NASA CR-
141472

Final Technical Report
on the
Surface Electrical Properties
Experiment

report prepared by

The University of Toronto

SEPTEMBER 1974

and submitted to MIT
in fulfillment of the
subcontract on NASA
contract NAS9-11540

(NASA-CR-141472) SURFACE ELECTRICAL PROPERTIES EXPERIMENT, PART 2 Final Technical Report (Toronto Univ.) 290 p HC $8.75 CSCL 03B

N75-15571

Unclas
G3/91 06657

Personnel

D.W. Strangway
A.P. Annan
J.D. Redman
J.R. Rossiter
J.A. Rylaarsdam
R.D. Watts

Part II of III Parts

A-3 (ii)

Digital Processing
2) Science Data Processing

R.D. Watts

SCIENCE DATA PROCESSING

The format of science (VCO) data from the acquisition system has been given in Figure 7 of J.D. Redman's report on data digitization. These data are processed in four major stages: 1) frequency and quality determination, 2) merging, 3) demultiplexing, 4) calibration.

In the notation of Figure 7 of Redman's report, one VCO measurement is recorded by 12 digits as follows

$N = (N_1\ N_2\ N_3)_{10}$ = number of 5.2 KHz cycles counted

$V = (V_1\ V_2\ V_3\ V_4\ V_5)_{10}$ = duration of M VCO cycles (μsec)

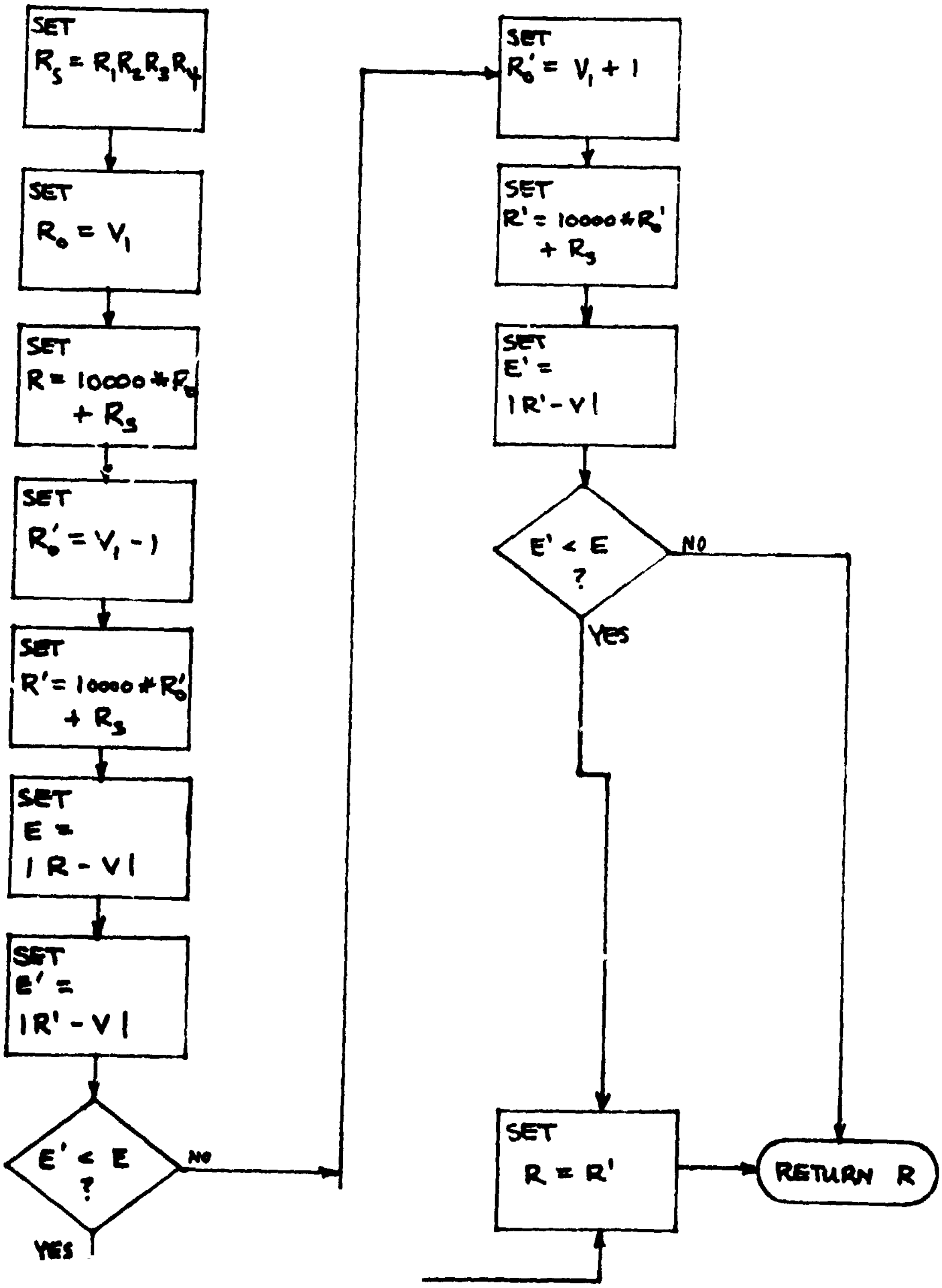
$R = (\{R_0\}\ R_1\ R_2\ R_3\ R_4)_{10}$ = duration of N 5.2 KHz cycles

Note that only the four low-order digits of R are recorded on tape. The high-order digit R_0 must be inferred from V and the operational characteristics of the DAS.

The determination of R_0 is based on the DAS design criterion that V and R should not differ by more than 1/500 sec $\cong$ 200 μsec. Since R_0 represents the 10000's unit of μsec, R_0 can conceivably be only one less, one greater, or equal to V_1. The algorithm diagrammed in Figure 1 bases the choice of one of these on the criterion of minimizing the difference between V and R. This function was performed in the routine FREQ.

The foregoing discussion assumed that the DAS output

FIGURE 1 DETERMINATION OF R_0



conformed to specifications. Becauce this was not always true, a syntax analysis was performed by the FREQ routine. Four types of errors were recognized, and an error indicator was set according to the seriousness of the errors which were found. The error indicator was set to the sum of the following individual error levels:

level 0 : no errors

1 : the measured VCO frequency was outside the range 300 - 3000 Hz., which spans the expected frequencies.

2 : the measured reference frequency was more than 100 Hz different from the mean reference frequency of 5213 Hz.

4 : the measured periods of VCO and reference cycles differed by more than 210 μsec (1/5200 sec + 9.2%).

8 : one or more of the N, V, or R counters read zero.

Error level 8 was a terminal error, for which the frequency could not be computed and was set to zero. Error level 4 indicated a malfunction of the zero-crossing detector for the reference signal or a malfunction in the start/stop circuitry for reference period counting. However, the effects of a level 4 error could be quite small, especially for low VCO frequencies. Error level 2 indicated either a large random fluctuation in the 5.2 KHz multivibrator in the DSEA, a tape-speed variation, or a counting problem. The tape-speed variation was probably the most frequent cause of this type of error, in which case the VCO frequency would still have been correctly determined. Error level 1 simply chopped off the allowable range of frequencies to a range spanning those observed in instrument calibration. VCO frequencies below 300 were set to 300, and those over 3000 were set to 3000.

For error levels below 8, the VCO frequency was computed by the following formulas:

$$T_{VCO} = \frac{V}{4} \times 10^{-6} \text{ sec.}$$

$$T_{REF} = \frac{R}{N} \times 10^{-6} \text{ sec.}$$

$$f_{VCO} = T_{VCO}^{-1} = \frac{4}{V} \times 10^{6} \text{ Hz}$$

$$f_{REF} = \frac{N}{R} \times 10^{6} \text{ Hz}$$

$$f_{VCO} - \text{CORRECTED} = \frac{5213}{f_{REF}} \quad f_{VCO}$$

$$= \frac{5213 \times 4 \times R}{V \times N} \quad \text{Hz}$$

The source of the correction frequencv 5213 has been discussed in Redman's report on data digitization. It represents the mean actual 5.2 KHz reference frequency.

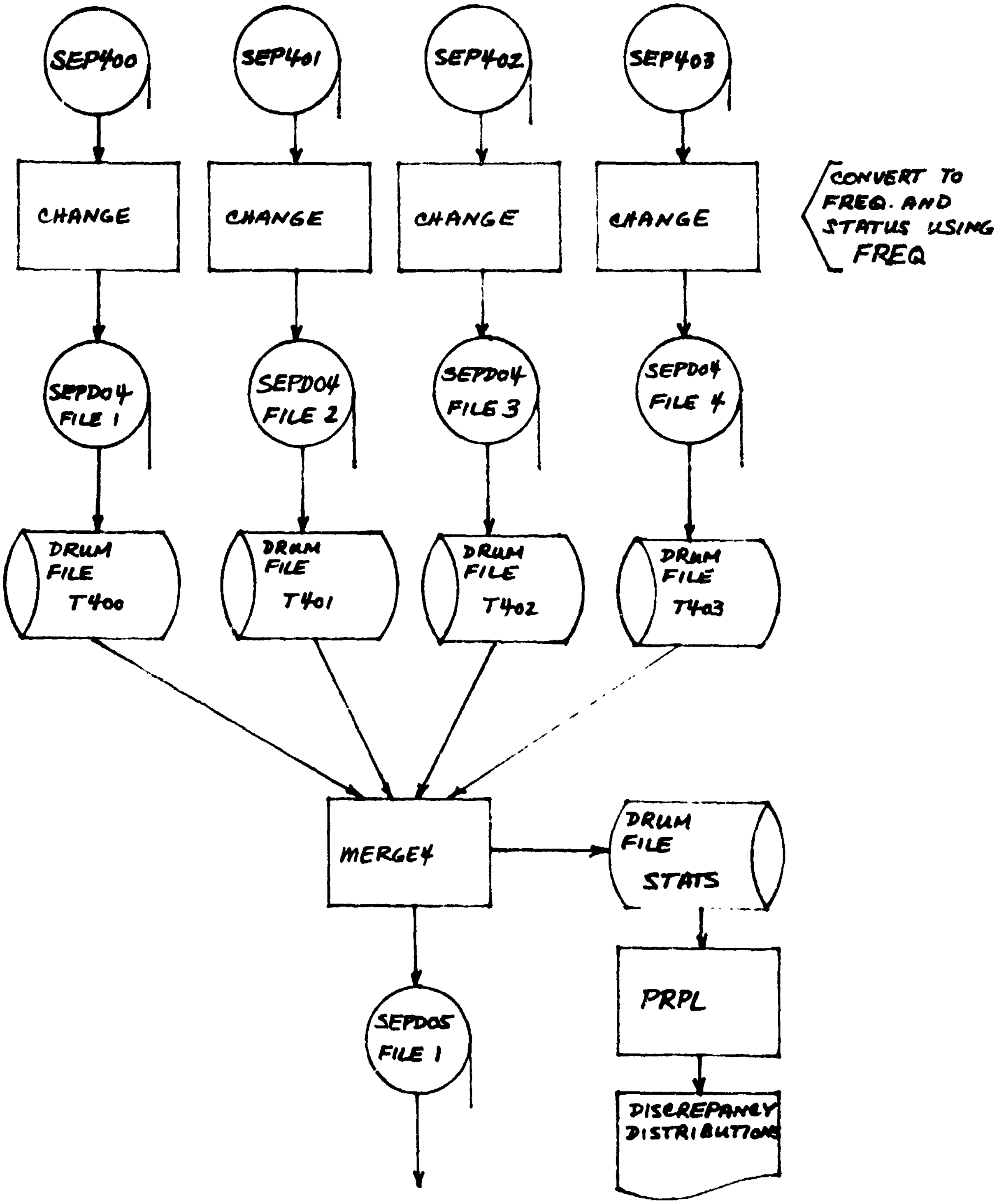
Each of the four final science tapes (SEP400-403) was processed with the program CHANGE, which used FREQ to change all VCO readings to frequency-status pairs. The output data were stored as four files on tape SEPDO4, as shown in Figure 2. Since there were two bad records on tape SEP403, the frequency values from these records were set to zero, and the error status values were set to 16.

The drum-file copies of CHANGE output (T400-403) were merged by the program MERGE4. Output was stored in the first file of tape SEPDO5. Merging was performed according to the following rules.

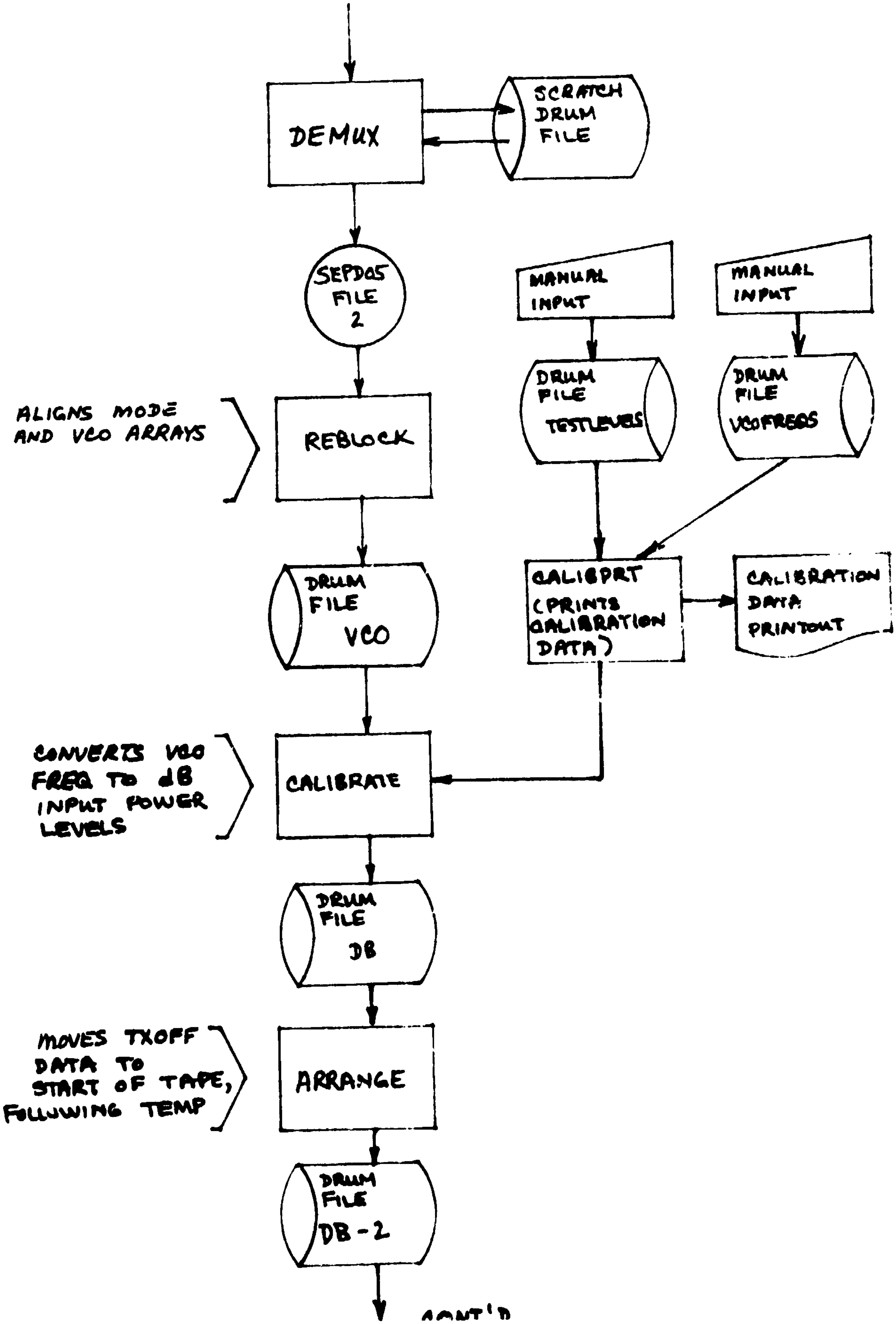
1. Taking the four corresponding readings from files T400-403, reject those not having the lowest error status.
2. Search the values with lowest error status for the pair of values with the smallest discrepancy.
3. Average the values from the pair with smallest discrepancy. Use this value.

The reason for rule 1 is simply to use the most trustworthy values available, judged according to the seriousness of syntax errors. Rule 2 guarantees that the most repeatable measurements are used. Rule 3, assuming that the errors of measurement are Gaussian (which they are only approximately),

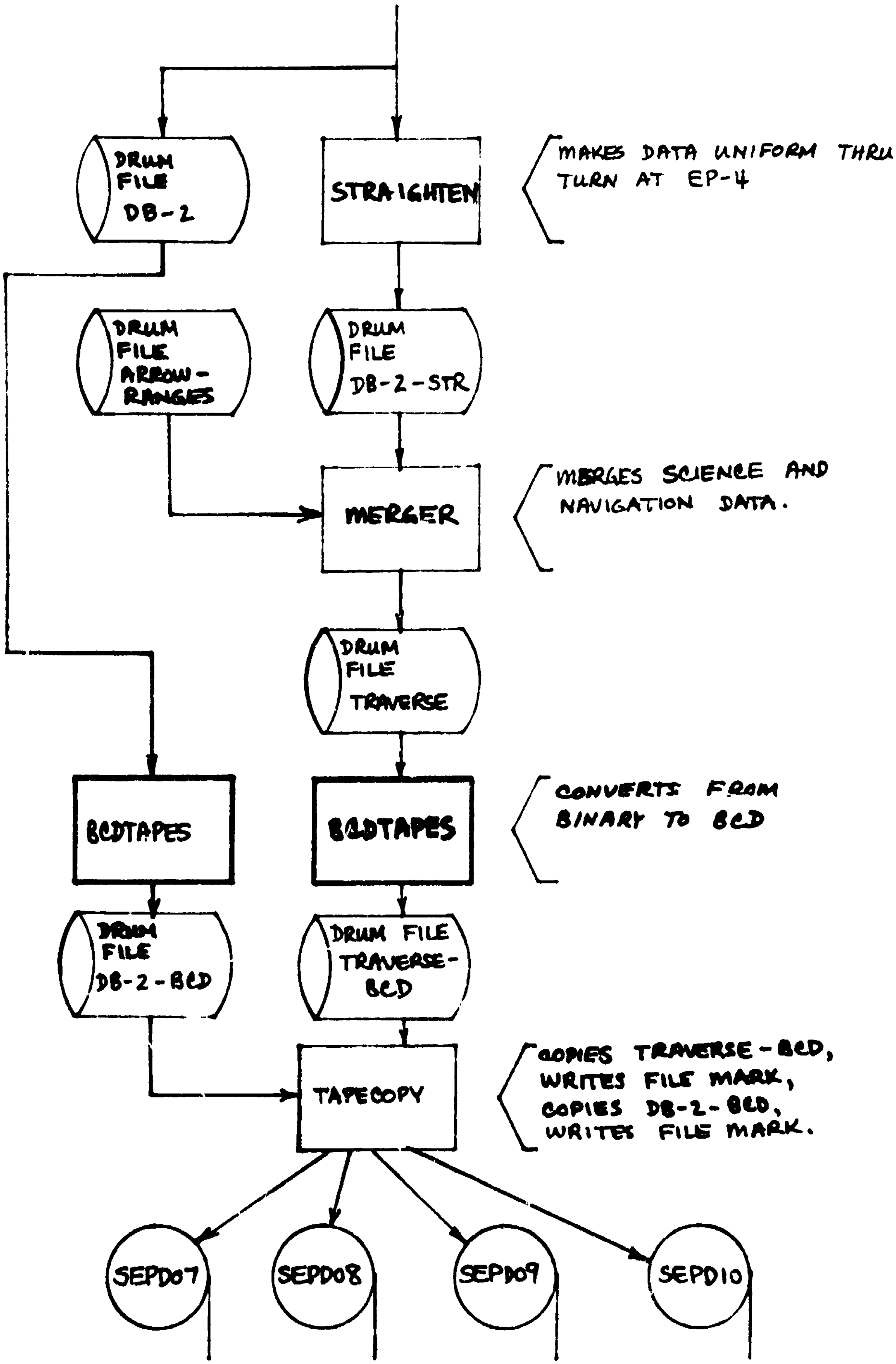
FIGURE 2 VCO DATA PROCESSING



CONT'D



DEMUX
SCRATCH DRUM FILE
SEPD05 FILE 2
MANUAL INPUT
MANUAL INPUT
ALIGNS MODE AND VCO ARRAYS
REBLOCK
DRUM FILE TESTLEVELS
DRUM FILE VCOFREQS
DRUM FILE VCO
CALIBPRT (PRINTS CALIBRATION DATA)
CALIBRATION DATA PRINTOUT
CONVERTS VCO FREQ TO dB INPUT POWER LEVELS
CALIBRATE
DRUM FILE DB
MOVES TXOFF DATA TO START OF TAPE, FOLLOWING TEMP
ARRANGE
DRUM FILE DB-2
CONT'D


DRUM FILE DB-2
STRAIGHTEN
MAKES DATA UNIFORM THRU TURN AT EP-4
DRUM FILE ARROW-RANGES
DRUM FILE DB-2-STR
MERGER
MERGES SCIENCE AND NAVIGATION DATA.
DRUM FILE TRAVERSE
BCDTAPES
BCDTAPES
CONVERTS FROM BINARY TO BCD
DRUM FILE DB-2-BCD
DRUM FILE TRAVERSE-BCD
TAPECOPY
COPIES TRAVERSE-BCD, WRITES FILE MARK, COPIES DB-2-BCD, WRITES FILE MARK.
SEPD07
SEPD08
SEPD09
SEPD10

reduces the measurment error by a factor $1/\sqrt{2}$.

MERGE4 provided the following ancillary functions: 1) printout of all non-zero status-value data which were used, 2) distributions of frequency discrepancies for the values from rule 3 above, broken down by 100 Hz intervals, with 5 Hz. granularity. These are shown in Figure 3. 3) printouts of all points where the discrepancy of accepted values was large (Figure 4).

The discrepancy distributions from MERGE4 were stored in the drum file STATS. Program PRPL graphed the distributions as histograms on the line printer (Figure 3). These graphs show the frequency of occurrence of various discrepancies, where the source of the first accepted value preceeds the source of the second accepted value in the sequence SEP400, 401, 402, 403, and the discrepancy is the first accepted value minus the second.

Science data output from MERGE4 were still in multiplexed form (i.e., data of various types were mixed together in a known sequence). The DEMUX program demultiplexed the data, collecting all data of one type (e.g. 4 MHz NS X) into a single array. Because the memory capacity of the computer was insufficient to hold all the data, the demultiplexing was done in two stages. Input data were first demultiplexed in core,

Figure 3

Distributions of discrepancies between the two accepted values (i.e. the values with lowest error level and smallest absolute discrepancy). Data are grouped according to the mean frequency, in 100 Hz intervals, and plotted with 5 Hz granularity.

FREQUENCY INTERVAL (100HZ.) STARTS AT 00 .

-100	.00
-95	.00
-90	.00
-85	.00
-80	.00
-75	.00
-70	.00
-65	.00
-60	.00
-55	.00
-50	.01
-45	.00
-40	.00
-35	.00
-30	.01
-25	.03
-20	.16
-15	.19
-10	.33
-5	1.00
0	95.37
5	.84
10	.23
15	.13
20	.11
25	.02
30	.01
35	.01
40	.00
45	.00
50	.00
55	.00
60	.00
65	.00
70	.00
75	.00
80	.00
85	.00
90	.00
95	.00
100	.00

AVERAGE ABSOLUTE ERROR = .3HZ.

```
FREQUENCY INTERVAL (10CHZ.) STARTS AT  400HZ.
     ------------------------------------------------------------------------
-100I                                                                  I   .00
 -95I                                                                  I   .00
 -90I                                                                  I   .00
 -85I                                                                  I   .00
 -80I                                                                  I   .00
 -75I                                                                  I   .00
 -70I                                                                  I   .00
 -65I                                                                  I   .00
 -60I                                                                  I   .00
 -55I                                                                  I   .00
 -50I                                                                  I   .00
 -45I                                                                  I   .00
 -40I                                                                  I   .00
 -35I                                                                  I   .00
 -30I                                                                  I   .00
 -25I                                                                  I   .00
 -20I                                                                  I   .03
 -15I                                                                  I   .02
 -10I                                                                  I   .03
  -5I                                                                  I   .03
   0I********************************************************************99.73
   5I                                                                  I   .06
  10I                                                                  I   .04
  15I                                                                  I   .01
  20I                                                                  I   .01
  25I                                                                  I   .00
  30I                                                                  I   .00
  35I                                                                  I   .00
  40I                                                                  I   .00
  45I                                                                      .00
  50I                                                                  I   .00
  55I                                                                  I   .00
  60I                                                                  I   .00
  65I                                                                  I   .00
  70I                                                                  I   .00
  75I                                                                  I   .00
  80I                                                                  I   .00
  85I                                                                  I   .00
  90I                                                                  I   .00
  95I                                                                  I   .00
 100I                                                                  I   .00
     ------------------------------------------------------------------------
AVERAGE ABSOLUTE ERROR =    .CHZ.
```

```
FREQUENCY INTERVAL (100HZ.) STARTS AT  500HZ.
   ----------------------------------------------------------------
-100I                                                              I  .00
 -95I                                                              I  .00
 -90I                                                              I  .00
 -85I                                                              I  .00
 -80I                                                              I  .00
 -75I                                                              I  .00
 -70I                                                              I  .00
 -65I                                                              I  .00
 -60I                                                              I  .00
 -55I                                                              I  .00
 -50I                                                              I  .00
 -45I                                                              I  .01
 -40I                                                              I  .00
 -35I                                                              I  .00
 -30I                                                              I  .00
 -25I                                                              I  .00
 -20I                                                              I  .00
 -15I                                                              I  .00
 -10I                                                              I  .03
  -5I                                                              I  .14
   0I**************************************************************99.74
   5I                                                              I  .06
  10I                                                              I  .01
  15I                                                              I  .00
  20I                                                              I  .00
  25I                                                              I  .00
  30I                                                              I  .00
  35I                                                              I  .00
  40I                                                              I  .00
  45I                                                              I  .00
  50I                                                              I  .00
  55I                                                              I  .00
  60I                                                              I  .00
  65I                                                              I  .00
  70I                                                              I  .00
  75I                                                              I  .00
  80I                                                              I  .00
  85I                                                              I  .00
  90I                                                              I  .00
  95I                                                              I  .00
 100I                                                              I  .00
   ----------------------------------------------------------------
AVERAGE ABSOLUTE ERROR =    .0HZ.
```

```
FREQUENCY INTERVAL (100HZ.) STARTS AT  500HZ.
     ------------------------------------------------------------------------------
-100I                                                                          I  .00
 -95I                                                                          I  .00
 -90I                                                                          I  .00
 -85I                                                                          I  .00
 -80I                                                                          I  .00
 -75I                                                                          I  .00
 -70I                                                                          I  .00
 -65I                                                                          I  .00
 -60I                                                                          I  .00
 -55I                                                                          I  .00
 -50I                                                                          I  .00
 -45I                                                                          I  .01
 -40I                                                                          I  .00
 -35I                                                                          I  .00
 -30I                                                                          I  .00
 -25I                                                                          I  .00
 -20I                                                                          I  .00
 -15I                                                                          I  .00
 -10I                                                                          I  .03
  -5I                                                                          I  .14
   0I***************************************************************************99.74
   5I                                                                          I  .06
  10I                                                                          I  .01
  15I                                                                          I  .00
  20I                                                                          I  .00
  25I                                                                          I  .00
  30I                                                                          I  .00
  35I                                                                          I  .00
  40I                                                                          I  .00
  45I                                                                          I  .00
  50I                                                                          I  .00
  55I                                                                          I  .00
  60I                                                                          I  .00
  65I                                                                          I  .00
  70I                                                                          I  .00
  75I                                                                          I  .00
  80I                                                                          I  .00
  85I                                                                          I  .00
  90I                                                                          I  .00
  95I                                                                          I  .00
 100I                                                                          I  .00
     ------------------------------------------------------------------------------
AVERAGE ABSOLUTE ERROR =    .0HZ.
```

FREQUENCY INTERVAL (100HZ.) STARTS AT 500HZ.

-100	I	.00
-95	I	.00
-90	I	.00
-85	I	.00
-80	I	.00
-75	I	.00
-70	I	.00
-65	I	.00
-60	I	.00
-55	I	.00
-50	I	.00
-45	I	.00
-40	I	.00
-35	I	.00
-30	I	.00
-25	I	.00
-20	I	.02
-15	I	.00
-10	I	.02
-5	I	.02
0	**99	99.95
5	I	.00
10	I	.00
15	I	.00
20	I	.00
25	I	.00
30	I	.00
35	I	.00
40	I	.00
45	I	.00
50	I	.00
55	I	.00
60	I	.00
65	I	.00
70	I	.00
75	I	.00
80	I	.00
85	I	.00
90	I	.00
95	I	.00
100	I	.00

AVERAGE ABSOLUTE ERROR = .0HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 700HZ.

-100	.00
-95	.00
-90	.00
-85	.00
-80	.00
-75	.00
-70	.00
-65	.00
-60	.00
-55	.00
-50	.00
-45	.00
-40	.00
-35	.00
-30	.00
-25	.00
-20	.00
-15	.00
-10	.00
-5	.00
0	99.93
5	.00
10	.00
15	.00
20	.00
25	.00
30	.00
35	.00
40	.00
45	.00
50	.00
55	.00
60	.00
65	.00
70	.00
75	.00
80	.00
85	.00
90	.00
95	.00
100	.00

AVERAGE ABSOLUTE ERROR = .0HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 800HZ.

```
------------------------------------------------------------
-100I                                          I   .00
 -95I                                          I   .00
 -90I                                          I   .00
 -85I                                          I   .00
 -80I                                          I   .00
 -75I                                          I   .00
 -70I                                          I   .00
 -65I                                          I   .00
 -60I                                          I   .00
 -55I                                          I   .00
 -50I                                          I   .00
 -45I                                          I   .00
 -40I                                          I   .00
 -35I                                          I   .00
 -30I                                          I   .00
 -25I                                          I   .03
 -20I                                          I   .00
 -15I                                          I   .00
 -10I                                          I   .00
  -5I                                          I   .05
   0I*******************************************99.89
   5I                                          I   .03
  10I                                          I   .00
  15I                                          I   .00
  20I                                          I   .00
  25I                                          I   .00
  30I                                          I   .00
  35I                                          I   .00
  40I                                          I   .00
  45I                                          I   .00
  50I                                          I   .00
  55I                                          I   .00
  60I                                          I   .00
  65I                                          I   .00
  70I                                          I   .00
  75I                                          I   .00
  80I                                          I   .00
  85I                                          I   .00
  90I                                          I   .00
  95I                                          I   .00
 100I                                          I   .00
------------------------------------------------------------
```

AVERAGE ABSOLUTE ERROR =   .0HZ.

```
FREQUENCY INTERVAL (100HZ.) STARTS AT  900HZ.
  ---------------------------------------------------------------------------------------------
-100I                                                                                         I  .00
 -95I                                                                                         I  .00
 -90I                                                                                         I  .00
 -85I                                                                                         I  .00
 -80I                                                                                         I  .00
 -75I                                                                                         I  .00
 -70I                                                                                         I  .00
 -65I                                                                                         I  .00
 -60I                                                                                         I  .00
 -55I                                                                                         I  .00
 -50I                                                                                         I  .00
 -45I                                                                                         I  .00
 -40I                                                                                         I  .00
 -35I                                                                                         I  .00
 -30I                                                                                         I  .00
 -25I                                                                                         I  .00
 -20I                                                                                         I  .00
 -15I                                                                                         I  .00
 -10I                                                                                         I  .00
  -5I                                                                                         I  .05
   0I*******************************************************************************************99.36
   5I                                                                                         I  .05
  10I                                                                                         I  .05
  15I                                                                                         I  .00
  20I                                                                                         I  .00
  25I                                                                                         I  .00
  30I                                                                                         I  .00
  35I                                                                                         I  .00
  40I                                                                                         I  .00
  45I                                                                                         I  .00
  50I                                                                                         I  .00
  55I                                                                                         I  .00
  60I                                                                                         I  .00
  65I                                                                                         I  .00
  70I                                                                                         I  .00
  75I                                                                                         I  .00
  80I                                                                                         I  .00
  85I                                                                                         I  .00
  90I                                                                                         I  .00
  95I                                                                                         I  .00
 100I                                                                                         I  .00
  ---------------------------------------------------------------------------------------------
AVERAGE ABSOLUTE ERROR =    .0HZ.
```

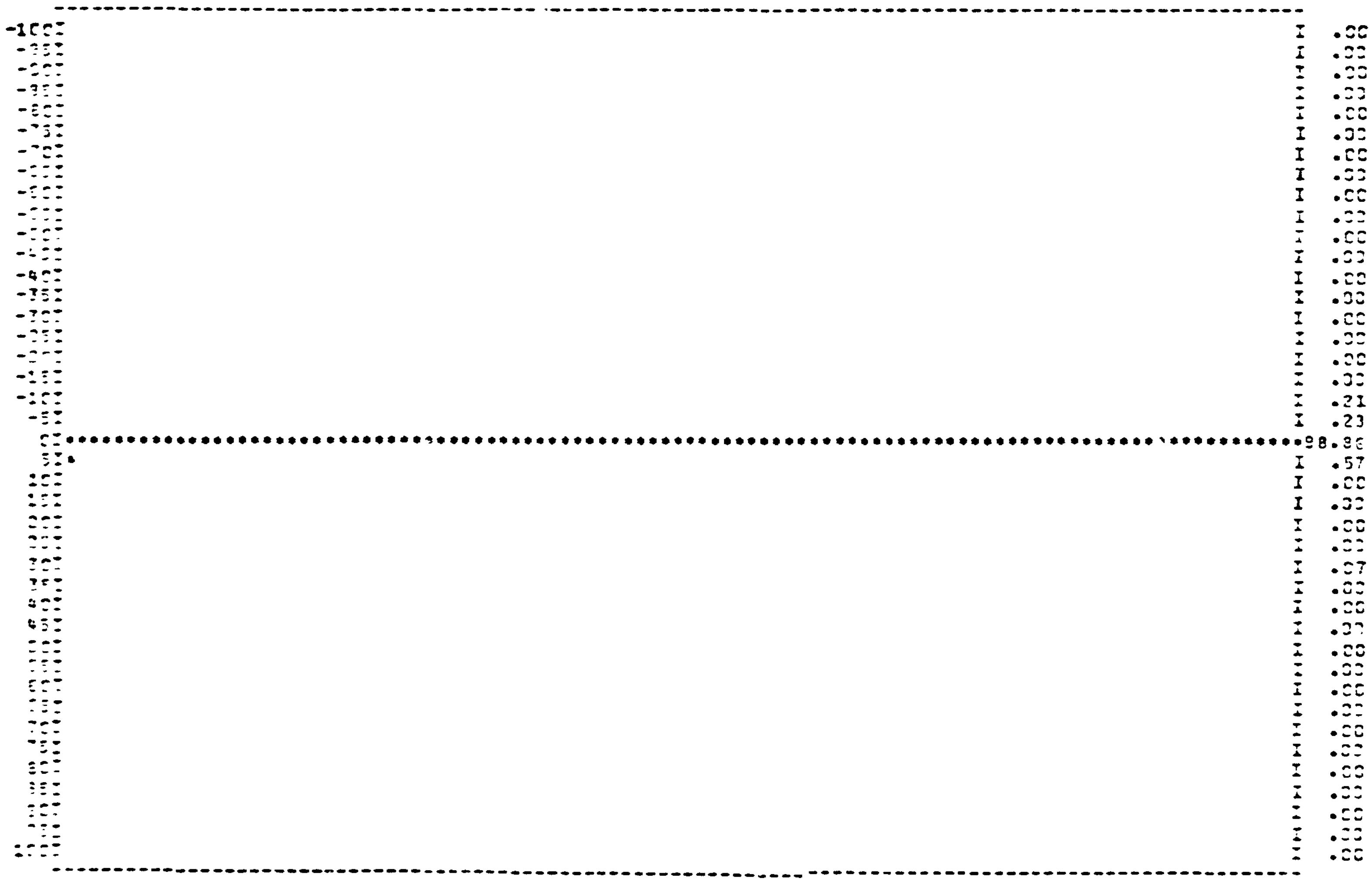
```
FREQUENCY INTERVAL (100HZ.) STARTS AT 1000HZ.
    --------------------------------------------------------------------------
-100I                                                                        I   .00
 -95I                                                                        I   .00
 -90I                                                                        I   .00
 -85I                                                                        I   .00
 -80I                                                                        I   .00
 -75I                                                                        I   .00
 -70I                                                                        I   .00
 -65I                                                                        I   .00
 -60I                                                                        I   .00
 -55I                                                                        I   .00
 -50I                                                                        I   .00
 -45I                                                                        I   .00
 -40I                                                                        I   .00
 -35I                                                                        I   .00
 -30I                                                                        I   .00
 -25I                                                                        I   .00
 -20I                                                                        I   .00
 -15I                                                                        I   .00
 -10I                                                                        I   .00
  -5I                                                                        I   .03
   0I************************************************************************* 99.80
   5I                                                                        I   .03
  10I                                                                        I   .04
  15I                                                                        I   .00
  20I                                                                        I   .00
  25I                                                                        I   .00
  30I                                                                        I   .00
  35I                                                                        I   .00
  40I                                                                        I   .00
  45I                                                                        I   .00
  50I                                                                        I   .00
  55I                                                                        I   .00
  60I                                                                        I   .00
  65I                                                                        I   .00
  70I                                                                        I   .00
  75I                                                                        I   .00
  80I                                                                        I   .00
  85I                                                                        I   .00
  90I                                                                        I   .00
  95I                                                                        I   .00
 100I                                                                        I   .00
    --------------------------------------------------------------------------
AVERAGE ABSOLUTE ERROR =    .0HZ.
```

FREQUENCY INTERVAL (100HZ.) STARTS AT 1100HZ.

```
    ---------------------------------------------------------------------------------------------------
-100I                                                                                                 I   .00
 -95I                                                                                                 I   .00
 -90I                                                                                                 I   .00
 -85I                                                                                                 I   .00
 -80I                                                                                                 I   .00
 -75I                                                                                                 I   .00
 -70I                                                                                                 I   .00
 -65I                                                                                                 I   .00
 -60I                                                                                                 I   .00
 -55I                                                                                                 I   .00
 -50I                                                                                                 I   .00
 -45I                                                                                                 I   .00
 -40I                                                                                                 I   .00
 -35I                                                                                                 I   .00
 -30I                                                                                                 I   .00
 -25I                                                                                                 I   .00
 -20I                                                                                                 I   .00
 -15I                                                                                                 I   .00
 -10I                                                                                                 I   .00
  -5I                                                                                                 I   .12
   0I*************************************************************************************************99.71
   5I                                                                                                 I   .06
  10I                                                                                                 I   .06
  15I                                                                                                 I   .00
  20I                                                                                                 I   .00
  25I                                                                                                 I   .06
  30I                                                                                                 I   .00
  35I                                                                                                 I   .00
  40I                                                                                                 I   .00
  45I                                                                                                 I   .00
  50I                                                                                                 I   .00
  55I                                                                                                 I   .00
  60I                                                                                                 I   .00
  65I                                                                                                 I   .00
  70I                                                                                                 I   .00
  75I                                                                                                 I   .00
  80I                                                                                                 I   .00
  85I                                                                                                 I   .00
  90I                                                                                                 I   .00
  95I                                                                                                 I   .00
 100I                                                                                                 I   .00
    ---------------------------------------------------------------------------------------------------
```

AVERAGE ABSOLUTE ERROR =   .0HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 1200HZ.



AVERAGE ABSOLUTE ERROR =   .1HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 1400HZ.

```
     ------------------------------------------------------------
-100I                                                           I   .27
 -95I                                                           I   .00
 -90I                                                           I   .00
 -85I                                                           I   .00
 -80I                                                           I   .00
 -75I                                                           I   .00
 -70I                                                           I   .00
 -65I                                                           I   .00
 -60I                                                           I   .00
 -55I                                                           I   .00
 -50I                                                           I   .00
 -45I                                                           I   .00
 -40I                                                           I   .00
 -35I                                                           I   .00
 -30I                                                           I   .00
 -25I                                                           I   .00
 -20I                                                           I   .00
 -15I                                                           I   .00
 -10I                                                           I   .27
  -5I                                                           I   .45
   0I*************************************************************98.41
   5I                                                           I   .45
  10I                                                           I   .00
  15I                                                           I   .00
  20I                                                           I   .00
  25I                                                           I   .00
  30I                                                           I   .00
  35I                                                           I   .00
  40I                                                           I   .00
  45I                                                           I   .00
  50I                                                           I   .00
  55I                                                           I   .00
  60I                                                           I   .00
  65I                                                           I   .00
  70I                                                           I   .00
  75I                                                           I   .00
  80I                                                           I   .00
  85I                                                           I   .00
  90I                                                           I   .00
  95I                                                           I   .00
 100I                                                           I   .27
     ------------------------------------------------------------
```

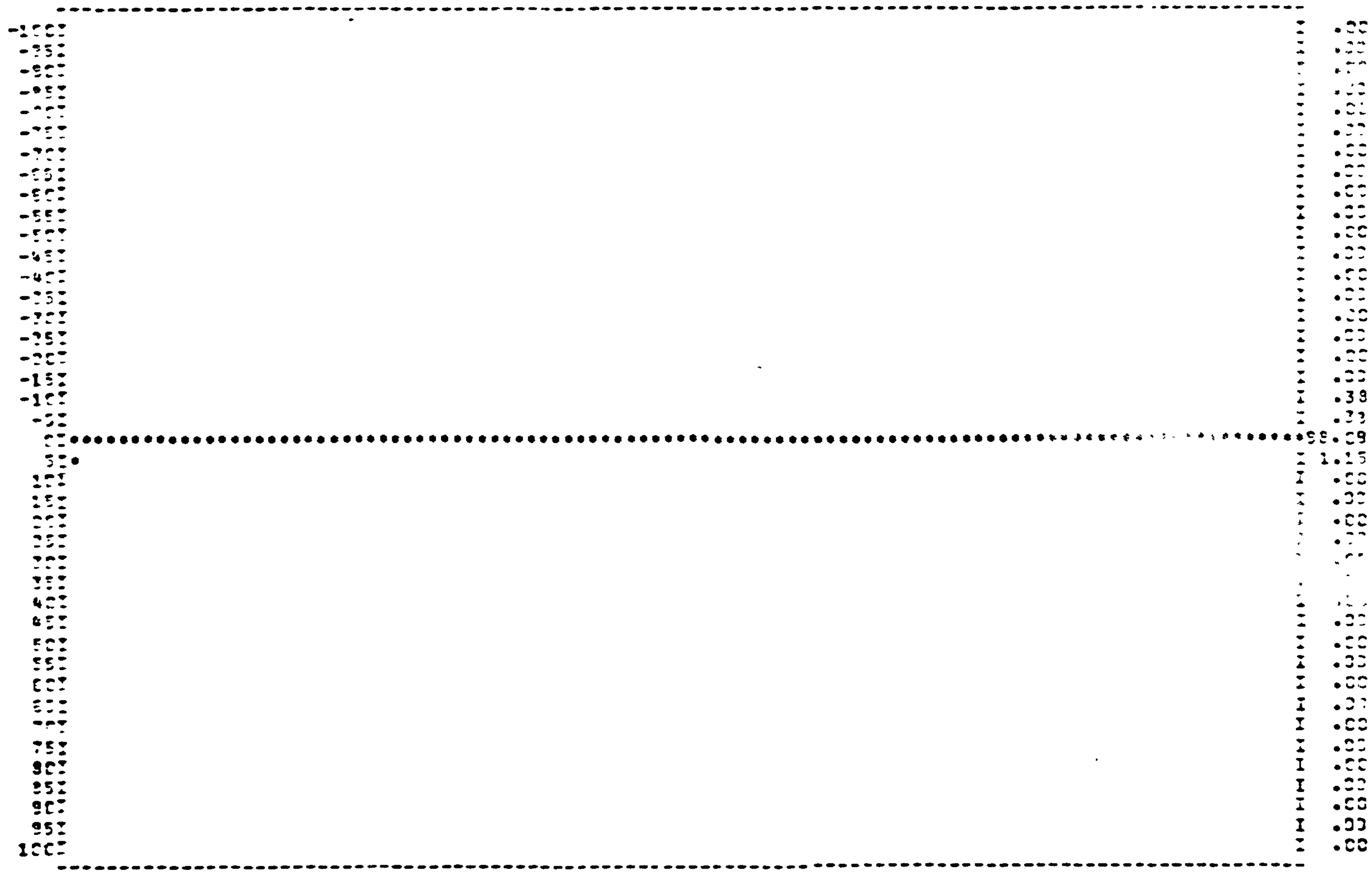
AVERAGE ABSOLUTE ERROR = .1HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 1500HZ.

-100	.00
-95	.00
-90	.00
-85	.00
-80	.00
-75	.00
-70	.00
-65	.00
-60	.00
-55	.00
-50	.00
-45	.00
-40	.00
-35	.00
-30	.00
-25	.00
-20	.00
-15	.00
-10	.00
-5	.85
0	37.93
5	1.15
10	.00
15	.00
20	.00
25	.00
30	.00
35	.00
40	.00
45	.00
50	.00
55	.00
60	.00
65	.00
70	.00
75	.00
80	.00
85	.00
90	.00
95	.00
100	.00

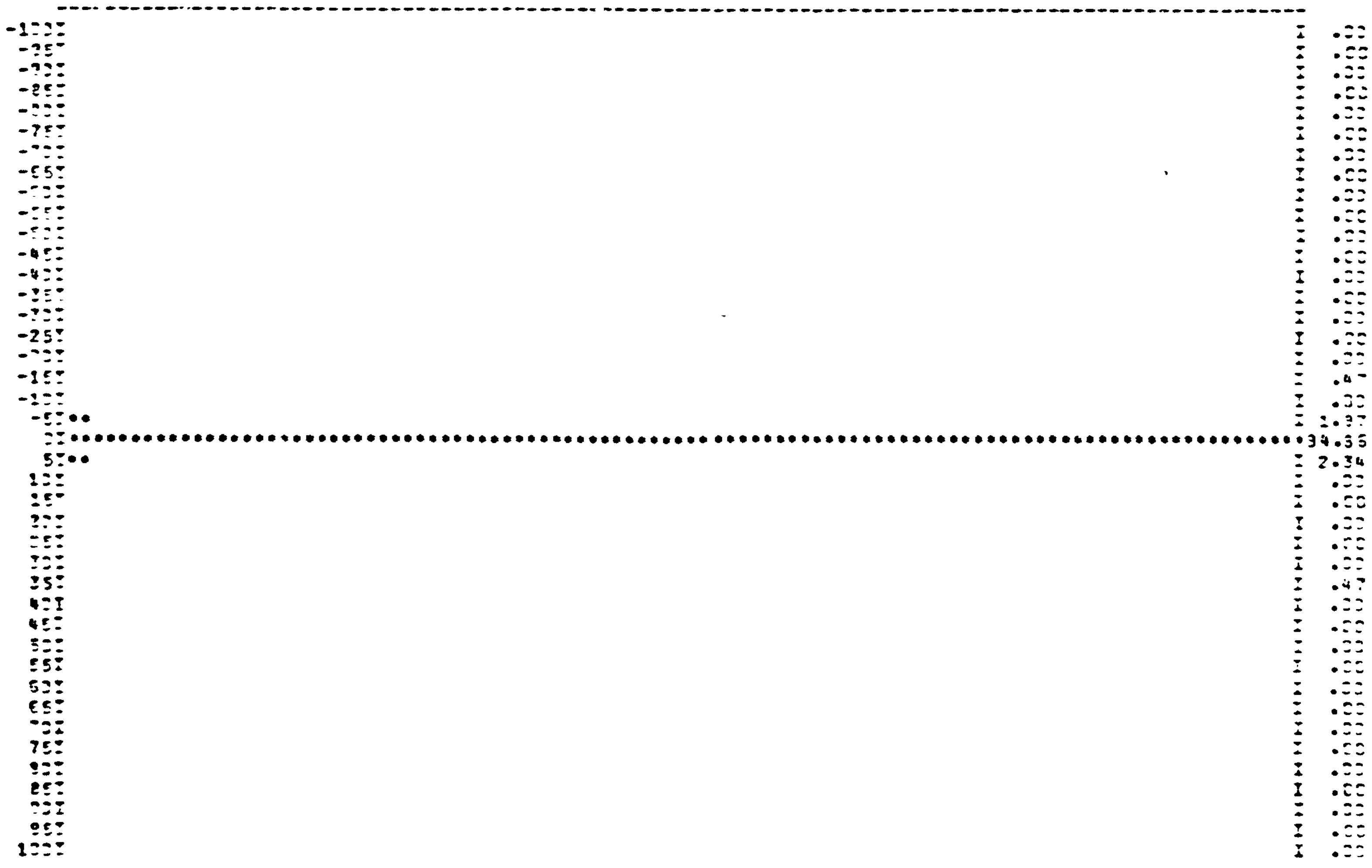
AVERAGE ABSOLUTE ERROR = .1HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 1000HZ.



AVERAGE ABSOLUTE ERROR = .1HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 1700HZ.



AVERAGE ABSOLUTE ERROR = .4HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 1800HZ.

-100	.34
-95	.00
-90	.00
-85	.00
-80	.00
-75	.00
-70	.00
-65	.00
-60	.00
-55	.00
-50	.00
-45	.00
-40	.00
-35	.00
-30	.00
-25	.00
-20	.00
-15	.34
-10	.00
-5	3.33
0	92.57
5	3.33
10	.00
15	.00
20	.00
25	.00
30	.00
35	.00
40	.00
45	.00
50	.00
55	.00
60	.00
65	.00
70	.00
75	.00
80	.00
85	.00
90	.00
95	.00
100	.00

AVERAGE ABSOLUTE ERROR = .4HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 1900HZ.

-100	.00
-95	.00
-90	.00
-85	.00
-80	.00
-75	.00
-70	.00
-65	.00
-60	.00
-55	.00
-50	.00
-45	.00
-40	.00
-35	.00
-30	.00
-25	.00
-20	.00
-15	.00
-10	.53
-5	5.71
0	34.33
5	7.76
10	.59
15	.23
20	.00
25	.00
30	.00
35	.00
40	.00
45	.00
50	.00
55	.00
60	.00
65	.00
70	.00
75	.00
80	.00
85	.00
90	.00
95	.00
100	.00

AVERAGE ABSOLUTE ERROR = .3HZ.

```
FREQUENCY INTERVAL (100HZ.) STARTS AT 2000HZ.
   ---------------------------------------------------------------------------------------------------------------------
-100I                                                                                                                   I   .00
 -95I                                                                                                                   I   .00
 -90I                                                                                                                   I   .00
 -85I                                                                                                                   I   .00
 -80I                                                                                                                   I   .00
 -75I                                                                                                                   I   .00
 -70I                                                                                                                   I   .00
 -65I                                                                                                                   I   .00
 -60I                                                                                                                   I   .00
 -55I                                                                                                                   I   .00
 -50I                                                                                                                   I   .00
 -45I                                                                                                                   I   .00
 -40I                                                                                                                   I   .00
 -35I                                                                                                                   I   .00
 -30I                                                                                                                   I   .00
 -25I                                                                                                                   I   .00
 -20I                                                                                                                   I   .00
 -15I                                                                                                                   I   .00
 -10I                                                                                                                   I   .42
  -5I***********                                                                                                        I  9.44
   0I*******************************************************************************************************************35.23
   5I******                                                                                                             I  5.43
  10I                                                                                                                   I   .42
  15I                                                                                                                   I   .00
  20I                                                                                                                   I   .00
  25I                                                                                                                   I   .00
  30I                                                                                                                   I   .00
  35I                                                                                                                   I   .00
  40I                                                                                                                   I   .00
  45I                                                                                                                   I   .00
  50I                                                                                                                   I   .00
  55I                                                                                                                   I   .00
  60I                                                                                                                   I   .00
  65I                                                                                                                   I   .00
  70I                                                                                                                   I   .00
  75I                                                                                                                   I   .00
  80I                                                                                                                   I   .00
  85I                                                                                                                   I   .00
  90I                                                                                                                   I   .00
  95I                                                                                                                   I   .00
 100I                                                                                                                   I   .00
   ---------------------------------------------------------------------------------------------------------------------
AVERAGE ABSOLUTE ERROR =    .8HZ.
```

FREQUENCY INTERVAL (10HZ.) STARTS AT 2100HZ.

```
      --------------------------------------------------------------------------------------------------------------
-100I***********                                                                                               I 7.83
 -95I                                                                                                          I  .00
 -90I                                                                                                          I  .00
 -85I                                                                                                          I  .00
 -80I                                                                                                          I  .00
 -75I                                                                                                          I  .00
 -70I                                                                                                          I  .00
 -65I                                                                                                          I  .00
 -60I                                                                                                          I  .00
 -55I                                                                                                          I  .00
 -50I                                                                                                          I  .00
 -45I                                                                                                          I  .00
 -40I                                                                                                          I  .00
 -35I                                                                                                          I  .00
 -30I                                                                                                          I  .00
 -25I                                                                                                          I  .00
 -20I                                                                                                          I  .00
 -15I*                                                                                                         I  .51
 -10I                                                                                                          I  .00
  -5I*************                                                                                             I 9.09
   0I**************************************************************************************************************75.76
   5I*********                                                                                                 I 6.06
  10I                                                                                                          I  .00
  15I                                                                                                          I  .00
  20I                                                                                                          I  .00
  25I                                                                                                          I  .00
  30I                                                                                                          I  .00
  35I                                                                                                          I  .00
  40I                                                                                                          I  .00
  45I                                                                                                          I  .00
  50I                                                                                                          I  .00
  55I                                                                                                          I  .00
  60I                                                                                                          I  .00
  65I                                                                                                          I  .00
  70I                                                                                                          I  .00
  75I                                                                                                          I  .00
  80I                                                                                                          I  .00
  85I                                                                                                          I  .00
  90I                                                                                                          I  .00
  95I                                                                                                          I  .00
 100I*                                                                                                         I  .51
      --------------------------------------------------------------------------------------------------------------
```

AVERAGE ABSOLUTE ERROR =   .9HZ.

FREQUENCY INTERVAL (10CHZ.) STARTS AT 220CHZ.

-100	.98
-95	.00
-90	.00
-85	.00
-80	.00
-75	.00
-70	.00
-65	.00
-60	.00
-55	.00
-50	.00
-45	.00
-40	.00
-35	.00
-30	.49
-25	.00
-20	.00
-15	.00
-10	2.44
-5	17.56
0	64.39
5	13.17
10	2.44
15	1.45
20	.98
25	.00
30	.00
35	.00
40	.00
45	.00
50	.00
55	.00
60	.00
65	.00
70	.00
75	.00
80	.00
85	.00
90	.00
95	.00
100	.00

AVERAGE ABSOLUTE ERROR = 2.4HZ.

```
FREQUENCY INTERVAL (100HZ.) STARTS AT 2300HZ.
     -----------------------------------------------------------------------------------------------
-100I                                                                                               I   .00
 -95I                                                                                               I   .00
 -90I                                                                                               I   .00
 -85I                                                                                               I   .00
 -80I                                                                                               I   .00
 -75I                                                                                               I   .00
 -70I                                                                                               I   .00
 -65I                                                                                               I   .00
 -60I                                                                                               I   .00
 -55I                                                                                               I   .00
 -50I                                                                                               I   .00
 -45I                                                                                               I   .00
 -40I                                                                                               I   .00
 -35I**                                                                                             I  1.19
 -30I                                                                                               I   .00
 -25I                                                                                               I   .00
 -20I*                                                                                              I   .59
 -15I                                                                                               I   .00
 -10I***                                                                                            I  2.38
  -5I*****************                                                                              I 11.67
   0I***********************************************************************************************55.83
   5I***************                                                                                I 10.39
  10I**                                                                                             I  1.49
  15I**                                                                                             I  1.49
  20I*                                                                                              I   .59
  25I*                                                                                              I   .59
  30I**                                                                                             I  1.19
  35I                                                                                               I   .00
  40I                                                                                               I   .00
  45I                                                                                               I   .00
  50I                                                                                               I   .00
  55I                                                                                               I   .00
  60I                                                                                               I   .00
  65I                                                                                               I   .00
  70I                                                                                               I   .00
  75I                                                                                               I   .00
  80I                                                                                               I   .00
  85I                                                                                               I   .00
  90I                                                                                               I   .00
  95I                                                                                               I   .00
 100I                                                                                               I   .00
     -----------------------------------------------------------------------------------------------
AVERAGE ABSOLUTE ERROR =  3.9HZ.
```

FREQUENCY INTERVAL (100HZ.) STARTS AT 2400 Z.

```
     ------------------------------------------------------------------------------------------
-100I**                                                                                       I 1.44
 -95I                                                                                         I  .00
 -90I                                                                                         I  .00
 -85I                                                                                         I  .00
 -80I                                                                                         I  .00
 -75I                                                                                         I  .00
 -70I                                                                                         I  .00
 -65I                                                                                         I  .00
 -60I                                                                                         I  .00
 -55I                                                                                         I  .00
 -50I                                                                                         I  .00
 -45I                                                                                         I  .00
 -40I                                                                                         I  .00
 -35I*                                                                                        I  .48
 -30I*                                                                                        I  .48
 -25I                                                                                         I  .00
 -20I                                                                                         I  .00
 -15I**                                                                                       I 1.44
 -10I***                                                                                      I 1.91
  -5I********************                                                                     I 9.57
   0I********************************************    ******************************************55.55
   5I****************************                                                             I13.40
  10I**                                                                                       I 1.44
  15I*                                                                                        I  .48
  20I**                                                                                       I 1.44
  25I                                                                                         I  .00
  30I*                                                                                        I  .48
  35I                                                                                         I  .00
  40I                                                                                         I  .00
  45I                                                                                         I  .00
  50I*                                                                                        I  .96
  55I                                                                                         I  .00
  60I*                                                                                        I  .48
  65I                                                                                         I  .00
  70I                                                                                         I  .00
  75I                                                                                         I  .00
  80I                                                                                         I  .00
  85I                                                                                         I  .00
  90I                                                                                         I  .00
  95I                                                                                         I  .00
 100I*                                                                                        I  .48
     ------------------------------------------------------------------------------------------
```

AVERAGE ABSOLUTE ERROR = 3.3HZ.

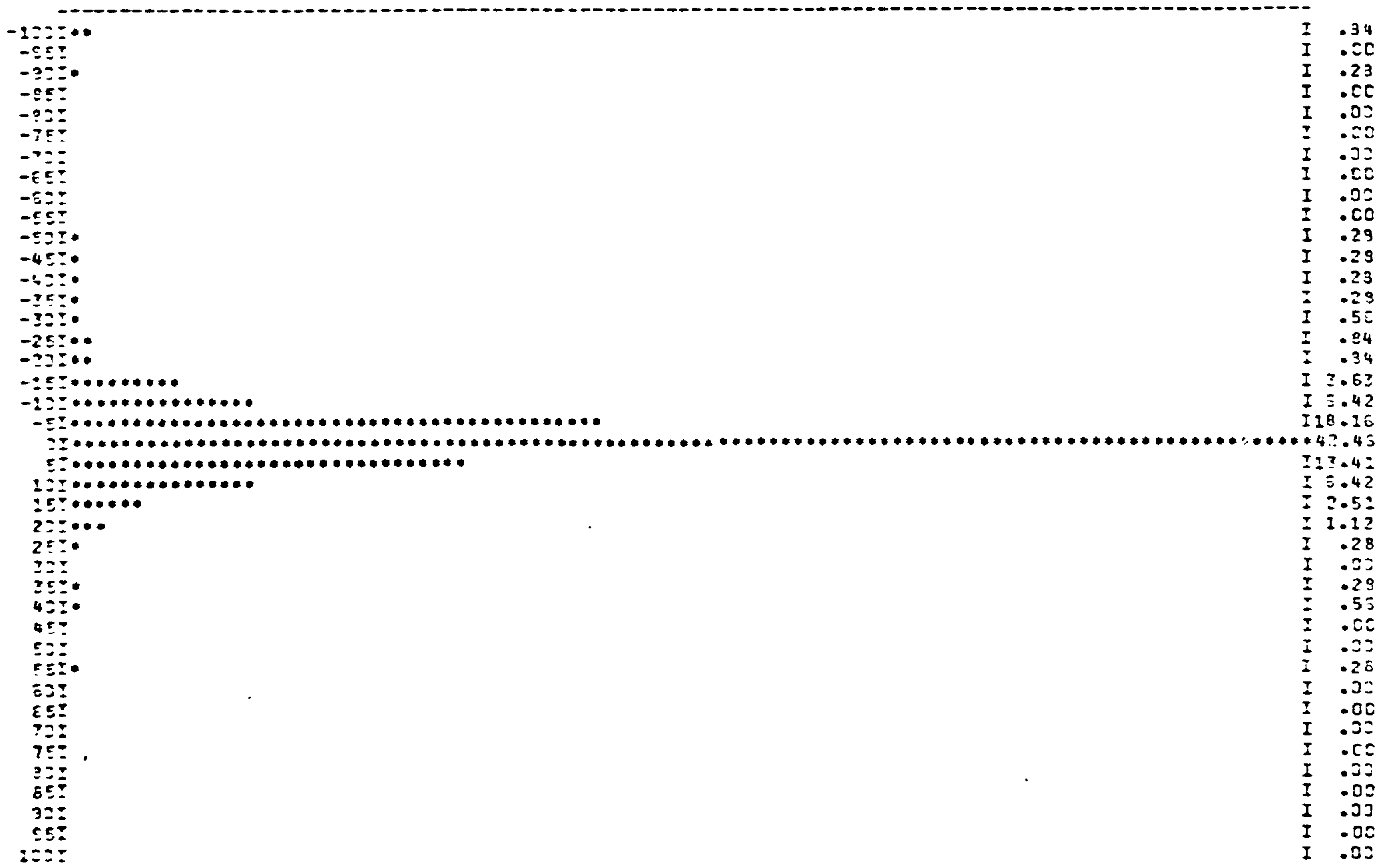
FREQUENCY INTERVAL (10CHZ.) STARTS AT 250CHZ.

Interval	%
-100	1.94
-95	.00
-90	.00
-85	.00
-80	.00
-75	.00
-70	.00
-65	.00
-60	.00
-55	.00
-50	.00
-45	.00
-40	.00
-35	.00
-30	.00
-25	.00
-20	.00
-15	.00
-10	1.94
-5	13.06
0	65.18
5	9.53
10	2.53
15	.53
20	.00
25	.00
30	.00
35	.00
40	.00
45	.00
50	.00
55	.00
60	.00
65	.00
70	.00
75	.00
80	.00
85	.00
90	.00
95	.00
100	.00

AVERAGE ABSOLUTE ERROR = 2.CHZ.

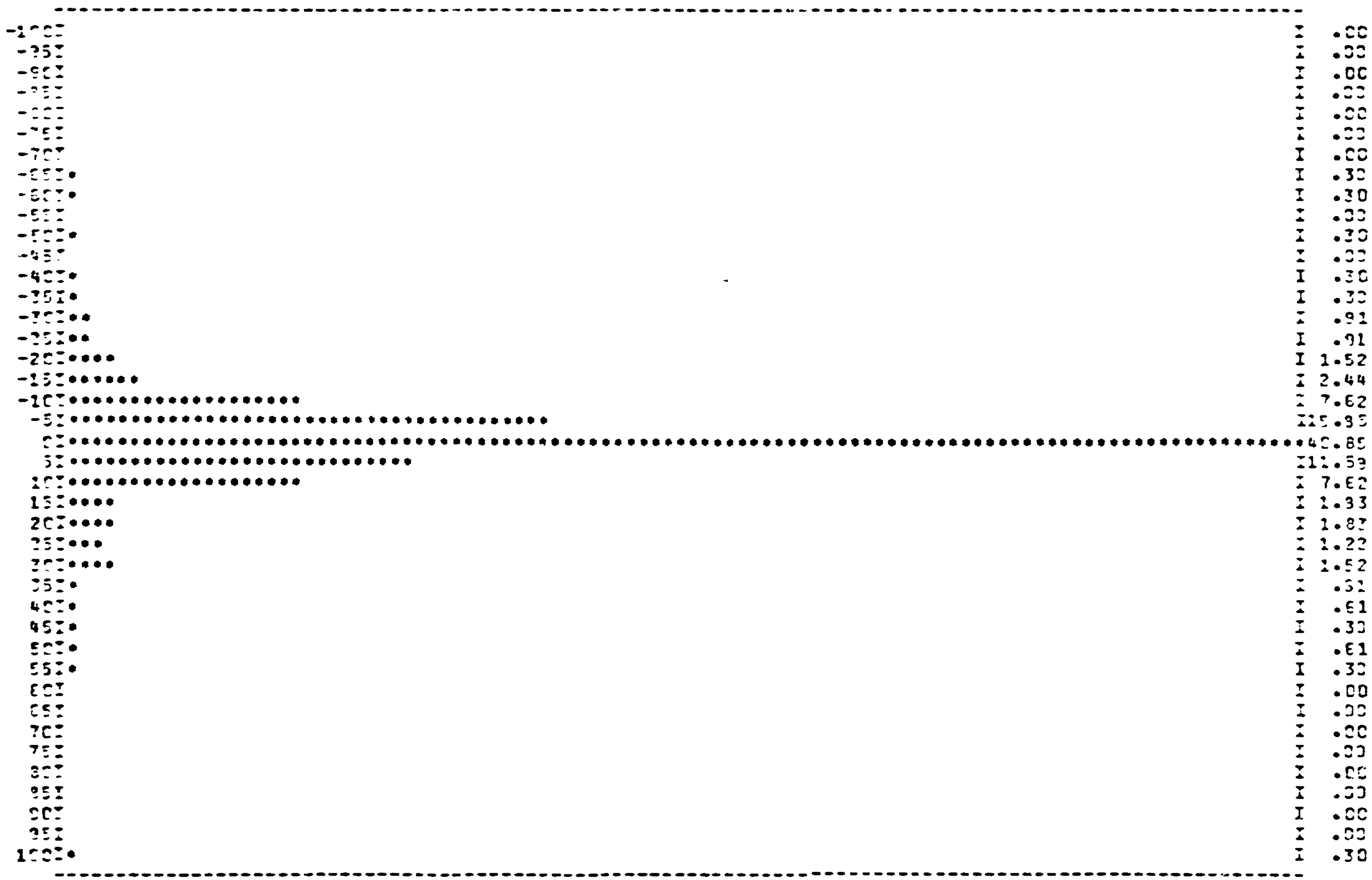
```
FREQUENCY INTERVAL (100HZ.) STARTS AT 2800HZ.
     ----------------------------------------------------------------------------------------------------------
-100I*********                                                                                              I 4.80
 -95I                                                                                                       I  .00
 -90I                                                                                                       I  .00
 -85I                                                                                                       I  .00
 -80I                                                                                                       I  .00
 -75I                                                                                                       I  .00
 -70I                                                                                                       I  .00
 -65I                                                                                                       I  .00
 -60I                                                                                                       I  .00
 -55I                                                                                                       I  .00
 -50I                                                                                                       I  .00
 -45I                                                                                                       I  .00
 -40I                                                                                                       I  .00
 -35I                                                                                                       I  .00
 -30I                                                                                                       I  .00
 -25I                                                                                                       I  .00
 -20I                                                                                                       I  .00
 -15I***                                                                                                    I 1.60
 -10I*******                                                                                                I 4.00
  -5I************************                                                                               I14.00
   0I**********************************************************************************************************61.20
   5I***************                                                                                        I 8.40
  10I*********                                                                                              I 5.20
  15I                                                                                                       I  .00
  20I*                                                                                                      I  .40
  25I                                                                                                       I  .00
  30I                                                                                                       I  .00
  35I                                                                                                       I  .00
  40I                                                                                                       I  .00
  45I                                                                                                       I  .00
  50I                                                                                                       I  .00
  55I                                                                                                       I  .00
  60I                                                                                                       I  .00
  65I                                                                                                       I  .00
  70I                                                                                                       I  .00
  75I                                                                                                       I  .00
  80I                                                                                                       I  .00
  85I                                                                                                       I  .00
  90I                                                                                                       I  .00
  95I                                                                                                       I  .00
 100I*                                                                                                      I  .40
     ----------------------------------------------------------------------------------------------------------
AVERAGE ABSOLUTE ERROR =  2.5HZ.
```

FREQUENCY INTERVAL (100HZ.) STARTS AT 2700HZ.



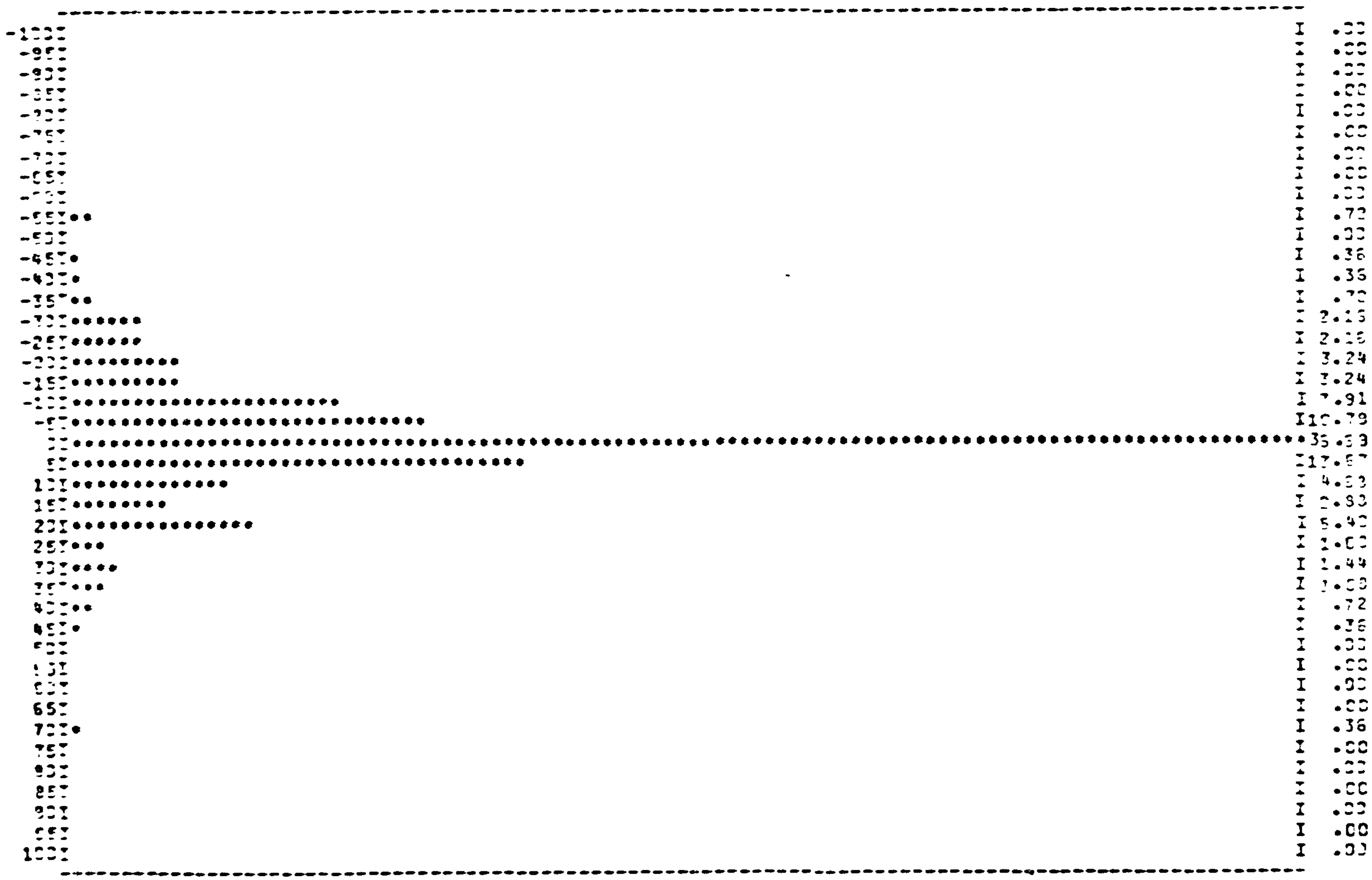
AVERAGE ABSOLUTE ERROR = 5.9HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 2800HZ.



AVERAGE ABSOLUTE ERROR = 7.3HZ.

FREQUENCY INTERVAL (100HZ.) STARTS AT 2900HZ.



AVERAGE ABSOLUTE ERROR = 3.0HZ.

RECORD	1	WORD	54	STATUS	0	TAPE	0	ERROR	1983.	TAPE	1	ERROR	2030.
RECORD	1	WORD	121	STATUS	0	TAPE	0	ERROR	2345.	TAPE	3	ERROR	2020.
RECORD	1	WORD	170	STATUS	0	TAPE	3	ERROR	2356.	TAPE	1	ERROR	2052.
RECORD	2	WORD	7	STATUS	0	TAPE	2	ERROR	2008.	TAPE	3	ERROR	2055.
RECORD	2	WORD	53	STATUS	0	TAPE	0	ERROR	2400.	TAPE	1	ERROR	2241.
RECORD	2	WORD	143	STATUS	0	TAPE	3	ERROR	2428.	TAPE	1	ERROR	2350.
RECORD	2	WORD	170	STATUS	0	TAPE	3	ERROR	1972.	TAPE	1	ERROR	2222.
RECORD	3	WORD	15	STATUS	0	TAPE	0	ERROR	2344.	TAPE	3	ERROR	2005.
RECORD	3	WORD	10	STATUS	0	TAPE	0	ERROR	1900.	TAPE	1	ERROR	2345.
RECORD	3	WORD	27	STATUS	0	TAPE	0	ERROR	2226.	TAPE	1	ERROR	2377.
RECORD	3	WORD	61	STATUS	0	TAPE	1	ERROR	1409.	TAPE	2	ERROR	1947.
RECORD	3	WORD	50	STATUS	2	TAPE	0	ERROR	2068.	TAPE	1	ERROR	2064.
RECORD	3	WORD	73	STATUS	0	TAPE	0	ERROR	2415.	TAPE	2	ERROR	2241.
RECORD	3	WORD	75	STATUS	0	TAPE	0	ERROR	2407.	TAPE	2	ERROR	2007.
RECORD	3	WORD	102	STATUS	0	TAPE	0	ERROR	2403.	TAPE	1	ERROR	2043.
RECORD	3	WORD	103	STATUS	0	TAPE	0	ERROR	2307.	TAPE	3	ERROR	2035.
RECORD	3	WORD	147	STATUS	0	TAPE	0	ERROR	2426.	TAPE	3	ERROR	2277.
RECORD	3	WORD	140	STATUS	0	TAPE	0	ERROR	2261.	TAPE	1	ERROR	2205.
RECORD	4	WORD	123	STATUS	0	TAPE	0	ERROR	1965.	TAPE	1	ERROR	2400.
RECORD	4	WORD	170	STATUS	0	TAPE	0	ERROR	2347.	TAPE	1	ERROR	2047.
RECORD	5	WORD	10	STATUS	0	TAPE	0	ERROR	1909.	TAPE	1	ERROR	2014.
RECORD	5	WORD	54	STATUS	0	TAPE	0	ERROR	2222.	TAPE	1	ERROR	2437.
RECORD	5	WORD	97	STATUS	0	TAPE	0	ERROR	2304.	TAPE	3	ERROR	2054.
RECORD	5	WORD	151	STATUS	0	TAPE	0	ERROR	2384.	TAPE	1	ERROR	2223.
RECORD	5	WORD	155	STATUS	0	TAPE	0	ERROR	2765.	TAPE	1	ERROR	2700.
RECORD	6	WORD	1	STATUS	0	TAPE	0	ERROR	2401.	TAPE	1	ERROR	1903.
RECORD	6	WORD	5	STATUS	0	TAPE	0	ERROR	2307.	TAPE	2	ERROR	2020.
RECORD	6	WORD	25	STATUS	0	TAPE	0	ERROR	1250.	TAPE	1	ERROR	2257.
RECORD	6	WORD	30	STATUS	0	TAPE	0	ERROR	2364.	TAPE	2	ERROR	2065.
RECORD	6	WORD	33	STATUS	0	TAPE	0	ERROR	2024.	TAPE	2	ERROR	2037.
RECORD	6	WORD	45	STATUS	0	TAPE	1	ERROR	2719.	TAPE	2	ERROR	2011.
RECORD	6	WORD	40	STATUS	0	TAPE	0	ERROR	1957.	TAPE	1	ERROR	2274.
RECORD	6	WORD	54	STATUS	0	TAPE	0	ERROR	2016.	TAPE	1	ERROR	2302.
RECORD	6	WORD	97	STATUS	0	TAPE	0	ERROR	1969.	TAPE	1	ERROR	2364.
RECORD	6	WORD	105	STATUS	0	TAPE	0	ERROR	2239.	TAPE	1	ERROR	2023.
RECORD	6	WORD	121	STATUS	0	TAPE	0	ERROR	2460.	TAPE	1	ERROR	2253.
RECORD	6	WORD	122	STATUS	0	TAPE	1	ERROR	1919.	TAPE	2	ERROR	2203.
RECORD	6	WORD	126	STATUS	1	TAPE	0	ERROR	3000.	TAPE	1	ERROR	3000.
RECORD	6	WORD	130	STATUS	0	TAPE	1	ERROR	2374.	TAPE	2	ERROR	2054.
RECORD	6	WORD	140	STATUS	0	TAPE	0	ERROR	2365.	TAPE	2	ERROR	2014.
RECORD	6	WORD	144	STATUS	0	TAPE	0	ERROR	1307.	TAPE	1	ERROR	1500.
RECORD	6	WORD	146	STATUS	0	TAPE	0	ERROR	1956.	TAPE	1	ERROR	2232.
RECORD	6	WORD	140	STATUS	0	TAPE	0	ERROR	1947.	TAPE	1	ERROR	2375.
RECORD	6	WORD	150	STATUS	0	TAPE	0	ERROR	2363.	TAPE	1	ERROR	2423.
RECORD	6	WORD	170	STATUS	0	TAPE	1	ERROR	2013.	TAPE	2	ERROR	2050.
RECORD	6	WORD	173	STATUS	0	TAPE	0	ERROR	1957.	TAPE	2	ERROR	2050.
RECORD	7	WORD	3	STATUS	0	TAPE	0	ERROR	1903.	TAPE	1	ERROR	2253.
RECORD	7	WORD	5	STATUS	0	TAPE	0	ERROR	1901.	TAPE	1	ERROR	2234.
RECORD	7	WORD	10	STATUS	0	TAPE	1	ERROR	2060.	TAPE	2	ERROR	2023.
RECORD	7	WORD	40	STATUS	0	TAPE	0	ERROR	1961.	TAPE	1	ERROR	2013.
RECORD	7	WORD	124	STATUS	0	TAPE	0	ERROR	2430.	TAPE	1	ERROR	2370.
RECORD	8	WORD	74	STATUS	0	TAPE	0	ERROR	2414.	TAPE	1	ERROR	2007.
RECORD	8	WORD	133	STATUS	0	TAPE	2	ERROR	2266.	TAPE	3	ERROR	2604.
RECORD	9	WORD	14	STATUS	0	TAPE	0	ERROR	2266.	TAPE	1	ERROR	2004.
RECORD	9	WORD	63	STATUS	0	TAPE	2	ERROR	2306.	TAPE	3	ERROR	2040.
RECORD	16	WORD	51	STATUS	0	TAPE	0	ERROR	1776.	TAPE	2	ERROR	1774.
RECORD	26	WORD	43	STATUS	0	TAPE	1	ERROR	1395.	TAPE	3	ERROR	1105.
RECORD	32	WORD	132	STATUS	0	TAPE	0	ERROR	1775.	TAPE	1	ERROR	1197.
RECORD	46	WORD	104	STATUS	0	TAPE	0	ERROR	1310.	TAPE	2	ERROR	1310.
RECORD	47	WORD	93	STATUS	0	TAPE	0	ERROR	940.	TAPE	1	ERROR	940.

