INSTRUMENTATION LABORATORY
DEPARTMENT OF ELECTRICAL ENGINEERING
UNITED STATES AIR FORCE ACADEMY, COLORADO

PROCESSING OF THE LEG VOLUME DATA OBTAINED ON THE M092 SKYLAB MISSIONS

RICHARD D. BARNETT
CRISTY M. SCHADE, LT, USAF
RICHARD J, GOWEN, LT COL, USAF



MAY, 1974

INSTRUMENTATION LABORATORY
DEPARTMENT OF ELECTRICAL ENGINEERING
UNITED STATES AIR FORCE ACADEMY, COLORADO

PROCESSING OF LEG VOLUME DATA OBTAINED ON THE M092 SKYLAB MISSIONS

RICHARD D. BARNETT
CRISTY M. SCHADE, LT, USAF
RICHARD J. GOWEN, LT COL, USAF

MAY, 1974

The Processing of Leg Volume Data from the MO92 Skylab Missions was performed in the Instrumentation Laboratory of the Department of Electrical Engineering at the United States Air Force Academy. This work was supported under NASA Project No. T66344(G).

TABLE OF CONTENTS

List of Figures..... iii
Introduction..... 1
Description of the Skylab leg volume measuring system.... 3
Description of LVMS signal processing system..... 6
Flowchart..... 7
Data format on magnetic tape..... 8
Noise and offset corrections..... 19
Entry of end of period values..... 36
Normalization of data..... 36
Foam correction..... 39
Sequencing of data for statistical processing..... 40
Special options..... 42
Summary..... 46
Appendix..... 47

LIST OF FIGURES

Figure 1 - Skylab Leg Volume Measuring Band..... 2

Figure 2 - Plethysmographs on the Legs of an Astronaut..... 4

Figure 3 - Output Parameters From MO92 Experiment..... 5

Figure 4 - Scheme for Leg Volume Processing..... 7

Figure 5 - Signal Processing at the Instrumentation Laboratory..... 9

Figure 6 - XDS Sigma 5 Digital Computer..... 10

Figure 7 - Interactive Graphics Terminal..... 11

Figure 8 - Interactive Graphics Terminal..... 12

Figure 9 - Block Diagram of Signal Processing Components... 13

Figure 10 - Plots of Unprocessed MO92 Data..... 20

Figure 11 - Plot of Data with Offsets on left leg output.... 21

Figure 12 - Plot of Corrected PLVC Data..... 22

Figure 13 - Plot of Data with Offsets on Right Leg Output... 24

Figure 14 - Plot of Corrected Left Leg Data..... 25

Figure 15 - Plot of Data with Noise on Both Left and Right Leg Outputs..... 26

Figure 16 - Final Processed PLVC Data..... 27

Figure 17 - Plot of Data with Noise on Both Left and Right Leg Outputs..... 28

Figure 18 - Final Processed PLVC Data..... 29

Figure 19 - Plot of Data with Noise and Electrical Spikes... 30

Figure 20 - Final Processed Data..... 31

Figure 21 - Plot of Data with Electrical Spikes..... 32

Figure 22 - Final Processed Data..... 33

Figure 23 - Plot of Data Badly Contaminated with Noise, Spikes and Offsets..... 34

Figure 24 - Final Processed Data..... 35

Figure 25 - Plot of Un-normalized Data..... 37

Figure 26 - Plot of Normalized Data..... 38
Figure 27 - Plot Composite Data..... 41
Figure 28 - Plot of Unprocessed Data..... 43
Figure 29 - Plot of Processed Data..... 44
Figure 30 - Plot of Data After Digital Filtering..... 45

INTRODUCTION

The design of the leg volume measuring system employed for the Skylab mission involved the development of a capacitive measurement system to detect small changes in leg volume. These changes in volume of the calf were produced in response to the stress of a Lower Body Negative Pressure Device (LBNPD) or by venous occlusion.

The prototype capacitance plethysmograph was designed at the Air Force Academy and space qualified by the Martin-Marietta Corporation, Denver, Colorado for use on the Skylab Missions.

A capacitive plethysmograph must be calibrated once to establish the correlation between change in capacitance and change in volume of the limb segment being monitored. Each leg band contained its own internal reference as shown in Figure 1 which was used to establish the final calibration number for each individual leg band. The devices and techniques used to obtain the calibration numbers and the associated calibration curves for the seventy-six Skylab leg bands have been described in a previous technical report dated April, 1974. All calibration testing was performed by the Instrumentation Laboratory.

This report describes the system, techniques and algorithms used to process the Skylab Mission data and deals specifically with the signal processing performed at the Instrumentation Laboratory.

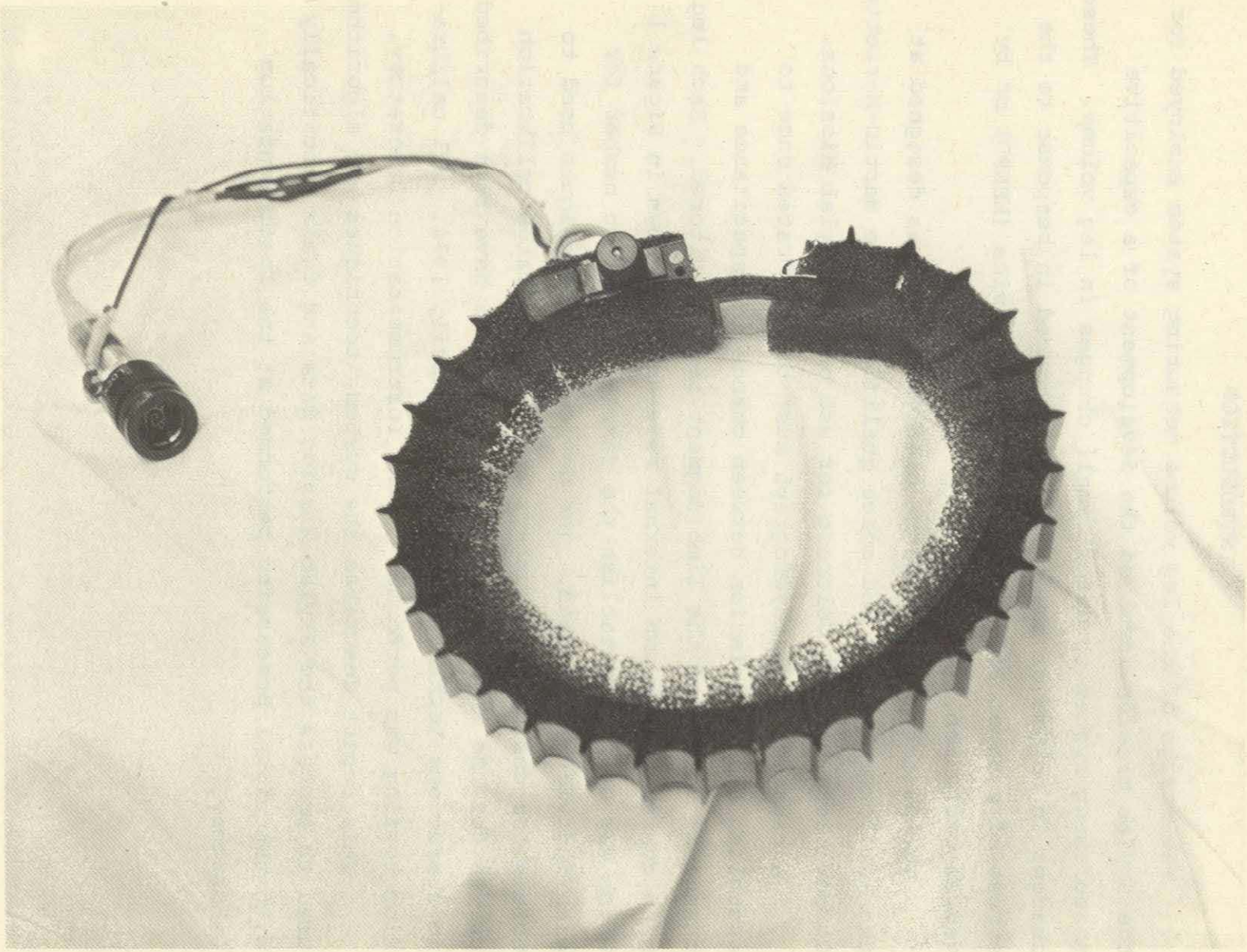


Figure 1. Skylab Leg Volume Measuring Band

SKYLAB LEG VOLUME MEASURING SYSTEM

The techniques used for measuring the volume change at the calf for the Skylab Missions seem at first appearance to be slightly unusual. The placement of the capacitive plethysmographs or leg bands on the legs of the astronaut is shown in Figure 2. In this configuration the right leg band which is located on a rigid reference band measures temperature and humidity changes within the LBNPD. The left leg band measures the same environmental conditions as well as the volume change occurring at the left calf. The left leg output (corrected) or difference signal is displayed as percent leg volume change (PLVC). These output parameters as well as the negative pressure (LBNP) are displayed on the graph of Figure 3. An increase in LBNPD humidity or a decrease in LBNPD temperature would cause an increased output from both the left and right leg bands, however, the PLVC output would not change.

This type of measuring technique was utilized after changes were incorporated to the LBNPD which restricted leakage flows to 1 SCFM of air. After the waist seal design was altered to minimize leakage, large variations in LBNPD temperature and humidity were observed during the 25 minute MO92 protocol. Since the LBNPD had no humidity sensor the decision was made to use one leg band as a temperature-humidity sensor by installing the band on a rigid reference adapter. By using the difference signal (i.e., left leg output- right leg output) this configuration produced data which was unaffected by changes in environmental conditions.

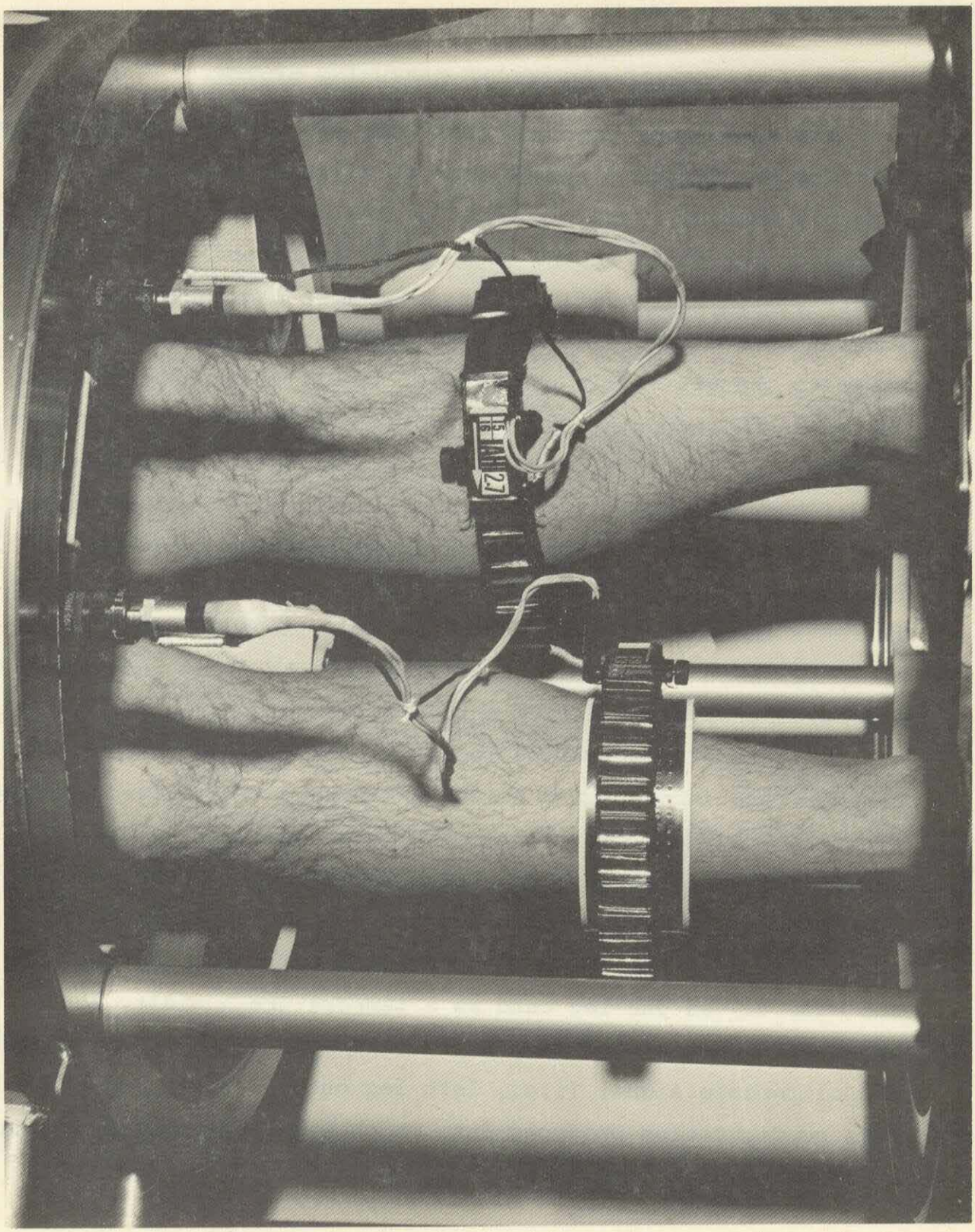


Figure 2. Capacitive Plethysmograph Installed
On The Legs Of An Astronaut In The Skylab Configuration

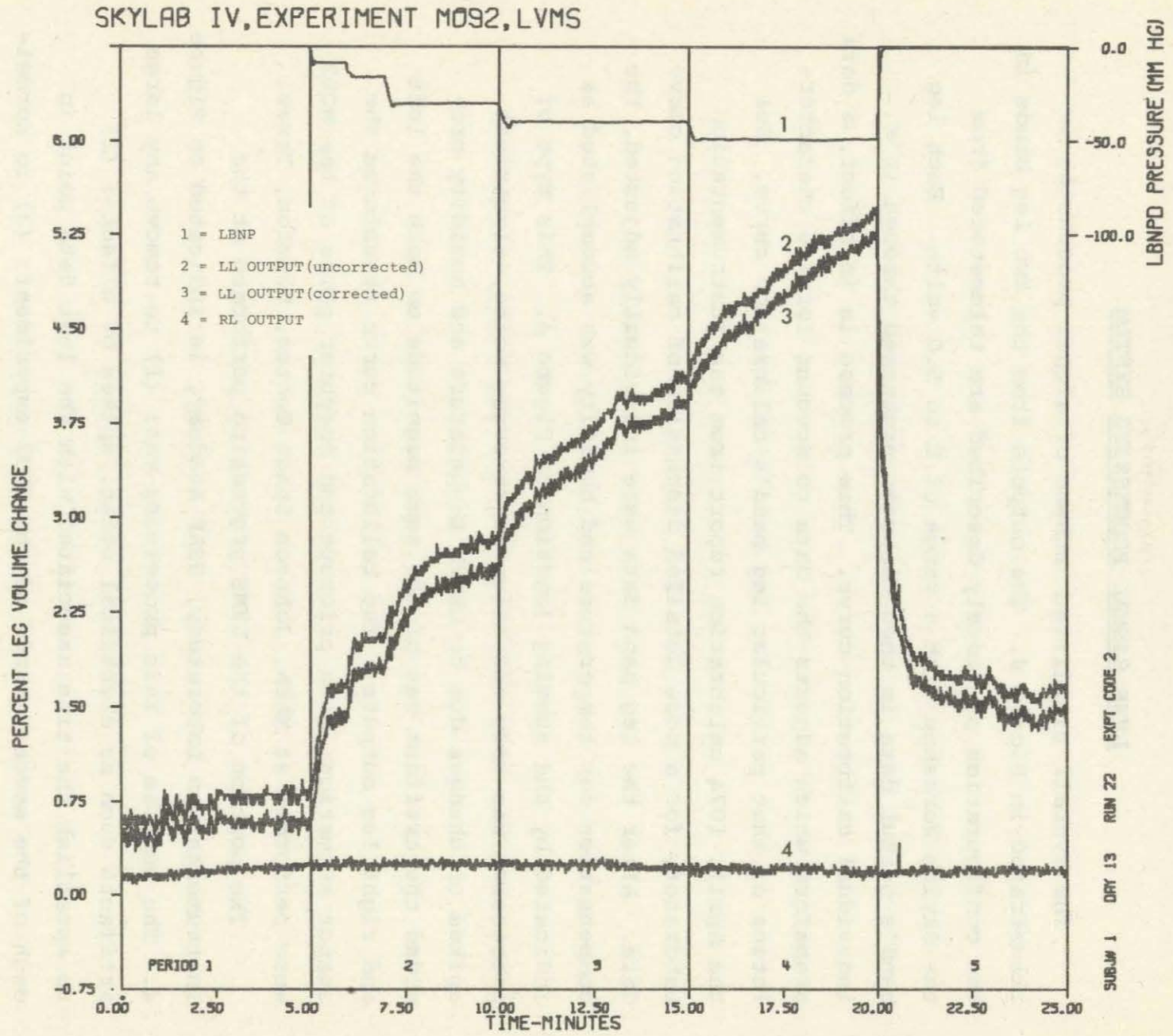


FIG. 3 OUTPUT PARAMETERS FROM MO92 EXPERIMENT

LVMS SIGNAL PROCESSING SYSTEM

The overall simplified scheme of signal processing is illustrated in Figure 4. The outputs from the two leg bands in the configuration previously described are telemetered from the Skylab Workshop with a range of 0 to 5.0 volts. Each leg band's output data is then computer processed through it's individual calibration curve. This process is in effect, a data transform which adjusts the data to account for the characteristics of that particular leg band's calibration curve. See the April, 1974 calibration report from the Instrumentation Laboratory for a more detailed discussion of calibration curve data. After the leg band data were individually adjusted, the compensation for temperature and humidity was accomplished as indicated by the summing junction of Figure 4. This type of processing was very effective at removing noise, electrical spikes or changes due to LBNPD temperature and humidity provided the artifact was of the same magnitude on both the left and right leg outputs. The calibration curve transforms, the output summation, data printout and computer plots of the M092 were performed at NASA, Johnson Space Center, Houston, Texas.

The portion of the LVMS processing performed at the Instrumentation Laboratory, USAF Academy, is indicated on Figure 4. The purpose of this processing was; (1) to remove any large artifacts such as electrical noise, spikes or offsets; (2) to establish the time associated with the last data point in each of the seven periods of the M092 experiment; (3) to normalize the data to a baseline just prior to onset of negative pressure; (4) to perform the foam correction required for each leg band; and (5) to add pertinent I.D. or experiment information to each M092 experiment.

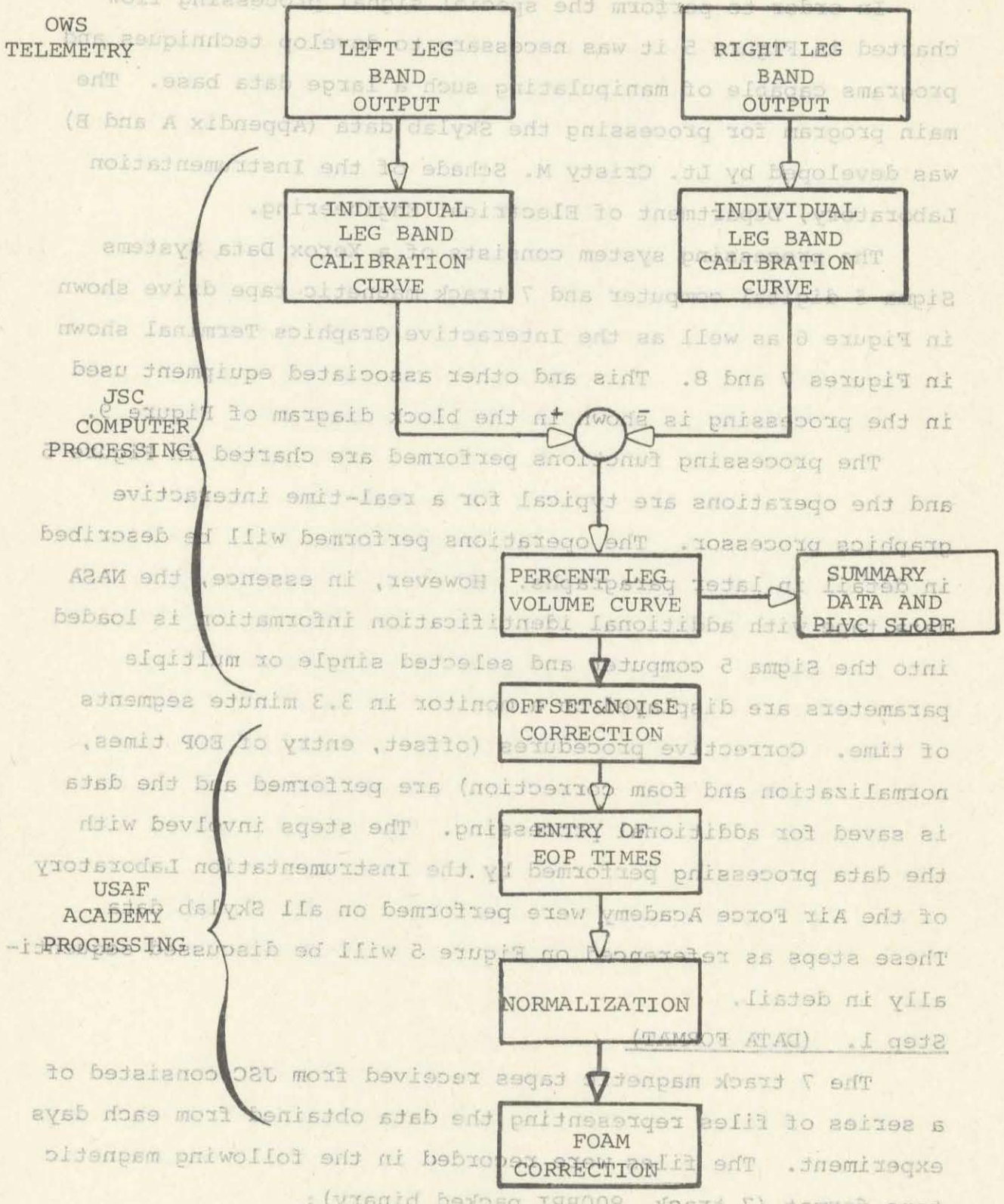


FIG. 4 SCHEME FOR LEG VOLUME PROCESSING

In order to perform the special signal processing flow charted in Figure 5 it was necessary to develop techniques and programs capable of manipulating such a large data base. The main program for processing the Skylab data (Appendix A and B) was developed by Lt. Cristy M. Schade of the Instrumentation Laboratory, Department of Electrical Engineering.

The processing system consists of a Xerox Data Systems Sigma 5 digital computer and 7 track magnetic tape drive shown in Figure 6 as well as the Interactive Graphics Terminal shown in Figures 7 and 8. This and other associated equipment used in the processing is shown in the block diagram of Figure 9.

The processing functions performed are charted in Figure 5 and the operations are typical for a real-time interactive graphics processor. The operations performed will be described in detail in later paragraphs. However, in essence, the NASA data tape with additional identification information is loaded into the Sigma 5 computer and selected single or multiple parameters are displayed on a monitor in 3.3 minute segments of time. Corrective procedures (offset, entry of EOP times, normalization and foam correction) are performed and the data is saved for additional processing. The steps involved with the data processing performed by the Instrumentation Laboratory of the Air Force Academy were performed on all Skylab data. These steps as referenced on Figure 5 will be discussed sequentially in detail.

Step 1. (DATA FORMAT)

The 7 track magnetic tapes received from JSC consisted of a series of files representing the data obtained from each days experiment. The files were recorded in the following magnetic tape format (7 track, 800BPI packed binary):

