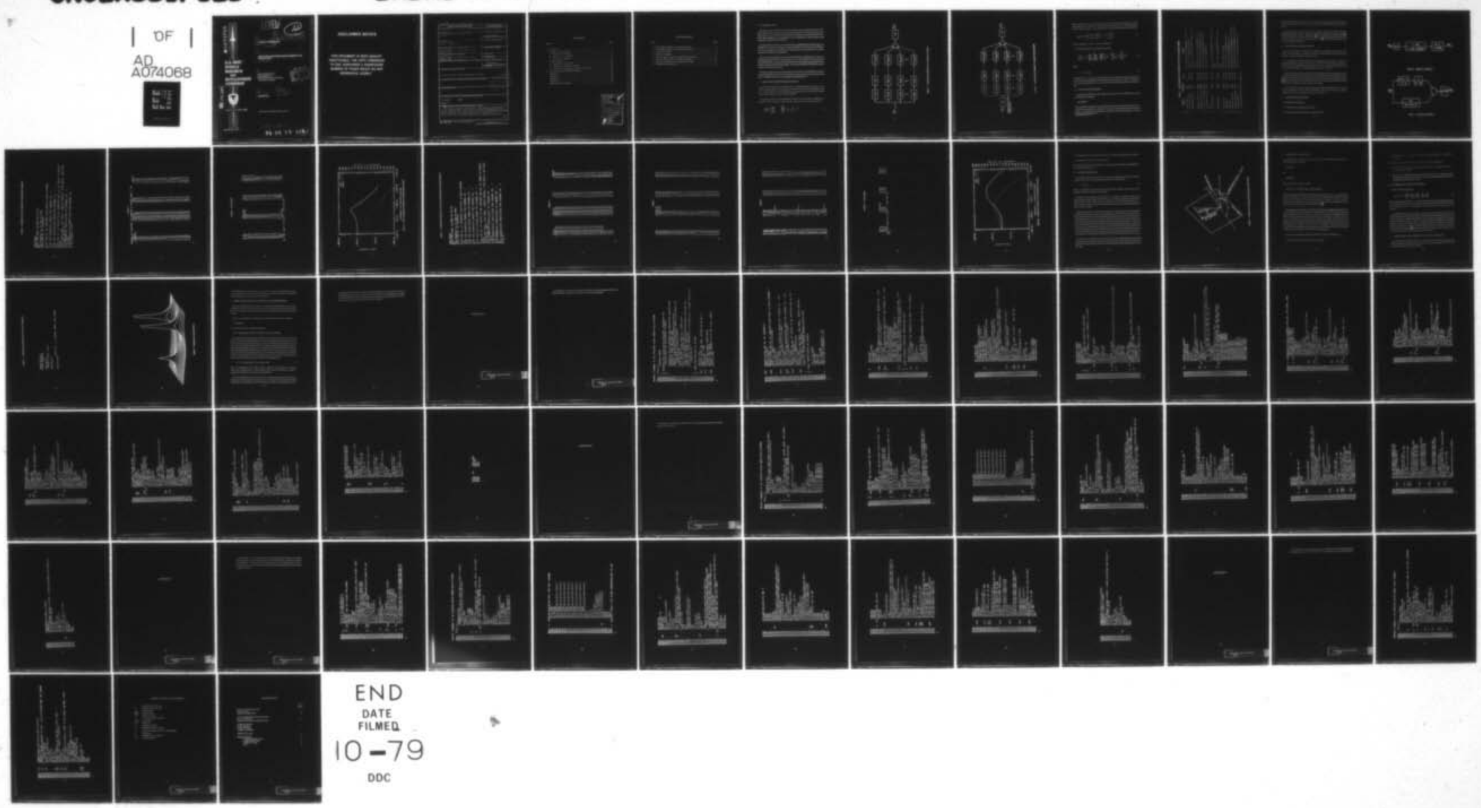
AD-A074 068 ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND REDSTO--ETC F/G 9/2
ADDITIONAL SOFTWARE DEVELOPMENTS FOR THE HP-21 MX.(U)
JUN 79 M C PITRUZZELLO, J R MITCHELL
UNCLASSIFIED DRDMI-T-79-65 NL

| OF |
AD
A074068



END
DATE
FILMED
10-79
DDC

1.0 2.8 2.5 3.2 2.2 3.6 4.0 2.0
1.1
1.8
1.25 1.4 1.6

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

TECHNICAL REPORT T-79-65

ADDITIONAL SOFTWARE DEVELOPMENTS FOR THE HP-21 MX

U.S. ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND

Michael C. Pitruzzello
Jerrel R. Mitchell
Guidance and Control Directorate
Technology Laboratory

26 June 1979

Redstone Arsenal, Alabama 35809

Approved for Public Release; Distribution unlimited.

DMI FORM 1000, 1 APR 77

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER T-79-65	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) ADDITIONAL SOFTWARE DEVELOPMENTS FOR THE HP-21 MX		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jerrel R. Mitchell Michael C. Pitruzzello		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS ATTN: Commander US Missile Research and Development Command DRSMI-TG (R&D) Redstone Arsenal, AL 35809		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS ATTN: Commander US Missile Research and Development Command DRSMI-TI (R&D) Redstone Arsenal, AL 35809		12. REPORT DATE 26 June 79
		13. NUMBER OF PAGES 77
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

M-II SYSTEM FRESP

FORTRAN IV

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report contains the description of four computer codes developed for the HP-21 MX minicomputer system housed in the Control System Branch of the Guidance and Control Directorate of the US Army Missile Research and Development Command (MIRADCOM). These codes (programs) were developed to aid in analyzing fire control systems being developed or proposed by the US Army.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

ILLUSTRATIONS

1. INTRODUCTION

This report contains the description of four computer codes developed for the HP-21MX minicomputer system housed in the Control System Branch of the Guidance and Control Directorate of the US Army Missile Research and Development Command (MIRADCOM). These codes (programs) were developed to aid in analyzing fire control systems being developed or proposed by the US Army.

The names of the symbolic versions of the programs are $FRESP, $PLOT3, $PLOTX, and $DREAD. The names of the binary versions are FRESP, PLOT3, PLOTX and DREAD. Uses of these programs are described in the following sections, and listings of the symbolic versions are contained in the appendices.

Both the symbolic and binary version of FRESP, PLOT3, and DREAD are stored on the flexible disc labeled SYSCPY. The symbolic version of PLOTX ($PLOTX) is stored on the M-II SYSTEM disc. It was stored here because the user must supply the subroutine DATA and then recompile and relocate the program. The libraries on the M-II SYSTEM disc must be searched in the relocation process.

All binary programs must be executed under the control of the M-II SYSTEM contained on the M-II SYSTEM disc. Even if relocated properly, none of the codes can be executed under an M-I SYSTEM since they were developed using FORTRAN IV.

2. FREQUENCY RESPONSE PROGRAM

The program FRESP was developed to aid in computing frequency responses of continuous or sampled-data control systems. This program is stored on the disc labeled M-II and can be executed under control of the M-II System contained on the same disc. The symbolic version of this program is stored in the file $FRESP.

The general continuous and sampled-data systems for which FRESP is applicable are, respectively, shown in *Figures 1* and *2*. For the continuous case, the program computes

$$\frac{C(s)}{R(s)} = \left[\sum_{\ell=1}^{L} \prod_{i=1}^{I(k)} G_{ji}(s) \right] e^{-ds} \tag{1}$$

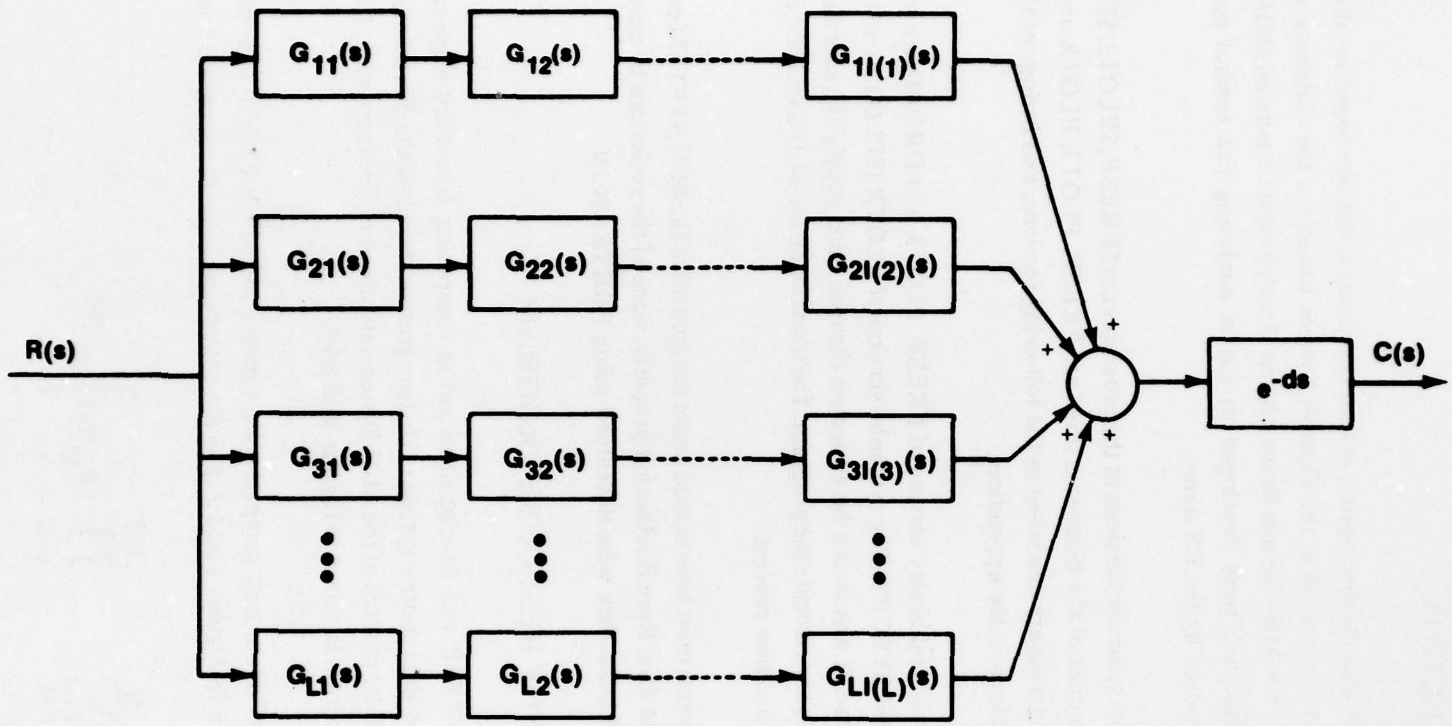


Figure 1. General block diagram for continuous analysis.

