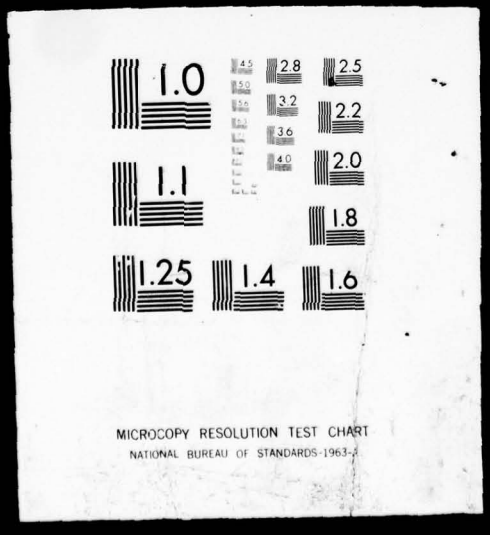
AD-A059 868 COLORADO STATE UNIV FORT COLLINS F/G 9/1
DEVELOPMENT OF IMPROVED DESIGN METHODS FOR DIGITAL FILTERING SY--ETC(U)
NOV 77 T A BRUBAKER F33615-75-C-1138
UNCLASSIFIED AFAL-TR-77-207 NL

1 OF 3
AD A059868



LEVEL II

AFAL-TR-77-207

DEVELOPMENT OF IMPROVED DESIGN METHODS FOR DIGITAL FILTERING SYSTEMS

Colorado State University
Fort Collins, CO 80523

November 1, 1977

TECHNICAL REPORT - AFAL-TR-77-207

Final Report for Period - October 1, 1975-September 30, 1977

Approved for public release; distribution unlimited.

AIR FORCE AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (IO) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Dale L. Harper
DALE L. HARPER
Project Engineer

Pearl Hoskins
PEARL HOSKINS
Supervisor

FOR THE COMMANDER

Raymond E Siferd
RAYMOND E. SIFERD, Lt Col, USAF
Chief, System Avionics Division
Air Force Avionics Laboratory

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

(19) REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (18) AFAL-TR-77-207	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* (6) Development of Improved Design Methods for Digital Filtering Systems.		5. TYPE OF REPORT & PERIOD COVERED (9) Final Report, Oct. 1, 1975-Sept. 30, 1977
		6. PERFORMING ORG. REPORT NUMBER 1 Oct 75-30 Sep 77
7. AUTHOR(s) (10) Thomas A. Brubaker		8. CONTRACT OR GRANT NUMBER(s) (15) F33615-75-C-1138
9. PERFORMING ORGANIZATION NAME AND ADDRESS Colorado State University Fort Collins, CO 80523		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS (11) 1 Air Force Avionics Laboratory Wright Patterson Air Force Base, OH 45433		12. REPORT DATE Nov. 1, 1977
		13. NUMBER OF PAGES 191
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* (12) 199 p.		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)*

Digital Filters -- Word Length -- Fault Detection
Sampling Rate -- Interactive Software

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

Design methods for digital filters used in discrete time control systems are described. The sampling interval is first used as a design parameter. Then new design methods for digital control algorithms that minimize multiplication requirements are developed. Interactive software for aiding the design and implementation of digital control algorithms is described.

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

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

78 0883 00

TABLE OF CONTENTS

PREFACE

This report describes work on new design methods for filters used in discrete data control systems. Design methods are developed first for sampling rates to minimize the bit requirements for each filter coefficient then new design methods for digital filters that minimize the need for digital multiplication are described. Interactive software for aiding design and implementation of digital filters was written and is described in the report.

INTRODUCTION

This report describes work on new design and implementation methods for filters used in discrete-data control systems. Specifically, the following tasks were undertaken:

1. The development of design methods that use the sampling interval as a design parameter to minimize the bits required to represent each filter coefficient.

2. The development of new design methods for digital control algorithms to minimize the need for digital multiplication.

3. The development of interactive software to aid in the design and implementation of digital control algorithms.

4. A method of fault analysis for digital control algorithms.

Each task is discussed with details given via copies of each report generated during the contract period. These reports are provided as appendices.

1. DESIGN USING THE SAMPLING RATE AS A DESIGN PARAMETER

Both digital and analog filter synthesis generally involve tradeoffs. For instance, low ordered filters may have either sharp rolloff or flat passbands but not both. High order filters can have excellent frequency response characteristics but involve a large number of components or multiplications, both of which increase errors. In sampling time synthesis there are tradeoffs as well. Exact coefficients can be easily found for quite a few first order filters but the sampling time which yields such coefficients causes the filters to have serious magnitude errors due to aliasing. On the other hand, a sampling time which is very short will cause the filter frequency response to be very

sensitive to coefficient quantization. The tradeoff between aliasing errors and coefficient quantization errors is to be kept in mind in synthesizing sampling times for bilinear z-transform filters. In fact, the tradeoff considerations are an important step in the synthesis procedure.

First Order Filters

The technique for synthesizing sampling time is essentially the same for both first and second order filters. However, because there are only two coefficients in the first order filters and because frequency independent bounds can be found for the first order filters, the first order case is developed first.

For the design a realistic approach is to make the magnitude (or phase) response errors as samll as possible while retaining a short enough sampling interval to avoid aliasing errors. One means of finding coefficients which will give small error is to generate a number of sets of coefficients and then find the truncated and bounded values for each. Next, take the difference between the designed coefficients and the respective quantized values and determine the maximum error for each set of coefficients. The set with the smallest maximum magnitude error is then the set of coefficients to use unless the sampling interval associated with that set is too long to meet the aliasing specifications. A maximum bound on frequency error could be given and the sampling times and coefficients which give a magnitude error less than the bound would be considered. If the frequency response error criterion is not met, then it is

necessary to generate more coefficients using different sampling times than used previously and repeat the procedure above. If the magnitude response error criterion is not satisfied after several hundred sampling times have been tried it would be necessary to use a longer word length to realize the coefficients.

While the procedure above seems quite long, it is possible to combine all of the steps of the process into an interactive computer program. The block diagram of such a program is shown in Figure 1.1.

The first block of Figure 1.1 asks for input of the analog filter coefficients, the word length desired, the maximum absolute value for the magnitude response error, $\Delta|H|$, and the maximum number of iterations to be done before it is decided that a longer word length is necessary. Block 2 initializes the sampling time for a certain pass. On the first pass the sampling time will be set to $.01 \cdot t_{max}$ as an initial value where t_{max} is the maximum value the sampling interval can be, set to avoid aliasing errors. Blocks 3 and 4 are self-explanatory, where block 3 uses equations for a digital filter found from an analog filter via the bilinear z-transform to generate the digital coefficients. The fifth and sixth blocks are similar to each other. In each, the difference is found between the designed (infinite precision) digital coefficients and the quantized coefficients. The differences are found for the rounded coefficients and then $\Delta|H|$ is derived by using the magnitude of the desired and actual filters. The same calculations are also done for the truncated coefficients. Then $\Delta|H|$ of the rounded values is compared to the $\Delta|H|$ of the truncated values and the smallest of those two $\Delta|H|$'s is chosen

for that sampling interval. If the $\Delta|H|$ chosen is less than the maximum allowable magnitude error, specified in the first block, then the sampling time and the associated digital coefficients are printed out along with the type of quantization to be used. Also, if the $\Delta|H|$ chosen on a particular iteration is smaller than any chosen on any previous iteration then it is stored along with its corresponding sampling interval and the previously stored values are discarded.

Whether or not $\Delta|H|$ is less than the previous smallest value, the sampling time is increased by $.01 \cdot t_{max}$. If the sampling time is then less than or equal to t_{max} a new set of coefficients, differences, and magnitude response errors is generated. If the sampling time is greater than t_{max} then the procedures of blocks 15 through 18 are executed. If there is at least one $\Delta|H|$ of those considered which meets the error criterion then the sampling time which gave the smallest $\Delta|H|$ is output along with the $\Delta|H|$. Otherwise a test is done to see if the maximum number of iterations have been run through. If they have then it is advisable to increase the word length and run through the iterations again. If the maximum number of iterations has not been reached then a new set of sampling times should be tried. A search can be made around the immediate area of the sampling time which gave the lowest $\Delta|H|$, using a smaller sampling time increment for the new iterations. Another possibility is to merely offset the new sampling times from those of the previous pass by a certain amount, for example $.001 \cdot t_{max}$. A FORTRAN program which realizes the block diagram of Figure 1.1

has been used to illustrate the procedure.

Second Order Filters

The block diagram in Figure 1.1 for first order filter synthesis serves as well for second order filters. If a second order low pass section is being designed, a possible evaluation technique would be to evaluate $\Delta|H|$ at $\Omega = 0$ and at the 3-db point for every sampling time considered. For a bandpass structure $\Delta|H|$ could be calculated at the 3-db points and the pole frequency. In the example programs the user is allowed to choose what radian frequencies the magnitude error is to be evaluated at, and how many frequencies are to be evaluated.

The block diagram in Figure 1.1 allows every sampling interval and corresponding set of coefficients which have a $\Delta|H|$ smaller than the maximum error bound to be printed out. The reason for this is very simple. In some cases a certain sampling rate will be more desirable than another even if it does not give the minimum magnitude response error. Such a case occurs when the clock rate for the filter is limited to a certain range of values. By printing all sampling times which have magnitude errors within the desired bound, there is more design freedom permitted.

So far the discussion has centered about magnitude response design. However, a bilinear z-transform digital filter will not generally have the same phase response as the corresponding analog filter. However, digital filters do have phase response

and it is sometimes desirable to retain the accuracy of that response. It is possible to make the phase response accurate in the same manner that was used for the magnitude response. In fact, the block diagram of Figure 1.1 can be used for the phase by replacing $|H|$ by Θ in the diagram. A longer flow chart could be developed which allows filters with accurate phase and magnitude to be designed.

In terms of design limitations, the primary considerations are those of aliasing and processor (or component) speed. When deriving a program to find a sampling time which results in small frequency response error, it is necessary to put an upper bound on the length of the sampling interval to reduce aliasing errors. If the sampling rate is too low, a filter will be aliased to the point where it no longer performs its designed task. To reduce aliasing errors, it could be required that the sampling rate be at least ten times the highest pole frequency. There are various such rules of thumb aimed at avoiding aliasing errors and which ever is appropriate should be used.

Digital hardware is limited in terms of clock rates it can operate at. Some computers can perform an instruction in a matter of nano-seconds while others require several micro-seconds to do the same instruction. Similarly, discrete digital components such as multipliers, adders, and shift registers are limited in speed. When designing a digital filter it is important to realize the constraint of digital hardware speed on sampling time. If a computer program is used to realize a filter, the program may be ten, twenty, or even over a hundred instructions

long. Often, large filter structures are better realized using discrete digital components which are dedicated to the filtering because the discrete components have an advantage in speed over a computer program. Whatever method of realization is used though, the design should not allow the clock rate of the filter exceed the speed of the structure used to realize it.

Example Designs

A commonly used filter design is the maximally flat, or Butterworth, filter. The low pass Butterworth filter has the property that the filter magnitude response is as flat as possible at $\omega = 0$. For the present example, a fifth order Butterworth low pass digital filter is synthesized using the method described in the first part of this chapter. The analog transfer function, H(s), is given by

$$H(s) = \frac{1}{s+1} \frac{1}{s^2+.618s+1.0000} \frac{1}{s^2+1.618s+1.0000} \qquad (1.1)$$

so that H(s) has unity gain and unity bandwidth. The example demonstrates the use of both the first and second order synthesis programs. The bilinear z-transform allows the transfer function to be broken up into first and second order cascade sections so there is no partial fraction expansion to worry about. For the example, the assumptions are that an eight bit word and a magnitude error of less than 10^{-5} are desired.

The first order section, $H_1(s)$, of H(s) is given by

$$H_1(s) = \frac{1}{s+1} = \frac{c}{s+a} \qquad (1.2)$$

Figure 1.2 shows the sequence of interactive inputs to the program. The first three inputs are self explanatory. After the word length was input, the program used 100 sampling times between 0 and the maximum sampling time allowed. An appropriate sampling time was not found and the graph of Figure 1.3 resulted. The graphs are not meant to be an absolute means of measuring the error of the filter but merely a way of determining whether to proceed or to try another word length or error bound. The next input given in Figure 1.2 was a 1(one) to indicate that on the next iteration the same range of sampling times was to be used but the sampling times would be offset from the previous set of sampling times. The amount that the second set was offset was one-tenth of the spacing of the first set of sampling times. Therefore, there was a sampling time selected between each of the first sampling times. Again the error bound was not met, resulting in Figure 1.4 which is very similar to Figure 1.3. Rather than continue on the same track, it was felt that a better approach would be to "blow up" the region around the sampling interval which gave the minimum error. 100 sampling times were chosen between the two sampling intervals which were adjacent to the point which gave the least error. The input of 2 in Figure 1.2 resulted in the expansion about the minimum point and the graph of Figure 1.5. The error bound was still not met but there seemed to be promise so another expansion was done. Figure 1.7 shows that the error bound was finally satisfied by three sampling times. The output lists the three sampling times which allowed the error criterion to be met and the corresponding coefficients

and the type of coefficient quantization to be used on each set of coefficients. Figure 1.6 shows the graph of the final expansion. The listing on Figure 1.7 prints, as a final set of values, the minimum magnitude error found and the sampling time which gave the minimum error. If there had been no sampling times which caused the error bound to be satisfied on that last round, then it probably would have been necessary to use a longer word length or accept a slightly relaxed error bound. Sometimes it is possible to do enough passes to find a sampling time which gives low enough error but in order to set the sampling time it would require an infinitesimal adjustment and so the sampling time is not practically realizable. Even with programmable clocks, the adjustment is usually only down to about 10^{-7} so adjustments below that level are not possible.

The two second order sections, $H_2(s)$ and $H_3(s)$, are given by

$$H_2(s) = \frac{1}{s^2+.618s+1.0000} \tag{1.3}$$

$$H_3(s) = \frac{1}{s^2+1.618s+1.0000} \tag{1.4}$$

Since there is no one general frequency which results in a maximum for the partial derivatives in the expansion ΔH [1,2], then several frequencies should be chosen to check those partial derivatives. In the examples, four frequencies were chosen for each section. Three were in the passband and one was in the

transition band of each filter section. The graphs of Figures 1.8 and 1.9 show the relative errors of the filters against the sampling times for $H_2(z)$ and $H_3(z)$, respectively. Some of the errors are so large that most of the error points appear to be very small but, in reality, only a few of the plotted points resulted in filters which satisfied the error bounds.

The second order filter errors behave much differently than the first order filter errors. The first order errors tend to decrease as the sampling interval gets longer, while the second order errors tend to increase. Also, the second order errors, with a few exceptions as noted on the graphs, are generally much lower than the first order errors. Therefore, it seems that the word length restrictions on a filter would come from the filter's first order section or sections. The design technique, then, should state that the first order sections should be synthesized before the second order sections in order to get a good bound on the word length requirements.

After considerable effort with more examples this approach appeared to be somewhat limited, in fact, the method does not generally work. To simplify the coefficient problem and to totally eliminate the multiplier, a different procedure was tried as shown in the next section.

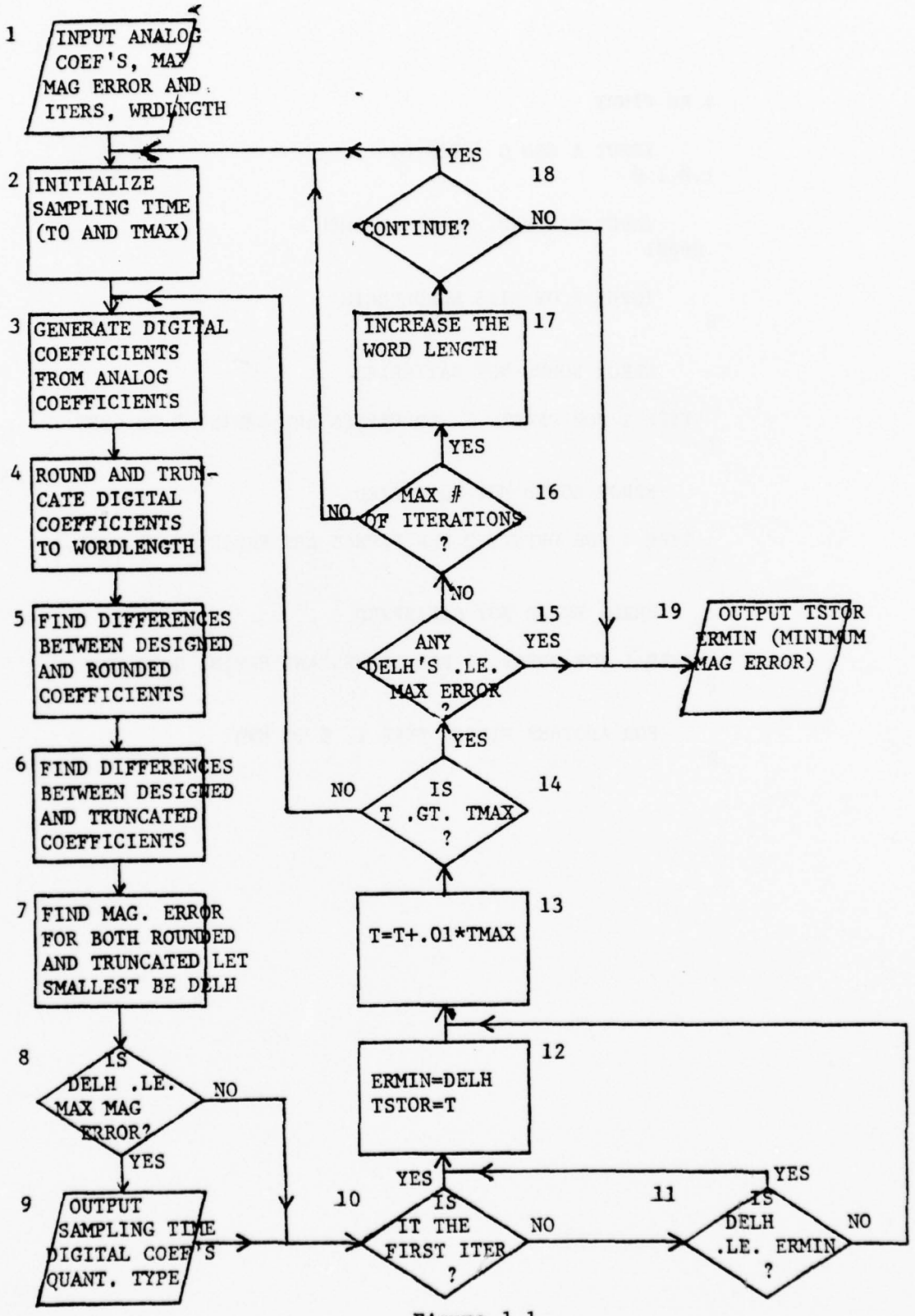

1
INPUT ANALOG COEF'S, MAX MAG ERROR AND ITERS, WRDLNGTH
2
INITIALIZE SAMPLING TIME (TO AND TMAX)
3
GENERATE DIGITAL COEFFICIENTS FROM ANALOG COEFFICIENTS
4
ROUND AND TRUNCATE DIGITAL COEFFICIENTS TO WORDLENGTH
5
FIND DIFFERENCES BETWEEN DESIGNED AND ROUNDED COEFFICIENTS
6
FIND DIFFERENCES BETWEEN DESIGNED AND TRUNCATED COEFFICIENTS
7
FIND MAG. ERROR FOR BOTH ROUNDED AND TRUNCATED LET SMALLEST BE DELH
8
IS DELH .LE. MAX MAG ERROR?
NO
YES
9
OUTPUT SAMPLING TIME DIGITAL COEF'S QUANT. TYPE
10
IS IT THE FIRST ITER ?
YES
NO
11
IS DELH .LE. ERMIN ?
YES
NO
12
ERMIN=DELH TSTOR=T
13
T=T+.01*TMAX
14
IS T .GT. TMAX ?
YES
NO
15
ANY DELH'S .LE. MAX ERROR ?
YES
NO
16
MAX # OF ITERATIONS ?
YES
NO
17
INCREASE THE WORD LENGTH
18
CONTINUE?
YES
NO
19
OUTPUT TSTOR ERMIN (MINIMUM MAG ERROR)

Figure 1.1

```
$ RU FINDT

     INPUT A AND C (2F1Ø.6)
1.Ø,1.Ø

     INPUT MAX MAG. ERROR ALLOWED
.ØØØØ1

     INPUT # OF BITS WORDLENGTH
8

     ERROR BOUND NOT SATISFIED

  TYPE 1 FOR OFFST, 2 FOR EXPNSN ABT ERMIN, 3 TO CONT.
1

     ERROR BOUND NOT SATISFIED

  TYPE 1 FOR OFFST, 2 FOR EXPNSN ABT ERMIN, 3 TO CONT.
2

     ERROR BOUND NOT SATISFIED

  TYPE 1 FOR OFFST, 2 FOR EXPNSN ABT ERMIN, 3 TO CONT.
2

     FOR ANOTHER FILTER TYPE 1, Ø TO END
Ø
```

Figure 1.2

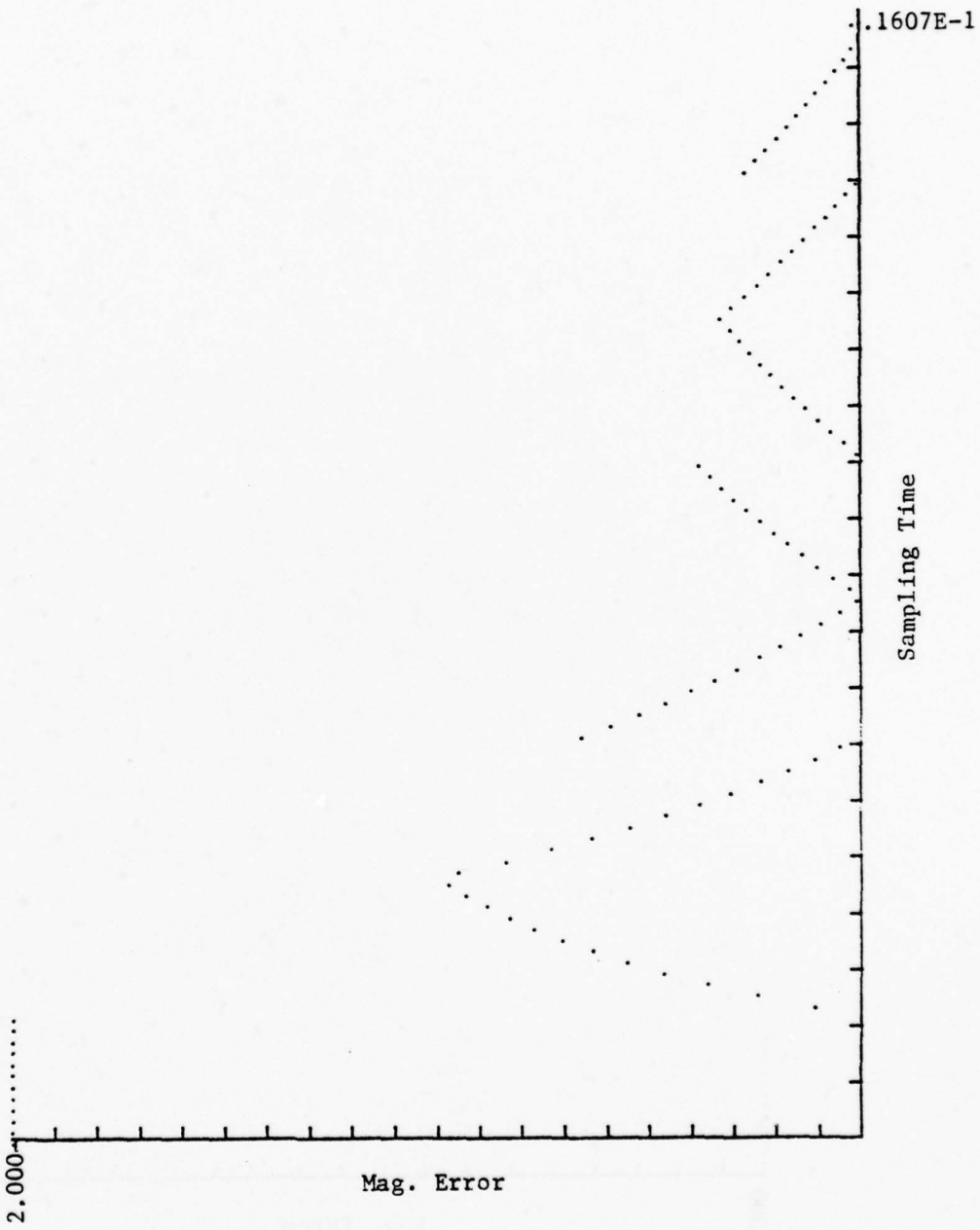


Figure 1.3

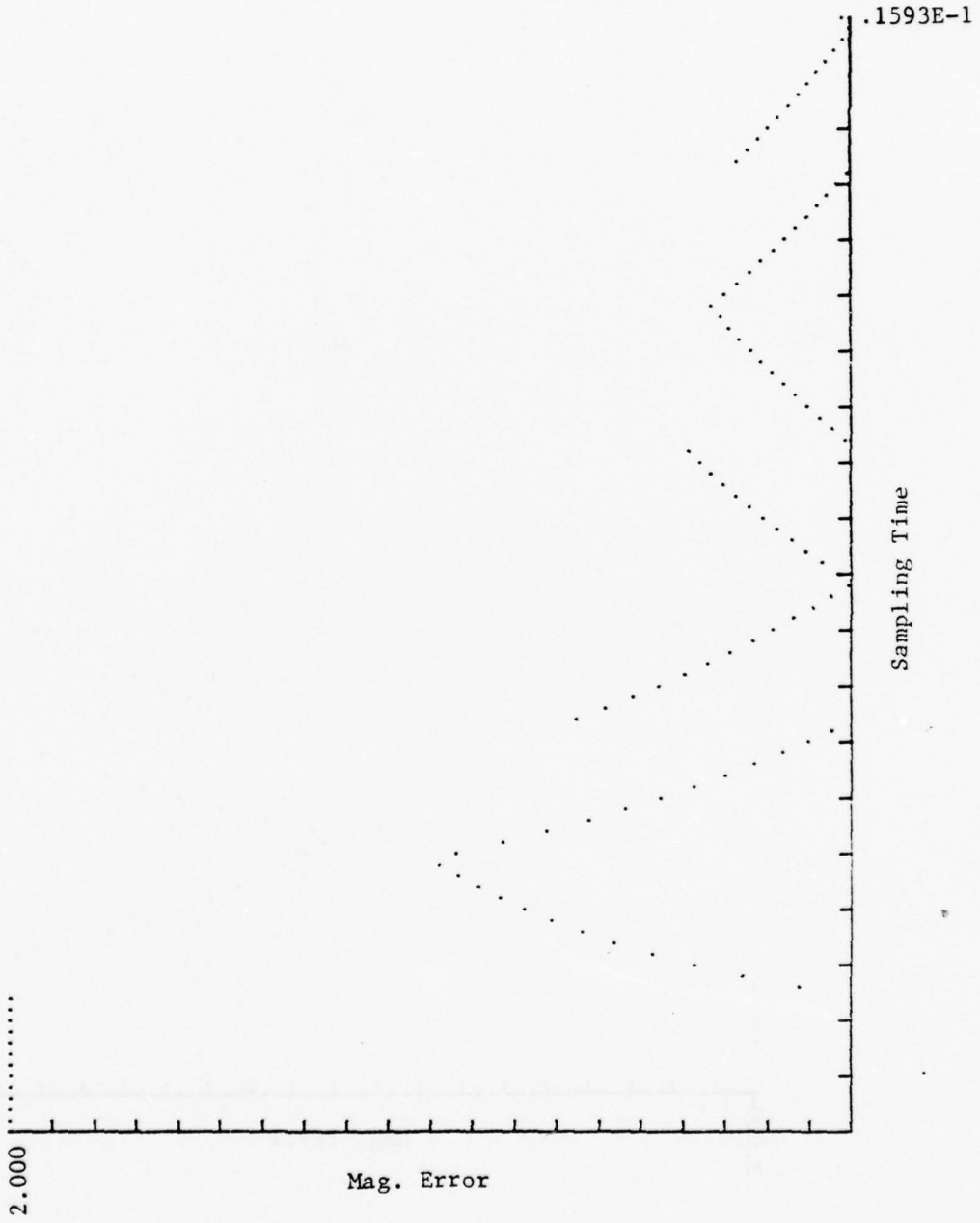


Figure 1.4

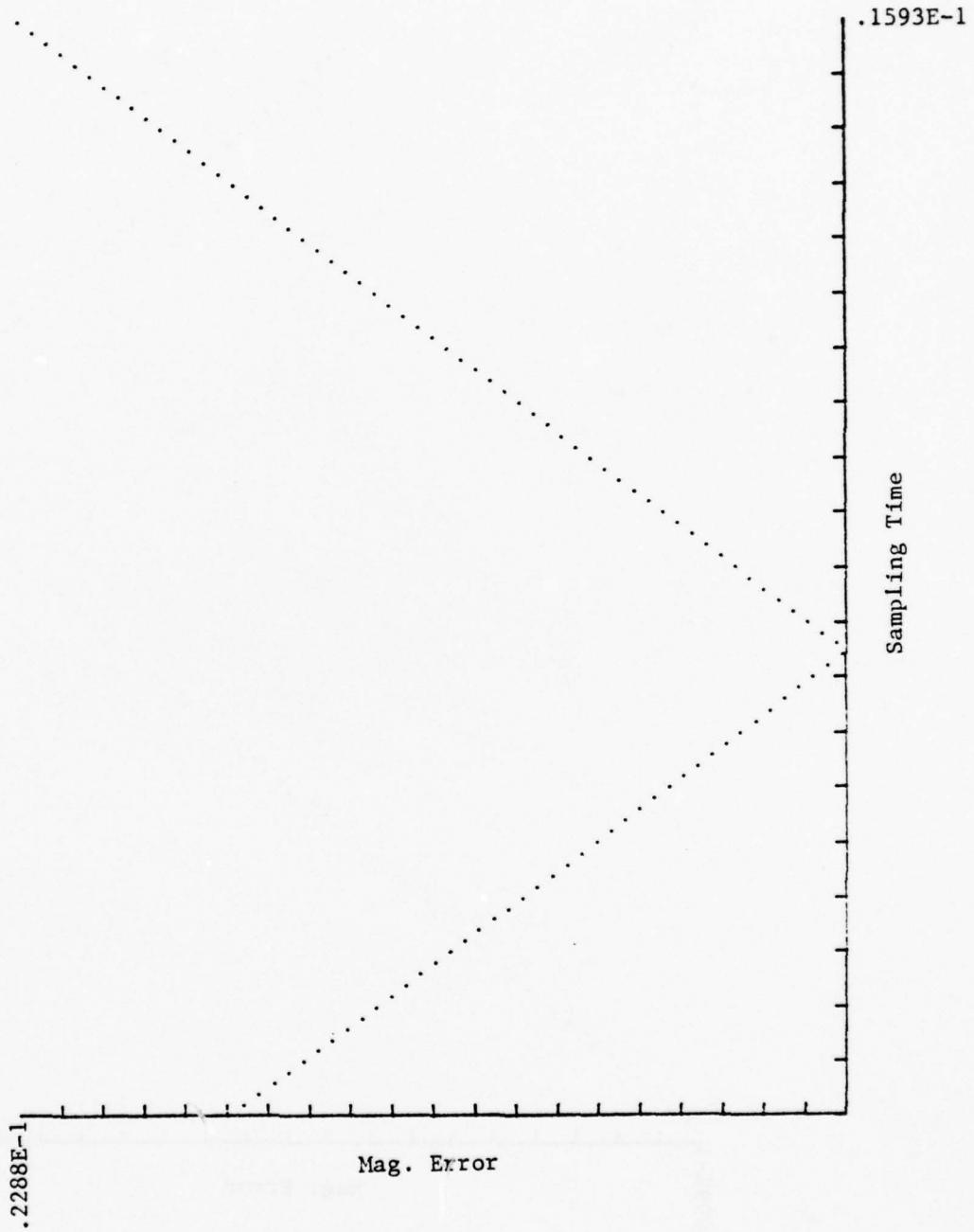


Figure 1.5

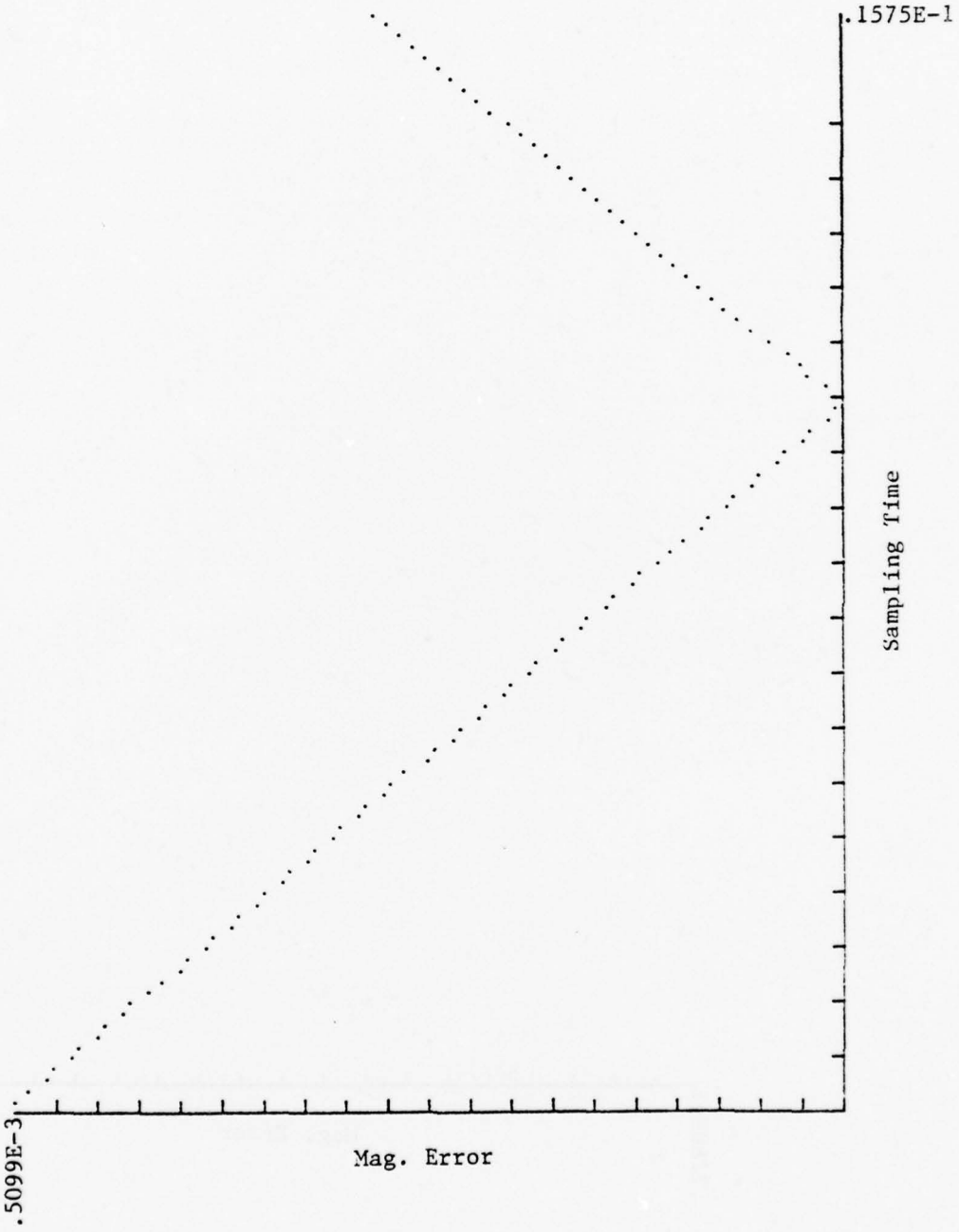


Figure 1.6

```
DELTA H LE. MAXIMUM ERROR
     SAMPLING TIME:      Ø.1574762E-Ø1
     ALPHA =      -Ø.9843751E ØØ      GAMMA =      Ø.7812456E-Ø2
     ROUND

DELTA H LE. MAXIMUM ERROR
     SAMPLING TIME:      Ø.1574768E-Ø1
     ALPHA =      -Ø.9843751E ØØ      GAMMA =      Ø.781249ØE-Ø2
     ROUND

DELTA H LE. MAXIMUM ERROR
     SAMPLING TIME:      Ø.1574775E-Ø1
     ALPHA =      -Ø.9843749E ØØ      GAMMA =      Ø.781252ØE-Ø2
     TRUNCATE

SAMPLING TIME FOR LEAST ERROR:      Ø.1574768E-Ø1
MINIMUM ERROR:      -Ø.5126Ø29E-Ø5
```

Figure 1.7

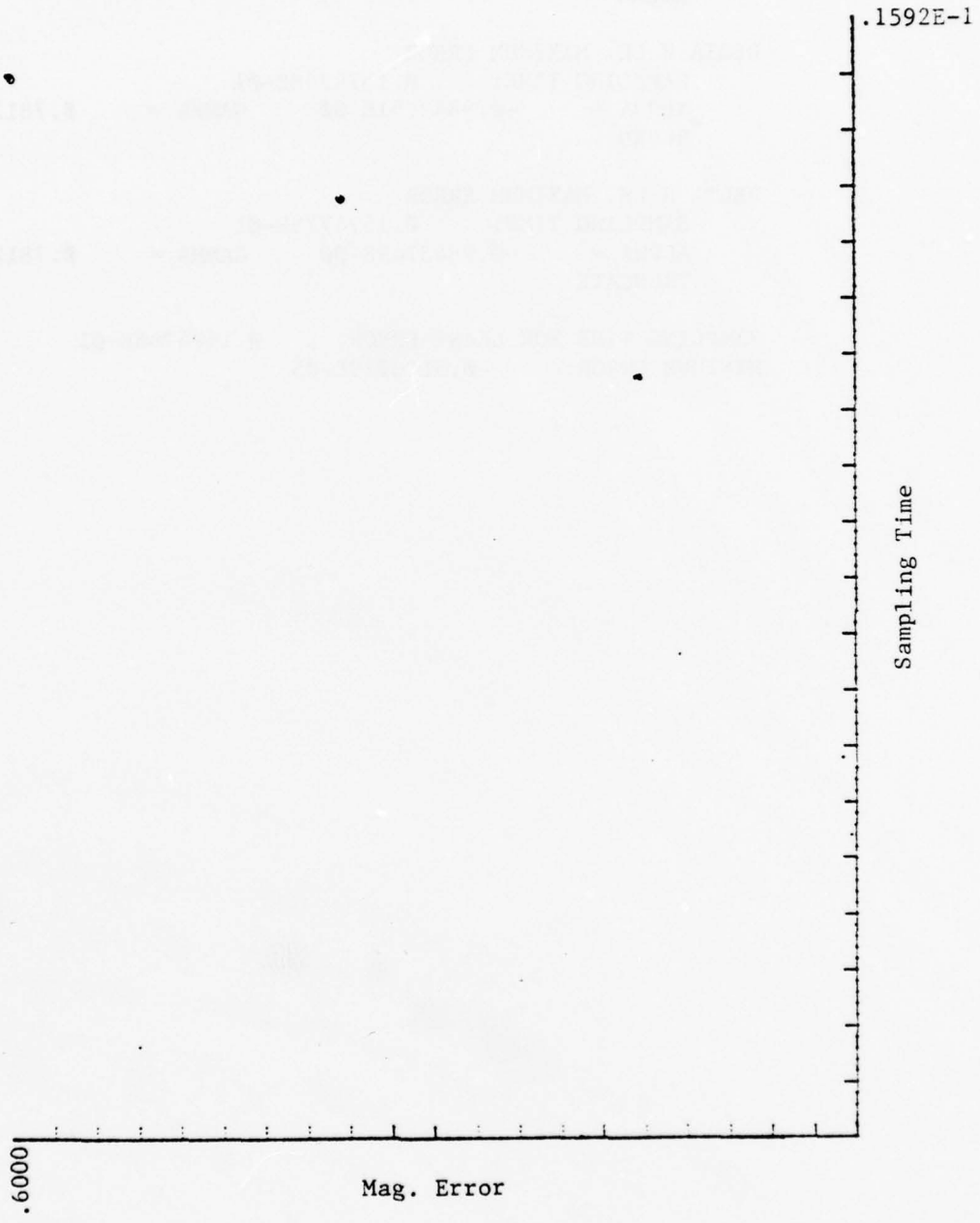


Figure 1.8

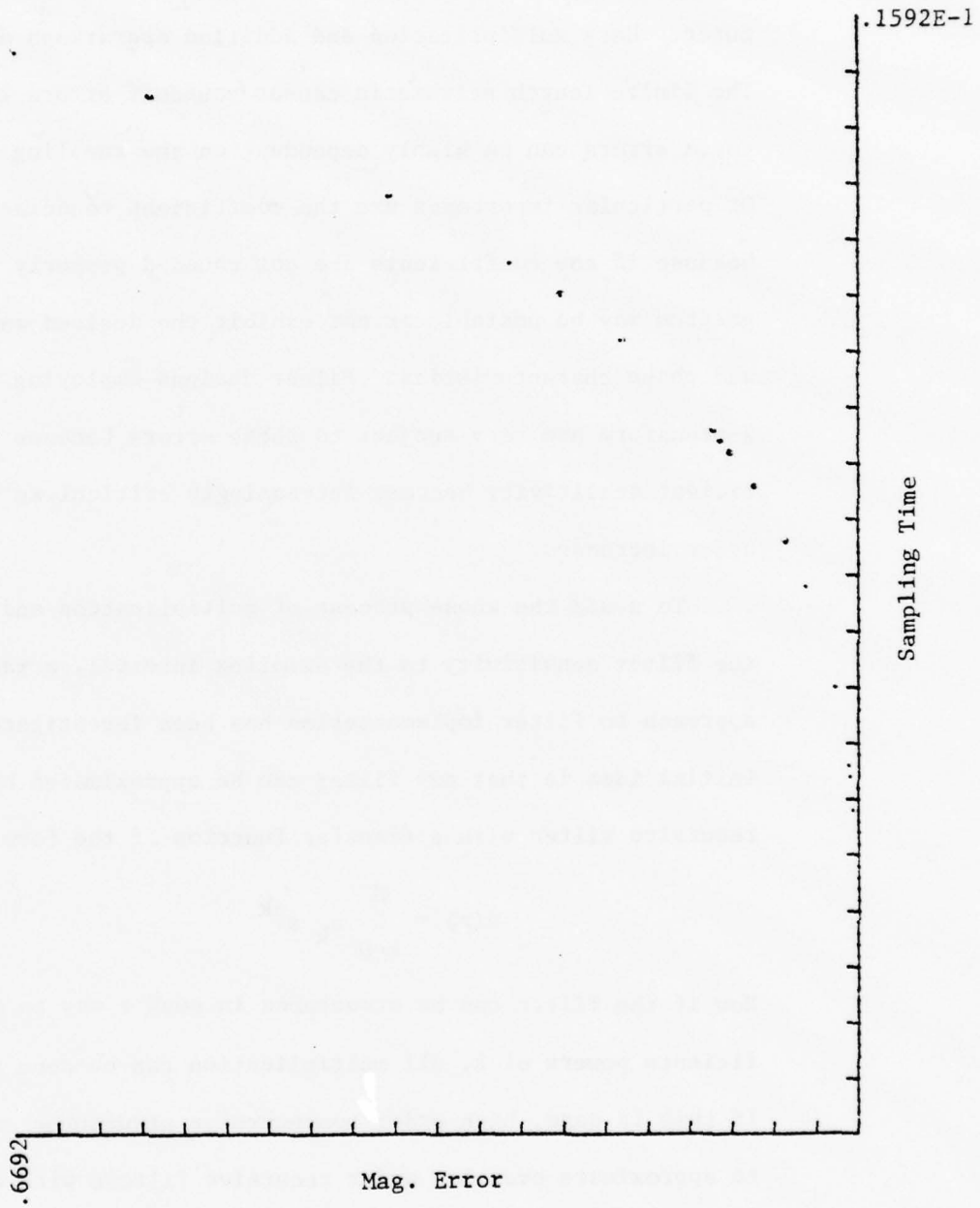


Figure 1.9

2. NEW DESIGN METHODS FOR DIGITAL CONTROL ALGORITHMS

The use of conventional digital control algorithms requires the implementation of a recursive difference equation on a computer. Here multiplication and addition operations are employed. The finite length arithmetic causes roundoff errors to occur and these errors can be highly dependent on the sampling interval. Of particular importance are the coefficient rounding errors, because if the coefficients are not rounded properly the algorithm may be unstable or not exhibit the desired magnitude and phase characteristics. Filter designs employing the bilinear z-transform are very subject to these errors because the coefficient sensitivity becomes increasingly critical as the filter order increases.

To avoid the whose process of multiplication and to decrease the filter sensitivity to the sampling interval, a radical new approach to filter implementation has been investigated. The initial idea is that any filter can be approximated by a nonrecursive filter with a transfer function of the form

$$H(z) = \sum_{k=0}^{N} a_k z^{-k}$$

Now if the filter can be structured in such a way to make the coefficients powers of 2, all multiplication can be done via shifting. If this is done, high order nonrecursive structures can be used to approximate even low order recursive filters with definite advantages.

The first advantage is that implementation can be done directly using large scale integrated circuits. Such an imple-

mentation allows the shifting operations to be designated under computer control so that a design can be used to implement a variety of different filters.

The second advantage is that linear phase can be implemented and this can be very useful in control system design.

The details of this work are provided by the report in Appendix 1. Here the design methodology along with several examples provide the necessary background for this implementation.

Experiments have also been tried on recursive filters using the implementation procedure given in the report. The results up to this time are mixed, however, using the integer design approach we feel that good recursive filter implementations can be obtained.