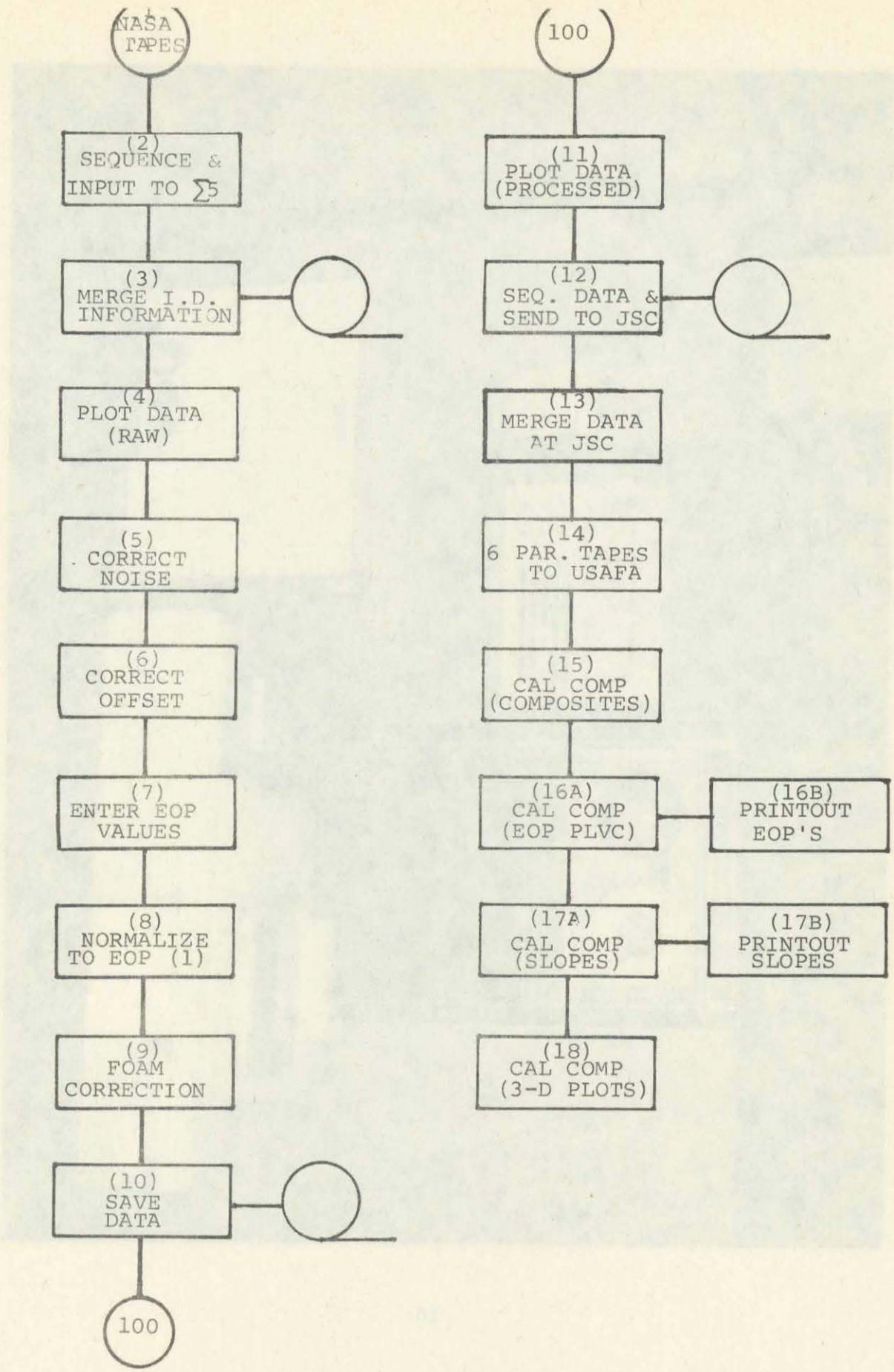


FIG. 5 SCHEME FOR PROCESSING LEG VOLUME

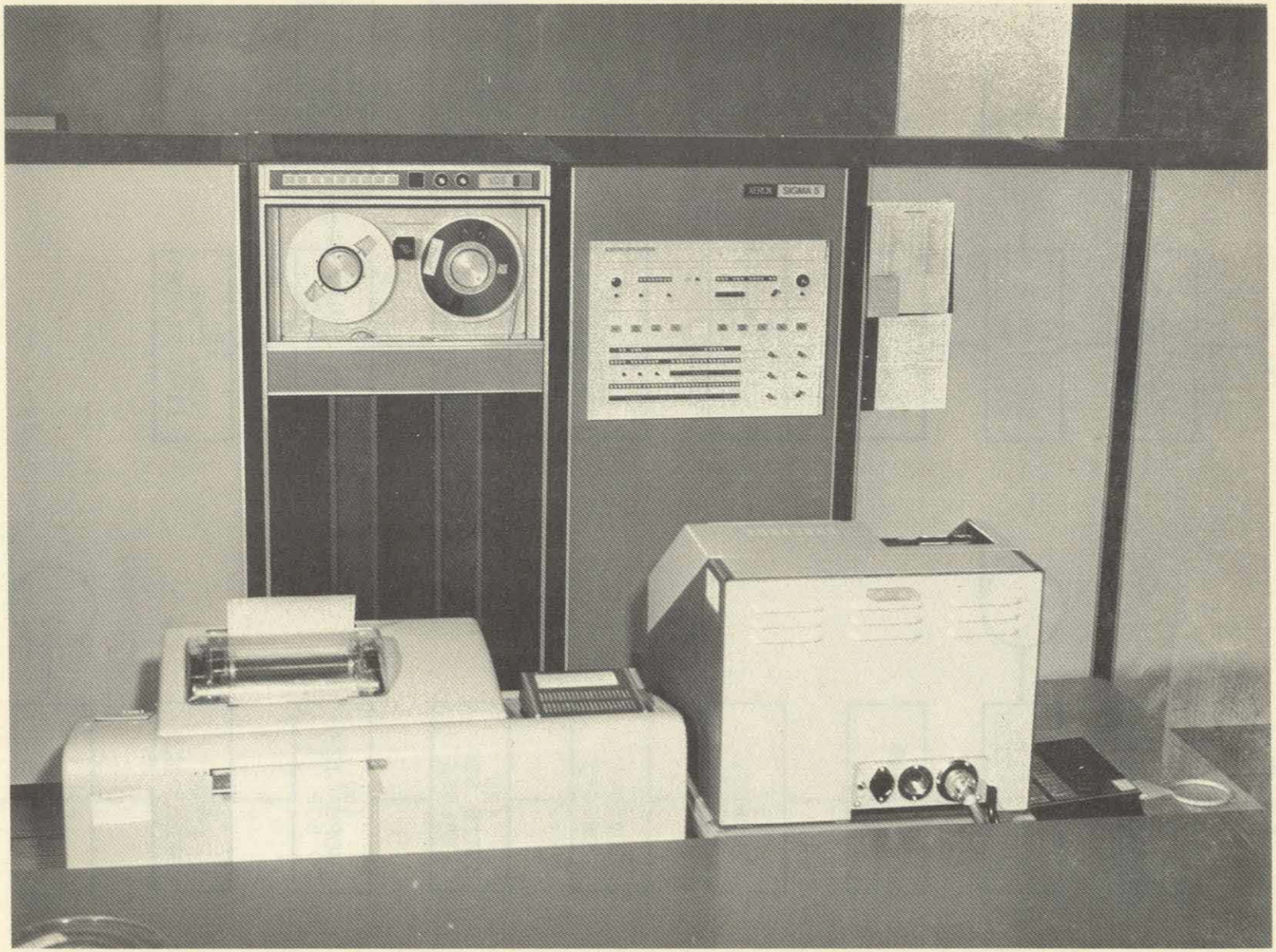


FIG. 6 XDS Sigma 5 Digital Computer

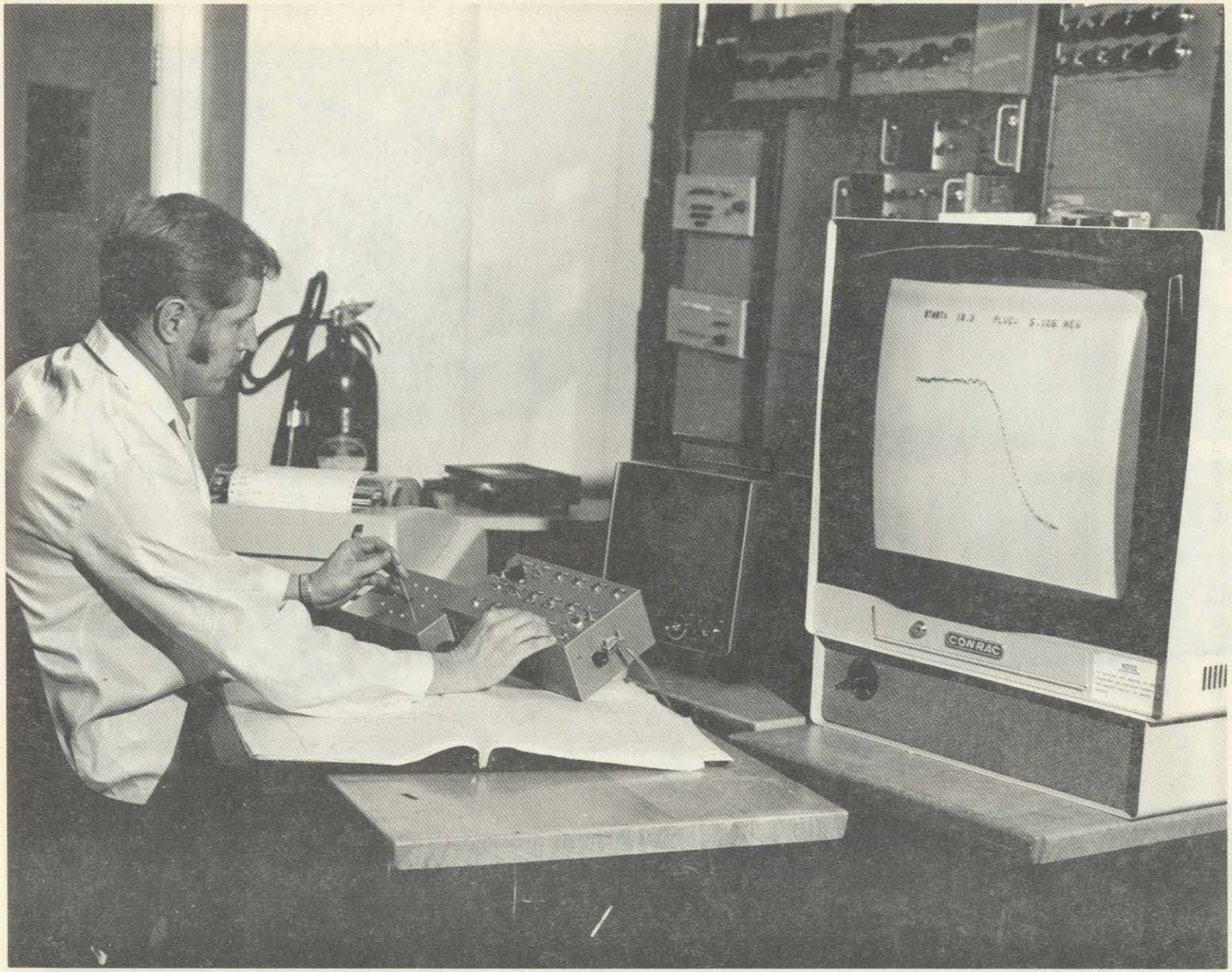


Figure 7. Interactive Graphics Terminal

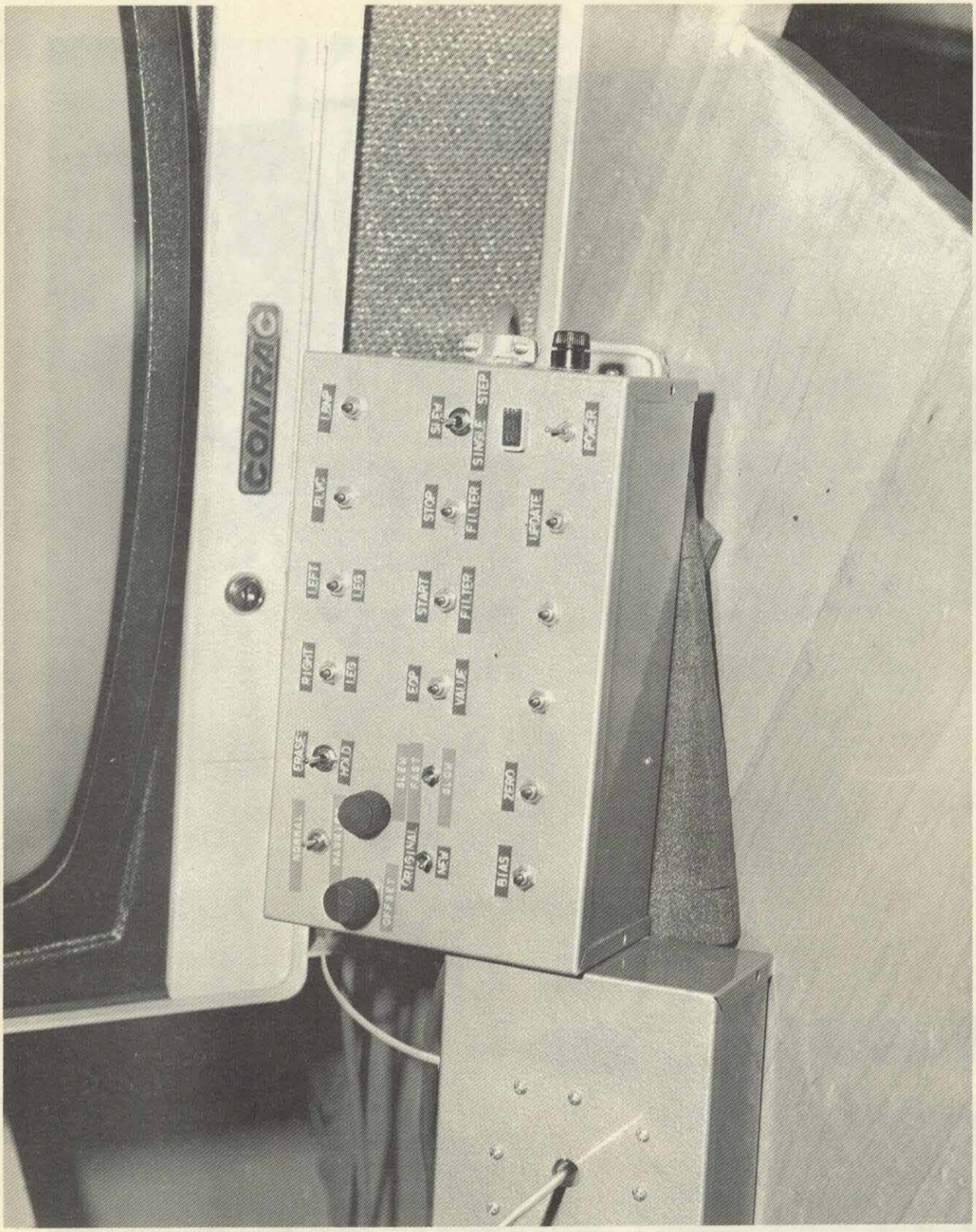


Figure 8. Interactive Graphics Terminal

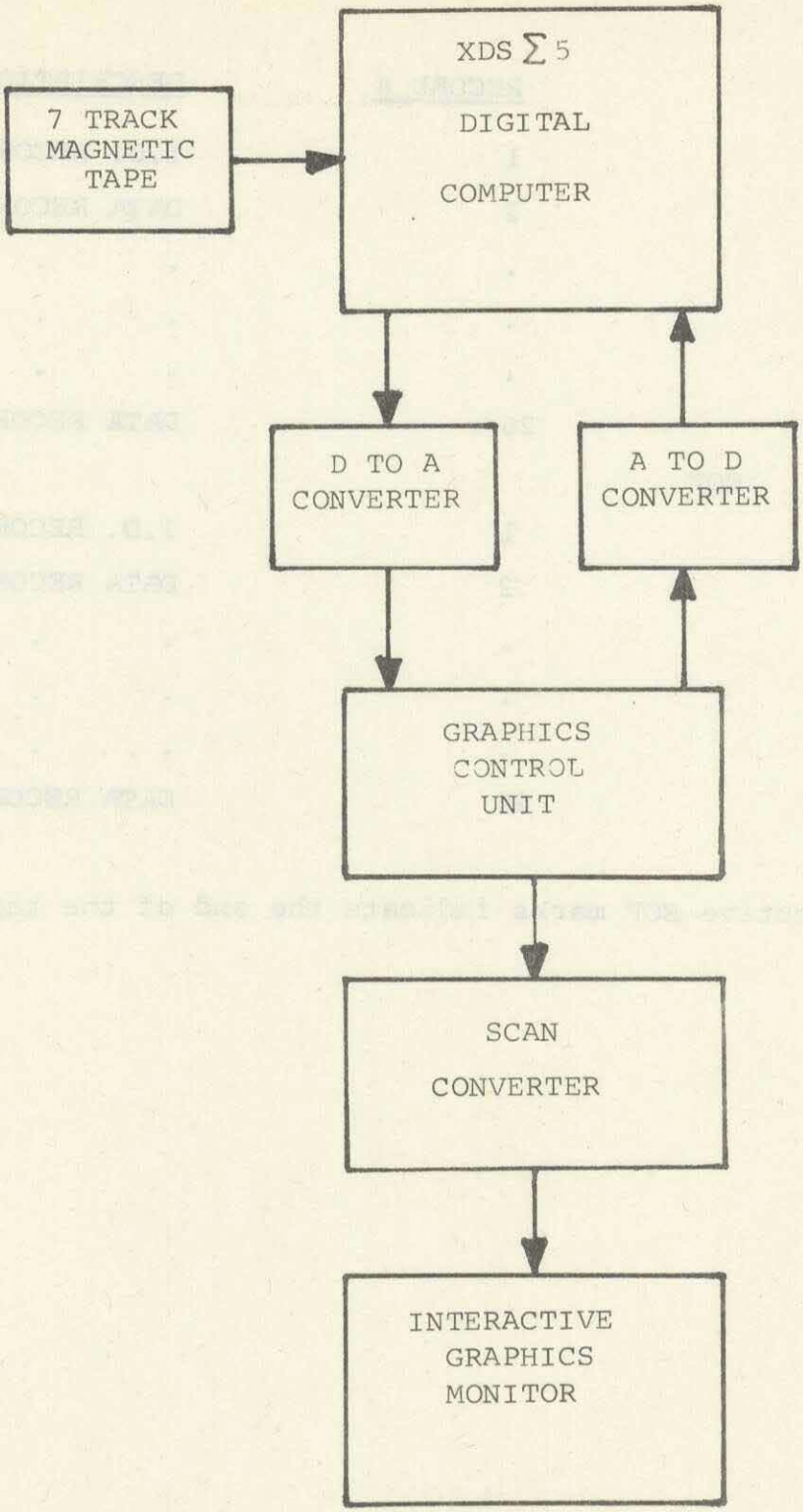


FIG. 9 BLOCK DIAGRAM FOR SIGNAL PROCESSING

FILE #	RECORD #	DESCRIPTION
1	1	I.D. RECORD
1	2	DATA RECORD
.	.	. .
.	.	. .
.	.	. .
1	26	DATA RECORD
	EOF	
2	1	I.D. RECORD
2	2	DATA RECORD
.	.	. .
.	.	. .
.	.	. .
2	26	DATA RECORD

Two consecutive EOF marks indicate the end of the tape.

Each I.D. record contained the following information:

32 bit	16 bit	INFORMATION
1	1	92
	2	Run No.
2	3	Subject I.D. No
	4	Julian Integer Day
3	5	Hours
	6	Minutes
4	7	Seconds
	8	0
5	9	Experiment Code (1,2,3)
	10	--
6	11	Beginning Period 2A
	12	
7	13	Beginning Period 2B
	14	
8	15	Beginning Period 2C
	16	
9	17	Beginning Period 3
	18	
10	19	Beginning Period 4
	20	
11	21	Beginning Period 5
	22	
12	23	End of Experiment
	24	

The One Minute Data Records Have The Following Format (32 Bit Word):

1	Elapsed Time in Seconds
2	Left Leg Volume
3	PLVC
4	Right Leg Volume
5	LBNP
.	.
.	.
.	.
375	LBNP

There are 75 data cycles per record since the data is sampled at 0.8 second intervals. When data is missing, $407FFFFF_{16}$ is used as fill.

Step 2. (SEQUENCE & INPUT DATA MAGNETIC TAPE)

The unsequenced magnetic tapes received from JSC were loaded into the Sigma 5 computer so that the files are sequenced by astronaut and by run number.

Step 3. (MERGE I.D. INFORMATION)

This step involved adding supplemental I.D., anthropometric, and environmental information to the I.D. record which expands the I.D. record from 12 to 42 elements. The program used to fill the additional 30 elements is contained in Appendix I. A sample output of 42 information elements for one experiment is printed in Appendix M. The additional information is contained in the I.D. array as follows:

USAF ADDED I.D. DATA

VARIABLE NAME*		DEFINITION
RID(13)		Julian Date
RID(14)	Left Leg	Band ID
RID(15)		Cal. Req.
RID(16)		Cal. Pre.
RID(17)		Cal. Post
RID(18)	Right Leg	Band ID
RID(19)		Cal. Req.
RID(20)		Cal. Pre.
RID(21)		Cal. Post
RID(22)		L.L. Circ.
RID(23)		R.L. Circ.
RID(24)		Body Mass
RID(25)		Body Temp.
RID(26)		Ambient Temperature (Start)
RID(27)		Ambient Temperature (End)
RID(28)	LBNPD Temperature	Start
RID(29)		P1
RID(30)		P4
RID(31)		P5
KID(32)		Subject ID Number
RID(33)		1-Elapsed Time End P1
RID(34)		2-Elapsed Time End P2A
RID(35)		3-Elapsed Time End P2B
RID(36)		4-Elapsed Time End P2C
RID(37)		5-Elapsed Time End P3
RID(38)		6-Elapsed Time End P4

VARIABLE NAME *	DEFINITION
RID(39)	7-Elapsed Time End P5
KID(40)	Run Type Code
KID(41)	Processing Code
KID(42)	Skylab Number

*"KID" and "RID" are the same array. "KID" is treated as integer data and "RID" is treated as real data.

Step 4. (PLOT OF UNPROCESSED DATA)

The unprocessed data, as received from JSC, was always plotted on a CAL COMP plotter with interconnecting lines between data points. These plots, such as the one in Figure 10, were then used for reference during the course of the processing. If fill characters were contained in the data, they were not plotted. Instead, the fill point was omitted and the curve was filled in by linear interpolation. These plots were very valuable in locating wild data points, electrical noise and offsets, timing errors and other variations in experimental conditions. In addition, after a file was processed it was again plotted and compared to the original plot to insure integrity of the processing. The program used for plotting is contained in Appendix G.

Steps 5 & 6. (CORRECTION OF NOISE AND OFFSETS)

For reasons that were not usually obvious some of the data contained unusual amounts of electrical noise and wild points combined with numerous offsets in the data. The noise and artifacts may have been produced by the telemetry system or may have been caused by workshop conditions or at times may have been caused by the electrical connection at the T-bar holder of the leg bands. Figures 11,13,15,17,19 and 21 illustrate some of the types of noise and offset conditions that were encountered. Figure 11 is a typical example of numerous offset conditions on the left leg data with no noise on the right leg data. As a consequence of the algorithm which subtracts the change in right leg band output from the left leg band output the PLVC curve displays all of the noise found on the left leg band output. In this case each offset must be manually removed from the PLVC curve. Figure 12 illustrates the PLVC after final processing.

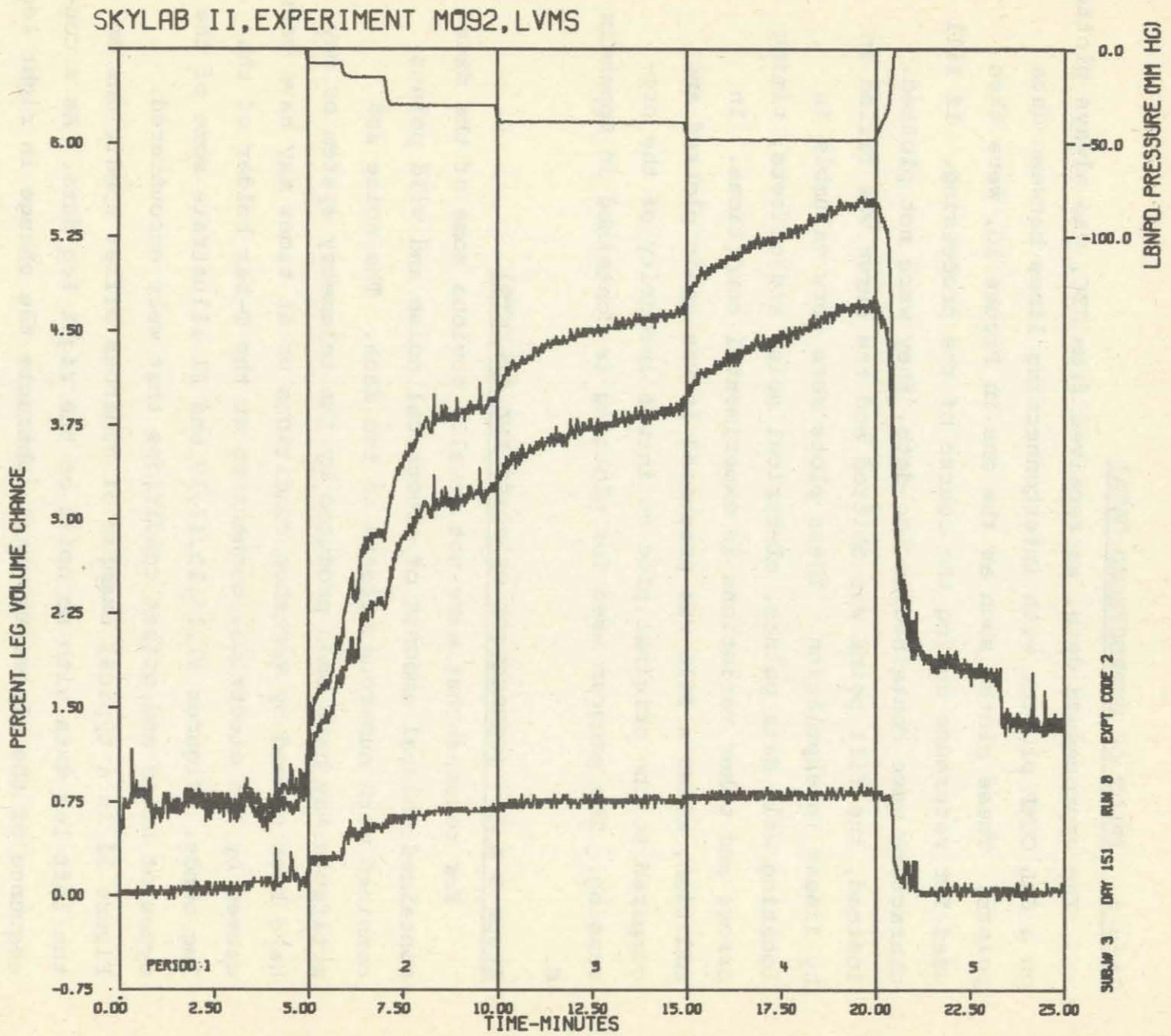


FIG. 10 PLOTS OF UNPROCESSED M092 DATA

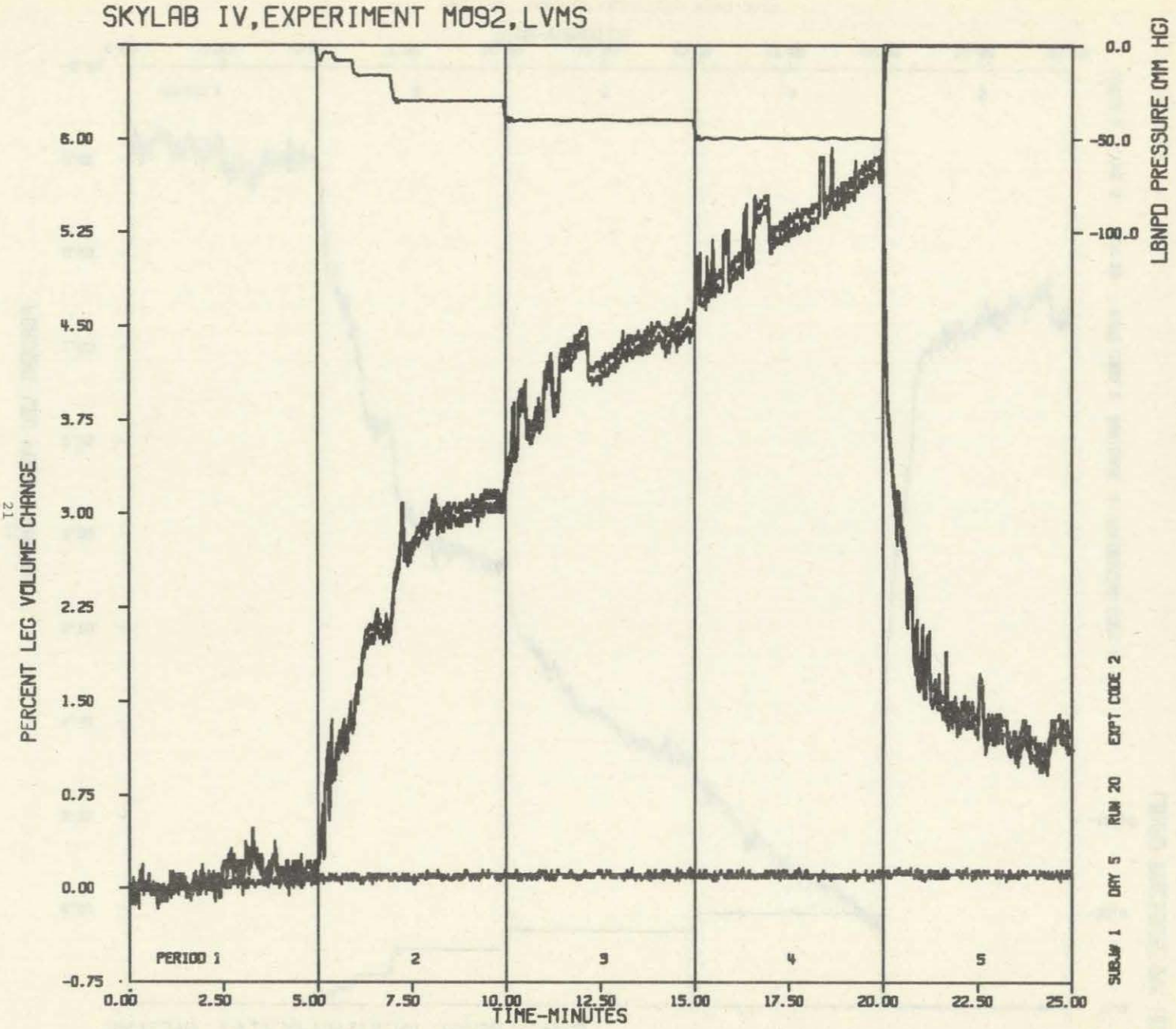


FIG. 11 PLOT OF DATA WITH OFFSETS ON LEFT LEG OUTPUT

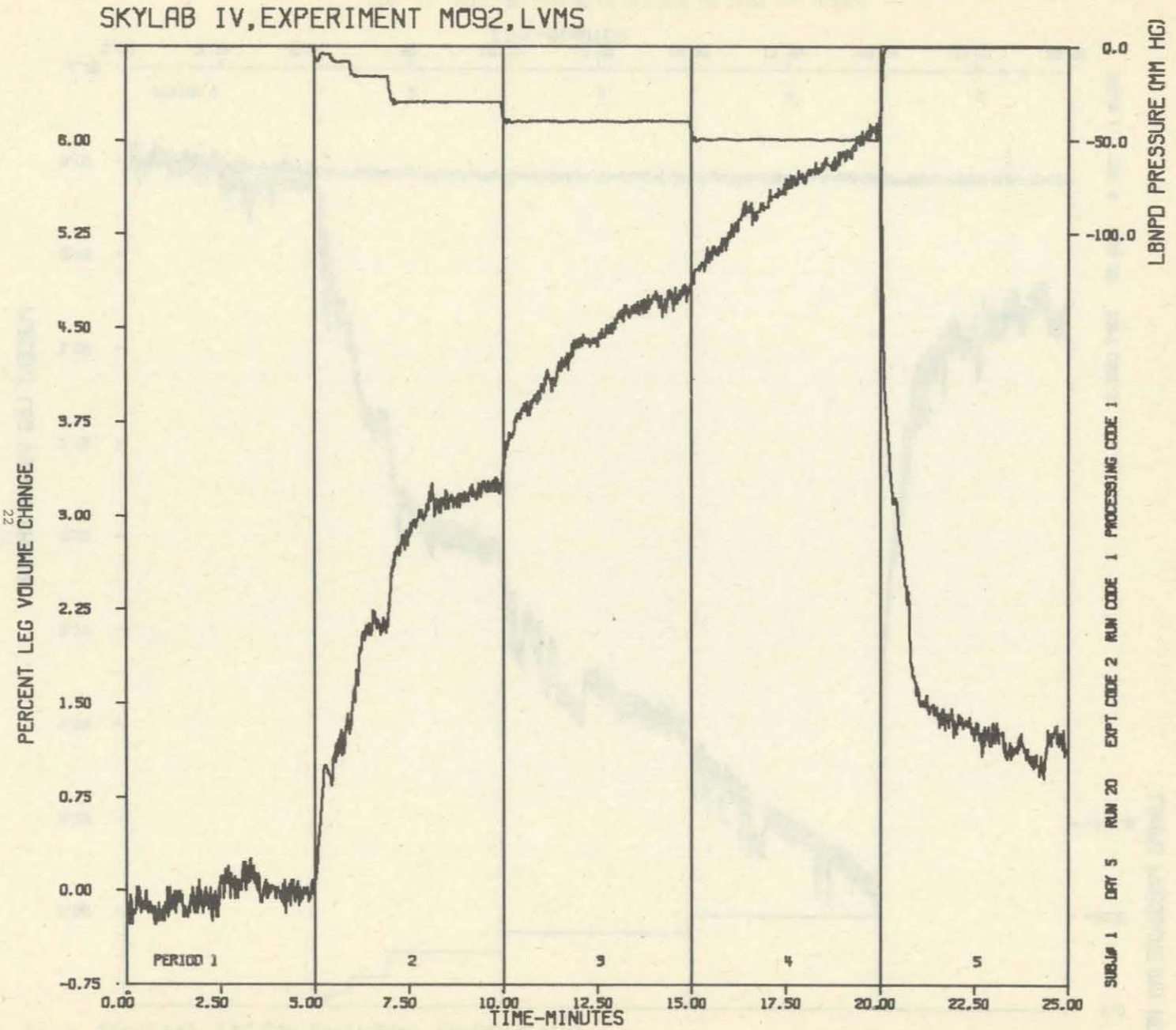


FIG. 12 PLOT OF CORRECTED PLVC DATA

The case where offsets are displayed on right leg data is shown in Figure 13. In this instance the right leg output had numerous offsets while the left leg data was reasonably free of offsets. In such instances the left leg data was processed instead of the PLVC data. This presents no particular problem except that the left leg data will not be temperature or humidity compensated. Figure 14 shows the data after final processing.

Figures 15 and 17 illustrate the case where system noise and offsets occurred on both the right and left leg outputs simultaneously. In cases such as these where system trouble occurred, the PLVC output will look very clean and present no problem for processing. Since the noise and offsets occur at both outputs, they cancel completely. The data in its final processed form is plotted in Figures 16 and 18.

