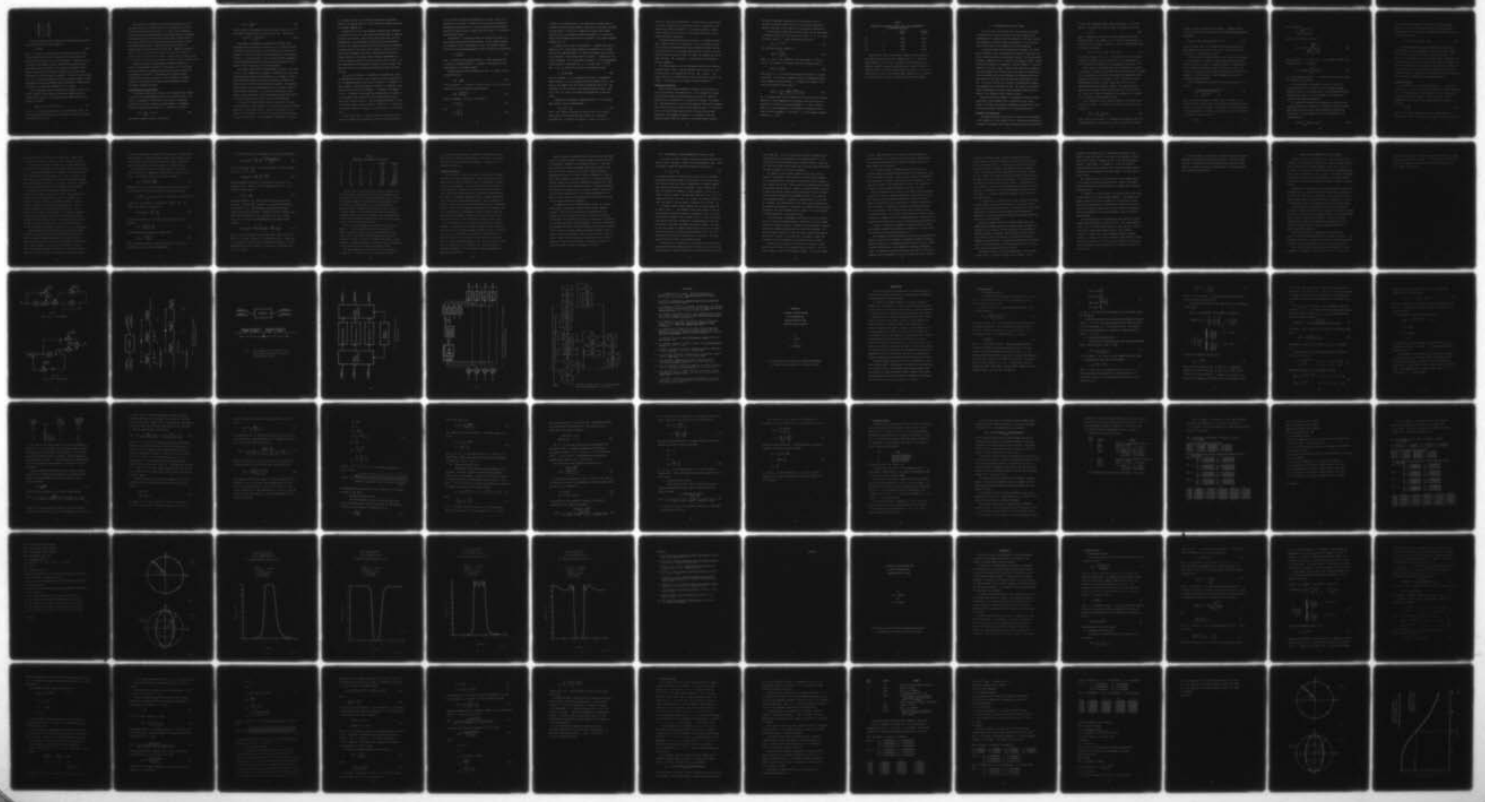
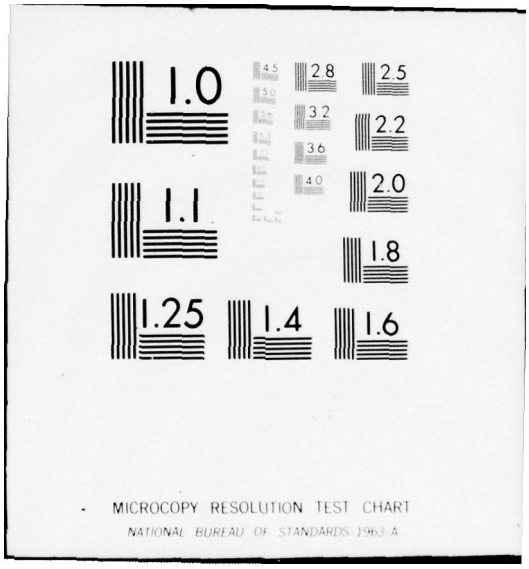
AD-A042 848 COLORADO STATE UNIV FORT COLLINS DEPT OF ELECTRICAL --ETC F/G 9/5
DIGITAL FILTER DESIGN AND IMPLEMENTATION METHODS.(U)
JUL 77 T A BRUBAKER F33615-73-C-1253

UNCLASSIFIED AFAL-TR-75-211 NL

1 OF 2
ADA042848






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

AD A042848

AFAL-TR-75-211

DIGITAL FILTER DESIGN AND IMPLEMENTATION METHODS

DDC
AUG 12 1977
C

DEPARTMENT OF ELECTRICAL ENGINEERING
COLORADO STATE UNIVERSITY
FORT COLLINS, COLORADO 80521

JULY 1977

164p.

Final TECHNICAL REPORT, APRIL 1973—JUNE 1974

Thomas A. Brubaker

Approved for public release; distribution unlimited.

2003 02

F33615-73-C-1253

62204F

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

DDC FILE COPY

406 434

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

Byron M. Allen
BYRON M. ALLEN
Supervisor

FOR THE COMMANDER

H. MARK GROVE, Acting 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)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFAL-TR-75-211	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DIGITAL FILTERING DESIGN AND IMPLEMENTATION METHODS.		5. TYPE OF REPORT & PERIOD COVERED Final Report 4 April 1973 - 30 June 1974 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Thomas A. Brubaker		8. CONTRACT OR GRANT NUMBER(s) F33615-73-C-1253
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Electrical Engineering Colorado State University Fort Collins, CO 80521		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 2003-02-04
11. CONTROLLING OFFICE NAME AND ADDRESS Wright Patterson Air Force Base Air Force Avionics Laboratory		12. REPORT DATE July 1977 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) 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)

DDC
AUG 12 1977
C

18. SUPPLEMENTARY NOTES

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

digital filters
design
implementation
multiplexing

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

This report describes methods for the design and implementation of digital filters that have applications in moder aircraft avionics and flight control systems. Design of the digital filters and the interaction between the design and implementation are emphasized. Multiplexing of digital filters is discussed.

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

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

TABLE OF CONTENTS

Preface iv

I. Design of Digital Filters 1

II. Implementation of Digital Filters 17

III. Multiplexing in the Implementation of Digital Filters . 28

IV. Conclusions and Recommendations for Future Work 34

APPENDICES:

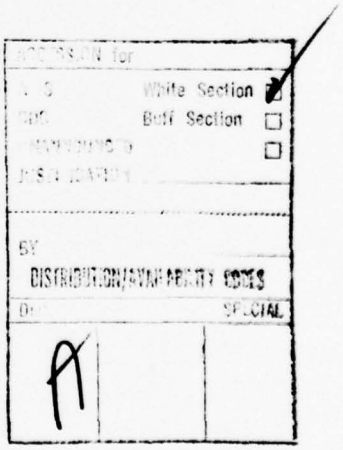
A. A Fortran IV Design Program for Butterworth and Chebýchev Band-Pass and Band-Stop Digital Filters . 43

B. A Fortran IV Design Program for Low-Pass Butterworth and Chebychev Digital Filters 72

C. Computation of the Discrete Autocovariance 99

D. Implementation of Digital Controller 113

E. A Strategy for Coefficient Quantization in Digital Control Algorithms 141



PRECEDING PAGE BLANK-NOT FILMED

PREFACE

This report describes methods for the design and implementation of digital filters that have application in modern aircraft avionics and flight control systems. The first section of the report deals with the design of digital filters and the interaction between the design and implementation tasks. The second section describes the implementation of digital filters and the procedures that are used to determine the computer word length. In the third section the use of multiplexing in the implementation of digital filters is described. The last section is concerned with conclusions and recommendations for future work that will result in less expensive more reliable digital filter structures.