3. INTERACTIVE SOFTWARE

Four interactive digital filter design packages were written that are valuable to the designer of digital control algorithms and/or digital signal processing algorithms. These are:

a. A Digital Filter Design Program Utilizing the Bilinear Z Transform

b. Programs for Weighted Least Squares Design of Nonrecursive and Recursive Digital Filters

c. A Fortran IV Design Program for Low-Pass Butterworth and Chebychev Digital Filters

d. A Fortran IV Design Program for Butterworth and Chebychev Band-Pass and Band-Stop Digital Filters

All of the programs are written in FORTRAN and run on the DEC PDP-11 and the DEC GT-40 Graphics System. The detailed descriptions are given in Appendices B, C, D and F. Card decks were sent to WPAFB during the course of the contract.

4. FAULT DETECTION

Because of the need to understand how well a digital control algorithm is operating, some work was done on detecting when a discrete time algorithm is not operating correctly due to hardware or software failure. A parameter identification algorithm was used with the method being described via the report of Appendix E.

REFERENCES

1. Final Report, Contract F33615-73-C-1255
Wright Patterson Air Force Base

2. Brubaker, T. A., A Strategy for Coefficient Quantization in Digital Control Algorithms," Computers and Electrical Engineering, Vol. 11, pp. 501-511, 1974.

Appendix A

IMPLEMENTATION OF FIR FILTERS VIA A DIFFERENCE ROUTING DIGITAL FILTER

by

Richard Schneider
Member IEEE

and

Thomas A. Brubaker
Senior Member IEEE

The authors are with the Electrical Engineering Department, Colorado State University, Fort Collins, CO 80523

R. Schneider is on leave from IBM.

ABSTRACT

The difference routing digital filter (DRDF) is a FIR filter whose coefficients are equal to zero, or integral powers of two. The basic DRDF structure is reviewed, and two coefficient restrictions are detailed that will insure bounded input, bounded output stability as well as a finite impulse response. Next, three parallel structures are presented. Each of these new structures will significantly reduce the RMS error between the desired impulse response and the actual filter response. The optimum structure appears to be a filter with a parallel structured transversal part with integer valued taps followed by a recursive part that in the low pass case is a digital integrator. For this new structure, an analysis is given of the RMS error performance in both the time and frequency domain. This analysis is supported by extensive computer simulation results.

I. INTRODUCTION

Finite impulse response (FIR) digital filter structures are attractive in a variety of applications. Among their advantages are the inherent stability and the ease of realizing a linear phase characteristic. Numerous methods now exist [1]-[3] for the design of FIR structures.

One disadvantage of conventional digital FIR filters, in many applications, is the slow operating speed due to the large number of required multiplies. Various methods [4], [5] and [6] have been proposed in the past to reduce or eliminate this multiplier requirement. This paper focuses on the low pass difference routing digital filter (DRDF) [4]. This filter structure consists of a transversal part with coefficients restricted to be zero, or integral powers of two. As originally proposed, the DRDF is limited in the minimum RMS error that can be obtained between the ideal and actual filter impulse response.

To reduce this error, three parallel structures are presented, each of which can significantly reduce the RMS error between the desired filter response and the actual filter response. These three methods are all structurely similar, but there are distinct differences in the design philosophy used. In the first two methods, a parallel structure is created that approximates the error that would have occurred in the original design. This error signal is added in such a way as to provide overall error reduction. In both cases, the parallel structure can be implemented with minimal additional hardware. The third approach, which appears to be optimum, also uses a parallel filter structure.

In this latter case, the purpose of the parallel structure is to provide integer valued taps that more accurately approximate the desired filter function. This third approach has two principle advantages over the first two methods: (1) reduced hardware requirements, and (2) a more straightforward and logical design procedure.

Several examples of improved performance with the new structures are given using computer simulation. An analysis of the RMS error performance of the optimum structure is also given in both the time and frequency domain. The analysis is seen to agree very favorably with computer simulation results.

This work was all done with low pass filter designs. However, similar results can also be obtained for both band pass and high pass circuits.

II. LOW PASS DRDF STRUCTURE

The structure of a low pass DRDF is shown in Figure 1. The coefficients a_o .... a_{N-1} of the transversal part are restricted to be zero or integer powers of two. The recursive part is a digital integrator with a single coefficient b_1 equal to minus one. The coefficients of the transversal part are chosen to be approximations to the differences between successive values of the desired filter impulse response $h_D[nT]$

$$a_j \simeq h_D[jT] - h_D[(j-1)T] \tag{1}$$

Thus, if the input signal to the filter is a unit impulse,

$$\begin{aligned} \delta[nT] &= 1 \qquad \text{for } n = 0 \\ &= 0 \qquad \text{for } n \neq 0. \end{aligned} \tag{2}$$

Therefore, the output y(nT) will be an approximation to the desired impulse response itself

$$h_D[nT] \simeq y[nT] = y[(n-1)T] + \sum_{j=0}^{N-1} a_j \, \delta[(n-j)T] \tag{3}$$

The use of a digital integrator places one restriction on the a_j coefficients to insure a finite impulse response (FIR) filter, the sum of the a_j coefficients must be zero.

This is shown as follows. If the filter is FIR, then we desire y[(N-1)T] to be zero.

Hence:

$$y[(N-1)T] = y[(N-2)T] + a_{N-1} = 0 \tag{4}$$

but,

$$y[(N-2)T] = \sum_{j=0}^{N-2} a_j \tag{5}$$

Therefore:

$$\sum_{j=0}^{N-1} a_j = 0 \tag{6}$$

The restriction of equation six is not a practical problem. Since the desired finite impulse response in practice will always damp out to zero at (N-1)T as in Figure 2, we have:

$$h_D[(N-1)T] = 0 = y[(N-1)T] = \sum_{j=0}^{N-1} a_j \tag{7}$$

Therefore, the restrictions on the a_j coefficients are:

$$a_j = 0, \pm 2^K \quad \text{for } K = 0, 1, 2, \ldots$$

$$\sum_{j=0}^{N-1} a_j = 0 \tag{8}$$

III. DRDF OPERATION AND DESIGN

The design of a DRDF is based on approximating a desired finite impulse response. Consider a desired impulse response $h_D[nT]$, the first few samples of which are shown in Figure 3. Without loss of generality, assume $h_D[0]$ is zero. Further, consider that $h_D[nT]$ has been amplitude scaled ($h_D[nT] = F \cdot h_s[nT]$) such that the maximum change is:

$$\max \left| h_s[jT] - h_s[(j-1)T] \right| = 2^{Km}$$

where Km is the largest exponent being considered in the design. Thus, the DRDF will approximate a scaled version of the desired impulse. This scaled value is then multiplied by a scale factor (F) to give the desired impulse response as in Figure 4. This will insure that the coefficient values will be able to follow the maximum slope of the scaled impulse response, $h_s[nT]$. Note that:

$$\max \left| h_D[jT] - h_D[(j-1)T] \right| = F\ 2^{Km}$$

Therefore, as Km increases, the scale factor F will get smaller. The value of the first coefficient, a_o is selected from 0, $\pm 2^K$; K = 0, 1, ..., Km so as to be closest to the first change $h_s[T] - h_s[0]$. Since $h_s[0]$ equals zero, we have

$$a_o \simeq h_s[T] \tag{9}$$

The design proceeds recursively, selecting a_j from 0, ± 1, ± 2 ... to minimize:

$$\left| a_j - \{h_s[(j+1)T] - \sum_{k=0}^{j-1} a_k\} \right| \tag{10}$$

Figure 5 is a comparison of the entire desired impulse response sequence $h_D[nT]$ and the error for the DRDF approximation for Km=4. Figure 6 is a comparison of the desired magnitude response and the error for the DRDF magnitude response. For this example T = 0.05 sec and N = 200. The "IDEAL" Chebychev impulse response used in this and all subsequent examples was obtained from synthetic division of the H(Z) found using the impulse invariant design [7].

Figure 7 is a plot of the RMS error between the desired impulse response and the actual DRDF impulse response for the four pole Chebychev filter. The RMS error is expressed as a percentage of the peak value of the impulse response, and it is plotted vs Km. It is seen that little improvement results beyond a Km of 3 or 4. The parallel structure introduced in the next section provides a method of significantly reducing this RMS error.

IV. PARALLEL DRDF STRUCTURE

It was shown in the previous section that increasing Km beyond 3 or 4 does little to further reduce the percentage RMS error. There may be many applications where further improvement is desirable. One way to do this is to both double the sampling rate and also the number of transversal stages. For example, doubling the sampling rate and doubling the number of transversal stages will cut the RMS error in half. Since it may not always be possible to double the sampling rate, and since doubling the required number of stages is not attractive, another alternative is desirable. Three different alternative designs are considered below.

In the first method, an error sequence e[nT] is defined as the difference between the desired impulse response and that actually generated by the DRDF. That is:

$$e[nT] \triangleq h_D[nT] - h_A[nT] \tag{11}$$

If the error sequence e[nT] of equation 11 could somehow be approximated, $\hat{e}[nT]$ and added to the DRDF output, the new signal $h_A'[nT] = h_A[nT] + \hat{e}[nT]$ would be a better approximation to the desired signal. Since e[nT] is itself a finite duration sequence, it is possible to approximate it with a second DRDF filter as shown in Figure 8. These two parallel filters can share much of the basic DRDF hardware as shown in Figure 9. Note that the parallel DRDF will have its own scale factor F_2. In some cases, F_2 can itself be satisfactorily approximated by an integral power of 2, however, in general this is not the case.

Conceptionally, this process of approximating the DRDF error could be continued to two, three or more parallel stages. Of course, at some point it will be more expedient to use a conventional filter structure.

Figure 10 is a plot of the percentage RMS error vs Km for the basic DRDF and for one and two parallel stages. This is for the Chebychev filter used in previous examples. It is seen that the RMS error is reduced by a factor of about 3 each time a parallel branch is added. Thus, RMS errors well below 1% of the peak value of the impulse response are feasible with this approach.

The parallel filter does not approximate the error waveform nearly as well as the basic DRDF matches the original desired impulse response. This is because the error sequence is quite noise-like with rapid changes. The error waveform for the Chebychev filter was shown in Figure 5. Except for the final sequence values, the error signal is very much like white noise. The sinusoidal appearance of the final sequence values is due to the fact that the small ripple values in the desired impulse response is being matched by a zero output from the DRDF.

It has been found that a consistently better approximation to the error signal may be made by roughly quantizing the error signal to integer powers of 2 or zero. Thus, the new filter shown in Figure 11 would be similar to that of Figure 9, but without a second integrator. This is the second design method.

Figure 12 is a plot of the percentage RMS error vs Km for the basic DRDF and for one and two parallel stages where the parallel sections are rough quantizations of the error signals to integer powers of 2 or zero. A comparison with Figure 10 shows that this second approach is clearly better. Similar improvements for other filters have also been found, and the results of Figures 10 and 12 may be considered typical.

The third method is aimed directly at the reason why the basic DRDF error performance does not improve as Km goes beyond 3 or 4. This is because as Km increases, the allowable tap values are spread further apart. If, however, as Km increased, all the integer values were allowable, then clearly the quantization would improve and result in reduced RMS error. It is

worth noting at this point that uniform quantization intervals of any desired value could be achieved with appropriate scaling. This then represents a rough quantization of the transversal coefficients with the subsequent integration acting to smooth the overall impulse response.

Because of the speed and cost benefits, it is desirable to obtain integer value taps with shifting and adding rather than by the use of hardware multipliers. A direct approach would be to have a parallel filter section for various integer values of 2. In this case, each tap weight would be constructed from its binary equivalent. For example, for a tap value of 5 (101) there would be three parallel filter sections with connections made to the first and third, but not to the second section. Each section would have its own adder. Figure 13 shows such a filter which is capable of producing tap values 0 to 7, and with suitable two's complement circuitry, -7 to +7.

A similar approach, but one with further hardware economies, is to permit both positive and negative values of the integer values of two. Refer to Table I. This shows how the integers up to 42 could be implemented with just three parallel sections. The first section could have up to five shifts, the second section up to three shifts, and the third section a single shift. Thus, 31 is implemented as 32-1 rather than as 16+8+4+2+1. The optimum implementation of this concept will be dependent on the application and device technology. One possible structure is presented by Kishi et al. [8].

Thus, we have considered a third method of reducing RMS error that of creating integer value taps. We have also looked

at several possible implementations of the integer tap concept. A major advantage of the integer tap approach is that separate scale factors (hardware multipliers) are not required in the integer tap approach. A second advantage with this method is that error performance continues to improve as the number of shifts is increased. In the basic DRDF and the other parallel filter approaches, improvement leveled off beyond a Km of 3 or 4.

Figure 14 compares the performance of the DRDF with the integer tap approach for the four pole Chebychev filter. In the graph at Km = 4, the basic DRDF is allowing taps values of 0, $\pm$1, $\pm$2, $\pm$4, $\pm$8 and $\pm$16, while the integer approach is allowing all the integer values 0, $\pm$1, $\pm$2, $\pm$3 . . . . $\pm$16. For this filter, the integer approach surpasses the best results of the other methods at a maximum integer value of $\pm$138.

Figure 14 compares the performance of the basic DRDF with the integer tap approach for the four pole Chebychev filter.

The design of the integer taps would proceed in a recursive fashion similar to the basic DRDF design. Tap values a_j would be selected from the allowable integers 0, $\pm$1, $\pm$2, ... $\pm$MAX to minimize:

$$\left| a_j - \{h_s[(j+1)T] - \sum_{k=0}^{j-1} a_k\} \right| \qquad (12)$$

V. TIME DOMAIN ERROR PERFORMANCE

An important measure of performance will be the RMS error between the desired impulse response sequence $h_D[nT]$ and the actual sequence $h_A[nT]$. In the time domain, the RMS error can easily be calculated for any filter design from:

$$E_T = \sqrt{\sum_{n=0}^{N-1} \frac{h_D[nT] - h_A[nT]}{N}} \qquad (13)$$

where the subscript T stands for time-domain.

It is desirable to be able to estimate what this error may be for a particular filter without going through the actual design procedure. In this section, the time-domain RMS error of the integer tap approach is estimated.

The errors measured at the filter output may be assumed to be uniformly distributed with zero mean and variance $Q^2/12$ [8]. In the case of integer taps Q = 1. Since the mean is zero, the RMS error will be $1/\sqrt{12}$.

In actual practice, the designer would want to know how the RMS error compared with the peak value of the impulse response sequence. Therefore, it is desirable to have an estimate of the peak value of the impulse response sequence. This estimate can be made using the following rules.

1. The main lobe of a typical high order low pass filter will have a width that is approximately equal to the reciprocal of the cutoff frequency.
2. The average slope of the main lobe will be about one half its maximum value.

Therefore, an estimate of the peak output of a DRDF can be made given the filter cutoff frequency, fco, the sampling interval, T, and the maximum integer, Im.

$$P \simeq \frac{Im}{2T} \cdot \frac{1}{2\ fco} \tag{14}$$

In Table II, a comparison is made of actual and estimated RMS errors as a percentage of the peak value of the impulse response sequence. The percentage estimates are obtained by dividing the RMS estimates obtained from equation (13) by the peak value estimates from equation (14). In all cases, the estimates represent a conservative bound on the actual error. The RMS error estimate as a percentage of the peak value (E_T') is thus seen to be:

$$E_T' = \frac{2T}{Im\sqrt{3}} \tag{15}$$

Therefore, for a fixed sample rate, the percentage RMS error is inversely proportional to the maximum integer value, Im.

VI. FREQUENCY DOMAIN ERROR PERFORMANCE

Since filter performance requirements are often given in terms of the frequency domain, it is important to evaluate the frequency response error performance.

The error sequence e[nT] is defined in equation (11) as the difference between the desired impulse response and that actually generated by the filter. In the frequency domain, the magnitude of the error at any frequency ω_i can be obtained by evaluating the z transform of e[nT] at $z = e^{j\omega_i T}$. Therefore,

the magnitude response of the error may be written as:

$$E(\omega) \triangleq E(z=e^{j\omega T}) = \sum_{n=0}^{N-1} e[nT]z^{-nT} \qquad (16)$$

If equation (15) is evaluated for a set of frequencies ω_o, ω_1, ... ω_{M-1}, then we can define an expression for the RMS error in the frequency domain.

$$E\omega = \sqrt{\frac{\sum_{j=0}^{M-1} \left(E(\omega j)\right)^2}{M}} \qquad (17)$$

We can see from equation (15) that the value of $E(\omega)$ and hence, the value of the RMS error $E\omega$ is a function of the number of coefficients, N in the FIR filter used to generate the impulse response. In 1973, Chan and Rabiner [9] showed that $E\omega$, the RMS error in the frequency domain is found from:

$$E\omega = \sqrt{N}\, E_T \qquad (18)$$

here E_T is the time domain RMS error defined in equation (13).

Estimates of the frequency domain RMS error can be derived from the time domain RMS error estimate by application of equation (18). Table III is a comparison of the estimated frequency response errors with the actual errors for the Chebychev filter previously used.

VII. OPERATION EXAMPLES

Specific examples are given in this section of the time and frequency domain performance of DRDF structures compared with the ideal time and frequency responses. The marked improvement

using the integer filter is also shown. Because the performance is so close to the ideal, especially if an integer filter is used, only the error (difference from ideal) is plotted along with the ideal waveforms.

Figure 15 shows the ideal impulse response for the Chebychev filter used in previous examples. Also, plotted with an expanded amplitude scale are the error waveforms for an integer filter with Im = 16. The improvement gained with the integer filter is readily apparent by comparing Figure 15 with the basic DRDF results of Figure 5. There are reductions in both the peak and RMS errors. This same improvement is mirrored in the frequency domain as seen in Figure 16 which shows the ideal magnitude response along with the magnitude errors for the integer filter. Compare Figure 16 with the results shown in Figure 6.

Figure 17 shows the ideal step response for the same Chebychev filter. Again, the error waveforms are plotted on the same time scale and we see the improvement achieved with the use of the integer filter. The step response also indicates that the filter is BIBO stable. Very similar results are obtained with all other low pass filter designs.

VIII. SUMMARY

The structure and performance of the difference routing digital filter (DRDF) has been explored. Design restrictions and the basic filter design have been detailed. Examples of the RMS error performance were given.

Three enhanced DRDF structures were presented. Each of these new approaches used parallel filter sections. The parallel

filters could share much of the basic DRDF hardware. The optimum approach was to use integer value taps constructed from three parallel sections. The need for hardware multipliers was avoided through the use of shifting and adding.

Expressions for the RMS error of the integer tap method were derived. It was shown that the RMS error is inversely proportional to the maximum integer used. A low pass four pole Chebychev filter was used as an example. Similar results have been obtained for other low pass structures and the results given in this paper are typical of what might be expected.

References

1. L. R. Rabiner, "Techniques for Designing Finite-Duration Impulse Response Digital Filters", IEEE Trans. Comm. Techn. Vol. COM-19, pp. 183-195, April 1971.

2. L. R. Rabiner and R. W. Schafer, "Recursive and Nonrecursive Realizations of Digital Filters Designed by Frequency Sampling Techniques", IEEE Trans. Audio Electroacoust., Vol. AU-21, pp. 477-484, Dec. 1973.

3. D. C. Farden and L. L. Scharf, "Statistical Design of Nonrecursive Digital Filters", IEEE Trans. Acoustics, Speech, and Signal Proc., Vol. ASSP-22, No. 3, pp. 188-196, June 1974.

4. P. J. Van Gerwin et al, "A New Type of Filter for Data Transmission", IEEE Trans. on Communications, Vol. COM-23, No. 2, pp. 222-234, Feb. 1975.

5. A. Peled and B. Liu, "A New Approach to the Realization of Nonrecursive Digital Filters", IEEE Trans. Audio Electroacoust., Vol. AU-21, pp. 477-484, Dec. 1973.

6. G. Kishi et al, "A New Realization Scheme of Digital Filters-Row-Wise Addition Configuration".

7. C. M. Rader and B. Gold, "Digital Filter Design Techniques in the Frequency Domain", Proc. IEEE, Vol. 55, pp. 149-171, Feb. 1967.

8. G. Kishi et al, "A New Realization Scheme of Digital Filters -- Row-Wise Addition Configuration", IEEE Trans. Acoust., Speech, Signal Processing, Vol, ASSP-25, No. 3, pp. 256-257, June 1977.

9. D. S. K. Chan and L. R. Rabiner, "Analysis of Quantization Errors in Direct Form for FIR Digital Filters", IEEE Trans. Audio Electroacoust., Vol. AU-21, pp. 354-366, Aug. 1973.

0.	0	15.	16-1	30.	32-2
1.	1	16.	16	31.	32-1
2.	2	17.	16+1	32.	32
3.	2+1	18.	16+2	33.	32+1
4.	4	19.	16+2+1	34.	32+2
5.	4+1	20.	16+4	35.	32+2+1
6.	4+2	21.	16+4+1	36.	32+4
7.	8-1	22.	16+4+2	37.	32+4+1
8.	8	23.	16+8-1	38.	32+4+2
9.	8+1	24.	16+8	39.	32+8-1
10.	8+2	25.	16+8+1	40.	32+8
11.	8+2+1	26.	16+8+2	41.	32+8+1
12.	8+4	27.	32-4-1	42.	32+8+2
13.	8+4+1	28.	32-4		
14.	16-2	29.	32-4+1		

Integer Values with Three Parallel Sections

TABLE I

Im	Estimated	Actual
2	2.8×10^{-2}	1.9×10^{-2}
4	1.4×10^{-2}	1.1×10^{-2}
8	7.2×10^{-3}	5.7×10^{-3}
16	3.6×10^{-3}	3.0×10^{-3}
32	1.8×10^{-3}	1.5×10^{-3}
64	0.9×10^{-3}	0.8×10^{-3}

Comparison of Estimated and Actual
Time Domain RMS Errors
for the Integer Tap Filter

TABLE II

Im	Estimated	Actual
2	0.396	0.240
4	0.198	0.140
8	0.102	0.065
16	0.051	0.033
32	0.025	0.012
64	0.012	0.007

Comparison of Estimated and Actual
Frequency Domain RMS Errors
for the Integer Tap Filter