Typical examples of data with numerous electrical noise spikes is illustrated in the plots of Figures 19 and 21. In this particular case most of the spikes were removed with a function of the Interactive Graphics Program- the Wild Point Editor. This function operates on the data by comparing a single point with a specific window of points. The function carries a running mean and if the single point being analyzed exceeds the mean by a specified number of standard deviations the point is edited out. Figures 20 and 22 show the data in its final processed form.

Figure 23 shows an example of data which is very close to being hopelessly contaminated with noise, electrical spikes and offsets. By careful and judicious use of the Wild Point Editor, Bias and Update controls the data can be reasonably reconstructed as shown in Figure 24.

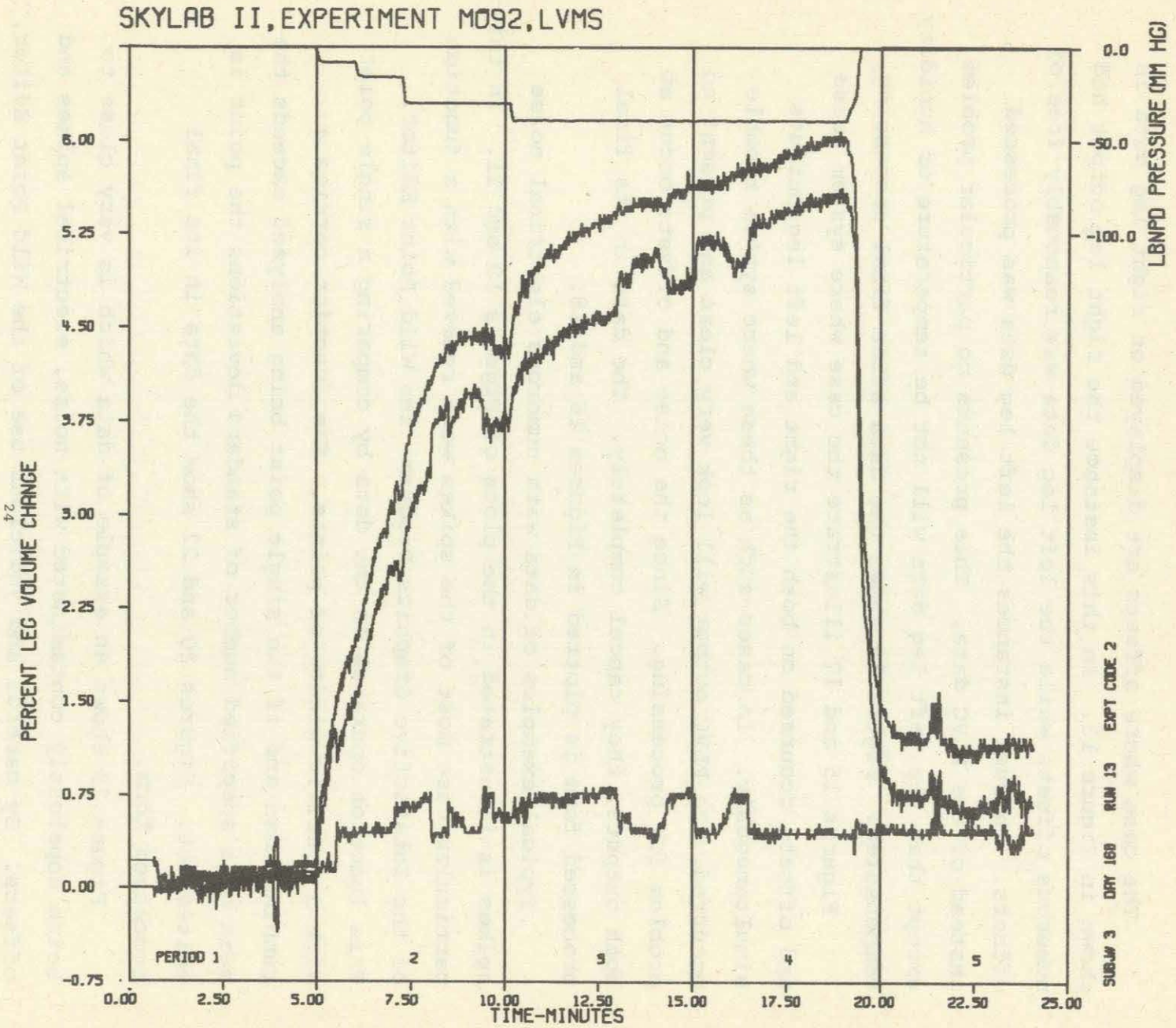


FIG. 13 PLOT OF DATA WITH OFFSETS ON RIGHT LEG OUTPUT

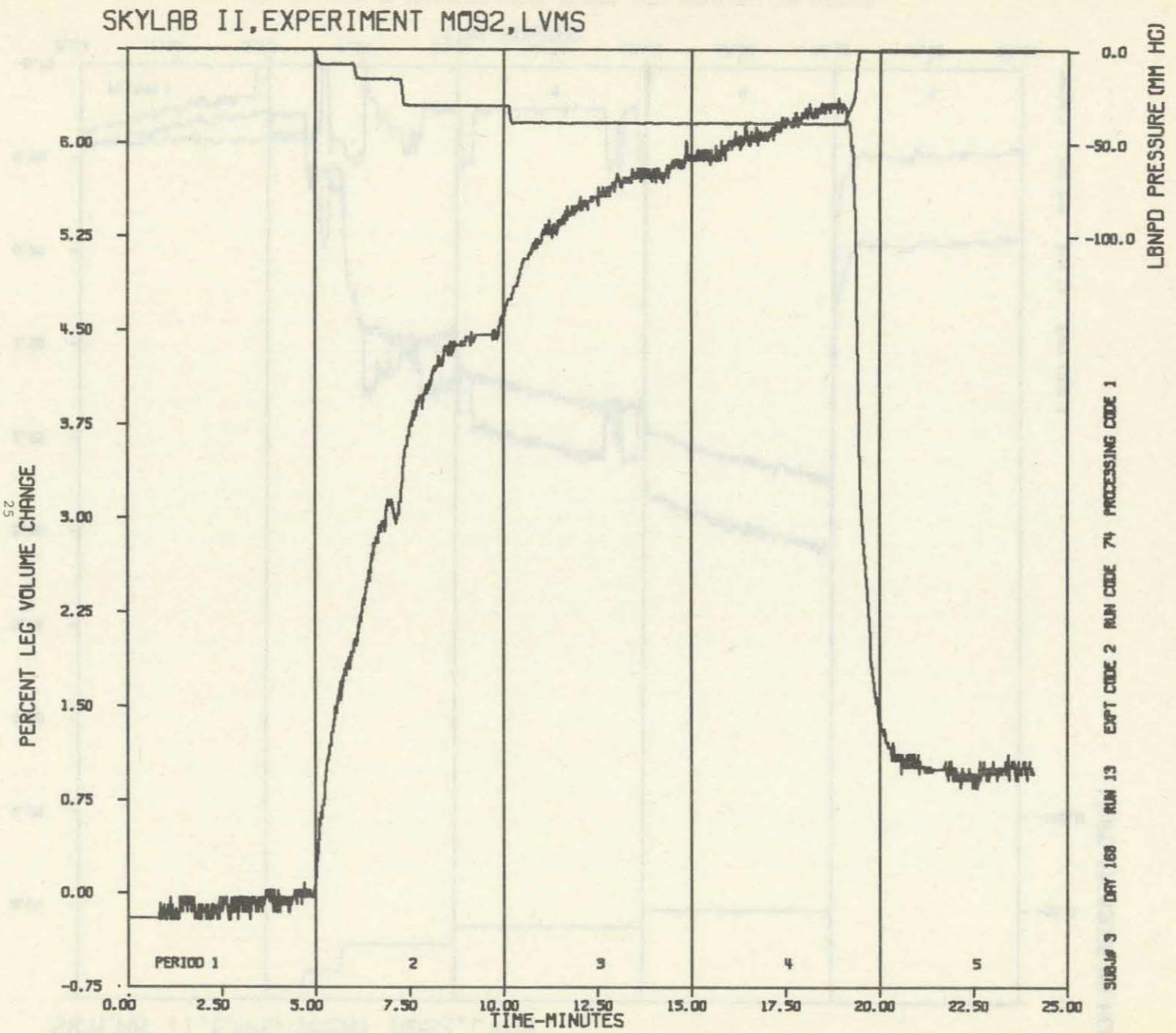


FIG. 14 PLOT OF CORRECTED LEFT LEG DATA

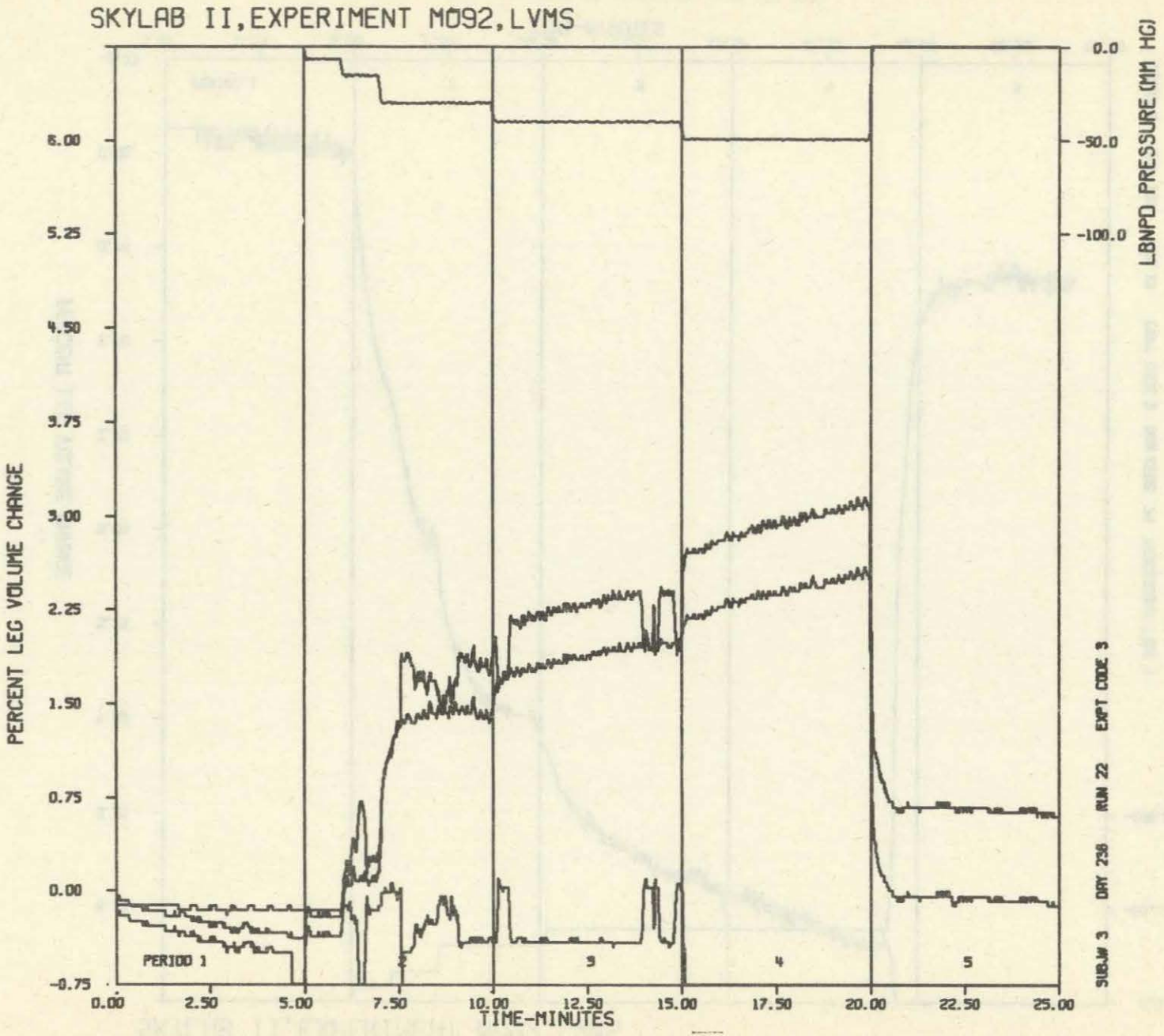


FIG. 15 PLOT OF DATA WITH NOISE ON BOTH LEFT AND RIGHT LEG OUTPUTS

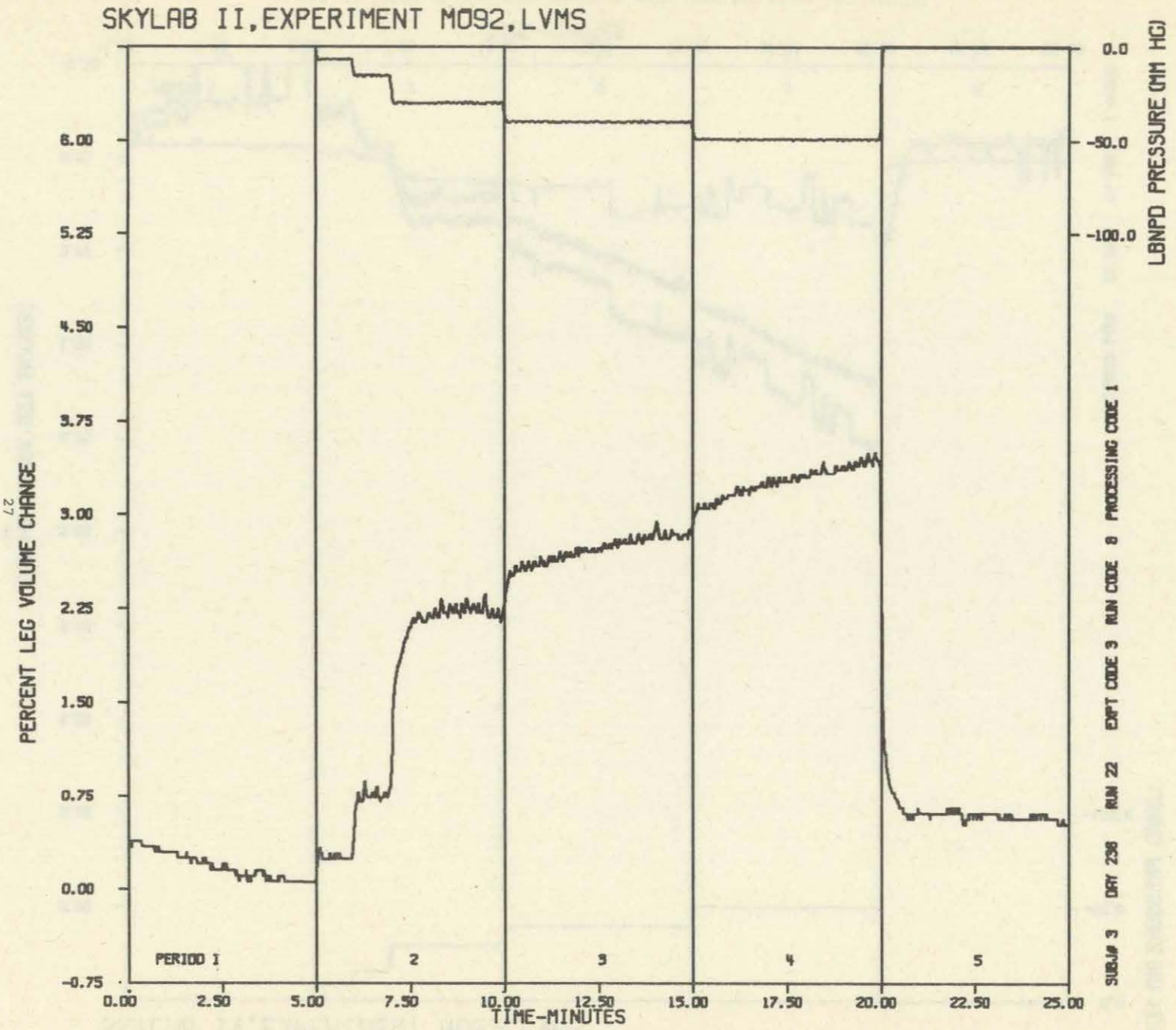


FIG. 16 FINAL PROCESSED PLVC DATA

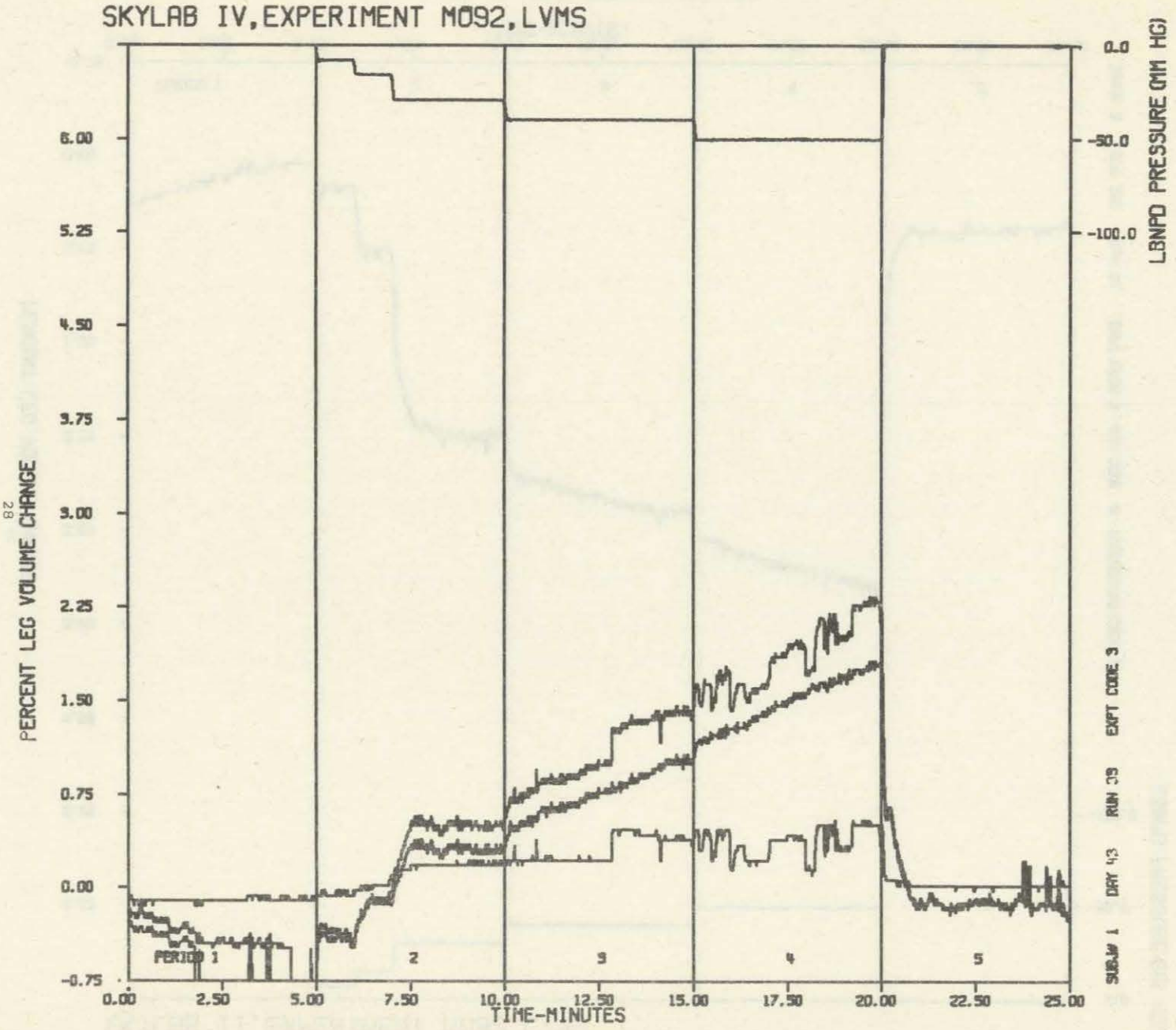


FIG. 17 PLOT OF DATA WITH NOISE ON BOTH LEFT AND RIGHT LEG OUTPUTS

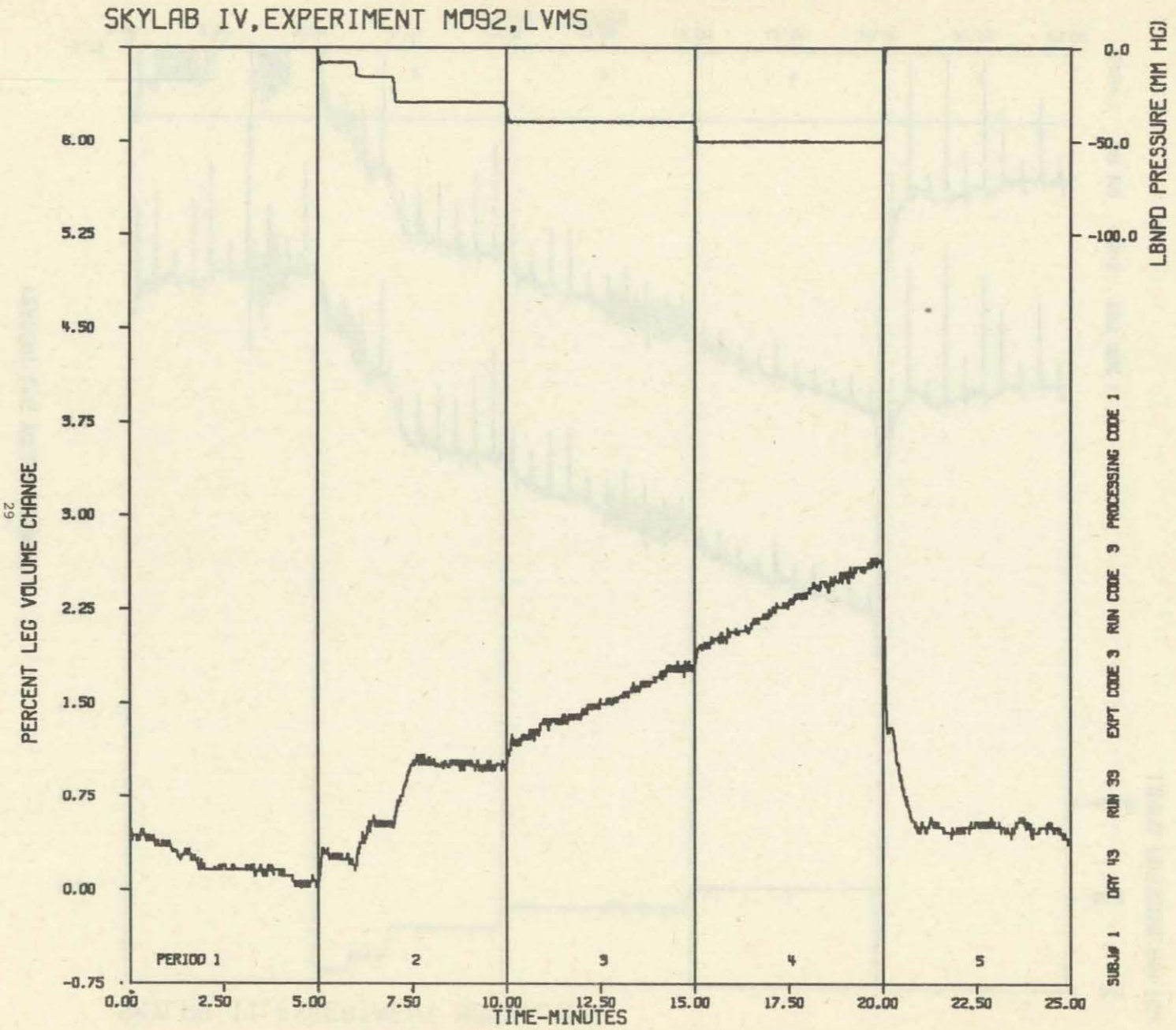


FIG. 18 FINAL PROCESSED PLVC DATA

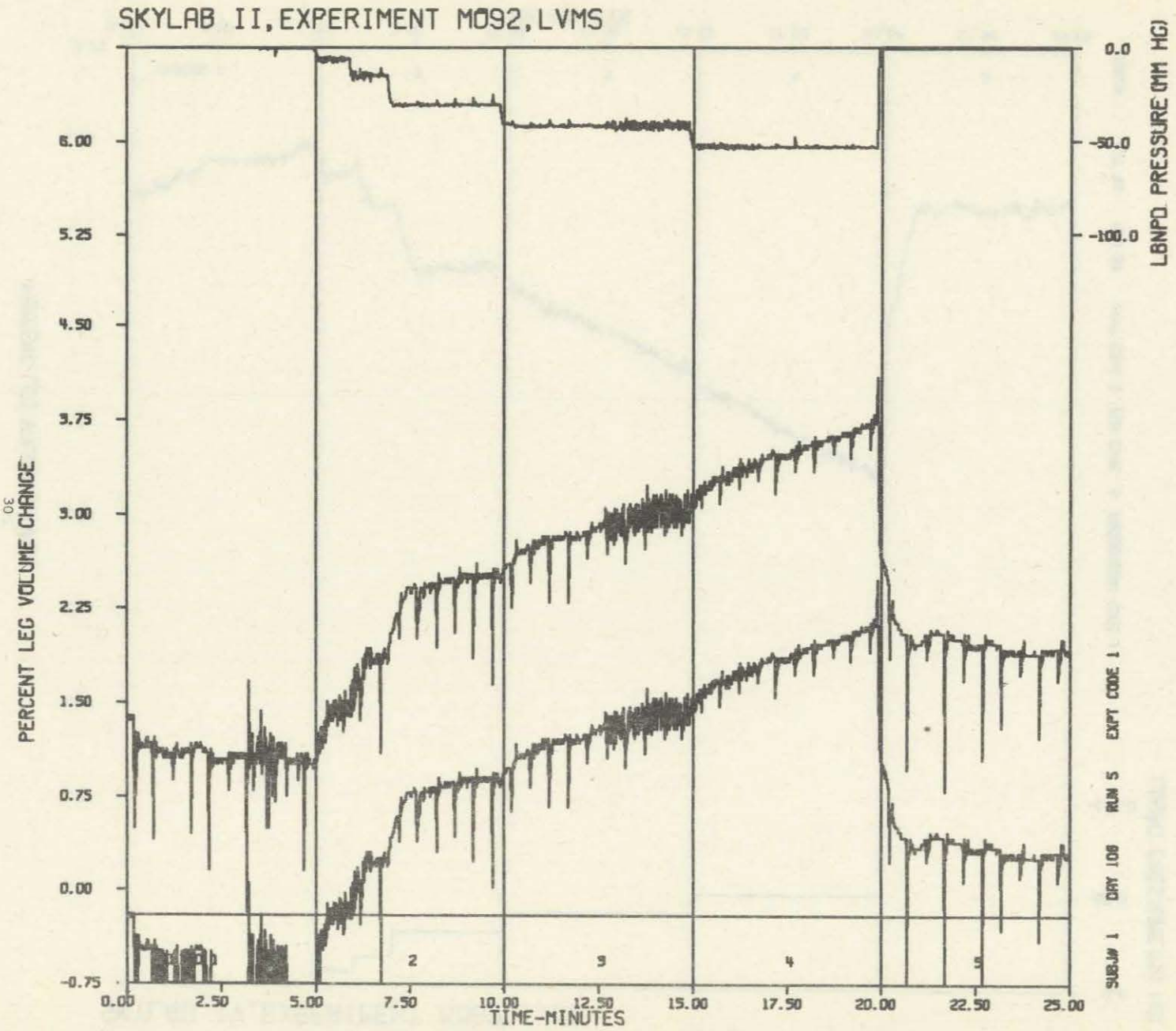


FIG. 19 PLOT OF DATA WITH NOISE AND ELECTRICAL SPIKES

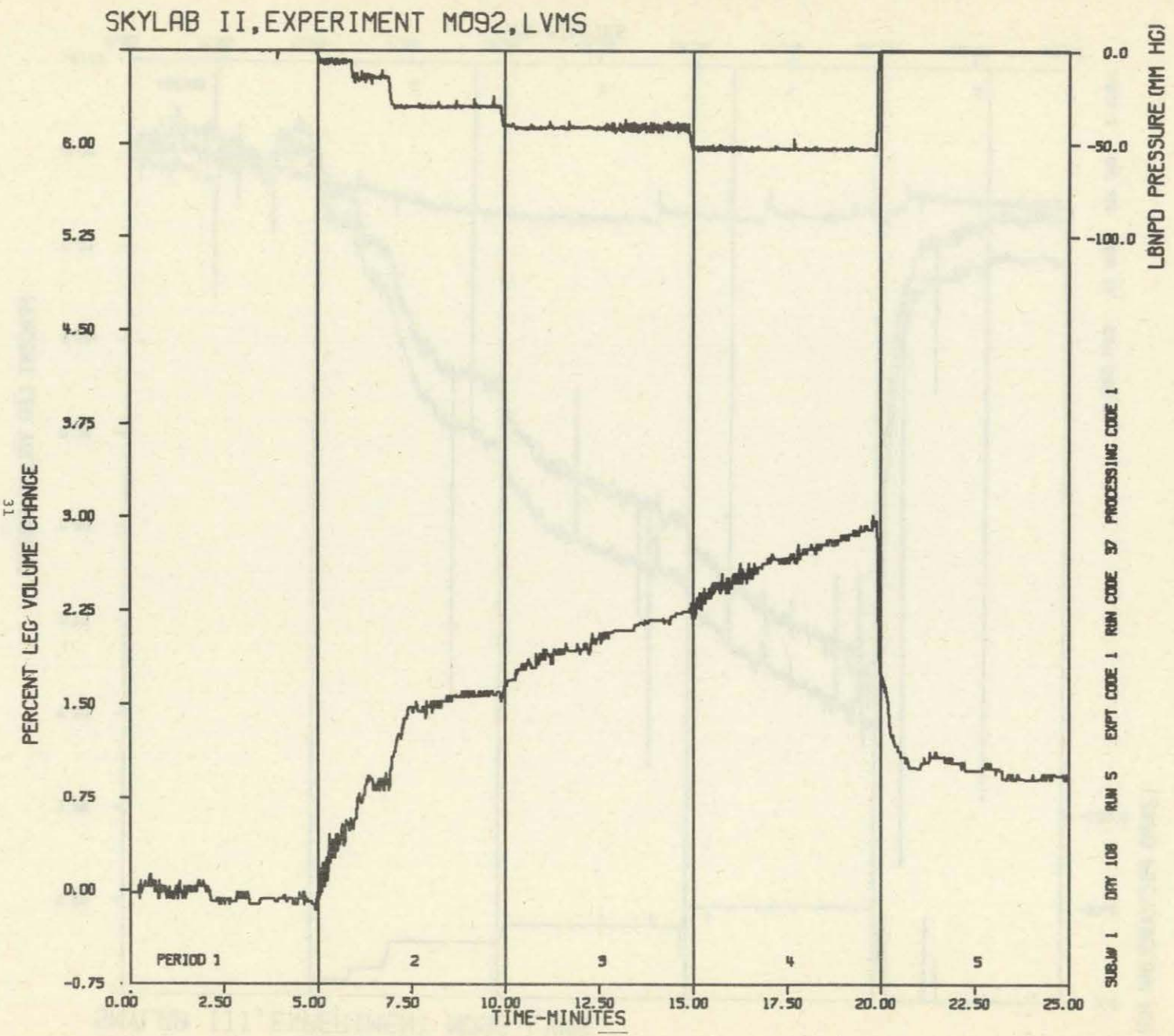


FIG. 20 FINAL PROCESSED DATA

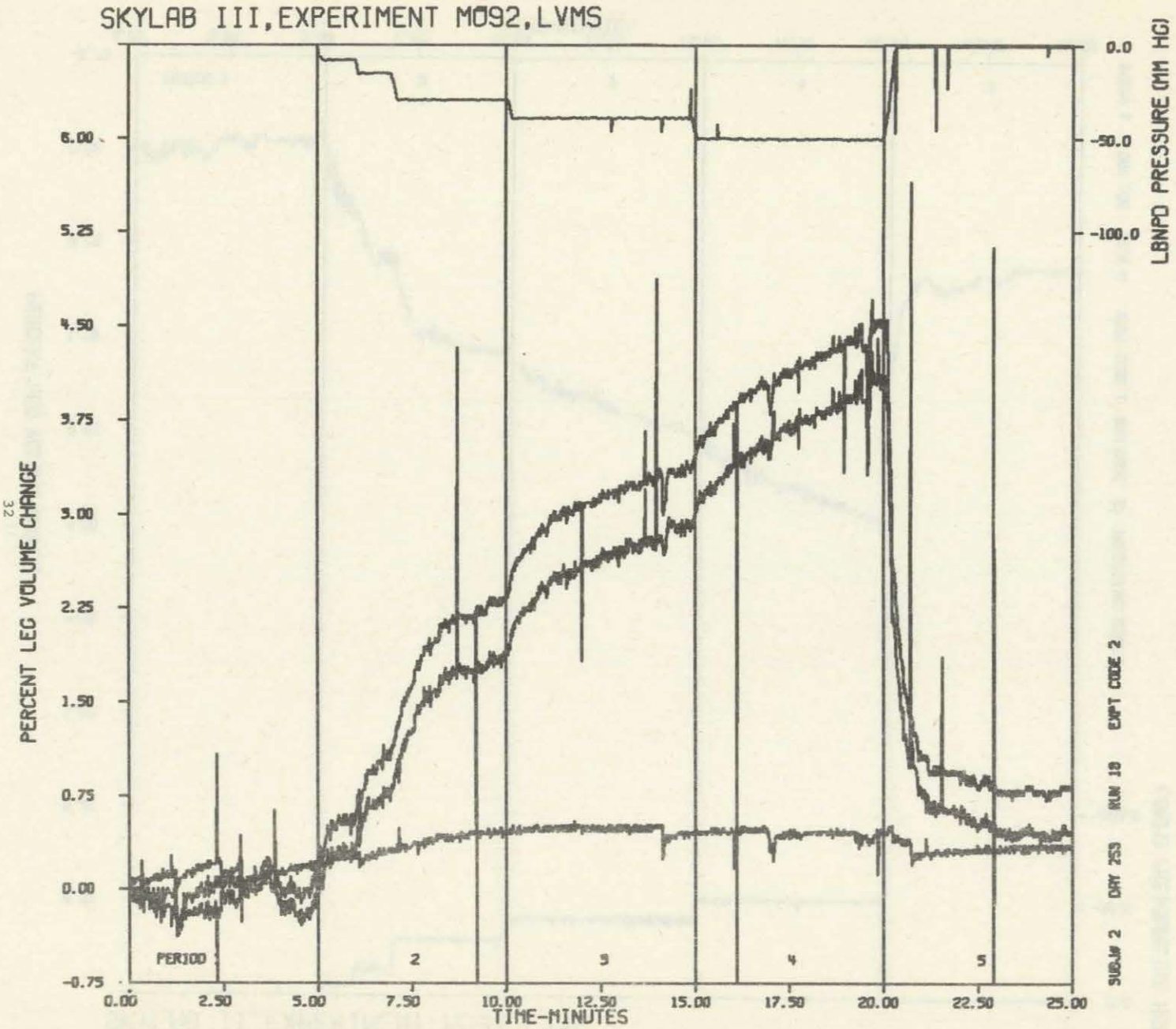


FIG. 21 PLOT OF DATA WITH ELECTRICAL SPIKES

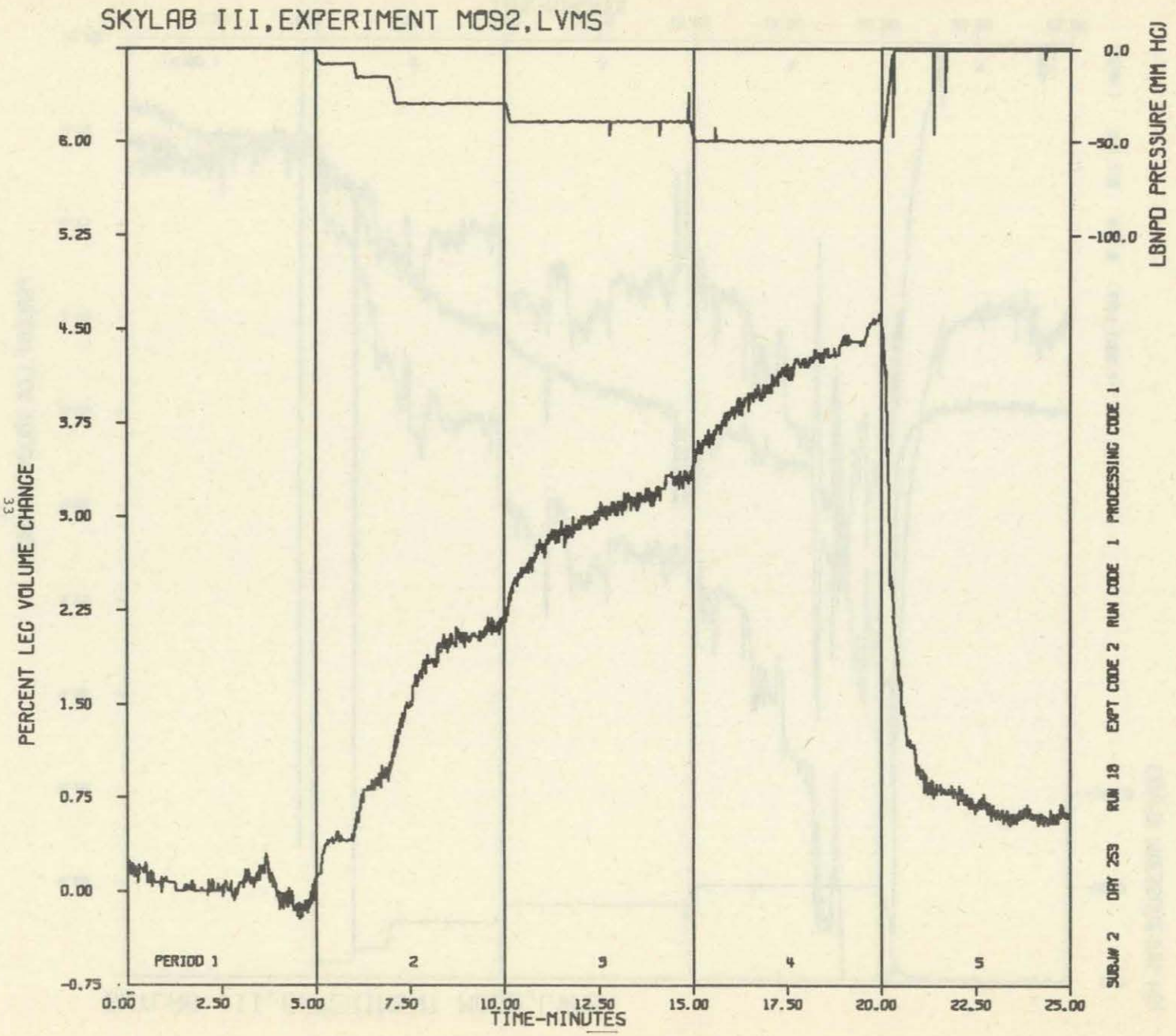


FIG. 22 FINAL PROCESSED DATA

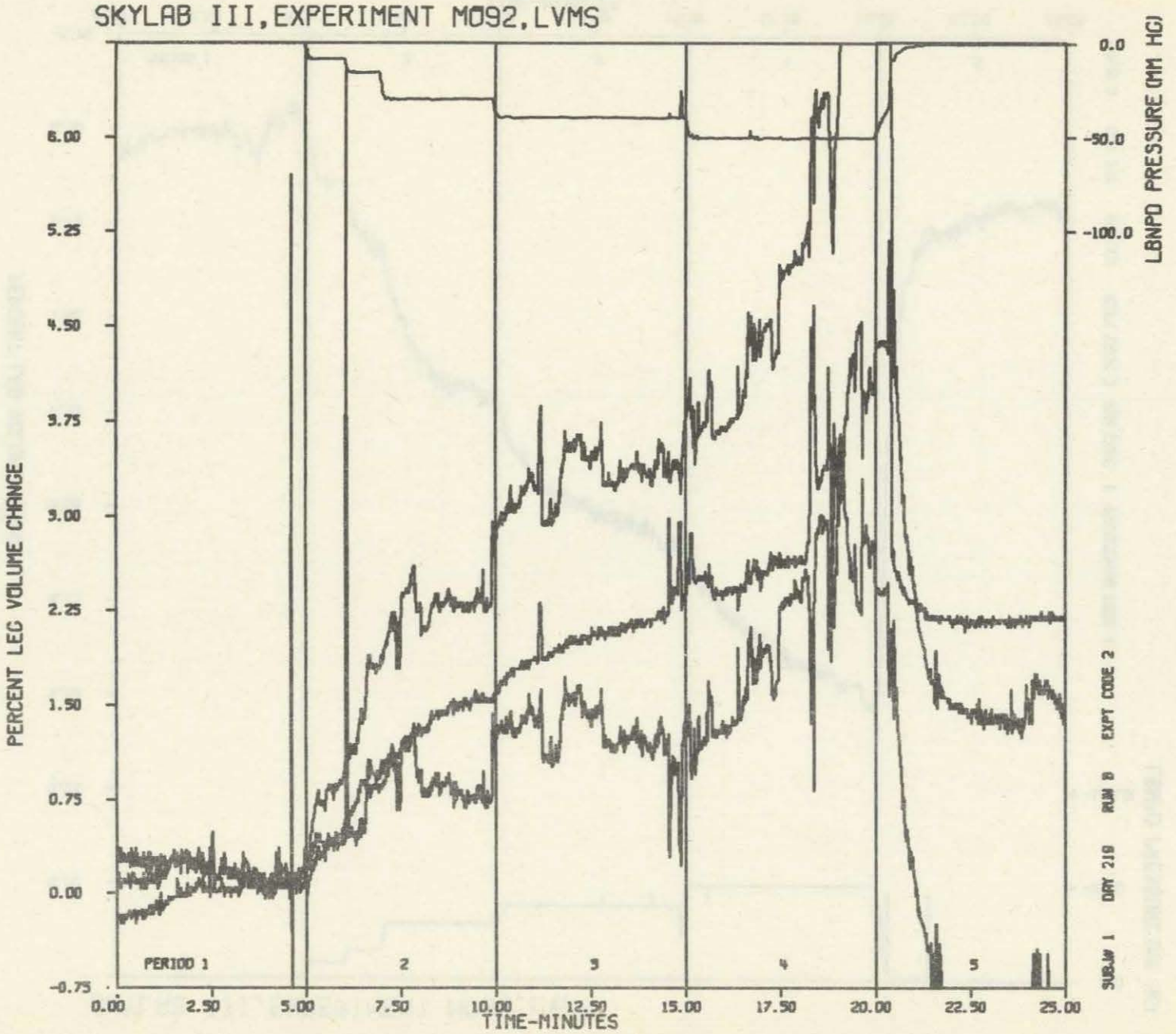


FIG. 23 PLOT OF DATA BADLY CONTAMINATED WITH NOISE, SPIKES AND OFFSETS

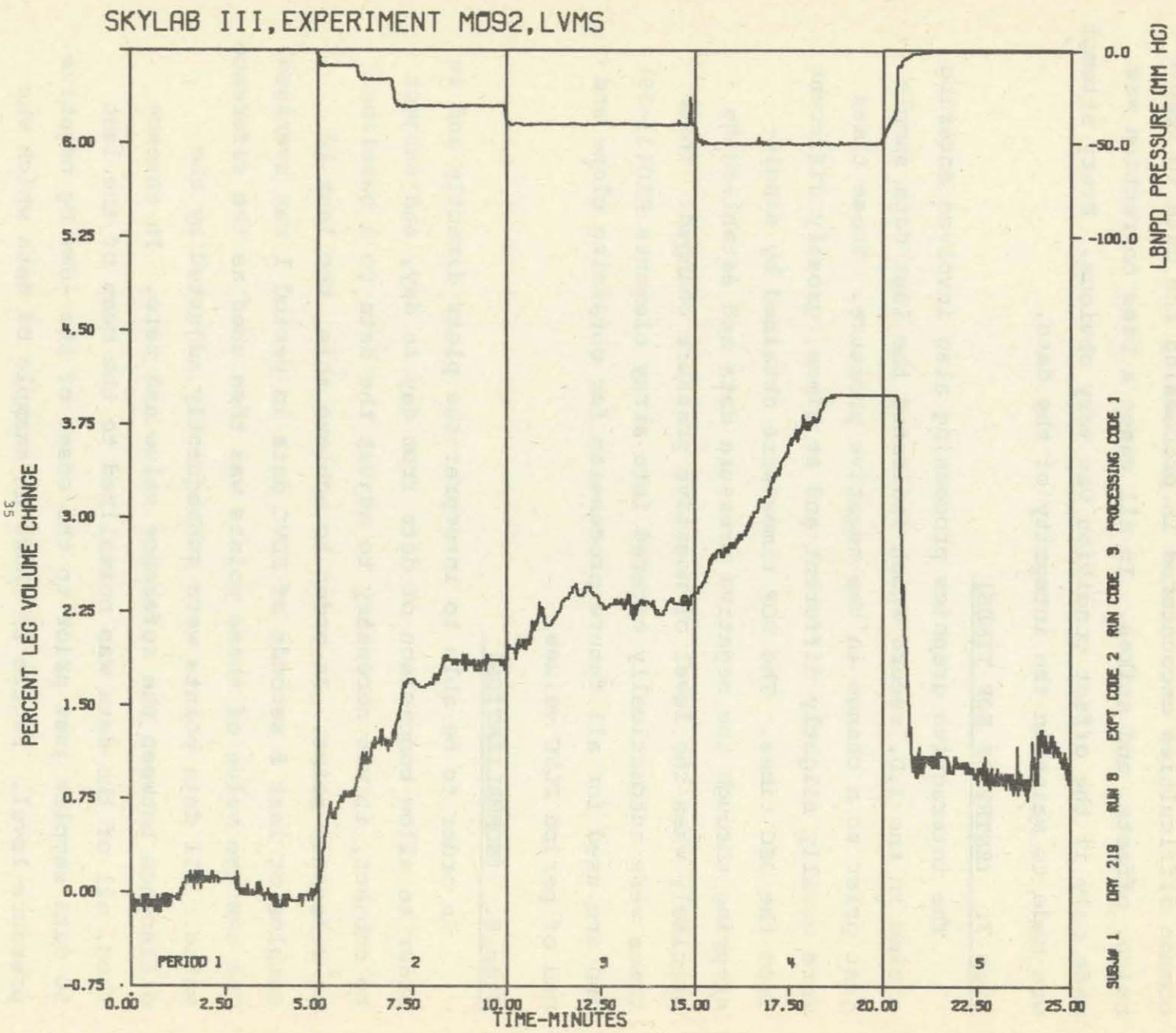


FIG. 24 FINAL PROCESSED DATA

The graphs of Figures 11-24 illustrate some of the more common difficulties encountered in processing the data to remove noise, offsets, and spikes. In all cases a bias correction was made only if the offset condition was very obvious. Every attempt was made to maintain the integrity of the data.

Step 7. (ENTRY OF EOP TIMES)

The interactive graphics processing also involved entering a time in the I.D. record which indicated the last data sample just prior to a change in the negative pressure. These times were usually slightly different and at times, grossly different from the JSC times. The EOP times were obtained by single-stepping through the negative pressure data and establishing precisely when the level of negative pressure changed. These times were automatically entered into array elements RID(33-39) and are used for all future processing for obtaining slope and end of period PLVC values.

Step 8. (NORMALIZATION)

In order to be able to interpret the plots directly and in order to allow comparison of data from day to day, and subject to subject, it was necessary to adjust the data to a baseline or reference point. In order to achieve this, the last 10 samples or last 8 seconds of PLVC data in period 1 was averaged. The average value of these points was then used as the reference value. All data points were subsequently adjusted by the difference between the reference value and zero. In essence then, all of the data was normalized to the mean of the last 10 data samples just prior to the onset of the -8mm Hg negative pressure level. Figure 25 shows an example of data which was far below the zero percent starting point and the final normalized data is shown in Figure 26.

