NASA TM X-62000

September 25, 1967

NASA TM X-62000

DIGITAL FILTER SYNTHESIS PROGRAM

By Robert M. Munoz and Richard A. Moyer

Ames Research Center, NASA
Moffett Field, Calif., 94035

ERRATA

NASA Technical Memorandum X-62000

DIGITAL FILTER SYNTHESIS PROGRAM

By Robert M. Munoz and Richard A. Moyer

Page 6 - Equation 20 should read:

$$O(Z) = -\frac{O(Z)\,B_1\,Z^{-1}}{B_0} - \frac{O(Z)\,B_2\,Z^{-2}}{B_0} + \frac{I(Z)\,A_0}{B_0} + \frac{I(Z)\,A_1\,Z^{-1}}{B_0} + \frac{I(Z)\,A_2\,Z^{-2}}{B_0}$$

Page 6 - Equation 21 should read:

$$O(Z) = O(Z)\,[-B_1'\,Z^{-1} - B_2'\,Z^{-2}] + I(Z)\,[A_0' + A_1'\,Z^{-1} + A_2'\,Z^{-2}]$$

Page 12 - example problem load deck second card from the end should read:

1. 12.5663706 10. .001

Page 13 - Appendix B
Example problem print-out third line should read:

1.0 12.57 10.00 .00100

TABLE OF CONTENTS

TECHNICAL MEMORANDUM X-62000

DIGITAL FILTER SYNTHESIS PROGRAM

By Robert M. Munoz and Richard A. Moyer

Ames Research Center
Moffett Field, Calif. 94035

ABSTRACT

In order to perform computations on the differential equations of continuous time varying quantities using the general purpose digital computer, it is necessary to form the difference equations or computational algorithms that approximate the differential equations. To do this, the "Digital Filter Synthesis Program" has been written. This program written in Fortran IV uses the bilinear transform method to approximate linear differential equations with constant coefficients. It allows inputs to be presented as functions of "s" (complex frequency) in either the factored or unfactored form and it presents the coefficients of the difference equations as intermediate outputs. The program tabulates input and output data lists representing the discrete form of time varying input and output quantities for the transfer functions represented by the differential equations. A subroutine for graphical display of the outputs by use of the print plot technique is also included in the program.

TECHNICAL MEMORANDUM X-62000

DIGITAL FILTER SYNTHESIS PROGRAM

By Robert M. Munoz and Richard A. Moyer

SUMMARY

In this report, a discussion of the theory and operation of a digital filter synthesis program is given. This program was written specifically to aid in the computer simulation studies of space instrument systems but it has broad general application. It allows any continuous function of a complex variable to be expressed in approximate form as a computational algorithm or difference equation.

INTRODUCTION

Digital filtering has received a great amount of attention in recent years and a number of methods have evolved for digital filter synthesis. Rader and Gold (ref. 1) have given a good summary of many of these methods and Kuo and Kaiser (ref. 2) have dedicated a chapter of their book to the same topics. In this report, a brief description will be given of the automatic implementation of one method of digital filter synthesis originated by Steiglitz (ref. 3) and discussed by Rader and Gold. This bilinear transform method allows synthesis of the difference equation for digital filtering with the initial specifications for filter performance given in terms of the analog prototype in the frequency domain. In other words, the digital filter can be derived from $F(s)$ the transfer function for the real time analog equivalent network. The pertinent mathematical results are also summarized briefly.

The digital filter synthesis program can perform the operation of digital filtering on any input data once the difference equation has been developed. Since the theory underlying synthesis of the analog prototype is well developed (see refs. 4 and 5), one can design a digital filter with any desired frequency or transient response by the use of this program.

The functions performed by the digital filter synthesis program are illustrated in figure 1. The first step, however, which is not a part of this program, is the synthesis of the analog prototype of the filter. The prototype is a filter configuration scaled for a frequency of 1 radian per second and 1 ohm impedance level. As an example of a prototype filter configuration, figure 2 shows the transfer function and the frequency response function for a Butterworth filter, a filter that has a second order maximally flat amplitude response. Any prototype filter form is possible; however, care should be exercised in using filters of fifth order and above because numerical instability may result from the cumulative effect of round off errors in such filters. Where higher order filters are required, they should be produced by a cascade of lower order filters developed from the factored form of the prototype transfer function. High pass, low pass, and band pass configurations as well as combinations are acceptable.

The digital filter synthesis program is written in Fortran IV and operates under the IBM 7040/7094 DCS (Direct Couple System) with the IBSYS executive monitor. Plotting is available for a maximum of 500 points per variable. The number of points plotted by the program may be any fraction less than the number of points computed. This fraction is determined by the plot ratio and the total number of points equals run time divided by the input sampling interval.

BILINEAR TRANSFORM METHOD

The bilinear transform method is an approximate method because digital data only approximately represent analog data and because the bilinear transform warps the frequency scale. Where very many samples of a given analog waveform are taken in an interval large with respect to filter time constants, which is the mode of operation expected, a very good approximation is obtained. However, the program is useful even when this is not the case, but care should be taken to compute or calibrate the performance of the filter when fewer than 10 samples are used in the interval corresponding to the shortest filter time constant. Methods for compensating the frequency warping effect are available but have not been incorporated in the program.

A bilinear transformation is the process of forming all the coefficients of the digital difference equation which controls the actual digital filtering. The form of the difference equation is such that the current data output sample is formed from the present data input, past data input, and past data output samples only. Other types of digital filter operations using future data input samples are not possible in this program. The structures of filters of this type are discussed at length by Arabadjis (ref. 6) and others.

DESIGN PROCEDURE

The design procedure used in this program starts with the choice of a sampling interval t. This is the time in seconds between each entry in the digital input data. Once this time interval is known and the digital filter critical frequencies have been established, it is possible to compute the equivalent analog filter critical frequencies according to the following formula:

$$\omega_{ai} = \tan \frac{\omega_{di}T}{2} \tag{1}$$

where the ω_{ai} are the fundamental quantities used in frequency scaling for the analog prototype, ω_{di} are the corresponding quantities for the digital filter, T is the interval between data points, and i is an index.

Given a prototype filter expressed in the following form:

$$\text{Filter function} = \frac{\Sigma a_n s^n}{\Sigma b_m s^m} \tag{2}$$

where a and b are arbitrary constants, m and n are real positive integers, and s is the complex frequency variable, the program computes a new frequency-scaled function as follows:

$$F(s) = \frac{\Sigma a_n' s^n}{\Sigma b_m' s^m} \tag{3}$$

where the primes represent scaled quantities. The next step in the synthesis process is the transformation from the s to the z plane by means of the bilinear transformation approximation.

Choice of Sampling Interval

The sampling frequency should be large in comparison to the frequencies of interest and especially the critical frequencies for reasons mentioned earlier. In order to simplify the discussion, let us continue using the example of figure 2 and carry out the numerical aspects of the computations as we go along. The data we will be dealing with in this example are in the order of cycles per second, we will choose 1000 samples of data per second. This is more than necessary but should make the approximation to the analog prototype quite good; namely,

$$T = 0.001 \tag{4}$$

Choice of the 3 dB Cutoff Frequency as a Critical Frequency

To pursue the numerical example further, we assign the 3 dB point for the Butterworth filter to be a critical frequency at 2 Hz:

$$\omega_d = 2\pi f_d = 12.56 \text{ rad/sec} \tag{5}$$

where f_d is the desired 3 dB cutoff frequency in Hz for the digital filter. Then ω_a is computed as follows:

$$\omega_a = \frac{\tan \omega_d T}{2} \approx 0.00628 \tag{6}$$

Choice of a Filter Form

The prototype of figure 2 has been chosen for this example and the analog transfer function is repeated here:

$$F(s) = \frac{1}{s^2 + \sqrt{2}\, s + 1} \tag{7}$$

Calculation of the Frequency-Scaled Parameters for the Prototype

For synthesis on the real frequency axis we may substitute $j\omega$ for s in equation (7) and, wherever $j\omega$ occurs, scale this variable by the frequency scaling ratio which, for this case, is equal to ω_a as follows:

$$F'(\omega) = \frac{1}{\left(\frac{j\omega}{\omega_a}\right)^2 + \sqrt{2}\left(\frac{j\omega}{\omega_a}\right) + 1} \tag{8}$$

F(s) can then be calculated by substituting back as follows:

$$F'(s) = \frac{1}{\frac{s^2}{\omega_a{}^2} + \frac{\sqrt{2}s}{\omega_a} + 1} \tag{9}$$

$$F'(s) = \frac{1}{b'_2 s^2 + b'_1 s + b'_o} \tag{10}$$

Coefficients are matched to give the b values where

$$b'_2 = \frac{1}{\omega_a{}^2} = 2.54 \times 10^4 \tag{11}$$

$$b'_1 = \frac{\sqrt{2}}{\omega_{a_1}} = 2.25 \times 10^2 \tag{12}$$

$$b'_o = 1 \tag{13}$$

and can be computed for any order filter by similar procedures. Thus,

$$F'(s) = \frac{1}{(2.54 \times 10^4)s^2 + (2.25 \times 10^2)s + 1} \tag{14}$$

Bilinear Transformation of $F(s)$

Now to calculate $F(z)$ replace s by $z - 1/z + 1$

$$F(z) = \frac{1}{(2.54\times10^4)\ \frac{z-1}{z+1} + (2.25\times10^2)\ \frac{z-1}{z+1} + 1} \tag{15}$$

Multiplying numerator and denominator by $(z + 1)^2$ gives:

$$F(z) = \frac{z^2 + 2z + 1}{2.54\times10^4 z^2 - 5.08\times10^4 z + 2.54\times10^4 + 2.25\times10^2 z^2 - 2.25\times10^2 + 1} \tag{16}$$

Multiplying numerator and denominator again by z^{-2} gives:

$$F(z) = \frac{1 + 2z^{-1} + z^{-2}}{25625 - 50800z^{-1} + 25176z^{-2}} \tag{17}$$

Since $F(z)$ represents the digital equivalent of a transfer function, it can be represented by the following expression:

$$F(z) = \frac{\Sigma A_n z^{-n}}{\Sigma B_n z^{-n}} = \frac{O(z)}{I(z)} \tag{18}$$

where

$O(z)$ output data function of z

$I(z)$ input data function of z

A_n numerator coefficients of the digital transfer function

B_n denominator coefficients of the digital transfer function

The function $O(z)$ represents the desired output digital data list and we can solve for this quantity by simply rearranging equation (18) as follows:

$$O(z) = \frac{I(z)A_o + I(z)A_1 z^{-1} + I(z)A_2 z^{-2}}{B_o + B_1 z^{-1} + B_2 z^{-2}} \tag{19}$$

Dividing through by B_o and simplifying yields

$$O(z) = -\frac{O(z)B_1z^{-1}}{B_O} - \frac{O(z)B_2z^{-2}}{B_O} + \frac{I(z)A_O}{B_O} + \frac{I(z)A_1z^{-1}}{B_O} + \frac{I(z)A_2z^{-2}}{B_O} \tag{20}$$

Since each term on the right-hand side of equation (20) is divided by B_O, a new set of constant coefficients is defined for simplicity as follows:

$$O(z) = O(z)[-B'_1z - 1 - B'_2z - 2] + I(z)[A'_O + A'_1z - 1 + A'z - 2] \tag{21}$$

This is the difference equation that represents the nearest digital equivalent to the analog prototype within the limitations allowed by the input sampling intervals for the example. The following is a list of these coefficients:

$$\left.\begin{aligned} A'_O &= 3.913\times10^{-5} \\ A'_1 &= 7.826\times10^{-5} \\ A'_2 &= 3.913\times10^{-5} \\ B'_1 &= 1.982 \\ B'_2 &= 0.98238 \end{aligned}\right\} \tag{22}$$

The program lists these coefficients for each filter configuration.