TABLE III

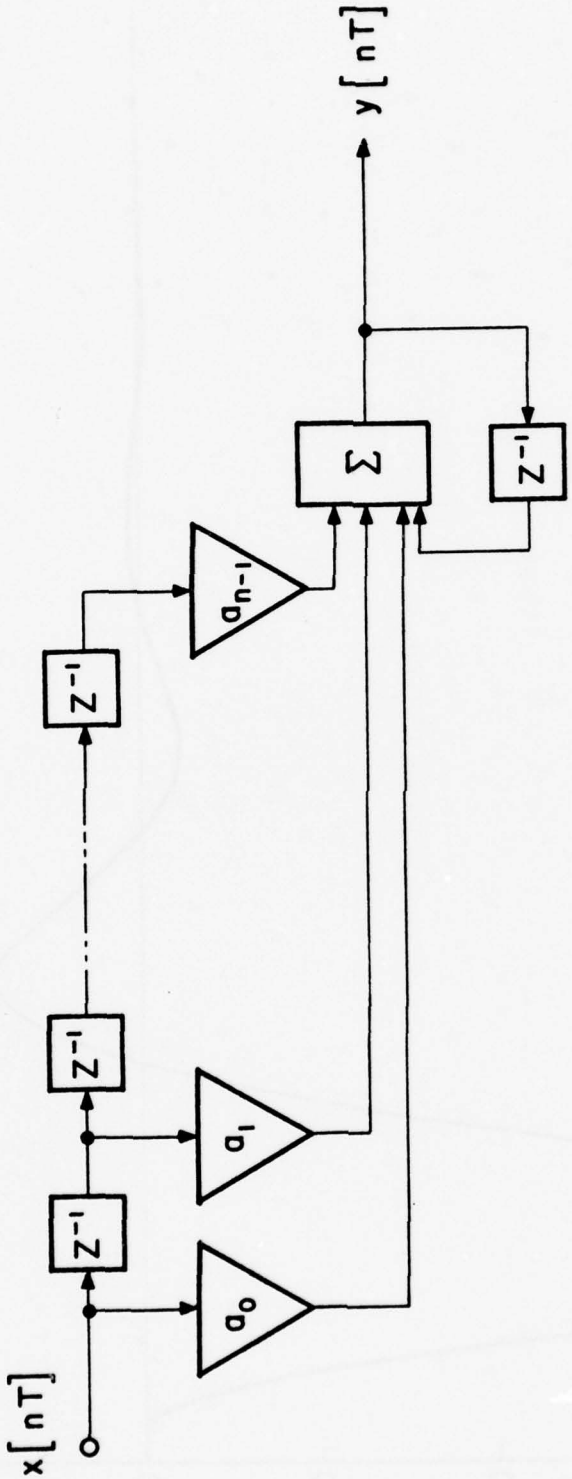


FIG. 1 - LOW PASS DRDF STRUCTURE

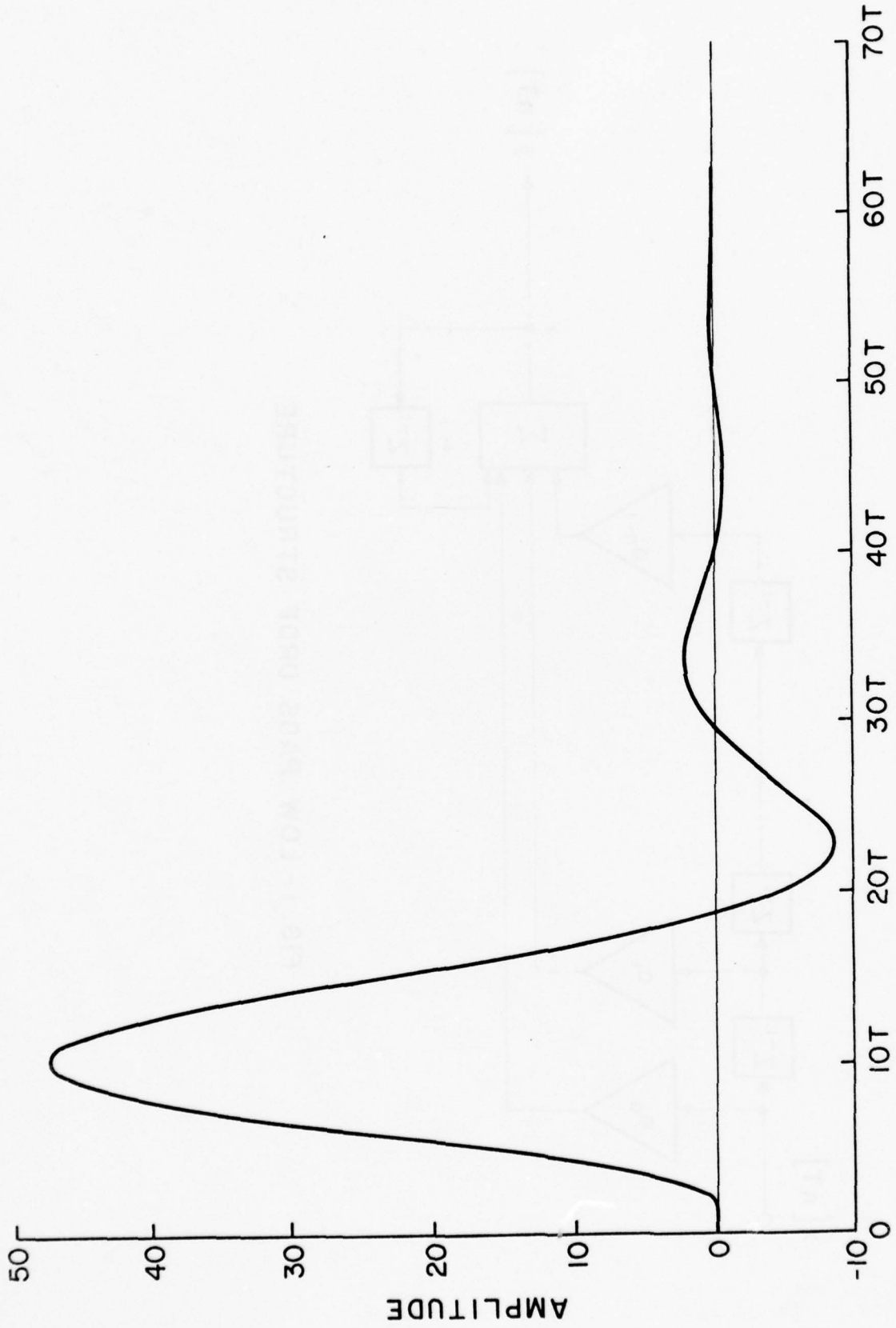


FIG. 2- TYPICAL LOW PASS FINITE IMPULSE RESPONSE

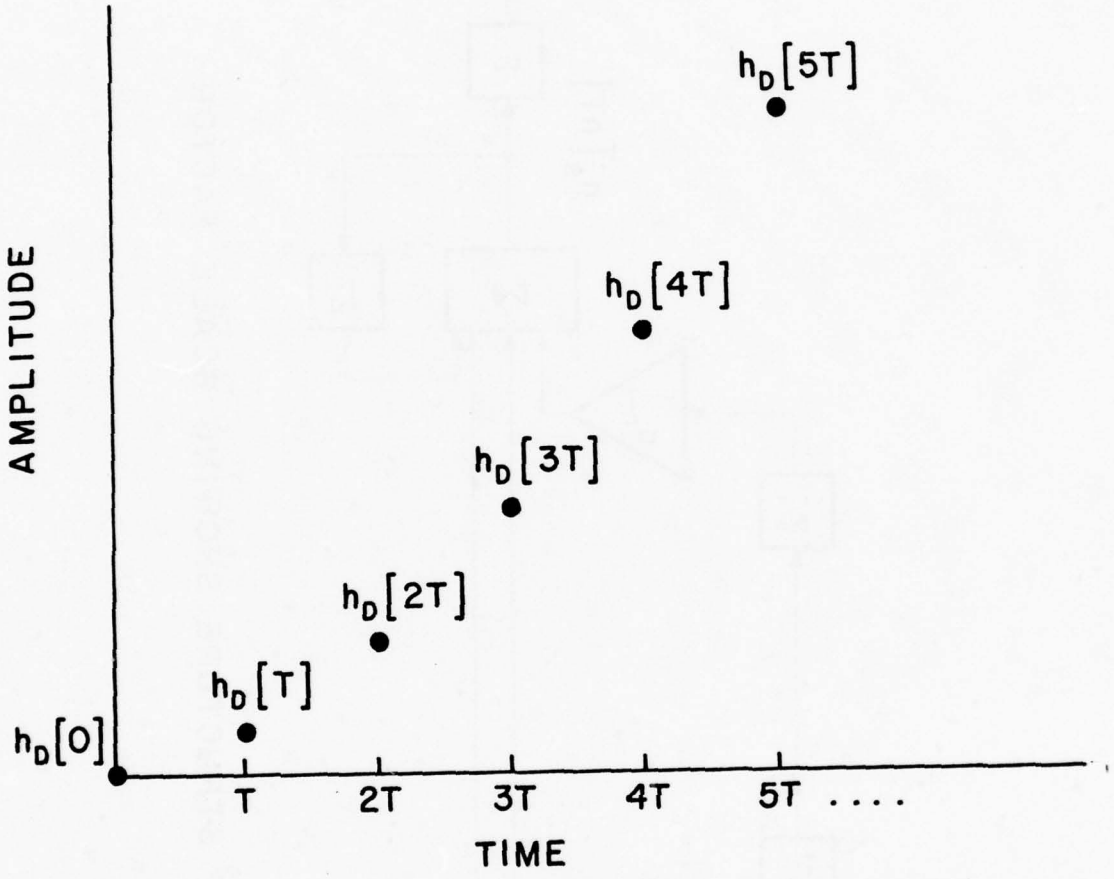


FIG. 3 - INITIAL PORTION OF DESIRED IMPULSE RESPONSE

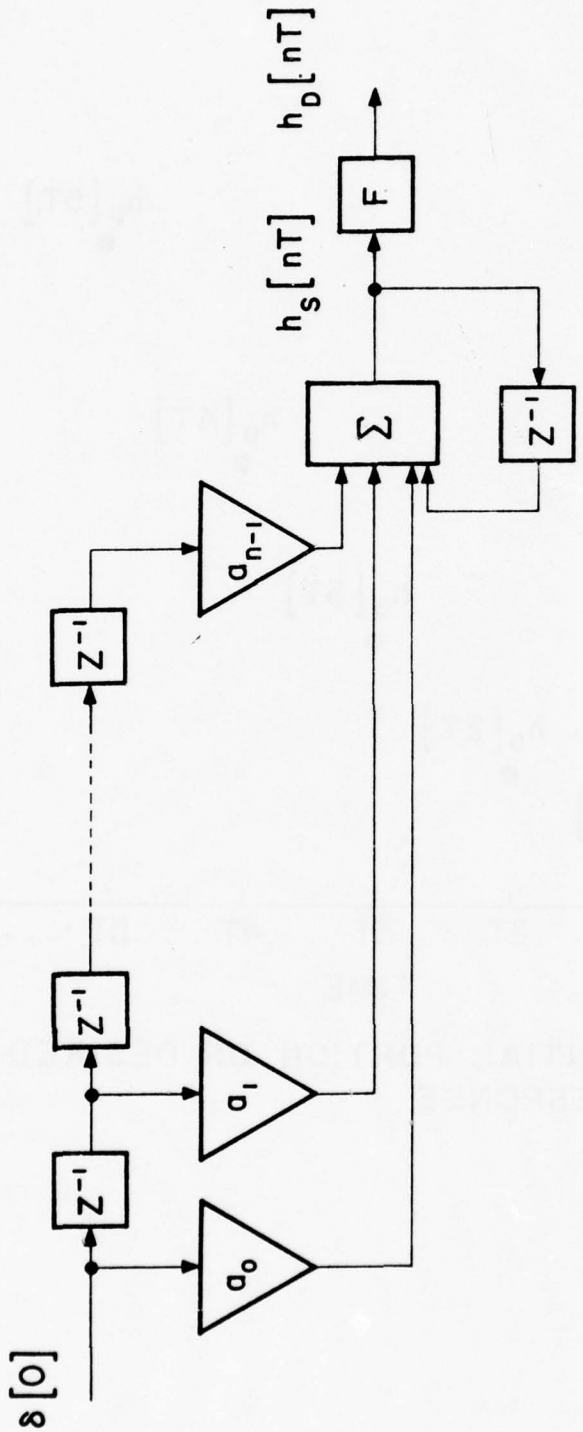


FIG. 4 - DRDF STRUCTURE SHOWING SCALE FACTOR

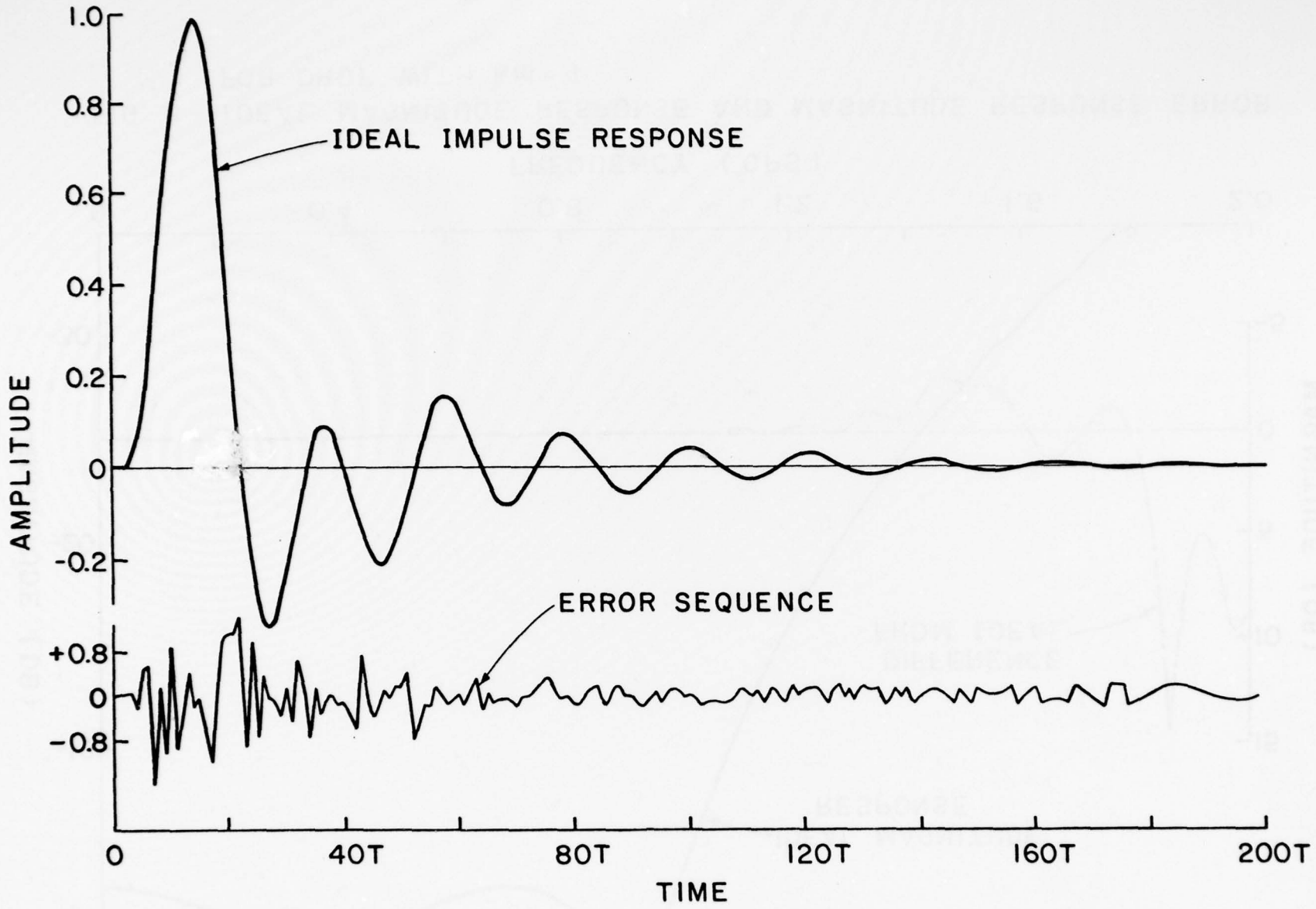


FIG. 5 - IDEAL IMPULSE RESPONSE AND ERROR SEQUENCE OF DRDF FOR Km = 4

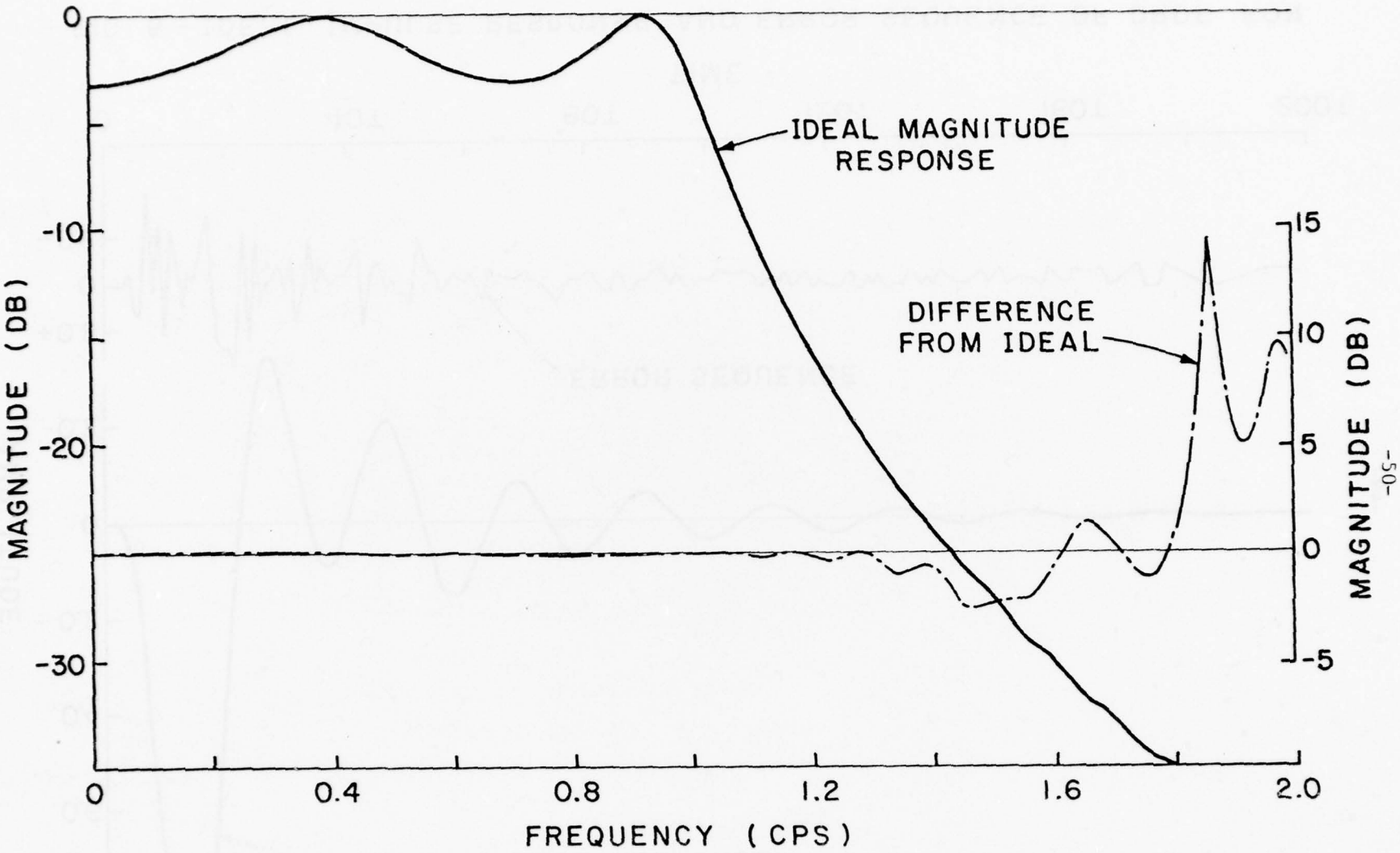


FIG. 6 - IDEAL MAGNITUDE RESPONSE AND MAGNITUDE RESPONSE ERROR FOR DRDF WITH Km = 4

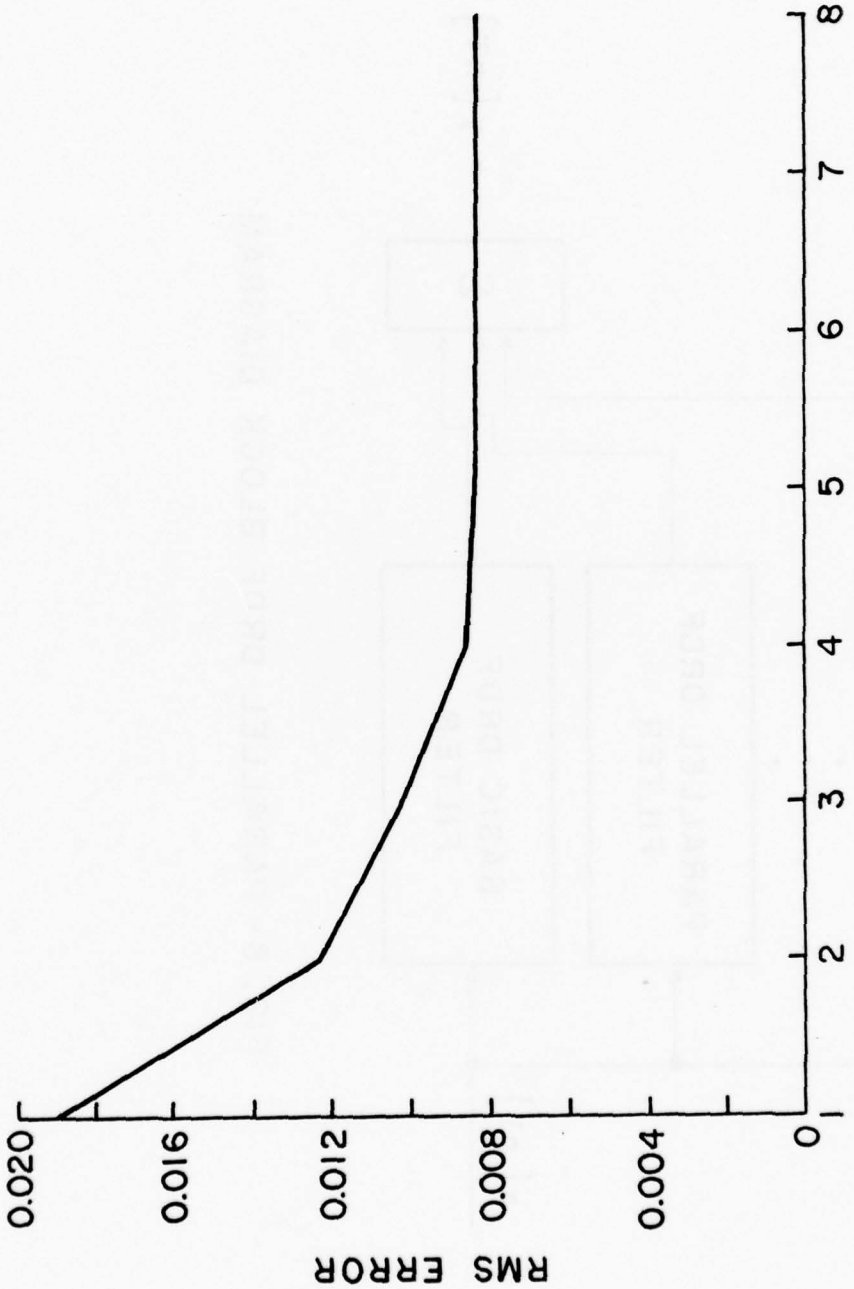


FIG. 7- RMS ERROR vs. Km FOR BASIC DRDF

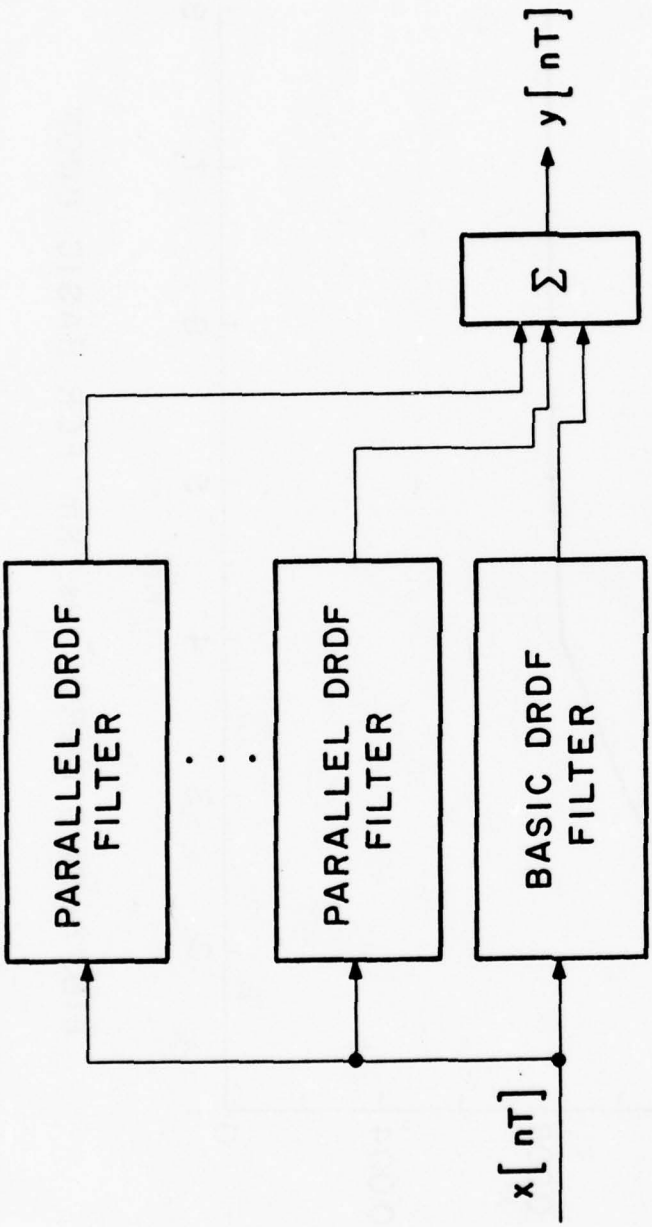


FIG. 8- PARALLEL DRDF BLOCK DIAGRAM

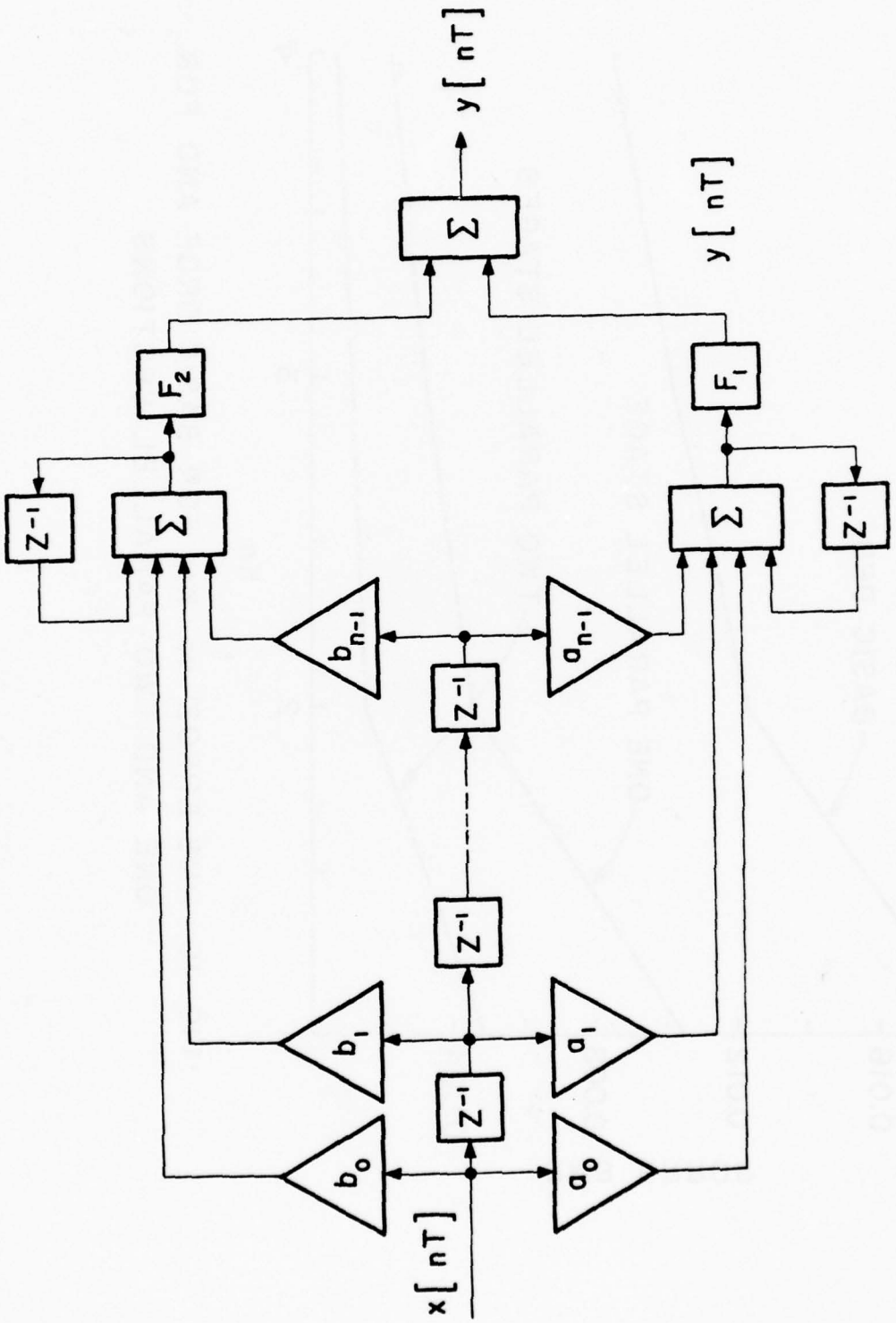


FIG. 9 – DETAIL STRUCTURE OF A PARALLEL DRDF

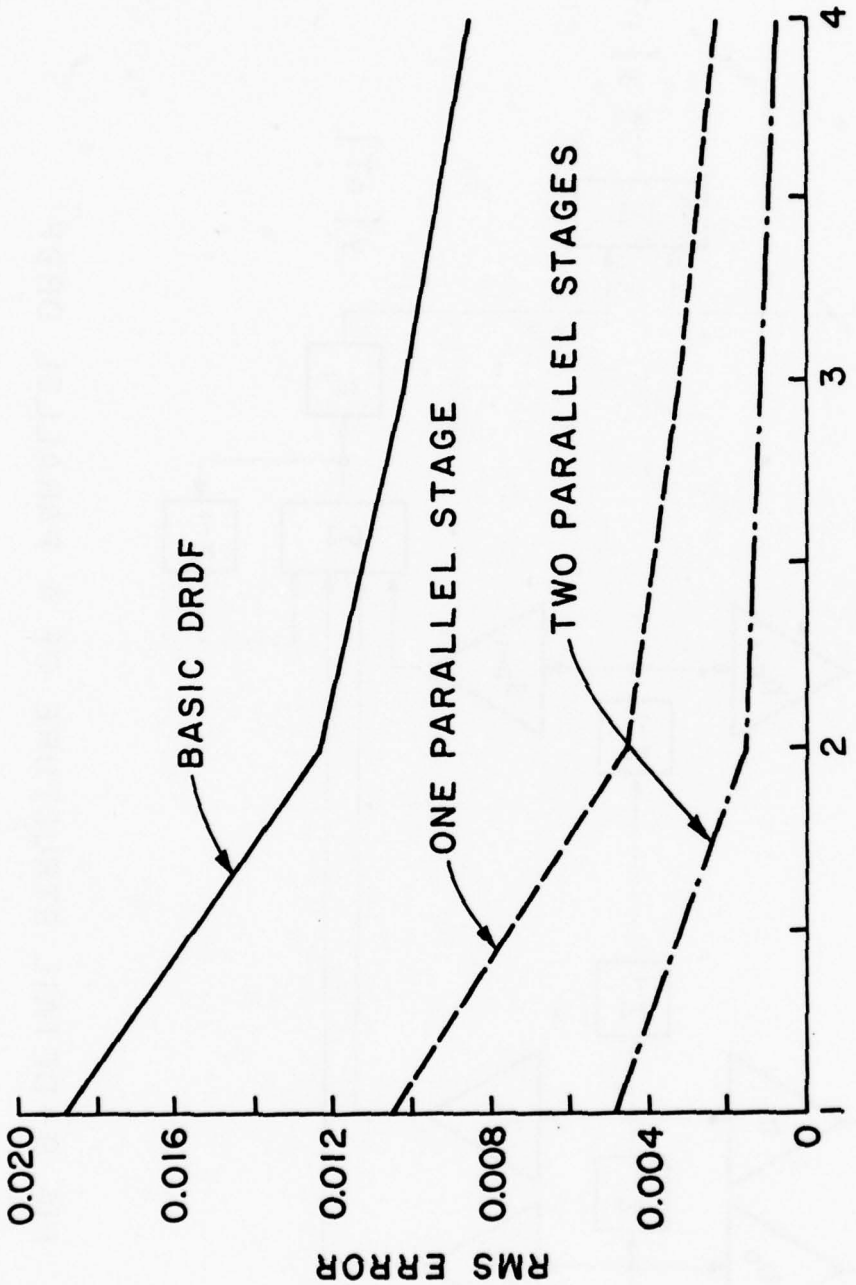


FIG. 10- RMS ERROR vs. Km FOR BASIC DRDF AND FOR ONE AND TWO PARALLEL SECTIONS

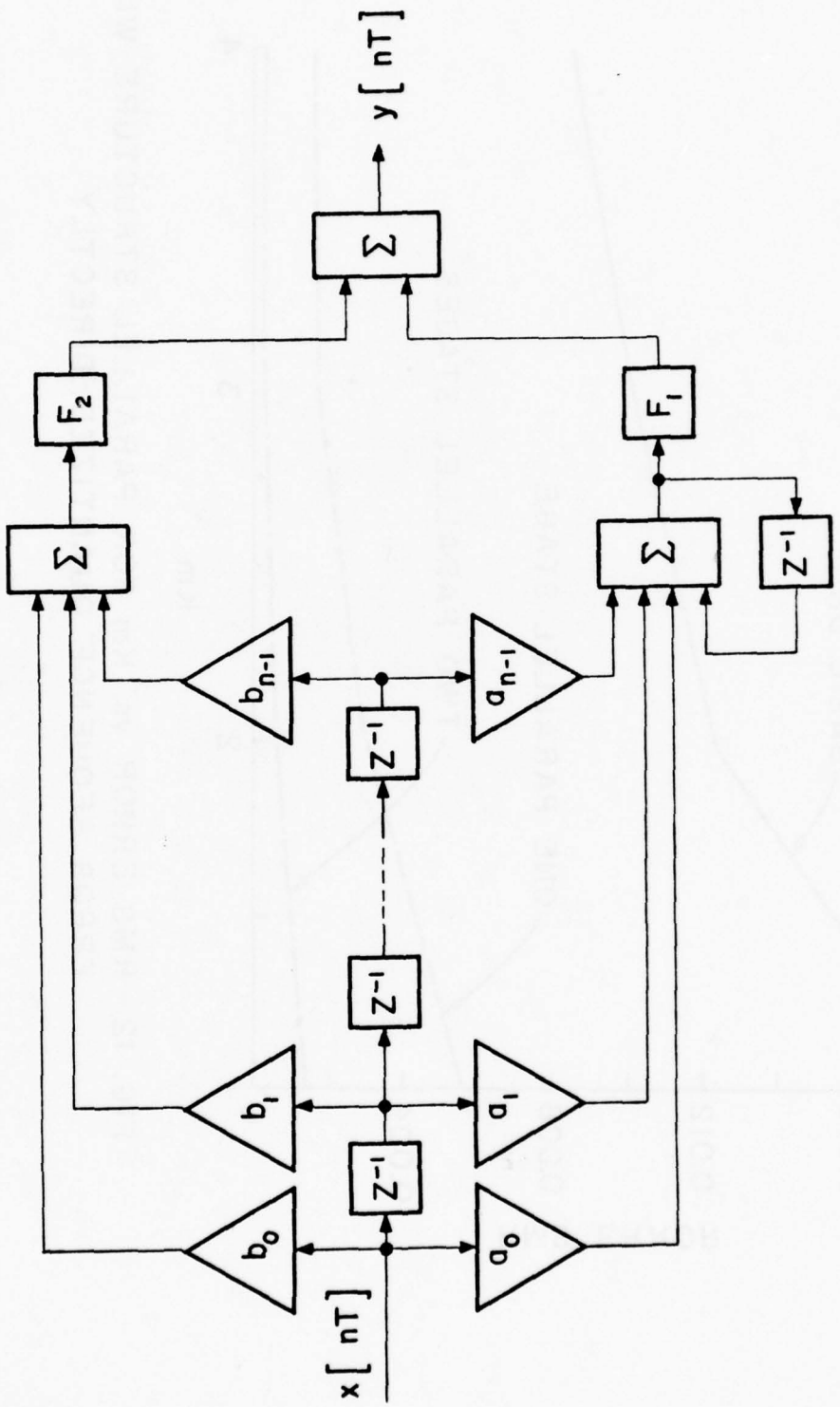


FIG. 11 - STRUCTURE OF A PARALLEL FILTER THAT QUANTIZES ERROR DIRECTLY

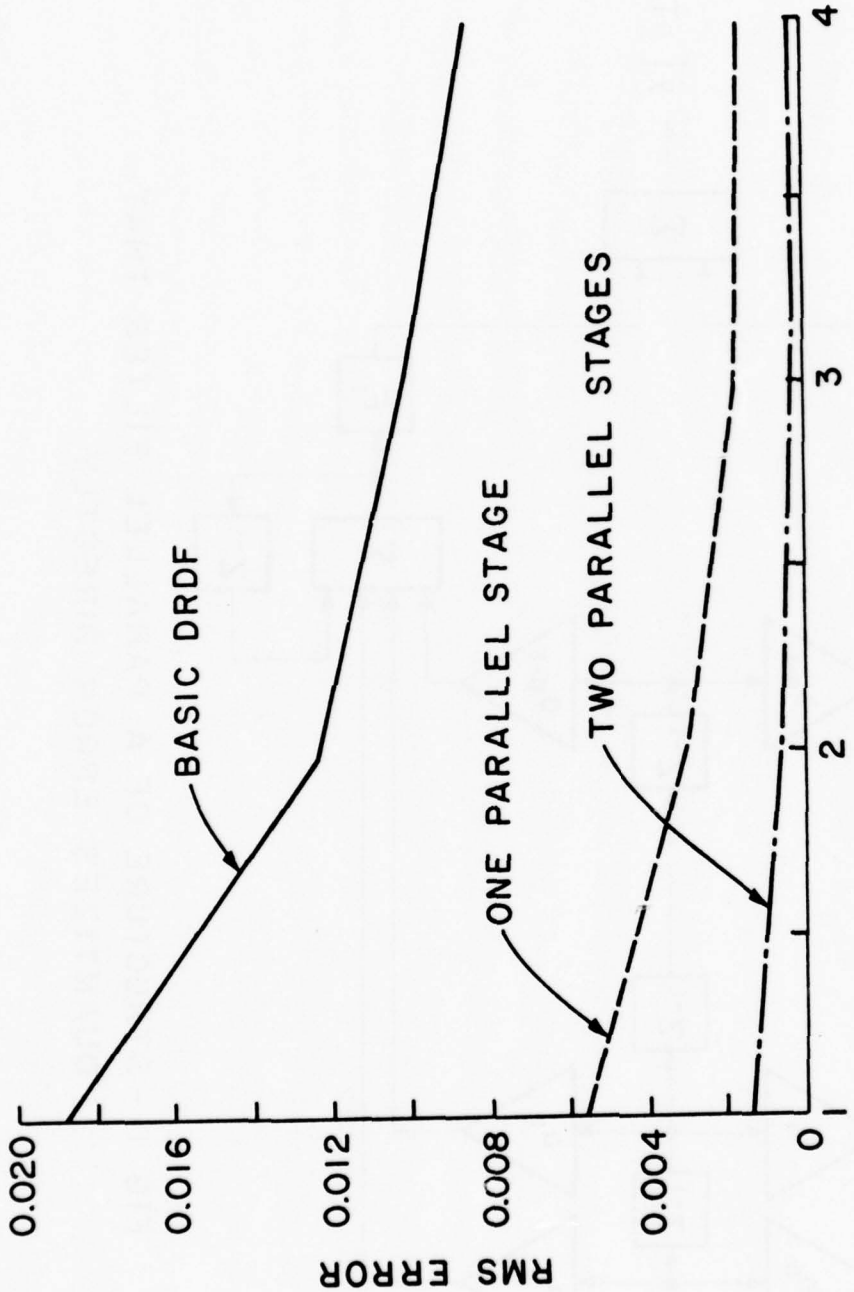


FIG. 12 - RMS ERROR vs. Km FOR PARALLEL STRUCTURE WITH ERROR SEQUENCE QUANTIZED DIRECTLY

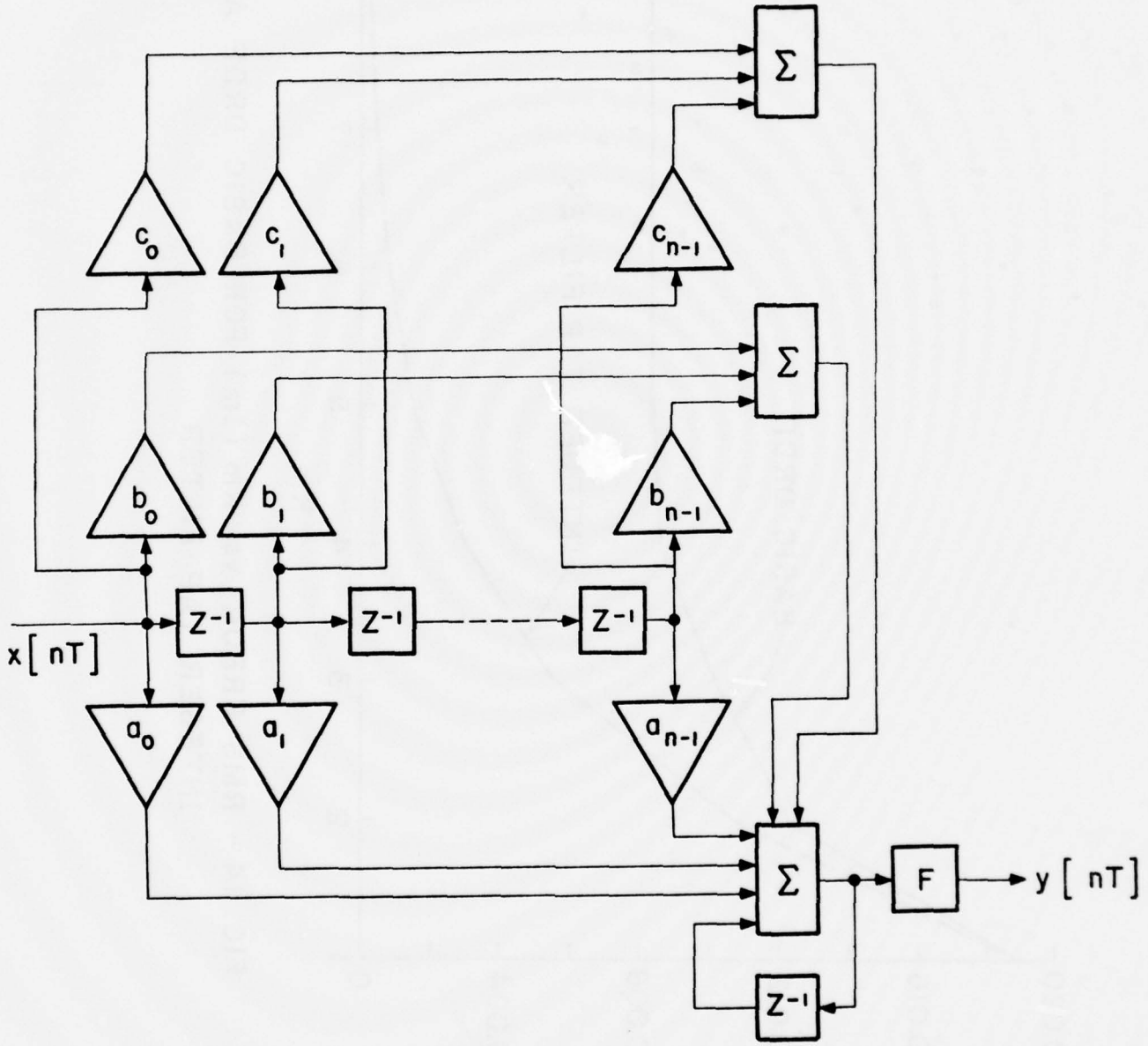


FIG. 13- INTEGER TAP FILTER STRUCTURE

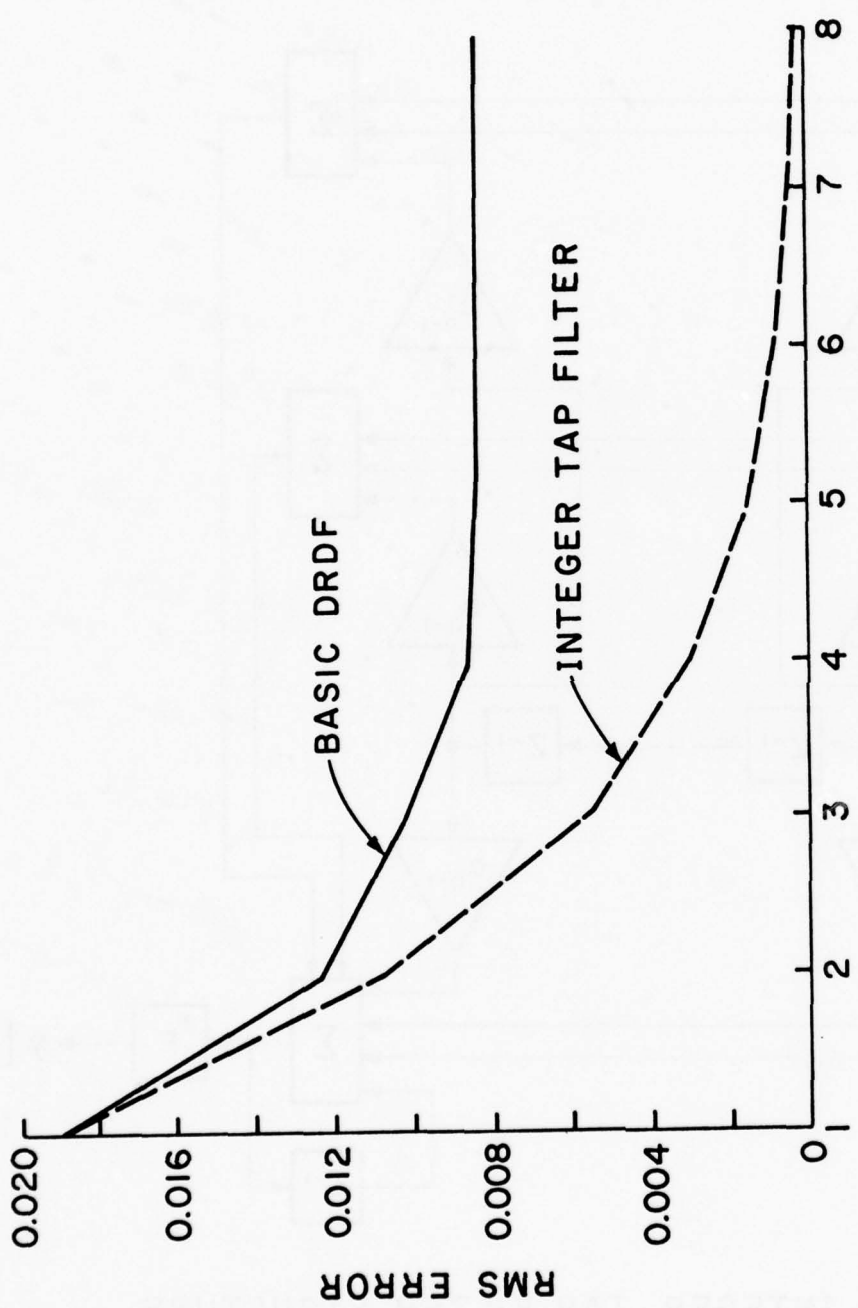


FIG. 14 - RMS ERROR vs. Km (Im) FOR BASIC DRDF AND INTEGER TAP FILTER

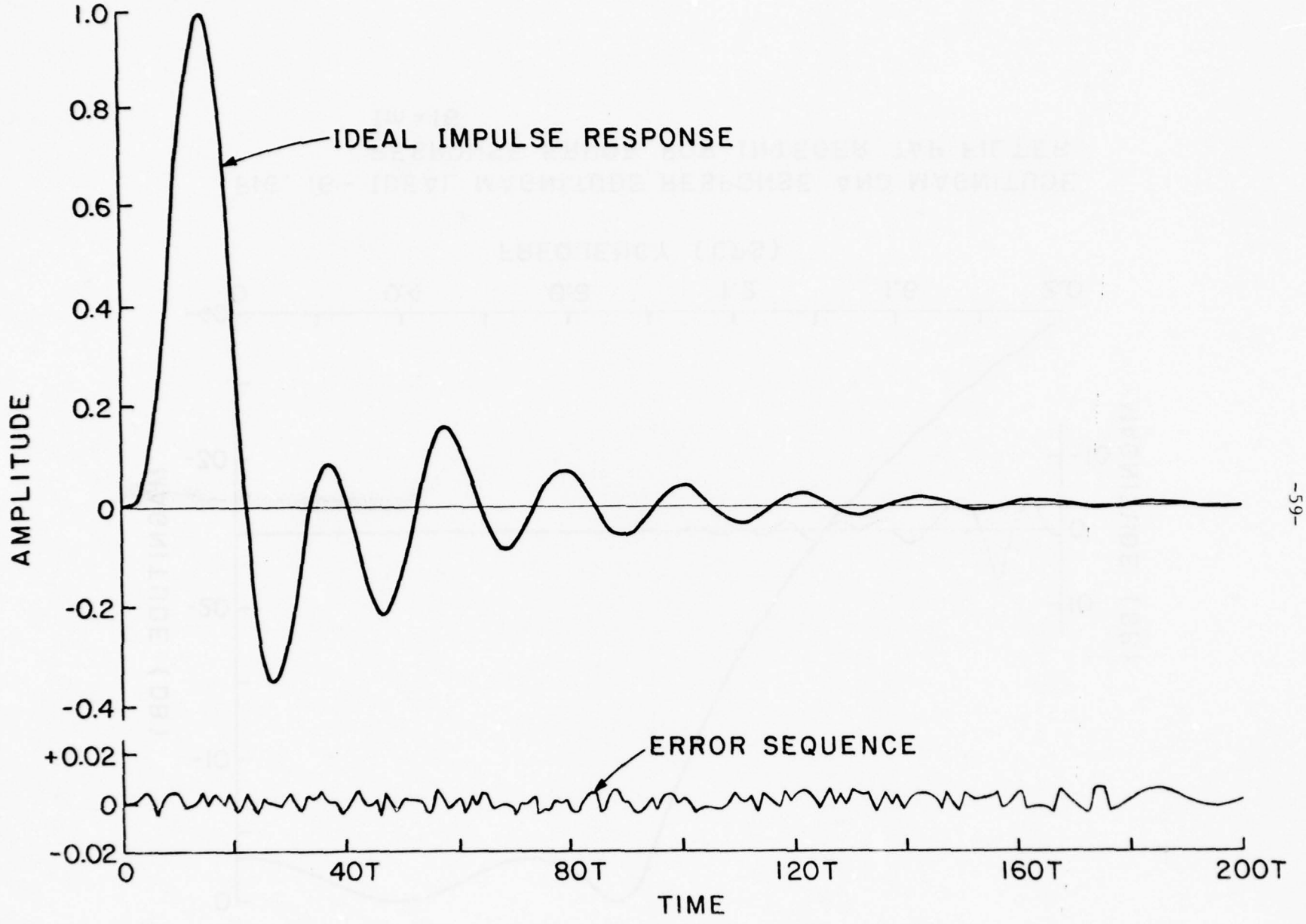


FIG. 15 - IDEAL IMPULSE RESPONSE AND ERROR SEQUENCE FOR INTEGER TAP FILTER Im = 16

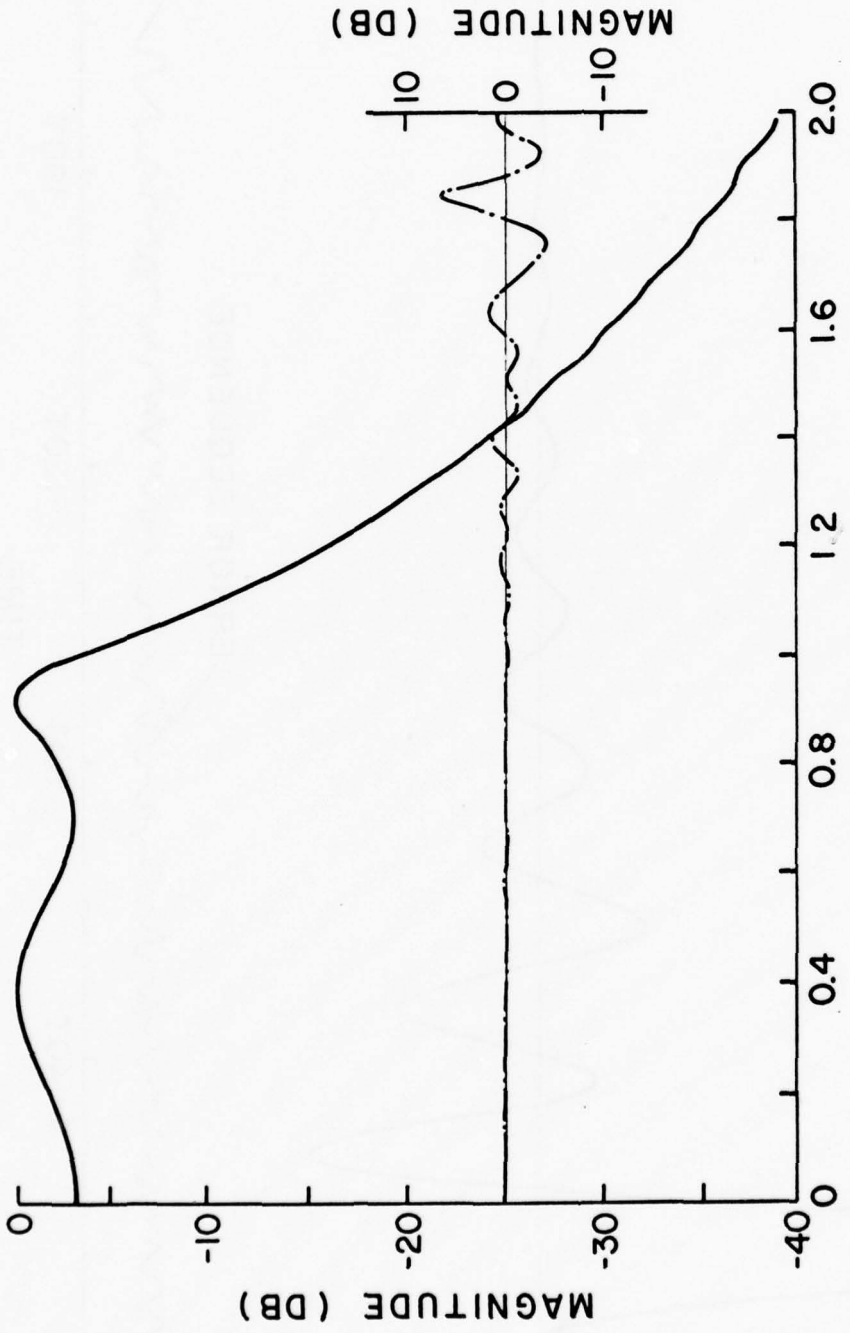


FIG. 16 - IDEAL MAGNITUDE RESPONSE AND MAGNITUDE RESPONSE ERROR FOR INTEGER TAP FILTER
Im=16

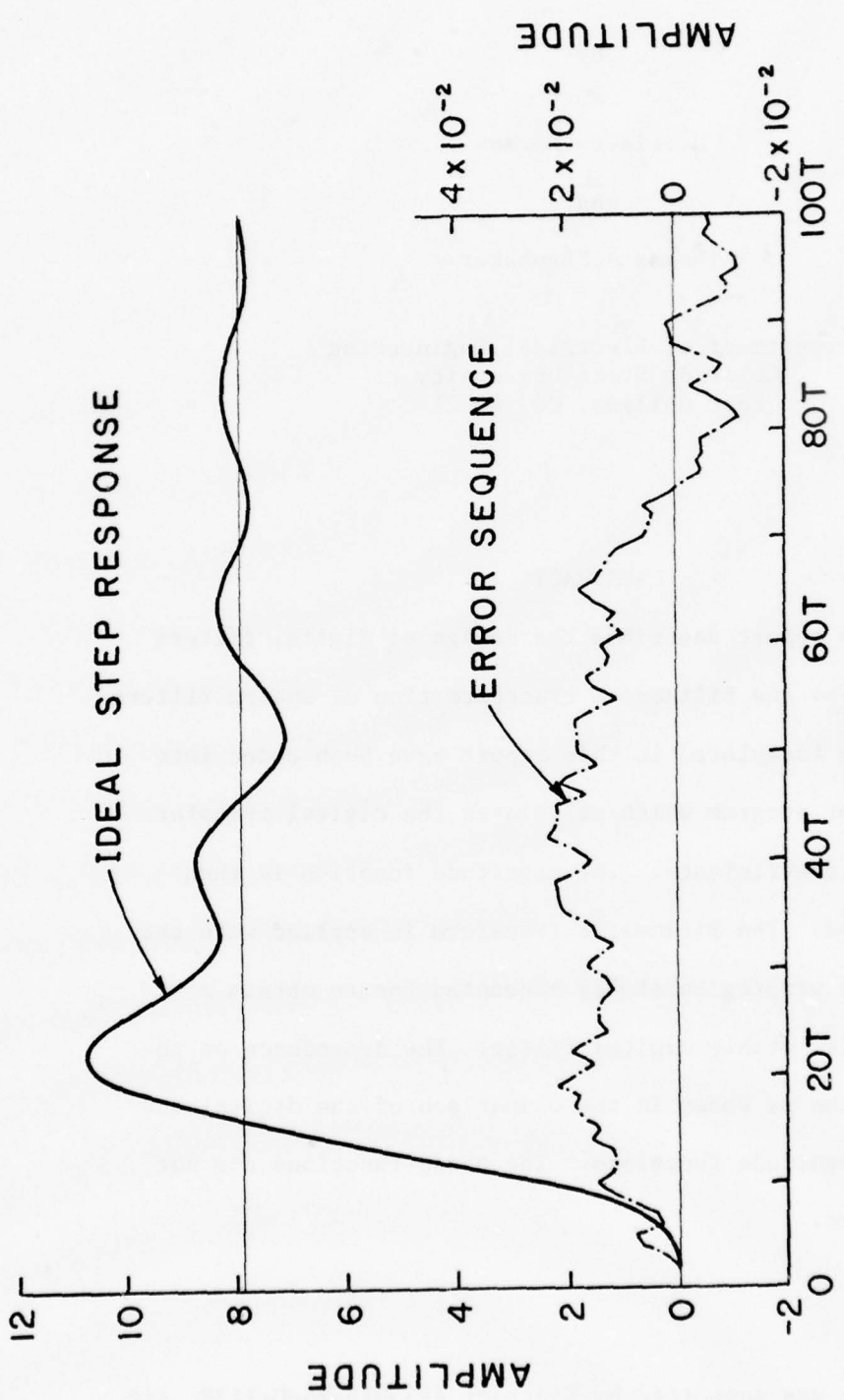


FIG. 17- IDEAL STEP RESPONSE AND RESPONSE ERROR FOR INTEGER TAP FILTER (Im=16)

Appendix B

A DIGITAL FILTER DESIGN PROGRAM UTILIZING THE BILINEAR Z TRANSFORM

by

Harriette Markos

and

Thomas A. Brubaker

Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80521

ABSTRACT

This report describes the design of digital filters obtained by the Bilinear-Z transformation of analog filters. The ideas formulated in this report have been coded into a computer program which calculates the digital transfer function coefficients. The magnitude function is then calculated. The Bilinear-Z transform is applied with the frequency warping carefully accounted for to obtain a realizable, stable digital filter. The dependence on the sample time is shown in the comparison of the digital and analog magnitude functions. The phase functions are not considered.

This work was supported by Contract #F33615-75-C-1138, Air Force Avionics Laboratory, Wright Patterson Air Force Base, Ohio 45433

I. INTRODUCTION:

In network analysis the transfer function is a fundamental characteristic of a system. The transfer function H(s) is defined by McGillem and Cooper [1] to be the ratio of the Laplace Transforms of the response or output signals to the excitation or input signals when the initial conditions are zero. If the excitation voltage is represented as $v_i(t)$ and the response voltage is represented by $v_o(t)$, then the transfer function is given by

$$H(s) = \frac{L[v_o(t)]}{L[v_i(t)]} = \frac{V_o(s)}{V_i(s)} \quad . \tag{1}$$

This H(s) transfer function is of interest because it describes the network behavior. For simplicity, it will be referred to as the transfer function.

This report and the computer program it describes are concerned with second order transfer functions

$$H(s) = \frac{C_o s^2 + C_1 s + C_2}{D_o s^2 + D_1 s + D_2} \quad . \tag{2}$$

These second order structures are fundamental building blocks for lowpass filters, highpass filters, bandpass filters, and bandstop filters. For more information on these structures see Budak [2].

Design of these filters is often done by specifying the magnitude function $H(\omega)$. In a filter design procedure the specifications on the magnitude function are usually concerned with critical frequencies such

as ω_{DC}, the frequency at DC; and ω_3, the frequency where the response has decreased by 3db from the DC value. For high Q filters, the frequency where the magnitude function peaks is typically specified. This frequency is referred to as the center frequency, ω_c. These frequencies are the critical frequencies under consideration in this report. This report is written to outline the procedure for use of the Bilinear-Z transform as applied to second order filters. The BLZ program is a computer program to perform the procedure outlined in the report.

The program accepts a second order function in s. Before applying the extended Bilinear-Z transform, the critical frequencies must be prewarped as explained in Rabiner and Gold [3]. Then, the extended Bilinear-Z transform described as

$$s = \frac{2}{T}\,\frac{Z-1}{Z+1} \tag{3}$$

is implemented and the equivalent discrete time transfer function in Z is obtained:

$$H(Z) = \frac{A_o Z^2 + A_1 Z + A_2}{Z^2 + B_1 Z + B_2} \quad . \tag{4}$$

The magnitude-squared function for (4) is obtained by setting $Z = e^{j\omega T}$ and taking the sum of the squares of the real and imaginary parts. In the design program, plots of both the analog and digital filter magnitude functions are available for presentation on a display screen.

The program recognizes real or complex poles as well as real or complex zeros. When equation (2) contains both zeros and poles, the program will consider two cascaded sections, one of zeros and one of poles

$$H(s) = \frac{D_2}{C_2} \cdot \frac{C_o s^2 + C_1 s + C_2}{D_2} \cdot \frac{C_2}{D_o s^2 + D_1 s + D_2} = K \cdot H_{ZRO}(s) \cdot H_{POL}(s) \tag{5}$$

where the cascaded sections may be considered as

$$H_{ZRO}(s) = \frac{C_o s^2 + C_1 s + C_2}{D_2} \tag{5a}$$

and

$$H_{POL}(s) = \frac{C_2}{D_o s^2 + D_1 s + D_2} \quad . \tag{5b}$$

The program is developed with specific consideration for second order, lowpass filters. However, any second order realizable structure can be processed. The program will proceed as described with consideration of the three critical frequencies: DC frequency ω_{DC}, -3db frequency ω_3, and center frequency ω_c. Proper operation of the program requires the filters to have an ω_3 or an ω_c in order to calculate the discrete time transfer function (4), with the magnitudes at the critical frequencies preserved. The magnitudes at any other frequency may not correspond due to the frequency warping.

II. DEVELOPMENT:

A. BILINEAR-Z TRANSFORM

The extended Bilinear-Z transformation is described in Rabiner and Gold [3] and McGillem and Cooper [4] as a mapping from the s-plane to the Z-plane with the following properties: the $s=j\omega$ axis is mapped onto the unit circle of the Z-plane; the left half of the s-plane ($s<o$) is mapped to the interior of the unit circle in the Z-plane; and the right half of the s-plane ($s>o$) is mapped to the exterior of the unit circle in the Z-plane. Thus, the analog transfer function H(s) given by (2) is transformed by (3) to the corresponding digital transfer function H(Z) given

by (4). The digital transfer function H(Z), evaluated at $Z=e^{j\omega T}$, is periodic in ω with period $\omega_p = \frac{2\pi}{T}$, as explained in McGillem and Cooper [4].

The extended Bilinear-Z transform is given as

$$s = \frac{2}{T}\frac{Z-1}{Z+1} \tag{3}$$

and is applied by direct substitution for s. When the $j\omega$ axis is mapped onto the unit circle in the Z-plane, there is a nonlinear relationship between the analog frequency ω and the corresponding digital frequency Ω. Rabiner and Gold [3] illustrate this nonlinearity as follows:
Using equation (3) let $s=j\omega$ and $Z=e^{j\Omega T}$ so

$$j\omega = \frac{2}{T}\frac{e^{j\Omega T}-1}{e^{j\Omega T}+1},$$

then multiplying numerator and denominator of the right side by $e^{-j\Omega T/2}$

$$j\omega = \frac{2}{T}\frac{e^{j\Omega T/2}-e^{-j\Omega T/2}}{e^{j\Omega T/2}+e^{-j\Omega T/2}}.$$

Recalling the exponential relations from Euler's equation

$$j\omega = \frac{2}{T}\, j\, \mathrm{TAN}\, \frac{\Omega T}{2}$$

and so
$$\omega = \frac{2}{T}\, \mathrm{TAN}\, \frac{\Omega T}{2}. \tag{6}$$

Therefore, the analog frequency ω is mapped to the digital frequency Ω by the relationship given in equation (6). When it is desirable to have the discrete transfer function possess the same magnitude as the analog transfer function at a particular frequency ω_D, the analog frequency must be prewarped or altered so that when the Bilinear-Z transformation is performed this prewarped frequency ω_A will map into the desired

digital frequency ω_D. The ω_A is determined by

$$\omega_A = \frac{2}{T} \text{ TAN } \frac{\omega_D T}{2} . \tag{7}$$

The relationship of this ω_D to the coefficients of (4) is determined and the prewarped coefficients for (4) are evaluated using the prewarped value for ω_A. The process is described in more detail in the following sections.

B. COMPLEX CONJUGATE ROOTS

A second order structure with complex conjugate roots is considered as

$$M_o s^2 + M_1 s + M_2 = (s-r_1)(s-r_2) \tag{8}$$

where the leading coefficient of s^2 is normalized to 1 by dividing through by M_o. The roots are complex conjugates

$$r_1 = \alpha + j\beta \tag{9a}$$

$$r_2 = \alpha - j\beta . \tag{9b}$$

Equation (8) may also be considered as

$$(s+\alpha)^2 + \beta^2 = s^2 + \frac{\omega_o}{Q} s + \omega_o^2 \tag{10}$$

where $\omega_o = \sqrt{\frac{M_2}{M_o}}$

and $Q = \frac{M_o}{M_1} \cdot \omega_o$

In equation (10), ω_o represents the distance from the origin to the location of the roots, and Q indicates the slopes of the radial lines that connect

the root locations to the origin. For $\alpha << \beta$, $Q >> 1$.

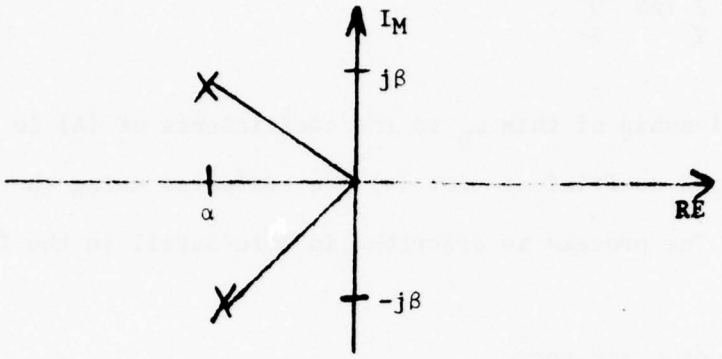


Figure 1

α, β, ω_o, Q are related as follows:

$$\alpha = \frac{\omega_o}{2Q}$$

$$\beta = \omega_o \sqrt{1 - \frac{1}{4Q^2}}$$

$$\omega_o = \sqrt{\alpha^2 + \beta^2}$$

$$Q = \frac{\sqrt{\alpha^2 + \beta^2}}{2\alpha} \quad .$$

A plot of the magnitude function $H(\omega)$ for complex conjugate poles reveals a peaking effect with the maximum value at the center frequency, ω_c. This is covered extensively by Budak [5]. The amount of peaking is indicated by Q, where Q may be considered as a measure of the ratio $\frac{\text{peak magnitude}}{\text{DC magnitude}}$. When Q is greater than five the peak will become prominent so that the center frequency ω_c may become more important than the -3db frequency ω_3. These high Q lowpass filters actually have a bandpass type structure. Therefore, it is important to preserve the appropriate critical frequency. The program utilizes a user input limiting value, Q-limit. If Q is less than or equal to Q-limit, the -3db frequency ω_3 is prewarped. But if Q is greater than Q-limit, the center frequency ω_3 is prewarped.