FIGURE 4 MERGE4 OUTPUT ACCEPTED VALUES WITH NON-ZERO ERROR LEVEL OR VOICE DISCREPANCY

```
RECORD  58 WORD  10 STATUS  2 TAPE 0 ERROR  550. TAPE 1 ERROR  350.
RECORD  84 WORD 105 STATUS  2 TAPE 0 ERROR  240. TAPE 1 ERROR  240.
RECORD  88 WORD 100 STATUS  2 TAPE 1 ERROR 1079. TAPE 2 ERROR 1080.
RECORD 114 WORD  44 STATUS  2 TAPE 1 ERROR  965. TAPE 2 ERROR  965.
RECORD 121 WORD  90 STATUS  2 TAPE 2 ERROR  374. TAPE
RECORD 127 WORD 110 STATUS  2 TAPE 0 ERROR  474. TAPE 1 ERROR  471.
RECORD 133 WORD  26 STATUS  2 TAPE 2 ERROR  971. TAPE 3 ERROR  971.
RECORD 154 WORD 143 STATUS  2 TAPE 2 ERROR  465. TAPE 3 ERROR  465.
RECORD 271 WORD 112 STATUS  2 TAPE 0 ERROR  500. TAPE 2 ERROR  500.
RECORD 343 WORD  73 STATUS  2 TAPE 0 ERROR  452. TAPE 3 ERROR  452.
RECORD 352 WORD  67 STATUS  2 TAPE 1 ERROR  562. TAPE 2 ERROR  562.
RECORD 358 WORD 132 STATUS  2 TAPE 1 ERROR  411. TAPE 2 ERROR  411.
RECORD 358 WORD 170 STATUS  2 TAPE 2 ERROR  558. TAPE 3 ERROR  558.
RECORD 393 WORD  70 STATUS  2 TAPE 0 ERROR  331. TAPE 1 ERROR  332.
END OF FILE WRITTEN ON UNIT 4
397 RECORDS PROCESSED.
STATUS COUNTS
    0    22736
    1        1
    2       19
    3        0
    4        0
    5        0
    6        0
    7        0
    8        0
    9        0
   10        0
   11        0
   12        0
   13        0
   14        0
   15        0
   16        0
NORMAL EXIT.  EXECUTION TIME:         53044 MILLISECONDS.
```

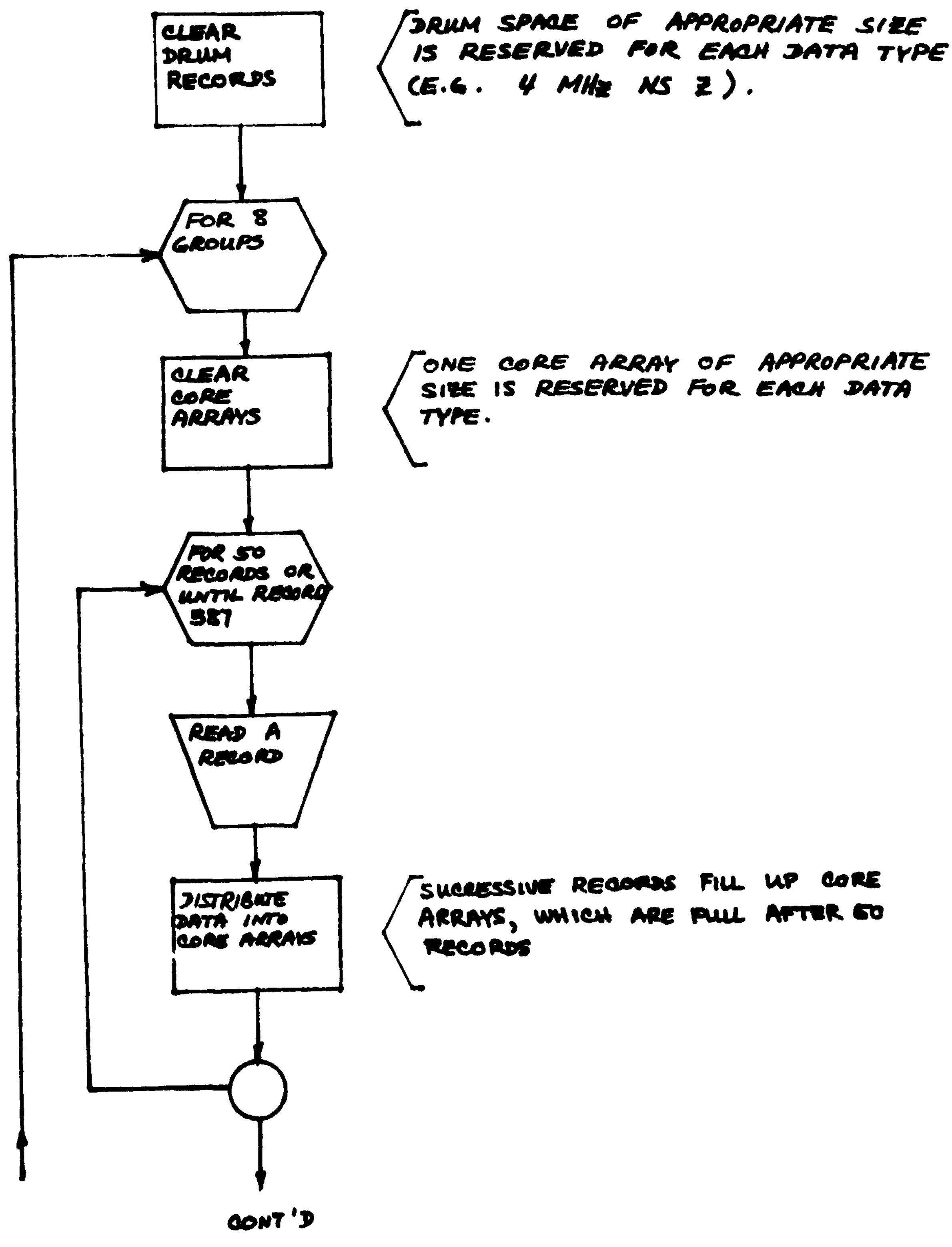
NUMBER OF ACCEPTANCES
AT EACH ERROR LEVEL.

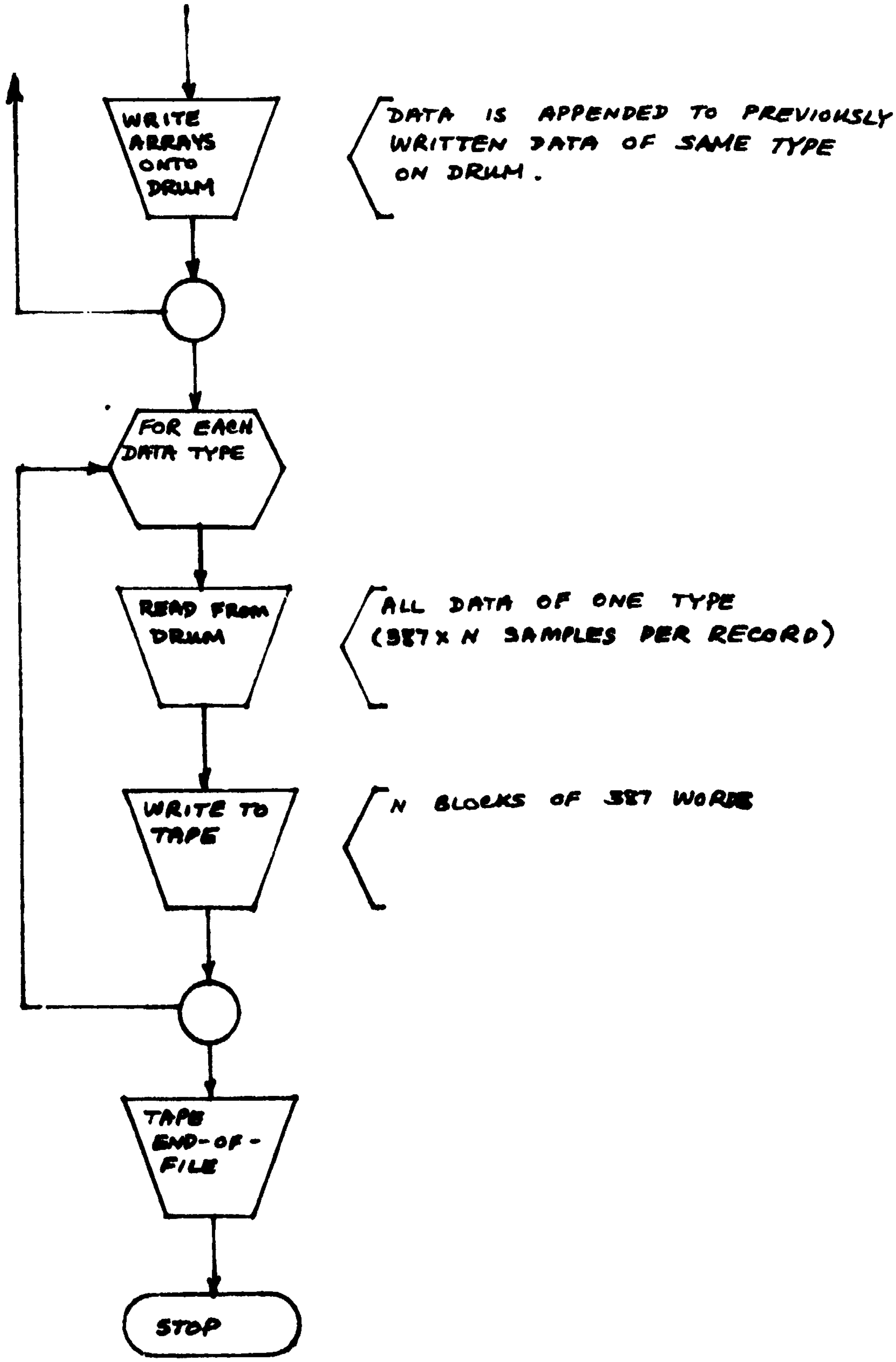
50 records at a time, then output to random-access drum. The next 50 records were then demultiplexed in core, and these data were concatenated to the previous data on the drum. The random access feature was used to skip over space reserved for as yet unprocessed data. Figure 5 shows the general flow of the DEMUX program as well as diagrams of core and drum space allocations.

The DEMUX program contained a coding error which deleted data from the arrays for 4, 8, 16, and 32 MHz. The error occurred in the final transfer of data from drum to tape, in subroutine TAPER (internal subroutine in DEMUX). Figure 6 diagrams the nature of the error. In the tape output routine, the entire 400*N data array (N is the number of samples per record of the data type being considered) should have been processed at once. In Figure 6, the data array in the drum diagram should have been written on tape as 4 contiguous parts of 387 values, truncated to 387. The effect was a compression of the data - 13 values missing after every 387 values on tape. There are (N-1) such error regions. At the end of the tape arrays, there is garbage of length 13* (N-1).

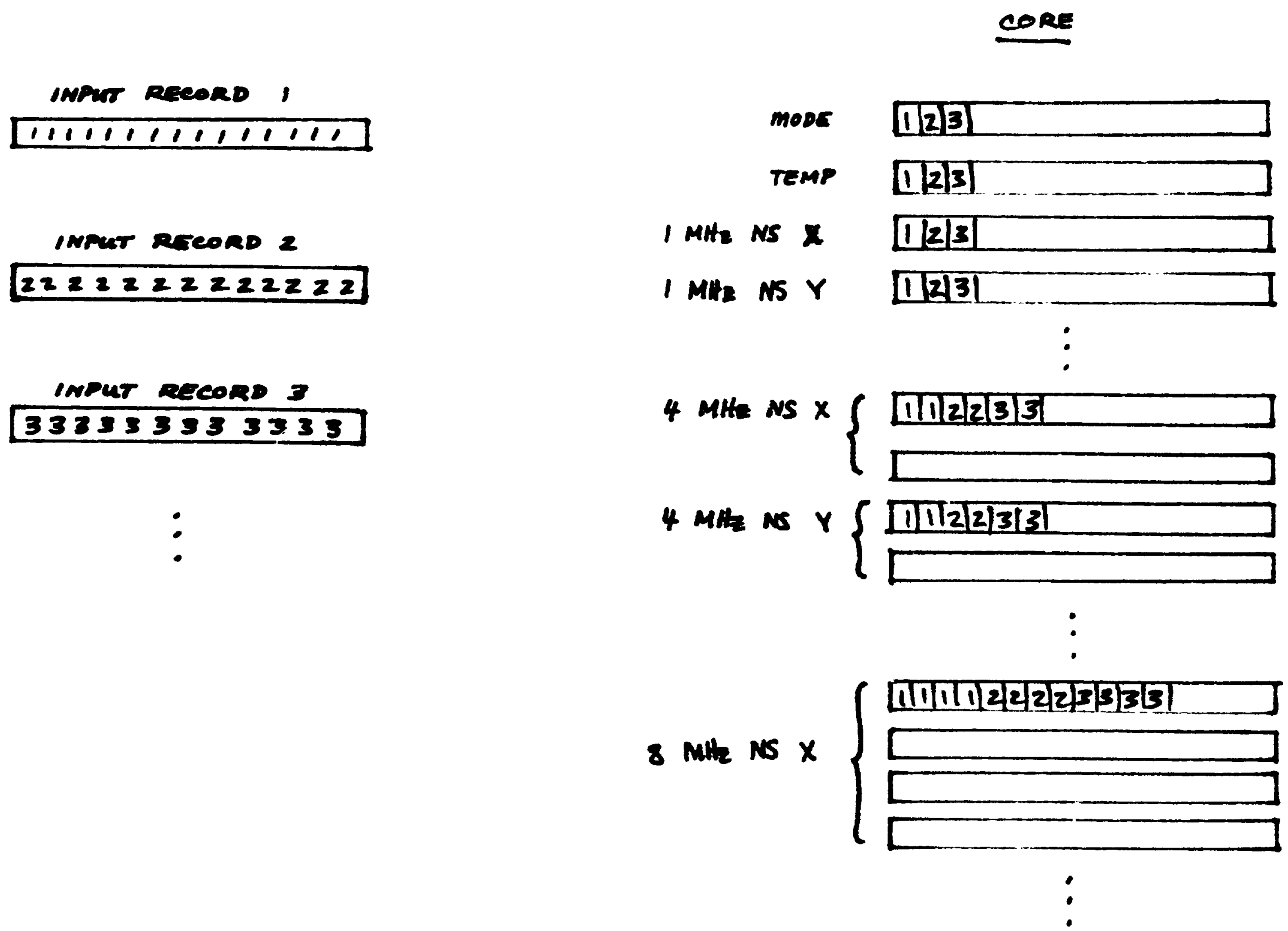
The error is complicated somewhat by the next stage of processing. Because of the design of the DAS, the mode word appearing in record i <u>predicted</u> the receiver mode in

FIGURE 5 DEMUX - DEMULTIPLEXING FLOW




WRITE ARRAYS ONTO DRUM
DATA IS APPENDED TO PREVIOUSLY WRITTEN DATA OF SAME TYPE ON DRUM.
FOR EACH DATA TYPE
READ FROM DRUM
ALL DATA OF ONE TYPE (387 x N SAMPLES PER RECORD)
WRITE TO TAPE
N BLOCKS OF 387 WORDS
TAPE END-OF-FILE
STOP

CORE DEMULTIPLEXING



DRUM DEMULTIPLEXING

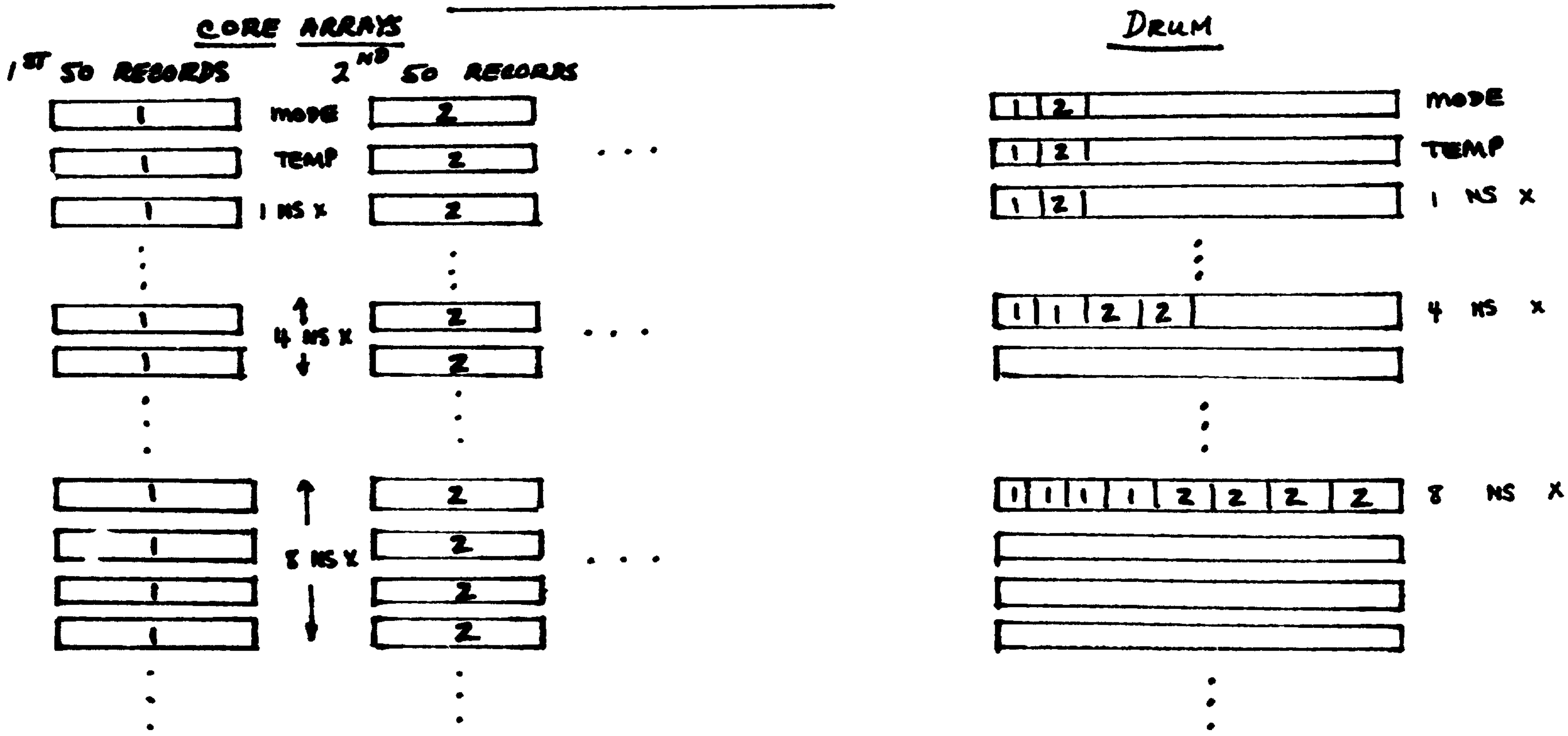
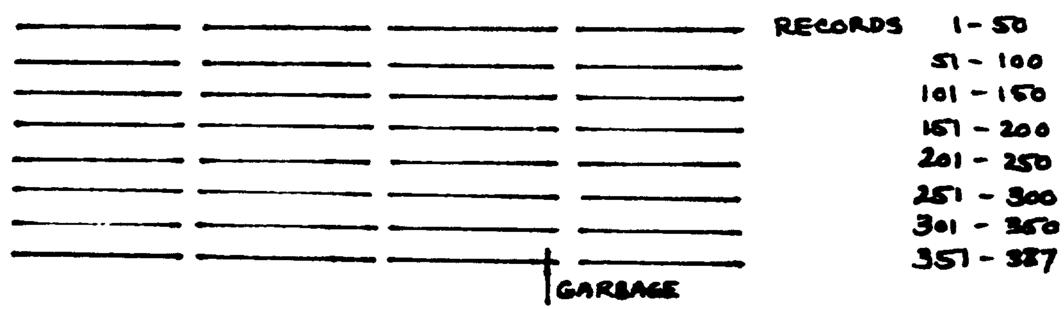


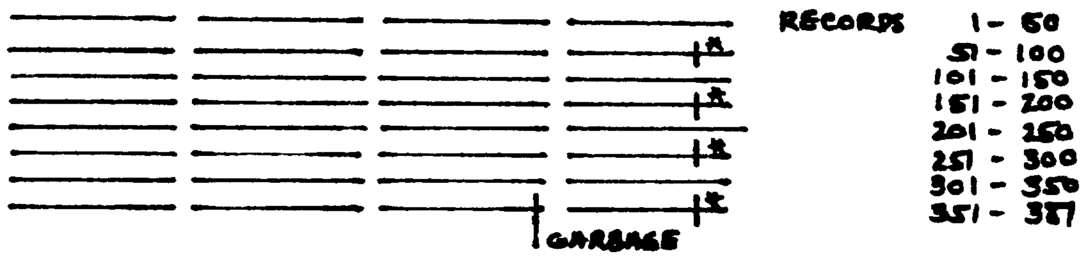
FIGURE 6 DEMULTIPLEXING ERROR

EXAMPLE: 8 MHz DATA (4 SAMPLES / INPUT RECORD)

DRUM:



TAPE:



|* DENOTES AND END-OF-RECORD AS WRITTEN ONTO TAPE, WITH THE NEXT 13 VALUES SKIPPED (THE NEXT RECORD STARTS 13 VALUES LATER), FOR THIS EXAMPLE, THE FINAL 39 VALUES ARE GARBAGE.

record i + 1. To make the mode array agree exactly with the data arrays, with no shift, N points were dropped from the start of each data array. This was accomplished by reading N blocks of 387 values (concatenated), deleting the first N values, and writing N blocks of 386 values. This was done by the REBLOCK program. The output was stored in drum file VCO.

Location of errors - data is missing after the following locations:

	SAMPLE #	APPARENT TIME FROM START
4 MHz	385	0:20:47
8 MHz	383	0:10:20
	770	0:20:47
	1157	0:31:14
16 MHz	379	0:05:07
	766	0:10:20
	1153	0:15:34
	1540	0:20:47
	1927	0:26:01
	2314	0:31:14
	2701	0:36:28
32 MHz	374	0:03:06
	761	0:06:19
	1148	0:09:32

SAMPLE #	APPARENT TIME FROM START
1535	0:12:45
1922	0:15:58
2309	0:19:11
2696	0:22:24
3083	0:25:37
3470	0:28:50
3857	0:32:03
4244	0:35:15

To correct the timing errors, 13 filler values should be inserted following each of the tabulated error locations. This should be done with care, since at 8, 16, and 32 MHz the insertion of the first 13 fillers moves the location for insertion of subsequent sets of 13 fillers. The last 13* (N-1) values should be discarded.

The next step of processing was conversion from VCO frequencies to dB values for scientific data, and from VCO frequency to temperature. The calibration data for the flight unit were typed in and stored in drum files TESTLEVELS and VCOFREQS. These data are printed out in Figure 7. The receiver was operating in the medium (65°) to hot (105°) and above temperature range during the entire traverse, so the cold calibration was ignored.

A two-dimensional linear interpolation was done by

CALIBRATE, first determining the temperature, then using the temperature to interpolate between the medium and high temperature dB vs. frequency curves. Since the calibration data were given as VCO frequency as a function of dB input power, the inverse function first had to be determined. This was done by linear interpolation for VCO frequencies in 1 Hz steps from 300 Hz to 3000 Hz. The dB levels for frequencies below/above the lowest/highest calibrated value were set to the lowest/highest value. Interpolation was not done between these 1 Hz values - they were used purely as a lookup table. The granularity of the data (near the center of the calibration curves) is about 1/27 dB, or .9% on a linear power scale.

Up to this time, there has been no mention of the disposition of the following arrays: MODE, TEMP, TXOFF, CAL. The contents of each will be described here.

The MODE array consists of 386 words of the form $(MAR)_{10}$, where M (the 100's digit) is 1 or 2 depending on whether the receiver was in sync'ed or sync-search mode during the i^{th} record, corresponding to the i^{th} word in the MODE array. The 10's digit, A, is the antenna indicator. Values 1, 2, and 3 correspond to the X, Y, and Z antennas during synch search and to the

FIGURE 7. FLIGHT UNIT CALIBRATION DATA

1 MHZ

DBM	X ANTENNA COLD	X ANTENNA MED	X ANTENNA HOT	Y ANTENNA COLD	Y ANTENNA MED	Y ANTENNA HOT	Z ANTENNA COLD	Z ANTENNA MED	Z ANTENNA HOT
-134.23	380	366	349	378	360	342	383	373	352
-129.24	418	400	376	412	390	366	424	408	383
-119.29	645	631	574	936	614	559	653	640	580
-109.37	947	940	899	944	936	893	950	943	901
-99.41	1202	1195	1168	1201	1196	1167	1203	1198	1170
-89.54	1420	1424	1418	1421	1426	1417	1422	1426	1420
-79.75	1684	1686	1679	1684	1686	1679	1684	1686	1680
-69.62	1967	1959	1952	1968	1959	1952	1968	1959	1953
-59.68	2259	2239	2225	2260	2239	2225	2259	2232	2225
-49.74	2517	2492	2471	2518	2492	2471	2517	2492	2472
-39.80	2769	2759	2744	2770	2758	2743	2769	2758	2744
-34.84	2897	2880	2873	2880	2876	2873	2879	2879	2874
-29.86	2946	2962	2970	2948	2960	2969	2946	2959	2969

2 MHZ

DBM	X ANTENNA COLD	X ANTENNA MED	X ANTENNA HOT	Y ANTENNA COLD	Y ANTENNA MED	Y ANTENNA HOT	Z ANTENNA COLD	Z ANTENNA MED	Z ANTENNA HOT
-134.56	387	376	356	394	378	356	387	379	355
-129.56	442	422	391	449	424	390	442	421	393
-119.61	691	680	620	698	682	620	694	683	625
-109.69	972	964	924	972	962	919	974	966	925
-99.72	1217	1213	1190	1216	1212	1186	1218	1215	1191
-89.78	1435	1441	1435	1434	1441	1433	1435	1442	1435
-79.88	1698	1701	1689	1698	1700	1698	1699	1703	1698
-69.93	1982	1973	1970	1982	1974	1970	1983	1975	1970
-60.01	2272	2253	2242	2272	2253	2242	2273	2254	2243
-50.04	2530	2507	2489	2529	2502	2489	2532	2508	2489
-40.10	2781	2774	2763	2781	2775	2763	2783	2776	2763
-35.14	2890	2895	2892	2890	2894	2892	2892	2895	2892
-30.16	2948	2967	2979	2947	2965	2978	2951	2966	2978

4 MHZ

DBM	X ANTENNA COLD	X ANTENNA MED	X ANTENNA HOT	Y ANTENNA COLD	Y ANTENNA MED	Y ANTENNA HOT	Z ANTENNA COLD	Z ANTENNA MED	Z ANTENNA HOT
-135.05	392	380	361	393	382	361	394	380	360
-130.06	447	430	399	448	429	401	449	428	400
-120.12	701	690	637	696	690	637	698	689	637
-110.20	980	974	938	980	975	938	979	974	937
-100.23	1446	1224	1204	1225	1225	1204	1224	1223	1203
-90.29	1626	1451	1447	1444	1454	1448	1444	1452	1447
-80.38	1709	1713	1712	1709	1714	1712	1707	1715	1712
-70.44	1993	1986	1984	1993	1987	1983	1993	1986	1983
-60.52	2280	2263	2255	2282	2265	2252	2281	2263	2254
-50.55	2540	2518	2501	2540	2519	2501	2539	2518	2501
-40.63	2789	2783	2774	2789	2784	2774	2789	2784	2774
-35.66	2895	2900	2900	2895	2901	2899	2895	2901	2899
-30.67	2947	2967	2980	2950	2966	2978	2952	2969	2978

8 MHZ

DBM	X ANTENNA COLD	X ANTENNA MED	X ANTENNA HOT	Y ANTENNA COLD	Y ANTENNA MED	Y ANTENNA HOT	Z ANTENNA COLD	Z ANTENNA MED	Z ANTENNA HOT
-134.86	437	427	401	433	430	401	435	430	403
-129.86	515	505	464	515	501	462	515	502	461
-119.98	789	795	747	789	795	747	792	796	745
-110.02	1059	1055	1018	1060	1055	1019	1060	1055	1019
-100.04	1292	1294	1283	1292	1295	1283	1293	1295	1284
-90.11	1524	1532	1527	1524	1532	1528	1527	1533	1528
-80.18	1797	1799	1798	1796	1801	1800	1798	1802	1798
-70.26	2083	2074	2068	2084	2074	2069	2084	2075	2068
-60.34	2361	2340	2327	2361	2341	2327	2361	2341	2327
-50.36	2618	2601	2583	2618	2600	2583	2618	2601	2583
-40.41	2855	2857	2851	2855	2857	2851	2856	2858	2851
-35.44	2937	2950	2955	2938	2949	2955	2938	2951	2955
-30.47	2949	2970	2984	2954	2969	2983	2953	2973	2983

16 MHZ

DBM	X ANTENNA COLD	X ANTENNA MED	X ANTENNA HOT	Y ANTENNA COLD	Y ANTENNA MED	Y ANTENNA HOT	Z ANTENNA COLD	Z ANTENNA MED	Z ANTENNA HOT
-134.58	430	430	408	433	437	412	433	431	410
-129.60	504	501	468	521	511	480	514	510	471
-119.80	777	795	756	780	801	765	781	797	758
-109.91	1051	1059	1033	1053	1061	1035	1053	1059	1032
-99.92	1285	1299	1297	1286	1300	1299	1287	1300	1298
-90.00	1517	1538	1543	1518	1538	1545	1519	1539	1544
-80.11	1799	1813	1817	1802	1810	1821	1799	1812	1822
-70.31	2076	2081	2082	2078	2080	2082	2077	2081	2083
-60.42	2354	2344	2338	2354	2345	2340	2354	2345	2339
-50.43	2612	2605	2596	2612	2605	2597	2612	2605	2596
-40.51	2853	2865	2868	2854	2865	2867	2854	2865	2866
-35.54	2940	2956	2967	2940	2955	2966	2940	2955	2965
-30.57	2955	2971	2987	2953	2969	2984	2952	2969	2983

222:

32 MHZ

DBM	COLD	ANTENNA	HOT	COLD	ANTENNA	HOT	COLD	ANTENNA	HOT
-133.45	415	399	378	430	411	385	423	407	383
-128.49	499	464	410	518	480	442	507	474	443
-119.06	762	745	697	770	760	698	769	755	700
-109.26	1044	1030	985	1046	1030	987	1044	1030	987
-99.34	1282	1276	1238	1282	1277	1257	1281	1276	1256
-89.42	1513	1513	1499	1513	1513	1501	1512	1513	1501
-79.62	1806	1800	1788	1782	1803	1789	1802	1794	1791
-70.21	2066	2048	2034	2064	2044	2031	2064	2042	2034
-60.41	2337	2310	2274	2338	2308	2290	2336	2308	2290
-50.49	2593	2566	2488	2593	2568	2540	2592	2565	2539
-40.57	2834	2826	2756	2835	2825	2810	2833	2825	2809
-35.62	2928	2931	2927	2925	2929	2968	2925	2930	2925
-30.65	2950	2969	2981	2949	2967	2980	2951	2969	2980

frequency pairs (32, 16), (8, 4), (2, 1) during calibration (CAL) and transmitter off (TXOFF) frames. The 1's digit, R, tells whether the receiver re-synced at the beginning of the record. The MODE array is the first array on the tape.

The TEMP array is self-explanatory. It gives the temperature in degrees Fahrenheit. It is the second array on the tape.

The TXOFF data fill 6 tape records (records 3-8). These were recorded during times when the transmitter was turned off, but the receiver was active. Figure 8 shows the contents of the TXOFF records.

Figure 8

TXOFF ARRAY - TRANSMITTER OFF DATA

6 RECORDS · 386 WORDS/RECORD

	MODE DIGIT A	⇒	FREQ
X	1		32
Y	2		8
Z	3		2
X	1		16
Y	2		4
Z	3		1

The TXOFF arrays were fully calibrated (converted to dB) in the same way as the science VCO data. This procedure required that the TXOFF data be present in core as the interpolation was being done for each frequency - antenna combination (the alternative was to generate the dB vs. VCO tables twice). The TXOFF array was therefore read into core preceeding the science VCO data, calibrated along with the science VCO data back to its position following the TEMP array.

The CAL array consists of three kinds of data: 1) receiver front-end noise measurement, 2) noise-diode source amplified by 20 dB, and 3) noise diode source unamplified. It contains 6 records, as diagrammed in Figure 9. These are the 9th-14th records on tape. Because these data are intended for use in calibrating the VCO - vs. - input power characteristics of the receiver, CAL data were left as VCO frequencies.

Figure 9

CAL ARRAY - CALIBRATION DATA

6 RECORDS

386 WORDS/RECORD

	MODE DIGIT A	⇒	FREQ
G	1		32
NA	2		8
N	3		2
G	1		16
NA	2		4
N	3		1

G = Input grounded (front-end noise)
NA = Noise diode amplified 20 dB
N = Noise diode unamplified

Drum file DB-2 represented the full set of final science data. It was copied verbatim to the second file of the distribution tapes SEPD07 - D10.

For use in scientific interpretation, the turn at EP4, which was "removed" from the navigation data, had to be specially handled for scientific data as well. The treatment which was applied was: the turn was identified by the stops preceeding and following the turn. The last values which existed prior to the execution of the turn, during the stop, were repeated through the time when the turn was completed. This gives the appearance for plotting, that the turn was not made. This function was performed by the STRAIGHTEN routine. The output was stored in drum file DB-2-STR.

The output from the navigation data processing, drum file ARROW-RANGES, was merged with DB-2-STR and stored in drum file TRAVERSE by MERGER.

The routine BCDTAPE was used in two versions (which differed only in the number of records they processed) to convert the binary data in TRAVERSE and DB-2 into BCD mode for tape transmission. Title arrays were placed at the beginning of each output file, named TRAVERSE-BCD and DB-2-BCD, respectively. These files were copied to tapes SEPD07-D10 for transmission.

The detailed format of transmission tapes SEPD07-D10 is given in Figure 10.

Figure 10

Distribution Tape Format

Tapes SEPD07, SEPD08, SEPD09, SEPD010

7-Track
Even Parity
800 BPI
BCD
Unlabeled
Fixed Unblocked Records
Record Size = Block Size = 386 - 6 char words
= 2316 chars.

Two Files

First File: Straightened Science & Nav. Data.
Second File: Unstraightened Science Data only.

FILE 1

RECORD	FORMAT	CONTENTS	
1	27(14A6),8A6	TITLE:	27 card images plus padding.
2	386I6	MODE:	$(MAR)_1$. M = 1 Data acquisition mode M = 2 Sync acquisition mode A = 1 X antenna, 32 or 16 MHz A = 2 Y antenna, 8 or 4 MHz A = 3 Z antenna, 2 or 1 MHz R = 0 No receiver resync R = 1 Receiver resync
3	386F6.1	TEMP	Temperature, degrees F.
4	386F6.1	TXOFF	X antenna, 32, 8, or 2 MHz
5	"	"	Y antenna, " " " MHz
6	"	"	Z antenna, " " " MHz
7	"	"	X antenna, 16, 4, or 1 MHz
8	"	"	Y antenna, " " " MHz
9	"	"	Z antenna, " " " MHz ------------- Determined by "A" digit of mode word
10	"	CAL	Front-end noise, 32, 8 or 2 MHz
11	"	"	Diode + 20 DB, " " "
12	"	"	Diode, " " "
13	"	"	Front-end noise, 16, 4, or 1 MHz

14	386F6.1	CAL		Diode + 20 DB, 16, 4, or 1 MHz
15	"	"		Diode, " " "
16	"	$RANGE_1$		Range array for 1 MHz data, meters
17	"	1MHz	Endfire	X antenna power in dBm
18	"	"	"	Y " "
19	"	"	"	Z " "
20	"	"	Broadside	X " "
21	"	"	"	Y " "
22	"	"	"	Z " "
23	"	$RANGE_2$		Range array for 2 MHz data
24	"	2 MHz	Endfire	X
25	"	"	"	Y
26	"	"	"	Z
27	"	"	Broadside	X
28	"	"	"	Y
29	"	"	"	Z
30	2(386F6.1/)	$RANGE_4$		
31				
32	"	4 MHz	Endfire	X
33				

34	2(386F6.1/)	4 MHz Endfire Y
35		
.		
.		
.		
44		
45	4(386F6.1/)	$RANGE_8$
46		
47		
48		
.	"	
51		8 MHz Endfire X
.		
.		
72		
.	8(386F6.1/)	$RANGE_{16}$
79		
80		
.		
87	"	16 MHz Endfire X
.		

128 . . 140	13(386F6.1/)	$RANGE_{32}$
141 . . 153	"	32 MHz Endfire X
. .		
206 . 218	"	32 MHz Broadside Z

End-of-file

FILE 2

Structured exactly like File 1 except range arrays are absent.

RECORD	CONTENTS	RECORD	CONTENTS
1	TITLE	112	32 MHz
2	MODE	.	"
3	TEMP	189	
4	TXOFF		
.	"		
10	CAL		
.	"		
16	1 MHz		
.	"		
22	2 MHz		
.	"		
28	4 MHz		
.	"		
40	8 MHz		
.	"		
64	16 MHz		
.	"		

APPENDIX

Programs for processing science (VCO) data and merginining it with Nav Data.

FREQ

```
U366RD**X*1 IR.FREQ
      FUNCTION FREQ(X,IND)
      INTEGER VCOPER,REFPER,SIG,STERR,UPER
      ERR(Y)=ABS(Y-VCOPER)
      REAL X(2)
      IND=0
      SIG=100000*FLD(24,4,X(2))
      VCOPER=SIG+1000*FLD(2,4,X(1))+
     .      100*FLD(8,4,X(1))+10*FLD(14,4,X(1))+FLD(20,4,X(2))
      REFPER=1000*FLD(20,4,X(1))+100*FLD(26,4,X(1))+
     .      10*FLD(32,4,X(1))+FLD(20,4,X(2))
      REFPER=REFPER+SIG
      STERR=ERR(REFPER)
      IF(STERR-ERR(REFPER+10000))1,1,2
    1 IF(STERR-ERR(REFPER-10000))3,3,4
    2 REFPER=REFPER+10000



      GO TO 1
    4 REFPER=REFPER-10000
    3 UPER=1000*FLD(2,4,X(2))+10*FLD(8,4,X(2))+FLD(14,4,X(2))
      IF(UPER)10,10,
      IF(REFPER)10,10,
      IF(VCOPER)10,10,
      FREF=FLOAT(UPER)/REFPER
      FREQ=5213.04/VCOPER/FREF
      IF(ABS(FREQ-1450.).LT.1350.)GO TO 5
      IND=1
      FREQ=MAX(300.,MIN(3000.,FREQ))
    5 IF(ABS(FREF-5213.E-6).GT.100.E-6)IND=IND+2
      IF(ERR(REFPER).GT.210)IND=IND+4
      RETURN
   10 IND=0
      FREQ=0.
      RETURN


      END
```

MODEF

```
U3AARD*V*LIB.MODEF
                FUNCTION MODEF(X)
                INTEGER X,XANT,YANT,RESYNC,MODE,ANT

C               GET BITS FROM END OF WORD
                MODE=2-FLD(35,1,X)
                XANT=1-FLD(34,1,X)
                YANT=1-FLD(33,1,X)
                RESYNC=FLD(29,1,X)

C               CHECK ERROR CONDITIONS
                IF(XANT.NE.0.AND.YANT.NE.0)GO TO 1

C               CORRECT THE MODE INDICATOR IF RESYNC WAS RECEIVED
                IF(RESYNC.EQ.1)MODE=1
C               GENERATE ANTENNA DIGIT
                ANT=3-2*XANT-YANT

C               STACK DIGITS INTO MODEF
                MODEF=100*MODE+10*ANT+RESYNC

                RETURN

C               ERRORS DETECTED IN DATA
              1 WRITE(6,600)MODE,XANT,YANT,RESYNC
            600 FORMAT(' MODE DIGIT ERROR.  RAW DATA =' , 6I1)
                MODEF=0
                RETURN
                END
```

CHANGE

```
U3AARD**ALTB.CHANGE
C       CHANGE
C       CONVERTS TAPE TO FREQUENCY AND STATUS
        REAL RECORD(411),DATA(189)
        INTEGER STATUS(189)
        EQUIVALENCE (DATA(189),MODE)

C       SKIP BAD RECORDS
      1 CALL RDWAIT(4,RECORD,411,$2,$2,$2)
        GO TO 1


C       PROCESS 387 RECORDS
      2 DO 100 IREC=1,387

C       READ A RECORD
        CALL RDWAIT(4,RECORD,411,$11,$10,$10)
        GO TO 12

C       END OF FILE - UNEXPECTED
     11 WRITE(6,600)IREC
    600 FORMAT(' AT RECORD',I4)
        STOP

C       ERROR RECORD - SET FREQUENCIES TO ZERO, STATUS TO 16
     10 WRITE(6,400)IREC
        DO 13 IWORD=1,189
        DATA(IWORD)=0.
     13 STATUS(IWORD)=16
C       GO WRITE RECORD
        GO TO 100

C       CONVERT GOOD RECORD
     12 DO 14 IWORD=1,188
     14 DATA(IWORD)=FREQ(RECORD(2*IWORD-1),STATUS(IWORD))
        MODE=MODEF(RECORD(380))
        STATUS(189)=0

C       WRITE DATA
    100 WRITE(7)DATA,STATUS

C       PROCESSING COMPLETED


        ENDFILE 7
        WRITE(6,601)
    601 FORMAT(' 387 RECORDS PROCESSED'/
     *        ' END OF FILE WRITTEN ON UNIT 7')
        STOP
        END
```

MERGE4

1/3

```
U3A6RD**91 18.MERGE4
C     MERGE4
C     MERGES FINAL FOUR SCIENCE TAPES
      REAL DATA(4,199),OUTPUT(180)
      INTEGER STATUS(4,180),COUNTS(17)/17*0/,MODE(4),
     .      DIST(41,27)/1107*0/,RECORD,TAPE,WORD,UNIT
      EQUIVALENCE (DATA(1,199),MODE(1))

C     FOR EACH INPUT RECORD
      DO 1 RECORD=1,387

C     READ EACH TAPE
      DO 2 TAPE=1,4
      UNIT=6+TAPE
    2 READ(UNIT)(DATA(TAPE,WORD),WORD=1,199),(STATUS(TAPE,WORD),
     .      WORD=1,180)

C     CHECK MODES FOR AGREEMENT
      DO 3 TAPE=2,4
      IF(MODE(TAPE).EQ.MODE(1))GO TO 3
      IF(STATUS(TAPE,1).EQ.16)GO TO 3
C     MODE ERROR
      WRITE(6,600)RECORD,MODE(1),TAPE,MODE(TAPE)
  600 FORMAT(' MODES DON''T AGREE AT RECORD',I4/
     .      ' MODE 1 =',I4,'   MODE',I2,' =',I4)
      STOP
    3 CONTINUE

C     FOR EACH WORD IN A RECORD, PROCESS THE DATA
      DO 4 WORD=1,180
    4 CALL PROCES(DATA(1,WORD),STATUS(1,WORD),OUTPUT(WORD))

C     OUTPUT THE RESULTS
    1 WRITE(4)OUTPUT,MODE(1)

C     FINISH PROCESSING
      ENDFILE 4
      WRITE(6,601)
  601 FORMAT(' END OF FILE WRITTEN ON UNIT 4'/
     .      ' 387 RECORDS PROCESSED.')

C     PRINT STATUS STATISTICS
      WRITE(6,602)(I,COUNTS(I+1),I=0,16)
  602 FORMAT('0STATUS COUNTS'/(I15,I9))

C     SAVE ERROR DISTRIBUTIONS FOR PLOTTING
      WRITE(3)DIST
      ENDFILE 3


C------------------------------------------------------------------
```

```
      SUBROUTINE PROCES(DATA,STATUS,OUTPUT)
C     SELECTS AND PROCESSES DATA
      REAL DATA(N),STACK(N)
      INTEGER COUNT,PSTAT,PIVOT,SECOND,COUNTS,PIVOT1,
     .      STATUS(N),UNIT(N)

C     COUNT VALUES AT MINIMUM ERROR LEVEL AND PLACE THEM
C     IN STACK.
      COUNT=1
C     MOVE FIRST VALUE INTO STACK, SET PSTAT
      STACK(1)=DATA(1)
      PSTAT=STATUS(1)
      UNIT(1)=0

C     FOR OTHER VALUES
      DO 1 IVAL=2,N

C     COMPARE NEW STATUS TO PSTAT
      IF(STATUS(IVAL)-PSTAT)2,3,1

C     NEW ERROR LEVEL IS LOWER - CLEAR THE LIST
    2 COUNT=0

C     NEW ERROR LEVEL IS SAME
C     INCREMENT COUNTER AND ADD TO LIST
    3 COUNT=COUNT+1
      STACK(COUNT)=DATA(IVAL)
      PSTAT=STATUS(IVAL)
      UNIT(COUNT)=IVAL-1

C     PROCEED FOR REMAINING VALUES
    1 CONTINUE

C     KEEP STATUS STATISTICS
      COUNTS(PSTAT+1)=COUNTS(PSTAT+1)+1

C     WAS SINGLE VALUE BEST
      IF(COUNT.NE.1)GO TO 4

C     YES - USE SINGLE VALUE
      OUTPUT=STACK(1)

C     IF ZERO STATUS, RETURN
      IF(PSTAT.EQ.0)RETURN

C     NON-ZERO STATUS MESSAGE
      WRITE(6,600)RECORD,WORD,PSTAT,UNIT(1),OUTPUT
  600 FORMAT(' RECORD',I4,' WORD',I4,' STATUS',I3,
     .      2(' TAPE',I2,' FREQ',F6.0))
      RETURN
```

```
C      MULTIPLE VALUES OF SAME STATUS
C      SET MINIMUM DIFFERENCE TO MAX, THEN SEEK ACTUAL MIN
     4 DIFMIN=200.

C      FOR EACH PIVOT VALUE
       COUNT1=COUNT-1
       DO 5 PIVOT=1,COUNT1
       PIVOT1=PIVOT+1
C      FOR EACH SECOND VALUE ABOVE PIVOT
       DO 6 SECOND=PIVOT1,COUNT

C      IS DISCREPANCY LARGER FOR THIS PAIR OF VALUES
       DIF=STACK(PIVOT)-STACK(SECOND)
       IF(ABS(DIF).GE.DIFMIN)GO TO 6

C      SMALLER - MAKE SUBSTITUTIONS, USE AVERAGE
       DIFMIN=ABS(DIF)
       DIFT=DIF
       FREQA=STACK(PIVOT)
       ITAPEA=UNIT(PIVOT)
       FREQB=STACK(SECOND)
       ITAPEB=UNIT(SECOND)
       OUTPUT=(FREQA+FREQB)/2.

C      CONTINUE FOR MORE PIVOTS AND SECONDS
     6 CONTINUE
     5 CONTINUE

C      KEEP DIFFERENCE STATISTICS
       IFREQ=(OUTPUT-300.)/100.+1.
       IFREQ=MIN(27,MAX(1,IFREQ))
       IDIF=DIFT/25.+21.5
       IDIF=MIN(41,MAX(1,IDIF))
       DIST(IDIF,IFREQ)=DIST(IDIF,IFREQ)+1

C      PRINT MESSAGE FOR LARGE DIFFERENCES OR NON-ZERO STATUS
       IF(DIFMIN.LT.50..AND.PSTAT.EQ.0)RETURN
       WRITE(6,400)RECORD,WORD,PSTAT,ITAPEA,FREQA,ITAPEB,FREQB


       RETURN

       END
```

1/2

PRPL

```
C      PRPL
C      PLOTS ACCURACY STATISTICS ON PRINTER-PLOTTER
       PARAMETER LWIDE=101
       INTEGER DMAX,TOT,BLANK/' '/,BAR/'I'/,STAR/'*'/,
     *        DIST(41,27),LINE(LWIDE),FORM1(10),FORM2(10)

C      SET UP FORMATS
       LWIDER=LWIDE
       ENCODE(400,FORM1)LWIDER
   400 FORMAT('(5X,',I3,'(1H-))')
       ENCODE(401,FORM2)LWIDER
   401 FORMAT('(I5,',I3,'A1,F5.2)')

C      READ ERROR DISTRIBUTION
       READ(4)DIST

C      LOOP THROUGH PLOTS
       DO 1 J=1,27
       JFREQ=300+100*(J-1)

C      SET UP LINE BUFFER
       LINE(1)=BAR
       DO 2 I=2,LWIDE
     2 LINE(I)=BLANK
       LINE(LWIDE)=BAR

C      LOCATE MAXIMUM VALUE ANDTOTAL WEIGHT
       DMAX=DIST(1,J)
       TOT=DMAX
       DO 3 I=2,41
       TOT=TOT+DIST(I,J)
     3 DMAX=MAX(DMAX,DIST(I,J))
       DDMAX=DMAX
       IF(TOT)1,1,
       ATOT=TOT/100.
```

2/2

```
C
C      PRINT TOP LINE
       WRITE(6,600)JFREQ
   600 FORMAT('1FREQUENCY INTERVAL (100HZ.) STARTS AT',I5,'HZ.')
C      PRINT HEADING
       WRITE(6,FORM1)

C      PRINT LINES
       DO 4 I=1,41
       IFREQ=5*(I-21)
       IPOS=1.5+DIST(I,J)*(LWIDE-1)/DDMAX
       IPOS=MIN(LWIDE,MAX(1,IPOS))
       IF(IPOS.EQ.1)GO TO 9
       DO 8 II=2,IPOS
     8 LINE(II)=STAR
     9 X=DIST(I,J)/ATOT
       WRITE(6,FORM2)IFREQ,LINE,X
       DO 10 II=2,IPOS
    10 LINE(II)=BLANK
       LINE(LWIDE)=BAR
     4 CONTINUE

C      WRITE BOTTOM LINE
       WRITE(6,FORM1)
C
C      COMPUTE MEAN ABSOLUTE ERROR
       IERR=0
       TOT=0
       DO 20 I=2,40
       K=DIST(I,J)
       TOT=TOT+K
    20 IERR=IERR+K*IABS(I-21)
       ERR=(5.*IERR)/TOT
       WRITE(6,601)ERR
   601 FORMAT(' AVERAGE ABSOLUTE ERROR =',F5.1,'HZ.')

     1 CONTINUE
       END
```

1/3

DEMUX

```
@ARD*XQL TR.DEMUX
C     DEMUX
C     DEMULTIPLEXES MODE, VCO DATA FROM MERGE4 OUTPUT
      INTEGER ST/15/,SC/16/,S1/31/,S2/29/,S4(2)/1,23/,
     .     S8(4)/3,11,19,27/,S16(8)/1,5,9,13,17,21,25,29/,
     .     S32(13)/2,4,6,8,10,12,14,18,20,22,24,26,30/,
     .     IDUMP
      REAL RECORD(189),REC(6,1),MODE(387),TEMP(387),
     .     T(50,6),C(50,6),D1(50,6),D2(50,6),D4(2,50,6),
     .     D8(4,50,6),D16(8,50,6),D32(13,50,6)
      EQUIVALENCE (RECORD,REC)

      DEFINE FILE 3(1600,50,U,IPOINT)

C     CLEAR CORE AND DRUM ARRAY COUNTERS
      ICORE=0
      IDRUM=0

C     FOR EACH INPUT RECORD
      DO 1 IREC=1,387

C     READ THE RECORD
      READ(4)RECORD

C     COUNT THE CORE ARRAY
      ICORE=ICORE+1


C     MOVE THE DATA INTO CORE ARRAYS
      MODE(IREC)=RECORD(189)
      TEMP(IREC)=RECORD(188)
      CALL MOVER(T,ST,1)
      CALL MOVER(C,SC,1)
      CALL MOVER(D1,S1,1)
      CALL MOVER(D2,S2,1)
      CALL MOVER(D4,S4,2)
      CALL MOVER(D8,S8,4)
      CALL MOVER(D16,S16,8)
      CALL MOVER(D32,S32,13)

C     IF CORE IS NOT FULL, PROCESS NEXT RECORD
      IF(ICORE.NE.50)GO TO 1

C     CORE IS FULL -- DUMP TO DRUM
C     COUNT DRUM RECORD, RESET CORE COUNTER
      IDRUM=IDRUM+1
      ICORE=0

      CALL DRUMMR

C     FINISH ALL DATA
    1 CONTINUE

C     EMPTY FINAL ARRAYS
      IDRUM=IDRUM+1
      CALL DRUMMR
```

```
C       DUMP TO TAPE
        WRITE(7)MODE
        WRITE(7)IFMO
        CALL TAPEW
        ENDFILE 7
        WRITE(6,6000)
  6000  FORMAT(' END OF FILE WRITTEN ON UNIT 7'/
     *        ' END OF DEMULTIPLEXING.')
        STOP


        SUBROUTINE MOVE(DATA,SOURCE,SIZE)
        INTEGER SIZE,SOURCE(SIZE)
        REAL DATA(SIZE,10,6)
        DO 1 I=1,SIZE
        J=SOURCE(I)
        DO 1 ICOMP=1,6
     1  DATA(I,ICORE,ICOMP)=REC(ICOMP,J)
        RETURN


        SUBROUTINE DRUMW
        WRITE(6,6000)IDRUM
  6000  FORMAT(' MOVING RECORDS TO DRUM, IDRUM=',I2)
        JUMP=0
        CALL DRUM(T,1)
        CALL DRUM(C,1)
        CALL DRUM(D1,1)
        CALL DRUM(D2,1)
        CALL DRUM(D4,2)
        CALL DRUM(D8,4)
        CALL DRUM(D16,8)

        CALL DRUM(D32,17)
        RETURN
```

3/3

```
      SUBROUTINE DRUM(DATA,SIZE)
      INTEGER SIZE,DEST
      REAL DATA(50,SIZE,6)
      DEST=JUMP+(IDRUM-1)*SIZE
      DO 1 ICOMP=1,6
      DO 2 J=1,SIZE
    2 CALL TRANS(DATA(1,J,ICOMP),DEST+J)
    1 DEST=DEST+8*SIZE
      JUMP=JUMP+48*SIZE
      RETURN



      SUBROUTINE TRANS(DATA,RECORD)
      INTEGER RECORD,DATA(50)
      WRITE(3'RECORD)DATA
      FIND(3'IPOINT)
      RETURN



      SUBROUTINE TAPE9
      REAL DATA(50,8),OUTPUT(387)
      EQUIVALENCE (DATA,OUTPUT)
      IREC=1
      WRITE(6,600)
  600 FORMAT(' WRITING DATA ON TAPE.')
      DO 1 I=1,31
      DO 1 J=1,6
      DO 2 K=1,8
      CALL INPUT(DATA(1,K),IREC)
    2 IREC=IREC+1
    1 WRITE(7)OUTPUT
      RETURN



      SUBROUTINE INPUT(DATA,IREC)
      REAL DATA(50)
      READ(3'IREC)DATA
      FIND(3'IPOINT)
      RETURN


      END
```

CALIBPRT

```
U36ARD*IO*CCPLIB.CALIBPRT
      REAL DB(13,6)
      INTEGER VCO(13,3,3,6)

C     INPUT D-LEVEL DATA
      DO 1 IFREQ=1,6
      READ(3,300)(DB(I,IFREQ),I=13,1)
  300 FORMAT(7F7.2)
C     CHANGE SIGNS
      DO 1 I=1,13
    1 DB(I,IFREQ)=-DB(I,IFREQ)

C     INPUT VCO DATA
      DO 2 ITEMP=3,1
      DO 2 IFREQ=1,6
      DO 2 IANT=1,3
    2 READ(4,400)(VCO(I,IANT,ITEMP,IFREQ),I=13,1)
  400 FORMAT(7I5)

C     OUTPUT BY FREQ
      DO 3 IFREQ=1,6
      JFREQ=2.*(IFREQ-1)
      WRITE(6,600)JFREQ
  600 FORMAT(////'1',I2,' GHZ'/
     .      T10,'X ANTENNA',T32,'Y ANTENNA',T49,'Z ANTENNA'/
     .      '    DBM',T9,3(3X,'COLD  MED  HOT'),//)

C     OUTPUT THE DATA
    3 WRITE(6,601)(DB(I,IFREQ),
     .      ((VCO(I,IANT,ITEMP,IFREQ),ITEMP=1,3),
     .      IANT=1,3),I=1,13)
  601 FORMAT((F8.2,3(1X,3I5)))

      STOP
      END
```

REBLOCK

```
0366RDNY*LIB.REBLOCK
      C     REBLOCK
      C     COPIES DATA DELETING FIRST RECORD AND SHIFTING MODE
            INTEGER MODE(387),SIZE(6)/1,1,2,4,8,16/
            REAL DATA(387,13)

      C     COPY MODE
            READ(7)MODE
            CALL WRITR(MODE)

      C     COPY TEMP
            CALL COPY(1)

      C     COPY TXOFF, CAL, 1, AND 2 MHZ DATA
            DO 1 I=1,24
          1 CALL COPY(1)

      C     COPY HIGHER FREQUENCY DATA
            DO 2 IFREQ=3,6
            ISIZE=SIZE(IFREQ)
            DO 2 ICOMP=1,6
          2 CALL COPY(ISIZE)

            ENDFILE 8
            WRITE(6,600)
        600 FORMAT(' END OF FILE WRITTEN ON UNIT 8')
            STOP


            SUBROUTINE COPY(N)
            DO 1 I=1,N
          1 CALL READR(DATA(1,I))
            CALL OUT(DATA(N+1,1),N)
            RETURN

            SUBROUTINE OUT(DATA,N)
            REAL DATA(386,N)
            DO 1 I=1,N
          1 CALL WRITR(DATA(1,N))
            RETURN

            SUBROUTINE READR(D)
            REAL D(387)
            READ(7)D
            RETURN

            SUBROUTINE WRITR(D)
            REAL D(386)
            WRITE(8)D
            RETURN

            END
```

```
JAGRDMX*LIB.CALIBRATE
C      CALIBRATE
C      USES DR-VS-VCO DATA TO CALIBRATE POWER LEVELS
       INTEGER MODE(386),SIZE(6)/1,1,2,4,8,12/,
      .     TXOTAB(6)/3,3,2,2,1,1/,TXOTAC(6)/1,2,1,2,1,2/
       REAL TEMP(386),TXOFF(384,2,2),WORK(384),
      .     TABLE(2,2201,3),DR(13,6),VCO(13,2,3,6)

C      READ CALIBRATION DATA
       CALL SETUP

C      READ MODE ARRAY
       READ(7)MODE
C      COPY MODE ARRAY
       WRITE(8)MODE

C      READ TEMP ARRAY
       READ(7)TEMP
C      CONVERT TO TEMPERATURE
       DO 1 I=1,386
     1 TEMP(I)=.073*TEMP(I)-47
C      WRITE TEMP ARRAY
       WRITE(8)TEMP

C      READ TXOFF ARRAYS
       DO 2 ITRANS=1,2
       DO 2 ICOMP=1,2
     2 CALL READR(TXOFF(1,ICOMP,ITRANS))

C      COPY CAL ARRAYS
       DO 3 ICOMP=1,6
       READ(7)WORK
     3 WRITE(8)WORK

C      FOR EACH FREQUENCY
       DO 10 IFREQ=1,6
       FREQ=2*(IFREQ-1)+.1*MOD(IFREQ-1,2)
       WRITE(6,600)FREQ
   600 FORMAT(' START PROCESSING FOR ',F4.1,' MHZ DATA.'/)
       ISIZE=SIZE(IFREQ)

C      COMPUTE INTERPOLATION TABLE
       DO 11 ICOMP=1,3
    11 CALL INTERP(DR(1,IFREQ),VCO(1,ICOMP,2,IFREQ),
      .       VCO(1,ICOMP,3,IFREQ),TABLE(1,1,ICOMP))

C      INTERPOLATE THE SCIENCE DATA
       DO 12 ITRANS=1,2
       DO 12 ICOMP=1,3
       CALL REFER(TABLE(1,1,ICOMP))
       DO 12 IREC=1,ISIZE
       READ(7)WORK
       DO 13 I=1,386
    13 WORK(I)=POW(WORK(I),TEMP(I))
    12 WRITE(8)WORK
```

2/4

```
C     INTERPOLATE THE TXOFF DATA
      DO 14 I=1,[illegible]
      M=MODE(I)
      MA=MOD(M,100)/10
C     MA IS THE MODE ANTENNA DIGIT
C     IF ANTENNA DIGIT INDICATES OTHER FREQUENCIES, SKIP THIS PT
      IF(MA.NE.TXOTAN(IFREQ))GO TO 14

C     FREQUENCY AGREES - GET CORRESPONDING TX FRAME DIGIT
      MTX=TXOTAC(IFREQ)
C     INTERPOLATE
      DO 15 ICOMP=1,3
      CALL REFER(TABLE(1,1,ICOMP))




   15 TXOFF(I,ICOMP,MTX)=POW(TXOFF(I,ICOMP,MTX),TEMP(I))
   14 CONTINUE

C     FINISH FOR ALL FREQUENCIES

   10 CONTINUE


C     WRITE TXOFF DATA
      DO 16 ITRANS=1,2
      DO 16 ICOMP=1,3
   16 CALL WRITR(TXOFF(1,ICOMP,ITRANS))

      WRITE(6,601)
  601 FORMAT(' END OF FILE WRITTEN ON UNIT 8'/
     *       ' END OF CALIBRATION PROCESSING')
      ENDFILE 8
      STOP



      SUBROUTINE SETUP
C     READS CALIBRATION DATA ARRAYS
      READ(3,300)((DB(I,IFREQ),I=13,1),IFREQ=1,6)
  300 FORMAT(7F7.2)
C     CHANGE SIGNS
      DO 1 IFREQ=1,6
      DO 1 I=1,13
    1 DB(I,IFREQ)=-DB(I,IFREQ)

C     READ VCO DATA
      DO 2 ITEMP=2,1
      DO 2 IFREQ=1,6
      DO 2 IANT=1,3
    2 READ(4,400)(VCO(I,IANT,ITEMP,IFREQ),I=13,1)
  400 FORMAT(7F5.0)

      RETURN
```

```
      SUBROUTINE INTEG(DR,VCO,POWER)
C     INTERPOLATES FOR SINGLE-TEMPERATURE ARRAY
      REAL POWER(2,2701),DR(13),VCO(13),X
      INTEGER I,J,NEXT
      DEFINE P(I)=POWER(1,I)

C     FILL THE ARRAY UP TO THE FIRST VCO VALUE
      J=1
      IF(VCO(1).LT.300.)GO TO 1
      J=VCO(1)-299.
      DO 2 I=1,J
    2 P(I)=DR(1)
      J=J+1

C     INTERPOLATE UP TO THE LAST VCO VALUE
    1 DO 3 NEXT=2,13
      DRL=DR(NEXT-1)
      VCOL=VCO(NEXT-1)
      VCON=VCO(NEXT)
      X=(DR(NEXT)-DRL)/(VCON-VCOL)
      DO 4 I=J,2701
      IF(I+299..GT.VCON)GO TO 3
    4 P(I)=(I+299-VCOL)*X+DRL
      RETURN
    3 J=I

C     FILL THE ARRAY BEYOND THE LAST VCO VALUE
      DRL=DR(13)
      DO 6 I=J,2701
    6 P(I)=DR(13)
      RETURN



      END
```

```
      SUBROUTINE READ2(X)
      REAL X(286)
      READ(7)X
      RETURN


      SUBROUTINE  WRIT2(X)
      REAL X(286)
      WRITE(8)X
      RETURN


C     FUNCTION FOR TEMPERATURE INTERPOLATION
      FUNCTION POW(FREQ,TEMP)
      REAL POWER(2,2701)
      INTEGER I
      IF(FREQ.EQ.0.)GO TO 1
      I=FREQ-998.
      I=MIN0(2701,MAX0(1,I))



      POW=POWER(1,I)+(TEMP-65.)/40.*(POWER(2,I)-POWER(1,I))
      RETURN
    1 POW=0.
      RETURN


      ENTRY INTERP(DR,VCOWEO,VCOUOT,POWER)
C     INTERPOLATES FOR ONE FREQUENCY-ANTENNA COMBINATION

C     INTERPOLATE FOR 75 AND 105- DEGREE ARRAYS
      CALL INTERV(DR,VCOWEO,POWER)
      CALL INTERV(DR,VCOUOT,POWER(2,1))
      RETURN





C     ENTRY TO REFER TO PREVIOUS INTERPOLATIONS
      ENTRY REFER(POWER)
      RETURN
```

U3ARNDEV*LIB.ARRANGE

ARRANGE

```
C      ARRANGE
C      MOVE TXOFF DATA BACK INTO POSITION AT START OF TAPE
       REAL WORK(1024)
       INTEGER SIZE(6)/1,1,2,4,8,13/,TITLE(6)/'1 MHZ',
      .   '2 MHZ','4 MHZ','8 MHZ','16 MHZ','32 MHZ'/

C      COPY MODE, TEMP
       CALL COPY(1,'MODE')
       CALL COPY(1,'TEMP')

C      SKIP CAL
       CALL SKIP(6,'CAL')

C      SKIP SCIENCE DATA
       DO 1 IFREQ=1,6
     1 CALL SKIP(6*SIZE(IFREQ),TITLE(IFREQ))

C      COPY TXOFF
       CALL COPY(6,'TXOFF')

C      START OVER
       REWIND 7
       WRITE(6,400)
   400 FORMAT(' REWIND UNIT 7')

C      SKIP MODE AND TEMP
       CALL SKIP(1,'MODE')
       CALL SKIP(1,'TEMP')

C      COPY CAL
       CALL COPY(6,'CAL')

C      COPY SCIENCE DATA
       DO 2 IFREQ=1,6
     2 CALL COPY(6*SIZE(IFREQ),TITLE(IFREQ))

       ENDFILE 8
       WRITE(6,401)
   401 FORMAT(' END OF FILE WRITTEN ON UNIT 8'/
      .   ' END PROCESSING')


       SUBROUTINE SKIP(N,TITLE)
       WRITE(6,400)N,TITLE
   400 FORMAT(' SKIPPING',I3,' RECORDS OF ',A6,' DATA.')
       DO 1 I=1,N
     1 READ(7) WORK
       RETURN


       SUBROUTINE COPY(N,TITLE)
       WRITE(6,400)N,TITLE
   400 FORMAT(' COPYING ',I3,' RECORDS OF ',A6,' DATA.')
       DO 1 I=1,N
       READ(7) WORK
     1 WRITE(8) WORK
       RETURN

       END
```

1/3

STRAIGHTEN

```
93AARD*YOSEPLIN,STRAIGHTE
C       STRAIGHTEN
C       PROPAGATES SEISMIC DATA THROUGH THE FLOW TUBE
        REAL DATA(100,13),DSCRIP(19)
        INTEGER N,MODE,IREC(/1H6/,...IREC(/)/1,1,2,4,8,12/

C       READ AND ECHO IN
        CALL RDCATT(3,DSCRIP,19,51,51,51)
        GO TO 2
      1 STOP
      2 DECODE(1000,DSCRIP)N
   1000 FORMAT(10AX,IA)
        WRITE(6,600)(DSCRIP(I),I=1,18),N
    600 FORMAT('1MODE',A7/' SITE',A7/
     .       ' DIRECTION',A//' FORWARD OR REVERSE = ',A7/
     .       1X,19A4/' IREC = ',I6)

C       COPY IN, CHANGE IREC
        ENCODE(1001,DATA)(DSCRIP(I),I=1,19),IREC
   1001 FORMAT(19A4,IA)
        WRITE(6,601)(DATA(I,1),I=1,19)
    601 FORMAT(1X(' .'/),' NEW IN ='/
     .       4(-7/),1X,19A4/A7/3(' .'/))
        CALL WRCATT(2,DATA,19,51,51)

C       COPY MODE
        CALL COPY(1)

C       SKIP RAW DATA
        CALL SKIP(3)

C       COPY TEMP
        CALL COPY(1)

C       SKIP TIME
        CALL SKIP(1)

C       COPY THREE AND ONE
        CALL COPY(12)

C       FOR EACH FREQUENCY
```

2/3

STRAIGHTEN

```
      DO 3 IFRAM=1,6
      NREC=PERREC(IFRAM)

C     SKIP MAP
      CALL SKIP(NREC)

C     FOR EACH COMPONENT
      DO 3 ICOMP=1,6
C     CONSTRUCT, CORRECT, OUTPUT THE DATA
      CALL CON(DATA)

    3 CONTINUE

C     WRITE END OF FILE
      CALL FILEND(2)

      STOP


      SUBROUTINE SKIP(NREC)
      WRITE(6,601)NREC
  601 FORMAT(' SKIPPING',I3,' RECORDS')
      DO 2 I=1,NREC
    2 CALL RDWAIT(3,DATA,1,$1,$1,$1)
      RETURN
    1 WRITE(6,600)I
  600 FORMAT(' SKIPPING AT RECORD',I5)
      STOP


      SUBROUTINE COPY(NREC)
      WRITE(6,600)NREC
  600 FORMAT(' COPYING ',I2,' RECORDS')
      DO 2 I=1,NREC
      CALL RDWAIT(3,DATA,N,$1,$1,$1)
    2 CALL WRWAIT(2,DATA,LREC,$1,$1)
      RETURN
    1 WRITE(6,601)I
  601 FORMAT(' COPYING AT RECORD',I5)
      STOP
```

```
      SUBROUTINE COR(DATA)
C     CORRECTS ONE COMPONENT
      REAL DATA(N,NREC)
      WRITE(6,601)NREC
  601 FORMAT(' CORRECTING',I1,' RECORDS')

C     INPUT THE ARRAY
      DO 1 I=1,NREC
    1 CALL RDWAIT(3,DATA(1,I),N,&9,&9,&9)

C     CORRECT IT
      CALL CORREC(DATA)

C     OUTPUT IT
      CALL OUTPUT(DATA)

      RETURN



C     ERROR
    9 WRITE(6,609)I
  609 FORMAT(' AT RECORD',I5)
      STOP


      SUBROUTINE OUTPUT(DATA)

      REAL DATA(LREC,NREC)
      DO 1 I=1,NREC
    1 CALL WRWAIT(2,DATA(1,I),LREC,&9,&9)
      RETURN
    9 WRITE(6,609)I
  609 FORMAT(' AT RECORD',I5)
      STOP


      SUBROUTINE CORREC(DATA)
      REAL DATA(NREC,1)
      DO 1 I=49,64
      DO 1 J=1,NREC
    1 DATA(J,I)=DATA(J,63)
      RETURN


      END
```

1/2

MERGER

```
C      MERGER
C      MERGE RANGE ARRAYS FROM RANGER WITH DATA FROM STRAIGHTE
C     INPUT FROM ARROW-RANGES, SERP01
C      OUTPUT TO SERP03
       REAL RANGES(386,30),DATA(386),LABEL(19)
       INTEGER DIMS(6)/1,1,2,4,8,13/RDIMS(6)/1,2,3,5,9,17/

       READ(2)RANGES
       CALL WINDER(3)
       CALL WINDER(4)

C      SKIP OLD LABEL
       CALL RDWAIT(3,LABEL,1,$9,$8,$9)
       GO TO 8
     8 WRITE(6,609)
   609 FORMAT(' AT LABEL')
     9 CONTINUE
C      SETUP NEW LABEL
       ENCODE(1000,LABEL)
  1000 FORMAT('1.1   APLOIZREST  CORRECT,
     .      'SEP SITE TO 4.3 KM, 2ND DIGZ''N, FOR TURN REMOVED',
     .      ', LRV NAV BEST RANGES',T109,'    386')

C      OUTPUT NEW LABEL
       CALL WRWAIT(4,LABEL,19,$19,$19)

C      COPY MODE, TEMP, IXOFF, CAL
       DO 1 I=1,14
       CALL RDWAIT(3,DATA,386,$10,$10,$10)
     1 CALL WRWAIT(4,DATA,386,$10,$10)

C      FOR EACH FREQUENCY
       DO 2 IFREQ=1,6
       IDIM=DIMS(IFREQ)
       JDIM=RDIMS(IFREQ)

C      COPY RANGE
       CALL OUT(RANGES(1,JDIM),IDIM)

C      FOR EACH COMPONENT
       DO 3 ICOMP=1,6
C      COPY DATA
       DO 3 I=1,IDIM
       CALL RDWAIT(3,DATA,386,$11,$11,$11)
     3 CALL WRWAIT(4,DATA,386,$11,$11)
     2 CONTINUE

       CALL FILEND(4)

       STOP
```

MERGER 2/2

```
   10 WRITE(6,600)I
  600 FORMAT(' AT RECORD',I4)
      STOP 10
   12 STOP 12




   11 WRITE(6,601)IFREQ,ICOMP,I
  601 FORMAT(' AT FREQ',I4/' AT COMPONENT',I4/
     .       ' AT RECORD',I4)
      STOP 11


      SUBROUTINE OUT(R,I)
      REAL R(386,I)
      DO 1 J=1,I
    1 CALL WRTATT(0,R(1,J),386,$9,$9)
      RETURN
    9 STOP


      END
```

BCDTAPES

```
C      BCDTAPES
C      CONVERTS FILES WITH 386-WORD RECORDS TO BCD
       DIMENSION INPUT(386,2),OUTPUT(386,2),TITLE(386)
C
C      SET OUTPUT TAPE PARITY
       CALL PWPAR(8,6)

C      READ AND COPY TITLE
       READ(2,200)TITLE
  200  FORMAT(13A6,A2)
       CALL WRWAIT(8,TITLE,386,$9,$9)

C      READ AND COPY MODE
       CALL RDWAIT(7,INPUT,386,$9,$9,$9)
       ENCODE(100,OUTPUT)(INPUT(I,1),I=1,386)
  100  FORMAT(22I6)
       CALL WRWAIT(8,OUTPUT,386,$9,$9)

C      READ, CONVERT, AND COPY EVERYTHING ELSE
       IBUF=1
 -     CALL PREAD(7,INPUT,386,$9,$9,$9)
       DO 1 IREC=1,216
       IBUF=3-IBUF
       CALL PREAD(7,INPUT(1,IBUF),386,$10,$10,$9)
       CALL CONVRT(INPUT(1,3-IBUF),OUTPUT(1,3-IBUF))
       CALL PWRITE(8,OUTPUT(1,3-IBUF),386,$9,$9)
    1  CONTINUE
       CALL PWWAIT(8,$9,$9)
       CALL FILEND(8)
       STOP
   10  WRITE(6,600)
  600  FORMAT(' UNEXPECTED EOF  REACHED ON UNIT 7')
    9  STOP


C      SUBROUTINE FOR REAL CONVERSIONS
       SUBROUTINE CONVRT(INPUT,OUTPUT)
       REAL INPUT(386),OUTPUT(386)
       ENCODE(100,OUTPUT)INPUT
  100  FORMAT(22F6.1)
       RETURN

       END
SCAN:44
```

TAPECOPY

```
C       TAPECOPY
C      COPIES FINAL DATA TAPES
       REAL DATA(386,2)
       CALL PWPAR(9,0)
       CALL TRANS(7,218)
       CALL TRANS(8,189)
       STOP
C
C
C
       SUBROUTINE TRANS(IUNIT,NREC)
       CALL PREAD(IUNIT,DATA,386,$9,$9,$9)
       IBUF=1
       DO 1 IREC=1,NREC
       CALL PREAD(IUNIT,DATA(1,3-IBUF),386,$9,$9,$9)
       CALL WRWAIT(9,DATA(1,IBUF),386,$9,$9)
     1 IBUF=3-IBUF
       CALL FILEND(9)
       CALL PRWAIT(IUNIT,$2,$2,$2)
     2 RETURN
     9 STOP
       END
SCAN:23
    23
```

A 4(i)

Apollo 17 SEP

Data Processing

John C. Rylaarsdam

July 1974

Contents

Figures

Tables

Introduction

This report is a summary of intermediate stage processing operations performed on the data from the Apollo 17 surface electrical properties experiment. The starting points for these operations are the files designated SCI1, SCI2, and SCI3; the contents of these and all generated files, plots, and listings are summarized in table 1. File SCI1 is a preliminary release of the data; files SCI2 and SCI3 are the final data sets, respectively with and without data for the EP-4 turn, prepared by R. Watts, tape number SEPD09. (N.B. - subsequent references to removal of turn data refer to work described in this report, rather than to Watts' initial processing.)

The diagrams in figure 1 give an indication of the sequences of processing operations. More detailed information is provided by the descriptions of the programs. Annotated listings are also included, as well as precise tabulations of the formats of the various files. The program listings include descriptions of all required card input data.

In addition, two complete sets of plots (SCI2B) are included as a record of the data; in one set dB values are plotted versus the range in metres, and in the second set versus the range in wavelengths.

	LABEL	MODE	TEMP	TX-OFF	CAL	RANGE	DB	Notes
CAL1 Listing			*		*	*		
EP4							*	dB data for 490 m. <= range <= 535 m.
EP4 Listing							*	
EP4 Plot							*	
EP4A Listing							*	dB data for 490 m. <= range <= 535 m.
EP4A Plot							*	and LRV in motion
NAV1								times and odometer counts relative to beginning of traverse
RT1 Listing						*		includes VLBI data, converted
RT1 Plot						*		to ranges
SCI1	*	*	*	*	*	*	*	
SCI1 Listing	*	*	*	*	*	*	*	
SCI1A	*						*	dB data sampled at intervals of 0.1
SCI1A Listing	*						*	wavelength
SCI1A Plot							*	

Table 1 - Data set summary.

	LABEL	MODE	TEMP	TX-OFF	CAL	RANGE	DB	Notes
SCI2	*	*	*	*	*	*	*	no data for EP-4
SCI2 Listing	*	*	*	*	*	*	*	
SCI2 Plot						*	*	
SCI2A	*						*	no data for EP-4; dB data sampled at intervals of 0.1 wavelength
SCI2A Listing	*						*	
SCI2A Plot							*	
SCI2B Plot				*		*	*	
SCI3	*	*	*	*	*		*	
SCI3 Listing	*	*	*	*	*	*	*	range data from SCI2
SCI3A	*						*	dB data sampled at intervals of 0.1 wavelength
SCI3A Listing	*						*	
SCI3A Plot							*	
STAT1			*	*	*	*		also contains speed data
TX01 Listing				*		*		all data except 490 m. <= range <= 535 m. and LRV in motion
TX01 Plot				*		*		

Table 1 - Data set summary (continued).

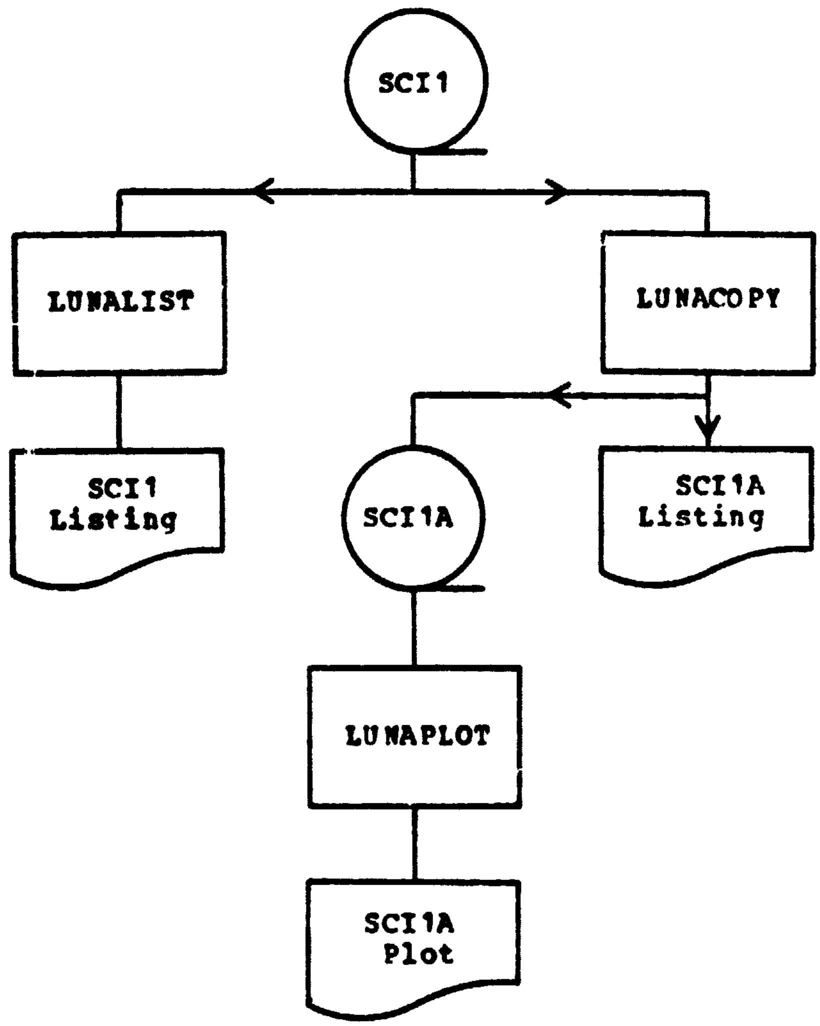


Figure 1(a) - Processing flow.