During the contract period a number of technical reports on the above topics were written and sent to the contract monitor at WPAFB. These reports are included as appendices. Many of the results described in the reports are briefly mentioned in the main body of this final report and the reader is referred to the appropriate appendix for further information.

I. DESIGN OF DIGITAL FILTERS

At the present time, the majority of the published literature on the design of linear time-invariant digital filters describes design using frequency domain. The most probable explanation for this is the wealth of knowledge about analog filter design. From the frequency domain point of view, a desired magnitude function is typically specified and a discrete transfer function is found whose steady state magnitude function satisfies the specifications. However it is also possible and in many cases desirable to consider time domain synthesis. In this section, the design of digital filters using the frequency and time domains is discussed.

In general digital filters are separated into two classes, nonrecursive or finite memory and recursive or infinite memory filters. A general real time linear time invariant nonrecursive digital filter is represented by the difference equation

$$Y[nT] = \sum_{k=1}^{M} a_k \, X[(n - k)T] \tag{1}$$

where $Y[nT]$ is the filter response at $t = nT$, the sequence $\{a_k\}$ is the sequence of filter coefficients and the $X[(n - k)T]$ are input data points. In (1) T represents the sampling interval which is assumed to be constant and M is the number of input data points forming the filter window.

The general expression for a real time linear time invariant recursive digital filter of order N is represented by the difference equation

$$Y[nT] = \sum_{k=1}^{M} a_k \, X[(n - k)T] - \sum_{j=1}^{N} b_j \, Y[(n - j)T]. \tag{2}$$

Here past outputs are also used to form the response at time $t = nT$ giving rise to the recursive nature of the filter. The design task for either filter is to determine the filter coefficients so that the filter satisfies specified requirements.

Design Using the Time Domain

Synthesis in the time domain is based on a signal model whose output is typically corrupted by noise. For nonrecursive filters the most common signal model consists of a polynomial over the finite window which implies that over M data points the signal is modeled by a polynomial. Because this type of digital filter is useful in a variety of applications such as radar signal processing, the general design procedure will be outlined. First the concept of a transition matrix for a polynomial signal model is developed. This will be done only for a third order polynomial, however, the results are easily generalized. Let a signal $X(t)$ be represented by a third order polynomial which is equivalent to the differential equation

$$\frac{d^4X(t)}{dt^4} = 0 \tag{3}$$

If a uniform sampling interval is assumed, the signal and the first three derivatives at the point $t = (n + h)T$ can be written as Taylor series expansions about the point $t = nT$ giving

$$X[(n + h)T] = X[nT] + hT\dot{X}[nT] + \frac{(hT)^2}{2!}\ddot{X}[nT] + \frac{(hT)^3}{3!}\dddot{X}[nT], \tag{4}$$

$$\dot{X}[(n + h)T] = \dot{X}[nT] + hT\ddot{X}[nT] + \frac{(hT)^2}{2!}\dddot{X}[nT], \tag{5}$$

$$\ddot{X}[(n + h)T] = \ddot{X}[nT] + hT\dddot{X}[nT] \tag{6}$$

and

$$\dddot{X}[(n + h)T] = \dddot{X}[nT]. \tag{7}$$

If the state vector is defined as

$$\underline{X}[(n + h)T] = \begin{pmatrix} X[(n + h)T] \\ \dot{X}[(n + h)T] \\ \ddot{X}[(n + h)T] \\ \dddot{X}[(n + h)T] \end{pmatrix} \tag{8}$$

then in vector form

$$\underline{X}[(n + h)T] = \Phi[hT]\ \underline{X}[nT] \tag{9}$$

where $\Phi[nT]$ is the state transition matrix

$$\Phi[hT] = \begin{bmatrix} 1 & hT & \frac{(hT)^2}{2!} & \frac{(hT)^3}{3!} \\ 0 & 1 & hT & \frac{(hT)^2}{2!} \\ 0 & 0 & 1 & hT \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

To eliminate the sampling interval T in (10) the state vector is now redefined as

$$\underline{X}[(n + h)T] = \begin{pmatrix} X[(n + h)T] \\ T\,\dot{X}[(n + h)T] \\ \frac{T^2}{2!}\,\ddot{X}[(n + h)T] \\ \frac{T^3}{3!}\,\dddot{X}[(n + h)T] \end{pmatrix} . \tag{11}$$

Using the new definition the vector form for the model is

$$\underline{X}[(n + h)T] = \Phi[h]\ \underline{X}[nT] \tag{12}$$

where the new state transition matrix is given by

$$\Phi[h] = \begin{bmatrix} 1 & h & h^2 & h^3 \\ 0 & 1 & 2h & 3h^2 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

The important characteristic of $\Phi[h]$ is that

$$\Phi[-h] = [\Phi[h]]^{-1} \ . \tag{14}$$

The determination of the optimal nonrecursive filter is now formulated. Given that the signal is corrupted by noise, the observation vector at time $t = nT$ is defined by

$$\underline{Y}[nT] = M\ \underline{X}[nT] + \underline{N}[nT] \tag{15}$$

where $\underline{Y}[nT]$ is an m element observation vector, M is a mxr matrix that designates which elements of the state vector are observed, $\underline{X}[nT]$ is an r element state vector and $\underline{N}[nT]$ is an m element noise vector with zero mean. Given ℓ observations, the total observation vector $\underline{P}[nT]$ is now defined as a $(m)(\ell)$ element vector

$$\underline{P}[nT] = \begin{pmatrix} \underline{Y}[nT] \\ - - - - - - \\ \underline{Y}[(n-1)T] \\ - - - - - - \\ \vdots \\ \underline{Y}[(n-\ell-1)T] \end{pmatrix} \ . \tag{16}$$

The design problem is to determine the weight matrix W that makes the estimate

$$\underline{Z}[nT] = W\ \underline{P}[nT] \ . \tag{17}$$

optimal.

In order to write $\underline{P}[nT]$ in terms of the state vector (15) is substituted into (16) giving

$$\underline{P}[nT] = \begin{pmatrix} M\ X[nT] \\ - - - - - - - \\ M\ X[(n-1)T] \\ - - - - - - - \\ \cdot \\ \cdot \\ M\ \underline{X}[(n-\ell-1)T] \end{pmatrix} + \begin{pmatrix} N[nT] \\ - - - - - - - \\ N[(n-1)T] \\ - - - - - - - \\ \cdot \\ \cdot \\ N[(n-\ell-1)T] \end{pmatrix} \tag{18}$$

Substitution of (12) into (18) now allows $\underline{P}[nT]$ to be written as

$$\underline{P}[nT] = T\ \underline{X}[nT] + \underline{u}[nT] \tag{19}$$

where $u[nT]$ is the total noise vector on the right side of (18). In (18) the matrix T is given by

$$T = \begin{pmatrix} M \\ - - - - - - \\ M\ \Phi[h] \\ - - - - - - - \\ \cdot \\ \cdot \\ M\ \Phi[h-\ell-1] \end{pmatrix} . \tag{20}$$

and as will be shown later the T matrix plays an essential role in the filter design.

Finally the optimal estimate is assumed to be an unbiased estimate which results in a constraint relationship between W and T. To derive this (19) is substituted into (17) giving

$$\underline{Z}[nT] = WT\ \underline{X}[nT] + W\ \underline{u}[nT] \tag{21}$$

Since $\underline{Z}[nT]$ is an unbiased estimate of $\underline{X}[nT]$, taking the mean of (21) now yields the equation

$$\underline{X}[nT] = WT\ \underline{X}[nT] \tag{22}$$

which is only satisfied if the constraint

$$WT = I \tag{23}$$

is valid.