1. Complex Conjugate Poles

Recalling equations (2), (5b), and (10) the transfer function with complex conjugate poles can be considered as

$$H(s) = C_2\left[\frac{1}{s^2 + D_1 s + D_2}\right] = C_2\left[\frac{1}{s^2 + \frac{\omega_o}{Q} s + \omega_o^2}\right] \tag{11}$$

where the s^2 coefficient has been normalized to 1 by dividing through by D_o.

To prewarp the -3db frequency, the first step is to calculate ω_3. At ω_3 the magnitude has decreased by a factor of $\frac{1}{\sqrt{2}}$ from its DC value so

$$\frac{|H(s=0)|}{\sqrt{2}} = |H(s=j\omega_3)|$$

or using the latter form of (11) and squaring both sides

$$\frac{C_2^2}{2\omega_o^4} = \frac{C_2^2}{(\omega_o^2 - \omega_3^2)^2 + \frac{(\omega_o\,\omega_3)^2}{Q^2}} .$$

Expanding and simplifying:

$$\omega_3^4 + \left[\frac{\omega_o^2}{Q^2} - 2\omega_o^2\right]\omega_3^2 - \omega_o^4 = 0 .$$

To solve for ω_3, apply the quadratic formula employing the positive root

$$\omega_3^2 = \frac{1}{2}\left(2\omega_o^2 - \frac{\omega_o^2}{Q^2}\right) + \sqrt{\left[2\omega_o^2 - \frac{\omega_o^2}{Q^2}\right]^2 + 4\omega_o^4}$$

and recognizing that $\omega_o^2 = D_2$

$$\omega_3 = \sqrt{\frac{D_2}{2}\left[\left(2 - \frac{1}{Q^2}\right) + \sqrt{\left(2 - \frac{1}{Q^2}\right)^2 + 4}\right]} . \tag{12}$$

At this time the digital ω_3 frequency can be prewarped by equation (6) to obtain the prewarped analog -3db frequency ω_3' where

$$\omega_3' = \frac{2}{T} \text{ TAN } \frac{\omega_3 T}{2} \quad .$$

Now utilizing equation (11) and ω_3', the prewarped coefficients for H(s) can be evaluated. Recognizing first that

$$D_2 = \omega_o^2 \tag{13}$$

$$D_1 = \frac{\omega_o}{Q} = \frac{\sqrt{D_2}}{Q} \tag{14}$$

then using (12)

$$D_2' = \frac{2(\omega_3')^2}{(2 - \frac{1}{Q^2})^2 + \sqrt{(2 - \frac{1}{Q^2})^2 + 4}} \tag{15}$$

$$D_1' = \frac{\sqrt{D_2'}}{Q} \tag{16}$$

and

$$D_o' = 1 \quad . \tag{17}$$

To evaluate C_2', note the DC gain at H(s=0) is $DC = \frac{C_2}{D_2} = \frac{C_2}{\omega_o^2}$. (18)

Since it is critical to retain the DC gain, C_2' is evaluated to be

$$C_2' = DC \cdot D_2' \quad . \tag{19}$$

The prewarped coefficients determined by (15), (16), (17), and (19) now form a new transfer function H'(s) which when transformed by the Bilinear-Z transformation will exactly match in magnitude the original transfer function H(s) at the critical frequencies ω_{DC}, and ω_3.

Instead of prewarping ω_3, the center frequency ω_c could be prewarped. Again, it is first necessary to locate ω_c. Recalling that ω_c is the center frequency of the peaking effect, differential calculus is used to find the location of this extrema. Taking the derivative with respect to ω of the magnitude-squared function for (11)

$$\frac{d}{d\omega}[|H(\omega=\omega_c)|^2] = \frac{d}{d\omega}\left[\frac{C_2^2}{(\omega_o^2-\omega_c^2)^2 + (\frac{\omega_o\omega_c}{Q})^2}\right].$$

Setting the derivative equal to zero will yield the location ω_c

$$\frac{-C_2^2[2(\omega_o^2-\omega_c^2)(-2\omega_c) + 2(\frac{\omega_o\omega_c}{Q})(\frac{\omega_o}{Q})]}{[(\omega_o-\omega_c^2)^2 + \frac{(\omega_o\omega_c)^2}{Q^2}]^2} = 0$$

simplifying

$$2(\omega_o^2-\omega_c^2)(-2\omega_c) + 2\left(\frac{\omega_o\omega_c}{Q}\right)\left(\frac{\omega_o}{Q}\right) = 0$$

or

$$4\omega_c^3 - 4\omega_c\omega_o^2 + 2\omega_c\frac{\omega_o^2}{Q^2} = 0$$

but $\omega_c \neq 0$ so

$$\omega_c^2 = \omega_o^2 - \frac{\omega_o^2}{2Q^2}$$

and finally

$$\omega_c = \omega_o\sqrt{1 - \frac{1}{2Q^2}}\ . \qquad (20)$$

Note, in (20) for Q greater than 10, ω_c is very nearly ω_o.

Now, employing equation (6) the prewarped analog center frequency ω_c' is obtained:

$$\omega_c' = \frac{2}{T} \text{ TAN } \frac{\omega_c T}{2} \quad .$$

The prewarped coefficients are found from (13), (14), (17), (19), and (20) to be:

$$D_2' = \omega_o^2 = \left[\frac{\omega_c'}{\sqrt{1 - \frac{1}{2Q^2}}} \right]^2 \tag{21}$$

$$D_1' = \frac{\sqrt{D_2'}}{Q} \tag{16}$$

$$D_o' = 1 \tag{17}$$

$$C_2' = DC \cdot D_2' \tag{19}$$

As in the case of prewarping ω_3, these prewarped coefficients determine a new transfer function H'(s) which when transformed by the Bilinear-Z transformation will exactly match in magnitude the original transfer function H(s) at the critical frequencies ω_{DC}, and ω_c.

Example 1:

Consider $H(s) = \dfrac{1}{s^2 + 0.1s + 1}$, where Q = 10.

Using a sample time of T = 1.0,

$$\omega_c = 0.997497 \text{ and } \omega_3 = 1.551026.$$

The digital transfer function coefficients can be obtained on print out during execution of the BLZ program.

Figures 2 and 3 are plots of the analog and digital magnitude functions with ω_3 prewarped and ω_c prewarped, respectively. In both cases, a sample time of 1.0 second was used. The sample time affects the H(Z) magnitude function: For small T values (less than 0.1 second) the two magnitude functions are nearly the same for all ω values. As the sample time increases, the two magnitude functions begin to differ from each other for all ω values except the critical frequencies. The sample time of 1.0 second shows this variation clearly. Values of T greater than 1.0 second start to show larger differences to the point that distortion causes the two magnitude functions to differ in an unacceptable manner.

2. Complex Conjugate Zeros

Recalling equations (2), (5a), and (10) the transfer function with complex conjugate zeros can be considered as

$$H(s) = \frac{1}{D_2}\left[\frac{s^2 + C_1 s + C_2}{1}\right] = \frac{1}{D_2}\left[\frac{s^2 + \frac{\omega_o}{Q} s + \omega_o^2}{1}\right]$$

where the s^2 coefficient is normalized to 1 by division throughout by C_o.

Considerations for the -3db frequency ω_3 and the center frequency ω_c follow immediately from the development for complex conjugate poles and are not reiterated here. The effect of Q in this case is to cause a dip in the magnitude function and the ω_c is found by locating a minimum rather than a maximum.

The prewarped coefficients for H(s) are found as follows:

I. Prewarped ω_3

$$\omega_3 = \sqrt{\frac{C_2}{2}\left[\left(2 - \frac{1}{Q}_2\right) + \sqrt{\left(2 - \frac{1}{Q}_2\right)^2 + 4}\right]} \qquad (22)$$

$$\omega_3' = \frac{2}{T} \text{ TAN } \frac{\omega_3 T}{2}$$

$$C_2' = \frac{2(\omega_3')^2}{(2 - \frac{1}{Q^2}) + \sqrt{(2 - \frac{1}{Q^2})^2 + 4}} \tag{23}$$

$$C_1' = \sqrt{\frac{C_2'}{Q}} \tag{24}$$

$$C_o' = 1 \tag{25}$$

$$D_2' = \frac{C_2'}{DC} \tag{26}$$

II. Prewarped ω_c:

$$\omega_c = \omega_o \sqrt{1 - \frac{1}{2Q^2}} \tag{27}$$

$$\omega_C' = \frac{2}{T} \text{ TAN } \frac{\omega_c T}{2}$$

$$C_2' = \omega_o{}^2 = \left[\frac{\omega_C'}{(1 - \frac{1}{2Q^2})}\right]^2 \tag{28}$$

$$C_1' = \sqrt{\frac{C_2'}{Q}} \tag{24}$$

$$C_o' = 1 \tag{25}$$

$$D_2' = \frac{C_2'}{DC} \tag{26}$$

In both cases the prewarped coefficients form a new transfer function $H'(s)$ which when transformed by the Bilinear-Z transformation will exactly match the original transfer function $H(s)$ at the critical frequencies ω_{DC}, and ω_3 in case I; or ω_{DC} and ω_c in case II.

C. REAL ROOTS

A second order structure with real roots is considered as

$$M_o s^2 + M_1 s + M_2 = (s - r_1)(s - r_2) \tag{27}$$

where the roots are evaluated by application of the Quadratic formula

$$r_1 = \frac{-M_1 + \sqrt{M_1^2 - 4M_oM_2}}{2M_o} \tag{28}$$

$$r_2 = \frac{-M_1 - \sqrt{M_1^2 - 4M_oM_2}}{2M_o} . \tag{29}$$

When complex conjugate roots are under consideration it is first necessary to locate the critical ω_3 or ω_c frequencies. In the real root case this is not necessary. If each root is prewarped and the prewarped H(s) coefficients determined from the prewarped roots, the new transfer function H´(s) will yield an H(z) after the Bilinear-Z transform which matches in magnitude the original H(s) at the DC frequency ω_{DC} and at one other frequency dependent on the sample time T used. The technique is to factor the filter coefficients to locate the roots. Then each root is prewarped and the sections multiplied together before applying the Bilinear-Z transform.

The roots are found by equations (28) and (29). Then using equation (6), the prewarped roots are

$$r_1' = \frac{2}{T} \text{ TAN } \frac{r_1 T}{2} \tag{30}$$

$$r_2' = \frac{2}{T} \text{ TAN } \frac{r_2 T}{2} . \tag{31}$$

Then the prewarped coefficients are found by (27), (28), and (29) to be

$$M_o' = 1 \tag{32}$$

$$M_1' = -(r_1 + r_2) \tag{33}$$

$$M_2' = r_1 r_2 \quad . \tag{34}$$

If $M_o = 0$ there is only one real root

$$r = \frac{-M_2}{M_1} \quad . \tag{35}$$

Then $$r' = \frac{2}{T} \text{ TAN } \frac{rT}{2}$$

and so

$$M_o' = 0 \tag{36}$$

$$M_1' = 1 \tag{37}$$

$$M_2' = -r' \quad . \tag{38}$$

A special case arises when $M_2 = 0$ and $M_o = 0$, or when $M_1 = 0$ and $M_2 = 0$, causing two roots at zero, or one root at zero, respectively. In both cases TAN (0) = 0 indicating a linear relationship, hence, no prewarping is required.

1. Real Poles

Referring to equations (2) and (5) the transfer functions with real poles can be considered as

$$H(s) = C_2 \left[\frac{1}{D_o s^2 + D_1 s + D_2} \right] \tag{39}$$

where

$$D_o \leftrightarrow M_o$$

$$D_1 \leftrightarrow M_1$$

$$D_2 \leftrightarrow M_2$$

in the previous development.

The roots (poles) and prewarped coefficients can be evaluated as described before with C_2' being determined from

$$DC = \frac{C_2}{D_2}$$

hence

$$C_2' = DC \cdot D_2' \quad . \tag{40}$$

2. Real Zeros

Referring to equations (2) and (5) the transfer function with real zeros can be considered as:

$$H(s) = \frac{1}{D_2}\left[C_o s^2 + C_1 s + C_2\right]$$

where

$$C_o \leftrightarrow M_o$$

$$C_1 \leftrightarrow M_1$$

$$C_2 \leftrightarrow M_2$$

in the previous development.

The roots (zeros) and prewarped coefficients can be evaluated as described with D_2' defined by (40) to be $D_2' = \frac{C_2'}{DC}$. (41)

Example 2:

Consider

$$H(s) = \frac{50}{s^2 + 15s + 50}$$

with poles at 5.0 and 10.0.

Figure 4 is a plot of the analog and digital magnitude functions for this lowpass, real pole filter using two sample times. For the sample

time of 0.3 sec., the analog and digital magnitude functions agree exactly at ω_{DC}, and at an intermediate frequency ω_1. However, for the sample time of 0.2 sec., the analog and digital magnitude functions agree at ω_{DC}, and at a frequency ω_2 different from ω_1. When evaluating real root structures, the sample time has a direct effect on the critical frequency value because the roots are prewarped, not the critical frequency. Note that for first order sections prewarping the root is equivalent to prewarping the -3db frequency. For the first order sections in cascade the net -3db frequency will not match because it was not prewarped.

D. H(Z) TRANSFER FUNCTION

After the prewarping is accomplished, the digital transfer function is evaluated. The transformation is made by direct substitution of (3) into the transfer function of prewarped coefficients giving

$$H(Z) = H'(s = \frac{2}{T}\frac{Z-1}{Z+1}) = \frac{C'_o\left[\frac{2}{T}\frac{Z-1}{Z+1}\right]^2 + C'_1\left[\frac{2}{T}\frac{Z-1}{Z+1}\right] + C'_2}{D'_o\left[\frac{2}{T}\frac{Z-1}{Z+1}\right]^2 + D'_1\left[\frac{2}{T}\frac{Z-1}{Z+1}\right] + D'_2} .$$

Multiplying numerator and denominator by $[T(Z+1)]^2$

$$H(Z) = \frac{C'_o[2(Z-1)]^2 + C'_1[2(Z-1)][T(Z+1)] + C'_o[T(z+1)]^2}{D'_o[2(Z-1)]^2 + D'_1[2(Z-1)][T(Z+1)] + D'_o[T(Z+1)]^2} .$$

Expanding, and then collecting terms

$$H(Z) = \frac{[4C'_o + 2TC'_1 + T^2C'_2]Z^2 + [2T^2C'_2 - 8c'_o]Z + [4C'_o - 2TC'_1 + T^2C'_2]}{[4D'_o + 2TD'_1 + T^2D'_2]Z^2 + [2T^2D'_2 - 8D'_o]Z + [4D'_o - 2TD'_1 + T^2D'_2]} . \quad (41)$$

Recalling equation (4)

$$H(Z) = \frac{A_oZ^2 + A_1Z + A_2}{Z^2 + B_1Z + B_2} , \quad (4)$$

where

$$DEN = 4D_o' + 2TD_1' + T^2D_2'$$

and

$$A_o = [4C_o' + 2TC_1' + T^2C_2']/DEN$$
$$A_1 = [2T^2C_2' - 8C_o']/DEN$$
$$A_2 = [4C_o' - 2TC_1' + T^2C_2']/DEN$$
$$B_1 = [2T^2D_2' - 8D_o']/DEN$$
$$B_2 = [4D_o' - 2TD_1' + T^2D_2']/DEN \quad .$$

E. CALCULATION OF MAGNITUDE FUNCTIONS

1. Analog Transfer Function H(s)

The magnitude function for the analog transfer function is obtained by setting $s=j\omega$ in (2) and finding the square root of the sum of the squares of the imaginary and real parts:

$$H_s(\omega) = |H(s=j\omega)| = \left| \frac{C_o (j\omega)^2 + C_1 (j\omega) + C_2}{D_o (j\omega)^2 + D_1 (j\omega) + D_2} \right|$$

and since $j = \sqrt{-1}$

$$H_s(\omega) = \left| \frac{(C_2 - C_o\omega^2) + j(C_1\omega)}{(D_2 - D_o\omega^2) + j(D_1\omega)} \right| .$$

Then the magnitude of a quotient is the quotient of the magnitudes

$$H_s(\omega) = \frac{\sqrt{(C_2 - C_o\omega^2)^2 + (C_1\omega)^2}}{\sqrt{(D_2 - D_o\omega^2)^2 + (D_1\omega)^2}}$$

or equivalently

$$H_s(\omega) = \sqrt{\frac{(C_2 - C_o\omega^2)^2 + (C_1\omega)^2}{(D_2 - D_o\omega^2)^2 + (D_1\omega)^2}} \quad . \tag{42}$$

2. Digital Transfer Function H(Z)

The magnitude function for the digital transfer function requires special consideration of the original analog transfer function. If H(s)

is composed of one section of either poles or zeros, the digital magnitude function is found using the prewarped $H'(s)$ transfer function. But, if $H(s)$ is composed of two sections in cascade as in equation (5), there are two transfer functions to be considered: the transfer function of prewarped coefficients for the pole section, $H'_{POL}(s)$; and the transfer function of prewarped coefficients for the zero section, $H'_{ZRO}(s)$. The resultant total transfer function is the product $H'_{POL}(s) \cdot H'_{ZRO}(s)$. The technique used here is to transform each section to obtain the digital transfer function in cascades $H'_{POL}(Z)$ and $H'_{ZRO}(Z)$, then evaluate the digital magnitude functions for each section, and the resultant total digital magnitude function is the product $H_{POL}(\omega) \cdot H_{ZRO}(\omega)$. In both cases the general technique to obtain the magnitude function $H(\omega)$ from a digital transfer function $H(Z)$ is to let $Z = e^{j\omega T}$, apply Euler's formula, and find the square root of the sum of the squares of the imaginary and real parts

$$H_Z(\omega) = \left|H(Z = e^{j\omega T})\right| = \left| \frac{A_o\, e^{j2\omega T} + A_1\, e^{j\omega T} + A_2}{e^{j2\omega T} + B_1\, e^{j\omega T} + B_2} \right| .$$

Recalling Euler's formula

$$e^{j\omega T} = \cos \omega T + j \sin \omega T$$

and combining real and imaginary parts yields

$$H_Z(\omega) = \left| \frac{[A_o \cos 2\omega T + A_1 \cos \omega T + A_2] + j[A_o \sin 2\omega T + A_1 \sin \omega T]}{[\ \cos 2\omega T + B_1 \cos \omega T + B_2] + j[\ \sin 2\omega T + B_1 \sin \omega T]} \right|$$

and since the quotient of the magnitudes is the magnitude of the quotients

$$H_Z(\omega) = \sqrt{\frac{[A_o \cos 2\omega T + A_1 \cos \omega T + A_2]^2 + [A_o \sin 2\omega T + A_1 \sin \omega T]^2}{[\ \cos 2\omega T + B_1 \cos \omega T + B_2]^2 + [\ \sin 2\omega T + B_1 \sin \omega T]^2}} \quad . \tag{43}$$

In the case of the analog transfer function being composed of two sections in cascade, the resultant total digital magnitude function is

found as the product of the magnitude functions of each section. This multiplication effectively destroys the correspondence of the critical 3db or center frequencies for each section, resulting instead in a new critical frequency. Only if the critical frequency of prewarping is identical in both sections will the resultant critical frequency be the same. However, if critical frequencies for the total transfer function are prewarped, such as in Butterworth filter design, then the individual sections are not treated separately in terms of prewarping. This allows the critical frequencies to be handled directly from input to output. This form of design has been treated in two separate reports now being revised. These are "A Fortran IV Design Program for Butterworth and Chebychev Band-Pass and Band-Stop Digital Filters", and "A Fortran IV Design Program for Low Pass Butterworth and Chebychev Digital Filters". The revisions will be completed in early August, 1976.

III. OPERATION OF THE PROGRAM

The principles developed so far are utilized in the BLZ program. Written in DEC Fortran IV, this program accepts a second order analog transfer function, performs the necessary prewarping to match critical frequencies, implements a Bilinear Z transform, and determines the digital transfer function by the methods explained in section II.

A. INITIALIZATION

The operation of the BLZ program assumes a high degree of user interaction via an external teletype or keyboard device. The user is asked to supply, through input, the following initialization factors:

1. The program will request input specifying real poles or complex poles, real zeros or complex zeros. Input is in the form of 1 for yes and 0 for no. As the program receives each response a flag is set to each affirmative

reply. If flags are set for both zero and pole sections a special flag is set to indicate two sections in cascade.

2. The program will request the range of ω values the magnitude functions are to be plotted over. A minimum ω and a maximum ω are input in 2I3 format. This option facilitates the opportunity of investigating any specific region of the magnitude function.
3. The program will then request the total number of points to be plotted. Input is in I3 format, however, only values up to 500 are allowed under the current dimension allotments. If more than 500 points are desired, the dimension statement for the arrays used must be altered to a value greater than or equal to the total number of points plus two.
4. The program will request a sample time T in seconds and the coefficients of the analog transfer function H(s). As stated before:

$$H(s) = \frac{C_o s^2 + C_1 s + C_2}{D_o s^2 + D_1 s + D_2} \quad . \tag{2}$$

The format statement here allows values up to F9.5. The leading coefficients need not be normalized to 1.

NOTE: All input values must be presented in the format specified or with decimal points and commas included in all relevant positions.

B. EXECUTION

After the initialization factors are received, the program proceeds to calculate the analog magnitude function defined by equation (42).

The user may have the computer output the DC value before responding to a request to display a plot of $H_s(\omega)$ versus ω to be generated on a display screen. The IDIOT subroutine is called for this purpose, and is included in the program package as explained under the section on software.

The program control now advances into the transformational section of its operation. If the special flag to indicate two sections in cascade is set, the program will evaluate a magnitude function $H_{POL}(\omega)$ for the pole section, then a magnitude function $H_{ZRO}(\omega)$ for the zero section. The resultant total digital magnitude function $H_Z(\omega)$ is the product of $H_{POL}(\omega)$ and $H_{ZRO}(\omega)$. If the special flag is not set, the program checks first for the pole flags and then for zero flags. In all cases, when a set flag is encountered, the control branches to the appropriate subroutine for the set flag. The subroutines serve to evaluate the prewarped analog transfer function coefficients as described in section II. In the case of complex conjugate poles or complex conjugate zeros, the user is requested to input a limit value for Q to determine prewarping of ω_3 or ω_c. In all cases, the user will have the option to print out such information as the -3db frequency, the center frequency, or the prewarped roots.

As each prewarping subroutine is completed, control returns to the main program. The Bilinear-Z transform is executed as described and the digital transfer function is obtained. The magnitude function defined in (43) is evaluated for the ω range indicated by the initialization. A plot of this magnitude function is available, and if the user so indicates, the digital magnitude function will then be superimposed on the plot of the analog transfer function. If the user so desires the program will now output the minimum and maximum values of the magnitude

function.

At this time the user is afforded the option to alter the sample time T. If he responds affirmatively, the program requests a new T value to be input. The control then branches to inquire if the user desires to replot the analog magnitude function. If the user responds negatively, the present display screen contents are retained and the next plot will be superimposed with these same plots. If the user responds affirmatively, the screen is cleared and the analog magnitude function is displayed and the program proceeds as before.

In the case of two sections in cascade the program evaluates for each section, then offers a plot of the resultant total digital magnitude function. The options for the minimum and maximum magnitude values and a new sample time T are offered as described before.

When the user is satisfied with the sample time used and rejects the option to alter T, the program continues with an offer to printout the table of values of the magnitude functions over the range of ω values.

At this point, in the case of two sections in cascade, the user is offered the opportunity to display plots of the magnitudes of each section individually.

Two more printout options are available at this time: the prewarped normalized analog transfer function coefficients; and the digital transfer function coefficients. The final option is to run the program over again. An affirmative response directs control to the initialization portion of the program, while a negative response terminates the program.

C. HARDWARE

The BLZ program is written in DOS Fortran version 9.02. It was developed on a PDP-11/20 with a DOS/BATCH operating system. The function

plots were obtained using a GT40 graphics display terminal, with hardcopy plots available on a Houston Instruments Plotter interfaced to the GT40. All printed results were available using a Centronics line printer. The program is written to be easily modified for use with systems of similar configuration.

D. SOFTWARE

Operation of the BLZ program requires considerable user interaction for data input. The program writes instructions and questions to unit 6. The user responds with the appropriate data which is read from unit 4. These two units must be assigned to appropriate output and input devices at run time. Additional output is available to unit 5, which must also be assigned to a device at run time. The program was written and tested with units 4 and 6 assigned to a teletype keyboard and unit 5 assigned to a line printer. A sample run using these devices is provided in figure 5.

The BLZ program is composed of a main block with four subroutines to perform the prewarping:

RZRO
RPOL
CZRO
CPOL

and a subfunction TAN to calculate the trigonometric function tangent. The program also calls the following routines from library files:

FLOAT	IDIOT
SQRT	RANGE
ABS	
COS	
SIN	

The routines on the left are called from the DOS/BATCH FTNLIB. The routines on the right are required for plotting. They are called from the PLTLIB file and are included in the program package.

The IDIOT subroutine was written for plotting applications on the GT40 graphics display terminal. Any subroutines required by IDIOT are included in PLTLIB, along with the GT40 plotting routine (PDP-11 assembly language MACRO). Explanation and documentation of the use of IDIOT can be obtained by listing the program. All of the required plotting software for the GT40 has been included with the program package, or the user may incorporate his own plotting routines.

Recommended set up procedures for use of the BLZ program are as follows:

1. Compile BLZ program. This assures system compatibility.
2. Compile PLTLIB.SRC.
3. Assemble SENDGT. This is the plotting routine to run the GT40.
4. Build the subroutine library PLTLIB.OBJ including
   1. PLTLIB.OBJ
   2. SENDGT.OBJ

   in that order.
5. Assemble and link PLOTGT.MAC for latter use in GT40 plotting.
6. Link together BLZ, PLTLIB, FTNLIB.
7. Load PLOTGT.LDA into GT40 and start it running.
8. Run BLZ (any changes on I/O devices may be made now by assignment statements for the devices 4, 5, and 6 as explained previously.)

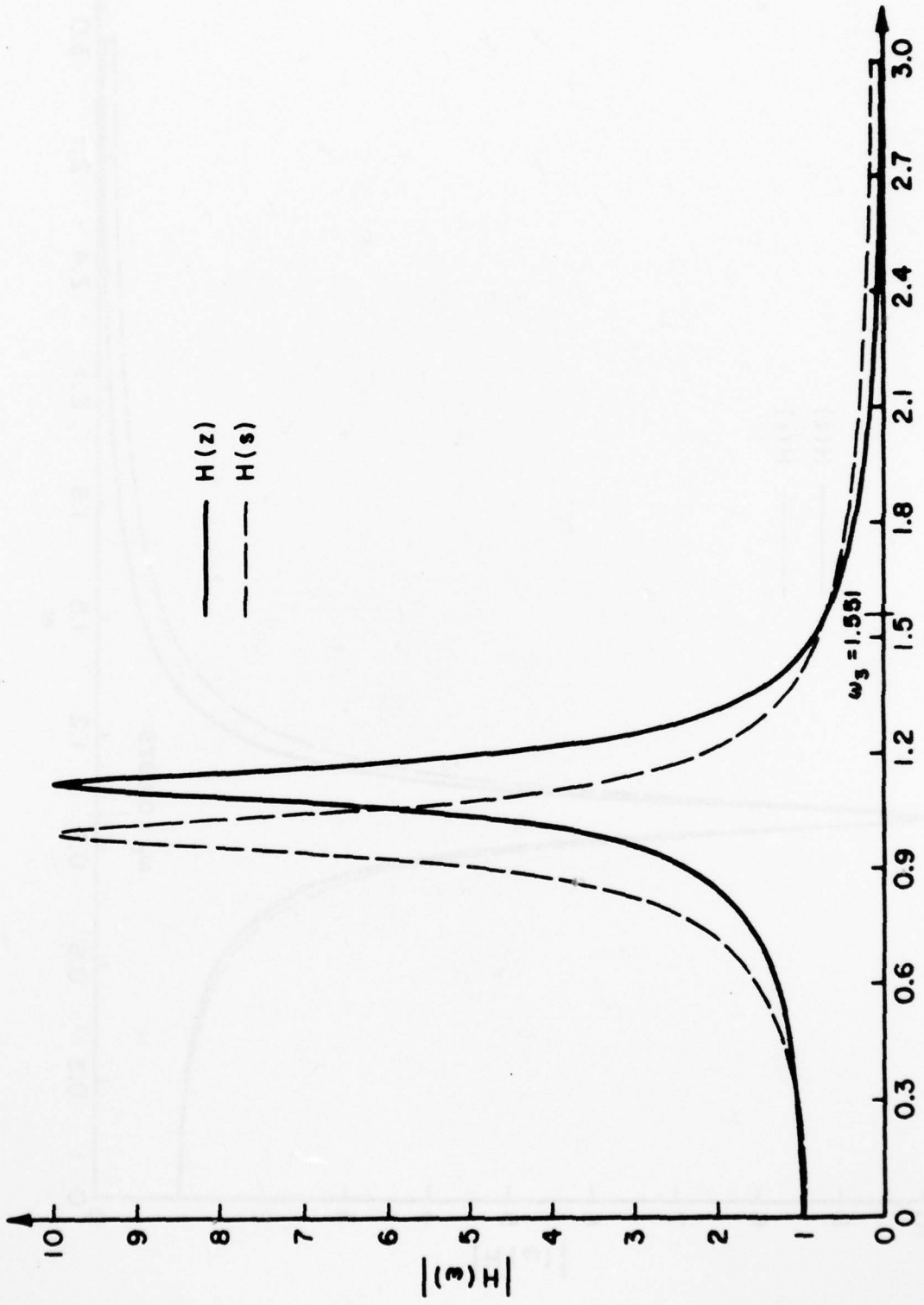


Figure 2

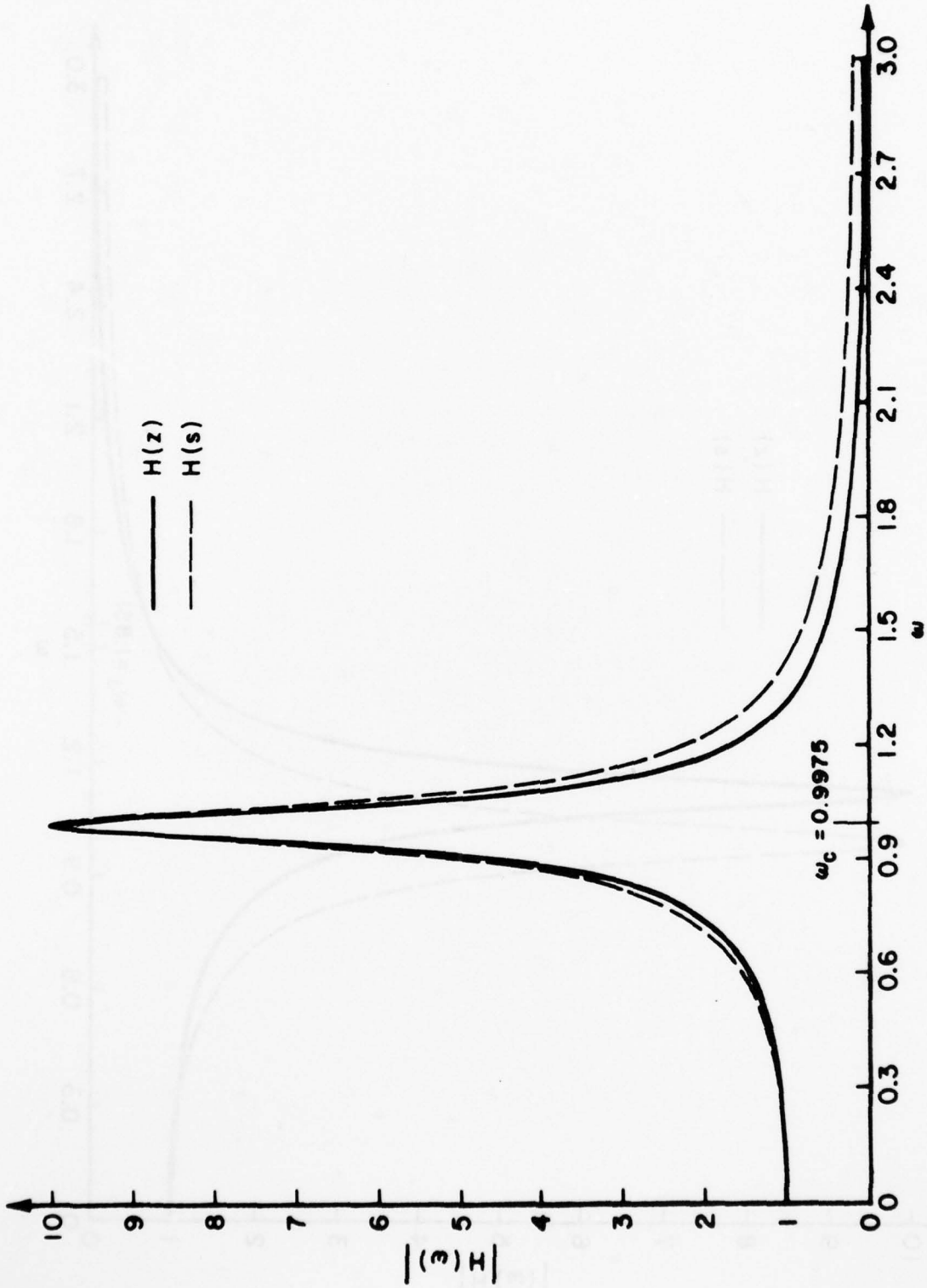


Figure 3

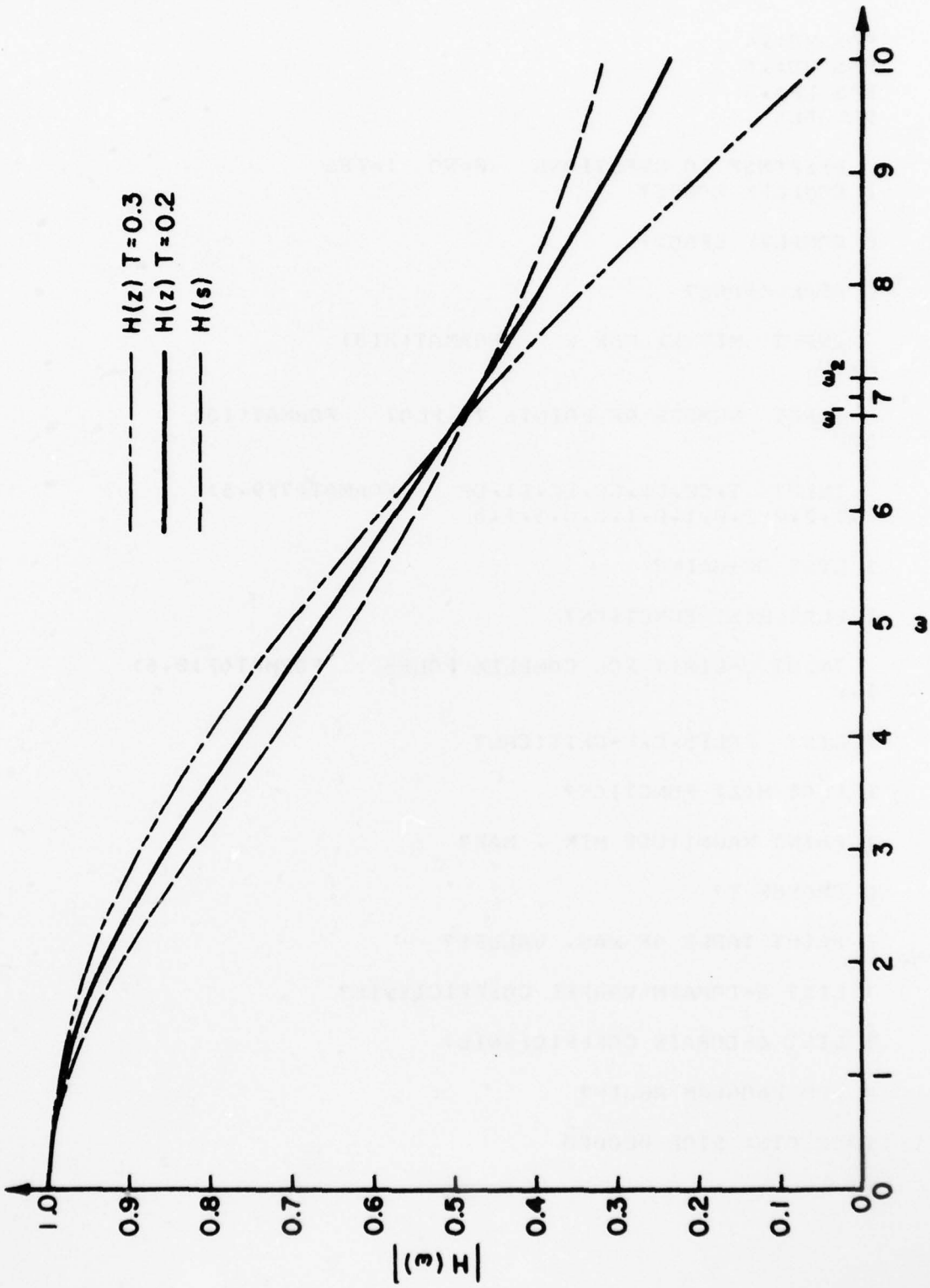

H(z) T=0.3
H(z) T=0.2
H(s)
|H(ω)|
ω
ω1 ω2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1
2
3
4
5
6
7
8
9
10

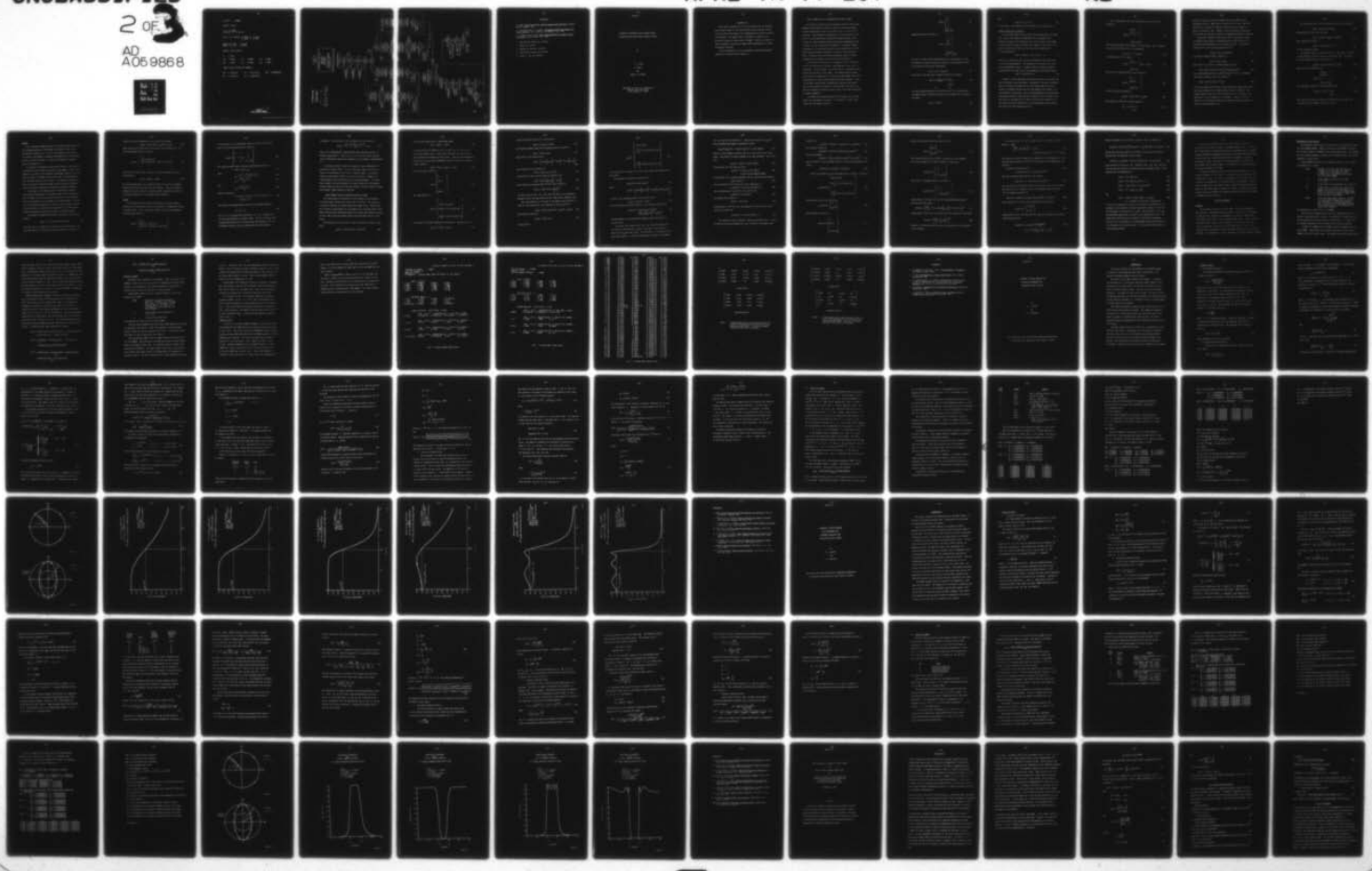
Figure 4

```
$AS KB:,4
$AS KB:,6
$AS LP:,5
$RU PLZ

  RESPONSE TO QUESTIONS    0=NO   1=YES
1 COMPLEX POLES?

0 COMPLEX ZEROS?

0 REAL ZEROS?

  INPUT  MIN W, MAX W     FORMAT(2I3)
0,10

  INPUT  NUMBER OF POINTS TO PLOT    FORMAT(I3)
500

  INPUT  T,C0,C1,C2,D0,D1,D2     FORMAT(7F9.5)
0.1,0.0,0.0,1.0,1.0,0.1,1.0

1 LIST DC-GAIN?

1 PLOT H(S) FUNCTION?

  INPUT Q-LIMIT FOR COMPLEX POLES     FORMAT(F10.6)
1.0

1 LIST  POLES,Q,W-CRITICAL?

1 PLOT H(Z) FUNCTION?

1 PRINT MAGNITUDE MIN , MAX?

0 CHANGE T?

0 PRINT TABLE OF MAG. VALUES?

1 LIST S-DOMAIN WARPED COEFFICIENTS?

1 LIST Z-DOMAIN COEFFICIENTS?

0 RUN PROGRAM AGAIN?

I35C OTS: STOP 000000

$
```

FIGURE 5
TTY Keyboard Output

AD-A059 868 COLORADO STATE UNIV FORT COLLINS F/G 9/1
DEVELOPMENT OF IMPROVED DESIGN METHODS FOR DIGITAL FILTERING SY--ETC(U)
NOV 77 T A BRUBAKER F33615-75-C-1138
UNCLASSIFIED AFAL-TR-77-207 NL

2 OF 3
AD A059868



```
DC GAIN     1.0000

COMPLEX POLES

Q IS  10.0000
W-CENTER IS  0.997497

POLES IN S-DOMAIN -0.0500 +J  0.9987
                  -0.0500 -J  0.9987

MAGNITUDE MAX=    9.99983
MAGNITUDE MIN=    0.00854

WARPED COEFFICIENTS

T=    0.1000

C0=    0.0000       C1=    0.0000       C2=    1.0017

D0=    1.0000       D1=    0.1001       D2=    1.0017

COEFFICIENTS AFTER BILINEAR-Z

A0=  0.2485E-02       A1=  0.4971E-02       A2=  0.2485E-02

B1= -0.1980E 01       B2=  0.9901E 00
```