GRAPHICAL OUTPUTS

One of the special features of this program is the graphical displays of data lists that are used as inputs and outputs of the data lists that are used as inputs and outputs of the filters. Graphs are obtained in the print out by the method of plotting that was developed at the University of Michigan. Special characters are used to form a reference grid and then variable points from the same listing machine are used to generate tabulated digital data. The graphs formed by this method are only accurate to approximately ±1 percent, but they do allow an almost instantaneous graphical output to be generated, thereby providing improved man-machine communications. Other methods such as off-line plotting have the advantage of greater accuracy but the disadvantage of long turn-around time. In trouble-shooting and debugging operations, this turn-around time is cumulative and under some circumstances can be devastating. Some of the limitations in accuracy of the list plotting method used in this program are compensated for by simultaneously listing the digital data so that if it is desired to measure a point more accurately than is possible on the graph, the tabulated data will give the result to any degree of precision required. It is not necessary to plot each point in input or output data. A plot ratio allows one out of every n points to be plotted where n is a positive integer. For example, if the plot ratio is

10, computations will be performed and the required accuracy will be achieved with all input digital data points though only 1 out of every 10 points will be plotted.

SUMMARY OF SUBROUTINES

In this program, a number of subroutines that accomplish the functions indicated in figure 1 are discussed here. The BILIN routine obtains filtering constants from a transfer function. ISCALE is a routine that sets up the time scale abscissa and number of samples for the graphing routines. The FILTER subroutine takes one input and gives one output for each entry, retaining only those past values which it needs. PLT subroutine will scale the ordinate values for the plot and give the first two calls to the UMPLOT subroutine (PLOT1 and PLOT2). The R array of the program is a collection of all ordinate points for the graph with LR = the number of such points. Any combination or arrangement of graphs of the different operations and filters is possible. The plot is then completed by successive PLOT3 calls (1-5) for each operation and a PLOT4 call. A plot exceeding 500 points is made possible only by changing the image dimension of subroutine PLT. The following is a listing of the subroutines with their arguments and definitions.

COEF (N,C) ST1002	Stores binomial coefficients. The coefficients of the x^2, x^3, . . . , x^N terms are placed in C(3), C(4), . . . , C(N + 1). C(1) and C(2) are fixed by the programmer.
XDEN (N,S,B) ST1003	Assumes coefficient of s^N to be 1. The coefficients of s^{N-1}, s^{N-2}, . . . , s constant are assumed to be in S(1), S(2), . . . , S(N - 1), S(N), respectively. XDEN makes the substitution $(z - 1)^N$, $(z - 1)^{N+1}(z + 1)$, $(z - 1)^{N-2}(z+1)^2$, . . . , $(z - 1)(z + 1)^{N-1}$ for s^{N-1}, s^{N-2}, . . . , s and multiplies the constant by $(z + 1)^N$. The coefficients of z^j, j = 0, 1, 2, . . . , N are given in B(1), B(2), . . ., B(N), B(N + 1). This subroutine is part of the bilinear transformation.
FILTER (VAL,YY, A,B,X,Y,NN,M,NNR) ST1004	Performs filter function
VAL	Xi - input data list I(z)
YY	Yi - output data list O(z)
A's and B's	Filtering coefficients supplied by BILIN
X's and Y's	Arrays containing past input and output quantities needed for the calculation of YY.
M	Order of the filter
NN	Equal to M + 1
NNR	Number of the sample

PSMPY (P,NP,Q, NQ,RR,NRR) ST1005	Multiplies polynomials P (an array with dimension NP) and Q (an array with dimension NQ) giving the resultant polynomial in array RR with dimension NRR
PROD (EK,A,B,MU, M,P,NP) ST1006	Produces coefficients of polynomials from its roots.
EK	Scaling factor (the coefficient of the highest term in degree; 1. for this program)
A	Real part of roots (an array)
B	Imaginary part of roots (an array)
MU	Array specifying number of times each root is repeated (all set to 1.)
M	Number of different roots
P	Array of coefficients starting with constant term rearranged in subroutine RCOEF.
NP	Degree of resulting polynomial
	A pair of conjugate roots is counted as one root. The conjugate is assumed where the root location has an imaginary component.
RCOEF (N,SS,S) ST1007	Generates coefficients of polynomial S from complex roots in SS
N	The number of roots
S	Real coefficients of resulting polynomial
XINP (VAL,TI,K) ST1008	Generates filter inputs and is a subroutine supplied by the programmer. A single input is calculated on each entry and placed in VAL.
TI	The sampling time interval
K	The nth input point
NOTE:	The programmer should compile this subroutine for each run to insure proper inputs to the filter.
PLT (KL,RT,R,LR, KLL,IFL) ST1009	Provides the first two calls (PLOT1 and PLOT2) to the plot routine UMPLOT. All arguments to this subroutine must be calculated and supplied to it.
KL	Number of samples to be plotted
RT	An array of times (abscissa) for the plot
R	An array containing all the ordinate values for scaling purposes.
LR	Number of elements in array R
KLL	Number of abscissa grid lines
IFL	Flag; positive formal entry; negative to call PLOT2 only. This subroutine scales the ordinate values with the help of subroutines RANGE and SCALE. SCALE is a library subroutine.

NOTE:	The abscissa and ordinate are interchanged so that the abscissa may be several pages long. This necessitates the change in sign for time (RT) values and this is accomplished by the subroutine automatically.
ISCALE (TIME,TI, IW,NP,KLL) ST1011	Calculates arguments NP and KLL from the other three arguments.
NP	Total number of points in the sample
KLL	Number of abscissa grid lines
TIME	Length of run in seconds
TI	Sampling period in seconds
IW	Plot ratio, total number points/number points plotted
RANGE (RMIN, RMINI, LS) ST1010	Calculates a rounded minimum for scaling.
RMIN	minimum value to be plotted, floating point
RMINI	Minimum value to be plotted, integer
LS	Number of shifts Enter with least value in RMIN Exit with new rounded value in RMIN Examples:

Enter RMIN	Exit RMIN
0.014+	0.0100--
0.0025+	0.0020--
1.4	1.0
-0.013	-0.020

SCALE (RNG, RMIN,10.,SF)	Calculates a maximum value
RNG	Maximum value to be plotted
RMIN	Minimum
10.	Available inches minus 1
SF	scale factor
RNG	11.xSF + RMIN, calculates new value of maximum based on 11 inches.
SCALE	is a NASA Ames utility subroutine written by J. A. Jeske available at this time on the disk. No deck card is needed.
BILIN (LD,LN,FR, TI,A,B,CØN,IFL) ST1012	Calculates coefficients for the difference equation
LD	Degree of the polynomial or number of poles in the denominator of the transfer function f(s)
LN	Degree of the polynomial or number of roots in the numerator of the transfer function f(s)
FR	The frequency scaling ratio
TI	Sampling period in seconds
A and B	Arrays containing coefficients for the difference equation

CØN	Leading constant of the transfer function
IFL	Input flag
Positive	f(s) function in terms of coefficients of polynomials
Negative	f(s) function in terms of poles and zeros of f(s)
	All arguments are inputs to the subroutine except arrays A and B. See flow chart and inputs to BILIN

The following plot subroutines are NASA Ames modifications from SHARE library routine UMPLOT.

PLØT1 (NSCALE, NHL, NSBH, NBL, NSBV)

NSCALE	=	An array of dimension 5 that supplies the subroutine with grid and scale factor information
NSCALE(1)	=	0, standard grid and scale factors (see note (a))
	$\neq$	0, grid and scale factors are as defined in NSCALE(2)-NSCALE(5)
NSCALE(2)	=	I, scale factor such that printed values of the ordinate are 10^I times the actual values
NSCALE(3)	=	J, J digits will appear to the right of the decimal point in printed ordinate values ($J < 8$)
NSCALE(4)	=	K, scale factor such that printed values of the abscissa are 10^K times the actual values
NSCALE(5)	=	M, M digits will appear to the right of the decimal point in printed abscissa values ($M < 8$)
NHL	=	The number of horizontal grid lines (NHL > 0)
NSBH	=	The number of spaces between horizontal grid lines (NSBH > 0)
NBL	=	The number of vertical grid lines (NBL > 0)
NSBV	=	The number of spaces between vertical grid lines (NSBV > 0, and NSBV*NBL $\leq$ 119)

NOTE (a): Standard scale factors correspond to values of I, J, K, and M of 0, 3, 0, 3, respectively.

PLØT2 (IMAGE, XMAX, XMIN, YMAX, YMIN, IDIM)

IMAGE	An array, dimensioned IDIM, which is used as the image region for the plot
XMAX	The value of the abscissa at the right most grid line
XMIN	The value of the abscissa at the left most grid line
YMAX	The value of the ordinate at the upper most grid line
YMIN	The value of the ordinate at the lower most grid line
IDIM	The dimension of the array IMAGE, where IDIM is at least equal to N*(NSBH*NHL + 1) where $N = [(K/6) + (1/2) + (1/2)(-1)^{(K+1-2[K/2])}]$ and where K = NSBV*NBL + 1 (The square brackets in the formula for N signify "integral value.")

PLØT3 (1HX,RX,RT,L)

1HX	X is the character used for plotting the points whose abscissas are stored in array RX and whose ordinates are stored in array RT
RX	The array containing the abscissas of the points to be plotted.
RT	The array containing the ordinates of the points to be plotted
L	The number of points to be plotted

PLØT4 (20,20Hbbbb TbbbIbbbMbbbEbbb)	This subroutine initiates the plot and provides the ordinate label

FLOW CHARTS

Flow charts for the main program and for the subroutine BILIN are given in figures 3 and 4.