For a conventional least squares estimate the minimization problem is handled by minimizing the cost function

$$E[nT] = [T\ \underline{Z}[nT] - \underline{P}[nT]]^t[T\ \underline{Z}[nT] - \underline{P}[nT]] \tag{24}$$

Note that if the total noise vector is zero, the cost function is zero and any weight matrix W that satisfies (23) will result in an optimal filter. Since $\underline{Z}[nT]$ is the unknown, the differentiation of (24) with respect to $\underline{Z}[nT]$ and the setting of the result to zero gives

$$T^tT\ \underline{Z}[nT] - T^t\ \underline{P}[nT] = 0. \tag{25}$$

The solution of (25) gives an optimal estimate denoted by $\hat{\underline{X}}[nT]$ which is

$$\hat{\underline{X}}[nT] = [T^tT]^{-1}\ T^t\ \underline{P}[nT] \tag{26}$$

so that the optimal weight vector is

$$\hat{W} = [T^tT]^{-1}T^t \tag{27}$$

In practice, the T matrix can often be written by inspection. For example since usually only the data and not the derivatives are known, the observation is a scalar. For this important case, the T matrix for a three point window and a second order polynomial signal is given by

$$T = \begin{bmatrix} 1 & 0 \\ 1 & -1 \\ 1 & -2 \end{bmatrix} \tag{28}$$

To derive this the state vector for the second order polynomial is

$$\underline{X}[nT] = \begin{pmatrix} X[nT] \\ \dot{} \\ TX[nT] \end{pmatrix} \tag{29}$$

Since only the data $X[nT]$ is available M is defined as the matrix $[1\ 0]$. The T matrix is now given as

$$T = \begin{bmatrix} 1 & 0 & & \\ - & - & - & - \\ [1 & 0] & 1 & -1 \\ & & 0 & 1 \\ - & - & - & - \\ [1 & 0] & 1 & -2 \\ & & 0 & 1 \end{bmatrix} \tag{30}$$

which when multiplied out gives (28). The optimal weight matrix is now given by

$$\hat{W} = \frac{1}{6} \begin{bmatrix} 5 & 2 & -1 \\ 3 & 0 & -3 \end{bmatrix} \tag{31}$$

Using (31) the two elements of the optimal estimate vector are estimates of the data and the first derivative and are given by

$$\hat{X}_1[nT] = \frac{5}{6} Y[nT] + \frac{2}{6} Y[(n-1)T] - \frac{1}{6} Y[(n-2)T] \tag{32}$$

and

$$\hat{X}_2[nT] = \frac{3}{6T} Y[nT] + 0\ Y[(n-1)T] - \frac{3}{6T} Y[(n-2)T]. \tag{33}$$

For this formulation, the T matrix can be easily generalized. The number of rows in the T matrix is the size of the data window. The first column consists of ones, and the elements in all other columns are given by

$$T_{ij} = (-1)^j\ (i)^j \qquad j,i \geq 0 \tag{34}$$

Thus for a five point filter the T matrix for a third order polynomial fit has five rows and three columns giving

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 1 \\ 1 & -2 & 4 \\ 1 & -3 & 9 \\ 1 & -4 & 16 \end{bmatrix} \tag{35}$$

Given the optimal estimate formed using conventional least squares, the covariance matrix for the estimate is

$$C = W^t R W \tag{36}$$

where R is the covariance matrix of the total noise vector $\underline{u}[nT]$. In most applications the noise is assumed to be statistically time invariant so the elements of R are all constant. Carefully note that conventional least squares does not consider any correlation between noise samples and the estimate is not optimum if correlation exists. However in many cases the estimate is satisfactory and the optimal weight matrix is easily found via a computer. In terms of a good design tool, (27) is easily programmed and with an interactive graphics terminal the designer can quickly determine filter coefficients by simply giving the number of row and column elements of the T matrix.

If the measurement noise is correlated from sample to sample, the optimal estimate is found using the concept of minimum variance or weighted least squares. The resulting optimal weight matrix for the estimate is given by

$$\hat{W}[\text{min var}] = [T^t R^{-1} T]^{-1} T^t R^{-1} \tag{37}$$

where R is the autocovariance matrix of the measurement noise. The corresponding covariance matrix for the estimate is given by (36) using the weight matrix of (37).

The extension of the theory to time invariant recursive filters is not simple because the covariance matrix of the estimate vector is time varying until steady state is reached. As a result little information in the literature is available, however, the problem has been investigated by Brubaker and Harper [1] for first and second order recursive filters. For the case where the state equation is driven by input noise the well known Kalman filter results. This filter has been the subject of numerous papers, and no theory will be given here. However its implementation is not as well known and the report that describes a modified Kalman filter where the effects of product rounding errors and random errors in the filter coefficients are considered is being sent under a separate cover.

Because time domain synthesis methods have great potential in a variety of applications of interest to the Air Force, it is anticipated that new procedures for designing time invariant recursive filters will be developed in the future. The advantage to be gained is greater noise reduction with less computing. A good reference on time domain synthesis is by Morrison [2] who gives a good background in current time domain synthesis procedures.

Design Using the Frequency Domain

The design of nonrecursive filters via the frequency domain normally starts with a specified magnitude function for the digital filter. The task is to determine the coefficients in (1) so that the magnitude function for the filter satisfies the specifications. The magnitude function is found by first taking the Z transform of (1) giving

$$Y(Z) = \sum_{k=1}^{M} a_k Z^{-k} X(Z) \ . \qquad (38)$$

The discrete transfer function is now given by

$$H(Z) = \sum_{k=1}^{M} a_k Z^{-k} \tag{39}$$

and the steady state magnitude function is found by letting $Z = e^{j\omega T}$ and taking the absolute value of both sides of (39). This yields

$$|H(e^{j\omega T})| = |\sum_{k=1}^{M} a_k e^{-j\omega kT}| \tag{40}$$

One method of determining the coefficients utilizes linear programming and is described in references [3, 4]. Another procedure described by Farden and Scharf [5] uses a Wiener filter structure to minimize the noise while giving a good approximation to the magnitude function. It is also possible to use a truncated Fourier expansion of the periodic magnitude function since the frequency function of any discrete filter is periodic in the frequency domain.

The primary problem with the above methods for design is the large size of the data window that is required to synthesize any filter with reasonably sharp attenuation characteristics. For example to obtain sharp attenuation between the pass and stop bands in reference [5] window sizes of 128 to 256 coefficients were required. The resulting filters require too much computational time to be useful in medium to high frequency applications. As a result the nonrecursive filter appears to most suitable for time domain synthesis where fewer coefficients are utilized and the filter is used for preprocessing.

For recursive digital filters there are a variety of frequency domain procedures. First Rabiner [6] has extended the linear programming procedure for nonrecursive filters. However the procedure does not appear to be satisfactory if the filter has sharp attenuation between the pass and stop bands. Also trigonometric polynomials are employed

in a procedure similar to the design of Butterworth and Chebyshev filters in the analog domain [7]. Other methods use impulse invariance and frequency sampling [7].

In general none of the above methods has gained wide acceptance. One reason for this is the lack of phase information which is important in digital control work. However the primary reason deals with the implementation. When high-order recursive filters are implemented directly, the accuracy with which the coefficients must be represented increases rapidly as the filter order increases. The same effect is present when implementing high-order analog filters and a general design principle is that analog filters should be implemented as a series or parallel connection of first and second order sections. The same implementation strategy holds for digital filters, however, with high order filters the polynomial factoring must be done with great accuracy.

The use of the bilinear Z transform on the other hand, allows the implementation directly from the series or parallel connection of the corresponding analog filter since the bilinear Z transform of a sum or product is the sum or product of bilinear Z transforms. Note that this is not true of the classical Z transform. For example, if the Z transform of two second order analog filters in cascade is to be determined, the sections must first be multiplied to give a fourth order transfer function. Then the weighting function is found by taking the inverse Laplace transform. Finally the Z transform of the weighting function is computed. In general this is not a pleasant task.

Because the bilinear Z transform method is widely used and at present is probably most appropriate for most Air Force applications

this is the only frequency design method for recursive filters that is considered in this report. However all of the results concerning the implementation apply directly to filters designed using other methods. Note that in digital control, design using the bilinear Z transform is called Tustin's method.

The bilinear Z transform method first requires the design of an analog filter that meets the specifications. If the transfer function for the analog filter is represented as H(s) the discrete transfer function for the corresponding digital filter is found by the substitution

$$s = \frac{2}{T}\frac{Z - 1}{Z + 1} \tag{41}$$

where T represents the sampling interval. In many references (41) is called the extended bilinear Z transform. The procedure is now illustrated by a very simple example.