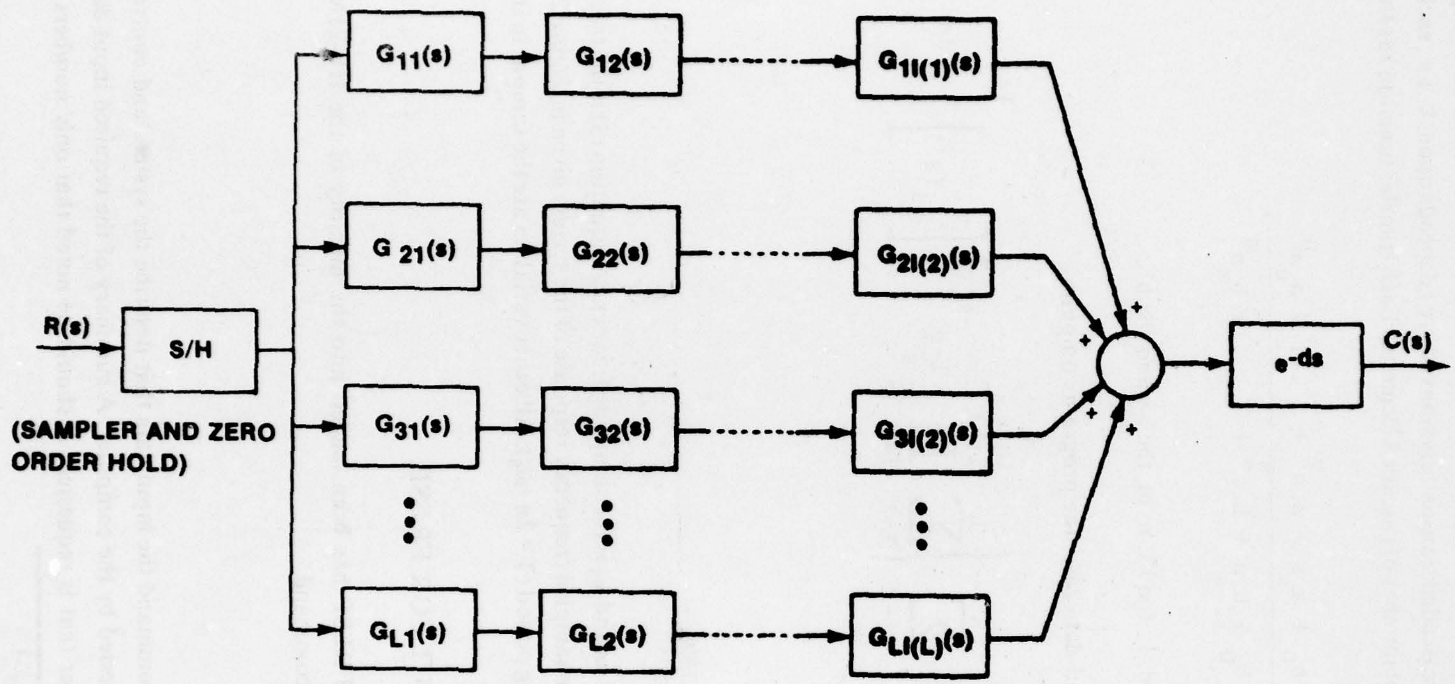


Figure 2. General block diagram for sampled-data analysis.

where s=jω.* There are L forward paths between the input, R(s), and the output, C(s). Each forward path has I cascaded transfer functions where I depends upon ℓ, i.e., each forward path can have a different number of cascaded elements. Each transfer function has the general form

$$G_{ki}(s) = \frac{a_0 + a_1 s + a_2 s^2 + \ldots + a_n s^n}{b_0 + b_1 s + b_2 s^2 + \ldots + b_m s^m} . \tag{2}$$

The user specifies d, L, the I's, n, m, the a's and the b's.

For the sampled-data case, the program computes

$$\frac{C^*(s)}{R^*(s)} = \frac{1}{T} \sum_{k=-K}^{K} \left\{ \sum_{\ell=1}^{L} \prod_{i=1}^{I(k)} \left[\frac{1-e^{-s'T}}{s'} \right] \left[G_{ji}(s') \right] \right\} e^{-ds} \tag{3}$$

where

$$s' = s + jk\omega_s . \tag{4}$$

In Equation (4), s = jω and ω_s is the sampling frequency. Equation (3) is the truncated infinite series version of the sampled frequency response of the system given in *Figure 2*. In Equation (3) T is the sampling period ($T = 2\pi/\omega_s$). All other variables are the same as in the continuous case.

A. DATA INPUT FOR FRESP

Assuming that FRESP has been loaded into the memory of the HP-21MX, it can be executed with the command

ON, FRESP

After issuing this command the input data that describe the system and control the flow of execution are requested by the computer. A summary of the required input data is given in *Table 1* in the order that is requested. It should be noted that only numbers 1 and 15 are

*The symbol j represents $\sqrt{-1}$.

TABLE 1. DEFINITIONS OF REQUIRED INPUTS FOR USING FRESP.

NO.	QUANTITY READ-IN	PROGRAM NAME OF QUANTITY	FORMAT	COMMENTS
1.	Type of Analysis	MODE	A2	CR for continuous on SR for sampled-data. Default is CR.
2.	No. of Parallel Paths	KCHNL	free field	Corresponds to L in Equations (1) and (3).
3.	Transfer functions/path	NUMC	free field	Array for storing number of cascaded transfer functions in each path.
4.	Numerator Orders of T.F.'s	NRATOR	free field	Two-dimensional array for storing numerator orders.
5.	Denominator Order of T.F.'s	NDENOM	free field	Two-dimensional array for storing denominator orders.
6.	No. Points to be Completed	FREQPT	free field	Number of frequency response points to be computed.
7.	No. of Terms to be used in SR.	NOTSIR	free field	Corresponds to K in Equation (3).
8.	Starting Frequency	FSTART	free field	Frequency at which the computation is to begin (units in rad/sec).
9.	Ending Frequency	FRQEND	free field	Frequency at which the computation is to end (units in rad/sec).
10.	Sampling Period	SAMPRD	free field	Units are assumed to be seconds.
11.	Transport Lag	DLFUNC	free field	Amount of time delay (d in Equations (1) and (3).
12.	Gain factor	GAIN	free field	The frequency response magnitude scale factor.
13.	Numerator Coefficients	XCOMN	free field	Two-dimensional array for storing numerator coefficients.*
14.	Denominator Coefficients	YCOMN	free field	Two-dimensional array for storing denominator coefficients.*
15.	Plotting Designator	M1	A2	YES for a plot. NO for no plot. Default is NO.

* The coefficients should be input in ascending order.

column dependent. When requested for these, the user must start his response in column one to be assured of a proper interpretation. It should also be noted that both of these have default values.

All other inputs are free field. If more than one number is required, the numbers should be separated by commas or blanks. Each line of input is terminated with a carriage return (CR). For a free field input, if a slash (/) followed by a (CR) is issued, the input can be continued to the next line. If a line of input has not been terminated, i.e., a (CR) given, the input of the line can be restarted by issuing a RUBOUT.

B. ILLUSTRATIVE USES OF FRESP

In order to illustrate the pragmatic use of FRESP, two examples are presented in this section. The first example is a single-forward path, sampled-data system, and the second example is a two-forward path continuous system.

(1) EXAMPLE 1. Consideration is given to the system shown in *Figure 3*. The goal is to compute the sampled frequency response when T = 0.02513 for $0.1 \leq \omega \leq 100$. Ten terms are to be used in the series and 50 frequency response points are to be computed and plotted. The session with the computer for accomplishing this is shown in *Table 2*.

On the first page of *Table 2* the dialogue with the computer for describing the system and controlling the computational flow is given. After the data is printed, the computer queries as to whether or not a plot is desired. In this case a plot was desired; therefore, YES followed by a (CR) was typed. (The YES is not shown in this case.) The screen was automatically erased, and the frequency response plot shown in *Figure 5* was made.

(2) EXAMPLE 2. In this example it is desired to compute and plot the frequency response of the system shown in *Figure 43* for $0.01 \leqslant \omega \leqslant 99.9$. The results of the session with the computer for this case are presented in *Table 3* and *Figure 6*.

C. LIMITATIONS OF FRESP

The limitations of FRESP are:

- Maximum of 5 forward paths (channels)

- Maximum of 10 transfer functions per forward path

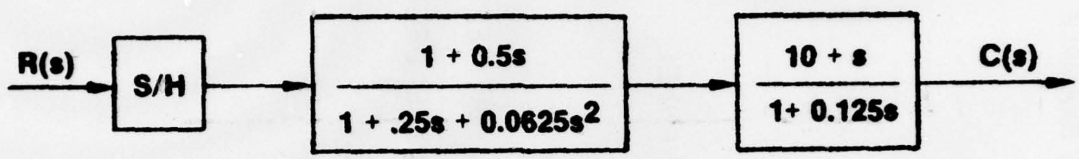


Figure 3. System for example 1.

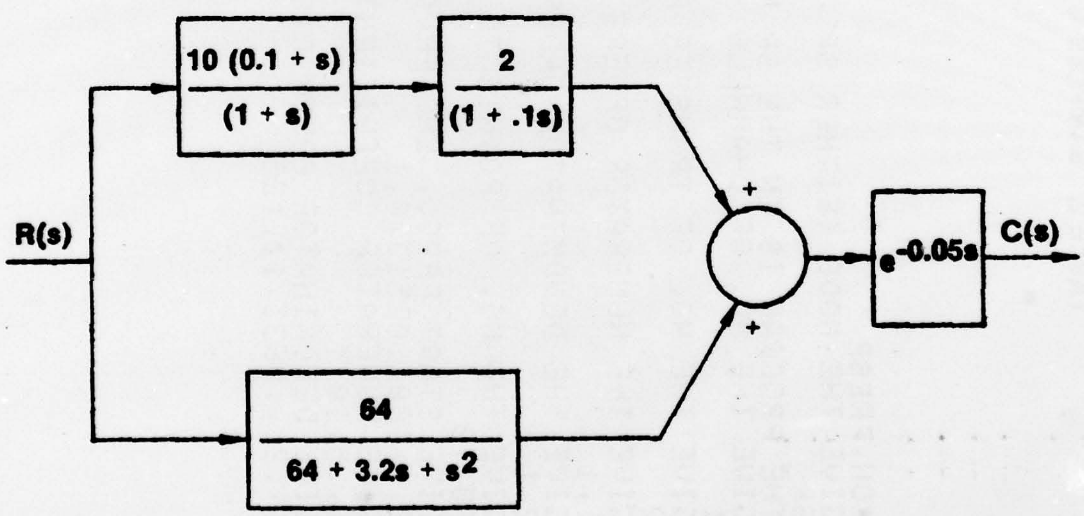


Figure 4. System for example 2.

TABLE 2. SAMPLED-DATA FREQUENCY RESPONSE EXAMPLE

```
*ON,FRESP
GIVE THE MODE (EITHER SR OR CR)
SR
THE PROGRAM IS IN THE SR MODE
GIVE THE NO. OF CHANNELS
1
GIVE THE NO. OF TRANSFER FUNCTIONS IN EACH CHANNEL
2
GIVE THE NUMERATOR ORDERS FOR CHANNEL NO.  1
1,1
GIVE THE DENOMINATOR ORDER FOR CHANNEL NO.  1
2,1
GIVE THE NO. OF POINTS TO BE COMPUTED AND NO. OF TERMS IN SERIES
50,10
GIVE START FREQ., END FREQ., SAMPLE PERIOD, TIME DELAY, AND GAIN
0.1,100,.02513,0,1
GIVE NUMERATOR COEFFICIENTS OF CHANNEL NO.  1
1,.5,10,1
GIVE DENOMINATOR COEFFICIENT FOR CHANNEL NO.  1
1,.25,.0625,1,.125
```

TABLE 2.

FRQEND IS CHANGED TO OMEGAS/2 SINCE MODE = SR

OMEGA	OMEGAW	MAG.	ANGLE
1.000E-01	.126E-02	10.0	1.21
.116	.145E-02	10.0	1.40
.134	.168E-02	10.0	1.62
.155	.194E-02	10.0	1.87
.179	.225E-02	10.0	2.16
.207	.260E-02	10.1	2.49
.239	.301E-02	10.1	2.87
.277	.348E-02	10.1	3.30
.320	.403E-02	10.2	3.80
.371	.466E-02	10.2	4.35
.429	.539E-02	10.3	4.98
.496	.623E-02	10.4	5.67
.573	.720E-02	10.5	6.43
.663	.833E-02	10.7	7.24
.767	.964E-02	10.9	8.07
.887	.111E-01	11.2	8.88
1.03	.129E-01	11.6	9.60
1.19	.149E-01	12.1	10.1
1.37	.173E-01	12.7	10.3
1.59	.200E-01	13.6	9.85
1.84	.231E-01	14.7	8.50
2.12	.267E-01	16.1	5.80
2.46	.309E-01	17.8	1.18
2.84	.357E-01	19.7	-6.03
3.29	.413E-01	21.2	-16.3
3.80	.478E-01	21.7	-29.3
4.40	.553E-01	20.7	-43.5
5.09	.640E-01	18.3	-56.5
5.88	.741E-01	15.5	-67.3

*

TABLE 2. (Concluded)

6.81	.857E-01	12.8	-75.5
7.87	.992E-01	10.6	-81.9
9.11	.115	8.75	-86.8
10.5	.133	7.27	-90.7
12.2	.154	6.07	-94.0
14.1	.179	5.10	-96.9
16.3	.208	4.31	-99.5
18.9	.241	3.66	-102.
21.8	.281	3.13	-105.
25.2	.328	2.68	-107.
29.2	.384	2.31	-111.
33.7	.451	1.99	-114.
39.0	.534	1.73	-118.
45.1	.637	1.51	-122.
52.2	.770	1.32	-128.
60.4	.948	1.17	-133.
69.8	1.20	1.04	-140.
80.8	1.61	.936	-148.
93.4	2.39	.859	-157.
108.	4.63	.809	-168.
125.	-.191E+06	.790	180.