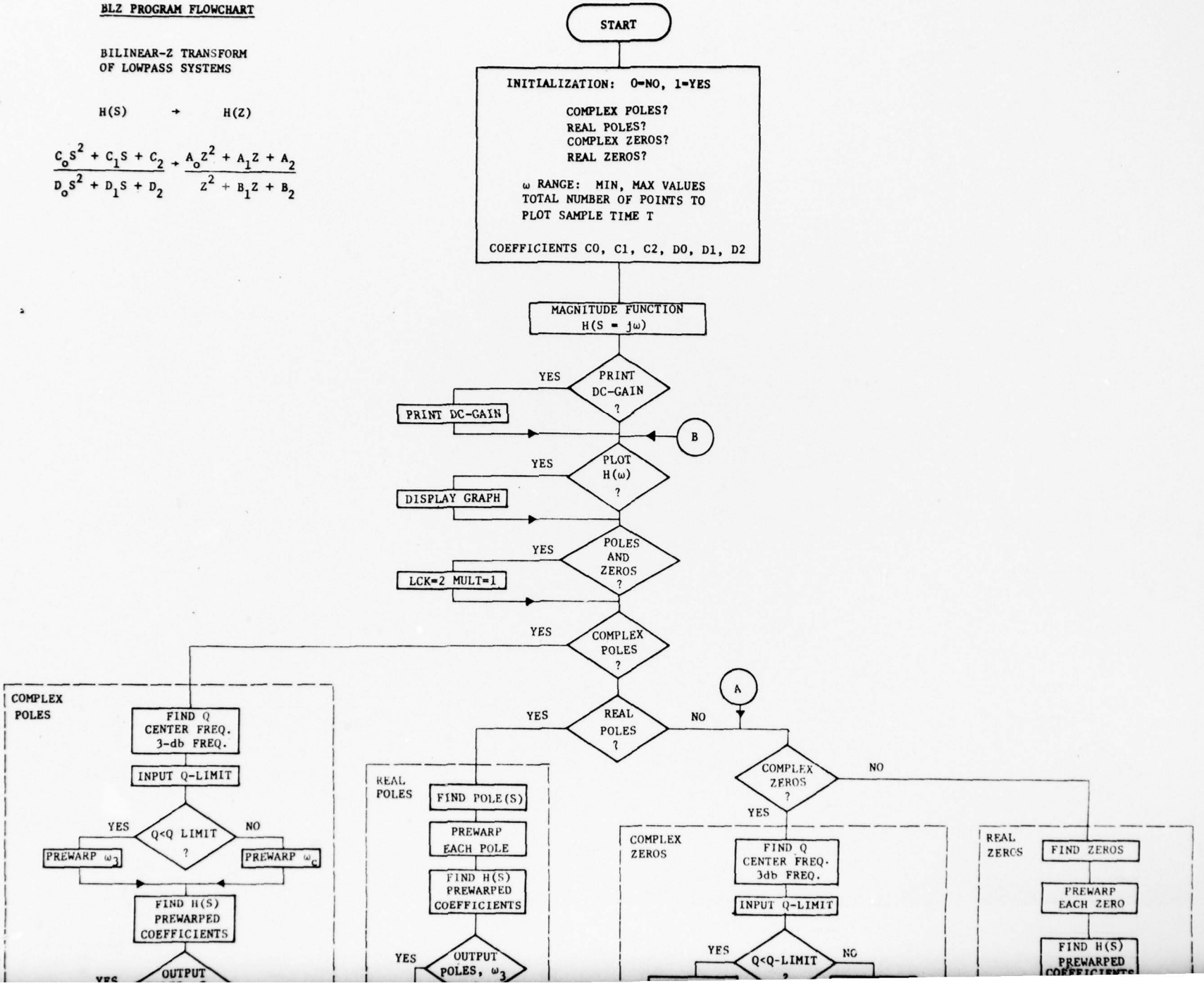
FIGURE 5
Line Printer Output

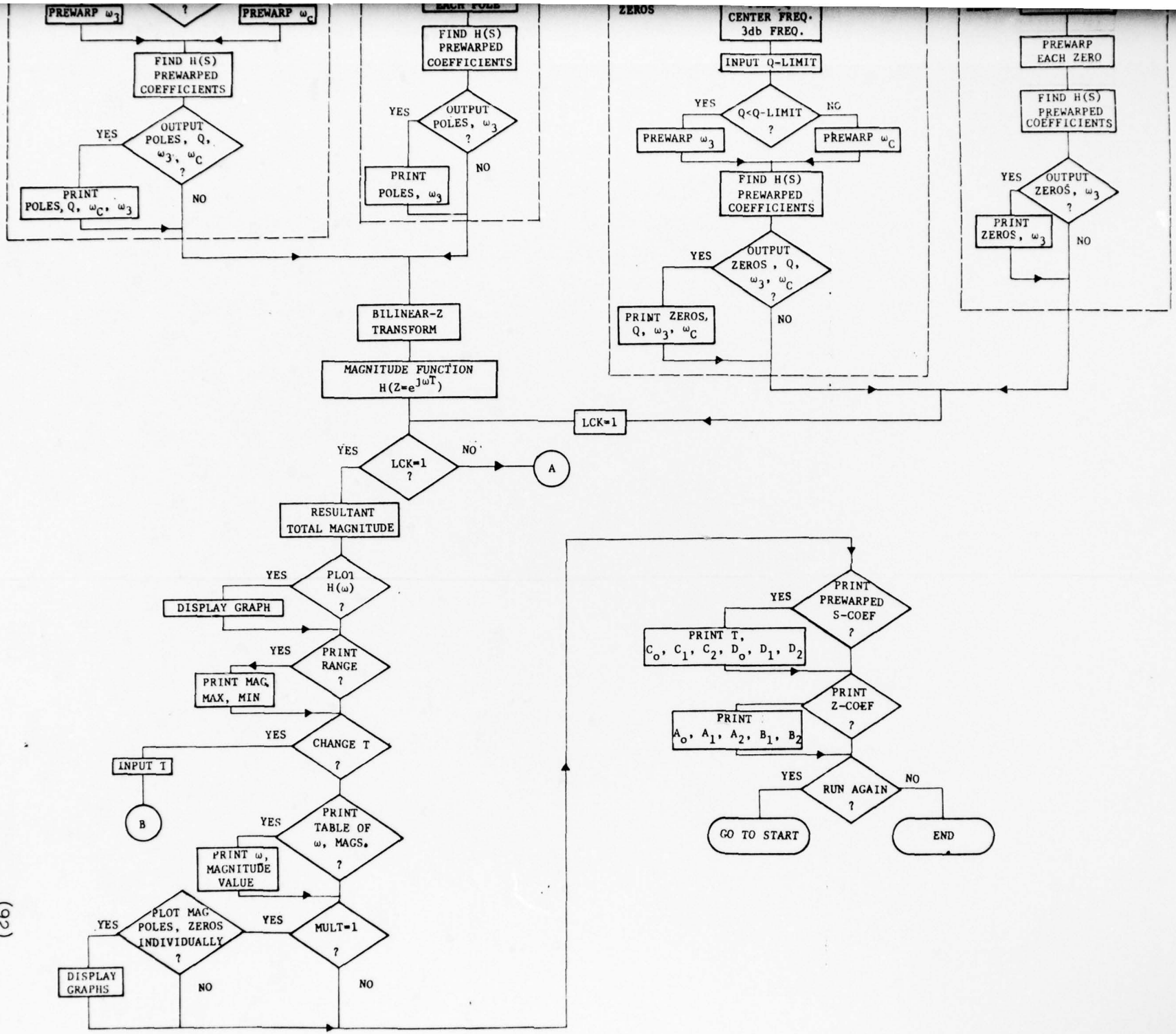
BLZ PROGRAM FLOWCHART

BILINEAR-Z TRANSFORM
OF LOWPASS SYSTEMS

$H(S) \rightarrow H(Z)$

$$\frac{C_o S^2 + C_1 S + C_2}{D_o S^2 + D_1 S + D_2} \rightarrow \frac{A_o Z^2 + A_1 Z + A_2}{Z^2 + B_1 Z + B_2}$$




PREWARP ω_3
PREWARP ω_C
FIND H(S) PREWARPED COEFFICIENTS
YES
OUTPUT POLES, Q, ω_3, ω_C ?
NO
PRINT POLES, Q, ω_C, ω_3
FIND H(S) PREWARPED COEFFICIENTS
YES
OUTPUT POLES, ω_3 ?
NO
PRINT POLES, ω_3
ZEROS
CENTER FREQ.
3db FREQ.
INPUT Q-LIMIT
YES
Q<Q-LIMIT ?
NO
PREWARP ω_3
PREWARP ω_C
FIND H(S) PREWARPED COEFFICIENTS
YES
OUTPUT ZEROS , Q, ω_3, ω_C ?
NO
PRINT ZEROS, Q, ω_3, ω_C
PREWARP EACH ZERO
FIND H(S) PREWARPED COEFFICIENTS
YES
OUTPUT ZEROS, ω_3 ?
NO
PRINT ZEROS, ω_3
BILINEAR-Z TRANSFORM
MAGNITUDE FUNCTION $H(Z=e^{j\omega T})$
LCK=1
YES
LCK=1 ?
NO
A
RESULTANT TOTAL MAGNITUDE
YES
PLOT $H(\omega)$?
DISPLAY GRAPH
YES
PRINT RANGE ?
PRINT MAG, MAX, MIN
YES
CHANGE T ?
INPUT T
B
YES
PRINT TABLE OF ω, MAGS. ?
PRINT ω, MAGNITUDE VALUE
YES
PLOT MAG. POLES, ZEROS INDIVIDUALLY ?
YES
MULT=1 ?
YES
DISPLAY GRAPHS
NO
NO
YES
PRINT PREWARPED S-COEF ?
PRINT T, C_o, C_1, C_2, D_o, D_1, D_2
YES
PRINT Z-COEF ?
PRINT A_o, A_1, A_2, B_1, B_2
YES
RUN AGAIN ?
NO
GO TO START
END

REFERENCES

A. Budak, Passive and Active Network Analysis and Synthesis, Houghton Mifflin Company, 1974.

C. D. McGillem and G. R. Cooper, Continuous and Discrete Signal and System Analysis, Holt, Rinehart, and Winston, Inc., 1974.

L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., 1975.

1. McGillem and Cooper, pp. 235-238.
2. Budak, pp. 533-644.
3. Rabiner and Gold, pp. 220-224.
4. McGillem and Cooper, pp. 301-302.
5. Budak, p. 146, pp. 319-321.

Appendix C

Programs for Weighted Least Squares Design of Nonrecursive and Recursive Digital Filters

by

F. L. Mann

E. Reuss

and

Thomas A. Brubaker

Department of Electrical Engineering
Colorado State University
Fort Collins, CO 80523

INTRODUCTION

This report describes how to use two programs for the weighted least squares design of nonrecursive and recursive digital filters. First the theoretical aspects are considered and the design equations are developed. The signal model in this work is assumed to be a polynomial because the state model is simple. However, the theory is easily extended to include any signal model represented by a linear differential equation.

Then the operation of the two programs is described along with examples to illustrate their operation.

LEAST SQUARES DESIGN OF NONRECURSIVE DIGITAL FILTERS

The design of nonrecursive and recursive digital filters using weighted least squares is based on a model for the input signal. The most common models that are currently in use are differential equations. The subsequent representation of the differential equation by a first-order vector differential equation leads to the concept of a state variable and the state space representation for a system. By use of the proper formulation, a continuous signal represented by a differential equation can be described by a first-order vector difference equation or a discrete time state space model. References that describe the essential aspects of state variables are by De Russo, Roy and Close [1] and Chen [2].

Since signals from laboratory instruments are not usually described by a differential equation, approximations often utilize a polynomial. For this reason, the vector form of a polynomial approximation will be used in this report. The reader should be aware that this can be generalized to include any signal model that can be represented as a linear time-varying differential equation. Later in the paper a scalar model representing a Gaussian time signal will also be utilized to develop a time-varying filter that can be used for reducing the base-line error and for the initial separation of signal components.

To develop the polynomial model let the signal $z(t)$ be represented by a polynomial of order m. At the time $t = nT$ the state vector for the signal is given by

$$\underline{z}(nT) = \begin{bmatrix} z \\ \dot{z} \\ \cdot \\ \cdot \\ D^m z \end{bmatrix}_{t=nT} \tag{1}$$

Redefining the state vector as

$$\underline{x}(nT) = \begin{bmatrix} z \\ \cdot \\ Tz \\ \frac{T^2}{2!}\ddot{z} \\ \cdot \\ \cdot \\ \frac{T^m}{m!} D^m z \end{bmatrix}_{t=nT} \tag{2}$$

the use of a Taylor series representation for each element of x(nT) now permits the state of system at t=(n+h)T to be described in terms of the state at t=nT by the relationship

$$\underline{x}[(n + h)T] = \Phi[h]\underline{x}[nT] \tag{3}$$

where $\Phi[h]$ is the mxm state transition matrix with elements

$$\Phi_{ij}[h] = \frac{j!}{i!(j-i)!} h^{j-i} \qquad \begin{matrix} 0 \le i \le m \\ i \le j \le m \end{matrix} \tag{4}$$

$$= 0 \qquad 1 \le j$$

The state transition matrix $\Phi[h]$ satisfies all of the relationships for general state transition matrices with the important two being for this work

$$\Phi[-h] = [\Phi(h)]^{-1} \tag{5}$$

and

$$\Phi[m]\ \Phi[p] = \Phi[m + p] \tag{6}$$

At this point, these models can be utilized in the design process.

Design of Nonrecursive Filters

Let the input signal start at time t=0 and assume that the signal over a finite data window is approximated by a polynomial z(t). Defining the state of the signal by (2), then the state of the signal at time t=(n+h)T is given in terms of the signal at time t=nT by (3). Given that the signal starts at t=0, the first ℓ observations are each defined as

$$\underline{m}[jT] = M\underline{x}[jT] + \underline{v}[jT] \quad j=0,1,2,\ldots\ell-1 \tag{7}$$

where M is a row matrix that relates the measurable state variables to the actual measurements. The elements of each noise vector $\underline{v}[jT]$ are the measurement noises. In general these are taken as random variables with zero mean and the time dependent autocovariance matrix

$$R[jT] = E[\underline{v}(jT)\underline{v}^{t}(jT)] \tag{8}$$

However, in most laboratory systems only the data is available and the derivatives are not measurable. Furthermore, the noise covariance matrix is usually not known and for scalar measurements the noise variance is assumed constant and the noise samples uncorrelated. For the remainder of this paper, only scalar measurements and uncorrelated measurement noise with time-invariant statistics will be assumed. The mean value of the noise will be taken as zero and the variance as σ^2. The results are easily extended to vector measurements and to measurement noise with time varying statistics.

For ℓ observations, the total observation vector at t=nT is defined as

$$\underline{m}_t[nT] = \begin{bmatrix} m[nT] \\ m[(n-1)T] \\ \vdots \\ M[(n-\ell+1)T] \end{bmatrix} \tag{9}$$

This vector now forms a data window of ℓ data points. For an estimate of the data at t=nT the use of the expression

$$\underline{x}[(n-j)T] = \Phi(-j)\ \underline{x}[nT] \tag{10}$$

is combined with (9) to yield

$$\underline{m}_t[nT] = \begin{bmatrix} M \\ M\Phi(-1) \\ \vdots \\ M\Phi[-\ell+1] \end{bmatrix} \underline{x}[nT] + \underline{v}_t[nT] \tag{11}$$

The matrix of constants H[nT] is now defined as

$$H[nT] = \begin{bmatrix} \underline{M} & \cdots & \cdots \\ \underline{M\Phi(-1)} & \cdots & \\ & \vdots & \\ \overline{M\Phi[-\ell+1]} & \cdots & \cdots \end{bmatrix} \tag{12}$$

so that (11) can be written as

$$\underline{m}_t[nT] = H[nT]\ \underline{x}[nT] + \underline{v}_t[nT] \tag{13}$$

The elements in H[nT] are constants given by

$$H_{ij} = (-i)^j \quad \begin{matrix} 0 \le i \le \ell \\ 0 \le j \le q \end{matrix} \tag{14}$$

where ℓ is the size of the data window and q is the order of the polynomial plus one. When i=j=0 the value of (14) is one. Note that since (12) is a matrix of constants there is no need to make the matrix a function of time. However, in the derivation for the recursive filters this provides a method for separating different H matrices.

The optimal estimate of the data at t=nT is now given in terms of weighted least squares or minimum variance because this form is utilized in the derivation of the recursive filters. If the covariance matrix for the total observation vector is

$$R_t(nT) = E[\underline{v}_t\ (nT)\ \underline{v}_t^t(nT)] \tag{15}$$

the optimal minimum variance estimate is

$$\underline{\hat{x}}[nT] = \hat{W}[nT]\ m_t[nT] \tag{16}$$

where $\hat{W}[nT]$ is a series of constant weights given by

$$\hat{W}[nT] = \left[H^t(nT)[R_t(nT)]^{-1}H(nT)\right]^{-1}H^t(nT)[R_t(nT)]^{-1} \tag{17}$$

For uncorrelated noise with constant variance σ^2 (15) is a diagonal matrix with elements σ^2 and (17) reduces to

$$\hat{W}[nT] = [H^t(nt)\ H(nt)]^{-1}\ H^t(nT) \tag{18}$$

This is the same result obtained using conventional least squares when the noise covariance is a diagonal matrix of equal constants. The reader should be aware that the estimate vector is an estimate of the data and all of the derivatives in the model. However, all of the weights in the derivative terms must be scaled by the scale factors in the state vector defined by (2).

The covariance matrix for the estimate given by (16) is given by

$$\hat{S}(nT) = \hat{W}(nT)\ R_t(nT)\ \hat{W}^t(nT) \tag{19}$$

Substituting (17) into (19) now yields

$$\hat{S}(nT) = [H^t(nT)\ [R_t(nT)]^{-1}\ H(nT)]^{-1} \tag{20}$$

which simplifies further to

$$\hat{S}(nT) = [H^t(nT)\ H(nT)]^{-1}\ \sigma^2 \tag{21}$$

for the uncorrelated noise.

By defining a delay or prediction factor α, estimates of the data αT units behind or ahead of the point $t=nT$ can be done. To do this the total observation vector is written as

$$\underline{m}_t(nT) = H_\alpha(nT)\ \underline{x}[(n-\alpha)T] + \underline{v}_t(nT) \tag{22}$$

where $H_\alpha(nT)$ now takes the form

$$H_\alpha(nT) = \begin{bmatrix} M\Phi(-\alpha) \\ M\Phi(-\alpha-1) \\ \vdots \\ M\Phi(-\ell-\alpha+1) \end{bmatrix} \tag{23}$$

The individual elements of the matrix now become

$$H_\alpha(nT)_{ij} = (\alpha-i)^j \quad \begin{matrix} 0\le i\le \ell \\ 0\le j\le q \end{matrix} \tag{24}$$

The form for the optimal estimate now utilizes $H_\alpha(nT)$ in (18). The covariance of the estimate uses $H_\alpha(nT)$ in (21).

Example

For a five-point window with $\alpha=0$, the optimal weight matrix and the covariance matrix for the estimate are shown in Table 1 for a third-order polynomial fit. For $\alpha=2$, the estimate of the data in the middle of the window is obtained and weight matrix and covariance matrices are shown in Table 2. This case corresponds to weights given in reference [3].

In practice, the design of nonrecursive filters is initiated by specifying the size of the window which is the number of rows in the H matrix, the order of the polynomial approximation which is one less than the number of columns in the H matrix and α which determines the coefficient values in the H matrix. This makes the design suitable for use with interactive graphics since only three parameters need be specified to generate the weight and covariance matrices.

If the model of the signal process is modified by additive uncorrelated driving noise, the variance terms of the driving noise fade the memory so that past data has less effect on the estimate. This can be thought of as uncertainty in the signal model. The concept is particularly important in recursive filter design. For nonrecursive filters, it can also be utilized and can be useful when using a non-recursive filter to initialize a recursive filter.

If the model includes driving noise, the state at time $t=nT$ is given by

$$\underline{x}[nT] = \Phi[-1]\ \underline{x}[(n-1)T] + \underline{w}[(n-1)T] \qquad (25)$$

where $\underline{w}[(n-1)T]$ is a sample from a noise process with mean zero and covariance matrix Q. This noise process is assumed to be white. In

terms of the total measurement vector, $\underline{m}_t[nT]$ now becomes

$$\underline{m}_t[nT] = H[nT]\ \underline{x}[nT] + \underline{p}_t(nT) + \underline{v}_t[nT] \tag{26}$$

where $\underline{p}_t[nT]$ is the total noise vector due to the driving noise. For scalar measurements this is given by

$$\underline{p}_t[nT] = \begin{bmatrix} 0 \\ -M\Phi[-1]\ \underline{w}[(n-1)T] \\ -M\Phi[-2]\ \underline{w}[(n-1)T] - M\Phi[-1]\ \underline{w}[(n-2)T] \\ \vdots \end{bmatrix} \tag{27}$$

The total noise vector that corrupts the total measurement vector is now defined as

$$\underline{r}_t[nT] = \underline{p}_t[nT] + \underline{v}_t[nT] \tag{28}$$

For scalar measurements the covariance matrix for $\underline{r}_t[nT]$ has diagonal elements whose value increases down the diagonal. Since the diagonal elements are not the same, the minimum variance expressions of (16) and (17) must be employed to find optimal linear estimates.

Example

To illustrate how the driving noise affects the filter weights, consider a zero order process which is equivalent to estimating a signal of constant value. For a three-point filter with scalar measurements, the total noise vector is

$$\underline{r}_t[nT] = \begin{bmatrix} v[nT] \\ \underline{v}[(n-1)T] - \underline{w}[(n-1)T] \\ \underline{v}[(n-2)T] - \underline{w}[(n-2)T] - \underline{w}[(n-1)T] \end{bmatrix} \tag{29}$$

If the variance of the measurement noise is σ^2 and of the driving noise ${\sigma_1}^2$ the covariance matrix of $\underline{r}_t[nT]$ is

$$R_t[nT] = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 + {\sigma_1}^2 & 0 \\ 0 & 0 & \sigma^2 + 2{\sigma_1}^2 \end{bmatrix} \tag{30}$$

The resulting weight matrix obtained using minimum variance is

$$W[nT] = [a_0 \; a_1 \; a_2] \tag{31}$$

where

$$a_0 = \frac{({\sigma_1}^2 + \sigma^2)\,(2{\sigma_1}^2 + \sigma^2)}{D}, \tag{32}$$

$$a_1 = \frac{\sigma^2(2{\sigma_1}^2 + \sigma^2)}{D}, \tag{33}$$

and

$$a_2 = \frac{\sigma^2({\sigma_1}^2 + \sigma^2)}{D}, \tag{34}$$

where D is given by

$$D = (2{\sigma_1}^2 + \sigma^2)\,({\sigma_1}^2 + \sigma^2) + \sigma^2\,(2{\sigma_1}^2 + \sigma^2) + \sigma^2({\sigma_1}^2 + \sigma^2) \tag{35}$$

The terms in the weight matrix satisfy the following inequality

$$a_0 \geq a_1 \geq a_2 \tag{36}$$

For ${\sigma_1}^2 = 0$, the equalities hold and $a_0 = a_1 = a_2 = 1/3$ which are the well known weights to estimate a mean. If ${\sigma_1}^2$ is not zero, the inequalities hold and as ${\sigma_1}^2$ becomes large with respect to σ^2, a_0 approaches one and a_1 and a_2 become smaller so that fading is

introduced. The covariance of the estimate is a scalar given by

$$\hat{S}(nT) = \frac{(\sigma_1^2 + \sigma^2)\ (2\sigma_1^2 + \sigma^2)\ \sigma^2}{D} = C\sigma^2 \qquad (37)$$

where C is a dimensionless constant whose value approaches one as σ_1^2 becomes larger than σ^2. Thus for $\sigma_1^2 >> \sigma^2$ only the current observation is effectively used and the variance of the estimate is approximately σ^2.

The fading obviously reduces the signal-to-noise enhancement of a nonrecursive digital filter. On the other hand, fading can be used to reduce the deterministic error due to a nonexact model. In practice, fading in nonrecursive filters is not often utilized. Instead, the window length is more typically used as a design parameter. In future work, however, it may be desirable to further explore the relationship between fading and the size of the data window to achieve improved designs when nonexact signal models are employed.

LEAST SQUARES DESIGN OF RECURSIVE DIGITAL FILTERS

The fixed memory or nonrecursive filter design is now extended to recursive digital filters that utilize all of the data. The result is a recursive form that is usually called the Kalman filter. While there are a variety of derivations for the Kalman filter, starting with a fixed memory filter using a polynomial model with driving noise gives the result in such a way to give the reader greater intuition about how the filter works.

The derivation of the recursive filter is started by using a signal model

$$\underline{x}[nT] = \Phi[1]\ \underline{x}[(n-1)T] + \underline{w}[(n-1)T] \qquad (38)$$

and the scalar observation or measurement model

$$y[nT] = M\underline{x}[nT] + \underline{v}[nT]. \tag{39}$$

In (38) and (39) the terms $w[(n-1)T]$ and $v[nT]$ are the driving noise and the measurement noise. These noise terms have zero mean and are uncorrelated with themselves and each other. Given the n+1 measurements starting at t=0, the total observation vector at time t=nT is given in terms of $\underline{x}[nT]$ as

$$\underline{y}[nT] = H[nT] + \underline{p}_t[nT] + \underline{v}_t[nT] \tag{40}$$

In (40) the matrix H[nT] is

$$H[nT] = \begin{bmatrix} M \\ \hline M\Phi[-1] \\ \\ \vdots \\ M\Phi[-n] \end{bmatrix} \tag{41}$$

the vector $\underline{p}_t[nT]$ is

$$\underline{p}_t[nT] = \begin{bmatrix} 0 \\ -M\Phi[-1]\ \underline{w}[(n-1)T] \\ -M\sum_{j=1}^{2} \Phi[-(3-j)]\ \underline{w}[(n-j)T] \\ \vdots \\ -M\sum_{j=1}^{n} \Phi[-(n+1-j)]\ \underline{w}[(n-j)T] \end{bmatrix} \tag{42}$$

and $\underline{v}_t[nT]$ is the total measurement noise vector. Defining the sum

$$\underline{r}_t[nT] = \underline{p}_t[nT] + \underline{v}_t[nT] \tag{43}$$

as the total noise vector with autocovariance

$$R_t[nT] = E\{\underline{r}_t[nT]\ \underline{r}_t^t[nT]\} \tag{44}$$

the optimal estimate using linear minimum variance is given by

$$\hat{\underline{x}}[nT] = W[nT]\ \underline{y}_t[nT] \tag{45}$$

where W[nT] is the weight matrix

$$W[nT] = \left[H^t[nT]\left[R_t[nT]\right]^{-1} H[nT]\right]^{-1} H^t[nT]\left[R_t[nT]\right]^{-1} \tag{46}$$

The covariance of the estimate is

$$\hat{S}[nT] = W[nT]\ R_t[nT]\ W^t[nT] \tag{47}$$

Substitution of (46) and (47) gives an alternate form

$$\hat{S}[nT] = \left[H^t[nT]\left[R_t[nT]\right]^{-1} H[nT]\right]^{-1} \tag{48}$$

which allows the alternate form for (46) to be written as

$$W[nT] = \hat{S}[nT]\ H^t[nT]\ R_t\left[[nT]\right]^{-1} \tag{49}$$

The forms given by (46), (47), (48) and (49) apply to the remaining estimates used in the derivation and the reader should remember them.

Next the prediction or forecast of the signal state X[(n+1)T] is found by first writing the total observation vector $\underline{y}_t[nT]$ as

$$Y_t[nT] = H_1[nT]\ \underline{x}[(n+1)T] + \underline{p}_{1t}[nT] + \underline{v}_t[nT] \tag{50}$$

where $H_1[nT]$ is given as

$$H_1[nT] = H[nT]\ \Phi[-1] \tag{51}$$

and $\underline{p}_{1t}[nT]$ is

$$\underline{p}_{1t}[nT] = \begin{bmatrix} -M\Phi[-1]\ \underline{w}[nT] \\ -M \sum_{j=1}^{2} \Phi[-(3-j)]\ \underline{w}[(n-1-j)T] \\ \vdots \\ -M \sum_{j=1}^{n+1} \Phi[-(n+1-j)]\ \underline{w}[(n-1-j)T \end{bmatrix} \tag{52}$$

The term $\underline{v}_t[nT]$ is the same as in (40) since no more observations have been taken.

The estimate of $\underline{x}[(n+1)T]$ is now given by

$$\underline{x}_1[(n+1)T] = W_1[nT]\ \underline{y}_t[nT] \tag{53}$$

where

$$W_1[nT] = \left[H_1[nT] \left[R_{1t}[nT] \right]^{-1} H_1[nT] \right] H_1[nT] \left[R_{1t}[nT] \right]^{-1} \tag{54}$$

In (54) is the covariance matrix of the noise vector

$$\underline{r}_{1t}[nT] = \underline{p}_{1t}[nT] + \underline{v}_t[nT] \tag{55}$$

The corresponding covariance matrix for the estimate $\underline{x}_1[(n+1)T]$ is

$$\begin{aligned} S_1[(n+1)T] &= W_1[nT]\ R_{1t}[nT]\ W_1^t[nT] \\ &= \left[H_1^t[nT] \left[R_{1t}[nT] \right]^{-1} H_1[nT] \right]^{-1} \end{aligned} \tag{56}$$

The development of the relationship between $R_{1t}[nT]$ and $R_t[nT]$ is the next issue.

To obtain this recognize that since the random variables $w[nT]$ and $\underline{v}[nT]$ are uncorrelated, $R_{1t}[nT]$ is the sum of the covariance matrices for $\underline{p}_{1t}[nT]$ and $\underline{v}_t[nT]$. For the latter, if the scalar noise $\underline{v}[nT]$ has variance σ^2, the covariance matrix of $\underline{v}_t[nT]$ is a diagonal

n+1 x n+1 matrix with elements σ^2. Taking the covariance of $\underline{p}_{1t}[nT]$ and performing some algebraic manipulation gives

$$E\{\underline{p}_{1t}[nT]\ \underline{p}_{1t}^t[nT]\} = E\{\underline{p}_1[nT]\ \underline{p}_1^t[nT]\} + H_1[nT]\ QH_1^t[nT] \tag{57}$$

where Q is the diagonal covariance matrix of the driving noise vector $\underline{w}[nT]$. This matrix is usually assumed to be time invariant. Thus, from (57)

$$R_{1t}[nT] = R_t[nT] + H_1[nT]\ QH_1^t[nT] \tag{58}$$

Substituting (58) into (56) now gives

$$S_1[(n+1)T] = W_1[nT]\ R_t[nT]\ W_1^t[nT] + W_1[nT]\ H_1[nT]\ QH_1^t[nT]\ W_1^t[nT] \tag{59}$$

If $X_1[(n+1)T]$ is an unbiased estimate then the constraint relationship

$$W_1[nT]\ H_1[nT] = I \tag{60}$$

must be satisfied [4] so that (59) can be simplified to

$$S_1[(n+1)T] = W_1[nT]\ R_t[nT]\ W_1^t[nT] + Q \tag{61}$$

Also recognizing that $\underline{x}_1[(n+1)T]$ can be written as

$$\underline{x}_1[(n+1)T] = \Phi[1]\ \hat{\underline{x}}[nT] \tag{62}$$

the weight matrix $W_1[nT]$ is

$$W_1[nT] = \Phi[1]\ W(nT). \tag{63}$$

Substitution at (58) into (61) and applying (47) now gives the final form for $S_1[(n+1)T]$ as

$$S_1[(n+1)T] = \Phi[1]\ \hat{S}[nT]\ \Phi^t[nT] + Q \tag{64}$$

The recursion is now formulated. When the observation at t = (n+1)T arrives the new total observation vector in terms of the signal state

$\underline{x}[(n+1)T]$ is

$$\underline{y}_t[(n+1)T] = H[(n+1)T] + \underline{p}_t[(n+1)T] + \underline{v}_t[(n+1)T]. \tag{65}$$

The estimate is given by

$$\hat{\underline{x}}[(n+1)T] = \hat{S}[(n+1)T]\ H^t[(n+1)T]\ R_t[(n+1)T]^{-1} \tag{66}$$

with the covariance

$$\hat{S}[(n+1)T] = \left[H^t[(n+1)T]\left[R_t[(n+1)T]\right]^{-1} H[(n+1)T]\right]^{-1} \tag{67}$$

where $R_t[(n+1)T]$ is the covariance matrix of the total measurement noise vector

$$\underline{r}_t[(n+1)T] = \underline{p}_t[(n+1)T] + \underline{v}_t[(n+1)T] \tag{68}$$

First the recursion for the covariance matrix is found. Given that

$$\underline{p}_t[(n+1)T] = \begin{bmatrix} -M\Phi[-1]\ \underline{w}[nT] \\ -M\sum_{j=1}^{2} \Phi[-(3-j)]\ \underline{w}[(n+1-j)T] \\ \vdots \\ -M\sum_{j=1}^{n+1} \Phi[-(n+1-j)]\ \underline{w}[(n+1-j)T] \end{bmatrix} \tag{69}$$

substitution of (52) into (69) gives

$$\underline{p}_t[(n+1)T] = \begin{bmatrix} 0 \\ \hline \underline{p}_{1t}[nT] \end{bmatrix} \tag{70}$$

Next $H[(n+1)T]$ is given as

$$H[(n+1)T] = \begin{bmatrix} M \\ \hline M\Phi[-1] \\ \hline \vdots \\ M\Phi[-(n+1)] \end{bmatrix} \tag{71}$$

which can be rewritten with the aid of (41) as

$$H[nT] = \begin{bmatrix} M\Phi[1] \\ \\ H[nT] \end{bmatrix} \Phi[-1] \tag{72}$$

The covariance matrix for $\underline{v}_t[(n+1)T]$ is now an n+2 x n+2 diagonal matrix with elements σ^2 so that $R_t[(n+1)T]$ is seen to be

$$R_t[(n+1)T] = \left[\begin{array}{c|c} \sigma^2 & 0 \\ \hline 0 & R_{1t}[nT] \end{array}\right] \tag{73}$$

Since this is a diagonal matrix its inverse is

$$R_t[(n+1)T] = \left[\begin{array}{c|c} \frac{1}{\sigma^2} & 0 \\ \hline 0 & \left[R_{1t}[nT]\right]^{-1} \end{array}\right] \tag{74}$$

Substitution of (74) and (72) into (67) and performing the matrix multiplication now yields

$$\hat{S}[(n+1)T] = \left[\frac{M^t M}{\sigma^2} + \Phi^t[-1]\, H^t[nT]\left[R_{1t}[nT]\right]^{-1} H[nT]\, \Phi[-1]\right]^{-1} \tag{75}$$

Substitution of (51) into (75) and the use of the inverse of (56) gives

$$\hat{S}[(n+1)T] = \left[\frac{M^t M}{\sigma^2} + \left[S_1[nT]\right]^{-1}\right]^{-1} \tag{76}$$

Equation (76) along with (64) now forms a recursion for the covariance of the estimate.

For the quantities in (76) the application of the matrix inversion lemma [5], gives

$$\left[\frac{M^t M}{\sigma^2} + \left[S_1[nT]\right]^{-1}\right]^{-1} = S_1[nT] - S_1[nT]\, M^t[M\, S_1[nT]\, M^t + \sigma^2]^{-1}\, M\, S_1[nT] \tag{77}$$

This form will be used to generate an expression for the Kalman gain. Post multiplying both sides of (75) by M^t/σ^2 and using (75) the Kalman gain is defined as

$$\begin{aligned} K[(n+1)T] &= \hat{S}[(n+1)T]\, M^t/\sigma^2 \\ &= S_1[(n+1)T]\, M^t[\sigma^2 + M\, S_1[(n+1)T]\, M^t]^{-1} \end{aligned} \tag{78}$$

Thus the covariance matrix given by (76) becomes

$$\hat{S}[(n+1)T] = [I - K[(n+1)T]\, M]\, S_1[(n+1)T] \tag{79}$$

The recursion for the optimal estimate is now formed that uses the Kalman gain given by (78). Using the form for the optimal estimate given by (47) the estimate $\hat{\underline{x}}[(n+1)T]$ is

$$\hat{\underline{x}}[(n+1)T] = \hat{S}[(n+1)T]\, H^t[(n+1)T]\left[R_{1t}[(n+1)T]\right]^{-1}\, \underline{y}_t[(n+1)T] \tag{80}$$

where $\underline{y}_t[(n+1)T]$ is the new total observation vector given by

$$\underline{y}_t[(n+1)T] = \begin{bmatrix} y[(n+1)T] \\ \hline \underline{y}_t[nT] \end{bmatrix} \tag{81}$$

Substitution of (74), (72) and (81) into (80) and carrying out the matrix multiplication yields

$$\begin{aligned} \hat{\underline{x}}[(n+1)T] = \hat{S}[(n+1)T] \Bigg[& \frac{M^t\, y[(n+1)T]}{\sigma^2} \\ & + \Phi[-1]\, H^t[nT]\left[R_{1t}[nT]\right]^{-1}\, \underline{y}_t[nT] \Bigg] \end{aligned} \tag{82}$$

Using the estimate of the forecast $x_1[(n+1)T]$ (82) is simplified to

$$\hat{\underline{x}}[(n+1)T] = \hat{S}[(n+1)T]\left[\frac{M^t}{\sigma^2}\, y[(n+1)T] + \left[S_1[(n+1)T]\right]^{-1} x_1[(n+1)T]\right] \tag{83}$$

Adding and subtracting $(M^tM/\sigma^2)\, x_1[(n+1)T]$, performing some algebraic manipulation, and applying (76) now yields

$$\underline{x}[(n+1)T] = \underline{x}_1[(n+1)T] + K[(n+1)T]\left[y[(n+1)T] - M\, \underline{x}_1[(n+1)T]\right] \tag{84}$$

where $K[(n+1)T]$ is given by (78). Equations (84), (79), (78), (64) and (62) now form the recursive formulation called the Kalman filter. These equations are now summarized as

$$\underline{x}_1(nT) = \Phi(1)\, \hat{\underline{x}}[(n-1)T] \,, \tag{85}$$

$$S_1(nT) = \Phi(1)\, \hat{S}[(n-1)T]\, \Phi^t(1) + Q \,, \tag{86}$$

$$K(nT) = S_1(nT)\, M^t[\sigma^2 + M\, S_1(nT)\, M^t]^{-1} \,, \tag{87}$$

$$\hat{S}(nT) = [I - K(nT)\, M]\, S_1(nT) \tag{88}$$

and

$$\begin{aligned} \hat{\underline{x}}(nT) &= \underline{x}_1(nT) + K(nT)\, [y(nT) - M\, \underline{x}_1(nT)] \\ &= [I - K(nT)M]\, \Phi(1)\, \hat{x}[(n-1)T] + K(nT)\, y(nT) \end{aligned} \tag{89}$$

In this set of equations, $x_1(nT)$ is the forecast or prediction of the estimate at $t=nT$ using the previously generated estimate at $t=[(n-1)T]$. The covariance of the forecast is $S_1(nT)$. The term $K(nT)$ is the time varying Kalman gain matrix and $y(nT)$ is the observation at $t=nT$. The terms $\hat{x}(nT)$ and $\hat{S}(nT)$ are the estimate at $t=nT$ and its covariance. The term σ^2 is the variance of the measurement noise and the term Q is the covariance of the driving noise. It is the term Q that serves as a key design parameter.

If there is no driving noise so that the diagonal elements of Q are all zero, the filter is simply an expanding memory filter. This means that if the filter is initialized properly, the estimates will correspond to those obtained by designing nonrecursive filters where the data window starts at zero and a new weight matrix $\hat{W}(nT)$ is computed as each new measurement is made. Obviously as the window expands, the variance of the estimate will decrease; however, if the model is not exact deterministic errors will begin to increase.

In using the equations, they must be initialized properly if a truly unbiased estimate is to be formed. In practice this is usually done using a nonrecursive filter. For exact initialization, driving noise must be included in the computation of the nonrecursive filter weights. To minimize the computation, the minimum H matrix should be used which means the number of rows should equal the number of columns. By using this minimum H matrix the filter weights are computed and the initial optimal estimate is formed from the actual data.

USING THE PROGRAMS

HARDWARE

The programs are written in DOS FORTRAN. They were developed on a PDP 11/20 with a DOS/BATCH operating system. Printed results are written to logical unit 5, which can be assigned at run time to a line printer, CRT terminal, disk file, or other suitable output device. Data is entered from units 6 and 3, which can be assigned to a card reader, disk data file, TTY keyboard, or any other suitable input device. Plots can be obtained with a GT40 graphics display terminal and the plotting subroutines provided. The programs can be easily modified to use other plotting routines.

NONRECURSIVE FILTER PROGRAM

The program for generating the coefficients for non-recursive filters is called PROGRAM WINDOW. WINDOW is written in DEC FORTRAN, but may be run on other versions of FORTRAN IV with minor modifications. The program can call plotting packages to produce CRT or hardcopy plots of the filter response to various inputs.

Program WINDOW reads the filter parameters SIGMA, N, M, IA, IPLOT from logical unit 6, where:

SIGMA	determines the input covariance matrix R. If SIGMA > 0.0, R becomes σ^2*I, where σ^2 = SIGMA and I is the identity matrix. If SIGMA $\leq$ 0.0, the matrix R is read from unit 6 in 10F8.2 FORMAT.
N	is the number of points in the window ($1 \leq N \leq 20$).
M	is the order of the polynomial fit ($1 \leq M \leq 9$).
IA	is the offset α from the first point in the window. If IA > 0, the filter predicts IA sample times ahead of the most recent sample. If IA < 0, the filter smooths IA sample times behind the most recent sample.
IPLOT	is the plotting control variable. If IPLOT = 0, the program finishes after the coefficient matrices are computed and printed out. If IPLOT $\neq$ 0, an input signal is read, and the input points are filtered using the coefficients computed.

The parameters are read in F10.4, 4I3 FORMAT.

Once the filter parameters have been read, WINDOW generates the required S (covariance) and T matrices , and computes the weight matrix W. All three matricies are then written to logical unit 5. If IPLOT = 0, the program terminates after the weight matrix W has been printed.

If IPLOT $\neq$ 0, WINDOW reads 101 sample points from logical unit 3 in G15.6 FORMAT. These points are provided by the user, and are used by WINDOW as the filter input signal. The program filters this input signal

using the weight matrix W, and tabulates the input signal, output signal (filter response), and error signal (input minus output) for each sample time. The tabulated results are printed on unit 5. The sum of the total absolute error is also computed and written to unit 5. WINDOW then calls the plotting package routines (subroutine IDIOT) to plot the input and output (estimate) signals vs. time. After a PAUSE, the program calls the plotting package to plot the error signal (input minus output) vs. time. The plotting packages are included to be used with WINDOW, or WINDOW may be changed to call other plotting routines.

Program WINDOW can be modified to write the derivative estimates. Note from equation 2 that the m^{th} derivative term is multiplied by a $T^m/m!$ factor, where T is the sample time. Thus, if the derivative estimates are written out, they will be scaled by this factor. To illustrate the use of the program, WINDOW was run with the parameters SIGMA = 1.0, N = 5, M = 3, IA = 0, and IPLOT = 0. The filter weighting coefficients are listed row by row, and are shown with the input parameters and S and T matrices in Fig. 1. The filter obtained using the weight matrix shown estimates the input data and the first three derivatives. In equation form, these estimates are given by

$$\hat{x}[nT] = 0.9857\, y[nT] + 0.05714\, y[(n-1)T] - 0.0857\, y[(n-2)T] + 0.05714\, y[(n-3)T] - 0.01429\, y[(n-4)T] ,$$

$$\dot{\hat{x}}[nT] = \frac{1.488\, y[nT] - 1.619\, y[(n-1)T] - .5714\, y[(n-2)T]}{T} + \frac{1.048\, y[(n-3)T] - 0.3452\, y[(n-4)T]}{T} ,$$

$$\ddot{\hat{x}}[nT] = \frac{0.6429\, y[nT] - 1.071\, y[(n-1)T] - 0.1429\, y[(n-2)T]}{T^{2}/_{2}} + \frac{0.9286\, y[(n-3)T] - 0.3571\, y[(n-4)T]}{T^{2}/_{2}} ,$$

$$\ldots$$

$$\hat{x}[nT] = \frac{0.08333\ y[nT] - 0.1667\ y[(n-1)T]}{T^{3}/_{6}}$$

$$+ \frac{0.1667\ y[(n-3)T] - 0.08333\ y[(n-4)T]}{T^{3}/_{6}} \quad .$$

RECURSIVE FILTERS

The Kalman filter program is called ADAPT. ADAPT is written in DEC FORTRAN, but may be run on other versions of FORTRAN IV with minor modifications. The program can call plotting packages to produce CRT or hardcopy plots of the filter response to various inputs.

Program ADAPT reads the filter parameters SIGMA, M, and Q from logical unit 6, where

SIGMA	determines the initial covariance matrix R. If SIGMA > 0.0, R becomes σ^2*I, where σ^2 = SIGMA and I is the identity matrix. If SIGMA $\leq$ 0.0, the matrix R is read from unit 6 in 10F8.2 FORMAT.
M	is the order of the polynomial fit ($1 \leq M \leq 9$).
Q	is the driving noise term.

The parameters are read in F10.4,I3,F10.4 FORMAT.

Once the filter parameters have been read, ADAPT generates an initial S (covariance) and T matrix. ADAPT then generates an initial weight matrix W, which is used to initialize the $\hat{x}$ vector. The initial covariance matrix is used to initialize the $\hat{S}$ matrix.

Once initialized, ADAPT reads 101 sample points from logical unit 3 in G15.6 FORMAT. These points are provided by the user, and are used by ADAPT as the Kalman filter input. The program filters the input using equations 85 through 89. The sample number filter input, filter output, error signal (input minus output), and Kalman gain are tabulated and written to unit 5. The total absolute error is also computed and written

to unit 5. ADAPT then calls the plotting package routines (subroutine IDIOT) to plot the input and output (estimate) signals vs. time. After a PAUSE, the program calls the plotting package to plot the error signal (input minus output) vs. time. After a second PAUSE, IDIOT is called to plot the Kalman gain vs. time.

Program ADAPT can be modified to write the derivative estimates. Note from equation 2 that the m^{th} derivative term is multiplied by a $T^m/m!$ factor, where T is the sample time. Thus, if the derivative estimates are written out, they will be scaled by this factor.

To illustrate the use of the program, ADAPT was run with the parameters SIGMA = 1.0, M = 3, Q = 0.0. The S (covariance) and T matrices used to generate the initial weight matrix (W) are shown in Fig. 2. The S and W matrices are used to initialize the Kalman filters' $\hat{S}$ and $\hat{\underline{x}}$ matrices. The Kalman filter was then used to filter an ideal sinusoidal signal. A portion of the tabulated results is shown in Fig. 3.

GETTING ON LINE

In order to run program WINDOW and ADAPT, first build two files named WINDOW.FTN and ADAPT.FTN from the sources provided (card deck or paper tape). Also create PLT.FTN and SENDGT.MAC from the sources. Compile programs WINDOW and ADAPT with the FORTRAN compiler to create WINDOW.OBJ and ADAPT.OBJ. The one word integer option should be selected for all compilations. Also, compile PLT.FTN to create PLT.OBJ. Assemble SENDGT.MAC under the MACRO assembler to create SENDGT.OBJ. Create a subroutine library called PLTLIB.OBJ from PLT.OBJ and SENDGT.OBJ (in that order). Next, LINK WINDOW.OBJ, PLTLIB.OBJ and the FTN library to create a file called WINDOW.LDA.

Also, create ADAPT.LDA by LINKing ADAPT.OBJ, PLTLIB.OBJ and the FTN library. The files WINDOW.OBJ, ADAPT.OBJ, PLT.OBJ and SENDGT.OBJ mav now be deleted.