Consider the first order analog filter with a dc gain of one and a transfer function

$$H(s) = \frac{a}{s + a} \tag{42}$$

Using the bilinear Z transform or Tustin's method, (41) is substituted into (42) to give the discrete transfer function

$$H(Z) = \frac{a_o(Z + 1)}{Z - b_1} \tag{43}$$

where the constants a_o and b_1 are given by

$$a_o = \frac{a}{\frac{2}{T} + a} \tag{44}$$

and

$$b_1 = \frac{\frac{2}{T} - a}{\frac{2}{T} + a} \tag{45}$$

In terms of the implementation, as the sampling time becomes small, it obviously takes more decimal digits or binary bits to accurately represent the coefficients. This effect is magnified in higher order digital filters so that a very important design consideration is to choose the lowest possible sampling rate which is equivalent to the largest possible sampling interval.

It appears that the use of the bilinear Z transform that does not contain the 2/T factor might eliminate the dependence of the coefficients on the sampling interval T, however the warping effect cancels out any benefits. This warping using the bilinear Z transform of (41) is now illustrated. For a given digital frequency ω_d the corresponding analog frequency is represented by ω_a. In steady state $Z = e^{j\omega_d T}$ and $s = j\omega_a$ so substitution into (41) and manipulating gives

$$\omega_a = \frac{2}{T} \operatorname{Tan}(\omega_d \frac{T}{2}) \tag{46}$$

Thus any frequency ω_d for the digital filter has a corresponding analog frequency ω_a that is warped by the relationship of (46). This implies, for example, that the critical frequencies such as the -3db frequencies are not the same in the analog and digital filters. Note that if the 2/T factor is not used, the warping will be much more severe.

To minimize this warping the sampling interval T can be chosen small enough so that the approximation

$$\operatorname{Tan}\ \omega_d \frac{T}{2} \tilde{=} \omega_d \frac{T}{2} \tag{47}$$

holds. This, however, increases the sampling rate for the digital filter which in turn requires greater accuracy for coefficient representation. In terms of the computer a longer word length is

needed for coefficient representation. Another method to avoid warping at critical frequencies is to prewarp the critical frequencies of the analog filter before the bilinear Z transform is applied. However, other frequencies in the filter are still warped with respect to the analog and digital frequencies.

Two design programs utilizing the bilinear Z transform have been sent to WPAFB as part of the current work. The first package is used for the design of digital Butterworth and Chebyshev band-pass and band-stop filters with up to six second order sections in cascade. The user need only enter the type of filter, the order, the sampling rate and the upper and lower -3db frequencies. A description of the program is given in Appendix A.

The second program is used for designing Butterworth and Chebyshev low pass digital filters. Here the designer enters the type of filter, the filter order, the sampling rate and the -3db frequency. The output consists of the coefficients for each second order section in the cascade. A description of the program is given in Appendix B.

Sampling Rate Selection

One of the key design parameters in digital filter design that is usually overlooked is the sampling rate. In many designs the sampling rate is chosen higher than necessary which can lead to a variety of problems such as instability due to coefficient rounding. One argument for a high sampling rate is the need for analog filtering before sampling. This is used to prevent aliasing of high frequency noise. If the analog filter is to cause minimal phase shift in the signal band and good attenuation at the sampling frequency it is necessary to make the sampling frequency high with respect to the maximum signal frequency.

Although the magnitude characteristic for any digital filter is periodic in frequency, random noise can still be reduced although the amount of reduction is less than that of an equivalent analog filter.

To provide some insight into the noise problem and its relationship to sampling, consider the filter given by (42) with $a = 1$. For a noise input with variance σ^2 and an autocorrelation function

$$R_{xx}(\tau) = \sigma^2 e^{-\omega_1 \tau} \tag{48}$$

the input power spectral density is

$$G_{xx}(\omega) = \frac{\sigma^2 \, 2\omega_1}{\omega_1^2 + \omega^2} \tag{49}$$

where ω_1 is the -3db bandwidth of the noise whose dc value is $2\sigma^2/\omega_1$. The variance of the output noise for the analog filter is

$$\mathrm{Var}_a\{Y\} = \frac{\sigma^2}{\omega_1 + 1} \tag{50}$$

Note that if $\omega_1 = 1$ the filter has reduced the variance of the noise by one-half. If the noise is sampled to drive a corresponding digital filter designed using the bilinear Z transform, the variance of the noise at the digital filter output is

$$\mathrm{Var}_d\{Y_n\} = \frac{\sigma^2(1 - e^{-\omega_1 T})}{\frac{2}{T}(1 - e^{-\omega_1 T}) + (1 + e^{-\omega_1 T})} \quad . \tag{51}$$

As T approaches zero the two variances given by (50) and (51) converge. However for a given sampling rate the digital filter will reduce the variance of the noise for any noise bandwidth ω_1. This is shown in Table 1 for a noise bandwidth of ten radians per second which is ten times the filter bandwidth. In the table ω_s is the sampling frequency defined as $\omega_s = 2\pi/T$.

Table 1

EVALUATION OF EOUATIONS 50 AND 51 FOR A NOISE BANDWIDTH OF TEN RADIANS PER SECOND

T	ω_s	$\frac{Var_a(Y)}{\sigma^2}$	$\frac{Var_d(Y_n)}{\sigma^2}$
2.0	π	0.091	0.500
1.0	2π	0.091	0.333
0.5	4π	0.091	0.202
0.25	8π	0.091	0.128

Looking at the table, the difference between the noise levels of the analog and digital filters at a sampling rate of 4π times the filter bandwidth is about 6db. In general, unless a serious noise problem is present, no analog prefiltering may be needed. However if filtering is done, wide-band filtering should be tried or a linear phase filter should be used to allow digital phase compensation to be employed. This can be done via the use of predictive nonrecursive filtering.

II. IMPLEMENTATION OF DIGITAL FILTERS

In this section, the procecures that can be used by the designer for determining the required computer word length needed for the implementation of a digital filter are discussed. This topic has been well developed in reports included as appendices C, D, and E and the reader should refer to these appendices for more information.

In a digital filter error is introduced by rounding. For fixed point arithmetic, errors are caused by the rounding of input data, the filter coefficients, and the intermediate products. For floating point arithmetic, additional error is caused by the rounding of sums. Because fixed point arithmetic is inexpensive, fast and reliable it is the most suitable for aircraft systems. However, when fixed point arithmetic is used, the designer must carefully specify the amount of rounding error that is acceptable and then must determine the configuration and the word length for the filter that will allow the specifications to be met. In practice this must also be done for floating point, however, most designers do not undertake the task. As a result, the specified computer requirements are almost always excessive resulting in more expensive less reliable aircraft computer systems.

Because fixed point arithmetic is satisfactory for most aircraft applications, this arithmetic will be considered in this report. However, the results can easily be extended to floating point to find the exact word length requirements for a floating point system.

Rounding of the Input Data

Any analog-to-digital converter has an output that is represented in the computer by a finite number of bits. Since most analog inputs to a converter are bipolar, most analog-to-digital converters are specified

to have a two's complement binary output consisting of P bits and a sign bit. This means that each data point is rounded with an error

$$e_1 \leq \frac{q}{2} \tag{52}$$

where q represents the value of the least significant bit scaled into signal units. For example, if P is ten bits and the output is scaled to a binary integer, a value of q equal to 10 millivolts means the signal has a range of

$$R = (2^{10} - 1)\ 10 \times 10^{-3} = 10.23 \text{ volts} \tag{53}$$

The binary equivalent of q is 1. If the output of the A/D converter is scaled to a binary fraction, the value of q in signal units is still 10 millivolts, however, the binary equivalent is given by 2^{-10}.

Input rounding or quantization has been investigated by a number of researchers. Bennett [8] and Widrow [9] have shown that for a rounding form of quantization, the errors can be modeled as independent random variables with zero mean and variance $q^2/12$. This model has been experimentally verified for input signals that traverse several quantization levels from sample to sample. The truncation form of quantization is similar except that the mean value for each error is $q/2$. Note that in terms of probability the rounding error is assumed to be uniformly distributed about the mean.