DO YOU WANT TO PLOT? (YES OR NO)

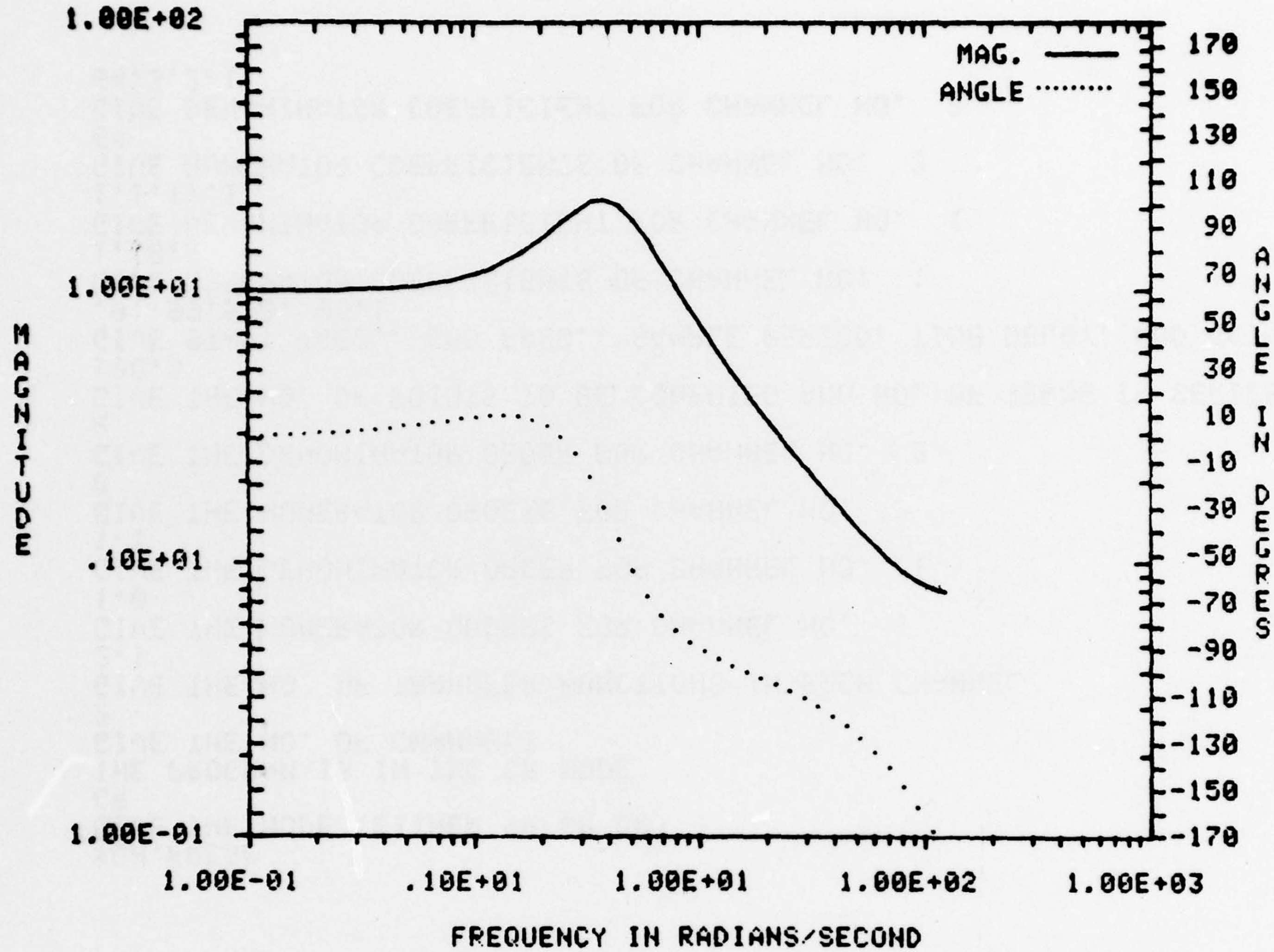


Figure 5. Plot of frequency response of sampled-data example.

TABLE 3. COMPUTER RESULTS FOR THE CONTINUOUS SYSTEM EXAMPLE

```
*ON,FRESP
GIVE THE MODE (EITHER SR OR CR)
CR
THE PROGRAM IS IN THE CR MODE
GIVE THE NO. OF CHANNELS
2
GIVE THE NO. OF TRANSFER FUNCTIONS IN EACH CHANNEL
2,1
GIVE THE NUMERATOR ORDERS FOR CHANNEL NO.  1
1,0
GIVE THE DENOMINATOR ORDER FOR CHANNEL NO.  1
1,1
GIVE THE NUMERATOR ORDERS FOR CHANNEL NO.  2
0
GIVE THE DENOMINATOR ORDER FOR CHANNEL NO.  2
2
GIVE THE NO. OF POINTS TO BE COMPUTED AND NO. OF TERMS IN SERIES
100,0
GIVE START FREQ., END FREQ., SAMPLE PERIOD, TIME DELAY, AND GAIN
.01,99.9,0,.05,1
GIVE NUMERATOR COEFFICIENTS OF CHANNEL NO.  1
1,10,2
GIVE DENOMINATOR COEFFICIENT FOR CHANNEL NO.  1
1,1,1,.1
GIVE NUMERATOR COEFFICIENTS OF CHANNEL NO.  2
64
GIVE DENOMINATOR COEFFICIENT FOR CHANNEL NO.  2
64,3.2,1
```

TABLE 3.

OMEGA	OMEGAW	MAG.	ANGLE
1.000E-02	.500E-02	3.01	3.35
.110E-01	.549E-02	3.01	3.68
.120E-01	.602E-02	3.01	4.04
.132E-01	.661E-02	3.01	4.43
.145E-01	.725E-02	3.02	4.86
.159E-01	.796E-02	3.02	5.33
.175E-01	.874E-02	3.02	5.84
.192E-01	.959E-02	3.03	6.40
.210E-01	.105E-01	3.03	7.01
.231E-01	.116E-01	3.04	7.68
.254E-01	.127E-01	3.05	8.42
.278E-01	.139E-01	3.06	9.21
.305E-01	.153E-01	3.07	10.1
.335E-01	.168E-01	3.08	11.0
.368E-01	.184E-01	3.10	12.0
.404E-01	.202E-01	3.12	13.2
.443E-01	.222E-01	3.14	14.4
.486E-01	.243E-01	3.17	15.6
.534E-01	.267E-01	3.20	17.0
.586E-01	.293E-01	3.24	18.5
.643E-01	.321E-01	3.28	20.0
.705E-01	.353E-01	3.34	21.7
.774E-01	.387E-01	3.40	23.4
.850E-01	.425E-01	3.48	25.2
.932E-01	.467E-01	3.57	27.1
.102	.512E-01	3.67	29.0
.112	.562E-01	3.80	30.9
.123	.617E-01	3.94	32.8
.135	.677E-01	4.10	34.6
.148	.744E-01	4.28	36.5

*

TABLE 3.

	.163	.816E-01	4.50	38.2
	.179	.896E-01	4.74	39.8
	.196	.984E-01	5.01	41.2
	.215	.108	5.31	42.5
	.236	.119	5.65	43.6
	.259	.130	6.03	44.5
	.285	.143	6.44	45.1
	.312	.158	6.90	45.5
	.343	.173	7.39	45.6
	.376	.190	7.92	45.4
	.413	.210	8.49	45.0
	.453	.231	9.10	44.3
	.497	.254	9.74	43.3
	.546	.280	10.4	42.1
	.599	.309	11.1	40.6
	.658	.341	11.8	38.9
	.722	.377	12.5	36.9
	.792	.418	13.2	34.7
	.869	.464	13.9	32.3
	.954	.517	14.6	29.8
	1.05	.577	15.3	27.0
	1.15	.647	15.9	24.2
	1.26	.730	16.5	21.3
	1.38	.829	17.0	18.2
	1.52	.950	17.5	15.1
	1.67	1.10	17.9	12.0
	1.83	1.30	18.3	8.76
	2.01	1.57	18.6	5.48
	2.20	1.97	18.8	2.16
	2.42	2.65	19.0	-1.23
	2.65	4.02	19.2	-4.68
	2.91	8.72	19.3	-8.21
*	3.20	-35.9	19.3	-11.8

TABLE 3.

3.51	-5.38	19.4	-15.6
3.85	-2.70	19.3	-19.4
4.23	-1.66	19.3	-23.4
4.64	-1.08	19.2	-27.6
5.09	-.679	19.0	-32.1
5.59	-.363	18.9	-36.8
6.13	-.759E-01	18.7	-42.1
6.73	.227	18.5	-48.3
7.39	.615	18.0	-55.7
8.11	1.29	16.8	-63.3
8.90	3.69	15.3	-68.8
9.76	-5.86	14.2	-72.9
10.7	-1.33	13.4	-77.2
11.8	-.427	12.7	-82.2
12.9	.171	12.0	-87.8
14.2	1.03	11.4	-93.8
15.5	12.2	10.7	-100.
17.1	-1.25	10.0	-107.
18.7	-.634E-01	9.34	-114.
20.5	1.14	8.69	-122.
22.6	-3.48	8.06	-129.
24.7	-.194	7.46	-138.
27.2	1.61	6.88	-146.
29.8	-1.04	6.34	-156.
32.7	.761	5.83	-166.
35.9	-1.25	5.35	-176.
39.4	1.15	4.91	172.
43.2	-.384	4.50	160.
47.5	-5.82	4.12	147.
52.1	1.29	3.77	132.
57.2	.321	3.44	117.
62.7	-.443E-01	3.15	99.8
* 68.9	-.128	2.87	81.5

TABLE 3. (Concluded)

```
     75.6              .876E-01          2.62              61.5
     82.9              .728              2.39              39.7
     91.0             25.0               2.18              15.9
     99.9             -.326              1.99             -10.1
DO YOU WANT TO PLOT?  (YES OR NO)
YES
```

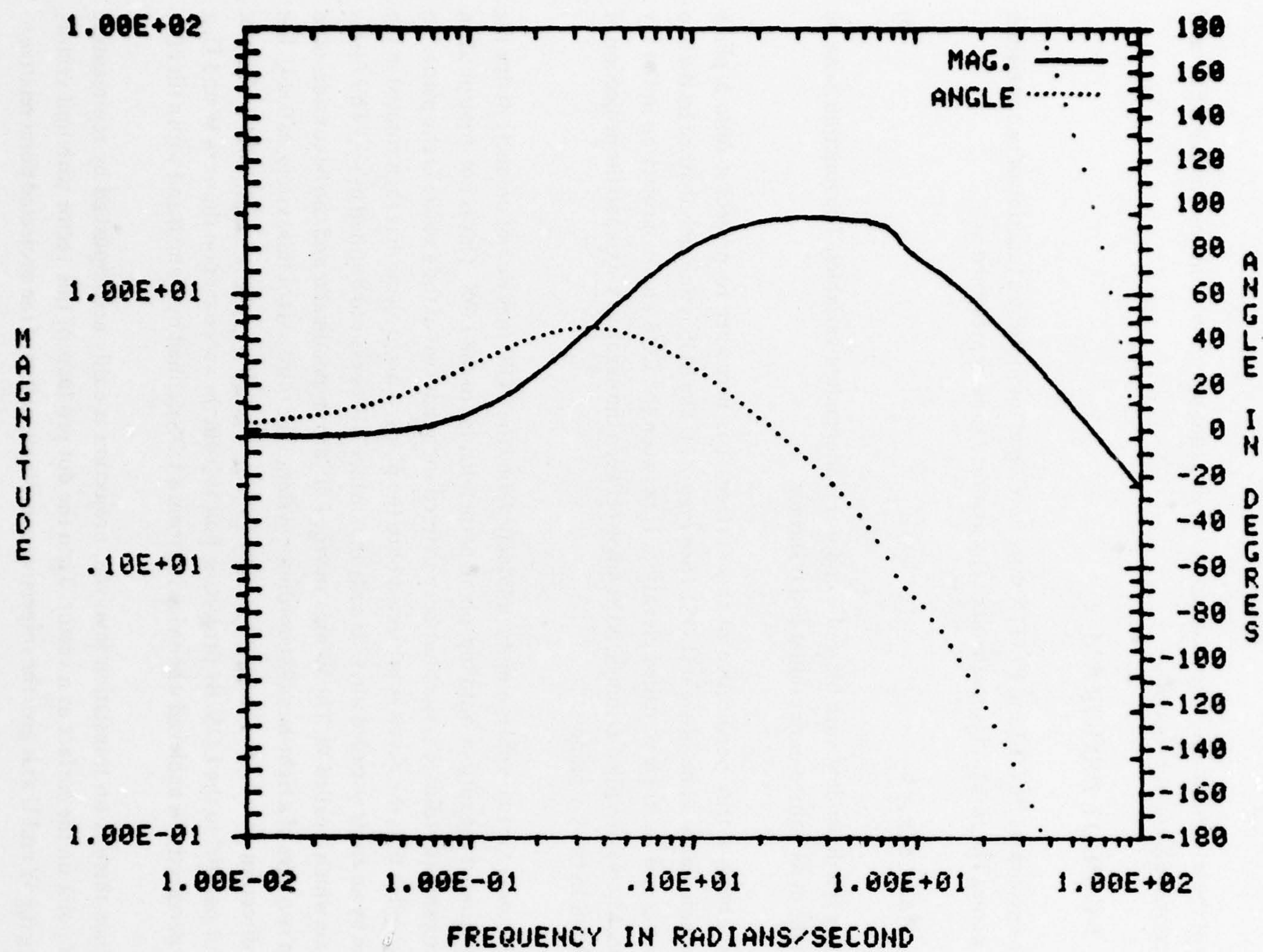


Figure 6. Frequency response for the continuous example.

