AD-769 211

A GENERAL METHOD AND FORTRAN PROGRAM FOR THE DESIGN OF RECURSIVE DIGITAL FILTERS

Joseph J. Wolcin

Naval Underwater Systems Center
New London, Connecticutt

25 September 1973

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-769211

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER TR 4629	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A GENERAL METHOD AND FORTRAN PROGRAM FOR THE DESIGN OF RECURSIVE DIGITAL FILTERS		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Joseph J. Wolcin		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Underwater Systems Center New London Laboratory New London, Conn. 06320		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 3ö0200 SPA400
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 25 September 1973 13. NUMBER OF PAGES 46
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)

Recursive Digital Filters
Frequency Selective Digital Filters

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

A general method for the design of digital filters is presented, and a FORTRAN program that follows the steps of the method is described. The computer program can design any low-pass, high-pass, bandpass, or band-reject digital filter by using either Butterworth, Chebyshev, or elliptic approximations to the ideal rectangular type frequency response. The user need only provide the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. program with a set of structured specifications, and as output he obtains the transfer function of the minimum order filter that meets his requiremer‹ ;. For convenience, the transfer function is expressed in a form that allows immediate implementation in a cascade type realization.

PREFACE

This report was prepared under a local indirect funding project (NUSC Job Order 300200, Cost Center SPA400).

The Technical Reviewer for this report was Dr. Albert H. Nuttall (Code TC).

REVIEWED AND APPROVED: 25 September 1973

R. M. Dunlap

R. M. Dunlap
Director for Plans and Analysis

Inquiries concerning this report may be addressed to the author, New London Laboratory, Naval Underwater Systems Center, New London, Connecticut 06320

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

A GENERAL METHOD AND FORTRAN PROGRAM FOR THE DESIGN OF RECURSIVE DIGITAL FILTERS

INTRODUCTION

A very tedious problem often confronting engineers is the design of a frequency selective digital filter that meets a desired set of specifications. The task is especially laborious and the theory quite complex when the filter has to be designed to satisfy very demanding requirements. Considerations such as these have often caused the designer to settle for a filter with characteristics further from the ideal rectangular type response than is desired. Described herein is a general method for the design of digital filters and a versatile FORTRAN program (listed in the appendix) that will carry out the design of any low-pass, high-pass, bandpass, or band-reject digital filter in any of the standard forms (i.e., Butterworth, Chebyshev, or elliptic). The user need only provide the program with a set of structured specifications, and as output he obtains the transfer function of the minimum order filter that meets his requirements. For convenience, the transfer function is expressed in a form that allows immediate implementation is a cascade type realization. The user also has the option of obtaining plots of the frequency response (both magnitude and phase) and the unit sample response of the filter.

Complicated design problems will no longer demand so much time and effort, as has especially been the case with elliptic filters. Moreover, it is suspected that elliptic filters may be preferred now for many applications since they normally require a lower order filter than that required by the currently more common Butterworth and Chebyshev filters. The advantage lies in the fact that a lower order implies fewer computations in the recursive scheme used in implementing the filter. The principal argument against elliptic filters in the past has been the time consuming and complicated design process.

In this report instructions for the use of the computer program are discussed first and then its use by example is illustrated. Discussion of the underlying theory is deferred to the latter part of the report since its comprehension is not necessary in order to successfully use the program and the resulting filter.

USE OF THE COMPUTER PROGRAM

One data card, which provides the program with the desired set of specifications, should be used for each filter to be designed. The structure of the data card is shown in table 1. Also, figure 1, which illustrates how the various input parameters should be interpreted on plots of typical frequency curves, should be helpful to the user. More detailed descriptions of the input parameters and other special notes now follow.

Table 1. Data Card Structure

Input Variable	Format	Columns	Brief Description
SRATE	F8.0	1-8	Sampling rate
FCLOW	F8.0	9-16	Lower cutoff frequency
FCHIGH	F8.0	17-24	Higher cutoff frequency
RIPPLE	F6.0	25-30	Passband ripple
FSLOW	F8.0	31-38	Lower stopband boundary
FSHIGH	F8.0	39-46	Higher stopband boundary
STPLVL	F4.0	47-50	Minimum stopband attenuation
IKIND	I1	51	Kind of filter desired (i.e., Chebyshev, Butterworth, elliptic)
ITYPE	I1	52	Type of filter desired (e.g., low-pass, or high-pass.)
NPOLES	I2	53-54	Order desired; set equal to zero for calculation of order
IPLOT	I1	56	Set ≠0 for plot of magnitude of frequency response
NPTS	I4	57-60	Number of points in unit sample response
FLOW	F10.0	61-70	Lowest frequency in plots
FHIGH	F10.0	71-80	Highest frequency in plots

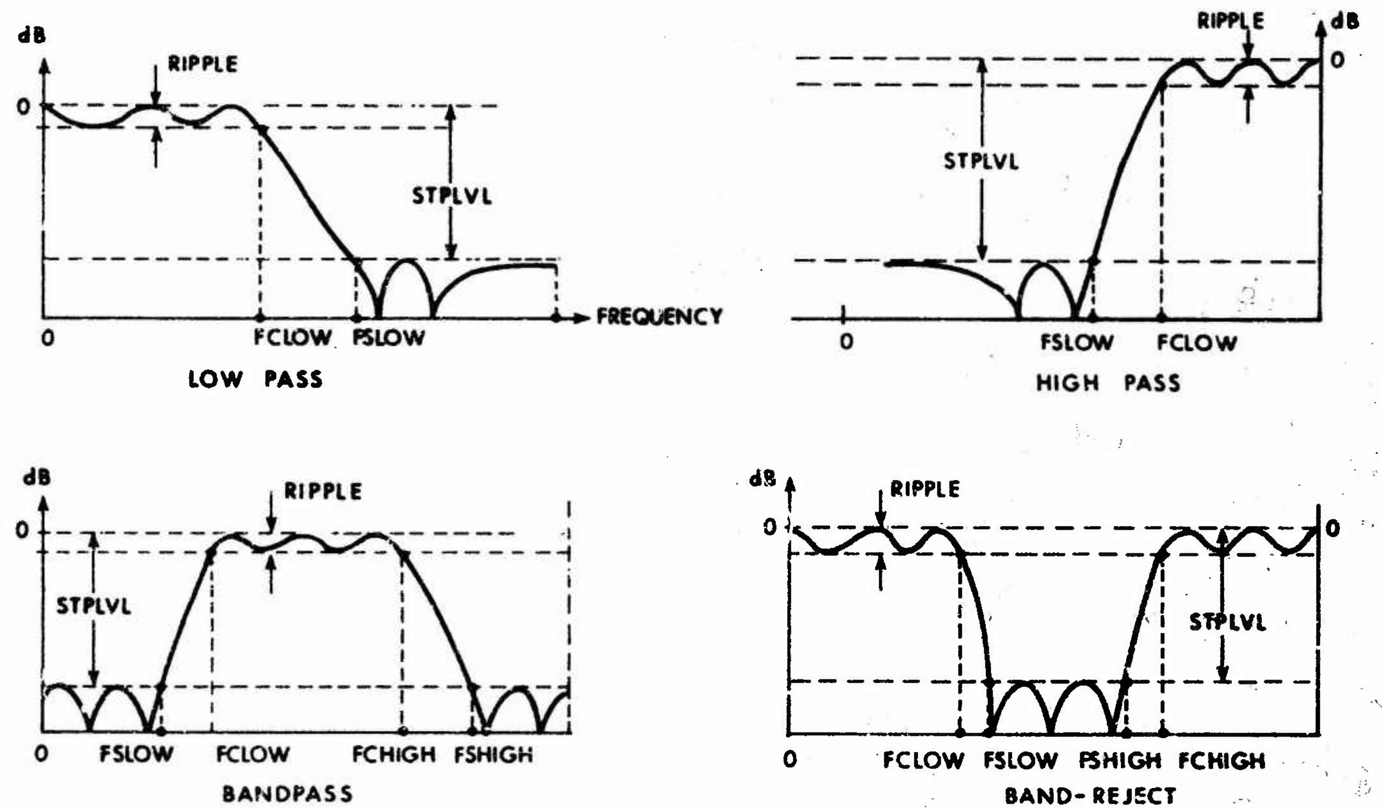

Figure 1. Illustrative Frequency Characteristics (Elliptic)

The first input parameter, SRATE, is the sampling rate and is normally the number of data samples per second. However, any time unit can be used, and in fact, one might choose a fictitious value here with the intention of constructing a filter whose cutoff is a certain percentage of the Nyquist rate. For example, a low-pass filter with a cutoff that is one-quarter of Nyquist could be obtained by choosing SRATE to be 200 Hz and the cutoff to be 25 Hz. Note that this same filter, if applied to data sampled at some other rate, would no longer have cutoff at 25 Hz, but its cutoff would shift to one-quarter of the new Nyquist rate.

The quantities FCLOW and FCHIGH are the cutoff frequencies of the filter, and must be given in the same units as SRATE. Note that if a low-pass or high-pass filter is desired, there is only one cutoff frequency and this value should be read into FCLOW.

The quantity RIPPLE is the maximum passband ripple (in decibels) to be allowed. In the case of a Butterworth filter, where the response is monotonic, RIPPLE is not used. Note also that RIPPLE should always be a positive quantity.

The parameters FSLOW and FSHIGH are similar to FCLOW and FCHIGH, but they mark the boundaries between stopband and transition band. Here too the quantity FSHIGH is not used in the low-pass or high-pass case.

The next specification, STPLVL, is the minimum stopband attenuation (in decibels) the user will tolerate. This should also always be a positive quantity.

The value 1, 2, or 3 of the variable IKIND determines which kind of filter will be designed, that is, Chebyshev, Butterworth, or elliptic, respectively.

Similarly, the value 1, 2, 3, or 4 of ITYPE determines the shape of the filter that will be designed, that is, low-pass, high-pass, bandpass, or band-reject, respectively.

The value of NPOLES should normally be read in as zero, and the correct order will be calculated. However, the user may (except in the elliptic case) choose a positive integer here if he desires a filter of some specific order. This integer should be the order of the basic low-pass structure desired before transformation. In converting a low-pass filter to either a bandpass or band-reject filter, the method doubles the number of poles; thus, to get a band-reject filter of order 18, NPOLES should be set equal to 9. If NPOLES is given as nonzero, the quantities FSLOW, FSHIGH, and STOLVL are not used since the given order determines these quantities uniquely. NPOLES is restricted to be less than or equal to 20, since it is anticipated that filters of these orders can meet any reasonable set of requirements. However, if necessary, one could merely increase the array dimensions used in the program in order to obtain filters of higher order.

The remaining input parameters control the plots produced by the program. If the user does not have access to the Stromberg Carlson 4060 Integrated Graphics System, he should remove the corresponding code or replace it by a code that is compatible with his plotter.