EXAMPLE

A load deck for the example filter given in figure 2 with a sinusoidal input is shown in appendix A. Note that the input function which may be any analytic or data input function is entered in subroutine XINP to be compiled each time the problem is run. A list of input values X, output values Y, and the graphical outputs of the program is also given for this example for a time varying between 0 and 10 seconds. The A's and B's listed are the coefficients of the difference equation for the digital filter.

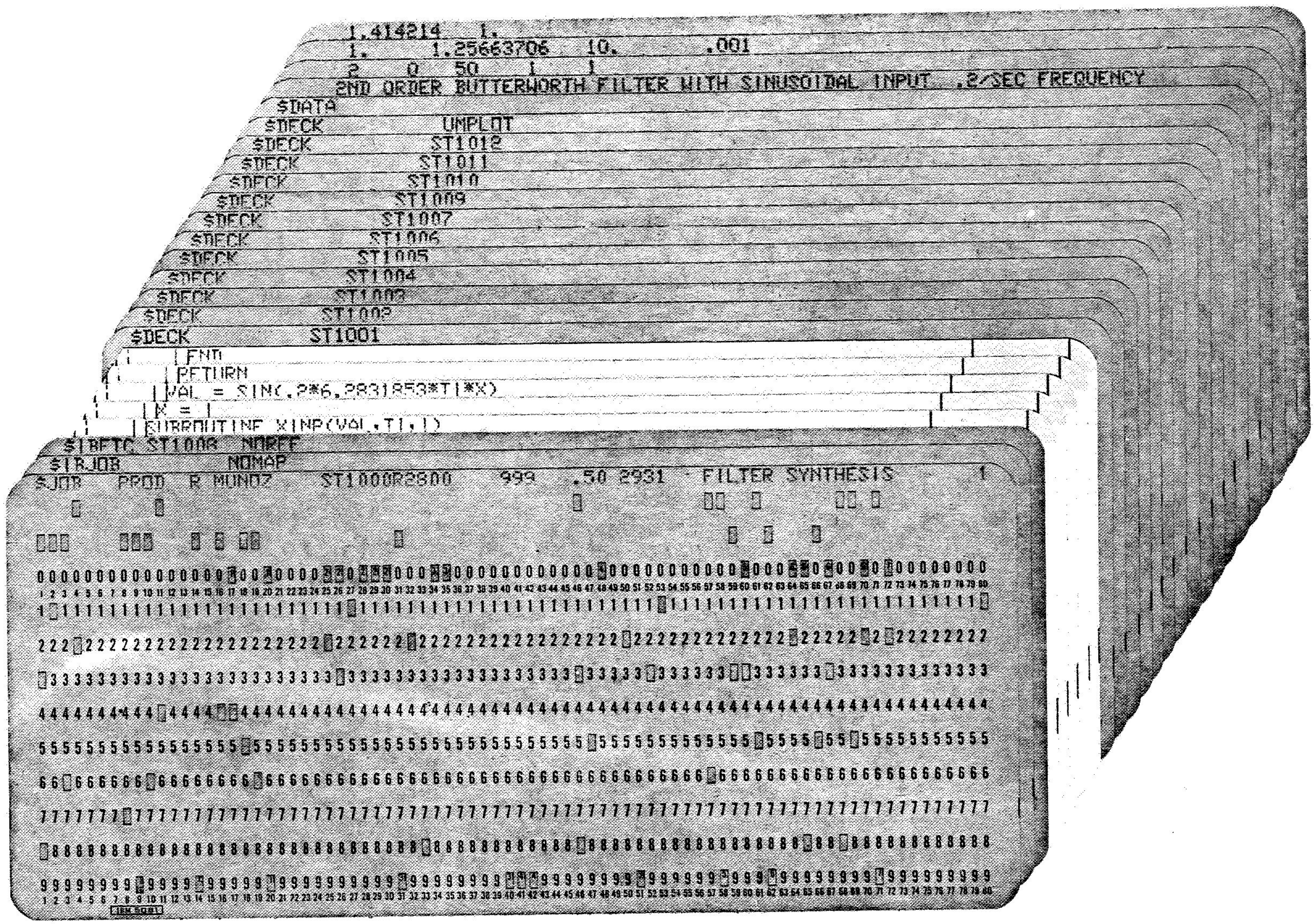
INPUT FORMAT

Tables I and II describe the input format for the Digital Filter Synthesis Program. This format contains the information necessary to produce the coefficients of the difference equation as well as ancillary inputs to determine the character of the input data lists, output data lists, run time, and plot characteristics.

Ames Research Center
National Aeronautics and Space Administration
Moffett Field, Calif., 94035, Sept. 25, 1967

APPENDIX A

EXAMPLE PROBLEM LOAD DECK



APPENDIX B

EXAMPLE PROBLEM PRINT OUT

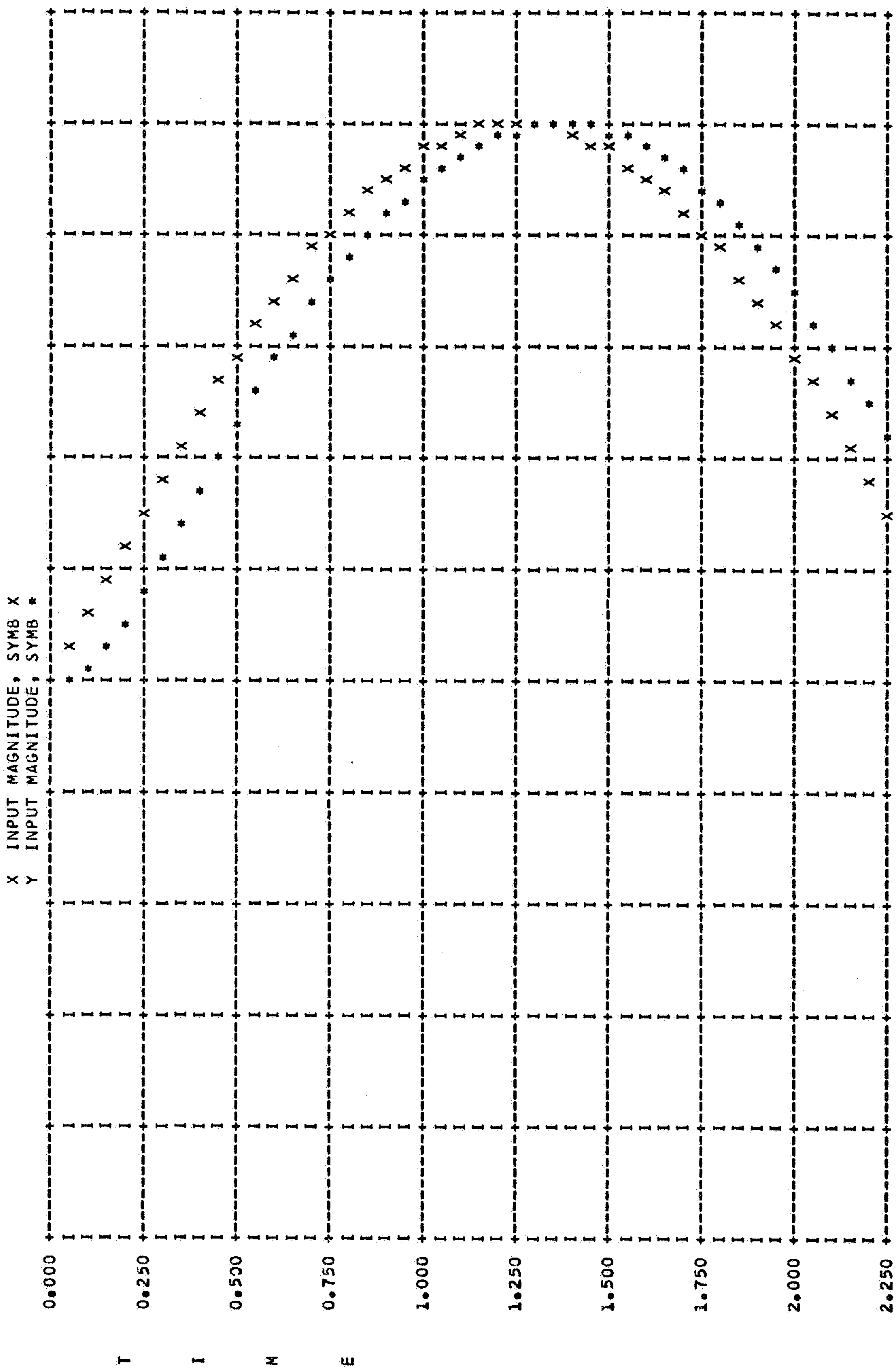
```
   2ND ORDER BUTTERWORTH FILTER WITH SINUSOIDAL INPUT  .2/SEC FREQUENCY

  2    0   50    1    1

   1.00       1.26      10.00   0.00100

 1.4142140E 00   1.0000000E 00

 A,S
   0.3913020E-04      0.7826041E-04      0.3913020E-04
 B,S
  -0.1982229E 01      0.9823854E 00
ORDER OF FILTER=    2   NO. OF PLOTTED SAMPLES= 200   TOTAL NO. OF SAMPLES=   10000

   X(INPUT)      Y(OUTPUT)

   0.62791E-01      0.32963E-02
   0.12533E-00      0.20893E-01
   0.18738E-00      0.55739E-01
   0.24869E-00      0.10455E-00
   0.30902E-00      0.16225E-00
   0.36812E-00      0.22424E-00
   0.42578E-00      0.28724E-00
   0.48175E-00      0.34926E-00
   0.53583E 00      0.40931E-00
   0.58779E 00      0.46700E-00
   0.63742E 00      0.52222E 00
   0.68455E 00      0.57496E 00
   0.72897E 00      0.62520E 00
   0.77051E 00      0.67289E 00
   0.80902E 00      0.71791E 00
   0.84433E 00      0.76013E 00
   0.87631E 00      0.79938E 00
   0.90483E 00      0.83550E 00
   0.92978E 00      0.86835E 00
   0.95106E 00      0.89778E 00
   0.96858E 00      0.92366E 00
   0.98229E 00      0.94591E 00
   0.99211E 00      0.96442E 00
   0.99803E 00      0.97912E 00
   0.10000E 01      0.98996E 00
   0.99803E 00      0.99689E 00
   0.99211E 00      0.99988E 00
   0.98229E 00      0.99893E 00
   0.96858E 00      0.99403E 00
   0.95106E 00      0.98521E 00
   0.92978E 00      0.97250E 00
```