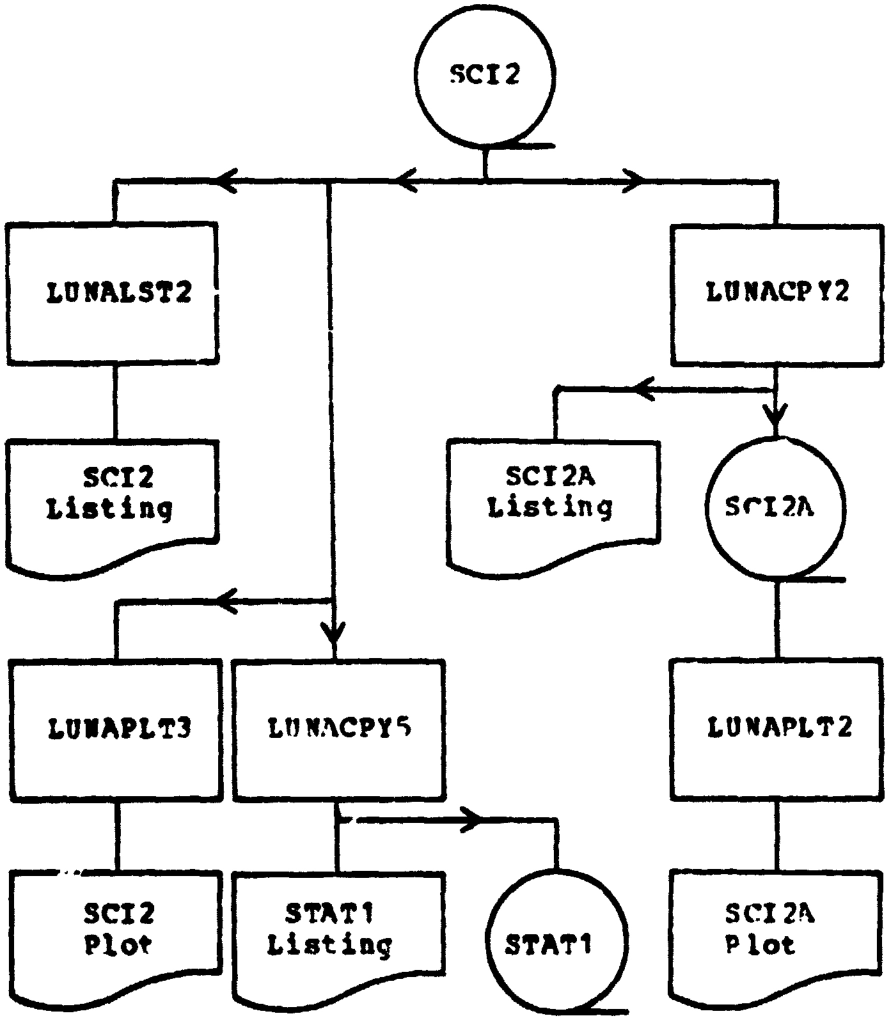


Figure 1(b) - Processing flow.

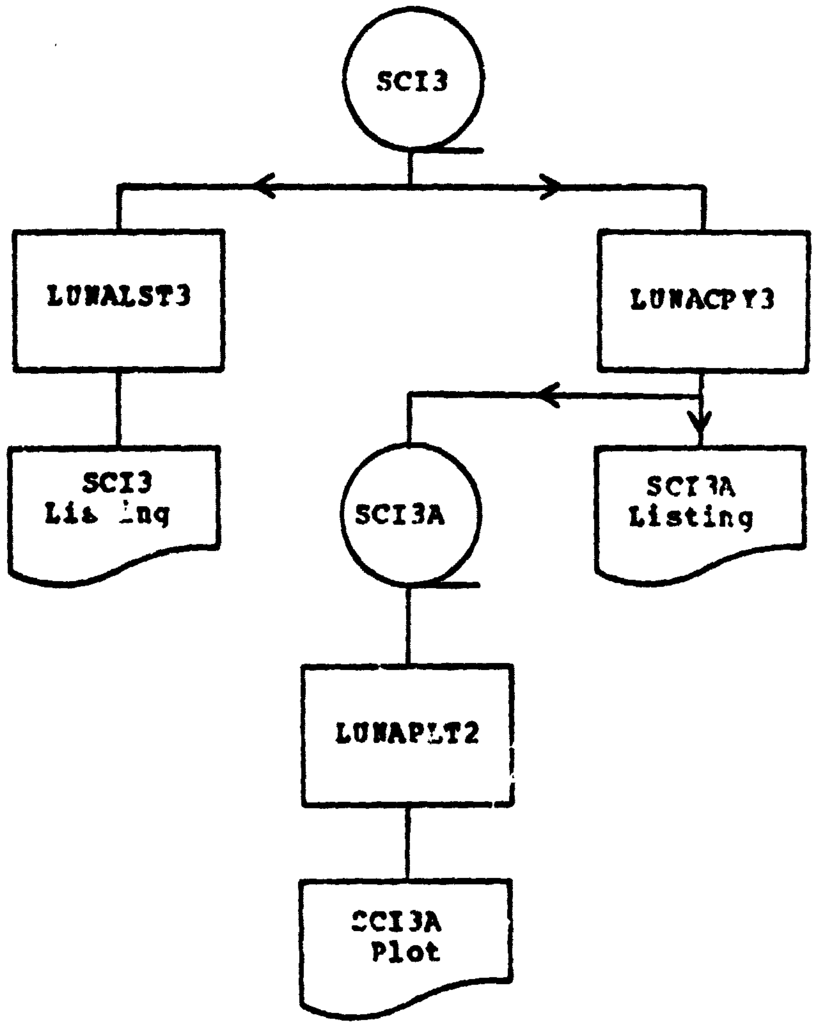


Figure 1(c) - Processing flow.

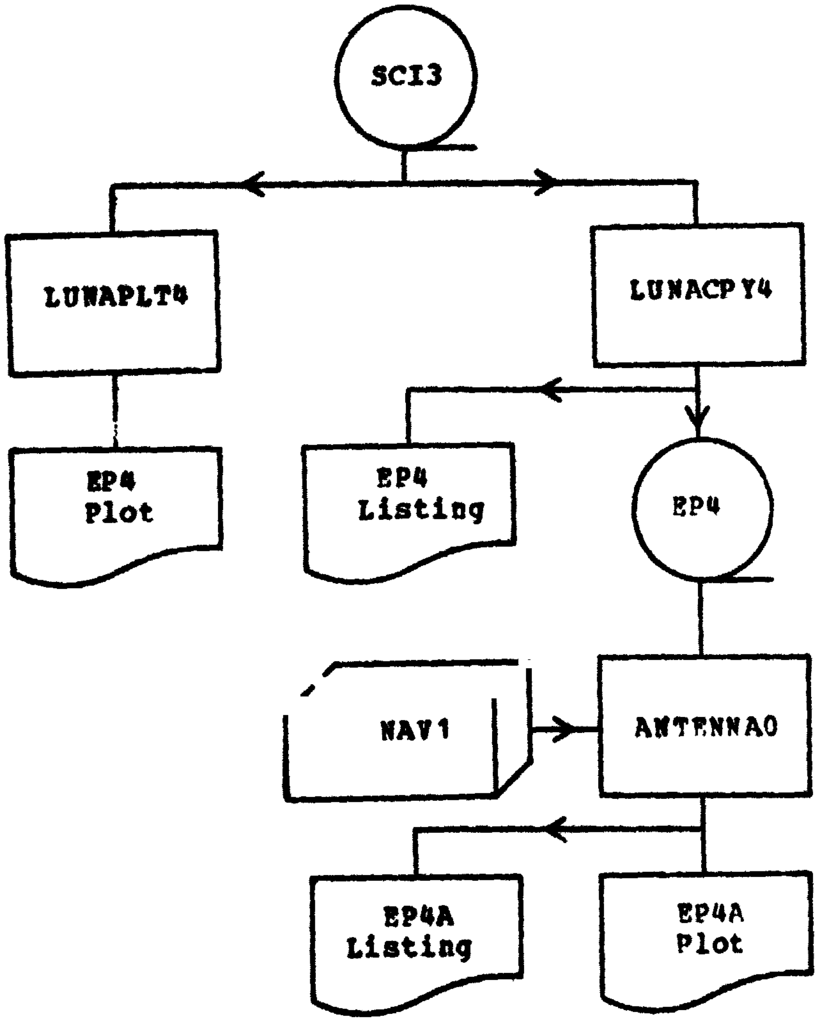


Figure 1(d) - Processing flow.

C-2

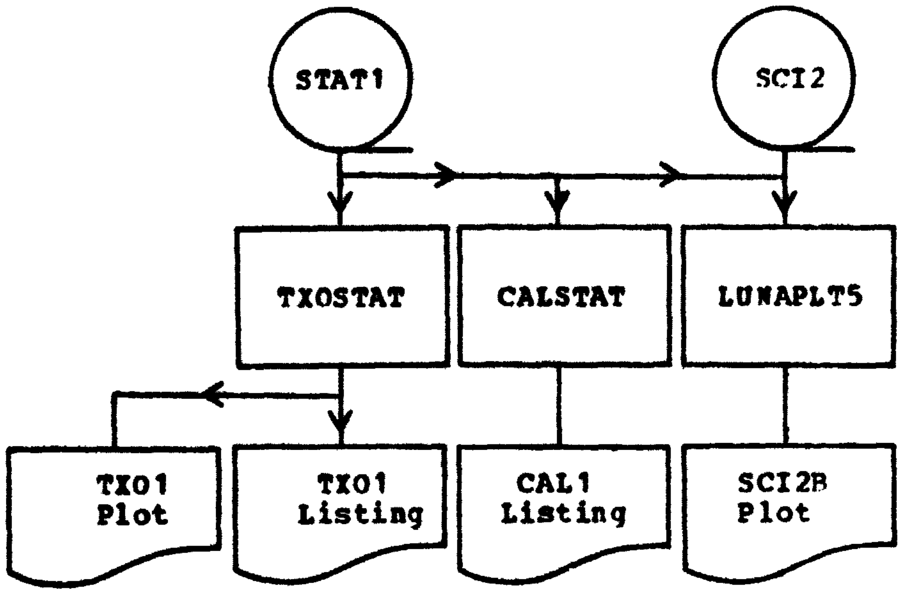


Figure 1(e) - Processing flow.

The Stack

A number of routines use a large array (named DATA, hereafter referred to by name, or as "the stack") and a set of indices to the array, as the basis of a system for manipulating range and dB data.

In most cases a particular set of range or dB information is contained in several blocks, which it is generally convenient to combine into one large block before processing. To accomplish this, three indices are associated with the stack as diagrammed in figure 2: IXX and IXY indicate the first words of range and dB data respectively; IORG gives the location of the first word into which data should be read to make an extension to the block currently being assembled. Other parameters relating to the stack may be defined where necessary.

The basic procedure to be used to process a complete file is outlined in figure 3. In this description "read a block" is taken to mean that the contents of one block from the input file are placed in locations IORG through IORG + N - 1 of the stack; the meaning of "last" is that given to it by the LUNIN routines.

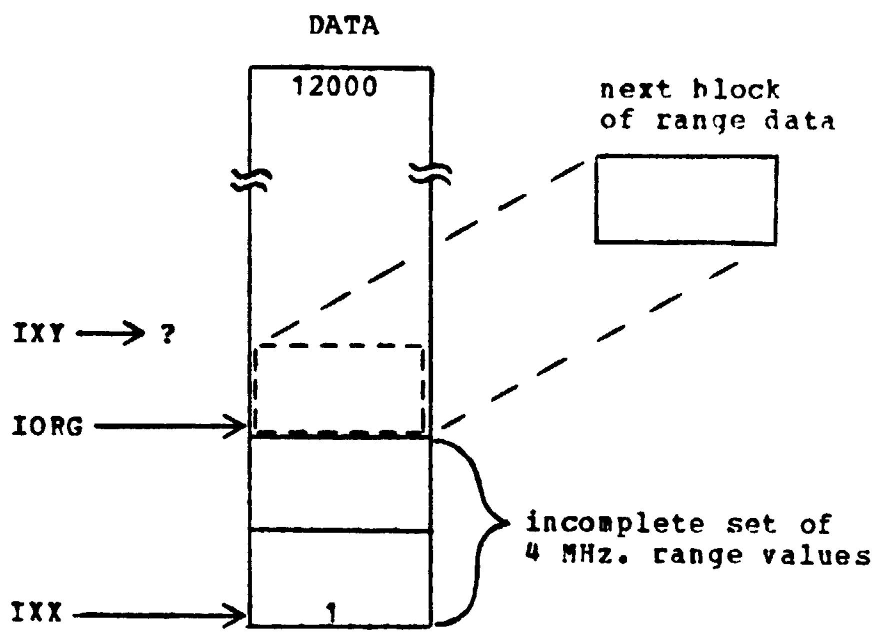


Figure 2(a) - Example of stack use: assembling a range array.

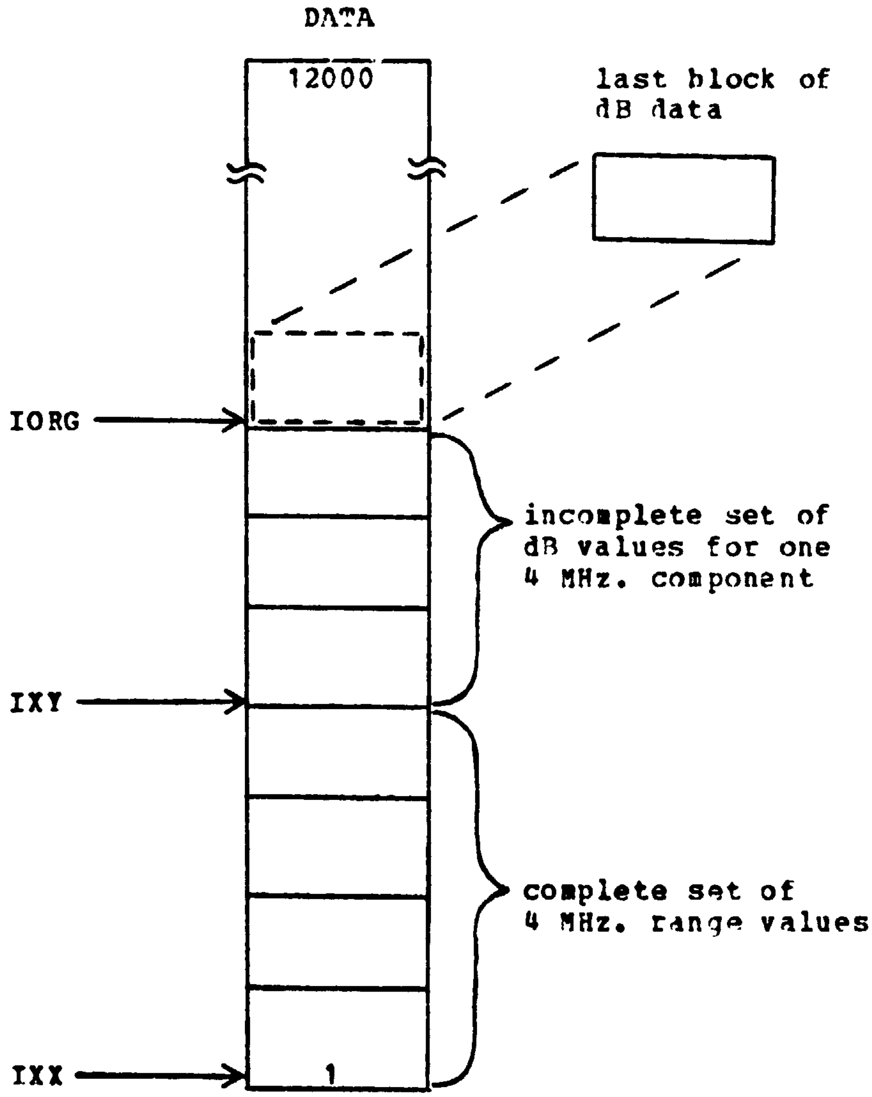


Figure 2(b) - Example of stack use: completing a dB array.

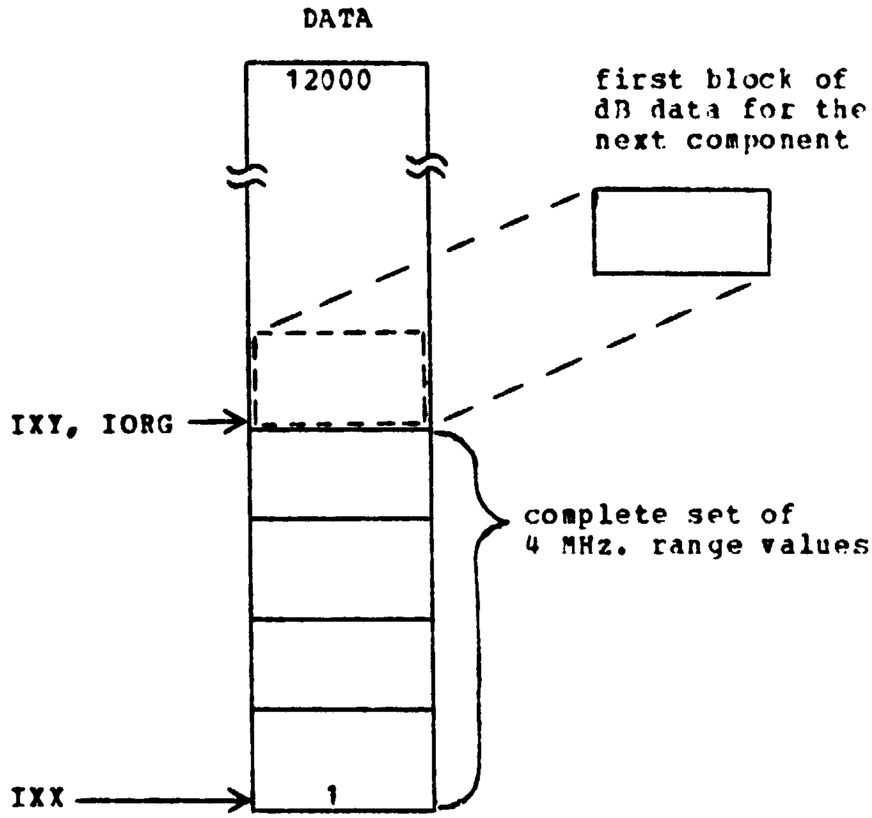


Figure 2(c) - Example of stack use: beginning the dB array for a new component.

```
do for frequency = 1, 2.1, 4, 8.1, 16, 32.1
    iorg = 1;  ixx = 1;  m = 0;

    ASSEMBLE THE RANGE ARRAY:
    repeat
        read a block;
        iorg = iorg + n;
        m = m + 1;
    until last;

    ixy = iorg;
    repeat
        l = 0;  iorg = ixy;

        ASSEMBLE THE DB ARRAY FOR ONE COMPONENT:
        repeat
            read a block;
            iorg = iorg + n;
            l = l + 1;
        until l = m;

        perform required processing;
    until last;
end;
```

Figure 3 - Algorithm for assembly of arrays of range and dB information.

Documentation Routines

LUNALIST, LUNALST2, LUNALST3

These routines produce complete listings of the data on files SCI1, SCI2, and SCI3 respectively.

The data are read from SCI1 and SCI2 by LUNIN and LUNIN2 respectively, and printed in blocks corresponding to the physical records on the files, fifteen values per line. Each block is preceded by a heading, containing the character information returned by the input routine, identifying the contents of the block.

The procedure for listing SCI3 is more complex, since the file contains no range data. LUNALST3 invokes LUNIN2 to read a record from SCI2, and inspects the returned value of ITYPE(2). If this value is five, indicating a block of range data, the block is listed. Otherwise, LUNIN3 is called to read a record from SCI3, which replaces the data from SCI2, and the new block is listed.

N.B. A bug, in all three routines, results in the identification of transmitter-off, calibration, and one megahertz range and dB data being printed incorrectly, as indicated in table 2. (The numbers in parentheses indicate how many records are affected.)

contents of block	label printed
temperature(1)	temperature
transmitter-off(1)	NONE
transmitter-off(5)	transmitter-off
calibration(1)	NONE
calibration(5)	calibration
1 MHz. range(1)	NONE
1 MHz. dB(6)	NONE

Table 2 - Incorrectly labelled blocks.

Editing Routines

LUNACOPY, LUNACPY2, LUNACPY3

These routines read files SCI1, SCI2, and SCI3 respectively, and produce the binary files SCI1A, SCI2A, and SCI3A, containing the label records from the input files, and six blocks of NPTS dB values each, for each frequency, after interpolation at intervals of 0.1 wavelength. NPTS varies with frequency, and is either the maximum number of interpolated values which could be generated, or 1000, whichever is smaller.

The stack mechanism is used to set up the arrays of range and dB values to be given to the interpolation routine, and the array RANGE is initialized with the appropriate equivalent in metres of 0, 0.1, 0.2, ... 99.9 wavelengths. Subroutine INTPOL is called to do the interpolation, and returns its results in array VCO.

After interpolation the values of NSTART and NPLOT (set by INTPOL) indicate the first and last elements of VCO which contain interpolated results. Before writing the array on the output file, the programs set VCO(1) through VCO(NSTART-1) and VCO(NPLOT+1) through VCO(1000) to zero.

LUNACPY4

This program produces a binary file (EP4) of range and dB values for the turn at EP-4, using file SCI3 as input. The program is a simple modification of LUNAPLT4, in which the call to subroutine GAPLOT is replaced by a write statement which generates a record on file EP4, and a second statement which writes the contents of the record on the printer.

LUNACPY5

This program is used to generate file STAT1, containing temperature, calibration, and transmitter-off data, and arrays of the ranges at which the data were obtained. Also, a set of crude LRV speed values, corresponding in time to the transmitter-off data, is computed and written on the output file.

The array of temperature values is simply copied from the input file (SCT2). The associated range data is the set of values for one megahertz, which is also copied directly onto the output file. (The one megahertz data were used since their occurrences are the closest in time to the temperature data - c.f. table 3.)

Transmitter-off data are read from the input file in two groups of three blocks each. For each group, the first, second, and third blocks contain data from the x, y, and z receiving antennae respectively. The frequency for each sample is dependant on the group, and on the tens digit of the corresponding element of the mode array: the first group contains data for frequencies of 32.1, 8.1, and 2.1 megahertz, and the second group for 16.0, 4.0, and 1.0 megahertz, corresponding respectively to tens digits of 1, 2, and 3. The contents of the blocks of calibration data are arranged similarly, with blocks one and four containing values for front-end noise, two and five containing values for the noise diode, and values for the noise diode plus 20 dB amplification in blocks three and six.

Using the number of the block on which it is working, and the appropriate contents of the mode array, tho program generates arrays of values which may be indexed by frequency, and either antenna in the case of the transmitter-off data, or noise source in the case of the calibration data.

Approximate range values corresponding to the above data are obtained by selecting the values closest to them in the timing sequence from the 32.1 megahertz range array, and again using the contents of the mode array. Corresponding to each range value, an approximation of the LRV speed is computed by taking the difference of the immediately succeeding and preceding elements of the 32.1 megahertz range array.

Record	Contents	Record	Contents
1	16 MHz. v.c.o.	17	16 MHz. v.c.o.
2	32 MHz. v.c.o.	18	32 MHz. v.c.o.
3	8 MHz. v.c.o.	19	8 MHz. v.c.o.
4	32 MHz. v.c.o.	20	32 MHz. v.c.o.
5	16 MHz. v.c.o.	21	16 MHz. v.c.o.
6	32 MHz. v.c.o.	22	32 MHz. v.c.o.
7	4 MHz. v.c.o.	23	4 MHz. v.c.o.
8	32 MHz. v.c.o.	24	32 MHz. v.c.o.
9	16 MHz. v.c.o.	25	16 MHz. v.c.o.
10	32 MHz. v.c.o.	26	32 MHz. v.c.o.
11	8 MHz. v.c.o.	27	8 MHz. v.c.o.
12	32 MHz. v.c.o.	28	2 MHz. v.c.o.
13	16 MHz. v.c.o.	29	16 MHz. v.c.o.
14	32 MHz. v.c.o.	30	32 MHz. v.c.o.
15	transmitter-off	31	1 MHz. v.c.o.
16	calibration	32	synchronization/reset (also contains temperature data)

Table 3 - Receiver timing sequence.
Each record is 202.5 ms. duration.

Plotting Routines

LUNAPLOT, LUNAPLT2

These routines produce (CALCOMP) plots of dB versus distance, using interpolated data. LUNAPLOT is used for plotting the data in SCI1A; LUNAPLT2 may be used to plot data from either SCI2A or SCI3A.

One namelist (FREQ) control record is read for each frequency. The parameters which may be specified allow choice of components, maximum and minimum wavelengths, and maximum dB value to be plotted. Filtering of the data before plotting may be requested, a dB level may be specified for plotting a reference mark, and optional plot annotation may be supplied.

Subroutine PLINIT is called to initialize the plotting software. The actual plots are produced using the DATPLT entry point of subroutine SEPLOT.

LUNAPLT3

This program is used to generate (CALCOMP) plots of the data on SCI2, similar to the plots produced by LUNAPLOT and LUNAPLT2. The program differs from the others in that the data are not interpolated, and cannot be filtered before plotting; also, the portion of the data corresponding to the turn at EP-4 is deleted before plotting.

The program uses the stack mechanism to prepare data for the plotting routine. Removal of the data for the turn is accomplished as follows. After an entire array of range values has been assembled, the program locates the values to be deleted; then succeeding values are copied downward to maintain a contiguous set, and IORG is reset to indicate what is then the first free location. The indices defining the gap are modified to apply to the dB segment of the stack, and each time a complete set of dB data is assembled, an equivalent compacting operation is performed.

Parameters to control the plotting operation are supplied on a set of namelist (CNTL) control records. The plotting is done using the DATPLT entry point of subroutine SEPLOT.

LUNAPLT4

This program produces (CALCOMP or GOULD) plots of dB values for the FP-4 turn versus an implicit time scale, using data from file SC13. The data are plotted as discrete points (marked by symbols) at equal horizontal intervals. Each component is plotted on a separate graph.

Each set of values to be plotted is assembled in the main program and passed to subroutine GAPLOT, which produces the plot. The method used to define the desired values is basically the complement of the method used in the two preceding programs to remove the same data. However, in this case the required segments of the arrays are merely located; no compacting operation is performed.

LUNAPLT5

This routine is a modification of LUNAPLT3 which allows the distances to be expressed in either metres or wavelengths. In addition, transmitter-off values from file STAT1 are plotted (as points, rather than continuous curves) as a baseline for each dB curve, using entry point BASEL of subroutine SEPLOT.

ANTENNAC

This program is used to generate plots of the patterns of the three receiving antennae, based on the data for the turn at EP-4 contained in file EP4. Two methods are available for computing the angle between the LRV and the SEP transmitter: the plotted points may be at equal angular increments throughout the whole range, or the navigation data may be used to compute an approximate angular displacement for each point.

One namelist (CNTL) control record is required for each frequency. Parameters in each record allow choice of components to be plotted, initial angle and the difference between final and initial angles, and indices defining the data points obtained while the LRV was in motion; a Boolean value (NAVDAT) may be included to indicate whether or not navigation data are to be used in computing angles, and if this value is "true", a time must be supplied corresponding to the first data point in the set for which the LRV was moving.

The main program organizes the control information, and then enters a loop, in each cycle of which it reads one record from file EP4, and if the data in that record are to be plotted, passes the data and required control information to subroutine ANTPAT.

Subroutine ANTPAT begins by drawing a set of x and y axes and plotting a label indicating frequency and component. The total angular range is divided into equal intervals, based on the number of points to be plotted. If navigation data are to be used in computing angular displacements, the number of odometer counts at the beginning and end of the range are obtained by invocations of function ODCINT, and their difference (ODCRAN) is computed. The first dB value and the initial angle are converted to rectangular coordinates and the point is plotted. A loop is then entered, which continues until the last data point has been plotted: the angle is decremented (to give clockwise rotation) either by the constant amount, or by using the result of an invocation of ODCINT to determine a fraction of the total angle; rectangular coordinates are computed, and the point is plotted; the index of the next value is determined, using the supplied parameters.

Function ODCINT is invoked with one argument, a time (T) in seconds. On the first entry a set of times and corresponding left- and right-wheel odometer counts are read from the card reader. For each triple the time and the average of the counts are saved. The value of the function is an odometer count obtained by linear interpolatior, using T and the arrays of times and corresponding average counts.

Statistical Routines

CALSTAT

This program is used to compute various statistics for each set of calibration data on file STAT1: the means and standard deviations of the front-end noise; differences between the experimental noise diode values (with and without amplification) and values for the same data obtained from earth-based testing; the means and standard deviations of these differences.

The computations for front-end noise data are straightforward, and should require no explanation. Most of the variable names begin with "FG". The results of the calculations are written on FORTRAN logical eight, which is used as an auxilliary printer.

The calculations for the noise-diode data (with and without amplification - "with" is indicated by "PA" as part of the name of each variable involved) involve computation of an earth-based value, using linear interpolation according to temperature, of the v.c.o. frequencies given in table 4. The difference between the experimental and interpolated values is computed; the remainder of the calculation consists of the accumulation and scaling of the appropriate sums.

transmitter frequency	noise diode		noise diode + amplifier	
	66°F	112°F	66°F	112°F
1.	525.6	508.6	1167.7	1157.7
2.1	564.2	546.2	1208.	1208.
4.	593.3	566.3	1230.	1230.
8.1	694.8	682.8	1298.6	1304.6
16.	756.1	770.1	1293.7	1306.7
32.1	833.9	888.9	1199.6	1198.6

Table 4 - Calibration data obtained from earth-based tests.

TXOSTAT

This routine and its associated subroutines compute mean values and standard deviations for each set of transmitter-off data on file STAT1. Separate statistics are calculated for periods when the rover was stopped and in motion; in the latter case, values for the EP-4 turn are excluded from the computations. Each set of values is displayed twice: once in order of increasing distance from the transmitter, and once in order of increasing LRV speed. The dB values are also plotted versus LRV speed (if plots are not required, subroutine TXPLOT is simply replaced by a dummy routine).

A set of bounds, the same as the 32.1 megahertz bounds input, but adjusted relative to the beginning of the 32.1 megahertz range data rather than the beginning of the turn data, is required as input. These values are used by function STOPT in deciding whether the rover was moving or stopped for each point within the EP-4 turn.

The statistical computations, which are performed in the main program, are relatively straightforward and should not require explanation. Data for ranges greater than 1667 metres are omitted from all calculations. A possibly confusing action is the assignment of -1 to certain elements of the speed array; the elements are those within the EP-4 turn for which the LRV was in motion. The speed values for these elements are computed (meaninglessly) as zero by LUNACPY5; the value of -1 indicates to the plotting routine that each such datum is to be ignored.

Ordering of the data according to increasing speed is accomplished by subroutine BUBBLE, which performs a bubble sort. Rather than interchanging elements within four parallel arrays (one of speeds and three of dB values) the routine uses an integer array (IX) of equivalent size, supplied by the calling program; IX(I) is initialized to I, and the contents of IX are used as indirect addresses to the actual data arrays, and it is these indices which are interchanged. When the sort is complete, the contents of IX indicate the order in which the other arrays should be indexed to obtain the data in order of increasing speeds.

The dB data are plotted versus speed by subroutine TXPLOT, which is entered once for each frequency. For each entry, three sets of labelled axes are plotted (one for each receiving antenna) within an 8.5 by 11 inch area. The appropriate data points are then simply plotted on each set of axes.

VLBIRT

This program is used to compare results from the VLBI experiment with SEP navigation data; the comparison is done on the basis of distance from the SEP transmitter.

The 16 megahertz range data are read from file SCI2, and an array of corresponding times is generated, using a starting value which is read as a control parameter. A parameter may also be supplied specifying a value to be added to each range value.

The VLBI times are converted from hours, minutes, and seconds to seconds, and each pair of x and y coordinates is converted to a distance. An interpolated SEP range value is computed for each VLBI datum, using the time arrays; the difference between VLBI and interpolated SEP ranges is calculated, and summations are taken of the differences and their squares, which yield the mean difference and standard deviation.

If PLOT is set to true on the namelist control record, subroutine RTPLOT is called. The subroutine plots a set of time and range axes, using the supplied parameter SCALE. The SEP and VLBI ranges are then plotted versus time in two simple loops.

Auxillary Routines

LUNIN, LUNIN2, LUNIN3

These subroutines are used to read data from files SCI1, SCI2, and SCI3 respectively. Floating-point data are returned to the invoking routine in the array DATA (not to be confused with the stack, although most of the programs which invoke these routines use a portion of the stack for passing data); fixed-point data are returned in array IDATA. The label record in SCI1 contains various fields, the contents of which are passed to the invoking routine through COMMON block LUNDAT. The label records of the other files consist solely of text, which is returned in IDATA.

The values in the fixed-point array TIDX are used to identify each record read from the input file. The values are arranged in seventeen groups of three: the first value in each group indicates the number of records of that type which are on the file; the second and third values are used to select alphanumeric identification from arrays TYPE1 and TYPE2 respectively. The alphanumeric identification is returned in TYPE, and the second and third values from TIDX are returned in ITYPE (both TYPE and ITYPE are in LUNDAT).

LUNIN obtains the value of N (the number of values in each record other than the label record) from the label record; the other two routines expect N to have been set by the invoking routine.

The logical variables FIRST and LAST are set to true if the record read is the first or last of its type respectively (e. g. - the first of the twenty-four eight megahertz dB records).

SFPLOT (DATPLT, BASEL)

This routine (written by J. J. Proctor, 1973) has been modified in a number of ways:

(1) dB values may be plotted versus either wavelengths or metres; the decision is made by examining XSCALE: if it is less than ten it is assumed to be the number of inches per twenty-wavelength segment, while a value of ten or greater is assumed to indicate the number of inches per kilometre.

(2) Entry point BASEL has been added in order to allow a set of points to be plotted in conjunction with each curve, using the same (relative) plotter origin. This is for the purpose of indicating a set of background values for each curve.

(3) Low dB values are no longer set to zero. Furthermore, the y-origin for each curve is now equivalent to (relative) zero dB, rather than the integral minimum dB value. This was necessary in order to keep the plot of background values on the page.

(4) All the labels for individual curves, except the component identification, have been eliminated.

(5) A reference mark for each curve is plotted on the y-axis, at a dB level set by the invoking routine.

N.B. The entry point (THEPLT) and associated code for plotting theoretical curves have not been changed. However, since the program was modified, no attempt has been made to verify the integrity of this feature.

Most of the code for the subroutine is concerned with setting up the axes and labels, and with placing a particular curve on the graph. The range axis markings depend on the value of XSCALE, as described above, and extend from zero to the first multiple of the chosen increment greater than the highest range value in the data. A displacement is computed such that curves on the graph will be equally spaced vertically. All these operations are performed on entry at DATPLT, the entry point for plotting data curves, if a new graph (not just a new curve) is to be plotted.

The curves themselves are plotted in a straightforward manner, and a label is plotted at the right-hand end of each, to provide component identification. After a component has been plotted, the parameters defining the position of the curve are left unchanged until the next entry at DATPLT; therefore an entry at BASEL will result in the baseline points being plotted on the same set of relative axes as the associated data curve. (If BASEL is invoked at any other time, it will not function properly.)

INTPOL

This is a general linear interpolation routine. It accepts "input" arrays XIN and YIN, and an array XOUT, of values on the same scale as XIN, for which interpolated values for YIN are required. The results are placed in array YOUT, and parameters NSTART and NPLOT are set to indicate which values in YOUT are the result of successful interpolation, and which are undefined due to the corresponding elements of XOUT being out of the range of XIN.

FILTER

This is a subroutine which accepts array A, and applies to its contents the filter whose coefficients are contained in array F. Array B is used for accumulation of sums, and its contents are copied into array A before return to the calling program.

PLINIT

This subroutine is used to systematize the initialization of the University of Toronto CALCOMP software package. Presumably it will be of interest only to users of that installation.

File Formats

SCI1, SCI2, SCI3

Each of these files begins with a label record. The format of this record on file SCI1 is given in table 5. The label records for the other two files are 2316 characters, consisting of 27 segments of 84 characters each, followed by 48 characters of padding. Each 84-character segment contains one card image and four padding characters.

The next record in each file is the mode array, in format 386I6. Each element is a three (decimal) digit number, MAR; the significance of the values of the digits is indicated in table 6. (The notations "f1" and "f2" in the table refer to transmitter-off and calibration data descriptions given in table 7.)

The format of all remaining records is 386F6.1; the contents of these records are given in table 7. (N.B. - file SCI3 contains no range data.)

Format	Contents
A6	run identification
A6	site identification
A6	traverse direction
A6	forward/reverse traverse
14A6	title
I6	number of values in each succeeding block

Table 5 - Format of SCI1 label record.

Digit	Value	Significance
M	1	receiver in data acquisition mode
	2	receiver in synchronization acquisition mode
A	1	f1 = 32.1 MHz.; f2 = 16 MHz.
	2	f1 = 8.1 MHz.; f2 = 4 MHz.
	3	f1 = 2.1 MHz.; f2 = 1 MHz.
R	0	synchronization not received
	1	synchronization received

Table 6 - Interpretation of mode data.

Number	Rec.	Tx.	Freq.	Contents
1				label
1				mode
1				temperature
1	x		f1	transmitter-off
1	y		f1	transmitter-off
1	z		f1	transmitter-off
1	x		f2	transmitter-off
1	y		f2	transmitter-off
1	z		f2	transmitter-off
1			f1	calibration - grounded input
1			f1	calibration - amplified noise
1			f1	calibration - noise
1			f2	calibration - grounded input
1			f2	calibration - amplified noise
1			f2	calibration - noise
1			1	range *
1	x	EW	1	dB
1	y	EW	1	dB
1	z	EW	1	dB
1	x	NS	1	dB
1	y	NS	1	dB
1	z	NS	1	dB
1			2.1	range *
1	x	EW	2.1	dB
1	y	EW	2.1	dB
1	z	EW	2.1	dB
1	x	NS	2.1	dB
1	y	NS	2.1	dB
1	z	NS	2.1	dB

Table 7 - Record contents on files SCI1, SCI2, and SCI3.
* not present on SCI3

Number	Rec.	Tx.	Freq.	Contents
2			4	range *
2	x	EW	4	dB
2	y	EW	4	dB
2	z	EW	4	dB
2	x	NS	4	dB
2	y	NS	4	dB
2	z	NS	4	dB
4			8.1	range *
4	x	EW	8.1	dB
4	y	EW	8.1	dB
4	z	EW	8.1	dB
4	x	NS	8.1	dB
4	y	NS	8.1	dB
4	z	NS	8.1	dB
8			16	range *
8	x	EW	16	dB
8	y	EW	16	dB
8	z	EW	16	dB
8	x	NS	16	dB
8	y	NS	16	dB
8	z	NS	16	dB
13			32.1	range *
13	x	EW	32.1	dB
13	y	EW	32.1	dB
13	z	EW	32.1	dB
13	x	NS	32.1	dB
13	y	NS	32.1	dB
13	z	NS	32.1	dB

Table 7 - Record contents on files SCI1, SCI2, and SCI3 (continued).
* not present on SCI3

SCI1A, SCI2A, SCI3A

These files are written in binary form (i.e. - without format control), and contain dB data equivalent to that in files SCI1, SCI2, and SCI3 respectively. Each file begins with the same label information as its parent file.

The remainder of each file consists of thirty-six blocks of dB data; each block is of the form given in table 8. The dB data are interpolated values, at intervals of 0.1 wavelength, beginning at zero wavelengths; the maximum value of NPTS is one thousand.

Position	Contents
1	eight characters: frequency
2	floating-point: frequency
3	integer: NSTART
4	integer: NPTS
5 through NSTART+3	floating-point: 0.0
NSTART+4 through NPTS+4	floating-point: dB values

Table 8 - dB data on files SCI1A, SCI2A, and SCI3A.

STAT1

The six blocks which comprise this binary file are summarized in table 9(a). The index I for the first two blocks selects temperatures and corresponding ranges at successive intervals of 6.48 seconds.

For the remaining blocks, the index I selects the various data at successive times (at varying intervals). Values of one through six for J select data for frequencies 1.0 through 32.1 megahertz respectively. The significance of K is indicated in table 9(b).

Block	Contents	Indexing
1	TEMP(I)	I<=386
2	RANGE(I)	I<=386
3	CAL(I,J,K), NCAL(J,K)	I<=NCAL(J,K) J<=6 K<=3
4	TXOFF(I,J,K), NTXOFF(J,K)	I<=NTXOFF(J,K) J<=6 K<=3
5	RANGE2(I,J), NR(J')	I<=NR(J') J<=6 J'=1+(J-1)/2
6	SPEED(I,J)	I<=NR(J') J<=6

Table 9(a) - Contents of file STAT1.

K	TXOFF	CAL
1	x antenna	grounded input
2	y antenna	noise diode +20dB amplification
3	z antenna	noise diode

Table 9(b) - Function of the index K for transmitter-off and calibration arrays on file STAT1.

EP4

This file contains range and dB data from file SCI3, only for the region of the turn at EP-4 (specifically for the range values on the interval 490 to 535 metres, inclusive. There are 36 records on the file, all of the same form, but of varying length. The form of a record is summarized in table 10.

Name	Words	Contents
F	1	frequency in megahertz
NCOMP	1	component identification: 1: rho endfire 2: phi endfire 3: zed endfire 4: rho broadside 5: phi broadside 6: zed broadside
YMIN	1	minimum and
YMAX	1	maximum dB values
N	1	number of range-dB pairs
RANGE	N	range data
DB	N	dB data

Table 10 - Record format for file EP4.

NAV1

This is actually a part of the card input data to ANTENNAC. Any number of cards may be included. Each card contains three values: a time in seconds, and corresponding right-front- and left-rear-wheel odometer counts, in format (3F10.).