- Maximum of 30 transfer function numerator or denominator coefficients per forward path

- Maximum of 100 frequency response points.

Any of these limitations can be changed by varying the appropriate dimensions in FRESP and the corresponding subroutines.

3. 3-D PLOT PROGRAM

The programs PLOT3 and PLOTX were developed for plotting a function of two variables in a plane. The function is mathematically assumed to be represented as

$$z = z(x,y) \tag{5}$$

where z is a dependent variable and x and y are independent variables. The function z can be thought of as a three-dimensional (3-D) surface.

To obtain a true perspective of this surface, it is necessary to project it onto a plane perpendicular to a line-of-sight (LOS). (See *Figure 7*). The LOS is uniquely defined by the two angles α and β. Angle β, called elevation, is between the LOS and its projection in the x-y plane, whereas α, called azimuth, is the angle between the negative y-axis and the projection of the LOS in the x-y plane.

Although the two angles α and β uniquely define the LOS, they do not uniquely define the projection of the surface onto the plane perpendicular to the LOS. This is not done until an orthogonal coordinate system has been defined such that two of the axes lie in the plane. To accomplish this, the z-axis is projected onto the plane. This projection is then rotated in the plane by an angle γ, called tilt. (Actually the surface will appear to be tilted by $-\gamma$.) This forms an axis which is called W. The W-axis and the LOS are perpendicular and can be considered to form two axes of a right-handed coordinate system. The third axis of this system, called U, lies in a direction parallel to the vector formed by crossing a unit vector parallel to W and a unit vector parallel to the LOS. By projecting points from the surface upon the axes W and U, a true perspective is achieved when viewing from a LOS defined by α and β and with a tilt of γ.

Since there is no translation involved, projection is easily accomplished by representing each point on the surface as a vector. Then the dot products of this vector with unit vectors along the W and U axis give the respective coordinate points of the projected point relative to these axes.

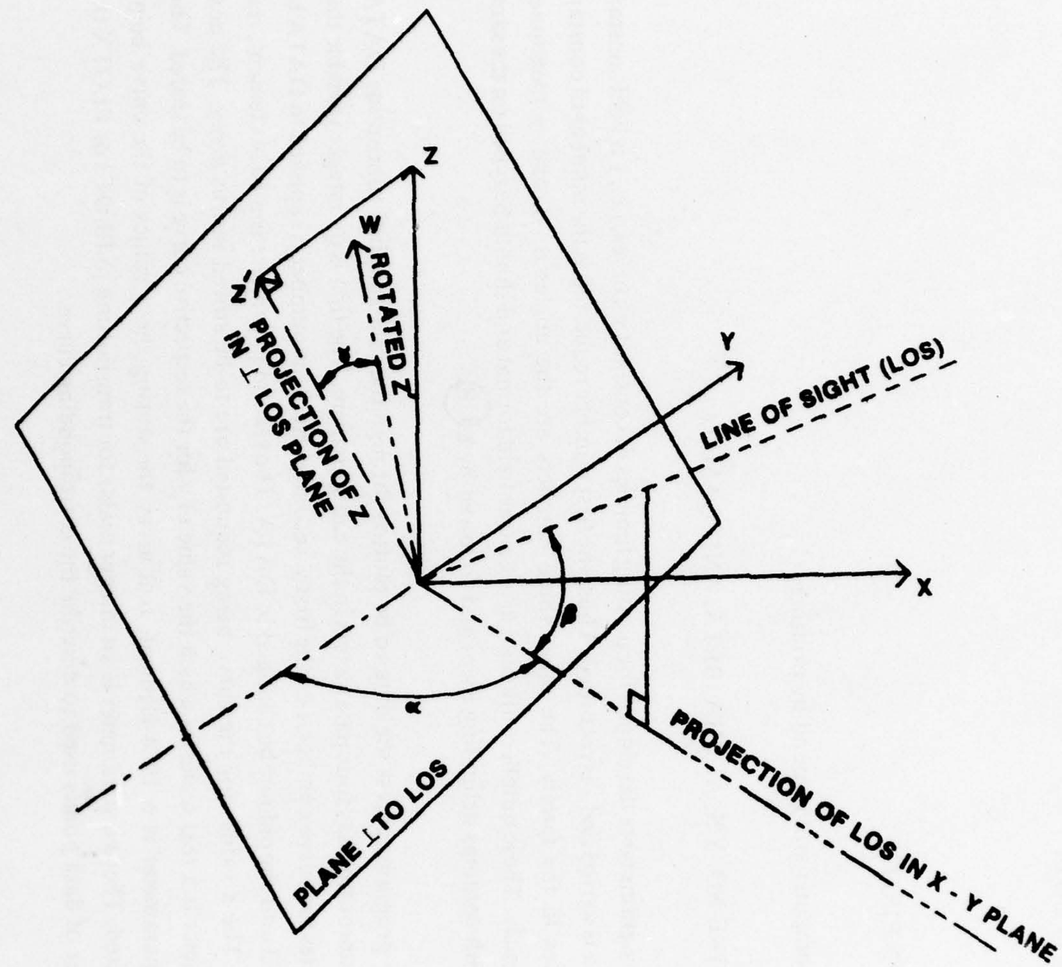


Figure 7. Pictorial for defining the LOS angle and tilt angle.

A. USING PLOT3 AND PLOTX

Assuming PLOT3 or PLOTX has been loaded into the HP-21MX, they can be run respectively with the commands

ON,PLOT3

and

ON,PLOTX.

The computer will respond by printing

GIVE NO. Y'S, ALPHA, BETA, AND GAMMA .

The programs were developed for plotting families of constant y-curves, i.e., y is held constant while x is varied and z is recorded. Thus, the first number requested is the number of constant y-curves in the family. The other three numbers are the angles α, β, and γ mentioned previously. These numbers are read using a free field format and should be typed on the same line with commas delimiting them and followed by a (CR).

The programs access the data to be plotted through the user supplied subroutine DATA. This subroutine has four parameters in the call statement. The first is an integer variable that indicates the curve number, i.e., the first y, second y, etc. This number is supplied to DATA by PLOT3 and should not be changed in DATA. The second parameter is a 1024 element, real array. The x values for the curve being requested are to be stored in this array. The next parameter is a real scalar in which the value of y for the respective curve is to be stored. The third parameter is a 1024 element, real array for storing the z values of the curve being requested. The last parameter is an integer scalar for transmitting to PLOT3 or PLOTX the number of data points used to describe the corresponding curve.

In essence the exchanges of data between PLOT3 or PLOTX and DATA are:

- PLOT3 or PLOTX gives DATA the curve number

- DATA returns the x's, a y value, the z's and the number of data points, i.e., the number of (z,x) pairs.

In developing the subrouting DATA the following rules should be adhered to:

- The x's must be either in ascending or descending order, i.e., random order of the x's will cause incorrect plotting.

- The curves are numbered starting with the lowest value of y if $-90° \leq \alpha \leq 90°$ and starting with the highest value of y if $90 < \alpha < 270$. (They are numbered starting with one and ending with the number of curves.)

B. EXAMPLE USE OF PLOT3 OR PLOTX

Consider the following function:

$$z_1(s) = 0.5 \left| \frac{s^2 + 4s + 53}{s^2 - 10s + 50} \right| \left| \frac{s}{s + 5} \right| \qquad (6)$$

where s is the complex variable $s = x + jy$. Thus, the function z can be represented by Equation (5). It is desired to plot z for x on the range of $-10 \leq x \leq 10$ and y on the range $-9.25 \leq y \leq 10.75$. Twenty constant y-curves are to be plotted. For each y-curve 101 (z,x) pairs are to be generated.

The SUBROUTINE DATA presented in the listings of both PLOT3 and PLOTX was written for generating the data to be plotted. This subroutine was attached to both PLOT3 and PLOTX; both were compiled and relocated. The session on the computer for loading and executing PLOTX is shown in *Table 4*. Initially, PLOT3 was occupying the computer memory. It was "OFF-ed" and PLOTX was loaded. PLOTX was then run with the ON, PLOTX command and it immediately requested the number of y's, α, β, and γ. The line 2Ø, 15, 1Ø, Ø was typed followed by a (CR). The execution began and finally the plot shown in *Figure 8* was produced. Using PLOT3 the same results were obtained.

C. LIMITATIONS AND COMMENTS ON PLOT3 AND PLOTX

The programs PLOT3 and PLOTX are, in essence, the same. The major difference is that PLOTX requires about 2048 less words of computer memory than does PLOT3. PLOTX was specifically developed to obtain this savings.

TABLE 4. COMPUTER SESSION FOR THE PLOT EXAMPLE

```
*OF,PLOT3,8
PLOT3 ABORTED

*LO,PLOTX
 APLDR: DONE- PLOTX

*ON,PLOTX

         GIVE NO. Y'S, ALPHA, BETA, AND GAMMA
20,15,10,0
```

