen:

*************************************************************************
*
DIGITIZING A DATA FILE
*
*************************************************************************

PLEASE ENTER RUN ID FOR THIS FILE:

1. User enters an ID of 6 digits.

PLEASE ENTER RUN TIME (IN SECS.) FOR TAPE FILE:

2. Determine maximum duration of data file in seconds (e.g. 30 minutes=1800 seconds); ADD 60 seconds to run length (e.g. , 1860). ENTER THIS NUMBER. NOTE: an additional 60 seconds is added to the file length to accommodate acquisition of calibration data.

HOW FAST TO SAMPLE THE DATA ON TAPE?

ENTER 1 FOR REAL-TIME

ENTER 2 FOR TWICE REAL TIME

[ENTER 1 OR 2]:

5
3. The option of digitizing data at twice real-time is determined by the memory capacity (512 K RAM) and the I/O response time (less than 0.025 sec) of the system implementing this program. This I/O response time is required to store four-channels of data, in two-byte segments. After entering 1 or 2, the screen displays:

HIT "S" TO START, "P" TO PAUSE, OR "A" TO ABORT:

FOLLOWED BY A [RETURN]

4. Position analog tape to beginning of run, then press the keys "S" and "Return" to begin computer acquisition of high calibration data. Data acquisition may be paused at any time by pressing the key "P" and will not continue until the user presses the "RESUME" key. Acquisition may be aborted at any time by pressing the key "A", at which time the program prompts to the user to either SAVE or DELETE the created digitized file. If the user has chosen to start data acquisition (i.e., have pressed the key "S"), the screen will display a flashing message and three columns will scroll continuously until acquisition ends. The first column indicates the active buffer (switches between buffers 1 and 2). The second column indicates error status (e.g., if A/D is turned off). No error is indicated by "0". The third column displays the data second currently being acquired.
5. After 30 seconds of high calibration data, the program will automatically pause. Advance tape to start of low calibration data. Press the key "RESUME". Again, the display indicates that data acquisition is in progress.

6. After 30 seconds of low calibration data, the program will automatically pause. When tape has been positioned at start of the data session, computer acquisition is started by pressing the key "RESUME".

7. When the total number of seconds defining this file have elapsed the program screen display will direct the user to: SAVE the file; or DELETE the created digitized file. The screen displays:
SELECT ONE OF THE FOLLOWING [ENTER 1 OR 2]

1 TO SAVE DIGITIZED FILE

2 TO DELETE DIGITIZED FILE

8. If the user has chosen to SAVE the file, it is written to hard disk, and the user may now proceed with data reduction to calculate stroke volume, cardiac output, cardiac index and Heather index. The original menu is displayed:

************
* SELECT FROM THE FOLLOWING *
* 1. DIGITIZE A DATA FILE FROM TAPE *
* 2. DATA REDUCTION OF DIGITIZED FILE *
* 3. EXIT *
************

[ENTER 1, 2, OR 3]:

Procedure for data reduction--SELECTION 2

1. The screen will request input from the user as follows:

PHASE 1 -- KEY FIELD ENTRY

**************************************************
ENTER RUN ID FOR THIS FILE: (6 digit file name used when digitizing).
ENTER LENGTH (CM) BETWEEN INNER ELECTRODES: (e.g., 26.75)
ENTER HEMATOCRIT COUNT: (e.g., 40)
ENTER HEIGHT (CM): (e.g., 183)
ENTER WEIGHT (KG): (e.g., 94.5)

PHASE 2 -- CALIBRATION VALUES

**************************************************

2. The display will show the calibration levels set internal to the program which the user may choose to modify. These default values are:
BASE IMPEDANCE (OHMS) 

\[
\begin{array}{cc}
\text{High} & \text{Low} \\
25.0000 & 0.0000 \\
1.0000 & 0.0000 \\
\end{array}
\]

WOULD YOU LIKE TO CHANGE THESE VALUES (Y/N)

NOTE: If the user enters "N", the screen will display:

```
-----------------------------------------------------------------------------------
PHASE 3 -- CALIBRATION ACQUISITION
-----------------------------------------------------------------------------------
CALIBRATION OF BASELINE IMPEDANCE
-----------------------------------------------------------------------------------

<table>
<thead>
<tr>
<th>CHANNEL</th>
<th>SIGNAL</th>
<th>A/DHIGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IMPEDANCE</td>
<td>5400</td>
</tr>
</tbody>
</table>
```

DO YOU WANT TO RE-RUN HIGH CALS [Y/N]?

NOTE: If the user enters "Y" then the program will return to the original menu and user must redigitize calibration data from tape. The condition that would require a "Y" response here is obtaining an A/DHIGH value which does not correspond to the voltage out of the Minnesota Cardiograph's internal "Hi CAL" setting. In this example, an A/D value of 5400 equals 0.8 volts. For additional information on determining the ratio of A/D values to voltage, refer to the: Pro/Tool Kit Real-Time Interface Library (PRTIL) User's Guide, Chapter 7, p. 12.

If the user enters "N" then the screen displays

```
<table>
<thead>
<tr>
<th>CHANNEL</th>
<th>SIGNAL</th>
<th>A/DLOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IMPEDANCE</td>
<td>433</td>
</tr>
</tbody>
</table>
```

DO YOU WANT TO RE-RUN LOW CALS [Y/N]?

NOTE: If the user enters "Y" then the program will return to the original menu and user must redigitize calibration data from tape. If the user enters "N" then the screen displays:
NOTE: If the user enters "Y" then the program will return to the original menu and user must redigitize calibration data from tape. If the user enters "N" then the screen displays:

```
CALIBRATION OF dZ/dt SIGNAL
```

```
CHANNEL  SIGNAL  ADLOW  ADHI  LOWCAL  HICAL
-------  ------  -----  -----  ------  ------
2        dZ/dt  491    1495   0.0000  1.0000
```

SLOPE 9.360159E-04  INTERCEPT -0.4890438

WOULD YOU LIKE TO REPEAT CALIBRATIONS [Y/N]?

NOTE: If the user enters "Y" then the program will return to the original menu and user must redigitize calibration data from tape. If the user enters "N" and this is the first time these data are to be analyzed, the program will display an informative error message: -FILE DOES NOT EXIST-. It will then create an interim file and proceed directly to DATA REDUCTION AND EDITING. If this is an editing session, the screen will display a list of all 15-second data epochs from which the user has selected at least one waveform for processing. If no waveform was chosen by the user for a particular epoch, that epoch is "ZEROED" (i.e., the epoch number is not listed below and will not be used). A 10-min file where all epochs contain selected data is displayed as follows:

```
FROM THE RECORD, THE FOLLOWING EPOCHS ARE NONZERO:

1  2  3  4  5  6  7  8  10  11
12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31
32 33 34 35 36 37 38 39 40
```

PRESS [RETURN] TO CONTINUE
When the [RETURN] key is pressed, the screen will display:

***********************************************************************
DATA REDUCTION AND EDITING
***********************************************************************

ENTER STARTING TIME CODE FOR DATA ON TAPE (e.g., enter 300 for 3:00).
ENTER EPOCH FOR START OF DATA REDUCTION: (e.g., 1)
ENTER LAST DATA EPOCH: (e.g., 40)

The time code entry is recordkeeping information which is not used by the
program. It is here for the convenience of the user so that he may keep track of
"where" in a particular tape file data reduction was started. If this is an editing
session, the user may choose to enter only those epochs needing correction, i.e.,
new waveforms will be selected. At this point, the screen will display the data of
the first 15-sec epoch selected:

Figure 1.- Example of a screen display showing a 15-sec data epoch.
The first waveform displayed in this figure is the dZ/dt. The horizontal lines through this waveform represent high cal level (upper line) and the low cal level (lower line). The two horizontal dots beneath each dZ/dt waveform mark the point where the rising (major peak) dZ/dt waveform first crosses the low-cal line and the maximum trough (downward spike) following this peak. The distance between these two dots (for a single waveform) is the ventricular ejection time.

The numerics displayed beneath the dZ/dt identify this epoch number and time code at the start of this epoch. In this figure, the display 1 300 means epoch 1, time code 3:00.

The second waveform is the electrocardiogram (ECG) and the horizontal dots beneath the ECG mark the peak of the R-wave. The third waveform shows respiratory cycles (Note that this waveform is inverted, such that the top of each cycle displayed actually represents the expiratory pause between breaths). Vertical dashed lines denote the dZ/dt peaks and their relation to ECG and respiration signals. Each dZ/dt peak (vertical dashed lines) is numbered at the bottom of the screen.

While this display is on the screen, the user is prompted with questions (displayed in upper left corner). The first question is:

**SKIP THIS EPOCH? [ENTER Y/N]:**

If the user does not want to have this data epoch represented in the stored output file, entering "Y" would cause the program to display the second 15-sec data epoch. No data from the first epoch are stored, and the second epoch is renumbered as epoch 1. If the user enters "N" in response to this question, the next query displayed is

**REDO ANALYSIS? [ENTER Y/N]:**

One condition that would make the user choose to redo the analysis is that the dZ/dt waveform is significantly distorted by artifact (e.g., electrical noise or movement) and therefore the horizontal dots beneath the waveform are incorrectly positioned (i.e., ventricular ejection time is incorrectly labeled). Answering "Y" to this question results in the prompt

**ENTER GATE AND DIFFERENCE FACTORS SEPARATED WITH A COMMA:**

**NOTE:** The GATE and DIFFERENCE factors are indigenous to the peak detection subroutine used in this program. They are used for determining the peak and delta-t (ventricular ejection time) values of the dZ/dt waveform. Both factors provide a criterion for distinguishing between high frequency noise and significant trends in the waveform. The GATE factor represents the minimum number of consecutive data points (A/D values) needed to establish a trend. The DIFFERENCE factor represents the minimum difference in magnitude between data points that is needed to establish a real increase in the trend. If the current maximum A/D value is greater than the current maximum plus the minimum DIFFERENCE (current minus previous maximum A/D
value), then the counter is incremented. When the counter is equal to the GATE, a trend toward significant peak is established. The default value for GATE is 2 and for DIFFERENCE is 100. If the user feels that the points marking delta-t are "too close together", then a higher DIFFERENCE factor is entered (e.g., 2,200 or 2,300). If the delta-t marks are "too far apart", then a lower DIFFERENCE factor is entered (e.g., 2,50 or 2,75). The GATE factor is rarely modified. After new GATE or DIFFERENCE factors have been entered, the screen blanks, and this epoch is again displayed with the dots marking delta-t repositioned. This process can be repeated until the user is satisfied that most dZ/dt waveforms in this epoch have delta-t correctly marked. If the user enters "N", the next query displayed is

OK TO CALCULATE HEATHE INDEX? [ENTER Y/N]:

If the user enters "N", the output file would contain zeroes for the heather index. The condition that would make the user choose to zero the heather index is when the ECG waveform is so distorted by artifact that the peak of the R-wave cannot be detected (i.e., marked by a dot) by the program. If the user enters "Y", the output file will contain the calculated heather index for those waveform peaks selected below. The next instruction from the program is

LIST SELECTED PEAKS, SEPARATE WITH :COMMA: (e.g., 1,5,9,10,14,15)

The criteria for selecting a peak are: (1) no significant artifact should be present in this waveform (and in the waveform immediately preceding it), (2) delta-t must be correctly marked, and (3) waveforms must occur during the exhalation respiratory plateau (pause between breaths). Refering to figure 1, the user would likely select peaks 1,5,9,10,14, and 15.