0.90483E 00	0.95595E 00
0.87631E 00	0.93564E 00
0.84433E 00	0.91163E 00
0.80902E 00	0.88402E 00
0.77051E 00	0.85292E 00
0.72897E 00	0.81845E 00
0.68455E 00	0.78076E 00
0.63742E 00	0.73998E 00
0.58779E 00	0.69629E 00
0.53583E 00	0.64984E 00
0.48175E-00	0.60083E 00
0.42578E-00	0.54945E 00
0.36812E-00	0.49589E-00
0.30902E-00	0.44039E-00
0.24869E-00	0.38315E-00
0.18738E-00	0.32441E-00
0.12533E-00	0.26437E-00
0.62791E-01	0.20329E-00
0.89407E-07	0.14141E-00
-0.62790E-01	0.78977E-01
-0.12533E-00	0.16228E-01
-0.18738E-00	-0.46585E-01
-0.24869E-00	-0.10921E-00
-0.30902E-00	-0.17141E-00
-0.36812E-00	-0.23293E-00
-0.42578E-00	-0.29354E-00
-0.48175E-00	-0.35298E-00
-0.53583E 00	-0.41103E-00
-0.58779E 00	-0.46746E-00
-0.63742E 00	-0.52205E 00
-0.68455E 00	-0.57457E 00
-0.72897E 00	-0.62482E 00
-0.77051E 00	-0.67261E 00
-0.80902E 00	-0.71775E 00
-0.84433E 00	-0.76005E 00
-0.87631E 00	-0.79936E 00
-0.90483E 00	-0.83551E 00
-0.92978E 00	-0.86837E 00
-0.95106E 00	-0.89780E 00
-0.96858E 00	-0.92368E 00
-0.98229E 00	-0.94592E 00
-0.99211E 00	-0.96443E 00
-0.99803E 00	-0.97913E 00
-0.10000E 01	-0.98996E 00
-0.99803E 00	-0.99689E 00
-0.99211E 00	-0.99988E 00
-0.98229E 00	-0.99893E 00
-0.96858E 00	-0.99403E 00
-0.95106E 00	-0.98521E 00
-0.92978E 00	-0.97251E 00
-0.90483E 00	-0.95596E 00

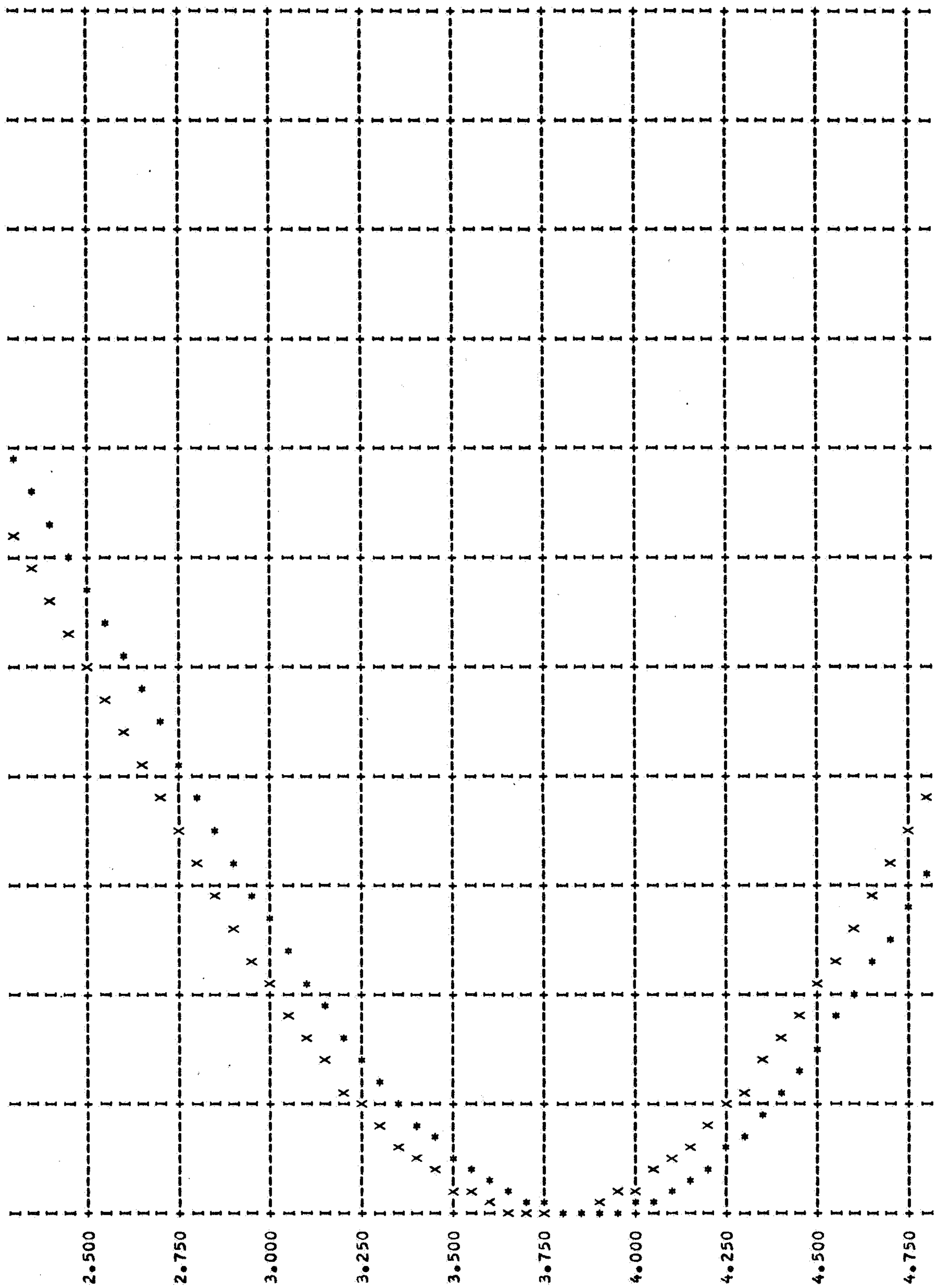
-0.87631E 00	-0.93564E 00
-0.84433E 00	-0.91163E 00
-0.80902E 00	-0.88402E 00
-0.77051E 00	-0.85292E 00
-0.72897E 00	-0.81846E 00
-0.68455E 00	-0.78076E 00
-0.63742E 00	-0.73999E 00
-0.58779E 00	-0.69629E 00
-0.53583E 00	-0.64984E 00
-0.48175E-00	-0.60083E 00
-0.42578E-00	-0.54945E 00
-0.36812E-00	-0.49590E-00
-0.30902E-00	-0.44039E-00
-0.24869E-00	-0.38316E-00
-0.18738E-00	-0.32441E-00
-0.12533E-00	-0.26437E-00
-0.62791E-01	-0.20329E-00
-0.18254E-06	-0.14141E-00
0.62790E-01	-0.78977E-01
0.12533E-00	-0.16228E-01
0.18738E-00	0.46585E-01
0.24869E-00	0.10921E-00
0.30902E-00	0.17141E-00
0.36812E-00	0.23293E-00
0.42578E-00	0.29354E-00
0.48175E-00	0.35298E-00
0.53583E 00	0.41103E-00
0.58779E 00	0.46746E-00
0.63742E 00	0.52205E 00
0.68455E 00	0.57457E 00
0.72897E 00	0.62482E 00
0.77051E 00	0.67261E 00
0.80902E 00	0.71775E 00
0.84433E 00	0.76005E 00
0.87631E 00	0.79936E 00
0.90483E 00	0.83551E 00
0.92978E 00	0.86836E 00
0.95106E 00	0.89779E 00
0.96858E 00	0.92367E 00
0.98229E 00	0.94592E 00
0.99211E 00	0.96442E 00
0.99803E 00	0.97912E 00
0.10000E 01	0.98996E 00
0.99803E 00	0.99689E 00
0.99211E 00	0.99988E 00
0.98229E 00	0.99892E 00
0.96858E 00	0.99403E 00
0.95106E 00	0.98521E 00
0.92978E 00	0.97250E 00
0.90483E 00	0.95596E 00
0.87631E 00	0.93564E 00