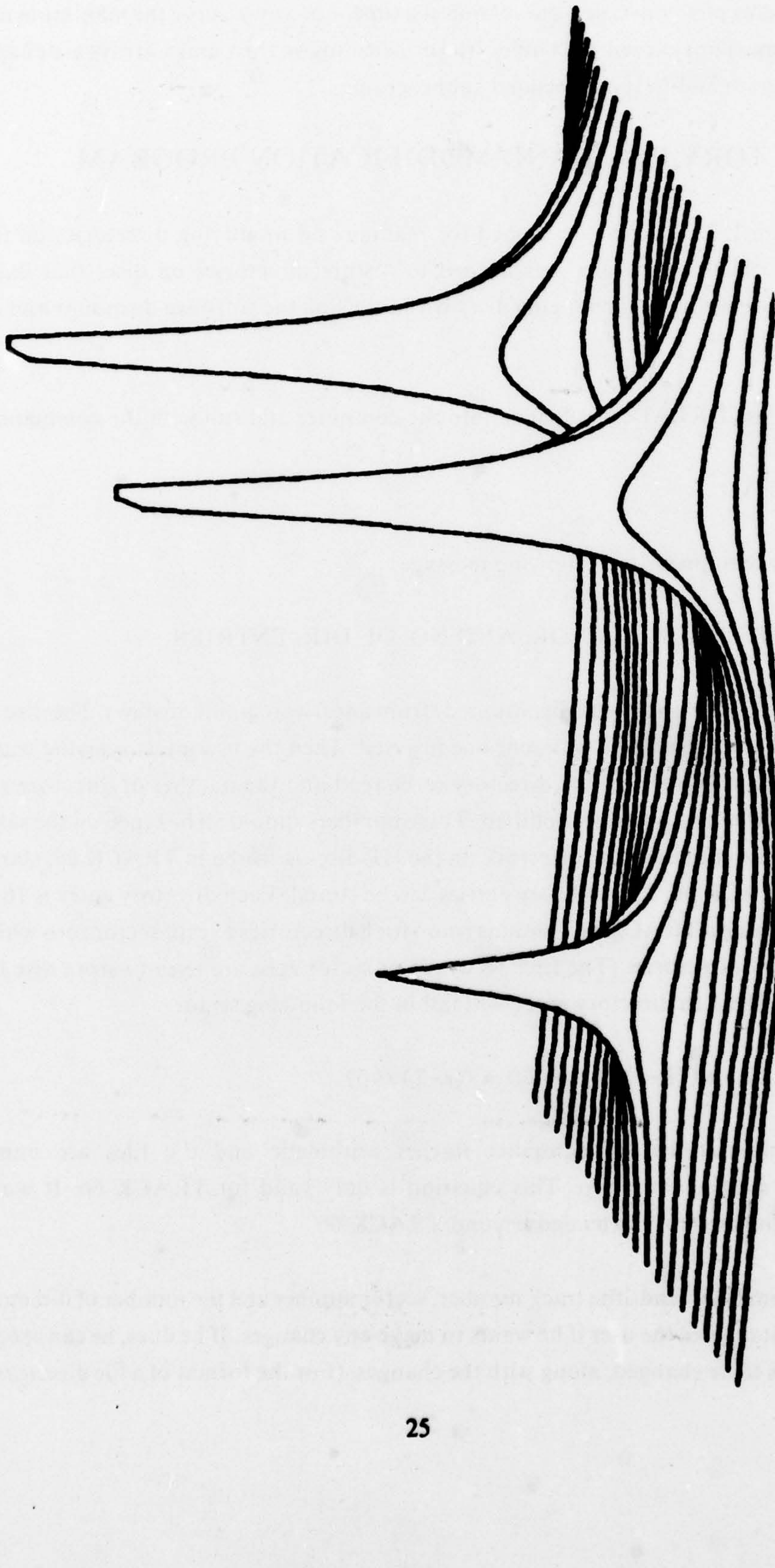


Figure 8. Example plot using PLOTX.

Both programs plot constant y-curves one at a time. For any y-curve the maximum number of (z,x) pairs must not exceed 1024 unless the dimensions on the z and x arrays are changed in the main program and in the associated subprograms.

4. DIRECTORY READ AND MODIFICATION PROGRAM

The program DREAD was developed for reading and modifying directories on the HP flexible discs. Such a program was needed to restore directories on discs that had been destroyed accidentally, e.g., changing discs without using the software dismount and mount routines.

In order to use DREAD, it is loaded into the computer and run with the command

*ON,DREAD

The computer then prints the following message

GIVE THE TRACK, SECTOR, AND NO. OF DIR. ENTRIES.

At this point the disc has been dismounted (from a software point of view). The disc can be removed from the drive and a different one inserted. Then the user must type the track and sector numbers of the part of the directory to be read and the number of directories in the sector to be viewed and possibly modified. These numbers should all be typed on the same line and separated by commas. The directory on the HP discs is stored in TRACK 66, starting at sector 0. In TRACK 66, 239 directory entries can be stored. Each directory entry is 16 words long. Each sector on TRACK 66 contains room for 8 directories except sector zero which has room for only 7 directories (The first 16 words of sector zero are used to store disc format information.) The Kth directory entry will fall in the following sector

```
IS = 14  *((K-1)/8) - 60 * ((K-1)/40)                                (7)
```

where all computations are computer integer arithmetic and the files are numbered consecutively starting with *two*. This equation is only valid for TRACK 66. It would be unusual that the directory extended beyond TRACK 66.

After the computer reads the track number, sector number and the number of directories to be displayed, it queries the user if he wants to make any changes. If he does, he can specify the word numbers to be changed, along with the changes. (For the format of a file directory entry see the RTE-M *Programmers Reference Manual*, Appendix F.) After a directory has been displayed and changes have been made, the changed directory entry is displayed. No changes have really been made at this point. When all the requested directory entries have been viewed, the process can be aborted by typing the number indicated.

APPENDIX A

PRECEDING PAGE NOT FILMED
BLANK

The following is a file manager listing of the symbolic version of the program FRESP. The symbolic listing is contained on the SYSCPY disc in the file $FRESP.

```
$FRPOT T=00004 IS ON CR32760 USING 00045 BLKS R=5648

FTN,L,T
      PROGRAM FRESP
C
C PROGRAM FOR CALCULATING FREQUENCY RESPONSE FOR
C      SAMPLED-DATA OR CONTINUOUS SYSTEMS
C
      DIMENSION ID(15),FREQRS(100),OMEGA(100),OMEGAW(100)
      DIMENSION AMAG(100),ANG(100)
      COMPLEX S,G,PHI,FREQRS,SIGMA,PHICAL,PHIDT
      COMMON XCOMN(5,30),YCOMN(5,30),NUMC(5),NRATOR(5,10)
     1  ,NDENOM(5,10),KNR(5),KDR(5)
      INTEGER FREQPT
      DATA IBLANK,ISDFR,ICDFR/2H  ,2HSR,2HCR/
      PHI(T,S)=(1.0-CEXP(-T*S))/S
      IXWRD(IPOS)=IPOS-(IPOS/32)*32+64+256*((IPOS/32)+32)
      IYWRD(IPOS)=IXWRD(IPOS)+32
      POS(W,IH,IL,JH,JL)=FLOAT(IL)+((W-FLOAT(JL))/(FLOAT(JH-JL)))*
     1   FLOAT(IH-IL)
C
C  READ DESCRIPTION OF SYSTEM
C
      WRITE(1,301)
301   FORMAT("GIVE THE MODE (EITHER SR OR CR)")
      READ(1,101) MODE
101   FORMAT(A2)
      IF(MODE.NE.ISDFR) MODE=ICDFR
      WRITE(1,200) MODE
200   FORMAT("THE PROGRAM IS IN THE ",A2," MODE")
      WRITE(1,302)

*
```

```
302   FORMAT("GIVE THE NO. OF CHANNELS")
      READ(1,*) KCHNL
      WRITE(1,303)
303   FORMAT("GIVE THE NO. OF TRANSFER FUNCTIONS IN EACH CHANNEL")
      READ(1,*) (NUMC(I),I=1,KCHNL)
      DO 2 I=1,KCHNL
      KNAT=NUMC(I)
      WRITE(1,304) I
304   FORMAT("GIVE THE NUMERATOR ORDERS FOR CHANNEL NO. ",I2)
      READ(1,*) (NRATOR(I,J),J=1,KNAT)
      WRITE(1,305) I
305   FORMAT("GIVE THE DENOMINATOR ORDER FOR CHANNEL NO. ",I2)
2     READ(1,*) (NDENOM(I,J),J=1,KNAT)
      WRITE(1,306)
306   FORMAT("GIVE THE NO. OF POINTS TO BE COMPUTED AND NO. OF "
     1 "TERMS IN SERIES")
      READ(1,*) FREQPT,NOTSIR
      WRITE(1,307)
307   FORMAT("GIVE START FREQ., END FREQ., SAMPLE PERIOD,"
     1 ," TIME DELAY, AND GAIN")
      READ(1,*) FSTART,FRQEND,SAMPRD,DLFUNC,GAIN
C
C   READ COEFFICIENTS OF BLOCKS
C
      DO 3 I=1,KCHNL
      LAMP=NUMC(I)
      KNR(I)=0
      KDR(I)=0
      DO 4 J=1,LAMP
      KNR(I)=KNR(I)+NRATOR(I,J)+1
4     KDR(I)=KDR(I)+NDENOM(I,J)+1

*
```

```
3     CONTINUE
      DO 5 I=1,KCHNL
      LNC=KNR(I)
      LDC=KDR(I)
      WRITE(1,308) I
308   FORMAT("GIVE NUMERATOR COEFFICIENTS OF CHANNEL NO. ",I2)
      READ(1,*) (XCOMN(I,J),J=1,LNC)
      WRITE(1,309) I
309   FORMAT("GIVE DENOMINATOR COEFFICIENT FOR CHANNEL NO. ",I2)
5     READ(1,*) (YCOMN(I,J),J=1,LDC)
      IF(MODE.EQ.ICDFR)SAMPRD=1.0
      OMEGAS=3.14159265*2.0/SAMPRD
      IF(MODE.EQ.ICDFR) NOTSIR=0
      IF(MODE.EQ.ISDFR) FRQEND=OMEGAS/2.0
      IF(MODE.EQ.ISDFR) WRITE(1,310)
310   FORMAT("FRQEND IS CHANGED TO OMEGAS/2 SINCE MODE = SR")
      FACTOR= (FRQEND/FSTART)**(1.0/(FLOAT(FREQPT)-1.0))
C
C   START FREQUENCY RESPONSE COMPUTATION LOOP
C
      WRITE(1,212)
212   FORMAT(3X,"OMEGA",10X,"OMEGAW",9X,"MAG.",11X,"ANGLE")
      OMEGA(1)=FSTART
      KOUNT=0
10    KOUNT=KOUNT+1
      I=KOUNT
      IF(I.GT.100)GO TO 11
      IF(OMEGA(I).GT.FRQEND+FRQEND/1.0E+4) GO TO 11
      SIGMA=CMPLX(0.,0.)
      NNN=2*NOTSIR+1

*
```

```
      DO 7 NN=1,NNN
      N=-NOTSIR+NN-1
      S=CMPLX(0.,OMEGA(I)+N*OMEGAS)
      IF(MODE.EQ.ICDFR) GO TO 12
      PHICAL=PHI(SAMPRD,S)
12    IF(MODE.EQ.ICDFR) PHICAL=1.0
      PHIDT=PHICAL*CEXP(-DLFUNC*S)*GAIN
      CALL EVAL(S,G,KCHNL)
7     SIGMA=SIGMA+PHIDT*G
      FREQRS(KOUNT)=(1.0/SAMPRD)*SIGMA
      OMEGAW(I)=TAN(OMEGA(I)*SAMPRD/2.0)
      XMAG=CABS(FREQRS(I))
      XANG=ATAN2(AIMAG(FREQRS(I)),REAL(FREQRS(I)))*57.295779265
      WRITE(1,213) OMEGA(I),OMEGAW(I),XMAG,XANG
      AMAG(I)=XMAG
      ANG(I)=XANG
213   FORMAT(G10.3,5X,G10.3,5X,G10.3,5X,G10.3)
      OMEGA(I+1)= OMEGA(I)*FACTOR
      GO TO 10
11    WRITE(1,500)
500   FORMAT("DO YOU WANT TO PLOT?  (YES OR NO)")
      READ(1,501) M1
501   FORMAT(A2)
      IF(M1.EQ.54505B)GO TO 503
      STOP
503   M1=15414B
      WRITE(1,501) M1
      I=I-1
      NPTS=I
      XMAX=AMAX(I,OMEGA)
      XMIN=AMIN(I,OMEGA)
      YMAX=AMAX(I,AMAG)
*
```

```
      YMIN=AMIN(I,AMAG)
      ZMAX=AMAX(I,ANG)
      ZMIN=AMIN(I,ANG)
C     PROGRAM PLOT
C     READ(1,*)YMAX,YMIN,XMAX,XMIN,ZMAX,ZMIN
C     READ(1,*)NPTS
C     READ(1,*) (AMAG(I),ANG(I),OMEGA(I),I=1,NPTS)
      IENT=29
      IEXT=31
C  DEFINE CORNERS OF GRAPH
      IX1=IXWRD(185)
      IY1=IYWRD(94)
      IX2=IX1
      IY2=IYWRD(778)
      IX3=IXWRD(909)
      IY3=IY2
      IX4=IX3
      IY4=IY1
C  FORM BORDER OF GRAPH
      WRITE(1,13)IENT,IY1,IX1,IY2,IX2,IY3,IX3,IY4,IX4,IY1,IX1,IEXT
13    FORMAT(12A2)
      IX1=IXWRD(756)
      IY1=IYWRD(742)
      IX2=IXWRD(826)
      IY2=IYWRD(752)
      IY3=IY2
      IX3=IXWRD(882)
C  PUT LEGEND ON GRAPH
      WRITE(1,15) IENT,IY1,IX1,IEXT,IENT,IY2,IX2,IY3,IX3,IEXT
15    FORMAT(4A2,"MAG.",6A2)
      IX1=IXWRD(742)
      IY1=IYWRD(710)
      WRITE(1,16) IENT,IY1,IX1,IEXT