If for some reason the user does not want any of the three available plots, he should set the parameter IPLOT to zero. The number of points desired in the unit sample response should be read into NPTS. This quantity must not be greater than 1000 and has a default value of 100 if the corresponding field in the data card is left blank. The values given to the quantities FLOW and FHIGH merely restrict the plots of magnitude and phase to the range between these two frequencies. All information can be seen by plotting from zero to Nyquist, since the curve is periodic with period SRATE and is symmetric about Nyquist. However, one can expand a small frequency range for closer examination.

If more than one filter is desired, a similar data card for each should be used and a blank card should be placed last.

The following typical example illustrates the points just discussed. Suppose that an elliptic bandpass digital filter with cutoffs at 2000 Hz and 3000 Hz is desired. Assume that the sampling rate is 10,000 Hz, that the filter is to be at least 30 dB down by 1800 Hz and 3200 Hz, and that the tolerable ripple in the passband is 0.5 dB. The following is a list of the proper input parameters for this problem:

SRATE	10000.0
FCLOW	2000.0
FCHIGH	3000.0
RIPPLE	0.5
FSLOW	1800.0
FSHIGH	3200.0
STPLVL	30.0
IKIND	3
ITYPE	3
NPOLES	0
IPLOT	1
NPTS	200
FLOW	0.0
FHIGH	5000.0

Figures 2 through 5 show the printed and plotted outputs for the example in question. Note that the range of the magnitude plot is from 0 dB to -50 dB, which can easily be increased if desired.

```
INPUT SPECIFICATIONS

SRATE = 10000.00

     FCLOW =  2000.00

          FCHIGH =  3000.00

               RIPPLE =  .500

                    FSLOW =  1800.00

                         FSHIGH =  3200.00

                              STPLVL =  30.00

                                   IKIND = 3

                                        ITYPE = 3

                                             NPOLES =  0

                                                  IPLOT = 1
THE MINIMUM ORDER FILTER WHICH MEETS THESE SPECIFICATIONS IS ORDER  8


THE TRANSFER FUNCTION,H(Z), OF THE DESIRED FILTER CAN BE EXPRESSED
IN THE FOLLOWING FORM


                           Z**2+B(1)*Z+B(2)        Z**2+B(3)*Z+B(4)                Z**2+B(N-1)*Z+B(N)
              H(Z) = CNST * ----------------  *  ----------------  * ..., *  ------------------
                           Z**2+C(1)*Z+C(2)        Z**2+C(3)*Z+C(4)                Z**2+C(N-1)*Z+C(N)

WHERE THE COEFFICIENTS ARE GIVEN BY

                 CNST =     .313180351429645-001
C( 1)=  -.293707645287690+000          B( 1)=   .146671954291418+001
C( 2)=   .744807703617405+000          B( 2)=   .100000000000000+001
C( 3)=   .293707645287690+000          B( 3)=  -.146671954291418+001
C( 4)=   .744807703617405+000          B( 4)=   .100000000000000+001
C( 5)=  -.608822957843175+000          B( 5)=   .892503533653899+000
C( 6)=   .930411492663660+000          B( 6)=   .100000000000000+001
C( 7)=   .608822957843175+000          B( 7)=  -.892503533653899+000
C( 8)=   .930411492663660+000          B( 8)=   .100000000000000+001


THE TRANSFER FUNCTION IS EXPRESSED IN THE ABOVE FORM SO THAT
THE COEFFICIENTS CAN BE USED AS GIVEN IN A CASCADE IMPLEMENTATION OF THE FILTER
```

Figure 2. Sample Printout

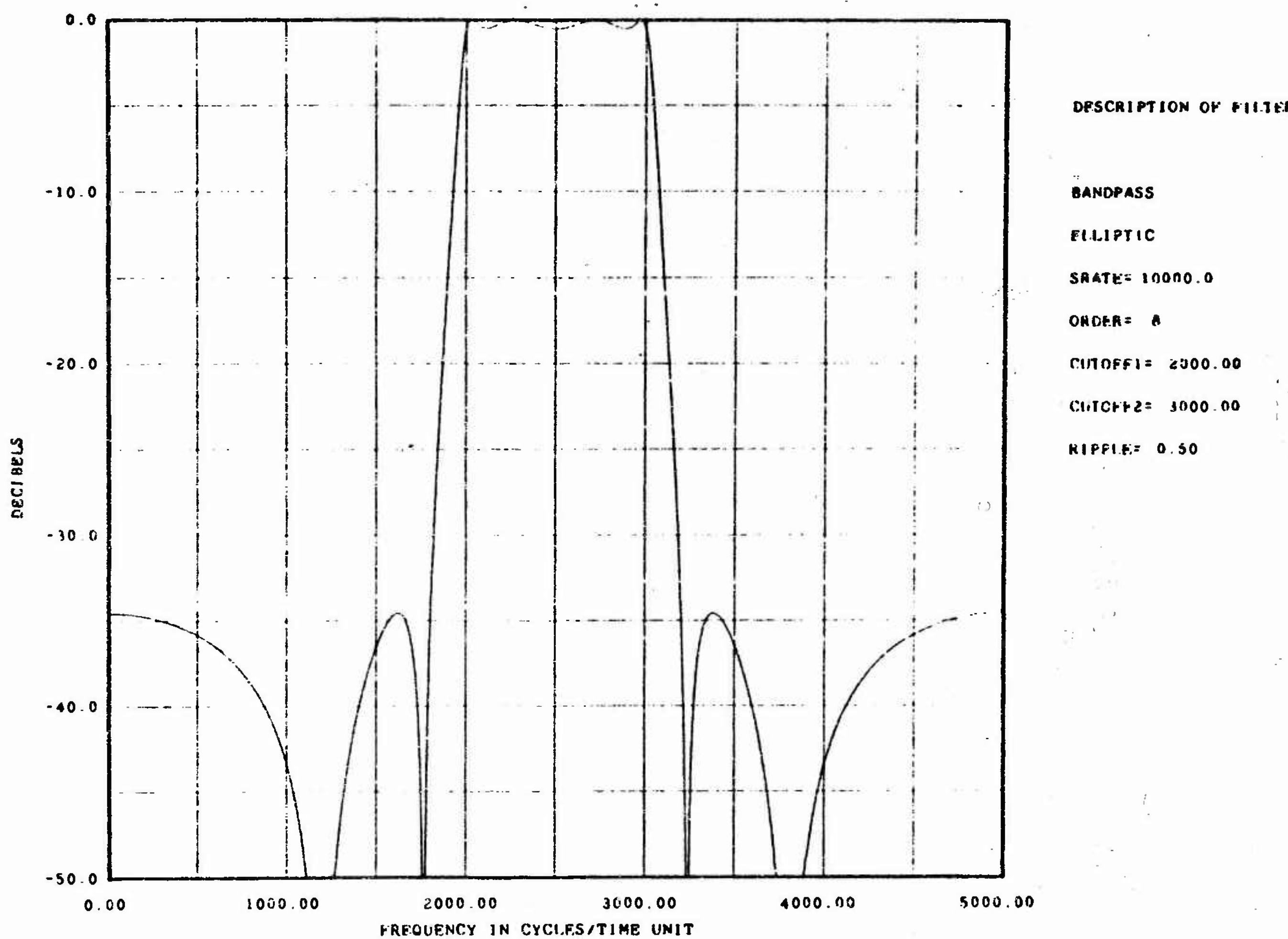

Figure 3. Frequency Characteristic (Magnitude)

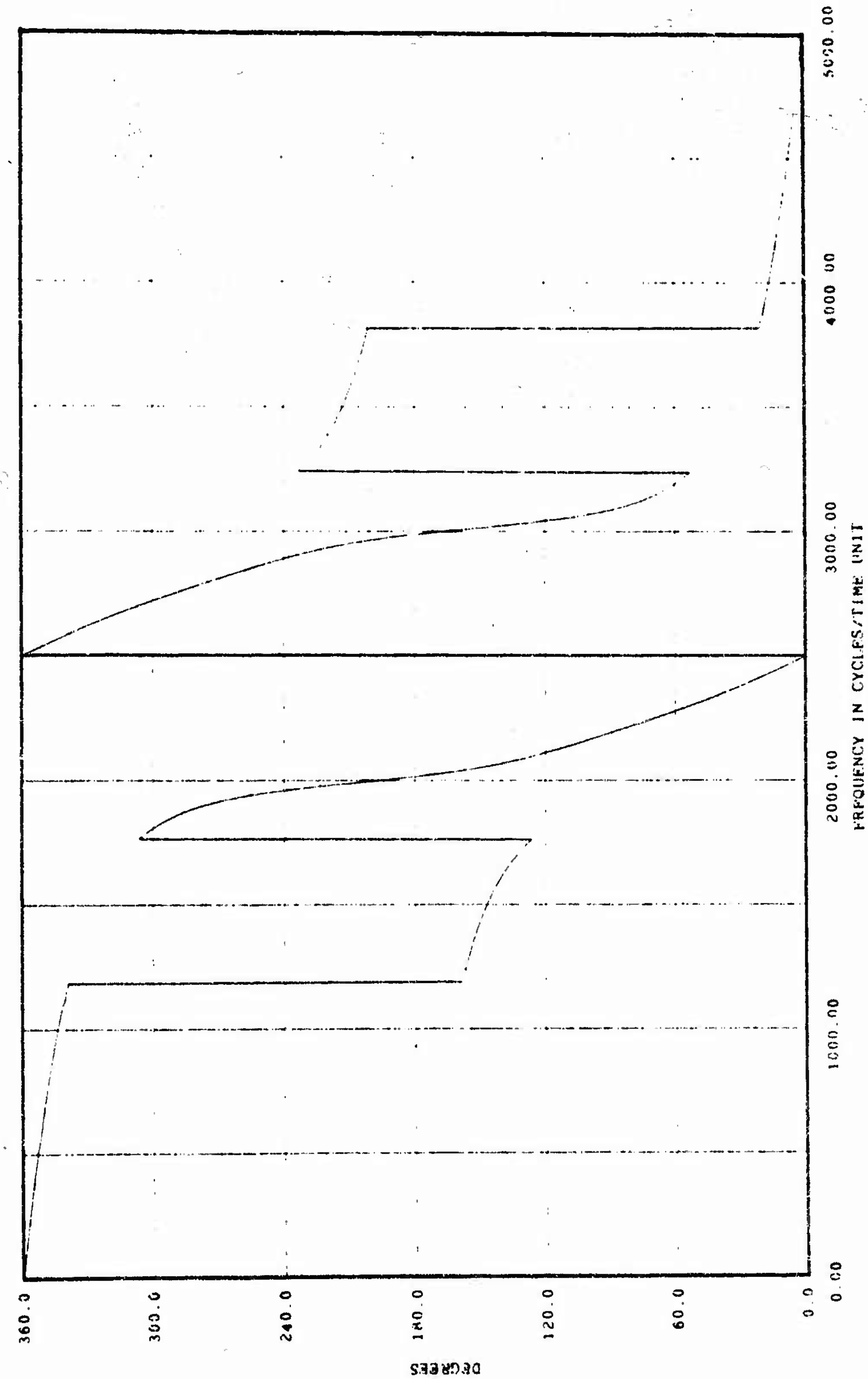

Figure 4. Frequency Characteristic (Phase)