After making peak selections, the user presses the "RETURN" key and is prompted with the query

DO YOU WANT A COPY [Y/N]:

Answering "Y" will produce a hard copy of the epoch being displayed on the screen. Answering "N" will prompt the query

DO YOU WANT TO EXIT [Y/N]:

Entering "N" causes the program to display the next 15-sec epoch of data from which the user may select peaks for data reduction. This is repeated until all 15-sec data epochs of this file have been processeled. Note that if the user has just completed the last epoch of data in this file, the program terminates automatically (i.e., responds as though a "Y" was entered). Entering "Y" terminates the data reduction phase of the program and the screen will display options for user selection:
ENTER 1 TO SAVE DATA ON FLOPPY AND EXIT
ENTER 2 TO EXIT (SAVE DATA ON HARD DISK)
ENTER 3 TO RESTART DATA EDITING
ENTER 4 TO RESTART CALIBRATION ACQUISITION

[ENTER 1, 2, 3 OR 4]:

Entering "1" will cause the program to prompt the user for additional key field information used to identify this file. The screen displays

GROUP TYPE (maximum of 8 characters): (e.g., CONTROL)

RUN NUMBER: (e.g., 1)

DIRECTION (cc,cw,nd): NOTE: cc=counterclockwise, cw=clockwise
nd=no direction (i.e., no rotation)

SUBJECT'S INITIALS (first and last): (e.g., PC)

AGE: (e.g., 49)

SUSCEPTIBILITY (1,2,3): NOTE: 1 = high motion sickness
susceptible, 2 = moderate and
3 = low.

SEX (m/f): (e.g., M)

TEST DATE (mmmddyy): (e.g., MAY0288)

TEST TIME (military in hr, min, e.g., for 1:00 pm enter 1300):

PRE HEART RATE (in beats/min): (e.g., 68)

PRE TEMPERATURE: (e.g., 97.6)

PRE B.P. (sys,dia, e.g., 120,80):

PRE/POST TEST BASELINE (minutes): (e.g., 10)
PLEASE INSERT DATA FLOPPY WITH AT LEAST 60 CONTIGUOUS FREE BLOCKS
INTO DRIVES DZ1 & DZ2. WHEN READY, PRESS RETURN

ARE YOU FINISHED WITH DIGITIZING DATA? [Y/N]

Entering "Y" will prompt the user as follows:

DO YOU WANT TO DELETE THE DIGITIZED FILE? [Y/N]:

(Note: After this input, the program returns
the user to the original menu).

-----------------------------------------------

Entering "2" will return the user to the original menu for this
program.

Entering "3" will return the user to the editing portion of
this program.

Entering "4" will return the user to the calibration acquisition
portion of this program.
REFERENCES


APPENDIX

SOURCE CODE

PROGRAM IMPMLT
AN IMPEDANCE CARDIOGRAPH PROGRAM USING DIGITIZED DATA FILES

PARAMETER (IFREQ=200)
DIMENSION SLOPE(2),ENTRCP(2),CALVAL(2,2)
COMMON ISPACE(4*IFREQ),SPACE(30*IFREQ)
REAL*8 DRAW,IMPCAL,IMPSTR

DATA IMPCAL/'IMPCAL'/,IMPSTR/'IMPSTR'/
DATA DRAW/'DRAWRM'/,ICLEAN/27,99/,IMPCAL/'IMPCAL'/
DATA CLEANS/27,99/,CATFIL/'00000000000000'

TYPE 500,CLEANS
500 FORMAT(X,4A1)
TYPE 101
101 FORMAT(10X,60(' '),//,27X,'IMPEDANCE CARDIOGRAPH PROGRAM', 1//,10X,60(' '),///)
125 CONTINUE
CATFIL(1:14)='00000000000000'
TYPE 102
100 FORMAT(20X,45('**'),/20X,'**',
  1 /,20X,'**',6X,' SELECT FROM THE FOLLOWING',/20X,'**',
  1 /,20X,'**',10X,' 1. DIGITIZE A DATA FILE FROM TAPE',
  2 /,20X,'**',10X,' 2. DATA REDUCTION OF DIGITIZED FILE',
  3 /,20X,'**',10X,' 3. EXIT',
  4 /,20X,45('**'),/,
  5 /* [ENTER 1, 2 OR 3]: '$)

150 CONTINUE
  ACCEPT * ,IANS
  IF(IANS.EQ.1) GOTO 200
  IF(IANS.EQ.2) GOTO 2000
  IF(IANS.EQ.3) GOTO 999
  TYPE ** ERROR -- TYPE 1 OR 2 OR 3 ONLY. NOW DO IT'
  GOTO 150

200 CONTINUE
  IEFN=6
  CALL IMPACQ(CATFIL)
  CALL SPAWN(READS(IMPACQ), IEFN, IESD, IDS)
  TYPE ** SPAWN CALLED ', 'IEFN=', IEFN, ' IESD=', IESD, 'IDS=', IDS
  CALL WAITSR(IEFN,IDS2)

2000 CONTINUE

C PHASE I -- KEY FIELD ENTRY
C
C
C TYPE 500,CLEANS
C TYPE 501

501 FORMAT(10X,60('**'),///,27X,' PHASE 1 -- KEY FIELD ENTRY',///,
  1 10X,60('**'),///)

TYPE 3998
3998 FORMAT (' ENTER RUN ID FOR THIS FILE: ',$)
  ACCEPT 3999,CATFIL(5:10)

3999 FORMAT(A6)
  TYPE 4000

4000 FORMAT('/ LENGTH (CM.) BETWEEN INNER ELECTRODES: ',$)
  ACCEPT * ,ELEN
  TYPE 4100

4100 FORMAT('/ HERMATOCRIT COUNT : ',$)
  ACCEPT * ,HEM
  TYPE 4200

4200 FORMAT('/ HEIGHT (CM) : ',$)
  ACCEPT * ,HEIGHT
  TYPE 4300

4300 FORMAT('/ WEIGHT (KG) : ',$)
  ACCEPT * ,WEIGHT

C
C SET DEFAULTS AND ZERO-currently unused fields
C

C

C

ORIGINAL PAGE IS
OF POOR QUALITY
RHO=53.2*EXP (.022*HEM)
BSA=HEIGHT**.725*WEIGHT**.425*.00714
C=RHO*ELEN**2
D TYPE ",' C=' ,C,'RHO= ',RHO
D CALL WAIT(5,2)
C
C PHASE 2 -- CALIBRATION VALUES
C
6000 CONTINUE
CALL ERRSET(59,.TRUE.,.FALSE.,.TRUE.,.FALSE.,MAX)
CALL IMPVAL (CALVAL)
C
C PHASE 3 -- CALIBRATION ACQUISITION
C
7010 CONTINUE
OPEN (UNIT=1,NAME='DW1:NUDRAW.TEL',TYPE='NEW',FORM='FORMATTED')
WRITE(1,2) CALVAL ,C,BSA,CATFIL.IMINS
2 FORMAT(6(G15.F),AI_,I10)
CLOSE (UNIT=1)
IEFN=6
CALL SPAWN (RAD50(IMPCAL),,,IEFN,,IESD,IDS)
D TYPE ",' SPAWN CALLED ',IEFN=',IEFN.' IESD=',IESD,'IDS=',IDS
C CALL WAITF(IEFN,IDS2)
D TYPE ",' IEFN SET NOW ',IDS2
C OPEN (UNIT=1,NAME='DW1:NUDRAW.TEL',TYPE='OLD',FORM='FORMATTED')
READ (1,2) SLOPE,ENTRCP,C,BSA,CATFIL,IMINS
CLOSE (UNIT=1)
CATFIL(11:14)=' .IMP'
D OPEN INTERNAL FILE THATSTORES RESULTS OF THE REDUCTION PROCESS
C IF IT EXISTS AND FIND OUT WHAT EPOCHS HAVE BEEN COMPLETED. OTHER
C WISE CREATE A NEW INTERNAL FILE TO BEGIN REDUCTION
C
C OPEN (UNIT=1,STATUS='OLD',NAME=CATFIL,
1 FORM='UNFORMATTED',ACCESS='DIRECT',RECL=8,ERR=50)
CLOSE(UNIT=1)
CALL INVEST(CATFIL) !DETERMINE WHAT EPOCHS HAVE BEEN COMPLETED
D TYPE ",'PRESS [RETURN] TO CONTINUE '
C ACCEPT 500 ,ANS
GOTO 55
50 CONTINUE
C
C OPEN (UNIT=1,STATUS='NEW',NAME=CATFIL,
1 FORM='UNFORMATTED',ACCESS='DIRECT',RECL=8)
CLOSE(UNIT=1)
MAXREC=380
IF(CATFIL(10:10).EQ.'0') MAXREC=120
IF(CATFIL(10:10).EQ.'4') MAXREC=220
CALL ZEREC (MAXREC,CATFIL) !ZERO CONTENTS OF NEW FILE FOR MAX RECS
55 CONTINUE
NOTE FOR THIS SPawning TO WORK TYPE INSTALL DRAWRM.TSK TO
INSTALL THE DRAWED.FTN PROGRAM

CONTINUE
CALL SPAW(9AD50(DRAW),,,IEFN,,IESD,,,,,,IDS)
CONTINUE
CALL WAITFR(IEFN,IDS2)
CONTINUE
CALL SPawn(CALLED ','IEFN=',IEFN,'IESD=',IESD,'IDS=',IDS
CONTINUE
TYPE *,' IEFN SET NOW ','IDS2
CONTINUE

CONTINUE
TYPE 500,ICLEAN

CONTINUE
TYPE 9109

CONTINUE
TYPE 9110

CONTINUE
IF(TEMP.EQ.'2'.OR .TEMP.EQ.'4') THEN
OPEN(UNIT=1, NAME='DWI:NUDRAW.TEL',STATUS='OLD')
CLOSE (UNIT=1,DISPOSE='DELETE')
END IF
CONTINUE
GO TO 9114

CONTINUE
DECODE (1,110,TEMP) ITEMP

CONTINUE
DECODE (9113,99990,80000,6000) ITEMP
CONTINUE
CALL SPAWN(RAD50(IMPSTR),,,IEFN,,IES),,IDS)
 TYPE *, ' SPAWN CALLED ', 'IEFN= ', 'IEFN,' IESD= ', 'IESD,' IDS= ', 'IDS
 CALL WAITFR(IEFN,IDS2)
 TYPE *, ' IEFN SET NOW ', IDS2
 CALL IMPSTR(CATFIL,IFLAG,GROUP,RUNNUM,DIRECT,OSN,AGE,SUSC,SEX,
 DATE,TESTIM,PREHR,TEMPRT,PRETEM,PRESYS,PREDIA,PRETST,PSTTST)
 TYPE 500,CLEANS
 OPEN(UNIT=1, NAME='DW1:NUDRAW.TEL',STATUS='OLD')
 CLOSE (UNIT=1,DISPOSE='DELETE')
 GOTO 125
 C10000 CONTINUE
 IFLAG=1 ! WRITE COMMENTARY FILE
 CALL COMENT (KCHAR)
 GOTO 9101
 99999 CONTINUE
 TYPE 500,CLEANS
 GOTO 125
 99999 CONTINUE
 END
 SUBROUTINE ZEREC(MAXREC,CATFIL)
 integer tcount
 CHARACTER*14 CATFIL
 OPEN (UNIT=I,STATUS='OLD',NAME=CATFIL,
 FORM='UNFORMATTED',ACCESS='DIRECT',RECL=8)
 do 100 tcount = I,MAXREC
 write(1,REC=TCOUNT,ERR=999)O.,O.,O.,O.,O.,O.,O.,O.
 continue
 100 CLOSE(UNIT=I, STATUS='SAVE')
 RETURN
 END
 SUBROUTINE INVEST(CATFIL)
 DIMENSION BUF(8,1),NONZER(400)
 CHARACTER*14 CATFIL
 CALL ERRSET(36,.TRUE.,.FALSE.,.TRUE.
 IREC=I
 INUM=0
 TYPE *, ' FROM THE RECORD, THE FOLLOWING EPOCHS ARE NONZERO :

 do 100 INUM = INUM+I
 IF(BUF(I,I)) 999,999,200
 continue
 INUM=INUM+I
 NONZER(INUM)=IREC
 100 CLOSE(UNIT=1, STATUS='SAVE')
CALL ERASE(106, IERR)
IF(IERR.EQ.1) THEN
TYPE *, (NONZERO(I), I=1, INUM)
RETURN
ELSE
IREC=IREC+1
GOTO 100
END IF
END

PROGRAM IMPCAL

C
(SLOPE, INTRCP, CATFIL)