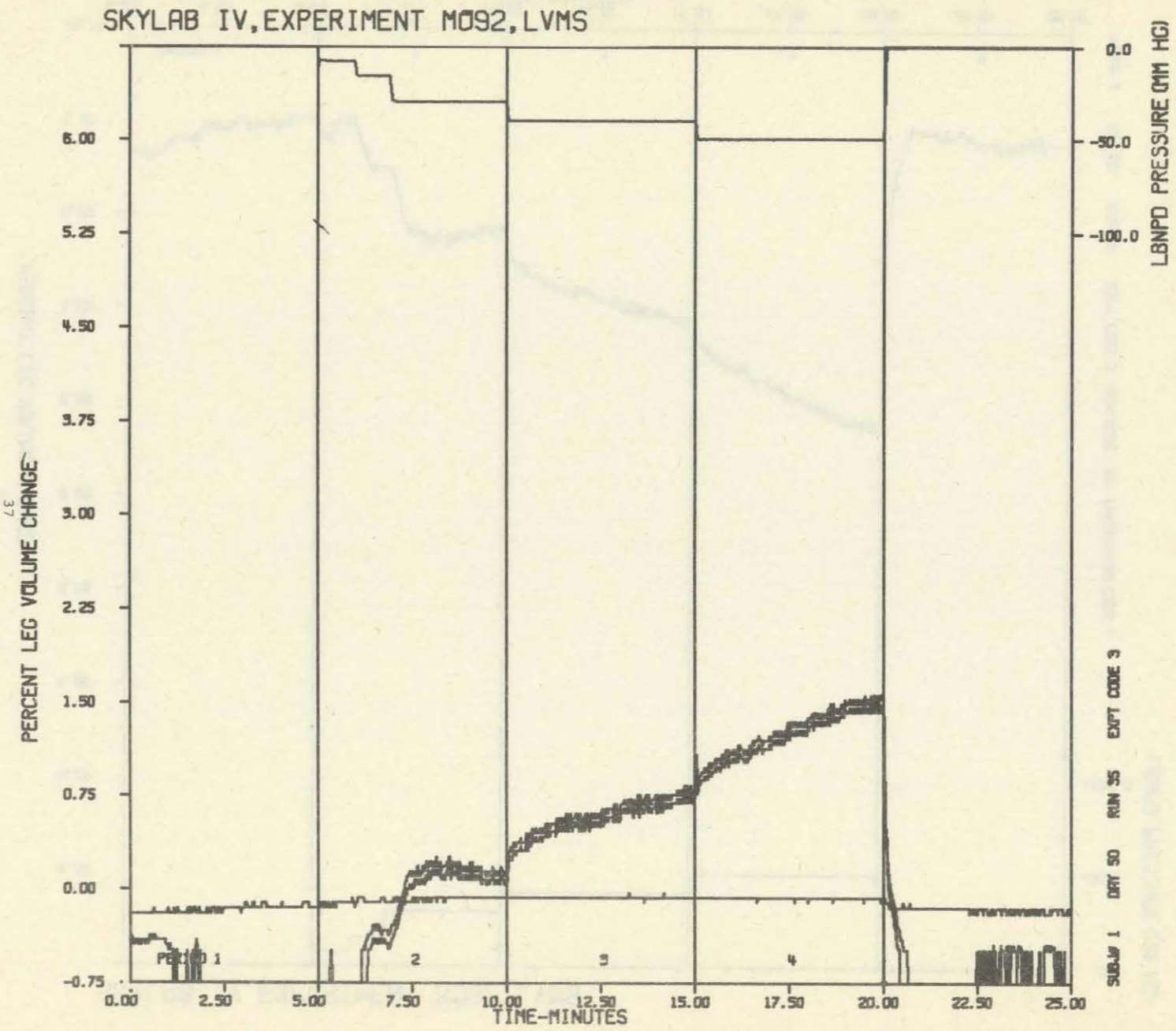


FIG. 25 PLOT OF UNNORMALIZED DATA

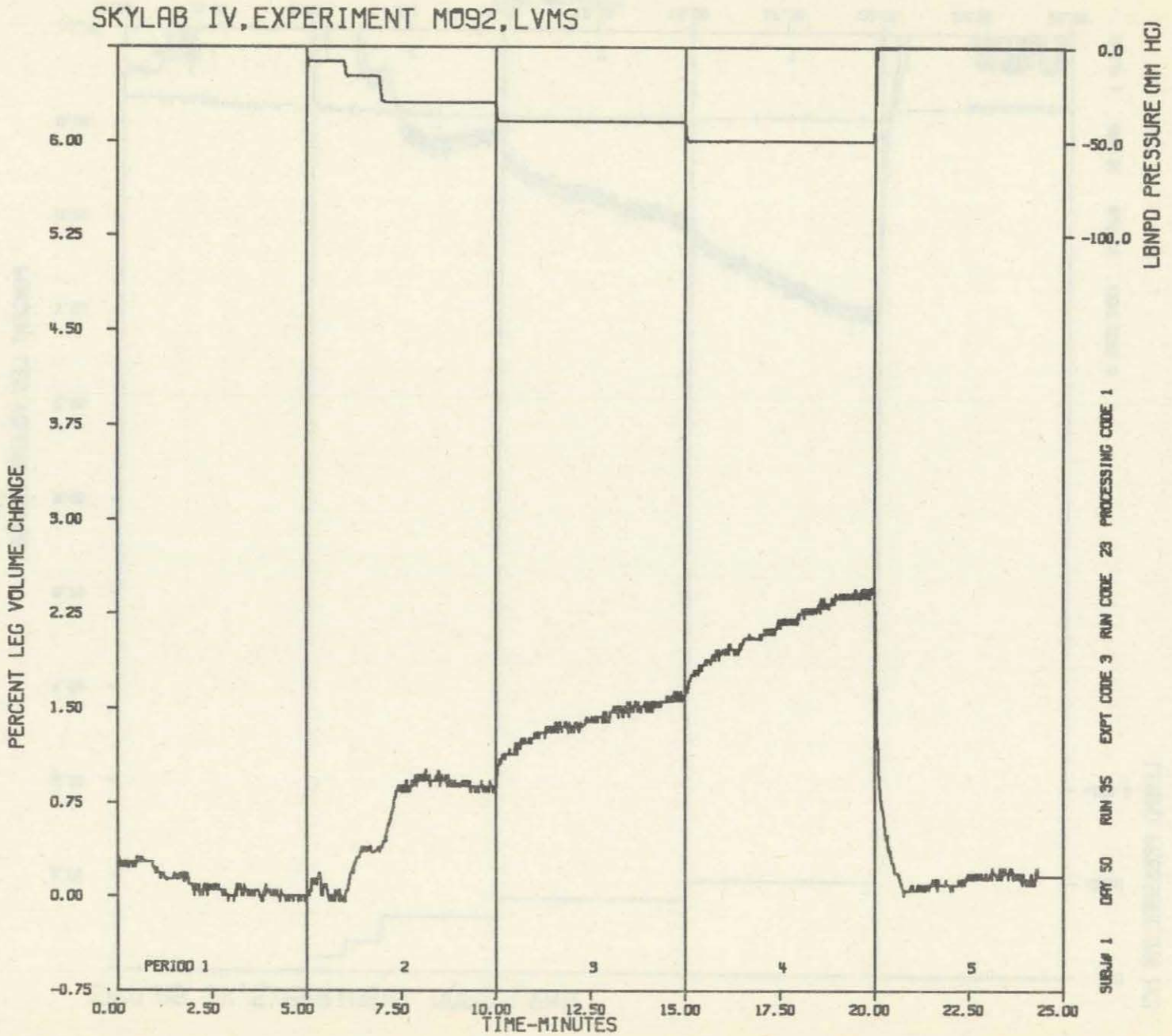


FIG. 26 PLOT OF NORMALIZED DATA

Step 9. (FOAM CORRECTION)

A foam correction factor was applied to the normalized percent leg volume change to account for the minor difference in leg volume resulting from the back pressure of the foam of the leg band upon the small vessels at the surface of the calf. The foam correction factor was determined from physiological tests comparing data taken with Whitney Guages and Capacitive Plethysmographs. The foam correction factor is a data transform using the best fit second order polynomial equation obtained experimentally for each band size. The equation used for the processing of all Skylab data was $Y=.001+.90X-.002X^2$. The Y value was the output for the capacitive leg bands and the X represented the Whitney output. This foam correction factor enables the leg band calibration derived from the calibration cylinders to reflect the soft tissue-foam compression interaction on the skin of the calf.

Step 10. (SAVE PROCESSED DATA)

After the processed data was completely reviewed to insure that all corrections had been made, that file was assigned a "RUN type" descriptor in element 40 of the KID array. Appendix C contains the descriptors used to identify the characteristics of a particular file. For example, if that file had missing data or a presyncope occurred then it's run code would be a 2 or 4. Multiple codes were used routinely. The data was then saved on magnetic tape.

Step 11. (PLOTTING OF THE PROCESSED DATA)

This step involved plotting the processed data on a large (10"x 10") graph to insure that no incorrect data manipulations had been performed. These plots were permanently filed with the original data plots for comparison. A copy of the program to generate these plots as well as sample plot is illustrated in Appendix J and Appendix K.

Steps 12 & 13 (SEQUENCING DATA & SHIPMENT to JSC)

After all data had been verified, each astronaut's data was sequenced by run number and copied onto a magnetic tape for shipment to JSC. These tapes were labled according to the tape numbering system identified in Appendix F. This processed data and I.D. information was then copied to the permanent data storage files at JSC.

Steps 14 & 15 (PLOTS OF COMPOSITE DATA)

The six parameter tapes returned from JSC contained the corrected LVMS data and in addition the M092 heart rate, systolic and diastolic blood pressure data. These data were displayed as composite data plots as shown in Fig. 27. This type of plot illustrates more completely the interaction between the cardiovascular parameters.

Steps 16 and 17 (PLOTS AND PRINTOUT of EOP and SLOPES)

The program listed in Appendix N was used to compute end of period values for the seven periods within the twenty-five minute protocol. End of period values were computed by obtaining the average of the last 10 PLVC data points just prior to the end of each period. Slopes of the rapid filling portions of the leg volume curves (S1) as well as the slopes of the infiltration (slowly changing slopes toward the ends of the negative pressure periods- S2) were computed using this program. The duration of time for slope computation could be selected as a program parameter. These values could then be displayed in printout or as Cal Comp plots.