For a random model, input rounding consists of an additive noise source at the input. The variance of the error at the filter output is given by

$$\text{Var}\{Y_n\} = \frac{q^2}{12} \sum_{k=0}^{n} H^2[kT] \tag{54}$$

where $H[kT]$ is the inverse Z transform of the transfer function for the digital filter. Note that (54) is time varying and the variance

becomes a constant in steady state by letting n approach infinity. In this case the variance is also given by the discrete Wiener-Khinchine relationship

$$\mathrm{Var}\{Y[\infty]\} = \frac{q^2}{12} \quad \frac{1}{2\pi j} \oint H[Z]\ \dot{H}[Z^{-1}]\ Z^{-1}\ dZ \tag{55}$$

Given a specified value for the variance at the filter output due to input rounding, the word-length of the filter can be determined from (54) and (55). This is done as follows.

Since the discrete transfer function is the ratio of two polynomials in Z, the inverse Z transform of the transfer function, called the weighting sequence is found via polynomial division. By squaring each term and adding, the asymptotic steady state variance is found. The steady state variance can also be found using the procedure developed in Appendix 3. This formulation is easily programmed for use with an interactive graphics terminal.

Given the summation and a specified value for the variance, q is given by

$$q \leq \left[\frac{12(\text{Specified Variance})}{\sum_{k=0}^{\infty} H^2[kT]} \right]^{1/2} . \tag{56}$$

Since the specified variance is usually given as a signal to noise ratio, the signal range is part of the specified variance and the actual number of bits can be determined. For example if a sinusoidal waveform with maximum amplitude is defined as the signal output, then the maximum signal power in an analog sense is

$$SP = \frac{R}{\sqrt{2}} . \tag{57}$$

The noise power is

$$NP = \sqrt{\frac{q^2}{12} \sum_{k=0}^{\infty} H^2[kT]}$$

and for a s/N ratio of say -20 db

$$-20 \leq 10 \log \frac{\frac{q(2^p-1)}{\sqrt{2}}}{\sqrt{\frac{q^2}{12} \sum_{k=0}^{\infty} H^2[kT]}} \tag{58}$$

where the range R is $(2^p-1)q$ and p is the number of binary bits. Solving (56) for 2^p now yields

$$2^p \geq (4.08)10^{-3} \sqrt{\sum_{k=0}^{\infty} H^2[kT]} + 1 \tag{59}$$

For a given weighting sequence, p is found to be the smallest number that satisfies the inequality.

It is also possible to determine q and/or p via use of a Gaussian distribution for the noise. Here, the designer assigns a confidence level for an acceptable error. Use of standardized tables then allows the value of p to be determined. This procedure is outlined in more detail in Appendix D.

If a statistical analysis is not satisfactory, then worst case design procedures can be initiated. A method for finding the maximum upper bound on the error due to input rounding was introduced by Bertram [10]. In this analysis, the error is represented as the convolutional sum

$$e[nT] = \sum_{k=0}^{n} H[kT]\ N\ [(n-k)T] \tag{60}$$

where the input noise samples are defined as the sequence $\{N[kT]\}$. Taking the absolute value of both sides and choosing the maximum absolute value of the error gives the maximum upper bound for the error at the filter output as

$$|e[nT]| \leq \frac{q}{2} \sum_{k=0}^{\infty} |H[kT]| = B_{max} \tag{61}$$

An alternative procedure has been developed by Slaughter [11]. Here each error source is considered as a step function with height $q/2$. The error is then given as the dc steady state output. In practice, the determination of the word length using a confidence level of 95 percent usually results in a smaller word length requirement. The method of Slaughter gives a smaller word length than that of Bertram's. In most applications, simulation shows the Bertran's method yields pessimistic results. Examples for each procedure are given in Appendix D. with the results including error due to product rounding.

Rounding of Products

In any digital filter, the multiplication of a P bit and sign number by an m bit and sign coefficient results in a product represented by $(P + m)$ bits and sign. In many systems, p and m are assumed to be equal so the product contains $2p$ bits plus sign. In most implementations, the products are rounded resulting in an error e_p bounded by

$$e_p \leq \frac{q_p}{2} \tag{62}$$

where q is the least significant bit of the rounded product. In practice this value of q_p may not be the same for various products and

not the same as the q for the input rounding error. However, for the purpose of analysis all values of q will be taken as the same.

As is the case with input rounding error, product rounding errors are assumed to be uncorrelated zero-mean random variables with variances $q^2/12$. Each product rounding error is assumed to be an additive noise source following the multiplication in the filter block diagram. The only difference in the analysis is the fact that the product rounding errors occur inside of the filter. As a result, the form of implementation plays an important factor in the implementation design phase.

Currently, most digital filters are implemented using the Direct Form 1 (DF1) or Direct Form 2 (DF2). Block diagrams for a third order filter are shown in Figs. 3 and 4 of Appendix D. In addition, it is well known that to minimize the effect of product rounding, a high order filter should be implemented as a cascade or parallel connection of first and second order sections. This is pointed out by Liu [12] and in Appendix C. For example in Appendix C. the autocovariance sequence for a fourth order Butterworth filter implemented in DF1 is shown in Fig. 2. The variance is about $48.75q^2$. Note that the variance is given when $k=0$. For two second order sections in cascade the variance is $5.83q^2$ which is a vast improvement.

Another problem that is present and closely related to product rounding is overflow. While a certain implementation structure may result in minimal product rounding, when the necessary scaling is employed to prevent overflow the structure can exhibit increased error due to product rounding. In practice the overflow often occurs during the transient response of the filter to a given input and it is difficult by analytical means to determine when overflow is present.

As a result, the design of a good filter structure that is satisfactory from both product rounding and overflow almost always requires the use of interactive design software that allows the designer to simulate various configurations over the specified set of input signals.

To demonstrate the ideas in a single example, a first order digital filter will be implemented in DF1 and DF2 as shown in Figs. 1a and 1b. The transfer function for the filter is given by

$$H(Z) = \frac{a_0 Z + a_1}{Z - b_1} = \frac{N(Z)}{D(Z)} \tag{63}$$

Referring to Fig. 1a, the three multiplication errors are operated on only by the pole so the error is given by the convolutional sum

$$e_{DF1}[nT] = \sum_{k=0}^{n} b_1^k \{\varepsilon_0[(n-k)T] + \varepsilon_1[(n-k)T] + \delta_1[(n-k)T]\} \tag{64}$$

In (62) b_1^k is the inverse Z transform at $1/P(Z)$ in (61). The steady state variance is now given by

$$\text{Var}\{e_{DF1}[\infty]\} = \frac{3q^2}{12} \frac{1}{1-b_1^2} \tag{65}$$

In terms of overflow, for a unit step input the response in the Z domain is

$$y(Z) = \frac{a_0 Z + a_1}{(Z-b_1)} \frac{Z}{Z-1} . \tag{66}$$

The corresponding time response in steady state is

$$(y_n)_{n\to\infty} = \frac{a_0 + a_1}{1-b_1} \tag{67}$$

Thus for unity gain (65) is set equal to one and for this case no overflow will occur in the DF1 implementation.

For the Direct Form 2, the variance of the error in steady state is

$$\mathrm{Var}\{e_{DF2}[nT]\} = \frac{2q^2}{12} + \frac{q^2}{12}\ \frac{(a_0^2+2a_0a_1b_1+a_1^2)}{1-b_1^2}\ . \tag{68}$$

For a dc gain of one $a_0+a_1 = 1-b_1$ and substitution of this into (66) for $a_0 = a_1 = \frac{1-b_1}{2}$ gives

$$\mathrm{Var}\{e_{DF2}[nT]\} = \frac{2q^2}{12} + \frac{q^2}{12}\ \frac{(1-b_1)}{2}\ . \tag{69}$$

Obviously without scaling (67) is less than (63) for values of b_1 between -1 and 1. However, for a step input, the value of T_n for the DF2 in steady state is

$$(T_n)_{n\to\infty} = \frac{1}{1-b_1} \tag{70}$$

For positive values of b_1, (68) is greater than one and overflow will occur. To avoid this, the input X_n is usually scaled down by a scale factor $1-b_1$. This additional multiplication introduces another noise term at the input. To obtain the correct scaling at the output the two coefficients $a_0' = a_0/(1-b_1)$ and $a_1' = a_1(1-b_1)$. For $a_0 = a_1 = (1-b_1)/2$, the resulting filter has a steady state variance

$$\mathrm{Var}\{e_{DF2}(nT)\} = \frac{2q^2}{12} + \frac{q^2}{12}\ (\frac{.5}{1-b_1}) + \frac{r^2q^2}{12}\ (\frac{.5}{1-b_1}) \tag{71}$$

where r is the scale factor for the quantization caused by the scale factor at the input. In general r is greater than one. A comparison of the variance of the two forms in the scaling as employed in the DF2 implementation is given in Table 2. A dc gain of one is assumed and $a_0 = a_1 = (1-b_1)/2$.