PARAMETER (:FREQ=200)
REAL CALVAL(2, 2), INTRCP(2), SLOPE(2)

INTEGER CLEANS(2), CHANEL, HIGH(2, 2)
COMMON ISPACE(4*IFREQ), SPACE(30*IFREQ)
EQUIVALENCE (CALVAL, SPACE(1))
CHARACTER*1 CATFIL
DATA CLEANS/27, 99 /

== Phase 3 -- Calibration Acquisition ==

CFREQ=IFREQ
OPEN(UNIT=1, NAME='DWI:NUDRAW.TEL', TYPE='OLD', FORM='FORMATTED')
READ (1, 2) CALVAL, C, BSA, CATFIL, IMINS
CLOSE(UNIT=1)

TYPE 500, CLEANS
500 FORMAT(X, 4A1)
TYPE 700
700 FORMAT(10X, 60(*), /////, 22X, 'Phase 3 -- Calibration
  Acquisition', /////, 10X, 60(*), /////)
CATFIL(1:4) = 'DWI:
CATFIL(11:14) = '.DIG'

CALIBRATE BASELINE IMPEDANCE - CHANNEL 2

CALL CALZ (SLOPE(1), INTRCP(1), CALVAL, CATFIL)

CALIBRATE dZ/dT SIGNAL - CHANNEL 3

CALL CALDZT (SLOPE(2), INTRCP(2), CALVAL, CATFIL)

OPEN(UNIT=1, NAME='DWI:NUDRAW.TEL', TYPE='OLD', FORM='FORMATTED')
WRITE(1, 2) SLOPE, INTRCP, C, BSA, CATFIL, IMINS
CLOSE(UNIT=1)
END

SUBROUTINE DIRECT (TCOUNT, CATFIL, NCHAN)

STORAGE SEQUENCE AS FOLLOWS:

FIRST - RAW RESPIRATION

OF POOR QUALITY
PARAMETER (IFREQ=200)
integer buf(IFREQ), tcount
DIMENSION X(15*IFREQ),Y(15*IFREQ)
COMMON BUF,X,Y
CHARACTER*14 CATFIL

OPEN (UNIT=1, status='old', NAME=CATFIL,
     FORM='UNFORMATTED', access='direct', recl=4*IFREQ)

read(1,REC=TCOUNT,ERR=999)(BUF(ICCUNT ),ICOUNT=1,4*IFREQ)
continue

CLOSE(UNIT=1, STATUS='SAVE')
FORMAT(x,i3,4(2X,14,2x,i4))
FORMAT(IX,15)
RETURN
END

SUBROUTINE CALZ (SLOPE,INTRCP,CALVAL,"CATFIL"
--------------------
CALIBRATE BASELINE IMPEDANCE - CHANNEL 2
--------------------

REAL CALVAL(2, 2),CFREQ,TFREQ,EPOCH,INTRCP
INTEGER*4 IADSUM
INTEGER CHANNEL,ICOUNT,HIGH( 2),
   CLEANS(2),MAXPTS,ISTAT(2),IEFN,MDSYN,
   ICHAN(8),ICONV,IFORM,ITRIG,ISTAT2(2),IEFN2,NPTS(2)

CHARACTER*4,1WORD(2)
CHARACTER*14,CATFIL

PARAMETER (IFREQ=200)
COMMON ISPACE(4*IFREQ),SPACE(30*IFREQ)
LOGICAL*1 ANS

DATA IWORD /'HIGH','LOW '/'
DATA CLEANS/27,99/

C
---

**INITIALIZATION**
---

**PERFORM PRELIMINARY ACQUISITION**
---

CALL WAIT(2,2)

**TYPE 500**, CLEANS

**FORMAT(X,4A1)**

**TYPE 700**

**FORMAT(10X,60(' '),/,' CALIBRATION OF BASELINE IMPEDANCE',/,'/',10X,60(' '),/)*/

CALL WAIT(2,2)

**CHANNEL=2**
---

**CONTINUE**
---

**OPEN(UNIT=2, NAME='SY:DUMP.TST', FORM='FORMATTED', TYPE='NEW')**

**ISTREC=1**

**CONTINUE**

N=1 ! HIGH CALIBRATIONS
---

**PERFORM HIGH CALIBRATION**
---

**CALIB= .TRUE.**

**CONTINUE**
---

**INITIALIZE SLOPE, INTERCEPT**
---

**SLOPE = 0.0**

**INTRCP = 0.0**

**NSAMP=0**

**IADSUM=0**

**DO 7000 I=0,29**

**IREC = 30*(N-1)+ISTREC+I**

**CALL DIRECT(IREC,CATFIL)**

**CALL CALCULATE MEAN**
---
ORIGINAL PAGE IS OF POOR QUALITY

NPTCHN=IPFREQ !NUMBER OF SAMPLES PER CHANNEL PER RECORD=PER SECOND
DO 704 J=1,NPTCHN
   IADSUM=IADSUM+ISPACE ( (J-1)*4+CHANEL )
C   TYPE *,IADSUM,ISPACE ( (J-1)*4+CHANEL )
   NSAMP=NSAMP+1
C   WRITE(2,19999)(IAD(I+(J-1)*CHANEL+(K-1)*MAXPTS),NPTCHN,CHANEL,19999)
   FORMAT(9I8)
   CONTINUE
C   WRITE(2,*),IADSUM,NSAMP,ISPACE
   CONTINUE
   HIGH(N)=IADSUM/NSAMP
C----------------------------------
C ECHO MEAN AND VERIFY
C----------------------------------
C
    TYPE 715,IWORD(N)
    FORMAT(/'---------',5X,'----------------',5X,'---------')

    TYPE 71550,HIGH(N)
    FORMAT(5X,'A/D',5X,'A/D',5X,'A/D',5X,'A/D',5X,'A/D',5X,'A/D')

    TYPE 717,IWORD(N)
    FORMAT(/' DO YOU WANT TO RE-RUN ',A4,' CALS [Y/N]?',$)
    ACCEPT 617,ANS

    IF (ANS .EQ. 'Y'.OR.ANS.EQ.'y') THEN
       TYPE *,WHAT STARTING RECORD '
       ACCEPT *,ISTREC
       GOTO 7110
    END IF
    IF (N.EQ.2) GOTO 724
    N=2
    GOTO 7110
C----------------------------------
C CALCULATE SLOPE AND INTERCEPT
C----------------------------------

    IF (HIGH(1) .NE. HIGH(2)) GOTO 7135
    INTRCP = 0.0
    SLOPE = 0.0
    GOTO 718

    SLOPE = (CALVAL(2,1)-CALVAL(1,1))/FLOAT(HIGH(1)-HIGH(2))
    INTRCP = CALVAL(1,1) - SLOPE*FLOAT(HIGH(2))
    GOTO 713
C----------------------------------
C ECHO SLOPE,INTERCEPT
C----------------------------------
TYPE 720

FORMAT(/' CHANNEL ',5X,' SIGNAL ',5X,' ADLOW ',5X,
1' ADHI ',5X,
2 ' LOWCAL ',5X,' HICAL ',/,'X,6('-------',5X))

TYPE 81550 , HIGH(2)

81550 FORMAT( 4X,' I',7X,' IMPEDANCE ',4X,110,$)

TYPE 722,HIGH (1),CALVAL(1,1),CALVAL (2,1)

721 CONTINUE

722 FORMAT('+' 4X 110 4X,F10.4)

723 FORMAT(/' WOULD YOU LIKE TO REPEAT CALIBRATIONS [Y/N]?',$)

ACCEPT 617,ANS

IF (ANS .EQ. 'Y'.OR.ANS.EQ.'y') THEN

TYPE *, WHAT STARING RECORD '

ACCEPT *,ISTREC

GOTO 7019

END IF

CLOSE (UNIT=2)

RETURN

FUNCTION MAXIAD (IAD,MAXPTS,NPTS)

DIMENSION IAD(NPTS*MAXPTS)

MAXIAD= IAD(3)

DO 100 J=1,MAXPTS

IF(IAD((J-1)*NPTS+3 ).LT.MAXIAD) GOTO 100

MAXIAD= IAD((J-1)*NPTS+3)

100 CONTINUE

RETURN

END

FUNCTION MINIAD (IAD,MAXPTS,NPTS)

DIMENSION IAD(NPTS*MAXPTS)

MINIAD= IAD(3)

DO 100 J=1,MAXPTS

IF(IAD((J-1)*NPTS+3 ).GT.MINIAD) GOTO 100

MINIAD= IAD((J-1)*NPTS+3)

100 CONTINUE

RETURN

END

SUBROUTINE CALDZT (SLOPE,INTRCP,CALVAL,CATFIL)

CALIBRATE dZ/dT SIGNAL - CHANNEL 3

REAL CALVAL(2, 2),CFREQ,TFREQ,EPOCH,INTRCP

INTEGER*4 IADSUM,IA4X,IAD4MX,MAD4ELT

INTEGER CHANEL,ICOUNT,HIGH(2),IAD(400),
CLEANS(2),MAXPTS,ISTAT(2),IEFN,MDSYN,
ICHAN(8),ICONV,IFORM,ITRIG,ISTAT2(2),IEFN2,NPTS(2)

PARAMETER (IFREQ=200)

COMMON ISPACE(4*IFREQ),SPACE(30*IFREQ)

EQUIVALENCE (IAD,ISPACF)
CHARACTER*4, IWORD(2)
CHARACTER*14, CATFIL

---

INITIALIZE
---

DATA CLEANS/27,99/

PERFORM PRELIMINARY ACQUISITION

TYPE 500, CLEANS

FORMAT(X,2A1)

TYPE 700

FORMAT(10X,50(' '),/10X,22X, 'CALIBRATION

OF dz/dt SIGNAL',/10X,10X,60(' '),/)

CALL WAIT(5,2)

CHANNEL=3

CONTINUE

CONTINUE

CONTINUE

CONTINUE

CONTINUE

OPEN(UNIT=2, NAME='SY:DUMP.TST', FORM::'FORMATTED', TYPE='NEW')

N=1  HIGH CALIBRATIONS
SLOPE=0.0
INTRCP=0.0

CONTINUE

NCHAN=0
NSAMP=0
IADSUM=0

READ IN A-D VALUES

IREC = 31 + I

CALL DIRECT(IREC, CATFIL)

DTYPE * , (IAD((J-1)*4 +3),J=1,100)
**CALCULATE MEAN**

```fortran
IF(N.EQ.1) THEN
  FIND MAX VALUE IF N=1
  IADMAX=MAXIAD(IAD,IFREQ,4)
  TYPE *, 'IADMAX = ', IADMAX
  IADELT=IADMAX
  IADELT=IADELT*5
  IF(IADELT.LT.0) IADELT=-IADELT
ELSE
  FOR N=2 FIND MAX AND MIN
  IADMAX=MAXIAD(IAD,IFREQ,4)
  IADMIN=MINIAD(IAD,IFREQ,4)
  IZERO=IADMIN+((IADMAX-IADMIN)*2/5)
END IF
NPTCHN=IFREQ
DO 704 J=1,NPTCHN
IF(N.EQ.1) THEN
  USE IAD VALUE IF IAD >= MAX VALUE - 5%*(MAX VALUE)
  IAD4=IAD((J-1)*4+CHANEL)
  IAD4MX=IADMAX
  IF((100*IAD4).GE.(IAD4MX*100-IADELT)) THEN
    NCHAN=NCHAN+1
    IADSUM=IADSUM+IAD((J-1)*4+CHANEL)
  END IF
ELSE
  IF(IAD((J-1)*4+CHANEL).LE.IZERO+80) THEN
    NCHAN=NCHAN+1
    IADSUM=IADSUM+IAD((J-1)*4+CHANEL)
  END IF
END IF
WRITE(2,19999)IAD(J+(K-1)*MAXPTS),NPTCHN,CHANEL,
1 NPTS(K),NCHAN,I,J,K,IADMAX
19999 CONTINUE
704 CONTINUE
7000 CONTINUE
  N=2
GOTO 714
714 CONTINUE
```
ECHO MEANS AND VERIFY