Step 18 (3-D CAL COMP PLOTS)

The use of three-dimensional plots was cursorily investigated as a technique for summation displays of each astronaut's mission data. This type of display, using the normalized data, presents a complete picture of the time course variation in leg volume changes during the mission as well as variation from preflight

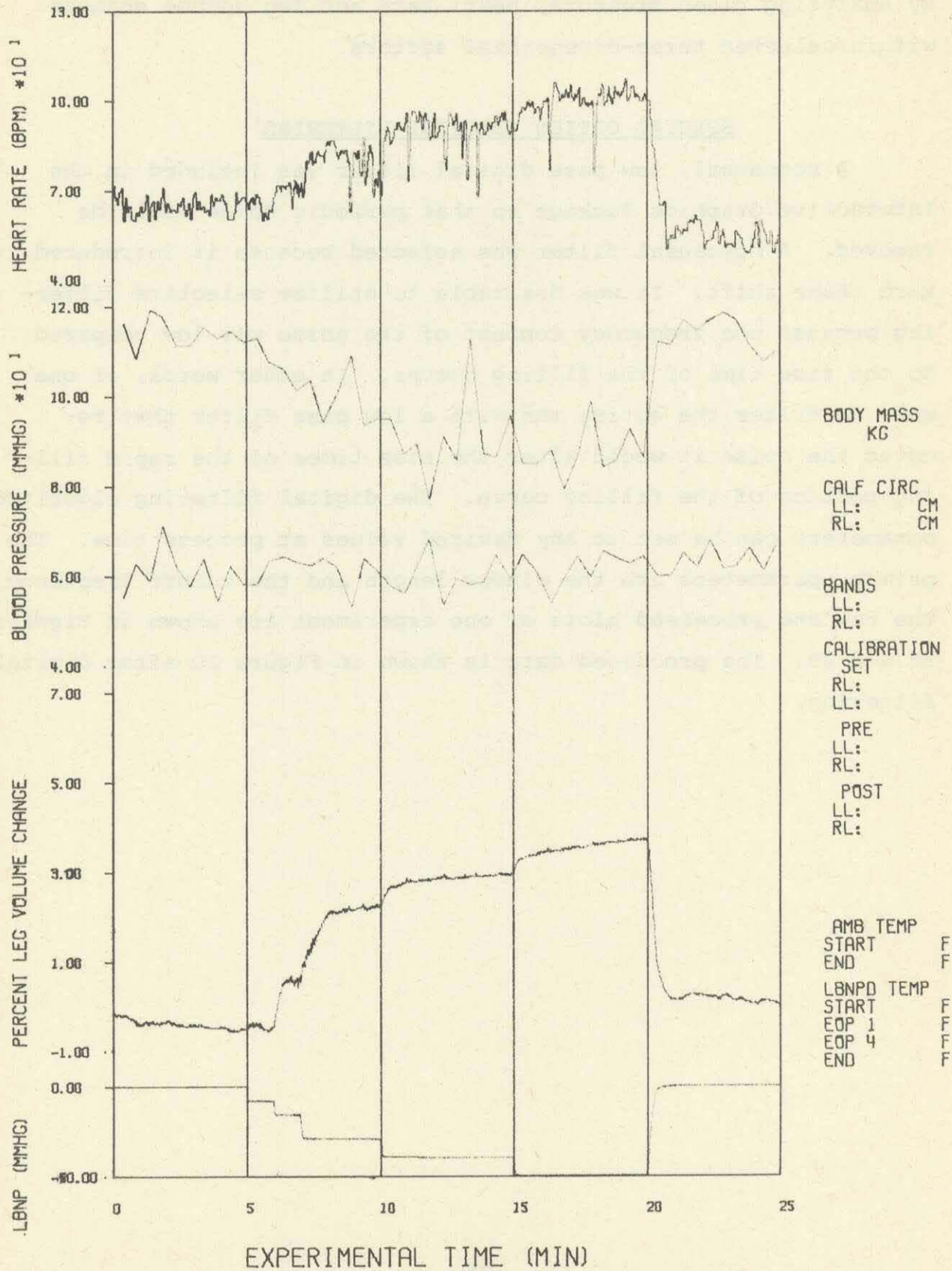


FIG. 27 PLOT COMPOSITE DATA

and postflight data. This type of capability can be extended to the study of the interaction of the cardiovascular parameters by analyzing blood pressure, heart rate and leg volume changes within selected three-dimensional sectors.

SPECIAL OPTION (DIGITAL FILTERING)

A noncausal, low pass digital filter was included in the Interactive Graphics Package so that periodic noise could be removed. A noncausal filter was selected because it introduced zero phase shift. It was desirable to utilize selective filtering because the frequency content of the noise was low compared to the rise time of the filling curves. In other words, if one were to filter the entire run with a low pass filter that removed the noise it would alter the rise times of the rapid filling portion of the filling curve. The digital filtering algorithm parameters can be set to any desired values at process time. The primary parameters are the window length and the cutoff frequency. The raw and processed plots of one experiment are shown in Figures 28 and 29. The processed data is shown in Figure 30 after digital filtering.

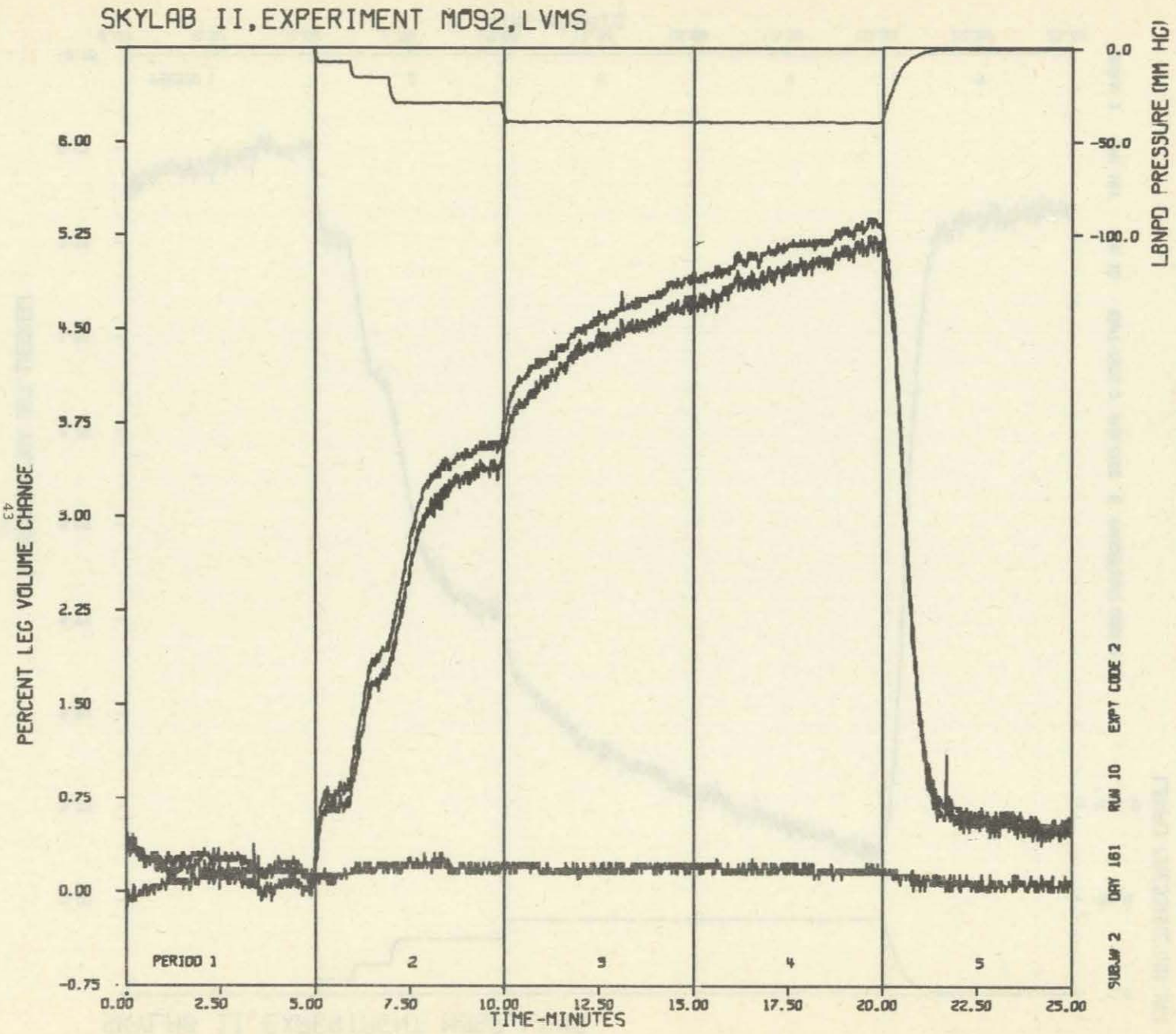


FIG. 28 PLOT OF UNPROCESSED DATA

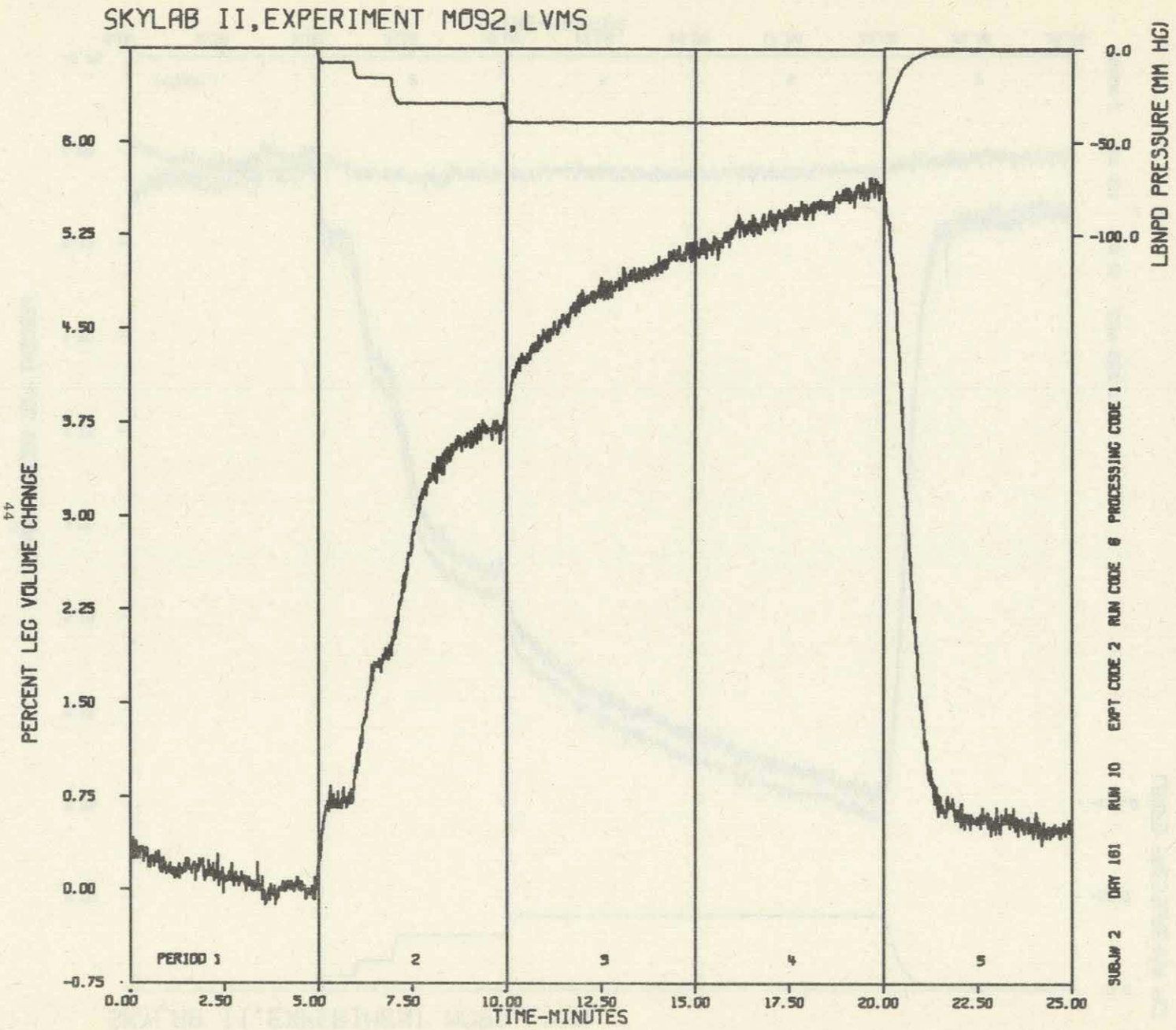


FIG. 29 PLOT OF PROCESSED DATA

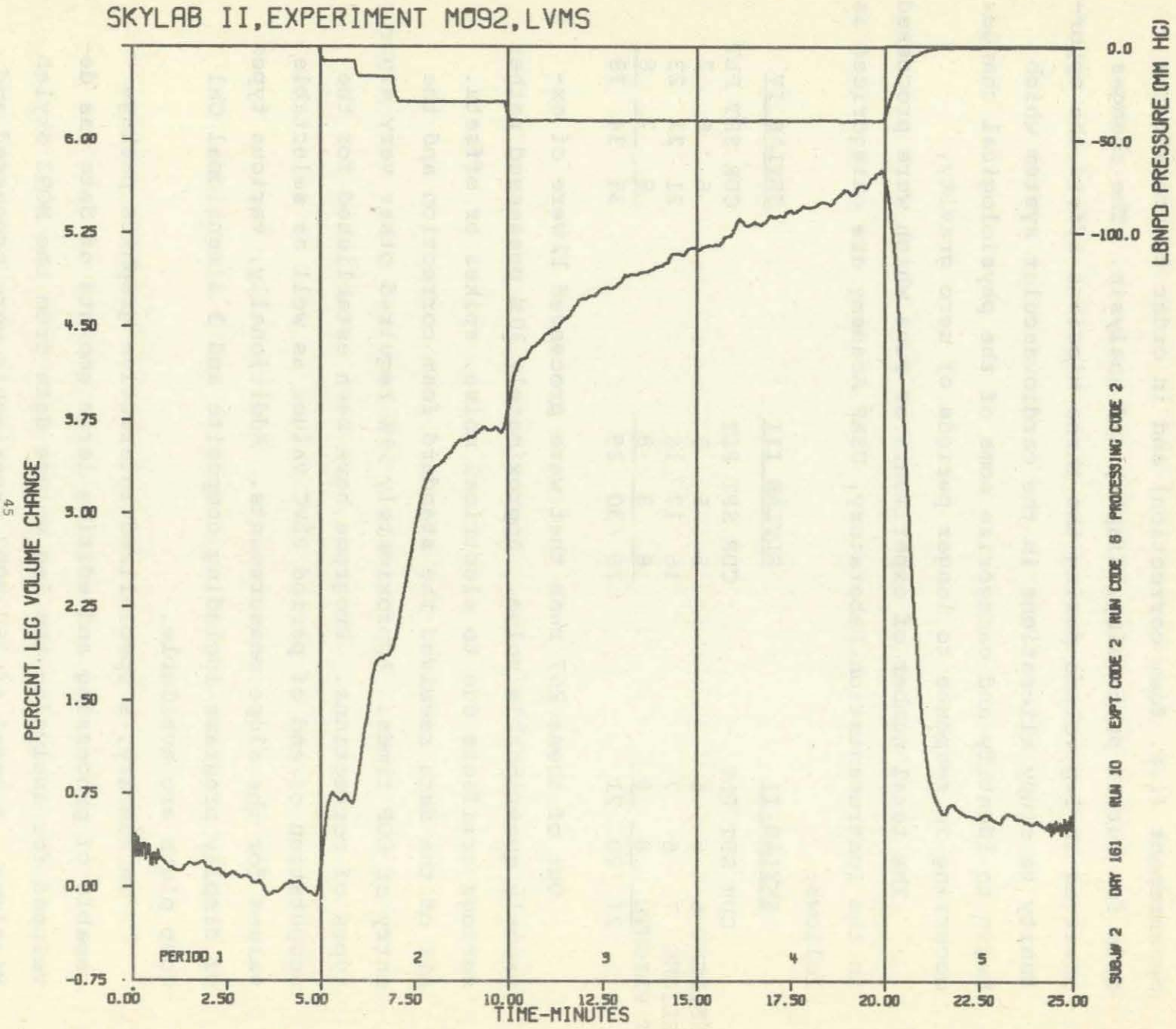


FIG. 30 PLOT OF DATA AFTER DIGITAL FILTERING

SUMMARY

The processing of the Skylab M092 leg volume data performed at the USAFA was necessary in order to complete the measurement (i.e. foam correction) and in order to prepare the data for future processing, display and analysis. The changes observed in leg volume during the three missions afford the opportunity to study alterations in the cardiovascular system which begin to identify and categorize some of the physiological changes occurring in response to longer periods of zero gravity.

The total number of experiments or runs which were processed in the Instrumentation Laboratory, USAF Academy are categorized as follows;

	SKYLAB II			SLYLAB III			SKYLAB IV		
	CDR	SPT	PLT	CDR	SPT	PLT	CDR	SPT	PLT
PREFLIGHT	6	6	6	5	5	5	6	6	7
FLIGHT	7	6	7	16	17	16	21	21	23
POST FLIGHT	8	8	8	8	8	8	8	7	8
	21	20	21	29	30	29	35	34	38

Out of these 257 runs that were processed 12 were of extremely questionable value. Approximately 30% possessed rather serious artifacts due to electrical noise, spikes or offsets. All of the data received the standard foam correction and the entry of EOP times. Approximately 55% required other very minor types of corrections. Programs have been established for the computation of end of period PLVC values as well as selectable values for the slope measurements. Additionally, various types of display programs including composite and 3 dimensional Cal Comp plots are available.

In summary, a specialized interactive graphics package capable of processing and editing large amounts of data was developed for analyzing the leg volume data from the M092 Skylab Missions. A total of 257 M092 experiments were processed and prepared for analysis using this system.

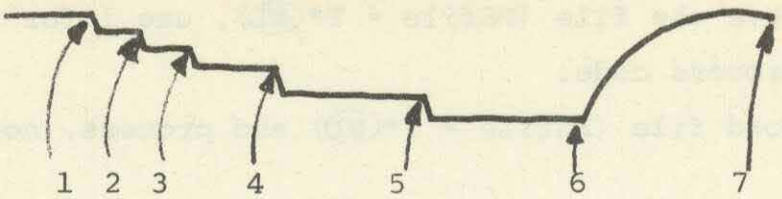
Appendix A	Operator instructions for using the Interactive Graphics Processor	48
Appendix B	Program for Interactive Graphics Processor	52
Appendix C	Descriptors for run type code	62
Appendix D	Interactive Graphics Flow Chart	63
Appendix E	Interactive Graphics Package Controls	64
Appendix F	Format for numbering data on magnetic tapes	65
Appendix G	Program for plotting unprocessed data from JSC.	66
Appendix H	Plot of unprocessed data	69
Appendix I	Program for merging supplementary I.D. information	70
Appendix J	Program for plotting USAF processed data	71
Appendix K	Plot of USAF processed data	74
Appendix L	Program for printout of I.D. and/or experimental data	75
Appendix M	Sample of printout program outputs	79
Appendix N	Program for Printout of EOP and Slope Information	81

APPENDIX A

PROCEDURE FOR PROCESSING MO92 LVMS DATA
USING THE INTERACTIVE GRAPHICS TERMINAL

1. Run RADMAP Program.
2. Delete & allot appropriate files. Use D1 area, Files F1-F11.
3. Load data to be processed into RAD (Files 1-5).
4. Load Graphics program (Program Can't be Reloaded by Commands.)
5. Insure that the Skylab patch panel (Panel 87) is in place.
6. Adjust gain on the scan converter so that the four reference dots can be seen at the corners of the monitor.
7. Input on the tty,
   A) interrupt
   B) type in T NL
   Start here for processing 2nd, 3rd, 4th, or 5th File.
   C) interrupt
   D) type in FNFILE= T*NL
   } C & D not necessary for first file after loading program
8. Enter File # NL.
9. Enter T or F (Wild Point Editing).
   A) will always be T unless you want to substitute left leg for PLVC in the data array. See paragraph 13 for substitution procedure.
10. Observe output to determine if correct day is being processed.
11. Should you desire to look at any other run - just,
   A) interrupt
   B) type in FNFILE = T* NL
   C) type in file # NL
12. Begin Processing
   A) make offset and minor corrections
   B) make bias corrections
   C) check PLVC for final appearance
   D) with LBNP on display, enter EOP values

a) Bring cursor to the last data point in that period.



b) Push EOP button.

c) Enter Period Number

E) Zero Data

a) Display PLVC

b) Bring cursor to EOP 1

c) observe value, push Zero button and observe shift in PLVC Value

F) Foam Correction

a) display PLVC

b) move cursor to EOP 6(-50)

c) observe value

d) interrupt

e) type in FOAM =T* (NL)

f) observe new value, it should be ~ 10% Higher

G) Save File

a) interrupt

b) type in FSFILE = T* (NL)

c) type in File No., run and process codes

d) record in processing log the information in Par. G.

H) Copy to Mag. Tape

a) copy on appropriate tape the data in files 6-10 (800BPI)

b) record in processing log

I) Cal Comp processed data

13) To input LL data into PLVC array

A) load data in normal manner (FNfile =T* (NL)

B) interrupt

C) type in FLLEG = T*(NL)

D) save the file (FSfile = T*(NL)), use 1 for run & process code.

E) Load file (FNfile = T*(NL)) and process, normally

SUMMARY OF OPERATIONS

STEP	OPERATION
1.	Load files in RAD
2.	Bring up Graphics Program
3.	Load file for processing
4.	Make offset & minor corrections
5.	Make bias corrections
6.	Check curve carefully
7.	Enter EOP values
8.	Zero the data
9.	Make foam correction
10.	Save the file using proper run & process code
11.	Copy files (6-10) to mag tape when finished.

PRIMARY CONTROL & DATA VARIABLES

VARIABLE NAME +	DEFINITION
TDATA (II, JJ)	MAIN DATA ARRAY -
JJ	Time index set by the X-axis value of the cursor
II = 1	Time
II = 2	Left Leg
II = 3	PLVC
II = 4	Right Leg
II = 5	LBNP
ISTART	Time index of the first datum on the display
CURSOR	Current Y-axis value of the cursor
FNFILE	Load a new file
FSFILE	Save the current file
FSTAT	Printout the values of II, JJ, TDATA (II, JJ), ISTART, CURSOR
FSTART	Input a new starting value
FILTERD	Design a new filter
FWTS	Printout the new filter values
FBUG	Printout the input bits that are equal to one
FEND	Release program
FRLS	Release program and return control to analog lab
NPTS	Number of points used by the wild point editor
WILDPT	Number of standard alleviations that defines a wild point
FOAM	Foam correct the PLVC data

+ Variable names that begin with F are logical variables. Setting a logical control variable equal to TRUE invokes the action indicated.