Before running WINDOW or ADAPT, build the file PLOTGT.MAC from the source. PLOTGT is the plotting routine that is loaded into the GT40. Assemble and LINK PLOTGT so that PLOTGT.LDA may be loaded into the GT40. After PLOTGT.LDA is running in the GT40, ADAPT.LDA or WINDOW.LDA may be executed using a RUN command. The source listings contain additional documentation on these programs.

```
                    ***FINITE MEMORY DIGITAL FILTER PACKAGE***

 VARIANCE OF ERROR=     1.0000
 SIZE OF WINDOW=  5
 ORDER OF FIT=  2
 TO PREDICT   0 UNITS AWAY FROM THE FRONT OF THE WINDOW


        MAGIC T MATRIX
 1.0000         0.0000         0.0000         0.0000
 1.0000        -1.0000         1.0000        -1.0000
 1.0000        -2.000          4.000         -8.000
 1.0000        -3.000          9.000         -27.00
 1.0000        -4.000          16.00         -64.00


        VARIANCE MATRIX, S
 0.9857          1.488          0.6429         0.8333E-01
  1.488          6.379          2.869          0.5972
 0.6429          2.869          2.571          0.4167
 0.8333E-01      0.5972         0.4167         0.6944E-01


        WINDOW WEIGHTS,  FROM FRONT TO BACK

                ROW  1 OF W.  CORRESPONDING TO THE INPUT SIGNAL
 0.9857         0.5714E-01   -0.8571E-01    0.5714E-01   -0.1429E-01


                ROW  2 OF W. CORRESPONDING TO DERIVATIVE NUMBER  1
  1.488        -1.619         -0.5714          1.048        -0.3452


                ROW  3 OF W. CORRESPONDING TO DERIVATIVE NUMBER  2
 0.6429         -1.071         -0.1429         0.9286        -0.3571


                ROW  4 OF W. CORRESPONDING TO DERIVATIVE NUMBER  3
 0.8333E-01   -0.1667         -0.3576E-06     0.1667        -0.8333E-01
```

Fig. 1 - Program WINDOW Sample Output

```
                    ***KALMAN STRUCTURE DIGITAL FILTER PACKAGE***

 DRIVING NOISE=     0 0000
 ORDER OF FIT=  3
 INITIAL ERROR VARIANCE     1 0000


        MAGIC T MATRIX
1 0000        0 0000        0 0000        0 0000
1 0000       -1 0000        1 0000       -1 0000
1 0000        -2 000         4 000        -8 000
1 0000        -3 000         9 000        -27 00


        VARIANCE MATRIX S
1 0000        1 833         1 0000        0 1667
 1 833        14 72          12 50         2 611
1 0000        12 50          11 50         2 500
0 1667        2 611          2 500         0 5555


        WINDOW WEIGHTS,  FROM FRONT TO BACK

              ROW  1 OF W,  CORRESPONDING TO THE INPUT SIGNAL
1 0000       0 4128E-05  -0 2503E-05   0 9537E-06


              ROW  2 OF W, CORRESPONDING TO DERIVATIVE NUMBER  1
 1 833        -3 000         1 500       -0 3333


              ROW  3 OF W, CORRESPONDING TO DERIVATIVE NUMBER  2
1 0000        -2 500         2 000       -0 5000


              ROW  4 OF W, CORRESPONDING TO DERIVATIVE NUMBER  3
0 1667       -0 5000        0 5000       -0 1667
```

Fig. 2 - Program ADAPT Sample Output

TIME	OUTPUT	ESTIMATE	ERROR	KAL GAIN
4.000	0.3894	0.3894	0.2682E-06	0.9857
5.000	0.4794	0.4794	0.1627E-04	0.9695
6.000	0.5646	0.5646	0.5460E-04	0.9482
7.000	0.6442	0.6441	0.1017E-03	0.9162
8.000	0.7174	0.7173	0.1563E-03	0.8794
9.000	0.7833	0.7831	0.2239E-03	0.8419
10.000	0.8415	0.8412	0.3122E-03	0.8050
11.00	0.8912	0.8908	0.4330E-03	0.7707
12.00	0.9320	0.9314	0.6002E-03	0.7382
13.00	0.9636	0.9627	0.8295E-03	0.7079
14.00	0.9855	0.9843	0.1140E-02	0.6797
15.00	0.9975	0.9959	0.1551E-02	0.6534
16.00	0.9996	0.9975	0.2086E-02	0.6289
17.00	0.9917	0.9889	0.2768E-02	0.6061
18.00	0.9738	0.9702	0.3620E-02	0.5848
19.00	0.9462	0.9416	0.4669E-02	0.5649
20.00	0.9093	0.9034	0.5938E-02	0.5462
21.00	0.8632	0.8558	0.7450E-02	0.5287
22.00	0.8085	0.7993	0.9229E-02	0.5123
23.00	0.7457	0.7344	0.1130E-01	0.4968
24.00	0.6755	0.6618	0.1367E-01	0.4822
25.00	0.5985	0.5821	0.1635E-01	0.4684
26.00	0.5155	0.4961	0.1937E-01	0.4553
27.00	0.4274	0.4047	0.2272E-01	0.4430
28.00	0.3350	0.3086	0.2640E-01	0.4312
29.00	0.2392	0.2088	0.3040E-01	0.4201
30.00	0.1411	0.1064	0.3472E-01	0.4095
31.00	0.4158E-01	0.2265E-02	0.3932E-01	0.3995
32.00	-0.5827E-01	-0.1025	0.4417E-01	0.3899
33.00	-0.1577	-0.2070	0.4924E-01	0.3807
34.00	-0.2555	-0.3100	0.5447E-01	0.3720
35.00	-0.3508	-0.4106	0.5981E-01	0.3637
36.00	-0.4425	-0.5077	0.6519E-01	0.3557
37.00	-0.5298	-0.6004	0.7054E-01	0.3480
38.00	-0.6119	-0.6876	0.7576E-01	0.3407
39.00	-0.6879	-0.7685	0.8076E-01	0.3337
40.00	-0.7568	-0.8422	0.8543E-01	0.3269
41.00	-0.8182	-0.9080	0.8968E-01	0.3204
42.00	-0.8716	-0.9649	0.9336E-01	0.3142
43.00	-0.9162	-1.013	0.9637E-01	0.3082
44.00	-0.9516	-1.050	0.9857E-01	0.3024
45.00	-0.9775	-1.077	0.9981E-01	0.2969
46.00	-0.9937	-1.094	0.9997E-01	0.2915
47.00	-0.9999	-1.099	0.9891E-01	0.2863
48.00	-0.9962	-1.093	0.9648E-01	0.2813
49.00	-0.9825	-1.075	0.9255E-01	0.2765
50.00	-0.9589	-1.046	0.8697E-01	0.2718
51.00	-0.9258	-1.005	0.7963E-01	0.2673
52.00	-0.8835	-0.9538	0.7039E-01	0.2629
53.00	-0.8322	-0.8914	0.5915E-01	0.2587
54.00	-0.7728	-0.8186	0.4580E-01	0.2546
55.00	-0.7055	-0.7358	0.3025E-01	0.2507
56.00	-0.6312	-0.6437	0.1241E-01	0.2468
57.00	-0.5507	-0.5429	-0.7760E-02	0.2431
58.00	-0.4646	-0.4343	-0.3022E-01	0.2395
59.00	-0.3739	-0.3186	-0.5528E-01	0.2360
60.00	-0.2794	-0.1968	-0.8265E-01	0.2325
61.00	-0.1822	-0.6975E-01	-0.1124	0.2292

Fig. 3 - Program ADAPT Sample Output

0.9857	0.05714	-0.0857	0.05714	-0.01429
1.488	-1.619	-0.5714	1.048	-0.3452
0.6429	-1.071	-0.1429	0.9286	-0.3571
0.08333	-0.1667	0.0	0.1667	-0.08333

Weight Matrix

0.9857	1.488	0.6429	0.08333
1.488	6.379	3.869	0.5972
0.6429	3.869	2.571	0.4167
0.08333	0.5972	0.4167	0.06944

Covariance Matrix

Table 1 Optimal Weight Matrix and Covariance Matrix For a Five Point Nonrecursive Filter Using a Third Order Polynomial Model. The Filter Estimates the Data at the End of the Window

-0.08571	0.3429	0.4857	0.3429	-0.08571
-0.08333	0.6667	0.0	-0.6667	0.08333
0.1429	-0.07143	-0.1429	-0.07143	0.1429
0.08333	-0.1667	0.0	0.1667	-0.08333

Weight Matrix

0.4857	0.0	-0.1429	0.0
0.0	0.9828	0.0	-0.2361
-0.1429	0.0	0.07143	0.0
0.0	-0.2361	0.0	0.06944

Covariance Matrix

Table 2 Optimal Weight Matrix and Covariance Matrix For a Five Point Nonrecursive Filter Using a Third Order Polynomial Model. The Filter Estimates the Data in the Center of the Window

REFERENCES

1. P. DeRusso, R. Roy and C. Close, State Variables for Engineers, John Wiley and Sons, 1965.

2. C. Chen, Introduction to Linear System Theory, Holt, Rinehart and Winston, 1970.

3. A. Savitzky and M. J. E. Golay, "Smoothing and Differentiation of Data by Simplified Least Squares Procedures," Analytical Chemistry, Vol. 36, No. 8, pp. 1627-1639, July 1964.

4. N. Morrison, Introduction to Sequential Smoothing and Prediction, McGraw-Hill, 1969.

5. A. Sage and J. Melso, Estimation Theory with Applications to Communications and Control, McGraw-Hill, 1971.

Appendix D

A FORTRAN IV DESIGN PROGRAM FOR LOW-PASS BUTTERWORTH AND CHEBYCHEV DIGITAL FILTERS

by

H. J. Markos

and

T. A. Brubaker

The authors are with the Electrical Engineering Department

at Colorado State University, Fort Collins, Colorado

INTRODUCTION

This report contains the documentation for the LPASS program. It consists of the design procedure used, a description of the program, and design examples using the program.

The purpose of the LPASS program is the design of a maximally flat Butterworth or an equiripple Chebychev lowpass digital filter. Starting with an analog filter, the bilinear Z transform is used to find an equivalent digital filter. The user enters the following parameters: the number of second order sections, the type of filter, the sampling interval, the -3db cutoff frequency, the starting frequency and the frequency increment. If a Chebychev filter is being designed, the ripple must also be entered.

The program calculates the digital filter coefficients for up to three second order sections in cascade. The program is designed to calculate up to a sixth order filter, thus the filter order is two times the number of cascaded second order sections. The filter magnitude response is generated over the frequency interval specified by the input.

The LPASS program, written in Fortran IV, is supplied as a card deck with this report. The program is in the form of a subroutine and can be used as is by a call statement from the main program. Data may be input via cards with output available through a line printer. The input/output devices may be altered as explained in this report. Graphics routines may easily be appended to the program.

I. Design Procedure

A. Preliminary Discussion

The transfer function of a second order digital filter in the Z domain is given by

$$H(Z) = \frac{K_1(A_0Z^2+A_1Z+A_2)}{Z^2+B_1Z+B_2} \qquad (1)$$

where the A's and B's are the coefficients of the numerator and denominator respectively. One common method of designing a digital filter is to start with an analog transfer function H(S) and transform it to the digital transfer function H(Z). This program will calculate the scale factor K_1 and the coefficients A_0, A_1, A_2, B_1, and B_2. The transformation used is the extended bilinear Z transform defined as

$$S \rightarrow \frac{2}{T}\left(\frac{Z-1}{Z+1}\right) , \qquad (2)$$

where T is the sampling interval. When this transform is employed, the desired frequencies must first be prewarped to make them compatible with the digital filter. The prewarped cutoff frequency is given by

$$WDC = \frac{2}{T}\,\mathrm{Tan}\left(\frac{\omega_c T}{2}\right) . \qquad (3)$$

This prewarping is done by the program.

B. Butterworth Low-Pass Filter

We start with a normalized second order low-pass filter in the S plane.

$$H(S) = \frac{1}{S^2 + 2S\cos\theta + 1} \qquad (4)$$

where the angle θ is in degrees (in the program). θ may be found from the Butterworth circle and the relationship

$$S = e^{\pm j\pi(2m-1)/2n} \tag{5}$$

where n is the order of the filter and $m = 1, 2, 3, \ldots, n$. This relationship is determined by the following procedure. By definition, a filter is n^{th} order Butterworth low-pass if its gain characteristic is

$$|H_n(j\omega)|^2 = \frac{a^2}{1 + (\frac{\omega}{\omega_c})^{2n}} \tag{6}$$

where a is the gain, ω_c is the desired cutoff frequency and n is the order of the filter. Note that $|H_n(j\omega)|^2$ goes to zero as ω goes to infinity, indicating the filter does attenuate the higher frequencies. To determine its efficiency as a low-pass filter we calculate

$$\frac{d}{d\omega}|H_n(j\omega)| = -\frac{an}{\omega_c} \frac{(\frac{\omega}{\omega_c})^{2n-1}}{\left[1+(\frac{\omega}{\omega_c})^{2n}\right]^{3/2}} \tag{7}$$

Thus

$$\frac{d}{d\omega}\left[|H_n(j\omega)|\right]_{\omega=0} = 0 \tag{8}$$

for all n and hence the gain characteristic stays flat for ω close to 0. Also

$$\left[\frac{d}{d\omega}|H_n(j\omega)|\right]_{\omega=\omega_c} = -\frac{an}{2\omega_c\sqrt{2}} \tag{9}$$

and hence, the decline rate or "roll-off" of the gain characteristic

at $\omega = \omega_c$ becomes sharper as n increases. In other words, the approximation to the ideal low-pass filter improves for larger n. The order n is chosen according to desired specifications. The references have equations, curves, and tables that select n, given the specifications. For example, page 227 of Rabiner and Gold gives an equation for calculating n when the transition band is specified.

In the design, the poles for the full frequency response, $H(S)$, of the n^{th} order Butterworth filter must be determined. The procedure is as follows:

$$|H_n(j\omega)|^2 = H_n(j\omega)\overline{H_n(j\omega)} = H_n(j\omega)H_n\overline{(j\omega)} = H_n(j\omega)H_n(-j\omega)$$

$$= [H(S)H(-S)]_{S=j\omega} = \left[\frac{a^2}{1 + (\frac{\omega}{\omega_c})^{2n}}\right]_{\omega = \frac{S}{j}} = \frac{a^2}{1 + (\frac{S}{j\omega_c})^{2n}}$$

$$= \frac{a^2}{1 + -\left[\frac{S^2}{\omega_c^2}\right]^n} = \begin{cases} \dfrac{a^2}{1 + \left[\frac{S^2}{\omega_c^2}\right]^n}, & \text{for } n \text{ even} \\ \dfrac{a^2}{1 - \left[\frac{S^2}{\omega_c^2}\right]^n}, & \text{for } n \text{ odd} \end{cases} \qquad (10)$$

Setting the denominators equal to zero,

$$\frac{S}{\omega_c} = (\pm 1)^{1/2n} \quad . \qquad (11)$$

Thus, the pole locations are the $2n$ roots of ± 1, depending on whether the order is odd or even. These roots are located on a circle with radius ω_c centered at the origin of the S plane and have symmetry

with respect to both real and imaginary axes. For n odd, a pair of roots are on the real axis and the rest are separated by π/n radians. For n even, a pair of roots are located $\pi/2n$ radians from the real axis and the rest are again separated by π/n radians. No roots are on the imaginary axis for either even or odd n.

Let $p_1, \ldots, p_{2n}$ be the roots. From the symmetry of the pole locations, if $p_1, \ldots, p_n$ are the roots lying in the right-half plane, the left-half plane roots are $-p_1, \ldots, -p_n$. The magnitude-squared function can then be written as

$$H_n(S)H_n(-S) = \frac{a^2(-1)^n\omega_c{}^{2n}}{(S+p_1)\ldots(S+p_n)(S-p_1)\ldots(S-p_n)} \qquad (12)$$

To be stable, $H_n(S)$ must have all its poles in the left-half plane, thus

$$H_n(S) = \frac{a\omega_c{}^n}{(S+p_1)\ldots(S+p_n)} \; . \qquad (13)$$

The program is written with unity gain at DC, (ω=0), therefore a = 1.

In order to locate the poles as specified above, consider the following set of equations.

$$\begin{aligned} 1 &= -e^{\pm j\pi(2m-1)} \quad , \; m = 1, 2, \ldots, n; \text{ for n even} \\ -1 &= -e^{\pm j2\pi k} \quad , \; k = 0, 1, \ldots, n; \text{ for n odd} \end{aligned} \qquad (14)$$

Substituting equations (14) into equation (11) yields

$$\begin{aligned} \left[\frac{S}{\omega_c}\right]_{\pm m} &= -e^{\pm j\pi(2m-1)/2n} \quad , \; m = 1, 2, \ldots, n; \text{ for n even} \\ \left[\frac{S}{\omega_c}\right]_{\pm k} &= -e^{\pm j\pi k/n} \quad , \; k = 0, 1, \ldots, n; \text{ for n odd} \end{aligned} \qquad (15)$$

Equations (15) will give the pole locations as described above. Consider the form of equations (15)

$$S = -\omega_c \, e^{\pm j\theta} = \omega_c[-\cos\theta \pm j\sin\theta]. \qquad (16)$$

From this relationship, it can be seen that the magnitude for each pole is ω_c, regardless of the angle, and thus all the poles lie on a circle with radius ω_c.

As an example consider a second order filter, n = 2.

$$\left[\frac{S}{\omega_c}\right]_{\pm m} = -e^{\pm j\pi(2m-1)/4} \qquad m = 1, 2$$

$$S_{\pm 1} = \omega_c \ \underline{/\pm 45^\circ}$$

$$S_{\pm 2} = \omega \ \underline{/\pm 135^\circ}$$

$$\theta = 45^\circ$$

The relationship of these roots about the circle of radius ω_c is illustrated in Figure 1. The angle θ is always measured from the negative real axis.

In the program, only the angle(s) less than 90° are considered so that poles lie in the left-half plane since poles in the left-half plane are stable. Putting $\theta = 45^\circ$ into equation (4) yields poles at $-0.707 \pm j0.707$. These locations are in the left-half plane.

In the program, only even order filters are considered.

Below are the values of θ for 1, 2, and 3 second order sections in cascade.

Cascaded Sections N	Filter Order n	Angle θ
1	2	45°
2	4	22.5°, 67.5°
3	6	75°, 45°, 15°

These calculated angles are incorporated in the program in the order given above.

For N second order sections there are N θ's. Only one specific θ is used per stage, because each stage has only one set of pole locations.

The following is the procedure to derive the magnitude of the i^{th} stage, where i varies from 1 to N.

Given the normalized second order low-pass transfer function equation (4), we employ the low-pass to low-pass transformation for an arbitrary cutoff frequency ω_c given by

$$S \rightarrow \frac{s}{\omega_c} \tag{17}$$

For the i^{th} stage, equation (4) becomes

$$H_i(S) = \frac{\omega_c^2}{S^2+2S\omega_c \cos\theta_i+\omega_c^2} \tag{18}$$

The extended bilinear Z transform, equation (2), is used to get to the digital domain. Employing equation (2) on equation (18) and substituting WDC for ω_c yields

$$H_i(Z) = \frac{WDC^2(Z^2+2Z+1)}{\frac{4}{T^2}(Z^2-2Z+1)+\frac{4}{T}(Z^2-1)WDC \cos\theta_i+WDC^2(Z^2+2Z+1)} \tag{19}$$

Putting the denominator of equation (19) in monic form yields the transfer function for the i^{th} stage of the filter

$$H_i(Z) = \frac{K_{1i}(A_0Z^2+A_1Z+A_2)}{Z^2+B_{1i}Z+B_{2i}} \quad . \tag{20}$$

Equation (20) is the same as equation (1) with the exception of the subscripts. In equation (20)

$$A_0 = A_2 = 1$$

$$A_1 = 2$$

$$G_i = \frac{4}{T^2} + \frac{4}{T}\,WDC\,\cos\theta_i + WDC^2 \qquad (21)$$

$$K_{1i} = \frac{WDC^2}{G_i}$$

$$B_{1i} = \frac{2WDC^2 - \frac{8}{T^2}}{G_i}$$

$$B_{2i} = \frac{\frac{4}{T^2} - \frac{4}{T}\,WDC\cos\theta_i + WDC^2}{G_i}$$

Letting $Z = e^{ST}$ and $S = j\omega$ and taking the magnitude of $H_i(j\omega)$ we have

$$|H_i(j\omega)| = K_{1i}\,\frac{\sqrt{(A_0\cos(2\omega T)+A_1\cos(\omega T)+A_2)^2+(A_0\sin(2\omega T)+A_1\sin(\omega T))^2}}{\sqrt{(\cos(2\omega T)+B_{1i}\cos(\omega T)+B_{2i})^2+(\sin(2\omega T)+B_{1i}\sin(\omega T))^2}} \qquad (22)$$

This magnitude function is the same for both the Butterworth and the Chebychev filters where i varies from 1 to N.

C. Chebychev Low-Pass Filter

The advantage of the Chebychev low-pass filter over the Butterworth low-pass filter is that the transition band of the response at frequencies greater than ω_c is sharper for the Chebychev low-pass filter. This is achieved by specifying a small percentage of ripple in the low-pass region. The amplitude of the ripple is specified by the quantity δ (labeled RIP in the program). Figures 6, 7 and 8 illustrate the rippling for second, fourth, and sixth order filters, respectively. The poles of the filter are found on an ellipse

described by two Butterworth circles of radii A and B with A<B. The location of the poles on the ellipse is a function of the ripple, δ, and is given by the following equation:

$$B,\ A = \frac{1}{2}\left(\left(\sqrt{\varepsilon^{-2}+1}+\varepsilon^{-1}\right)^{1/2N} \pm \left(\sqrt{\varepsilon^{-2}+1}+\varepsilon^{-1}\right)^{-1/2N}\right) \tag{23}$$

where

$$\varepsilon = \left[\frac{1}{(1-\delta)^2} - 1\right]^{1/2} \tag{24}$$

B is given for the plus sign and A for the minus sign. The Chebychev ellipse then has major axis B and minor axis A. The location of the S plane poles on the ellipse is given by

$$\begin{aligned} &\text{Real Part} = A\cos\theta \\ &\text{Imaginary Part} = B\sin\theta \end{aligned} \tag{25}$$

The θ's are the same as given for the corresponding order Butterworth filter. An example of Chebychev pole locations is illustrated in Figure 2. For A = 1/2 and B = 1 in a fourth order filter, $\theta = 22.5°$ and $67.5°$. The Chebychev pole locations are determined from equations (23), (24), and (25).

The analog second order Chebychev low-pass filter is

$$H(S) = \frac{K_2 a}{S^2+K_8 S+K_2} \tag{26}$$

where

$$a = \left[\frac{1}{(1+\varepsilon^2)^{1/2}}\right]^{1/N} . \tag{27}$$

ε is calculated from equation (24) and N is the number of second order sections. K_8 and K_2 are calculated by

$$K_8 = 2A\cos\theta \tag{28}$$

$$K_2 = A^2\cos^2\theta + B^2\sin^2\theta \ . \tag{29}$$

The substitution of the low-pass to low-pass transformation for some cutoff frequency ω_c, equation (17), into equation (26) yields

$$H(S) = \frac{K_2 a\omega_c^{\ 2}}{S^2 + SK_8\omega_c + \omega_c^{\ 2}K_2} \tag{30}$$

Using the extended bilinear Z transform, equation (2), and substituting WDC for ω_c we have for any section

$$H(Z) = \frac{K_2 aWDC^2(Z^2+2Z+1)}{\frac{4}{T^2}(Z^2-2Z+1) + \frac{2}{T}K_8 WDC(Z^2-1) + WDC^2K_2(Z^2+2Z+1)} \ . \tag{31}$$

Collecting terms yields the following for the i^{th} section

$$H_i(Z) = \frac{K_{1i}(A_0Z^2+A_1Z+A_2)}{Z^2+B_{1i}Z+B_{2i}} \tag{32}$$

where

$$A_0 = A_2 = 1$$

$$A_1 = 2$$

$$G_i = \frac{4}{T^2} + \frac{2}{T}\,WDC\cdot K_8 + WDC^2K_2$$

$$K_{1i} = \frac{aK_2WDC^2}{G_i} \tag{33}$$

$$B_{1i} = \frac{2WDC^2K_2 - \frac{8}{T^2}}{G_i}$$

$$B_{2i} = \frac{\frac{4}{T^2} - \frac{2}{T}\,WDC\cdot K_8 + WDC^2 K_2}{G_i}$$

i varies from 1 to N. These coefficients are used to find $|H_i(j\omega)|$ given in (22).

For applications where a sharper roll-off is required the Chebychev filters are used. The roll-off increases with n for any fixed ε. For fixed n, the roll-off decreases as ε decreases. For small ε the ripple width, δ, is small, see equation (23), but so is the roll-off. For larger ε the roll-off improves but the ripple width increases. In the first case the filter will be good at DC and low frequencies, unsatisfactory at high frequencies. The converse is true in the second case.

The above observations suggest the procedure to be used in selecting a Chebychev filter to match a set of specifications. The permissible ripple width specifies ε. With ε fixed, select n to attain the required roll-off.

II. Using the Program

The first data card read into the program contains the number of second order sections to be cascaded, N, and the type of filter desired, KN. N is equal to 1, 2, or 3, which corresponds to the 2^{nd}, 4^{th}, or 6^{th} order filter respectively. KN = 1 yields a Butterworth filter, while KN = 2 yields a Chebychev filter. The format on the N, KN card is 2I2. The second data card read in is the sampling interval T in F10.6 format. When choosing T, 1/T should be approximately equal to ten times the cutoff frequency, ω_c. The third data card contains the value of ω_c in F10.4 format. For the Butterworth low-pass filter, ω_c is the -3db cutoff frequency. For the Chebychev filter the magnitude of the response is $1/(1+\varepsilon^2)^{1/2}$ $= 1 - \delta$ at $\omega = \omega_c$. ω is in radians. δ is the ripple factor.

If the desired filter is Chebychev, i.e., KN = 2, the next data card is the ripple factor (RIP) in F5.3 format. The filter response for all even order Chebychev low-pass filters passes through $1/(1+\varepsilon^2)^{1/2} = 1 - \delta$ for $\omega = 0$ and ω_c. For odd order filters, the magnitude is 1 for $\omega = 0$ and $1/(1+\varepsilon^2)^{1/2} = 1 - \delta$ for $\omega = \omega_c$. This program produces only even order filters. If the desired filter is Butterworth, i.e., KN = 1, this data card is omitted from the data deck.

The final data card is the starting frequency (FREQ1) and the frequency increments (DELT) in radians. The format of the FREQ1, DELT card is 2F10.4. Determine DELT by the following:

$$\text{DELT} = \frac{\text{final frequency - starting frequency}}{1024}$$

This is necessary because there are 1024 frequency data points calculated in the program. Choose FREQ1 and DELT to insure that calculated values

will include the data of interest. For maximum efficiency of the program, DELT should be a multiple of 2^{-K} so no decimal to binary conversion errors are incurred.

The digital filter coefficients are computed and printed out for each second order section. The full filter magnitude response, as well as each section magnitude response, is printed for each of the frequency increments specified. When N = 1, the section magnitude response is the full filter magnitude response and is only printed once.

The program may be easily modified to incorporate a graphics display of the magnitude response. There is a comment card in the LPASS program indicating where the graphics subroutine call card should be inserted.

The program is written with input obtained via device 4 and output written to device 6. These numbers should be assigned to the appropriate devices prior to running the program.

The program was developed on the PDP-11/20 with a DOS/BATCH operating system. Trial runs frequently used a TTY terminal as well as a card reader for input (device 4); and a TTY terminal as well as a line printer for output (device 6).

Double precision arithmetic is employed. To decrease required memory storage, only the frequency interval values and the full magnitude response are saved. The section magnitude responses are printed out, but are not stored. The program will produce approximately 21 pages of output.

Shown below are sample deck set-ups for the Chebychev and Butterworth low-pass filters.

Data Card	Format	Example
1	2I2	0302 (3 sections Chebychev low-pass)
2	F10.6	0.001 (T = 0.001)
3	F10.4	100 (ω_c = 100 radians)
4	F5.3	0.10 (Ripple amplitude = 0.10)
5	2F10.4	70 0.06 (Start at ω = 70. Steps of 0.06 radians. Will finish just past ω = 131 radians.)
1	2I2	0201 (2 sections, 4^{th} order, Butterwo[illegible])
2	F10.6	0.005 (T = 0.005)
3	F10.4	20 (ω_c = 20 radians)
4	2F10.4	0 0.04 (Start at ω = 0 finish just past ω = 40 radians in steps of 0.04 radians.)

The following pages contain annotated examples of output data.

This is an example of the output for a 4^{th} ofder Butterworth low-pass filter with T = 0.005 and ω_c = 20 radians. The starting frequency is 0 radians and the frequency increment is 0.04 radian.

WDC = 20.01668 WC = 20.00000 T = 0.50000E-02

FOR I = 1 A_0 = 0.10000000E+01 A_1 = 0.20000000E+01
A_2 = 0.10000000E+01 K_1 = 0.22869799E-02
B_1 = 0.18219614E+01 B_2 = 0.83110937E+00

FOR I = 2 A_0 = 0.10000000E+01 A_1 = 0.20000000E+01
A_2 = 0.10000000E+01 K_1 = 0.24059972E-02
B_1 = 0.19167786E+01 B_2 = 0.92640257E+00

W	H	H1	H2
0.0000	0.10000E+01	0.10000E+01	0.10000E+01
0.0400	0.10000E+01	0.10000E+01	0.10000E+01
0.0800	0.99999E+00	0.99999E+00	0.10000E+01
0.1200	0.10000E+01	0.99997E+00	0.10000E+01
0.1600	0.10000E+01	0.99995E+00	0.10000E+01
0.2000	0.10000E+01	0.99993E+00	0.10000E+01
0.2400	0.10000E+01	0.99990E+00	0.10001E+01
.	.	.	.
.	.	.	.
.	.	.	.

I is the i^{th} stage. I varies from 1 to N.

WDC is the prewarped cutoff frequency.

WC is the cutoff frequency.

T is the sampling interval.

A_0, A_1, and A_2 are the low-pass filter numerator coefficients.

B_1 and B_2 are the low-pass filter denominator coefficients.

K_1 is the gain factor.

W is the frequency.

H is the overall magnitude of the digital transfer function.

H1 is the magnitude of the digital transfer function (1^{st} stage).

H2 is the magnitude of the digital transfer function (2^{nd} stage).

H = H1*H2.

See Figure 4.

This is an example of the output for a 6^{th} order Chebychev low-pass filter (three second order stages cascaded) with T = 0.005 and ω_c = 20 radians. The starting frequency is 0 and the frequency increment is 0.04 radian. The ripple is equal to 0.100.

WDC = 20.01668 WC = 20.00000 T = 0.50000E-02

A = 0.24783947 B = 1.03025433 K_8 = 0.12829114 K_2 = 0.99443709
A = 0.24783947 B = 1.03025453 K_8 = 0.35049793 K_2 = 0.56142438
A = 0.24783947 B = 1.03025453 K_8 = 0.47878908 K_2 = 0.12841170

FOR I = 1 A_0 = 0.10000000E+01 A_1 = 0.20000000E+01
A_1 = 0.10000000E+01 K_1 = 0.23830688E-02
B_1 = -0.19774006E+01 B_2 = 0.98727357E+00

WDC2 = 0.40066761E+03 G(I) = 0.16142562E+06 A = 0.96548939E+00

FOR I = 2 A_0 = 0.10000000E+01 A_1 = 0.20000000E+01
A_2 = 0.10000000E+01 K_1 = 0.13321469E-02
B_1 = -0.19600541E+01 B_2 = 0.96557320E+00

WDC2 = 0.40066761E+03 G(I) = 0.16303127E+06 A = 0.96548939E+00

FOR I = 3 A_0 = 0.10000000E+01 A_1 = 0.20000000E+01
A_2 = 0.10000000E+01 K_1 = 0.30310788E-03
B_1 = -0.19519613E+01 B_2 = 0.95321709E+00

WDC2 = 0.40066761E+03 G(I) = 0.16388496E+06 A = 0.96548939E+00

W	H	H1	H2	H3
0.0000	0.89998E+00	0.96549E+00	0.96549E+00	0.96547E+00
0.0400	0.89999E+00	0.96549E+00	0.96549E+00	0.96548E+00
0.0800	0.90001E+00	0.96550E+00	0.96551E+00	0.96547E+00
0.1200	0.90009E+00	0.96552E+00	0.96554E+00	0.96550E+00
0.1600	0.90016E+00	0.96555E+00	0.96558E+00	0.96551E+00
0.2000	0.90028E+00	0.96558E+00	0.96564E+00	0.96555E+00
0.2400	0.90042E+00	0.96563E+00	0.96571E+00	0.96559E+00
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

WDC is the prewarped cutoff frequency.

WC is the cutoff frequency.

T is the sampling interval.

B, $A = \frac{1}{2}\left(\left(\sqrt{\varepsilon^{-2}+1}+\varepsilon^{-1}\right)^{1/2N} \pm \left(\sqrt{\varepsilon^{-2}+1}+\varepsilon^{-1}\right)^{-1/2N}\right)$

I is the i^{th} stage, I varies from 1 to N.

$K_8 = 2A\cos\theta$.

$K_2 = A^2\cos^2\theta + B^2\sin^2\theta$.

A_0, A_1, and A_2 are the low-pass filter numerator coefficients.

B_1 and B_2 are the low-pass filter denominator coefficients.

K_1 is the gain factor.

WDC2 = $(WDC)^2$.

$G(I) = \frac{4}{T^2} + \frac{2}{T}WDC \cdot K_8 + (WDC)^2 K_2$.

The A following G(I) is $a = \left[\frac{1}{1+\varepsilon^2}\right]^{1/2N}$

W is the frequency.

H is the overall magnitude of the digital transfer function.

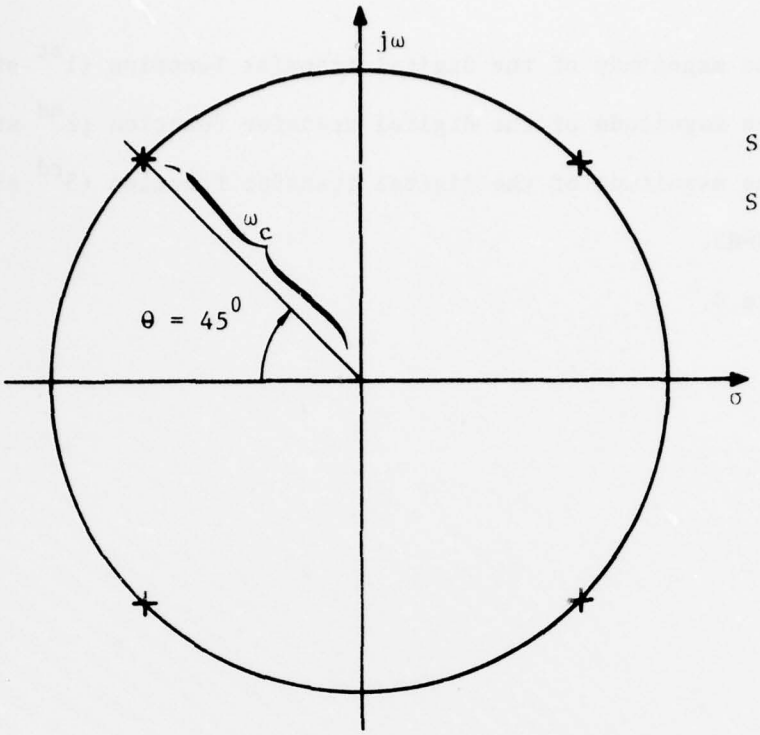
H1 is the magnitude of the digital transfer function (1^{st} stage).

H2 is the magnitude of the digital transfer function (2^{nd} stage).

H3 is the magnitude of the digital transfer function (3^{rd} stage).

H = H1*H2*H3.

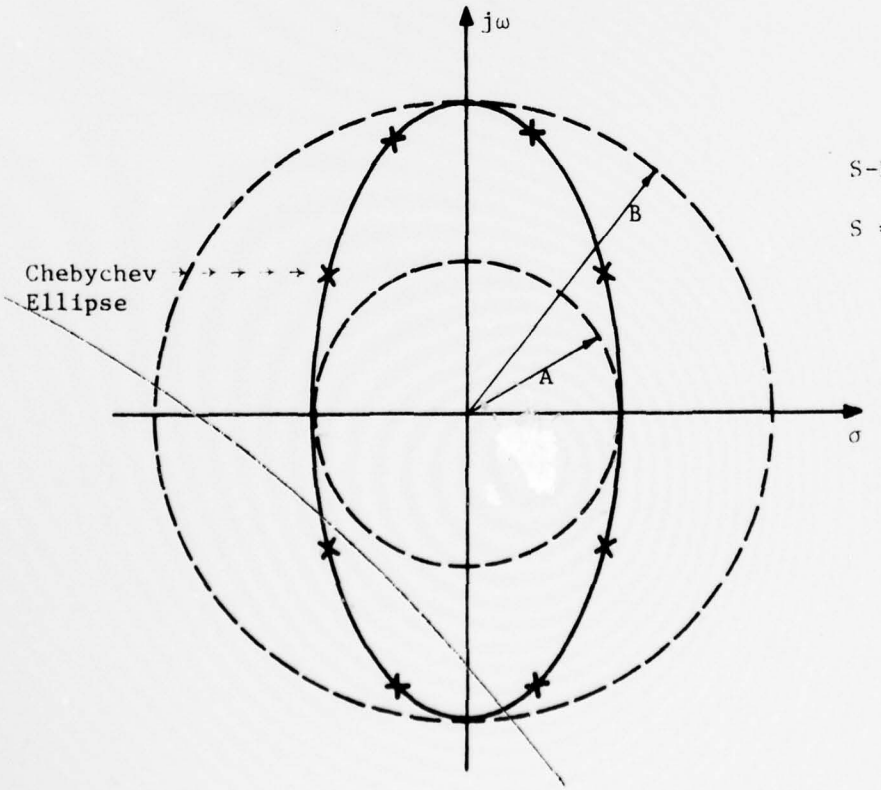
See Figure 8.



S-Plane

$S = \sigma + j\omega$

Figure 1



S-Plane

$S = \sigma + j\omega$

Figure 2

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

2ND ORDER BUTTERWORTH LOW-PASS FILTER

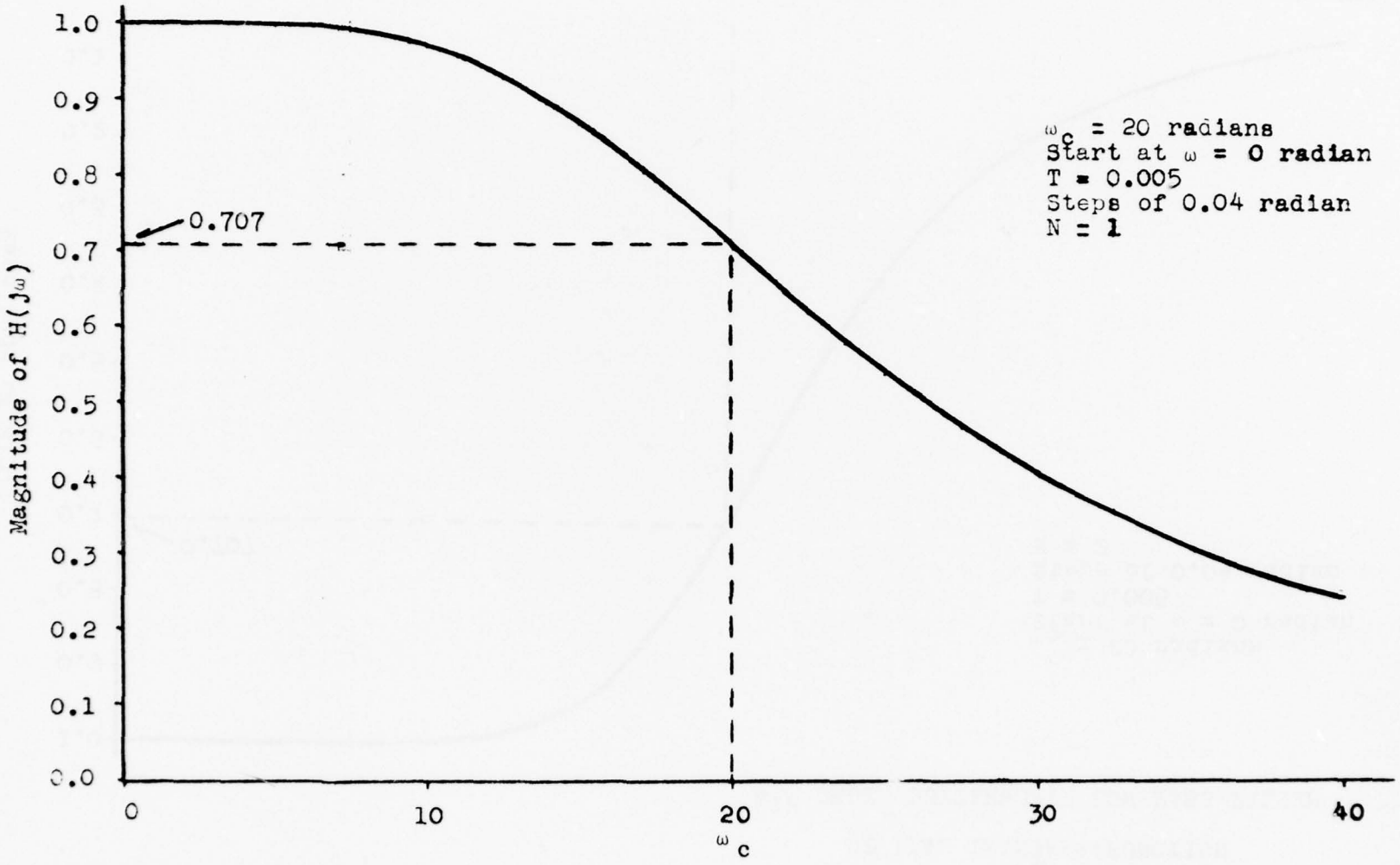


Figure 3

M GNITUDE VS FREQUENCY
FOR
DI(ITAL TRANSFER FUNCTION

4TH ORDE: BUTTERWORTH LOW-PASS FILTER

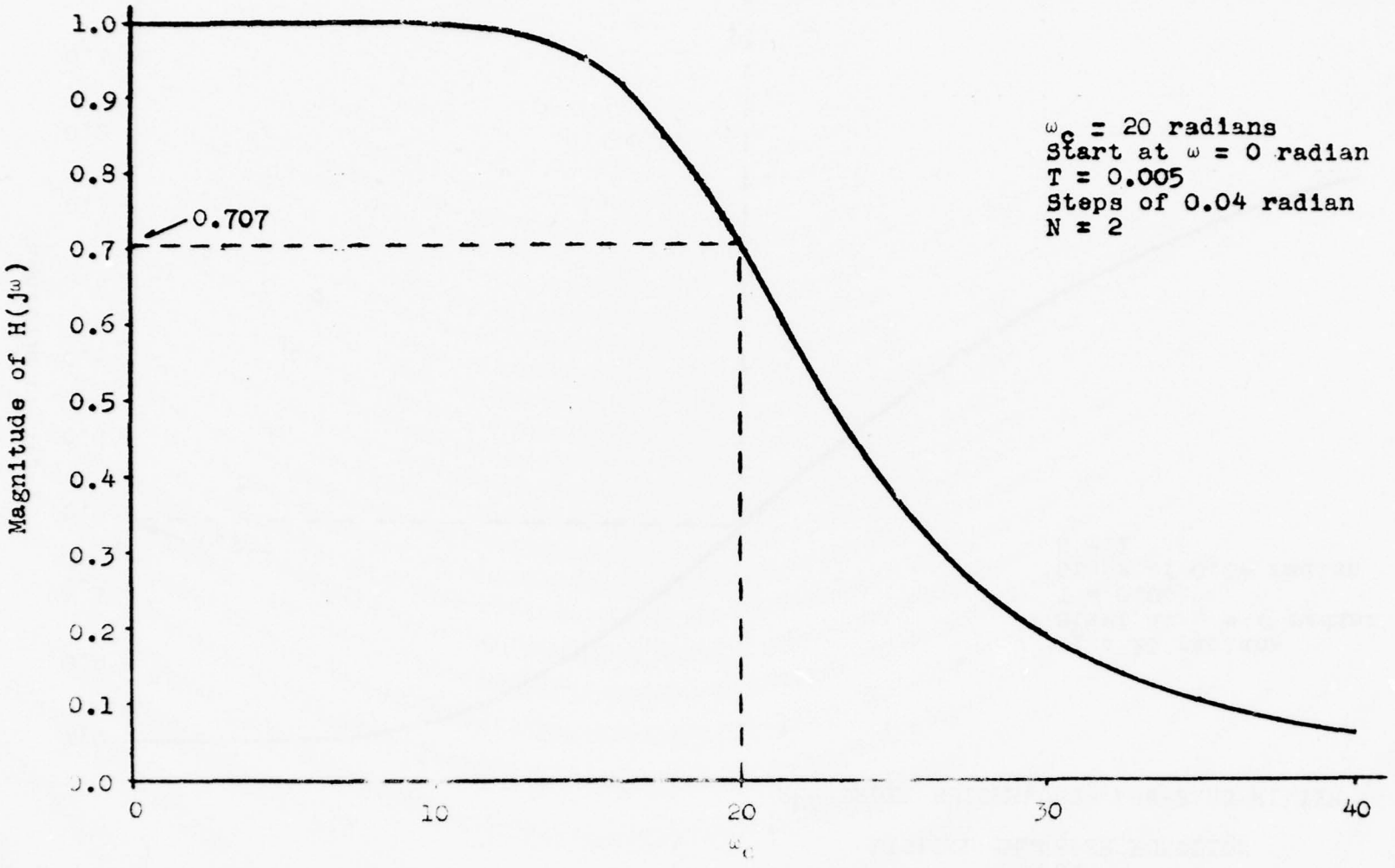
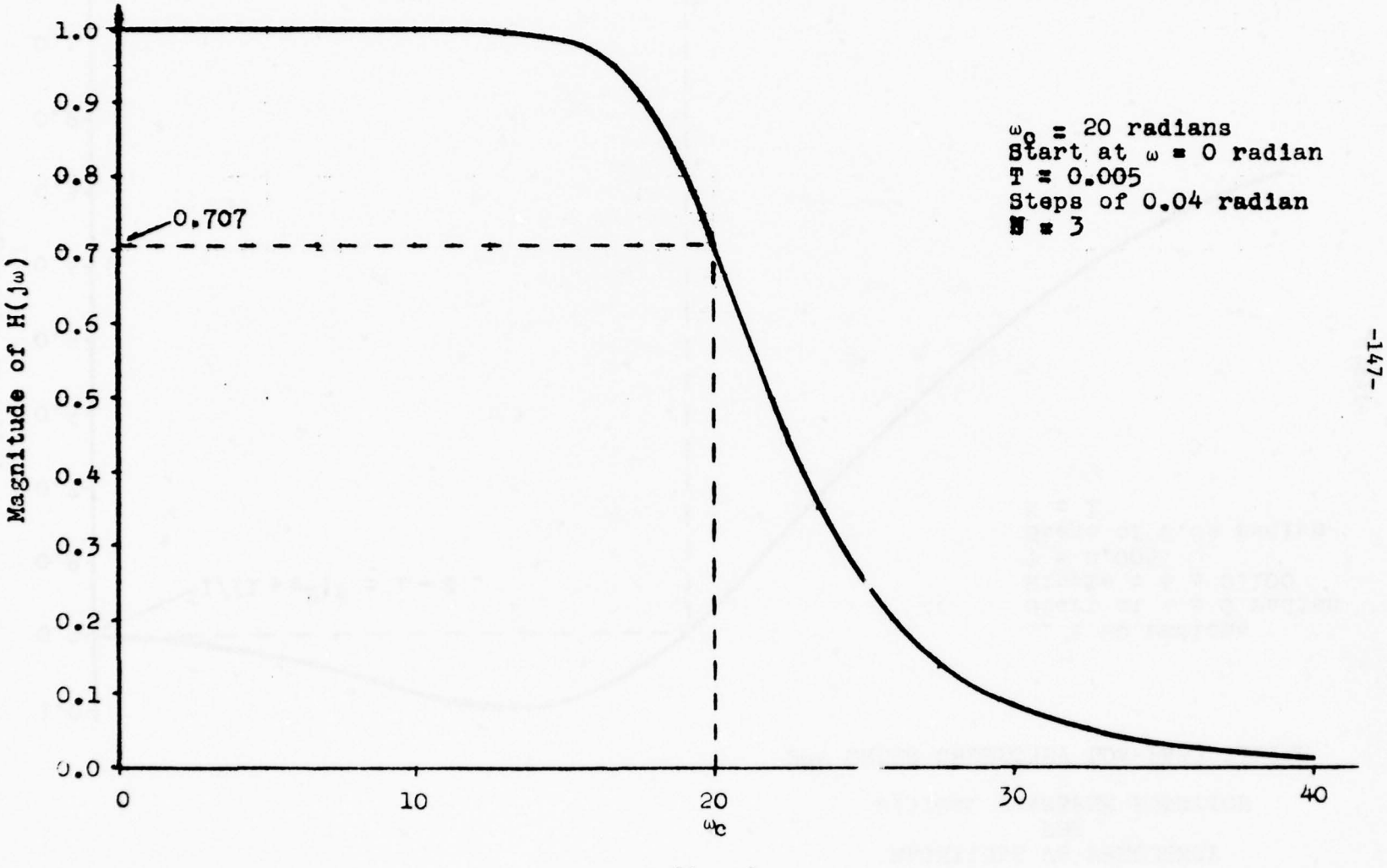