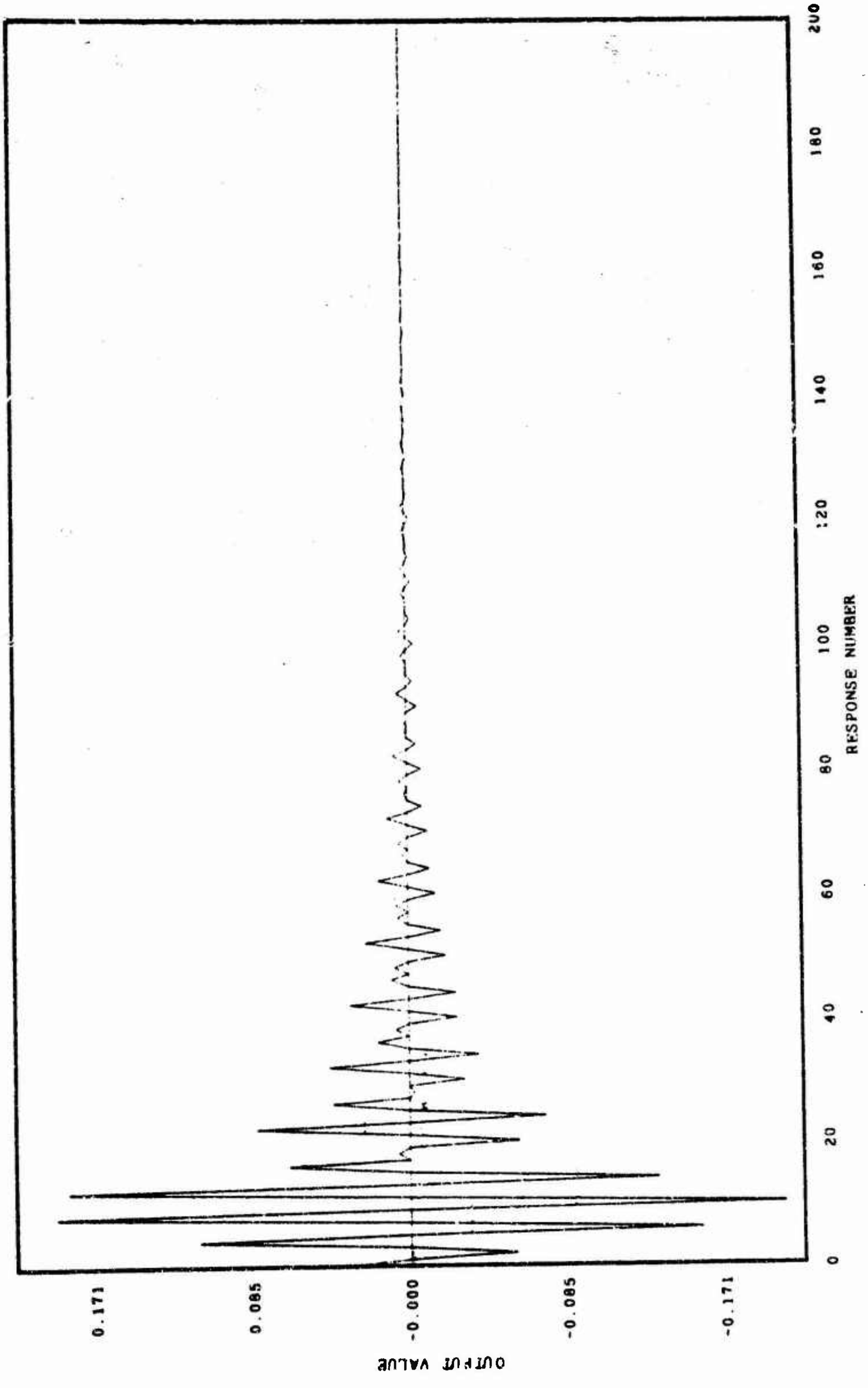

Figure 5. Unit Sample Response

THEORY [illegible] PROGRAMMING METHOD

PRELIMINARIES

Each of the major steps presented in the following synopsis of operations performed by the program will be discussed in greater detail in succeeding sections:

a. Certain critical frequencies in the digital z-plane are given as input parameters. These must be transformed to continuous s-plane critical frequencies since much of the design process is carried out in the s-plane.

b. The minimum order filter of the appropriate kind that will meet the specifications is computed.

c. The s-plane pole-zero pattern of a unity bandwidth low-pass analog filter of proper order is determined.

d. These s-plane poles and zeros are mapped to the z-plane poles and zeros of the required filter.

e. The poles and zeros of the desired transfer function determine the function up to some constant factor; the required constant is obtained by forcing the maximum value of the amplitude characteristic to be unity (0 dB).

f. Next, the transfer function is manipulated into a form that can be immediately implemented in cascade form.

g. The magnitude and phase of the transfer function evaluated on the unit circle in the z-plane (i.e., the frequency response) is plotted in the frequency range specified by the user.

h. Finally, the unit sample response of the desired filter is plotted.

A few facts from filter theory are now presented as background for later discussion. The three kinds of filters considered in this report have magnitude characteristics in the analog plane defined by the following equations:

Butterworth $$|H(j\Omega)|^2 = \frac{1}{1+\Omega^{2n}} \tag{1a}$$

Chebyshev $$|H(j\Omega)|^2 = \frac{1}{1+\epsilon^2 V_n^2(\Omega)} \tag{1b}$$

elliptic $$|H(j\Omega)|^2 = \frac{1}{1+\epsilon^2 \psi_n^2(\Omega)} , \tag{1c}$$

where

$H(\cdot)$ = the transfer function of the unity bandwidth low-pass filter

Ω = radian frequency

n = the order of the filter

ϵ^2 = $10^{R/10} - 1$, where R is the amount of passband ripple (in decibels)

$V_n(\cdot)$ = Chebyshev polynomial of order n

$$\psi_n(\Omega) = \begin{cases} \mathrm{sn}\left[n \dfrac{K(k_1)}{K(k)} \mathrm{sn}^{-1}(\Omega, k), k_1\right], & n \text{ odd} \\ \mathrm{sn}\left[K(k_1) + n \dfrac{K(k_1)}{K(k)} \mathrm{sn}^{-1}(\Omega, k), k_1\right], & n \text{ even,} \end{cases}$$

where

$\mathrm{sn}(\cdot,\cdot)$ = Jacobian elliptic function

$K(\cdot)$ = complete elliptic integral of the first kind

$k = 1/\Omega^s$, Ω^s being the start of the stopband

$k_1 = \epsilon/\sqrt{A^2 - 1}$, with $A = 10^{S/20}$ and S equal to the minimum stopband attenuation (in decibels).

Two types of mappings that play a major role in the method will now be presented. The first is a map that converts a unity bandwidth low-pass analog filter, say $H(s)$, into an analog filter of another type with different cutoff(s). If $H(s)$ is such a filter, then we see that

$H(s/\Omega^c)$ is low-pass with cutoff Ω^c (2a)

$H(\Omega^c/s)$ is high-pass with cutoff Ω^c (2b)

$$H\left[\frac{s^2 + \Omega_1^c \Omega_2^c}{s\left(\Omega_2^c - \Omega_1^c\right)}\right] \quad \text{is bandpass with cutoffs } \Omega_1^c \text{ and } \Omega_2^c \tag{2c}$$

and

$$H\left[\frac{s(\Omega_2^c - \Omega_1^c)}{s^2 + \Omega_1^c \Omega_2^c}\right] \quad \text{is band-reject with cutoffs } \Omega_1^c \text{ and } \Omega_2^c . \tag{2d}$$

A second mapping, called the bilinear transformation, maps the s-plane to the z-plane and can be used to convert an analog filter to a digital filter. The mapping is given by

$$z = \frac{1 + s}{1 - s} . \tag{3}$$

Note that the imaginary axis in the s-plane is mapped to the unit circle in the z-plane and that the left half of the s-plane is mapped inside the unit circle in the z-plane. Thus, a stable analog filter will be mapped to a stable digital filter and the digital frequency response from zero to Nyquist will take on exactly the same values as the analog frequency response from zero to infinity. Also, note that the aliasing problem inherent in filters designed by the method of impulse invariance is not present here since the mapping is invertible. It is often stated that this method has the drawback of warping the frequency scale. However, no real problem exists, since the critical frequencies in a design problem can be "prewarped," as described in the next section.

In the literature, the terms Butterworth, Chebyshev, and elliptic are used in the description of low-pass analog filters. In this report these terms are also used in describing filters of other types that have been obtained by the previous two mappings.

ANALOG CRITICAL FREQUENCIES

The given digital critical frequencies must be transformed to the analog plane in such a way that when they are later mapped back to the z-plane by the bilinear transformation they will be mapped to the proper values. This "prewarping" of the critical frequencies is carried out by the map

$$\Omega = \tan\left(\frac{\omega T}{2}\right), \tag{4}$$

where

Ω = continuous radian frequency

ω = digital radian frequency

T = time between samples.

The following argument will convince the reader of this result. Using the inverse of the bilinear transformation (3), we can relate any z-plane point to its corresponding s-plane point, i.e.,

$$s = \frac{z - 1}{z + 1}.$$

For $z = e^{j\omega T}$, at which points we obtain the digital frequency response, the corresponding s-plane point is

$$s = \frac{e^{j\omega T} - 1}{e^{j\omega T} + 1} = j \tan\left(\frac{\omega T}{2}\right).$$

But the analog frequency response is obtained by evaluating the transfer function at $s = j\Omega$. Thus, we have the correspondence dictated by (4). Now when we apply the bilinear transformation to our s-plane filter, we can be sure we will end up with a filter with proper digital critical frequencies.

DETERMINATION OF MINIMUM ORDER

We will first solve the problem for the low-pass case and then show how any of the other cases can be reduced to an equivalent problem.

Let ω^c and Ω^c be the desired digital and analog radian cutoff frequencies, respectively, and let ω^s and Ω^s be the frequencies at which the stopband begins. By (4), these frequencies are connected by the relations

$$\Omega^c = \tan\left(\frac{\omega^c T}{2}\right)$$

and

$$\Omega^s = \tan\left(\frac{\omega^s T}{2}\right) ,$$

where

$$\omega^c = 2\pi f^c$$

$$\omega^s = 2\pi f^s$$

f^c and f^s are the given input specifications in hertz.

Next, define the "transition ratio," an important factor in determining the required order, by

$$\Omega^T = \frac{\Omega^s}{\Omega^c} .$$

Using the expressions (1a)-(1c) given previously for the three magnitude characteristics, we can determine N, the minimum order that will satisfy the filter requirements. Omitting details of the algebra, we now present general expressions for calculation of N.