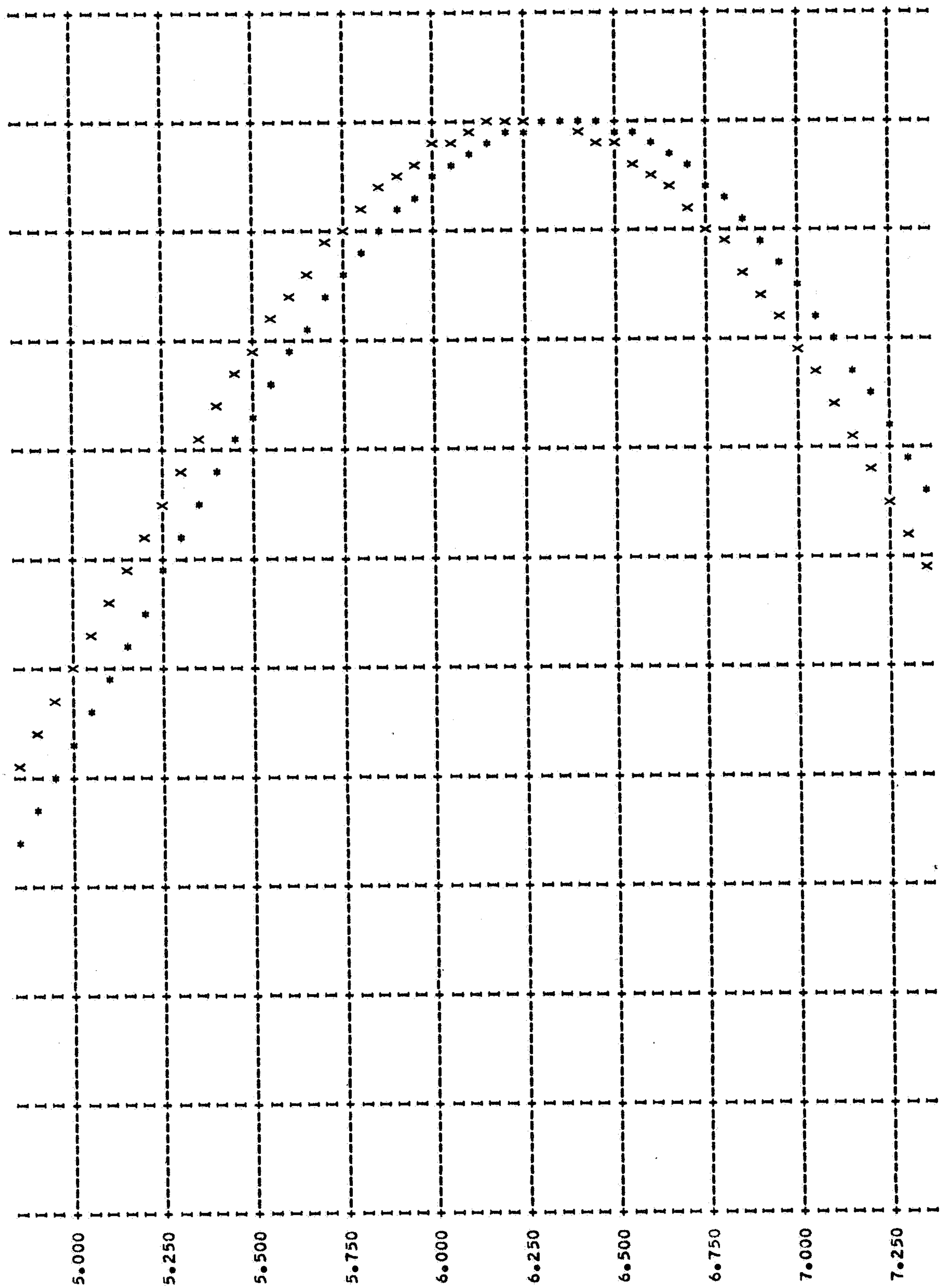
0.84433E 00	0.91163E 00
0.80902E 00	0.88402E 00
0.77051E 00	0.85292E 00
0.72897E 00	0.81845E 00
0.68455E 00	0.78076E 00
0.63742E 00	0.73998E 00
0.58779E 00	0.69628E 00
0.53583E 00	0.64984E 00
0.48175E-00	0.60083E 00
0.42578E-00	0.54944E 00
0.36812E-00	0.49589E-00
0.30902E-00	0.44039E-00
0.24869E-00	0.38316E-00
0.18738E-00	0.32441E-00
0.12533E-00	0.26437E-00
0.62791E-01	0.20329E-00
0.33155E-06	0.14141E-00
-0.62790E-01	0.78977E-01
-0.12533E-00	0.16229E-01
-0.18738E-00	-0.46584E-01
-0.24869E-00	-0.10921E-00
-0.30902E-00	-0.17141E-00
-0.36812E-00	-0.23293E-00
-0.42578E-00	-0.29354E-00
-0.48175E-00	-0.35298E-00
-0.53583E 00	-0.41103E-00
-0.58778E 00	-0.46746E-00
-0.63742E 00	-0.52205E 00
-0.68455E 00	-0.57457E 00
-0.72897E 00	-0.62482E 00
-0.77051E 00	-0.67261E 00
-0.80902E 00	-0.71774E 00
-0.84433E 00	-0.76005E 00
-0.87631E 00	-0.79935E 00
-0.90483E 00	-0.83551E 00
-0.92978E 00	-0.86836E 00
-0.95106E 00	-0.89779E 00
-0.96858E 00	-0.92368E 00
-0.98229E 00	-0.94592E 00
-0.99211E 00	-0.96443E 00
-0.99803E 00	-0.97913E 00
-0.10000E 01	-0.98996E 00
-0.99803E 00	-0.99689E 00
-0.99211E 00	-0.99988E 00
-0.98229E 00	-0.99893E 00
-0.96858E 00	-0.99403E 00
-0.95106E 00	-0.98521E 00
-0.92978E 00	-0.97251E 00
-0.90483E 00	-0.95596E 00
-0.87631E 00	-0.93564E 00
-0.84433E 00	-0.91163E 00

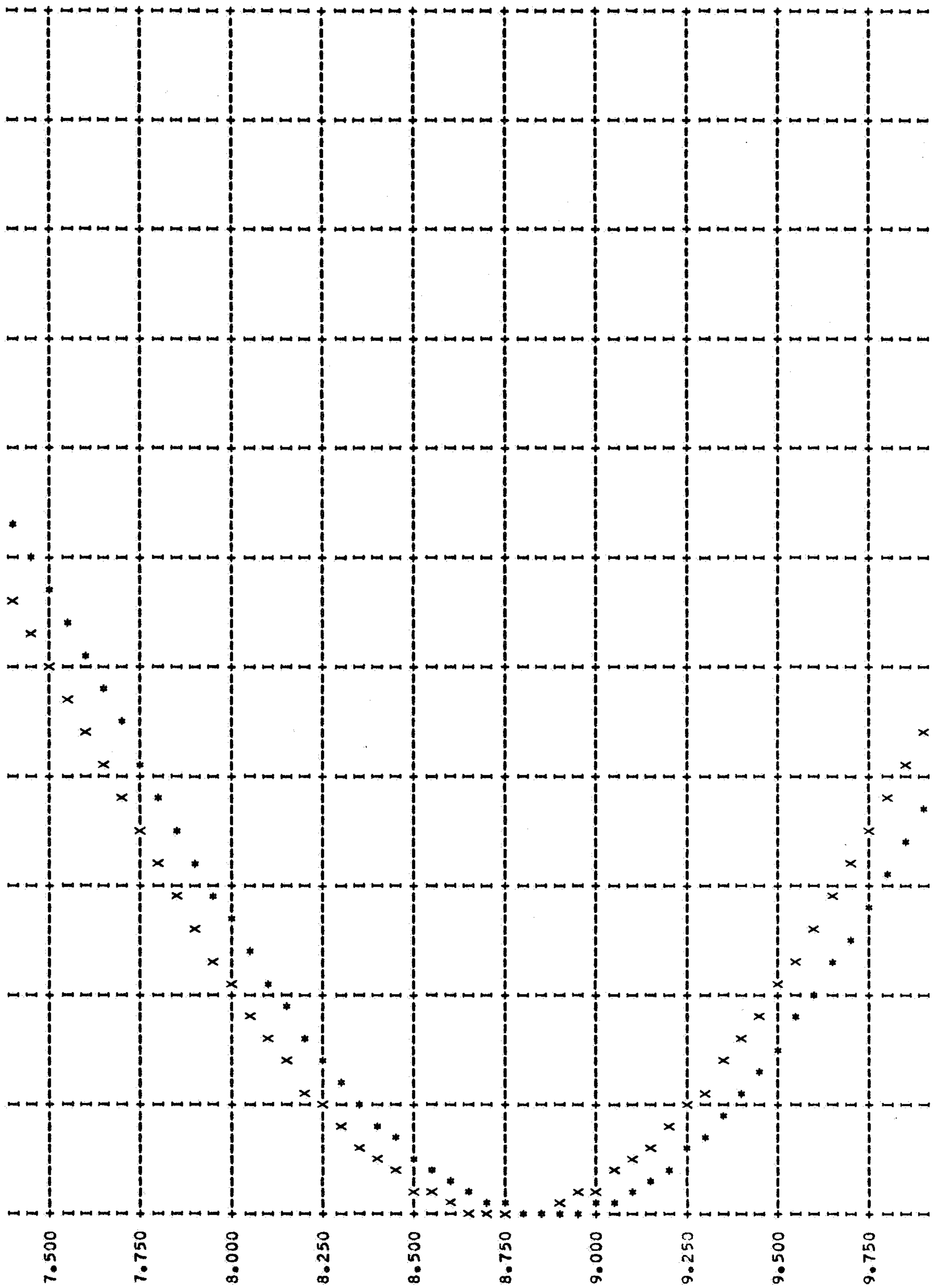
-0.80902E 00	-0.88402E 00
-0.77051E 00	-0.85292E 00
-0.72897E 00	-0.81845E 00
-0.68455E 00	-0.78076E 00
-0.63742E 00	-0.73998E 00
-0.58779E 00	-0.69628E 00
-0.53583E 00	-0.64984E 00
-0.48175E-00	-0.60083E 00
-0.42578E-00	-0.54945E 00
-0.36812E-00	-0.49590E-00
-0.30902E-00	-0.44039E-00
-0.24869E-00	-0.38316E-00
-0.18738E-00	-0.32441E-00
-0.12533E-00	-0.26437E-00
-0.62791E-01	-0.20329E-00
-0.36508E-06	-0.14141E-00



X INPUT MAGNITUDE, SYMB X
Y INPUT MAGNITUDE, SYMB *
0.000
0.250
0.500
0.750
1.000
1.250
1.500
1.750
2.000
2.250
T I M E


2.500
2.750
3.000
3.250
3.500
3.750
4.000
4.250
4.500
4.750


5.000
5.250
5.500
5.750
6.000
6.250
6.500
6.750
7.000
7.250


7.500
7.750
8.000
8.250
8.500
8.750
9.000
9.250
9.500
9.750

```
         I         I         I         I         *    X  I         I         I         I         I         I         I
10.000   +---------+---------+---------+---------+--*----X---------+---------+---------+---------+---------+---------+
       -1.000    -0.800    -0.600    -0.400    -0.200    -0.000     0.200     0.400     0.600     0.800     1.000     1.200
```

JOB 001

JOB TYPE	NAME	CAB NO.	JOB ORDER	TIME ON	COMP/LOAD TIME MIN.	EXECUTE TIME MIN.	DATE
PROD	MOYER	ST1000	R2800	13.56	.64	.26	07/13/67