TABLE 2

COMPARISON OF THE VARIANCE OF DFI AND DFZ

b_1	a_0	a_1	r	$\frac{\text{Var DF1}}{q^2/12}$	$\frac{\text{Var DF2}}{q^2/12}$
.1	.45	.45	2	3.03	4.77
.2	.4	.4	2	3.13	5.13
.3	.35	.35	2	3.30	5.57
.4	.3	.3	2	3.57	6.17
.5	.25	.25	4	4.0	19.00
.6	.2	.2	4	4.69	23.25
.7	.15	.15	4	5.88	30.33
.8	.1	.1	8	8.33	164.00
.9	.05	.05	16	15.79	1287.00
.99	.005	.005	128	150.75	819252.00

From the table it is obvious that when scaling is employed in the first order filter, the output error due to product rounding is much greater for the DF2 form. In general, scaling must be done via the use of interactive graphics for the desired set of input signals. Also it is important to recognize that for a given filter there is no analytical method to determine the best form for a given section and an optimal ordering of the sections that will result in no overflow and minimum variance due to product rounding. Thus, the designer must carefully utilize interactive computation along with a graphics terminal to scale the filter and order the sections.

More detail on the implementation of digital filters is given in Appendix . While the procedures described in the Appendix are pointed toward digital control, they are directly applicable to any digital filter. Both input and product rounding errors along with scaling are discussed. The method for computing the variance is shown in Appendix along with an example for product rounding. This algorithm is well suited for use with an interactive graphics terminal.

One other phenomenon that is present due to product rounding is the limit cycle. At present there is no general theory; however,

it is well known that rounding or truncation of products can cause a digital filter to exhibit an oscillation or limit cycle. This occurs with both fixed and floating point arithmetic. A reference is by Brubaker and Gowdy [13].

Coefficient Rounding

The last source of error in a filter is caused by the rounding or truncation of the filter coefficients. While some writers describe these errors as random, in practice they are deterministic and under control of the programmer. In filters with rigid specifications, the actual rounding or trucation of the coefficients must be done with care or the filter charcteristics will not satisfy the specifications. For example, for low-pass filters the dc gain can change dramatically if the coefficients are not handled properly. This is illustrated in Appendix E where a second order lag-lead digital filter is investigated. By use of a strategy for rounding and truncating the coefficients, the filter coefficients can be handled with twelve bits plus a sign bit. When conventional rounding is employed the use of twelve bits plus sign gives a dc gain of 4.88×10^5 when the desired dc gain is one.

At present there are not many general procedures for coefficient rounding. In Appendix E, the rounding is handled by first specifying how much error in the filter magnitude and phase functions is acceptable. Then, the use of a differential approximation allows the error to be represented in terms of the filter coefficients and the coefficient errors. A linear programming method is then used to establish a region where the coefficient error vector must lie. Note that while the strategy in Appendix E is used for digital control algorithms it is applicable to any filter.

Another important consideration that was pointed out in the design section is the fact that the accuracy with which the coefficients must be represented is dependent on the sampling interval T. For high order filters the accuracy requirement drastically increases as the order of the filter increases. As a result, another reason for implementing filters as a cascade or parallel connection of first and second order sections is to reduce error due to coefficient rounding. Thus the designer must carefully compromise the sampling time to meet the specifications and to minimize the effect of coefficient rounding. Then the filter is implemented as a cascade or parallel connection of first and second order sections. The actual determination of the number of bits needed for each first and second order section is done via the strategy described in Appendix E.

To summarize the overall implementation strategy, the designer first designs a given filter using the lowest possible sampling frequency. The filter is then factored for implementation as a cascade or parallel connection of first and second order sections. The ordering of the sections and the type of connection to be utilized is done using interactive computer packages to establish a configuration that minimizes the effect of product rounding and still has no overflow. Then the error due to input rounding and product rounding is used to compute the word length needed to meet the specifications on these two errors. Finally the coefficients are rounded and the word length is determined to meet the specifications on coefficient rounding. The final word length is chosen as the larger of the two.

III. MULTIPLEXING IN THE IMPLEMENTATION OF DIGITAL FILTERS

In a sense, any time a computer containing a single multiplier and adder is used to implement a digital filter multiplexing is done. To see this simply consider the nonrecursive digital filter given by

$$Y_n = \sum_{k=1}^{M} a_k x_{n-k} \tag{72}$$

If a single multiplier is employed each product in (72) is computed sequentially and multiplexing of the multiplier is effectively done in time. When more than one filter is implemented on a digital computer time division multiplexing can be employed. This is shown in Fig. 2 where two channels are handled by a single computer. At time t_{10} a sample is taken on channel 1 and an output generated at time t_{11}. The computational time is less than $T/2$ seconds. At time t_{20} a sample is taken from channel 2 and the output is computed for use at time t_{21}.

If, however, there is ample computational time, both samples may be taken at time t_o. The response to each input sample is then computed sequentially and both outputs are generated at time t_1 for use by the separate channels. This is shown in Fig. 3. Here, at the sampling instants each input is sampled by a sample-and-hold circuit and the inputs are converted into binary and stored in memory. The output for the first filter is computed during Computer Time 1 and the output for the second filter is generated during Computer Time 2. The two outputs are stored in buffer registers that are clocked to the output channel lines at the end of the sampling interval.

At this point it is important to realize that most computer and/or data acquisition system manufacturers do not design the control for the analog-to-digital conversion and digital-to-analog conversion subsystems

in the right way. Since the coefficients and thus the digital filter characteristics are very dependent on the sampling interval T the control data acquisition system must be carefully designed. Thus, the real time clock should be an integral part of the data acquisition system and control should not be done via interrupts.

For example, in many computer systems the real time clock is a feature of the computer. This consists of a preset counter and when the count reaches zero an interrupt is activated to tell the computer to take a sample from the data. This is done via a subroutine that controls the analog-to-digital converter. However, in normal operation, the time to jump to the service routine may vary and in many cases the variation will be a large percentage of the sampling interval. As a result the actual sampling interval is varying in a random fashion or is in error by a constant. The characteristics of the filter will now deviate from the desired characteristics. For example, if a band-stop filter is being implemented, the center frequency will be shifted. It is also possible to achieve instability in the digital filter.

In a good system, a programmable real-time clock will be specified as part of the data acquisition system. This clock will clock a sample point from the analog-to-digital converter and then interrupt the computer to indicate a sample has been taken and is ready for processing. If a digital-to-analog converter is used, the filter output will be computed and stored in a buffer. At the time when a new sample is taken, the output data will be transferred to the digital-to-analog converter.

When more than one channel is serviced by the computer, a multiplexer is used in front of the analog-to-digital converter. Here care must be taken to avoid a time lag between samples. This can be handled

by using a sample-and-hold on each channel before the multiplexer. Another alternative is to allow a fixed delay between sampling points. By including the same delay at the output the sampling interval for a given channel will be constant.

If a computer is used to implement the digital filters in an aircraft control and avionics system, the data will be transmitted to and from the computer in two ways. First for every channel, an individual wire can be used. Secondly, a bus structure can be employed. In this case, the data will be time multiplexed on the bus. If a single bus is employed, both the input and output data will use the bus.

When an individual wire is used for each channel, the synchronizing of the data to achieve a uniform sampling time is not difficult and the data acquisition system should be specified as outlined previously. When a bus structure is used, unless the bus is operated synchronously there is a danger of introducing variations in the sampling interval for a given channel. These variations can cause the digital filter to perform improperly. As a result, the designer must carefully specify the amount of allowable error in the sampling interval to establish a bus operating procedure that will satisfy the sampling interval specification.

In terms of reliability, redundancy is usually employed by using both redundant computers and redundant bus structures. In actual operation faults are detected using a majority vote.