If a Butterworth filter is desired, N can be computed as the smallest integer that is greater than

$$\frac{\log_{10}\left(10^{S/10} - 1\right)}{2 \log_{10} \Omega^T} .$$

For a Chebyshev filter, we compute the required order by finding the smallest value of N:

$$V_N(\Omega^T) \geq \left(\frac{10^{S/10} - 1}{10^{R/10} - 1}\right)^{1/2} .$$

Finally, if an elliptic filter is desired, we determine N by finding the smallest integer that is greater than

$$\frac{K\left(\sqrt{1 - k_1^2}\right) K\,(1/\Omega^T)}{K(k_1)\, K\left[\sqrt{1 - (1/\Omega^T)^2}\right]} .$$

All the parameters used in these expressions have been defined in this or previous sections.

The only additional problem arises in the elliptic case and results from the fact that the expression above will ordinarily not be an integer and must be rounded upwards. Since it is required that

$$N = \frac{K\left(\sqrt{1 - k_1^2}\right) K(1/\Omega^T)}{K(k_1)\, K\left[\sqrt{1 - (1/\Omega^T)^2}\right]} ,$$

the value of k_1 must be recalculated after N is determined. A convenient formula* for recomputation of k_1 is

$$k_1 = \left\{ \frac{2q^{0.25}\left[1 + \sum_{i=1}^{\infty} q^{i(i+1)}\right]^2}{1 + 2\sum_{i=1}^{\infty} q^{i2}} \right. ,$$

where

$$q = \exp\left\{-\frac{N\pi K(\sqrt{1 - k^2})}{K(k)}\right\} .$$

Note here that $k = 1/\Omega^T$.

---

*R. M. Fano, "A Note on the Solution of Certain Approximation Problems in Network Synthesis," J. Franklin Institute, vol. 249, 1950, pp. 189-205.

Now the problem of determination of minimum order for the low-pass case is solved. For any other case we define Ω^T differently but proceed exactly as in the low-pass case. For the high-pass case the appropriate definition is

$$\Omega^T = \frac{\Omega^c}{\Omega^s}.$$

For the bandpass case choose

$$\Omega^T = \min\left(\Omega_1^T,\ \Omega_2^T\right),$$

and for the band-reject case choose

$$\Omega^T = \min\left(\frac{1}{\Omega_1^T},\ \frac{1}{\Omega_2^T}\right),$$

where the Ω_i^T are given by

$$\Omega_i^T = \left|\frac{(\Omega_i^s)^2 - \Omega_1^c\,\Omega_2^c}{\Omega_i^s\,(\Omega_2^c - \Omega_1^c)}\right|, \quad i = 1,2.$$

By Ω_1^c and Ω_2^c we mean the two cutoff frequencies, and by Ω_1^s and Ω_2^s we mean the two boundaries between stopband and transition band.

These rules for determination of Ω^T and, hence, N, for filter types other than low-pass, follow directly from (2a)-(2d), the transformations that convert a low-pass filter to one of another type. Note, however, that in the bandpass and band-reject cases, the computed N will equal the order of the basic low-pass structure before transformation to the filter of proper type. The actual order of the final filter will be 2N.

In the above we have chosen the minimum order filter which meets the user's requirements. It should be mentioned, however, that the user may find another order more appealing in certain special cases. For example, for band-reject Chebyshev or elliptic filters that are derived from even order low-pass structures, we find that the magnitude characteristic is asymptotic to -R dB in the passband, and is actually near -R dB over a large percentage of the frequency range. Thus, a filter derived from an odd order low-pass structure may be preferred here, since its frequency response is asymptotic to 0 dB, the ideal value in the passband.

DETERMINATION OF S-PLANE POLE-ZERO PATTERN

A lengthy but straightforward task is the computation of the pole-zero pattern of a continuous unity bandwidth low-pass filter of order N. The techniques are adequately reviewed by Gold and Rader* and will not be repeated here. However, it is not necessary to keep track of all poles and zeros since they always occur in complex conjugate pairs, and it is not necessary to compute any zeros, except in the elliptic case, since they are always at infinity in the s-plane.

Also, the complete elliptic integral of the first kind, which is needed in the elliptic design problem, is given exactly by

$$K(k) = \int_0^1 \frac{1}{\left[(1 - t^2)(1 - k^2 t^2)\right]^{1/2}} ,$$

but in the program is approximated by the formula†

$$K(m) \doteq \left(a_0 + a_1 m_1 + \cdots + a_4 m_1^4\right) + \left(b_0 + b_1 m_1 + \cdots + b_4 m_1^4\right) \ln \left(\frac{1}{m_1}\right),$$

*B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Company, Inc., 1969, pp. 48-97.

†M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions, Dover Publications, 1964, pp. 567-607.

where

$a_0 = 1.38629436112$ $b_0 = 0.5$

$a_1 = 0.09666344259$ $b_1 = 0.12498593597$

$a_2 = 0.03590092383$ $b_2 = 0.06880248576$

$a_3 = 0.03742563713$ $b_3 = 0.03328355346$

$a_4 = 0.01451196212$ $b_4 = 0.00441787012$

and

$$m = k^2$$

$$m_1 = 1 - m.$$

The maximum error in this formula is 2×10^{-8} when $0 \leq m < 1$, in which range m will always be.

Also, other approximations* are used in the calculation of elliptic poles and zeros. Iterative and/or series calculations can be used to replace these approximations but the results obtained with them were deemed sufficiently accurate.

CALCULATION OF POLE-ZERO PATTERN OF THE DESIRED DIGITAL FILTER

Once the s-plane poles and zeros of a unity bandwidth low-pass continuous filter of proper order and kind have been determined, it is possible to map these to the poles and zeros of the desired digital filter, which may be of any type with any cutoff(s).

Although the calculations vary by type of filter desired, the method is basically the same, and will be shown for the bandpass case only. If H(s) is the transfer function of a unity bandwidth low-pass continuous filter of order N, then

$$H_A(s) = H\left[\frac{s^2 + \Omega_1^c \Omega_2^c}{s\left(\Omega_2^c - \Omega_1^c\right)}\right]$$

*Gold and Rader, op. cit.

is the transfer function of a continuous bandpass filter of order 2N with cutoffs Ω_1^c and Ω_2^c.

Now applying the bilinear transformation (3), we see that

$$H_D(z) = H_A\left[\frac{z-1}{z+1}\right] = H\left[\frac{\left(\frac{z-1}{z+1}\right)^2 + \Omega_1^c\,\Omega_2^c}{\left(\frac{z-1}{z+1}\right)\left(\Omega_2^c - \Omega_1^c\right)}\right]$$

is the transfer function of a bandpass digital filter of order 2N with cutoffs at

$$\omega_1^c = \frac{2}{T}\tan^{-1}\left(\Omega_1^c\right)$$

and

$$\omega_2^c = \frac{2}{T}\tan^{-1}\left(\Omega_2^c\right).$$

Now if we solve the equation

$$s = \frac{\left(\frac{z-1}{z+1}\right)^2 + \Omega_1^c\,\Omega_2^c}{\left(\frac{z-1}{z+1}\right)\left(\Omega_2^c - \Omega_1^c\right)}$$

for z in terms of s, we obtain

$$z = \frac{(p-1) \pm (s^2 d^2 - 4p)^{1/2}}{sd - 1 - p},$$

where

$$p = \Omega_1^c\,\Omega_2^c$$

$$d = \Omega_2^c\,\Omega_1^c\,.$$

We can merely substitute the previously computed s-plane poles and zeros into this expression to obtain the poles and zeros of the desired digital bandpass transfer function $H_D(z)$. Note that each point in the s-plane is mapped to two z-plane points, and thus the number of poles and zeros doubles. Actually, the computations are not carried out for the zeros unless we are working with an elliptic filter since $H_D(z)$ will always have N zeros at $z = 1$ and N zeros at $z = -1$. This follows from the fact that the s-plane zeros are all at infinity.

When we have calculated the z-plane poles and zeros, we have determined the desired transfer function up to some constant factor. In the next section we will determine that constant.

CALCULATION OF THE CONSTANT MULTIPLIER

Thus far we have determined the transfer function, $H_D(z)$, of the desired filter as

$$H_D(z) = \frac{K \prod_{i=1}^{N} (z - z_i)}{\prod_{i=1}^{N} (z - p_i)} ,$$

where

K = constant multiplier to be determined

N = order of filter

z_i = calculated zeros

p_i = calculated poles.

Knowing the value of $H_D(z)$ for any z ($z \neq z_i$ or p_i) will enable us to determine K uniquely.

Let E be defined by

$$E = \begin{cases} 1 & \text{, for N odd or Butterworth} \\ 1/\sqrt{1+\epsilon^2} & \text{, otherwise} \end{cases} .$$

Now for any low-pass continuous filter of the kinds considered in this report, we have

$$H_A(s)\Big|_{s=0} = E$$

and, in particular, this holds true for the unity bandwidth low-pass filter constructed previously. Thus, if we knew the point z_0 in the z-plane to which the point $s = 0$ is mapped, we could merely set

$$H_D(z)\Big|_{z=z_0} = E$$

and solve for the quantity K.

From the theory of the previous section it follows that the proper values for z_0 are

low-pass $\quad z_o = 1$

high-pass $\quad z_o = -1$

bandpass $\quad z_o = \frac{1-p}{1-p} \pm j\frac{2\sqrt{p}}{1+p}$, where $p = \Omega_1^c \Omega_2^c$, as before,

band-reject $\quad z_o = \pm 1.$

For example, in the bandpass case we know that any point s_0 in the s-plane will be mapped to the point

$$z_o = \frac{(p-1) \pm (s_o^2 d^2 - 4p)^{1/2}}{s_o d - 1 - p} .$$

Hence, the point $s_o = 0$ is mapped to

$$z_o = \frac{1 - p}{1 + p} \pm j \frac{2\sqrt{p}}{1 + p} .$$

Note that in the bandpass and band-reject cases we have a choice of two values for z_o, since each is an image of the point $s = 0$.

No matter what type filter we are designing, it follows that we can determine the quantity K by

$$K = \frac{E \prod_{i=1}^{N} (z_o - p_i)}{\prod_{i=1}^{N} (z_o - z_i)} .$$

In computing this expression, computational techniques make it unnecessary to work with complex quantities.