APPENDIX C

MAIN PROGRAM AND SUBROUTINE LISTING

```
$IBFTC ST1001  NOREF
C
C       M A I N   P R O G R A M
C
      DIMENSION FIRST(1)
      DIMENSION XID(14),A(50),B(50),RX(500),RY(500),RT(500),R(2500)
      DIMENSION X(20),Y(20)
      READ (5,100) XID
      WRITE (6,107) XID
      READ(5,101) LD,LN,IW,JJ,IFL
      READ(5,106) CON,FS,TIME,TI
      WRITE(6,102) LD,LN,IW,JJ,IFL
      WRITE(6,103) CON,FS,TIME,TI
      IF(JJ.LT.1) JJ=1
      IF(JJ.GT.4) JJ=4
      CALL ISCALE(TIME,TI,IW,NP,KLL)
      CALL BILIN(LD,LN,FS,TI,A,B,CON,IFL)
      M=MAXO(LD,LN)
      NZ = M+1
      WRITE (6,104) (A(I),I=1,NZ)
      WRITE (6,105) (B(I),I=1,M)
  100 FORMAT(13A6,A2)
  101 FORMAT(5I5)
  102 FORMAT(/5I5)
  103 FORMAT(/3F10.2,F10.5)
  104 FORMAT(1H0,5H  A,S/(6E18.7/))
  105 FORMAT(6H   B,S/(6E18.7/))
  106 FORMAT(4E10.0)
  107 FORMAT(2H1 13A6,A2)
      NUM=NP/IW
      WRITE(6,111)M,NUM,NP
  111 FORMAT(1H0,16HORDER OF FILTER=,I4,3X23HNO. OF PLOTTED SAMPLES=,I4,
     13X21HTOTAL NO. OF SAMPLES=,I8//)
      L=0
      NNR = 0
   41 CONTINUE
      NNR=NNR+1
      CALL XINP(VAL,TI,NNR)
      CALL FILTER(VAL,YY,A,B,X,Y,NZ,M,NNR)
      MNR=MOD(NNR,IW)
      IF(MNR.EQ.0) GO TO 43
   42 CONTINUE
      IF(NNR.LE.NP)  GO TO 41
      GO TO 45
   43 IF(JJ.LT.3) GO TO 44
```

```
      IF(JJ.EQ.3) GO TO 46
      GO TO 42
   44 L = L + 1
      RX(L) = VAL
      RY(L) = YY
      XNR = NNR
      RT(L) = XNR*TI
      IF(JJ.NE.1) GO TO 42
   46 IF(JKI-2521)60,62,60
   60 JKI = 2521
      WRITE(6,112)
  112 FORMAT(1H ,5X,8HX(INPUT),5X,9HY(OUTPUT)//)
   62 WRITE(6,113)VAL,YY
      GO TO 42
   45 IF(JJ.GT.2) GO TO 47
      LR=2*L
      WRITE(6,156)
  156 FORMAT(1H1,43X26HX  INPUT MAGNITUDE, SYMB X/44X26HY  INPUT MAGNITU
     1DE, SYMB *)
      DO 55 I=1,L
      III=2*I
      II =III-1
      R(II) = RX(I)
   55 R(III)= RY(I)
      CALL PLT(L,RT,R,LR,KLL,1)
      CALL  PLOT3(1HX,RX,RT,L)
      CALL  PLOT3(1H*,RY,RT,L)
      CALL PLOT4(20,20H    T    I    M    E    )
  113 FORMAT (2E17.5)
   47 LAST = 1001
      STOP
      END
$IBFTC ST1002  NOREF
      SUBROUTINE COEF (N,C)
C
C     SUBROUTINE TO GIVE COEF. OF BINOMIAL EXPANSION
C
      DIMENSION LPT(50),LPB(50),C(50)
      IF(N.EQ.1) RETURN
      DO 1 M=2,N
      K=M-1
      LPT(M)=N
      LPB(M)=M
      KK=N-1
      LK=M-1
```

```
      DO 2 I=1,K
      LPT(M)=LPT(M)*KK
      LPB(M)=LPB(M)*LK
      KK=KK-1
      LK=LK-1
    2 CONTINUE
      C(M+1)=LPT(M)/LPB(M)
    1 CONTINUE
      LAST=0
      RETURN
      END
$IBFTC ST1003  NOREF
      SUBROUTINE XDEN (N,S,B)
C
C     SUBROUTINE TO EVALUATE DENOMINATOR MAKES SUBSTITUTION S=Z/1/Z+1
C     THEN MULTIPLIES BY (Z+1)**N
C
      DIMENSION C(50),B(50),P(50),Q(50),RR(50),S(50)
C
C     DO FIRST AND LAST PART (Z-1)**N+S(N)*(Z+1)**N
C
      NN=N+1
      DO 1 I=1,NN
      B(I)=0.0
    1 CONTINUE
      C(1)=1
      C(2)=N
      CALL COEF(N,C)
      DO 2 I=1,NN
      B(I)=B(I)+C(I)*S(N)
    2 CONTINUE
      DO 3 I=2,NN,2
      C(I)=-C(I)
    3 CONTINUE
      DO 4 I=1,NN
      B(I)=B(I)+C(I)
    4 CONTINUE
      KN=N
C
C     DO MIDDLE PART S(1)*(Z-1)**N-1 *(Z+1)**1 ....S(N-1)*(Z-1)*(Z-1)**N-1
C
    6 P(1)=1
      Q(1)=1
      KN=KN-1
      K=N-KN
```

```
      P(2)=KN
      Q(2)=K
      IF(K.EQ.N) GO TO 7
      CALL COEF (KN,P)
      CALL COEF (K,Q)
      DO 9 I=1,KN,2
      P(I+1)=-P(I+1)
    9 CONTINUE
      CALL PSMPY (P,KN,Q,K,RR,NRR)
      DO 5 I=1,NN
      B(I)=B(I)+RR(I)*S(K)
    5 CONTINUE
      GO TO 6
C
C     NORMALIZE RESULTANT COEFFICIENTS
C
    7 CONTINUE
      LAST=0
   10 RETURN
      END
$IBFTC ST1004  NOREF
      SUBROUTINE FILTER(VAL,YY,A,B,X,Y,NN,M,NNR)
      DIMENSION FIRST(1)
      DIMENSION X(20),Y(20),A(50),B(50)
      IF(NNR.GT.NN) GO TO 1
      NR = NNR
      NR1 = NR - 1
    1 X(NR) = VAL
      Y(NR)=0.0
      DO 32 J=1,NN
      NK=NR-J+1
      IF(NK.LE.0) GO TO 34
      Y(NR)=Y(NR)+A(J)*X(NK)
   32 CONTINUE
   34 DO 33 J=1,M
      NK=NR-J
      IF(NK.LE.0) GO TO 35
      Y(NR)=Y(NR)-B(J)*Y(NK)
   33 CONTINUE
   35 CONTINUE
      YY = Y(NR)
      IF(NR.LT.NN) RETURN
      DO 40 I=1,NR1
      X(I)=X(I+1)
      Y(I)=Y(I+1)
```

```
   40 CONTINUE
      LAST = 1013
      RETURN
      END
$IBFTC ST1005  NOREF
C...   MULTIPLY POLYNOMIALS P(S) AND Q(S)
      SUBROUTINE  PSMPY(P, NP, Q, NQ, RR, NRR)
      DIMENSION  P(1), Q(1), RR(1), R(52)
      NNR = NP + NQ
      NR = NNR + 1
    2 NP1 = NP + 1
      K = 0
      DO 9  J= 1, NR
    3 J = J
      K = K +1
    4 L1 = MAXO(1, J-NQ)
    5 L2 = MINO(J, NP1)
      SUM = 0.0
      DO 8  I= L1, L2
      J1 = J-I
    8 SUM = SUM + P(I)  *Q(J1+1)
    9 R(K) = SUM
      NRR = NNR
      DO 10 L = 1,  NR
   10 RR(L)   = R(L)
      LAST=0
  100 RETURN
      END
$IBFTC ST1006  NOREF
      SUBROUTINE PROD(EK, A, B, MU, M, P, NP)
C
C  EK IS A SCALING FACTOR
C  A =  ARRAY OF REAL PARTS OF ROOTS
C  B = ARRAY OF IMAGINARY PART OF ROOTS
C  MU = ARRAY SPECIFYING NUMBER OF TIMES EACH ROOT IN A IS REPEATED
C  M = NUMBER OF DIFFERENT ROOTS
C  P = ARRAY OF COEFFICIENTS OF POLYNOMIAL,STARTING WITH THE CONSTANT TERM

C  NP = DEGREE OF RESULTING POLYNOMIAL
C  NOTE. IF ANY VALUE IN ARRAY B NOT ZERO, A COMPLEX CONJUGATE IS ASSUMED
C  AND IS NOT ENTERED AS A SEPERATE ROOT, NOR IS THAT ROOT REPEATED
C
C PROD    LIST,REF,DECK,M94,XR7
CPROD   (F4)      T417     W.M.SYN     (2/18/64)
C...   CALCULATE THE COEFFICIENTS OF A POLYNOMIAL FROM ITS ROOTS
```

```
      DIMENSION A(1), B(1), MU(1), P(1), R(3)
      KZ = 0
      P(1) = EK
      NO = 0
      IF(M) 10, 10, 7
    7 DO 3 K = 1,M
      IF(B(K)) 4, 6, 4
C     P(S) = P(S)*(S-A(K))**MU(K)
    6 R(1) = -A(K)
      R(2) = 1.0
      NR = 1
      GO TO 5
C     P(S) = P(S)*((S-A(K))**2 + B(K)**2)**MU(K)
    4 R(1) = A(K)**2 + B(K)**2
      R(2) = -2.0*A(K)
      R(3) = 1.0
      NR = 2
    5 ICTL = MU(K)
C     FORM P(S) = P(S)*R(S)  A TOTAL OF MU(K) TIMES
      DO 3 J = 1,ICTL
      NBAR = NO + NR
      L = NBAR
C     LOOP TO FORM  P(S) = P(S)*R(S)
      DO 2 L1 =KZ,NBAR
      JMAX = MINO(L,NR)
      JMIN = MAXO(0,L-NO)
      J1 = L - JMIN + 1
      SUM = 0.0
      DO 1 N = JMIN,JMAX
      SUM = SUM + R(N+1)*P(J1)
    1 J1 = J1 - 1
      P(L+1) = SUM
    2 L = L - 1
    3 NO = NBAR
   10 NP = NO
      RETURN
      END
$IBFTC ST1007
      SUBROUTINE RCOEF(N,SS,S)
      DIMENSION FIRST(1)
      DIMENSION S(50),D(50),DI(50),MU(50),B(50)
```

```
      COMPLEX SS(50)
      DATA MU/50*1/
      DO 10 I=1,N
      D(I) = REAL(SS(I))
   10 DI(I) = AIMAG(SS(I))
      CALL PROD(1.0,D,DI,MU,N,B,NP)
      L=0
      DO 24 I=1,N
      IF(DI(I))22,24,22
   22 L=L+1
   24 CONTINUE
      N=N+L
      DO 20 I=1,N
      J = N - I + 1
   20 S(J) = B(I)
      LAST=2
      RETURN
      END
$IBFTC ST1008  NOREF
      SUBROUTINE XINP(VAL,TI,I)
      X = RANDOM(+1)
      VAL = 2.*X-1.
      RETURN
      END
$IBFTC ST1009  NOREF
      SUBROUTINE PLT(KL,RT,R,LR,KLL,IFL)
       DIMENSION NSCALE(5),IMAGE(9600)
      DIMENSION FIRST(1)
      DIMENSION R(2500),RT(500)
   25 FORMAT(30HOTHE PLOT LIMIT IS 500 POINTS.)
      IF(KL.GT.500) GO TO 50
      IF(IFL.LE.0) GO TO 45
      IF(RT(KL).LE.0.0) GO TO 3
      OMGMIN = 0.0
      OMGMAX = -RT(KL)
      DO 120 I=1,KL
  120 RT(I) = -RT(I)
    3 CONTINUE
      DATA NSCALE/1,0,3,0,3/
C  FIND PROPER ABSCISSA RANGE
      RMIN = 100000.
      RNG  = 0.0
      DO 130 I=1,LR
      IF(R(I).GT.RNG) RNG=R(I)
      IF(R(I).LT.RMIN) RMIN=R(I)
```

```
  130 CONTINUE
      CALL RANGE(RMIN,RMINI,LS)
      CALL SCALE(RNG,RMIN,10.,SF)
      RNG=11.*SF+RMIN
      LAST=2
      S=PLOT1(NSCALE,KLL,5,11,10)
   45 S=PLOT2(IMAGE,RNG,RMIN,OMGMIN,OMGMAX,9600)
      GO TO 51
   50 WRITE(6,25)
   51 RETURN
      END
$IBFTC ST1010  NOREF
      SUBROUTINE RANGE(A,C,K)
      FIRST=FIRST
      IF(ABS(A).GT.1.) GO TO 70
      B=ABS(A)
      DO 60 I=1,15
      J=B*10.
      IF(J.NE.0) GO TO 61
60    B=B*10.
61    C=J
      IF(AMOD(B,C).NE.0.) C=C+1.
      C=SIGN(C,A)
      IF (C.LE.0.0) GO TO 62
      A = 10.**(-I)*(C-1.)
  63  K=I
      RETURN
   62 A = C*(10.**(-I))
      GO TO 63
70    B=AINT(A)
      B=ABS(B)
      IF(AMOD(A,B).NE.0.) B=B+1.
      B=SIGN(B,A)
      A=B
      C=B
      K=0
      LAST=LAST
      RETURN
      END
$IBFTC ST1011
      SUBROUTINE ISCALE(TIME,TI,IW,NP,KLL)
C  NP IS THE NUMBER OF POINTS IN THE EXPERIMENT
      NP=TIME/TI
      ID=IW*5
C       KLL IS NUMBER OF GRID LINES FOR ABSCISSA
```

```
      KLL = NP/ID
      IC = ID*KLL
      IF(NP.GT.IC) GO TO 50
      RETURN
   50 KLL = KLL + 1
      NP = KLL*ID
      RETURN
      END
$IBFTC ST1012  NOREF
      SUBROUTINE BILIN(LD,LN,FS,TI,A,B,CON,IFL)
      DIMENSION FIRST(1)
      DIMENSION A(50),B(50),S(50),T(50),C(50),C1(50)
      COMPLEX SS(50),TT(50)
C   WAI IS THE CUTOFF FREQUENCY
C
      WAI = TAN(FS*TI/2.)
      IF(IFL)1,1,2
    1 READ (5,102) (SS(I),I=1,LD)
      WRITE(6,104)  (SS(I),I=1,LD)
      CALL RCOEF(LD,SS,S)
      GO TO 3
    2 READ (5,103) (S(I),I=1,LD)
      WRITE(6,104)   (S(I),I=1,LD)
    3 CONTINUE
      IF(LN)4,14,4
    4 IF(IFL)20,20,5
   20 CONTINUE
      READ (5,102) (TT(I),I=1,LN)
      WRITE(6,104)  (TT(I),I=1,LN)
      CALL RCOEF(LN,TT,T)
      GO TO 6
    5 READ (5,103) (T(I),I=1,LN)
      WRITE(6,104)   (T(I),I=1,LN)
    6 K = LD - LN
      DO 7 I=1,LD
    7 S(I)=S(I)*WAI**I
      DO 8 I=1,LN
    8 T(I)=T(I)*WAI**I
      CON=CON*WAI**K
      CALL XDEN(LD,S,B)
      CALL XDEN(LN,T,A)
      IF(K)11,12,9
    9 CONTINUE
      C(1) = 1.
      C(2) = FLOAT(K)
```

```
      CALL COEF(K,C)
      CALL PSMPY(C,K,A,LN,C1,IORD)
      NK = IORD + 1
      DO 10 I=1,NK
   10 A(I) = C1(I)
      GO TO 12
   11 K = -K
      C(1) = 1.
      C(2)=FLOAT(K)
      CALL COEF(K,C)
      CALL PSMPY(C,K,B,LD,C1,IORD)
      NK = IORD + 1
      DO 110 I=1,NK
  110 B(I) = C1(I)
   12 CONTINUE
      BN = B(1)
      A(1) = A(1)*CON/BN
      DO 13 I=2,NK
      II=I-1
      A(I)=A(I)*CON/BN
   13 B(II)=B(I)/BN
      GO TO 16
   14 CONTINUE
      DO 15 I=1,LD
   15 S(I)=S(I)*WAI**I
      CON=CON*WAI**LD
      CALL XDEN(LD,S,B)
      A(1) = 1.
      A(2) = FLOAT(LD)
      CALL COEF(LD,A)
      NK = LD + 1
      GO TO 12
   16 CONTINUE
  102 FORMAT(8E10.0)
  103 FORMAT(8E10.0)
  104 FORMAT(/(1P8E16.7/))
      LAST=5
      RETURN
      END
```