WRITE(2,* ) IAD,HIGH,IADM,IA,IZERO
TYPE 715,1WORD(N)
FORMAT(\\\\\\\\\\\\\\\:\\\\\\\\:\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\ CHANNEL ',5X' SIGNAL ',5X,'A/D',A4,
1 ' /',------------------',5X,'------------------',5X,'------------------'
) TYPE 71551 , HIGH(N)

TYPE 71551 FORMAT( 5X,' 2',8X,'dZ/dT ',5X,I10)

TYPE 717
FORMAT(\'/' DO YOU WANT TO RE-RUN dZ/dT CALS [Y/N]? ',$)
ACCEPT 617,ANS
IF (ANS .EQ. 'Y'.OR.ANS.EQ.'y') GOTOC 7110
CLOSE(UNIT=2)

CALCULATE SLOPE AND INTERCEPT

IF (HIGH(1) .NE. HIGH(2)) GOTO 7135
INTRCP = 0.0
SLOPE = 0.0
GOTO 718
SLOPE = (CALVAL(2,2)-CALVAL(I,2))/FLOAT (HIGH(1)-HIGH(2))
INTRCP = CALVAL(1,2) - SLOPE*FLOAT(HIGH(2))
CONTINUE

ECHO SLOPE,INTERCEPT

TYPE 720
FORMAT(\'/' CHANNEL ',5X,' SIGNAL ',5X,'ADLOW ',5X,
1 ' ADHI ',5X,
2 ' LOWCAL ',5X,' HICAL ',5X,' 6(--------',5X)
) TYPE 81551 , HIGH(2)

FORMAT( 4X,' 2',7X,' dZ/dT ',4X.I10,$)
TYPE 722, HIGH(1),CALVAL(1,2),CALVAL(2,2)
2 TYPE *, ' SLOPE',' SLOPE,' INTERCEPT ', INTRCP
TYPE 723
FORMAT(\'/' WOULD YOU LIKE TO REPEAT CALIBRATIONS [Y/N]? ',$,)
ACCEPT 617,ANS
IF (ANS .EQ. 'Y'.OR.ANS.EQ.'y') THEN
TYPE *,' WHAT STARTING RECORD '
ACCEPT * ,ISTREC
GOTO 7019
END IF
PROGRAM IMPACQ

REAL TEMP, CALVAL(2, 2), MEAN(2), SD(2), SLOPE(2), INTRCP(2),
@ ADSUM, VARIAN(2), ZERO, CFREQ, TFREQ, DEV(2), EPOCH

INTEGER IOSB(2), BLINK(3), RESET(2),
@ ISTA(4), ICONT(2),
@ MIN, CHANNEL, ICLK, IADCSR,
@ ICOUNT, IMINS, IANS, LOW(10), HIGH(10), DIAG(15),
@ CLEANS(2), MAXRUN, MAXPTS, ISTAT(2), IEFN(2), MDSYN,
@ CHAN(8), ICONV, IFORM, ITRIG, ISTAT2, IEFN2, NPTS(2)
INTEGER IEFN1, IPAR(6), ISTAT3(2)

LOGICAL*1 EXP, SEX, ANS, TEMP, DMY,
@ DATFIL(14), STRNG(8),
@ FRMT(12)
CHARACTER*14 CATFIL
CHARACTER*1 IBUF, IPAWS, IABORT, IRSP
PARAMETER (IFREQ = 200) \! SETS PROGRAM SIZE WRT SAMPLE RATE
PARAMETER (MDSYN = 1)
PARAMETER (MODE = 0)
PARAMETER (ICONV = 0)
PARAMETER (IFORM = 0)
PARAMETER (ITRIG = 0)
PARAMETER (ITIME = 0)
PARAMETER (CHAN = 4)

INTEGER*4 IADSUM, AD(8), ISIZE
INTEGER*2 IAD(8*IFREQ), IAD2(4*IFREQ), IAD1(4*IFREQ)
EQUIVALENCE (IAD1, IAD), (IAD2, IAD(4*IFREQ+1)), (ISTAT, ISTA),
1 (ISTAT2, ISTAT(3)), (IEFN(1), IEFN1), (IEFN2, IEFN2)
COMMON IAD

C C C

C INITIALIZATION
C

DATA CLEANS/27, 99/
DATA BLINK/155, 53, 109/, RESET/155, 109/
DATA FRMT('(', ' ', ' ', '(', ' ', 'X', ' ', 'F', '9', ' ', '.', '4', ')', ')
DATA ICLK/"172540/
DATA IREAD/"001000/
DATA IABORT/ "101 /, IPAWS/"120/
500 FORMAT(X, 4A1)
510 FORMAT(X, 5A1)
617 FORMAT(A1)
618 FORMAT(A6)
7125 FORMAT(' AIN STATUS WAS:')
7077 FORMAT(' STAT1 (OCTAL) = ', O6, ' ISTAT2 (DECIMAL) = ', I6)
7135 FORMAT(' TRY AGAIN.')
10 FORMAT(I2)

RETURN
END
PHASE 5 -- DATA ACQUISITION

FREQ=IFREQ
MAXPTS=4*IFREQ
CALL GETADR(IPAR(1),IBUF)
IPAR(2)=1

9000 TYPE 500,CLEANS
TYPE 900
900 FORMAT(10X,60('**'),/////,22X,'DIGITIZING A DATA FILE ',
1 /////,10X,60('**'),/////

C REQUEST MAXIMUM RUN DURATION

---------

TYPE 90109
90109 FORMAT(///'PLEASE ENTER RUN ID FOR THIS DATA FILE: ',$)
ACCEPT 618,(CATFIL(5:10))
CATFIL(1:4)='MW1:
CATFIL(11:14)='.DIG'
TYPE 90110
90110 FORMAT(///'PLEASE ENTER RUN TIME (IN SECS.) FOR TAPE FILE: ',$)
ACCEPT *,MAXRUN
ISIZE=6.25*MAXRUN+1
TYPE 90111
90111 FORMAT(///'HOW FAST TO SAMPLE THE DATA ON TAPE ? ',
1 /10X,'ENTER 1 FOR REAL TIME RATE ',
2 /10X,'ENTER 2 FOR TWICE REAL TIME ',
3 /'ENTER 1 OR 2 NOW: ',$)
ACCEPT *,ISAMP

C NOTE: MAXPTS=FREQ*4
CFREQ=FREQ*ISAMP
CALL SETFRQ(CFREQ,TFREQ)
DO 190 I=1,4
ICHAN(I)=I
190 CONTINUE
IEFN1=8
IEFN2=10
ICOUNT=2*MAXPTS

C OPEN TEMPORARY DATA FILE

---------

9015 CONTINUE
C OPEN(UNIT=2,NAME='SY:TEMP.DAT',TYPE='NEW')
OPEN (UNIT=1,STATUS='NEW',NAME=CATFIL,INITIALIZE=-ISIZE,
1 FORM='UNFORMATTED',ACCESS='DIRECT',RECL=4*IFREQ)

MIN = 0
IMINS = 0
C  ---------- INITIALIZE SUM,ICOUNT
C  ----------

9011 CONTINUE
ICNT(1)=ICOUNT/2
ICONT(2)=ICOUNT/2
CALL AINIT (ISTAT,IEFN1)
IF(ISTAT(1).EQ."40000.OR.ISTAT(1).EQ."40040) GOTO 7000
TYPE *, ' ISTAT(1) = ', ISTAT(1)
PAUSE
7000 CONTINUE

TYPE *, *,
IF (IRSP .NE. IPAWS) GOTO 9022
PAUSE
TYPE 500,BLINK
TYPE 7100

7100 FORMAT(/'******************************************,
1          ' */ DATA COLLECTION RESUMED *',
2          '******************************************'/)
TYPE 500,RESET
GO TO 901

9022 TYPE 9012

9012 FORMAT (' HIT "S" TO START, "P" TO PAUSE, ',
          ' "A" TO ABORT:/' FOLLOWED BY A [RETURN]' )
ACCEPT 617, ANS
IF (ANS .EQ. 'A') GOTO 9100
IF (ANS .EQ. 'P') PAUSE
TYPE 500,CLEANS
TYPE 500,BLINK
TYPE 97100

97100 FORMAT(/'------------------------------,
1          ' */ DATA COLLECTION UNDERWAY *',
2          '------------------------------'/)

TYPE 500,RESET

901 CONTINUE
CALL QIO(IREAD,7,4,IOSB,IPAR,IDSW)
CALL AIN (ISTAT, IAD, ! START CLOCK
2 IAD ,
3 ICONT(1),
4 IEFN1,
5 MDSYN,
6 ICHAN,
7 ICONV,
8 IFORM,
9 ITRIG,ITIME,CFREQ)
CALL ABUF(ISTAT2,IAD2,ICONT(2),IEFN2,MDDE ,ITIME)

908 CONTINUE
IMINS = IMINS + 1
C TYPE *, ',IMINS = ', IMINS
K=1
N=1
IEND=ICOUNT
CONTINUE
ISTAT=ISTA(1+2*(K-1))
IF (ISTAT .EQ."40000. OR. ISTAT .EQ."40040"
1 GOTO 81300
C
IF (IISTAT.NE.0) TYPE *,K,IISTAT
TYPE *,K,IISTAT,IMINS
GOTO 80001
CONTINUE

STORE DIGITIZED DATA FOR EACH SECOND

CALL DIRECT(IMINS,K)
NCONT=ICONT(K)
CALL ABUF(ISTA(1+2*(K-1)),IAD(1+(K-1)*MAXPTS),NCONT ,IEFN(K),
1 MODE,ITIME)

ECHO MEANS AND STANDARD DEVIATIONS

IF END OF SESSION, END DATA ACQUISITION
MIN = IMINS
IF(MIN.EQ. 30) THEN
CALL ACSTAT(ISTAT,JSTAT, I,IEFN1,0)
CALL QIO("12,9")
CALL QIO("12,7")
PAUSE ' - ADVANCE TAPE TO START OF LOW CALS NOW; THEN ' CALL AINIT(ISTAT,IEFN1,0)
TYPE 500,BLINK
TYPE 7100
TYPE 500,RESET
GOTO 901
END IF
IF(MIN.EQ. 60) THEN
CALL ACSTAT(ISTAT,JSTAT, I,IEFN1,0)
CALL QIO("12,9")
CALL QIO("12,7")
PAUSE ' - ADVANCE TAPE TO START OF DATA NOW; THEN ' CALL AINIT(ISTAT,IEFN1,0)
TYPE 500,BLINK
TYPE 7100
TYPE 500,RESET
GOTO 901
END IF
IF (MIN .GE. MAXRUN) GO TO 9100

9100 CONTINUE
IF PAUSE OR ABORT, END DATA ACQUISITION

CALL READF(4,IEF)
IF(IEF .GT.0) THEN
  IRSP = IBUF
  IF (IRSP .EQ. IPAWS) THEN
    CALL ACSTAT(ISTAT,JSTAT,1,IEFN1,0)
    CALL QIO("12,9")
    CALL QIO("12,7")
    CALL AINIT (ISTAT,IEFN1,0)
    GO TO 7000
  ELSE
    IF (IRSP .EQ. IABORT) GO TO 9100
  END IF
ELSE
  SOMETHING ELSE TYPED IN; QUE I/O REQUEST AGAIN
  CALL QIO(IREAD,7, 4,,IOSB,IPAR,IDSW)
END IF
IK=K
IF(IK.EQ.1) K=2
IF(IK.EQ.2) K=1
IMINS=IMINS+I
GOTO 8000

CLEAN UP AND EXIT
CONTINUE
TYPE 500,RESET
CLOSE(UNIT=2)
CALL QIO("12,7")
TYPE 91000

91000 FORMAT (///////,10X,'SELECT ONE OF THE FOLLOWING [ENTER 1 OR 2]',
1/20X,'1 TO SAVE DIGITIZED FILE ' ,
2/20X,'2 TO DELETE DIGITIZED FILE ',$)
ACCEPT 617,ANS
IF(ANS.EQ.'1') THEN
  CLOSE(UNIT=1)
ELSE IF(ANS.EQ.'2') THEN
  CLOSE(UNIT=1, STATUS='DELETE')
ELSE
  GO TO 9100
END IF
CALL ACSTAT(ISTAT3,JSTAT,1, 5,0)
CALL QIO("12,7")
CALL QIO("12,9")
END
SUBROUTINE DIRECT (TCOUNT,K)

STORAGE SEQUENCE AS FOLLOWS:
FIRST - RAW RESPIRATION
SECOND - DELTA Z
THIRD - DZ/DT
FOURTH - ECG
PARAMETER (IFREQ=200)
integer buf(8*IFREQ), tcount

COMMON BUF

do 100 tcount = 1, 2
WRITE(1,REC=TCOUNT,ERR=999)(BUF((K-1)*4+4*IFREQ+ICOUNT),
ICOUNT=1,4*IFREQ)
   TYPE *, 'OK ON READ'
   do 200 i = 1, 200
      write(5,10)i, (BUF(ICOUNT,1),ICOUNT=(i-1)*4+1,i*4)
   C200
   continue
   TYPE *, 'OK ON WRITE'
C100 continue
10 FORMAT(x,i3,4(2x,i4))
20 FORMAT(1x,i5)

RETURN
END

SUBROUTINE SETFRQ(CFREQ,TFREQ)

SUBROUTINE GETFRQ(CFREQ,TFREQ)

LANGUAGE: FORTRAN-77

FUNCTION:
This SUBROUTINE will prompt the user for the desired clock frequency to
be used in acquiring analog data.