*
```

```
16    FORMAT(4A2,"ANGLE")
      IY1=IYWRD(723)
      IX1=IXWRD(820)
      IY2=IY1
      IX2=IX1
      DO 19 I=1,9
      WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
      IX1=IXWRD(820+I*7)
19    IX2=IX1
C  PUT LABELS ON AXES
       WRITE(1,17)
17    FORMAT(//////71X,"A"/71X,"N"/71X,"G"/"M",70X,"L"/"A",70X,
     1   "E"/"G"/"N",70X,"I"/"I",70X,"N"/"T"/"U",70X,"D"/"D",70X
     2   "E"/"E",70X,"G"/71X,"R"/71X,"E"/71X,"S")
      IY1=IYWRD(0)
      IX1=IXWRD(0)
      WRITE(1,18)IENT,IY1,IX1,IEXT
18    FORMAT(4A2,25X,"FREQUENCY IN RADIANS/SECOND")
C  DETERMINE NO. OF DECADES OF MAG. AND FREQ. AXES
      XMAX=XMAX-XMAX/100000.
      XMIN=XMIN+XMIN/100000.
      YMAX=YMAX-YMAX/100000.
      YMIN=YMIN+YMIN/100000.
      XMAX=ALOGT(XMAX)
      XMIN=ALOGT(XMIN)
      YMAX=ALOGT(YMAX)
      YMIN=ALOGT(YMIN)
      IYMAX=IFIX(YMAX)+1
      IXMAX=IFIX(XMAX)+1
      IYMIN=IFIX(YMIN)-1
      IXMIN=IFIX(XMIN)-1
      IDECX=IXMAX-IXMIN
      IDECY=IYMAX-IYMIN
```

*

```
C  DETERMINE DECADE INCREMENT FOR MAG. AND FREQ. AXES
      ANCX=724./FLOAT(IDECX)
      ANCY=684./FLOAT(IDECY)
C  PUT NUMBERS ON MAG. AXIS
      IY1=IYWRD(87)
      IX1=IXWRD(0)
      IUSE=IYMIN
      IDY1=IDECY+1
      DO 50 J=1,IDY1
      XNUM=10.**IUSE
      WRITE(1,40) IENT,IY1,IX1,IEXT,XNUM
40    FORMAT(4A2,2X,E9.2)
      IUSE=IUSE+1
      APTS=FLOAT(J)*ANCY
50    IY1=IYWRD(87+IFIX(APTS))
C  PUT MAJOR TIC MARKS ON MAG. AXIS
      IY1=IYWRD(94)
      IX1=IXWRD(173)
      IY2=IY1
      IX2=IXWRD(185)
      DO 65 M=1,2
      DO 60 J=1,IDECY
      WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
55    FORMAT(6A2)
      APTS=FLOAT(J)*ANCY
      IY1=IYWRD(94+IFIX(APTS))
60    IY2=IY1
      IX1=IXWRD(897)
65    IX2=IXWRD(909)
C  PUT MINOR TIC MARKS ON MAG. AXIS
      IY1=IYWRD(94)
      IX1=IXWRD(177)

*
```

```
      IY2=IY1
      IX2=IXWRD(185)
      DO 75 M=1,2
      DO 70 J=1,IDECY
      DO 70 I=1,8
      Z=ALOGT(FLOAT(I+1))
      APTS=FLOAT(J-1)*ANCY+Z*ANCY
      IY1=IYWRD(94+IFIX(APTS))
      IY2=IY1
70    WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
      IX1=IXWRD(901)
75    IX2=IXWRD(909)
C  PUT NUMBERS ON FREQUENCY AXIS
      IY1=IYWRD(47)
      IX1=IXWRD(75)
      IUSE=IXMIN
      IDX1=IDECX+1
      DO 80 J=1,IDX1
      XNUM=10.**IUSE
      IUSE=IUSE+1
      WRITE(1,40) IENT,IY1,IX1,IEXT,XNUM
      APTS=FLOAT(J)*ANCX
80    IX1=IXWRD(75+IFIX(APTS))
C  PUT MAJOR TIC MARKS ON FREQUENCY AXIS
      IY1=IYWRD(84)
      IX1=IXWRD(185)
      IY2=IYWRD(94)
      IX2=IX1
      DO 85 M=1,2
      DO 84 J=1,IDECX
      WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT

*
```

```
          APTS=FLOAT(J)*ANCX
          IX1=IXWRD(185+IFIX(APTS))
   84     IX2=IX1
          IY1=IYWRD(768)
   85     IY2=IYWRD(778)
   C  PUT MINOR TIC MARKS ON FREQUENCY AXIS
          IY1=IYWRD(87)
          IX1=IXWRD(185)
          IY2=IYWRD(94)
          IX2=IX1
          DO 95 M=1,2
          DO 93 J=1,IDECX
          DO 93 I=1,8
          Z=ALOGT(FLOAT(I+1))
          APTS=FLOAT(J-1)*ANCX+Z*ANCX
          IX1=IXWRD(185+IFIX(APTS))
          IX2=IX1
   93     WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
          IY1=IYWRD(771)
   95     IY2=IYWRD(778)
   C  PUT NUMBERS ON ANGLE AXIS
          IZMAX=IFIX(ZMAX/10.)+1
          IZMIN=IFIX(ZMIN/10.)-1
          ITEN=IZMAX-IZMIN
          ANCZ=684./FLOAT(ITEN)
          IY1=IYWRD(87)
          IX1=IXWRD(925)
          IUSE=IZMIN
          IDZ1=ITEN+1
          KJUMP=1
          DO 100 J=1,IDZ1
          INUM=10*IUSE

*
```

```
      IF(KJUMP.LT.0)GO TO 98
      WRITE(1,96) IENT,IY1,IX1,IEXT,INUM
98    KJUMP=-KJUMP
96    FORMAT(4A2,I4)
      IUSE=IUSE+1
      APTS=FLOAT(J)*ANCZ
100   IY1=IYWRD(97+IFIX(APTS))
C  PUT TIC MARKS ON ANGLE AXIS
      IX1=IXWRD(909)
      IX2=IXWRD(919)
      DO 110 M=1,2
      IY1=IYWRD(94)
      IY2=IY1
      DO 105 J=1,IDZ1
      WRITE(1,55) IENT,IY1,IX1,IY2,IX2,IEXT
      APTS=FLOAT(J)*ANCZ
      IY1=IYWRD(94+IFIX(APTS))
105   IY2=IY1
      IX1=IXWRD(185)
110   IX2=IXWRD(195)
      IZMAX=IZMAX*10
      IZMIN=IZMIN*10
      IENT=29
      IEXT=31
      DO 49 I=1,NPTS
      W=ALOGT(AMAG(I))
      Y1=POS(W,778,94,IYMAX,IYMIN)
      W=ALOGT(OMEGA(I))
      X1=POS(W,909,185,IXMAX,IXMIN)
      IY1=IYWRD(IFIX(Y1))
      IX1=IXWRD(IFIX(X1))

*
```

```
      IF(I.LE.1)GO TO 45        *
      WRITE(1,30)IENT,IY1,IX1,IY2,IX2,IEXT
30    FORMAT(6A2)
45    IY2=IY1
      IX2=IX1
      Z1=POS(ANG(I),778,94,IZMAX,IZMIN)
      IZ1=IYWRD(IFIX(Z1))
49    WRITE(1,30)IENT,IZ1,IX1,IZ1,IX1,IEXT
      READ(1,*)XYZ
      STOP
      END
      SUBROUTINE EVAL(S,G,KCHNL)
      DIMENSION CNUM(30),CDOM(30)
      COMPLEX S,G,FACT1,FACT2,RATIO
      COMMON XCOMN(5,30),YCOMN(5,30),NUMC(5),NRATOR(5,10),
     1  NDENOM(5,10),KNR(5),KDR(5)
      G=CMPLX(0.,0.)
      DO 10 I=1,KCHNL
      KCOMP=NUMC(I)
      KNOT=0
      KLOT=0
      RATIO=1.0
      DO 20 J=1,KCOMP
      NTR=NRATOR(I,J)+1
      NTD=NDENOM(I,J)+1
      DO 30 M=1,NTR
30    CNUM(M)=XCOMN(I,M+KNOT)
      DO 40 M=1,NTD
40    CDOM(M)=YCOMN(I,M+KLOT)
      KNOT=KNOT+NTR
      KLOT=KLOT+NTD
      K2=NTR-1
      K3=NTD-1
      CALL FUNC(CNUM,K2,S,FACT1)
```

```
      CALL FUNC(CDOM,K3,S,FACT2)
20    RATIO=(FACT1/FACT2)*RATIO
10    G=G+RATIO
      RETURN
      END
      SUBROUTINE FUNC(FW,K,X,SUM)
      DIMENSION FW(30)
      COMPLEX X,SUM
      SUM=CMPLX(FW(K+1),0.)
      IF(K.EQ.0) GO TO 20
      DO 10 II=1,K
      I=K-II+1
10    SUM=SUM*X+CMPLX(FW(I),0.)
20    RETURN
      END
      FUNCTION AMAX(N,X)
      DIMENSION X(100)
      AMAX=X(1)
      DO 10 I=1,N
      IF(X(I)-AMAX)10,10,5
5     AMAX=X(I)
10    CONTINUE
      RETURN
      END
      FUNCTION AMIN(N,X)
      DIMENSION X(100)
      AMIN=X(1)
      DO 10 I=1,N
      IF(X(I)-AMIN)5,10,10
5     AMIN=X(I)

*
```

```
10    CONTINUE
      RETURN
      END
      END$
:
```

APPENDIX B

The following is a file manager listing of PLOT3. It is contained on the disc SYSCPY and is located in the file $PLOT3.

PRECEDING PAGE NOT FILMED
BLANK

$PLOT3 T=00004 IS ON CR32760 USING 00010 BLKS R=5652

```
FTN,L,T
      PROGRAM PLOT4
      DIMENSION ZMAX(1024),ZMIN(1024),X(1024),Z(1024),
     1   U(3),W(3),Y1(50),P(8),W1(3),U1(8),S(3)
      RAD=57.2958
      KJUMP=0
      DO 5 I=1,1024
      ZMAX(I)=-5000.
5     ZMIN(I)=5000.
      WRITE(1,2)
2     FORMAT(/10X,36HGIVE NO. Y'S, ALPHA, BETA, AND GAMMA    )
      READ(1,*)N,ALPHA,BETA,GAMMA
C  GET MAX. AND MIN. VALUES BEFORE ROTATION
      I=1
      CALL DATA(I,X,Y,Z,NPTS)
      XMX=X(1)
      XMN=X(1)
      YMX=Y
      YMN=Y
      ZMX=Z(1)
      ZMN=Z(1)
      DO 11 I=1,N
      CALL DATA(I,X,Y,Z,NPTS)
      IF(Y.GT.YMX)YMX=Y
      IF(Y.LT.YMN)YMN=Y
      DO 8 J=1,NPTS
      IF(X(J).GT.XMX)XMX=X(J)
      IF(X(J).LT.XMN)XMN=X(J)
      IF(Z(J).GT.ZMX)ZMX=Z(J)
8     IF(Z(J).LT.ZMN)ZMN=Z(J)
```

*

```
11    CONTINUE
C   ASSIGN MAX. AND MIN VALUES FOR GRAPH
      KORGX=0
      KORGZ=0
      KXMAX=1000
      KZMAX=700
C   COMPUTATION OF UNIT VECTOR ALONG LINE OF SIGHT (LOS)
      THETA=ALPHA-90.
      P(1)=COS(BETA/RAD)*COS(THETA/RAD)
      P(2)=COS(BETA/RAD)*SIN(THETA/RAD)
      P(3)=COS((90.-BETA)/RAD)
C   PROJECTION OF Z-AXIS IN PERPENDICULAR PLANE OF LOS
C   THE W-AXIS IS FORMED HERE
      W1(1)=-P(1)*P(3)
      W1(2)=-P(2)*P(3)
      W1(3)=1.-P(3)*P(3)
      WMAG=0.
      DO 12 I=1,3
12    WMAG=WMAG+W1(I)*W1(I)
      WMAG=SQRT(WMAG)
      DO 14 I=1,3
14    W1(I)=W1(I)/WMAG
C   CROSS W1 AND LOS AXIS TO FORM U-AXIS
      U1(1)=W1(2)*P(3)-W1(3)*P(2)
      U1(2)=W1(3)*P(1)-W1(1)*P(3)
      U1(3)=W1(1)*P(2)-W1(2)*P(1)
C   TILT W AND U AXES IN PLANE
      DO 15 I=1,3
      W(I)=SIN(GAMMA/RAD)*U1(I)+COS(GAMMA/RAD)*W1(I)
15    U(I)=COS(GAMMA/RAD)*U1(I)-SIN(GAMMA/RAD)*W1(I)
C   COMPUTE PROJECTED MAX. AND MIN. VALUES
      W1(1)=XMX