MANIPULATION INTO CASCADE FORM

By combining the complex conjugate poles and zeros of the transfer function $H_D(z)$ into second order factors with real coefficients, we can write the transfer function as a product of terms of the form

$$\frac{z^2 + b_1 z + b_2}{z^2 + c_1 z + c_2} .$$

There will also be one factor of the form

$$\frac{z + b}{z + c} ,$$

if the order of the filter is odd, corresponding to the real pole. However, when one actually implements a filter, some recursive scheme is needed to calculate the present filtered data point. With the transfer function in the above form we can immediately write a set of difference equations that describes the filter in the time domain, and this set will correspond to a cascade realization. In the general case where the transfer function is

$$H(z) = \frac{Y(z)}{X(z)} = K\left(\frac{z + b_1}{z + c_1}\right)\left(\frac{z^2 + b_2\, z + b_3}{z^2 + c_2\, z + c_3}\right)\ldots\left(\frac{z^2 + b_{N-1}\, z + b_N}{z^2 + c_{N-1}\, z + c_N}\right),$$

the following set of difference equations equivalently describes the filter:

$$y_1(n) = x(n) - c_1 y_1(n-1)$$

$$y_2(n) = y_1(n) + b_1 y_1(n-1)$$

$$y_3(n) = y_2(n) - c_2 y_3(n-1) - c_3 y_3(n-1)$$

$$y_4(n) = y_3(n) + b_2 y_3(n-1) + b_3 y_3(n-2)$$

$$y_5(n) = y_4(n) - c_4 y_5(n-1) - c_5 y_5(n-2)$$

$$y_6(n) = y_5(n) + b_4 y_5(n-1) + b_5 y_5(n-2)$$

$$\vdots$$

$$y_N(n) = y_{N-1}(n) - c_{N-1}\, y_N(n-1) - c_N\, y_N(n-2)$$

$$y(n) = K(y_N(n) + b_{N-1}\, y_N(n-1) + b_N\, y_N\, (n-2)) \;,$$

where $x(n)$ and $y(n)$ are the filter inputs and outputs, respectively, and $y_1(n)$, $y_1(n)$, ..., $y_N(n)$ are intermediate values determined by the above equations.

The above filter could also be described by a single difference equation of order N, which would be an entirely equivalent time domain representation, if all coefficients could be represented exactly. However, due to the necessary quantization of coefficients, this so-called "direct form" realization is not recommended since it is much more sensitive to coefficient accuracy than the corresponding cascade implementation (5).

THE MAGNITUDE AND PHASE OF THE FREQUENCY RESPONSE

To obtain the frequency response of a digital filter, we must evaluate the transfer function on the unit circle of the complex z-plane. In particular, $|H_D(e^{j\omega T})|$, the magnitude of the frequency response, is usually of primary interest.

Thus far we have $H_D(z)$ expressed as a product of terms of the form

$$\frac{z^2 + b_1 z + b_2}{z^2 + c_1 z + c_2} .$$

The magnitude and phase of this term for $z = e^{j\omega T}$ can easily be expressed as a trigonometric function of ωT and can be simply evaluated for any ω. The product of the magnitudes of all such terms and the constant multiplier is then the magnitude of the frequency response. Similarly, the sum of the phases of all such terms gives the phase of the frequency response. The possibility of a single term of the form

$$\frac{z + b}{z + c}$$

poses no additional problem. In the program we compute the magnitude and phase as above, convert the magnitude to decibels and phase to degrees, and then provide plots in the frequency range desired.

THE UNIT SAMPLE RESPONSE

The rate of decay of the unit sample response (i.e., the response of the filter to a pulse of unit height at time zero) of a digital filter is often of interest to the designer. For this reason a plot is provided if requested.

Since the transfer function, $H_D(z)$, can be viewed as the Z-transform of $h(n)$, the unit sample response, we can write

$$h(n) = Z^{-1} \left[H_D(z)\right] ,$$

or

$$h(n) = \frac{1}{2\pi j} \oint_C H_D(z)\, z^{n-1}\, dz,$$

where C is any contour in the complex plane enclosing all poles of the integrand. Replacing $H_D(z)$ by its pole-zero factorization, we obtain

$$h(n) = \frac{K}{2\pi j} \oint_C \frac{\prod_{i=1}^{N} (z - z_i)}{\prod_{i=1}^{N} (z - p_i)} z^{n-1}\, dz.$$

Since the p_i, $i = 1,2,\ldots,N$ will always be simple poles, the residue at a specific p_j of the integrand is

$$\frac{\prod_{i=1}^{N} (p_j - z_i)}{\prod_{\substack{i=1 \\ i \neq j}}^{N} (p_j - p_i)} p_j^{n-1} .$$

For the special case $n = 0$ there is also a pole at $z = 0$, but the complications introduced by the additional pole need not be considered since we know by examining (5) that $h(0)$ must equal K. Defining

$$\beta_j = \frac{\prod_{i=1}^{N} (p_j - z_i)}{\prod_{\substack{i=1 \\ i \neq j}}^{N} (p_j - p_i)},$$

we can then easily compute the unit sample response as

$$h(n) = \begin{cases} K \sum_{i=1}^{N} \beta_i \, p_i^{\,n-1} , & n \geq 1 \\ K , & n=0 \end{cases} .$$

Although the unit sample response is a discrete function of time, the plot produced by the program is a continuous function for ease in viewing.

SUMMARY

Butterworth, Chebyshev, and elliptic digital filters are realizable networks that approximate physically unrealizable filters having rectangular magnitude characteristics. Each of these filters come arbitrarily close to the ideal response as the order of the approximating network becomes large. Thus, the question of the order necessary to meet a set of specifications is usually critical. The computer program described in this report computes the minimum order filter that meets the particular requirements of the user, and then proceeds to design that filter. The program also produces plots of the magnitude and phase of the frequency response and the unit sample response of the filter. The reader need not fully understand the theory presented in this report in order to use the program. Finally, for those unfamiliar with digital signal processing techniques, the instruction on how the filter transfer function can be implemented as a set of difference equations should be useful.