Program Listings

```
**********************************************************************
*                                                                    *
*                             ANTENNAO                               *
*                                                                    *
**********************************************************************

C....ANTENNAO.....PLOT DATA FROM EP-4 TURN
C
C      DB VALUES ARE PLOTTED ON A POLAR GRID; THE ANGULAR COORDINATES ARE
C      ESTIMATES OF THE ANGLE BETWEEN THE LRV AXIS AND THE LINE FROM THE
C      LRV TO THE SEP TRANSMITTER.
C
C      SEVEN INPUT RECORDS ARE REQUIRED: SIX AS DESCRIBED BELOW, AND ONE
C      NAMELIST (PLTID) RECORD, READ BY PLINIT.  IN ADDITION, ANY
C      NUMBER OF CARDS CONTAINING NAVIGATION DATA (TIME, RIGHT- AND
C      LEFT-WHEEL ODOMETER COUNTS IN FORMAT 3F10.0) MAY FOLLOW THE PLTID
C      RECORD.
C
C      NAMELIST (CNTL):
C
C            IFREQ  - BASE TWO LOG OF FREQUENCY; NO DEFAULT
C
C            ICOMP  - (6) CODES FOR COMPONENTS TO BE PLOTTED, PADDED WITH
C                     ZEROES; DEFAULT SIX ZEROES.  THE CODES ARE:
C
C                              E  PIPE  BROADSIDE
C
C                         RHO    212         211
C                         PHI    222         221
C                         ZED    232         231
C
C            AO     - ANGLE FOR THE FIRST POINT; DEFAULT 3.14159
C            ARANGE - ANGULAR DIFFERENCE BETWEEN THE FIRST AND LAST POINTS
C                     DEFAULT 6.28318
C
C            BOUNDS - THREE PAIRS OF INDICES DEFINING POINTS TO BE PLOTTED
C                     EACH PAIR DEFINES THE FIRST AND LAST OF A SEQUENCE
C                     OF POINTS TO BE PLOTTEDE NO DEFAULT
C
C            NAVDAT - (LOGICAL) ODOMETER  COUNTS TO BE READ (FOLLOWING THE
C                     PLTID RECORD) AND USED IN COMPUTATION OF ANGLES;
C                     DEFAULT FALSE
C
C            TIMEO  - (REQUIRED IF NAVDAT IS TRUE) - TIME ( ON SCALE OF
C                     NAVIGATION DATA) FOR THE FIRST POINT TO BE PLOTTED;
C                     NO DEFAULT
C
```

```
C
      REAL*4 X(600), Y(600), TIME(600)
      REAL*4 TZERO(6)
      REAL*4 DT(6) / 6.48, 6.48, 3.24, 1.62, .81, .40846 /
      REAL*8 PROGNM(2) / 'OOGP.ANT', 'ENNA' /
      INTEGER*2 BOUND(6, 6), BOUNDS(6)
      LOGICAL*1 DECIDE(6, 6) / 36 * .TRUE. /, NAVDAT / .FALSE. /
      INTEGER*2 COMP(6) / 212, 222, 232, 211, 221, 231 /
      INTEGER*2 ICOMP(6)
C
      NAMELIST / CNTL / IFREQ, ICOMP, A0, ARANGE, TIMEO, BOUNDS, NAVDAT
C
C     SET UP CONTROL INFORMATION
C
      A0 = 3.14159
      ARANGE = 6.28318
      DO 100 I=1,6
C
        DO 10 J=1,6
          ICOMP(J) = 0
          BOUNDS(J) = 0
   10   CONTINUE
C
C       GET FREQUENCY INDICATOR AND COMPONENTS TO PLOT
C
        READ(5,CNTL)
        TZERO(IFREQ + 1) = TIMEO
        DO 50 J = 1,6
C
C         IF A COMPONENT IS NOT TO BE PLOTTED,
C         RESET ITS MATRIX ENTRY.
C
          IC = COMP(J)
          DO 30 K = 1,6
            IF(IC .EQ. ICOMP(K))
     .        GO TO 50
   30     CONTINUE
          DECIDE(IFREQ + 1, J) = .FALSE.
   50   CONTINUE
C
        DO 80 J = 1,6
          BOUND(J, IFREQ + 1) = BOUNDS(J)
   80   CONTINUE
  100 CONTINUE
C
C     INITIALIZE THE PLOTTER
C
      CALL PLINIT(PROGNM)
```

```
      CALL PLOT( 4.25, 5.0, -3)
C
C     LOOP THROUGH FREQUENCIES
C
      DO 140 IFREQ = 1,6
C
C       SET UP THE TIME ARRAY
C
        IB1 = BOUND(1, IFREQ)
        IB6 = BOUND(6, IFREQ)
        DO 110 I = IB1, IB6
          TIME(I) = TZERO(IFREQ) + DT(IFREQ) * (I - IB1)
  110   CONTINUE
C
C       LOOP THROUGH COMPONENTS
C
        DO 130 JCOMP = 1,6
          READ(1) FREQ, NCOMP, YMIN, YMAX, N,
     .            (X(I), I = 1,N), (Y(I), I = 1,N)
          IF(.NOT. DECIDE(IFREQ, JCOMP))
     .      GO TO 130
        CALL ANTPAT(TIME, Y, N, FREQ, JCOMP, BOUND(1, IFREQ),
     .                     A0, ARANGE, NAVDAT)
  130   CONTINUE
  140 CONTINUE
C
      CALL PLOTND
C
      RETURN
      END

************************************************************************
*                                                                      *
*                               ANTPAT                                 *
*                                                                      *
************************************************************************

      SUBROUTINE ANTPAT(T, H, N, F, JC, B, A0, ARANGE, NAVDAT)
C
C     ROUTINE TO PLOT THE ANTENNA PATTERN
C     THROUGH THE TURN AT EP-4
C
C     PARAMETERS ARE:
C
C        T  - TIME ARRAY
C
```

```
C          P  - V.C.O. ARRAY
C
C          N  - NUMBER OF POINTS IN P OR H
C
C          F  - FREQUENCY
C
C          JC - COMPONENT IDENTIFIER:
C
C                    ENDFIRE  BROADSIDE
C
C                RHO     1         4
C                PHI     2         5
C                ZED     3         6
C
C          B  - BOUNDS WITHIN H;  THE VALUES H(B(1)) - H(B(2)) INCLUSIVE
C             H(B(3)) - H(B(4)) INCLUSIVE
C             AND H(B(5)) - H(B(6)) INCLUSIVE ARE PLOTTED.
C
C          AO - INITIAL ANGLE - DEFAULTS TO PI
C
C          ARANGE - RANGE OF ANGLES - DEFAULTS TO 2*PI
C
C
      REAL*8 LAB(6) / 'RHO END.','PHI END.', 'ZED END.',
     .               'RHO BRD.','PHI BRD.', 'ZED BRD'         /
      REAL*4 H(N), T(N)
      INTEGER*4 COMP(6,2) / Z3B, Z24, Z69, Z3B, Z24, Z69,
     .                     3 * 'END',  3 * 'BRD'           /
      INTEGER*2 B(6)
      LOGICAL*1 NAVDAT
C
      WRITE(6,1000) F, LAB(JC), N, B
C
C     PLOT X AND Y AXES FOR REFERENCE
C
      CALL PLOT(-4.  ,  0. ,  3)
      CALL PLOT( 4.  ,  0. ,  2)
      CALL PLOT( 0.  ,  4. ,  3)
      CALL PLOT( 0.  , -4. ,  2)
C
C     PLOT LABELS: FREQUENCY AND COMPONENT
C
      CALL NUMBER(-1.12,  -5., .14, F,            0.,   1)
      CALL SYMBOL( 999., 999., .14, 7H MHZ. H,    0.,   7)
      CALL SYMBOL( 999., 999., .14, COMP(JC,1), 0., -1)
      CALL SYMBOL( 999., 999., .14, COMP(JC,2), 0.,   3)
C
C     INITIALIZE THE VALUE OF THE ANGLE
```

```
C         AND PLOT THE FIRST POINT
C
          DA = ARANGE / ((B(2) - B (1)) + (B(4) - B(3)) + (B(6) - B(5)) + 2)
   20     A=A0
          X = H(B(1))  * COS(A)  * 0.1
          Y = H(B(1))  * SIN(A)  * 0.1
          WRITE(6,2000) B(1), H(B(1)), A, X, Y
          CALL SYMBOL(X, Y, .07, 10, 0., -1)
          I = B(1) + 1
          IF(I .EQ. B(2) + 1) I = B(3)
          IF(.NOT. NAVDAT) GO TO 30
          ODCMIN = ODCINT(T(B(1)))
          ODCRAN = ODCINT(T(B(6))) - ODCMIN
C
C         DECREMENT THE ANGLE (ROTATION IS CLOCKWISE)
C         AND PLOT THE NEXT POINT
C
   30     IF(NAVDAT) GO TO 40
          A = A - DA
          GO TO 45
   40     A = A0 - ARANGE * (ODCINT(T(I)) - ODCMIN) / ODCRAN
   45     IF(A .LT. -3.14159) A = A + 6.28318
          X = H(I)      * COS(A)  * 0.1
          Y = H(I)      * SIN(A)  * 0.1
          WRITE(6,3000) I, H(I), A, X, Y
          CALL SYMBOL  (X, Y, .07, 10, 0., -2)
          I = I + 1
          IF(I .EQ. B(2) + 1) I = B(3)
          IF(I .EQ. B(4) + 1) I = B(5)
          IF(I .EQ. B(6) + 1) GO TO 90
C
          GO TO 30
C
C         REDEFINE THE PLOTTER ORIGIN FOR A POSSIBLE NEXT ENTRY;
C         THEN RETURN
C
   90     CALL PLOT(8.50,  0.0, -3)
          RETURN
 1000     FORMAT('0', F5.1, ' MHZ., ', A8, ' COMPONENT' / 1X, I5, ' POINTS'
     .                // 1X, 'POINTS', I5, ' TO ', I5 /
     .                   7X, I5, ' TO ', I5  /
     .                   4X, 'AND', I5, ' TO ', I5, ' WILL BE PLOTTED')
 2000     FORMAT('0    '     DB', 7X, 'ANGLE', 9X, 'X (PLOT) Y' /
     .                '0', I4, 2X, F6.2, F10.5, 5X, 2F9.3)
 3000     FORMAT(1X, I4, 2X, F6.2, F10.5, 5X,2F9.3)
          END
```

```
*****************************************************************************
*                                                                           *
*                                 BUBBLE                                    *
*                                                                           *
*****************************************************************************

      SUBROUTINE BUBBLE(X, IX, N)
C
      REAL*4 X(N)
      INTEGER*4 IX(N)
C
C
      DO 10 I = 1, N
        IX(I) = I
   10 CONTINUE
C
C
      NN = N - 1
C
C
      DO 50 I = 1, NN
        IF(X(IX(I)) .LE. X(IX(I + 1))) GO TO 50
C
C             SWITCH
C
        IT          = IX(I)
        IX(I)       = IX(I + 1)
        IX(I + 1)   = IT
C
C
        II = I - 1
        JJ = I
C
C             BUBBLE
C
        DO 40 J = 1, II
          JJ = JJ - 1
          IF(X(IX(JJ)) .LE. X(IX(JJ + 1))) GO TO 50
C
C             SWITCH
C
          IT          = IX(JJ)
          IX(JJ)      = IX(JJ + 1)
          IX(JJ + 1)  = IT
   40   CONTINUE
   50 CONTINUE
C
```

```
C
      RETURN
C
C
      END

************************************************************************
*                                                                      *
*                              CALSTAT                                 *
*                                                                      *
************************************************************************

C
C            PROGRAM TO COMPARE CALIBRATION DATA OBTAINED ON THE MOON
C            WITH VALUES FROM TEST RUNS ON EARTH.
C
      REAL*4 TRANGE(386), TEMP(386), CRANGE(140,6), CAL(140,6,3)
      REAL*4 AN(6) / -.3696, -.3913, -.5870, -.2609, .3043, 1.1957 /
      REAL*4 BN(6) / 550., 590., 632., 712., 736., 755.  /
      REAL*4 ANPAMP(6)/-.2174, 0., 0., .1304, .2826, -.0217 /
      REAL*4 BNPAMP(6) / 1182., 1208., 1230., 1290., 1275., 1301.  /
      REAL*4 EDN(140), EDNPA(140), DN(140), DNPAMP(140), T(140)
      INTEGER*4 NCAL(6,3), NR(3)
C
C            READ THE REQUIRED DATA  (THE FOURTH READ STATEMENT
C            SKIPS OVER THE TRANSMITTER-OFF ARRAYS.)
C
      READ(3)  TEMP
      READ(3)  TRANGE
      READ(3)  CAL, NCAL
      READ(3)
      READ(3)  CRANGE, NR
C
C
C     LOOP THROUGH FREQUENCIES
C
      DO 100 IFREQ = 1, 6
        IFR = 2 ** (IFREQ - 1)
        WRITE (6,1000) IFR
        WRITE(8, 6000) IFR
        EMN = 0.
        ESIG = 0.
        EPAMN = 0.
        EPASIG = 0.
        EGMN = 0.
        EGSIG = 0.
```

```
      N = 0
C
C           LOOP THROUGH THE RANGE ARRAY FOR THIS FREQUENCY.
C
      DO 60 I = 1, 140
C
C           FIND TEMPERATURE VALUE FOR CRANGE(I)
C
        IF(CRANGE(I, IFREQ) .GT. 1667.) GO TO 70
        IF(CRANGE(I, IFREQ) .GT. TRANGE(1)) GO TO 20
        T(I) = TEMP(1)
        GO TO 50
   20   CONTINUE
        DO 40 J = 1, 385
          IF(CRANGE(I, IFREQ) .GT. TRANGE(J + 1)) GO TO 40
          T(I) = TEMP(J) +    (TEMP(J + 1) - TEMP(J))
     .                 * (CRANGE(I, IFREQ) - TRANGE(J))
     .                 / (TRANGE(J + 1) - TRANGE(J))
          GO TO 50
   40   CONTINUE
   50   EBN(I) = AN(IFREQ) * (T(I) - 66.) + BN(IFREQ)
        EBNPA(I) = ANPAMP(IFREQ) * (T(I) - 66.) + BNPAMP(IFREQ)
        DN(I) = CAL(I, IFREQ, 3) - EBN(I)
        DNPAMP(I) = CAL(I, IFREQ, 2) - EBNPA(I)
        N = N + 1
        EMN = EMN + DN(I)
        EPAMN = EPAMN + DNPAMP(I)
        EGMN = EGMN + CAL(I, IFREQ, 1)
   60 CONTINUE
   70 EMN = EMN / N
      EPAMN = EPAMN / N
      EGMN = EGMN / N
      DO 90 I = 1, N
        DDN = ABS(DN(I) - EMN)
        DDNPA = ABS(DNPAMP(I) - EPAMN)
        ESIG = ESIG + DDN * DDN
        EPASIG = EPASIG + DDNPA * DDNPA
        WRITE(6, 2000) CRANGE(I, IFREQ), T(I), CAL(I, IFREQ, 3),
     .                EBN(I), DN(I), DDN, CAL(I, IFREQ, 2),
     .                EBNPA(I), DNPAMP(I), DDNPA
        CIG = CAL(I, IFREQ, 1) - EGMN
        EGSIG = EGSIG + CIG * CIG
        WRITE(8, 4000) CRANGE(I, IFREQ), CAL(I, IFREQ, 1), CIG
   90 CONTINUE
      ESIG = SQRT(ESIG / (N - 1))
      EPASIG = SQRT(EPASIG / (N - 1))
      WRITE(6, 3000) EMN, EPAMN, ESIG, EPASIG
      EGSIG = SQRT(EGSIG / (N - 1))
```

```
      WRITE(8, 5000) EGMN, EGSIG
  100 CONTINUE
      RETURN
C
 1000 FORMAT('1', I4, ' MHZ.            CALIBRATION COMPARISON'   /
     .       43X, 'NOISE DIODE', 28X, 'NOISE DIODE + 20 DB AMP.'
     .      / 9X, 'RANGE', 4X, 'TEMPERATURE',
     .        2(10X, 'LUNAR', 5X, 'EARTH', 5X, 'ERROR',
     .        2X, 'D(ERROR)' )  / 2X )
 2000 FORMAT(1X, 2(F10.3, 5X), 2(4F10.3, 5X))
 3000 FORMAT('-', 25X, 2(22X, 'MEAN ERROR = ', F10.3) /
     .              26X, 2(14X, 'STANDARD DEVIATION = ', F10.3))
 4000 FORMAT(16X, F10.3, 5X, 2F10.3)
 5000 FORMAT('-', 33X, 'MEAN = ', F10.3 /
     .              20X, 'STANDARD DEVIATION = ', F10.3)
 6000 FORMAT('1', I4, ' MHZ.                    BACKGROUND' /
     .       '0', 20X, 'RANGE', 10X, 'NOISE  D(NOISE)' / 2X )
C
      END


*****************************************************************************
*                                                                           *
*                                  FILTER                                   *
*                                                                           *
*****************************************************************************

      SUBROUTINE FILTER(A,N,F,M,B)
C..N-POINT FILTER FOR ARRAY *A* OF DIMENSION *N*, USING *M* FILTER COEFF
C..IN ARRAY *F*. ARRAY *B* OF DIMENSION *N* IS USED TO STORE FILTERED EF
C..TEMPORARILY.
C
      DIMENSION A(N), B(N), F(M)
C
C..AVOID ATTEMPTING TO FILTER DATA AT START AND END OF ARRAY.
      K=M/2+1
      L=N-K+1
C
      IF(N.LT.M) GO TO 5
C
C..MAIN LOOP FOR ALL DATA POINTS.
      DO 2 I=K,L
      ISUB=I-K
C
C..LOOP TO APPLY THE FILTER COEFFICENTS FOR ONE DATA POINT.
      B(I)=0.
      DO 1 J=1,M
```

```
    1  B(I)=B(I)+A(ISUB+J)*F(J)
C
    2  CONTINUE
C
C..COPY B BACK INTO A.
       DO 3 I=K,L
    3  A(I)=B(I)
C
       WRITE(6,4) M,N,F
    4  FORMAT('0',I3,'-POINT FILTERING COMPLETED ON',I4,' POINTS. FILTER
     *COEFFICENTS WERE:'/(7X,14F9.4/))
C
       RETURN
C
    5  WRITE(6,6) M,N
    6  FORMAT('0***ERROR***  ATTEMPT TO USE',I4,'-POINT FILTER ON',I4,
     *' POINTS: FILTER REQUEST IGNORED.'/)
       RETURN
       END

************************************************************************
*                                                                      *
*                               GAPLOT                                 *
*                                                                      *
************************************************************************

       SUBROUTINE GAPLOT(FREQ, X, Y, N, YMIN, YMAX, NCOMP)
C
C      PLOT RANGE VS. RECORD NUMBER AND V.C.O. VS. RECORD NUMBER FOR
C      DATA IN THE AREA OF THE TURN AT EP-4
C
C      PARAMETERS ARE:
C
C         FREQ  - FREQUENCY
C
C         X     - RANGE ARRAY
C
C         Y     - V.C.O. ARRAY
C
C         N     - NUMBER OF VALUES IN X OR Y
C
C         YMIN  - MINIMUM V.C.O. VALUE
C
C         YMAX  - MAXIMUM V.C.O. VALUE
C
C         NCOMP - COMPONENT IDENTIFIER:
```

```
C
C                      ENDFIRE  BROADSIDE
C
C              RHO     1          4
C              PHI     2          5
C              ZED     3          6
C
C
      REAL*4 X(N),Y(N)
      INTEGER*4 ICOMP(6,2) / Z3B, Z24, Z69, Z3B, Z24, Z69,
     *                                3 * 'END',     3 * 'BRD'       /
      WRITE (6,100) FREQ, NCOMP, N, YMIN, YMAX
      WRITE(6,200) X
      WRITE(6,300) Y
C
C     N POINTS ARE TO BE PLOTTED OVER A RANGE IN (PLOTTER) Y
C     OF 8.589 INCHES.
C
      DX = 8.589 / N
      CALL PLOT(0., 8.,3)
      CALL PLOT(0., 0.,2)
C
C     SET AN INTEGRAL MINIMUM V.C.O. VALUE;  IF THE RANGE OF V.C.O.
C     VALUES IS GREATER THAN 35 DB. ADJUST THE DATA TO FIT WITHIN
C     A 7 INCH PLOTTER RANGE - OTHERWISE THE SCALE IS FIXED
C     AT 5 DB. / INCH.
C
      YMINN=AINT(YMIN)
      DY= AMAX1(5., AINT((AINT(YMAX -YMINN) + 1.) / 7.) + 1.)
      WRITE (6, 150) DY
      DY = 1. / DY
      DO 20 I = 1, 7
      YY = I
      YYY = YMINN + YY / DY
      CALL SYMBOL(0.,YY,.07,13,90.,-1)
      CALL NUMBER(-.1,YY-.05,.07,YYY,90.,-1)
   20 CONTINUE
C
C     FINISH LABELLING THE AXIS;  THEN PLOT A LABEL
C     GIVING FREQUENCY AND COMPONENT
C
      CALL SYMBOL(-.25, 6.25, .14, 5HV C O, 90., 5)
      CALL NUMBER(4.5,0.,.14,FREQ,0.,1)
      CALL SYMBOL(999.,999.,.14,7H MHZ. H,0.,7)
      CALL SYMBOL(999.,999.,.14,ICOMP(NCOMP,1),0.,-1)
      CALL SYMBOL(999.,999.,.14,ICOMP(NCOMP,2),0., 3)
C
C     PLOT A SYMBOL FOR EACH V.C.O. VALUE.
```

```
C
      DO 40 I=1,N
      XX=DX*I
      YY = DY * (Y(I) - YMINN)
      CALL SYMBOL(XX,YY,.07,10,0.,-1)
   40 CONTINUE
C
C     REDEFINE THE ORIGIN FOR THE NEXT PLOT;
C     THEN RETURN.
C
      CALL PLOT(9.5, 0., -3)
      RETURN
  100 FORMAT('OFREQ.=',F6.1,'  COMPONENT',I2/I6,' POINTS'/
     .      ' MIN. V.C.O.=', F6.1 / ' MAX. V.C.O.=', F6.1)
  150 FORMAT('OSCALE =', F6.1, ' DB / INCH')
  200 FORMAT('ORANGE  ARRAY:'/100(1X,10F10.1/))
  300 FORMAT('OV.C.O. ARRAY:'/100(1X,10F10.1/))
      END


*************************************************************************
*                                                                       *
*                              INTPOL                                   *
*                                                                       *
*************************************************************************

      SUBROUTINE INTPOL (XIN,YIN,N,XOUT,YOUT,NSTART,NPLOT)
C
C      LINEAR INTERPOLATION OF YIN VS XIN AT POINTS XOUT.  FEB 18/73.
C
C     INPUT:
C     XIN = INPUT X ARRAY
C     YIN = INPUT Y ARRAY
C     N = DIMENSION OF XIN AND YIN
C     XOUT = POINTS AT WHICH YIN WILL BE INTERPOLATED
C     NPLOT = DIMENSION OF XOUT AND YOUT
C
C     OUTPUT:
C     YOUT = INTERPOLATED VALUES OF YIN AT POINTS XOUT
C     NSTART = NUMBER OF FIRST POINT INTERPOLATED
C     NPLOT = NUMBER OF LAST POINT INTERPOLATED
C
      DIMENSION XOUT(NPLOT),YOUT(NPLOT),XIN(N),YIN(N)
      NSTART=1
      I=1
C                                                                     ***
C     DO LOOP TO INTERPOLATE YOUT AT EACH XOUT POINT.
```

```
C     CHECKS ARE MADE FOR BEGINNING AND END OF YIN.
C
      DO 50 J=1,NPLOT
   40 IF(XIN(I)-XOUT(J)) 10,20,30
C
   10 IF(I.EQ.N) GO TO 60
      I=I+1
      GO TO 40
C
   20 YOUT(J)=YIN(I)
      GO TO 50
C
   30 IF(I.EQ.1) GO TO 33
      YOUT(J)=YIN(I-1)+(XOUT(J)-XIN(I-1))*(YIN(I)-YIN(I-1))/(XIN(I)-
     .     XIN(I-1))
      GO TO 50
C
   33 NSTART=J+1
   50 CONTINUE
C                                                                    ***
      RETURN
   60 NPLOT=J-1
      RETURN
      END


************************************************************************
*                                                                      *
*                              LUNACOPY                                *
*                                                                      *
************************************************************************

      REAL*8    TYPE(2), RUN, SITE, DIRECT, FORREV, TITLE(11)
      REAL*4    DATA(12000), RANGE(1000), VCO(1000)
      REAL*4    FREQ(6) /1.0, 2.1, 4.0, 8.1, 16.0, 32.1 /
      INTEGER*4 IDATA(400)
      INTEGER*2 ITYPE(2)
      LOGICAL*4 FIRST, LAST
      EQUIVALENCE (DATA(1), IDATA(1))
      COMMON /LUNDAT/ TITLE, RUN, SITE, DIRECT, FORREV, TYPE,
     .                ITYPE, N, FIRST, LAST
C
C
C     READ LUNAR SEP FILE (#1) AND PRODUCE A FILE OF V.C.O. DATA
C     INTERPOLATED AT INTERVALS OF 0.1 WAVELENGTH
C
C     THE RANGE AND V.C.O. DATA ARE ACCUMULATED IN ARRAY "DATA".
```

```
C     IORG IS THE INDEX OF THE NEXT FREE LOCATION INTO WHICH DATA
C     MAY BE STORED.  IXX IS THE INDEX OF THE FIRST RANGE VALUE,
C     AND IXY IS THE INDEX OF THE FIRST V.C.O. VALUE.
C
C
C     READ AND WRITE THE LABEL RECORD.
C     THIS RECORD CONTAINS N - THE NUMBER OF
C     VALUES IN EACH SUBSEQUENT RECORD.
C
      CALL LUNIN(DATA, IDATA, &980, &990)
      WRITE(6, 3000) TYPE
      WRITE(6, 1000) RUN, SITE, DIRECT, FORREV, TITLE, N
      WRITE(3)       RUN, SITE, DIRECT, FORREV, TITLE, N
C
C     INITIALIZE THE STACK
C
   10 IORG=1
      M=0
      L=0
C
C     CHECK FOR STACK OVERFLOW BEFORE READING THE NEXT RECORD.
C
   20 IF(IORG+N .GT. 12000) GO TO 970
      CALL LUNIN(DATA(IORG), IDATA, &980, &990)
      IF(ITYPE(1) .GE. 6) GO TO 40
      WRITE(6, 2000) TYPE
      GO TO 20
C
C
C     THIS SECTION IS ENTERED ONLY FOR
C     RANGE AND V.C.O. RECORDS.
C
C
   40 WRITE(6, 3000) TYPE
      IF(ITYPE(2) .EQ. 6) GO TO 60
C
C     FOR RANGE DATA - MOVE IORG TO POINT ONE LOCATION BEYOND THE
C     LAST VALUE, AND INCREMENT THE COUNT OF RANGE BLOCKS (M).
C
      IF(FIRST) IXX=IORG
      IORG=IORG+N
      M=M+1
      IF(.NOT. LAST) GO TO 20
C
C     AFTER READING THE LAST RANGE BLOCK FOR THIS FREQUENCY,
C     FILL ARRAY "RANGE" WITH DISTANCES IN METERS CORRESPONDING
C     TO 0.1 WAVELENGTH INCREMENTS;  THEN COMPUTE THE NUMBER OF VALUES.
C
```

```
      DWL=29.97925/FREQ(ITYPE(1)-5)
      DO 50 I=1, 1000
      RANGE(I)=DWL*FLOAT(I-1)
   50 CONTINUE
      NPTSIN=M*N
      GO TO 20
C
C
C     TREATMENT OF V.C.O. DATA IS SIMILAR;  ONE SET OF V.C.O. VALUES
C     HAS BEEN ACCUMULATED WHEN THE COUNT OF V.C.O. BLOCKS (L) EQUALS M.
C
C
   60 IF(FIRST) IXY=IORG
      IORG=IORG+N
      L=L+1
      IF(L .LT. M) GO TO 20
C
C     CALL INTPOL TO OBTAIN V.C.O. VALUES AT EQUAL RANGE INTERVALS;
C     THE NEW VALUES ARE RETURNED IN ARRAY "VCO".
C
      NSTART=1
      NPLOT=1000
      CALL INTPOL(DATA(IXX), DATA(IXY),NPTSIN,RANGE, VCO, NSTART, NPLOT)
C
C     SET TO ZERO V.C.O. VALUES WHICH HAVE NOT BEEN INTERPOLATED.
C
      NSTM1=NSTART-1
      IF(NSTM1 .LE.0) GO TO 80
      DO 70 I=1, NSTM1
      VCO(I)=0.0
   70 CONTINUE
   80 NPLTP1=NPLOT+1
      IF(NPLTP1 .GT. 1000) GO TO 100
      DO 90 I=NPLTP1, 1000
      VCO(I)=0.0
   90 CONTINUE
  100 NPTS=NPLOT-NSTM1
C
C     WRITE HEADER INFORMATION AND ARRAY "VCO".
C
      WRITE(6,4000) TYPE(1),FREQ(ITYPE(1)-5),NSTART,NPTS,
     .              (VCO(I),I=1,NPTS)
      WRITE(3)      TYPE(1),FREQ(ITYPE(1)-5),NSTART,NPTS,
     .              (VCO(I),I=1,NPTS)
C
C
C     IF LAST IS TRUE THEN READ A NEW SET OF RANGE VALUES;  OTHERWISE
C     READ V.C.O. DATA FOR THE NEXT COMPONENT (THE CURRENT RANGE DATA
```

```
C     ARE RETAINED).
C
C
      IF(LAST) GO TO 10
      IORG=IXY
      L=0
      GO TO 20
C
C
C     STACK OVERFLOW MESSAGE
C
C
  970 WRITE(6,7000)
      GO TO 999
C
C
C     END OF FILE ON INPUT WHEN MORE DATA WERE EXPECTED
C
C
  980 WRITE(6, 5000)
      GO TO 999
C
C
C     PROCESSING COMPLETED NORMALLY
C
C
  990 WRITE(6, 6000) TYPE
  999 END FILE 3
      RETURN
C
C
 1000 FORMAT('ORUN ',A6/'OSITE ',A6/'ODIRECTION ',A6/
     .        '0',A6,' TRANSMITTER'/'0',10A8,A4/'0',I4,' POINTS')
 2000 FORMAT('0',2A8,' RECORD SKIPPED')
 3000 FORMAT('0',2A8,' RECORD READ')
 4000 FORMAT('1LABEL="',A8,'"'/ 'OFREQ.= ',F5.1,' MHZ.'/
     .        'OFIRST POINT=',I4/ '0# OF POINTS=',I4/
     .        '0',10F10.3/99(1X,10F10.3/))
 5000 FORMAT('ONORMAL END OF JOB')
 6000 FORMAT('OEND OF FILE OCCURRED WHILE ATTEMPTING TO READ ',
     .        2A8, ' RECORD')
 7000 FORMAT('-***  INSUFFICIENT SPACE ON STACK  ***')
C
C
      END
```

```
*************************************************************************
*                                                                       *
*                             LUNACPY2                                  *
*                                                                       *
*************************************************************************

      REAL*8    TYPE(2), RUN, SITE, DIRECT, FORREV, TITLE(11)
      REAL*4    DATA(12000), RANGE(1000), VCO(1000)
      REAL*4    FREQ(6) /1.0, 2.1, 4.0, 8.1, 16.0, 32.1 /
      INTEGER*4 IDATA(400)
      INTEGER*2 ITYPE(2)
      LOGICAL*4 FIRST, LAST
      EQUIVALENCE (DATA(1), IDATA(1))
      COMMON /LUNDAT/ TITLE, RUN, SITE, DIRECT, FORREV, TYPE,
     .                ITYPE, N, FIRST, LAST
C
C
C     READ LUNAR SEP FILE (#2) AND PRODUCE A FILE OF V.C.O. DATA
C     INTERPOLATED AT INTERVALS OF 0.1 WAVELENGTH
C
C     THE RANGE AND V.C.O. DATA ARE ACCUMULATED IN ARRAY "DATA".
C     IORG IS THE INDEX OF THE NEXT FREE LOCATION INTO WHICH DATA
C     MAY BE STORED.  IXX IS THE INDEX OF THE FIRST RANGE VALUE,
C     AND IXY IS THE INDEX OF THE FIRST V.C.O. VALUE.
C
C
C     READ AND WRITE THE LABEL RECORD.
C     THIS RECORD CONTAINS N - THE NUMBER OF
C     VALUES IN EACH SUBSEQUENT RECORD.
C
      N=386
      CALL LUNIN2(DATA,        IDATA, 8980, 8990)
      WRITE(6, 3000) TYPE
      WRITE(6, 1000) (IDATA(I), I=1, 297)
C
C     INITIALIZE THE STACK
C
   10 IORG=1
      M=0
      L=0
C
C     CHECK FOR STACK OVERFLOW BEFORE READING THE NEXT RECORD.
C
   20 IF(IORG+N .GT. 12000) GO TO 970
      CALL LUNIN2(DATA(IORG), IDATA, 8980, 8990)
      IF(ITYPE(1) .GE. 6) GO TO 40
      WRITE(6, 2000) TYPE
```

```
      GO TO 20
C
C
C     THIS SECTION IS ENTERED ONLY FOR
C     RANGE AND V.C.O. RECORDS.
C
C
C
   40 WRITE(6, 3000) TYPE
      IF(ITYPE(2) .EQ. 6) GO TO 60
C
C     FOR RANGE DATA - MOVE IORG TO POINT ONE LOCATION BEYOND THE
C     LAST VALUE, AND INCREMENT THE COUNT OF RANGE BLOCKS (M).
C
      IF(FIRST) IXX=IORG
      IORG=IORG+N
      M=M+1
      IF(.NOT. LAST) GO TO 20
C
C     AFTER READING THE LAST RANGE BLOCK FOR THIS FREQUENCY,
C     FILL ARRAY "RANGE" WITH DISTANCES IN METERS CORRESPONDING
C     TO 0.1 WAVELENGTH INCREMENTS;  THEN COMPUTE THE NUMBER OF VALUES.
C
      DWL=29.97925/FREQ(ITYPE(1)-5)
      DO 50 I=1, 1000
      RANGE(I)=DWL*FLOAT(I-1)
   50 CONTINUE
      NPTSIN=M*N
      GO TO 20
C
C
C     TREATMENT OF V.C.O. DATA IS SIMILAR;  ONE SET OF V.C.O. VALUES
C     HAS BEEN ACCUMULATED WHEN THE COUNT OF V.C.O. BLOCKS (L) EQUALS M.
C
C
   60 IF(FIRST) IXY=IORG
      IORG=IORG+N
      L=L+1
      IF(L .LT. M) GO TO 20
C
C     CALL INTPOL TO OBTAIN V.C.O. VALUES AT EQUAL RANGE INTERVALS;
C     THE NEW VALUES ARE RETURNED IN ARRAY "VCO".
C
      NSTART=1
      NPLOT=1000
      CALL INTPOL(DATA(IXX), DATA(IXY),NPTSIN,RANGE, VCO, NSTART, NPLOT)
C
C     SET TO ZERO V.C.O. VALUES WHICH HAVE NOT BEEN INTERPOLATED.
```

```
C
      NSTM1=NSTART-1
      IF(NSTM1 .LE.0) GO TO 80
      DO 70 I=1, NSTM1
      VCO(I)=0.0
   70 CONTINUE
   80 NPLTP1=NPLOT+1
      IF(NPLTP1 .GT. 1000) GO TO 100
      DO 90 I=NPLTP1, 1000
      VCO(I)=0.0
   90 CONTINUE
  100 NPTS=NPLOT-NSTM1
C
C     WRITE HEADER INFORMATION AND ARRAY "VCO".
C
      DO 120 I=NSTART, NPLOT
      VCO(I)=VCO(I)+135.0
  120 CONTINUE
      WRITE(6,4000)  TYPE(1),FREQ(ITYPE(1)-5),NSTART,NPTS,
     .               (VCO(I),I=1,NPTS)
      WRITE(3)       TYPE(1),FREQ(ITYPE(1)-5),NSTART,NPTS,
     .               (VCO(I),I=1,NPTS)
C
C
C     IF LAST IS TRUE THEN READ A NEW SET OF RANGE VALUES:  OTHERWISE
C     READ V.C.O. DATA FOR THE NEXT COMPONENT (THE CURRENT RANGE DATA
C     ARE RETAINED).
C
C
      IF(LAST) GO TO 10
      IORG=IXY
      L=0
      GO TO 20
C
C
C     STACK OVERFLOW MESSAGE
C
C
  970 WRITE(6,7000)
      GO TO 999
C
C
C     END OF FILE ON INPUT WHEN MORE DATA WERE EXPECTED
C
C
  980 WRITE(6, 5000)
      GO TO 999
C
```

```
C
C      PROCESSING COMPLETED NORMALLY
C
C
  990 WRITE(6, 6000) TYPE
  999 END FILE 3
      RETURN
C
C
 1000 FORMAT(27(1X,11A4/))
 2000 FORMAT('0',2A8,' RECORD SKIPPED')
 3000 FORMAT('0',2A8,' RECORD READ')
 4000 FORMAT('1LABEL="',A8,'"'/ '0FREQ.= ',F5.1,' MHZ.'/
     .        '0FIRST POINT=',I4/ '0# OF POINTS=',I4/
     .        '0',10F10.3/99(1X,10F10.3/))
 5000 FORMAT('0NORMAL END OF JOB')
 6000 FORMAT('0END OF FILE OCCURRED WHILE ATTEMPTING TO READ ',
     .        2A8, ' RECORD')
 7000 FORMAT('-***  INSUFFICIENT SPACE ON STACK  ***')
C
C
      END


************************************************************************
*                                                                      *
*                              LUNACPY3                                *
*                                                                      *
************************************************************************

      REAL*8    TYPE(2), RUN, SITE, DIRECT, FORREV, TITLE(11)
      REAL*4    DATA(12000), RANGE(1000), VCO(1000)
      REAL*4    FREQ(6) /1.0, 2.1, 4.0, 8.1, 16.0, 32.1 /
      INTEGER*4 IDATA(400)
      INTEGER*2 ITYPE(2)
      LOGICAL*4 FIRST, LAST
      EQUIVALENCE (DATA(1), IDATA(1))
      COMMON /LUNDAT/ TITLE, RUN, SITE, DIRECT, FORREV, TYPE,
     .                ITYPE, N, FIRST, LAST
C
C
C      READ LUNAR SEP FILE (#3) AND PRODUCE A FILE OF V.C.O. DATA
C      INTERPOLATED AT INTERVALS OF 0.1 WAVELENGTH
C      RANGE DATA ARE TAKEN FROM FILE #2
C
C      THE RANGE AND V.C.O. DATA ARE ACCUMULATED IN ARRAY "DATA".
C      IORG IS THE INDEX OF THE NEXT FREE LOCATION INTO WHICH DATA
```

```
C     MAY BE STORED.  IXY IS THE INDEX OF THE FIRST RANGE VALUE,
C     AND IXY IS THE INDEX OF THE FIRST V.C.O. VALUE.
C
C
C     READ AND WRITE THE LABEL RECORD.
C     THIS RECORD CONTAINS N - THE NUMBER OF
C     VALUES IN EACH SUBSEQUENT RECORD.
C
      N=386
      CALL LUNIN2(DATA,        IDATA, 6980, 6990)
      CALL LUNIN3(DATA,        IDATA, 6980, 6990)
      WRITE(6, 3000) TYPE
      WRITE(6, 1000) (IDATA(I), I=1, 297)
C
C     INITIALIZE THE STACK
C
   10 IORG=1
      M=0
      L=0
C
C     CHECK FOR STACK OVERFLOW BEFORE READING THE NEXT RECORD.
C
   20 IF(IORG+N .GT. 12000) GO TO 970
      CALL LUNIN2(DATA(IORG), IDATA, 6980, 6990)
      IF(ITYPE(2) .NE. 5)
     .  CALL LUNIN3(DATA(IORG), IDATA, 6980, 6990)
      IF(ITYPE(1) .GE. 6) GO TO 40
      WRITE(6, 2000) TYPE
      GO TO 20
C
C
C     THIS SECTION IS ENTERED ONLY FOR
C     RANGE AND V.C.O. RECORDS.
C
C
   40 WRITE(6, 3000) TYPE
      IF(ITYPE(2) .EQ. 6) GO TO 60
C
C     FOR RANGE DATA - MOVE IORG TO POINT ONE LOCATION BEYOND THE
C     LAST VALUE, AND INCREMENT THE COUNT OF RANGE BLOCKS (M).
C
      IF(FIRST) IXX=IORG
      IORG=IORG+N
      M=M+1
      IF(.NOT. LAST) GO TO 20
C
C     AFTER READING THE LAST RANGE BLOCK FOR THIS FREQUENCY,
C     FILL ARRAY "RANGE" WITH DISTANCES IN METERS CORRESPONDING
```

```
C     TO 0.1 WAVELENGTH INCREMENTS;  THEN COMPUTE THE NUMBER OF VALUES.
C
      DWI=29.97925/FREQ(ITYPE(1)-5)
      DO 50 I=1, 1000
      RANGE(I)=DWL*FLOAT(I-1)
   50 CONTINUE
      NPTSIN=M*N
      GO TO 20
C
C
C     TREATMENT OF V.C.O. DATA IS SIMILAR;  ONE SET OF V.C.O. VALUES
C     HAS BEEN ACCUMULATED WHEN THE COUNT OF V.C.O. BLOCKS (L) EQUALS M.
C
C
   60 IF(FIRST) IXY=IORG
      IORG=IORG+N
      L=L+1
      IF(L .LT. M) GO TO 20
C
C     CALL INTPOL TO OBTAIN V.C.O. VALUES AT EQUAL RANGE INTERVALS;
C     THE NEW VALUES ARE RETURNED IN ARRAY "VCO".
C
      NSTART=1
      NPLOT=1000
      CALL INTPOL(DATA(IXX), DATA(IXY),NPTSIN,RANGE, VCO, NSTART, NPLOT)
C
C     SET TO ZERO V.C.O. VALUES WHICH HAVE NOT BEEN INTERPOLATED.
C
      NSTM1=NSTART-1
      IF(NSTM1 .LE.0) GO TO 80
      DO 70 I=1, NSTM1
      VCO(I)=0.0
   70 CONTINUE
   80 NPLTP1=NPLOT+1
      IF(NPLTP1 .GT. 1000) GO TO 100
      DO 90 I=NPLTP1, 1000
      VCO(I)=0.0
   90 CONTINUE
  100 NPTS=NPLOT-NSTM1
C
C     WRITE HEADER INFORMATION AND ARRAY "VCO".
C
      DO 120 I=NSTART, NPLOT
      VCO(I)=VCO(I)+135.0
  120 CONTINUE
      WRITE(6,4000)  TYPE(1),FREQ(ITYPE(1)-5),NSTART,NPTS,
     .               (VCO(I),I=1,NPTS)
      WRITE(3)       TYPE(1),FREQ(ITYPE(1)-5),NSTART,NPTS,
```

```
     .                    (VCO(I),I=1,NPTS)
C
C
C     IF LAST IS TRUE THEN READ A NEW SET OF RANGE VALUES;  OTHERWISE
C     READ V.C.O. DATA FOR THE NEXT COMPONENT (THE CURRENT RANGE DATA
C     ARE RETAINED).
C
C
      IF(LAST) GO TO 10
      IORG=IYY
      L=0
      GO TO 20
C
C
C     STACK OVERFLOW MESSAGE
C
C
  970 WRITE(6,7000)
      GO TO 999
C
C
C     END OF FILE ON INPUT WHEN MORE DATA WERE EXPECTED
C
C
  980 WRITE(6, 5000)
      GO TO 999
C
C
C     PROCESSING COMPLETED NORMALLY
C
C
  990 WRITE(6, 6000) TYPE
  999 END FILE 3
      RETURN
C
C
 1000 FORMAT(27(1X,11A4/))
 2000 FORMAT('0',2A8,' RECORD SKIPPED')
 3000 FORMAT('0',2A8,' RECORD READ')
 4000 FORMAT('1LABEL="',A8,'"'/ '0FREQ.= ',F5.1,' MHZ.'/
     .        '0FIRST POINT=',I4/ '0# OF POINTS=',I4/
     .        '0',10F10.3/99(1X,10F10.3/))
 5000 FORMAT('0NORMAL END OF JOB')
 6000 FORMAT('0END OF FILE OCCURRED WHILE ATTEMPTING TO READ ',
     .       2A8, ' RECORD')
 7000 FORMAT('-***  INSUFFICIENT SPACE ON STACK  ***')
C
C
```

```
      END

*********************************************************************
*                                                                   *
*                            LUNACPY4                               *
*                                                                   *
*********************************************************************

C
C     ROUTINE TO COPY THE RANGE AND VCO (UNCORRECTED) DATA FOR THE
C     EP-4 TURN, FOR USE BY THE ANTENNA PATTERN PLOT PROGRAM(S)
C
C
C     THE RANGE AND V.C.O. DATA ARE ACCUMULATED IN ARRAY "DATA".
C     IORG IS THE INDEX OF THE NEXT FREE LOCATION INTO WHICH DATA
C     MAY BE STORED.  IXX IS THE INDEX OF THE FIRST RANGE VALUE,
C     AND IXY IS THE INDEX OF THE FIRST V.C.O. VALUE.
C
C
C     SIX NAMELIST CARDS ARE REQUIRED AS DESCRIBED BELOW.
C
C     NAMELIST / CNTL /
C
C       IFREQ  - FREQUENCY INDICATOR (BASE 2 LOG OF FREQUENCY)
C                NO DEFAULT
C
C       ICOMP  - ARRAY OF COMPONENTS TO BE COPIED,  OR ZEROES TO PAD
C                THE ARRAY OUT TO 6 ELEMENTS, DEFAULT 6 ZEROES
C                CODES FOR THE COMPONENTS ARE:
C
C                    ENDFIRE   BROADSIDE
C
C                RHO   212         211
C                PHI   222         221
C                ZED   232         231
C
C
      REAL*8    TYPE(2), RUN, SITE, DIRECT, FORREV, TITLE(11)
      REAL*8 PROGNM(2) / 'OOGP.JCR', 'GAP' /
      REAL*4    DATA(12000), RANGE(1000), VCO(1000)
      REAL*4    FREQ(6) /1.0, 2.1, 4.0, 8.1, 16.0, 32.1 /
      INTEGER*4 IDATA(400)
      INTEGER*2 ITYPE(2)
      LOGICAL*4 FIRST, LAST
      INTEGER*2  ICOMP(6)
      INTEGER*2 COMP(6) / 212,222,232,211,221,231 /
```

```
      LOGICAL*1 DECIDE(6,6) / 36 * .TRUE. /
      NAMELIST / CNTL / IFREQ, ICOMP
      EQUIVALENCE (DATA(1), IDATA(1))
      COMMON /LUNDAT/ TITLE, RUN, SITE, DIRECT, FORREV, TYPE,
     .                ITYPE, N, FIRST, LAST
C
C     BEGIN BY SETTING DEFAULT VALUES FOR COMPONENT SELECTION
C     (NO COMPONENTS), READING CONTROL CARDS, AND SETTING
C     CONTROL PARAMETERS
C
      DO 10500 I=1,6
      DO 10100 J=1,6
10100 ICOMP(J)=0
      READ(5,CNTL,END=10600)
      IDX=IFREQ+1
      DO 10120 J=1,6
      IC= COMP(J)
      DO 10110 K=1,6
      IF(IC.EQ.ICOMP(K)) GO TO 10120
10110 CONTINUE
      DECIDE(IDX,J)=.FALSE.
10120 CONTINUE
10500 CONTINUE
10600 CONTINUE
      N=386
C
C     SKIP THE LABEL BLOCK
C
      CALL LUNIN2(DATA,        IDATA, 8980, 8990)
      CALL LUNIN3(DATA,        IDATA, 8980, 8990)
C
C     INITIALIZE THE STACK
C
   10 IORG=1
      M=0
      L=0
   20 IF(IORG+N .GT. 12000) GO TO 970
      CALL LUNIN2(DATA(IORG), IDATA, 8980, 8990)
      IF(ITYPE(2) .NE. 5)
     .  CALL LUNIN3(DATA(IORG), IDATA, 8980, 8990)
      IF(ITYPE(1) .GE. 6) GO TO 40
      GO TO 20
C
C
   40 CONTINUE
      IF(ITYPE(2) .EQ. 6) GO TO 60
C
C     ACCUMULATE RANGE BLOCKS
```

```
C
      IF(FIRST) IXX=IORG
      IORG=IORG+N
      M=M+1
      IF(.NOT. LAST) GO TO 20
      NPTSIN=N*M
      IGX=IXX
      IGXEND=IXX
C
C     FIND THE POINTS WHICH LIE BETWEEN 490 AND 535 METERS:
C     THESE WILL BE COPIED
C
      DO 50 I=IXX,NPTSIN
      IF(DATA(I).LE.490.) IGX=IGX+1
      IF(DATA(I) .LE. 535.) IGXEND = IGXEND + 1
   50 CONTINUE
      NCOMP=0
      GO TO 20
C
C     ACCUMULATE VCO BLOCKS
C
   60 IF(FIRST) IXY=IORG
      IORG=IORG+N
      L=L+1
      IF(L .LT. M) GO TO 20
      IF(NCOMP .GT. 0) GO TO 65
C
C     ISOLATE THE POINTS TO BE COPIED
C
      IGY=IXY+IGX-IXX
      IGYEND=IXY+IGXEND-IXX
      NPTS=IGYEND-IGY
   65 CONTINUE
      NCOMP=NCOMP+1
      IF(.NOT.DECIDE(ITYPE(1)-5,NCOMP)) GO TO 150
      YMIN=DATA(IGY)+135.
      YMAX=YMIN
      IY=IGY
C
C     ADJUST THE DATA VALUES TO RELATIVE DB, AND FIND
C     MAXIMUM AND MINIMUM VALUES
C
      DO 70 I=1,NPTS
      DATA(IY)=DATA(IY)+135.
      IF(DATA(IY) .LT. YMIN) YMIN=DATA(IY)
      IF(DATA(IY) .GT. YMAX)  YMAX=DATA(IY)
      IY=IY+1
   70 CONTINUE
```

```
      IF(NCOMP .GT. 1) GO TO 75
      IGXEND = IGXEND - 1
      IGYEND = IGYEND - 1
   75 CONTINUE
C
C     WRITE OUT THE ACCUMULATED DATA
C
      WRITE(1) FREQ(ITYPE(1)-5), NCOMP, YMIN, YMAX, NPTS,
     .          (DATA(I), I=IGX,IGXEND), (DATA(I), I=IGY,IGYEND)
C
C     IF THIS WAS THE SIXTH COMPONENT FOR THIS FREQUENCY, READ
C     RANGE DATA FOR THE NEXT FREQUENCY;  OTHERWISE READ
C     VCO DATA FOR THE NEXT COMPONENT
C
  150 IF(LAST) GO TO 10
      IORG=IXY
      L=0
      GO TO 20
C
C     STACK ARRAY TOO SMAIL
C
  970 WRITE(6,7000)
      GO TO 999
C
C     ALL PROCESSING COMPLETED NORMALLY
C
  980 WRITE(6, 5000)
      GO TO 999
C
C     PREMATURE END OF INPUT FILE
C
  990 WRITE(6, 6000) TYPE
  999 END FILE 1
      RETURN
C
C
 1000 FORMAT(27(1X,11A4/))
 2000 FORMAT('0',2A8,' RECORD SKIPPED')
 3000 FORMAT('0',2A8,' RECORD READ')
 4000 FORMAT('1LABEL="',A8,'"'/ '0FREQ.= ',F5.1,' MHZ.'/
     .       '0FIRST POINT=',I4/ '0# OF POINTS=',I4/
     .       '0',10F10.3/99(1X,10F10.3/))
 5000 FORMAT('0NORMAL END OF JOB')
 6000 FORMAT('0END OF FILE OCCURRED WHILE ATTEMPTING TO READ ',
     .       2A8, ' RECORD')
 7000 FORMAT('-***  INSUFFICIENT SPACE ON STACK  ***')
C
C
```

```
      END

      ***********************************************************************
      *                                                                     *
      *                            LUNACPY5                                 *
      *                                                                     *
      ***********************************************************************

C            PROGRAM TO EXTRACT TEMPERATURE, CALIBRATION,
C            TRANSMITTER-OFF, AND SELECTED RANGE INFORMATION FROM
C            LUNAR SEP FILE # 2.
C
C            THE RANGE DATA ASSOCIATED WITH THE TEMPERATURE DATA ARE
C            A DIRECT COPY OF THE 1 MHZ. RANGE ARRAY.
C
C            A SECOND ARRAY CONTAINS RANGE VALUES MATCHED WITH THE
C            CALIBRATION AND TXOFF DATA BY SELECTING EVERY 13-TH POINT
C            FROM THE 32 MHZ. RANGE ARRAY, BEGINNING WITH THE 7-TH.
C
C            THE ARRAY OF TEMPERATURE DATA IS COPIED DIRECTLY OFF THE
C            INPUT FILE.  THE CALIBRATION AND TXOFF DATA ARE IN A
C            MULTIPLEXED FORM ON THE INPUT FILE (C.F. - L. WATTS NOTES
C            - 18.I.74).  THE PROGRAM DEMULTIPLEXES THIS INFORMATION
C            AND STORES IT IN TWO ARRAYS:
C
C                    CAL(I, IFREQ, J), AND
C
C                  TXOFF(I, IFREQ, K),
C
C            WHERE I INDICATES THE I-TH VALUE IN SEQUENCE AND IFREQ IS
C            THE (INTEGRAL) BASE-2 LOGARITHM OF THE FREQUENCY.
C            J = 1, 2, 3 CORRESPOND TO CALIBRATION FOR GROUND,
C            NOISE DIODE + 20 DB, AND NOISE DIODE SOURCES RESPECTIVELY.
C            K = 1, 2, 3 INDICATE TXOFF INFORMATION FOR THE X, Y,
C            AND Z ANTENNAE RESPECTIVELY.
C
C
C
      REAL*4 RANGE(386), DATA(5018), RANGE2(140, 6)
      REAL*4 CAL(140, 6, 3), TXOFF(140, 6, 3)
      REAL*4 SPEED(140, 6)
      INTEGER*4 MODE(386)
      INTEGER*4 NCAL(6, 3) / 18 * 0 /
      INTEGER*4 NTXOFF(6, 3) / 18 * 0 /
      INTEGER*4 NR(3) / 3 * 0 /
      INTEGER*2 ITYPE(2)
```

```
      LOGICAL*4 FIRST, LAST
C
      COMMON / LUNDAT / JUNK(34), ITYPE, N, FIRST, LAST
C
C
      N = 386
      IORG = 1
      DO 15 K = 1, 6
        DO 10 J = 1, 140
          RANGE2(J, K) = 0.
          DO 5 L = 1, 3
            CAL(J, K, L)    = 0.
            TXOFF(J, K, L)  = 0.
    5     CONTINUE
   10   CONTINUE
   15 CONTINUE
C
C
   20 CALL LUNIN2(DATA(IORG), MODE, &900, &300)
C
      IT = ITYPE(1)
      GO TO (20, 40, 100, 200, 300, 400, 20, 20, 20, 20, 500), IT
C
C
C             DELETE THE '9' DIGIT FROM EACH ELEMENT OF THE MODE ARRAY.
C
   40 CONTINUE
      DO 60 I = 1, N
        MODE(I) = MODE(I) / 10
   60 CONTINUE
      GO TO 20
C
C
C             THE TEMPERATURE ARRAY IS WRITTEN OUT IMMEDIATELY.
C
  100 WRITE(3)        (DATA(I), I = 1, N)
      WRITE(6,1000)   (DATA(I), I = 1, N)
      GO TO 20
C
C
C             DEMULTIPLEX TXOFF DATA INTO  ARRAY TXOFF;
C             NTXOFF(IFREQ, M) CONTAINS THE MAXIMUM K FOR
C             TXOFF(K, IFREQ, M).
C
  200 IF(FIRST) MM = 0
      M = MOD(MM, 3) + 1
      MM = MM + 1
      L = 0
```

```
      IF(MM .GT. 3) L = 1
      DO 250 I = 1, N
        IFREQ = 2 * (4 - MOD(MODE(I), 10)) - L
        NTXOFF(IFREQ, M) = NTXOFF(IFREQ, M) + 1
        TXOFF(NTXOFF(IFREQ, M), IFREQ, M) = DATA(I)
  250 CONTINUE
      GO TO 20
C
C
C             FOLLOW THE SAME PROCEDURE TO DEMULTIPLEX THE
C             CALIBRATION DATA.
C
  300 IF(FIRST) MM = 0
      M = MOD(MM, 3) + 1
      MM = MM + 1
      L = 0
      IF(MM .GT. 3) L = 1
      DO 350 I = 1, N
        IFREQ = 2 * (4 - MOD(MODE(I), 10)) - L
        NCAL  (IFREQ, M) = NCAL  (IFREQ, M) + 1
        CAL  (  NCAL(IFREQ, M), IFREQ, M) = DATA(I)
  350 CONTINUE
      GO TO 20
C
C
C             THE RANGE ARRAY FOR 1 MHZ. IS PAIRED WITH THE
C             TEMPERATURE ARRAY.
C
  400 IF(ITYPE(2) .EQ. 6) GO TO 20
      WRITE(3)        (DATA(I), I = 1, N)
      WRITE(6,1020)  (DATA(I), I = 1, N)
      GO TO 20
C
C
C             ACCUMULATE 32 MHZ. RANGE BLOCKS
C
  500 IF(FIRST) NN = 0
      NN = NN + N
      IORG = IORG + N
      IF(.NOT. LAST) GO TO 20
C
C
C             DEMULTIPLEX THE RANGE DATA TO MATCH THE CALIBRATION
C             AND TXOFF ARRAYS.
C
      DO 600 I = 7, NN, 13
        II = (I - 7) / 13 + 1
        IFREQ = 4 - MOD(MODE(II), 10)
```

```
         NR(IFREQ) = NR(IFREQ) + 1
         RANGE2(NR(IFREQ), 2 * IFREQ    ) = DATA(I)
         RANGE2(NR(IFREQ), 2 * IFREQ - 1) = DATA(I)
         SPEED(NR(IFREQ), 2 * IFREQ) =
     .     1.234568 * (DATA(I + 1) - DATA(I - 1))
         SPEED(NR(IFREQ), 2 * IFREQ - 1) =
     .     SPEED(NR(IFREQ), 2 * IFREQ)
  600 CONTINUE
C
C
C              WRITE AND LIST THE CALIBRATION, TXOFF, AND
C              ASSOCIATED RANGE INFORMATION.
C
      WRITE(3) CAL, NCAL
      WRITE(3) TXOFF, NTXOFF
      WRITE(3) RANGE2, NR
      WRITE(3) SPEED
C
      DO 700 IFREQ = 1, 6
        IFR = 2 ** (IFREQ - 1)
        NN = NR((IFREQ - 1) / 2 + 1)
        WRITE(6,1050) IFR, ( RANGE2(L, IFREQ),
     .                       (CAL(L, IFREQ, K), K = 1, 3),
     .                       (TXOFF(L, IFREQ, J), J = 1, 3), L = 1, NN)
  700 CONTINUE
C
C
  900 RETURN
C
C
 1000 FORMAT(' TEMPERATURE' // 26(1X, 15F8.1 / ))
C
 1020 FORMAT('0RANGES FOR TEMPERATURE ARRAY' // 26(1X, 15F8.1 / ))
C
 1050 FORMAT('1', I3, ' MHZ.', 29X, 'CALIBRATION', 37X,
     .       'TRANSMITTER-OFF' / 11X, 'RANGE', 14X, 'GROUND', 6X,
     .       'NOISE +20', 10X, 'NOISE', 19X, 'X', 14X, 'Y', 14X, 'Z'
     .       // 10(1X, F15.1, 2(5X, 3F15.1) / ))
C
      END
```

```
*******************************************************************************
*                                                                             *
*                                 LUNALIST                                    *
*                                                                             *
*******************************************************************************

C
C      PROGRAM TO LIST LUNAR DATA
C
       REAL*8    TITLE(11), RUN, SITE, DIRECT, FORREV, TYPE(2)
       REAL*4    DATA(825)
      INTEGER*4 IDATA(825)
      INTEGER*2 ITYPE(2)
       INTEGER*2 ITYSAV / 0 /,  LINCNT / 0 /
C
       COMMON /LUNDAT/ TITLE,  RUN,    SITE,   DIRECT, FORREV,
     .               TYPE,   ITYPE,  N
C
C      FLAGS FOR TEMPORARY TRAP
C
       LOGICAL*1 TRAP / .FALSE. /,  SKIP / .FALSE. /
C
C      BEGIN EXECUTABLE CODE
C
C      GET (NEXT) INPUT RECORD
C
   20 CALL LUNIN  (DATA, IDATA, &400, &500)
   21 CONTINUE
       IF(.NOT.SKIP)
     .         GO TO 30
C          ELSE
               IF(ITYPE(1) .NE. 1)
     .                 GO TO 20
C                  ELSE
                       IF(LINCNT .LE. 44)
     .                         GO TO 22
C                          ELSE
                               WRITE(6, 35) TYPE
                               LINCNT = 0
                               GO TO 28
   22                  WRITE(6, 25) TYPE
   25                  FORMAT('0<<< ', 2A8, ' >>>' / 2X)
   28                  LINCNT = LINCNT + 16
                       GO TO 80
   30  IF(ITYPE(1) .LE. 3 .OR. ITYPE(2) .NE. ITYSAV)
     .         WRITE(6, 35) TYPE
   35          FORMAT('1<<< ', 2A8, ' >>>' / 2X)
```

```
                LINCNT = 3
                ITYSAV = ITYPE(1)
C
C       CHOOSE APPROPRIATE OUTPUT FORMAT
C
   80   IF(ITYPE(1) - 2)  100, 200, 300
C
C             HEADER
C
  100         WRITE(6, 105) RUN, SITE, DIRECT, FORREV, TITLE, N
  105         FORMAT ('0RUN ', A6 / '0SITE ', A6 / '0DIRECTION ', A6 /
     .                '0', A6, ' TRANSMITTER' / '0', 10A8, A4 /
     .                '0', I6, ' POINTS' )
C
C             CHECK FOR ARRAY OVERFLOW
C
              IF(N .GT. 825)
     .            N = 825
C
C             COMPUTE NUMBER OF LINES REQUIRED FOR LISTING
C
              NL = MAX0(N / 15 ,   N / 15 + 1 ) + 2
C
C             TEMPORARY TRAP TO RESTRICT PRINTOUT
C
              IF(TRAP)
     .            SKIP = .TRUE.
              TRAP = .TRUE.
              GO TO 20
C
C             MODE
C
  200         CONTINUE
C
C             TRAP
C
              IF(SKIP)
     .            GO TO 20
C
              WRITE(6, 1000)
              WRITE(6, 225)  (IDATA(I), I=1,N)
  225         FORMAT(54 (1X, 15I7 / ), 1X, 15I7)
              GO TO 20
C
C             ALL OTHER DATA
C
C             TRAP
C
```

```
  300       CONTINUE
            IF(SKIP)
     .          GO TO 20
C
            IF(LINCNT + NL .LE. 60)
     .              GO TO 320
C               ELSE
                    WRITE(6, 35) TYPE
                    LINCNT = 3
  320       WRITE(6, 1000)
            WRITE(6, 350) (DATA(I), I=1,N)
  350       FORMAT(54 (1X, 15F7.1 / ), 1X, 15F7.1)
            LINCNT = LINCNT + NL
            GO TO 20
C
C       RETURN POINTS FOR END OF FILE CONDITIONS
C
  400   WRITE(6, 410)
  410   FORMAT('-NORMAL END OF FILE DETECTED')
        GO TO 900
C
  500   WRITE(6, 510)
  510   FORMAT('-ABNORMAL END OF FILE DETECTED')
C
  900   RETURN
C
 1000   FORMAT('0 ')
C
        END
```

```
*******************************************************************************
*                                                                             *
*                               LUNALST2                                      *
*                                                                             *
*******************************************************************************
```

```
C
C       PROGRAM TO LIST LUNAR DATA
C
        REAL*8    TITLE(11), RUN, SITE, DIRECT, FORREV, TYPE(2)
        REAL*4    DATA(825)
       INTEGER*4 IDATA(825)
       INTEGER*2 ITYPE(2)
        INTEGER*2 ITYSAV / 0 /,  LINCNT / 0 /
C
        COMMON /LUNDAT/ TITLE,  RUN,    SITE,   DIRECT, FORREV,
```

```
     .                  TYPE,    ITYPE,   N
C
C       FLAGS FOR TEMPORARY TRAP
C
        LOGICAL*1 TRAP / .FALSE. /,  SKIP / .FALSE. /
C
C       BEGIN EXECUTABLE CODE
C
C       GET (NEXT) INPUT RECORD
C
      N=386
   20 CALL LUNIN2(DATA, IDATA, 6400, 6500)
   21 CONTINUE
        IF(.NOT.SKIP)
     .            GO TO 30
C            ELSE
                IF(ITYPE(1) .NE. 1)
     .                      GO TO 20
C                   ELSE
                        IF(LINCNT .LE. 44)
     .                            GO TO 22
C                             ELSE
                                WRITE(6, 35) TYPE
                                LINCNT = 0
                                GO TO 29
   22                   WRITE(6, 25) TYPE
   25                   FORMAT('0<<< ', 2A8, ' >>>' / 2X)
   28                   LINCNT = LINCNT + 16
                        GO TO 90
   30   IF(ITYPE(1) .LE. 3 .OR. ITYPE(2) .EQ. ITYSAV)
     .            WRITE(6, 35) TYPE
   35             FORMAT('1<<< ', 2A8, ' >>>' / 2X)
                  LINCNT = 3
                  ITYSAV = ITYPE(1)
C
C       CHOOSE APPROPRIATE OUTPUT FORMAT
C
   90   IF(ITYPE(1) - 2)   100, 200, 300
C
C            HEADER
C
  100        WRITE(6, 105) (IDATA(I), I=1, 297)
  105        FORMAT (27(1X, 11A4 /), 2X)
C
C            CHECK FOR ARRAY OVERFLOW
C
             IF(N .GT. 825)
     .            N = 825
```

```
C
C           COMPUTE NUMBER OF LINES REQUIRED FOR LISTING
C
            NL = MAX0(N / 15 ,   N / 15 + 1 ) + 2
C
C           TEMPORARY TRAP TO RESTRICT PRINTOUT
C
            IF(TRAP)
     .         SKIP = .TRUE.
            TRAP = .TRUE.
            GO TO 20
C
C           MODE
C
  200       CONTINUE
C
C           TRAP
C
            IF(SKIP)
     .         GO TO 20
C
            WRITE(6, 1000)
            WRITE(6, 225)  (IDATA(I), I=1,N)
  225       FORMAT(54 (1X, 15I7 / ), 1X, 15I7)
            GO TO 20
C
C           ALL OTHER DATA
C
C           TRAP
C
  300       CONTINUE
            IF(SKIP)
     .         GO TO 20
C
            IF(LINCNT + NL .LE. 60)
     .            GO TO 320
C              ELSE
                  WRITE(6, 35)  TYPE
                  LINCNT = 3
  320       WRITE(6, 1000)
            WRITE(6, 350)  (DATA(I), I=1,N)
  350       FORMAT(54 (1X, 15F7.1 / ), 1X, 15F7.1)
            LINCNT = LINCNT + NL
            GO TO 20
C
C       RETURN POINTS FOR END OF FILE CONDITIONS
C
  400   WRITE(6, 410)
```

```
  410   FORMAT('-NORMAL END OF FILE DETECTED')
        GO TO 900
C
  500   WRITE(6, 510)
  510   FORMAT('-ABNORMAL END OF FILE DETECTED')
C
  900   RETURN
C
 1000   FORMAT('0 ')
C
        END


*****************************************************************************
*                                                                           *
*                                LUNALST3                                   *
*                                                                           *
*****************************************************************************

C
C       PROGRAM TO LIST LUNAR DATA
C
        REAL*8     TITLE(11), RUN, SITE, DIRECT, FORREV, TYPE(2)
        REAL*4     DATA(825)
       INTEGER*4 IDATA(825)
       INTEGER*2 ITYPE(2)
        INTEGER*2 ITYSAV / 0 /,  LINCNT / 0 /
C
        COMMON /LUNDAT/ TITLE,  RUN,     SITE,    DIRECT, FORREV,
     .               TYPE,    ITYPE,   N
C
C       FLAGS FOR TEMPORARY TRAP
C
        LOGICAL*1 TRAP / .FALSE. /,  SKIP / .FALSE. /
C
C       BEGIN EXECUTABLE CODE
C
C       GET (NEXT) INPUT RECORD
C
       N=386
   20 CALL LUNIN2(DATA, IDATA, &400, &500)
       IF(ITYPE(2) .NE. 5)
     .  CALL LUNIN3(DATA, IDATA, &400, &500)
   21 CONTINUE
        IF(.NOT.SKIP)
     .          GO TO 30
C           ELSE
```

```
      IF(ITYPE(1) .NE. 1)
     .        GO TO 20
C         ELSE
              IF(LINCNT .LE. 44)
     .             GO TO 22
C                 ELSE
                   WRITE(6, 35) TYPE
                   LINCNT = 0
                   GO TO 28
   22         WRITE(6, 25) TYPE
   25         FORMAT('0<<< ', 2A8, ' >>>' / 2X)
   28         LINCNT = LINCNT + 16
              GO TO 80
   30 IF(ITYPE(1) .LE. 3 .OR. ITYPE(2) .NE. ITYSAV)
     .     WRITE(6, 35) TYPE
   35      FORMAT('1<<< ', 2A8, ' >>>' / 2X)
           LINCNT = 3
           ITYSAV = ITYPE(1)
C
C     CHOOSE APPROPRIATE OUTPUT FORMAT
C
   80 IF(ITYPE(1) - 2)  100, 200, 300
C
C         HEADER
C
  100     WRITE(6, 105) (IDATA(I), I=1, 297)
  105     FORMAT (27(1X, 11A4 /), 2X)
C
C         CHECK FOR ARRAY OVERFLOW
C
          IF(N .GT. 825)
     .         N = 825
C
C         COMPUTE NUMBER OF LINES REQUIRED FOR LISTING
C
          NL = MAX0(N / 15 ,   N / 15 + 1 ) + 2
C
C         TEMPORARY TRAP TO RESTRICT PRINTOUT
C
          IF(TRAP)
     .         SKIP = .TRUE.
          TRAP = .TRUE.
          GO TO 20
C
C         MODE
C
  200     CONTINUE
C
```

```
C             TRAP
C
              IF(SKIP)
     .            GO TO 20
C
              WRITE(6, 1000)
              WRITE(6, 225) (IDATA(I), I=1,N)
  225         FORMAT(54 (1X, 15I7 / ), 1X, 15I7)
              GO TO 20
C
C             ALL OTHER DATA
C
C             TRAP
C
  300         CONTINUE
              IF(SKIP)
     .            GO TO 20
C
              IF(LINCNT + NL .LE. 60)
     .                GO TO 320
C                 ELSE
                      WRITE(6, 35) TYPE
                      LINCNT = 3
  320         WRITE(6, 1000)
              WRITE(6, 350) (DATA(I), I=1,N)
  350         FORMAT(54 (1X, 15F7.1 / ), 1X, 15F7.1)
              LINCNT = LINCNT + NL
              GO TO 20
C
C         RETURN POINTS FOR END OF FILE CONDITIONS
C
  400     WRITE(6, 410)
  410     FORMAT('-NORMAL END OF FILE DETECTED')
          GO TO 900
C
  500     WRITE(6, 510)
  510     FORMAT('-ABNORMAL END OF FILE DETECTED')
C
  900     RETURN
C
 1000     FORMAT('0 ')
C
          END
```

```
**************************************************************************
*                                                                        *
*                              LUNAPLOT                                  *
*                                                                        *
**************************************************************************

C
C     ROUTINE TO PLOT LUNAR DATA
C     FROM FILE #1
C     AFTER INTERPOLATION BY THE COPY PROGRAM
C
C     SEVEN NAMELIST CARDS ARE REQUIRED AS INPUT, SIX AS DESCRIBED
C     BELOW, AND ONE PLTID CARD AS DESCRIBED IN PLINIT.
C
C
C     NAMELIST / FRED /
C
C       IFREQ  - FREQUENCY INDICATOR (BASE 2 LOG OF FREQUENCY)
C                NO DEFAULT
C
C       XMIN   - MINIMUM WAVELENGTH TO BE PLOTTED, DEFAULT 0.0
C
C       XMAX   - MAXIMUM WAVELENGTH TO BE PLOTTED, DEFAULT 100.0
C
C       YMAX   - MAXIMUM (RELATIVE) DB VALUE TO BE PLOTTED, DEFAULT +7.5
C
C       ICOMP  - ARRAY OF COMPONENTS TO BE PLOTTED, OR ZEROES TO PAD
C                THE ARRAY OUT TO 6 ELEMENTS, DEFAULT 6 ZEROES
C                CODES FOR THE COMPONENTS ARE:
C
C                    ENDFIRE   BROADSIDE
C
C                RHO    212          211
C                PHI    222          221
C                ZED    232          231
C
C       FILT   - (LOGICAL) FILTERING REQUIRED, DEFAULT .FALSE.
C
C       COEFF  - FILTER COEFFICIENTS, OR ZEROES TO PAD THE ARRAY
C                TO 100 ELEMENTS (COEFFICIENTS SHOULD BE LEFT-JUSTIFIED
C                IN THE ARRAY, FOR DEFAULTS SEE DECLARATION OF COEFF
C
C       NCOEFF - NUMBER OF FILTER COEFFICIENTS, DEFAULT 11
C
C       REF    - RELATIVE DB VALUE AT WHICH A REFERENCE MARK IS TO BE
C                PLOTTED ON THE Y AXIS, DEFAULT 45.0
C
```

```
C        ABSREF - ABSOLUTE DB VALUE CORRESPONDING TO REF, DEFAULT 45.0
C
C        NOTES  - UP TO 32 CHARACTERS OF ANNOTATION, NO DEFAULT
C
C
C
      REAL*8      TITLE(2) / 2*'        ' /
      REAL*4      VCO(1000), RANGE(1000), WORK(1000)
      REAL*4      NOTES(8)  / 8 * '    ' /
      REAL*4 XLIM(6,2)  / 6*0.0, 16.0, 32.0, 60.0, 3*100.0 /
      REAL*4 YMAXS(6)  / 6*100.0 /
      REAL*4 REF
      REAL*4 XMIN, XMAX, YMAX
      REAL*4 COEFF(100) /-.0023, .0041, .0445, .1239, .2078,
     .                 .2440, .2078, .1239, .0445, .0041, -.0023,
     .              89*0.0                                          /
      INTEGER*4 COMP(6)  / 212, 222, 232, 211, 221, 231 /
      INTEGER*4 IFREQ, ICOMP(6), CFREQ
      INTEGER*4 NCOEFF/ 11 /
      LOGICAL*1 DECIDE(6,6) /36*.TRUE./, FILTER(6)/6*.FALSE./
      LOGICAL*1 BOTH, FILT
      NAMELIST /FREQ/ IFREQ, XMIN, XMAX, YMAX, ICOMP, FILT, COEFF,
     .                NCOEFF, REF, NOTES, ABSREF
C
C     INITIALIZE RANGE ARRAY
C
      DO 5 I=1,1000
      RANGE(I)=0.1*FLOAT(I-1)
    5 CONTINUE
C
C     SKIP THE LABEL RECORD
C
      READ(3)
      DO 500 I=1,6
C
C     INITIALIZE PLOTTING PARAMETERS TO DEFAULT VALUES , IF ANY
C
      XMIN=0.
      XMAX=100.
      FILT=.FALSE.
      YMAX=67.5
      REF=45.0
      ABSREF=45.0
      DO 100 J=1,6
  100 ICOMP(J)=0
C
C     READ PLOTTING PARAMETERS
C
```

```
      READ(5,FREQ,END=520)
      IDX=IFREQ+1
      DO 120 J=1,6
      IC=COMP(J)
      DO 110 K=1,6
      IF(IC .EQ. ICOMP(K)) GO TO 120
  110 CONTINUE
      DECIDE(IDX,J)=.FALSE.
  120 CONTINUE
      IF(XMIN .GT. XLIM(IDX,1)) XLIM(IDX,1)=XMIN
      IF(XMAX .LT. XLIM(IDX,2)) XLIM(IDX,2)=XMAX
      IF(YMAX .LT. YMAXS(IDX)) YMAXS(IDX)=YMAX
      FILTPE(IDX)=FILT
  500 CONTINUE
  520 DX=0.
C
C     INITIALIZE PLOTTER
C
      CALL PLINIT('OOGP.JCR.LUNAR  ')
C
C     LOOP THROUGH FREQUENCIES
C
      DO 900 I=1,6
C
C     DETERMINE NUMBER OF CURVES PER GRAPH
C
      NA=0
      NB=0
      XMIN=XLIM(I,1)
      XMAX=XLIM(I,2)
      DO 550 J=1,3
      IF(DECIDE(I,J  )) NA=NA+1
      IF(DECIDE(I,J+3)) NB=NB+1
  550 CONTINUE
      BOTH=.FALSE.
      CPERG=NA
      IF(NA+NB .GT.3) GO TO 560
      BOTH=.TRUE.
      CPERG=CPERG+NB
  560 CONTINUE
      MST=IFIX(XMIN*10.0+1.5)
      MPT=IFIX(XMAX*10.0+0.5)
C
C     LOOP THROUGH COMPONENTS
C
      DO 580 J=1,6
      IF(DECIDE(I,J)) GO TO 563
      READ(3,END=999)
```

```
      GO TO 580
  563 READ(3,END=999) TITLE(1), F,NST,NPT, (VCO(K),K=1,NPT)
      DO 565 K=1,NPT
      IF(VCO(K) .GT. YMAXS(I)) NST=K+1
  565 CONTINUE
C
C     COMPUTE FIRST POINT AND NUMBER OF POINTS TO BE PLOTTED
C
      NST=MAX0(NST,MST)
      NPT=MIN0(NPT,MPT)-NST+1
C
C     FILTER IF REQUESTED
C
      IF(FILTRE(I)) CALL FILTER(VCO(NST),NPT,COEFF,NCOEFF,WORK)
C
C     PLOT THE CURVE
C
      CALL DATPLT(TITLE,NOTES,CPERG,6.18,15.0,COMP(J),VCO(NST),
     .            RANGE(NST),NPT,1,17.0,1.1,F,REF,ABSREF)
      IF(BOTH .OR. J .NE. 4) GO TO 580
      CPERG=NP
  580 CONTINUE
  900 CONTINUE
C
C     TERMINATE THE PLOT WHEN EOF IS DETECTED ON TAPE
C
  999 CALL PLOTND
      RETURN
      END

*************************************************************************
*                                                                       *
*                               LUNAPLT2                                *
*                                                                       *
*************************************************************************

C
C     ROUTINE TO PLOT LUNAR DATA
C     FROM FILE #2
C     AFTER INTERPOLATION BY THE COPY PROGRAM
C
C     SEVEN NAMELIST CARDS ARE REQUIRED AS INPUT, SIX AS DESCRIBED
C     BELOW, AND ONE PLTID CARD AS DESCRIBED IN PLINIT.
C
C
C     NAMELIST / MREQ /
```

```
C
C        IFREQ  - FREQUENCY INDICATOR (BASE 2 LOG OF FREQUENCY)
C                 NO DEFAULT
C
C        XMIN   - MINIMUM WAVELENGTH TO BE PLOTTED, DEFAULT 0.0
C
C        XMAX   - MAXIMUM WAVELENGTH TO BE PLOTTED, DEFAULT 100.0
C
C        YMAX   - MAXIMUM (RELATIVE) DB VALUE TO BE PLOTTED, DEFAULT 67.5
C
C        ICOMP  - ARRAY OF COMPONENTS TO BE PLOTTED, OR ZEROES TO PAD
C                 THE ARRAY OUT TO 6 ELEMENTS, DEFAULT 6 ZEROES
C                 CODES FOR THE COMPONENTS ARE:
C
C                      ENDFIRE   BROADSIDE
C
C                 RHO    212         211
C                 PHI    222         221
C                 ZED    232         231
C
C        FILT   - (LOGICAL) FILTERING REQUIRED, DEFAULT .FALSE.
C
C        COEFF  - FILTER COEFFICIENTS, OR ZEROES TO PAD THE ARRAY
C                 TO 100 ELEMENTS (COEFFICIENTS SHOULD BE LEFT-JUSTIFIED
C                 IN THE ARRAY, FOR DEFAULTS SEE DECLARATION OF COFFF
C
C        NCOEFF - NUMBER OF FILTER COEFFICIENTS, DEFAULT 11
C
C        REF    - RELATIVE DB VALUE AT WHICH A REFERENCE MARK IS TO BE
C                 PLOTTED ON THE Y AXIS, DEFAULT 45.0
C
C        ABSREF - ABSOLUTE DB VALUE CORRESPONDING TO REF, DEFAULT 45.0
C
C        NOTES  - UP TO 32 CHARACTERS OF ANNOTATION, NO DEFAULT
C
C
C
      REAL*8     TITLE(2) / 2*'        ' /
      REAL*4     VCO(1000), RANGE(1000), WORK(1000)
      REAL*4     NOTES(8) / 8 * '    ' /
      REAL*4 XLIM(6,2) / 6*0.0, 16.0, 32.0, 60.0, 3*100.0 /
      REAL*4 YMAXS(6) / 6*100.0 /
      REAL*4 REF
      REAL*4 XMIN, XMAX, YMAX
      REAL*4 COEFF(100) /-.0023, .0041, .0445, .1239, .2078,
     .             .2440, .2078, .1239, .0445, .0041, -.0023,
     .           89*0.0                                              /
      INTEGER*4 COMP(6) / 212, 222, 232, 211, 221, 231 /
```

```
      INTEGER*4 IFREQ, ICOMP(6), CPERG
      INTEGER*4 NCOEFF/ 11 /
      LOGICAL*1 DECIDE(6,6) /36*.TRUE./, FILTRE(6)/6*.FALSE./
      LOGICAL*1 BOTH, FILT
      NAMELIST /FREQ/ IFREQ, XMIN, XMAX, YMAX, ICOMP, FILT, COEFF,
     .                   NCOEFF, REF, NOTES, ABSREF
C
C     INITIALIZE RANGE ARRAY
C
      DO 5 I=1,1000
      RANGE(I)=0.1*FLOAT(I-1)
    5 CONTINUE
C
      DO 500 I=1,6
C
C     INITIALIZE PLOTTING PARAMETERS TO DEFAULT VALUES , IF ANY
C
      XMIN=0.
      XMAX=100.
      FILT=.FALSE.
      YMAX=67.5
      REF=45.0
      ABSREF=45.0
      DO 100 J=1,6
  100 ICOMP(J)=0
C
C     READ PLOTTING PARAMETERS
C
      READ(5,FREQ,END=520)
      IDX=IFREQ+1
      DO 120 J=1,6
      IC=COMP(J)
      DO 110 K=1,6
      IF(IC .EQ. ICOMP(K)) GO TO 120
  110 CONTINUE
      DECIDE(IDX,J)=.FALSE.
  120 CONTINUE
      IF(XMIN .GT. XLIM(IDX,1)) XLIM(IDX,1)=XMIN
      IF(XMAX .LT. XLIM(IDX,2)) XLIM(IDX,2)=XMAX
      IF(YMAX .LT. YMAXS(IDX)) YMAXS(IDX)=YMAX
      FILTRE(IDX)=FILT
  500 CONTINUE
  520 DX=0.
C
C     INITIALIZE PLOTTER
C
      CALL PLINIT('OQGP.JCR.LUNAR  ')
C
```

```
C     LOOP THROUGH FREQUENCIES
C
      DO 900 I=1,6
C
C     DETERMINE NUMBER OF CURVES PER GRAPH
C
      NA=0
      NB=0
      XMIN=XLIM(I,1)
      XMAX=XLIM(I,2)
      DO 550 J=1,3
      IF(DECIDE(I,J  )) NA=NA+1
      IF(DECIDE(I,J+3)) NB=NB+1
  550 CONTINUE
      BOTH=.FALSE.
      CPERG=NA
      IF(NA+NB .GT.3) GO TO 560
      BOTH=.TRUE.
      CPERG=CPERG+NB
  560 CONTINUE
      MST=IFIX(XMIN*10.0+1.5)
      MPT=IFIX(XMAX*10.0+0.5)
C
C     LOOP THROUGH COMPONENTS
C
      DO 580 J=1,6
      IF(DECIDE(I,J)) GO TO 563
      READ(3,END=999)
      GO TO 580
  563 READ(3,END=999) TITLE(1), F,NST,NPT, (VCO(K),K=1,NPT)
      DO 565 K=1,NPT
      IF(VCO(K) .GT. YMAXS(I)) NST=K+1
  565 CONTINUE
C
C     COMPUTE FIRST POINT AND NUMBER OF POINTS TO BE PLOTTED
C
      NST=MAX0(NST,MST)
      NPT=MIN0(NPT,MPT)-NST+1
C
C     FILTER IF REQUESTED
C
      IF(FILTPE(I)) CALL FILTER(VCO(NST),NPT,COEFF,NCOEFF,WORK)
C
C     PLOT THE CURVE
C
      CALL DATPLT(TITLE,NOTES,CPERG,6.18,15.0,COMP(J),VCO(NST),
     .            RANGE(NST),NPT,1,17.0,1.1,F,REF,ABSREF)
      IF(BOTH .OR. J .NE. 4) GO TO 580
```

```
      CPP? I=NR
  550 CONTINUE
  900 CONTINUE
C
C     TERMINATE THE PLOT WHEN EOF IS DETECTED ON TAPE
C
  999 CALL PLOTND
      RETURN
      END

*************************************************************************
*                                                                       *
*                              LUNAPIT3                                 *
*                                                                       *
*************************************************************************

C
C
C     ROUTINE TO PLOT LUNAR SEP DATA FROM FILE #2;
C     NO INTERPOLATION;
C     VALUES WITH 510. M. <= RANGE <= 520. M.
C     ARE DELETED BEFORE PLOTTING.
C
C     THE RANGE AND V.C.O. DATA ARE ACCUMULATED IN ARRAY "DATA".
C     IORG IS THE INDEX OF THE NEXT FREE LOCATION INTO WHICH DATA
C     MAY BE STORED.  IXX IS THE INDEX OF THE FIRST RANGE VALUE,
C     AND IXY IS THE INDEX OF THE FIRST V.C.O. VALUE.
C
C     SEVEN NAMELIST CARDS ARE REQUIRED AS INPUT, SIX AS DESCRIBED
C     BELOW, AND ONE PLTIT CARD AS DESCRIBED IN PLINIT.
C
C
C     NAMELIST / CNTL /
C
C       IFREQ  - FREQUENCY INDICATOR (BASE 2 LOG OF FREQUENCY)
C                NO DEFAULT
C
C       XMIN   - MINIMUM WAVELENGTH TO BE PLOTTED, DEFAULT 0.0
C
C       XMAX   - MAXIMUM WAVELENGTH TO BE PLOTTED, DEFAULT 100.0
C
C       YMAX   - MAXIMUM (RELATIVE) DB VALUE TO BE PLOTTED, DEFAULT 67.5
C
C       ICOMP  - ARRAY OF COMPONENTS TO BE PLOTTED, OR ZEROES TO FILL
C                THE ARRAY OUT TO 6 ELEMENTS, DEFAULT 6 ZEROES
C                CODES FOR THE COMPONENTS ARE:
```

```
C
C                       ENDFIRE   BROADSIDE
C
C                   RHO    212         211
C                   PHI    222         221
C                   ZED    232         231
C
C         REF     - RELATIVE DB VALUE AT WHICH A REFERENCE MARK IS TO BE
C                   PLOTTED ON THE Y AXIS, DEFAULT 45.0
C
C         ABSREF  - ABSOLUTE DB VALUE CORRESPONDING TO REF, DEFAULT 45.0
C
C         NOTES   - UP TO 32 CHARACTERS OF ANNOTATION, NO DEFAULT
C
C
      REAL*8    TYPE(2), RUN, SITE, DIRECT, FORREV, TITLE(11)
      REAL*4    DATA(12000), RANGE(1000), VCO(1000)
      REAL*4    FREQ(6) /1.0, 2.1, 4.0, 8.1, 16.0, 32.1 /
      INTEGER*4 IDATA(400)
      INTEGER*2 ITYPE(2)
      LOGICAL*4 FIRST, LAST
      EQUIVALENCE (DATA(1), IDATA(1))
      REAL*8 NOTES(4) /4*' '/
      REAL*4 XLIM(6,2)/6*0.0,16.0,32.0,60.0,3*100.0/
      REAL*4 YMAXS(6)/6*100.0/
      INTEGER*4 ICOMP(6),NA(6),NB(6),CPERG(6,6)
      INTEGER*4 COMP(6)/212,222,232,211,221,231/
      LOGICAL*1 DECIDE(6,6)/36*.TRUE./,BOTH(6)
      NAMELIST/CNTL/IFREQ,XMIN,XMAX,YMAX,ICOMP,REF,NOTES,ABSREF
      COMMON /LUNDAT/ TITLE, RUN, SITE, DIRECT, FORREV, TYPE,
     .                ITYPE, N, FIRST, LAST
C
C
      DO 10500 I=1,6
C
C     INITIALIZE PLOTTING PARAMETERS TO DEFAULT VALUES , IF ANY
C
      XMIN=0.0
      XMAX=100.0
      YMAX=67.5
      REF=45.0
      ABSREF=45.0
      DO 10100 J=1,6
10100 ICOMP(J)=0
C
C     READ PLOTTING PARAMETERS
C
      READ(5,CNTL,END=10600)
```

```
      IDX=IZERO+1
      DO 10120 J=1,6
      IC= COMP(J)
      DO 10110 K=1,6
      IF(IC.EQ.ICOMP(K)) GO TO 10120
10110 CONTINUE
      DECIDE(IDX,J)=.FALSE.
10120 CONTINUE
C
C     SET MIN AND MAX RANGE, AND MAX VCO FOR THIS COMPONENT
C
      IF(XMIN.GT.XLIM(IDX,1)) XLIM(IDX,1)=XMIN
      IF(XMAX.LT.XLIM(IDX,2)) XLIM(IDX,2)=XMAX
      IF(YMAX.LT.YMAXS(IDX)) YMAXS(IDX)=YMAX
      NA(IDX)=0
      NB(IDX)=0
C
C     DECIDE ON THE NUMBER OF CURVES PER GRAPH
C
      DO 10200 J=1,3
      IF(DECIDE(IDX,J   )) NA(IDX)=NA(IDX)+1
      IF(DECIDE(IDX,J+3)) NB(IDX)=NB(IDX)+1
10200 CONTINUE
      BOTH(IDX)=.TRUE.
      IF(NA(IDX)+NB(IDX).GT.3) BOTH(IDX)=.FALSE.
      DO 10300 J=1,3
      IF(BOTH(IDX)) GO TO 10250
      CPERG(IDX,J   )=NA(IDX)
      CPERG(IDX,J+3)=NB(IDX)
      GO TO 10300
10250 CPERG(IDX,J   )=NA(IDX)+NB(IDX)
      CPERG(IDX,J+3)=CPERG(IDX,J)
10300 CONTINUE
10500 CONTINUE
10600 CONTINUE
C
C     INITIALIZE THE PLOTTER
C
      CALL PLINIT('OOGP.JCR.LUNAR   ')
      N=306
C
C     READ THE LABEL RECORD
C
      CALL LUNIN2(DATA,       IDATA, 6980, 6990)
      WRITE(6, 3000) TYPE
      WRITE(6, 1000) (IDATA(I), I=1, 297)
C
C     INITIALIZE THE STACK BEFORE READING RANGE DATA
```

```
C
   10 IORG=1
      M=0
      L=0
   20 IF(IORG+N .GT. 12000) GO TO 970
      CALL IUNIN2(DATA(IORG), IDATA, &980, &990)
      IF(ITYPE(1) .LE. 5) GO TO 20
C
C
      IF(ITYPE(2) .NE. 0) GO TO 60
C
C     ACCUMULATE RANGE BLOCKS
C
      IF(FIRST) IXX=IORG
      IORG=IORG+N
      M=M+1
      IF(.NOT. LAST) GO TO 20
      XMTOWL=FREQ(ITYPE(1)-5)/299.7925
      NPTSIN=M*N
      NXBGAP=0
      NXAGAP=0
C
C     FIND RANGE VALUES TO BE OMITTED
C
      DO 45 I=1,NPTSIN
      IF(DATA(I).GE.510.0) GO TO 50
      NXBGAP=NXBGAP+1
      DATA(I)=XMTOWL*(DATA(I)+3.0)
   45 CONTINUE
   50 INEXT=NXBGAP+1
      ISTART=INEXT
C
C     DELETE VALUES BY COMPRESSING THE ARRAY
C
      DO 55 I=ISTART,NPTSIN
      IF(DATA(I).LE.520.0) GO TO 55
      DATA(INEXT)=XMTOWL*(DATA(I)+3.0)
      NXAGAP=NXAGAP+1
      INEXT=INEXT+1
   55 CONTINUE
C
C     RESET THE ORIGIN FOR VCO DATA, AND ZERO THE COMPONENT COUNTER
C
      IORG=NXBGAP+NXAGAP+1
      NCOMP=0
      GO TO 20
C
C
```

```
C     ACCUMULATE VCO BLOCKS
C
C
   60 IF(FIRST) IXY=IORG
      IORG=IORG+N
      L=L+1
      IF(L .LT. M) GO TO 20
      IDX=ITYPE(1)-5
      NCOMP=NCOMP+1
      IF(.NOT.DECIDE(IDX,NCOMP)) GO TO 150
C
C     FIND THE LAST VALUE BEFORE THE GAP, AND ADJUST DB VALUES
C     TO A RELATIVE SCALE
C
      IY=IXY+NXBGAP-1
      DO 70 I=IXY,IY
   70 CONTINUE
      INEXT=IXY+NXBGAP
      ISTART=IORG-NXAGAP
      IEND=IORG-1
C
C     COMPRESS THE VCO DATA, ADJUSTING
C     TO RELATIVE SCALE IN THE PROCESS
C
      DO 80 I=ISTART,IEND
      DATA(INEXT)=DATA(I)+135.0
      INEXT=INEXT+1
   80 CONTINUE
      NSTX=IXX
      NX=IXX
      NPLOT=IXX+NXBGAP+NXAGAP-1
      IY=IXY
C
C     OMIT VALUES OUTSIDE THE OUTER BOUNDS
C
      DO 90 IX=NX,NPLOT
      IF(DATA(IX).LT.XLIM(IDX,1).OR.DATA(IY).GT.YMAXS(IDX))
     .        NSTX = IX + 1
      IY=IY+1
   90 CONTINUE
      NSTY=IXY+NSTX-IXX
      IX=NPLOT
      NX=NPLOT
      DO 100 I=NSTX,NX
      IF(DATA(IX).GT.XLIM(IDX,2)) NPLOT=NPLOT-1
      IX=IX-1
  100 CONTINUE
      NPTS=NPLOT-NSTX+1
```

```
C
C      PLOT THE CURVE
C
       CALL DATPLT(TYPE,NOTES,CPURG(IDY,NCOMP),6.18,15.0,COMP(NCOMP),
     .                 DATA(NSTY),DATA(NSTX),NPTS,1,17.,1.1,FREQ(IDY),
     .                 REF,ABSREF)
C
C
C      IF THIS WAS THE SIXTH COMPONENT, GET A NEW RANGE ARRAY, OTHERWISE
C      GET A VCO ARRAY FOR THE NEXT COMPONENT
C
C
  150  IF(LAST) GO TO 10
       IORG=IYY
       L=0
       GO TO 20
C
C      STACK ARRAY TOO SMALL
C
  970  WRITE(6,7000)
       GO TO 999
C
C      NORMAL COMPLETION
C
  980  WRITE(6, 5000)
       GO TO 999
C
C      PREMATURE END OF FILE ("TAPE" INPUT)
C
  990  WRITE(6, 6000) TYPE
  999  CALL PLOTND
       RETURN
C
C
 1000  FORMAT(27(1X,11A4/))
 2000  FORMAT('0',2A8,' RECORD SKIPPED')
 3000  FORMAT('0',2A8,' RECORD READ')
 4000  FORMAT('1LABEL="',A8,'"'/ '0FREQ.= ',F5.1,' MHZ.'/
     .         '0FIRST POINT=',I4/ '0# OF POINTS=',I4/
     .         '0',10F10.3/99(1X,10F10.3/))
 5000  FORMAT('0NORMAL END OF JOB')
 6000  FORMAT('0END OF FILE OCCURRED WHILE ATTEMPTING TO READ ',
     .         2A8, ' RECORD')
 7000  FORMAT('-***  INSUFFICIENT SPACE ON STACK  ***')
C
C
       END
```

```
***************************************************************************
*                                                                         *
*                               LUNAPLT4                                  *
*                                                                         *
***************************************************************************

C
C
C     ROUTINE TO PLOT SEP DATA THROUGH THE TURN AT LP-4 VS. RECORD NUMBER
C
C
C      THE RANGE AND V.C.O. DATA ARE ACCUMULATED IN ARRAY "DATA".
C      IORG IS THE INDEX OF THE NEXT FREE LOCATION INTO WHICH DATA
C      MAY BE STORED.  IXX IS THE INDEX OF THE FIRST RANGE VALUE,
C      AND IXY IS THE INDEX OF THE FIRST V.C.O. VALUE.
C
C
C      SEVEN NAMELIST CARDS ARE REQUIRED AS INPUT, SIX AS DESCRIBED
C      BELOW, AND ONE PLTID CARD AS DESCRIBED IN PLINIT.
C
C      NAMELIST / CNTL /
C
C        IFREQ  - FREQUENCY INDICATOR (BASE 2 LOG OF FREQUENCY)
C                 NO DEFAULT
C
C        ICOMP  - ARRAY OF COMPONENTS TO BE PLOTTED, OR ZEROES TO PAD
C                 THE ARRAY OUT TO 6 ELEMENTS, DEFAULT 6 ZEROES
C                 CODES FOR THE COMPONENTS ARE:
C
C                     ENDFIRE   BROADSIDE
C
C                 RHO   212         211
C                 PHI   222         221
C                 ZED   232         231
C
C
      REAL*8    TYPE(2), RUN, SITE, DIRECT, FOLREV, TITLE(11)
      REAL*8 PROGNM(2) / 'OOGP.JCR', 'GAP' /
      REAL*4    DATA(12000), RANGE(1000), VCO(1000)
      REAL*4    FREQ(6) /1.0, 2.1, 4.0, 8.1, 16.0, 32.1 /
      INTEGER*4 IDATA(400)
      INTEGER*2 ITYPE(2)
      LOGICAL*4 FIRST, LAST
      INTEGER*2 ICOMP(6)
      INTEGER*2 COMP(6) / 212,222,232,211,221,231 /
      LOGICAL*1 DECIDE(6,6) / 36 * .TRUE. /
      NAMELIST / CNTL / IFREQ, ICOMP
```

```
      EQUIVALENCE (DATA(1), IDATA(1))
      COMMON /LUNDAT/ TITLE, RUN, SITE, DIRECT, FORREV, TYPE,
     .                ITYPE, N, FIRST, LAST
C
C
      DO 10500 I=1,6
C
C     NO COMPONENTS PLOTTED UNLESS REQUESTED
C
      DO 10100 J=1,6
10100 ICOMP(J)=0
C
C     READ FREQUENCY INDICATOR AND COMPONENTS TO BE PLOTTED
C
      READ(5,CNTL,END=10600)
      IDX=IFREQ+1
      DO 10120 J=1,6
      IC= COMP(J)
      DO 10110 K=1,6
      IF(IC.EQ.ICOMP(K)) GO TO 10120
10110 CONTINUE
      DECIDE(IDX,J)=.FALSE.
10120 CONTINUE
10500 CONTINUE
10600 CONTINUE
C
C     INITIALIZE THE PLOTTER
C
      CALL PLINIT(PROGNM)
      N=386
C
C     SKIP THE LABEL BLOCK
C
      CALL LUNIN2(DATA,        IDATA, 8980, 8990)
      CALL LUNIN3(DATA,        IDATA, 8980, 8990)
C
C     INITIALIZE THE STACK
C
   10 IORG=1
      M=0
      L=0
   20 IF(IORG+N .GT. 12000) GO TO 970
      CALL LUNIN2(DATA(IORG), IDATA, 8980, 8990)
      IF(ITYPE(2) .NE. 5)
     .  CALL LUNIN3(DATA(IORG), IDATA, 8980, 8990)
      IF(ITYPE(1) .GE. 6) GO TO 40
      GO TO 20
C
```

```
C     ACCUMULATE RANGE BLOCKS
C
   40 CONTINUE
      IF(ITYPE(2) .EQ. 6) GO TO 60
      IF(FIRST) IXX=IORG
      IORG=IORG+N
      M=M+1
      IF(.NOT. LAST) GO TO 20
      NPTSIN=N*M
      IGX=IXX
      IGXEND=IXX
C
C     FIND THE GAP
C
      DO 50 I=IXX,NPTSIN
      IF(DATA(I).LE.490.) IGX=IGX+1
      IF(DATA(I) .LE. 535.) IGXEND = IGXEND + 1
   50 CONTINUE
      NCOMP=0
      GO TO 20
C
C     ACCUMULATE VCO BLOCKS
C
   60 IF(FIRST) IXY=IORG
      IORG=IORG+N
      L=L+1
      IF(L .LT. M) GO TO 20
      IGY=IXY+IGX-IXX
      IGYEND=IXY+IGXEND-IXX
      NPTS=IGYEND-IGY
      NCOMP=NCOMP+1
      IF(.NOT.DECIDE(ITYPE(1)-5,NCOMP)) GO TO 150
      YMIN=DATA(IGY)+135.
      YMAX=YMIN
      IY=IGY
C
C     ADJUST DB VALUES TO RELATIVE SCALE, AND FIND MINIMUM AND
C     MAXIMUM VCO THROUGH THE TURN
C
      DO 70 I=1,NPTS
      DATA(IY)=DATA(IY)+135.
      IF(DATA(IY) .LT. YMIN) YMIN=DATA(IY)
      IF(DATA(IY) .GT. YMAX)  YMAX=DATA(IY)
      IY=IY+1
   70 CONTINUE
C
C     PLOT THE POINTS
C
```

```
      CALL GAPLOT(FREQ(ITYPE(1)-5),DATA(IGX),DATA(IGY),
     .            NPTS, YMIN, YMAX, NCOMP)
C
C     IF THIS WAS THE SIXTH COMPONENT, GET NEW RANGE DATA;  OTHERWISE
C     GET NEW VCO DATA
C
  150 IF(LAST) GO TO 10
      IORG=IYY
      L=0
      GO TO 20
C
C     STACK ARRAY TOO SMALL
C
  970 WRITE(6,7000)
      GO TO 999
C
C     NORMAL COMPLETION
C
  980 WRITE(6, 5000)
      GO TO 999
C
C     PREMATURE END OF DATA FILE
C
  990 WRITE(6, 6000) TYPE
  999 CALL PLOTND
      RETURN
C
C
 1000 FORMAT(27(1X,11A4/))
 2000 FORMAT('0',2A8,' RECORD SKIPPED')
 3000 FORMAT('0',2A8,' RECORD READ')
 4000 FORMAT('1LABEL="',A8,'"'/ '0FREQ.= ',F5.1,' MHZ.'/
     .       '0FIRST POINT=',I4/ '0# OF POINTS=',I4/
     .       '0',10F10.3/99(1X,10F10.3/))
 5000 FORMAT('0NORMAL END OF JOB')
 6000 FORMAT('0END OF FILE OCCURRED WHILE ATTEMPTING TO READ ',
     .       2A8, ' RECORD')
 7000 FORMAT('-***  INSUFFICIENT SPACE ON STACK  ***')
C
C
      END
```

```
************************************************************************
*                                                                      *
*                              LUNAPLTS                                *
*                                                                      *
************************************************************************

C
C
C     ROUTINE TO PLOT LUNAR SEP DATA FROM FILE #2;
C     NO INTERPOLATION;
C     VALUES WITH 510. M. <= RANGE <= 520. M.
C     ARE DELETED BEFORE PLOTTING.
C
C     TRANSMITTER-OFF DATA ARE PLOTTED AS A BASELINE FOR EACH COMPONENT.
C
C     THE RANGE AND V.C.O. DATA ARE ACCUMULATED IN ARRAY "DATA".
C     IORG IS THE INDEX OF THE NEXT FREE LOCATION INTO WHICH DATA
C     MAY BE STORED.  IAX IS THE INDEX OF THE FIRST RANGE VALUE,
C     AND IXY IS THE INDEX OF THE FIRST V.C.O. VALUE.
C
C     SEVEN NAMELIST CARDS ARE REQUIRED AS INPUT, SIX AS DESCRIBED
C     BELOW, AND ONE PLTID CARD AS DESCRIBED IN PLINIT.
C
C
C     NAMELIST / CNTL /
C
C       IFREQ  - FREQUENCY INDICATOR (BASE 2 LOG OF FREQUENCY)
C                NO DEFAULT
C
C       XMIN,  - MINIMUM AND MAXIMUM RANGE VALUES TO BE PLOTTED, IN
C       XMAX     WAVELENGTHS IF VSWL = .TRUE., OTHERWISE IN METERS;
C                DEFAULTS: 0.0, 100.0
C
C       YMAX   - MAXIMUM (RELATIVE) DB VALUE TO BE PLOTTED, DEFAULT 67.5
C
C       ICOMP  - ARRAY OF COMPONENTS TO BE PLOTTED, OR ZEROES TO PAD
C                THE ARRAY OUT TO 6 ELEMENTS, DEFAULT 6 ZEROES
C                CODES FOR THE COMPONENTS ARE:
C
C                    ENDFIRE   BROADSIDE
C
C                RHO   212         211
C                PHI   222         221
C                ZED   232         231
C
C       VSWL   - (LOGICAL) IF TRUE (DEFAULT), DB VALUES ARE PLOTTED
C                VS. RANGE IN WAVELENGTHS, OTHERWISE VS. RANGE
```

```
C                 IN METRES.
C
C        REF    - RELATIVE DB VALUE AT WHICH A REFERENCE MARK IS TO BE
C                 PLOTTED ON THE Y AXIS, DEFAULT 45.0
C
C        ABSREF - ABSOLUTE DB VALUE CORRESPONDING TO REF, DEFAULT 45.0
C
C        NOTES  - UP TO 32 CHARACTERS OF ANNOTATION, NO DEFAULT
C
C
      REAL*8    TYPE(2), RUN, SITE, DIRECT, FORREV, TITLE(11)
      REAL*4    DATA(12000), RANGE(1000), VCO(1000)
      REAL*4 RANGE2(140, 6), TXOFF(140, 6, 3)
      REAL*4    FREQ(6) /1.0, 2.1, 4.0, 8.1, 16.0, 32.1 /
      INTEGER*4 IDATA(400)
      INTEGER*4 NTXOFF(6, 3), NP(3)
      INTEGER*2 ITYPE(2)
      LOGICAL*4 FIRST, LAST
      EQUIVALENCE (DATA(1), IDATA(1))
      REAL*8 NOTES(4) /4*' '/
      REAL*4 XLIM(6,2)
      REAL*4 YMAXS(6)
      INTEGER*4 ICOMP(6),NA(6),NB(6),CPREG(6,6)
      INTEGER*4 COMP(6)/212,222,232,211,221,231/
      INTEGER*4 NN(6)
      LOGICAL*1 DECIDE(6,6)/36*.TRUE./,BOTH(6)
      LOGICAL*1 VSNL / .TRUE. /
      NAMELIST/CNTL/IFREQ,XMIN,XMAX,YMAX,ICOMP,REF,ABSREF,NOTES,VSNL
      COMMON /LUNDAT/ TITLE, RUN, SITE, DIRECT, FORREV, TYPE,
     *                ITYPE, N, FIRST, LAST
C
C
      DO 10500 I=1,6
C
C     INITIALIZE PLOTTING PARAMETERS TO DEFAULT VALUES , IF ANY
C
      XMIN=0.0
      XMAX=100.0
      YMAX=67.5
      REF=45.0
      ABSREF=45.0
      DO 10100 J=1,6
10100 ICOMP(J)=0
C
C     READ PLOTTING PARAMETERS
C
      READ(5,CNTL,END=10600)
      IDX=IFREQ+1
```

```
      DO 10120 J=1,6
      IC= COMP(J)
      DO 10110 K=1,6
      IF(IC.EQ.ICOMP(K)) GO TO 10120
10110 CONTINUE
      DECIDE(IDX,J)=.FALSE.
10120 CONTINUE
C
C     SET MIN AND MAX RANGE, AND MAX VCO FOR THIS COMPONENT
C
      XLIM(IDX,1) = XMIN
      XLIM(IDX,2) = XMAX
      YMAXS(IDX) = YMAX
      NA(IDX)=0
      NB(IDX)=0
C
C     DECIDE ON THE NUMBER OF CURVES PER GRAPH
C
      DO 10200 J=1,3
      IF(DECIDE(IDX,J   )) NA(IDX)=NA(IDX)+1
      IF(DECIDE(IDX,J+3)) NB(IDX)=NB(IDX)+1
10200 CONTINUE
      BOTH(IDX)=.TRUE.
      IF(NA(IDX)+NB(IDX).GT.3) BOTH(IDX)=.FALSE.
      DO 10300 J=1,3
      IF(BOTH(IDX)) GO TO 10250
      CPERG(IDX,J   )=NA(IDX)
      CPERG(IDX,J+3)=NB(IDX)
      GO TO 10300
10250 CPERG(IDX,J   )=NA(IDX)+NB(IDX)
      CPERG(IDX,J+3)=CPERG(IDX,J)
10300 CONTINUE
10500 CONTINUE
10600 CONTINUE
      XSCALE = 10.
      IF(VSWL) XSCALE = 6.18
C
C     READ TRANSMITTER-OFF DATA AND THE ASSOCIATED RANGE VALUES.
C
      READ(3)
      READ(3)
      READ(3)
      READ(3) TXOFF, NTXOFF
      READ(3) RANGE2, NR
C
C     REMOVE DATA FOR 510 M. <= RANGE <= 520 M.
C
      DO 20500 IF3 = 1, 6
```

```
      INEXT = 1
      N = NN((IFR - 1) / 2 + 1)
      XMTOWL = FREC(IFR) / 299.7925
      IF(.NOT. VSWL) XMTOWL = 1.
      DO 20400 I = 1, N
      IF(RANGE2(I,IFR) .LT. XLIM(IFR,1) / XMTOWL - 3.) GO TO 20400
      IF(RANGE2(I,IFR) .GT. XLIM(IFR,2) / XMTOWL - 3.) GO TO 20500
      IF(RANGE2(I, IFR) .GE. 510.) GO TO 20200
      RANGE2(INEXT,IFR) = (RANGE2(I,IFR) + 3.) * XMTOWL
      DO 20100 J = 1, 3
      TXOFF(INEXT,IFR,J) = TXOFF(I,IFR,J) + 135.
20100 CONTINUE
      GO TO 20350
20200 IF(RANGE2(I,IFR) .LE. 520.) GO TO 20400
      RANGE2(INEXT, IFR) =(RANGE2(I, IFR) + 3.) * XMTOWL
      DO 20300 J = 1, 3
      TXOFF(INEXT, IFR, J) = TXOFF(I, IFR, J) + 135.
20300 CONTINUE
20350 NN(IFR) = INEXT
      INEXT = INEXT + 1
20400 CONTINUE
20500 CONTINUE
C
C     INITIALIZE THE PLOTTER
C
      CALL PLINIT('DOGP.JCR.LUNAR  ')
      N=3[illegible]
C
C     READ THE LABEL RECORD
C
      CALL LUNIN2(DATA,        IDATA, &980, &990)
      WRITE(6, 3000) TYPE
      WRITE(6, 1000) (IDATA(I), I=1, 297)
C
C     INITIALIZE THE STACK BEFORE READING RANGE DATA
C
   10 IORG=1
      M=0
      L=0
   20 IF(IORG+N .GT. 12000) GO TO 970
      CALL LUNIN2(DATA(IORG), IDATA, &980, &990)
      IF(ITYPE(1) .LE. 5) GO TO 20
C
C
      IF(ITYPE(2) .EQ. 6) GO TO 60
C
C     ACCUMULATE RANGE BLOCKS
C
```

```
      IF(FIRST) IXX=IORG
      IORG=IORG+N
      M=M+1
      IF(.NOT. LAST) GO TO 20
      XMTOWL=FREQ(ITYPE(1)-5)/299.7925
      IF(.NOT. VSWL) XMTOWL = 1.
      NPTSIN=M*N
      NXBGAP=0
      NXAGAP=0
C
C     FIND RANGE VALUES TO BE OMITTED
C
      DO 45 I=1,NPTSIN
      IF(DATA(I).GE.510.0) GO TO 50
      NXBGAP=NXBGAP+1
      DATA(I)=XMTOWL*(DATA(I)+3.0)
   45 CONTINUE
   50 INEXT=NXBGAP+1
      ISTART=INEXT
C
C     DELETE VALUES BY COMPRESSING THE ARRAY
C
      DO 55 I=ISTART,NPTSIN
      IF(DATA(I).LE.520.0) GO TO 55
      DATA(INEXT)=XMTOWL*(DATA(I)+3.0)
      NXAGAP=NXAGAP+1
      INEXT=INEXT+1
   55 CONTINUE
C
C     RESET THE ORIGIN FOR VCO DATA, AND ZERO THE COMPONENT COUNTER
C
      IORG=NXBGAP+NXAGAP+1
      NCOMP=0
      GO TO 20
C
C
C     ACCUMULATE VCO BLOCKS
C
C
   60 IF(FIRST) IXY=IORG
      IORG=IORG+N
      L=L+1
      IF(L .LT. M) GO TO 20
      IDX=ITYPE(1)-5
      NCOMP=NCOMP+1
      IF(.NOT.DECIDE(IDX,NCOMP)) GO TO 150
C
C     FIND THE LAST VALUE BEFORE THE GAP, AND ADJUST DB VALUES
```

```
C       TO A RELATIVE SCALE
C
        IY=IXY+NXBGAP-1
        DO 70 I=IXY,IY
        DATA(I)=DATA(I)+135.0
   70   CONTINUE
        INEXT=IXY+NXBGAP
        ISTART=IOPG-NXAGAP
        IEND=IOPG-1
C
C       COMPRESS THE VCO DATA, ADJUSTING
C       TO RELATIVE SCALE IN THE PROCESS
C
        DO 80 I=ISTART,IEND
        DATA(INEXT)=DATA(I)+135.0
        INEXT=INEXT+1
   80   CONTINUE
        NSTX=IXY
        NX=IXY
        NPLOT=IXY+NXBGAP+NXAGAP-1
        IY=IXY
C
C       OMIT VALUES OUTSIDE THE OUTER BOUNDS
C
        DO 90 IX=NX,NPLOT
        IF(DATA(IX).LT.XLIM(IDX,1).OR.DATA(IY).GT.YEAXS(IDX))
      .         NSTX = IX + 1
        IY=IY+1
   90   CONTINUE
        NSTY=IXY+NSTX-IXX
        IX=NPLOT
        NX=NPLOT
        DO 100 I=NSTX,NX
        IF(DATA(IX).GT.XLIM(IDX,2)) NPLOT=NPLOT-1
        IX=IX-1
  100   CONTINUE
        NPTS=NPLOT-NSTX+1
C
C       PLOT THE CURVE
C
        CALL DATPLT(TYPE,NOTES,CPERG(IEX,NCOMP),XSCALE,'.,  [illegible](NCOMP),
      .              DATA(NSTY),DATA(NSTX),NPTS,1,17.,'.1,[illegible](IDX),
      .              REF,ABSREF)
C
C       PLOT THE BASELINE.
C
        CALL BASEL(TXOFF(1, IDX, MOD(NCOMP - 1, 3) + 1),
      .            RANGE2(1, IDX), NW(IDX)                  )
```

```
C
C
C     IF THIS WAS THE SIXTH COMPONENT, GET A NEW RANGE ARRAY, OTHERWISE
C     GET A VCO ARRAY FOR THE NEXT COMPONENT
C
C
  150 IF(LAST) GO TO 10
      IORG=IXY
      L=0
      GO TO 20
C
C     STACK ARRAY TOO SMALL
C
  970 WRITE(6,7000)
      GO TO 999
C
C     NORMAL COMPLETION
C
  980 WRITE(6, 5000)
      GO TO 999
C
C     PREMATURE END OF FILE ("TAPE" INPUT)
C
  990 WRITE(6, 6000) TYPE
  999 CALL PLOTND
      RETURN
C
C
 1000 FORMAT(27(1X,11A4/))
 2000 FORMAT('0',2A8,' RECORD SKIPPED')
 3000 FORMAT('0',2A8,' RECORD READ')
 4000 FORMAT('1LABEL="',A8,'"'/ '0FREQ.= ',F5.1,' MHZ.'/
     .        '0FIRST POINT=',I4/ '0# OF POINTS=',I4/
     .        '0',10F10.3/99(1X,10F10.3/))
 5000 FORMAT('0NORMAL END OF JOB')
 6000 FORMAT('0END OF FILE OCCURRED WHILE ATTEMPTING TO READ ',
     .        2A8, ' RECORD')
 7000 FORMAT('-***  INSUFFICIENT SPACE ON STACK  ***')
C
C
      END
```

```
****************************************************************************
*                                                                          *
*                                  LUNIN                                   *
*                                                                          *
****************************************************************************

      SUBROUTINE LUNIN(DATA, IDATA, *, *)
C
C      READ LUNAR DATA TAPE
C
C      TAPE DATA ARRAYS
C
      REAL*4 DATA(1)
      INTEGER*4 IDATA(1)
C
C      CHARACTER DATA
C
       REAL*8 TYPE1(11)  / 'LABEL   ', 'MODE    ', 'TEMPERAT',
     .                     'TRANSMIT', 'CALIBRAT', ' 1 MHZ. ',
     .                     ' 2 MHZ. ', ' 4 MHZ. ', ' 8 MHZ. ',
     .                     '16 MHZ. ', '32 MHZ. '                  /
C
       REAL*8 TYPE2(6)   / '        ', 'HEP     ', 'TEL OFF ',
     .                     'TON     ', 'RANGE   ', 'V.   . C.'     /
C
       REAL*8 RUN, SITE, DIRECT, FORREV, TITLE(11), TYPE(2)
C
C      INDICES TO TYPE ARRAYS
C
       INTEGER*2  TIDX(3,17) /  1,  1,  1,    1,  2,  1,    1,  3,  2,
     .                          6,  4,  3,    6,  5,  4,    1,  [illegible],  5,
     .                          6,  6,  6,    1,  7,  5,    6,  7,  6,
     .                          2,  8,  5,   12,  8,  6,    4,  [illegible],  5,
     .                         24,  9,  6,    8, 10,  5,   [illegible], 10,  [illegible],
     .                         13, 11,  5,   78, 11,  [illegible]              /
C
      LOGICAL*4 FIRST, LAST
C
C
C      TYPE CODE RETURNED
C
       INTEGER*2 ITYPE(2)
C
C      COMMON BLOCK FOR RETURNED DATA
C
       COMMON /LUNDAT/ TITLE,  RUN,    SITE,   DIRECT, FORREV, TYPE,
     .                 ITYPE,  N,      FIRST,  LAST
```

```
C
C       RECORD COUNTERS
C
        INTEGER*4 IBLK /17/,  IREC /78/
C
C       BEGIN EXECUTABLE CODE
C
C       RESET RECORD COUNTER
C
       FIRST= FALSE.
        IREC = IREC + 1
        IF(IREC .LE. TIDX(1,IBLK))
     .           GO TO 10
C            ELSE
                 IREC = 1
                 FIRST=.TRUE.
                 IBLK = IBLK + 1
                 IF(IBLK .GT. 17)
     .               IBLK = 1
                 ITYPE(1) = TIDX(2, IBLK)
                 ITYPE(2) = TIDX(3, IBLK)
                 TYPE(1)  = TYPE1(ITYPE(1))
                 TYPE(2)  = TYPE2(ITYPE(2))
C
C       SELECT APPROPRIATE RECORD TYPE
C
   10   IF(IBLK - 2) 100,  200,  300
C
C            HEADER RECORD
C
  100        READ(4, 1000, END=990) RUN, SITE, DIRECT, FORREV, TITLE, N
             GO TO 999
C
C            MODE RECORD
C
  200        READ(4, 2000, END=995)           (IDATA(I), I=1, N)
             GO TO 999
C
C            ALL OTHER TYPES
C
  300        READ(4, 3000, END=995)           (DATA(I), I = 1, N)
             GO TO 999
C
C       END OF FILE CONDITIONS
C
C       PREDICTABLE
C         ( LABEL RECORD EXPECTED      )
C         ( => BEGINNING OF A NEW RUN )
```