APPENDIX B

PROGRAM FOR INTERACTIVE GRAPHICS PROCESSOR

```
C*****M0-92 INTERACTIVE GRAPHICS PACKAGE
C*****MAIN PROGRAM
      IMPLICIT LOGICAL (F)
      LOGICAL SSW
      LOGICAL UNPACK
      DIMENSION ALPHA(2),WTS(0:200)
      DIMENSION RDATA(5,75),RID(100),ID(10),TID(7),KID(100)
      DIMENSION TDATA(5,1875)
      EQUIVALENCE (KID,RID)
      DATA F1,F2,F3,F4,FWILD,FILL/6*.FALSE./
      DATA FZERO,FILTERD,FWTS,FOLD,FCHNG/5*.FALSE./
      DATA DFILL/8Z407FFFFF/,PI/3.14159/
      NAME LIST
      DATA FNFILE,FSFILE,FSTAT,FOAM/4*.FALSE./
      DATA XSLEWS,XSLEWF,YSLEWS,YSLEWF,YSLEWNP,YSLEWNPF,STEP,STEPNP
     * /1.0,9.0,0.05,0.2,1.0,5.0,0.02,0.5/
      DATA FEND,FLAG,FBUG,FSTART/4*.FALSE./
      DATA FSCALE,FXSLEW,FYSLEW,FRLS/4*.FALSE./
      CONNECT (2Z68,MONITOR)
      CONNECT (2Z6A,SCHEDLR)
      OUTPUT(6) 'INSTRUMENTATION LAB INTERACTIVE GRAPHICS PACKAGE'
      OUTPUT(6) 'DO NOT DISTURB----UNTIL THIS PROGRAM IS RELEASED'
      OUTPUT(8) 'M0-92 INTERACTIVE GRAPHICS PACKAGE'
      IDS=42
      FNFILE=.TRUE.
      FLLEG=.FALSE.
      WILDPT=2.0
      II=3; JJ=1
      CURSOR=0.0; ISTART=1
      NPTS=101
      IFLT=50
      CUTF=0.1
      SF=CUTF*0.8*PI
      WTS(0)=CUTF; SUM=WTS(0)
      DO 1 I=1,IFLT
      WTS(I)=SIN(SF*I)/(PI*I)
1     SUM=SUM+2.0*WTS(I)
      DO 2 I=0,IFLT
2     WTS(I)=WTS(I)/SUM
      CALL OCTYA02
      ION=1
      IF (SSW(1)) ION=0; CALL OCTYA01
      OUTPUT(8) 'INPUT  ION=T   WHEN THE SCAN CONV IS SET UP'
      CALL SETUP(ION,0)
      CALL SSCALE(1.0,30.0,0,30)
      CALL STARTCLK(800,2Z6A)
      CALL REXIT
C*****
C*************************************************************************
C*****
```

```
      SUBROUTINE MONITOR
C***** CHECK FOR START-UP PHASE
      IF(ION.EQ.0) GO TO 5
      OUTPUT(8) 'SCAN CONV RDY(T OR F)?'
      INPUT(7) FION; IF (FION) ION=0
      RETURN
5     IF (FCHNG) GO TO 8
      OUTPUT(8) 'COMMAND=?'
      INPUT(7)
8     CONTINUE
C***** INPUT NEW FILE?
      IF (.NOT.FNFILE) GO TO 150
      OUTPUT(8) 'FILE NUMBER=?'
      INPUT(7) NFILE
      IFILE=NFILE+10
      NREC=26
      REWIND IFILE
      CALL BUFFER IN(IFILE,1,KID,IDS,ISTAT,ICNT); ERR=0
10    GO TO (10,11,40,50) ISTAT
11    CONTINUE
      IF (ICNT.NE.IDS) GO TO 60
      CALL STOPCLK
      REWIND 21
      CALL BUFFER OUT(21,1,KID,IDS,ISTAT,ICNT); ERR=1
13    GO TO (13,15,40,55) ICNT
15    IF (ICNT.NE.IDS) GO TO 60
C*****UNPACK INTEGER DATA
      DO 20 I=1,5
      ID(2*I-1)=ISA(KID(I),-16)
      ID(2*I)=ISL(KID(I),16)
20    ID(2*I)=ISA(ID(2*I),-16)
      DO 30 I=1,7
30    TID(I)=RID(I+5)
      N=1
      DO 120 IREC=2,NREC
      CALL BUFFER IN(IFILE,1,TDATA(1,N),375,ISTAT,ICNT); ERR=IREC
35    GO TO (35,70,37,50) ISTAT
37    OUTPUT(8) 'EOF ENCOUNTERED',IREC;NREC=IREC-1
C***** INPUTS ZERO FOR SHORT DATA RECORDS
      DO 38 J=N,1875
38    TDATA(1,J)=TDATA(2,J)=TDATA(3,J)=TDATA(4,J)=TDATA(5,J)=0.0
      GO TO 121
40    OUTPUT(8) 'EOF ENCOUNTERED'; GO TO 65
50    OUTPUT(8) 'INPUT ERROR'; GO TO 65
55    OUTPUT(8) 'OUTPUT ERROR'; GO TO 65
60    OUTPUT(8) 'COUNT ERROR'
65    OUTPUT(8) 'IRRECOVERABLE BUFFER IN/OUT ERROR'
      OUTPUT(8) ERR,IFILE,JFILE,ISTAT,ICNT
      CALL STOPCLK; CALL RLSBARN
70    IF (ICNT.NE.375) GO TO 60
      CALL BUFFER OUT(21,1,TDATA(1,N),375,ISTAT,ICNT); ERR=IREC+30
80    GO TO  (80,90,40,55) ISTAT
90    IF (ICNT.NE.375) GO TO 60
```

```
104.          N=N+75
105.        120 CONTINUE
106.      121   REWIND IFILE,21
107.          WRITE(8,122)
108.      122   FORMAT(T2,'RUN#',T13,'SUBJ#',T28,'DAY',T34,'EXPT CODE')
109.          WRITE(8,125) ID(2),ID(3),ID(4),ID(9)
110.      125   FORMAT(5(I5,7X))
111.          FLAG=.TRUE.
112.          FNFILE=.FALSE.
113.          CALL STARTCLK(800,2Z6A)
114.      C*****REMOVE FILL POINTS
115.          LCNT=0
116.          DO 130 I=2,5
117.          IF (TDATA(I,1).EQ.DFILL) WRITE(8,126) I; INPUT(7) TDATA(I,1)
118.      126   FORMAT (' FIRST POINT IN PARAMETER #',I2,'IS BAD INPUT NEW VALUE')
119.          DO 128 J=1,1875
120.          IF (TDATA(I,J).EQ.DFILL) GO TO 131
121.      128   CONTINUE
122.      130   CONTINUE
123.          GO TO 138
124.      131   IF (J.NE.1875) GO TO 132
125.          TDATA(I,J)=TDATA(I,J-1)
126.          LCNT=LCNT+1; IF (FILL) OUTPUT(8) I,TDATA(1,J)
127.          GO TO 128
128.      132   DO 133 L=J,1875
129.      133   IF (TDATA(I,L).NE.DFILL) GO TO 134
130.      134   IF (L.NE.1875) GO TO 136
131.          DO 135 L=J,1875
132.          LCNT=LCNT+1; IF (FILL) OUTPUT(8) I,TDATA(1,L)
133.      135   TDATA(I,L)=0.0
134.          GO TO 130
135.      136   SLOPE=(TDATA(I,L)-TDATA(I,J-1))/(TDATA(1,L)-TDATA(1,J-1))
136.          DO 137 LL=J,L-1
137.          LCNT=LCNT+1; IF (FILL) OUTPUT(8) I,TDATA(1,LL)
138.      137   TDATA(I,LL)=SLOPE*(TDATA(1,LL)-TDATA(1,J-1))+TDATA(I,J-1)
139.          J=L
140.          GO TO 128
141.      138   WRITE (8,139) LCNT
142.      139   FORMAT (2X,I4,' FILL POINTS WERE REMOVED')
143.      C***** REMOVE WILD POINTS?
144.          OUTPUT(8) 'REMOVE WILD POINTS? (T OR F)'
145.          INPUT(7) FW
146.          IF (.NOT.FW) GO TO 150
147.          LCNT=0; AVGN=1.0/NPTS; DPTS=1.0/(NPTS*(NPTS-1)); NUM=0
148.          NSTART=NPTS/2+1; WSQ=WILDPT**2
149.          SUM=0.0; SUMSQ=0.0
150.          DO 141 J=1,NPTS
151.          TEMP=TDATA(3,J)
152.          SUM=SUM+TEMP
153.      141   SUMSQ=SUMSQ+TEMP**2
154.          SD=(NPTS*SUMSQ-SUM*SUM)*DPTS
155.          IF (SD.LT. 0.0) OUTPUT(8) 'DATA ERROR',SD; GO TO 148
156.          AVG=SUM*AVGN
157.          RANGE=WSQ*SD
```

```
158.          DO 142 J=1,NSTART
159.          TEST=ABS(TDATA(3,J)-AVG)**2
160.          IF (TEST.LT.RANGE) GO TO 142
161.          TDATA(3,J)=AVG        ;LCNT=LCNT+1
162.          IF (FWILD) OUTPUT(8) TEST,RANGE,TDATA(1,J)
163.      142 CONTINUE
164.          DO 143 J=NSTART+1,1876-NSTART
165.          TEMP1=TDATA(3,J-NSTART)
166.          TEMP2=TDATA(3,J+NSTART-1)
167.          SUM=SUM-TEMP1+TEMP2
168.          SUMSQ=SUMSQ-TEMP1**2+TEMP2**2
169.          SD=(NPTS*SUMSQ-SUM**2)*DPTS
170.          IF (SD.LT. 0.0) OUTPUT(8) 'DATA ERROR',TDATA(1,J),SD; GO TO 148
171.          AVG=SUM*AVGN
172.          TEST=ABS(TDATA(3,J)-AVG)**2
173.          RANGE=WSQ*SD
174.          IF (TEST.LT.RANGE) GO TO 143
175.          TDATA(3,J)=AVG; LCNT=LCNT+1
176.          IF (FWILD) OUTPUT(8) TEST,RANGE,TDATA(1,J)
177.      143 CONTINUE
178.          DO 145 J=1877-NSTART,1875
179.          TEST=ABS(TDATA(3,J)-AVG)
180.          IF (TEST.LT.RANGE) GO TO 145
181.          TDATA(3,J)=AVG; LCNT=LCNT+1
182.          IF (FWILD) OUTPUT(8) TEST,RANGE,TDATA(1,J)
183.      145 CONTINUE
184.      148 CONTINUE
185.          WRITE(8,149) LCNT
186.      149 FORMAT (2X,I4,' WILD POINTS WERE EDITED')
187.      C***** OUTPUT STATUS?
188.      150 IF (.NOT.FSTAT) GO TO 155
189.          OUTPUT(8) II,JJ,TDATA(II,JJ),ISTART,CURSOR
190.          FSTAT=.FALSE.
191.      155 CONTINUE
192.      C***** INPUT NEW STARTING VALUE?
193.          IF (.NOT.FSTART) GO TO 160
194.          OUTPUT(8) 'START=?(MINUTES)'
195.          INPUT(7) TIME
196.          JJ=(60.0*TIME)/0.8
197.          ISTART=JJ
198.          FLAG=.TRUE.
199.          FSTART=.FALSE.
200.      160 CONTINUE
201.      C***** SAVE THE CURRENT FILE?
202.          IF (.NOT.FSFILE) GO TO 180
203.      161 OUTPUT(8) 'FILE NUMBER=?(6,7,8,9,10)'
204.          INPUT(7) MFILE
205.          IF ((MFILE.LT.6).OR.(MFILE.GT.10)) OUTPUT(8) 'ERROR'; GO TO 161
206.          JFILE@MFILE+10
207.          OUTPUT(8) 'RUN TYPE CODE=?'
208.          INPUT(7) KID(40)
209.          OUTPUT(8) 'PROCESSING CODE=?'
210.          INPUT(7) KID(41)
211.          CALL STOPCLK
```

```
212.          REWIND JFILE
213.          CALL BUFFER OUT(JFILE,1,RID,IDS,ISTAT,ICNT); ERR=60
214.     162  GO TO (162,163,40,55) ISTAT
215.     163  IF(ICNT.NE.IDS) GO TO 60
216.          N=1
217.          DO 170 IREC=2,NREC
218.          CALL BUFFER OUT(JFILE,1,TDATA(1,N),375,ISTAT,ICNT); ERR=60+IREC
219.     167  GO TO (167,168,40,55) ISTAT
220.     168  IF (ICNT.NE.375) GO TO 60
221.          N=N+75
222.     170  CONTINUE
223.          END FILE JFILE
224.          REWIND JFILE
225.          WRITE(8,175) NFILE,MFILE
226.     175  FORMAT (' FILE #',I2,' SAVED IN FILE#',I2)
227.          FSFILE=.FALSE.
228.          CALL STARTCLK(800,2Z6A)
229.     180  CONTINUE
230.     C*****DESIGN A NEW FILTER?
231.          IF (.NOT.FILTERD) GO TO 190
232.     181  OUTPUT(8) '# OF POINTS=? (ODD VALUE)'
233.          INPUT(7) IFLT
234.          IF (IFLT.GT.401) OUTPUT(8) 'ERROR GT 401'; GO TO 181
235.          IFLT=IFLT/2
236.          OUTPUT(8) 'CUTOFF FREQ=?(HZ)'
237.          INPUT(7) CUTF
238.          SF=CUTF*0.8*PI
239.          WTS(0)=CUTF; SUM=WTS(0)
240.          DO 185 I=1,IFLT
241.          WTS(I)=SIN(SF*I)/(PI*I)
242.     185  SUM=SUM+2*WTS(I)
243.          DO 186 I=0,IFLT
244.     186  WTS(I)=WTS(I)/SUM
245.          IF (FWTS) WRITE(8,187) (WTS(I),I=0,IFLT)
246.     187  FORMAT(5(2X,G))
247.          FWTS=.FALSE.
248.          FILTERD=.FALSE.
249.     190  CONTINUE
250.     C*****CHANGE THE DISPLAY?
251.          IF (.NOT.FCHNG) GO TO 510
252.          CALL STOPCLK
253.          REWIND 21
254.          CALL BUFFER OUT(21,1,KID,IDS,ISTAT,ICNT); ERR=90
255.          KSTART=1; KSTOP=75; N=1
256.     501  GO TO (501,502,40,55) ICNT
257.     502  IF (ICNT.NE.IDS) GO TO 60
258.          DO 509 IREC=2,NREC
259.          J=0
260.          DO 504 M=KSTART,KSTOP
261.          J=J+1
262.          DO 503 I=1,5
263.     503  RDATA(I,J)=TDATA(I,M)
264.     504  CONTINUE
265.          CALL BUFFER IN(21,1,TDATA(1,N),375,ISTAT,ICNT); ERR=100+IREC
```

```
      505   GO TO (505,506,40,50) ISTAT
      506   IF (ICNT.NE.375) GO TO 60
            BACKSPACE 21
            CALL BUFFER OUT(21,1,RDATA,375,ISTAT,ICNT); ERR=130+IREC
      507   GO TO (507,508,40,55) ISTAT
      508   IF (ICNT.NE.375) GO TO 60
            N=N+75
            KSTART=KSTART+75
            KSTOP=KSTOP+75
      509   CONTINUE
            REWIND 21
            FLAG=.TRUE.; FCHNG=.FALSE.
            CALL STARTCLK(800,2Z6A)
      510   CONTINUE
      C*****MAKE THE FOAM CORRECTION?
            IF (.NOT.FOAM) GO TO 520
            DO 515 I=1,1875
      515   TDATA(3,I)=(-.9+SQRT(.81-.008*(.001-TDATA(3,I))))/.004
            FLAG=.TRUE.; FOAM=.FALSE.
      520   CONTINUE
      C******FLLEG PUTS LL DATA INTO PLVC ARRAY******
            IF(.NOT. FLLEG) GO TO 530
            DO 525 J=1,1875
      525   TDATA(3,J)=TDATA(2,J)
            OUTPUT(8) 'SAVE THIS FILE AND BEGIN PROCESSING'
            FLLEG=.FALSE.
      530   CONTINUE
            RETURN
      C*****
      C*************************************************************************
      C*****
            SUBROUTINE SCHEDLR
      C*****THIS SUBROUTINE READS THE SWITCHES
            IF (FRLS.OR.SSW(4)) CALL STOPCLK; CALL OCTYAO1;CALL RLSBARN
            IF (FEND) CALL STOPCLK; CALL RLSBARN
      C*****INPUT SWITCH SETTING
            CALL DIGIN(1,IDATA)
            FLBNP=UNPACK(IDATA,15)
            FLL=UNPACK(IDATA,14)
            FRL=UNPACK(IDATA,13)
            FPLVC=UNPACK(IDATA,12)
            FBIAS=UNPACK(IDATA,11)
            FUPDATE=UNPACK(IDATA,10)
            FSLEW=UNPACK(IDATA,9)
            FXP=UNPACK(IDATA,8)
            FXM=UNPACK(IDATA,7)
            FYP=UNPACK(IDATA,6)
            FYM=UNPACK(IDATA,5)
            FSWITCH=UNPACK(IDATA,4)
            FZERO=UNPACK(IDATA,3)
            FBEGIN=UNPACK(IDATA,2)
            FLAST=UNPACK(IDATA,1)
            FEOP=UNPACK(IDATA,0)
            CALL DIGIN(2,IDATA)
```

```
320.          FAST=UNPACK(IDATA,15)
321.    C*****CHANGE THE DATA VALUE?
322.          IF (FUPDATE) TDATA(II,JJ)=CURSOR
323.    C*****CHANGE THE DISPLAY?
324.          IF (FLBNP) II=5; CALL SSCALE(1.0,4.8,0,240)
325.          IF (FLL) II=2; CALL SSCALE(1.0,30.0,0,30)
326.          IF (FRL) II=4; CALL SSCALE(1.0,30.0,0,30)
327.          IF (FPLVC) II=3; CALL SSCALE(1.0,30.0,0,30)
328.          IF (FLBNP.OR.FLL.OR.FRL.OR.FPLVC) CALL DISPLAY
329.    C*****REMOVE THE BIAS?
330.          IF (.NOT.FBIAS) GO TO 210
331.          IF (F1) OUTPUT(8) 'BIAS'
332.          OFFSET=CURSOR-TDATA(II,JJ)
333.          DO 200 J=JJ,1875
334.    200   TDATA(II,J)=TDATA(II,J)+OFFSET
335.          CALL DISPLAY
336.    210   CONTINUE
337.    C*****SET ZERO OFFSET?
338.          IF (.NOT.FZERO) GO TO 220
339.          DO 212 J=1,1875
340.    212   IF (TDATA(1,J).EQ.RID(33)) GO TO 213
341.    213   IF (J.EQ.1875) OUTPUT(8) 'ERROR...NO P1'; GO TO 220
342.          IF (J.LT.11) OUTPUT(8) 'ERROR...P1 TOO SMALL'; GO TO 220
343.          K=J-10
344.          SUM=0.0
345.          DO 215 I=K,J
346.    215   SUM=SUM+TDATA(3,I)
347.          SMALL=SUM/10.0
348.          DO 216 J=1,1875
349.    216   TDATA(3,J)=TDATA(3,J)-SMALL
350.          CALL DISPLAY
351.    220   CONTINUE
352.    C*****UPDATE CURSOR POSITION
353.          IF (II.EQ.5) GO TO 225
354.          IF (.NOT.FSLEW) XS=1.0;YS=STEP;GO TO 230
355.          IF (FAST) XS=XSLEWF; YS=YSLEWF; GO TO 230
356.          XS=XSLEWS; YS=YSLEWS; GO TO 230
357.    225   IF (.NOT.FSLEW) XS=1.0; YS=STEPNP; GO TO 230
358.          IF (FAST) XS=XSLEWF; YS=YSLEWNPF; GO TO 230
359.          XS=XSLEWS; YS=YSLEWNP
360.    230   IF (FXP) JJ=JJ+XS
361.          IF (FXM) JJ=JJ-XS
362.          IF (FYP) CURSOR=CURSOR+YS
363.          IF (FYM) CURSOR=CURSOR-YS
364.          IF (FSLEW) GO TO 250
365.    240   CALL DIGIN(1,IDATA)
366.          IF (UNPACK(IDATA,5).OR.UNPACK(IDATA,6).OR.UNPACK(IDATA,7)
367.         *  .OR.UNPACK(IDATA,8)) GO TO 240
368.    250   CONTINUE
369.    C*****CHECK FOR WRAP-AROUND
370.          IF (JJ.LT.1) JJ=1
371.          IF (JJ.GT.1875) JJ=1875
373.          IF (JJ.LT.ISTART) ISTART=ISTART-250; FLAG=.TRUE.
373.          IF (ISTART.LT.1) ISTART=1
```

```
374.          IF (JJ.GT.(ISTART+250)) ISTART=ISTART+250; FLAG=.TRUE.
375.          IF (ISTART.GT.1625) ISTART=1625
376.          IF (FLAG) CALL DISPLAY
377.          FLAG=.FALSE.
378.          IF (II.EQ.5) GO TO 260
379.          IF (CURSOR.GT.7.0) CURSOR=7.0
380.          IF (CURSOR.LT.-1.0) CURSOR=-1.0
381.          GO TO 270
382.    260   IF (CURSOR.GT.0.0) CURSOR=0.0
383.          IF (CURSOR.LT.-50.0) CURSOR=-50.0
384.    270   CONTINUE
385.    C*****DISPLAY NEW CURSOR POINT
386.          X=JJ-ISTART+1
387.          IF (FXP.OR.FXM.OR.FYP.OR.FYM) CALL PLOTS(X,CURSOR)
388.    C***** SWITCH FILES?
389.          IF (.NOT.(FSWITCH.EOR.FOLD)) GO TO 275
390.          FOLD=FSWITCH
391.          FCHNG=.TRUE.
392.          CALL TRIGGER(2Z68)
393.    275   CONTINUE
394.    C*****SAVE EOP ELAPSED TIME?
395.          IF (.NOT.FEOP) GO TO 280
396.    276   OUTPUT(8) 'PERIOD9=?'
397.          INPUT(7) I; IF ((ABS(I).LT.1).OR.(ABS(I).GT.7)) GO TO 276
398.          IF (I.LT.0) RID(32-I)=DFILL;GO TO 280
399.          IF(TDATA(1,JJ).EQ.DFILL) OUTPUT(8)'ERR*TIME=FILL CHAR';GO TO 280
400.          RID(32+I)=TDATA(1,JJ)
401.    280   CONTINUE
402.    C*****SET THE BEGINNING POINT FOR THE FILTER?
403.          IF (.NOT.FBEGIN) GO TO 282
404.          IFIRST=JJ
405.          IF (IFIRST.LT.(IFLT+1)) IFIRST=IFLT+1; OUTPUT(8) 'TOO SMALL'
406.          TIME=TDATA(1,IFIRST)/60.0
407.          OUTPUT(8) TIME
408.    282   CONTINUE
409.    C*****SET THE LAST POINT FOR THE FILTER AND FILTER THE DATA?
410.          IF (.NOT.FLAST) GO TO 290
411.          ILAST=JJ
412.          IF (ILAST.GT.(1875-IFLT)) ILAST=1875-IFLT;OUTPUT(8) 'TOO LARGE'
413.          TIME=TDATA(1,ILAST)/60.0
414.          OUTPUT(8) TIME
415.          DO 287 J=IFIRST,ILAST
416.          SUM=TDATA(3,J)*WTS(0)
417.          DO 285 INC=1,IFLT
418.    285   SUM=SUM+(TDATA(3,J-INC)+TDATA(3,J+INC))*WTS(INC)
419.          TDATA(3,J)=SUM
420.    287   CONTINUE
421.          CALL DISPLAY
422.    290   CONTINUE
423.          RETURN
424.    C*****
425.    C*************************************************************************
426.    C*****
```

```
      SUBROUTINE DISPLAY
C*****THIS SUBROUTINE CAN ONLY BE CALLED BY "SCHEDLR"
      CALL ERASE
      CALL SCHAR(0,246,6,'START=')
      TIME=TDATA(1,ISTART)/60.0
      ENCODE(8,310,ALPHA,N) TIME
310   FORMAT(F6.1)
      CALL SCHAR(48,246,N,ALPHA)
      IF (II.EQ.2) CALL SCHAR(144,246,3,'LL=')
      IF (II.EQ.4) CALL SCHAR(144,246,3,'RL=')
      IF (II.EQ.3) CALL SCHAR(128,246,5,'PLVC=')
      IF (II.EQ.5) CALL SCHAR(128,246,5,'LBNP=')
      ENCODE(8,320,ALPHA,N) TDATA(II,JJ)
320   FORMAT(F7.3)
      CALL SCHAR(168,246,N,ALPHA)
      IF (.NOT.FOLD) CALL SCHAR(232,246,3,'NEW')
      IF (FOLD) CALL SCHAR(232,246,3,'ORG')
      DO 350 I=1,250
      X=I
      J=I+ISTART-1
      Y=TDATA(II,J)
      IF (II.EQ.5) GO TO 330
      IF (Y.LT.-1.0) Y=-1.0
      IF (Y.GT.7.0) Y=7.0
      GO TO 340
330   IF (Y.LT.-50.0) Y=-50.0
      IF (Y.GT.0.0) Y=0.0
340   CONTINUE
350   CALL PLOTS(X,Y)
      X=JJ-ISTART+1
      CALL PLOTS(X,CURSOR)
      RETURN
C*****
C*************************************************************************
C*****
      LOGICAL FUNCTION UNPACK(ITEM,IBIT)
C*****THIS FUNCTION RETURNS TRUE IF THE I TH BIT=1
      ISHIFT =IBIT-15
      IVAL=ISL(ITEM,ISHIFT)
      IVAL=IAND(IVAL,1)
      UNPACK=.FALSE.
      IF (IVAL.EQ.1) UNPACK=.TRUE.
      IF (FBUG.AND.UNPACK) OUTPUT(8) IBIT
      RETURN
      END
!!JCP

!PAUSE INTERRUPT & KEY-IN FSC
!PAUSE INTERRUPT & KEY-IN FSC
!!KEY-IN
FSC
```

```
!0LØAD GØ,(UDCB,15),(FILE,FP,BARNETT),(FØRE,1000),(TEMP,2000),(LIB,USER,SYSTEM)
!!0LØAD
:ASSIGN (F:5,TYA01),VFC
:ASSIGN (F:6,TYA01)
:ASSIGN (F:7,TYA02),VFC
:ASSIGN (F:8,TYA02)
:ASSIGN (F:11,D1,F1),BIN
:ASSIGN (F:12,D1,F2),BIN
:ASSIGN (F:13,D1,F3),BIN
:ASSIGN (F:14,D1,F4),BIN
:ASSIGN (F:15,D1,F5),BIN
:ASSIGN (F:16,D1,F6),BIN
:ASSIGN (F:17,D1,F7),BIN
:ASSIGN (F:18,D1,F8),BIN
:ASSIGN (F:19,D1,F9),BIN
:ASSIGN (F:20,D1,F10),BIN
:ASSIGN (F:21,D1,F11),BIN
LØADING WAS COMPLETED
!!JCP

!FIN
!!BEGIN IDLE
```

APPENDIX C

DESCRIPTORS FOR RUN TYPE CODE

1. Data OK
2. Missing Data
3. Questionable or bad data
4. Presyncope
5. Data over range
6. 10 min at -40 mm Hg
7. Timing off
8. Rt leg offset - used left leg instead of PLVC data
9.
10.

Combinations - 2nd digit indicates 2nd condition

21 -- 30 .Missing data

31 -- 40 Questionable or bad data

41 -- 50 Presyncope

51 -- 60 Data over range

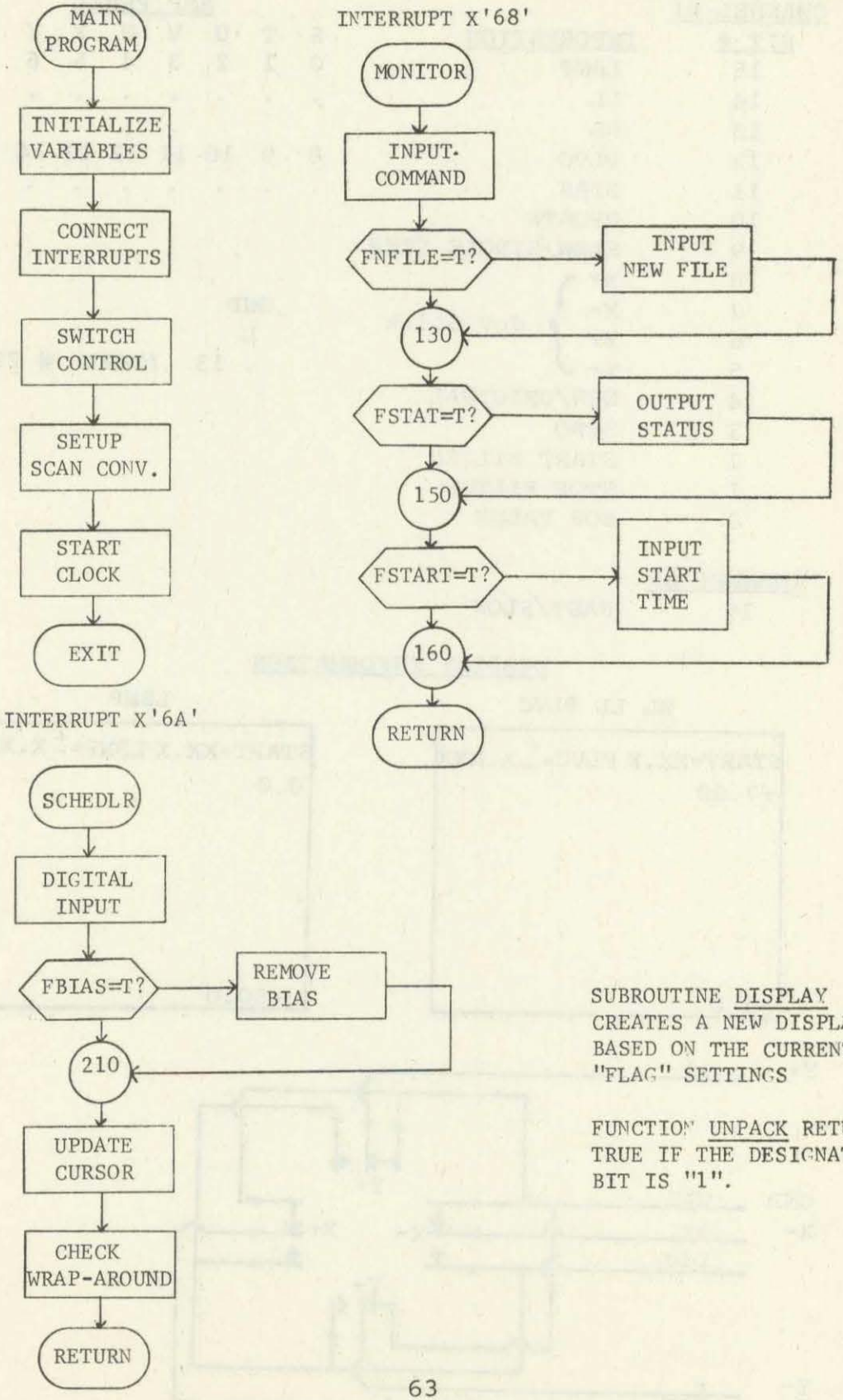
61 -- 70 10 min at -40 mm Hg

71 -- 80 Timing off

81 -- 90 Rt leg offset - used left leg

FOR COMBINATION CODES THE 2ND DIGIT SHOULD ALWAYS BE LARGER THAN THE FIRST DIGIT (i.e. 28,78)

APPENDIX D
MO-92 INTERACTIVE GRAPHICS PACKAGE
BASIC FLOW CHARTS



SUBROUTINE DISPLAY
CREATES A NEW DISPLAY
BASED ON THE CURRENT
"FLAG" SETTINGS

FUNCTION UNPACK RETURNS
TRUE IF THE DESIGNATED
BIT IS "1".

APPENDIX E

MO-92 INTERACTIVE GRAPHICS PACKAGE

CONTROLS

DIGITAL INPUTS

CHANNEL #1

BIT #	INFORMATION
15	LBNP
14	LL
13	RL
12	PLVC
11	BIAS
10	UPDATE
9	SLEW/SINGLE STEP
8	X+
7	X-
6	Y+
5	Y-
4	NEW/ORIGINAL
3	ZERO
2	START FILTER
1	STOP FILTER
0	EOP VALUE

(X+, X-, Y+, Y- : Joy Stick)

AMP PANEL

S	T	U	V	W	X	Y	Z	
0	1	2	3	4	5	6	7	1
.	.	.	.	.	.	.	.	
8	9	10	11	12	13	14	15	2
.	.	.	.	.	.	.	.	

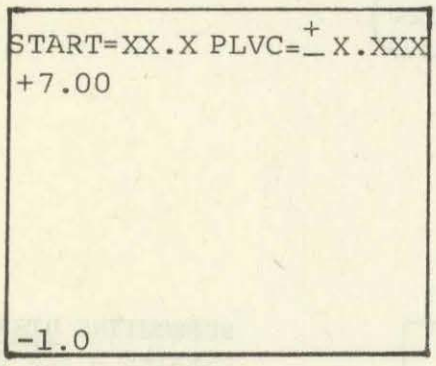
GND
L
. 13 (CABLE # 22)

CHANNEL #2

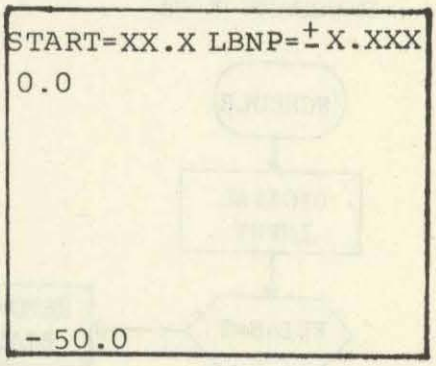
15	FAST/SLOW

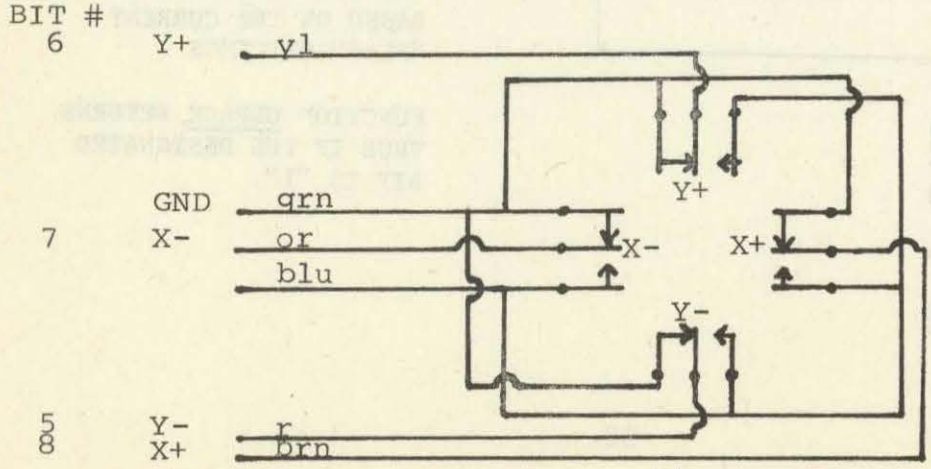
DISPLAY INFORMATION

RL LL PLVC



LBNP

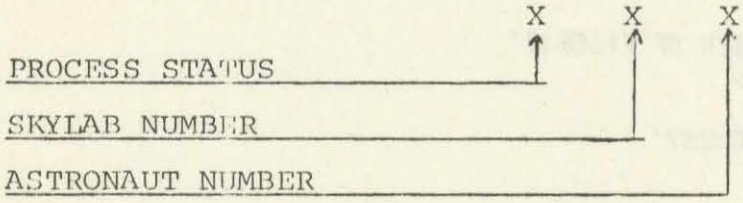




APPENDIX F

CODE FOR M092 PROCESSED MAGNETIC TAPE DATA

Code consists of a 3 digit number which indicates the process status, skylab number and astronaut number.



PROCESS STATUS		MAG TAPE NUMBER
1. Merged and Sequenced Data:		
SL 2	CONRAD	121
	KERWIN	122
	WEITZ	123
SL 3	BEAN	131
	GARRIOTT	132
	LOUSMA	133
SL 4	CARR	141
	GIBSON	142
	POGUE	143
2. Academy Processed Data: - Not Sequenced by Run #		221
		222
		223 etc.
3. Academy Processed and Sequenced Data: (Send to NASA)		321
		322
		323 etc.
4. Academy Sequenced And Processed Data: Merged with NASA M092 Data		421
		422
		423 etc.