Appendix

LISTING OF THE COMPUTER PROGRAM

```
C     CHEBYSHEV BUTTERWORTH OR ELLIPTIC DIGITAL FILTER DESIGN PROGRAM ***S1889***
C     WRITTEN BY J.J.WOLCIN
C
C     THIS PROGRAM WILL DESIGN ANY LOW-PASS, HIGH-PASS, BANDPASS OR
C     BAND-REJECT FILTER BY BILINEAR TRANSFORMATION TECHNIQUES
C
C     LAST UPDATED 6 JUNE 1973
C
C
C * * * * * * * * * *INSTRUCTIONS FOR INPUT SPECIFICATIONS  * * * * * * * * * * *
C
C     VARIABLE   TYPE     COLUMNS     DESCRIPTION
C     --------   ----     -------     -----------
C
C     SRATE      REAL     1-8         THE NUMBER OF DATA SAMPLES PER
C                                     TIME UNIT (USUALLY SECONDS)
C
C     FCLOW      REAL     9-16        THE LOWER CUTOFF FREQUENCY
C                                     (IN CYCLES/TIME UNIT)
C
C     FCHIGH     REAL     17-24       THE UPPER CUTOFF FREQUENCY
C                                     (IN CYCLES/TIME UNIT)
C
C     RIPPLE     REAL     25-30       THE DESIRED PASSBAND RIPPLE (IN DB)
C
C     FSLOW      REAL     31-38       THE LOWER FREQUENCY MARKING THE BOUNDARY
C                                     BETWEEN STOPBAND AND TRANSITION BAND
C                                     (IN CYCLES/TIME UNIT)
C
C     FSHIGH     REAL     39-46       THE HIGHER FREQUENCY MARKING THE BOUNDARY
C                                     BETWEEN STOPBAND AND TRANSITION BAND
C                                     (IN CYCLES/TIME UNIT)
C
C     STPLVL     REAL     47-50       THE MINIMAL STOPBAND ATTENUATION(IN DB)
C
C     IKIND      INTEGER  51          THE KIND OF FILTER DESIRED I.E.
C                                      1 CHEBYSHEV
C                                      2 BUTTERWORTH
C                                      3 ELLIPTIC
C
C     ITYPE      INTEGER  52          TYPE OF FILTER DESIRED I.E.
C                                      1 LOW-PASS
C                                      2 HIGH-PASS
C                                      3 BANDPASS
C                                      4 BAND-REJECT
C
C     NPOLES     INTEGER  53-54       SET=0  FOR CALCULATION OF REQUIRED ORDER
C                                     OTHERWISE SET=ORDER DESIRED FOR BASIC
C                                     LOW-PASS STRUCTURE BEFORE TRANSFORMATION
C
C     IPLOT      INTEGER  56          SET=1 FOR PLOT OF MAGNITUDE OF H(Z)
C                                     SET=0 FOR NO PLOT
C
```

```
C     NPTS        INTEGER  57-60        NO. OF PTS IN UNIT SAMPLE RESPONSE
C                                       (DEFAULT VALUE = 100)
C
C     FLOW        REAL     61-70        LOW END FREQUENCY OF PLOT
C
C     FHIGH       REAL     71-80        HIGH END FREQUENCY OF PLOT
C
C
C     NOTE---IF NPOLES.NE.0 (I.E. THE ORDER IS SPECIFIED), FSLOW,FSHIGH AND
C            STPLVL ARE NOT USED (UNLESS ELLIPTIC) SINCE THE GIVEN ORDER
C            DETERMINES THESE QUANTITIES
C
C            IF LOW-PASS OR HIGH-PASS, FCHIGH AND FSHIGH ARE NOT USED
C
C            IF BUTTERWORTH IS DESIRED, RIPPLE IS NOT USED
C
C            THE QUANTITIES RIPPLE AND STPLVL SHOULD ALWAYS BE GIVEN .GT. 0
C
C            REPEAT DATA CARD OF THIS TYPE FOR EACH FILTER TO BE DESIGNED
C            PLACE A BLANK DATA CARD LAST
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C
C
      DOUBLE PRECISION COEF(20,2),POLES(10,2),POLEZ(40,2),ZEROS(10,2)
      DOUBLE PRECISION COEFN(20,2),CNST,ZEROZ(40,2)
      DOUBLE PRECISION EPS2,PI,WCUT1,WCUT2,EXPNT,ANGLE,DANGLE,WSTOP1,
     1WSTOP2,OMEGA,OMEGA1,A,B,C,D,E,F,G,PROD,DIFF,RMAG,RANGLE,THETA,
     2SN1,SN3,CN1,CN3,DN1,DN3,SN,Q
      DIMENSION X(1000),Y(1000),PHI(1000),Z(200)
      DIMENSION W(21)
      DIMENSION  H1(1000),H2(1000)
      COMPLEX ZPOLES(40),ZZEROS(40),ZTEMP(40),BETA(40)
      COMPLEX PTEMP1,PTEMP2,ZTEMP2
C
      ELLPTC(T)=1.38629436112D0+T*(.09666344259D0+T*(.03590092383D0+
     1T*(.03742563713D0+T*.01451196212D0)))+DLOG(1.0/T)*(.5+T*
     2(.12498593597D0+T*(.06880248576D0+T*(.03328355346D0+T*.00441787012
     3D0))))
C
C
C ---------------------------------------------------------------------
C            READ INPUT SPECIFICATIONS OF FILTER TO BE DESIGNED
C ---------------------------------------------------------------------
C
    5 READ(5,6) SRATE,FCLOW,FCHIGH,RIPPLE,FSLOW,FSHIGH,STPLVL,IKIND,
     1ITYPE,NPOLES,IPLOT,NPTS,FLOW,FHIGH
    6 FORMAT(3F8.0,F6.0,2F8.0,F4.0,2I1,2I2,I4,2F10.0)
      IF(ITYPE.EQ.0) GO TO 1000
C
C ---------------------------------------------------------------------
C            WRITE OUT INPUT SPECIFICATIONS ON PRINTER
C ---------------------------------------------------------------------
C
      WRITE(6,7) SRATE,FCLOW,FCHIGH,RIPPLE,FSLOW,FSHIGH,STPLVL,IKIND,
     1ITYPE,NPOLES,IPLOT
    7 FORMAT(1H1/18X,'INPUT SPECIFICATIONS'///18X,'SRATE = ',F8.2//23X,
     1'FCLOW = ',F8.2//28X,'FCHIGH = ',F8.2//33X,'RIPPLE = ',F5.3//38X,
```

```
     2'FSLOW = ',F8.2//43X,'FSHIGH = ',F8.2//48X,'STPLVL = ',F6.2//53X,
     3'IKIND = ',I1//58X,'ITYPE = ',I1//63X,'NPOLES = ',I2//68X,'IPLOT =
     4 ',I1)
      FNYQ=SRATE/2.
      IF(FCLOW.GT.FNYQ.OR.FCHIGH.GT.FNYQ.OR.FSHIGH.GT.FNYQ.OR.FSLOW.GT.
     1FNYQ)  GO TO 8
      IF(NPOLES.EQ.0.AND.STPLVL.LT.0.)  GO TO 8
      IF(IKIND.NE.2.AND.RIPPLE.LT.0.)  GO TO 8
      IF(NPOLES.NE.0.AND.IKIND.EQ.3)  GO TO 8
      IF(ITYPE.EQ.1.AND.FCLOW.GT.FSLOW)  GO TO 8
      IF(ITYPE.EQ.2.AND.FCLOW.LT.FSLOW)  GO TO 8
      IF(ITYPE.EQ.3.AND.(FCLOW.LT.FSLOW.OR.FCHIGH.GT.FSHIGH))  GO TO 8
      IF(ITYPE.EQ.4.AND.(FCLOW.GT.FSLOW.OR.FCHIGH.LT.FSHIGH))  GO TO 8
      GO TO 11
    8 WRITE(4,9)
    9 FORMAT(18X,'DESIGN IMPOSSIBLE---INCONSISTENT SPECIFICATIONS')
      GO TO 5
   11 CONTINUE
C
C
      PI=3.141592653589793240D0
      WCUT1=DTAN(PI*FCLOW/SRATE)
      IF(ITYPE.LT.3)  GO TO 12
      WCUT2=DTAN(PI*FCHIGH/SRATE)
      DIFF=WCUT2-WCUT1
      PROD=WCUT1*WCUT2
   12 IF(IKIND.NE.2) EPS2=(10.0D0)**(RIPPLE/10.0D0)-1.0D0
      IF(NPOLES.NE.0.AND.IKIND.NE.3)  GO TO 50
C
C -----------------------------------------------------------------------
C             DETERMINE MINIMUM ORDER FILTER WHICH MEETS INPUT
C             SPECIFICATIONS UNLESS ORDER IS SPECIFIED
C -----------------------------------------------------------------------
C
      WSTOP1=DTAN(PI*FSLOW/SRATE)
      GO TO (13,14,15,15),ITYPE
C
   13 OMEGA=WSTOP1/WCUT1
      GO TO 18
C
   14 OMEGA=WCUT1/WSTOP1
      GO TO 18
C
   15 WSTOP2=DTAN(PI*FSHIGH/SRATE)
      IF(ITYPE.EQ.4)  GO TO 16
      OMEGA=DABS((WSTOP1**2-PROD)/(WSTOP1*DIFF))
      OMEGA1=DABS((WSTOP2**2-PROD)/(WSTOP2*DIFF))
      GO TO 17
C
   16 OMEGA=DABS((WSTOP1*DIFF)/(PROD-WSTOP1**2))
      OMEGA1=DABS((WSTOP2*DIFF)/(PROD-WSTOP2**2))
   17 IF(OMEGA1.LT.OMEGA)    OMEGA=OMEGA1
C
   18 GO TO (19,30,40),IKIND
C
   19 VN=SQRT((10.**(STPLVL/10.)-1.)/EPS2)
      V(1)=1.0
      V(2)=OMEGA
```

```
      IF(OMEGA.GT.VN)  GO TO 25
      DO 20 I=3,21
      V(I)=2.0*OMEGA*V(I-1)-V(I-2)
      IF(V(I).LT.VN)  GO TO 20
      NPOLES=I-1
      GO TO 50
   20 CONTINUE
C
   21 WRITE(4,22(
   22 FORMAT(40X,'THE SPECIFICATIONS REQUIRE A FILTER OF ORDER .GT. 20')
      GO TO 5
C
   25 NPOLES=I
      GO TO 50
C
   30 NPOLES=ALOG10(10.**(STPLVL/10.)-1.)/(2.*ALOG10(OMEGA))+1
      IF(NPOLES.GT.20)  GO TO 21
      GO TO 50
C
   40 A=10.**(STPLVL/20.)
      B=EPS2/(A*A-1.)
      C=1./(OMEGA*OMEGA)
      D=ELLPIC(1.-B)
      E=ELLPIC(1.-C)
      F=ELLPIC(B)
      G=ELLPIC(C)
      NPOLES=1+(F*E)/(D*G)
      IF(NPOLES.GT.20)  GO TO 21
      G=DEXP(-NPOLES*PI*G/E)
      D=1.
      DO 42 I=1,100
      B=G**(I*I+I)
      IF(B.LT.1.D-30)  GO TO 45
      D=D+B
   42 CONTINUE
   45 F=0.
      DO 47 I=1,100
      B=G**(I*I)
      IF(B.LT.1.D-30)  GO TO 48
      F=F+B
   47 CONTINUE
   48 B=(2.*G**0.25*D/(1.+2.*F))**4
      D=ELLPIC(1.-B)
      F=ELLPIC(B)
C
   50 N=NPOLES*((ITYPF+1)/2)
      WRITE(4,51)  N
   51 FORMAT(18X,'THE MINIMUM ORDER FILTER WHICH MEETS THESE SPECIFICATI
     1ONS IS ORDER ',I2,
C
C -----------------------------------------------------------------------