OUTPUTS:
CFREQ = REAL*4 variable containing the user's desired frequency in hz.
TFREQ = REAL*4 variable containing the user's actual frequency in hz.

SUBROUTINES REFERENCED:
SUBROUTINE CLKFRQ.
REAL*4 CFREQ,TFREQ  
!Declare desired, actual frequencies.

CONTINUE
CALL CKFREQ(CFREQ,TFREQ)
IF(TFREQ.NE.-999.0) GO TO 2  
!Skip ahead if CFREQ is ok.

TYPE 9015
9015 FORMAT(/,1X,'Bad frequency, please try again',/)  
GO TO 1  
!Prompt for frequency again.

CONTINUE
RETURN  
!Return to caller.
END

PROGRAM IMPSTR

!AN IMPEDANCE CARDIOGRAPH PROGRAM USING DIGITIZED DATA FILES

PARAMETER (!FREQ=200)
DIMENSION SLOPE(2),ENTRCP(2)
COMMON ISPACE(4*IFREQ),SPACE(30*IFREQ)
REAL DRAW,IMPACQ,IMPCAL

REAL TEMPRT,
  ZERO,CFREQ,TFREQ,EPOCH

INTEGER PREDIA,PREHR,PRETEM,PRESYS,PSTSYS,NUMDIA,
  PSTHR,PSTTEM,RUNNUM,IDAY,NSN,AGE,SUSC,
  TESTIM,MIN,PSTSTT,PAR,TOTMIN,PSTST,
  IMINS,IANS, HIGH(2,2),DIAG(15),
  CLEANS(4),ICLEAN(2)

LOGICAL*1 SEX,ANS,TEMP,DMY,IRSP,IABORT,IPAWS,TYMEAN,STDDEV,
  GROUP(8),DATE(7),DIRECT(2),OSN(2),STRNG(8),
  FRMT(12)
CHARACTER*1 CMT(80),RUN(4),RUNTYP,DATFIL(14),EXP
CHARACTER*14 CATFIL
EQUIVALENCE (DATFIL,CATFIL)
DATA CLEANS/27,91,50,74/

OPEN (UNIT=1,NAME='NUDRAW.TEL',FORM='FORMATTED',TYPE='OLD')
READ(1,2) SLOPE,ENTRCP,C,CATFIL,MAXREC
CLOSE(UNIT=1)
CONTINUE

TYPE 5020
5020 FORMAT(/' EXPERIMENT DESIGNATION (A,B,...): ',$)
CALL ACCEPT 5021,EXP
5021 FORMAT (A)
TYPE 5030
5030 FORMAT(/' GROUP TYPE (maximum of 8 characters): ',$)
CALL ACCEPT 5031,(GROUP(I),I=1,8)
FOR _AT(8A_)
TYPE 5040
FORMAT(/' RUN TYPE (maximum of 4 characters): ',$)
ACCEPT 5041,(RUN(I),I=1,4)
TYPE 5050
FORMAT(/' RUN NUMBER: ',$)
ACCEPT *,RUNNUM
TYPE 5060
FORMAT(/' DIRECTION (cc,cw,nd): ',$)
ACCEPT 5061,(DIRECT(I),I=1,2)
TYPE 5070
FORMAT(/' DAY NUMBER: ',$)
ACCEPT *,IDAY
C
TYPE 5080
FORMAT(/' NEW SUBJECT NUMBER: ',$)
ACCEPT *,NSN
TYPE 5090
FORMAT(/' SUBJECT'S INITIALS (first and last): ',$)
ACCEPT 5091,(OSN(I),I=1,2)
TYPE 5100
FORMAT(/' AGE: ',$)
ACCEPT *,AGE
TYPE 5110
FORMAT(/' SUSCEPTIBILITY (1,2,3): ',$)
ACCEPT *,SUSC
TYPE 5120
FORMAT(/' SEX (m/f): ',$)
ACCEPT 5121,SEX
TYPE 5130
FORMAT(/' TEST DATE (mmmddyy): ',$)
ACCEPT 5131,(DATE(I),I=1,7)
TYPE 5140
FORMAT(/' TEST TIME (military in hrs and mins, e.g. for 1:30 p.m. enter 1330): ',$)
ACCEPT *,TESTIM
TYPE 5150
FORMAT(/' PRE HEART RATE (in beats/min): ',$)
ACCEPT *,PREHR
TYPE 5160
FORMAT(/' PRE TEMPERATURE: ',$)
ACCEPT *,TEMPRT
PRETEM = IIFIX(TEMPRT*10.0)
TYPE 5170
FORMAT(/' PRE B.P. (sys,dias ,e.g. 120,80): ',$)
ACCEPT *,PRESYS,PREDIAS
TYPE 5180
FORMAT(/' PRE/POST TEST BASELINE (in minutes): ',$)
ACCEPT *,PRETST
PSTTST = PRETST
AN IMPEDANCE CARDIOGRAPH PROGRAM USING DIGITIZED DATA FILES

```
CALL STORE (CATFIL,IFLAG,GROUP,RUNNUM,DIRECT,OSN,AGE,SUSC,SEX,  
  DATE,TESTIM,PREHR,TEMPRT,PRETEM,PRESYS,PREDIA,PRETST,PSTTST)

TYPE 500,CLEANS

COMMON DATCAR(4,2)
REAL*8 DRAW

AN IMPEDANCE CARDIOGRAPH PROGRAM USING DIGITIZED DATA FILES

DIMENSION SLOPE(2),ENTRCP(2)
REAL TEMPRT,MEAN(4),SD(4),  
  ZERO,CFREQ,TFREQ,EPOCH

INTEGER PREDIA,PREHR,PRETEM,PRESYS,PSYSTIA,PSYSYS,NUMDIA,  
  PSTHR,PSSTEM,RUNNUM,IDAY,NSN,AGE,SUSC,  
  TESTIM,MIN,PSTTST,PART,CHANEL,TOTMIN,PRETST,  
  IMINS,IANS, HIGH(2,2),DIAG(15),  
  CLEANS(4),ICLEAN(2)

LOGICAL*1 SEX,ANS,TEMP,DMY,IRSP,IABORT,IPAWS,TYMEAN,STDDEV,  
  GROUP(8),DATE(7),DIRECT(2),OSN(2),STRNG(8),  
  FRMT(12)
CHARACTER*1 CMT(80),RUNTYP,DATFIL(14),EXP  
CHARACTER*4 RUN
CHARACTER*14 CATFIL,IMPFIL
EQUIVALENCE (DATCAR,MEAN),(DATCAR(1,2),SD)

DATA DRAW/'DRAWRM'/,ICLEAN/27,99/
DATA CLEANS/27,91,50,74/

IFLAG=0

OPEN (UNIT=1,STATUS='OLD',NAME='DW1:NUDRAW.TEL',FORM='FORMATTED')
READ(1,2) SLOPE,ENTRCP,C,BSA,IMPFIL,MAXREC
CLOSE(UNIT=1,DISPOSE='SAVE')
```

AN IMPEDANCE CARDIOGRAPH PROGRAM USING DIGITIZED DATA FILES
ASSEMBLE DATAFILE NAME

CONTINUE
CALL ERRSET(63,.TRUE.,.FALSE.,.FALSE.,.FALSE.,.FALSE.,MAX)
CALL ERRSET(36,.TRUE.,.FALSE.,.TRUE.,.FALSE.,.FALSE.,.FALSE.,MAX)
CALL ERRSET(29,.TRUE.,.FALSE.,.TRUE.,.FALSE.,.FALSE.,.FALSE.,MAX)
EXP=IMPFIL(5:5)
DECODE(2,9112,IMPFIL(6:7)) NSN
DECODE(2,9112,IMPFIL(8:9)) IDAY
IF (IMPFIL(10:10) .EQ. '0') RUN(1:4)='BSLN'
IF (IMPFIL(10:10) .EQ. '1') RUN(1:4)='CSSI'
IF (IMPFIL(10:10) .EQ. '2') RUN(1:4)='TRAN'
IF (IMPFIL(10:10) .EQ. '3') RUN(1:4)='AMBL'
IF (IMPFIL(10:10) .EQ. '4') RUN(1:4)='TASK'
IF (IMPFIL(10:10) .EQ. '5') RUN(1:4)='VARD'
IF (IMPFIL(10:10) .EQ. '6') RUN(1:4)='DRUM'
IF (IMPFIL(10:10) .EQ. '7') RUN(1:4)='ZERO'

CATFIL(1:1) = 'D'
CATFIL(2:2) = 'Z'
CATFIL(3:3) = 'I'
CATFIL(4:4) = ':'
CATFIL(5:10)=IMPFIL(5:10)

FORMAT(A2)
FORMAT(I2)
FORMAT(I5)

OPEN (UNIT=1, TYPE='OLD', NAME=CATFIL, FORM='FORMATTED', ERR=45)
GOTO 100

CONTINUE
CALL ERRREN (IRENUM)
IF(IRENUM.EQ.29) THEN
OPEN (UNIT=1, TYPE='NEW', NAME=CATFIL, FORM='FORMATTED',
: ERR=40
ELSE
GOTO 40
END IF
CONTINUE

STORE KEY FIELD INFO TO DATA FILE

D
TYPE *, 'EPOCHS, RUN DURATION (MINUTES)= ', IMINS, MIN
WRITE(1,9120) EXP, (GROUP(I),I=1,8), (RUN ),
1 RUNNUM,(DIRECT(I),I=1,2), IDAY, NSN, (OSN(I),I=1,2),
2 AGE, SUSC, SEX, (DATE(I),I=1,7), TESTIM, PREHR, PRETEM, PRESYS, PREDIA,
3 PSTD, PSTTEM, PSTSYS, PSTDIA, MIN, PREST,
4 MIN-PRETEST-PSTSTT, PSTSTT, NUMDIA, 1, 8, DIAG(1)
9120 FORMAT( A1,X,8A1,X, A4,X, I2,X, 2A1,2(X,I2),X,2A1,X, I2,X, I1,
2 X,I3,X, I4,5(X,I3), 5(X,I2))
DO 9130 I = 2,14
   WRITE(1,9140) DIAG(I)
9130 CONTINUE
9140 FORMAT(109X,I2)
WRITE(1,9150) DIAG(15), MAXREC
9150 FORMAT(109X,I2, I3)
FRMT(2) = '0'
FRMT(3) = '6'
ZERO = 0.0
IERR=0
9650 CONTINUE
IREC=IREC+1
IF(IREC.GT.MAXREC) GOTO 9165
CALL DRECT (IREC,IERR,IMPFIL)
IF(IERR.EQ.1) GOTO 9160
WRITE(1,9550) MEAN, 0., 0., 0., 0., SD, 0. , 0. , 0. , 0.
WRITE(1,9550) (DATCAR(I,1),I=1,4), 0. , 0. , 0. , 0.
9550 FORMAT(8(X,F9.4))
GOTO 9650
9160 CONTINUE
WRITE(1,9550) 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0. , 0.
GOTO 9650
9165 CONTINUE
CLOSE(UNIT=1)
CATFIL(12:12)='C'
DATFIL(13)='A'