Figure 4

MA NITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

6^{TH} ORDER BUTTERWORTH LOW-PABS FILTER



ω_c = 20 radians
Start at ω = 0 radian
T = 0.005
Steps of 0.04 radian
N = 3

Figure 5

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

2ND ORDER CHEBYCHEV LOW-PASS FILTER

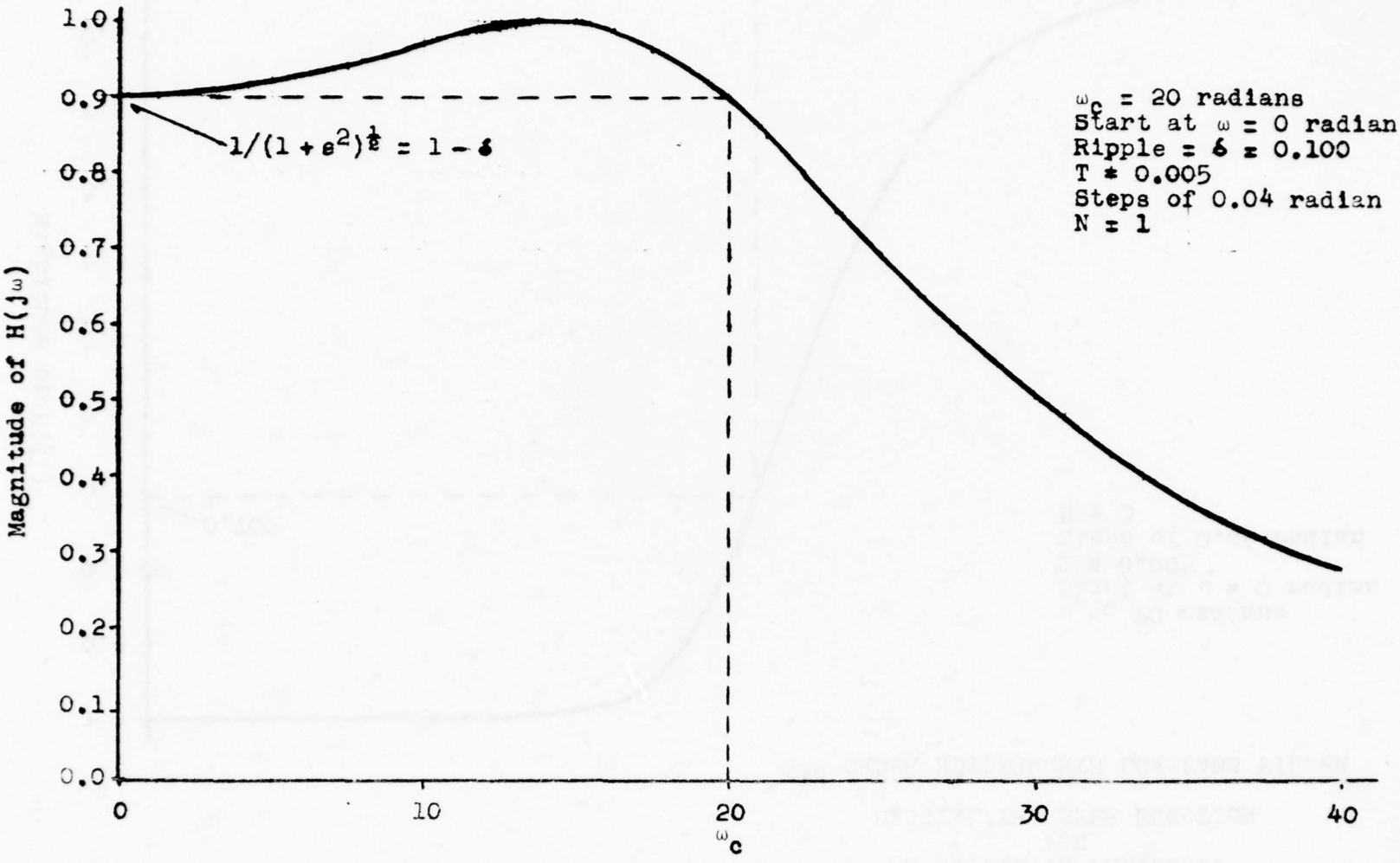
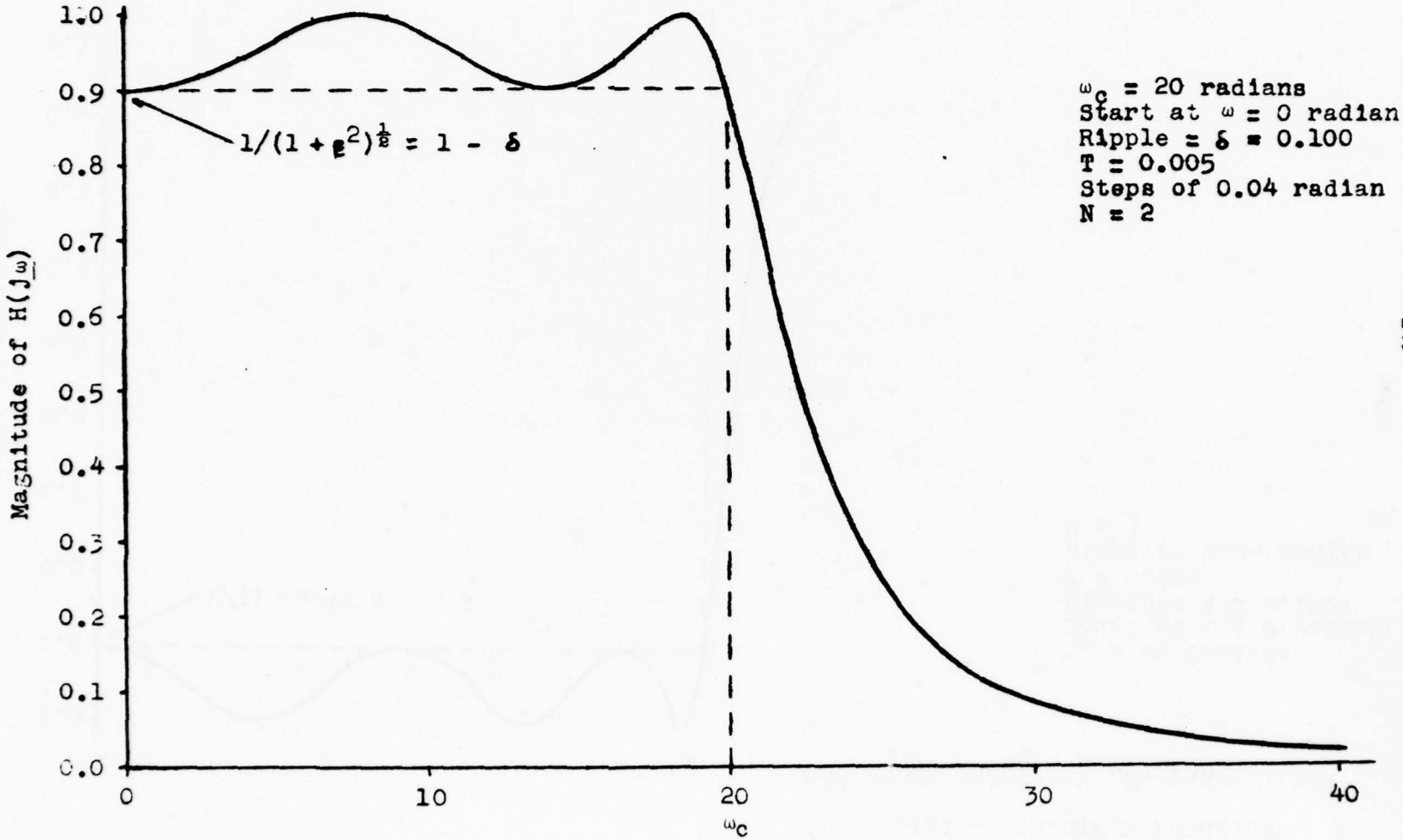


Figure 6

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

4^{TH} ORDER CHEBYCHEV LOW-PASS FILTER

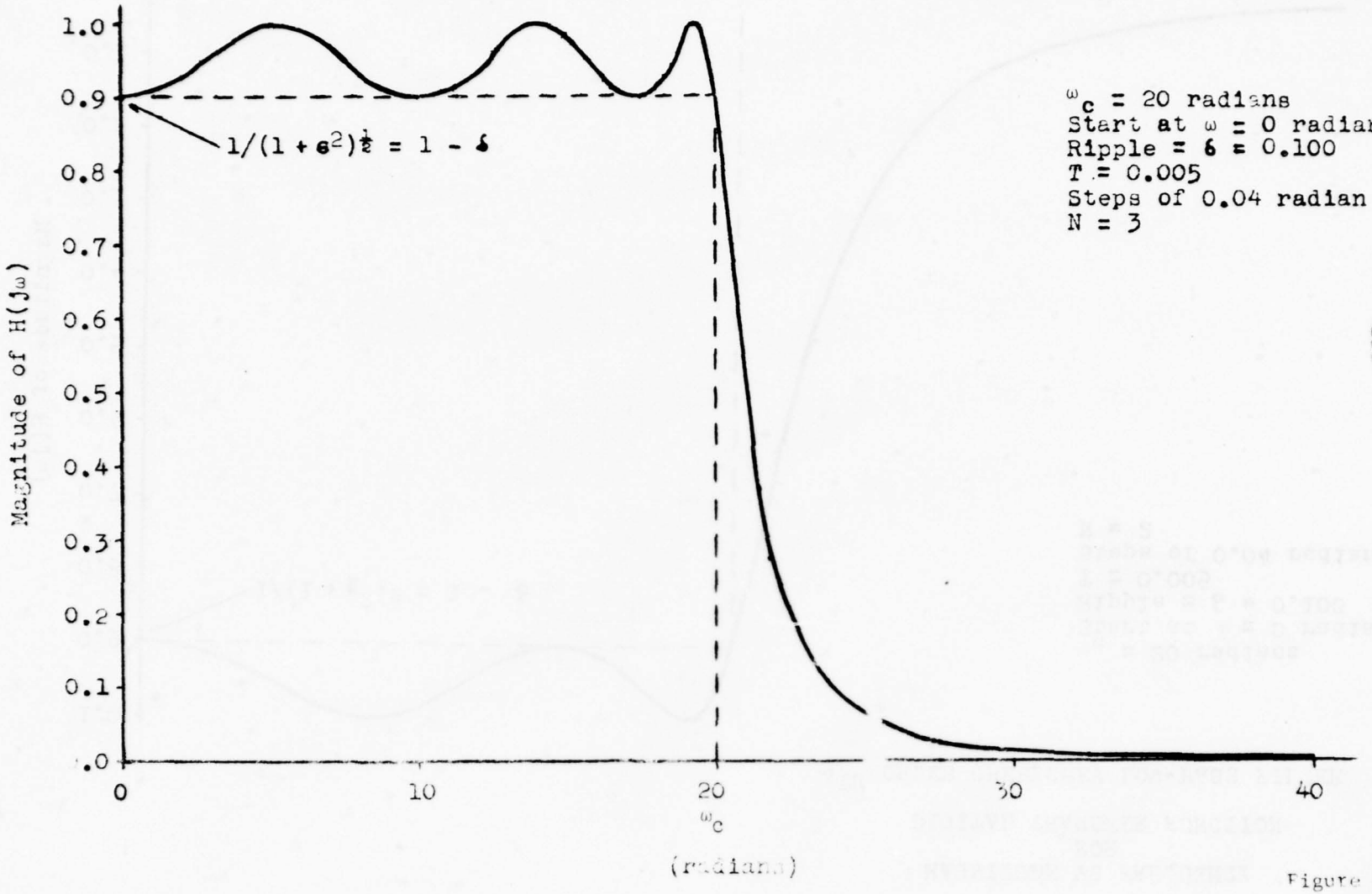


ω_c = 20 radians
Start at ω = 0 radian
Ripple = δ = 0.100
T = 0.005
Steps of 0.04 radian
N = 2

Figure 7

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

6^{TH} ORDER CHEBYCHEV LOW-PASS FILTER



ω_c = 20 radians
Start at ω = 0 radian
Ripple = δ = 0.100
T = 0.005
Steps of 0.04 radian
N = 3

Figure 8

References

A. Budak, Passive and Active Network Analysis and Synthesis, Houghton Mifflin Co., Boston, 1974.

D. Childers and A. Durling, Digital Filtering and Signal Processing, West Publishing Company, New York, 1975.

J. J. D'Azzo and C. H. Houpis, Linear Control System Analysis and Design, McGraw-Hill, Inc., New York, 1975.

B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill, Inc., New York, 1969.

B. J. Leon and P. A. Wintz, Basic Linear Networks for Electrical and Electronics Engineers, Holt, Rinehart, and Winston, Inc., New York, 1970.

L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.

L. Weinberg, Network Analysis and Synthesis, McGraw-Hill, Inc., New York, 1962.

M. E. Van Valkenburg, Modern Network Synthesis, John Wiley & Sons, Inc., New York, 1960.

Appendix E

A FORTRAN IV DESIGN PROGRAM FOR BUTTERWORTH AND CHEBYCHEV BAND-PASS AND BAND-STOP DIGITAL FILTERS

by

H. J. Markos

and

T. A. Brubaker

The authors are with the Electrical Engineering Department at Colorado State University, Fort Collins, Colorado.

INTRODUCTION

This report contains the documentation for the BPASS program. It consists of the design procedure used, a description of the program, and design examples using the program.

The purpose of the BPASS program is the design of either a maximally flat Butterworth or a Chebychev filter with equal ripple in the pass band. For each type of filter there is a choice of band-pass or band-stop filters. Starting with an analog filter, the bilinear Z transform is used to design an equivalent digital filter. The user enters the low-pass filter order, the type of filter desired, the sampling interval, the upper and lower cutoff frequencies, the starting frequency and frequency increment, and if a Chebychev filter is being designed, the ripple. The low-pass filter sections are transformed to second order band-pass or band-stop sections. Then the program generates the digital filter coefficients for up to six second order sections in cascade or up to a 12th order filter. The design is carried out in the frequency domain. The program calculates the transfer function coefficients for each second order section, the magnitude function for each section, and the final cascaded filter magnitude response over the frequency interval specified by the input.

The BPASS program, written in Fortran IV is supplied as a card deck with this report. The program is in the form of a subroutine and can be used as is by a call statement from the main program. Data may be input via cards with output available through a line printer. The input/output devices may be altered as explained in this report. Graphic routines may easily be appended to the program.

I. Design Procedure

A. Preliminary Discussion

One common method of designing a digital filter is to start with an analog transfer function $H(S)$ and transform it to the digital transfer function $H(Z)$.

The transfer function of a second order digital filter in the Z domain is given by

$$H(Z) = \frac{K_1(A_0Z^2 + A_1Z + A_2)}{Z^2 + B_1Z + B_2} \qquad (1)$$

where the A's and B's are the coefficients of the numerator and denominator respectively. This program will calculate the scale factor K_1 and the coefficients A_0, A_1, A_2, B_1, and B_2. The transformation used is the extended bilinear Z transform

$$S \rightarrow \frac{2}{T}\left(\frac{Z - 1}{Z + 1}\right) \quad , \qquad (2)$$

where T is the sampling interval. When the extended bilinear Z transform is employed, the desired frequencies must first be pre-warped to make them compatible with the digital filter. In the band-pass and band-stop filters, the upper and lower cutoff frequencies and the center frequency of the filter are of interest. Calling the upper and lower frequencies ω_u and ω_l respectively, the pre-warped upper (WDU), lower (WDL), and center (WDM) frequencies and the bandwidth between WDU and WDL are found by

$$
\begin{aligned}
WDU &= \frac{2}{T}\tan\left(\frac{\omega_u T}{2}\right) \\
WDL &= \frac{2}{T}\tan\left(\frac{\omega_l T}{2}\right) \\
WDM &= \frac{2}{T}\tan\left[\frac{\sqrt{\omega_u \omega_l}\,T}{2}\right] \\
WB &= WDU - WDL
\end{aligned}
\tag{3}
$$

ω_u and ω_l are specified by the designer and the prewarping is done by the program.

In the design procedure for all band-pass and band-stop filters of order n', (n' even), the program begins by first finding the poles for the corresponding $n'/2$ order low-pass filter. The low-pass filter is then transformed into a band-pass or band-stop filter of order n', ($n' = 2n$).

B. Butterworth Band-Pass Filter

We start with a normalized second order low-pass Butterworth filter transfer function in the S plane

$$H(S) = \frac{1}{S^2 + 2S\cos\theta + 1} \tag{4}$$

where the angle θ is in degrees (in the program) and may be found from the Butterworth circle and the relationship

$$S = e^{\pm j\pi(2m - 1)/2n} \tag{5}$$

where n is the order of the low-pass filter and $m = 1, 2, \ldots, n$. This relationship is determined by the following procedure. By definition, a filter is nth order Butterworth low-pass if its gain characteristic is

$$|H_n(j\omega)|^2 = \frac{a^2}{1 + (\frac{\omega}{\omega_c})^{2n}} \tag{6}$$

where a is the DC gain, ω_c is the desired cutoff frequency and n is the order of the low-pass filter.

In the design, the poles of $H(S)$ must be found. The procedure is as follows:

$$|H_n(j\omega)|^2 = H_n(j\omega)\overline{H_n(j\omega)} = H_n(j\omega)H_n\overline{(j\omega)} = H_n(j\omega)H_n(-j\omega)$$

$$= [H(S)H(-S)]_{S = j\omega} = \left[\frac{a^2}{1 + (\frac{\omega}{\omega_c})^{2n}}\right]_{\omega = \frac{S}{j}} = \frac{a^2}{1 + (\frac{S}{j\omega_c})^{2n}}$$

$$= \frac{a^2}{1 + \left[-\frac{S^2}{\omega_c^2}\right]^n} = \begin{cases} \dfrac{a^2}{1 + \left[\dfrac{S^2}{\omega_c^2}\right]^n} & \text{, for } n \text{ even} \\ \dfrac{a^2}{1 - \left[\dfrac{S^2}{\omega_c^2}\right]^n} & \text{, for } n \text{ odd} \end{cases} \tag{7}$$

Setting the denominators equal to zero,

$$\frac{S}{\omega_c} = (\pm 1)^{1/2n} \quad . \tag{8}$$

Thus, the pole locations are the $2n$ roots of ± 1, depending on whether the low-pass filter order is odd or even. These roots are located on a circle with radius ω_c centered at the origin of the S plane and have symmetry with respect to both real and imaginary axes.

For n odd, a pair of roots are on the real axis and the rest are separated by π/n radians. For n even, a pair of roots are located $\pi/2n$ radians from the real axis and the rest are again separated by π/n radians. No roots are on the imaginary axis, for either even or odd n.

Let $p_1,\ldots,p_{2n}$ be the roots. From the symmetry of the pole locations, if $p_1,\ldots,p_n$ are the roots lying in the right-half plane, the left-half plane roots are $-p_1,\ldots,-p_n$. The magnitude-squared function can then be written as

$$H_n(S)H_n(-S) = \frac{a^2(-1)^n\omega_c{}^{2n}}{(S + p_1)\ldots(S + p_n)(S - p_1)\ldots(S - p_n)} \quad . \tag{9}$$

To be stable, $H_n(S)$ must have all its poles in the left-hand plane, thus

$$H_n(S) = \frac{a\omega_c{}^n}{(S + p_1)\ldots(S + p_n)} \quad . \tag{10}$$

The program is written with unity gain at DC, ($\omega = 0$), therefore $a = 1$.

In order to locate the poles as specified above, consider the following set of equations.

$$\begin{aligned} 1 &= -e^{\pm j\pi(2m - 1)} \quad , m = 1, 2,\ldots, n; \text{ for } n \text{ even} \\ -1 &= -e^{\pm j2\pi k} \quad , k = 0, 1,\ldots, n; \text{ for } n \text{ odd} \end{aligned} \tag{11}$$

Substituting equations (11) into equations (8) yields

$$\begin{aligned} [\frac{S}{\omega_c}]_{\pm m} &= -e^{\pm j\pi(2m - 1)/2n} \quad , \quad m = 1, 2,\ldots, n; \text{ for } n \text{ even} \\ [\frac{S}{\omega_c}]_{\pm k} &= -e^{\pm j\pi k/n} \quad , k = 0, 1,\ldots, n; \text{ for } n \text{ odd} \end{aligned} \tag{12}$$

Equations (12) will give the pole locations as described above. Consider the form of equations (12)

$$S = -\omega_c e^{\pm j\theta} = \omega_c[-\cos\theta \pm j\sin\theta] \quad . \tag{13}$$

From this relationship, it can been seen that the magnitude for each pole is ω_c, regardless of the angle, and thus all the poles lie on a circle with radius ω_c.

As an example, consider a second order filter, n = 2.

$$[\frac{S}{\omega_c}]_{\pm m} = -e^{\pm j\pi(2m - 1)/4} \qquad m = 1, 2$$

$$S_{\pm 1} = \omega_c \underline{/\pm 45°}$$

$$S_{\pm 1} = \omega_c \underline{/\pm 135°}$$

$$\theta = 45°$$

The relationship of these roots about the circle of radius ω_c is illustrated in Figure 1. The angle θ is always measured from the negative real axis.

In the program, only the angle(s) less than 90° are considered so that the poles lie in the left-half plane because poles in the left-half plane are stable. Putting $\theta = 45°$ into equation (4) yields poles at -0.707 ±j0.707. These locations are in the left-half plane. From equations (12), for low-pass filter orders n = 1, 2,..., 6, the values of θ are given below.

Low-Pass Filter Order	Angle	Second Order Cascaded Sections	Band-Pass Band-Stop Filter Order
n	θ	N	n'
1	0°	1	2
2	45°	2	4
3	60°, 0°	3	6
4	22.5°, 67.5°	4	8
5	72°, 36°, 0°	5	10
6	75°, 45°, 15°	6	12

n is the order of the low-pass filter and is used to determine pole locations. n is also the number of second order band-pass or band-stop sections which results from the transformation of the low-pass filter sections and which will be cascaded to form the band-pass or band-stop filters of order n'. The transformation is explained below. The calculated angles are incorporated in the program in the order given above.

Given the normalized second order low-pass transfer function equation (4), we transform this low-pass into a band-pass transfer function for some bandwidth WB, and center frequency WDM by using the transform

$$S \rightarrow \frac{S^2 + WDM^2}{SWB} \quad . \tag{14}$$

Equation (4) then transforms to a 4th order transfer function

$$H(S) = \frac{S^2 WB^2}{S^4 + S^3\, 2WB\cos\theta + S^2(2WDM^2 + WB^2) + S2WB\, WDM^2\, \cos\theta + WDM^4} \quad . \tag{15}$$

Using the root finding subroutine "POLRT" from the IBM Scientific Subroutine Package (SSP), the roots of the denominator of equation (15)

are found. (Note: POLRT has been attached to BPASS as a double precision subroutine and is included in the card deck). The roots found will be complex conjugate pairs. Calling the real and imaginary parts of the pairs RE_1, AIM_1, RE_2, AIM_2 equation (15) is factored to yield two cascaded second order sections

$$H(S) = \frac{SWB}{S^2 - 2SRE_1 + RE_1^2 + AIM_1^2} \cdot \frac{SWB}{S^2 - 2SRE_2 + RE_2^2 + AIM_2^2} . \qquad (16)$$

For each θ of a given N, the program calculates roots for both sections of equation (16) and labels them the ith and the ith + 1 section. If N, the number of second order sections specified, is even, the program will calculate N pairs of RE and AIM values or $2N = n'$ roots. If N is odd, the last value of θ is 0. Substituting $\theta = 0$ into equation (4) and factoring yields two identical first order sections, $1/(S + 1)$. The program will calculate N + 1 pairs of RE and AIM values, but because the last two pairs are the same due to the identical first order sections, the last pair will not be used.

Because both second order sections of equation (16) are of the same format, we will deal with only one section, the ith section and let

$$\begin{aligned} -2RE_i &= D_i \\ RE_i^2 + AIM_i^2 &= C_i \end{aligned} \qquad (17)$$

The design of an n'th order band-pass or band-stop filter leads to n'/2 second order sections. Substituting equations (17) into one

section of equation (16) yields the transfer function for the ith section

$$H_i(S) = \frac{SWB}{S^2 + SD_i + C_i} . \tag{18}$$

The extended bilinear Z transform, equation (2) is used to get to the digital domain. Employing equation (2) on equation (18) yields $H_i(Z)$ for the ith second order section.

$$H_i(Z) = \frac{\frac{2}{T}WBZ^2 - \frac{2}{T}WB}{Z^2(\frac{4}{T^2} + \frac{2D_i}{T} + C_i) + Z(2C_i - \frac{8}{T^2}) + (\frac{4}{T^2} - \frac{2D_i}{T} + C_i)} \tag{19}$$

Putting the denominator of equation (19) in monic form yields the transfer function for the ith second order stage of the filter

$$H_i(Z) = \frac{K_{1i}(A_0Z^2 + A_1Z + A_2)}{Z^2 + B_{1i}Z + B_{2i}} \tag{20}$$

This equation is the same as equation (1) with the exception of the subscripts. For all four filter types discussed here, the scale factor, K_1, and coefficients B_1 and B_2 are a function of the section calculated, while the coefficients A_0, A_1, and A_2 are the same for all sections calculated. In going from equation (19) to equation (20) we have

$$A_0 = \frac{2}{T}WB$$

$$A_1 = 0$$

$$A_2 = -\frac{2}{T}WB$$

$$G_i = \frac{4}{T^2} + \frac{2D_i}{T} + C_i \qquad (21)$$

$$K_{1i} = \frac{1}{G_i}$$

$$B_{1i} = \frac{2C_i - \frac{8}{T^2}}{G_i}$$

$$B_{2i} = \frac{\frac{4}{T^2} - \frac{2D_i}{T} + C_i}{G_i}$$

Letting $Z = e^{ST} = e^{j\omega T}$ for $S = j\omega$ and taking the magnitude of $H_i(j\omega)$ we have

$$|H_i(j\omega)| = K_{1i} \frac{\sqrt{(A_0\cos(2\omega T) + A_1\cos(\omega T) + A_2)^2 + (A_0\sin(2\omega T) + A_1\sin(\omega T))^2}}{\sqrt{(\cos(2\omega T) + B_{1i}\cos(\omega T) + B_{2i})^2 + (\sin(2\omega T) + B_{1i}\sin(\omega T))^2}} \qquad (22)$$

The magnitude function, equation (22) is the same for all the filters discussed in this report.

C. Butterworth Band-Stop Filter

The design procedure is almost exactly the same as that of the Butterworth band-pass filter, except that the transformation to band-stop is the reciprocal of equation (14), i.e.

$$S \rightarrow \frac{SWB}{S^2 + WDM^2} \qquad (23)$$

and we find $H_i(S)$ to be

$$H_i(S) = \frac{S^2 + WDM^2}{S^2 + SD_i + C_i} \quad . \tag{24}$$

After employing the extended bilinear Z transform, equation (2), we have

$$\begin{aligned} A_0 &= A_2 = \frac{4}{T^2} + WDM^2 \\ A_1 &= 2WDM^2 - \frac{8}{T^2} \end{aligned} \tag{25}$$

and B_{1i}, B_{2i}, K_{1i} are the same functions of C_i and D_i as in equation (21). These coefficients are then used in the calculation of equation (22) to find $|H_i(j\omega)|$.

D. Chebychev Band-Pass Filter

The Chebychev filter ripples with equal amplitude in the pass-band. The amount of ripple is specified by the quantity δ (labeled RIP in the program). The poles of the filter are found on an ellipse described by two Butterworth circles of radii A and B with A < B. The location of the poles on the ellipse is a function of the ripple and is given by the following equation:

$$B, A = \frac{1}{2}\left(\left(\sqrt{\varepsilon^{-2} + 1} + \varepsilon^{-1}\right)^{1/N} \pm \left(\sqrt{\varepsilon^{-2} + 1} + \varepsilon^{-1}\right)^{-1/N}\right) \tag{26}$$

where

$$\varepsilon = \left[\frac{1}{(1 - \delta)^2} - 1\right]^{\frac{1}{2}} \tag{27}$$

and N is numerically equal to the order of the low-pass filter which is transformed to yield the band-pass filter. B is given

for the plus sign and A for the minus sign. The Chebychev ellipse then has major axis B and minor axis A. The location of the S plane poles on the ellipse is given by

$$\text{Real Part} = A\cos\theta$$
$$\text{Imaginary Part} = \beta\sin\theta \qquad (28)$$

The θ's are the same as given for the corresponding order Butterworth filter. An example of Chebychev pole locations is illustrated in Figure 2. For $A = \frac{1}{2}$ and $B = 1$ in a fourth order filter, $\theta = 22.5°$ and $67.5°$. The Chebychev pole locations are determined from equations (26), (27) and (28).

The analog second order Chebychev low-pass filter is

$$H(S) = \frac{K_2\left[\frac{1}{\sqrt{1+\varepsilon^2}}\right]^{2/N}}{S^2 + K_8 S \quad K_2} . \qquad (29)$$

ε is calculated from equation (27) and N is equal to the order of the low-pass filter which is transformed to yield the band-pass filter. K_8 and K_2 are calculated by

$$K_8 = 2A\cos\theta \qquad (30)$$
$$K_2 = A^2\cos^2\theta + B^2\sin^2\theta \quad . \qquad (31)$$

The substitution of the low-pass to band-pass transformation, equation (14), into equation (29) yields

$$H(S) = \frac{S^2 WB^2 K_2\left[\frac{1}{\sqrt{1+\varepsilon^2}}\right]^{2/N}}{S^4 + S^3 K_8 WB + S^2(2WDM^2 + K_2 WB^2) + SK_8 WDM^2 WB + WDM^4} . \qquad (32)$$

After finding the roots of equation #32) and making the substitutions given by equations (17) we find the ith second order section

$$H_i(S) = \frac{SWK_3}{S^2 + SD_i + C_i} \quad ,$$

$$K_3 = \sqrt{K_2}\left[\frac{1}{\sqrt{1 + \varepsilon^2}}\right]^{1/N} \quad . \tag{33}$$

Applying the extended bilinear Z transform equation (2) yields an equation of the form of equation (20) where

$$\begin{aligned} A_0 &= 1 \\ A_1 &= 0 \\ A_2 &= -1 \\ K_{1i} &= \frac{2WB}{T} \cdot \frac{K_3}{G_i} \end{aligned} \tag{34}$$

B_{1i} and B_{2i} are the same functions of C_i and D_i given by equations (21). These coefficients are then used in equation (22) to find $|H_i(j\omega)|$.

E. Chebychev Band-Stop Filter

Given equation (29) for H(S) we apply the low-pass to band-stop transformation equation (23) to obtain the 4th order transfer function

$$H_i(S) = \frac{(S^2 + WDM^2)^2 K_2\left[\frac{1}{\sqrt{1 + \varepsilon^2}}\right]^{2/N}}{K_2S^4 + S^3K_8WB + S^2(WB^2 + 2K_2WDM^2) + SK_8WDM^2WB + K_2WDM^4} \quad . \tag{35}$$

N is equal to the order of the low-pass filter which is transformed to yield the band-stop filter.

After finding the roots of equation (35) and making the substitutions given by equations (17) the ith second order section is

$$H_i(S) = \frac{(S^2 + WDM^2)K_3}{S^2 + SD_i + C_i} \quad ,$$

$$K_3 = \sqrt{K_2}\left[\frac{1}{\sqrt{1 + \varepsilon^2}}\right]^{1/N} \quad . \tag{36}$$

Applying the extended bilinear Z transform equation (2) yields an equation of the form of equation (20) where

$$A_0 = A_2 = \frac{4}{T^2} + WDM^2$$

$$A_1 = 2WDM^2 - \frac{8}{T^2} \tag{37}$$

$$K_{1i} = \frac{K_3}{G_i}$$

B_{1i} and B_{2i} are the same functions of C_i and D_i given by equations (21). These coefficients are then used in equation (22) to find $|H_i(j\omega)|$.

II. Using the Program

The first data card read into the program contains the number of second order sections to be cascaded, N, and the type of filter desired, KN. N is equal to 1, 2,..., or 6, which corresponds to the order of the low-pass filter, and hence corresponds to the 2nd, 4th,..., or 12th order band pass or band stop filter respectively. KN is the type of filter desired. The values of KN specifies one of the four choices given by

KN	Type
1	Butterworth Band-Pass
2	Butterworth Band-Stop
3	Chebychev Band-Pass
4	Chebychev Band-Stop

The format on the N, KN card is 2I2.

The second data card read in is the sampling interval T in F10.6 format. When choosing T, 1/T should be approximately equal to ten times the center frequency (WDM).

The third data card read in contains the values of the upper and lower cutoff frequencies, ω_u and ω_l, in 2F10.4 format. For the Butterworth filters, the cutoff frequencies are the -3db cutoff frequencies. For the Chebychev filters, the magnitude of the response is $1/(1 + \varepsilon^2)^{1/2} = 1 - \delta$ at the cutoff frequencies. ω is in radians. δ is the ripple factor.

If the desired filter is Chebychev, i.e., KN = 3 or 4, the next data card contains the ripple (RIP) factor in F5.3 format. If the desired filter is Butterworth, i.e., KN = 1 or 2, this card is omitted from the data deck.

The final data card is the starting frequency (FREQ1) and the frequency increments (DELT) in radians. The format of the FREQ1, DELT card is 2F10.4. Determine DELT by the following:

$$\text{DELT} = \frac{\text{final frequency - starting frequency}}{1024} \quad .$$

This is necessary because there are 1024 frequency data points calculated in the program. Choose FREQ1 and DELT to insure that calculated values will include the data of interest. For maximum efficiency of the program, DELT should be a multiple of 2^{-K} so no decimal to binary conversion errors are incurred.

The digital filter coefficients are computed and printed out for each second order section. The full filter magnitude response, as well as each section magnitude response, is printed for each of the specified frequency increments. When there is only one second order section, the section magnitude response is the full filter magnitude response and is only printed once.

The program may be easily modified to incorporate a graphics display of the magnitude response. There is a comment card in the BPASS program indicating where the graphics subroutine call card should be inserted.

The program is written with input obtained via device 4 and output written to device 6. These numbers should be assigned to the appropriate devices prior to running the program.

The program was developed on a PDP-11/20 with a DOS/BATCH operating system. Trial runs frequently used a TTY terminal as well as a card reader for input (device 4); and a TTY terminal as well as a line printer for output (device 6). Double precision arithmetic

is employed. To decrease required memory storage, only the frequency interval values and the full magnitude response are saved. The section magnitude responses are printed out, but are not stored. The program will produce approximately 21 pages of output.

Shown below are sample deck set-ups.

Data Card	Format	Example
1	2I2	0504 (5 sections Chebychev band-stop)
2	F10.6	0.002 (T = 0.002)
3	2F10.4	60 40 (ω_u = 60, ω_l = 40 radians)
4	F5.3	0.10 (Ripple amplitude = 0.10)
5	2F10.4	0 0.1 (Start at ω = 0. Steps of 0.1 radian. Will finish just past ω = 102 radians.)
1	2I2	0401 (4 sections Butterworth band-pass)
2	F10.6	0.002 (T = 0.002)
3	2F10.4	60 40 (ω_u = 60, ω_l = 40 radians)
4	2F10.4	0 0.1 (Start at ω = 0. Steps of 0.1 radian. Will finish just past ω = 102 radians).

The following pages contain annotated examples of output data.

This is an example of the output for an 8th order Butterworth band-stop filter (0402) with $T = 0.002$, $\omega_u = 60$ radians, and $\omega_l = 40$ radians. The starting frequency is 0 radian and the frequency increment is 0.1 radian.

```
WDU = 60.07210 WDL = 40.02135 WDM = 49.02902 WB = 20.05076
     T = 0.20000E-02

THE ROOTS OF THE FILTER ARE GIVEN BELOW
REAL(1) = -8.52659476  IMAGINARY(1) = -44.46786740
REAL(2) = -9.95788916  IMAGINARY(2) = -52.14095970
REAL(3) = -4.50676557  IMAGINARY(3) = -59.01588870
REAL(4) = -3.12232691  IMAGINARY(4) = -40.49140500

THE COEFFICIENTS OF EACH DIGITAL FILTER SECOND ORDER SECTION ARE
     GIVEN BELOW
FOR I = 1  A0 =  0.10024038E+07  A1 = -0.19951923E+07
           A2 =  0.10024038E+07  K1 =  0.98125481E-06
           B1 = -0.19584863E+01  B2 =  0.96653295E+00

FOR I = 2  A0 =  0.10024038E+07  A1 = -0.19951923E+07
           A2 =  0.10024038E+07  K1 =  0.97769448E-06
           B1 = -0.19498774E+01  B2 =  0.96090048E+00

FOR I = 3  A0 =  0.10024038E+07  A1 = -0.19951923E+07
           A2 =  0.10024038E+07  K1 =  0.98755180E-06
           B1 = -0.19681836E+01  B2 =  0.98202353E+00

FOR I = 4  A0 =  0.10024038E+07  A1 = -0.19951923E+07
           A2 =  0.10024038E+07  K1 =  0.99216788E-06
           B1 = -0.19[illegible]E+01  B2 = [illegible]
```

W	H	H1	H2	H3	H4
0.0000	0.10000E+01	0.11726E+01	0.85284E+00	0.68611E+00	0.14575E+01
0.1000	0.10000E+01	0.11726E+01	0.85284E+00	0.68611E+00	0.14575E+01
0.2000	0.10000E+01	0.11726E+01	0.85284E+00	0.68611E+00	0.14575E+01
0.3000	0.99999E+00	0.11726E+01	0.85283E+00	0.68610E+00	0.14575E+01
0.4000	0.10000E+01	0.11726E+01	0.85283E+00	0.68609E+00	0.14576E+01
0.5000	0.10000E+01	0.11726E+01	0.85282E+00	0.68609E+00	0.14576E+01
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

WDU is the prewarped upper frequency.

WDL is the prewarped lower frequency.

WDM is the prewarped center frequency.

WB is the bandwidth, WDU - WDL.

T is the sampling interval.

The Real and Imaginary part of the roots of the filter are given next.

I is the ith stage. I varies from 1 to N.

A_0, A_1, A_2 are the Butterworth band-stop filter numerator coefficients.

K_1 is the gain factor.

B_1, and B_2 are the Butterworth band-stop filter denominator coefficients.

W is the frequency

H is the overall magnitude of the digital transfer function

H1 is the magnitude of the digital transfer function (1st stage).

H2 is the magnitude of the digital transfer function (2nd stage).

H3 is the magnitude of the digital transfer function (3rd stage).

H4 is the magnitude of the digital transfer function (4th stage).

See Figure 4.

This is an example of the output for an 8th order Chebychev band-pass filter (0403) with T = 0.002, ω_u = 60 radians, and ω_l = 40 radians. The starting frequency is 0 radian, the frequency increment is 0.1 radian, and the ripple is 0.1.

```
WDU = 60.07210 WDL = 40.02135 WDM = 49.02902 WB = 20.05076
     T = 0.20000E-02

A = 0.37642105  B = 1.06850027  K8 = 0.69553541  K2 = 0.28813942
A = 0.37642105  B = 1.06850027  K8 = 0.28810020  K2 = 0.99524620

THE ROOTS OF THE FILTER ARE GIVEN BELOW
REAL(1) = -3.19528085    IMAGINARY(1) = -44.97792290
REAL(2) = -3.77772492    IMAGINARY(2) = -53.17662810
REAL(3) = -1.15829660    IMAGINARY(3) = -40.10115290
REAL(4) = -1.73001696    IMAGINARY(4) = -59.89456990

THE COEFFICIENTS OF EACH DIGITAL FILTER SECOND ORDER SECTION ARE
     GIVEN BELOW
FOR  I = 1   A0 =  0.10000000E+01   A1 = 0.00000000E+00
             A2 = -0.10000000E+01   K1 = 0.10395603E-01
             B1 = -0.19792607E+01   B2 = 0.98732565E+00

FOR  I = 2   A0 =  0.10000000E+01   A1 = 0.00000000E+00
             A2 = -0.10000000E+01   K1 = 0.10375296E-01
             B1 = -0.19737935E+01   B2 = 0.98504460E+00

FOR  I = 3   A0 =  0.10000000E+01   A1 = 0.00000000E+00
             A2 = -0.10000000E+01   K1 = 0.19406846E-01
             B1 = -0.19889723E+01   B2 = 0.99538493E+00

FOR  I = 4   A0 =  0.10000000E+01   A1 = 0.00000000E+00
             A2 = -0.10000000E+01   K1 = 0.19346636E-01
             B1 = -0.19788675E+01   B2 = 0.99312838E+00
```

W	H	H1	H2	H3	H4
0.0000	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
0.1000	0.12493E-12	0.51560E-03	0.36886E-03	0.12106E-02	0.54265E-03
0.2000	0.19990E-11	0.10312E-02	0.73773E-03	0.24211E-02	0.10853E-02
0.3000	0.10121E-10	0.15468E-02	0.11066E-02	0.36318E-02	0.16280E-02
0.4000	0.31991E-10	0.20625E-02	0.14755E-02	0.48426E-02	0.21707E-02
0.5000	0.78116E-10	0.25783E-02	0.18445E-02	0.60536E-02	0.27134E-02
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

WDU is the prewarped upper frequency.

WDL is the prewarped lower frequency.

WDM is the prewarped center frequency.

WB is the bandwidth, WDU - WDL.

T is the sampling interval.

$$B, A = \frac{1}{2}((\sqrt{\epsilon^{-2} + 1} + \epsilon^{-1})^{1/N} \pm (\sqrt{\epsilon^{-2} + 1} + \epsilon^{-1})^{-1/N})$$

$$K_8 = 2A\cos(\theta)$$

$$K_2 = A^2\cos^2(\theta) + B^2\sin^2(\theta)$$

The Real and Imaginary part of the roots of the filter are given next.

I is the ith stage. I varies from 1 to N.

A_0, A_1, A_2 are the Chebychev band-pass filter numerator coefficients.

K_1 is the gain factor.