REFERENCES

1. Rader, C. M.; and Gold, B.: Digital Filter Design Techniques in the Frequency Domain. Proc. IEEE, vol. 55, no. 2, Feb. 1967, pp. 149-171.

2. Kuo, F. F.; and Kaiser, J. F.: System Analysis by Digital Computer. John Wiley and Sons, Inc., New York, N.Y., 1966.

3. Steiglitz, K.: The Approximation Problem for Digital Filters. Tech. Rep. 400-56, Dept. Electrical Engr., New York Univ. College of Engr., March 1962.

4. Chen, W. H.: Linear Network Design and Synthesis, McGraw-Hill Engineering Series, New York, N. Y., 1964.

5. Geffe, P. R.: Simplified Modern Filter Design. John F. Rider pub., New York, N.Y., 1963.

6. Arabadjis, C.: An Investigation of Linear Finite Optimum and Frequency Constrained Digital Filters for Random and Deterministic Signals. Defense Electronics Div., General Electric Co., Syracuse, New York, Aug. 1963.

TABLE I.- INPUTS FOR DIGITAL FILTER SYNTHESIS PROGRAM

Card	Column		
1	Comment card to be printed verbatim (columns 1-80)		
2	Column		
(all integers right justified)	1-5	LD	Degree of the polynomial in the denominator of the filter transfer function f(s). Exception is noted with conjugate roots (poles of F(s)).
	6-10	LN	Degree of the polynomial in the numerator of the transformation function f(s). Exception is noted with conjugate roots (zeros of F(s)). LN is zero if the numerator is a constant.
	11-15	IW	Plot ratio; ratio of the total number of samples to number of plotted samples
	16-20	JJ	Flag for print/plot options
		if JJ =	1 Print and plot 2 Plot only 3 Print only 4 Do neither
	21-25	IFL	Input flag
		positive	f(s) function in terms of coefficients of polynomials
		negative	f(s) function in terms of poles and zeros of f(s).
3	1-10	CON	Leading constant of the transfer function f(s)
(all floating point)	11-20	FR	The scaling frequency ratio
	21-30	TIME	Total time of the filter run in seconds
	31-40	TI	Time interval between successive points in the input data in seconds

The next card (or cards if required) provide inputs to subroutine BILIN in either the factored or unfactored form of the prototype transfer function depending on whether IFL is positive or negative. The fields are 10 columns of 8 floating point numbers per card, as illustrated in table II. The denominator of the transfer function is given first and the numerator follows immediately. The coefficients of these polynomials must be entered as real numbers. Roots of the polynomials entered in factored form must be complex numbers. If a root has a nonzero imaginary part, its conjugate is assumed to be another root and must not be entered as input, that is, complex conjugate pairs are always assumed and one complex number represents the pair.

TABLE II.- SAMPLE INPUTS FOR SUBROUTINE BILIN

1. $$F(s) = \frac{s^2 + 3s + 8}{s^3 + 3s^2 + 3s + 1}$$

LN = 2
LD = 3
IFL = 1
CØN = 1

Floating point mode

Column 1	11	21	30
3.	3.	1.	
3.	8.		

2. $$F(s) = \frac{3}{(s + 1.5 - j2.4)(s + 1.5 - j2.4)}$$

LN = 0
LD = 1
IFL = -1
CØN = 3

Complex mode

Column 1	11	20
-1.5	2.4	

3. $$F(s) = \frac{4(s + 1)(2 - 1 + j3)(s - 1 - j3)}{(s + 2)(s + 3 + j1)(s + 3 - j1)}$$

LN = 2
LD = 2
IFL = -1
CØN = 4

Complex mode

Column 1	11	21	31	40
-2.	0.0	-3.	1.	
-1.	0.0	1.	3.	

4. $$F(s) = \frac{1.5(s^2 + 2s + 3)}{s^{10}+3s^9+8s^8+7s^7+2s^6+2s^5+s^4+3s^3+4s^2+s+6}$$

LN = 2
LD = 10
IFL = 1
CØN = 1.5

Floating point mode

Column 1	11	21	31	41	51	61	71	80
3.	8.	7.	2.	2.	1.	3.	4.	
1.	6.							
2.	3.							

PRECEDING PAGE BLANK NOT FILMED.