40
C DATFIL(14)='L'
C OPEN (UNIT=1, TYPE='NEW', NAME=CATFIL, FORM='FORMATTED',
C  1 ERR=40)
C DO 9700 I=1,3
C WRITE(1,9500) HIGH(I,2),HIGH(I,1),CALVAL(1,I),CALVAL(2,I)
C9500  FORMAT( 2(X, 110),2(X,F10.4))
C9700  CONTINUE
C CLOSE(UNIT=1)
C IF(IFLAG.EQ.0) GOTO 9875
C CATFIL(13:13)='O'
C CATFIL(14:14)='M'
C OPEN (UNIT=1, TYPE='NEW', NAME=CATFIL, FORM='FORMATTED',
C  1 ERR=40)
C OPEN (UNIT=2, TYPE='OLD', NAME='SY:TEMP.COM', FORM='FORMATTED',
C  1 READDONLY)
C CONTINUE
C J=0
C KCNT=90
C9750 CONTINUE
C J=J-KCNT
C IF(J.GT.KCHAR) KCNT=KCHAR-J+KCNT
C READ(2,9800) (CMT(I),I=1,KCNT)
C WRITE(1,9801) (CMT(I),I=1,KCNT)
C9800  FORMAT(8OA1)
C9801  FORMAT(X,8OA1)
C IF(J.GE.KCHAR) GOTO 9850
C GOTO 9750
C9850 CONTINUE
C CLOSE(UNIT=1)
C CLOSE(UNIT=2)
C9875 CONTINUE
C IF(N.EQ.2) GOTO 99990
C N=2
C CATFIL(3:3)='2'
C GOTO 9510
C99990 CONTINUE
C TYPE *, ' ARE YOU FINISHED WITH THE DIGITIZED DATA ?[Y/N]'
C ACCEPT 10000,ANS
C10000 FORMAT(A1)
C IF(ANS.EQ.'Y') THEN
C TYPE *, ' THEN YOU WANT TO DELETE THE DIGITIZED FILE ?[Y/N]'
C ACCEPT 10000,ANS
C IF(ANS.EQ.'Y') THEN
C OPEN (UNIT=2,NAME=IMPFIL ,TYPE='OLD')
C CLOSE (UNIT=2,DISPOSE='DELETE')
C IMPFIL(11:14)='.DIG'
C OPEN (UNIT=2,NAME=IMPFIL ,TYPE='OLD')
C CLOSE (UNIT=2,DISPOSE='DELETE')
C END IF
C END IF
C IF(IFLAG.EQ.1) THEN
C OPEN (UNIT=2,NAME='SY:TEMP.COM',TYPE='OLD')
C CLOSE (UNIT=2,DISPOSE='DELETE')
C END IF
C
OPEN (UNIT=1, STATUS='OLD', NAME='DRAW.TEL', FORM='FORMATTED')
CLOSE(UNIT=1, DISPOSE='DELETE')
TYPE 91100
91100 FORMAT (' SUCCESSFUL COMPLETION ')
99999 CONTINUE
RETURN
END

SUBROUTINE DRECT (T2COUNT, I9ERR, IMPFIL)

STORAGE SEQUENCE AS FOLLOWS:

FIRST - RAW RESPIRATION
SECOND - DELTA Z
THIRD - DZ/DT
FOURTH - ECG

TKB FOR TASK BUILD

TKB>FILENAME=FILENAME
TKB://
TKB>MAXBUF=1600
TKB://

PARAMETER (IFREQ=200)

integer buf(4*IFREQ,1)
INTEGER tcount
DIMENSION X(15*IFREQ), Y(15*IFREQ)
DIMENSION DATCAR(4,2)
COMMON DATCAR
EQUIVALENCE (DATCAR,BUF)
CHARACTER*14 IMPFIL

OPEN (UNIT=2, STATUS='old', NAME=IMPFIL
       FORM='UNFORMATTED', access='direct', recl=8 )

do 100 tcount = 1,2
read(2,REC=T2COUNT, ERR=990)((DATCAR(I,J),I=I,4),J=1,2)
   TYPE *,'OK ON READ'
   do 200 i = 1, 200
      write(5,10)i,(BUF(ICOUNT,I),ICOUNT=(i-I)*4+I,i*4)
   continue
   TYPE *,'OK ON WRITE'
   continue
GOTO 999
100 continue
GO999
990 CONTINUE
I9ERR=1
999 CLOSE(UNIT=2, STATUS='SAVE')
10 FORMAT(x,i3,4(2x,i4))
20 FORMAT(1X,I5)

RETURN
END

PROGRAM DRAWED
PARAMETER (IFREQ=200) ! SET SAMPLING SIZE FOR DIGITIZED DATA
DIMENSION ISTAT(2), IRES(75), OUTPUT(10,70),
1 SLOPE(2), ENTRCP(2)
CHARACTER*1 ANS
CHARACTER*1 ICHRPK(2), IBUF, ISTOP, IREPET
CHARACTER*4 ICHRT, ETIME
VIRTUAL IDATA(9000)
DIMENSION Y(1450), X(1000), NCHAN(3), SMIN(3), SMAX(3), DATCAR(5,50),
1 DATSD(4), IPAR(6)
INTEGER*2 IRESP(IFREQ)
CHARACTER*14 CATFIL
INTEGER LPTS(3), IPEAKS(50), IOSB(2), CLEANS(2)
BYTE 1233.173,
COMMON /PLOT/X,Y
EQUIVALENT (DATCAR,Y(1201)), (Y(501), OUTPUT)
DATA NCHAN/3,4,1/ , SMIN/350. ,150. ,0. / , SMAX/550. ,350. ,150. /
DATA LPTS/3,3,10/ , NSMFQ/100/, NREC/15/, IREAD/"001000/
DATA A/.01/, B/0./, C/O./, IBRAC/"/ , ICHAN/3/
DATA IGATE/0/, IRAC/0/, I233/"233/,"173/"73/,"162/"162/
DATA IHOME/155,63,54,108/, CLEANS/27,99/
CALL GETIDR(IPAR(1), IBUF) ! GET START ADDR OF IBUF & STOR
IPAR(2) ! SET TO BYTE SIZE OF IBUF
TYPE 4000, CLEANS
4000 FORMAT(X, 4 A1)
TYPE 4500
4500 FORMAT (10X, 60(\*'), ///, 22X, ' DATA REDUCTION & EDITING ',
1 ///, 10X, 60(\*'), ///)
CONTINUE
CALL QIO(IREAD,7,4,,IOSB,IPAR,IDSW)
CONTINUE
XMIN=0.
XMAX=IFREQ*NREC
CALL CGL(96) !INITIALIZE GRAPHICS
CALL CGL(92) !NEW FRAME
CALL CGL(103,'DW1:FILE2.GID',13) !INITIALIZE PLOT FILE
CALL CGL(105,'DW1:FILE2.GID',13) !SELECT PLOT FILE
CALL CGL(80,XMIN,XMAX,0.  ,600. ) !SET WINDOW TO DEFAULT VALUES
CALL CGL(86,0) !SET ORIGIN
OPEN (UNIT=1, status='old', NAME=CATFIL, !OPEN DIGITIZED DATA FILE
      FORM='UNFORMATTED', access='direct', reci='4*IFREQ')
DO 125 I=1,NREC
  IREC=IREC+I-1
  read(I,REC=IREC,ERR=999)((IDATA(J+(I-I)*IFREQ),ID,ID,ID),
    1 J=1,IFREQ)
125  CONTINUE
CLOSE(UNIT=1, STATUS='SAVE')
M=0
NPTS=LPTS(3)
DO 135 L=1,IFREQ*NREC,NPTS
  M=M+1
  X(M)=L
  Y(M)=IDATA(L)
135  CONTINUE
M=0 !SET COUNTER OF RESPIRATION VALUES
IFRQRC=IFREQ*NREC !SET AREA SIZE FOR DIG (IDATA) BLOCKS
DO 150 I=1,NREC
  IREC=KREC+I-1
150  CONTINUE
STORAGE SEQUENCE OF DIGITIZED DATA AS FOLLOWS:
FIRST - RAW RESPIRATION - STORED IN RESP FROM 1 TO IFREQ
SECOND - DELTA Z - STORED IN IDATA FROM 1 TO IFRQRC
THIRD - DZ/DT - STORED IN IDATA FROM 1+IFRQRC TO 2*IFRQRC
FOURTH - ECG - STORED IN IDATA FROM 1+2*IFRQRC TO 3*IFRQRC
read(1,REC=IREC, ERR=999)((IRES(J),
1 IDATA(J+(I-I)*IFREQ), IDATA(IFRQRC+J+(I-I)*IFREQ)),
2 IDATA(2*IFRQRC+ J+(I-I)*IFREQ)),J=1,IFREQ)
PRINT *,(IECG(J+(I-I)*IFREQ ),J=1,IFREQ),J
STORE A SUBSET( EVERY LPTS(3) VALUE) OF RESP IN Y FOR PLOTTING
DO 175 J=1,IFREQ,LPTS(3)
  M=M+1
175  CONTINUE

ORIGINAL REPLAY IS OF POOR QUALITY.
I(M)=IRESPP(C)
173 CONTINUE
150 CONTINUE
999 CLOSE(UNIT=1, STATUS='SAVE') !CLOSE DIG. DATA FILE
C
C PLOT FIRST RESPIRATION(K=3) THEN ECG(K=2) AND LAST DZ/DT(K=1)
C
DO 3000 K=3,1,-1
YSMIN=SMIN(K) !PARTITION SCREEN ARE., MIN FOR K PLOT
YSMAX=SMAX(K) !PARTITION SCREEN ARE., MAX FOR K PLOT
NPTS=LPTS(K)

SET PLOT ARRAY PAIRS (X,Y)

IF(K.EQ.3) THEN
DO 275 L=1,M !SETUP X VALUES FOR RESPIRATION PLOTTING
X(L)=1+(L-1)*NPTS
ELSE
Y=0
DO 300 L=1,IFREQ*NREC,NPTS !SETUP (X,Y) PAIRS FOR ECG AND DZ/DT PLOT:
M=M+1
X(M)=L
Y(M)=IDATA(IFRQRC*K+L)
CONTINUE
END IF
CALL INDEX(XMIN,XMAX,YMIN,YMAX,M ) !DETERMINE PLOT LIMITS
IF(K.EQ.1) THEN
IF(IDZDTI-IDZDTO.LT.(YMAX-YMIN)/3) THEN !REDONE LIMITS IF NEEDED
YMAX=IDZDTI+IDZDTI-IDZDTO
YMIN=IDZDTO-(IDZDTI-IDZDTO)
END IF
IF(IDZDT1.GT.YMAX) YMAX=IDZDT1
DZDTO=(IDZDTO-(IDZDT1-IDZDTO)
DZDT1=(IDZDT1-YMIN)/(YMAX-YMIN)*(YSMAX-YSMIN)+YSMIN !SCALE DZDTO
DZDTI=(IDZDT1-YMIN)/(YMAX-YMIN)*(YSMAX-YSMIN)+YSMIN !SCALE DZDT1
C
PLOT DZ/DT CALIBRATION VALUES FOR OBSERVATION