APPENDIX G

PROGRAM FOR PLOTTING UNPROCESSED DATA FROM JSC

```
C***SKYLAB LVMS DATA(RAW)200 BPI
C***
      DIMENSION RDATA(5,75),FID(12),ID(10),TID(7),KID(12)
      DATA FILL /8Z407FFFFF/
      NAME LIST
      OUTPUT(6) 'SET MT TO 200 BPI'
    8 OUTPUT(6) 'SKYLAB NO. = ?'
      INPUT(5) NEXPR
      OUTPUT(6) 'NUMBER OF FILES=?'
      INPUT(5) NFILES
      IEND=NFILES+10
      OUTPUT(6) 'NREC=287'
      NREC=28
      INPUT(5)
      DO 600 IFILE =11,IEND
      NFILE=IFILE-10
      OUTPUT(6) NFILE
C***  THIS CALL MUST BE FIRST TO INITIALIZE THE MAG TAPE
      CALL CLRPLT(NFILE)
C***     LABEL PLOT
      GO TO(6,11,12,13),NEXPR
    6 OUTPUT(6) 'YOU DUMMY,SKYLAB I IS THE WORKSHOP'
      GO TO 8
   11 CALL SYMBOL(0.25,10.7,0.2,0.,1,0,
     130,'SKYLAB II,EXPERIMENT M092,LVMS')
      GO TO 15
   12 CALL SYMBOL(0.25,10.7,0.2,0.,1,0,
     131,'SKYLAB III,EXPERIMENT M092,LVMS')
      GO TO 15
   13 CALL SYMBOL(0.25,10.7,0.2,0.,1,0,
     130,'SKYLAB IV,EXPERIMENT M092,LVMS')
   15 CONTINUE
C***     DRAW HORIZONTAL AXIS
      CALL AXIS(0.5,0.5,'TIME-MINUTES',-12,10.0,0,0.,2.5,0)
C***     DRAW VERTICAL AXIS
      CALL AXIS(0.5,0.5, 'PERCENT LEG VOLUME CHANGE',25,9.0,90.,-.75,
     1.75,-1)
      CALL PLOT(.5,9.5,1)
      CALL PLOT(.5,10.5,2)
      CALL PLOT(.4,10.5,2)
      CALL SYMBOL(0.8,.7,.1,0.,1,0,8,'PERIOD 1')
      CALL SYMBOL(3.5,.7,.1,0.,1,0,1,'2')
      CALL SYMBOL(5.5,.7,.1,0.,1,0,1,'3')
      CALL SYMBOL(7.5,.7,.1,0.,1,0,1,'4')
      CALL SYMBOL(9.5,.7,.1,0.,1,0,1,'5')
C***  DRAW RT. HAND VERTICAL AXIS
      CALL PLOT(10.5,0.5,1)
C***  DRAW RT. HAND VERTICAL AXIS
      CALL PLOT(10.5,10.5,2)
      CALL PLOT(10.6,10.5,2)
      CALL PLOT(10.5,9.5,1)
      CALL PLOT(10.6,9.5,2)
```

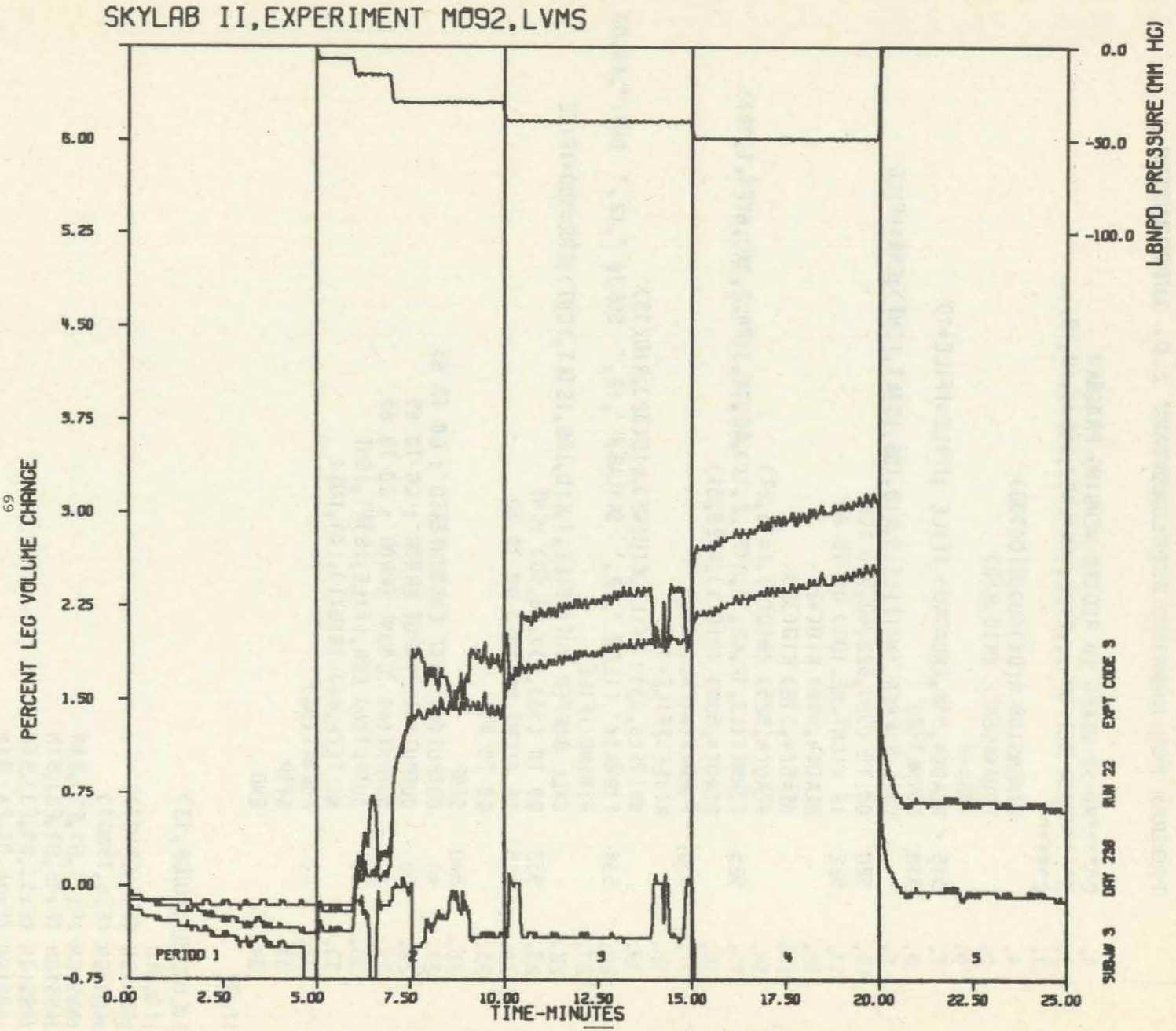
```
      CALL PLOT(10.5,8.5,1)
      CALL PLOT(10.5,8.5,2)
      CALL SYMBOL(10.7,10.45,.1,0.,1,0,5,'  0.0')
      CALL SYMBOL(10.7,9.45,.1,0.,1,0,5,'-50.0')
      CALL SYMBOL(10.7,8.45,.1,0.,1,0,6,'-100.0')
      CALL SYMBOL(11.5,8.2,.15,90.,1,0,21,'LBNPD PRESSURE(MM HG)')
C***  DRAW VERTICAL GRID LINES
      CALL PLOT(10.5,10.5,1)
      CALL PLOT(8.5,10.5,2)
      CALL PLOT(8.5,.5,2)
      CALL PLOT(6.5,.5,1)
      CALL PLOT(6.5,10.5,2)
      CALL PLOT(8.5,10.5,1)
      CALL PLOT(4.5,10.5,2)
      CALL PLOT(4.5,.5,2)
      CALL PLOT(2.5,.5,1)
      CALL PLOT(2.5,10.5,2)
      CALL PLOT(4.5,10.5,1)
      CALL PLOT(.5,10.5,2)
      CALL BUFFER IN(IFILE,1,KID,12,ISTAT,ICNT)
2     GO TO (2,3,40,50) ISTAT
3     CONTINUE
      IF (ICNT.NE.12) OUTPUT(6) 'COUNT ERROR'; STOP
C     UNPACK INTEGER DATA
      DO 20 I=1,5
      ID(2*I-1)=ISA(KID(I),-16)
      ID(2*I)=ISL(KID(I),16)
20    ID(2*I)=ISA(ID(2*I),-16)
      CALL SYMBOL(11.0,9.5,0.1,90.0,8,0,6,'SUBJ# ',
     * 1,-1,ID(3),0,8,'    DAY ',1,-1,ID(4),0,8,'    RUN ',
     * 1,-1,ID(2),0,14,'    EXPT CODE ',1,-1,ID(9))
      DO 300 K=2,5
      REWIND IFILE
      CALL BUFFER IN(IFILE,1,FID,12,ISTAT,ICNT)
7     GO TO (7,9,40,50) ISTAT
9     CONTINUE
      IF (ICNT.NE.12) OUTPUT(6) 'COUNT ERROR'; STOP
      DO 30 I=1,7
30    TID(I)=FID(I+5)
      CALL OFFSET(-75.,150.,-1.125,.75)
      IF(K .EQ. 5) CALL OFFSET(-75.,150.,-525.,50.)
      DO 200 IREC=2,NREC
      CALL BUFFER IN(IFILE,1,RDATA,375,ISTAT,ICNT)
100   GO TO (100,110,700,50) ISTAT
110   CONTINUE
      IF (ICNT.NE.375) OUTPUT(6) 'COUNT ERROR'; STOP
C***     PLOT DATA
C***        CALL PEN TO ORIGIN
      IF(IREC .EQ. 2) CALL PLOT(RDATA(1,1),RDATA(K,1),-1)
      DO 200 J=1,75
      X=RDATA(1,J)
      Y=RDATA(K,J)
      IF(X .EQ. FILL) GO TO 200
```

```
106.            IF(Y .EQ. FILL) GO TO 200
107.            IF((K.NE.5).AND.(Y.LT.-.75)) Y=-.75
108.            IF((K.NE.5).AND.(Y.GT.7.00)) Y=7.0
109.            IF((K .EQ. 5).AND.(Y .LT. -60.0)) Y=-60.0
110.            IF((K .EQ. 5).AND.(Y .GT. 2.0)) Y=2.0
111.            IF(X .LT. 0.0) X=0.0
112.            IF(X .GT. 1550.0) X=1550.0
113.            CALL PLOT(X,Y,-2)
114.          192 CONTINUE
115.      200   CONTINUE
116.            CALL PEN(1,'9')
117.      300   CONTINUE
118.            CALL PLOT(20.0,0.0,1)
119.       600  CONTINUE
120.      C***     THIS CALL MUST BE LAST TO FINISH AND REWIND THE MAG TAPE
121.            CALL CLRPLT(999)
122.            OUTPUT(6) 'SET MT TO 556 BPI'
123.            CALL RLSBARN
124.      700   IF(K.NE.2) GO TO 300
125.            IREC=IREC-1
126.            WRITE(6,710) IREC
127.      710   FORMAT (' # OF RECORDS= ',I3)
128.            GO TO 300
129.      40    OUTPUT(6) 'EOF ENCOUNTERED@; STOP
130.      50    OUTPUT(6) 'INPUT ERROR'; STOP
131.            END
!!JCP

!PAUSE INTERRUPT & KEY IN FSC
!PAUSE INTERRUPT & KEY IN FSC
!!KEY-IN
FSC

!@LOAD GO,(UDCB,15),(FILE,FP,BARNETT),(FORE,1000),(TEMP,2000),(LIB,USER,SYSTEM)
!!@LOAD
:ASSIGN (F:4,CRA03)
:ASSIGN (F:5,TYA01),VFC
:ASSIGN (F:6,TYA01)
:ASSIGN (F:7,7TA81),VFC,(RECL,132),BCD
:ASSIGN (F:11,D1,F1),BIN
:ASSIGN (F:12,D1,F2),BIN
:ASSIGN (F:13,D1,F3),BIN
:ASSIGN (F:14,D1,F4),BIN
:ASSIGN (F:15,D1,F5),BIN
:ASSIGN (F:16,D1,F6),BIN
:ASSIGN (F:17,D1,F7),BIN
:ASSIGN (F:18,D1,F8),BIN
:ASSIGN (F:19,D1,F9),BIN
:ASSIGN (F:20,D1,F10),BIN
:ASSIGN (F:21,D1,F11),BIN
LOADING WAS COMPLETED
!!JCP

!FIN
!!BEGIN IDLE
```



· APPENDIX H PLOT OF UNPROCESSED DATA

APPENDIX I

PROGRAM FOR MERGING SUPPLEMENTARY I.D. INFORMATION

```
      C*****MØ-92 USAFA ID RECØRD MERGING PRØGRAM
      C*****DECK SET-UP ...FILE#,SKYLAB#,SUBJ#,ID,ID...
      C*****
            DIMENSIØN RID(100),KID(100)
            EQUIVALENCE (KID,RID)
            IDS=42
      515   READ(4,518,END=540) IFILE ;IFILE=IFILE+10
      518   FØRMAT(G)
            CALL BUFFER IN(IFILE,1,KID,IDS,ISTAT,ICNT);ERR=IFILE
      520   GØ TØ (520,522,40,50) ICNT
      522   IF (ICNT.NE.IDS) GØ TØ 60
            READ(4,518) KID(42)
            READ(4,518) KID(32)
            READ(4,525) (RID(I),I=13,27)
      525   FØRMAT(I3,1X,A2,1X,3(F4.2,1X),A2,1X,3(F4.2,1X),6(F4.1,1X))
            READ(4,530) (RID(I),I=28,31)
      530   FØRMAT(4(F4.1,1X))
            KFILE=IFILE-10
            WRITE(6,531) KFILE,KID(42),KID(32),RID(13)
      531   FØRMAT(' FILE# ',I2,'  SKYLAB# ',I2,'  SUBJ# ',I2,'  DAY ',F4.0)
            REWIND IFILE
            CALL BUFFER ØUT(IFILE,1,KID,IDS,ISTAT,ICNT);ERR=100+IFILE
      533   GØ TØ (533,535,40,50) ICNT
      535   IF (ICNT.NE.IDS) GØ TØ 60
            GØ TØ 515
      540   STØP
      40    ØUTPUT(6) 'EØF ENCØUNTERED'; GØ TØ 65
      50    ØUTPUT(6) 'INPUT ERRØR'; GØ TØ 65
      60    ØUTPUT(6) 'CØUNT ERRØR'; GØ TØ 65
      65    ØUTPUT(6) ERR,IFILE,ISTAT,ICNT
            WRITE(6,66) (RID(I),I=1,IDS)
      66    FØRMAT(5G)
            STØP
            END
!!JCP

!ØLØAD GØ,(UDCB,13)
!!ØLØAD
:ASSIGN (F:4,CRA03)
:ASSIGN (F:6,TYA01)
:ASSIGN (F:11,D1,F1),BIN
:ASSIGN (F:12,D1,F2),BIN
:ASSIGN (F:13,D1,F3),BIN
:ASSIGN (F:14,D1,F4),BIN
:ASSIGN (F:15,D1,F5),BIN
:ASSIGN (F:16,D1,F6),BIN
:ASSIGN (F:17,D1,F7),BIN
:ASSIGN (F:18,D1,F8),BIN
:ASSIGN (F:19,D1,F9),BIN
:ASSIGN (F:20,D1,F10),BIN
:ASSIGN (F:21,D1,F11),BIN
LØADING WAS CØMPLETED
```

APPENDIX J

PROGRAM FOR PLOTTING USAF PROCESSED DATA

```
1.    C***** M0-92 CALCOMP OUTPUT,200 BPI *****
2.    C***SKYLAB LVMS DATA(PROCESSED)
3.    C***** PROGRAM PLOTS LEFT LEG ONLY AND DATA MUST HAVE ID RECORDS *****
4.    C***
5.          DIMENSION RDATA(5,75),FID(100),ID(10),TID(7),KID(100)
6.          EQUIVALENCE (FID,KID)
7.          DATA FILL /8Z407FFFFF/
8.          NAME LIST
9.          OUTPUT(6) 'SET MT TO 200 BPI'
10.         OUTPUT(6) 'TO STOP THIS PROGRAM INPUT 0 FOR THE FILE #'
11.         IPLT=0
12.         IDS=42
13.   8     OUTPUT(6) 'FILE# =?'
14.         INPUT(5) NFILE
15.         OUTPUT(6) 'NREC=?'
16.         INPUT(5) NREC
17.         NREC=28
18.         IF ((NFILE.LE.0).OR.(NFILE.GT.11)) GO TO 600
19.         IFILE=NFILE+10
20.         CALL BUFFER IN(IFILE,1,KID,IDS,ISTAT,ICNT)
21.   2     GO TO (2,3,40,50) ISTAT
22.   3     CONTINUE
23.         IF (ICNT.NE.IDS) OUTPUT(6) 'COUNT ERROR'; GO TO 8
24.         IPLT=IPLT+1
25.   C***  THIS CALL MUST BE FIRST TO INITIALIZE THE MAG TAPE
26.         CALL CLRPLT(IPLT)
27.   C***     LABEL PLOT
28.         GO TO(6,11,12,13) KID(42)
29.         GO TO 15
30.       6 OUTPUT(6) 'YOU DUMMY,SKYLAB I IS THE WORKSHOP'
31.         GO TO 8

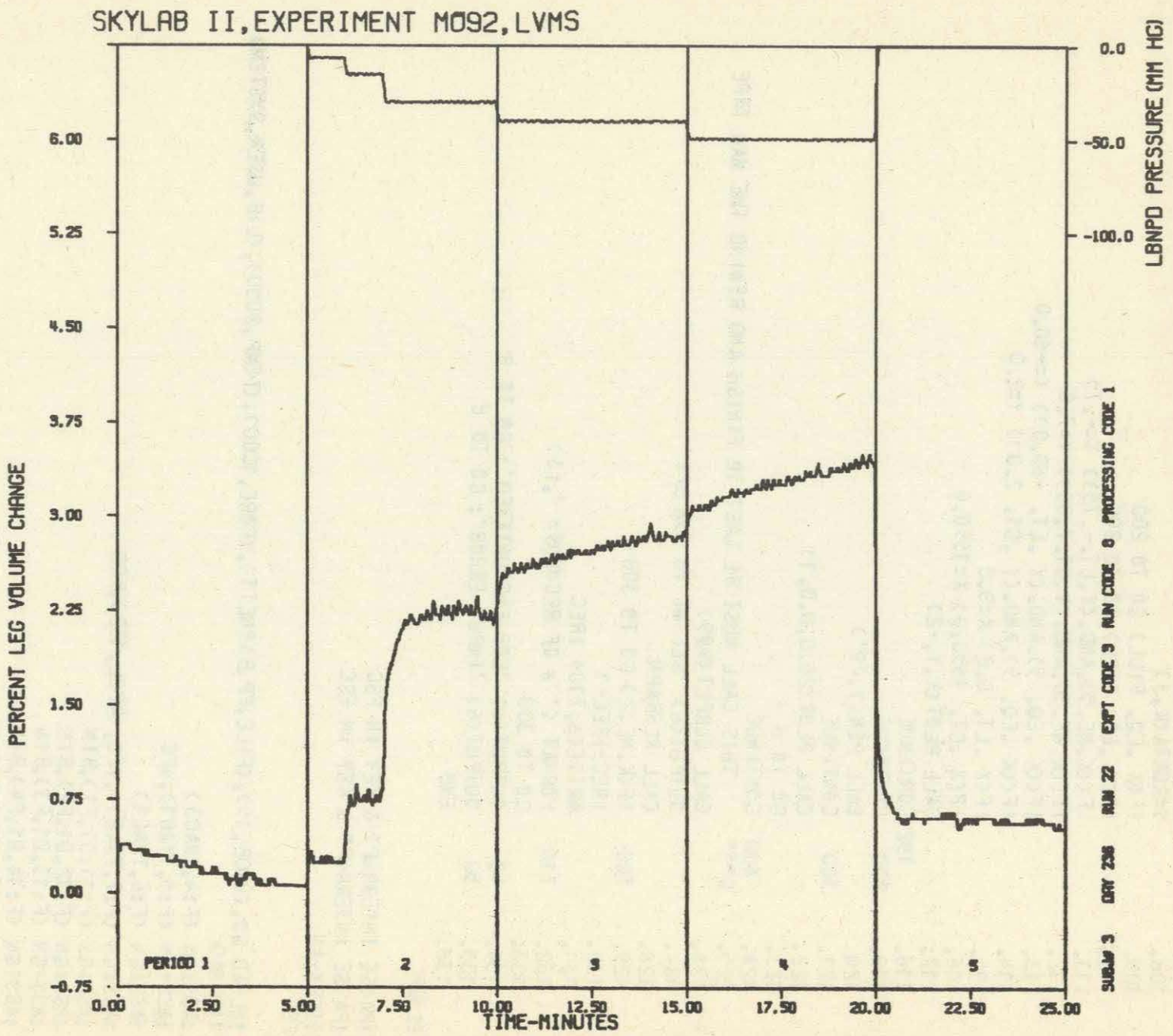
32.      11 CALL SYMBOL(0.25,10.7,0.2,0.,1,0,
33.         130,'SKYLAB II,EXPERIMENT M092,LVMS')
34.         GO TO 15
35.      12 CALL SYMBOL(0.25,10.7,0.2,0.,1,0,
36.         131,'SKYLAB III,EXPERIMENT M092,LVMS')
37.         GO TO 15
38.      13 CALL SYMBOL(0.25,10.7,0.2,0.,1,0,
39.         130,'SKYLAB IV,EXPERIMENT M092,LVMS')
40.      15 CONTINUE
41.   C***     DRAW HORIZONTAL AXIS
42.         CALL AXIS(0.5,0.5,'TIME-MINUTES',-12,10.0,0,0.,2.5,0)
43.   C***     DRAW VERTICAL AXIS
44.         CALL AXIS(0.5,0.5, 'PERCENT LEG VOLUME CHANGE',25,9.0,90.,-.75,
45.        1.75,-1)
46.         CALL PLOT(.5,9.5,1)
47.         CALL PLOT(.5,10.5,2)
48.         CALL PLOT(.4,10.5,2)
49.         CALL SYMBOL(0.8,.7,.1,0.,1,0,8,'PERIOD 1')
50.         CALL SYMBOL(3.5,.7,.1,0.,1,0,1,'2')
51.         CALL SYMBOL(5.5,.7,.1,0.,1,0,1,'3')
52.         CALL SYMBOL(7.5,.7,.1,0.,1,0,1,'4')
53.         CALL SYMBOL(9.5..7..1.0..1.0.1.'5')
```

```
C***  DRAW RT. HAND VERTICAL AXIS
      CALL PLOT(10.5,0.5,1)
C***  DRAW RT. HAND VERTICAL AXIS
      CALL PLOT(10.5,10.5,2)
      CALL PLOT(10.6,10.5,2)
      CALL PLOT(10.5,9.5,1)
      CALL PLOT(10.6,9.5,2)
      CALL PLOT(10.5,8.5,1)
      CALL PLOT(10.6,8.5,2)
      CALL SYMBOL(10.7,10.45,.1,0.,1,0,5,'  0.0')
      CALL SYMBOL(10.7,9.45,.1,0.,1,0,5,'-50.0')
      CALL SYMBOL(10.7,8.45,.1,0.,1,0,6,'-100.0')
      CALL SYMBOL(11.5,8.2,.15,90.,1,0,21,'LBNPD PRESSURE(MM HG)')
C***   DRAW VERTICAL GRID LINES
      CALL PLOT(10.5,10.5,1)
      CALL PLOT(8.5,10.5,2)
      CALL PLOT(8.5,.5,2)
      CALL PLOT(6.5,.5,1)
      CALL PLOT(6.5,10.5,2)
      CALL PLOT(8.5,10.5,1)
      CALL PLOT(4.5,10.5,2)
      CALL PLOT(4.5,.5,2)
      CALL PLOT(2.5,.5,1)
      CALL PLOT(2.5,10.5,2)
      CALL PLOT(4.5,10.5,1)
      CALL PLOT(.5,10.5,2)
C     UNPACK INTEGER DATA
      DO 20 I=1,5
      ID(2*I-1)=ISA(KID(I),-16)
      ID(2*I)=ISL(KID(I),16)
20    ID(2*I)=ISA(ID(2*I),-16)
      CALL SYMBOL(11.0,0.5,0.1,90.0,12,0,6,'SUBJ# ',
     * 1,-1,ID(3),0,8,'    DAY ',1,-1,ID(4),0,8,'    RUN ',
     * 1,-1,ID(2),0,14,'    EXPT CODE ',1,-1,ID(9),
     *0,12,'  RUN CODE  ',1,-1,KID(40),
     * 0,18,'  PROCESSING CODE ',1,-1,KID(41))
      DO 300 K=3,5,2
      REWIND IFILE
      CALL BUFFER IN(IFILE,1,KID,IDS,ISTAT,ICNT)
7     GO TO (7,9,40,50) ISTAT
9     CONTINUE
      IF (ICNT.NE.IDS) OUTPUT(6) 'COUNT ERROR'; GO TO 8
      CALL OFFSET(-75.,150.,-1.125,.75)
      IF(K .EQ. 5) CALL OFFSET(-75.,150.,-525.,50.)
      DO 200 IREC=2,NREC
      CALL BUFFER IN(IFILE,1,RDATA,375,ISTAT,ICNT)
100   GO TO (100,110,700,50) ISTAT
110   CONTINUE
      IF (ICNT.NE.375) OUTPUT(6) 'COUNT ERROR'; GO TO 8
C***     PLOT DATA
C***        CALL PEN TO ORIGIN
      IF(IREC .EQ. 2) CALL PLOT(RDATA(1,1),RDATA(K,1),-1)
      DO 200 J=1,75
```

```
107.            X=RDATA(1,J)
108.            Y=RDATA(K,J)
109.            IF(X .EQ. FILL) GO TO 200
110$            IF(Y .EQ. FILL) GO TO 200
111.            IF((K.NE.5).AND.(Y.LT.-.75)) Y=-.75
112.            IF((K.NE.5).AND.(Y.GT.7.00)) Y=7.0
113.            IF((K .EQ. 5).AND.(Y .LT. -60.0)) Y=-60.0
114.            IF((K .EQ. 5).AND.(Y .GT. 2.0)) Y=2.0
115.            IF(X .LT. 0.0) X=0.0
116.            IF(X .GT. 1550.0) X=1550.0
117.            CALL PLOT(X,Y,-2)
118.          192 CONTINUE
119.        200   CONTINUE
120.            CALL PEN(1,'9')
121.        300   CONTINUE
122.            CALL PLOT(20.0,0.0,1)
123.            GO TO 8
124.         600  CONTINUE
125.        C***     THIS CALL MUST BE LAST TO FINISH AND REWIND THE MAG TAPE
126.            CALL CLRPLT(999)
127.            OUTPUT(6) 'SET MT TO 556 BPI'
128.            CALL RLSBARN
129.        700   IF(K.NE.2) GO TO 300
130.            IREC=IREC-1
131.            WRITE(6,710) IREC
132.        710   FORMAT (' # OF RECORDS= ',I3)
133.            GO TO 300
134.        40    OUTPUT(6) 'EOF ENCOUNTERED'; GO TO 8
135.        50    OUTPUT(6) 'INPUT ERROR'; GO TO 8
136.            END
!!JCP

!PAUSE INTERRUPT & KEY IN FSC
!PAUSE INTERRUPT & KEY IN FSC
!!KEY-IN
FSC

!0LOAD GO,(UDCB,15),(FILE,FP,BARNETT),(FORE,1000),(TEMP,2000),(LIB,USER,SYSTEM)
!!0LOAD
:ASSIGN (F:4,CRA03)
:ASSIGN (F:5,TYA01),VFC
:ASSIGN (F:6,TYA01)
:ASSIGN (F:7,7TA81),VFC,(RECL,132),BCD
:ASSIGN (F:11,D1,F1),BIN
:ASSIGN (F:12,D1,F2),BIN
:ASSIGN (F:13,D1,F3),BIN
:ASSIGN (F:14,D1,F4),BIN
:ASSIGN (F:15,D1,F5),BIN
:ASSIGN (F:16,D1,F6),BIN
:ASSIGN (F:17,D1,F7),BIN
:ASSIGN (F:18,D1,F8),BIN
:ASSIGN (F:19,D1,F9),BIN
:ASSIGN (F:20,D1,F10),BIN
:ASSIGN (F:21,D1,F11),BIN
```