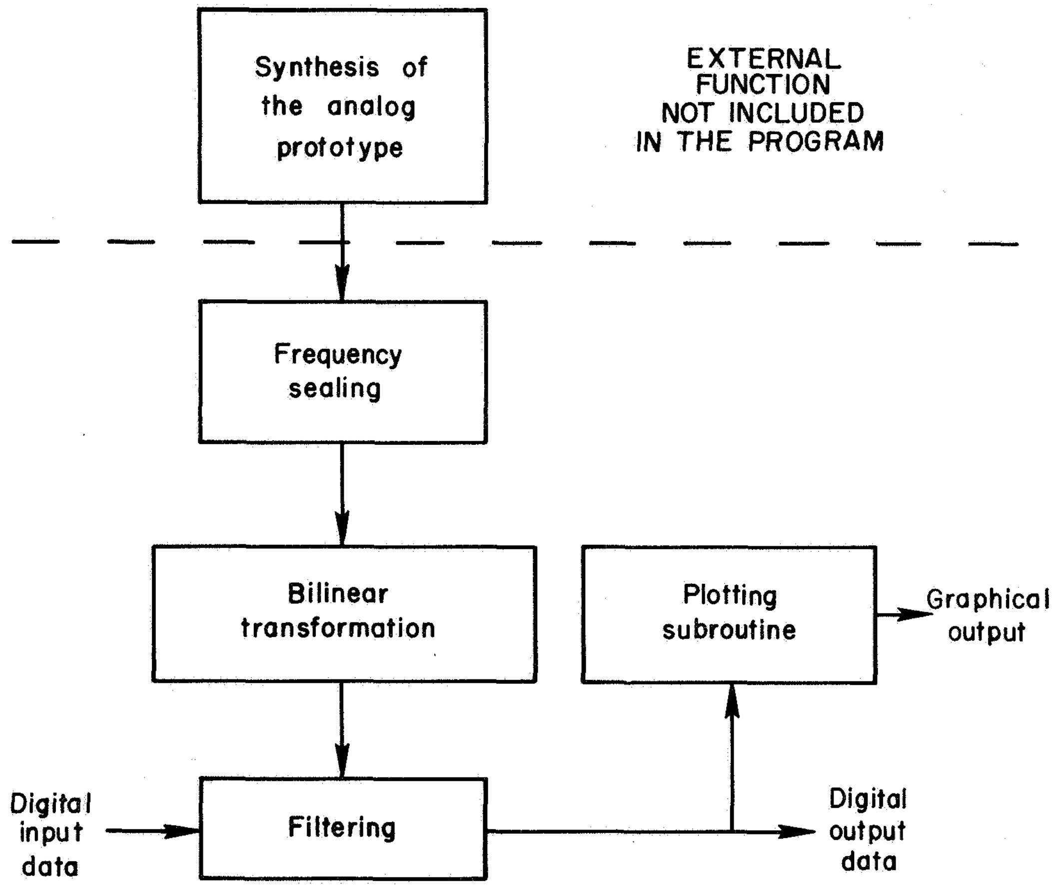
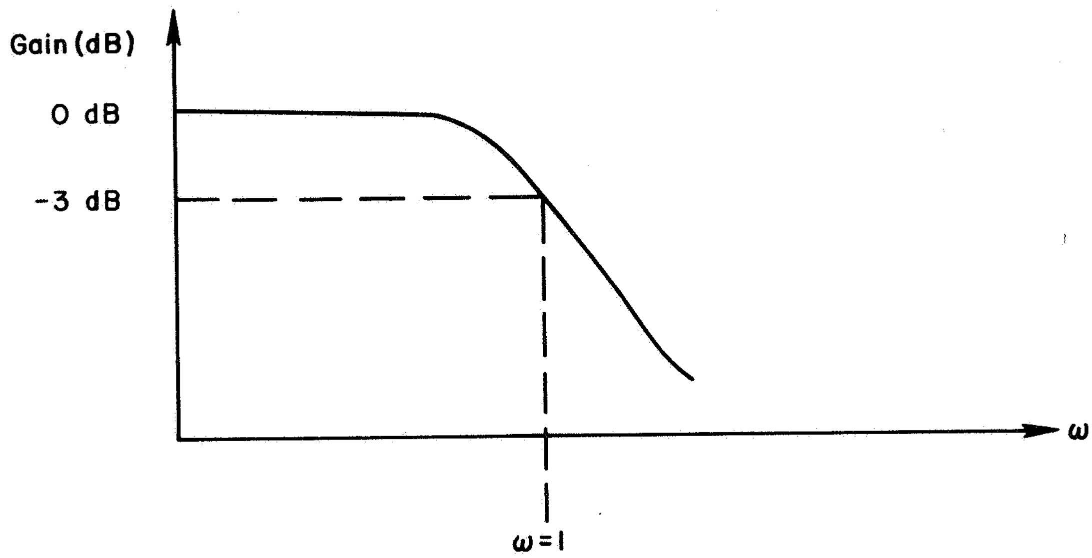


Figure 1.- Functional block diagram of the digital filter synthesis program.



$$F(s) = \frac{out}{in} = \frac{1}{s^2 + \sqrt{2}\,s + 1}$$

Figure 2.- Characteristics of a second order Butterworth filter.

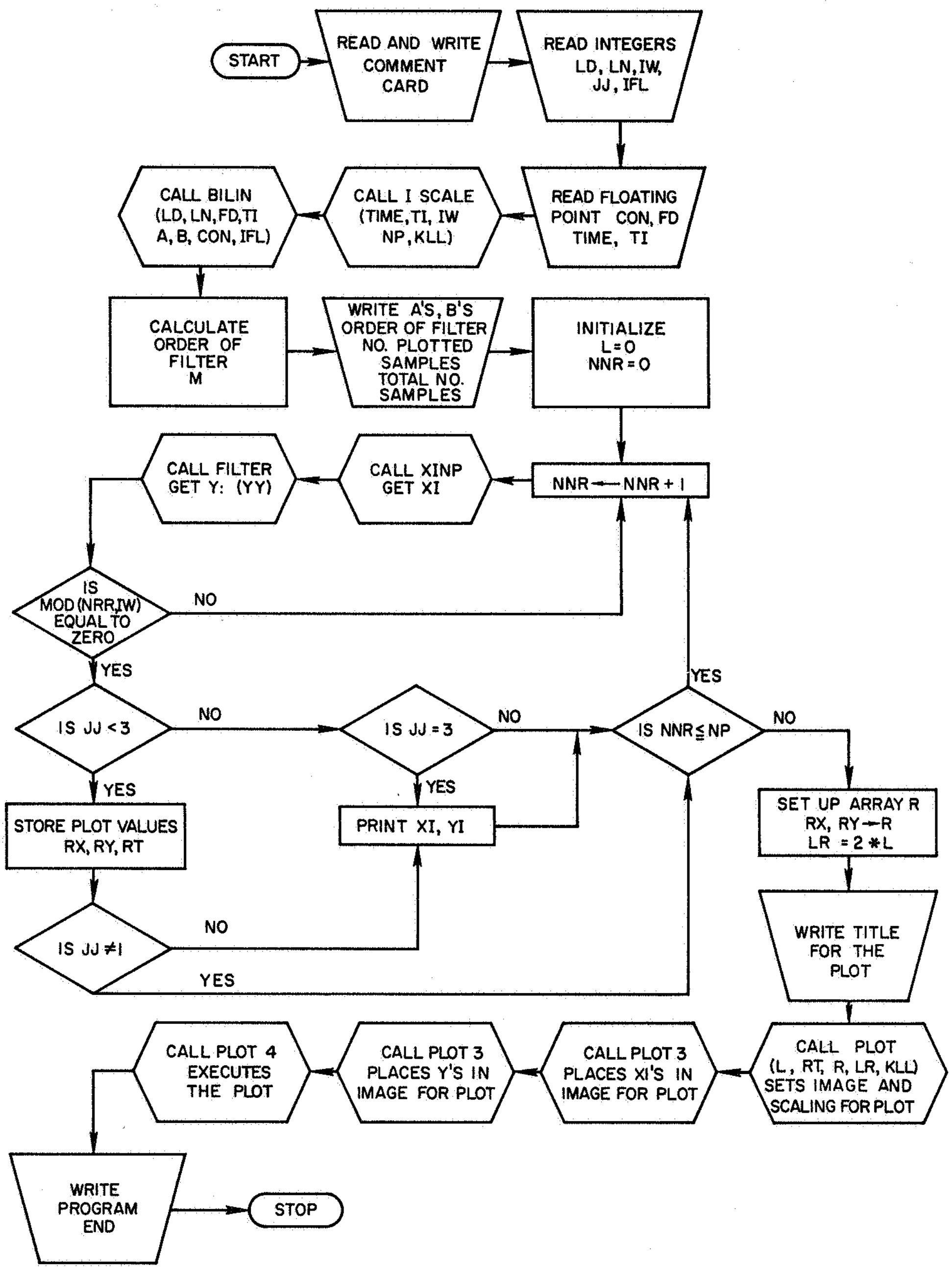


Figure 3.- Flow chart for main program.

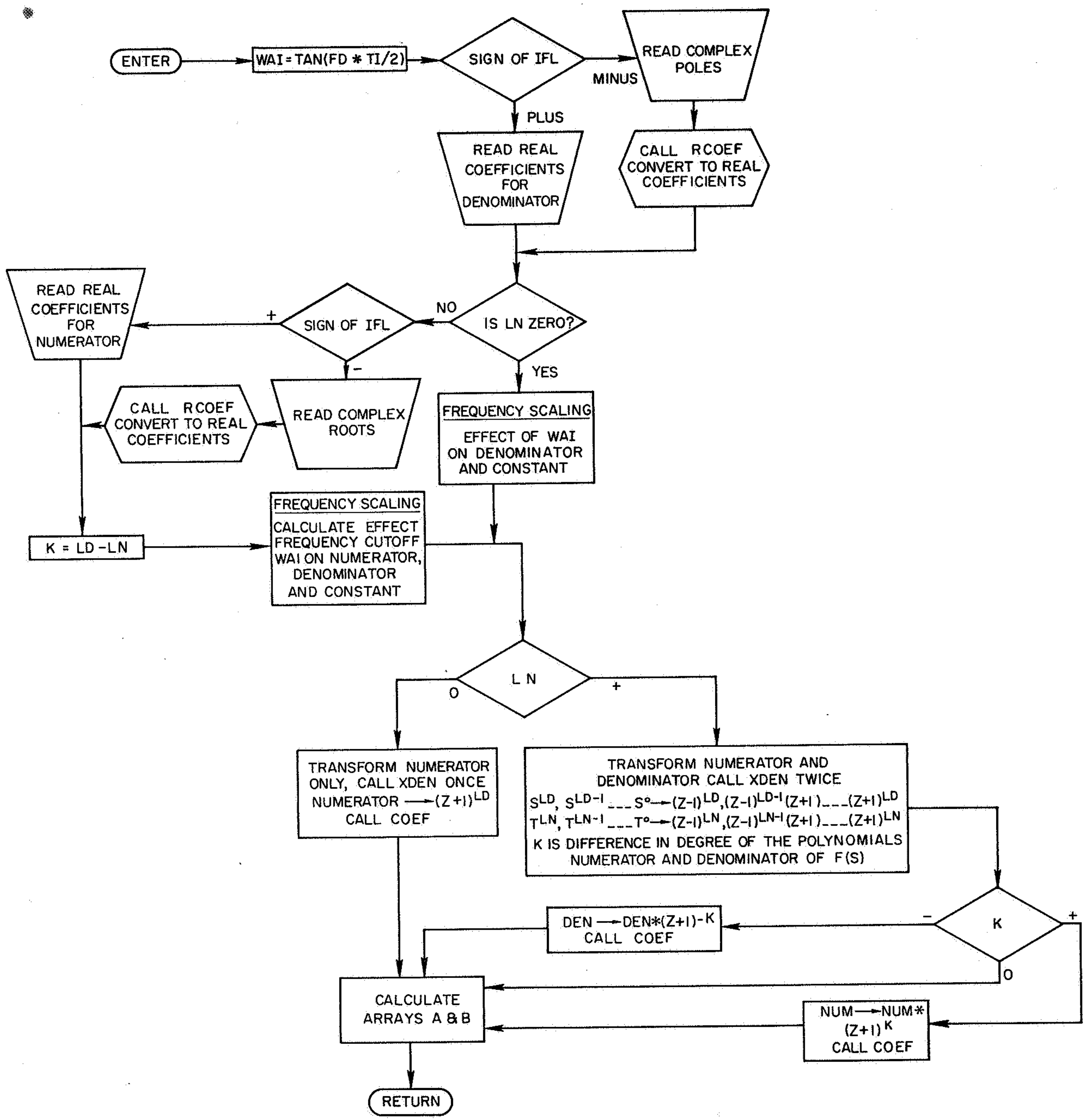


Figure 4.- Flow chart for subroutine BILIN.