CALL CGL( 1,X(1), DZDTO) !MOVE "PEN" TO POSITION
CALL CGL( 4,XMAX, DZDTO) !DRAW TO POSITION
CALL CGL( 1,X(1), DZDT1) !MOVE "PEN" TO POSITION
CALL CGL( 4,XMAX, DZDT1) !DRAW TO POSITION
END IF
DO 322 L=1,10
TYPE *,X(L),Y(L),YMIN,YMAX,L
Y(L)=(Y(L)-YMIN)/(YMAX-YMIN)*(YSMAX-YSMIN)+YSMIN
CONTINUE
DO 325 L=1,M !SCALE Y
PRINT *,X(L),Y(L),YMIN,YMAX,L
Y(L)=(Y(L)-YMIN)/(YMAX-YMIN)*(YSMAX-YSMIN)+YSMIN
CONTINUE
FACT=5./32767.
CALL CGL(1,X,Y,1)) MOVE PEN TO POSITION
C XOLD=X(600)=1.7260.
C NY=NREC*IFREQ
NYPLT=IFREQ*60
CALL CGL(6,X,Y,NYPLT) !PLOT NYPLT X,Y PAIRS
IF (I.EQ.1) THEN !PLOT EPOCH NUMBER AND START TIME OF EPOCH
IEPOCH=IMOD(KREC-61,60)+(KREC-61)/60*100-ISTREC
ENCOD(4,32500,ETIME)IEPOCH
ENCOD(4,32500,EPOCH)IDKREC
3250 FORMAT(14)
CALL CGL(1,X,MIN,SIN(1))
CALL CGL(16,EPoch,4)
CALL CGL(1,X,MIN-240,SIN(1))
CALL CGL(16,ETIME,4)
END IF
500 CONTINUE
3000 CONTINUE END OF PLOTTING
C C NEXT CALL PEAK DETECTION
C C OPEN (UNIT=2,-status=NEW,NAME='DWT:IMPTEMP.TST',
D 1 FORM='FORMATTED',ACCESS='SEQUENTIAL')
CALL VOL(IFRQRC,NPEAKS,IDATA,IDADMIN,IGATE,IM,IDZDTO,IDZDFI)
C IDZDT - INPUT OF DZ/DT DATA
C IFRQRC - NUMBER OF DZ/DT DATA POINTS
C OUTPUT(3,I) - ARRAY CONTAINING TIME VALUES FOR DETECTED PEAKS
C NPEAKS - NUMBER OF PEAKS DETECTED
IF(NPEAKS.GT.0) THEN
CALL CGL(12,4,0,0) !SET LINE STYLE TO DOTTED
C C PLOT PEAK INDICATORS
C DO 680 I=1,NPEAKS
YLOW=SMIN(3)+40
XL=(OUTPUT(3,I)-1.)
CALL CGL(1,XL,SMAX(1)) !POSITION TO X=XL,Y=SMAX(1)
CALL CGL(4,XL,YLOW) !DRAW LINE TO X=XL,Y=YLOW
ENCOD(2,610,ICHRPK),I
610 FORMAT(12)
XLOW=XL-10.
DO 620 J=1,2
CALL CGL(1,XLOW,YLOW)
CALL CGL(16,ICHRPK(J),1) !WRITE PEAK NUMBER J
YLOW=YLOW-20.
620 CONTINUE
680 CONTINUE
CALL CGL(12,1,0,0) !RESET LINE STYIE TO SOLID
C C USE A SUBROUTINE HERE TO CALCULATE THE TIME WHEN
dZ/dT=0 BEFORE EACH PEAK ALSO CALCULATE STROKE VOLUME
C CALL TFACT(NPEAKS,IFREQ,C,NY,K,TYMEAN,
1 SLOPE,ENTRCP,BSA,IDATA,KREC,IDZDTO,IDADMIN,CATFIL,IDKREC)
C ELSE
SET MEAN AND SD TO ZERO
END IF

IF(_COUNT.GT.I) THEN
CALL CGL(12,4,O,O)
DO 700 I=I,_COUNT-I
XL=600.*I/200.
CALL CGL(1,XL,YMAX )
CALL CGL(4,XL ,YMIN )
CONTINUE
CLOSE(UNIT=2)

SET UP ONE LINE DIALOG AREA NOW

L20=49
L24=49
WRITE(5,40000) I233,L20,I73,L24,I162
FORMAT(IX,5AI)
DO 725 I=1,50 !ZERO ALL IPEAKS - ARRAY OF USER SELECTED PEAKS
IPEAKS(I)=0
CONTINUE

QUIRY EVENT FLAG _ TO SEE IF USER REQUEST MADE

CALL READEF(4,IEF)
IF(IEF.GT.O) THEN ! REQUEST MADE VIA KEYBOARD
END IF
CALL QIO("12,7) ! CANCEL QUE I/O REQUEST

DIALOG SECTION

TYPE *, ' SKIP THIS EPOCH ? [ENTER Y,N]'
ACCEPT 10000,ANS
IF(ANS.EQ.'Y') THEN
KREC=KREC+NREC !INCREMENT RECORD INDEX
MAXREC=MAXREC-1 !REDUCE MAX RECORD SIZE BY ONE
CALL CGL(106,'DWI:FILE2.GID',13) !DELETE SELECT PLOT FILE
CALL CGL(104,'DWI:FILE2.GID',13) !TERMINATE PLOT FILE
CLOSE (UNIT=1,DISPOSE='DELETE')
IF (KREC.GE.INDREC) GOTO 99999
GOTO 105
END IF

TYPE *, ' REDO ANALYSIS ? [ENTER Y/N]'
ACCEPT 10000,ANS
IF(ANS.EQ.'Y') THEN
TYPE *, ' ENTER GATE FACTOR, MINIMUM DIFFERENCE FACTOR NOW'
ACCEPT *,IGATE,IM
CALL CGL(106,'DWI:FILE2.GID',13)
CALL CGL(104,'DWI:FILE2.GID',13)
CLOSE (UNIT=1,DISPOSE='DELETE')
GOTO 10
END IF

TYPE *, 'OK TO CALCULATE HEATHER INDEX ? [Y/N]'
ACCEPT 10000,ANS

47
IF(ANS.EQ.'Y') THEN
    CALL HICALC(NPEAKS,IPREQ,IDATA
                   ,KREC,NREC,CATFIL)
ELSE
    DO 7250 I=1,NPEAKS
         DATCAR(4,I)=0.0
7250 CONTINUE
END IF
CALL CGL(106,'DW1:FILE2.GID',13)
CALL CGL(104,'DW1:FILE2.GID',13)
CALL CGL(93,INAM,ICODE) !REPORT ERROR
IF(ICODE.NE.0) THEN
    TYPE *,'INAME=',INAM,'ICODE=',ICODE
    CALL CGL(91) !TERMINATE GRAPHICS
END IF
7025 CONTINUE
CALL ERRSNS
C
PRINT *,NPEAKS=NPEAKS
C
PRINT *,(IPEAKS(I),I=1,NPEAKS)
TYPE *,LIST SELECTED PEAKS NOW (SEPERATE W/ COMMA)'
READ(8,7725) (IPEAKS(I),I=1,NPEAKS)
7725 FORMAT(<NPEAKS>13)
730 CONTINUE
740 CONTINUE
C
PRINT *,NPEAKS=NPEAKS
C
PRINT *,(IPEAKS(I),I=1,NPEAKS)
KPEAKS=0
DO 750 I=1,NPEAKS !DETERMINE NUMBER OF PEAKS SELECTED
    IF(IPEAKS(I).EQ.0) GO TO 775
    KPEAKS=KPEAKS+1
750 CONTINUE
775 CONTINUE
C
TYPE *,KPEAKS=KPEAKS,(IPEAKS(I),I=1,KPEAKS)
IF(KPEAKS.GT.0) THEN
    PRINT *,IDKREC,(IPEAKS(I),(DATCAR(K,IPEAKS(I)),K=I,4),I=I,KPEAKS)
    CALL STRVOL(KPEAKS,DATSD,IPEAKS)
ELSE !ZERO STROKE VALUE,CARDIAC OUTPUT,CARDIAC INDEX,HEATHER INDEX
    DO 810 I=1,4
         DATCAR(I,1)=0.0 !MEANS
         DATSD(I)=0.0 !STANDARD DEVIATIONS
810 CONTINUE
END IF
CALL ERRSNS(IERR)
IF(IERR.NE.0) GOTO 7025
820 CONTINUE
C
WRITE RESULTS OF EPOCH(IDKREC) CALCS TO INTERNAL FILE
C
CATFIL(11:14)='.IMP'
OPEN (UNIT=1,STATUS='OLD',NAME=CATFIL,
      FORM='UNFORMATTED',ACCESS='DIRECT',RECL=8)
WRITE(1,REC=IDKREC) ((DATCAR(K,1),K=1,4)),DATSD
CATFIL(11:14)='.DIG'
CLOSE(UNIT=1)
CLOSE(UNIT=2)
TYPE *,' DO YOU WANT A COPY? [Y/N]'
ACCEPT 10000, ANS

10000 FORMAT(A1)
IF(ANS.EQ.'Y') THEN
   IRES(1)=1
   CALL CPRNT(ISTAT, IRES, 'DW1:FILE2.GID', 13)
ELSE
   OPEN (UNIT=1, STATUS='OLD', NAME='DW1:FILE2.GID')
   CLOSE (UNIT=1, DISPOSE='DELETE')
ENDIF

TYPE *,' EXIT? [Y/N]'
ACCEPT 10000, ANS
IF(ANS.EQ.'Y') GOTO 99999

IDKREC=IDKREC+1 !INDEX EPOCH COUNTER FOR PLOT AND RESULTS FILE
KREC=KREC+NREC !INDEX DIGITIZED DATA FILE RECORD COUNTER
IF(KREC.GE.INDREC) GOTO 99999 !AT END OF DATA FILE YET
GOTO 105

CONTINUE

WRITE(1,2) SLOPE, ENTRCP, C, BSA, CATFIL, MAXREC !RECORD MAXREC VALUE
CLOSE (UNIT=1)

WRITE(5,40000) 1233, 173, 1162

STOP
END

SUBROUTINE INDEX (XMIN, XMAX, YMIN, YMAX, NPTS)
PARAMETER (IFREQ=200)
DIMENSION X(1000), Y(1450)
COMMON /PLOT/X, Y

PRINT *, K, NPTS, (Y(I), I=I, NPTS)
SX=639./NPTS
YMIN=Y(1)

DO 300 I=I, NPTS
   PRINT *, Y(I), I
   IF(Y(I).LT.YMIN) YMIN=Y(I)
   IF(Y(I).GT.YMAX) YMAX=Y(I)
300 CONTINUE

SY=479./(YMAX-YMIN)
ENY=-479.*YMIN/(YMAX-YMIN)
XMIN=0.
XMAX=X(NPTS)

PRINT *, Y, YMIN, YMAX, NPTS
RETURN
END

SUBROUTINE VOL(NY, KPEAKS, IY, IADMIN, IGATE, IM, IDZDTO, IDZDTI)
DO PEAK DETECTION USING DIGITAL ROUTINE 'PEAK'
PARAMETER (IFREQ=200)
VIRTUAL IY (9000)
DIMENSION INPUT(15*IFREQ)
DIMENSION OUTPUT(10, 70)
DIMENSION ITABLE(68), VTYPE(2, 2)
DIMENSION Y(1450)
COMMON /PLOT/IX(2000), Y

DATA ITABLE/1, 5, 5,800,1,0,0,61*0/
DATA INLAST, IDIMO/600,70/
CALL ERRSET(112,TRUE,TRUE,TRUE,TRUE,TRUE)

DZREF=IDZDTO+(IDZDT1-IDZDTO)*.5

TYPE *,' HERE IN VOL', NY, IY

SET TABLE VALUES FOR DIGITAL PEAK DETECTION ROUTINE

ITABLE(6)=0
ITABLE(7)=0
INPTR=0
INLAST=NY
NPEAKS=0

ITABLE(3)=0 !SET GATE FACTOR
ITABLE(4)=100 !SET MIN DIFFERENCE FACTOR
IF(IGATE.NE.0) THEN !RESET IGATE AND IM IF REQUESTED BY USER
   ITABLE(3)=IGATE
   ITABLE(4)=IM
END IF