APPENDIX K PLOT OF USAF PROCESSED DATA

APPENDIX L

PROGRAM FOR PRINTOUT OF I.D. AND/OR EXPERIMENTAL DATA

```
C***** M0-92 PRINT OUT PROGRAM
C***** SSW1 ON-OUTPUTS ON TTY-ELSE MT(556 BPI) *****
C***** SSW2 ON-LIMITS OUTPUT *****
C***** SSW3 ON-OUTPUTS ID(4) AND FID(13) DAYS *****
      DIMENSION RDATA(5,75),FID(100),ID(10),TID(7),KID(100),RID(100)
      EQUIVALENCE (FID,KID)
      LOGICAL SSW
      NAME LIST
      NREC=28
      IDS=42
      OUTPUT(6) 'TO STOP THIS PROGRAM INPUT 0 FOR THE FILE#'
1     OUTPUT(6) 'FILE# =?'
      INPUT(5) IFILE
      IF ((IFILE.LE.0).OR.(IFILE.GT.11)) GO TO 600
      IFILE=IFILE+10
      IF (SSW(1)) L=6
      IF (.NOT.SSW(1)) L=7
      CALL BUFFER IN(IFILE,1,KID,IDS,ISTAT,ICNT)
2     GO TO (2,3,40,50) ISTAT
3     CONTINUE
      IF (ICNT.NE.IDS) OUTPUT(6) 'COUNT ERROR'; CALL RLSBARN
C     UNPACK INTEGER DATA
      DO 5 I=1,5
      ID(2*I-1)=ISA(KID(I),-16)
      ID(2*I)=ISL(KID(I),16)
5     ID(2*I)=ISA(ID(2*I),-16)
      DO 10 I=1,7
10    TID(I)=FID(I+5)
      IF(SSW(3))OUTPUT(L)ID(2);OUTPUT(L)ID(4);OUTPUT(L)FID(13);GO TO 90
      WRITE(L,901)
      OUTPUT(L) '  RECORD NUMBER 1¢¢¢¢¢¢'
      WRITE(L,902)
      WRITE(L,70)
70    FORMAT(T5,'EXPT#',T16,'RUN#',T25,'SUBJ#',T37,'DAY',
     * T47,'HRS',T56,'MINS',T66,'SECS',T81,'EXPT CODE')
      WRITE(L,75) ID
75    FORMAT(4X,10(I5,5X))
      WRITE(L,902)
      WRITE(L,80) I,I=1,7
80    FORMAT (7(8X,'T',I1,5X))
      WRITE(L,85) TID
85    FORMAT(10(5X,F10.5))
      WRITE(L,902)
      GO TO(11,12,20,30) KID(42)
11    WRITE(L,13)
13    FORMAT(T2,'ERROR-SL1=WORKSHOP')
      GO TO 39
12    GO TO(14,4,15) KID(32)
14    WRITE(L,6)
 6    FORMAT(T45,'SKYLAB II,CONRAD ¢¢¢¢¢¢')
      GO TO 39
 4    WRITE(L,7)
```

```
      7 FORMAT(T45,'SKYLAB II,KERWIN ¢¢¢¢¢¢¢')
        GO TO 39
   15   WRITE(L,8)
      8 FORMAT(T45,'SKYLAB II,WEITZ ¢¢¢¢¢¢¢')
        GO TO 39
   20   GO TO(21,23,25) KID(32)
   21   WRITE(L,22)
   22   FORMAT(T45,'SKYLAB III,BEAN ¢¢¢¢¢¢¢',//)
        GO TO 39
   23   WRITE(L,24)
   24   FORMAT(T45,'SKYLAB III,GARRIOT ¢¢¢¢¢¢¢',//)
        GO TO 39
   25   WRITE(L,26)
   26   FORMAT(T45,'SKYLAB III,LOUSMA ¢¢¢¢¢¢¢',//)
        GO TO 39
   30   GO TO(32,34,36) KID(32)
   32   WRITE(L,33)
   33   FORMAT(T45,'SKYLAB IV,CARR ¢¢¢¢¢¢¢',//)
        GO TO 39
   34   WRITE(L,35)
   35   FORMAT(T45,'SKYLAB IV,GIBSON ¢¢¢¢¢¢¢',//)
        GO TO 39
   36   WRITE(L,37)
   37   FORMAT(T45,'SKYLAB IV,POGUE ¢¢¢¢¢¢¢',//)
   39   CONTINUE
        WRITE(L,902)
        WRITE(L,41)
   41   FORMAT(T45,'M092 SUMMARY DATA')
        WRITE(L,902)
        WRITE(L,42)
   42   FORMAT(
       1T10,'LEFT LEG LVMS DATA',T37,'RIGHT LEG LVMS DATA',
       1T93,'AMBIENT',T112,'LBNPD')
        WRITE(L,43)
   43   FORMAT(
       1T2,'JUL.',T8,'BAND',T14,'CAL.',T21,'CAL.',T28,'CAL.', T35,'BAND',
       1T42,'CAL.',T49,'CAL.',T56,'CAL.',T63,'L.L.',T70,'R.L.',T77,'BODY',
       1T84,'BODY',T91,'TEMPERATURE',T109,'TEMPERATURE')
        WRITE(L,44)
   44   FORMAT(
       1 T2,'DATE',T8,'I.D.',T14,'REQ.',T21,'PRE.',T28,'POST',T35,'I.D.',
       1T42,'REQ.',T49,'PRE.',T56,'POST',T63,'CIRC.',T70,'CIRC.',T77,
       1'MASS',T84,'TEMP.',T91,'START',T98,'END',T105,'START',T112,'P1',
       1T119,'P4',T126,'P5')
        WRITE(L,902)
        DO 95 J=13,31
        IF (FID(J).EQ.9.0) FID(J)=100000.
   95   CONTINUE
        WRITE(L,99) (FID(J),J=13,31)
   99   FORMAT(T2,I3,T8,A2,T14,F4.2,T21,F4.2,T28,F4.2,T36,A2,T42,F4.2,
       2T49,F4.2,T56,F4.2,T63,F4.1,T70,F4.1,T77,F4.1,T84,F4.1,T91,F4.1,
       1T98,F4.1,T105,F4.1,T112,F4.1,T119,F4.1,T126,F4.1)
```

```
105.            WRITE(L,902)
106.            WRITE(L,120)
107.      120   FORMAT(' ELAPSED TIME END',T20,'P1',T30,'P2A',T40,'P2B',
108.           1T50,'P2C',T60,'P3',T70,'P4',T80,'P5')
109.            WRITE(L,125) (FID(I),I=33,39)
110.      125   FORMAT(T15,F10.4,T25,F10.4,T35,F10.4,T45,F10.4,T55,F10.4,
111.           1T65,F10.4,T75,F10.4,T85,'¢¢¢¢¢¢¢¢')
112.            WRITE(L,902)
113.            WRITE(L,130) KID(40)
114.      130   FORMAT(' RUN TYPE CODE=',I6)
115.            WRITE(L,902)
116.            WRITE(L,135) KID(41)
117.      135   FORMAT(' PROCESSING CODE=',I6)
118.            DO 500 IREC=2,NREC
119.      90    CALL BUFFER IN(IFILE,1,RDATA,375,ISTAT,ICNT)
120.      100   GO TO (100,110,700,50) ISTAT
121.      110   CONTINUE
122.            IF (ICNT.NE.375) OUTPUT(6) 'COUNT ERROR'; STOP
123.            IF (SSW(2)) GO TO 500
124.            WRITE(L,901)
125.            WRITE(L,150) IREC
126.      150   FORMAT(2X,'RECORD NUMBER ',I4)
127.            WRITE(L,902)
128.            WRITE(L,155)
129.      155   FORMAT(T11,'TIME',T23,'LEFT LEG',T37,'RIGHT LEG',
130.           * T56,'PLVC',T71,'LBNP')
131.            WRITE(L,900)
132.            WRITE(L,210)  RDATA(1,J),RDATA(2,J),RDATA(4,J),
133.           * RDATA(3,J),RDATA(5,J),J=1,75
134.      210   FORMAT(5(5X,F10.5))
135.      500   CONTINUE
136.            GO TO 1
137.      600   CONTINUE
138.            IF (L.EQ.6) CALL RLSBARN
139.            END FILE 7
140.            END FILE 7
141.            END FILE 7
142.            REWIND 7
143.            CALL RLSBARN
144.      700   IREC=IREC-1
145.            WRITE(6,710) IREC
146.      710   FORMAT (' # OF RECORDS= ',I3)
147.            GO TO 1
148.      40    OUTPUT(6) 'EOF ENCOUNTERED'; CALL RLSBARN
149.      50    OUTPUT(6) 'INPUT ERROR'; CALL RLSBARN
150.      900   FORMAT(1H0,7('¢'))
151.      901   FORMAT(1H1,7('¢'))
152.      902   FORMAT(1H2,7('¢'))
153.            END
```

```
!!JCP

!PAUSE INTERRUPT & KEY IN FSC
!PAUSE INTERRUPT & KEY IN FSC
!!KEY-IN
FSC

!0LOAD G0,(UDCB,15),(FILE,FP,BARNETT),(FORE,1C00),(TEMP,2000),(LIB,USER,SYSTEM)
!!0LOAD
:ASSIGN (F:4,CRA03)
:ASSIGN (F:5,TYA01),VFC
:ASSIGN (F:6,TYA01)
:ASSIGN (F:7,7TA81),VFC,(RECL,132),BCD
:ASSIGN (F:11,D1,F1),BIN
:ASSIGN (F:12,D1,F2),BIN
:ASSIGN (F:13,D1,F3),BIN
:ASSIGN (F:14,D1,F4),BIN
:ASSIGN (F:15,D1,F5),BIN
:ASSIGN (F:16,D1,F6),BIN
:ASSIGN (F:17,D1,F7),BIN
:ASSIGN (F:18,D1,F8),BIN
:ASSIGN (F:19,D1,F9),BIN
:ASSIGN (F:20,D1,F10),BIN
:ASSIGN (F:21,D1,F11),BIN
LOADING WAS COMPLETED
!!JCP

!FIN
!!BEGIN IDLE
```

APPENDIX M SAMPLE OF PRINTOUT PROGRAM OUTPUTS

RECORD NUMBER 1

EXPT#	RUN#	SUBJ#	DAY	HRS	MINS	SECS		EXPT CODE	
92	7	1	310	14	10	4	0	1	0

T1	T2	T3	T4	T5	T6	T7
300.00000	356.79980	416.79980	508.39990	896.79980	1196.79980	1496.79980

SKYLAB IV,CARR

MD92 SUMMARY DATA

JUL. DATE	LEFT LEG LVMS DATA BAND I.D.	CAL. REQ.	CAL. PRE.	CAL. POST	RIGHT LEG LVMS DATA BAND I.D.	CAL. REQ.	CAL. PRE.	CAL. POST	L.L. CIRC.	R.L. CIRC.	BODY MASS	BODY TEMP.	AMBIENT TEMPERATURE START	AMBIENT TEMPERATURE END	LBNPD TEMPERATURE START	LBNPD TEMPERATURE P1	LBNPD TEMPERATURE P4	LBNPD TEMPERATURE P5
310	CY	3.60	3.28	3.64	AP	3.00	3.35	2.94	33.7	34.2	00.0	97.0	71.1	71.9	73.0	73.1	73.2	73.2

ELAPSED TIME END P1 P2A P2B P2C P3 P4 P5
209.2000 356.0000 416.0000 508.3999 896.7998 1196.0000 1496.0000

RUN TYPE CODE= 1

PROCESSING CODE= 1

RECORD NUMBER 2

TIME	LEFT LEG	RIGHT LEG	PLVC	LBNP
.00000	-.02288	-.00970	-.21327	-.20841
.80000	-.02288	-.00970	-.21327	-.20841
1.60000	-.02288	-.00970	-.21327	-.20841
2.40000	-.02288	-.00970	-.21327	-.20841
3.20000	-.02288	-.00970	-.21327	-.20841
4.00000	-.06401	-.00970	-.25898	-.20841
4.80000	-.02288	-.00970	-.21327	-.20841
5.60000	-.02288	-.00970	-.21327	-.20841
6.40000	-.02288	-.00970	-.21327	-.20841
7.20000	-.02288	-.00970	-.21327	-.20841
8.00000	-.02288	-.00970	-.21327	-.20841
8.80000	-.02288	-.00970	-.21327	-.20841
9.60000	-.02288	-.00970	-.21327	-.20841
10.40000	-.02288	-.00970	-.21327	-.20841
11.20000	-.02288	-.00970	-.21327	-.20841
12.00000	-.02288	-.00970	-.21327	-.20841
12.80000	-.02288	-.00970	-.21327	-.20841
13.60000	-.02288	-.00970	-.21327	-.20841
14.40000	-.02288	-.00970	-.21327	-.20841
15.20000	-.02288	-.00970	-.21327	-.20841
16.00000	-.06401	-.00970	-.25898	-.20841
16.79999	-.02288	-.00970	-.21327	-.20841
17.59999	-.02288	-.00970	-.21327	-.20841
18.39999	-.02288	-.00970	-.21327	-.20841
19.20000	-.02288	-.00970	-.21327	-.20841
20.00000	-.06401	-.00970	-.25898	-.20841
20.79999	-.06401	-.00970	-.25898	-.20841
21.59999	-.06401	-.00970	-.25898	-.20841
22.39999	-.06401	-.00970	-.25898	-.20841
23.20000	-.06401	-.05389	-.20985	-.20841
24.00000	-.06401	-.05389	-.20985	-.20841
24.79999	-.06401	-.05389	-.20985	-.20841
25.59999	-.06401	-.05389	-.20985	-.20841
26.39999	-.06401	-.05389	-.20985	-.20841
27.20000	-.10522	-.05389	-.25569	-.20841
28.00000	-.10522	-.05389	-.25569	-.20841
28.79999	-.14652	-.05389	-.21222	-.20841
29.59999	-.10522	-.05389	-.23195	-.20841
30.39999	-.18791	-.05389	-.23195	-.20841
31.20000	-.18791	-.05389	-.23195	-.20841
32.00000	.22194	-.05389	-.23007	-.20841
32.79999	-.18791	-.05389	-.23195	-.20841
33.59999	-.31262	-.05389	-.23195	-.20841
34.39999	-.27096	-.05389	-.18562	-.20841
35.20000	-.27096	-.05389	-.18562	-.20841
36.00000	-.27096	-.05389	-.18562	-.20841
36.79999	-.22039	-.05389	-.21222	-.20841
37.59999	-.22039	-.05389	-.21222	-.20841
38.39999	-.22039	-.05389	-.21222	-.20841
39.20000	-.22039	-.05389	-.21222	-.20841
40.00000	-.22039	-.05389	-.21222	-.20841
40.79999	-.22039	-.05389	-.21216	-.20841
41.59999	-.22039	-.05389	-.21164	-.20841
42.39999	-.22039	-.05389	-.21157	-.20841
43.20000	-.27096	-.05389	-.18562	-.20841
44.00000	-.31262	-.05389	-.23195	-.20841
44.79999	-.27096	-.05389	-.18562	-.20841
45.59999	-.31262	-.05389	-.23195	-.20841
46.39999	-.27096	-.05389	-.18562	-.20841
47.20000	-.27096	-.05389	-.18562	-.20841
48.00000	-.27096	-.05389	-.18562	-.20841
48.79999	-.31262	-.05389	-.23195	-.20841
49.59999	-.31262	-.05389	-.23195	-.20841
50.39999	-.31262	-.05389	-.23195	-.20841
51.20000	-.31262	-.05389	-.23195	-.20841
52.00000	-.39622	-.05389	-.23195	-.20841
52.79999	-.35438	-.05389	-.18543	-.20841
53.59999	-.39622	-.05389	-.23195	-.20841
54.39999	-.39622	-.05389	-.23195	-.20841
55.20000	-.43815	-.05389	-.23195	-.20841
56.00000	-.43815	-.05389	-.23195	-.20841
56.79999	-.39622	-.05389	-.18533	-.20841
57.59999	-.43815	-.05389	-.23195	-.20841
58.39999	-.48017	-.05389	-.23195	-.20841
59.20000	-.48017	-.05389	-.23335	-.20841

APPENDIX N

PROGRAM FOR PRINTOUT of EOP and SLOPE INFORMATION

```
C***SKYLAB LVMS DATA(PLVC AND SLOPE) 6 PARAMETER DATA DECK
C***** SW(1) ON-OUTPUT TTY *****
C***** SW(2) ON-COMPUTE EOP PLVC *****
C***** SW(3) ON-COMPUTE S(1) SLOPES *****
C***** SW(4) ON-COMPUTE S(2) SLOPES *****
      DIMENSION RDATA(5,75),RID(42),ID(10),TID(7),KID(42)
      EQUIVALENCE (RID,KID)
      DIMENSION FDATA(5,1875),PLVC(7)
      DIMENSION DERV(1875),AVSL(10),SDSL(10),S(10),E(10)
      DATA FILL/8Z407FFFFF/
      NAME LIST
      LOGICAL SSW
      OUTPUT(6) 'NUMBER OF FILES=?'
      INPUT(5) NFILES
      IEND=NFILES+10
      OUTPUT(6) 'SKYLAB NO. = ?'
      INPUT(5) NSKYLB
      OUTPUT(6) 'ASTRONAUT NO. = ?'
      INPUT(5) NUMSUB
      IF(SSW(1)) L=6
      IF(.NOT.SSW(1)) L=7
      NREC=26
      IF(SSW(2)) GO TO 1
      IF(SSW(3)) GO TO 200
      IF(SSW(4)) GO TO 599
      OUTPUT (6) 'ERROR-SET SENSE SWITCHES';STOP
1     WRITE(L,8) NSKYLB,NUMSUB
8     FORMAT(T2,'SKYLAB NO. = ',I1,T20,'SUBJECT NO. = ',I1,
     1T50,'END OF PERIOD VALUES FOR PLVC');WRITE(L,992)
      WRITE(L,10)
10    FORMAT(T2,'RUN',T10,'DAY',T18,'CODE',T27,'DATA',
     1T34,'1',T42,'2A',T50,'2B',T58,'2C',T66,'3',T74,'4',T83,'5' )
      WRITE(L,992)
      DO 195 IFILE =11,IEND
      NFILE=IFILE-10
      OUTPUT(6) NFILE
      CALL BUFFER IN(IFILE,1,KID,42,ISTAT,ICNT)
2     GO TO (2,3,40,50) ISTAT
3     CONTINUE
      IF (ICNT.NE.42) OUTPUT(6) 'COUNT ERROR 1';STOP
C     UNPACK INTEGER DATA
      DO 20 I=1,5
      ID(2*I-1)=ISA(KID(I),-16)
      ID(2*I)=ISL(KID(I),16)
20    ID(2*I)=ISA(ID(2*I),-16)
      DO 30 I=1,7
30    TID(I)=RID(I+5)
      ISTART=1
      ISTOP=75
      DO 120 IREC=2,NREC
      CALL BUFFER IN(IFILE,1,RDATA,375,ISTAT,ICNT)
100   GO TO(100,110,122,50) ISTAT
110   CONTINUE
```

```
      IF (ICNT.NE.375) OUTPUT(6) 'COUNT ERROR 2';STOP
      J=0
      DO 115 N=ISTART,ISTOP
      J=J+1
      FDATA(1,N)=RDATA(1,J)
      FDATA(6,N)=RDATA(3,J)
  115 CONTINUE
      ISTART=ISTART+75
      ISTOP=ISTOP+75
  120 CONTINUE
      GO TO 124
  122 M=(IREC-1); WRITE(6,123) M
  123 FORMAT(T2,'EOF ENCOUNTERED AT NREC = ',T28,I2)
  124 DO 190 I=1,7
      K=0
      SVOL=0
      DO 180 N=1,1875
      T1=FDATA(1,N)-.2
      T2=FDATA(1,N)
      T3=FDATA(1,N)+.2
      IF(RID(32+I).EQ. FILL) GO TO 180
      IF((T1.LE.RID(32+I)).AND.(T3.GE.RID(32+I))) GO TO 129
      GO TO 180
C***** JJ = NO. OF SAMPLES OVER WHICH EOP PLVC IS AVERAGED *****
  129 JJ=10
  130 DO 150 NN=(N-JJ),N
      VOL=FDATA(6,NN)
      K=K+1
  150 SVOL=SVOL+VOL
      PLVC(I)=SVOL/K
  180 CONTINUE
  190 CONTINUE
      WRITE(L,191) ID(2),ID(4),ID(9),(PLVC(I),I=1,7)
  191 FORMAT(T2,I2,T10,I3,T19,I1,T27,'PLVC',T32,F6.2,
     1T40,F6.2,T48,F6.2,T56,F6.2,T64,F6.2,T72,F6.2,T80,F6.2)
      WRITE(L,990)
      REWIND IFILE
  195 CONTINUE
      WRITE(L,991)
      IF (SSW(3)) GO TO 200
      IF (.NOT. SSW(3)) GO TO 510
  200 IEND=NFILES+10
      WRITE(L,205) NSKYLB,NUMSUB
  205 FORMAT(T2,'SKYLAB NO. = ',I1,T20,'SUBJECT NO. = ',I1,
     1T50,'SLOPE OF LEG VOLUME');WRITE(L,992)
      WRITE(L,208)
  208 FORMAT(T2,'RUN',T10,'DAY',T18,'CODE',T27,'DATA',
     1       T41,'2A',T49,'2B',T57,'2C',T65,'3',T73,'4',T83,'5' )
      WRITE(L,992)
      DO 500 IFILE =11,IEND
      NFILE@IFILE-10
      OUTPUT(6) NFILE
      CALL BUFFER IN(IFILE,1,KID,42,ISTAT,ICNT)
  212 GO TO (212,213,42,50) ISTAT
  213 CONTINUE
```

```
109.          IF (ICNT.NE.42) OUTPUT(6) 'COUNT ERROR'; STOP
110.    C     UNPACK INTEGER DATA
111.          DO 220 I=1,5
112.          ID(2*I-1)=ISA(KID(I),-16)
113.          ID(2*I)=ISL(KID(I),16)
114.    220   ID(2*I)=ISA(ID(2*I),-16)
115.          DO 230 I=1,7
116.    230   TID(I)=RID(I+2)
117.          ISTART=1
118.          ISTOP=75
119.          DO 320 IREC=2,NREC
120.          CALL BUFFER IN(IFILE,1,RDATA,375,ISTAT,ICNT)
121.    300   GO TO(300,310,322,50) ISTAT
122.    310   CONTINUE
123.          IF (ICNT.NE.375) OUTPUT(6) 'COUNT ERROR'; STOP
124.          J=0
125.          DO 315 N=ISTART,ISTOP
126.          J=J+1
127.          FDATA(1,N)=RDATA(1,J)
128.          FDATA(6,N)=RDATA(3,J)
129.    315   CONTINUE
130.          ISTART=ISTART+75
131.          ISTOP=ISTOP+75
132.    320   CONTINUE
133.          GO TO 324
134.    322   M=(IREC-1);WRITE(6,323) M
135.    323   FORMAT(T2,'EOF ENCOUNTERED AT NREC = ',T28,I2)
136.    324   DO 410 I=1,6
137.          SQSL=0.0
138.          SUMSL=0.0
139.          XNUM=0.0
140.          DO 400 N=1,1875
141.          X1=FDATA(1,N)-.2
142.          X2=FDATA(1,N)
143.          X3=FDATA(1,N)+.2
144.          K=32+I
145.          IF(RID(K).EQ.FILL) GO TO 400
146.          IF(X1.LE.RID(K).AND.X3.GE.RID(K)) GO TO 340
147.          GO TO 400
148.    340   KK=20
149.          DO 400 NN=N+1,N+KK
150.          IF(FDATA(6,NN).EQ.FILL) GO TO 400
151.          IF(FDATA(6,NN-1).EQ.FILL) GO TO 400
152.          IF(FDATA(6,NN+1).EQ.FILL) GO TO 400
153.    350   DERV(NN)=((FDATA(6,NN+1)-FDATA(6,NN-1))/(2*.8))*60.0
154.    360   SQSL=SQSL+DERV(NN)**2
155.          XNUM=XNUM+1
156.          SUMSL=SUMSL+DERV(NN)
157.    400   CONTINUE
158.          AVSL(I)=SUMSL/XNUM
159.          SDSL(I)=SQRT((SQSL-(XNUM*AVSL(I)**2))/(XNUM-1.0))
160.    410   CONTINUE
161.          WRITE(L,420) ID(2),ID(4),ID(9),(AVSL(I),I=1,6)
162.    420   FORMAT(T2,I2,T10,I3,T19,I1,T27,'SLOPE(S1)',
163.         1T40,F5.3,T48,F5.3,T56,F5.3,T64,F5.3,          T72,F5.3,T80,F7.3)
164.          WRITE(L,992)
165.          REWIND IFILE
166.    500   CONTINUE
```

```
167.            KK=KK-1
168.            WRITE(L,430) KK
169.      430   FORMAT(T2,'NUMBER OF SAMPLES FOR SLOPE COMPUTATION = ',T44,I2)
170.            WRITE(L,990)
171.      510   CONTINUE
172.            WRITE(L,991)
173.      599   CONTINUE
174.            IF(SSW(4)) GO TO 600
175.            IF(.NOT.SSW(4)) GO TO 900
176.      600   IEND=NFILES+10
177.            WRITE(L,602) NSKYLB,NUMSUB
178.      602   FORMAT(T2,'SKYLAB NO. = ',I1,T20,'SUBJECT NO. = 'I1,
179.           1T50,'SLOPE OF LEG VOLUME');WRITE(L,992)
180.            WRITE(L,605)
181.      605   FORMAT(T2,'RUN',T8,'DAY',T14,'CODE',T21,'DATA',
182.           1T33,'1',T41,'2A',T49,'2B',T57,'2C',T65,'3',T73,'4',T83,'5' )
183.            WRITE(L,992)
184.            DO 900 IFILE =11,IEND
185.            NFILE=IFILE-10
186.            OUTPUT(6) NFILE
187.            CALL BUFFER IN(IFILE,1,KID,42,ISTAT,ICNT)
188.      612   GO TO (612,613,44,50) ISTAT
189.      613   CONTINUE
190.            IF (ICNT.NE.42) OUTPUT(6) 'COUNT ERROR'; STOP
191.      C     UNPACK INTEGER DATA
192.            DO 620 I=1,5
193.            ID(2*I-1)=ISA(KID(I),-16)
194.            ID(2*I)=ISL(KID(I),16)
195.      620   ID(2*I)=ISA(ID(2*I),-16)
196.            DO 630 I=1,7
197.      630   TID(I)=RID(I+5)
198.            ISTART=1
199.            ISTOP=75
200.            DO 720 IREC=2,NREC
201.            CALL BUFFER IN(IFILE,1,RDATA,375,ISTAT,ICNT)
202.      700   GO TO (700,710,722,50) ISTAT
203.            GO TO 724
204.      710   CONTINUE
205.            IF (ICNT.NE.375) OUTPUT(6) 'COUNT ERROR'; STOP
206.            J=0
207.            DO 715 N=ISTART,ISTOP
208.            J=J+1
209.            FDATA(1,N)=RDATA(1,J)
210.            FDATA(6,N)=RDATA(3,J)
211.      715   CONTINUE
```

```
212.            ISTART=ISTART+75
213.            ISTOP=ISTOP+75
214.      720   CONTINUE
215.      722   M=(IREC-1);WRITE(6,723) M
216.      723   FORMAT(T2,'EOF ENCOUNTERED AT NREC = ',T28,I2)
217.      724   DO 810 I=1,7
218.            SQSL=0.0
219.            SUMSL=0.0
220.            XNUM=0.0
221.            DO 800 N=1,1875
222.            X1=FDATA(1,N)-.2
223.            X2=FDATA(1,N)
224.            X3=FDATA(1,N)+.2
225.            K=32+I
226.            IF(RID(K).EQ.FILL) GO TO 800
227.            IF(X1.LE.RID(K).AND.X3.GE.RID(K)) GO TO 740
228.            GO TO 800
229.      740   IP1=290
230.            IP2=38
231.            IP3=38
232.            IP4=75
233.            IP5=225
234.            IP6=225
235.            IP7=225
236.            IF(I.EQ.1) II=N-IP1
237.            IF(I.EQ.2) II=N-IP2
238.            IF(I.EQ.3) II=N-IP3
239.            IF(I.EQ.4) II=N-IP4
240.            IF(I.EQ.5) II=N-IP5
241.            IF(I.EQ.6) II=N-IP6
242.            IF(I.EQ.7) II=N-IP7
243.            DO 800 NN=II,N
244.            IF(FDATA(6,NN).EQ.FILL) GO TO 800
245.            IF(FDATA(6,NN-1).EQ.FILL) GO TO 800
246.            IF(FDATA(6,NN+1).EQ.FILL) GO TO 800
247.      750   DERV(NN)=((FDATA(6,NN+1)-FDATA(6,NN-1))/(2*.8))*60.0
248.      760   SQSL=SQSL+DERV(NN)**2
249.            XNUM=XNUM+1
250.            SUMSL=SUMSL+DERV(NN)
251.      800   CONTINUE
252.            AVSL(I)=SUMSL/XNUM
253.            SDSL(I)=SQRT((SQSL-(XNUM*AVSL(I)**2))/(XNUM-1.0))
254.      810   CONTINUE
255.            WRITE(L,820) ID(2),ID(4),ID(9),(AVSL(I),I=1,7)
256.      820   FORMAT(T2,I2,T8,I3,T14,I1,T21,'SLOPE(S2)',T32,F5.3,
257.           1T40,F5.3,T48,F5.3,T56,F5.3,T64,F5.3,          T72,F5.3,T80,F7.3)
258.            WRITE(L,992)
259.            WRITE(L,990)
260.            REWIND IFILE
261.      900   CONTINUE
```

```
   262.           WRITE(L,830)
   263.     830   FORMAT(T2,'NUMBER OF SAMPLES FOR PERIODS INDICATED');WRITE(L,992)
   264.           WRITE(L,840) IP1,IP2,IP3,IP4,IP5,IP6,IP7
   265.     840   FORMAT(T2,'P1 = ',I3,T11,'P2A = ',I2,T20,'P2B = ',I2,T29,
   266.          1'P2C = ',I2,T38,'P3 = ',I3,T47,'P4 = ',I3,T57,'P5 = ',I3)
   267.     910   CONTINUE
   268.           IF(L.EQ.6) CALL RLSBARN
   269.           END FILE 7
   270.           END FILE 7
   271.           END FILE 7
   272.           REWIND 7
   273.           CALL RLSBARN
   274.     40    OUTPUT(6) 'EOF ENCOUNTERED-EOP PLWC ID'; STOP
   275.     41    OUTPUT(6) 'EOF ENCOUNTERED-EOP PLWC RDATA'; STOP
   276.     42    OUTPUT(6) 'EOF ENCOUNTERED-SLOPE(S1) ID'; STOP
   277.     43    OUTPUT(6) 'EOF ENCOUNTERED-SLOPE(S1) RDATA';STOP
   278.     44    OUTPUT(6) 'EOF ENCOUNTERED-SLOPE(S2) ID';STOP
   279.     45    OUTPUT(6) 'EOF EMCOUNERED-SLOPE(S2) RDATA';STOP
   280.     50    OUTPUT(6) 'INPUT ERROR'; STOP
   281.       990 FORMAT(1H0,7('¢'))
   282.       991 FORMAT(1H1,7('¢'))
   283.       992 FORMAT(1H2,7('¢'))
   284.       998 FORMAT(1H8,7('¢'))
   285.           END
!!JCP

!PAUSE INTERRUPT & KEY IN FSC
!PAUSE INTERRUPT & KEY IN FSC
!!KEY-IN
FSC

!LOAD GO,(UDCB,15),(FILE,FP,BARNETT),(FORE,1C00),(TEMP,2000),(LIB,USER,SYSTEM)
!!LOAD
:ASSIGN (F:4,CRA03)
:ASSIGN (F:5,TYA01),VFC
:ASSIGN (F:6,TYA01)
:ASSIGN (F:7,7TA81),VFC,(RECL,132),BCD
:ASSIGN (F:11,D1,H1),BIN
:ASSIGN (F:12,D1,H2),BIN
:ASSIGN (F:13,D1,H3),BIN
:ASSIGN (F:14,D1,H4),BIN
:ASSIGN (F:15,D1,H5),BIN
:ASSIGN (F:16,D1,H6),BIN
:ASSIGN (F:17,D1,H7),BIN
:ASSIGN (F:18,D1,H8),BIN
:ASSIGN (F:19,D1,H9),BIN
LOADING WAS COMPLETED
!!JCP

!FIN
!!BEGIN IDLE
```


USAFA/USAF Academy, CO - Jun 74