C-3

```
C
  990  RETURN 1
C
C      UNEXPECTED
C        ( NON-LABEL RECORD EXPECTED )
C        ( => MIDDLE OF A RUN         )
C
  985  WRITE(6, 4000) TYPE, IREC
       RETURN 2
C
C      RETURN DATA
C
  999  LAST=.FALSE.
       IF(IREC .EQ. TIDX(1,IBLK)) LAST = .TRUE.
       RETURN
C
C
 1000  FORMAT(4A6, 10A8, A4, I6)
 2000  FORMAT(5 (200I6) )
 3000  FORMAT(5 (200F6.1) )
 4000  FORMAT('0***  END OF FILE FOUND WHILE ATTEMPTING TO READ "',
     .                    2A8, '" RECORD ',I3)
C
C
       END

*************************************************************************
*                                                                       *
*                                 LUNIN2                                *
*                                                                       *
*************************************************************************

      SUBROUTINE LUNIN2(DATA, IDATA, *, *)
C
C      READ LUNAR DATA TAPE
C
C      TAPE DATA ARRAYS
C
      REAL*4 DATA(1)
      INTEGER*4 IDATA(1)
C
C      CHARACTER DATA
C
       REAL*8 TYPE1(11)   / 'LABEL   ', 'MODE    ', 'TEMPERAT',
     .                      'TRANSMIT', 'CALIBRAT', ' 1 MHZ. ',
     .                      ' 2 MHZ. ', ' 4 MHZ. ', ' 8 MHZ. ',
```

```
     .                       '16 MHZ. ', '32 MHZ. '                /
C
      REAL*8 TYPE2(6)       / '            ', 'URE      ', 'TEP OFF ',
     .                       'ION      ', 'RANGE    ', 'V. C. O.'   /
C
      REAL*8 RUN, SITE, DIRECT, FORREV, TITLE(11), TYPE(2)
C
C     INDICES TO TYPE ARRAYS
C
      INTEGER*2  TIDX(3,17) /  1,  1,  1,    1,  2,  1,    1,  3,  2,
     .                         6,  4,  3,    6,  5,  4,    1,  6,  5,
     .                         6,  6,  6,    1,  7,  5,    6,  7,  6,
     .                         2,  8,  5,   12,  8,  6,    4,  9,  5,
     .                        24,  9,  6,    8, 10,  5,   48, 10,  6,
     .                        13, 11,  5,   78, 11,  6                /
C
     LOGICAL*4 FIRST, LAST
C
C
C     TYPE CODE RETURNED
C
      INTEGER*2 ITYPE(2)
C
C     COMMON BLOCK FOR RETURNED DATA
C
      COMMON /LUNDAT/ TITLE,  RUN,    SITE,   DIRECT, FORREV, TYPE,
     .                ITYPE,  N,      FIRST,  LAST
C
C     RECORD COUNTERS
C
      INTEGER*4 IBLK /17/,  IREC /78/
C
C     BEGIN EXECUTABLE CODE
C
C     RESET RECORD COUNTER
C
     FIRST=.FALSE.
      IREC = IREC + 1
      IF(IREC .LE. TIDX(1,IBLK))
     .        GO TO 10
C         ELSE
              IREC = 1
              FIRST=.TRUE.
              IBLK = IBLK + 1
              IF(IBLK .GT. 17)
     .            IBLK = 1
              ITYPE(1) = TIDX(2, IBLK)
              ITYPE(2) = TIDX(3, IBLK)
```

```
                 TYPE(1)  = TYPE1(ITYPE(1))
                 TYPE(2)  = TYPE2(ITYPE(2))
C
C        SELECT APPROPRIATE RECORD TYPE
C
   10    IF(IBLK - 2) 100,  200,  300
C
C            HEADER RECORD
C
  100       READ(4,1000,END=990) (IDATA(I), I=1, 297)
             GO TO 999
C
C            MODE RECORD
C
  200        READ(4, 2000, END=995)            (IDATA(I), I=1, N)
             GO TO 999
C
C            ALL OTHER TYPES
C
  300        READ(4, 3000, END=995)            (DATA(I), I = 1, [illegible])
             GO TO 999
C
C        END OF FILE CONDITIONS
C
C        PREDICTABLE
C          ( LABEL RECORD EXPECTED      )
C          ( => BEGINNING OF A NEW RUN )
C
  990    RETURN 1
C
C        UNEXPECTED
C          ( NON-LABEL RECORD EXPECTED )
C          ( => MIDDLE OF A RUN        )
C
  995    WRITE(6, 4000) TYPE, IREC
         RETURN 2
C
C        RETURN DATA
C
  999    LAST=.FALSE.
         IF(IREC .EQ. IIDX(1,IBLK)) LAST = .TRUE.
         RETURN
C
C
 1000 FORMAT(27(11A4))
 2000  FORMAT(5 (200I6) )
 3000  FORMAT(5 (200F6.1) )
 4000  FORMAT('0***  END OF FILE FOUND WHILE ATTEMPTING TO READ "',
```

```
     .              2A8, '" RECORD ',I3)
C
C
      END

************************************************************************
*                                                                      *
*                               LUNIN3                                 *
*                                                                      *
************************************************************************

      SUBROUTINE LUNIN3(DATA, IDATA, *, *)
C
C       READ LUNAR DATA TAPE
C
C       TAPE DATA ARRAYS
C
      REAL*4 DATA(1)
      INTEGER*4 IDATA(1)
C
C       CHARACTER DATA
C
        REAL*8 TYPE1(11)  / 'LABEL   ', 'MODE    ', 'TEMPERAT',
     .                      'TRANSMIT', 'CALIBRAT', ' 1 MHZ. ',
     .                      ' 2 MHZ. ', ' 4 MHZ. ', ' 8 MHZ. ',
     .                      '16 MHZ. ', '32 MHZ. '                /
C
        REAL*8 TYPE2(6)   / '        ', 'UPF     ', 'TEP OFF ',
     .                      'TON     ', 'RANGE   ', 'V. C. O.'   /
C
        REAL*8 RUN, SITE, DIRECT, FORREV, TITLE(11), TYPE(2)
C
C       INDICES TO TYPE ARRAYS
C
        INTEGER*2  TIDX(3,17) /  1,  1,  1,    1,  2,  1,    1,  3,  2,
     .                           6,  4,  3,    6,  5,  4,    1,  6,  5,
     .                           6,  6,  6,    1,  7,  5,    6,  7,  6,
     .                           2,  8,  5,   12,  8,  6,    4,  9,  5,
     .                          24,  9,  6,    8, 10,  5,   48, 10,  6,
     .                          13, 11,  5,   78, 11,  6                /
C
      LOGICAL*4 FIRST, LAST
C
C
C       TYPE CODE RETURNED
C
```

```
      INTEGER*2 ITYPE(2)
C
C     COMMON BLOCK FOR RETURNED DATA
C
      COMMON /LUNDAT/ TITLE,  RUN,    SITE,   DIRECT, FORREV, TYPL,
     .                ITYPE,  N,      FIRST,  LAST
C
C     RECORD COUNTERS
C
      INTEGER*4 IBLK /17/,  IREC /78/
C
C     BEGIN EXECUTABLE CODE
C
C     RESET RECORD COUNTER
C
     FIRST=.FALSE.
      IREC = IREC + 1
      IF(IREC .LE. TIDX(1,IBLK))
     .        GO TO 10
C         ELSE
              IREC = 1
              FIRST=.TRUE.
              IBLK = IBLK + 1
              IF(TIDX(3,IBLK) .EQ. 5)
     .           IBLK = IBLK + 1
              IF(IBLK .GT. 17)
     .           IBLK = 1
              ITYPE(1) = TIDX(2, IBLK)
              ITYPE(2) = TIDX(3, IBLK)
              TYPE(1)  = TYPE1(ITYPE(1))
              TYPE(2)  = TYPE2(ITYPE(2))
C
C     SELECT APPROPRIATE RECORD TYPE
C
   10 IF(IBLK - 2) 100,  200,  300
C
C         HEADER RECORD
C
  100     READ(2,1000,END=990) (IDATA(I), I=1, 297)
          GO TO 999
C
C         MODE RECORD
C
  200     READ(2,2000,END=995) (IDATA(I), I=1, ")
          GO TO 999
C
C         ALL OTHER TYPES
C
```

```
  300      READ(2,3000,END=995) ( DATA(I), I=1, N)
           GO TO 999
C
C       END OF FILE CONDITIONS
C
C       PREDICTABLE
C         ( LABEL RECORD EXPECTED      )
C         ( => BEGINNING OF A NEW RUN )
C
  990   RETURN 1
C
C       UNEXPECTED
C         ( NON-LABEL RECORD EXPECTED )
C         ( => MIDDLE OF A RUN         )
C
  995   WRITE(6, 4000) TYPE, IREC
        RETURN 2
C
C       RETURN DATA
C
  999   LAST=.FALSE.
        IF(IREC .EQ. TIDX(1,IBLK)) LAST = .TRUE.
        RETURN
C
C
 1000 FORMAT(27(11A4))
 2000  FORMAT(5 (200I6) )
 3000  FORMAT(5 (200F6.1) )
 4000  FORMAT('0***  END OF FILE FOUND WHILE ATTEMPTING TO READ "',
     .                   2A8, '" RECORD ',I3)
C
C
        END

********************************************************************************
*                                                                              *
*                                    ODCINT                                    *
*                                                                              *
********************************************************************************

      FUNCTION ODCINT(T)
C
      LOGICAL*1 FIRST / .TRUE. /
      REAL*4 TIME(500), ODC(500)
C
C
```

```
      IF(.NOT. FIRST) GO TO 100
        FIRST = .FALSE.
        N = 0
   20   READ(5, 1000, END = 30) TT, ORF, OLR
        N = N + 1
        TIME(N) = TT
        ODC(N) = 0.5 * (ORF + OLR)
        GO TO 20
C
   30 NN = N / 5
      IF(MOD(N, 5) .NE. 0) NN = NN + 1
      DO 50 I = 1, NN
        IF(MOD(I - 1, 50) .EQ. 0) WRITE(6, 2000)
        JJ = 4 * NN + I
        IF(JJ .GT. N) JJ = JJ - NN
        WRITE(6, 3000) (TIME(II), ODC(II), II = I, JJ, NN)
   50 CONTINUE
      WRITE(6, 4000)
C
  100 IF(T .LE. TIME(1)) GO TO 300
      IF(T .GE. TIME(N)) GO TO 400
      DO 200 I = 2, N
        IF(T .GE. TIME(I)) GO TO 200
        ODCINT = ODC(I - 1) + (ODC(I) - ODC(I - 1))
     .                      * (T - TIME(I - 1))
     .                      / (TIME(I) - TIME(I - 1))
        GO TO 500
  200 CONTINUE
C
C
  300 ODCINT = ODC(1)
      GO TO 500
C
C
  400 ODCINT = ODC(N)
C
C
  500 RETURN
C
C
 1000 FORMAT(3F10.0)
 2000 FORMAT('1NAVIGATION DATA ' / '0', 6X, 'TIME  OD. CNT.',
     .        4(14X, 'TIME  OD. CNT.') / '0 ')
 3000 FORMAT(1X, 2F10.1, 4(8X, 2F10.1))
 4000 FORMAT('- ')
C
      END
```

```
*******************************************************************
*                                                                 *
*                             PLINIT                              *
*                                                                 *
*******************************************************************

      SUBROUTINE PLINIT(NAME)
C     PLOTTER INITIALIZATION AND SETUP
      REAL*4 NAME(4),INIT/'.JCR'/
      LOGICAL ZIP/.TRUE./
      REAL*8 CODE/'PGS1410'/, SETUP(5)/5*'  '/,BLANK/'  '/
      DATA LIMIT,PLTLEN,PAGWID/90, 20., 11. /
      NAMELIST /PLTID/ INIT,CODE,SETUP,LIMIT,PLTLEN,PAGWID,ZIP
      NAMELIST /PLECHO/ LIMIT,PLTLEN,PAGWID,ZIP
      COMMON /PLTCOM/IT,L,II,IL
      IT=12
      LSET=0
      READ(5,PLTID)
      NAME(4)=INIT
      IF(SETUP(1).NE.BLANK) LSET=40
      IZIP=-1822
      IF (ZIP) IZIP=-IZIP
      CALL PLTSET(LIMIT,SETUP,LSET)
      CALL PLOTST (NAME,16,CODE,IZIP)
      WRITE (6,PLECHO)
      CALL PLTPAG(PAGWID)
      CALL PLTXMX(PLTLEN)
      RETURN
      END
      SUBROUTINE SYMBOL (X,Y,Z,IBCD,ANGLE,N)
C..***** NOTE *****..USE THIS SUBROUTINE ONLY IN PRODUCTION; REMOVE FOR
      CALL SYMBOL        (X,Y,Z,IBCD,ANGLE,N)
      RETURN
      END

*******************************************************************
*                                                                 *
*                             RTPLOT                              *
*                                                                 *
*******************************************************************

      SUBROUTINE RTPLOT
C
      REAL*4 SR(3088), ST(3088), VR(2565), VT(2565)
      COMMON / BLOCK / SR, ST, VR, VT, SCALE
```

```
C
C     INITIALIZE THE PLOTTER SOFTWARE AND LOCAL VARIABLES.
C     (TS IS REQUIRED LATER, FOR DRAWING THE TIME AXIS.)
C
      CALL PLINIT('QOGP.RANGES.     ')
      SC=1. / SCALE
      TORG = AINT(AMAX1(ST(3088), VT(2565)) * SC) + 1.
      T = AINT(AMIN1(ST(1), VT(1)) * SC)
      TS = T
C
C     DRAW THE RANGE AXIS.
C
      CALL PLOT(0., TORG - T, 3)
      R = AINT(AMAX1(SR(3088), VR(2565)) * SC) + 1.
      CALL SYMBOL(R, TORG - T, .07, 6, -90., -2)
      N = IFIX(R) - 1
      X = R
C
C     AND LABEL IT.
C
      DO 20 I = 1, N
        X = X - 1.
        CALL SYMBOL(X, TORG - T, .07, 13, 0., -1)
        CALL NUMBER(X - .14, TORG - T + .07, .07, X * SCALE, 0., -1)
   20 CONTINUE
      CALL SYMBOL(R * .6, TORG - T + .2, .14, 'RANGE (METRES)', 0., 14)
C
C     IDENTIFY THE TWO PLOTS: SEP IS A SOLID LINE; VLBI IS SOLID
C     AND MARKED BY A SYMBOL AT EVERY 100'TH POINT
C
      CALL PLOT(R - 2., TORG - T - 2.4, 3)
      CALL PLOT(R - 2., TORG - T - 3.4, 2)
      CALL SYMBOL(R - 2.07, TORG - T - 3.6, .14, 'SEP', -90., 3)
      CALL SYMBOL(R - 2.2, TORG - T - 2.4, .03, 0, 0., -1)
      CALL SYMBOL(R - 2.2, TORG - T - 2.9, .03, 0, 0., -2)
      CALL SYMBOL(R - 2.2, TORG - T - 3.4, .03, 0, 0., -2)
      CALL SYMBOL(R - 2.27, TORG - T - 3.6, .14, 'VLBI', -90., 4)
C
C     LABEL THE TIME AXIS.
C
      CALL SYMBOL(-.34, (TORG - T) * .5, .14, 'TIME (SECONDS)', -90.,14)
      N = IFIX(TORG - T) - 1
      DO 40 I = 1, N
        T = T + 1.
        CALL SYMBOL(0., TORG - T, .07, 13, 90., -1)
        CALL NUMBER(-.14, TORG - T + .14, .07, T * SCALE, -90., -1)
   40 CONTINUE
C
```

```
C     AND THEN DRAW IT.
C
      CALL SYMBOL(0., TORG - T - 1., .07, 6, 180., -1)
      CALL PLOT(0., TORG - TS, 2)
C
C     MOVE TO THE FIRST SEP POINT, AND THEN DRAW THE LINE.
C
      CALL PLOT(SR(1) * SC, TORG - ST(1) * SC, 3)
      DO 60 I = 2, 3088
        CALL PLOT(SR(I) * SC, TORG - ST(I) * SC, 2)
   60 CONTINUE
C
C     PLOT THE VLBI DATA WITH A SYMBOL AT EVERY 100'TH POINT.
C
      CALL SYMBOL(VR(1) * SC, TORG - VT(1) * SC, .03, 0, 0., -1)
      DO 80 I = 2, 2564
        IF(MOD(I, 100) .EQ. 1) GO TO 70
        CALL PLOT(VR(I) * SC, TORG - VT(I) * SC, 2)
        GO TO 80
   70   CALL SYMBOL(VR(I) * SC, TORG - VT(I) * SC, .03, 0, 0., -2)
   80 CONTINUE
      CALL SYMBOL(VR(2565) * SC, TORG - VT(2565) * SC, .03, 0, 0., -2)
      CALL PLOTND
      RETURN
      END

***************************************************************************
*                                                                         *
*                                SEPLOT                                   *
*                                                                         *
***************************************************************************

      SUBROUTINE SEPLOT(TITLE,NOTES,CPERG,XSCALE,YSCALE,COMP,H,P,MIN,
     *  INDEX,D,K1,LT1,K2,LT2,SITE,RUN,FREQ,REF,ABSREF)
C
C     PLOT OF EITHER THEORETICAL OR EXPERIMENTAL SEP DATA.
C     WRITTEN BY J.J.PROCTOR, SPRING 1973.  UNIVERSITY OF TORONTO.
C
C     INPUT:
C     TITLE = PLOT TITLE (16 DIGITS)
C     NOTES = ADDITIONAL NOTES (32 DIGITS)
C     CPERG = CURVES PER GRAPH (<= 6)
C     XSCALE = NUMBER OF INCHES PER 20.0 WL (6.18 IS STANDARD)
C     YSCALE = NUMBER OF INCHES PER CURVE FOR LINEAR PLOTS (< 5.),
C            = DB PER INCH FOR DB PLOTS (=> 5.)
C     COMP = COMPONENT LABEL - A 3-DIGIT INTEGER:
```

```
C               FIRST DIGIT       1=E, 2=H;
C               SECOND DIGIT      1=RHO, 2=PHI, 3=ZED;
C               THIRD DIGIT       1=BROADSIDE, 2=ENDFIRE
C       H = FIELD-STRENGTH ARRAY
C       R = RANGE ARRAY (IN WL)
C       NIN = DIMENSION OF H AND R
C       INDEX = INDEXING THROUGH H AND R ARRAYS (USUALLY =1)
C
        REAL*4 K1,K2,LT1,LT2
        INTEGER*4 TITLE(4),NOTES(8)
        INTEGER*4 COMTAB(7)/'E','H',73B,724,Z69,'PRD','END'/
        INTEGER*4 CTR/0/,GCTR/0/,CPERG,COMP
        LOGICAL*4 DATOLD,DATNEW
        DIMENSION H(NIN),R(NIN)
        INTEGER*4 LABELS(3)
        INTEGER*4 PXA(3) / 'RHO', 'PHI', 'ZED' /
        RETURN
C
C
C..ENTRY POINT FOR THEORY CURVES
        ENTRY THEPLT(TITLE,NOTES,CPERG,XSCALE,YSCALE,COMP,H,R,NIN,INDEX,
      *  D,K1,LT1,K2,LT2)
        DATNEW=.FALSE.
        GO TO 2
C
C
C..ENTRY POINT FOR DATA TYPE CURVES
        ENTRY DATPLT(TITLE,NOTES,CPERG,XSCALE,YSCALE,COMP,H,R,NIN,INDEX,
      *         SITE,RUN,FREQ,REF,ABSREF)
        DATNEW=.TRUE.
C
C
    2   CTR=CTR+1
C
C..IF THIS IS THE FIRST CURVE ON THE GRAPH, PLOT GRAPH OUTLINE
        IF(CTR.EQ.1) GO TO 10
C
C..TEST FOR A FULL GRAPH
        IF(CTR.LE.CPERG) GO TO 70
C
C..FULL-GRAPH LOGIC
        CTR=1
        CALL PLOT(XXX+4.20,0.,-3)
C
C..GRAPH OUTLINE
   10   GCTR=GCTR+1
C
C..SET RANGE AND FIELD STRENGTH ARRAY DIMENSION ON INDEX BOUNDARY
```

```
      N=((NIN-1)/INDEX)*INDEX+1
      NFIRST=N
C
C..CONVERT X-AXIS SCALE TO INCHES-PER-WAVELENGTH
C  OR INCHES-PER-METRE
      XSCALD = XSCALE / 20.
      IF(XSCALE .GT. 10.)XSCALD = XSCALE / 1000.
C
C..DISTANCE BETWEEN 4 WL SEGMENTS
      XSPACE=4.*XSCALD
      IF(XSCALE .GT. 10.)XSPACE = 200. * XSCALD
C
C..ROUND DOWN FIRST RANGE POINT TO DETERMINE GRAPH ORIGIN
      IRIST=0
C
C..NUMBER OF 4 WL SEGMENTS IN THE RANGE VALUES
      IF(XSCALE .LT. 10.) NUMR=(IFIX(R(N))-IRIST)/4+1
      IF(XSCALE .GE. 10.) NUMR=(IFIX(R(N))-IRIST)/200+1
C
C..LAST SEGMENT NUMBER + ONE
      ILAST=NUMR+1
C
C..HALFWAY POINT IN THE NUMBER OF SEGMENTS
      IHALF=ILAST/2
C
C..CO-ORDINATE OF Y-AXIS LABEL
      XLABEL=2*NUMR*XSCALD-.5
      IF(XSCALE .GT. 10) XLABEL = 100 * NUMR * XSCALD - .5
C
C..CO-ORDINATE OF GRAPH TITLE
      XTITLE=AMAX1(.5,XLABEL-1.3)
C
C..DRAW Y-AXIS
      CALL SYMBOL(0.,10.,.1,6,0.,-2)
C
C..TITLE GRAPH AND PLOT ANY NOTES (EXPLANATION, ETC)
      CALL NUMBER(XTITLE,10.,.15,FREQ,0.,1)
      CALL SYMBOL(999.,999.,.15,5H MHZ.,0.,5)
      CALL SYMBOL(999.,999.,.15,14H     APOLLO 17,0.,14)
      CALL SYMBOL(XTITLE,9.8,.07,NOTES,0.,32)
C
C..SET *XXX* VARIABLE WHICH IS RIGHTMOST POSITION OF X-AXIS SEGMENTS
      XXX=NUMR*XSPACE
C
C..DRAW X-AXIS  (BACKWARDS)
      XAX=XXX+XSPACE
      CALL SYMBOL(XAX,0.,.1,6,270.,-1)
      DO 35 I=1,ILAST
```

```
      XAX=XAX-XSPACE
   35 CALL SYMBOL(XAX,0.,.05,13,0.,-2)
C
C..NUMBER FIRST HALF OF X-AXIS
      DNUM=IFIST-3.9
      IF(XSCALE .GE. 10.) DNUM = -199.9
      XNUM=-XSPACE-.05
      DO 36 I=1,IHALF
      IF(XSCALE .LT. 10.) DNUM = DNUM + 4.
      IF(XSCALE .GE. 10.) DNUM = DNUM + 200.
      XNUM=XNUM+XSPACE
   36 CALL NUMBER(XNUM,-.15,.07,DNUM,0.,-1)
C
C..LABEL X-AXIS
      CALL SYMBOL(XLABEL,-.3,.1,6HRANGE ,0.,6)
      IF(XSCALE .GE. 10.) CALL SYMBOL(999., 999., .1, 2HM., 0., 2)
      IF(XSCALE .LT. 10.) CALL SYMBOL(999.,999.,.14,41,0.,-1)
C
C..NUMBER SECOND HALF OF X-AXIS
      DO 37 I=IHALF,NUMB
      IF(XSCALE .LT. 10.) DNUM = DNUM + 4.
      IF(XSCALE .GE. 10.) DNUM = DNUM + 200.
      XNUM=XNUM+XSPACE
   37 CALL NUMBER(XNUM,-.15,.07,DNUM,0.,-1)
C
C..LABEL Y-AXIS
      IF(YSCALE.GT.5.) GO TO 38
      CALL SYMBOL(-.15,4.5,.1,6HLINEAR,90.,6)
      GO TO 39
   38 CALL NUMBER(-.15,4.5,.1,YSCALE,90.,-1)
      CALL SYMBOL(999.,999.,.1,3H DB,90.,3)
      CALL SYMBOL(-.1,4.25,.06,13,90.,-1)
      CALL SYMBOL(-.1,4.27,.04,6,180.,-1)
      CALL SYMBOL(-.1,5.23,.04,6,0.,-2)
      CALL SYMBOL(-.1,5.25,.06,13,90.,-1)
      CALL SYMBOL(-.15,6.0,.1,7HREF. AT,90.,7)
      CALL NUMBER(-.15,6.8,.1,ABSREF,90.,1)
      CALL SYMBOL(999.,999.,.1,4H DBM,90.,4)
C
C..END OF GRAPH VARIABLE SET-UPS
   39 DATOLD=.NOT.DATNEW
      CPERG=MINO(6,CPERG)
      SHIFT=6./(CPERG-1.)
      ORIGIN=6.+SHIFT
      GO TO 71
C
C
C..ENTRY POINT FOR PLOTTING A CURVE
```

```
   70 N=((NIN-1)/INDEX)*INDEX+1
      NFIRST=N
C
C..SET THE ORIGIN FOR THIS CURVE
   71 ORIGIN=ORIGIN-SHIFT
      IC3=5+MOD(COMP,10)
      IC2=2+MOD(COMP/10,10)
      IC1=COMP/100
      LABELS(1) = COMTAB(IC1)
      LABELS(2) = BXA(IC2 - 2)
      LABELS(3) = COMTAB(IC3)
      WRITE(6, 90050) FREQ, LABELS
90050 FORMAT('0PLOTTING ', F6.1, 3X, A2, 2A4)
C
C..FIND MAXIMUM AND MINIMUM FIELD STRENGTH VALUES
      YMAX=H(1)
      YMIN=YMAX
      DO 97 I=1,N,INDEX
      YMIN=AMIN1(H(I),YMIN)
   97 YMAX=AMAX1(H(I),YMAX)
C
C..TEST FOR LINEAR PLOTS
      IF(YSCALE.LT.5.) GO TO 700
C
C..CONVERT DB VALUES TO INCHES AND ZERO LOW VALUES
      DO 98 I=1,N,INDEX
   98 H(I) = H(I) / YSCALE
      GO TO 99
C
C..CONVERT LINEAR VALUES TO INCHES
  700 HDELTA=YSCALE/YMAX
      DO 701 I=1,N,INDEX
  701 H(I)=H(I)*HDELTA
C
C..PLOT THE CURVE
   99 CALL PLOT(-.05, REF / YSCALE + ORIGIN, 3)
      CALL PLOT( .05, REF / YSCALE + ORIGIN, 2)
      CALL PLOT((R(1)-IRIST)*XSCALD, H(1)+ORIGIN, 3)
      NST=1+INDEX
      DO 800 I=NST,N,INDEX
  800 CALL PLOT((R(I)-IRIST)*XSCALD, H(I)+ORIGIN, 2)
C
C..IF FIRST CURVE ON GRAPH, PLOT *COMP* AND *MAX* HEADINGS
      YYNO=H(N)+ORIGIN
      YYHD=YYNO+.15
      IF(CTR.EQ.1) CALL SYMBOL(XXX+.035,YYHD,.070,4HCOMP,0.,4)
C
C..PLOT COMPONENT AND MAXIMUM
```

```
      CALL SYMBOL(XXX+.035,YYNO,.070,COMTAB(IC1),0.,1)
      CALL SYMBOL(999.,999.,.070,47,0.,-1)
      CALL SYMBOL(999.,999.,.070,COMTAB(IC2),0.,-1)
      CALL SYMBOL(999.,999.,.070,46,0.,-1)
      CALLSYMBOL(999.,999.,.070,COMTAB(IC3),0.,3)
C
C
C..CURVE LABELLING TESTS FOLLOW
      IF (DATNEW) GO TO 40
C
C..THEORY CURVE
      IF (.NOT.DATOLD) GO TO 41
C
C..THEORY HEADINGS
      CALL SYMBOL(XXX+1.015,YYHD,.07,32HDEPTH   K1      LT1      K2    LT2
     *,0.,32)
C
C..THEORY VARIABLES
   41 CALL NUMBER(XXX+1.015,YYNO,.070,D  ,0.,3)
      CALL NUMBER(XXX+1.505,YYNO,.070,K1 ,0.,2)
      CALL NUMBER(XXX+1.925,YYNO,.070,LT1,0.,4)
      CALL NUMBER(XXX+2.555,YYNO,.070,K2 ,0.,2)
      CALL NUMBER(XXX+2.975,YYNO,.070,LT2,0.,3)
C
   40 CONTINUE
  999 DATOLD=DATNEW
      RETURN
      ENTRY BASEL(H, R, NIN)
      DO 90100 I = 1, NIN
      CALL SYMBOL((R(I)-TRIST)*XSCALD, H(I)/YSCALE+ORIGIN,.07,11,0.,-1)
90100 CONTINUE
      RETURN
      END
```

```
******************************************************************
*                                                                *
*                             STOPT                              *
*                                                                *
******************************************************************
```

```
      LOGICAL FUNCTION STOPT*1 (I, IFP)
C
C            RETURNS .TRUE. IF THE LRV WAS STOPPED DURING THE EP-4
C            TURN, .FALSE. OTHERWISE.  (THIS DECISION IS BASED ON THE
C            VALUES PLACED IN THE ARRAY "H" BY THE CALLING ROUTINE.)
C
```

```
C
      INTEGER*2 B(6)
      INTEGER*2 C(6) / 33, 33, 20, 20, 7, 7 /
C
      COMMON /BOUNDS/ B
C
C
      J = 13 * I + C(IPR)
      STOPT = (         J .GT. B(2)
     .            .AND. J .LT. B(3) )
     .    .OR. (        J .GT. B(4)
     .            .AND. J .LT. B(5) )
      RETURN
      END
```

```
**************************************************************************
*                                                                        *
*                               TXOSTAT                                  *
*                                                                        *
**************************************************************************
```

```
C
C
C              PROGRAM TO COMPARE TRANSMITTER-OFF DATA
C              WITH APPROXIMATE LRV SPEED.
C
C     ONE INPUT CARD IS REQUIRED, CONTAINING SIX INTEGER VALUES IN
C     FORMAT 6I5; THESE VALUES SHOULD BE THE SAME AS THE "BOUNDS" FOR
C     THE 32.1 MHZ. INPUT TO ANTENNAO.
C
C
      REAL*4 SST(3), SMO(3), SSTSO(3), SMOSQ(3)
      REAL*4 RANGE(140, 6), SPEED(140, 6), TXOFF(140, 6, 3)
      REAL*4 TXOFS(140, 6, 3), SMOFS(3), SSTFS(3)
      REAL*4 SSTFSQ(3), SMOFSQ(3)
      INTEGER*4 NTXOFF(6, 3), NR(3), INDEX(140)
      INTEGER*2 BOUND(6)
      LOGICAL*1 STOPT
      LOGICAL*1 SWITCH
C
      COMMON /BOUNDS/ BOUND
C
C
      CALL PLOTST('OOGP.JCR.TXOFF  ', 16, 'FGS1410 ')
C
      READ(3)
```

```
      READ(3)
      READ(3)
      READ(3) TXOFF, NTXOFF
      READ(3) RANGE, NR
      READ(3) SPEED
      READ(5, 3000) BOUND
C
C             LOOP THROUGH FREQUENCIES
C
      DO 100 IFR = 1, 6
        IFREQ = 2 ** (IFR - 1)
        I = 0
        N = 0
        SWITCH = .FALSE.
        NST = 0
        NMO = 0
        DO 5 J = 1, 3
          SST(J) = 0.
          SMO(J) = 0.
          SMOFS(J) = 0.
          SSTFS(J) = 0.
          SSTSQ(J) = 0.
          SMOSQ(J) = 0.
          SSTFSQ(J) = 0.
          SMOFSQ(J) = 0.
    5   CONTINUE
   10   WRITE(6, 1000) IFREQ
        LIN = 0
   20   I   = I + 1
        IF(RANGE(I, IFR) .GT. 1667.) GO TO 50
        N = N + 1
        IF(RANGE(I, IFR) .EQ. 513.9 .AND. .NOT. STOPT(I, IFR)) GO TO 20
        DO 23 J = 1, 3
          TXOFS(I, IFR, J) = 10. ** (.05 * TXOFF(I, IFR, J))
   23   CONTINUE
        LIN = LIN + 1
        WRITE(6, 2000) RANGE(I, IFR), SPEED(I, IFR),
     .                 (TXOFF(I, IFR, J), J = 1, 3),
     .                 (TXOFS(I, IFR, J), J = 1, 3)
        IF(SPEED(I, IFR) .EQ. 0.) GO TO 30
        NMO = NMO + 1
        DO 25 J = 1, 3
          SMO(J) = SMO(J) + TXOFF(I, IFR, J)
          SMOFS(J) = SMOFS(J) + TXOFS(I, IFR, J)
   25   CONTINUE
        GO TO 40
   30   NST = NST + 1
        DO 35 J = 1, 3
```

```
          SST(J) = SST(J) + TXOFF(I, IFR, J)
          SSTFS(J) = SSTFS(J) + TXOFS(I, IFR, J)
   35   CONTINUE
   40   IF(LIN .LT. 55) GO TO 20
        GO TO 10
C
C             DO A SORT ON THE APPROPRIATE SEGMENT OF THE SPEED ARRAY.
C             SUBROUTINE BUBBLE RETURNS  THE VECTOR INDEX CONTAINING
C             INDICES TO THE DATA ARRAYS SUCH THAT IF I < J, THEN
C             SPEED(INDEX(I), IFR) < SPEED(INDEX(J), IFR)
C
   50   CALL BUBBLE(SPEED(1, IFR), INDEX, N)
C
C       LIST THE SPEED VALUES IN ASCENDING ORDER WITH THE CORRESPONDING
C             TXOFF VALUES FOR EACH ANTENNA.
C
        LIN = 0
        DO 60 J = 1, 3
          SMO(J) = SMO(J) / NMO
          SST(J) = SST(J) / NST
          SMOFS(J) = SMOFS(J) / NMO
          SSTFS(J) = SSTFS(J) / NST
   60   CONTINUE
        DO 80 I = 1, N
          IF(LIN .EQ. 0) WRITE(6, 1010) IFREQ
          IX = INDEX(I)
          IF(RANGE(IX, IFR) .NE. 513.9 .OR. STOPT(IX, IFR)) GO TO 70
            SPEED(IX, IFR) = -1.
            GO TO 80
   70     IF(SWITCH .OR. SPEED(IX, IFR) .EQ. 0.) GO TO 75
          DO 72 J = 1, 3
            SSTSQ(J) = SQRT(SSTSQ(J) / (NST - 1))
            SSTFSQ(J) = SQRT(SSTFSQ(J) / (NST - 1))
   72     CONTINUE
          WRITE(6, 4000) NST, SST, SSTFS, SSTSQ, SSTFSQ
          LIN = LIN + 5
          SWITCH = .TRUE.
   75     CONTINUE
          DO 79 J = 1, 3
            IF(SWITCH) GO TO 77
            A = TXOFF(IX, IFR, J) - SST(J)
            B = TXOFS(IX, IFR, J) - SSTFS(J)
            SSTSQ(J) = SSTSQ(J) + A * A
            SSTFSQ(J) = SSTFSQ(J) + B * B
            GO TO 79
   77       A = TXOFF(IX, IFR, J) - SMO(J)
            B = TXOFS(IX, IFR, J) - SMOFS(J)
            SMOSQ(J) = SMOSQ(J) + A * A
```

```
            SMOFSQ(J) = SMOFSQ(J) + B * B
   70     CONTINUE
          WRITE(6, 2010) SPEED(IX, IFR),
     .                   (TXOFF(IX, IFR, J), J = 1, 3),
     .                   (TXOFS(IX, IFR, J), J = 1, 3)
          LIN = MOD(LIN + 1, 50)
   80   CONTINUE
        DO 85 J = 1, 3
          SMOSQ(J) = SQRT(SMOSQ(J) / (NMO - 1))
          SMOFSQ(J) = SQRT(SMOFSQ(J) / (NMO - 1))
   85   CONTINUE
        WRITE(6, 4010) NMO, SMO, SMOFS, SMOSQ, SMOFSQ
C
C             PLOT TXOFF VS. SPEED
C
        CALL TXPLOT(SPEED(1, IFR), TXOFF(1, IFR, 1), TXOFF(1, IFR, 2),
     .                             TXOFF(1, IFR, 3), N, IFREQ          )
C
  100 CONTINUE
C
      CALL PLOTND
C
      RETURN
C
C
 1000 FORMAT('1',I4,' MHZ. -- LRV SPEED AND TXOFF DATA ORDERED BY RANGE'
     .       / 45X, 'TXOFF DB', 25X, 'TXOFF FIELD STRENGTH' /
     .       6X, 'RANGE', 10X, 'SPEED', 12X, 2(     'X', 11X, 'Y', 11X,
     .       'Z', 14X) / 2X )
 1010 FORMAT('1', I4, ' MHZ. -- TXOFF DATA ORDERED BY LRV SPEED' /
     .       47X, 'TXOFF DB', 25X, 'TXOFF FIELD STRENGTH' /
     .       24X, 'SPEED', 12X, 2('X', 11X, 'Y', 11X, 'Z', 14X ) / 2X )
 2000 FORMAT(1X, F10.1, 5X, F10.4, 3X, 3(2X, F10.1), 3X, 3(2X, 1PE10.3))
 2010 FORMAT(19X, F10.4, 3X, 3(2X, F10.1), 3X, 3(2X, 1PE10.3))
 3000 FORMAT(6I5)
 4000 FORMAT('0', I4, ' RECORDS WITH LRV STOPPED' /
     .        6X, 'MEAN VALUES', 15X, 3(2X, F10.2), 3X, 3(2X, 1PE10.3) /
     .        6X, 'STANDARD DEVIATIONS', 7X, 3(2X, 0PF10.5),
     .        3X, 3(2X, 1PE10.3) / 2X )
 4010 FORMAT('0', I4, ' RECORDS WITH LRV MOVING ' /
     .        6X, 'MEAN VALUES', 15X, 3(2X, F10.2), 3X, 3(2X, 1PE10.3) /
     .        6X, 'STANDARD DEVIATIONS', 7X, 3(2X, 0PF10.5),
     .        3X, 3(2X, 1PE10.3) )
C
      END
```

```
***************************************************************************
*                                                                         *
*                                TXPLOT                                   *
*                                                                         *
***************************************************************************

      SUBROUTINE TXPLOT(S, TXX, TXY, TXZ, N, IER)
C
      REAL*4 S(N), TXX(N), TXY(N), TXZ(N)
      REAL*4 TXORG(3) / 0., 3., 6. /
      REAL*4 SCALE
      SCALE(ARG) = .1 * (ARG + 135.)
C
C
C              DRAW THE SPEED AXES
C
      X = 6.
      DO 5 I = 1, 3
        CALL PLOT(0., TXORG(I), 3)
        CALL SYMBOL(X, TXORG(I), .07, 6, -90., -2)
    5 CONTINUE
C
C              AND LABEL THEM
C
      DO 10 I = 1, 5
        X = X - 1.
        DO 8 J = 1, 3
          CALL SYMBOL(X, TXORG(J), .07, 13, 0., -1)
    8   CONTINUE
        CALL NUMBER(X - .105, -.2, .07, X, 0., 1)
   10 CONTINUE
      CALL SYMBOL(1.5, -.5, .14, 22HLRV SPEED  (M. / SEC.), 0., 22)
C
C              DRAW THE DB AXES
C
      DO 30 J = 1, 3
        CALL PLOT(0., TXORG(J), 3)
        CALL SYMBOL(0., TXORG(J) + 2.5, .07, 6, 0., -2)
C
C              AND LABEL THEM
C
        DO 20 I = 1, 3
          CALL SYMBOL(0., TXORG(J) + 3 - I, .07, 13, 90., -1)
          CALL NUMBER(-.13, TXORG(J) + 2.79 - I, .07, -105. - 10. * I,
     *                90., 1)
   20   CONTINUE
   30 CONTINUE
```

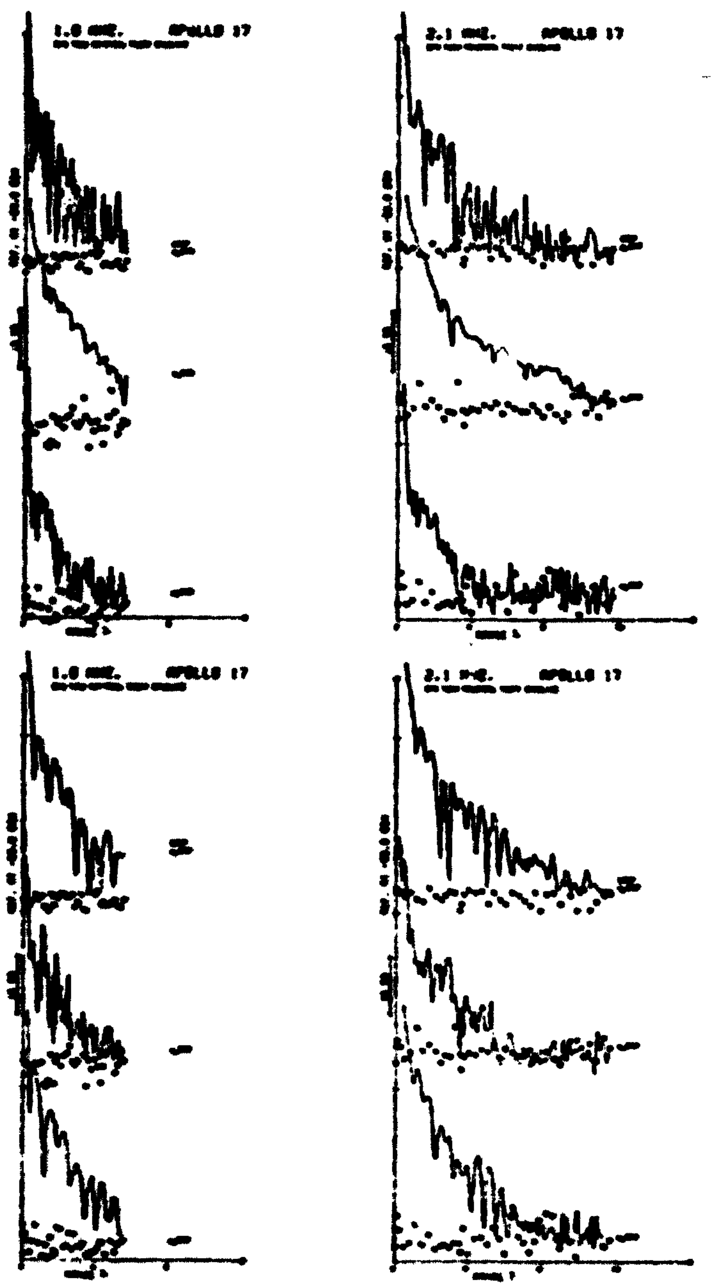
```
      CALL SYMBOL(-.36, 3., .14, 22HTRANSMITTER-OFF (DEM.), 90., 22)
C
C               LABEL THE GRAPH
C
      CALL NUMBER(3., 9.16, .14, FLOAT(IFR), 0., -1)
      CALL SYMBOL(999., 999., .14, 18H MHZ.    APOLLO 17, 0., 18)
C
C               PLOT THE DATA POINTS
C
      DO 40 I = 1, N
        IF(S(I) .LT. 0.) GO TO 40
          CALL SYMBOL(S(I), SCALE(TXX(I)) +  6., .07, 4, 0., -1)
          CALL SYMBOL(S(I), SCALE(TXY(I)) +  3., .07, 9, 0., -1)
          CALL SYMBOL(S(I), SCALE(TXZ(I))      , .07, 8, 0., -1)
   40 CONTINUE
C
C               MOVE ON TO BEGIN A POSSIBLE NEW PLOT
C
      CALL PLOT(8.5, 0., -3)
C
      RETURN
C
      END

******************************************************************************
*                                                                            *
*                                  VLPIRT                                    *
*                                                                            *
******************************************************************************

C
C     PROGRAM TO COMPARE VLBI DATA WITH SEP NAVIGATION DATA.  VLBI DATA
C     MAY BE EITHER HIGH- OR LOW-SPEED; SEP DATA ARE OBTAINED FROM THE
C     16 MHZ. RANGE ARRAY ON FILE SCI2, AND CORRESPONDING TIMES ARE
C     GENERATED INTERNALLY.
C
C     ONE NAMELIST (CNTL) CONTROL CARD IS REQUIRED:
C
C     TO     - TIME (GM) OF FIRST 16 MHZ. RANGE POINT.
C     RO     - DISTANCE OF FIRST 16 MHZ. RANGE POINT FROM
C              SEP TRANSMITTER.
C     STAT   - (BOOLEAN) OUTPUT COMPARISON STATISTICS.
C     PLOT   - (BOOLEAN) PLOT RANGES FOR VLBI AND SEP VS. TIME.
C     SCALE  - INDICATES NUMBER OF METRES AND 100-WAVELENGTH INTERVALS
C              PER INCH ON THE PLOT; NOT REQUIRED IF PLOT = FALSE.
C
```

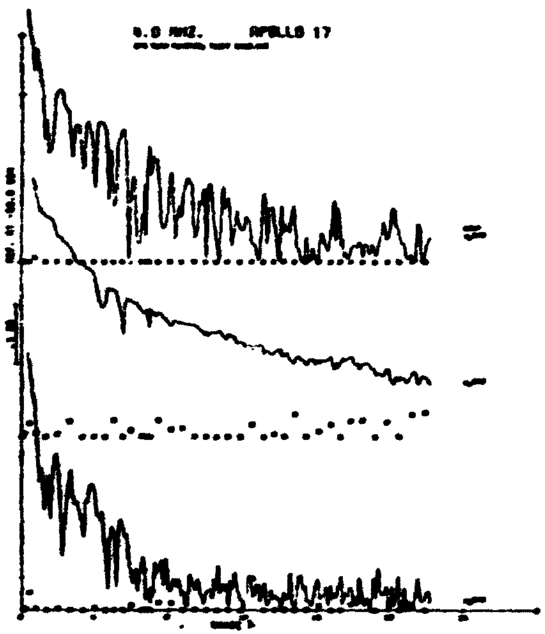
```
C     ALSO A PLTID NAMELIST CARD IS REQUIRED BY PLINIT (IF PLOT = TRUE).
C
      INTEGER*4 H, M, S
      REAL*4 X(5), Y(5)
      REAL*4 SR(3088), ST(3088), VR(2565), VT(2565)
      REAL*4 TO / 1452. /, RO / 0. /
      LOGICAL*1 STAT / .TRUE. /, PLOT / .FALSE. /
      COMMON / BLOCK / SR, ST, VR, VT, SCALE
      NAMELIST / CNTL / TO , RO, STAT, PLOT, SCALE
C
C     SET AND READ CONTROL PARAMETERS, AND SKIP OVER UNWANTED SEP DATA.
C
      SCALE = 500.
      READ(5, CNTL)
      DO 10 I = 1, 71
        READ(3, 1000)
   10 CONTINUE
C
C     READ 16 MHZ. RANGE DATA.
C
      K = 1
      L = 386
      DO 20 I = 1, 8
        READ(3, 2000) (SR(J), J = K, L)
        K = K + 386
        L = L + 386
   20 CONTINUE
C
C     ADJUST SEP RANGE AND DB DATA USING SUPPLIED PARAMETERS
C
      DO 30 I = 1, 3088
        ST(I) = .81 * (I - 1) + TO
        SR(I) = SR(I) + RO
   30 CONTINUE
C
C     READ VLBI DATA IN GROUPS OF ONE TIME AND FIVE X-Y PAIRS, CONVERT
C     TIME TO SECONDS AND X-Y PAIRS TO RANGES, AND STORE.
C
      DO 50 I = 1, 2565, 5
        READ(4, 100) H, M, S, (X(J), Y(J), J = 1, 5)
        T = S + 60 * (M + 60 * H)
        DO 40 J = 1, 5
          II = J - 1
          JJ = I + II
          VT(JJ) = T + II
          VR(JJ) = SQRT(X(J) * X(J) + Y(J) * Y(J))
   40   CONTINUE
   50 CONTINUE
```

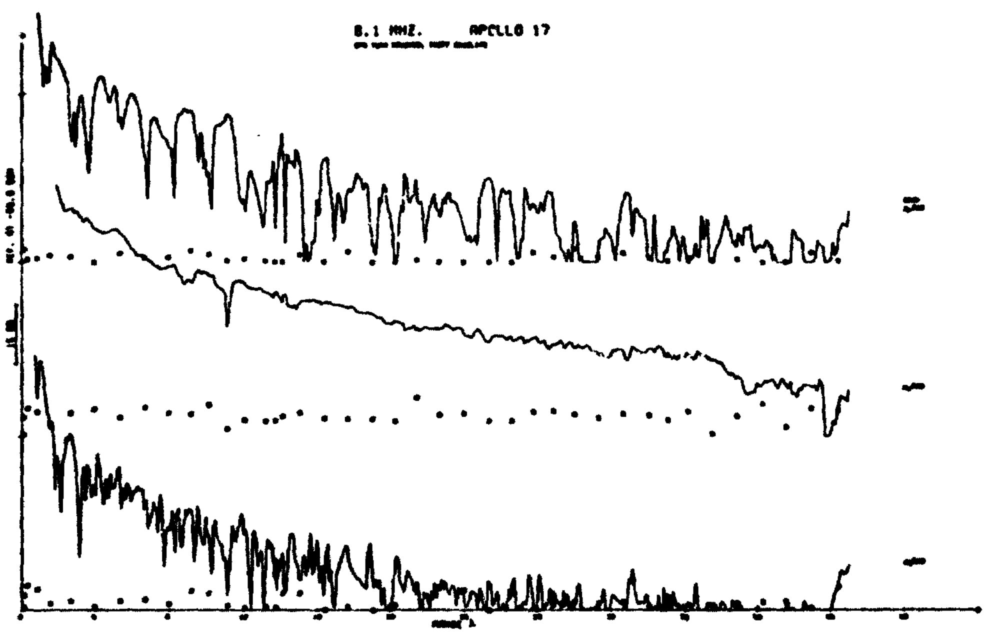
```
C
      IF(PLOT) CALL RTPLOT
      IF(.NOT. STAT) GO TO 99
      JST = 1
C
C     SKIP ALL VLBI TIMES WHICH ARE LESS THAN THE FIRST SEP TIME.
C
      DO 60 J = 1, 2565
        IF(VT(J) .GE. ST(1)) GO TO 65
          JST = JST + 1
   60 CONTINUE
C
   65 LIN = 0
      E = 0.
      S = 0.
      I = 1
C
C     FOR EACH VLBI TIME-RANGE PAIR
C
      DO 90 J = 1, 2565
C
C       FIND THE PAIR OF SEP TIMES WHICH BRACKET THE VLBI TIME.
C
        DO 70 K = I, 3088
          IF(VT(J) .LT. ST(K)) GO TO 80
   70   CONTINUE
        GO TO 95
   80   I = K
        II = I - 1
C
C       COMPUTE AN INTERPOLATED SEP RANGE, AND THE DIFFERENCE BETWEEN
C       IT AND THE VLBI RANGE, AND INCREMENT THE SUM OF DIFFERENCES AND
C       THE SUM OF SQUARES OF DIFFERENCES.
C
        R = SR(II) + (SR(I) - SR(II)) * (VT(J) - ST(II))
     *                                  / (ST(I) - ST(II))
        D = VR(J) - R
        E = E + D
        S = S + D * D
        IF(MOD(LIN, 50) .EQ. 0) WRITE(6, 200)
        WRITE(6, 300) VT(J), VR(J), R, D
        LIN = LIN + 1
   90 CONTINUE
C
C     COMPUTE THE MEAN AND STANDARD DEVIATION.
C
   95 E = E / LIN
      S = SQRT(S / FLOAT(LIN - 1))
```

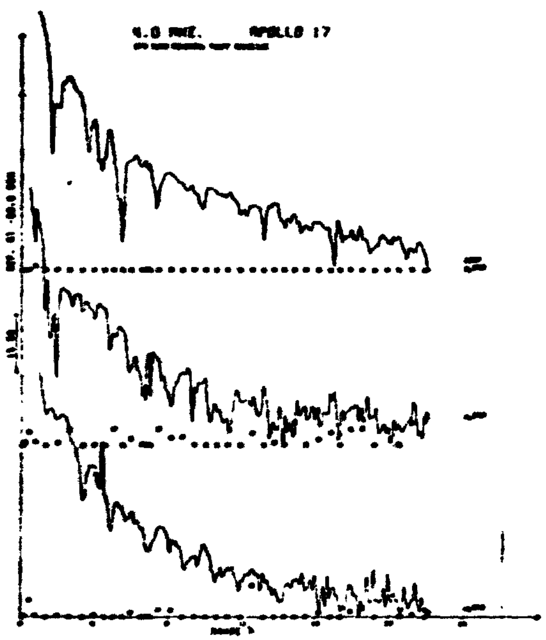
```
      WRITE(6, 400) E, S
C
   99 RETURN
C
  100 FORMAT(3(I2,1X), 1X, 10F5.0)
  200 FORMAT('1' / '-', 33X, 'INTERPOLATED', 5X, 'DIFFERENCE' /
     .                7X, 'VLBI TIME', 5X, 'VLBI RANGE',
     .                6X, 'SEP RANGE', 7X, 'IN RANGE' / 1X )
  300 FORMAT(10X, F6.0, 9X, F6.0, 8X, F7.2, 8X, F7.2)
  400 FORMAT('-SUM(DIFF.) / N = ', F7.2 /
     .       '0SQRT(SUM(DIFF. ** 2) / (N - 1)) = ', F7.3)
 1000 FORMAT(200A6, 186A6)
 2000 FORMAT(200F6.1, 186F6.1)
C
      END
```