TRANSCRIBE DZDT DATA TO ARRAY INPUT

N=0
DO 1000 I=1,NY
   INPUT(I)=IY(3000+I)
   TYPE *,I,INPUT(I),IY(I)
1000 CONTINUE
IADMIN=-16000
IADMIN=IMIN(INPUT,NY)
OPEN (UNIT=2,TYPE='NEW',NAME='INPUT.DAT',FORM='FORMATTED')
WRITE (2,1980) (INPUT(I),I=I,NY)
FORMAT(1110)
CLOSE (UNIT=2)
IF(IADMIN.LT.0) THEN !OFFSET INPUT VALUES SO ALL ARE NON NEGATIVE
   DO 2000 I=1,NY
      INPUT(I)=INPUT(I)-IADMIN
   CONTINUE
END IF
TYPE *, (INPUT(I),I=1,NY),IADMIN
WRITE(2,*)((INPUT(I),I=1,NY),IADMIN,SLOPE,INTRCP

CALL ROUTINE TO FIND PEAKS AND TRAILING MINS TIMES
FOR dZ/dT MIN=PEAKS HEIGHT AND T=TIME AT TRAILING
MINS - TIME WHEN dZ/dT=0 BEFORE PEAKS
OUTPUT ARRAY CONTAINS RESULTS OF PEAK DETECTION

CALL PEAK(ITABLE,INPUT,INLAST,INPTR,OUTPUT,IDIMO,NPEAKS)
TYPE *,NPEAKS,INLAST,INPTR

50
NEXT SECTION NEEDED BECAUSE PEAK ROUTINE MAY AVERAGE INPUT FOR PEAKS

DO 2500 I=1,NPEAKS
OUTPUT(2,I)=INPUT(IFIX(OUTPUT(3,I)))+IADMIN

IF(IADMIN.LT.0) THEN !CONVERT RESULTS IF DATA OFFSET
DO 3000 I=1,NPEAKS
OUTPUT(2,I)=OUTPUT(2,I)+IADMIN
OUTPUT(4,I)=OUTPUT(4,I)+IADMIN
OUTPUT(7,I)=OUTPUT(7,I)+IADMIN

END IF

REFINE PEAKS BASED ON SOME CRITERION IF NECESSARY

KPEAKS=0
DO 4000 I=1,NPEAKS
IF(OUTPUT(2,I).LE.DZREF) GOTO 4000
KPEAKS=KPEAKS +1
IF(KPEAKS.EQ.1) GOTO 4000
DO 3500 J=1,10
OUTPUT(J,KPEAKS)=OUTPUT(J,I)
CONTINUE

WRITE (2, * ) NPEAKS,INLAST,INPTR
WRITE (2,20000)
WRITE (I ,20000)
FORMAT(' PEAK NO.',8X,'AREA',4X,'P HEIGHT',6X,'P TIME',4X,
'A L HEIGHT',6X,'L TIME'/,11X,'HALF WIDTH',4X,'T HEIGHT',6X,
'B T TIME',8X,'TYPE',8X,'RATE'//)
DO 4 L=1,NPEAKS
KK=OUTPUT(9,L)+I
WRITE(2, 30000) (L,(OUTPUT(I,L),I=I,8),(VTYPE(M,KK),M=I,2),
OUTPUT(IO,L))
WRITE(I , 30000) (L,(OUTPUT(I,L),I=I,8)_(VTYPE(M,KK),M=I,2),
OUTPUT(IO,L))
FORMAT(I9,5F12.0,/,9X,3F12.0,4X,2A4,F12.0)
CONTINUE
RETURN
END

FUNCTION IMIN (INPUT,ICOUNT)
DIMENSION INPUT(ICOUNT)
IMIN=INPUT(1)
DO 1000 I=2,ICOUNT
IF(INPUT(I).LT. IMIN) THEN
IMIN=INPUT(I)
END IF
1000 CONTINUE
RETURN
END

SUBROUTINE TFACT (NPEAKS,IFREQ,C,INL_ST,K,TYMEAN,
I SLOPE,INTRCP,BSA,IZO,IREC,IDZDTO,IADMIN,CATFIL,IDKREC)
PARAMETER (JFREQ=200)
PARAMETER (NREC=15 )
PARAMETER (NUM=220)
DIMENSION IREC(NUM),ZOEPC(NUM)
DIMENSION INPUT(15*JFREQ),OUTPUT(IO,70),DATCAR(5,50)
VIRTUAL IZO(9000)
COMMON /PLOT/X(1000),Y(1450)
REAL SLOPE(2),INTRCP(2)
CHARACTER_14 CATFIL
EQUIVALENCE(Y(501),OUTPUT),(X,INPUT),(DATCAR,Y(1201))
LOGICAL_I TYMEAN
DATA ICOUNT/O/
DATA IEPC/ ,2,3,4,
15,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,
125,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,
146,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,
167,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,
1 87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,
119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,
2135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,
3151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,
3166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,
5181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,
6196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,
7211,212,213,214,215,216,217,218,219,220/
DATA ZOEPC/20.9,21.0,21.0,21.0,20.9,21.0,20.9,20.9,21.0,21.0,
1 21.0,20.9,20.9,21.0,20.9,20.9,20.9,20.9,20.9,20.9,20.9,21.0,
1 21.0,20.9,20.9,21.0,20.9,20.9,20.9,20.9,20.9,20.9,20.9,21.0,
2 20.9,21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,
7 21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,21.0,
DO 200 I=1,NREC
KREC=IREC+1-I
CALL DIRECT (KREC,CATFIL,2,1,IZO)
CONTINUE
150 CONTINUE

ORIGINAL PAGE OF POOR QUALITY
CONTINUE
NDZDTO=IDZDTO
IF(IADMIN.LT.0.0) NDZDTO=IDZDTO-IADMIN
DO 100 I=1,2
SLOPE(I)=1.
INTRCP(I)=1.
100 CONTINUE
C=1.
BSP=1.
DO 1000 I=1,NPEAKS
ISTRT=OUTPUT(3,I)
DO 400 J=ISTRT,2,-1
IF(INPUT(J).GT.NDZDTO ) GOTO 400
ISTRT= J
GOTO 500
400 CONTINUE
TSTRT=0
DO THE BEGINNING OF THE CIRCLE BUT STILL GOING DOWN; CHECK
ISTRT FOR STARTING TIME
ISTRT=170
425 CONTINUE
IF(ICOUNT.GT.1) THEN
DO 450 J=ISTRT,2,-1
IF(INTRCP(J-1).LT.INTRCP(J) ) GOTO 450
ISTRT=600.*(ICOUNT-1)-(170.-J+1)
GOTO 500
450 CONTINUE
TYPE *, ' ERROR NO BOTTOM '
END IF
ISTRT=600.*(ICOUNT-1)
500 CONTINUE
TIME=(OUTPUT(8,1)-TSTRT)
CALL CGL(37,3,0)
CALL CGL(33,TSTRT,350.)
CALL CGL(37,5,0)
CALL CGL(33,OUTPUT(8,1),350.)
ZNOT:ZOEPC (ISTRT)
DO 550 J=1,NUM
IF(IDKREC.EQ.IEPC(J)) THEN
ZNOT: ZOEPC (J)
GOTO 575
ENDIF
550 CONTINUE
ENDIF
575 CONTINUE
DZDT=OUTPUT(2,I)*SLOPE(2)+INTRCP(2)
VNTVOL=C*TIME*(DZDT
DATCAR(I,I)=VNTVOL
IF(I.EQ.I) THEN
BPM=IFREQ*60./(OUTPUT(3,2)-OUTPUT(3,1))
ELSE
BPM=IFREQ*60./(OUTPUT(3,1)-OUTPUT(3,1-1))
END IF
53
SAVE TIME FACTORS FOR HI FROM DZDT DATA AND DZDT PEAK VALUES

WHERE HI = DATCAR(4,1)/[Q-DATCAR(5,1)]
AND Q IS TIME OF Q WAVE BEFORE DZDT PEAK

WRITE(2,* ) OUTPUT(2,1),TIME,VMTVOL,TSTRT,J,ISTRT,
ZNOT,C,DZDT,BPM,SLOPE,INTRCP,IZO(ISTRT)

CONTINUE

OLDOUT=OUTPUT(3,NPEAKS)
DO 4000 J=431,600
CONTINUE
RETURN
END

SUBROUTINE STRVOL (KPEAKS,DATSD,IPEAKS)
PARAMETER (IFREQ=200)
DIMENSION DATCAR(5,50),DATSD(4),IPEAKS(KPEAKS)

CONTINUE

DATCAR(I,I)=SUM/KPEAKS
DATSD(1)=SQRT(KPEAKS SUMSQ-SUM SUM)/KPEAKS
CONTINUE
RETURN
END

SUBROUTINE HICALC (NPEAKS,IFREQ,IECG,KREC,NREC,CATFIL)
PARAMETER (JFREQ=200)
VIRTUAL IECG(9000 )
DIMENSION OUTPUT(10,70),DATCAR(5,50),INPUT(15*JFREQ)
CHARACTER*14 CATFIL
COMMON /PLOT/X(IOOO),Y(1450)
EQUIVALENCE (Y(501),OUTPUT),(Y(1201),DATCAR),(X,INPUT)
N=0
DO 200 I=1,NREC
CONTINUE
RETURN
END

ORIGINAL PAGE IS OF POOR QUALITY
CALL DIRECT (IRREG, SATFIL, 5, IECG)

DO 150 J = 1, IFREQ

IMAX = N + 1

IECG (N) = ISPACE ( J , 1 )

150 CONTINUE

DO 250 CONTINUE

DO 1000 I = 1, NPEAKS

IEND = DATCAR ( 5, I )

IF ( I .NE. 1 ) THEN

ISTRT = ( IEND + DATCAR ( 5, I - 1 ) ) / 2

ELSE

ISTRT = ( I - IEND ) / 2

END IF

ICOUNT = IEND - ISTRT + 1

DO 500 J = 1, ICOUNT

INPUT ( J ) = IECG ( ISTRT + J - 1 + 6000 )

500 CONTINUE

TIME = ISTRT - IMAX ( INPUT, ICOUNT ) + 1

TIME = TIME

CALL CCL ( 32, TIME, 150, )

TIME = ( DATCAR ( 5, I ) - TIME ) / IFREQ

IF ( TIME ) .GT. 600, 600, 700

600 CONTINUE

DATCAR ( 4, I ) = 0.0

GOTO 1000

700 CONTINUE

DATCAR ( 4, I ) = DATCAR ( 4, I ) / TIME

1000 CONTINUE

RETURN

END

FUNCTION IMAX ( INPUT, ICOUNT )

DIMENSION INPUT ( ICOUNT )

IMAX = ICOUNT

DO 1000 I = ICOUNT - 1, 1, - 1

IF ( INPUT ( I ) .GT. INPUT ( IMAX ) ) THEN

IMAX = I

END IF

1000 CONTINUE

RETURN

END
A Computer Program for Processing Impedance Cardiographic Data: Improving Accuracy Through User-Interactive Software

Patricia S. Cowings, Karen Naifeh,* and Chet Thrasher*

Ames Research Center
Moffett Field, CA 94035
*University of California, San Francisco, CA 94122

This report contains the source code and documentation for a computer program used to process impedance cardiography data. The cardiodynamic measures derived from impedance cardiography are ventricular stroke volume, cardiac output, cardiac index and Heather index. The program digitizes data collected from the Minnesota Impedance Cardiograph, electrocardiography (ECG), and respiratory cycles and then stores these data on hard disk. It computes the cardiodynamic functions using interactive graphics and stores the means and standard deviations of each 15-sec data epoch on floppy disk. This software was designed on a Digital PRO380 microcomputer and used version 2.0 of P/OS, with (minimally) a 4-channel 16 bit analog/digital (A/D) converter. Applications software is written in Fortran 77, and uses Digital's Pro-tool Kit Real Time Interface Library (PRTIL), CORE Graphic Library (CGL), and laboratory routines. Source code can be readily modified to accommodate alternative detection, A/D Conversion and interactive graphics. The object code utilizing overlays and multitasking has a maximum of 50 Kbytes.