B_1, and B_2 are the Chebychev band-pass filter denominator coefficients.

W is the frequency.

H is the overall magnitude of the digital transfer function.

H1 is the magnitude of the digital transfer function (1st stage).

H2 is the magnitude of the digital transfer function (2nd stage).

H3 is the magnitude of the digital transfer function (3rd stage).

H4 is the magnitude of the digital transfer function (4th stage).

See Figure 5.

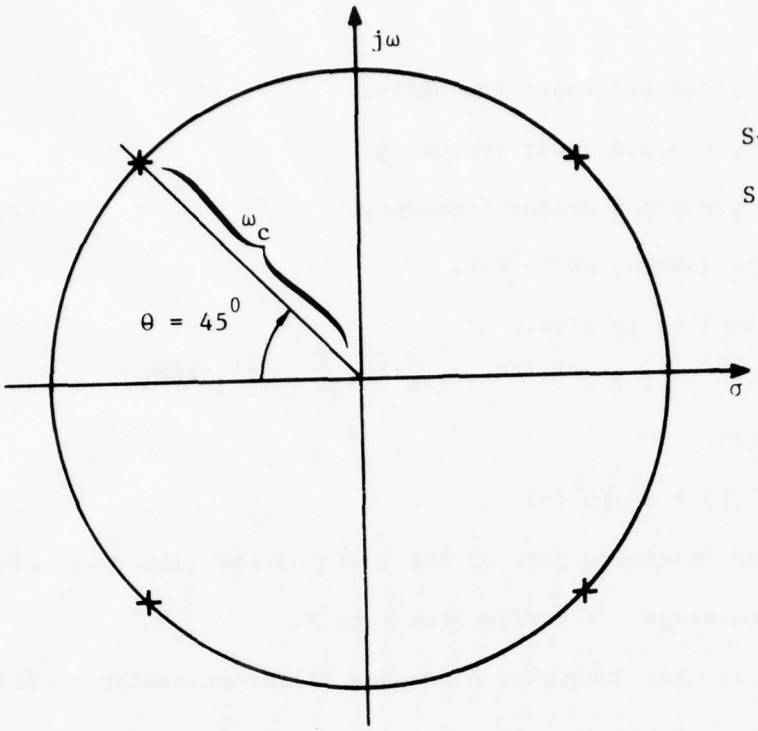


Figure 1

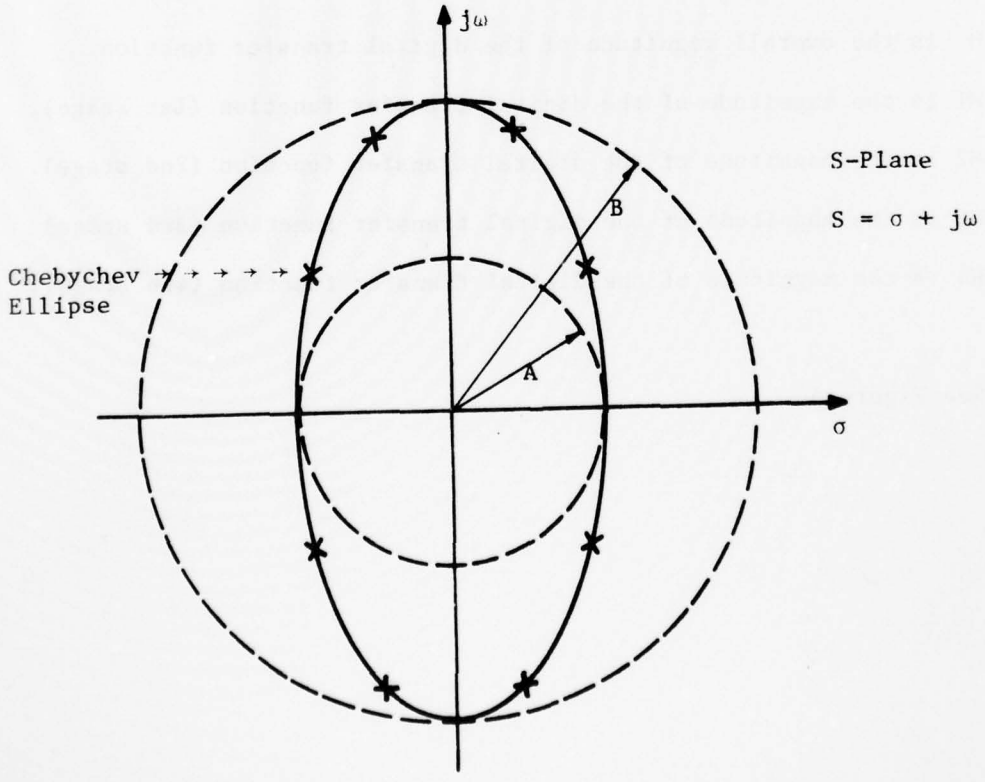


Figure 2

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

8^{th} ORDER BUTTERWORTH BAND-PASS FILTER

N = 4
Start at ω = 0 radian
T = 0.002
Steps of 0.1 radian
ω_u = 60 radians
ω_l = 40 radians

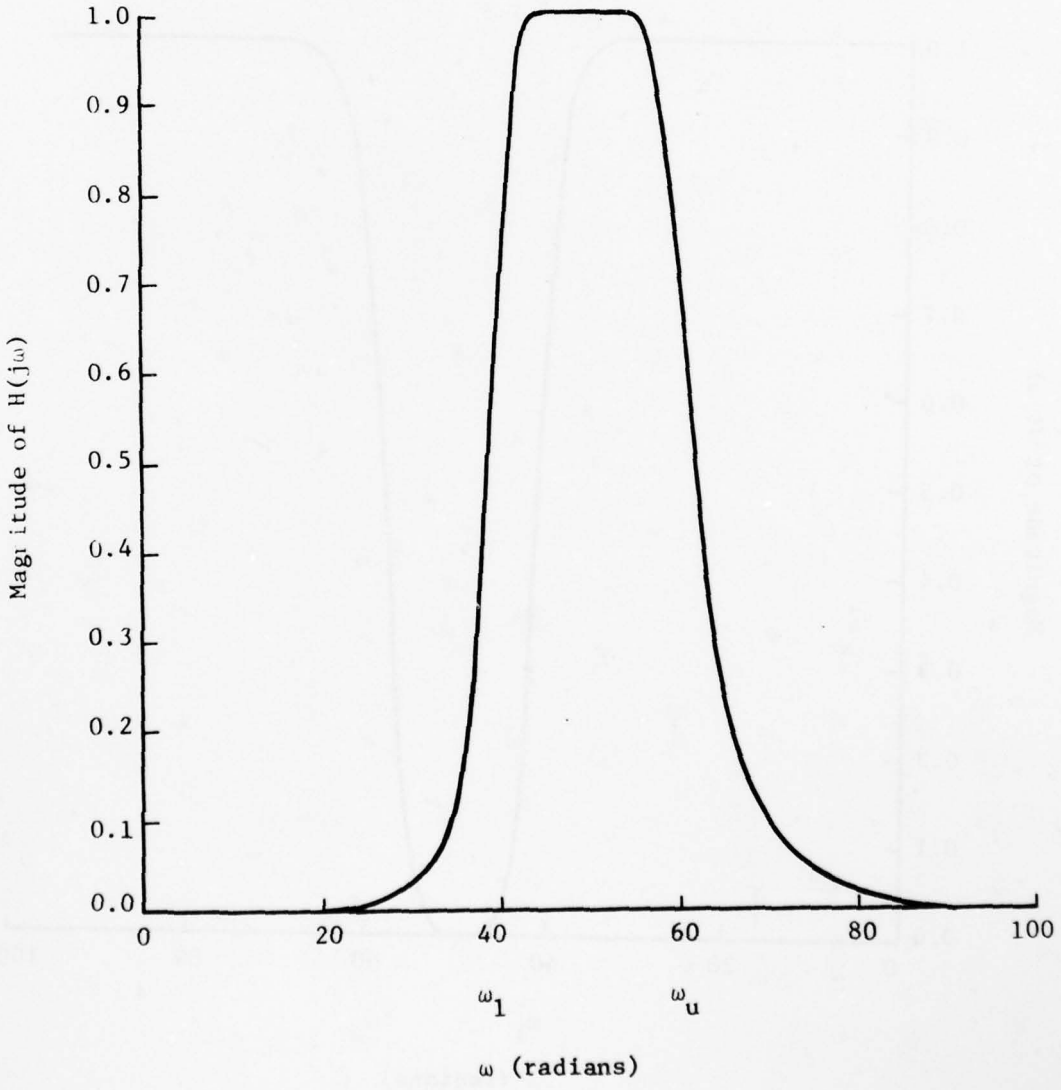


Figure 3

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

8^{th} ORDER BUTTERWORTH BAND-STOP FILTER

N = 4
Start at ω = 0 radian
T = 0.002
Steps of 0.1 radian
ω_u = 60 radians
ω_l = 40 radians

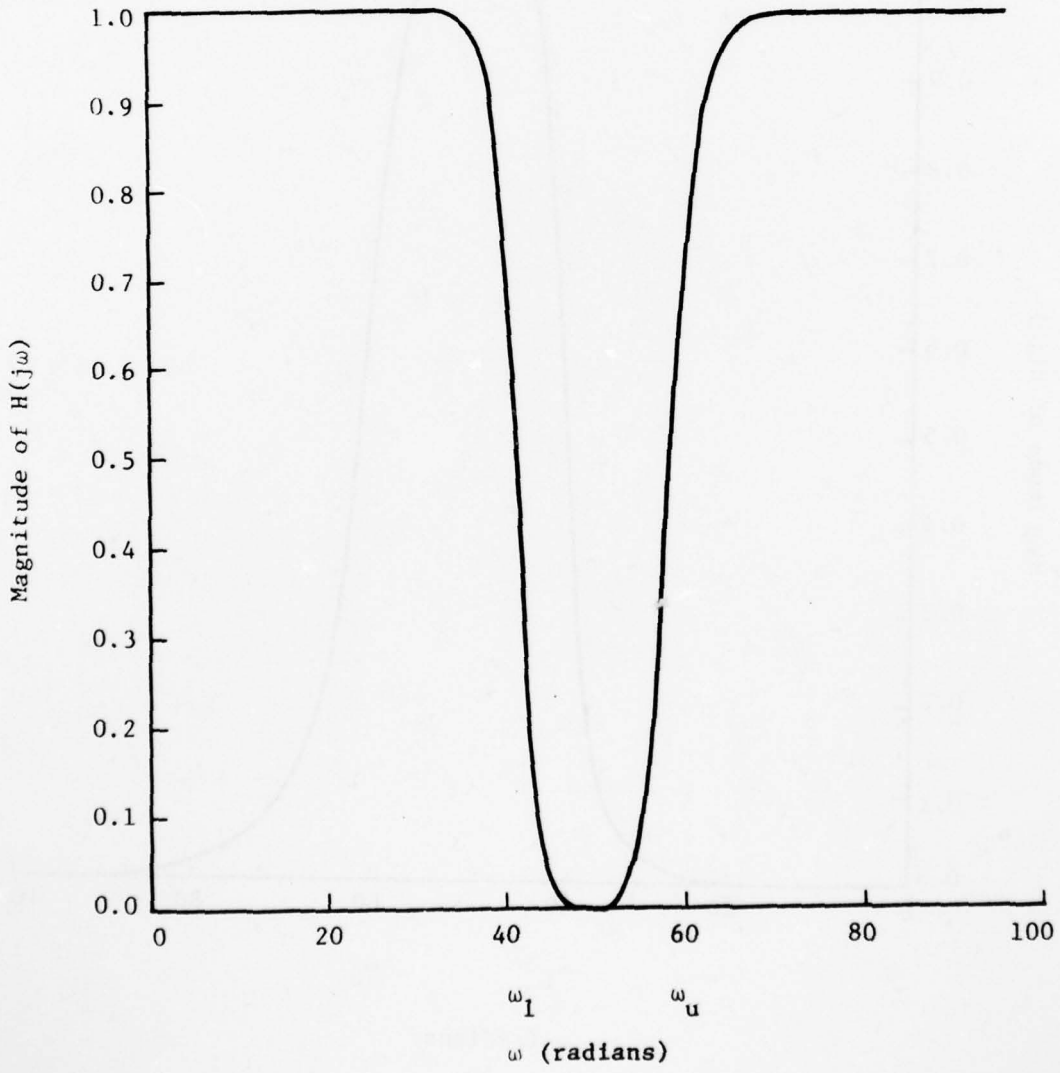


Figure 4

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

8^{th} ORDER CHEBYCHEV BAND-PASS FILTER

N = 4
Start at ω = 0 radian
Ripple = σ = 0.100
T = 0.002
Steps of 0.1 radian
ω_u = 60 radians
ω_l = 40 radians

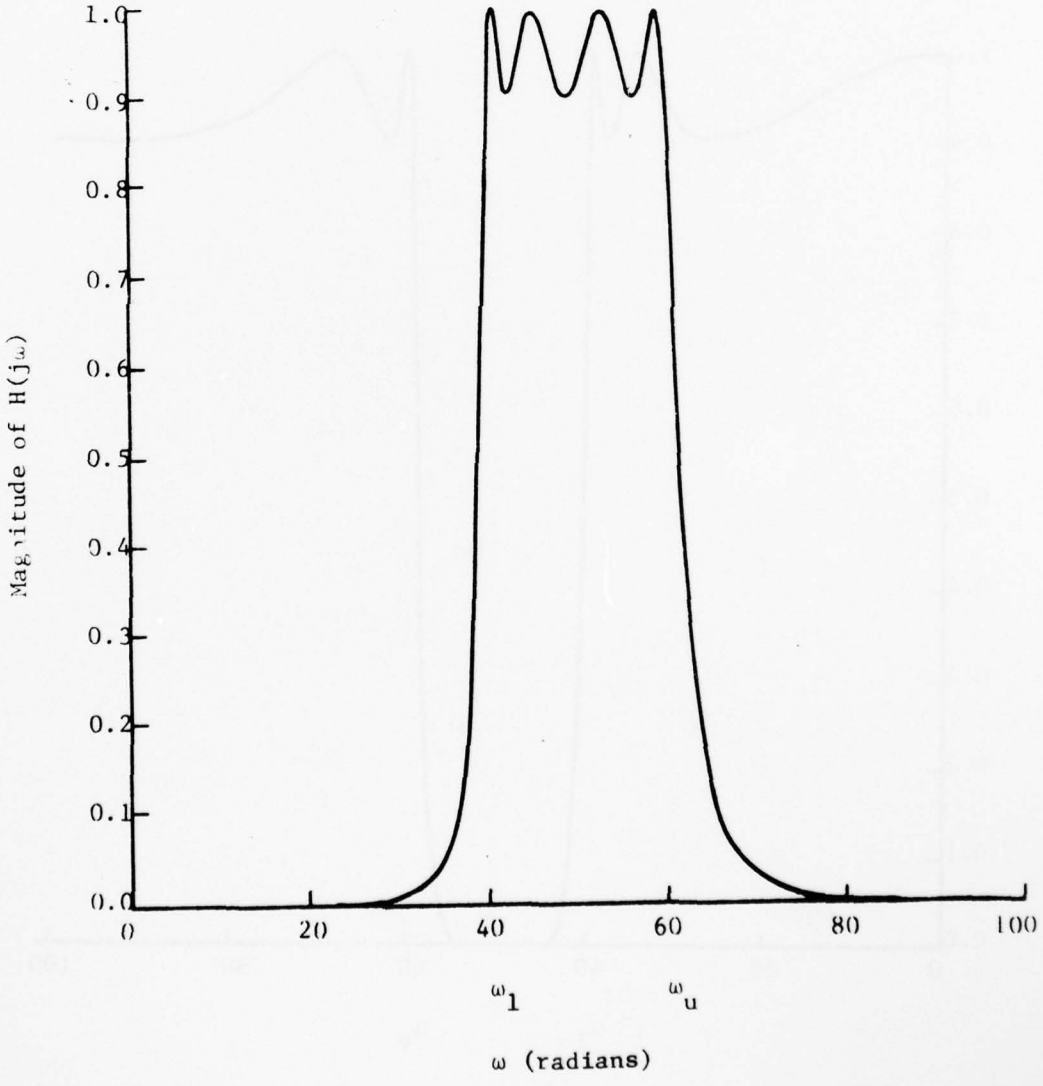


Figure 5

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

10^{th} ORDER CHEBYCHEV BAND-STOP FILTER

N = 5
Start at ω = 0 radian
Ripple = σ = 0.100
T = 0.002
Steps of 0.1 radian
ω_u = 60 radians
ω_l = 40 radians

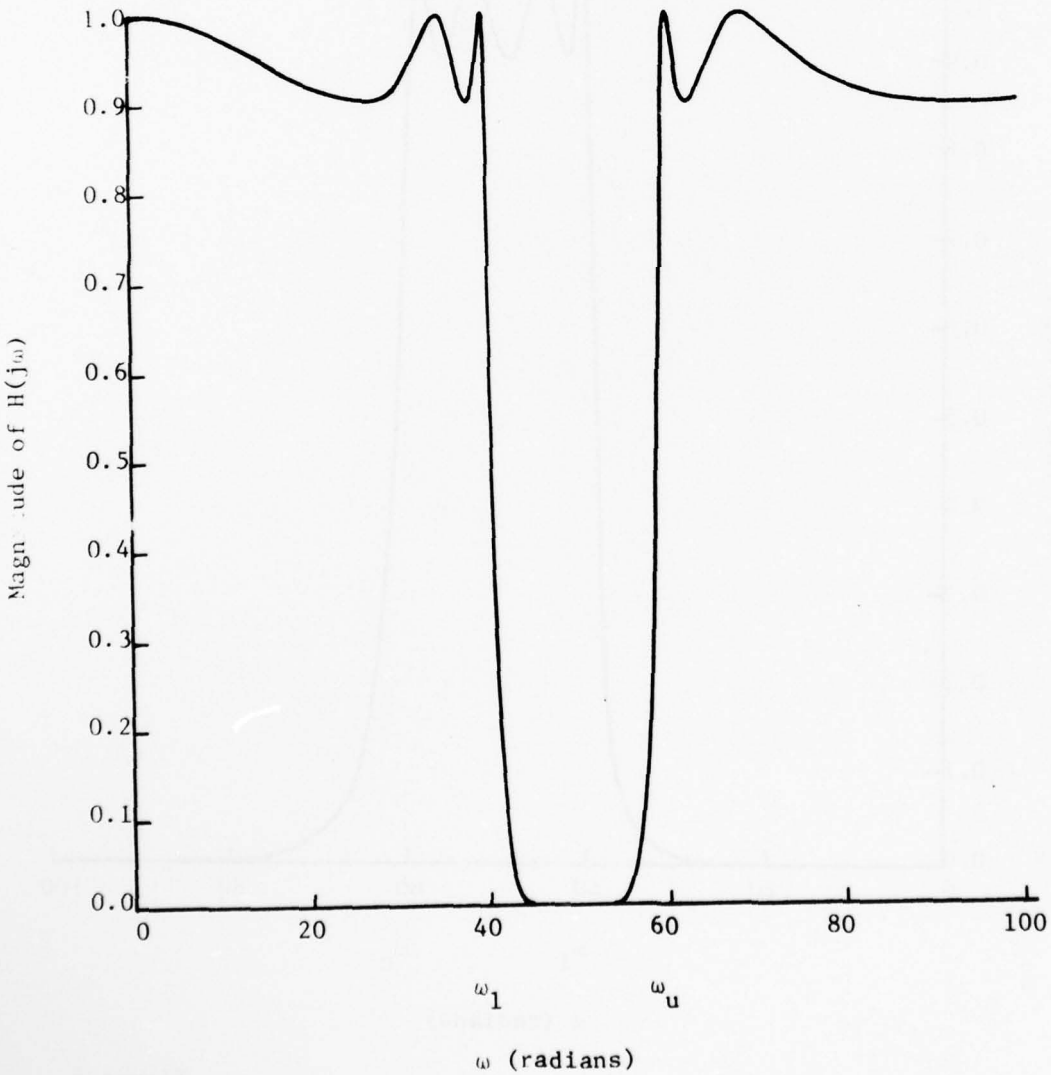


Figure 6

References

A. Budak, Passive and Active Network Analysis and Synthesis, Houghton Mifflin Co., Boston, 1974.

D. Childers and A. Durling, Digital Filtering and Signal Processing, West Publishing Company, New York, 1975.

J. J. D'Azzo and C. H. Houpis, Linear Control System Analysis and Design, McGraw-Hill, Inc., New York, 1975.

B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill, Inc., New York, 1969.

B. J. Leon and P. A. Wintz, Basic Linear Networks for Electrical and Electronics Engineers, Holt, Rinehart, and Winston, Inc., New York, 1970.

L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.

M. E. Van Valkenburg, Modern Network Synthesis, John Wiley & Sons, New York, 1960.

L. Weinberg, Network Analysis and Synthesis, McGraw-Hill, Inc., New York, 1962.

IBM S/360, Scientific Subroutine Package Version 3, Publication 6H 20-0205-4, pp. 181,182.

Appendix F

Fault Detection in Digital Filter Systems

by

Rex W. Tracy, Student Member IEEE

and

Thomas A. Brubaker, Senior Member IEEE
Department of Electrical Engineering
Colorado State University
Fort Collins, Colorado 80523

November 2, 1975

Abstract

The use of a parameter identification procedure to detect faults in hardware used to implement a broad class of linear algorithms defined as digital filters is presented. Using the filter coefficient estimates produced by the identifier a method of measuring the acceptability of the filtering algorithm is suggested and a numerical example is given.

INTRODUCTION

Today's integrated circuit technology has provided inexpensive digital hardware that can be used for the direct implementation of the digital signal processing algorithms utilized in a variety of communication and control systems. In these applications, there is a need for new methods of failure analysis. While it is desirable to know when a hardware failure occurs, and to know where the failure is located, it is also important to know what effect the failure has on algorithm performance. In many cases a failure, such as losing the least significant bit in a register, will not significantly degrade performance and there is no need to consider switching to a redundant implementation.

Currently, most of the work on failure analysis is described under the heading of fault detection. From an operational point of view, designers are concerned with the development of fault tolerant computer systems. However, in both cases the hardware is usually considered and little attention is given to the interaction between the hardware and the algorithm to be implemented.

To provide more information about algorithm performance, the problem of mathematically describing failure detection and diagnosis has recently been given attention. Mehra and Peschon [1] suggested implementing a Kalman filter in parallel with an algorithm implementation and using the statistics of the innovations process for detecting system failures. Davis [2] shows a method for using a Kalman filter to estimate the time when a failure occurs. He then recommends readjustment of the filter parameters to obtain new state estimates after the occurence of the fault. Krischer [3], in an applications oriented approach, applied a parameter identification approach to estimate the state of a biological system as the system parameters slowly vary.

In this paper, a parameter identification procedure similar to that presented by Mendel and Fu [4] is used to detect faults in the implementation of a broad class of algorithms defined as digital filters. Given a design, the filter coefficients are used as initial conditions for the identifier operating in parallel with the filter. The identifier output consists of an estimate of the filter coefficients and an error signal. When a fault occurs, the error signal and the coefficient estimates will change rapidly during the next few sampling times. If the fault is very serious, such as complete failure of the multiplier, the algorithm implementation no longer exists and a total failure has occured. If, however, the failure is not total so that the effect is to modify the filter coefficients, the coefficient estimates will converge to the new values. At this point, the effect of the coefficient changes must be evaluated to determine if the algorithm characteristics are still satisfactory. In this paper this is done by establishing bounds on the steady-state magnitude and phase functions. If the change in the coefficients allows these bounds to be satisfied, the filter operation is considered to be acceptable.

The procedures described have several advantages. First, remote sensing on an algorithm implementation can be accomplished. Secondly, the identifier can be used with any linear system that can be modeled by a difference equation. Thirdly, a better assessment as to the need for switching in an alternate algorithm implementation is available.

THE DIGITAL FILTER MODEL

An nth order time invariant linear digital filter is represented by the expression

$$y[nT]=\sum_{k=0}^{m} a_k \times [(n-K)T] - \sum_{j=1}^{n} b_j y[(n-j)T] \tag{1}$$

Where the coeffieient sequences $\{a_k\}$ and $\{b_j\}$ are chosen to achieve the desired filter characteristics. Using a vector representation (1) can be rewritten as

$$y[nT] = \underline{a}^t \underline{x}[nT] - \underline{b}^t \underline{y}\,[(n-1)T] \tag{2}$$

where

$$\underline{a}^t = [a_0\ a_1\ \cdot\ \cdot\ a_m], \tag{3}$$

$$\underline{b}^t = [b_1\ b_2\ \cdot\ \cdot\ b_n], \tag{4}$$

$$\underline{x}[nT] = \begin{bmatrix} x[nT] \\ x[(n-1)T] \\ \vdots \\ x[(n-m)T] \end{bmatrix}, \tag{5}$$

and

$$\underline{y}[(n-1)T] = \begin{bmatrix} y[(n-1)T] \\ y[(n-2)T] \\ \vdots \\ y[0T] \end{bmatrix}. \tag{6}$$

Letting

$$\underline{c} = \begin{bmatrix} \underline{a} \\ \hline \underline{b} \end{bmatrix}, \tag{7}$$

$$z[nT] = y[nT] \tag{8}$$

and

$$\underline{u}[nT] = \begin{bmatrix} \underline{x}[nT] \\ --- \\ \underline{y}[(n-1)T] \end{bmatrix} \tag{9}$$

(2) can be rewritten as

$$z[nT] = \underline{u}^t[nT]\underline{c} = \langle \underline{u}[nT], \underline{c} \rangle \tag{10}$$

Equation (10) now represents the model for the digital filter that is used in the identification algorithm.

THE IDENTIFICATION ALGORITHM

The identification procedure is a sequential gradient descent algorithm utilizing a quadratic cost function. The model is depicted in the block diagram of Fig. 1. Here, at time t=nT the actual digital filter output is y[nT] and the current input is $\underline{u}[nT]$. From (10) the filter model is now expressed as

$$z[nT]=\underline{u}^t[nT]\hat{\underline{c}}[nT] \tag{11}$$

Where $\hat{\underline{c}}[nT]$ is the approximation of the parameter vector $\underline{c}$ at time t=nT. Defining the error as

$$e[nT]=y[nT]-z[nT] \tag{12}$$

the quadradic error function is

$$J[\hat{\underline{c}}[nT]]=\tfrac{1}{2}[y[nT]-z[nT]]^2 \tag{13}$$

To minimize $J[\hat{\underline{c}}[nT]]$ a multidimensional form of Newton's method is used giving the recursive expression

$$\hat{\underline{c}}[(n+1)T]=\hat{\underline{c}}[nT] - R[nT] \text{ grad } [J[\hat{\underline{c}}[nT]]]. \tag{14}$$

Where R[nT] is an nxn matrix whose coefficients are yet to be specified. From (13) the expression

$$\text{grad } J[\hat{\underline{c}}[nT]]=-e[nT]\underline{u}[nT] \tag{15}$$

is obtained. Substitution of (15) into (14) now yields the recursive

expression

$$\hat{\underline{c}}[(n+1)T]=\hat{\underline{c}}[nT]+R[nT]e[nT]\underline{u}[nT] \tag{16}$$

Mendel and Fu [7], show that for the choice

$$R[nT]=\frac{I}{\underline{u}^t[nT]\underline{u}[nT]} \tag{17}$$

convergence in the mean for $\hat{\underline{c}}[nT]$ as $n\rightarrow\infty$ is obtained.

The addition of the measurement noise $v[nT]$ and $\underline{n}[nT]$ as shown in Fig. 1 causes the parameter estimates using (16) and (17) to become biased. To avoid this a new error function

$$e[nT]=\tfrac{1}{2}[y[nT]-[z[nT]+v[nT]]]^2 \tag{18}$$

is used to give

$$\hat{\underline{c}}[(n+1)T]=[I+R[nT]\hat{\Phi}[nT]]\hat{\underline{c}}[nT]+R[nT]e[nT][\underline{u}[nT]+\underline{n}[nT]] \tag{19}$$

where $\hat{\Phi}[nT]$ is the nth estimate of the measurement noise covariance.

FILTER PERFORMANCE

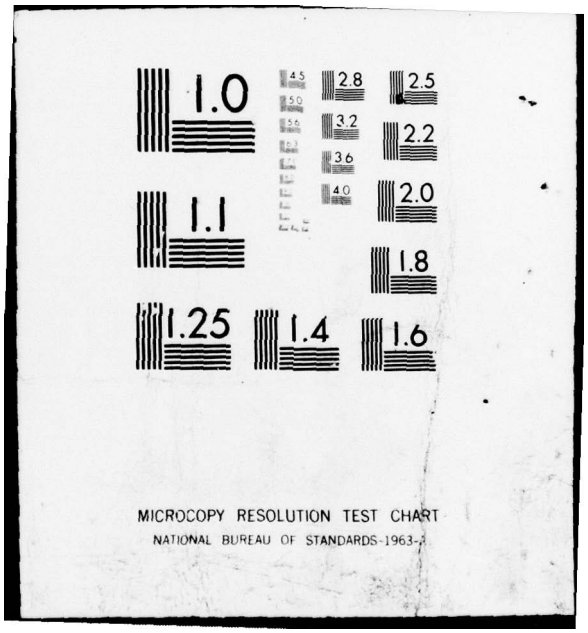
Given a digital filter described by (1), the implementation of the filter is usually done to minimize the effects of coefficient and rounding errors. In practice this results in most filters being implemented as a cascade or parallel connection of first and second order sections [5].

Given a cascade or parallel connection, several options for utilizing the identifier are possible. First, the identifier can treat the complete structure as a single filter. However, for high order filters, the determination of the coefficient sensitivities with respect to any error criterion is generally very difficult. Secondly, the identifier can operate on each first or second order section. This allows a performance evaluation at the section level which can be interpreted in terms of overall performance. Because magnitude and phase are often used to specify a digital filter design, a change in the filter coefficients, due to a fault, will be

AD-A059 868 COLORADO STATE UNIV FORT COLLINS F/G 9/1
DEVELOPMENT OF IMPROVED DESIGN METHODS FOR DIGITAL FILTERING SY--ETC(U)
NOV 77 T A BRUBAKER F33615-75-C-1138
UNCLASSIFIED AFAL-TR-77-207 NL

3 OF 3
AD A059868

END
DATE FILMED
12-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

evaluated to see if the specifications are still satisfied. This follows the coefficient design procedure described by Brubaker [6].

For a second order filter with the transfer function

$$H(Z) = \frac{a_0+a_1Z^{-1}+a_2Z^{-2}}{1+b_1Z^{-1}+b_2Z^{-2}} \tag{20}$$

the magnitude function is described by

$$|H[j\omega]| = \frac{A(\omega)}{B(\omega)} \tag{21}$$

where

$$A(\omega) = \left|a_0^2+a_1^2+a_2^2+(2a_0a_1+2a_1a_2)\cos\omega T + 2a_0a_2\cos 2\omega T\right|^{\frac{1}{2}} \tag{22}$$

and

$$B(\omega) = \left|1 + b_1^2 + b_2^2 + 2b_1(1 + b_2)\cos\omega T + 2b_2\cos 2\omega T\right|^{\frac{1}{2}} \tag{23}$$

Using (21) a differential magnitude approximation can be written as

$$\Delta|H[j\omega]| \cong \frac{\partial|H(j\omega)|}{\partial a_0}\Delta a_0 + \frac{\partial|H(j\omega)|}{\partial a_1}\Delta a_1 + \frac{\partial|H(j\omega)|}{\partial a_2}\Delta a_2 +$$

$$\frac{\partial|H(j\omega)|}{\partial b_1}\Delta b_1 + \frac{\partial|H(j\omega)|}{\partial b_2}\Delta b_2 \tag{24}$$

The partial derivatives in (24) are given in [6] and by evaluating these derivatives over a frequency range of interest a region in parameter space can be established where filter performance is satisfactory for a given $\Delta|H[j\omega]|$. A similar strategy can be implemented for the phase function.

expression

$$\hat{\underline{c}}[(n+1)T]=\hat{\underline{c}}[nT]+R[nT]e[nT]\underline{u}[nT] \tag{16}$$

Mendel and Fu [7], show that for the choice

$$R[nT]=\frac{I}{\underline{u}^t[nT]\underline{u}[nT]} \tag{17}$$

convergence in the mean for $\hat{\underline{c}}[nT]$ as $n\rightarrow\infty$ is obtained.

The addition of the measurement noise $v[nT]$ and $\underline{n}[nT]$ as shown in Fig. 1 causes the parameter estimates using (16) and (17) to become biased. To avoid this a new error function

$$e[nT] = \tfrac{1}{2}[y[nT] - [z[nT]+v[nT]]]^2 \tag{18}$$

is used to give

$$\hat{\underline{c}}[(n+1)T] = [I+R[nT]\hat{\Phi}[nT]]\hat{\underline{c}}[nT]+R[nT]e[nT][\underline{u}[nT]+\underline{n}[nT]] \tag{19}$$

where $\hat{\Phi}[nT]$ is the nth estimate of the measurement noise covariance.

FILTER PERFORMANCE

Given a digital filter described by (1), the implementation of the filter is usually done to minimize the effects of coefficient and rounding errors. In practice this results in most filters being implemented as a cascade or parallel connection of first and second order sections [5].

Given a cascade or parallel connection, several options for utilizing the identifier are possible. First, the identifier can treat the complete structure as a single filter. However, for high order filters, the determination of the coefficient sensitivities with respect to any error criterion is generally very difficult. Secondly, the identifier can operate on each first or second order section. This allows a performance evaluation at the section level which can be interpreted in terms of overall performance. Because magnitude and phase are often used to specify a digital filter design, a change in the filter coefficients, due to a fault, will be

EXAMPLE

To illustrate using identification for fault detection consider the first order digital filter

$$H(Z) = \frac{a_1 Z^{-1}}{1-b_1 Z^{-1}} = \frac{0.8Z^{-1}}{1-0.85Z^{-1}} \qquad (15)$$

For $\Delta|H[j\omega]| = 0.1$, the acceptable region for the filter coefficients is shown in Fig. 2. The identifier response to the filter is shown in Fig. 3 as b_1 changes from 0.85 to 0.4. Initially the identifier tracks the correct coefficients. When b_1 changes the error signal changes rapidly and converges to zero. The coefficient estimates converge to the new values and the b_1 coefficient of 0.4 does not allow satisfactory performance through use of the region shown in Fig. 2. A redundant filter would then be set into operation.

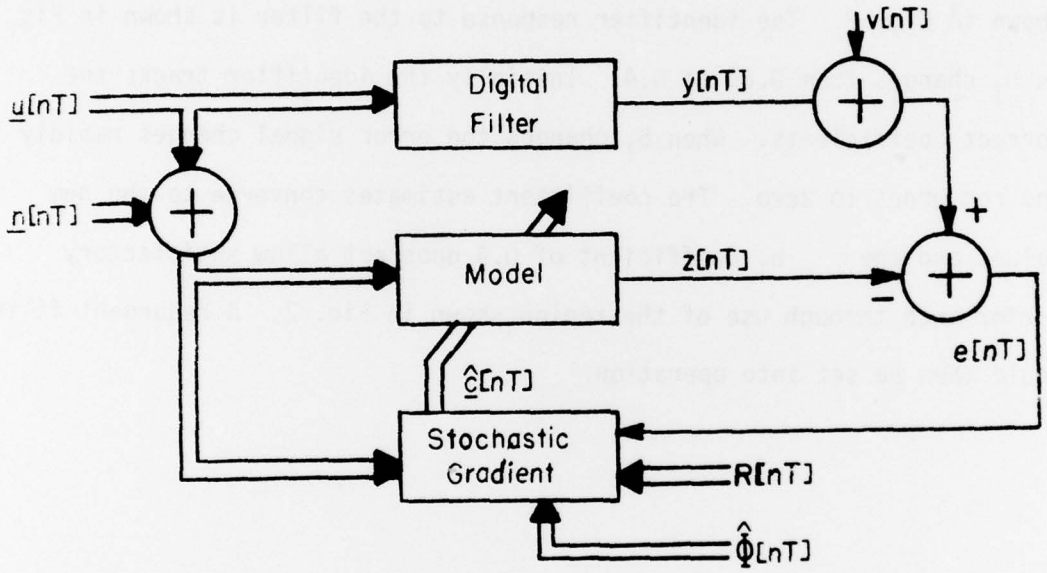


Figure 1. Block Diagram for the Gradient Identifier

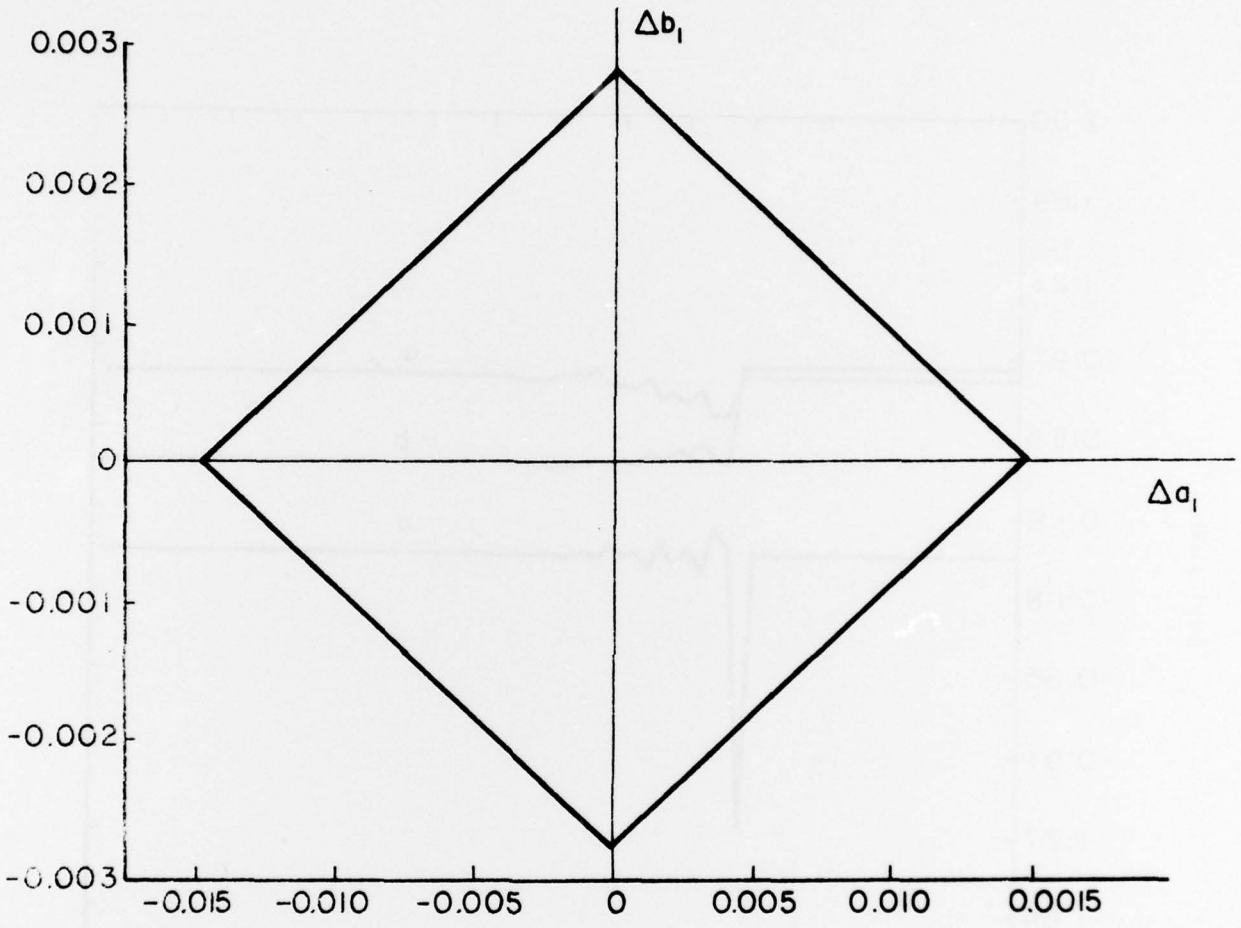


Figure 2. Coefficient Region for the Transfer Function of Equation 15

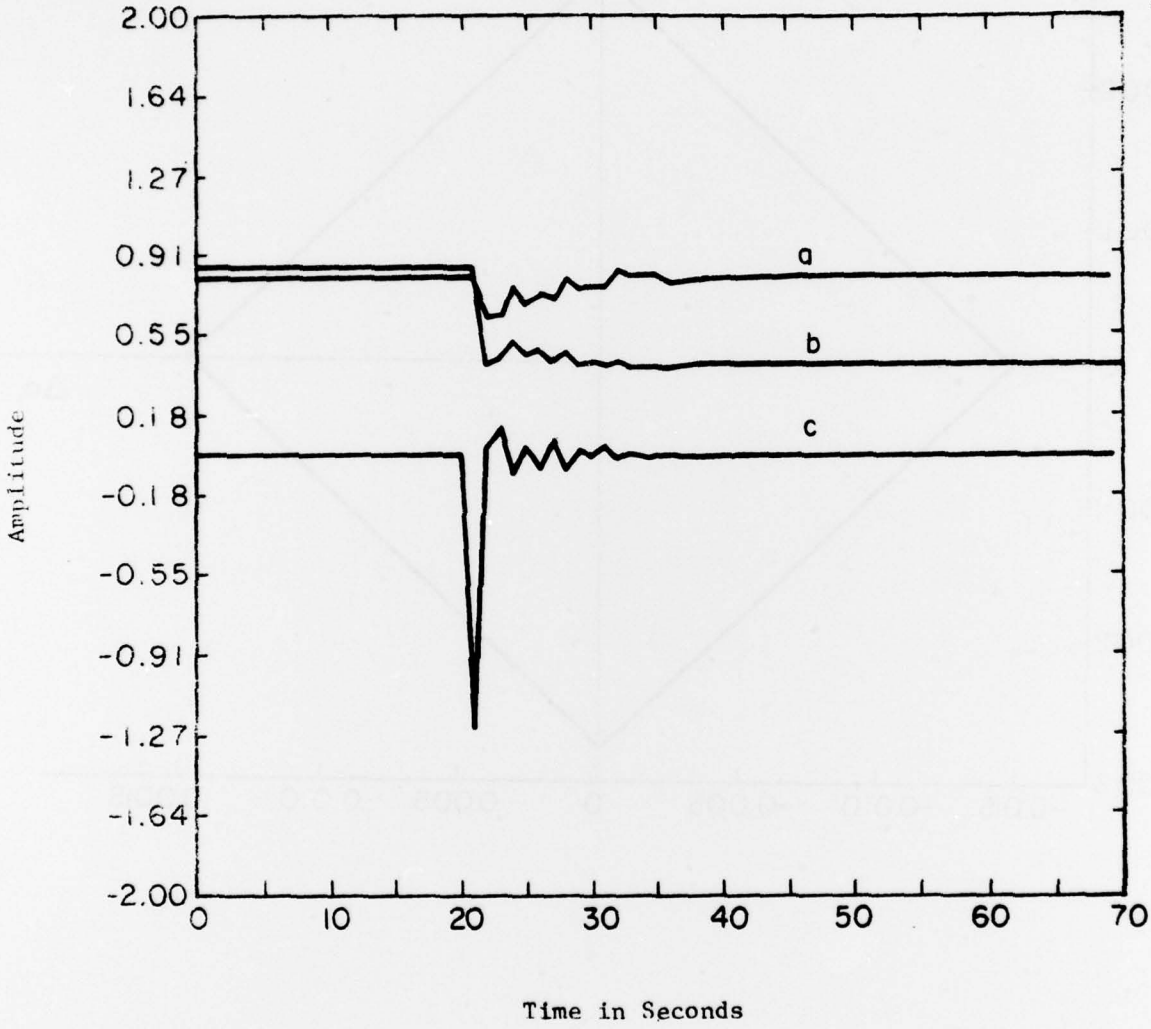


Figure 5. Identifier Response to the Transfer Function of Equation 15

REFRENCES

1. R. K. Mehra and J. Peschon, "An Innovations Approach to Fault Detection and Diagnosis in Dynamic Systems." Automatica, Vol. 7, pp 637-640, 1971.

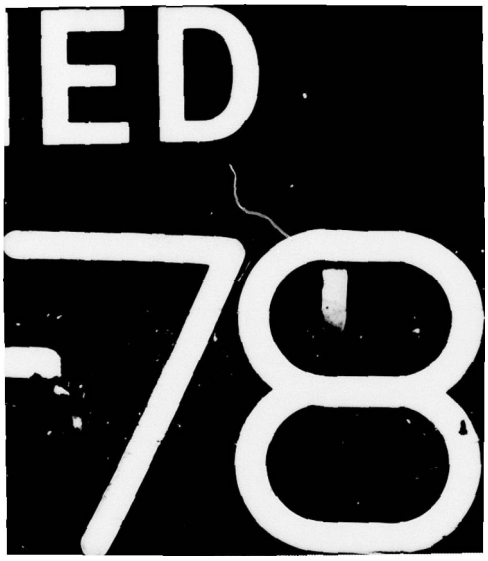
2. M. H. A. Davis, "The Application of Nonlinear Filtering to Fault Detection in Linear Systems", I.E.E.E. Trans. on Automatic Control Vol. AC-20 pp 257-258 April, 1975.

3. J. P. Krischer, "Application of Sequential Methods in Pattern Recognition to Diagnosis", Math. Biosc. Vol. 13, pp 33, 1972.

4. J. M. Mendel and K. S. Fu, Adaptive Learning and Pattern Recognition Systems: Theory and Application Academic Press, 1970.

5. B. Liu "Effect of Finite Word Length on the Accuracy of Digital Filters", I.E.E.E. Trans. on Circuit Theory November, 1971.

6. T. A. Brubaker, "A Strategy for Coefficient Quantization in Digital Control Algorithums", Computers and Electrical Engineering Vol. 1, pp 501-511, 1974.


ED
78