C             CALCULATE THE SECOND QUADRANT POLE POSITIONS IN THE S-PLANE
C             OF A UNITY BANDWIDTH LOW PASS FILTER OF THE REQUIRED ORDER
C             REALIZING THAT THEIR COMPLEX CONJUGATES ARE ALSO POLES
C -----------------------------------------------------------------------
C
   60 IODD=MOD(NPOLES,2)
      K=(NPOLES+1)/2
```

```
      EXPNT=1.0D0/NPOLES
      IF(IKIND.EQ.3) GO TO 90
C
      DANGLE=PI*EXPNT
      A=1.0
      B=1.0
      IF(IKIND.EQ.2) GO TO 70
      C=(DSQRT(1.0D0+1.0D0/EPS2)+1.0D0/DSQRT(EPS2))**EXPNT
      A=.5D0*(C-1.0D0/C)
      B=.5D0*(C+1.0D0/C)
   70 ANGLE=0.0
      IF(IODD.EQ.0)  ANGLE=DANGLE/2.0D0
      DO 80 I=1,K
      POLES(I,1)=-1.0D0*A*DCOS(ANGLE)
      POLES(I,2)=B*DSIN(ANGLE)
      ANGLE=ANGLE+DANGLE
   80 CONTINUE
      GO TO 110
C
   90  A=-(G/F)*DLOG((DSQRT(1.+EPS2)+1.)/DSQRT(EPS2))
      F=DSQRT(1.-C)
      INDIC=1
C
   92 G=ELLPIC(F*F)
      B=ELLPIC(1.-F*F)
      Q=DEXP(-PI*G/B)
      ANGLE=PI*A/(2.*B)
      SN=0.
      DO 94 J=0,1000
      D=(Q**(J+.5))/(1.-Q**(2*J+1))
      SN=SN+D*DSIN((2*J+1.)*ANGLE)
      IF(D.LT.1.D-30)  GO TO 95
   94 CONTINUE
   95 SN=(2.*PI*SN)/(F*G)
C
      GO TO (96,98),INDIC
C
   96 SN1=SN
      CN1=DSQRT(1.-SN1*SN1)
      DN1=DSQRT(1.-(F*SN1)**2)
      INDIC=2
      A=(1-IODD)*E*EXPNT
      F=DSQRT(C)
      I=0
      GO TO 92
C
   98 SN3=SN
      CN3=DSQRT(1.-SN3*SN3)
      DN3=DSQRT(1.-C*SN3*SN3)
      I=1+I
      POLES(I,1)=(SN1*CN1*CN3*DN3)/(1.-SN1*SN1*DN3*DN3)
      POLES(I,2)=(SN3*DN1)/(1.-SN1*SN1*DN3*DN3)
      IF(I.EQ.1.AND.IODD.EQ.1)  GO TO 100
      ZEROS(I,1)=0.
      ZEROS(I,2)=1./(DSQRT(C)*SN3)
  100 A=A+(2.*E*EXPNT)
      IF(I.LT.K)  GO TO 92
C
```

```
C -----------------------------------------------------------------------
C          MAP THE S-PLANE POLES TO THE Z-PLANE USING THE BILINEAR
C          TRANSFORMATION TAKING THE CUTOFF FREQUENCY AND TYPE OF
C          FILTER INTO ACCOUNT
C          ALSO, IF ELLIPTIC IS DESIRED CARRY OUT THE SAME PROCEDURE
C          FOR THE ZEROS
C -----------------------------------------------------------------------
C
  110 IF(ITYPE.GT.2)  GO TO 170
      J=1
      IF(ITYPE.EQ.1)  GO TO 140
      IF(IODD.EQ.0)  GO TO 120
      POLEZ(1,1)=(POLES(1,1)+WCUT1)/(POLES(1,1)-WCUT1)
      POLEZ(1,2)=0.0
      IF(NPOLES.EQ.1)  GO TO 300
      J=2
  120 L=J
C
      DO 130 I=L,K
      A=POLES(I,1)+WCUT1
      B=POLES(I,2)
      D=POLES(I,1)-WCUT1
      POLEZ(J,1)=(A*D+B*B)/(D*D+B*B)
      POLEZ(J,2)=(B*D-A*B)/(D*D+B*B)
      POLEZ(J+1,1)=POLEZ(J,1)
      POLEZ(J+1,2)=-1.0D0*POLEZ(J,2)
      IF(IKIND.NE.3)  GO TO 125
      ZEROZ(J,1)=(ZEROS(I,2)*ZEROS(I,2)-WCUT1*WCUT1)/(ZEROS(I,2)*
     1ZEROS(I,2)+WCUT1*WCUT1)
      ZEROZ(J,2)=2.*ZEROS(I,2)*WCUT1/(ZEROS(I,2)*ZEROS(I,2)+WCUT1*WCUT1)
      ZEROZ(J+1,1)=ZEROZ(J,1)
      ZEROZ(J+1,2)=-ZEROZ(J,2)
  125 J=J+2
  130 CONTINUE
      GO TO 300
C
  140 C=1.0D0/WCUT1
      IF(IODD.EQ.0)  GO TO 150
      POLEZ(1,1)=(POLES(1,1)+C)/(C-POLES(1,1))
      POLEZ(1,2)=0.0
      IF(NPOLES.EQ.1)  GO TO 300
      J=2
  150 L=J
C
      DO 160 I=L,K
      A=POLES(I,1)+C
      B=POLES(I,2)
      D=C-POLES(I,1)
      POLEZ(J,1)=(A*D-B*B)/(D*D+B*B)
      POLEZ(J,2)=(B*D+A*B)/(D*D+B*B)
      POLEZ(J+1,1)=POLEZ(J,1)
      POLEZ(J+1,2)=-1.0D0*POLEZ(J,2)
      IF(IKIND.NE.3)  GO TO 155
      ZEROZ(J,1)=(C*C-ZEROS(I,2)*ZEROS(I,2))/(C*C+ZEROS(I,2)*ZEROS(I,2))
      ZEROZ(J,2)=2.*ZEROS(I,2)*C/(C*C+ZEROS(I,2)*ZEROS(I,2))
      ZEROZ(J+1,1)=ZEROZ(J,1)
      ZEROZ(J+1,2)=-ZEROZ(J,2)
  155 J=J+2
```

```
  160 CONTINUE
      GO TO 300
C
  170 IF(ITYPE.EQ.3)  GO TO 190
      DO 180 I=1,K
      A=POLES(I,1)
      POLES(I,1)=POLES(I,1)/(POLES(I,1)**2+POLES(I,2)**2)
      POLES(I,2)=-POLES(I,2)/(A**2+POLES(I,2)**2)
      IF(IKIND.NE.3)  GO TO 180
      IF(I.EQ.1.AND.IODD.EQ.1) GO TO 180
      A=ZEROS(I,1)
      ZEROS(I,1)=ZEROS(I,1)/(ZEROS(I,1)**2+ZEROS(I,2)**2)
      ZEROS(I,2)=-ZEROS(I,2)/(A**2+ZEROS(I,2)**2)
  180 CONTINUE
C
  190 INDIC=0
  195 CONTINUE
      J=1
C     .
      DO 200 I=1,K
      IF(I.EQ.2.AND.IODD.EQ.1)  J=J+2
      IF((I.GT.2).OR.(I.EQ.2.AND.IODD.EQ.0))  J=J+4
      A=DIFF*DIFF*(POLES(I,1)**2-POLES(I,2)**2)-4.*PROD
      B=2.*POLES(I,1)*POLES(I,2)*DIFF*DIFF
      RMAG=(A*A+B*B)**0.25
      RANGLE=DATAN2(B,A)
      A=RMAG*DSIN(RANGLE/2.)
      B=RMAG*DCOS(RANGLE/2.)
      C=1.+PROD-DIFF*POLES(I,1)
      D=DIFF*POLES(I,2)
      IF(INDIC.EQ.1)  GO TO 198
      INCRMT=2
      IF(I.EQ.1.AND.IODD.EQ.1)  INCRMT=1
      POLEZ(J,1)=((1.-PROD+B)*C-D*A)/(C*C+D*D)
      POLEZ(J,2)=(D*(1.-PROD+B)+C*A)/(C*C+D*D)
      POLEZ(J+INCRMT,1)=((1.-PROD-B)*C+D*A)/(C*C+D*D)
      POLEZ(J+INCRMT,2)=(D*(1.-PROD-B)-C*A)/(C*C+D*D)
      IF(I.EQ.1.AND.IODD.EQ.1)  GO TO 200
      POLEZ(J+1,1)=POLEZ(J,1)
      POLEZ(J+1,2)=-POLEZ(J,2)
      POLEZ(J+3,1)=POLEZ(J+2,1)
      POLEZ(J+3,2)=-POLEZ(J+2,2)
      GO TO 200
  198 IF(I.EQ.1.AND.IODD.EQ.1)  GO TO 200
      ZEROZ(J,1)=((1.-PROD+B)*C-D*A)/(C*C+D*D)
      ZEROZ(J,2)=(D*(1.-PROD+B)+C*A)/(C*C+D*D)
      ZEROZ(J+2,1)=((1.-PROD-B)*C+D*A)/(C*C+D*D)
      ZEROZ(J+2,2)=(D*(1.-PROD-B)-C*A)/(C*C+D*D)
      ZEROZ(J+1,1)=ZEROZ(J,1)
      ZEROZ(J+1,2)=-ZEROZ(J,2)
      ZEROZ(J+3,1)=ZEROZ(J+2,1)
      ZEROZ(J+3,2)=-ZEROZ(J+2,2)
  200 CONTINUE
C
      IF(IKIND.NE.3.OR.INDIC.EQ.1)   GO TO 300
C
      DO 210 I=1,K
      POLES(I,1)=ZEROS(I,1)
```

```
      POLES(I,2)=ZEROS(I,2)
  210 CONTINUE
      INDIC=1
      GO TO 195
C
  300 CONTINUE
      NZ=N
      IF(IKIND.EQ.3.AND.IODD.EQ.0)  GO TO 410
      IF(IKIND.EQ.3)  NZ=(ITYPE+1)/2
C
C ------------------------------------------------------------------------
C           FILL UP THE ZEROS ARRAY WITH THE Z-PLANE ZEROS (UNLESS ELLIPTIC)
C           IN WHICH CASE THE ZEROS HAVE BEEN CALCULATED.
C ------------------------------------------------------------------------
C
      DO 400 I=1,NZ
      GO TO (310,320,330,340),ITYPE
  310 ZEROZ(I,1)=-1.
      ZEROZ(I,2)=0.
      GO TO 400
  320 ZEROZ(I,1)=1.
      ZEROZ(I,2)=0.
      GO TO 400
  330 ZEROZ(I,1)=(-1.)**(I+1)
      ZEROZ(I,2)=0.
      GO TO 400
  340 ZEROZ(I,1)=(1.-PROD)/(1.+PROD)
      ZEROZ(I,2)=(-1.)**(I+1)*2.*DSQRT(PROD)/(1.+PROD)
  400 CONTINUE
C
C ------------------------------------------------------------------------
C           DETERMINE THE COEFFICIENTS IN THE DENOMINATOR POLYNOMIAL
C           OF THE TRANSFER FUNCTION BY COMBINING THE COMPLEX CONJUGATE
C           POLE PAIRS INTO REAL SECOND ORDER FACTORS. IF THE ORDER IS
C           ODD THERE WILL ALSO BE ONE TERM OF ORDER ONE, CORRESPONDING
C           TO THE REAL POLE
C           FOLLOW THE SAME PROCEDURE TO DETERMINE THE NUMERATOR COEFFICIENTS
C ------------------------------------------------------------------------
C
  410 J=1
      IF(ITYPE.GT.2.OR.IODD.EQ.0)  GO TO 450
      COEF(1,1)=-POLEZ(1,1)
      COEF(1,2)=0.
      COEFN(1,1)=-ZEROZ(1,1)
      COEFN(1,2)=0.
      IF (NPOLES.NE.1)  GO TO 440
      E=1.0D0
      IF(ITYPE.EQ.1)  C=(1.+COEF(1,1))/(1.+COEFN(1,1))
      IF(ITYPE.EQ.2)  C=(1.-COEF(1,1))/(1.-COEFN(1,1))
      GO TO 500
  440 J=2
  450 L=J
      M=N
      IF(ITYPE.GT.2)  M=NPOLES
C
      DO 460 I=L,M
      COEF(I,1)=-POLEZ(J,1)-POLEZ(J+1,1)
      COEF(I,2)=POLEZ(J,1)*POLEZ(J+1,1)+POLEZ(J,2)*POLEZ(J,2)
```

```
      COEFN(I,1)=-ZEROZ(J,1)-ZEROZ(J+1,1)
      COEFN(I,2)=ZEROZ(J,1)*ZEROZ(J+1,1)+ZEROZ(J,2)*ZEROZ(J,2)
      J=J+2
  460 CONTINUE
C
C ----------------------------------------------------------------------
C           NOW THAT WE HAVE THE POLE-ZERO PATTERN OF THE DESIRED TRANSFER
C           FUNCTION, H(Z), WE HAVE DETERMINED H(Z) UP TO SOME CONSTANT.
C           NEXT DETERMINE THAT CONSTANT.
C ----------------------------------------------------------------------
C
      E=1.0D0
      IF(IODD.EQ.0.AND.IKIND.NE.2)  E=1.0D0/DSQRT(1.0D0+EPS2)
      C=1.0D0
      GO TO (470,480,490,470),ITYPE
C
  470 DO 475 I=1,M
      A=1.+COEFN(I,1)+COEFN(I,2)
      B=1.0D0+COEF(I,1)+COEF(I,2)