*
```

```
      W1(2)=YMX
      W1(3)=ZMX
      CALL DOTP(W1,U,W,P(1),U1(1))
      W1(3)=ZMN
      CALL DOTP(W1,U,W,P(2),U1(2))
      W1(2)=YMN
      CALL DOTP(W1,U,W,P(3),U1(3))
      W1(3)=ZMX
      CALL DOTP(W1,U,W,P(4),U1(4))
      W1(1)=XMN
      CALL DOTP(W1,U,W,P(5),U1(5))
      W1(3)=ZMN
      CALL DOTP(W1,U,W,P(6),U1(6))
      W1(2)=YMX
      CALL DOTP(W1,U,W,P(7),U1(7))
      W1(3)=ZMX
      CALL DOTP(W1,U,W,P(8),U1(8))
      A=P(1)
      B=P(1)
      C=U1(1)
      D=U1(1)
      DO 16 I=2,8
      IF(P(I).GT.A)A=P(I)
      IF(P(I).LT.B)B=P(I)
      IF(U1(I).GT.C)C=U1(I)
 16   IF(U1(I).LT.D)D=U1(I)
      XMX=A
      XMN=B
      ZMX=C
      ZMN=D
C  PROJECT POINT INTO PLANE PERPENDICULAR TO LOS
      M1=15414B

*
```

```
      WRITE(1,98) M1
 98   FORMAT(A2)
      DO 7 J=1,N
      CALL DATA(J,X,Y,Z,NPTS)
      S(2)=Y
      DO 20 I=1,NPTS
      S(1)=X(I)
      S(3)=Z(I)
 20   CALL DOTP(S,U,W,X(I),Z(I))
 7    CALL PLOT3(XMX,XMN,ZMX,ZMN,Z,X,NPTS,ZMAX,ZMIN,
     1   KORGX,KXMAX,KORGZ,KZMAX,KJUMP)
      READ(1,*) XYZ
      STOP
      END
      SUBROUTINE DOTP(S,U,V,X,Y)
      DIMENSION U(3),V(3),S(3)
      X=0.
      Y=0.
      DO 10 I=1,3
      X=X+S(I)*U(I)
 10   Y=Y+S(I)*V(I)
      RETURN
      END
      SUBROUTINE PLOT3(XMAX,XMIN,ZMX,ZMN,Z,X,
     1   NPTS,ZMAX,ZMIN,KORGX,KXMAX,KORGZ,KZMAX,KJUMP)
      DIMENSION ZMAX(1024),ZMIN(1024),Z(50),X(50)
      IXWRD(IPOS)=IPOS-(IPOS/32)*32+64+256*((IPOS/32)+32)
      IYWRD(IPOS)=IXWRD(IPOS)+32
      POS(W,IH,IL,AH,AL)=FLOAT(IL)+((W-AL)/(AH-AL))*FLOAT(IH-IL)
C  FILL EDGE POINTS FOR MAX. AND MIN. ARRAY
      XS1=X(1)
      XE1=X(NPTS)
      ZS1=Z(1)

*
```

```
      ZE1=Z(NPTS)                    *
      IF(KJUMP.EQ.0)GO TO 100
      X2=XS2
      X1=XS1
      Z2=ZS2
      Z1=ZS1
      KSKIP=0
10    CONTINUE
      KSTART=IFIX(POS(X1,KXMAX,KORGX,XMAX,XMIN))+1
      KEND=IFIX(POS(X2,KXMAX,KORGX,XMAX,XMIN))+1
      KPTS=KEND-KSTART
      IF(KPTS.EQ.0)GO TO 50
      KSGN=1
      IF(KPTS.LT.0)KSGN=-1
      KPTS=IABS(KPTS)
      DELX=(X2-X1)/FLOAT(KPTS)
      XV=X1
      SLOPE=(Z2-Z1)/(X2-X1)
      KEND=KPTS+1
      DO 20 I=1,KEND
      J=KSTART+KSGN*(I-1)
      ZV=SLOPE*(XV-X1)+Z1
      ZV=POS(ZV,KZMAX,KORGZ,ZMX,ZMN)
      IF(ZMAX(J).LT.ZV)ZMAX(J)=ZV
      IF(ZMIN(J).GT.ZV)ZMIN(J)=ZV
20    XV=XV+DELX
50    IF(KSKIP.EQ.1)GO TO 100
      X2=XE2
      X1=XE1
      Z2=ZE2
      Z1=ZE1
      KSKIP=1
      GO TO 10
100   KJUMP=1
```

```
      XS2=XS1                        *
      XE2=XE1
      ZS2=ZS1
      ZE2=ZE1
C  FILL-IN POINTS AND PLOT
      IENT=29
      IEXT=31
      I=0
290   I=I+1
      IF(I.GE.NPTS)RETURN
      JX=0
      KSTART=IFIX(POS(X(I),KXMAX,KORGX,XMAX,XMIN))+1
      KEND=IFIX(POS(X(I+1),KXMAX,KORGX,XMAX,XMIN))+1
      NTOTAL=IABS(KEND-KSTART)
      IF(NTOTAL.EQ.0)RETURN
      DELX=(X(I+1)-X(I))/FLOAT(NTOTAL)
      KSGN=1
      IF(KSTART.GT.KEND)KSGN=-1
      K=KSTART-KSGN
      SLOPE=(Z(I+1)-Z(I))/(X(I+1)-X(I))
300   XV=FLOAT(JX)*DELX+X(I)
      K=K+KSGN
      JX=JX+1
      IF(KSGN)301,301,302
301   IF(XV.LT.X(I+1))GO TO 290
      GO TO 303
302   IF(XV.GT.X(I+1))GO TO 290
303   ZV=SLOPE*(XV-X(I))+Z(I)
      Z1=POS(ZV,KZMAX,KORGZ,ZMX,ZMN)
      IF(Z1.GE.ZMAX(K))GO TO 305
      IF(Z1.GT.ZMIN(K))GO TO 300
305   X1=POS(XV,KXMAX,KORGX,XMAX,XMIN)
      IZ1=IYWRD(IFIX(Z1))
      IX1=IXWRD(IFIX(X1+0.1))
```

```
      IF(Z1.GE.ZMAX(K))ZMAX(K)=Z1
      IF(Z1.LE.ZMIN(K))ZMIN(K)=Z1
310   XV=FLOAT(JX)*DELX+X(I)
      K=K+KSGN
      JX=JX+1
      IF(KSGN)311,311,312
311   IF(XV.LT.X(I+1))GO TO 320
      GO TO 314
312   IF(XV.GT.X(I+1))GO TO 320
314   KL=2
      ZV=SLOPE*(XV-X(I))+Z(I)
      Z3=POS(ZV,KZMAX,KORGZ,ZMX,ZMN)
      IF(Z3.GE.ZMAX(K))GO TO 315
      IF(Z3.GT.ZMIN(K))GO TO 325
315   Z2=Z3
      X2=POS(XV,KXMAX,KORGX,XMAX,XMIN)
      IF(Z2.GE.ZMAX(K))ZMAX(K)=Z2
      IF(Z2.LE.ZMIN(K))ZMIN(K)=Z2
      GO TO 310
320   XV=X(I+1)
      ZV=Z(I+1)
      Z2=POS(ZV,KZMAX,KORGZ,ZMX,ZMN)
      X2=POS(XV,KXMAX,KORGX,XMAX,XMIN)
      KL=1
325   IZ2=IYWRD(IFIX(Z2))
      IX2=IXWRD(IFIX(X2+0.1))
      WRITE(1,328)IENT,IZ1,IX1,IZ2,IX2,IEXT
328   FORMAT(6A2)
      IF(KL.EQ.1)GO TO 290.
      GO TO 300
      END
      SUBROUTINE DATA(II,X,Y,Z,NPTS)

*
```

```
      DIMENSION X(1024),Z(1024)
      COMPLEX S
      Z1(S)=.5*CABS((S**2+4.*S+53.)/(S**2-10.*S+50.)*(S/(S+5.)))
      IF(II.EQ.1)Y=-10.25
      Y=Y+1.0
      X(1)=-10.
      DO 10 I=1,101
      S=CMPLX(X(I),Y)
      Z(I)=Z1(S)
      IF(Z(I).GT.10.)Z(I)=10.
10    X(I+1)=X(I)+.2
      NPTS=101
      RETURN
      END
      END$

:
```

APPENDIX C

The following is a file manager listing of the program PLOTX and the user supplied subroutine DATA. These are contained on the M-II SYSTEM disc, and both are located in the file $PLOTX. The subroutine DATA generates the data for the plot shown in *Figure 8*. For plotting other data, the user must substitute DATA with the appropriate version and then compile and relocate.

```
       IF(Z(J).GT.ZMX)ZMX=Z(J)
8      IF(Z(J).LT.ZMN)ZMN=Z(J)
11     CONTINUE
C   ASSIGN MAX. AND MIN VALUES FOR GRAPH
       KORGX=0
       KORGZ=0
       KXMAX=1000
       KZMAX=700
C   COMPUTATION OF UNIT VECTOR ALONG LINE OF SIGHT (LOS)
       THETA=ALPHA-90.
       P(1)=COS(BETA/RAD)*COS(THETA/RAD)
       P(2)=COS(BETA/RAD)*SIN(THETA/RAD)
       P(3)=COS((90.-BETA)/RAD)
C   PROJECTION OF Z-AXIS IN PERPENDICULAR PLANE OF LOS
C   THE W-AXIS IS FORMED HERE
       W1(1)=-P(1)*P(3)
       W1(2)=-P(2)*P(3)
       W1(3)=1.-P(3)*P(3)
       WMAG=0.
       DO 12 I=1,3
12     WMAG=WMAG+W1(I)*W1(I)
       WMAG=SQRT(WMAG)
       DO 14 I=1,3
14     W1(I)=W1(I)/WMAG
C   CROSS W1 AND LOS AXIS TO FORM U-AXIS
       U1(1)=W1(2)*P(3)-W1(3)*P(2)
       U1(2)=W1(3)*P(1)-W1(1)*P(3)
       U1(3)=W1(1)*P(2)-W1(2)*P(1)
C   TILT W AND U AXES IN PLANE
       DO 15 I=1,3
       W(I)=SIN(GAMMA/RAD)*U1(I)+COS(GAMMA/RAD)*W1(I)
15     U(I)=COS(GAMMA/RAD)*U1(I)-SIN(GAMMA/RAD)*W1(I)

*
```

```
$PLOTX T=00004 IS ON CR32760 USING 00028 BLKS R=5652

FTN,L,T
      PROGRAM PLOT4
      DIMENSION IZMX(1024),IZMN(1024),X(1024),Z(1024),
     1   U(3),W(3),P(8),W1(3),U1(8),S(3)
      RAD=57.2958
      KJUMP=0
      DO 5 I=1,1024
      IZMX(I)=-5000
5     IZMN(I)=5000
      WRITE(1,2)
2     FORMAT(/10X,36HGIVE NO. Y'S, ALPHA, BETA, AND GAMMA    )
      READ(1,*)N,ALPHA,BETA,GAMMA
C   GET MAX. AND MIN. VALUES BEFORE ROTATION
      I=1
      CALL DATA(I,X,Y,Z,NPTS)
      XMX=X(1)
      XMN=X(1)
      YMX=Y
      YMN=Y
      ZMX=Z(1)
      ZMN=Z(1)
      DO 11 I=1,N
      CALL DATA(I,X,Y,Z,NPTS)
      IF(Y.GT.YMX)YMX=Y
      IF(Y.LT.YMN)YMN=Y
      DO 8 J=1,NPTS
      IF(X(J).GT.XMX)XMX=X(J)
      IF(X(J).LT.XMN)XMN=X(J)