As the integrated circuit manufacturers develop the new microprocessor systems, a more realistic procedure for handling the digital filtering operations consists of a distributed computing system. In such a system a separate microprocessor can be employed to implement each digital filter. By placing each microprocessor under the control of the central computer rapid reprogramming of the filter coefficients can be achieved

as operating conditions change. Improved reliability in terms of aircraft operation can also be achieved by a filter priority structure. This is illustrated in Fig. 4. Referring to the figure, three microprocessors are used to implement three digital filters in channels 1, 2 and 3. The priority has been predetermined to be channel 1 highest priority, channel 2 next highest priority and channel 3 lowest priority. The channel inputs and outputs are switched into the appropriate microprocessors under control of the central computer. If microprocessor 1 fails, channel 1 is switched into microprocessor 3 and channel 3 is disconnected. Microprocessor 3 is then reprogrammed to handle the filter for channel 1. If desired, additional redundancy can be included in each microprocessor.

If a single data bus is employed, the control computer will program the proper address into each microprocessor so that the data will be pointed into the correct microprocessor. In the event of a high priority microprocessor failure, the control computer would reprogram the address into a low priority microprocessor and discontinue pointing data from the low priority channel.

Because the present structure of a digital flight control and digital avionics systems appears to be based on using one computer for handling all of the digital filtering operations, the development of simulation software for multiple digital filters for digital control was started as a part of the project. The computer was a Digital Equipment Corporation PDP-11. The complete description of this work was sent to the contract monitor as a thesis referenced in [14]. In this report only the necessary details are included.

The general structure of the multiplexed digital control system is shown in Fig. 5 for four individual control channels. In the

simulation, each channel input is sequentially multiplexed. When a sample is taken from channel 1, the proper filter algorithm is called and the filter output is computed and sent to the digital-to-analog converter. Then channel 2 is sampled etc. Thus, each channel is handled separately with the proper filter coefficients being placed in a memory stack corresponding to the given channel. The flow chart is shown in Fig. 6.

The software package for this simulation is not complete because of the change in the data acquisition system. However, the program using an outdated data acquisition system is complete and is described in Reference [14].

To summarize, the current trend in the digital avionics and flight control system is to utilize redundant computers. This implies that a single computer will be used to handle all of the digital filtering operations in the aircraft. The data will be handled via a bus and it will be necessary to design this in such a way that the sampling of data will correspond to the way that sampling is done in the filter design process.

The multiplexing for this configuration will consist of a single computer used to implement a number of filters. For a given number of filters, all of the input sample points at a given time can be taken at once and all filter output computed during a single sampling interval or a single filter can be implemented during a sampling interval. In either case the data acquisition system must be carefully specified and designed. If this is not done timing will be difficult and a considerable software overhead will be required to acquire and send out data.

In future designs it is desirable to consider the use of distributed computing systems with microprocessors being used to implement digital filtering. By employing a priority multiplexed switching system under control of a central computer, the highest priority filtering operations can be maintained during flight.

IV. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Currently it appears the digital filter design using the bilinear z transform is most suitable for digital flight control. However for many avionics applications, time domain synthesis may be more effective. In the future it is important that attention be given to developing design methods that are more suited to the z domain. For example, the sampling interval should be used as a design parameter. For effective design it is necessary to develop good interactive software to work with a graphics terminal.

In addition it is important that the designer carefully consider the implementation of a digital filter when doing the design. Specifically the effects of rounding must be considered. By the use of interactive design software, and the procedures developed in Appendices 3, 4, and 5, the word length needed for an implementation can easily be determined.

When the design for the complete aircraft system is developed, the method of moving data within the aircraft must be carefully considered. If this is not done the shift in sampling interval can cause severe problems in the way a digital filter performs. As part of this work, attention must be given to the specification and design of the data acquisition systems in order to minimize software overhead and to insure proper filter operation.

While multiplexing can be employed future systems should utilize a distributed computing structure for greater reliability. This will reduce the need for a large central computer and will allow greater flexibility in the implementation of digital filtering algorithms.

Last, the need for developing good interactive graphics is again emphasized. In addition, a simulation system should be established to

allow Air Force personnel to carefully develop specifications for digital avionics and flight control systems. The effective use of simulation and interactive design will result in more effective, less expensive and more reliable aircraft systems.