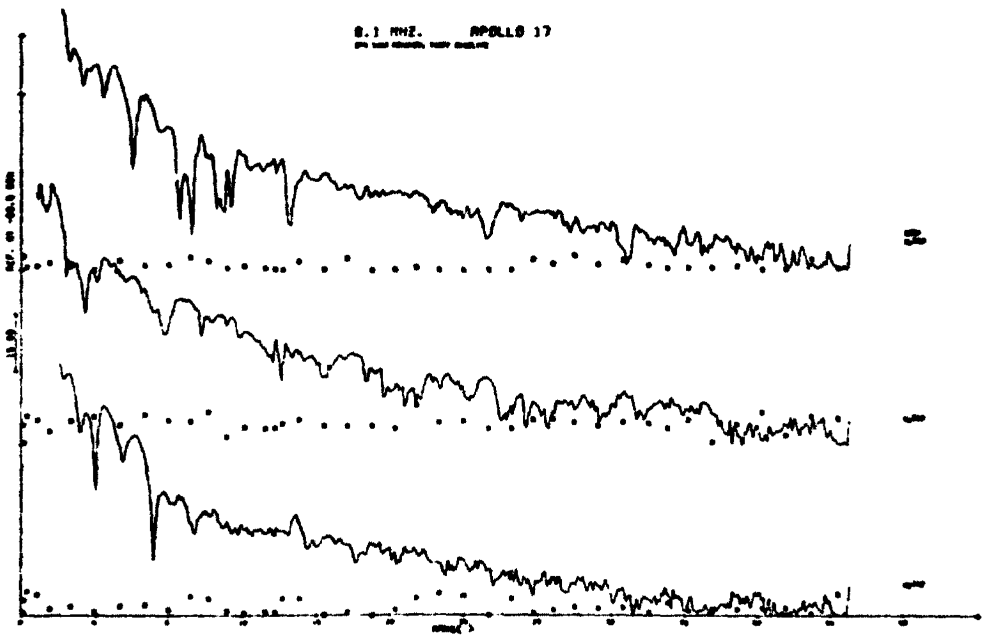
SCI2B Plots


1.8 MHZ. APOLLO 17
2.1 MHZ. APOLLO 17
1.8 MHZ. APOLLO 17
2.1 MHZ. APOLLO 17

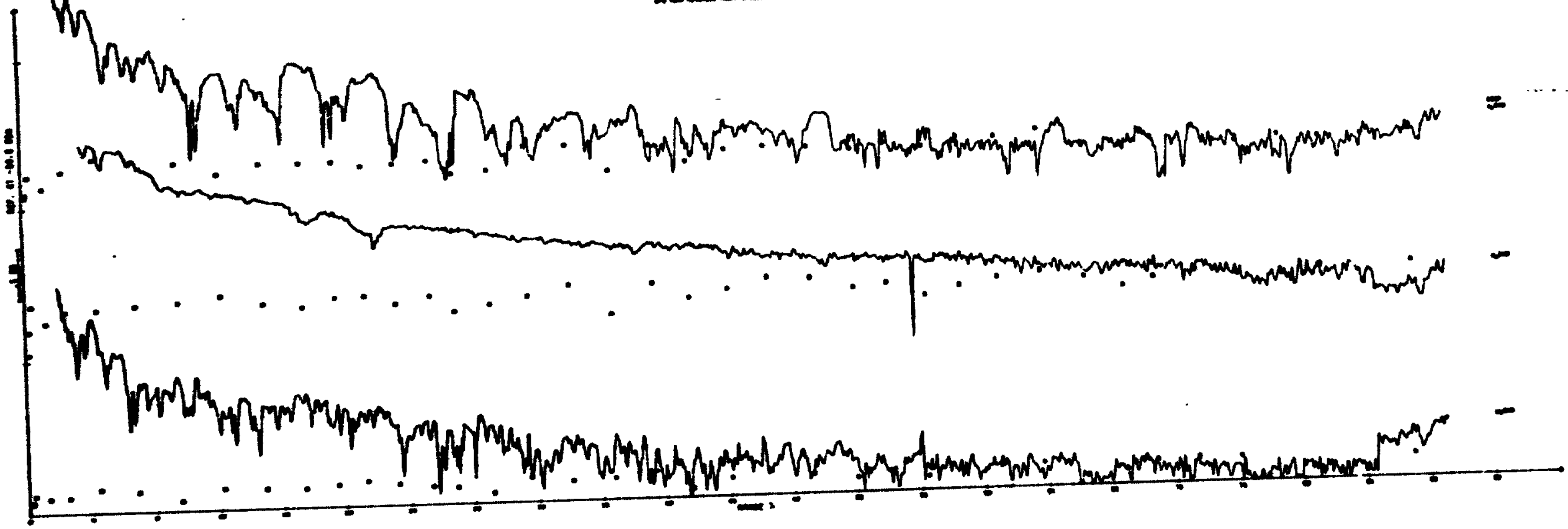

4.0 MHZ. APOLLO 17


8.1 MHZ. APOLLO 17

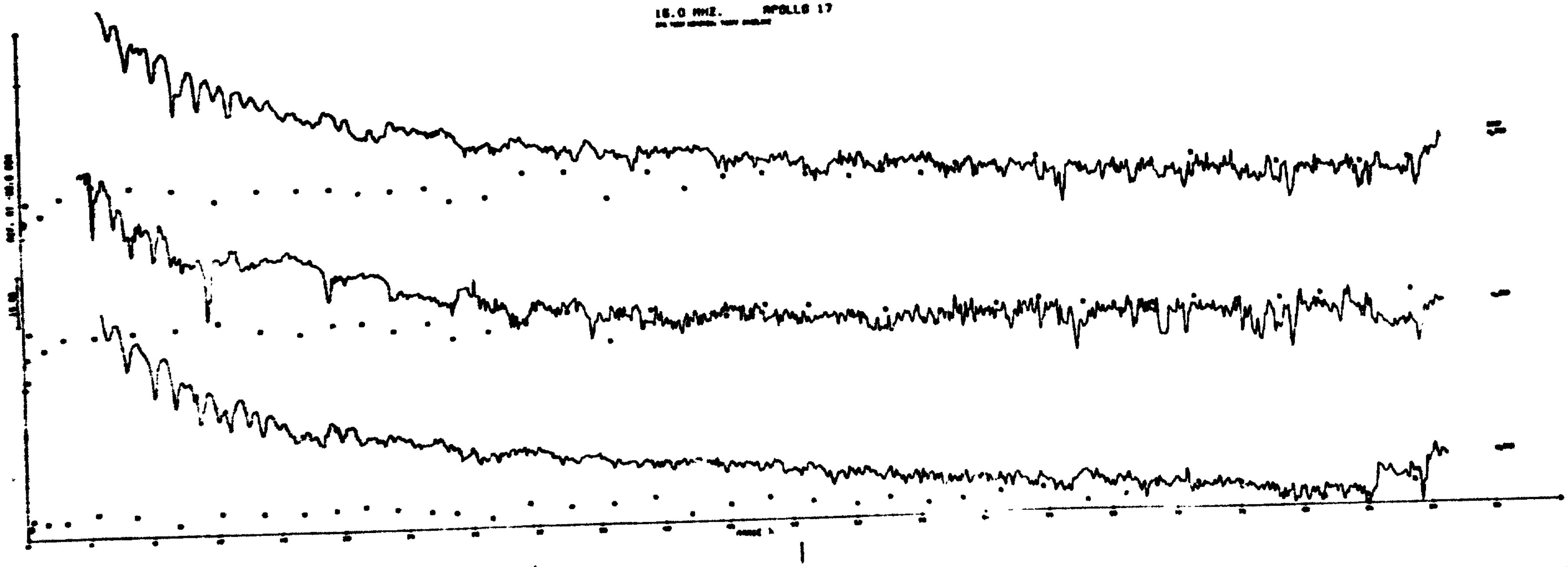

4.0 MHZ. APOLLO 17

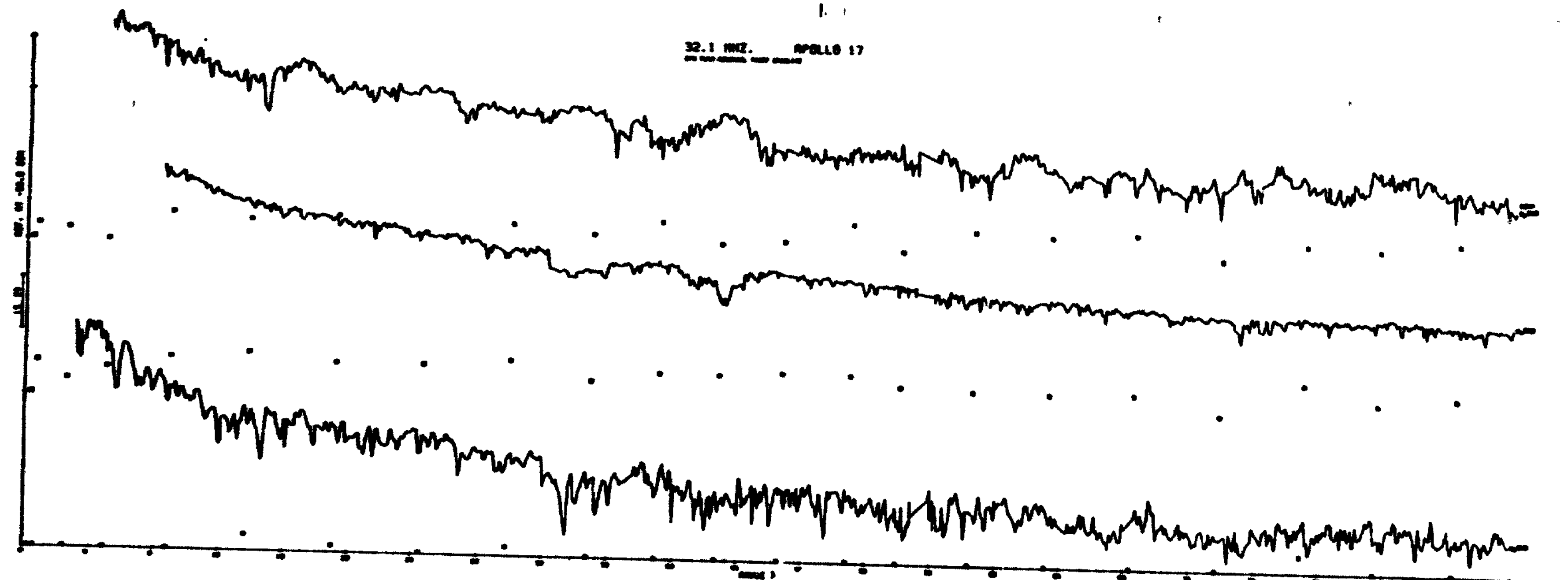

8.1 MHZ. APOLLO 17

16.0 MHZ. APOLLO 17

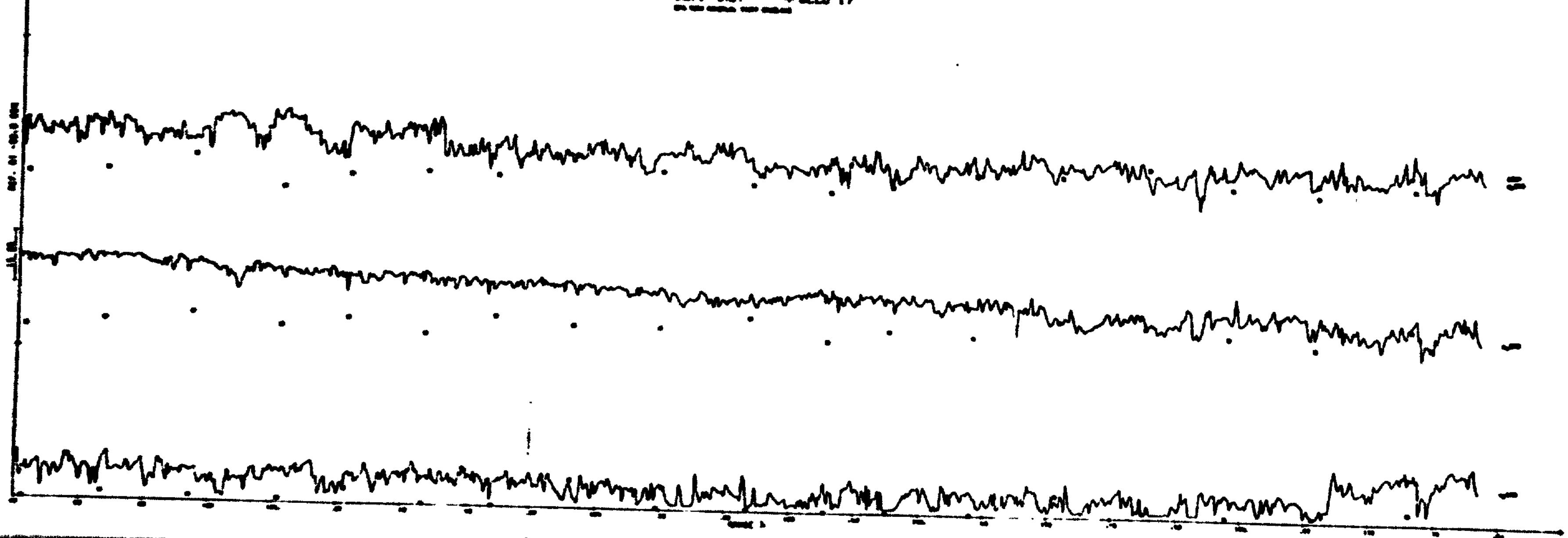


16.0 MHZ. APOLLO 17

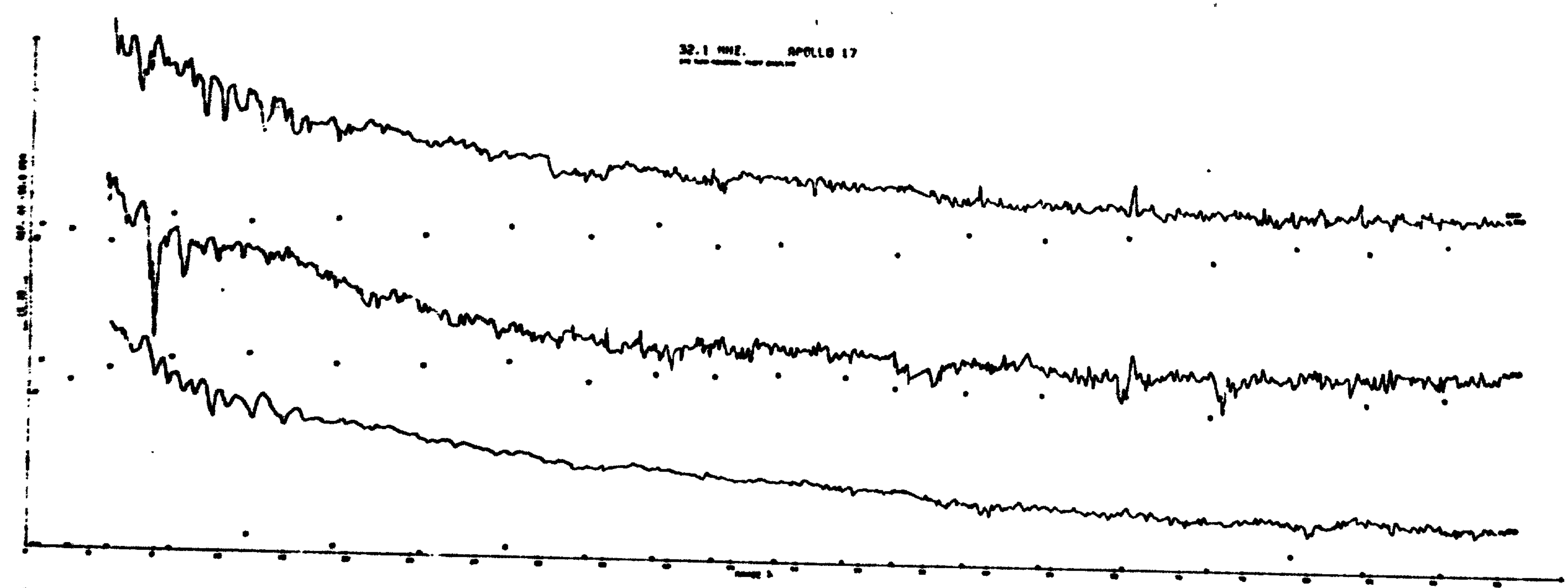




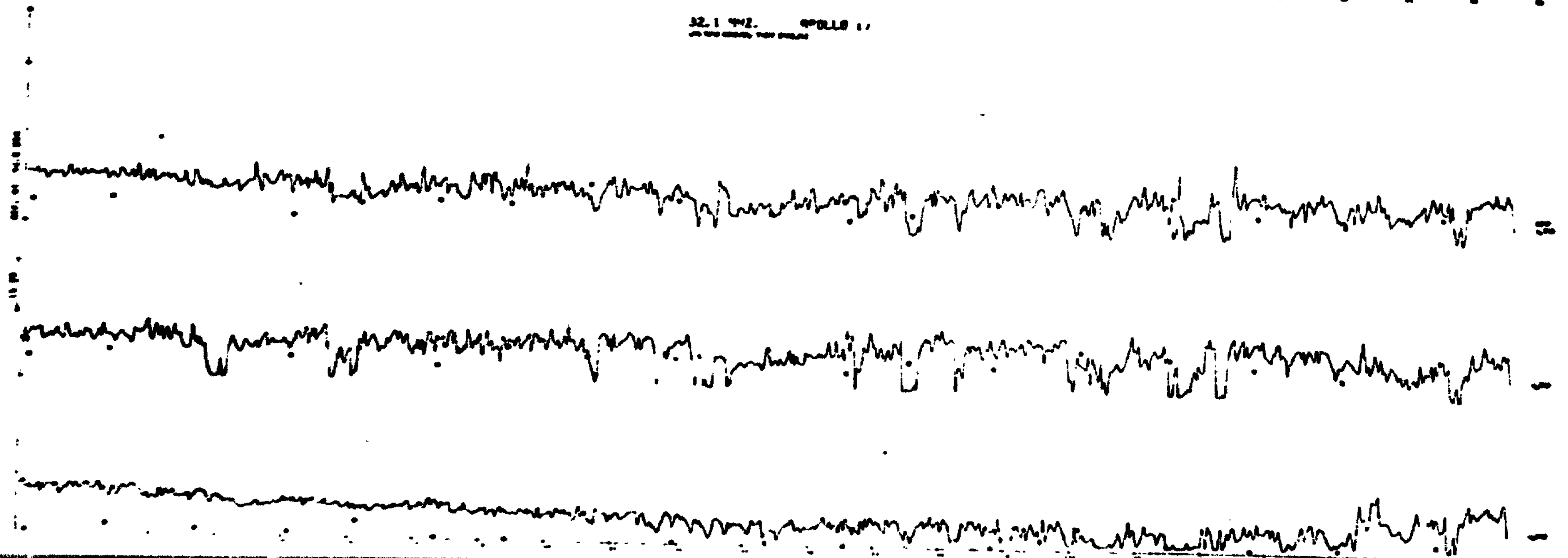
32.1 MHZ. APOLLO 17



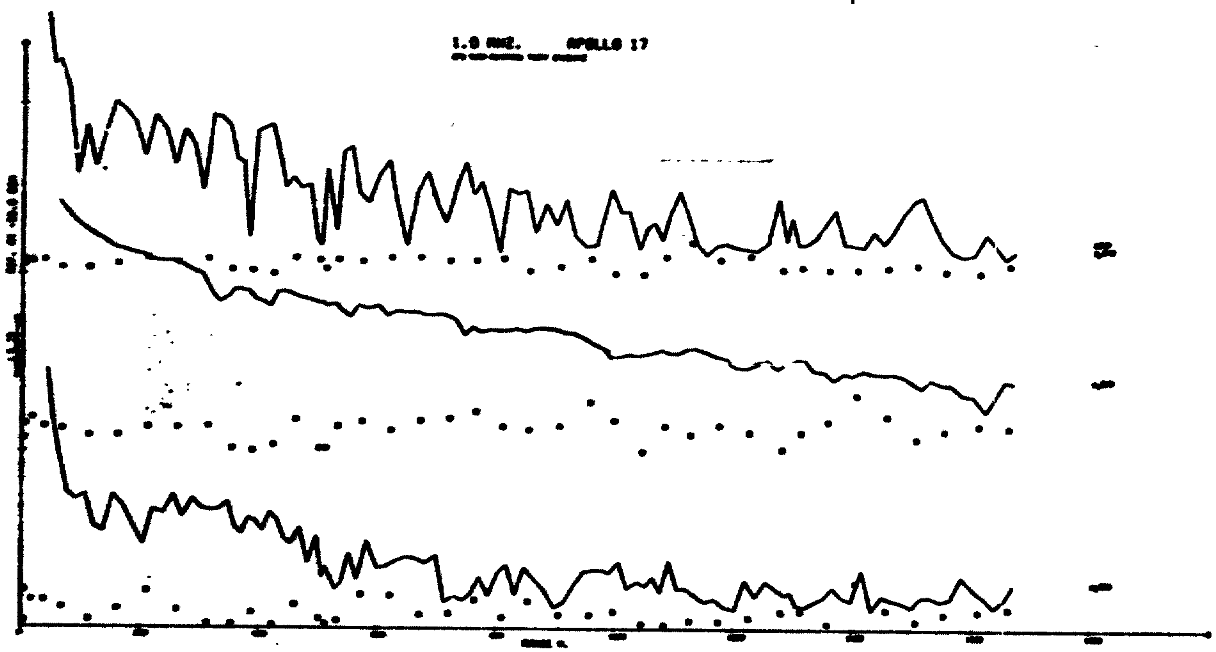
32.1 MHZ. APOLLO 17

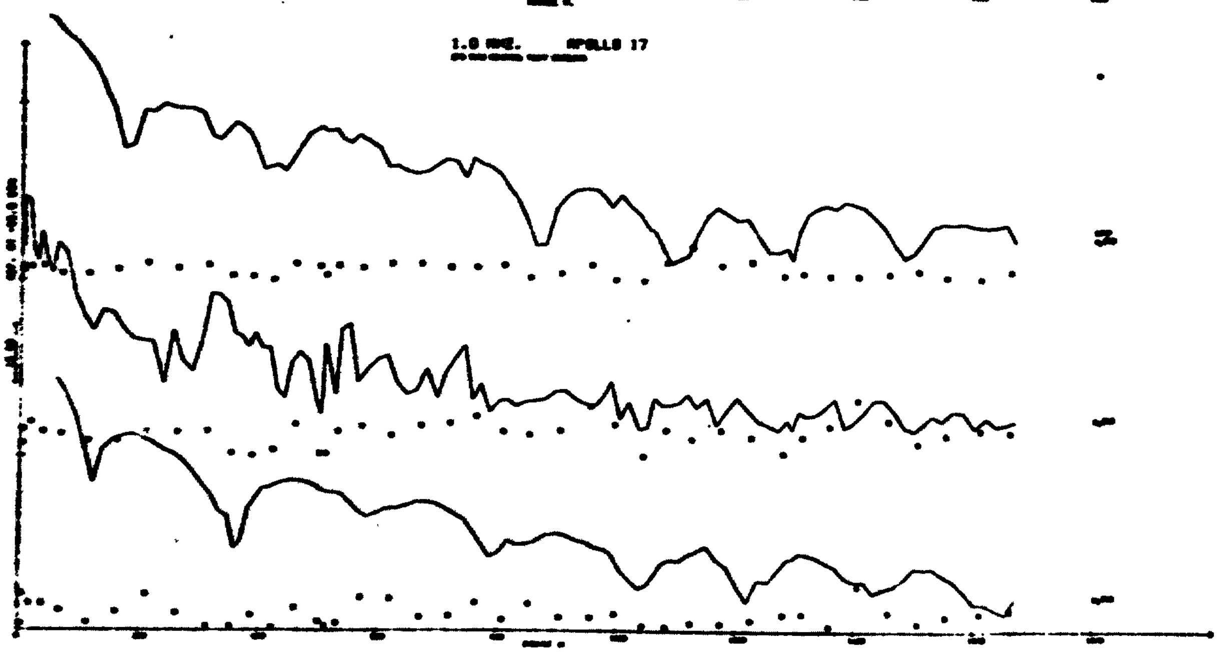


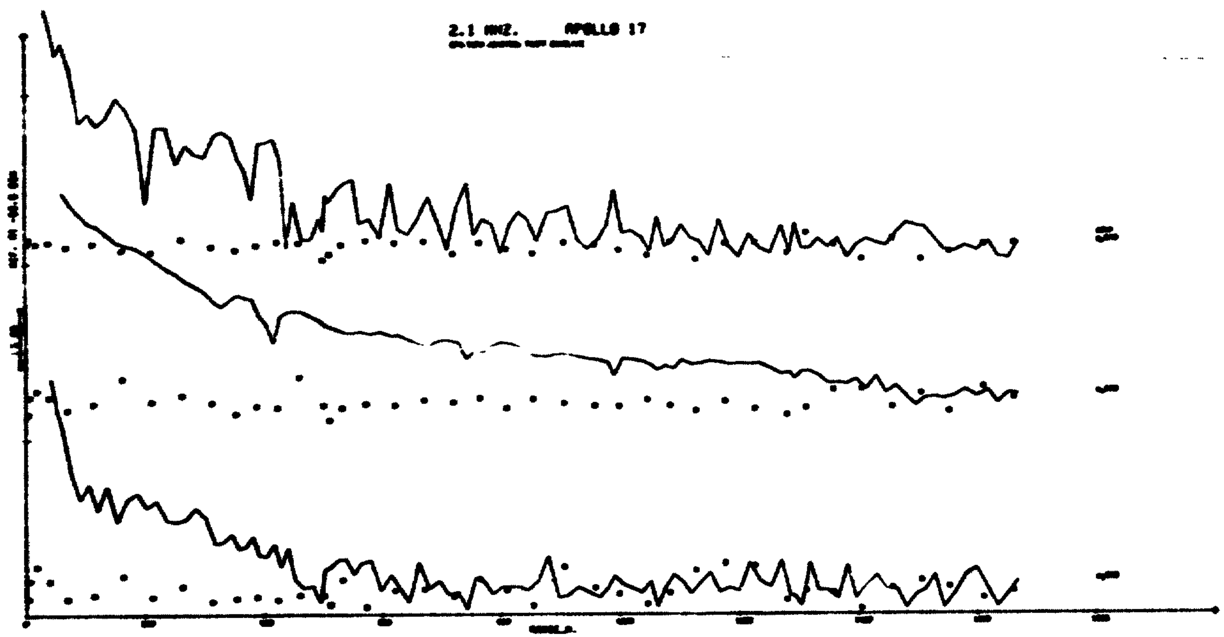
32.1 MHZ. APOLLO 17

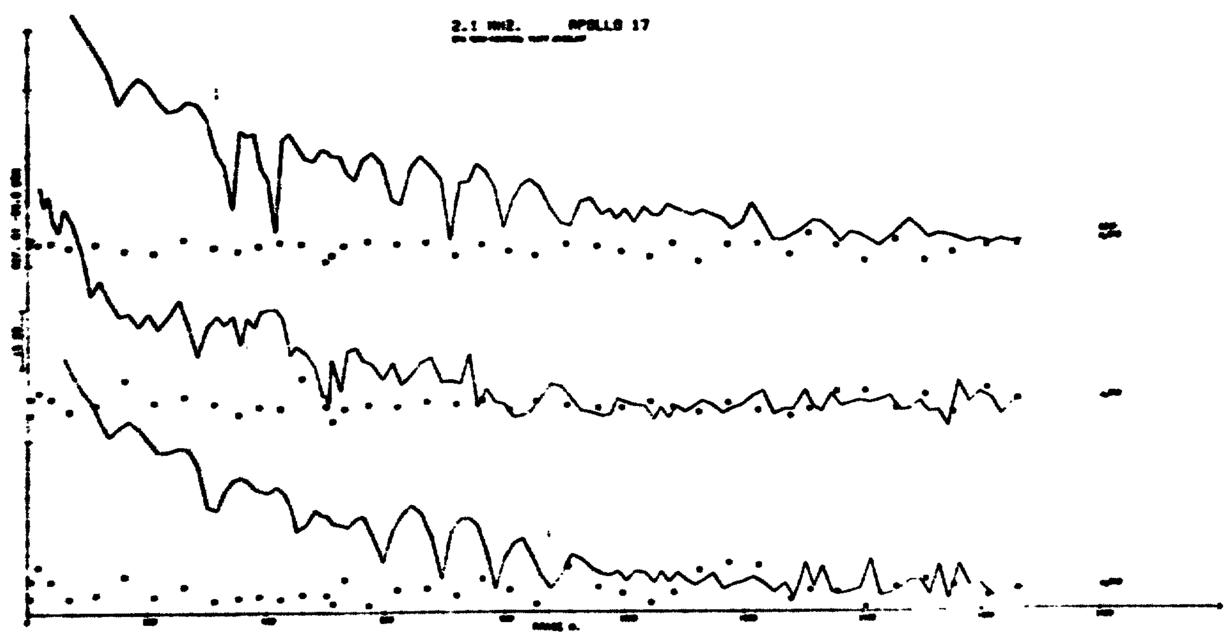


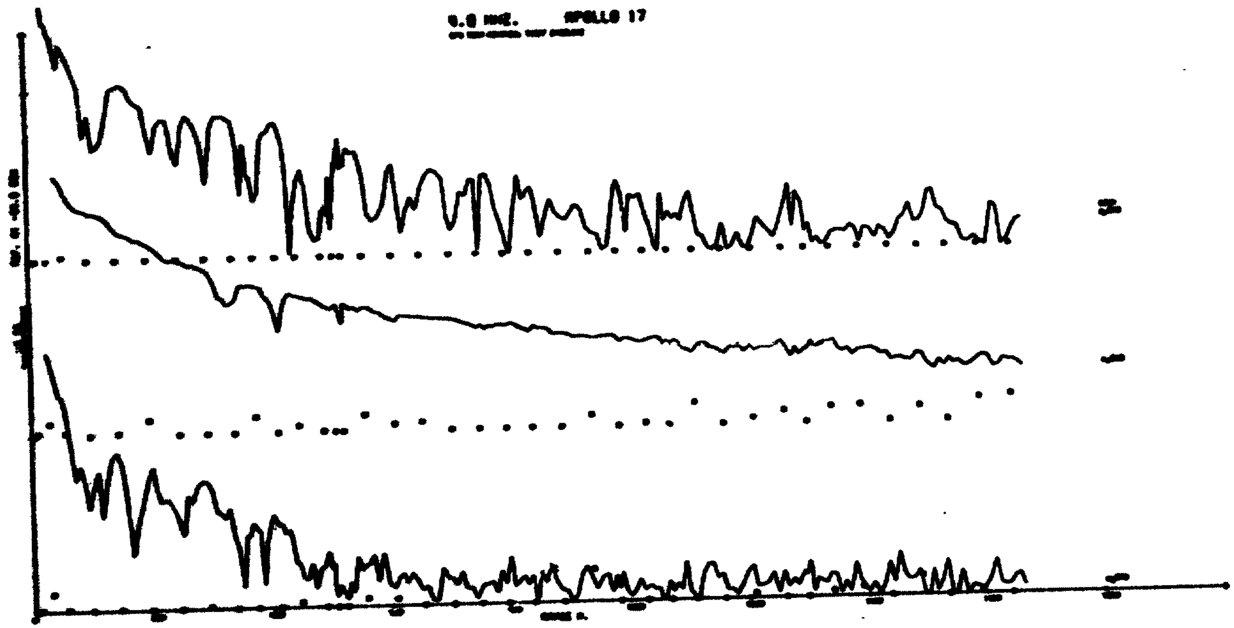
32.1 MHZ. APOLLO 17

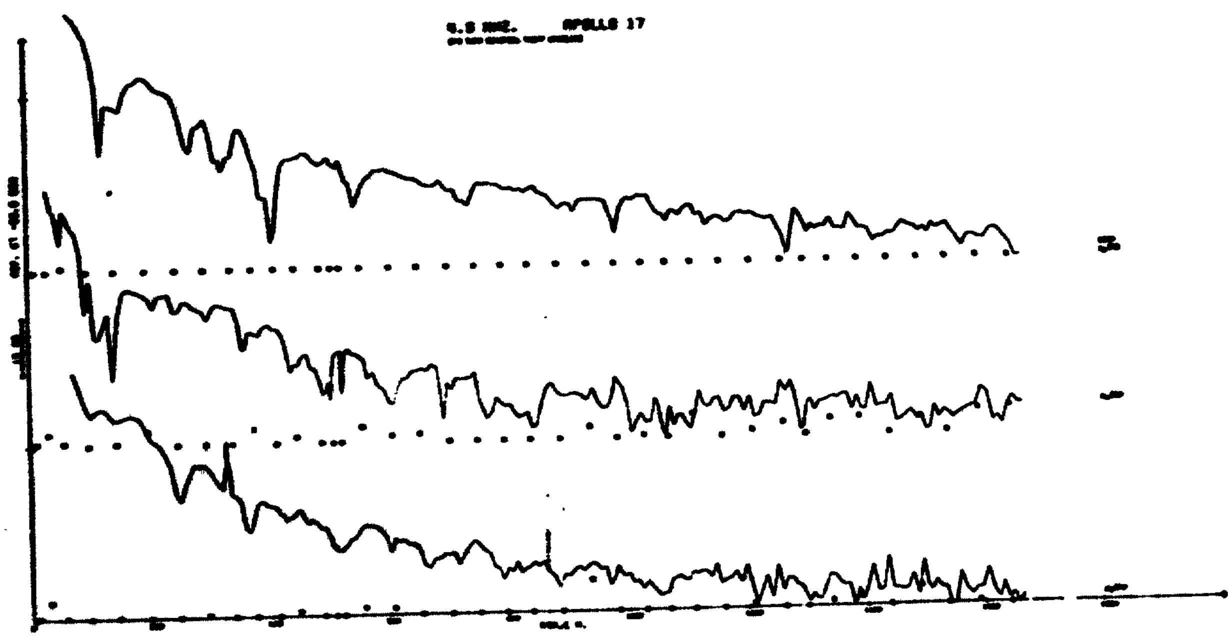

1.0 MHZ. APOLLO 17

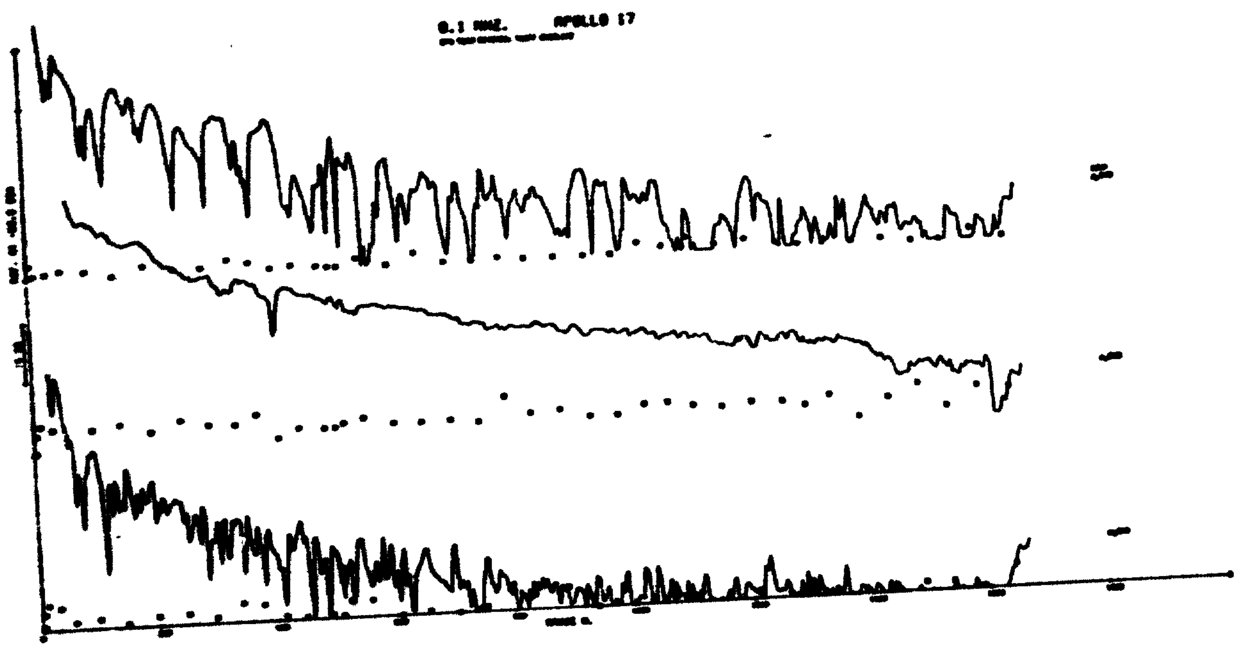

1.0 MHZ. APOLLO 17

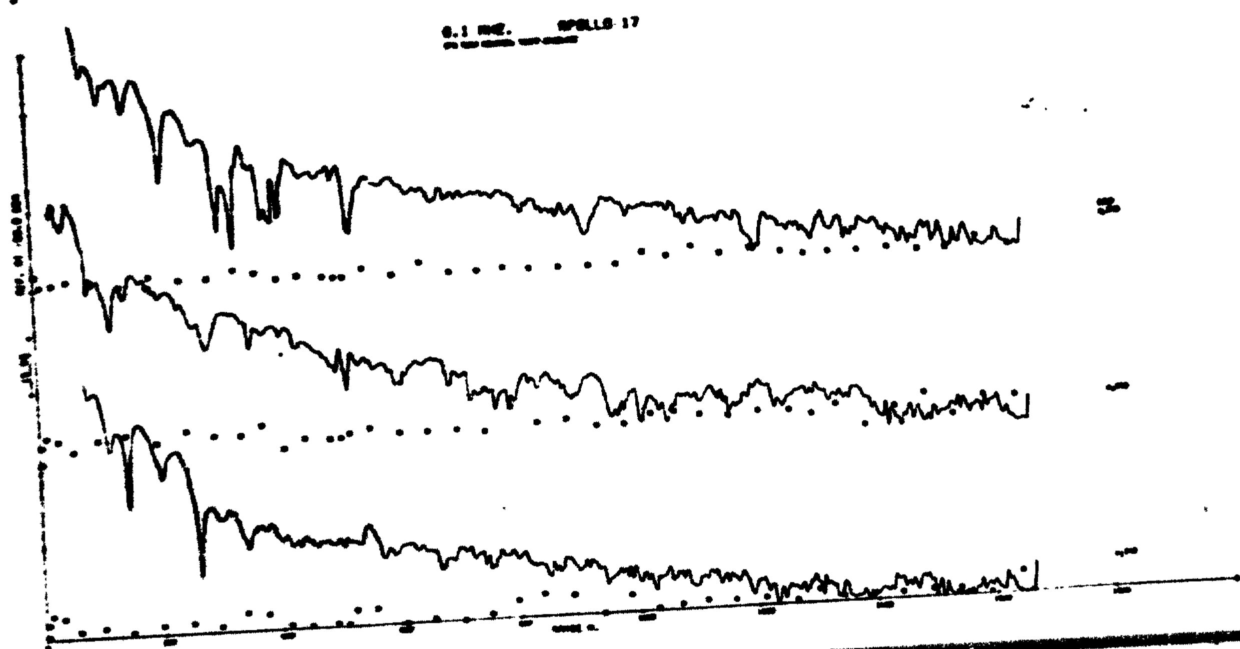

2.1 MHZ. APOLLO 17

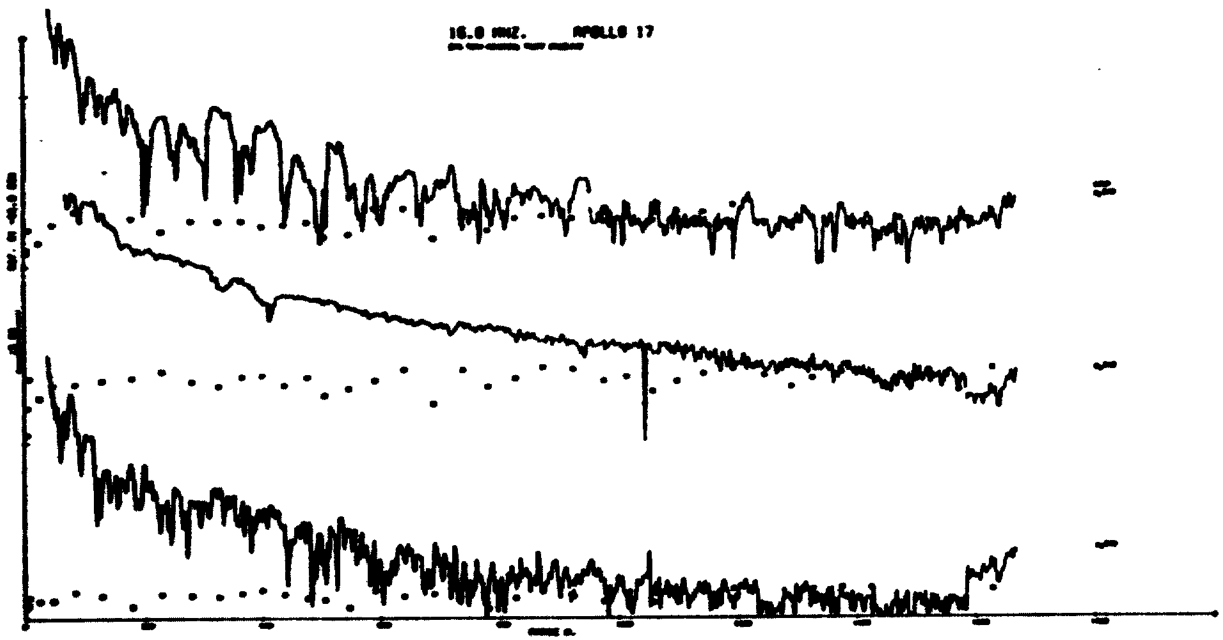

2.1 MHZ. APOLLO 17

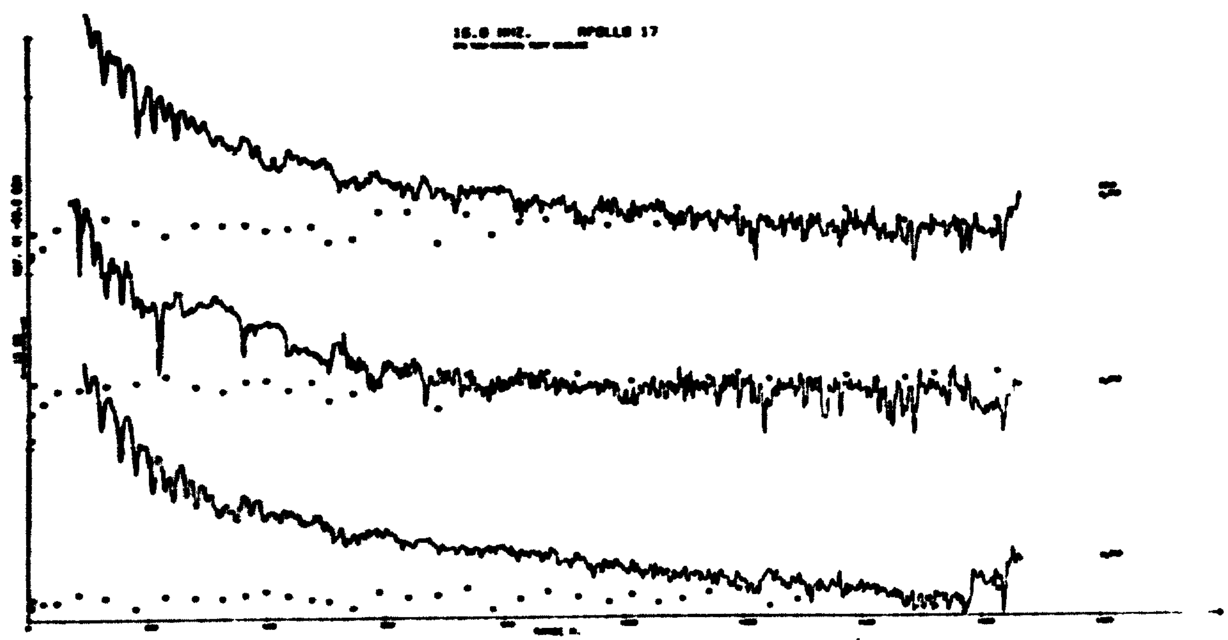

APOLLO 17

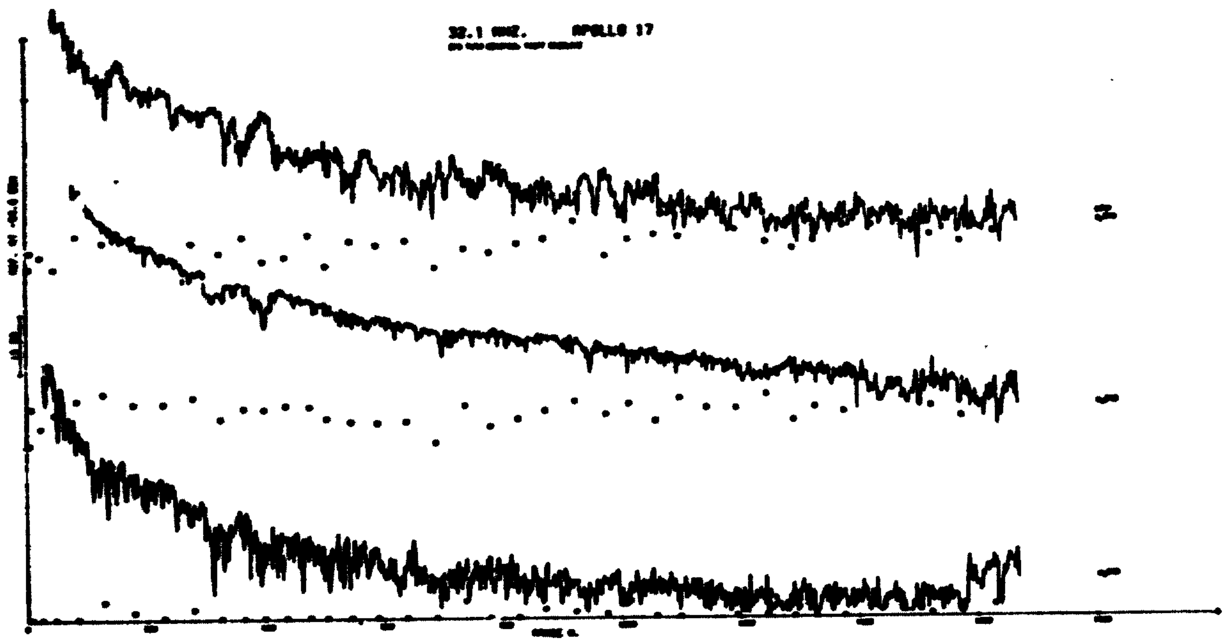

APOLLO 17

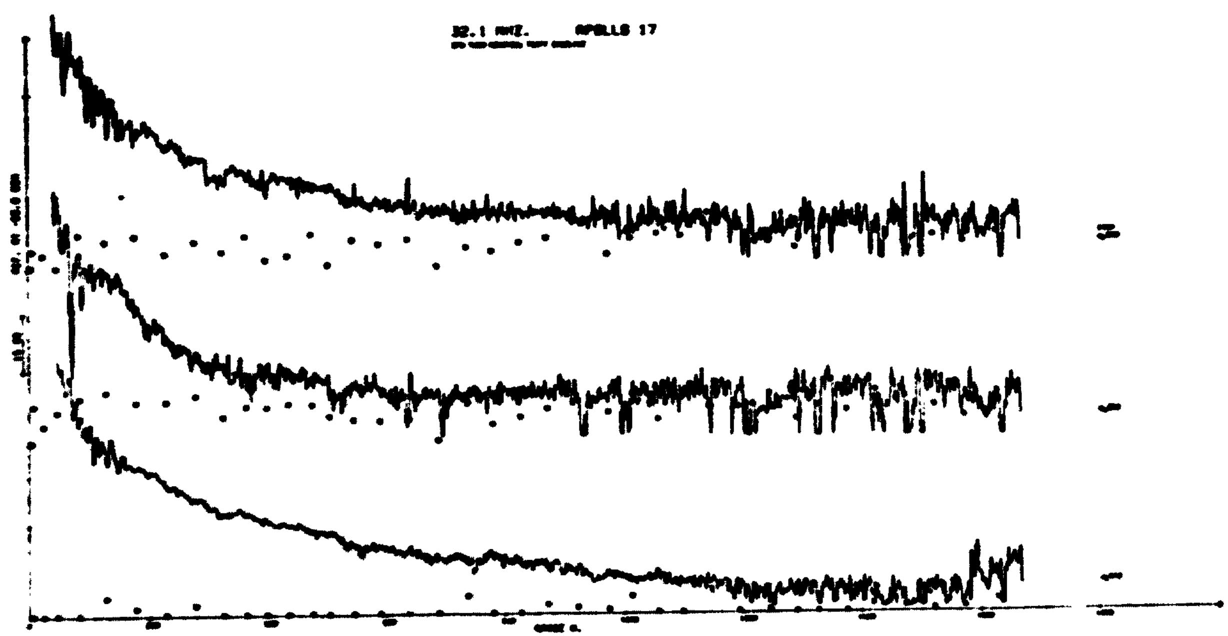

0.1 MHZ. APOLLO 17


0.1 MHZ. APOLLO 17


16.0 MHZ. APOLLO 17


16.0 MHZ. APOLLO 17


32.1 MHZ. APOLLO 17


32.1 MHZ. APOLLO 17

ii-4(..)

MEMORANDUM July 29, 1974

TO: Distribution

FROM: J. C. Rylaarsdam

SUBJECT: Modifications to data on tape SEPD09

As described in Watts' memorandum of July 2, 1974, a processing error during generation of data tapes SEPD07 through SEPD10 resulted in the loss of small amounts of dB data for 4, 8.1, 16, and 32.1 megahertz. These losses lead to erroneous correlation of the dB data with the range information, which was processed correctly. This memorandum describes a procedure for producing a set of data which is correctly matched, by removing range data corresponding the the dB data which were lost.

In the context of my report (Apollo 17 SEP Data Processing - July 1974) the processing is performed by program LUNACPY6, using file SCI2 as input. The modified data are designated as file SCI2M; this file is of exactly the same format as file SCI2, and contains all the same data, except for the changes described. File SCI2M is intended as a replacement for file SCI2 in any of the processing functions described in the report. (However, since some of the missing data occurred during the turn at EP-4, no processing by LUNAPLT4, LUNACPY4, or ANTENNAO was attempted, and none is reccommended using the modified data.)

The following is a recapitulation of Watts' description of the error, including one item which was not explained in his memorandum.

For each frequency, M * 400 data words were assembled in memory for each component, where M = 1, 1, 2, 4, 8, and 13 for frequencies of 1, 2.1, 4, 8.1, 16, and 32.1 megahertz respectively. Than M blocks of length 387 words were defined, with origins of 1, 401, ..., (M - 1) * 400 + 1; the origins should have been 1, 388, ..., (M - 1) * 387 + 1. What is not made clear by Watts' memorandum is that the first M words of the first block did not contain data, and were discarded; hence M blocks, each <u>of length 386 words</u>, were written on the output tape. Then output block i would contain

(a) the last 387 - (M - i + 1) words of block i, followed by

(b) the first M - i words of block i + 1.

N. B.

(1) part (b) above does not apply to output block M (M - i = 0, and block i + 1 does not exist).

(2) if the above-mentioned origins had been defined correctly, then the above definition of output blocks would yield the desired result.

(3) In the cases where M = 1, the data on the output file are correct.

(4) The last (M - 1) * 13 words of the last output block do not contain data.

The words which should have been used to assemble the output blocks are given in table 1; those which were used are given in table 2.

In the light of the above discussion the following procedure can be derived for matching the range data correctly to the erroneous dB data.

Having assembled the M blocks of length 386 in memory, define a set of "incorrect" block origins corresponding to those used in Watts' processing; since the M words are no longer present at the beginning of the data, this series is now 1 - M, 401 - M, ..., 400 * (M - 1) + 1 - M. Taking the length of these blocks as 387, a new set of M blocks, each of length 386, may be generated by selecting and reassembling portions of the blocks as described in (a) above, and adding 13 * (M - 1) words of padding to the end of the last block.

The words used to assemble the blocks of modified range data are indicated in table 3.

A listing of program LUNACPY6 begins on page seven. Following the listing is an updated set of plots, designated SCI2BM, produced by LUNAPLT5 from file SCI2M.

Block	4 MHz.	8.1 MHz.	16 MHz.	32.1 MHz.
1	3- 388	5- 390	9- 394	14- 399
2	389- 774	391- 776	395- 780	400- 785
3		777-1162	781-1166	786-1171
4		1163-1548	1167-1552	1172-1557
5			1553-1938	1558-1943
6			1939-2324	1944-2329
7			2325-2710	2330-2715
8			2711-3096	2716-3101
9				3102-3487
10				3488-3873
11				3874-4259
12				4260-4645
13				4646-5031

Table 1 - Locations which should have been used to assemble blocks of dB data for file SCI2.

Block	4 MHz.	8.1 MHz.	16 MHz.	32.1 MHz.
1	3- 387 401	5- 387 401- 403	9- 387 401- 407	14- 387 401- 412
2	402- 774 (775- 787)	404- 787 801- 802	408- 787 801- 806	413- 787 801- 811
3		803-1187 1201	807-1187 1201-1205	812-1187 1201-1210
4		1202-1548 (1549-1587)	1206-1587 1601-1604	1211-1587 1601-1609
5			1605-1987 2001-2003	1610-1987 2001-2008
6			2004-2387 2401-2402	2009-2387 2401-2407
7			2403-2787 2801	2408-2787 2801-2806
8			2802-3096 (3097-3187)	2807-3187 3201-3205
9				3206-3587 3601-3604
10				3605-3987 4001-4003
11				4004-4387 4401-4402
12				4403-4787 4801
13				4802-5031 (5032-5187)

Table 2 - Locations which were used to assemble blocks of dB data for file SDI2. (Locations in parentheses contain meaningless information.)

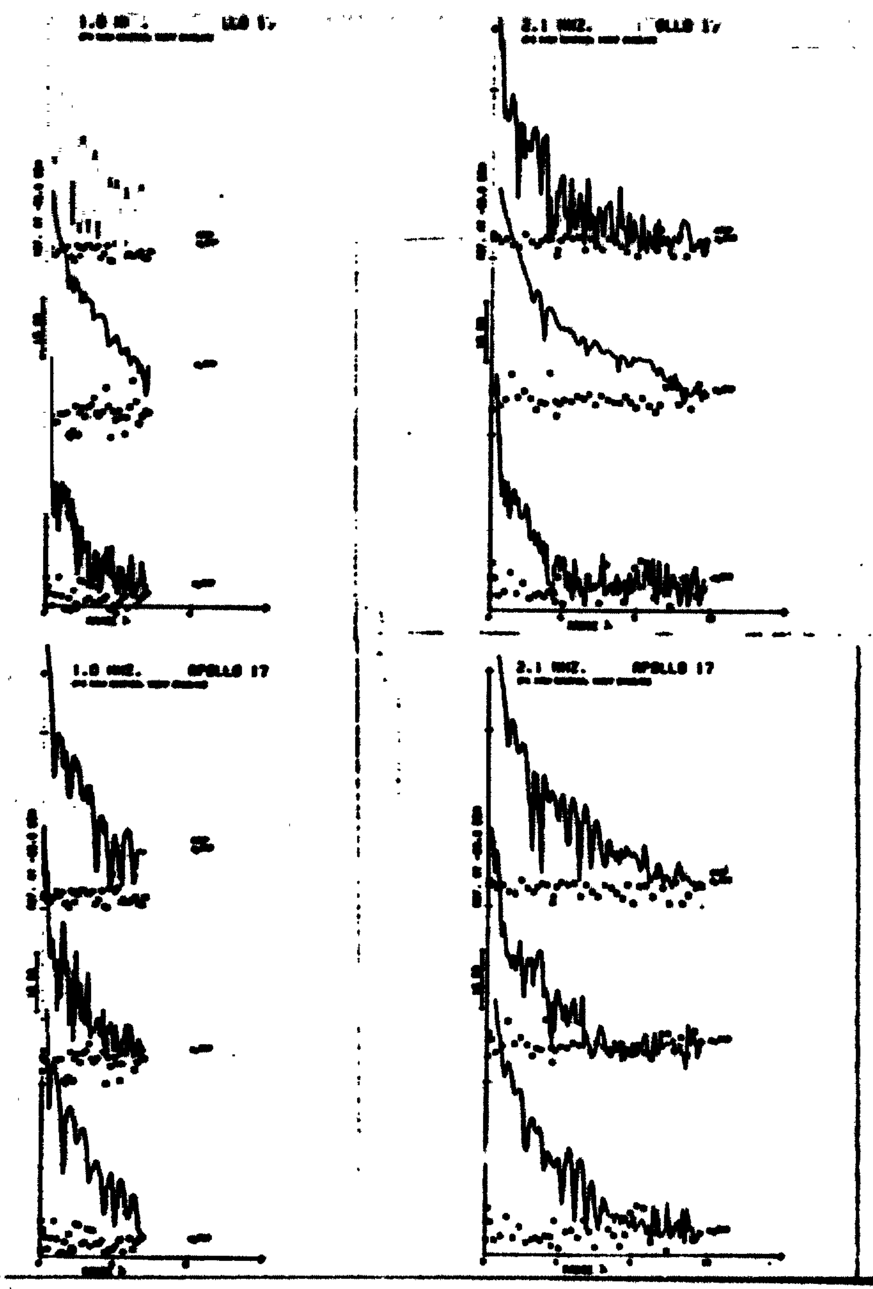
Block	4 MHz.	8.1 MHz.	16 MHz.	32.1 MHz.
1	1- 385 399	1- 383 397- 399	1- 379 393- 399	1- 374 388- 399
2	400- 772 (773- 785)	400- 783 797- 798	400- 779 793- 798	400- 774 788- 798
3		799-1183 1197	799-1179 1193-1197	799-1174 1188-1197
4		1198-1544 (1545-1583)	1198-1579 1593-1596	1198-1574 1588-1596
5			1597-1979 1993-1995	1597-1574 1988-1995
6			1996-2379 2393-2394	1996-2374 2388-2394
7			2395-2779 2793	2395-2774 2788-2793
8			2794-3088 (3089-3179)	2794-3174 3188-3192
9				3193-3574 3598-3591
10				3592-3974 3988-3990
11				3991-4374 4388-4389
12				4390-4774 4788
13				4789-5018 (5019-5174)

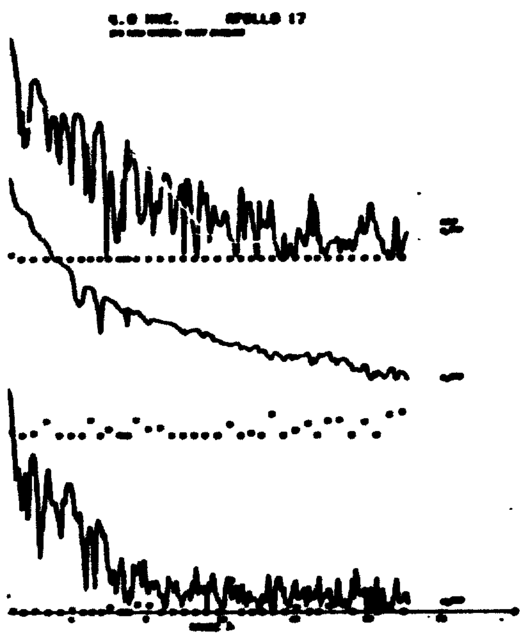
Table 3 - Locations used to assemble blocks of range data for file SCI2M. (Locations in parentheses contain padding.)

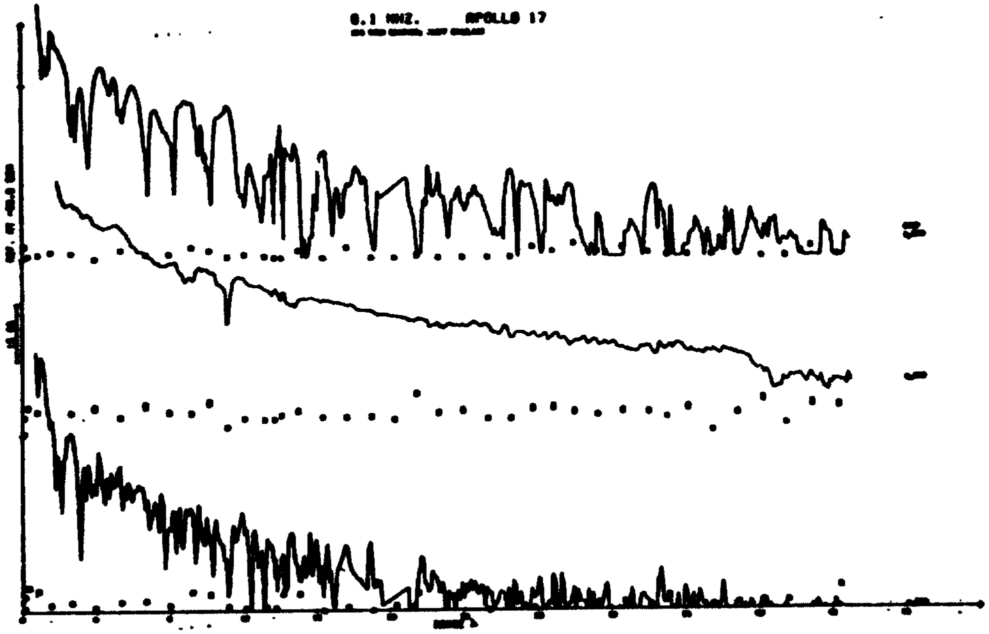
```
*****************************************************************************
*                                                                           *
*                                LUNACPY6                                   *
*                                                                           *
*****************************************************************************

C     PROGRAM TO GENERATE A MODIFIED VERSION OF FILE #2 , IN WHICH
C     THE RANGE DATA CORRESPONDING TO MISSING DB INFORMATION HAVE
C     BEEN DELETED.
C
      INTEGER*4 MM(4) / 2, 4, 8, 13 /
      REAL*4 DATA(6000)
C
C     COPY ALL THE DATA WHICH REQUIRE NO MODIFICATION - I. E. THE
C     LABEL RECORD THROUGH THE 2 MHZ. DATA.
C
      DO 10 I = 1, 29
        READ (1, 1000)  (DATA(J), J = 1, 579)
        WRITE(2, 1000)  (DATA(J), J = 1, 579)
   10 CONTINUE
C
C     LOOP OVER THE FOUR FREQUENCIES WHICH REQUIRE MODIFICATION.
C
      DO 150 I = 1, 4
        M = MM(I)
        IORG = 1
        IEND = 386
C
C       READ THE M BLOCKS OF RANGE DATA INTO MEMORY.
C
        DO 20 J = 1, M
          READ (1, 2000) (DATA(K), K = IORG, IEND)
          IORG = IORG + 386
          IEND = IEND + 386
   20   CONTINUE
C
C       ID IS INITIALIZED AS THE FIRST WORD OF THE FIRST GROUP
C       OF 13 WORDS TO BE DELETED.
C
        ID = 388 - M
C
C       MM1 IS THE NUMBER OF GROUPS OF 13 WORDS TO BE DELETED.
C
        MM1 = M - 1
C
C       N IS THE NUMBER OF WORDS TO BE MOVED FROM THE BEGINNING OF
C       BLOCK J + 1 TO THE END OF BLOCK J. (INITIALLY M - 1)
```

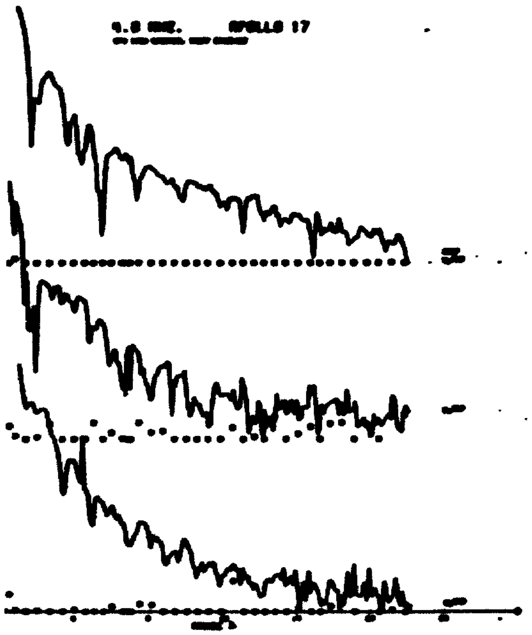
```
C
      N = MM1
C
C     LOOP OVER THE SET OF 13-WORD GROUPS.
C
      DO 60 J = 1, MM1
C
C       TRANSFER N WORDS FROM BLOCK J + 1 TO BLOCK J.
C
        DO 40 K = 1, N
          JD = ID + K - 1
          JS = JD + 13
          DATA(JD) = DATA(JS)
   40   CONTINUE
C
C       FOR GROUP J + 1, N IS DECREMENTED BY 1.
C
        N = N - 1
C
C       THE BEGINNING OF GROUP J + 1 IS 400 WORDS FROM THE
C       BEGINNING OF GROUP J.
C
        ID = ID + 400
C
C       JD IS CURRENTLY THE INDEX OF THE 386'TH WORD OF OUTPUT BLOCK
C       J; SET JS TO INDEX THE 1'ST WORD, AND THEN WRITE THE BLOCK
C       ON THE OUTPUT FILE.
C
        JS = JD - 385
        WRITE(2, 2000) (DATA(K), K = JS, JD)
        WRITE(6, 3000) (DATA(K), K = JS, JD)
   60 CONTINUE
C
C     COMPLETE OUTPUT BLOCK N BY PLACING THE LAST CORRECT RANGE
C     VALUE (CONTAINED IN WORD N)
C
      N = 386 * M
C
C     IN THE NN LOCATIONS BEGINNING AT LOCATION N + 1.
C
      NN = 13 * MM1
      DO 80 J = 1, NN
        JD = N + J
        DATA(JD) = DATA(N)
   80 CONTINUE
C
C     THE FIRST WORD OF THE OUTPUT BLOCK IS COMPUTED AS ABOVE,
C     AND THE BLOCKIS WRITTEN ON THE OUTPUT FILE.
```

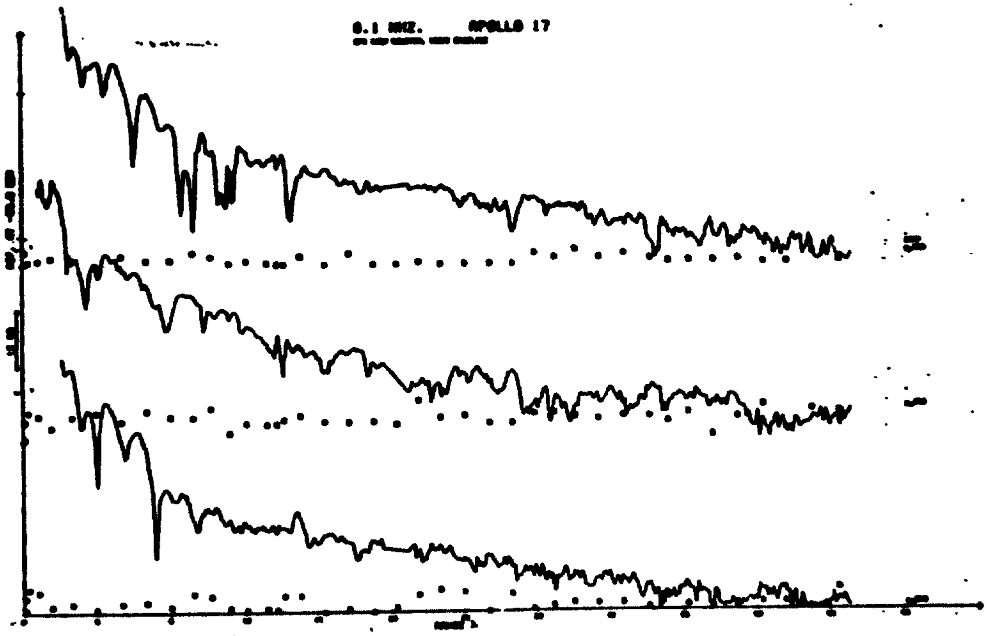
```
C
         JS = JD - 385
         WRITE(2, 2000)  (DATA(K), K = JS, JD)
         WRITE(6, 3000)  (DATA(K), K = JS, JD)
C
C        REDEFINE N AS THE LAST VALID WORD (WITHIN THE LAST BLOCK)
C        OF DB DATA
C
         N = 386 - NN
C
C        FOR EACH COMPONENT,
C
         DO 140 IC = 1, 6
C
C          READ M BLOCKS OF DB DATA.
C
           DO 120 J = 1, M
             READ (1, 2000)  (DATA(K), K = 1, 386)
C
C            WRITE BLOCKS 1 THROUGH M - 1 ON THE OUTPUT FILE IMMEDIATELY.
C
             IF(J .NE. M) GO TO 110
C
C              FILL THE LAST NN WORDS OF BLOCK M WITH PADDING
C              BEFORE WRITING IT ON THE OUTPUT FILE.
C
               DO 100 K = 1, NN
                 JD = N + K
                 DATA(JD) = -135.
  100          CONTINUE
  110        CONTINUE
             WRITE(2, 2000)  (DATA(K), K = 1, 386)
  120      CONTINUE
  140    CONTINUE
  150  CONTINUE
       RETURN
C
 1000  FORMAT(200A4, 200A4, 179A4)
 2000  FORMAT(200F6.1, 186F6.1)
 3000  FORMAT('0 ' / '-', 15F6.1 / 25(1X, 15F6.1/))
       END
```

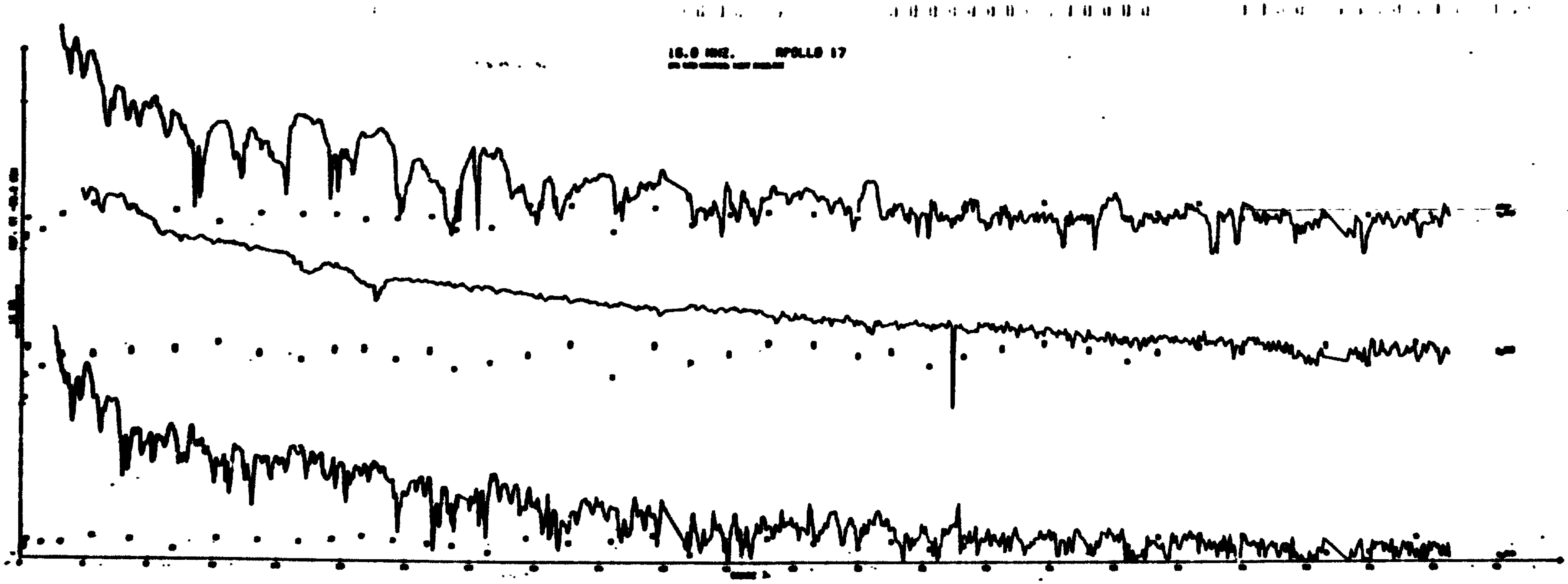

1.0 MHZ. APOLLO 17
2.1 MHZ. APOLLO 17

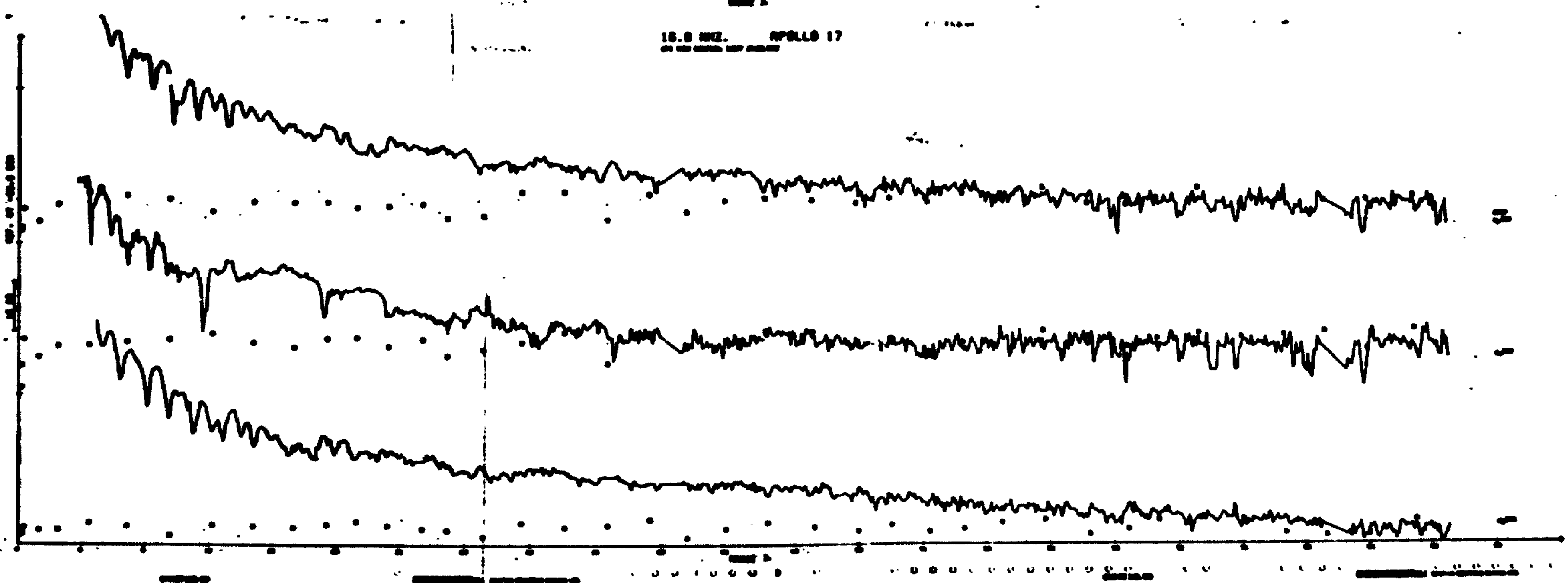

APOLLO 17

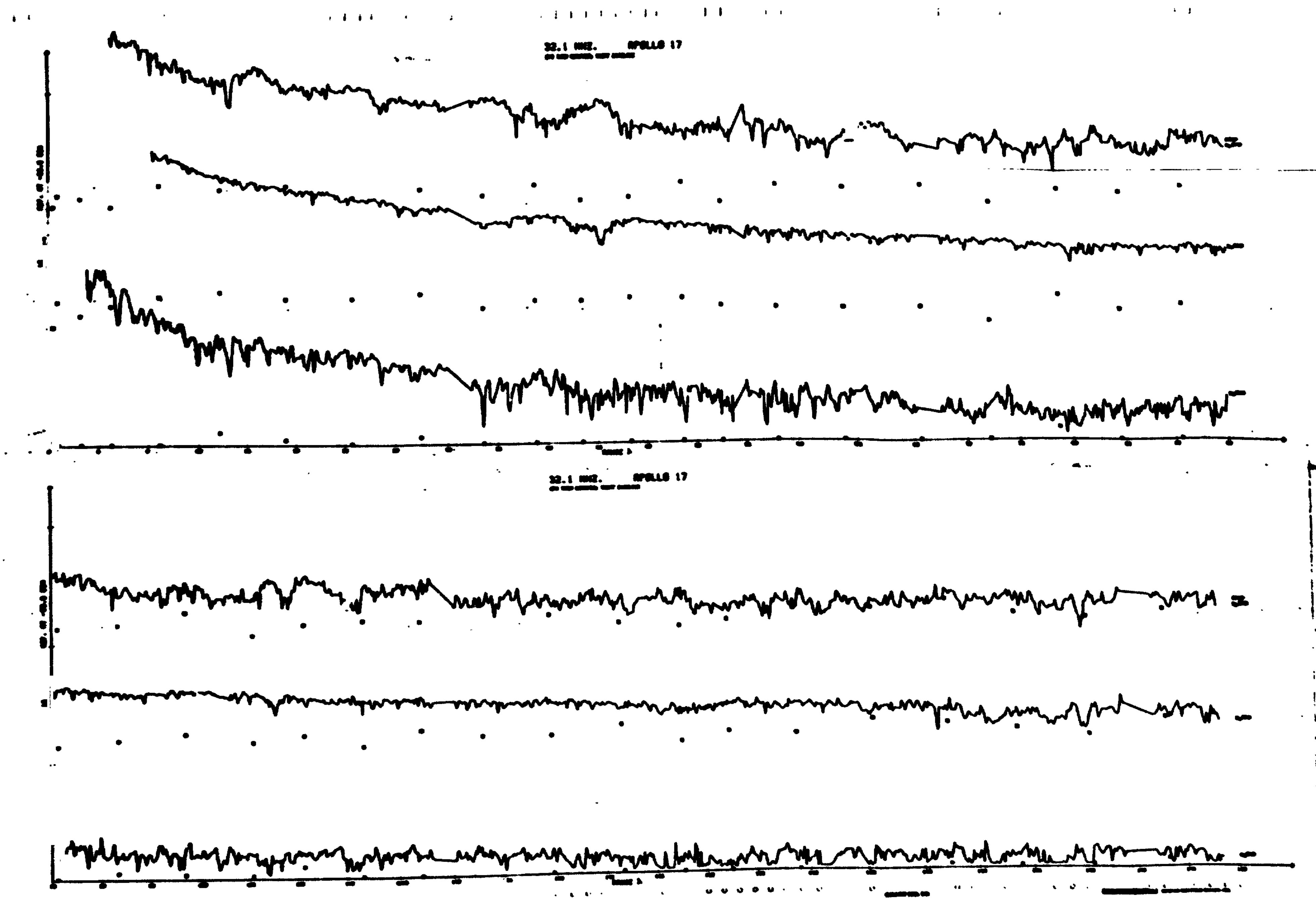

0.1 MHZ. APOLLO 17


APOLLO 17

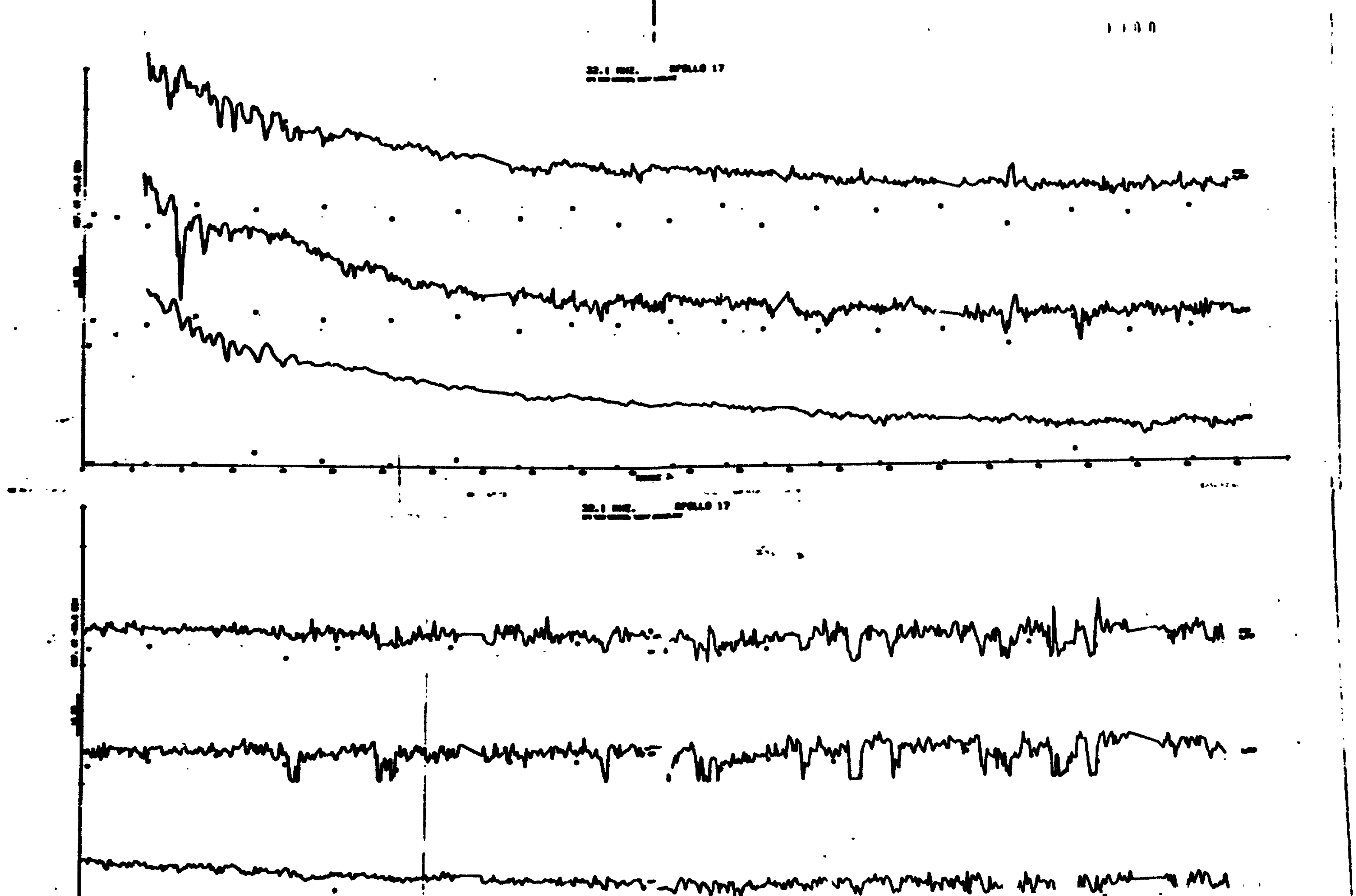

0.1 MHZ. APOLLO 17

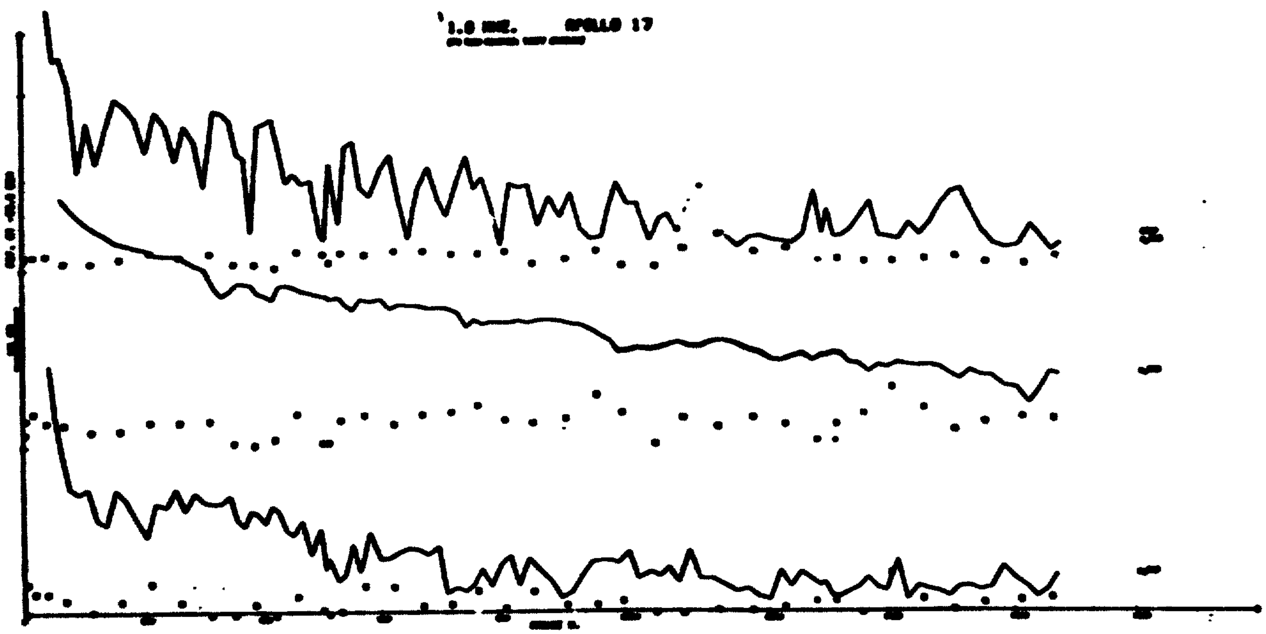

15.0 MHZ. APOLLO 17

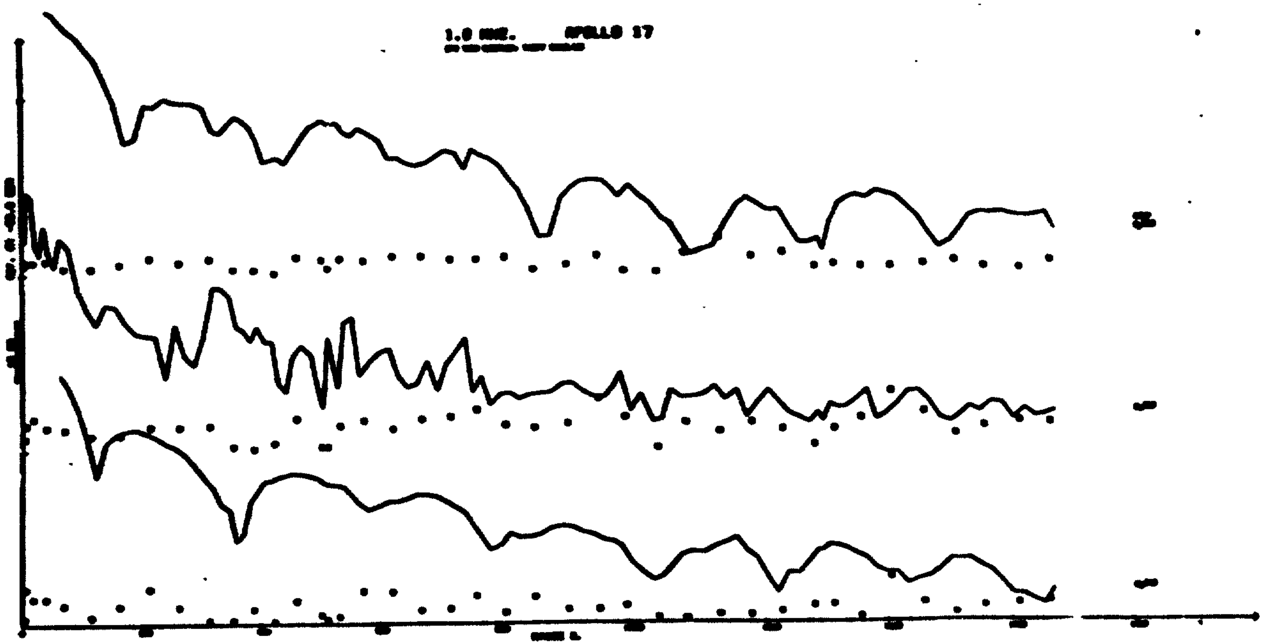

15.0 MHZ. APOLLO 17

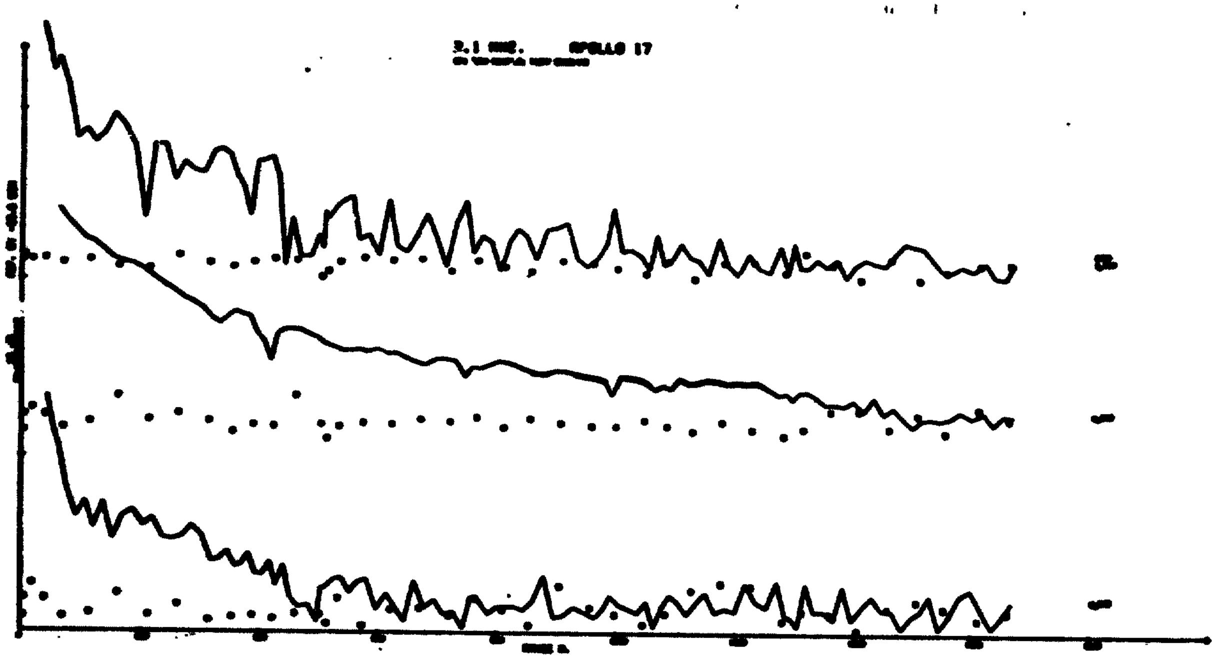


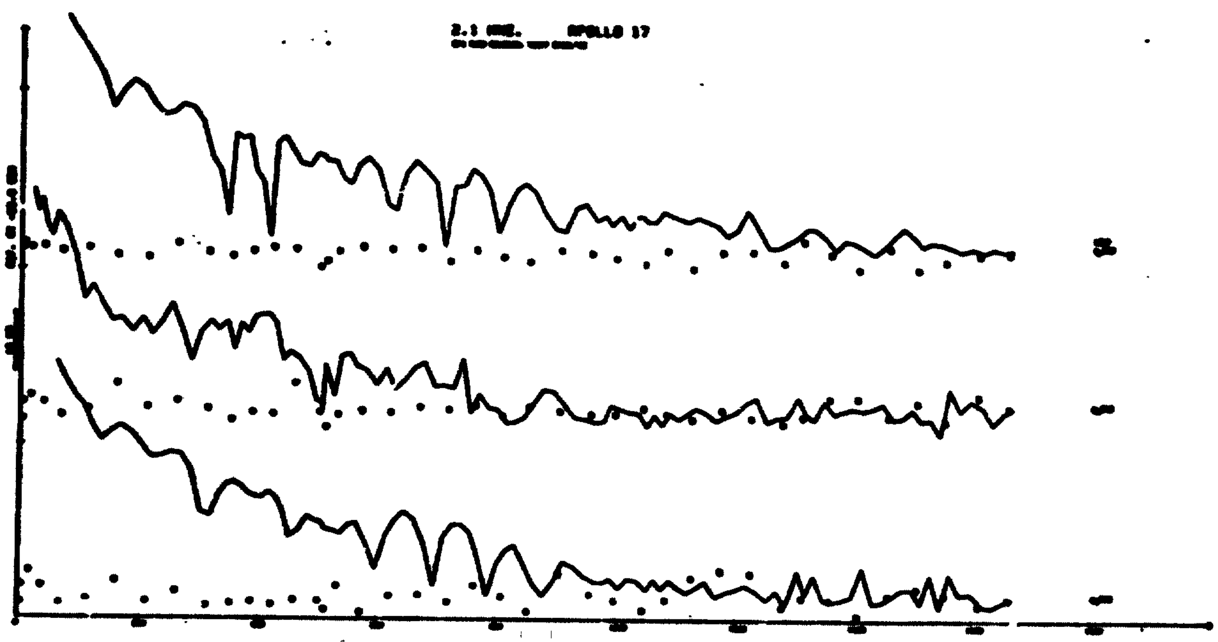
32.1 HRS. APOLLO 17
32.1 HRS. APOLLO 17

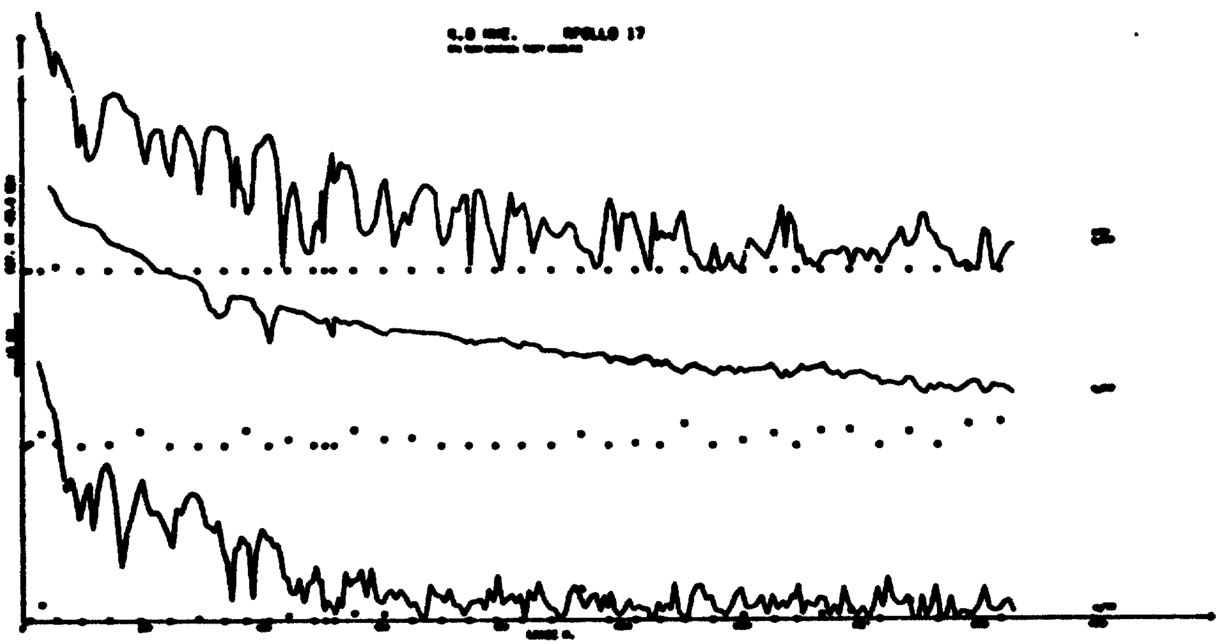

APOLLO 17
APOLLO 17

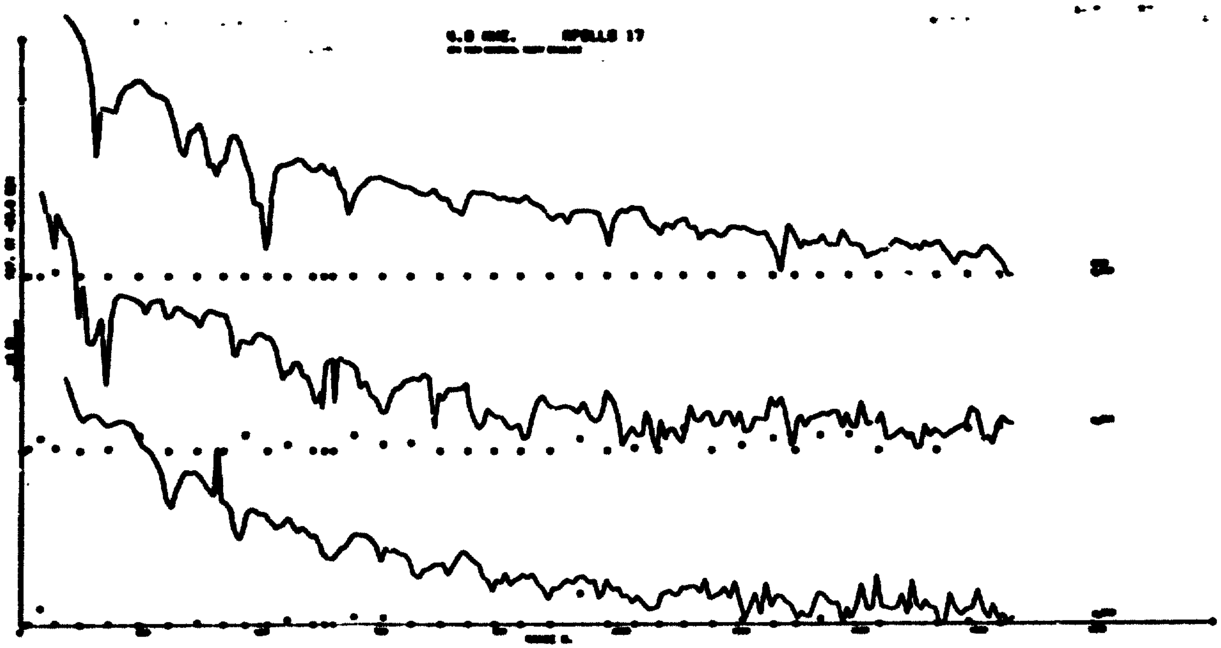

1.0 MHZ. APOLLO 17

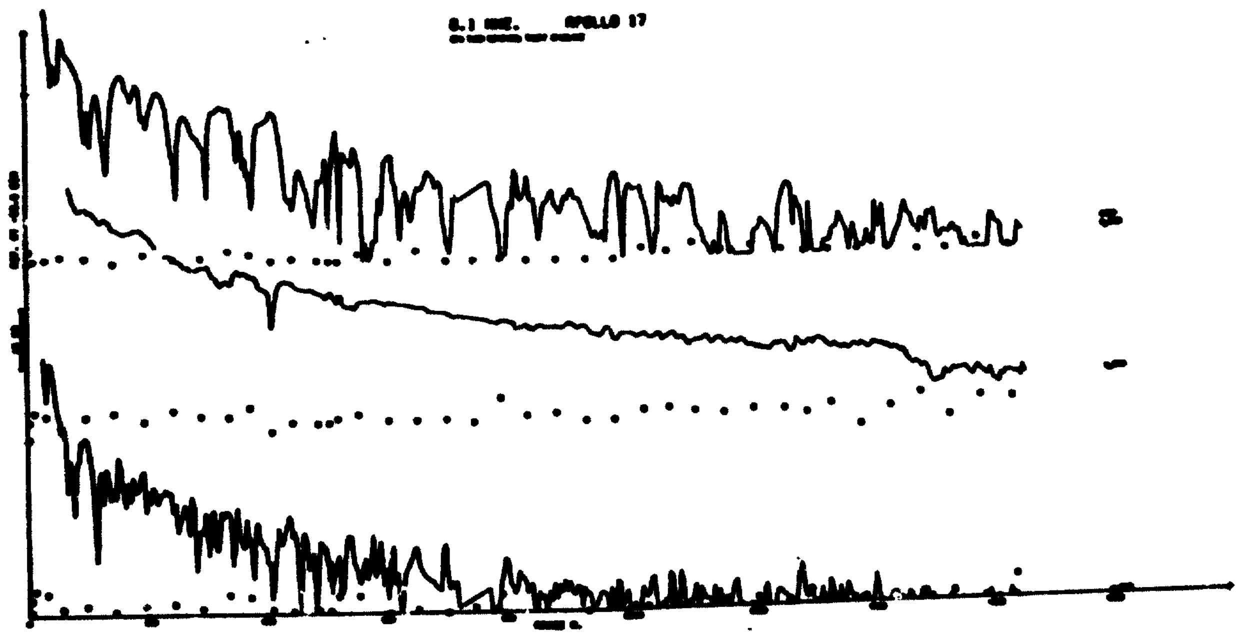

1.0 MHZ. APOLLO 17

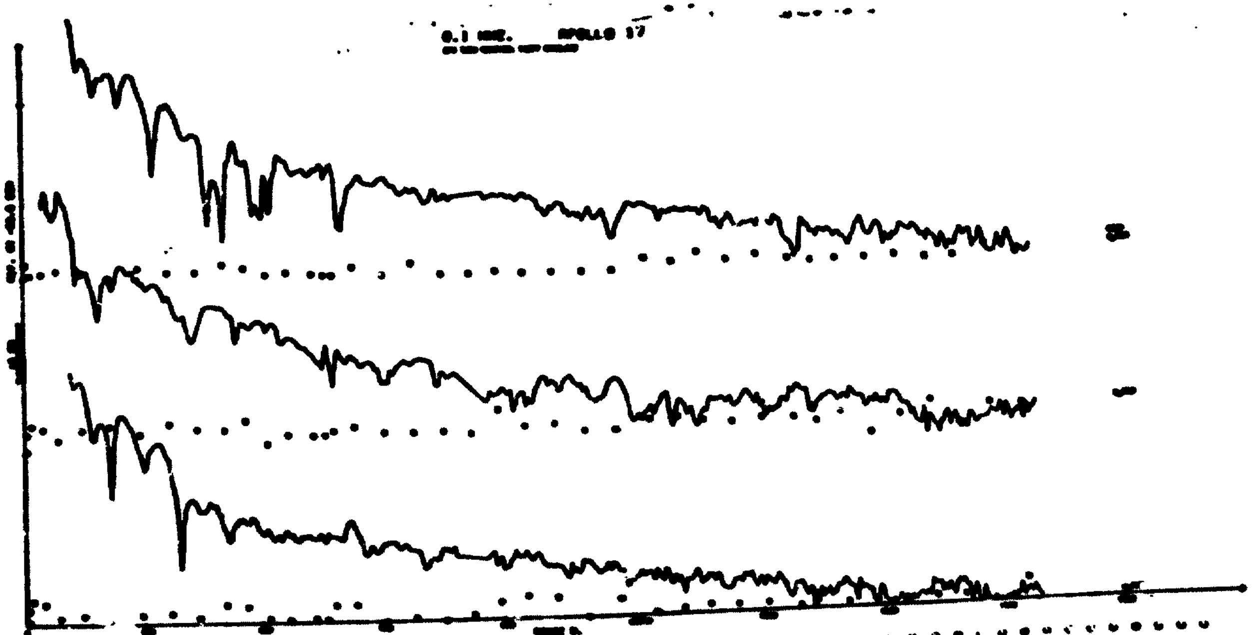

APOLLO 17

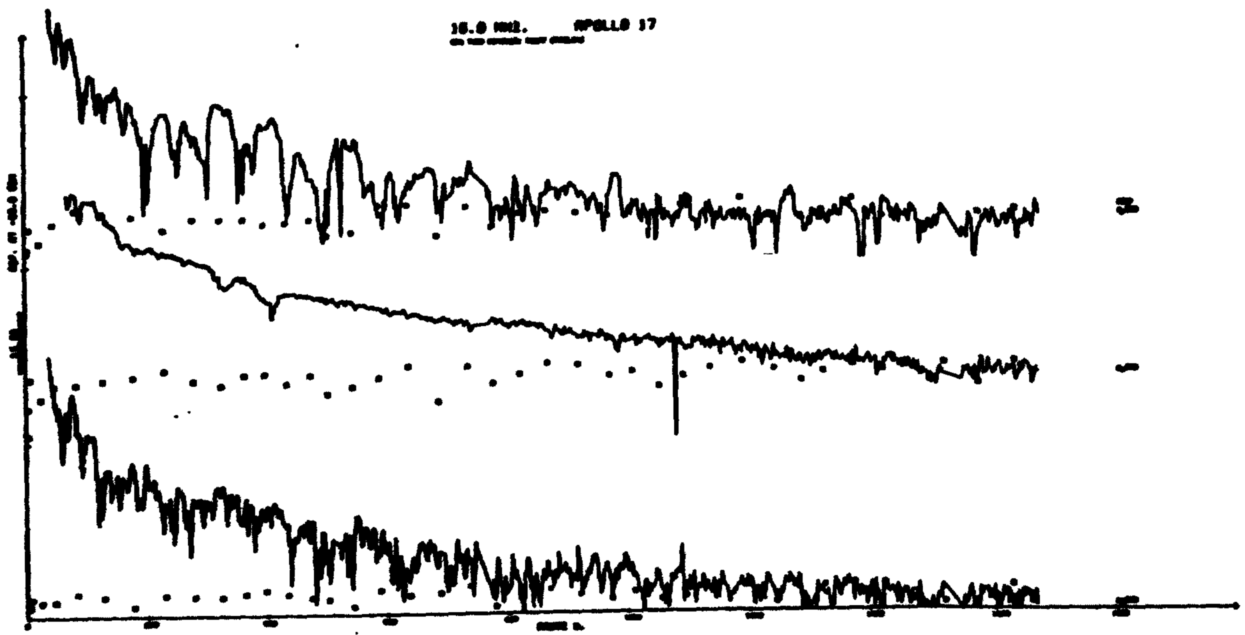

APOLLO 17

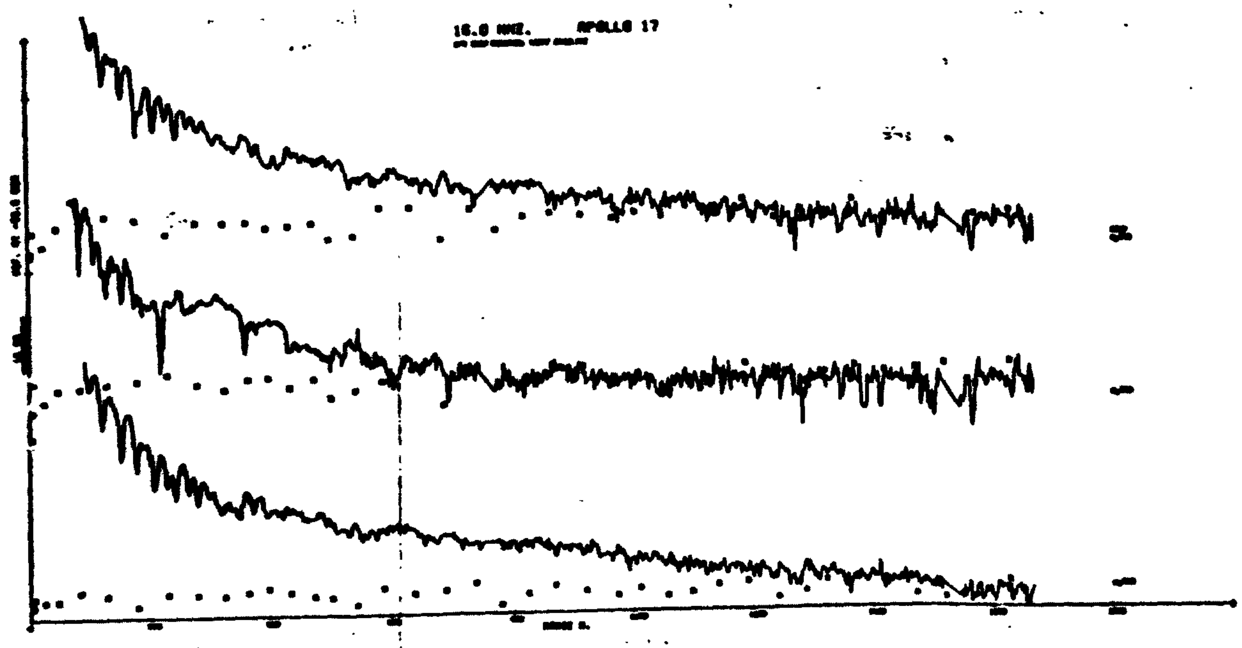

APOLLO 17

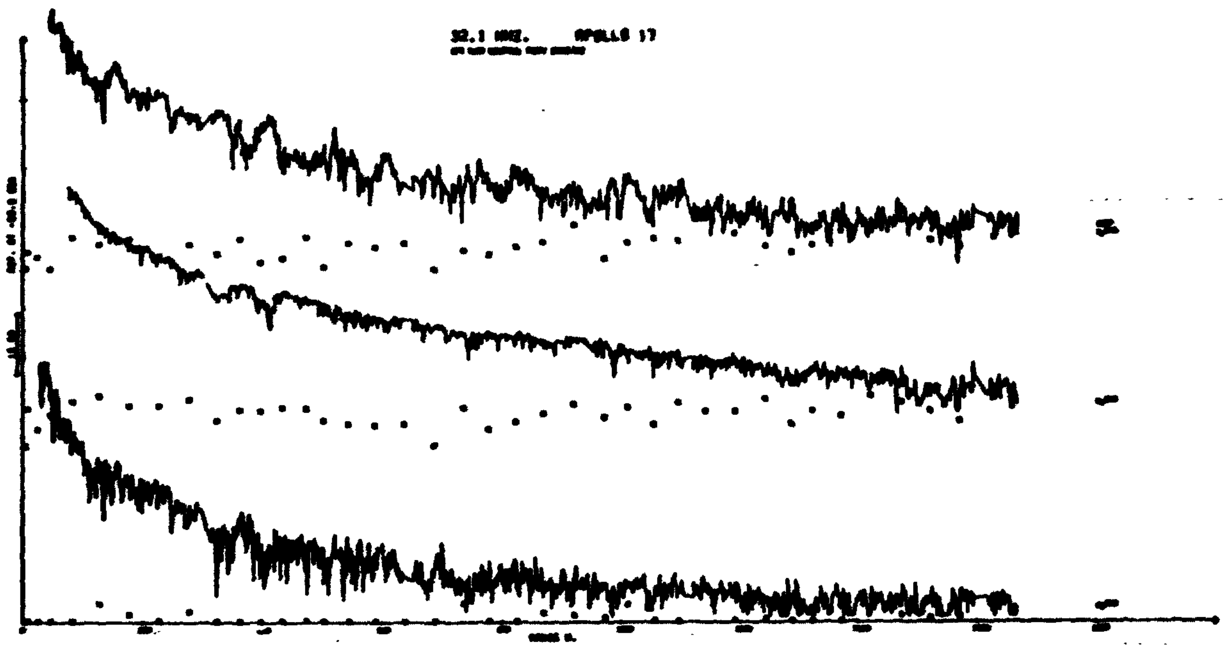

APOLLO 17

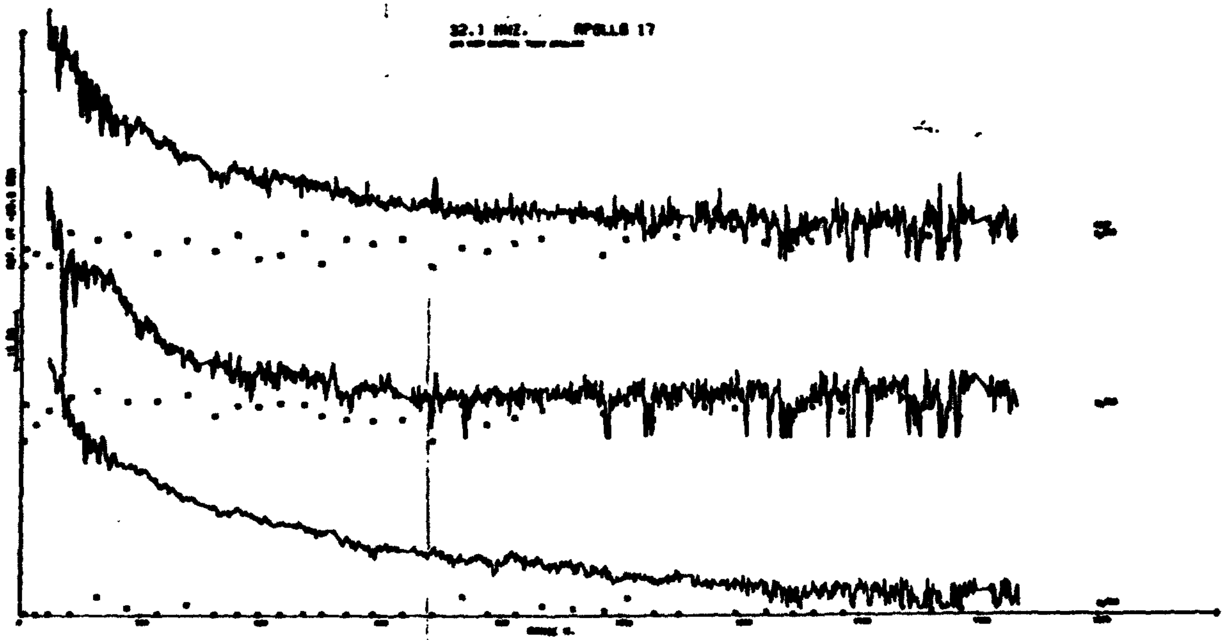

0.1 SEC. APOLLO 17


0.1 SEC. APOLLO 17


15.0 MHZ. APOLLO 17


15.0 MHZ. APOLLO 17


32.1 MHZ. APOLLO 17


32.1 MHZ. APOLLO 17

A-6

MEMORANDUM July 22, 1974

TO: Distribution

FROM: J. C. Rylaarsdam

SUBJECT: Comparison of SFP Range Data and Data from the VLBI Experiment

The VLBI data used for this study were obtained from tape number G2TMS (Goddard Space Flight Centre) as a set of x-y coordinate pairs; associated with the first pair in each group of five was a time (Grenwich Mean) expressed in hours, minutes, and seconds. These times were converted to seconds, and times for the four remaining pairs in each group were generated by adding values of one, two, three, and four seconds to the initial value. The x-y pairs were converted to distances.