*
```

```
C   COMPUTE PROJECTED MAX. AND MIN. VALUES
        W1(1)=XMX
        W1(2)=YMX
        W1(3)=ZMX
        CALL DOTP(W1,U,W,P(1),U1(1))
        W1(3)=ZMN
        CALL DOTP(W1,U,W,P(2),U1(2))
        W1(2)=YMN
        CALL DOTP(W1,U,W,P(3),U1(3))
        W1(3)=ZMX
        CALL DOTP(W1,U,W,P(4),U1(4))
        W1(1)=XMN
        CALL DOTP(W1,U,W,P(5),U1(5))
        W1(3)=ZMN
        CALL DOTP(W1,U,W,P(6),U1(6))
        W1(2)=YMX
        CALL DOTP(W1,U,W,P(7),U1(7))
        W1(3)=ZMX
        CALL DOTP(W1,U,W,P(8),U1(8))
        A=P(1)
        B=P(1)
        C=U1(1)
        D=U1(1)
        DO 16 I=2,8
        IF(P(I).GT.A)A=P(I)
        IF(P(I).LT.B)B=P(I)
        IF(U1(I).GT.C)C=U1(I)
16      IF(U1(I).LT.D)D=U1(I)
        XMX=A
        XMN=B
        ZMX=C
        ZMN=D
C   PROJECT POINT INTO PLANE PERPENDICULAR TO LOS
```

*

```
      M1=15414B
      WRITE(1,98) M1              *
98    FORMAT(A2)
      DO 7 J=1,N
      CALL DATA(J,X,Y,Z,NPTS)
      S(2)=Y
      DO 20 I=1,NPTS
      S(1)=X(I)
      S(3)=Z(I)
20    CALL DOTP(S,U,W,X(I),Z(I))
7     CALL PLOT3(XMX,XMN,ZMX,ZMN,Z,X,NPTS,IZMX,IZMN,
     1   KORGX,KXMAX,KORGZ,KZMAX,KJUMP)
      READ(1,*) XYZ
      STOP
      END
      SUBROUTINE DOTP(S,U,V,X,Y)
      DIMENSION U(3),V(3),S(3)
      X=0.
      Y=0.
      DO 10 I=1,3
      X=X+S(I)*U(I)
10    Y=Y+S(I)*V(I)
      RETURN
      END
      SUBROUTINE PLOT3(XMAX,XMIN,ZMX,ZMN,Z,X,
     1   NPTS,IZMX,IZMN,KORGX,KXMAX,KORGZ,KZMAX,KJUMP)
      DIMENSION IZMX(1024),IZMN(1024),Z(50),X(50)
      IXWRD(IPOS)=IPOS-(IPOS/32)*32+64+256*((IPOS/32)+32)
      IYWRD(IPOS)=IXWRD(IPOS)+32
      POS(W,IH,IL,AH,AL)=FLOAT(IL)+((W-AL)/(AH-AL))*FLOAT(IH-IL)
C   FILL EDGE POINTS FOR MAX. AND MIN. ARRAY
      XS1=X(1)
      XE1=X(NPTS)
      ZS1=Z(1)
      ZE1=Z(NPTS)
```

```
      IF(KJUMP.EQ.0)GO TO 100  *
      X2=XS2
      X1=XS1
      Z2=ZS2
      Z1=ZS1
      KSKIP=0
10    CONTINUE
      KSTART=IFIX(POS(X1,KXMAX,KORGX,XMAX,XMIN))+1
      KEND=IFIX(POS(X2,KXMAX,KORGX,XMAX,XMIN))+1
      KPTS=KEND-KSTART
      IF(KPTS.EQ.0)GO TO 50
      KSGN=1
      IF(KPTS.LT.0)KSGN=-1
      KPTS=IABS(KPTS)
      DELX=(X2-X1)/FLOAT(KPTS)
      XV=X1
      SLOPE=(Z2-Z1)/(X2-X1)
      KEND=KPTS+1
      DO 20 I=1,KEND
      J=KSTART+KSGN*(I-1)
      ZV=SLOPE*(XV-X1)+Z1
      IZ1=IFIX(POS(ZV,KZMAX,KORGZ,ZMX,ZMN))
      IF(IZMX(J).LT.IZ1)IZMX(J)=IZ1
      IF(IZMN(J).GT.IZ1)IZMN(J)=IZ1
20    XV=XV+DELX
50    IF(KSKIP.EQ.1)GO TO 100
      X2=XE2
      X1=XE1
      Z2=ZE2
      Z1=ZE1
      KSKIP=1
      GO TO 10
100   KJUMP=1
      XS2=XS1
```

```
      XE2=XE1
      ZS2=ZS1
      ZE2=ZE1
C  FILL-IN POINTS AND PLOT
      IENT=29
      IEXT=31
      I=0
290   I=I+1
      IF(I.GE.NPTS)RETURN
      JX=0
      KSTART=IFIX(POS(X(I),KXMAX,KORGX,XMAX,XMIN))+1
      KEND=IFIX(POS(X(I+1),KXMAX,KORGX,XMAX,XMIN))+1
      NTOTAL=IABS(KEND-KSTART)
      IF(NTOTAL.EQ.0)RETURN
      DELX=(X(I+1)-X(I))/FLOAT(NTOTAL)
      KSGN=1
      IF(KSTART.GT.KEND)KSGN=-1
      K=KSTART-KSGN
      SLOPE=(Z(I+1)-Z(I))/(X(I+1)-X(I))
300   XV=FLOAT(JX)*DELX+X(I)
      K=K+KSGN
      JX=JX+1
      IF(KSGN)301,301,302
301   IF(XV.LT.X(I+1))GO TO 290
      GO TO 303
302   IF(XV.GT.X(I+1))GO TO 290
303   ZV=SLOPE*(XV-X(I))+Z(I)
      IZ1=IFIX(POS(ZV,KZMAX,KORGZ,ZMX,ZMN))
      IF(IZ1.GE.IZMX(K))GO TO 305
      IF(IZ1.GT.IZMN(K))GO TO 300
305   X1=POS(XV,KXMAX,KORGX,XMAX,XMIN)
      IF(IZ1.GE.IZMX(K))IZMX(K)=IZ1
      IF(IZ1.LE.IZMN(K))IZMN(K)=IZ1

*
```

```
      X2=X1
      IZ2=IZ1
310   XU=FLOAT(JX)*DELX+X(I)
      K=K+KSGN
      JX=JX+1
      IF(KSGN)311,311,312
311   IF(XU.LT.X(I+1))GO TO 320
      GO TO 314
312   IF(XU.GT.X(I+1))GO TO 320
314   KL=2
      ZU=SLOPE*(XU-X(I))+Z(I)
      IZ3=IFIX(POS(ZU,KZMAX,KORGZ,ZMX,ZMN))
      IF(IZ3.GE.IZMX(K))GO TO 315
      IF(IZ3.GT.IZMN(K))GO TO 325
315   X2=POS(XU,KXMAX,KORGX,XMAX,XMIN)
      IZ2=IZ3
      IF(IZ2.GE.IZMX(K))IZMX(K)=IZ2
      IF(IZ2.LE.IZMN(K))IZMN(K)=IZ2
      GO TO 310
320   XU=X(I+1)
      ZU=Z(I+1)
      IZ2=IFIX(POS(ZU,KZMAX,KORGZ,ZMX,ZMN))
      X2=POS(XU,KXMAX,KORGX,XMAX,XMIN)
      KL=1
325   IX2=IXWRD(IFIX(X2+0.1))
      IZ2=IYWRD(IZ2)
      IX1=IXWRD(IFIX(X1+0.1))
      IZ1=IYWRD(IZ1)
      WRITE(1,328)IENT,IZ1,IX1,IZ2,IX2,IEXT
328   FORMAT(6A2)
      IF(KL.EQ.1)GO TO 290
      GO TO 300
      END

*
```

```
      SUBROUTINE DATA(II,X,Y,Z,NPTS)
      DIMENSION X(1024),Z(1024)
      COMPLEX S
      Z1(S)=.5*CABS((S**2+4.*S+53.)/(S**2-10.*S+50.)*(S/(S+5.)))
      IF(II.EQ.1)Y=-10.25
      Y=Y+1.0
      X(1)=-10.
      DO 10 I=1,101
      S=CMPLX(X(I),Y)
      Z(I)=Z1(S)
      IF(Z(I).GT.10.)Z(I)=10.
10    X(I+1)=X(I)+.2
      NPTS=101
      RETURN
      END
      END$
:
```

APPENDIX D

The following is a file management listing of the symbolic version of the program DREAD. This program is located on the disc SYSCPY. The symbolic version is in file $DREAD.

```
$DREAD T=00004 IS ON CR32760 USING 00007 BLKS R=0245

FTN,L,T
      PROGRAM DREAD
      DIMENSION IBUF(128)
      CALL DCMC(1,-2,0)
      WRITE(1,6)
6     FORMAT(47HGIVE THE TRACK, SECTOR, AND NO. OF DIR. ENTRIES
      READ(1,*) IT,IS,NFDS
      CALL DCMC(0,2,0)
C  READ TRACT IT AND SECTOR IS
      CALL EXEC(1,2,IBUF,128,IT,IS)
C  GET CONTENTS OF A AND B REG.'S
      CALL ABREG(IA,IB)
      IA=IAND(IA,377B)
      IF(IA)50,10,50
10    IF(IS.NE.0)GO TO 11
      WRITE(1,9)
9     FORMAT(/20X,10HDISC INFO. )
      WRITE(1,15) (IBUF(I),I=1,16)
15    FORMAT(3A2,1X,13(I4,1X))
      KM=0
      GO TO 12
11    KM=-16
12    CONTINUE
      DO 20 J=1,NFDS
      WRITE(1,14) J
14    FORMAT(/20X,8HDIR. NO.,I2)
      KS=J*16+1+KM
      KE=J*16+16+KM
      WRITE(1,16) (IBUF(I),I=KS,KE)

*
```

```
16    FORMAT(3A2,1X,5(I4,1X),A2,7(1X,I4))
      WRITE(1,17) J
17    FORMAT(/19HFOR CHANGES IN DIR. ,I2,1X,21HGIVE WRD. NO.,CHANG
     1  ,9HOR GIVE 0)
19    READ(1,*) I1,I2
      IF(I1.LE.0)GO TO 22
      I1=I1+16*J
      IBUF(I1)=I2
      GO TO 19
22    WRITE(1,23)
23    FORMAT(/30HTHIS IS THE WAY IT WILL LOOK.   )
      WRITE(1,16) (IBUF(I),I=KS,KE)
20    CONTINUE
      WRITE(1,25)
25    FORMAT(/44HDO YOU REALLY WANT TO DO THIS? (0-YES. 1-NO)     )
      READ(1,*)KX
      IF(KX.NE.0)STOP
      CALL EXEC(2,2,IBUF,128,IT,IS)
      CALL ABREG(IA,IB)
      IA=IAND(IA,377B)
      IF(IA.NE.0)GO TO 50
      STOP
50    WRITE(1,55) IA
55    FORMAT(K5)
      STOP
      END
      END$

:
```

ABBREVIATIONS AND SYMBOLS

α	Azimuth angle of the LOS
β	Elevation angle of the LOS
$\vert \quad \vert$	Absolute value
Ⓒ CR	Carriage Return
T.F.	Transfer Function
j	$\sqrt{-1}$ unless otherwise indicated
LOS	Line-of-sight
M-II	RTE-M II
ω	Frequency in rad/sec
ω_S	Sampling frequency in rad/sec
Π	Product if a large symbol; Pi if a small symbol
Σ	Summation
T	Sampling Period in seconds
3-D	Three-dimensional

DISTRIBUTION

	No. of Copies
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
US Army Materiel Systems Analysis Activity ATTN: DRXSY-MP Aberdeen Proving Ground, Maryland 21005	2
IIT Research Institute ATTN: GACIAC 10 West 35th Street Chicago, Illinois 60616	1
DRSMI-LP, Mr. Voigt	1
DRSMI-TBD (R&D)	3
-TI (R&D) (Reference Copy)	1
-TI (R&D) (Record Set)	1
-T (R&D), Dr. Kobler	1
-TGC	5