      C=C*(B/A)
  475 CONTINUE
      GO TO 500
C
  480 DO 485 I=1,K
      A=1.-COEFN(I,1)+COEFN(I,2)
      B=1.0D0-COEF(I,1)+COEF(I,2)
      C=C*(B/A)
  485 CONTINUE
      GO TO 500
C
  490 THETA=DATAN2(2.*DSQRT(PROD),1.-PROD)
      F=DCOS(2.*THETA)
      G=DCOS(THETA)
      O=DSIN(2.*THETA)
      Q=DSIN(THETA)
      DO 495 I=1,NPOLES
      B=DSQRT((F+COEF(I,1)*G+COEF(I,2))**2+(O+COEF(I,1)*Q)**2)
      A=DSQRT((F+COEFN(I,1)*G+COEFN(I,2))**2+(O+COEFN(I,1)*Q)**2)
      C=C*(B/A)
  495 CONTINUE
C
  500 CNST=E*C
C
C ----------------------------------------------------------------------
C           CALCULATIONS ARE COMPLETE. WRITE OUT THE TRANSFER FUNCTION
C           AS A RATIO OF POLYNOMIALS WHICH CAN BE IMMEDIATELY
C           IMPLEMENTED IN 'CASCADE' FORM
C ----------------------------------------------------------------------
C
      WRITE(4,510)
  510 FORMAT (////18X,'THE TRANSFER FUNCTION,H(Z), OF THE DESIRED FILTER
     1 CAN BE EXPRESSED',/18X,'IN THE FOLLOWING FORM')
      IF(IODD.EQ.1.AND.ITYPE.LT.3)  WRITE(4,520)
      IF(IODD.EQ.0.OR.ITYPE.GT.2)  WRITE(4,530)
  520 FORMAT(//50X,'Z+B(1)',7X,'Z**2+B(2)*Z+B(3)',13X,'Z**2+B(N-1)*Z+B(N
     1)',/35X,'H(Z) = CNST *  ------   *   ----------------   * ... *
     2-----------------',/50X,'Z+C(1)',7X,'Z**2+C(2)*Z+C(3)',13X,'Z**2+
     3C(N-1)*Z+C(N)')
```

```
530 FORMAT(//50X,'Z**2+B(1)*Z+B(2)',7X,'Z**2+B(3) Z+B(4)',13X,'Z**2+B(
   1N-1)*Z+B(N)',/35X,'H(Z) = CNST *  ----------------   *   ---------
   2--------   * ... *    -------------------',/50X,'Z**2+C(1)*Z+C(2)',7X
   3,'Z**2+C(3)*Z+C(4)',13X,'Z**2+C(N-1)*Z+C(N)')
    WRITE(4,540)  CNST
540 FORMAT (//18X,'WHERE THE COEFFICIENTS ARE GIVEN BY',//35X,'CNST =
   1',D23.15)
    INDIC=0
    IF(IODD.EQ.1 .AND.ITYPE.LT.3)   INDIC=1
    DO 560 I=1,N
    J1=(I+INDIC+1)/2
    J2=2-MOD(I+INDIC,2)-1/(I-INDIC+1)
    WRITE(4,550)  I,COEF(J1,J2), I ,COEFN(J1,J2)
555 FORMAT (18X,'C(',I2,')=',D23.15,10X,'B(',I2,')=',D23.15)
560 CONTINUE
    WRITE(4,570)
570 FORMAT(///18X,'THE TRANSFER FUNCTION IS EXPRESSED IN THE ABOVE FOR
   1M SO THAT'/18X,'THE COEFFICIENTS CAN BE USED AS GIVEN IN A CASCADE
   2 IMPLEMENTATION OF THE FILTER')
    IF(IPLOT.EQ.0)  GO TO 5
C
C ----------------------------------------------------------------------
C           PLOT THE MAGNITUDE AND PHASE OF H(Z) AS Z VARIES ON THE
C           UNIT CIRCLE IN THE COMPLEX PLANE
C ----------------------------------------------------------------------
C
    WLOW=2.*FLOW*PI/SRATE
    WHIGH=2.*FHIGH*PI/SRATE
    RANGE=WHIGH-WLOW
    DX=(FHIGH-FLOW)/10.0
    TWODX=2.*DX
    L=1
C
    DO 600 I=1,1000
    X(I)=WLOW+FLOAT(I)*RANGE/1000.0
    SINX=SIN(X(I))
    SIN2X=SIN(2.*X(I))
    COSX=COS(X(I))
    COS2X=COS(2.*X(I))
    RR=0.
    D=1.0
    IF(IODD.EQ.0.OR.ITYPE.GT.2)  GO TO 580
    A=(COSX+COEFN(1,1))**2+SINX**2
    B=(COSX+COEF(1,1))**2+SINX**2
    D=DSQRT(A/B)
    R1=COEFN(1,1)+COSX
    R2=COEF(1,1)+COSX
    IF(ABS(SINX).LT.1.E-10.AND.ABS(R1).LT.1.E-10)  R1=1.E-10
    IF(ABS(SINX).LT.1.E-10.AND.ABS(R2).LT.1.E-10)  R2=1.E-10
    RR=ATAN2(SINX,R1)-ATAN2(SINX,R2)
    L=2
    IF(NPOLES.EQ.1)  GO TO 595
580 CONTINUE
    DO 590 J=L,M
    C=(COS2X+COEF(J,1)*COSX+COEF(J,2))**2+(SIN2X+COEF(J,1)*SINX)**2
    A=(COS2X+COEFN(J,1)*COSX+COEFN(J,2))**2+(SIN2X+COEFN(J,1)*SINX)**2
    B=B*DSQRT(A/C)
    R1=SIN2X+COEFN(J,1)*SINX
```

```
      R2=COS2X+COEFN(J,1)*COSX+COEFN(J,2)
      R3=SIN2X+COEF(J,1)*SINX
      R4=COS2X+COEF(J,1)*COSX+COEF(J,2)
      IF(ABS(R1).LT.1.E-10.AND.ABS(R2).LT.1.E-10)  R2=1.E-10
      IF(ABS(R3).LT.1.E-10.AND.ABS(R4).LT.1.E-10)  R4=1.E-10
      RR=RR+ATAN2(R1,R2)-ATAN2(R3,R4)
  590 CONTINUE
  595 IF(B.LT.0.001/CNST)  GO TO 598
      Y(I)=20.*DLOG10(CNST*B)
      IF(Y(I).LT.-50.0)  Y(I)=-50.0
      GO TO 599
  598 Y(I)=-50.
  599 X(I)=X(I)*SRATE/(2.0*PI)
      NPI=IFIX(0.5*RR/PI)
      PHI(I)=RR-2.*NPI*PI
      IF(PHI(I).LT.0.)  PHI(I)=PHI(I)+2.*PI
      PHI(I)=180.*PHI(I)/PI
  600 CONTINUE
C
      CALL MODESG(Z,0)
      CALL OBJCTG(Z,500.0,500.0,3050.0,3050,0)
      CALL SUBJEG(Z,FLOW,-50.0,FHIGH,0,0)
      CALL GRIDG(Z,DX,5.0,0.0,0.0)
      CALL LABELG(Z,0,TWODX,0,8,2)
      CALL LABELG(Z,1,10.0,0,5,1)
      CALL LINESG(Z,1000,X,Y)
      CALL TITLEG(Z,29,'FREQUENCY IN CYCLES/TIME UNIT',8,'DECIBELS',61,
     1'PLOT OF THE MAGNITUDE OF THE FREQUENCY RESPONSE OF THE FILTER')
C
      CALL OBJCTG(Z,3050.,500.,4095.,3050.)
      CALL SUBJEG(Z,0.,0.,1045.,100.)
      CALL LEGNDG(Z,200.,90.,21,'DESCRIPTION OF FILTER')
      GO TO (701,702,711,712),ITYPE
  701 CALL LEGNDG(Z,200.,80.,8,'LOW PASS')
      GO TO 705
  702 CALL LEGNDG(Z,200.,80.,9,'HIGH PASS')
  705 CALL LEGNDG(Z,200.,60.,8,'CUTOFF= ')
      CALL NUMBRG(Z,450.,60.,7,2,FCLOW)
      IF(IKIND.EQ.2)  GO TO 750
      CALL LEGNDG(Z,200.,55.,8,'RIPPLE= ')
      CALL NUMBRG(Z,450.,55.,4,2,RIPPLE)
      GO TO 750
  711 CALL LEGNDG(Z,200.,80.,8,'BANDPASS')
      GO TO 715
  712 CALL LEGNDG(Z,200.,80.,11,'BAND REJECT')
  715 CALL LEGNDG(Z,200.,60.,9,'CUTOFF1= ')
      CALL LEGNDG(Z,200.,55.,9,'CUTOFF2= ')
      CALL NUMBRG(Z,480.,60.,7,2,FCLOW)
      CALL NUMBRG(Z,480.,55.,7,2,FCHIGH)
      IF(IKIND.EQ.2)  GO TO 750
      CALL LEGNDG(Z,200.,50.,8,'RIPPLE= ')
      CALL NUMBRG(Z,450.,50.,4,2,RIPPLE)
  750 GO TO (801,802,803),IKIND
  801 CALL LEGNDG(Z,200.,75.,9,'CHEBYSHEV')
      GO TO 850
  802 CALL LEGNDG(Z,200.,75.,11,'BUTTERWORTH')
      GO TO 850
  803 CALL LEGNDG(Z,200.,75.,8,'ELLIPTIC')
```

```
850 CALL LEGNDG(Z,200.,70.,7,'SRATE= ')
    CALL NUMBRG(Z,400.,70.,7.1,SRATE)
    CALL LEGNDG(Z,200.,65.,7,'ORDER= ')
    CALL NUMBRG(Z,400.,65.,2,N)
    CALL PAGEG(Z,0,1,1)
    CALL OBJCTG(Z,500.,500.,4000.,2800.)
    CALL SUBJEG(Z,FLO*,0.,FHIGH,360.)
    CALL GRIDG(Z,DX,60.,0.,0.)
    CALL LABELG(Z,0,TAUDX,0,8.2)
    CALL LABELG(Z,1,60.,0,5.I)
    CALL LINESG(Z,I000,X,PHI)
    CALL TITLEG(Z,29,'FREQUENCY IN CYCLES/TIME UNIT',7,'DEGREES',57,'P
   1LOT OF THE PHASE OF THE FREQUENCY RESPONSE OF THE FILTER')
    CALL PAGEG(Z,0,1,1)
C
C ---------------------------------------------------------------------
C         COMPUTE AND PLOT THE UNIT SAMPLE RESPONSE AS THE INVERSE
C         Z-TRANSFORM OF H(Z)
C ---------------------------------------------------------------------
C
    DO 900 I=1,N
    R1=POLEZ(I,1)
    R2=POLEZ(I,2)
    R3=ZEROZ(I,1)
    R4=ZEROZ(I,2)
    ZPOLES(I)=CMPLX(R1,R2)
    ZTEMP(I)=1.
900 ZZEROS(I)=CMPLX(R3,R4)

    DO 920 I=1,N
    PTEMPI=CMPLX(1.,0.)
    PTEMP2=CMPLX(1.,0.)
    DO 910 J=1,N
    PTEMP1=PTEMP1*(ZPOLES(I)-ZZEROS(J))
    IF(J.EQ.I)  GO TO 910
    PTEMP2=PTEMP2*(ZPOLES(I)-ZPOLES(J))
910 CONTINUE
    BETA(I)=PTEMP1/PTEMP2
920 CONTINUE

    H1(1)=0.
    H2(1)=CNST
    HMAX=CNST
    IF(NPTS.EQ.0)  NPTS=100
    PTS=FLOAT(NPTS)
    DO 950 I=2,NPTS
    ZTEMP2=0.
    DO 930 J=1,N
    ZTEMP2=ZTEMP2+BETA(J)*ZTEMP(J)
    ZTEMP(J)=ZTEMP(J)*ZPOLES(J)
930 CONTINUE

    H1(I)=FLOAT(I-1)
    H2(I)=CNST*REAL(ZTEMP2)
    IF(ABS(H2(I)).GT.HMAX)  HMAX=ABS(H2(I))
950 CONTINUE

    HMAX=1.05*HMAX
```

```
      IF(HMAX.LT.0.002)  HMAX=0.002
      HMIN=-HMAX
      DY=HMAX/2.5
      DX=NPTS/10

      CALL OBJCTG(Z,500.,500.,4000.,2800.)
      CALL SUBJCG(Z,0.,HMIN,PTS,HMAX)
      CALL [illegible](Z,PTS,HMAX,0.,0.)
      CALL LABELG(Z,0,DX,0,4)
      CALL LABELG(Z,1,DY,0,6,3)
      CALL LINESG(Z,NPTS,H1,H2)
      CALL TITLEG(Z,15,'RESPONSE NUMBER',12,'OUTPUT VALUE',38,'PLOT OF U
     1NIT SAMPLE RESPONSE OF FILTER')
      CALL PAGEG(Z,0,1,1)
      GO TO 5
C
 1000 CONTINUE
      CALL EXITG(Z)
      END

   END OF COMPILATION:       NO  DIAGNOSTICS.
S1069            SYMBOLIC.                      23 JUL 73  08:41:14  0  03246670  14   711  (DELETED)
S1069   CODE     RELOCATABLE                    23 JUL 73  08:41:14  1  03272232  48     1  (DELETED)
                                                                     0  03272312  14   347


   @AN   XQT S1069                                                          14:48:34. 50
```