The SFP data consisted of the sixteen megahertz range values from tape number SEPD99. The time (GM) corresponding to the first datum was set at 1407.4 seconds; times for succeeding values were generated by repeated addition of 0.81 second, the time interval between samples.

For each VLBI range datum, a corresponding SFP range value was computed by linear interpolation, using the two arrays of time data; the difference between these ranges was calculated as the VLBI range minus the interpolated SFP range. The differences obtained are plotted in figure 1; in this plot, the data are grouped into ten-second intervals, and the maximum and minimum differences over each interval are displayed. The mean of these differences was found to be -23.94 metres (indicating a lag in the VLBI data), and the standard deviation of the differences was effectively zero, considering the limits of precision of the calculations.

A more direct visual comparison is provided by figure 2, in which both sets of data are plotted on a single set of axes.

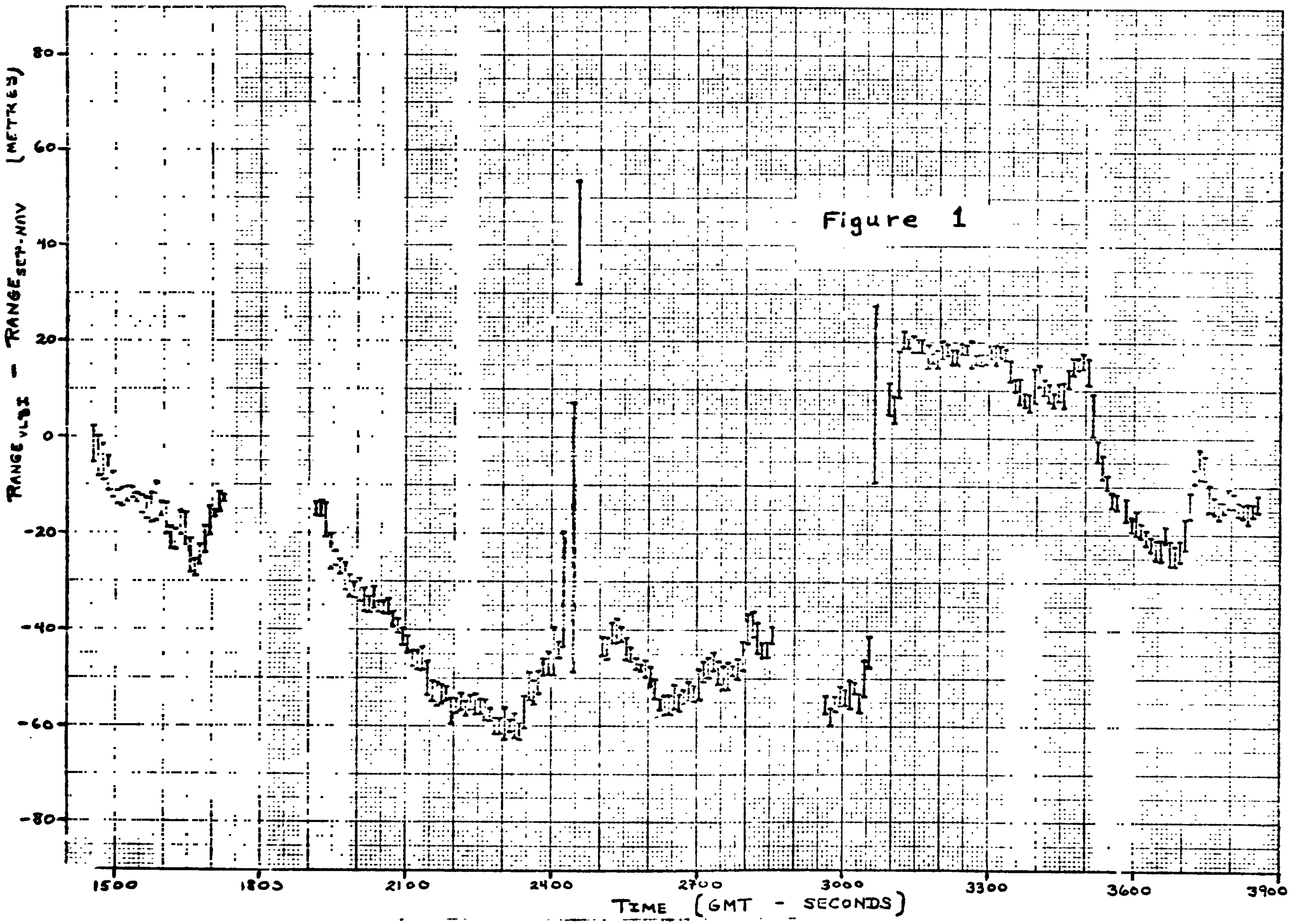


Figure 1
RANGE VLBI - RANGE SET-NAV (METRES)
80
60
40
20
0
-20
-40
-60
-80
1500
1800
2100
2400
2700
3000
3300
3600
3900
TIME (GMT - SECONDS)

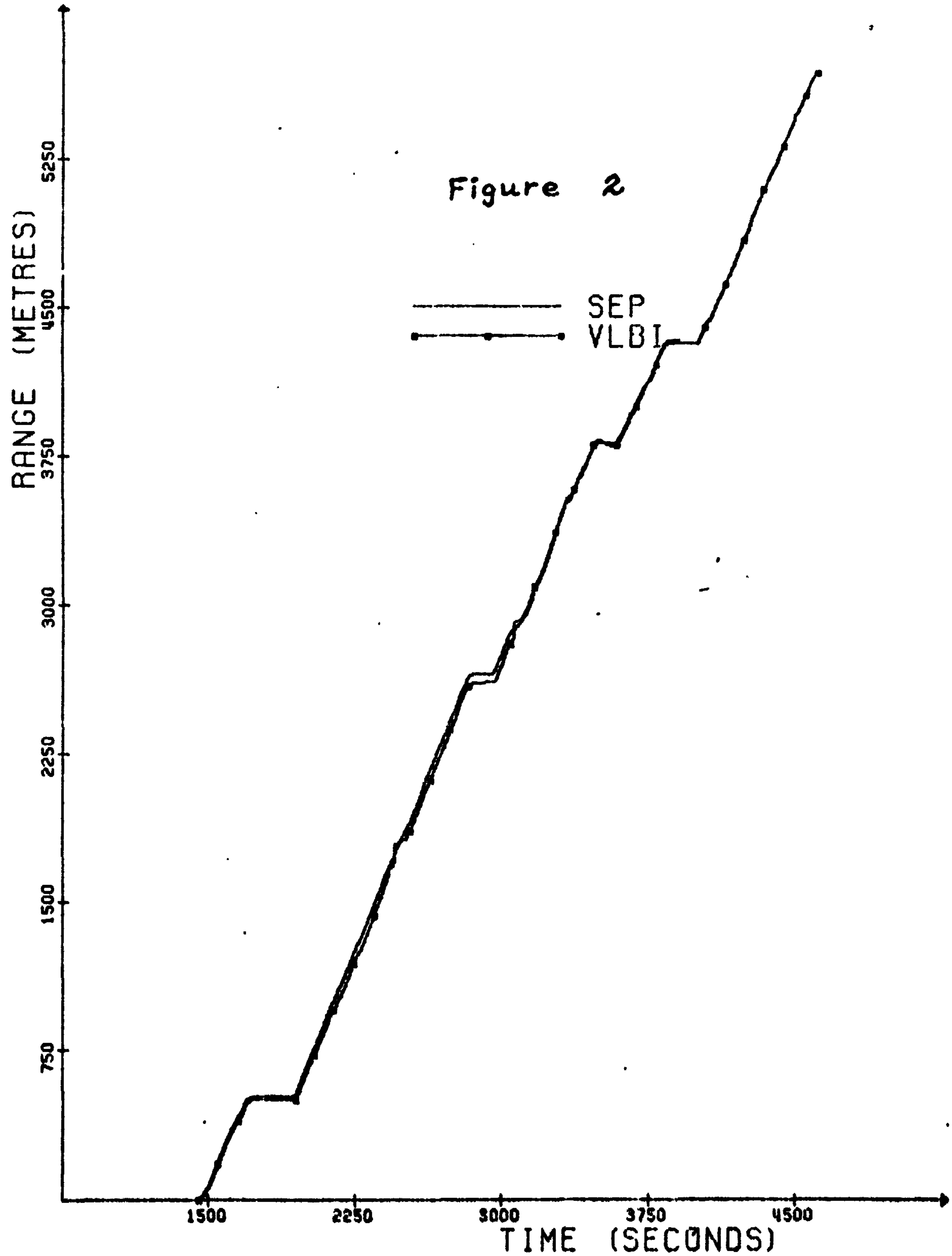

RANGE (METRES)
750
1500
2250
3000
3750
4500
5250
SEP
VLBI
1500
2250
3000
3750
4500
TIME (SECONDS)

Figure 2

A-6

July 22, 1974

MEMO TO: D.W. Strangway

FROM: James Rossiter

RE: SEP Antenna Patterns Reconstructed from EP-4 Turn

Introduction

During EVA II of Apollo 17, the Lunar Roving Vehicle (LRV) made a complete 360° turn around the deployment site of Seismic Explosion Package 4 (EP-4), about 525 m. from the SEP transmitter site. This turn provided an opportunity to estimate the directional characteristics of the three orthogonal SEP loop receiving antennas, as mounted on the LRV, over a dielectric earth.

Ideally, any signal received by the H_ρ (radial) antenna should smoothly interchange with the signal received by the H_ϕ (tangential) antenna as the LRV goes through each 90° of the turn. The H_z (vertical) signal should remain constant throughout the turn. If the turn were of zero radius, any deviations from the above could then be attributed to interference by the Rover and/or mount.

Data Reduction

Data were taken from Watt's lunar tape SEPDO9, which included the error noted in his memo (July 2, 1974). Details of the organization of the data after their removal from tape are given by Rylaarsdam ("Apollo 17 SEP Data Processing", July 1974). The following steps were then taken in order to get antenna patternplots.

(1) The values of all 36 components over the entire range of the turn (493 to 538 m. from the SEP site) were removed from Rylaarsdam's file SCI3 (which included no pre-processing of the turn by Watts). These values were stored in a new file called EP4, listed using program LUNACPY4, and plotted using program LUNAPLT4 (and routine GAPLOT). The points were spaced according to the time scale implicit in the data; an example is shown in Figure 1.

(2) Odometer counts (one count = 0.49 m. of wheel turn), received from both the right front and left rear wheels of the LRV, are available for each 1.02 seconds of the traverse (see memo by Redman, July 16/73). Ideally, given a high density of odometer pulses, and assuming no wheel slippage or sticking, LRV speed and rate of turn could be completely determined. However, the coarseness of the odometer pulses prevented this detailed reconstruction (see Figures 2 and 3). Antenna patterns plotted using the navigation data (see Rylaarsdam's report) were far less consistent from component to component than were those plotted assuming the LRV speed to be constant during the non-stationary portions of the turn.

Therefore a template with three pairs of bounds was set up to separate the points that were recorded while the LRV was actually turning from those recorded while the LRV was either on its traverse leg or stopped. By using components that had a good deal of character, the times during which the LRV was stationary were easily distinguished on the set of plots like Figure 1. The end-points of the turn were more difficult to estimate, and consistency from component to component was the only criterion available.

Unfortunately the 16 and 32 MHz data could not be used to construct the template, since both of these frequencies contain a drop-out due to Watt's spooling error during the turn.

(3) The total angle through which the LRV turned was calculated in the following way:

Assume there is no net slippage or sticking of either wheel over the turn. Then, for each wheel,

$$c = n\pi r, \qquad (1)$$

where c = the circumference of the turn made by the wheel
= total number of counts x 0.49 meters;
$n\pi$ = number of radians of the turn; and,
r = radius of the turn (m.).

Therefore,

$$n\pi = \frac{(c_o - c_i) \times 0.49}{r_o - r_i} \qquad (2)$$

where $c_o - c_i$ = the difference in odometer counts between the two wheels over the turn (see Figure 3); and,
$r_o - r_i$ = the distance between the two wheels
= 1.73 m. (Apollo 17 LRV Manual).

For the turn, $c_o - c_i = 21\pm2$,

therefore, $n\pi = 6.0\pm0.5$ radians.

Although this is evidently a fairly crude estimate, it indicates that the turn was close to 360°.

(4) The portions of file EP4 determined by step 2 to be actually in the turn proper were plotted as a function of angle using program ANTENNAO (and routine ANTPAT). A complete set of patterns is shown in Figure 4. The angles sta: t along the negative x-axis, and increment uniformly clockwise over 2π radians.

Discussion

Basically the plots show the expected type of behaviour. The vertical components are fairly smooth (except those which have very low signals), with few lobes, while the H_ρ and H_ϕ components do interchange. It must be pointed out that the 16 and 32 MHz plots do not contain any angular correction for the missing points, and this wil: certainly create some amount of distortion in the patterns.

Several of the plots do not align well with the north-south and east-west axes - e.g. 4MHz H_ϕ endfire. This is possibly due to an incorrect choice of either the bounds or of the total angle.

The major obstacles in obtaining good patterns from this analysis are as follows:

(1) Very poor sampling for the lower frequencies (as few as 8 points for a complete turn at 1 and 2 MHz), giving virtually no resolution of any lobe structure.

(2) Non-constant range of the LRV through the turn for the higher frequencies. The turn had a diameter of approximately 15 m - or about 1.5 wavelengths at 32 MHz. Therefore the signal received during the turn could have changed substantially quite independently of LRV rotation.

(3) Lack of direct knowledge of a) the exact position of the turn in the data stream, b) the complete angle of rotation, or c) the speed of rotation. These could only be estimated, and compatibility from component to component used to improve the estimate. The problem was particularly severe because of the drop-outs at 16 and 32 MHz.

(4) Unknown source signal. It is evidently not a plane wave, since the SEP transmitter was used. Reflections and scattering from the subsurface may well have had important influences on the type of pattern.

Conclusions

Considering the above problems, the amount of distortion of the patterns is within the error of the analysis. H_z appears non-directional at all frequencies; H_ρ and H_ϕ interchange smoothly through the turn. It is therefore not possible to attribute any large degree of interference to the

LRV or mount. This does not imply that such interferences did not exist - only that this analysis was not able to detect it.

It would probably be worth while in the future to analyse the data without the drop-outs at 16 and 32 MHz. These frequencies have both the highest resolution and are most likely to be susceptible to interference from the LRV or mount. A systematic attempt to use a number of different possible bounds and rotation angles may locate the turn in the data stream better. If so, such a study could be more definitive.

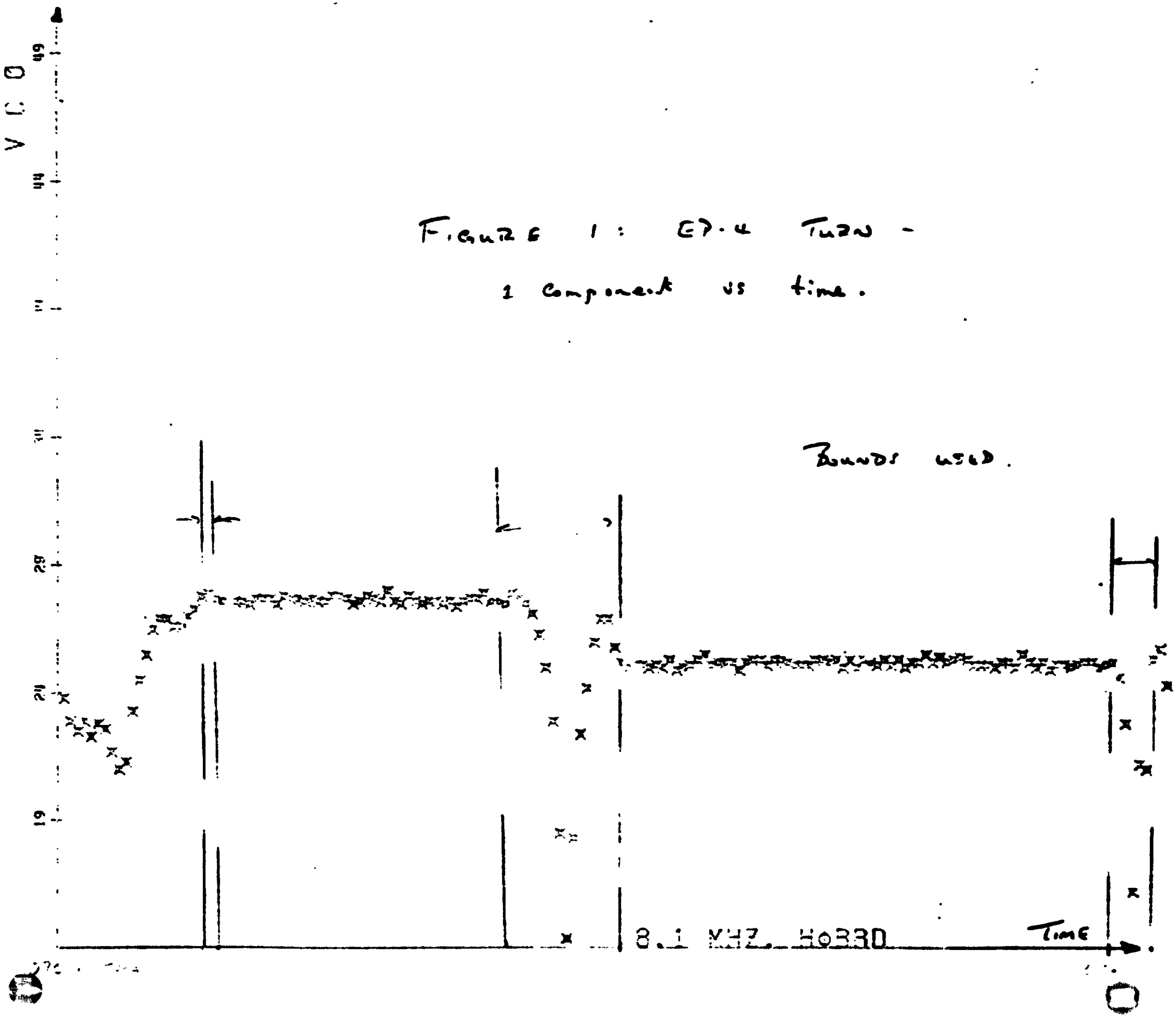
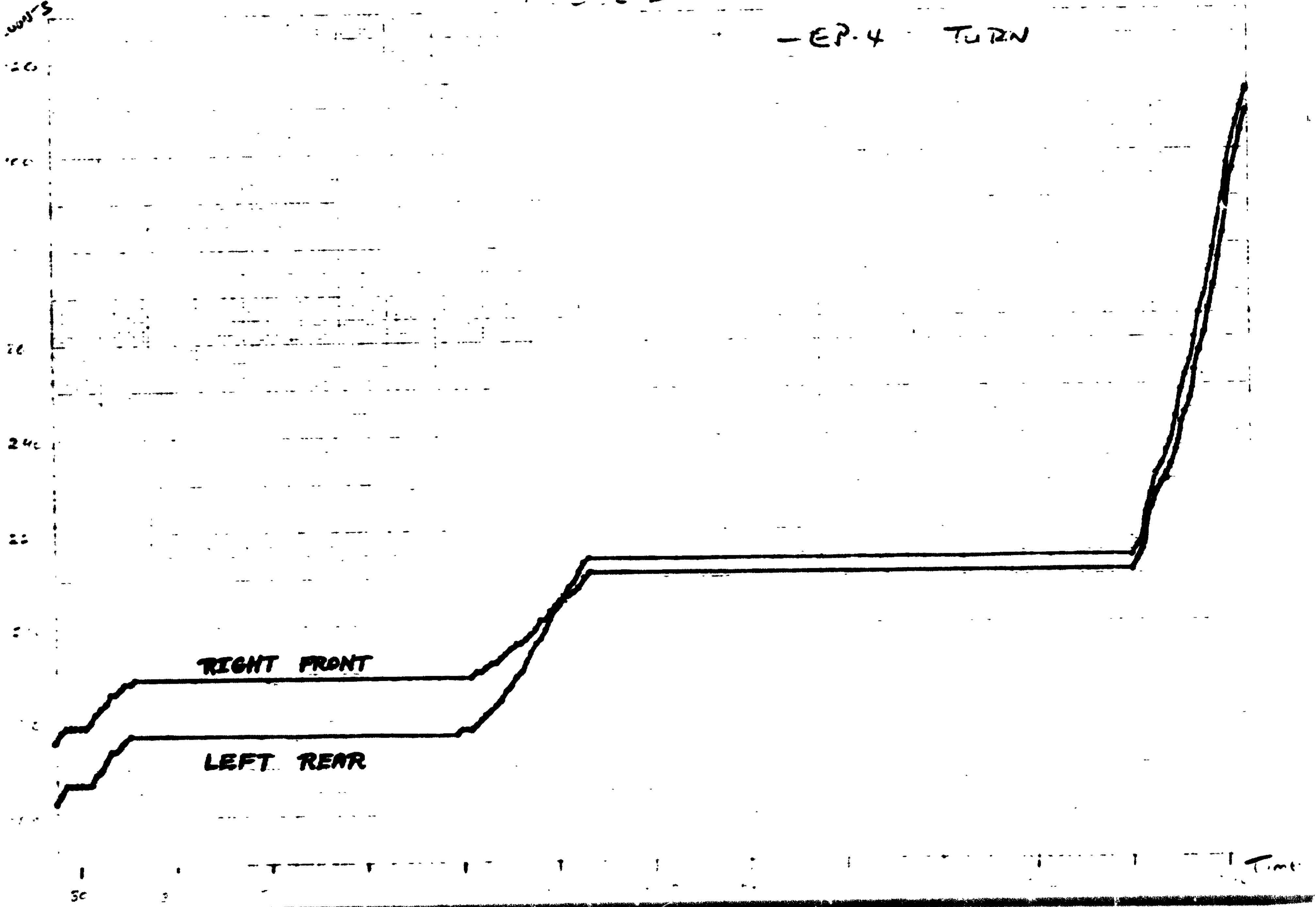


Figure 1: E7.4 Turn -
1 component vs time.

- EP. 4 TURN



LRV WHEELS - E7-6 TURN.

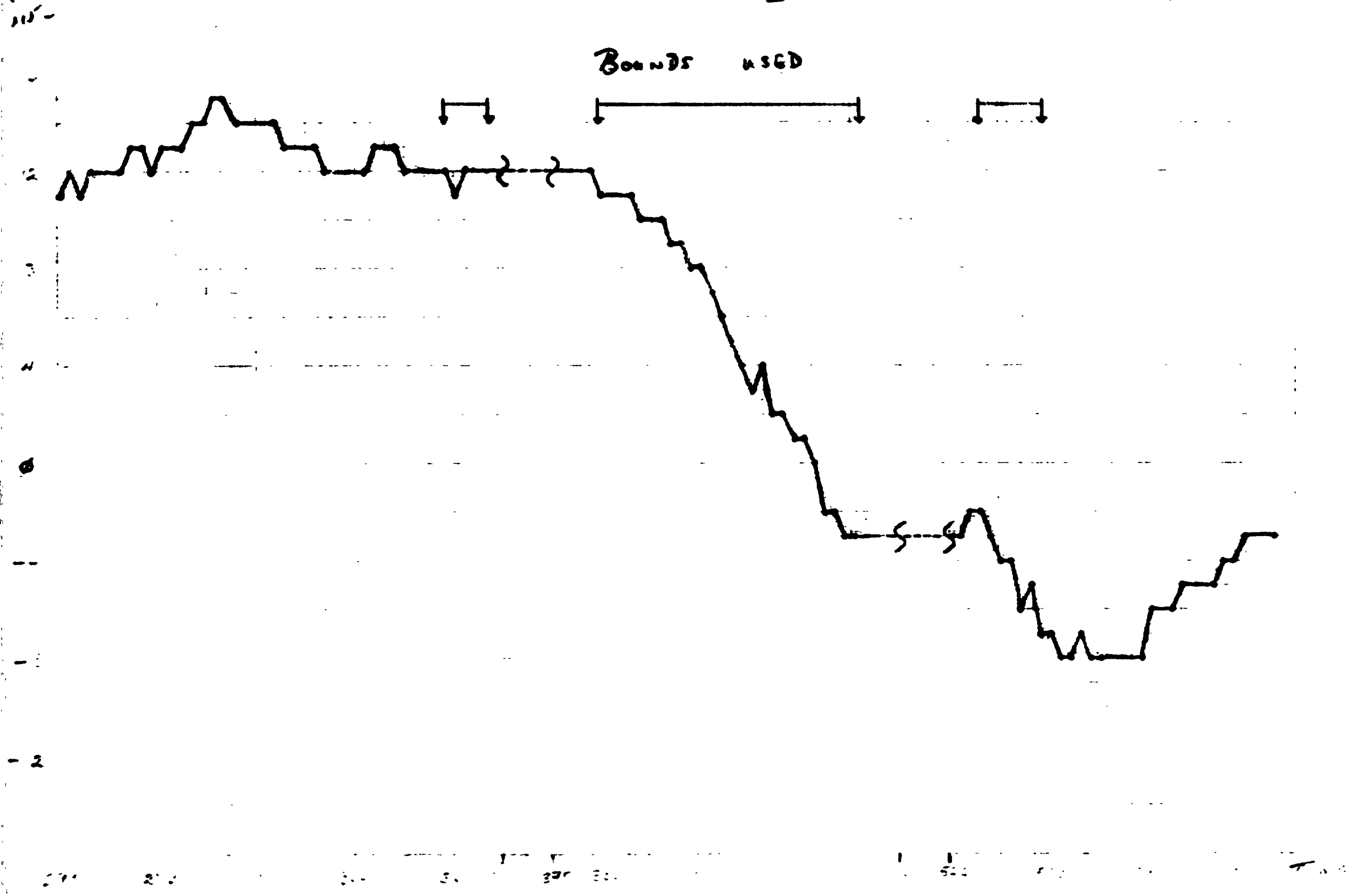
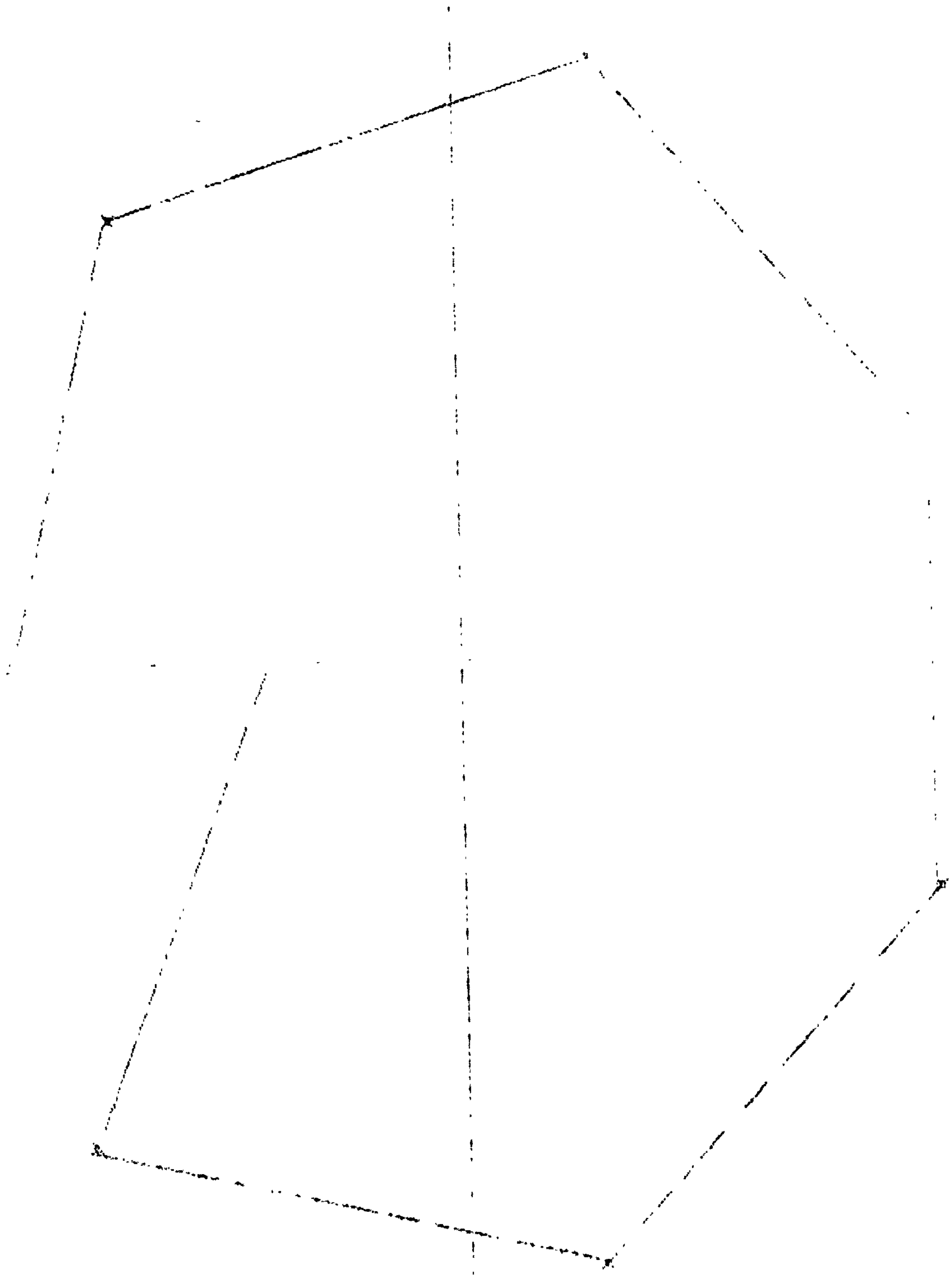
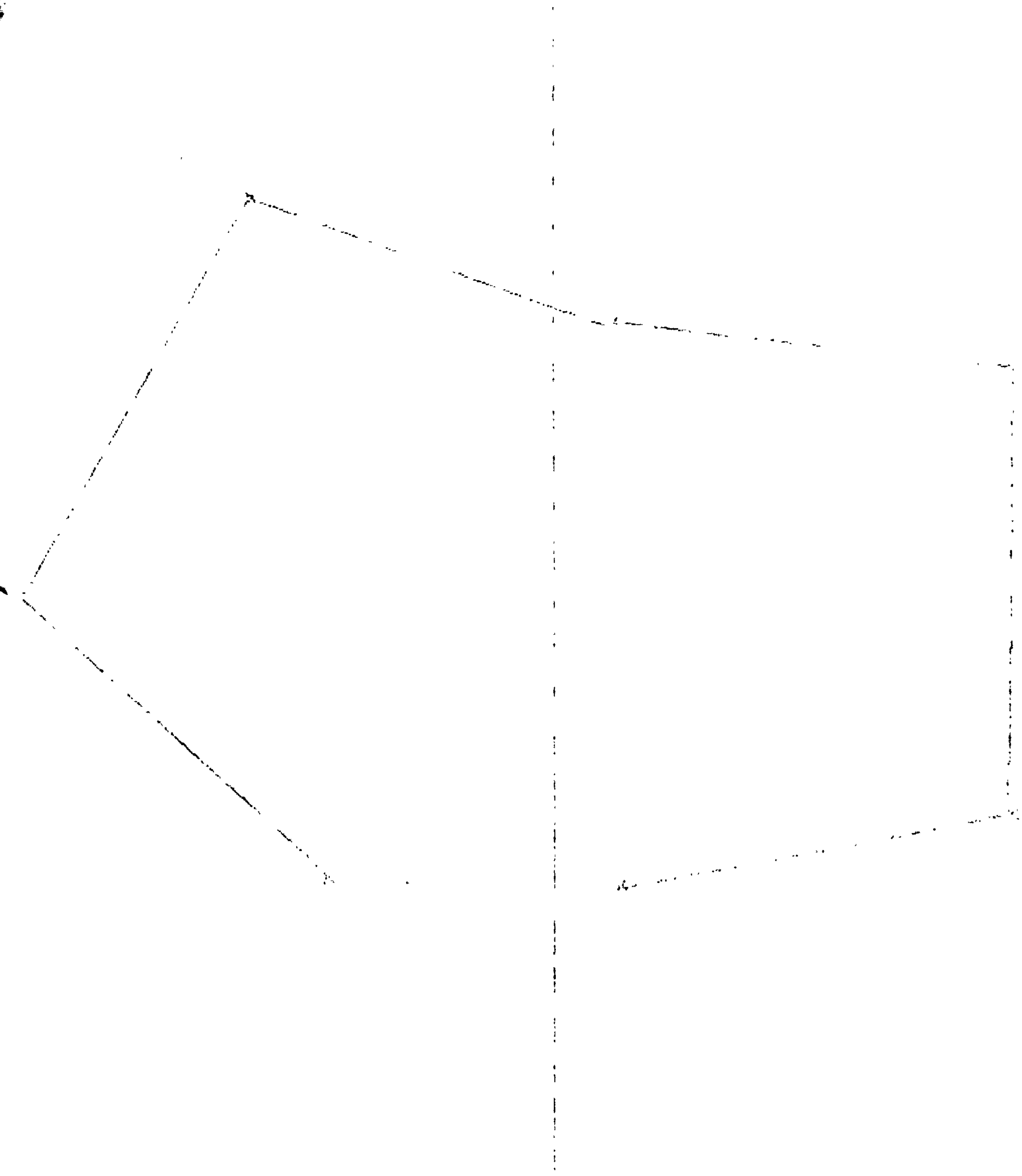


Figure 4. SEP antenna radiation patterns from EP-4 turn for all components. Since choice of the exact position of the turn is somewhat arbitrary (see text), these patterns are only approximate.

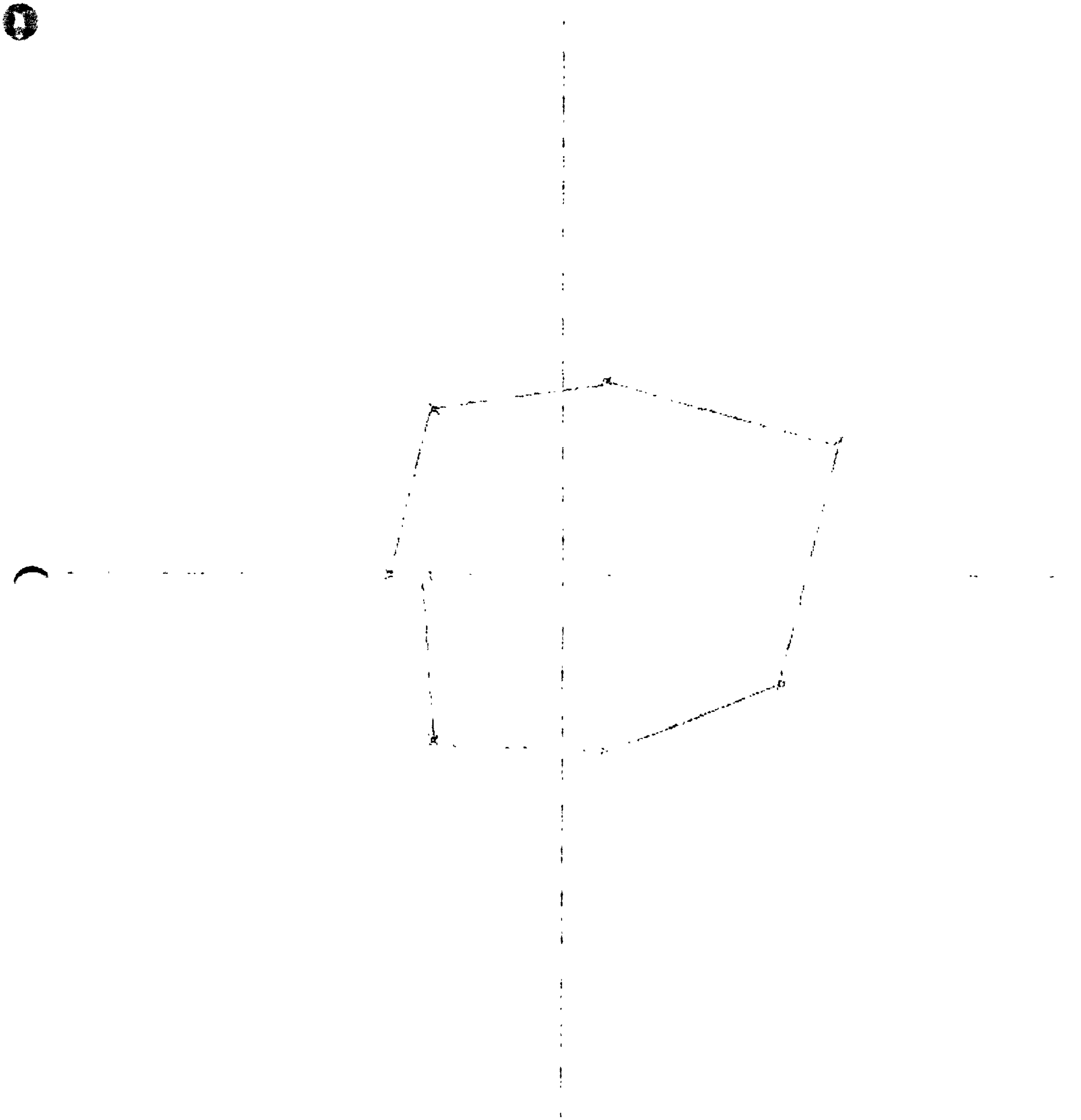
16 and 32 MHz each suffer a 13-point data dropout during the turn in these plots. This has not been corrected for in any way.



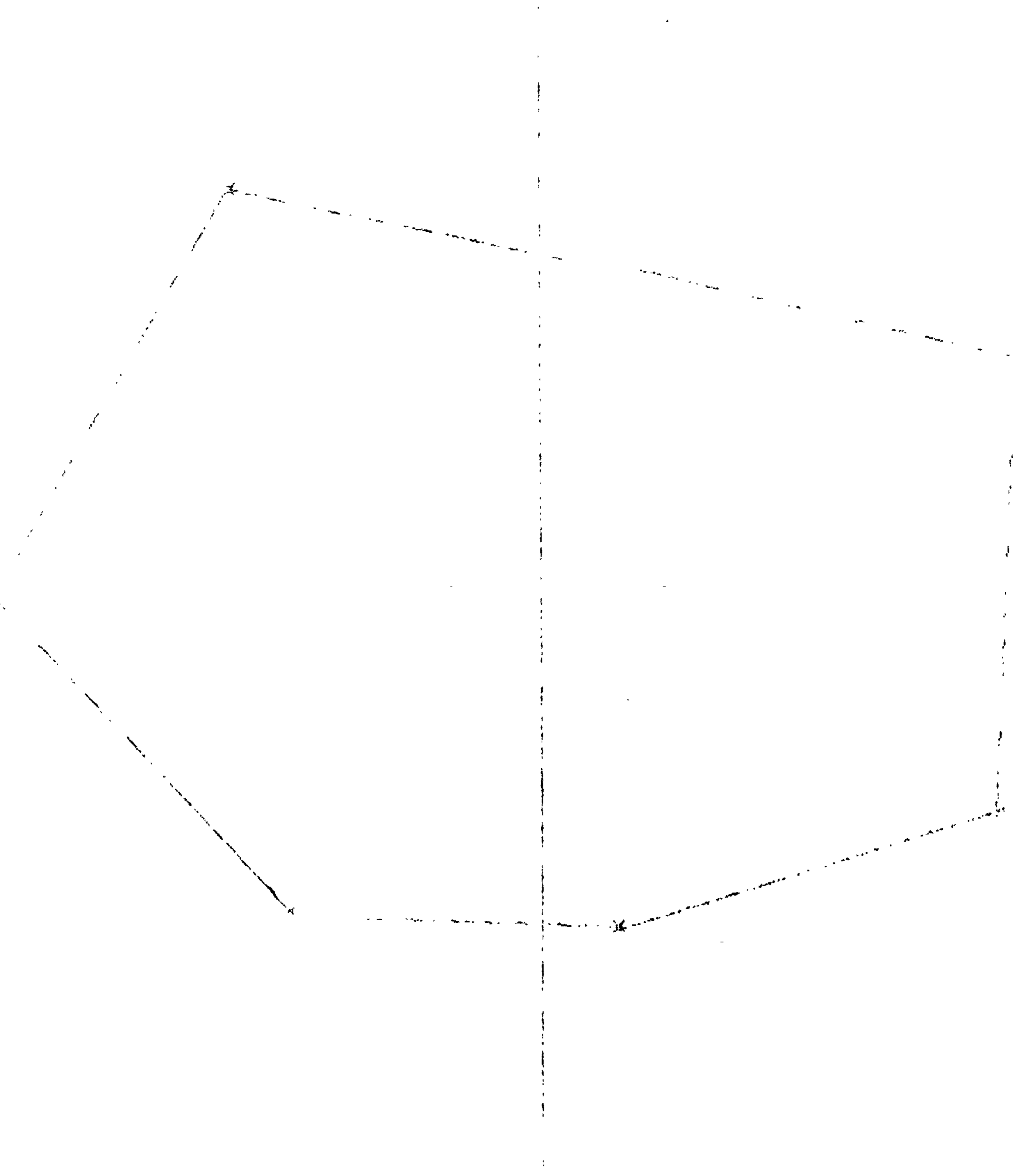
1.0 MHZ. BEND



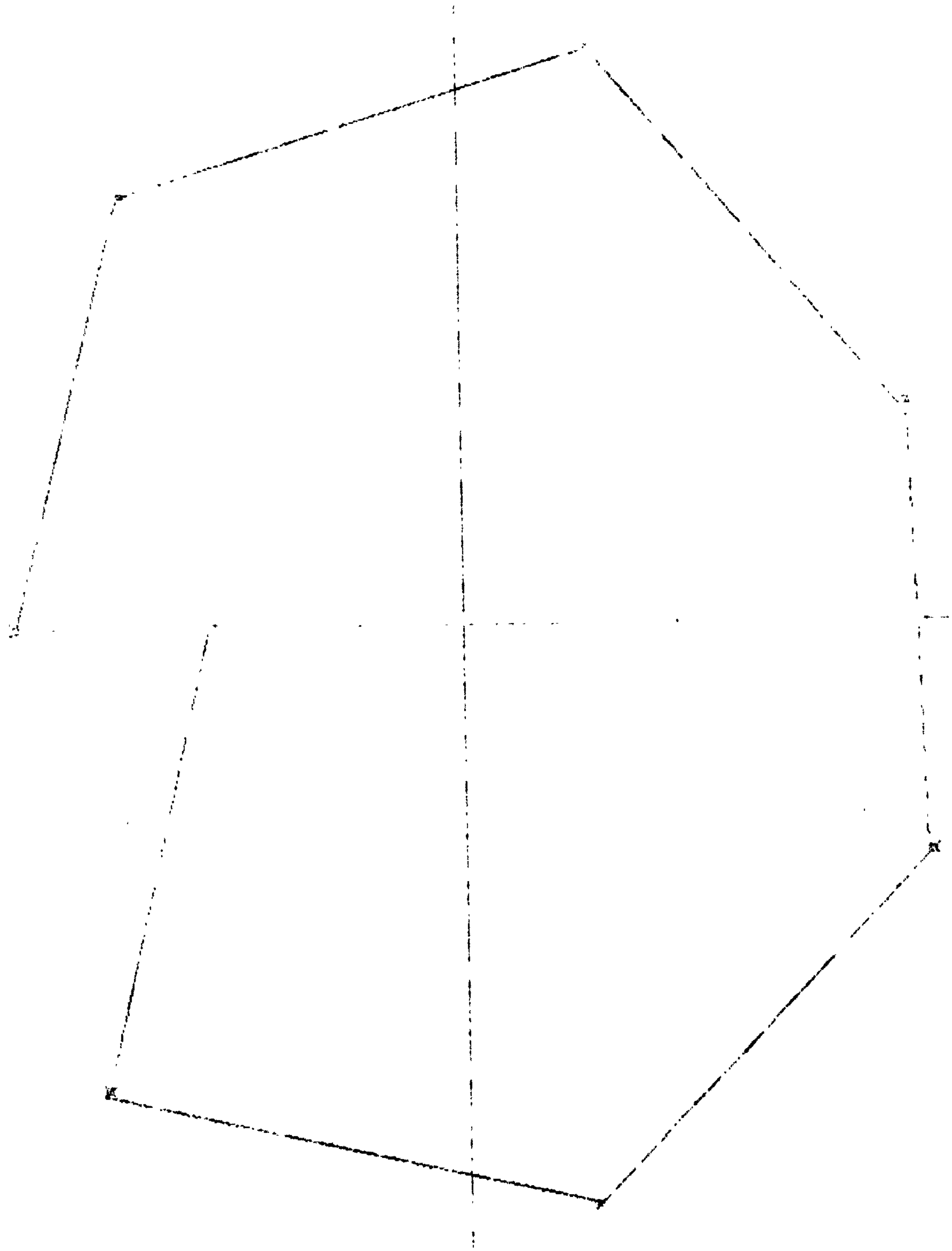
1.0 MHZ. HΦEND



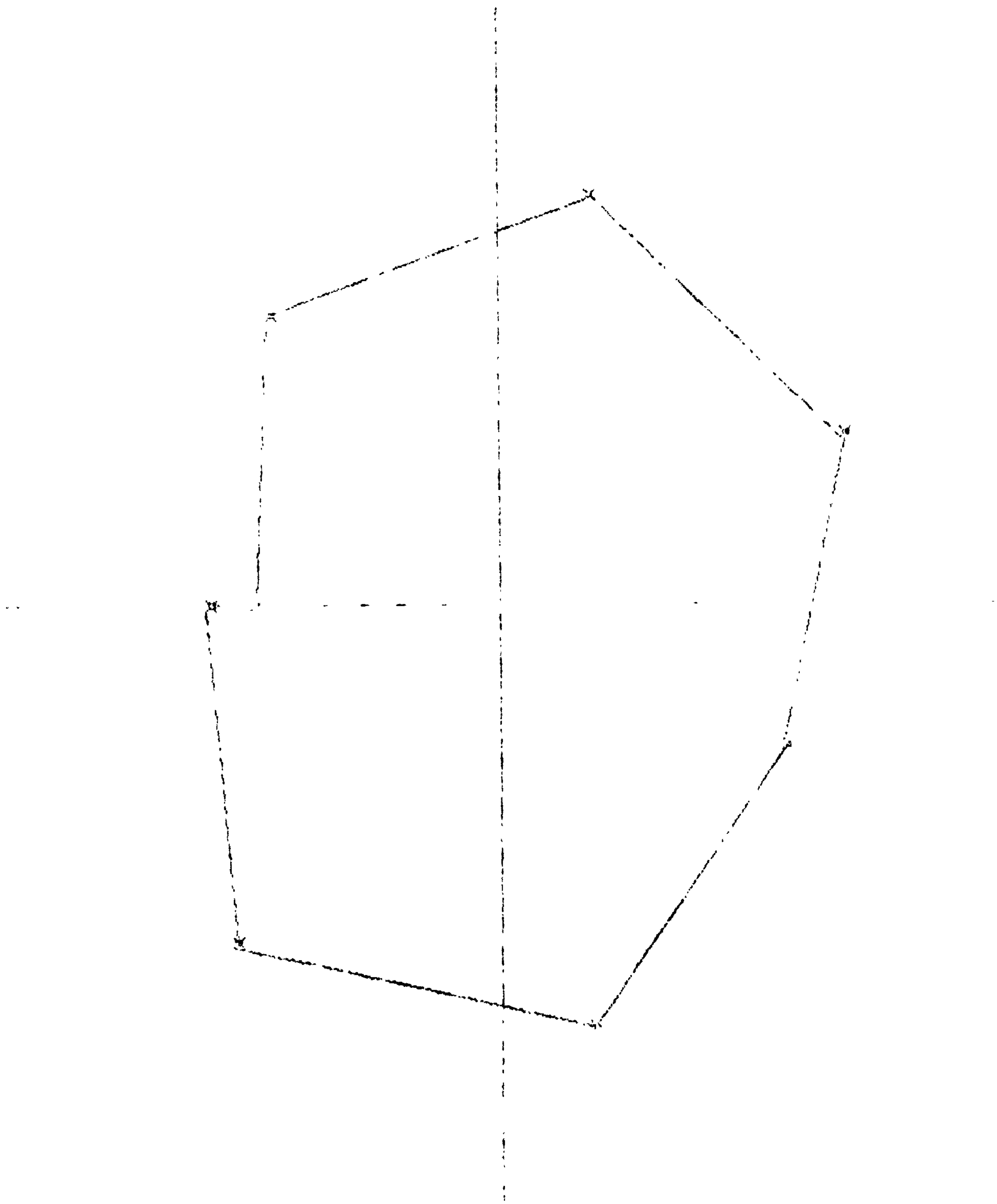
1.0 MHZ. HZENB



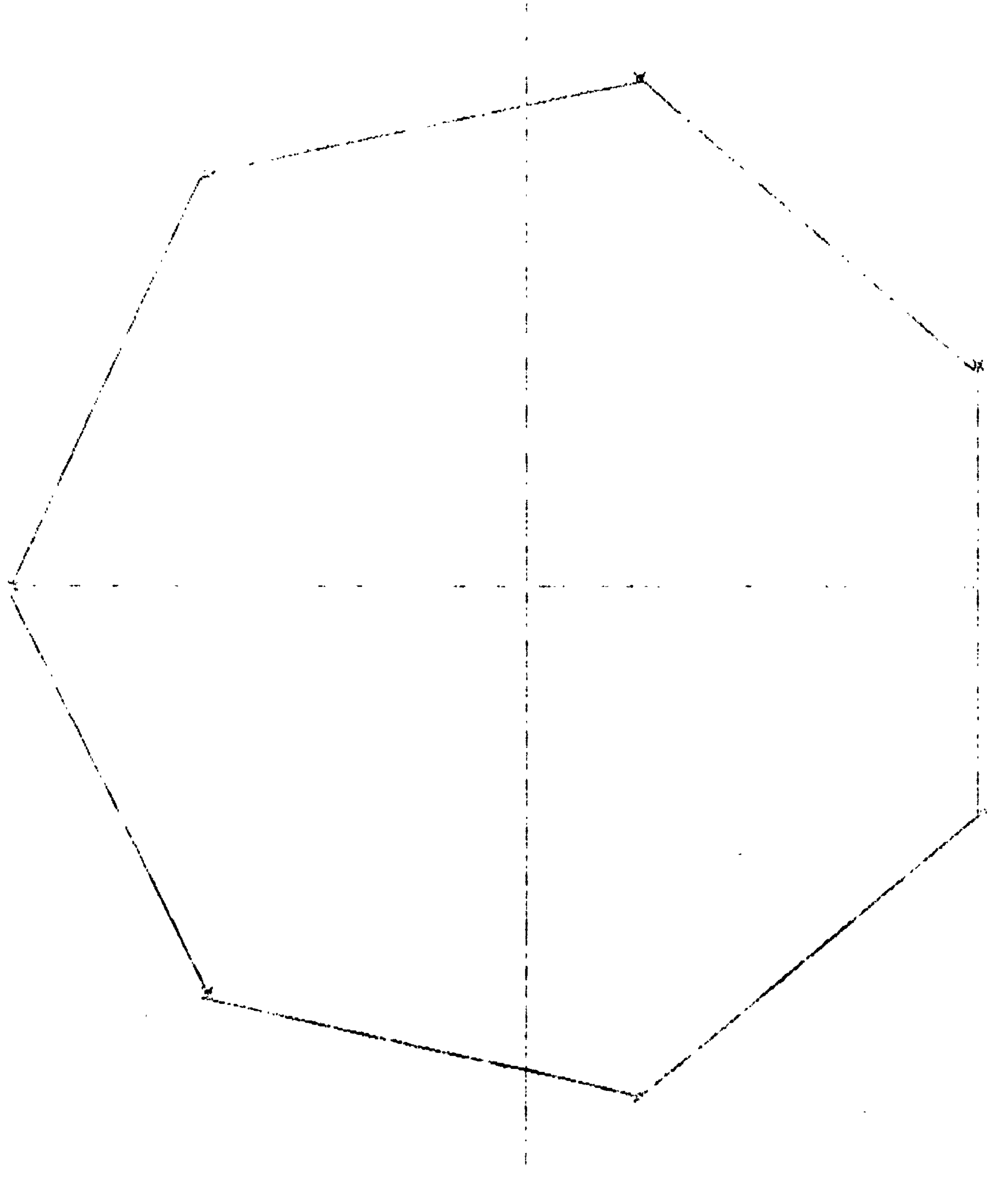
1.0 MHZ. HRSPD



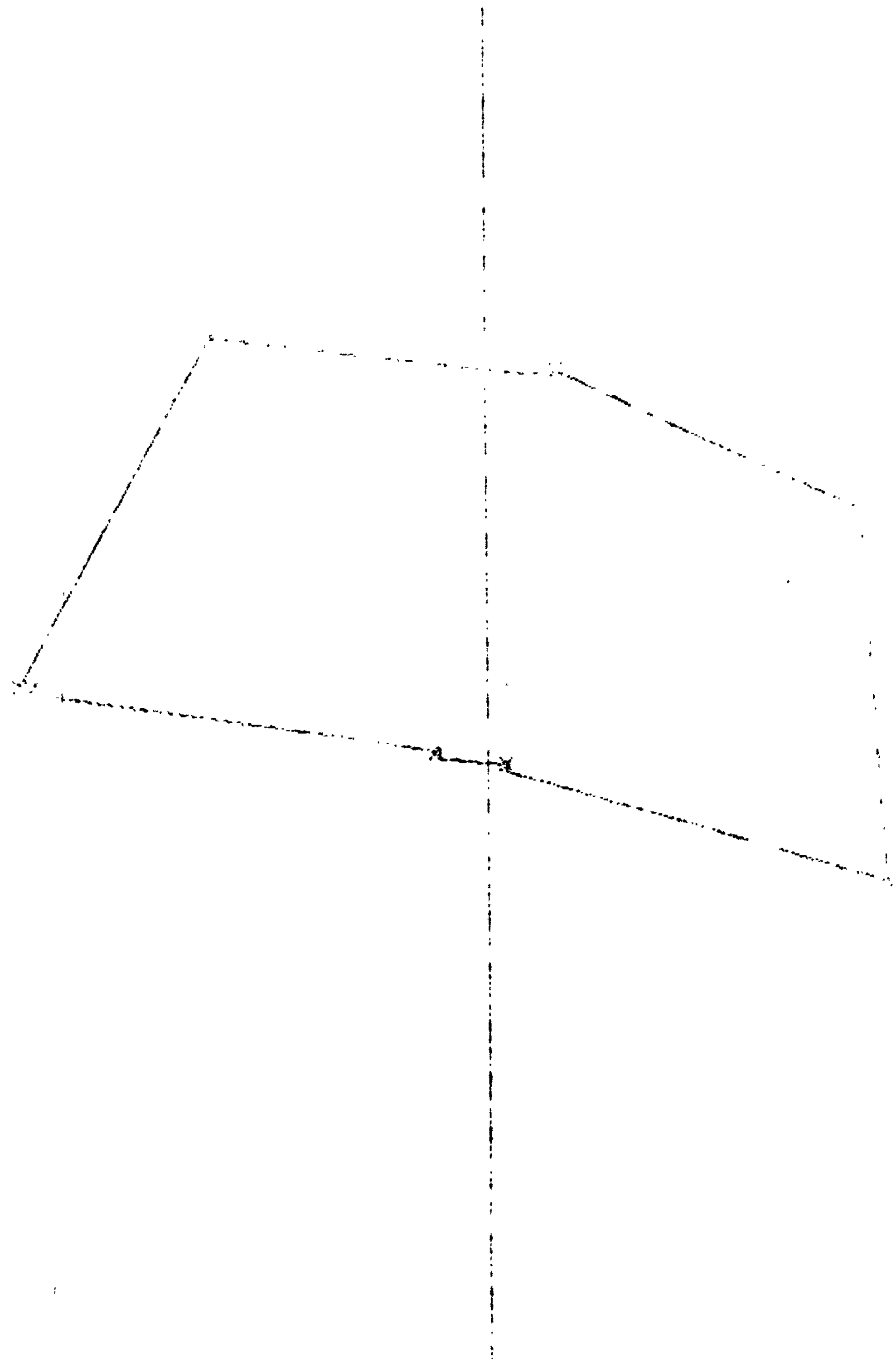
1.0 MHZ. HΦBRD



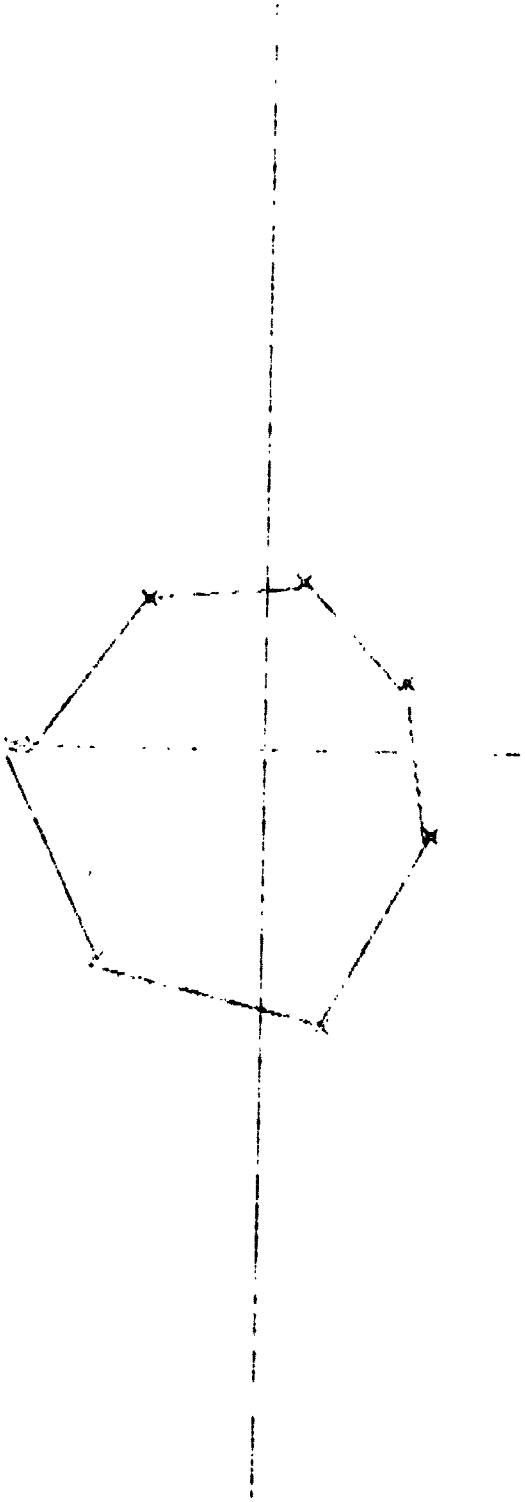
2.1 MHZ. HPEND



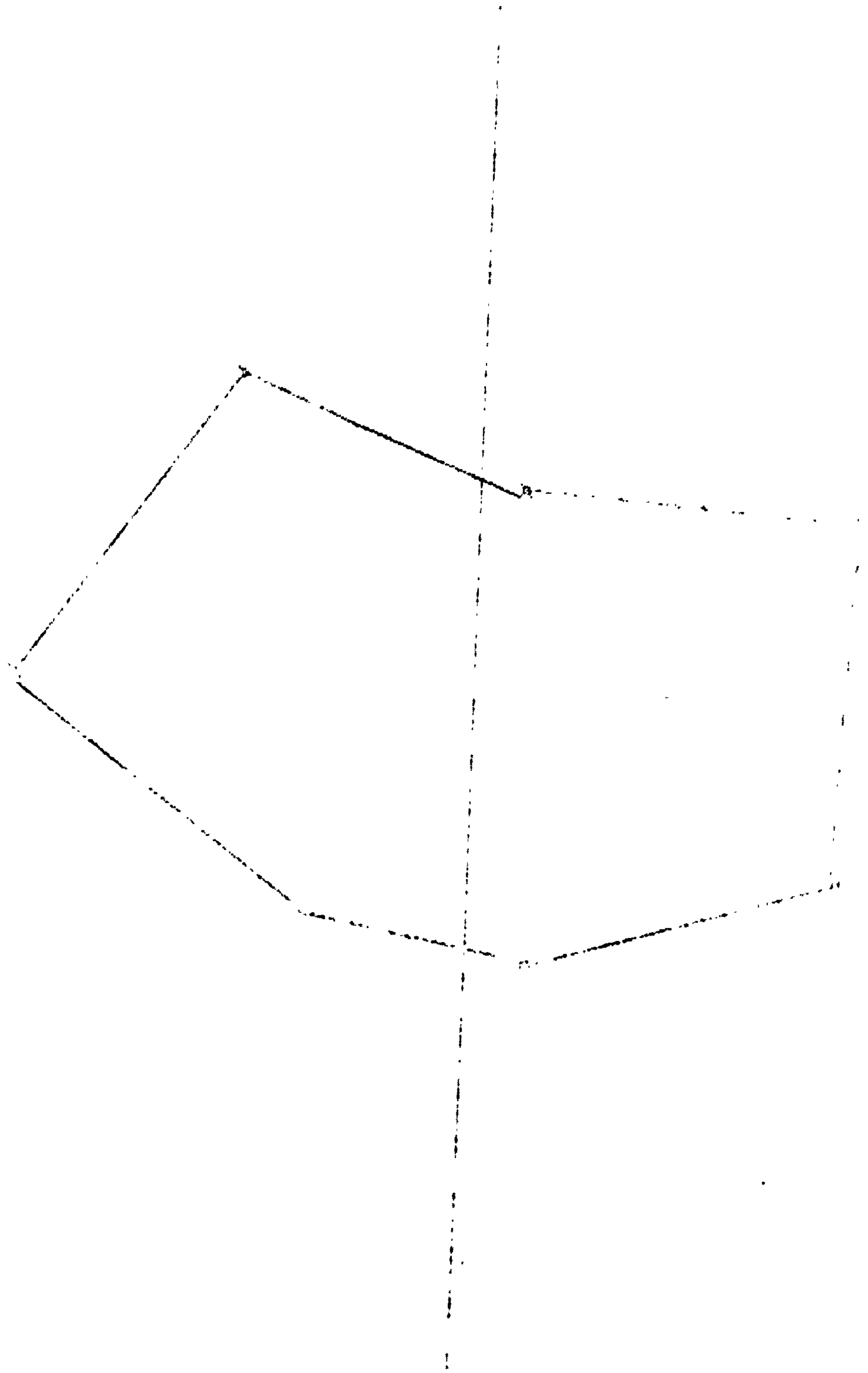
1.0 MHZ. HZBRD



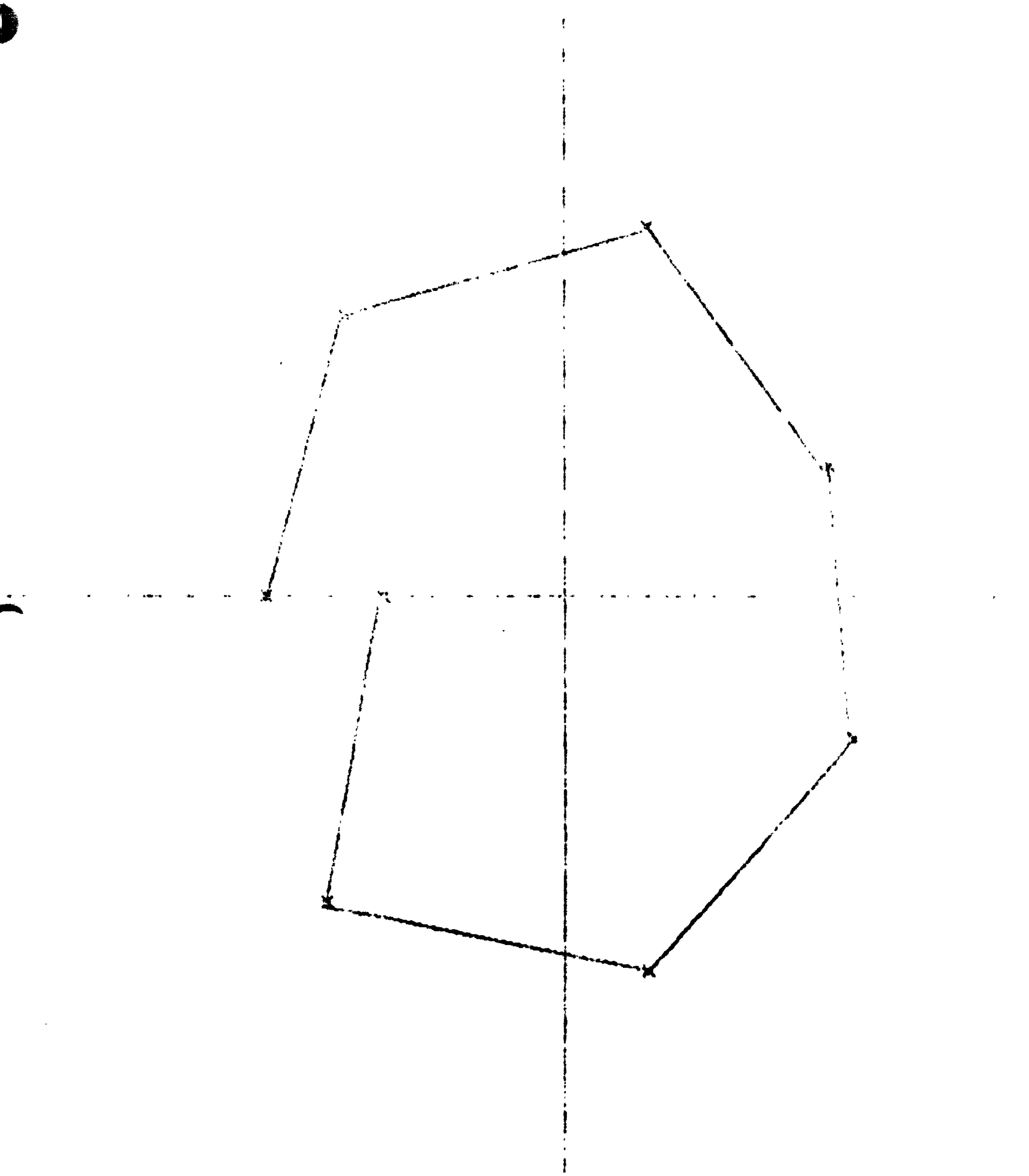
2.1 MHZ. HOENT



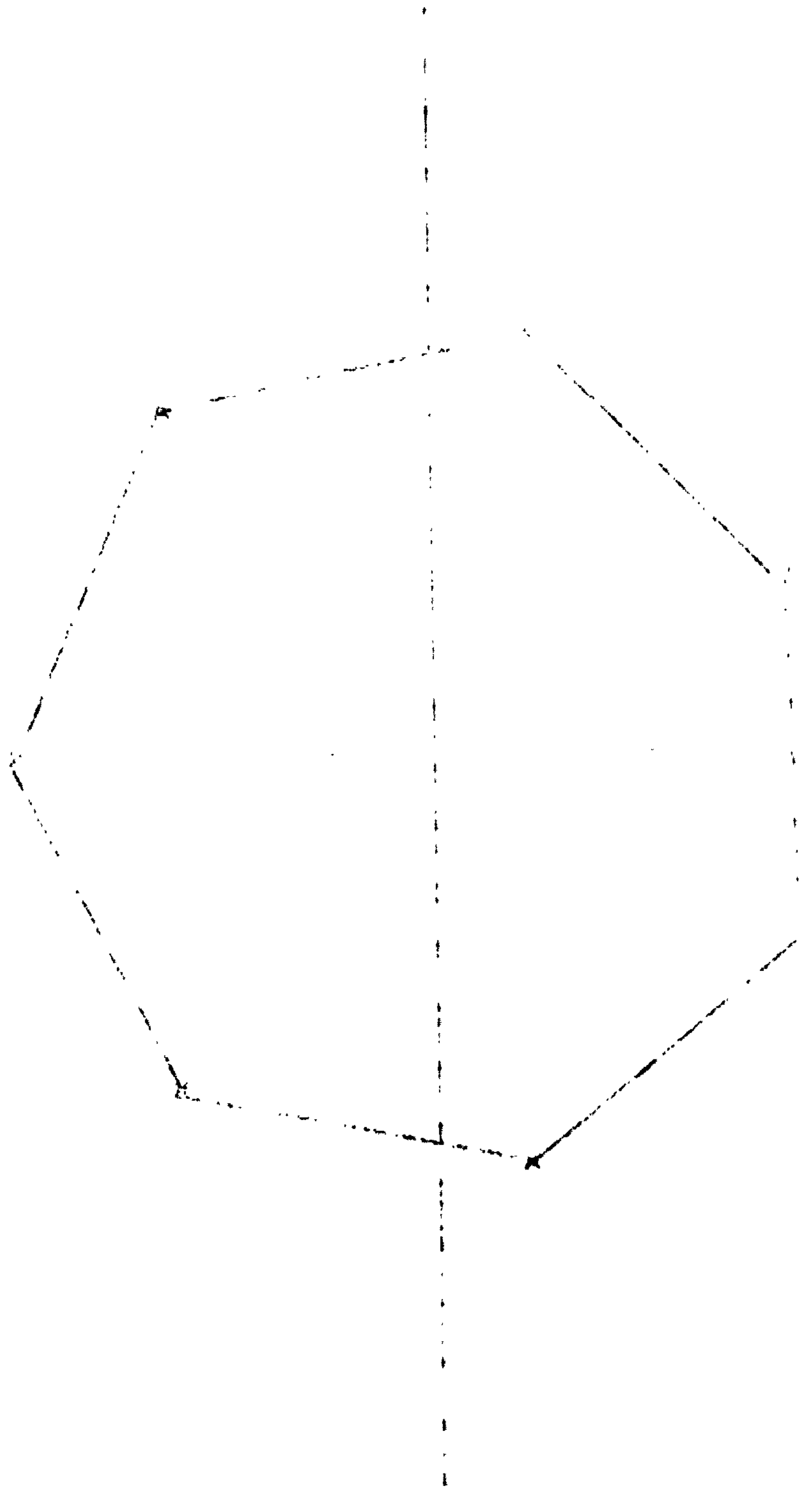
2.1 MHZ. HZEND



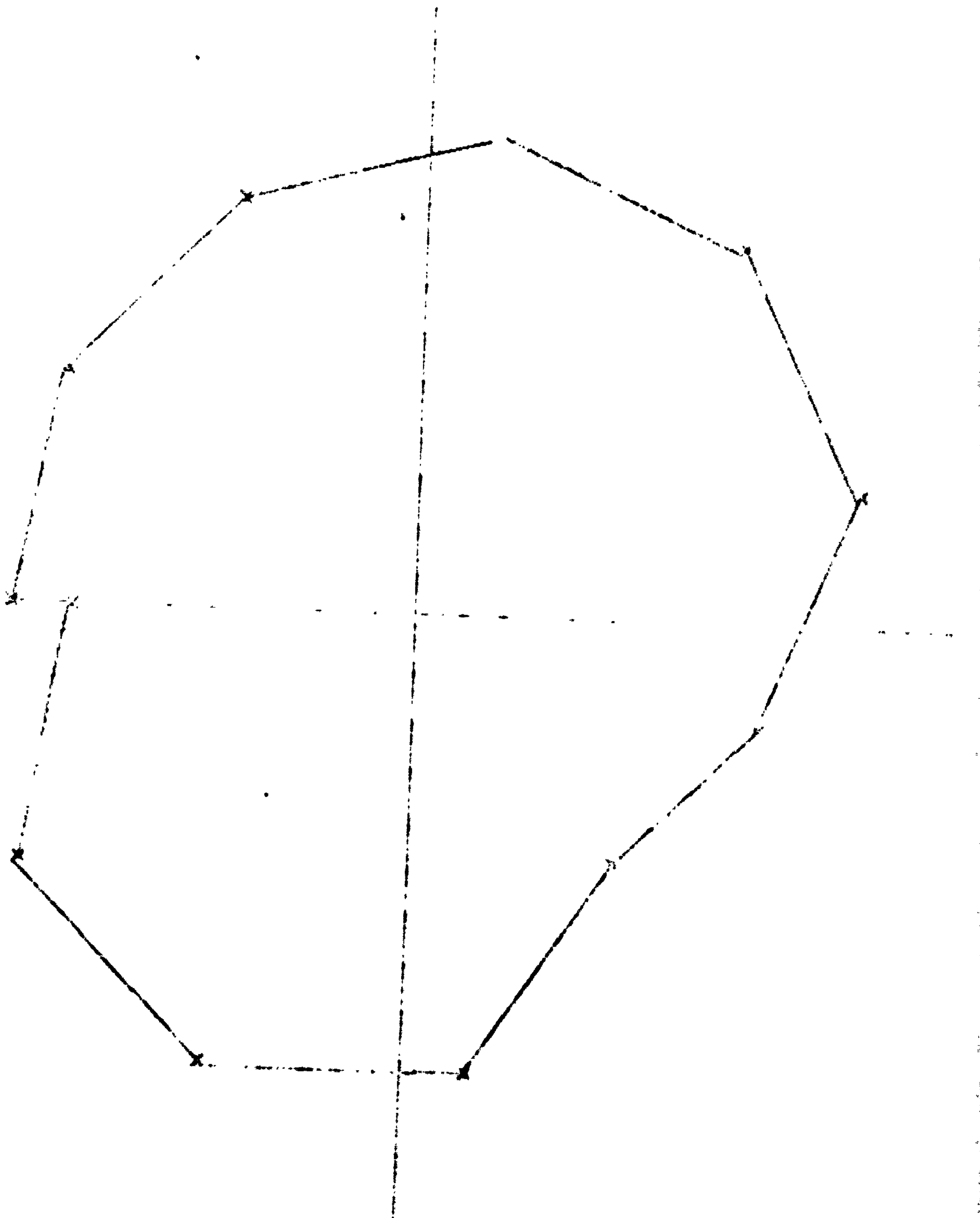
2.1 MHZ. HPDED



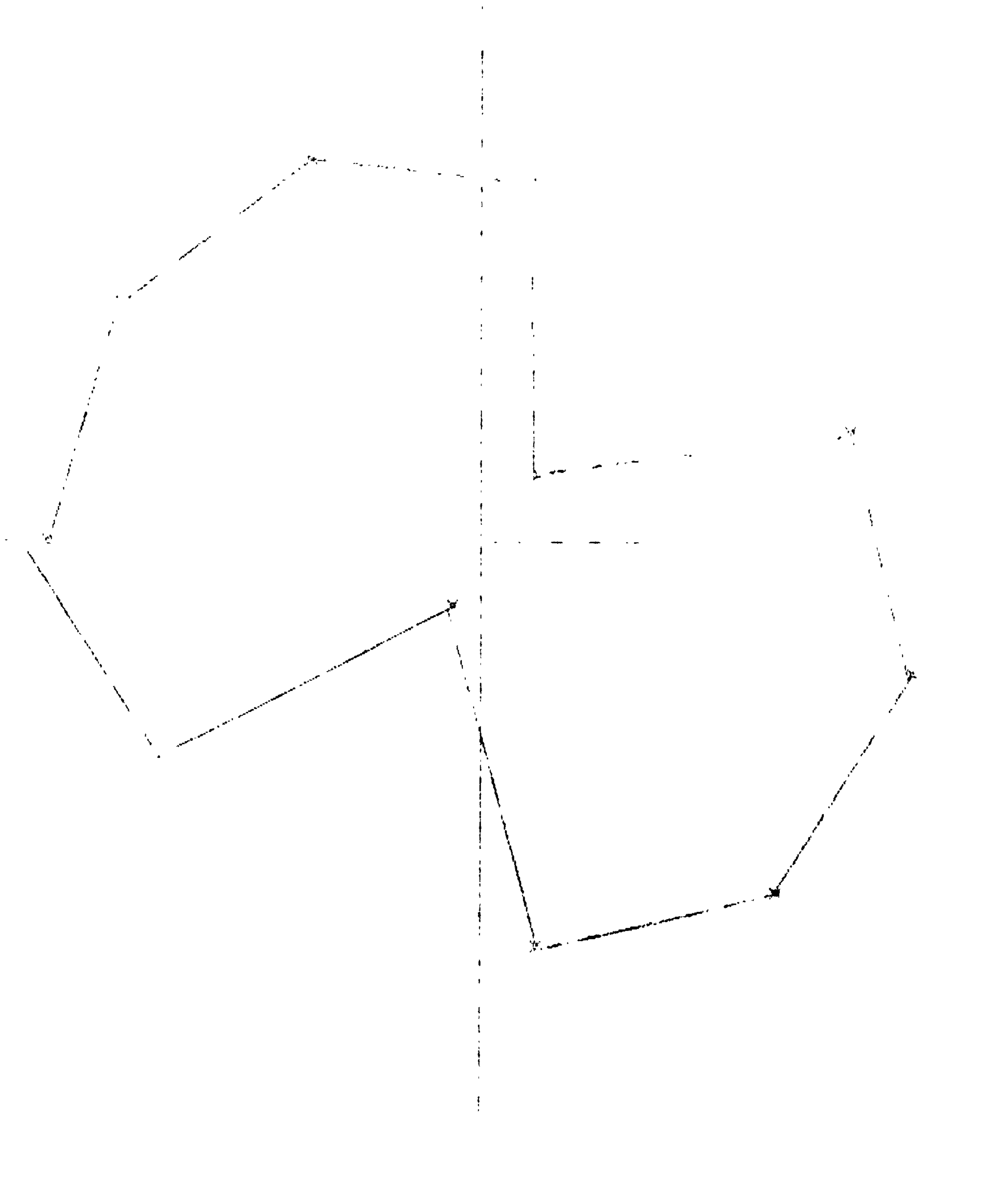
2.1 MHZ. HΦBRD



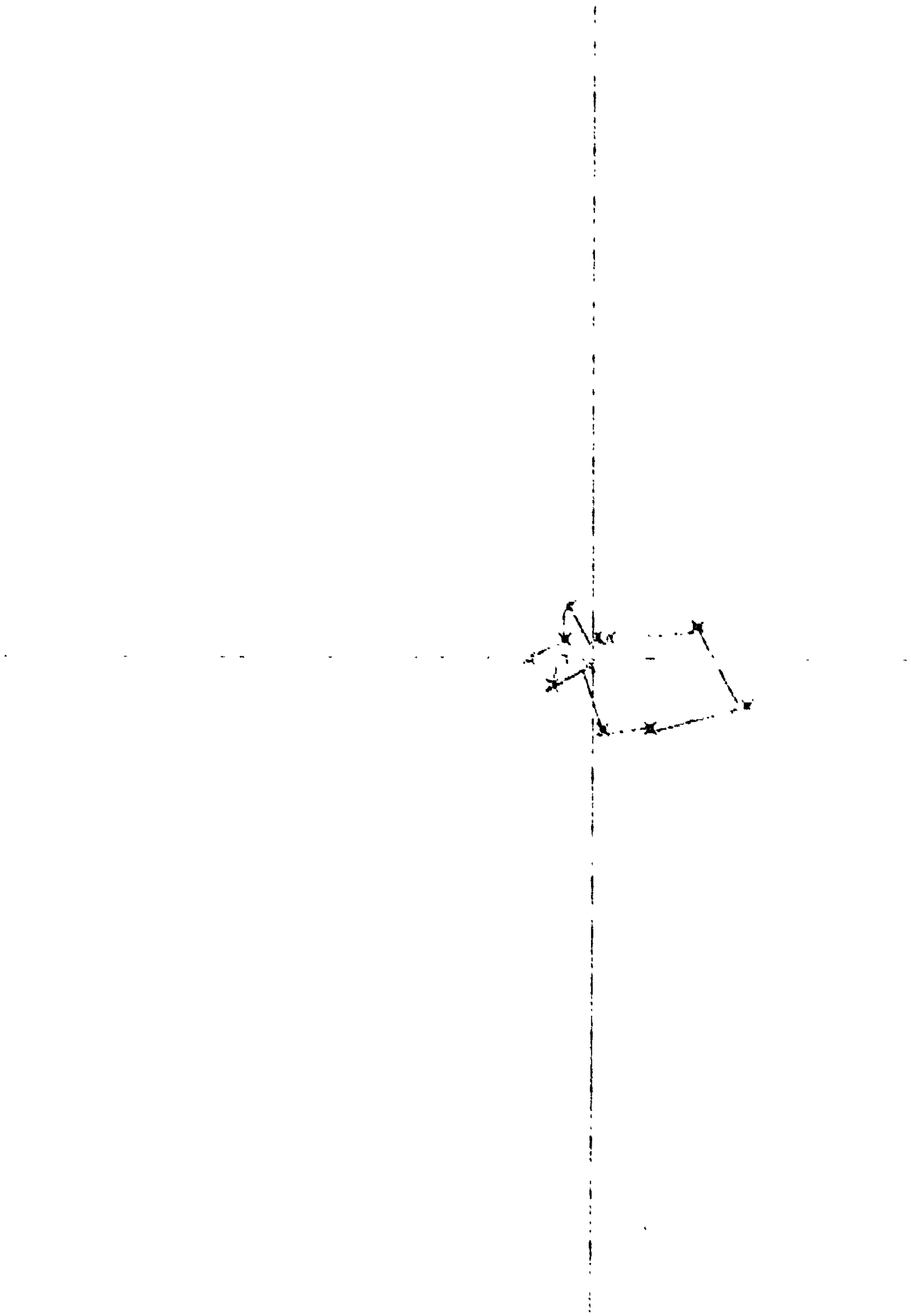
2.1 MHZ. HZRRD



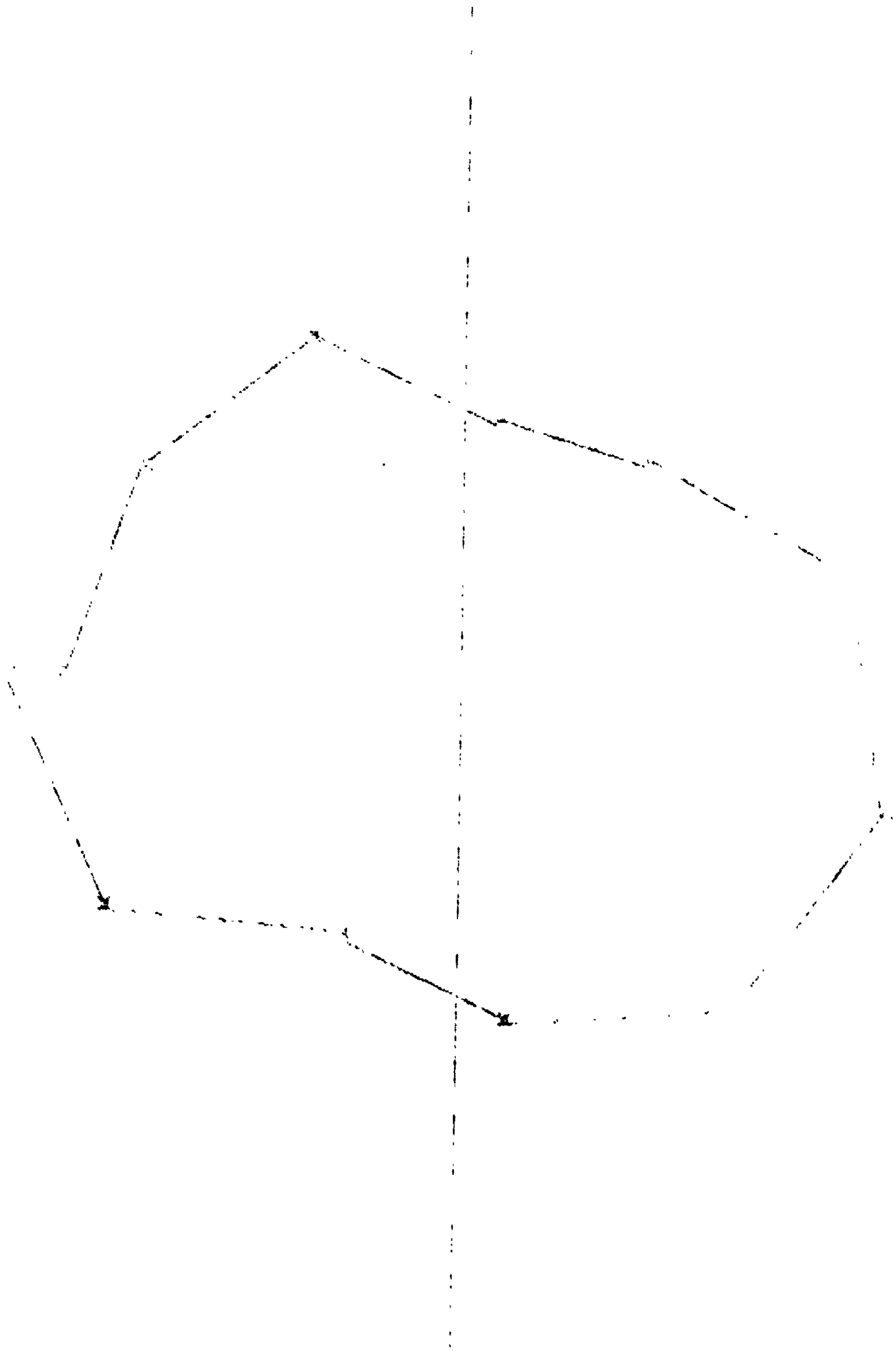
4.0 MHZ. HBEND



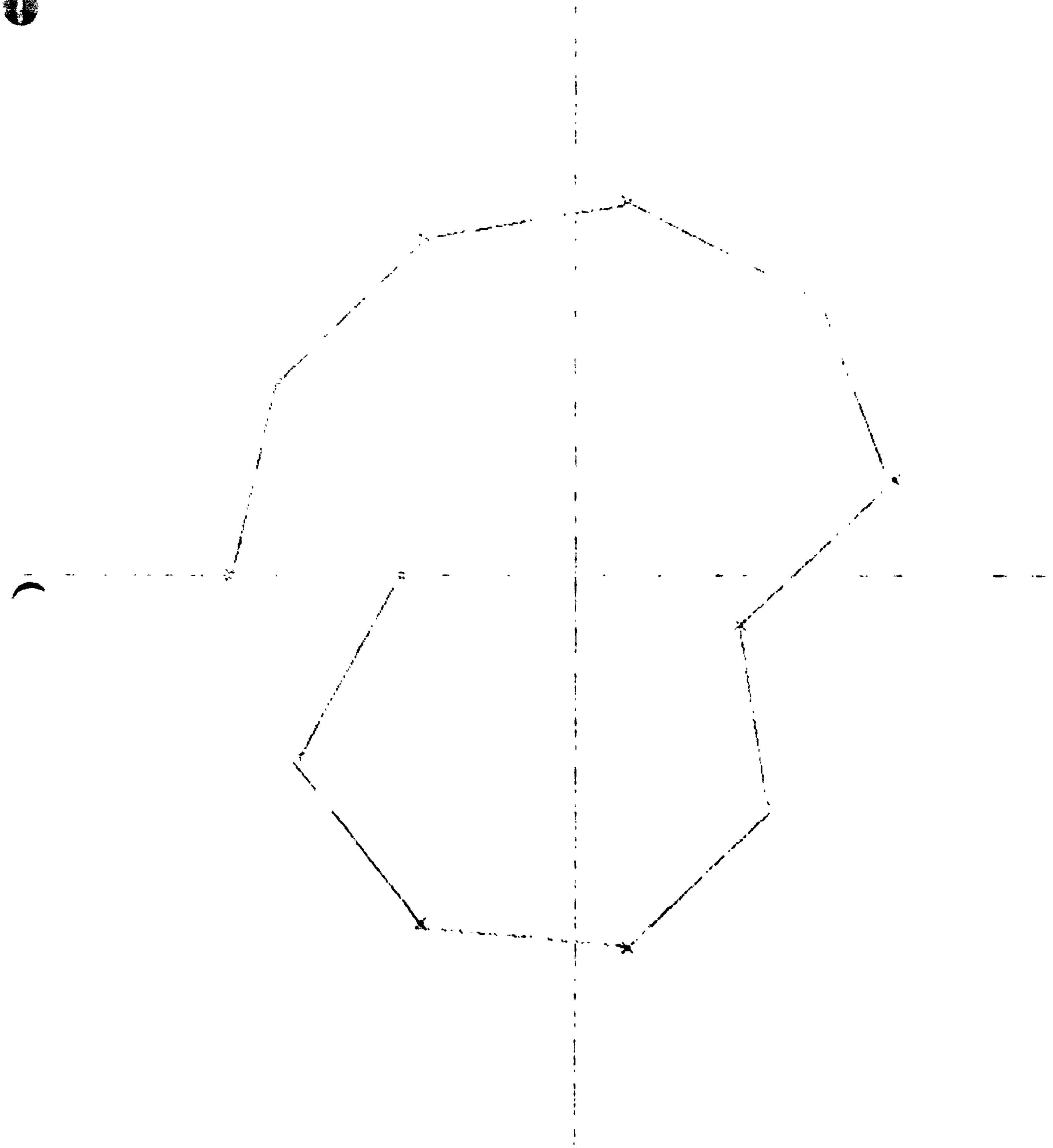
4.0 MHZ. HeENO



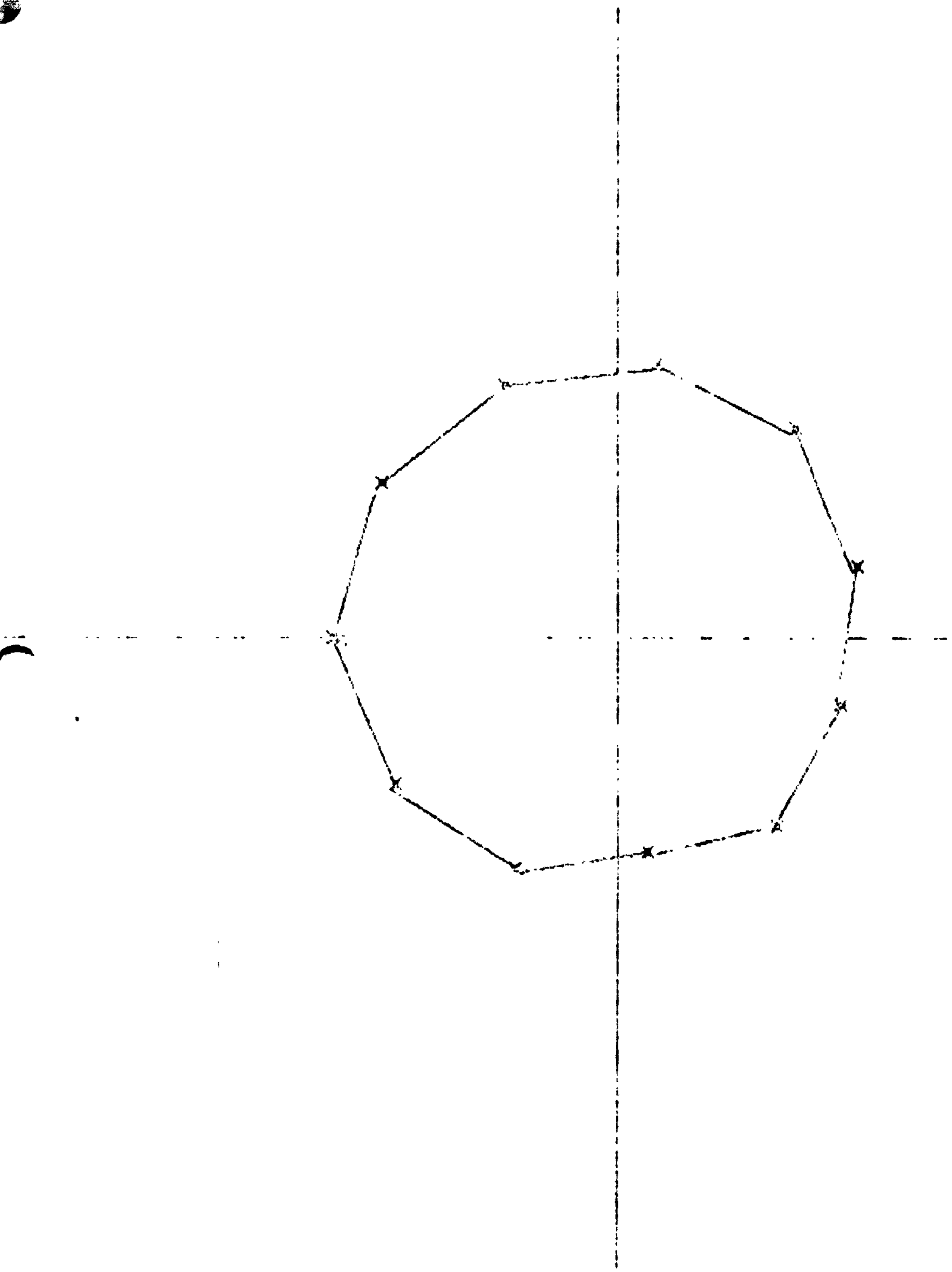
4.0 MHZ. HZEND



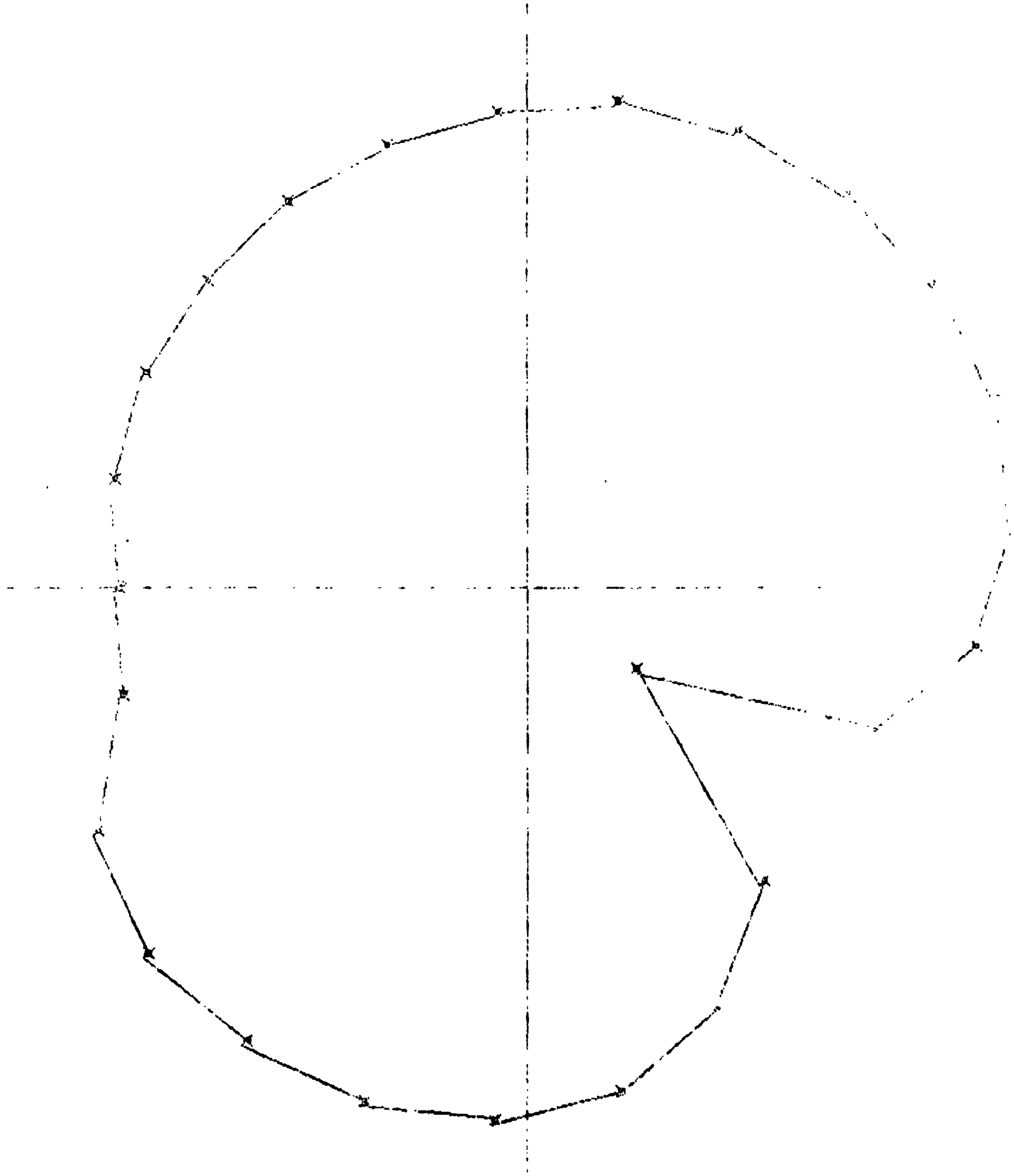
4.0 MHZ. H[illegible]ERD



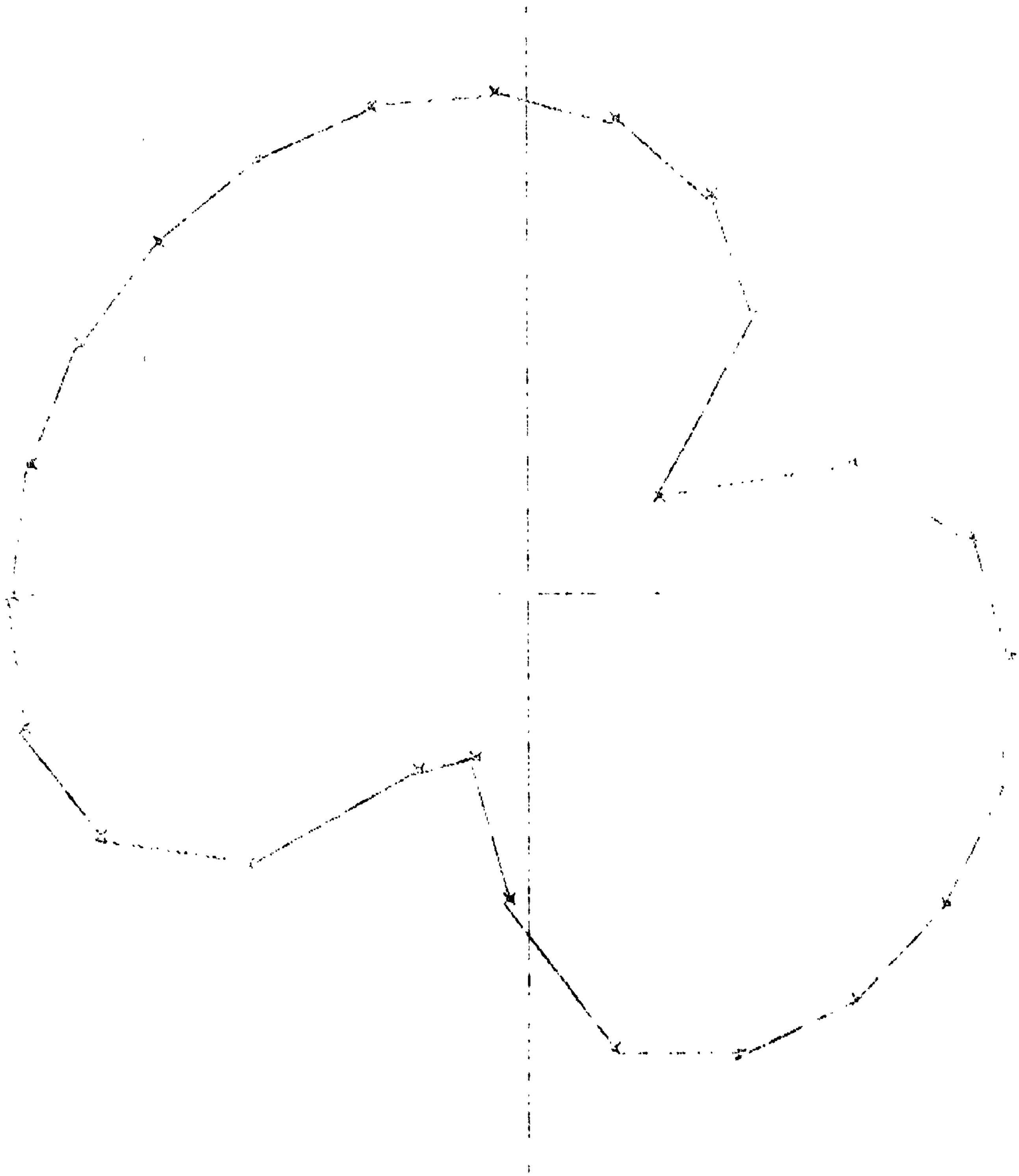
4.0 MHZ. HΦBRO



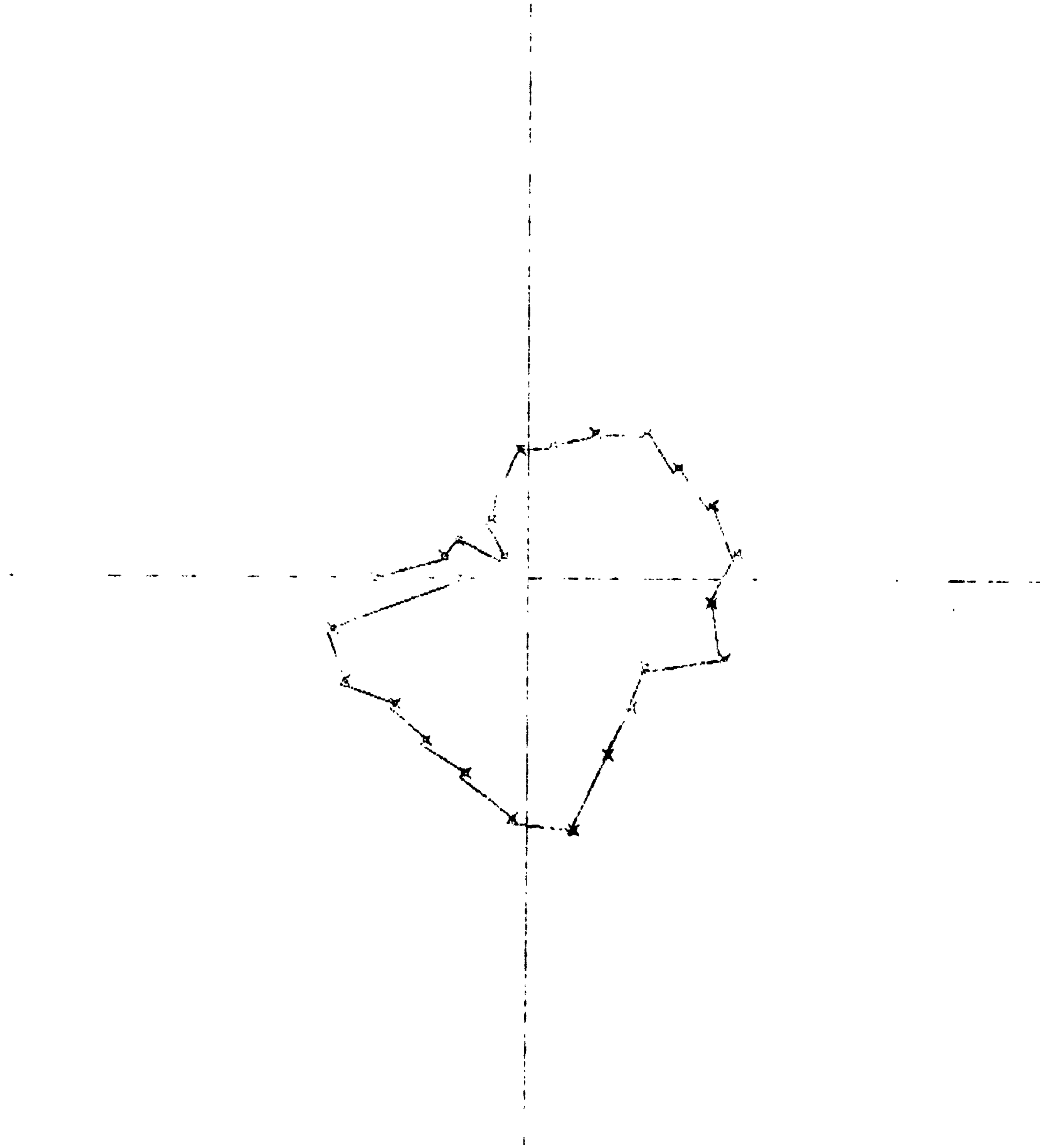
4.0 MHZ. HZSRD



8.1 MHZ. H[illegible]END

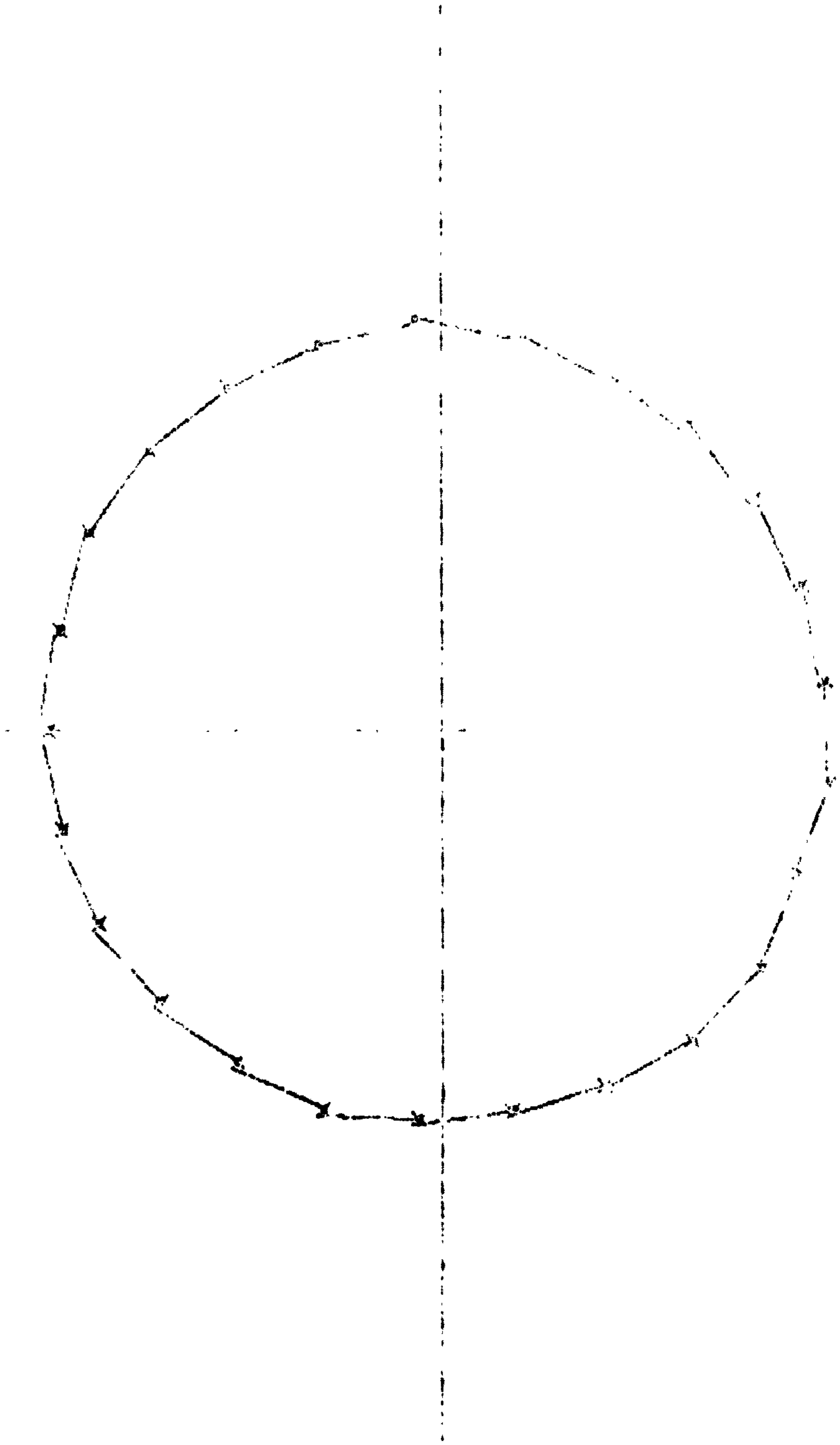


8.1 MHZ. H-END

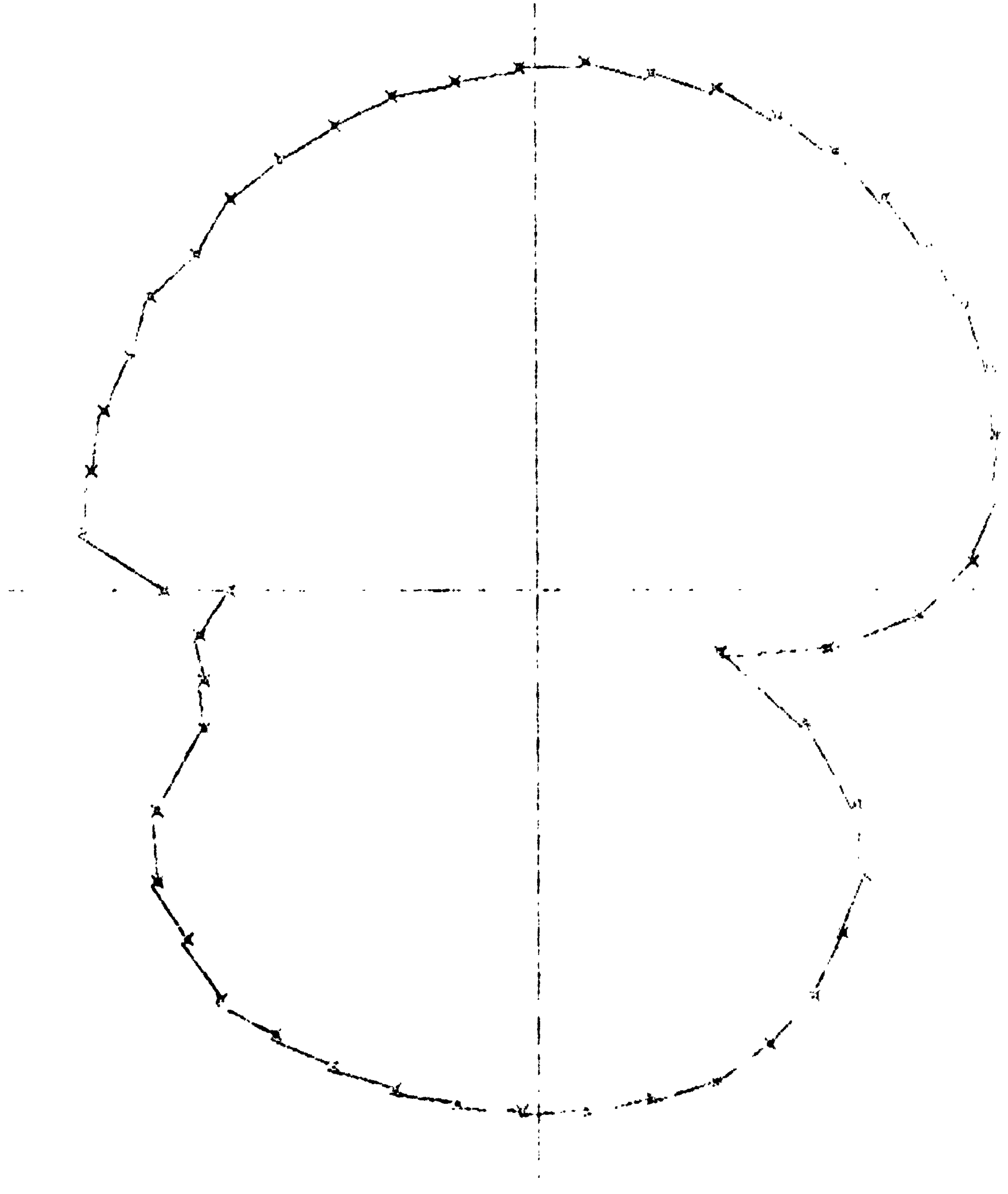


8.1 MHZ. HZENO

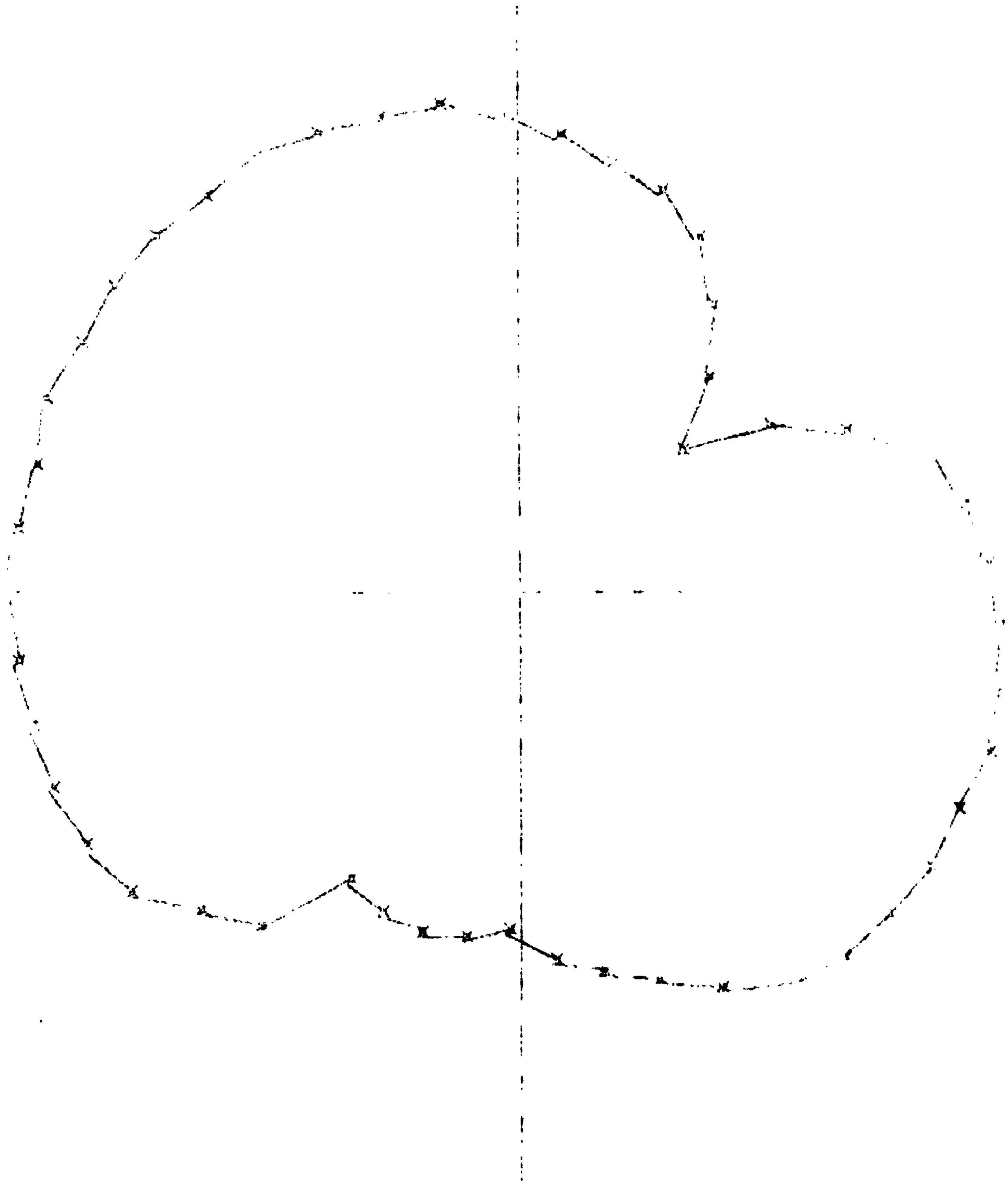
8.1 MHZ. HΦBRC



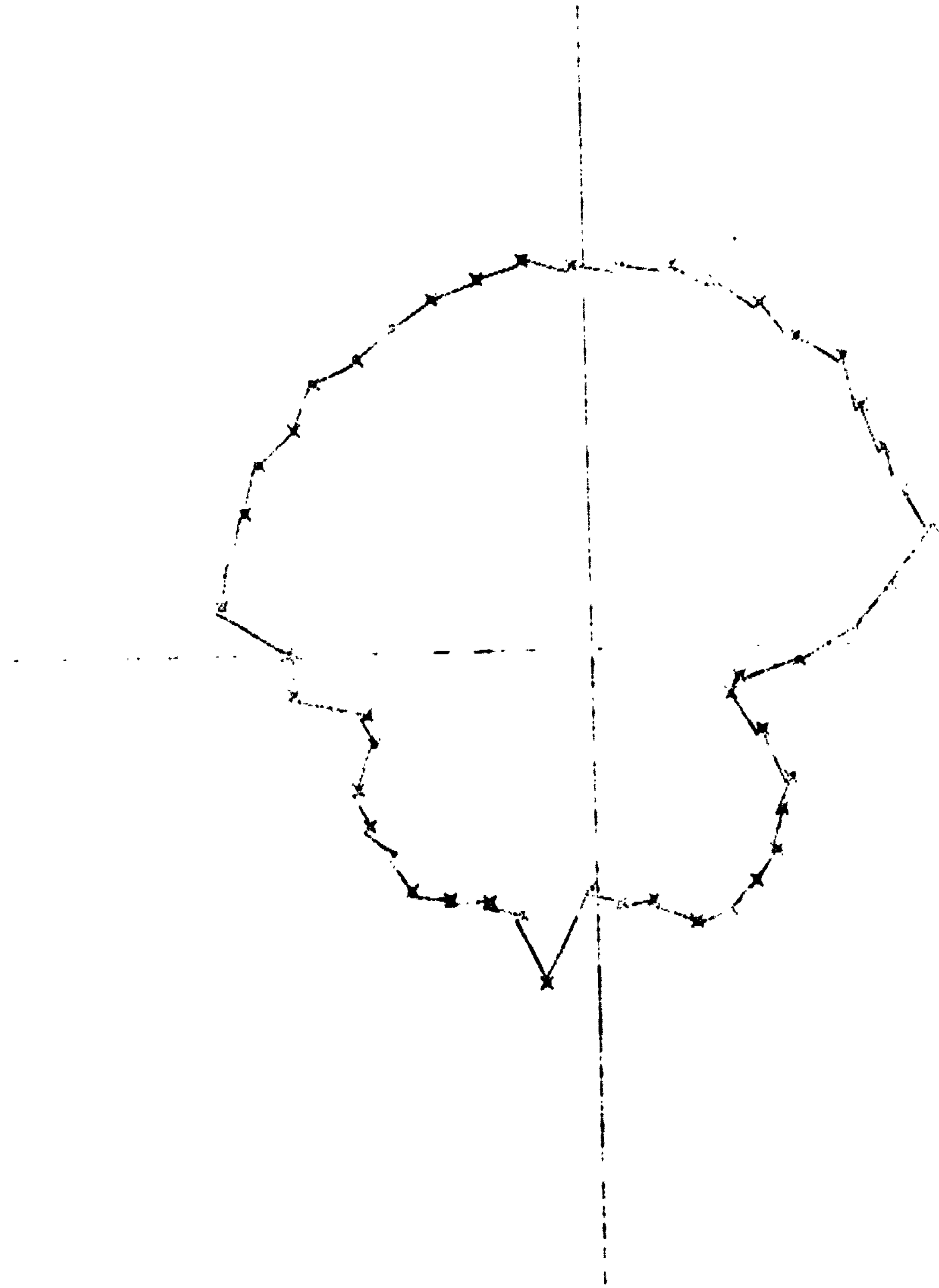
8.1 MHZ. HZ3RC



16.0 MHZ. HPENC

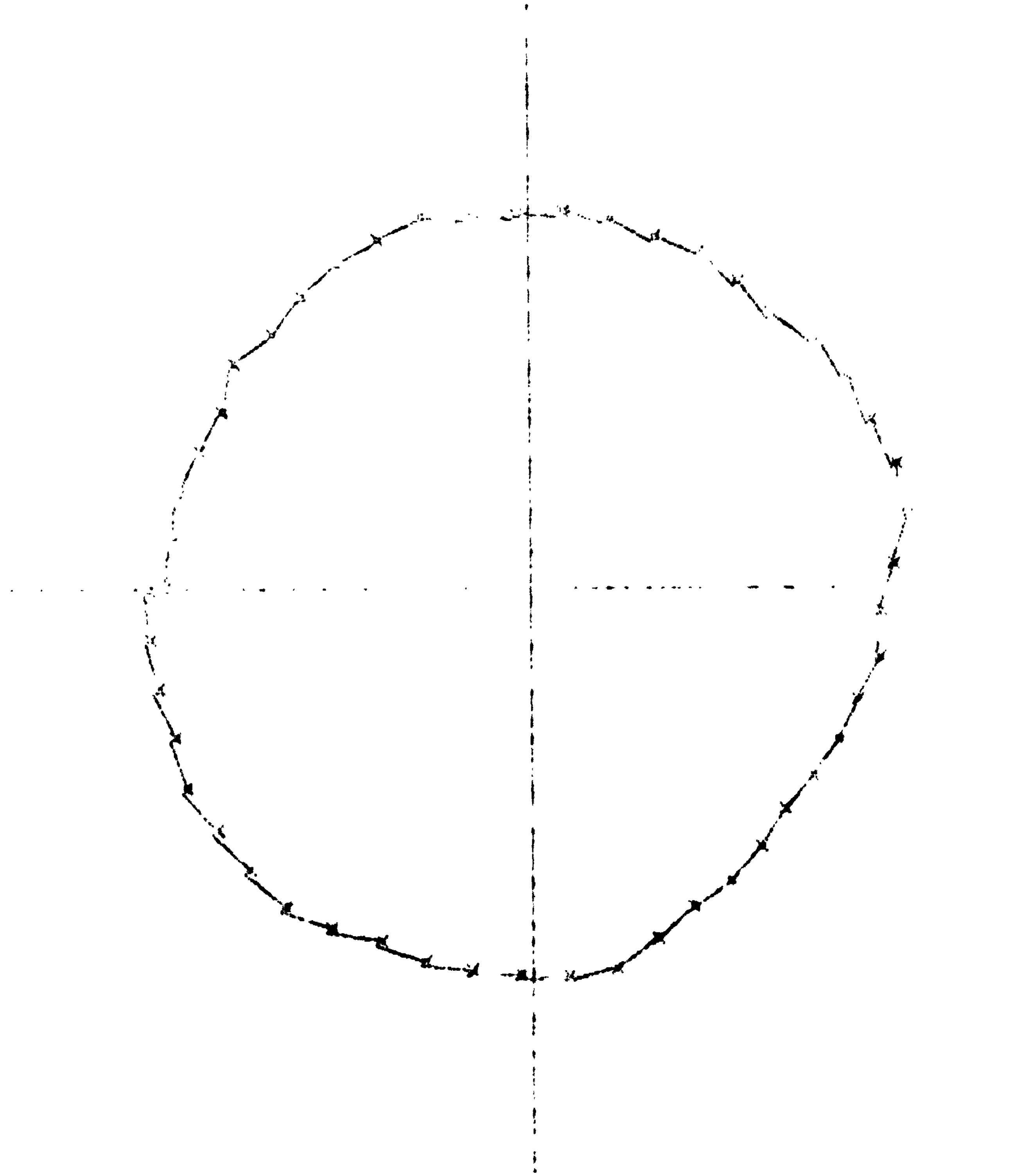


16.0 MHZ. HΦEND

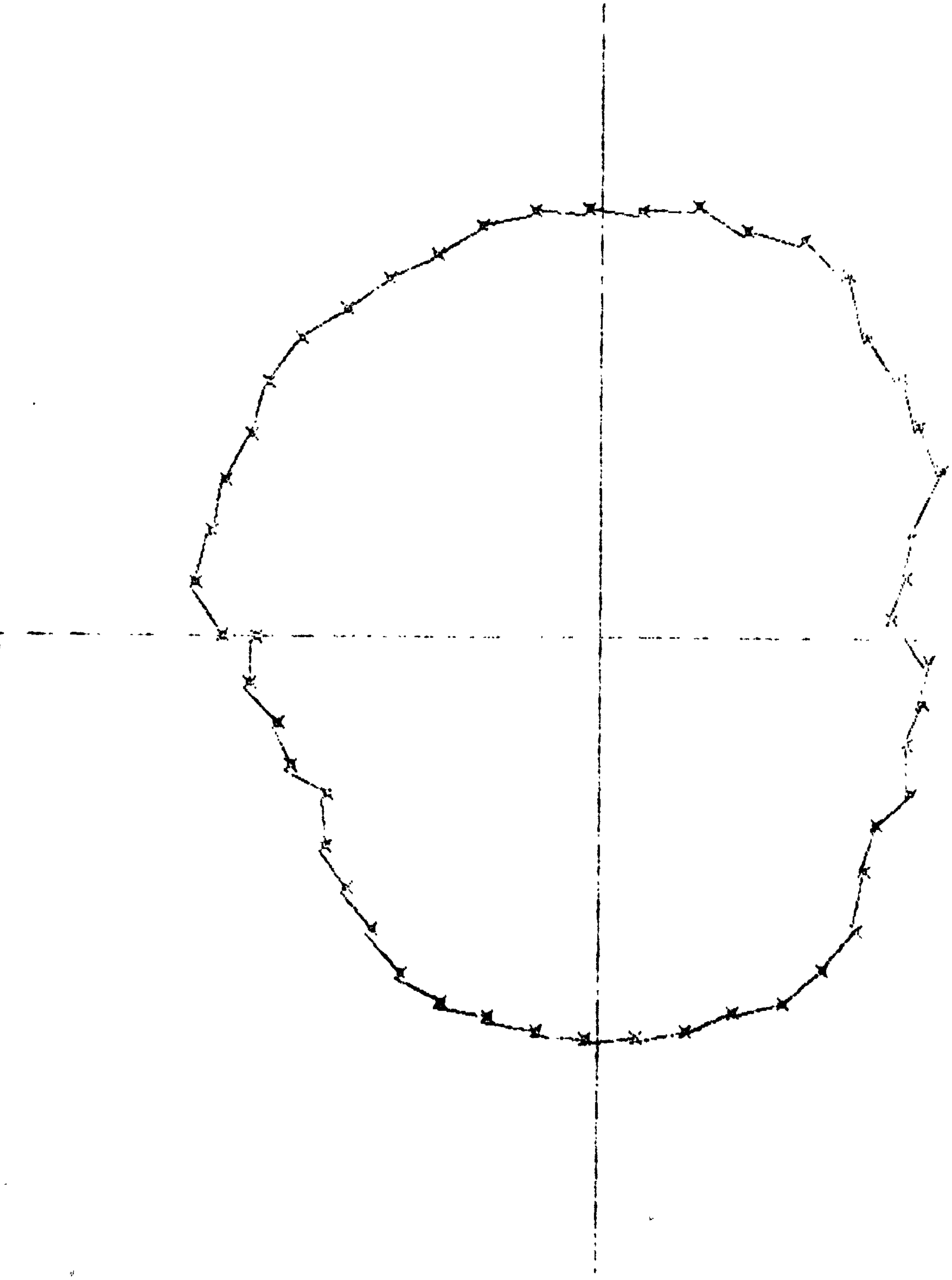


6

16.0 MHZ. HZENC

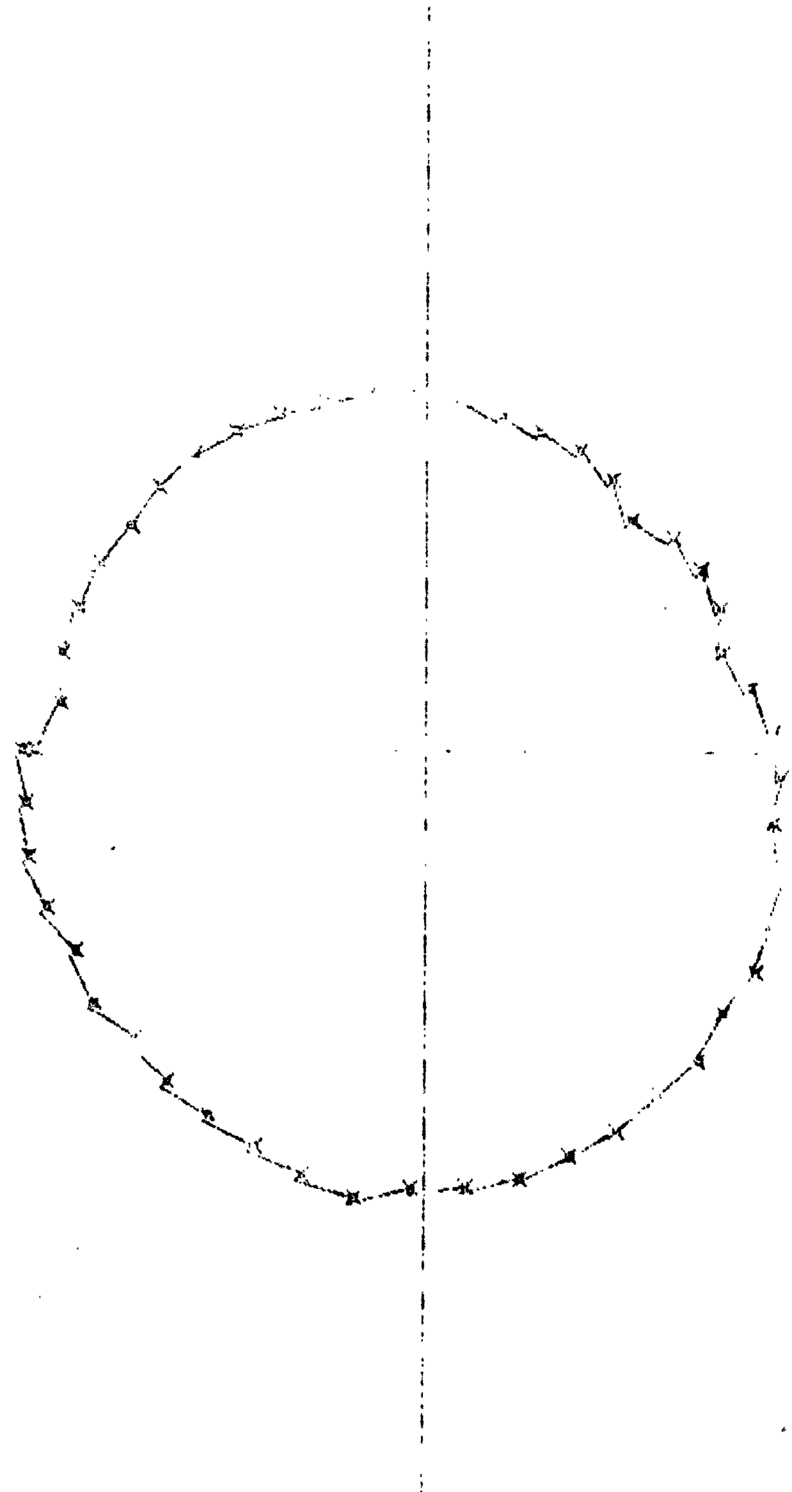


15.0 MHZ. HYBRD

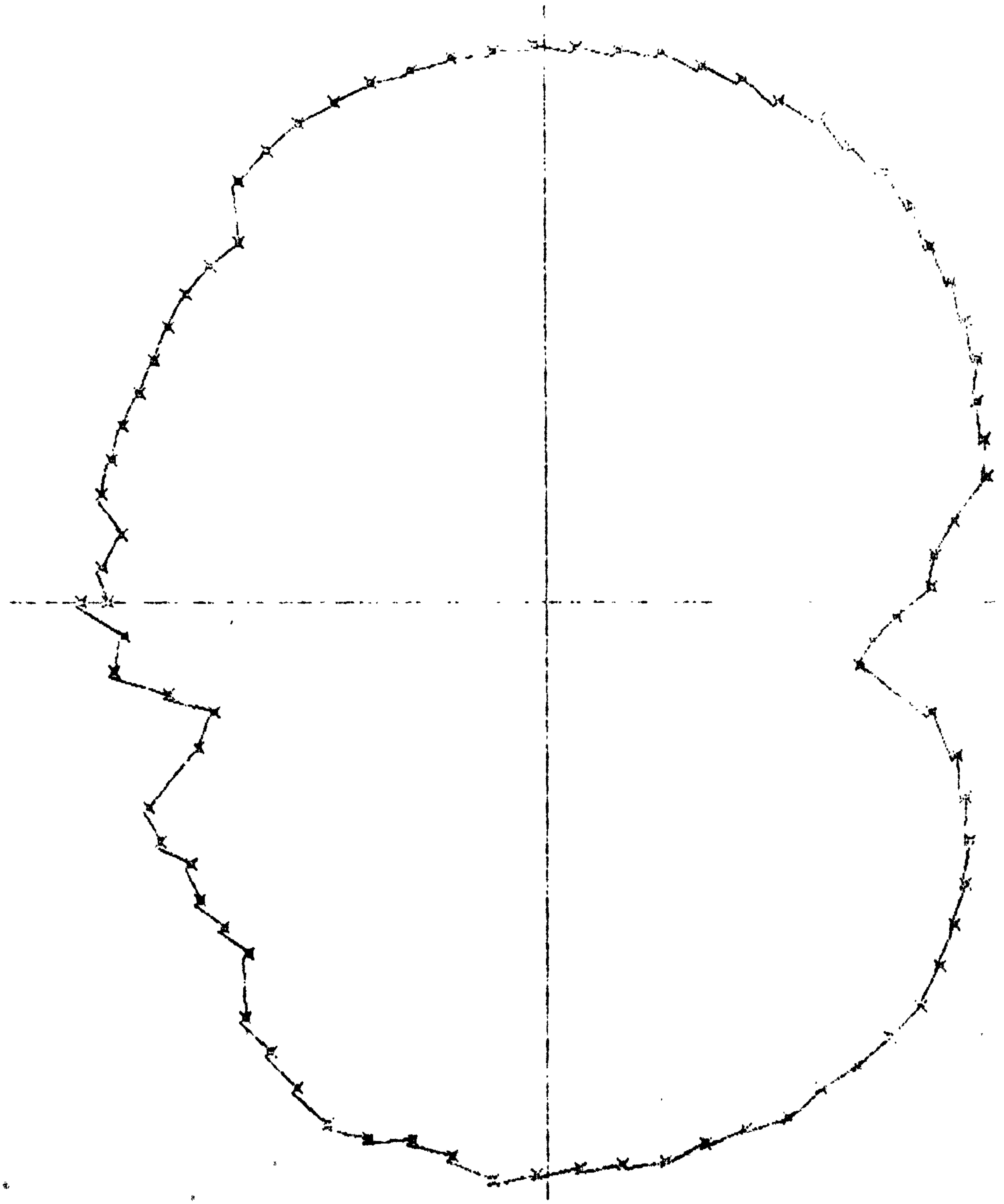


16.0 MHZ. HΦBRD

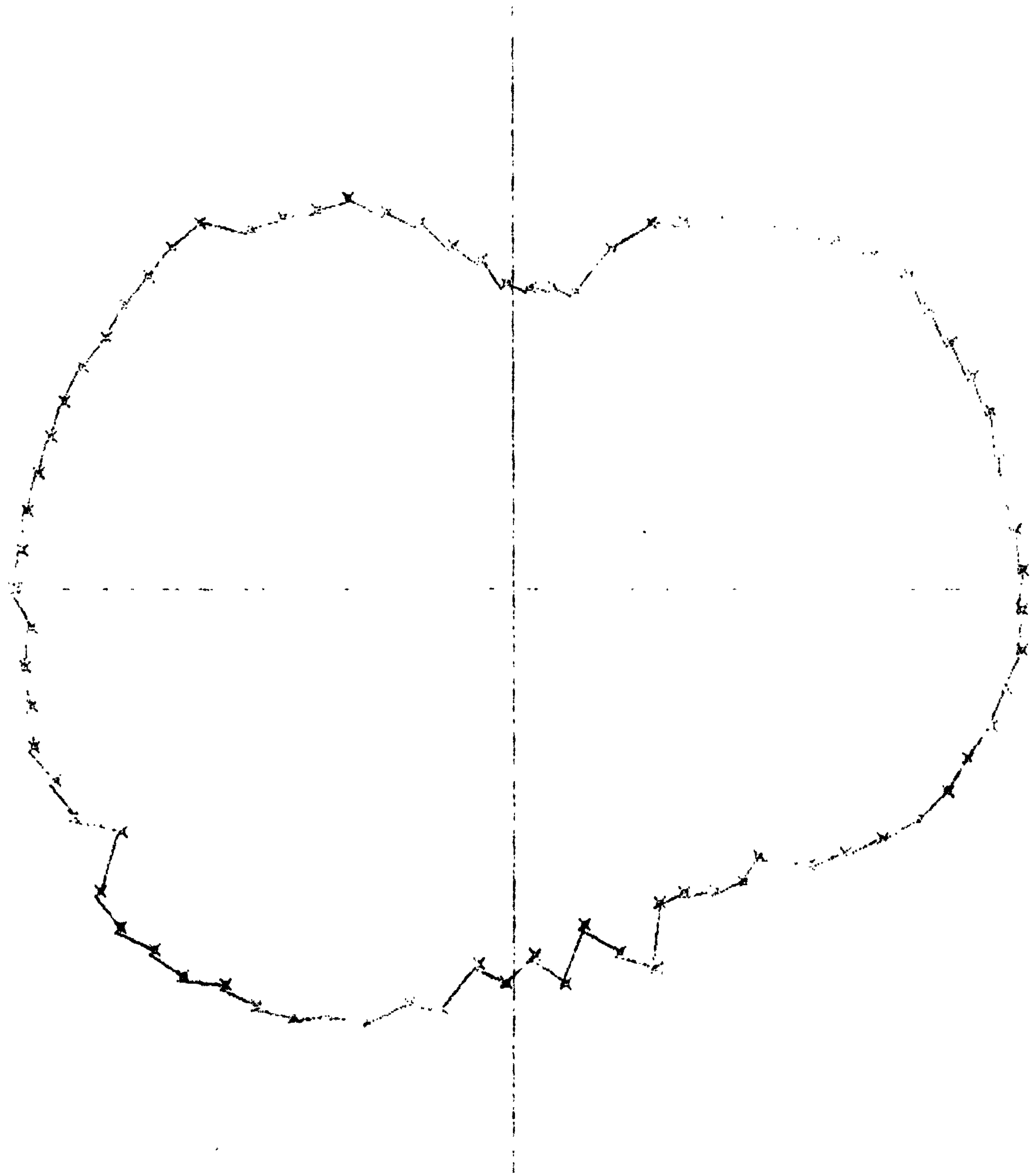
C-4



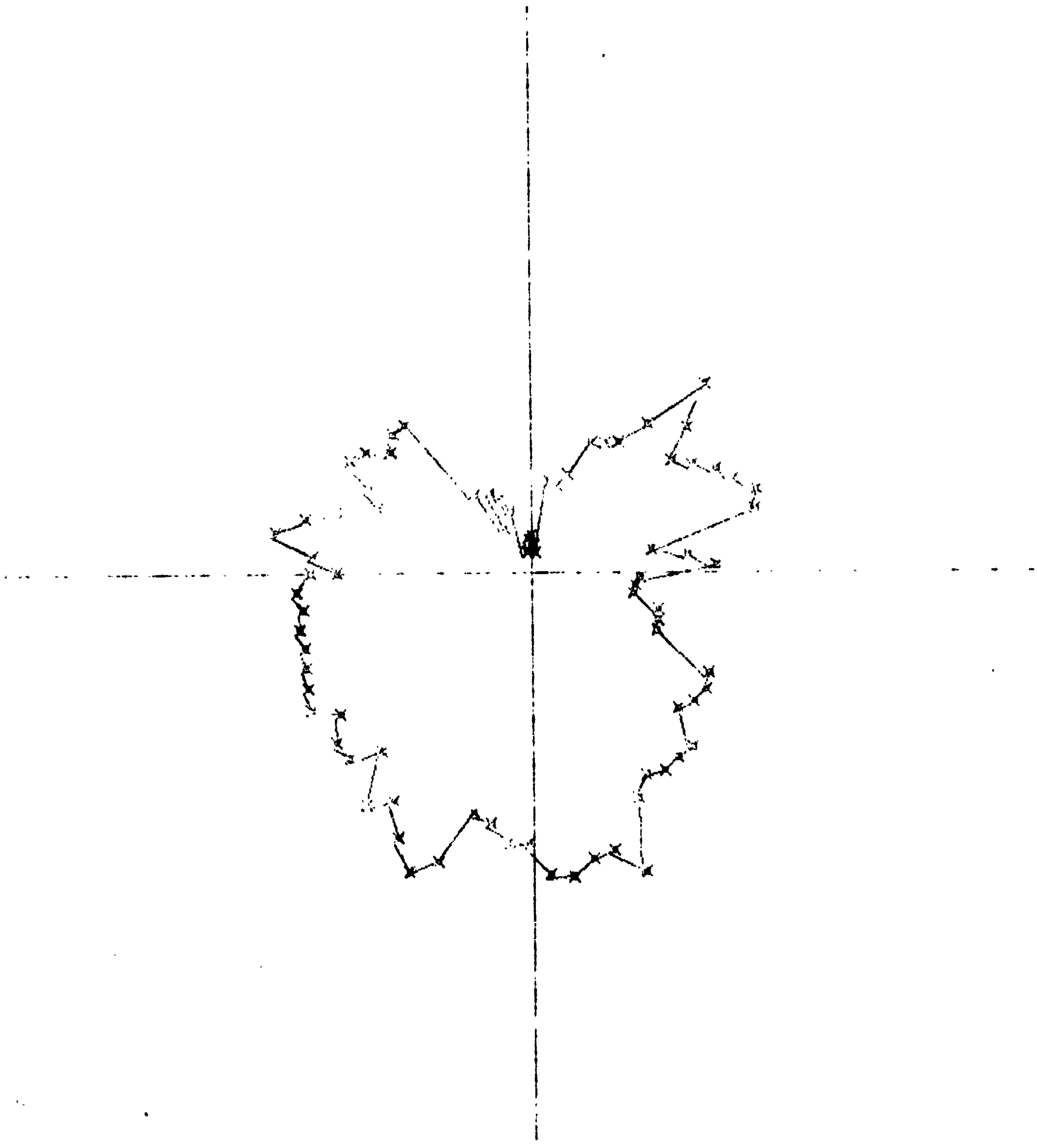
15.0 MHZ. HZBRD



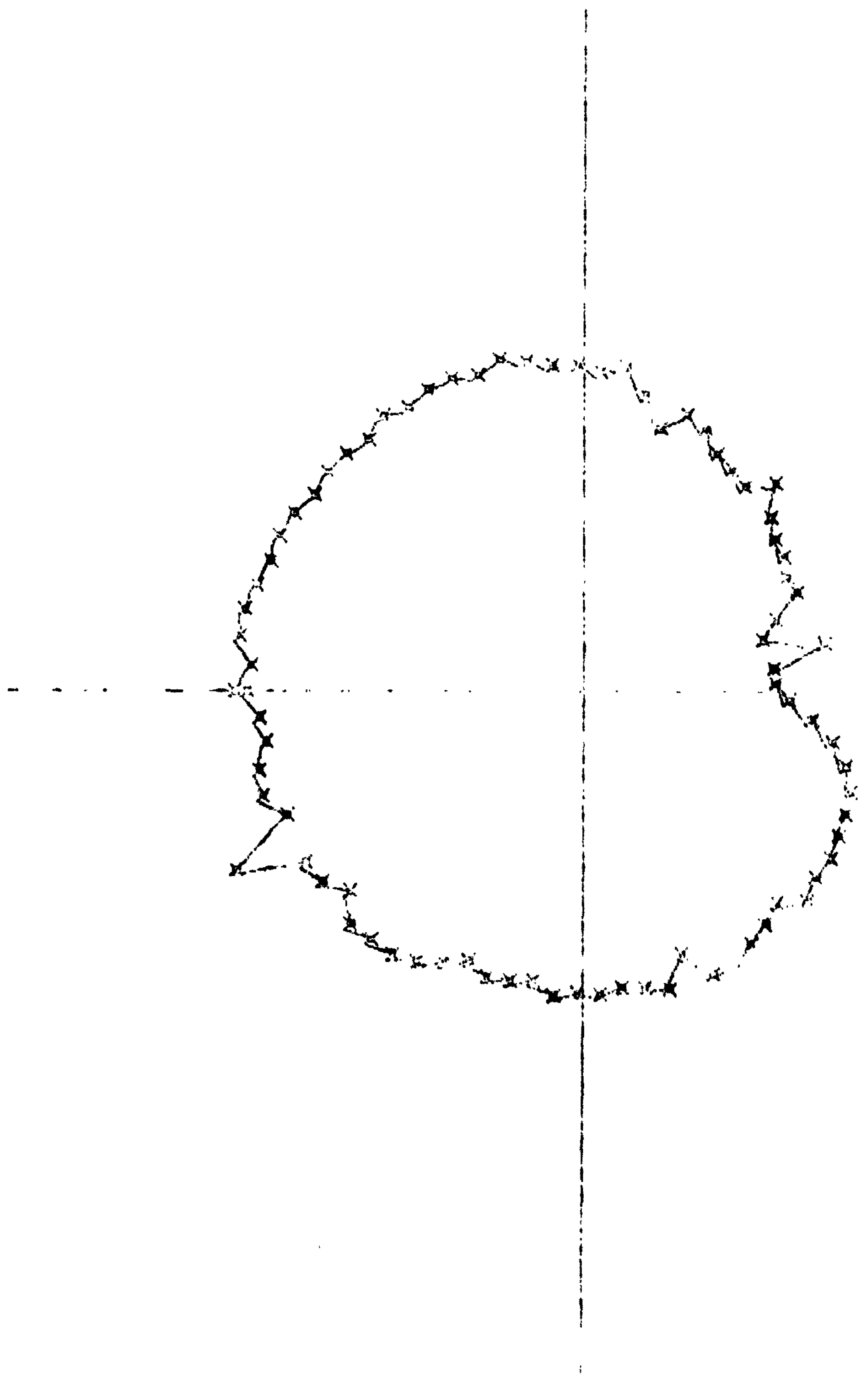
32.1 MHZ. HPEND



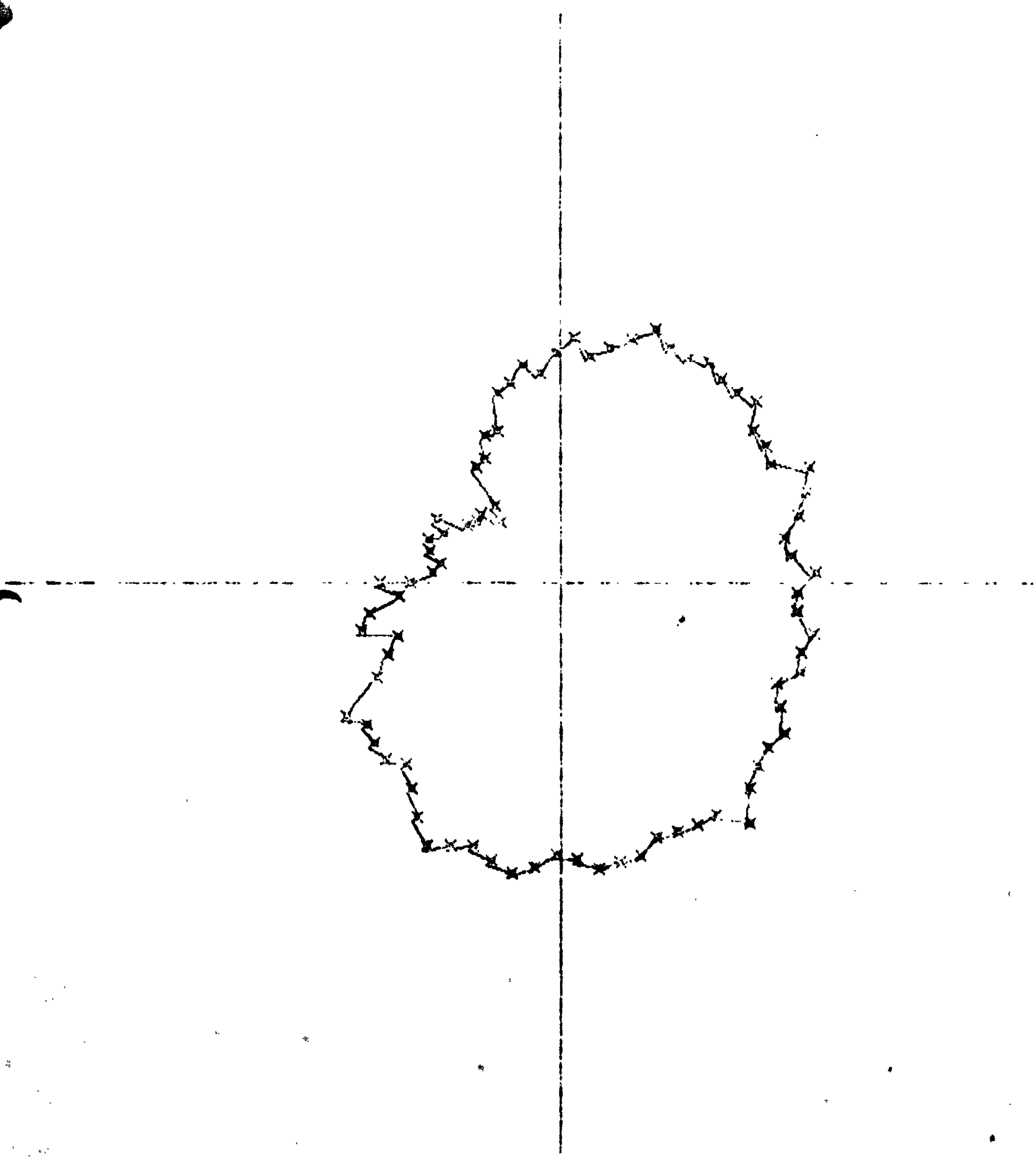
32.1 MHZ. HΦEND



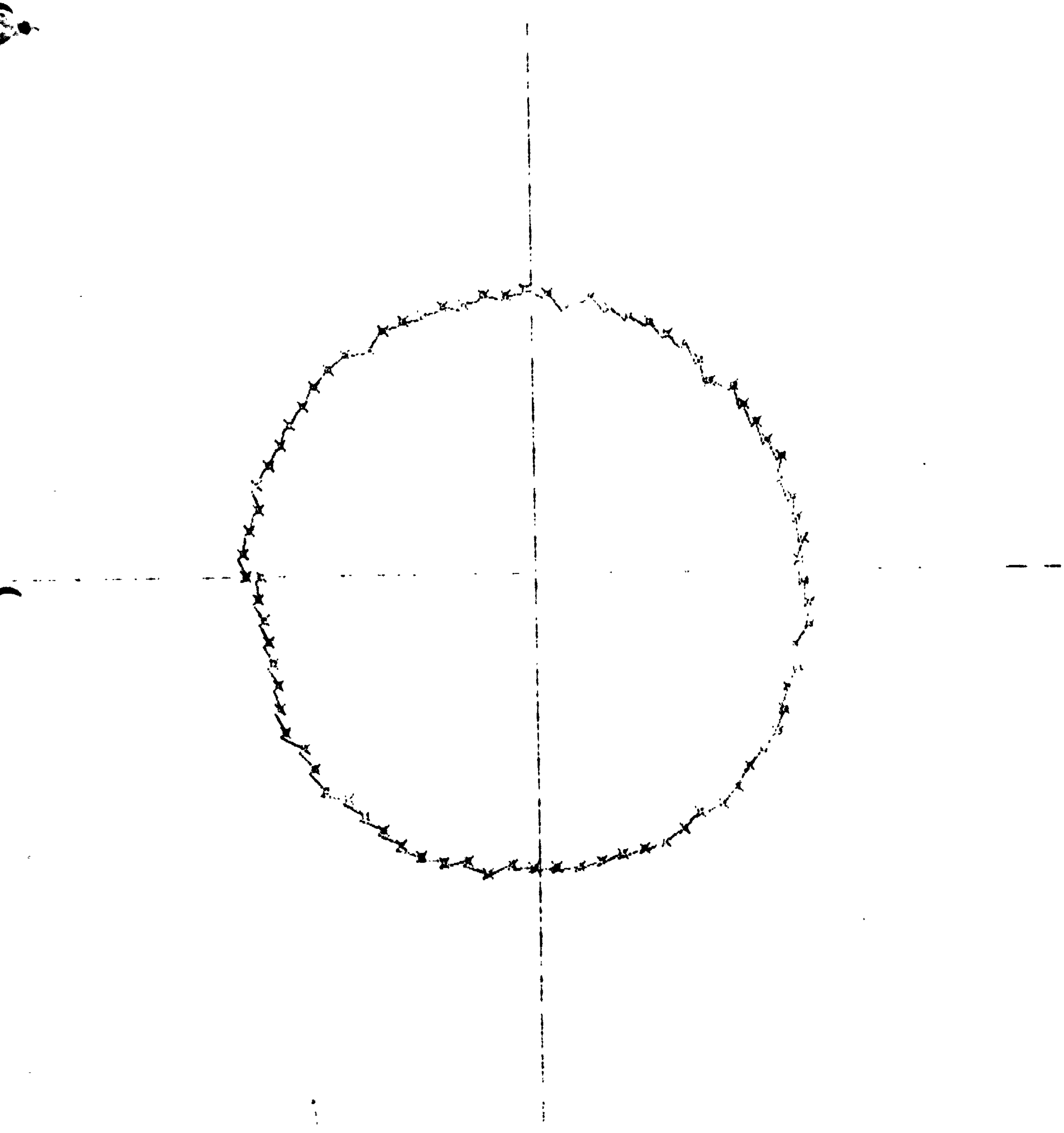
32.1 MHZ. HZEND



32.1 MHZ. H[illegible]BRD



32.1 MHZ. HΦBRD



32.1 MHZ. HZ8RD