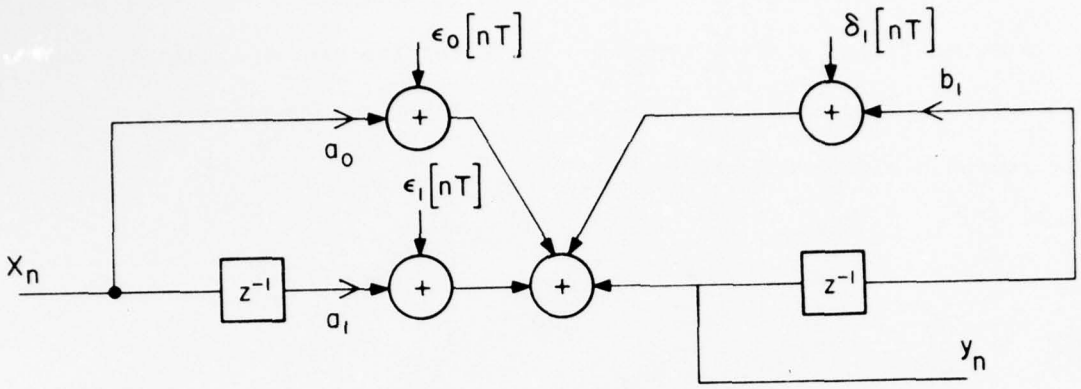


Fig. 1a

Direct Form 1 Block Diagram

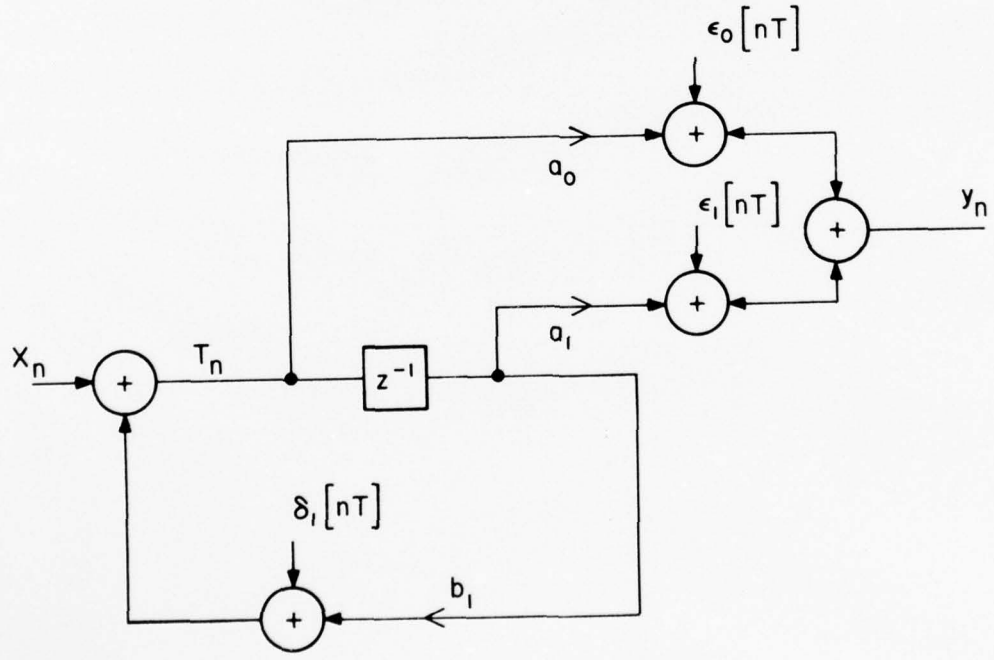


Fig. 1b

Direct Form 2 Block Diagram

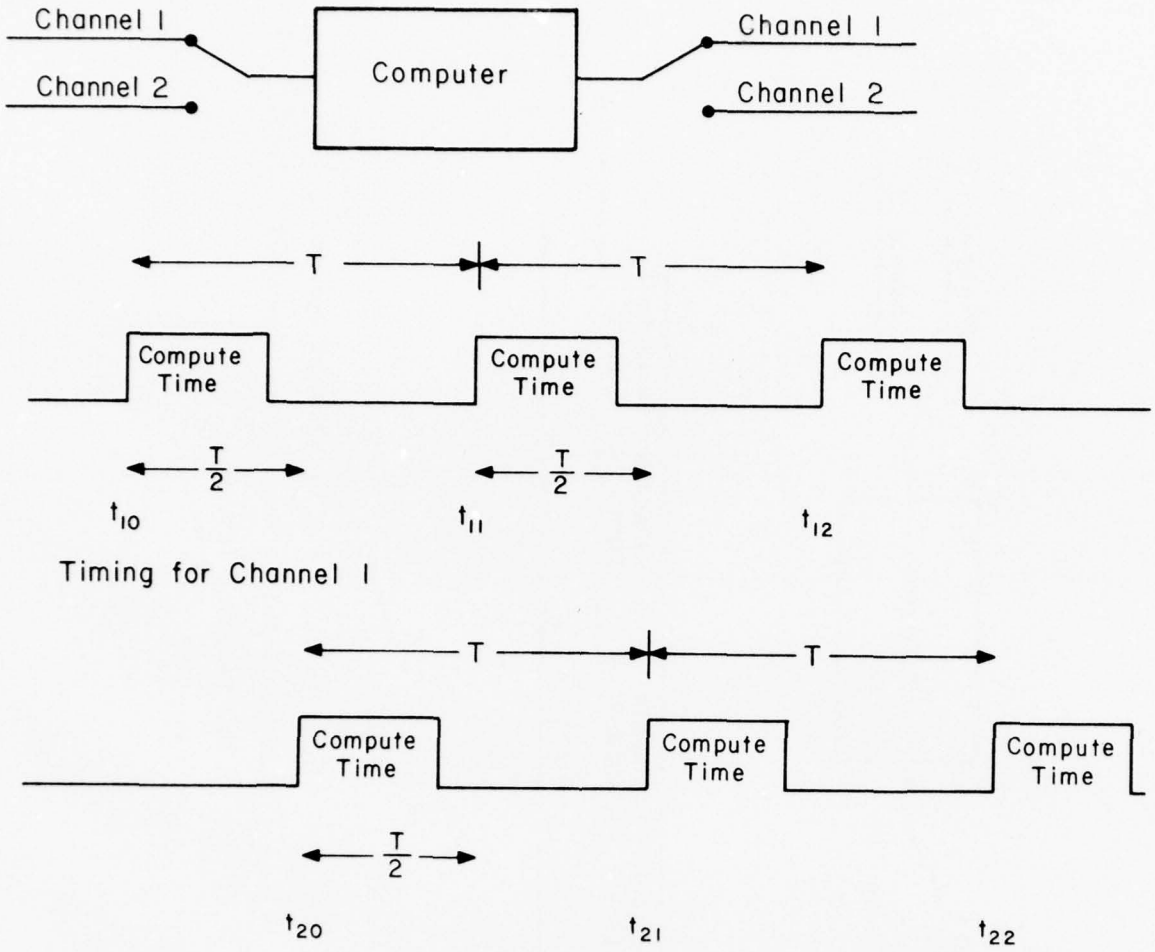


Fig. 2 Block Diagram for a Two Channel Time Multiplexed Filter

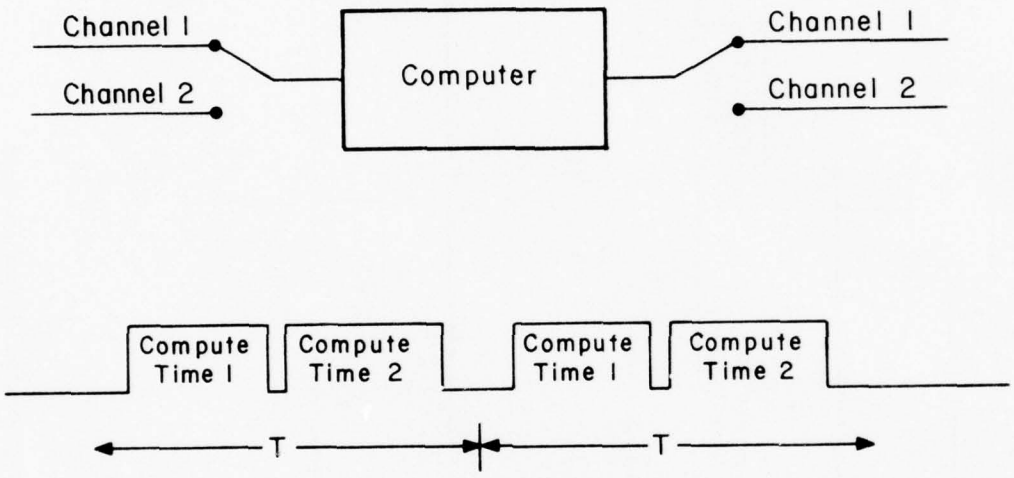


Fig. 3 Block Diagram for a Two Channel System Where Both Filter Outputs are Generated in One Sampling Interval

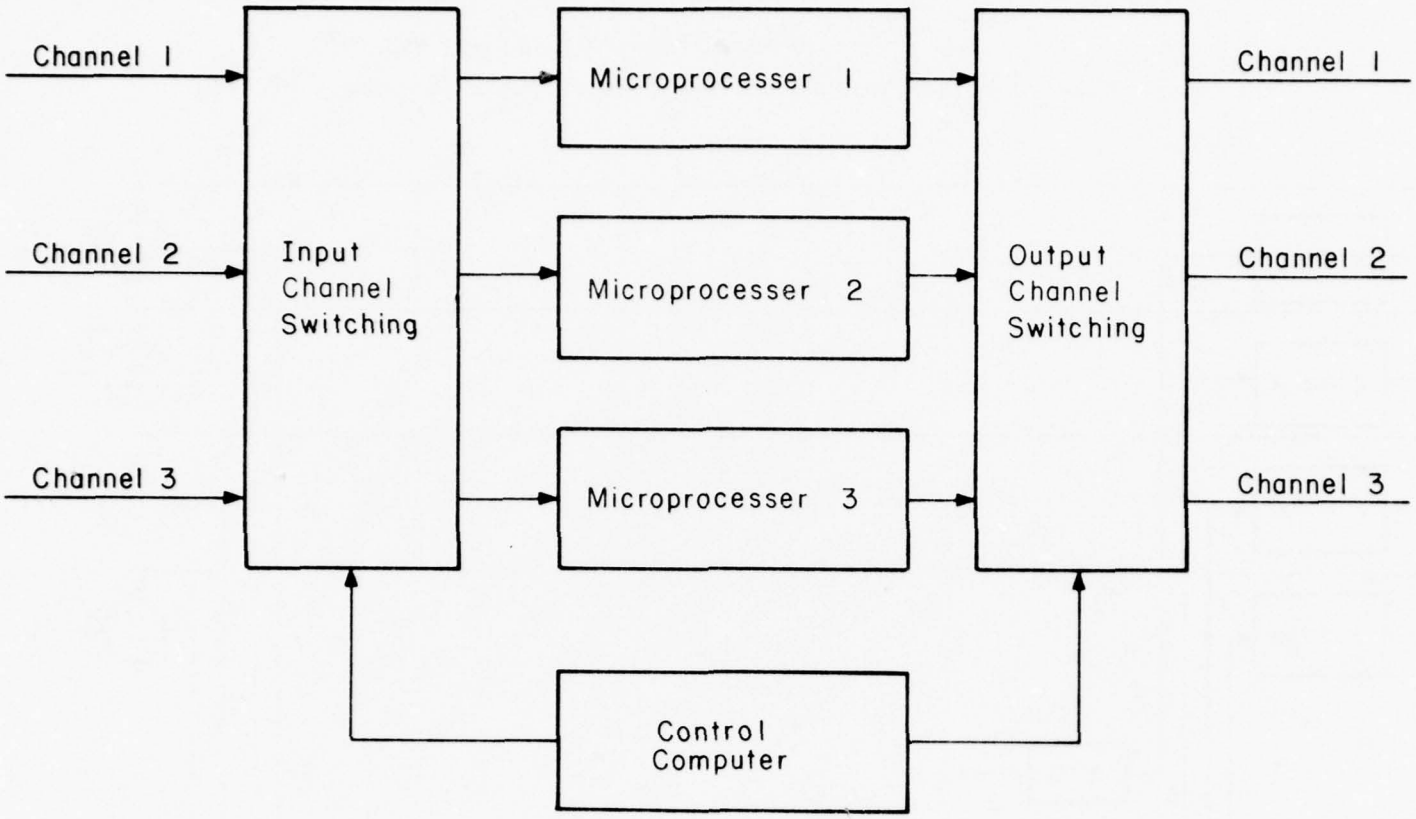


Fig. 4 Block Diagram for Priority Channel Control

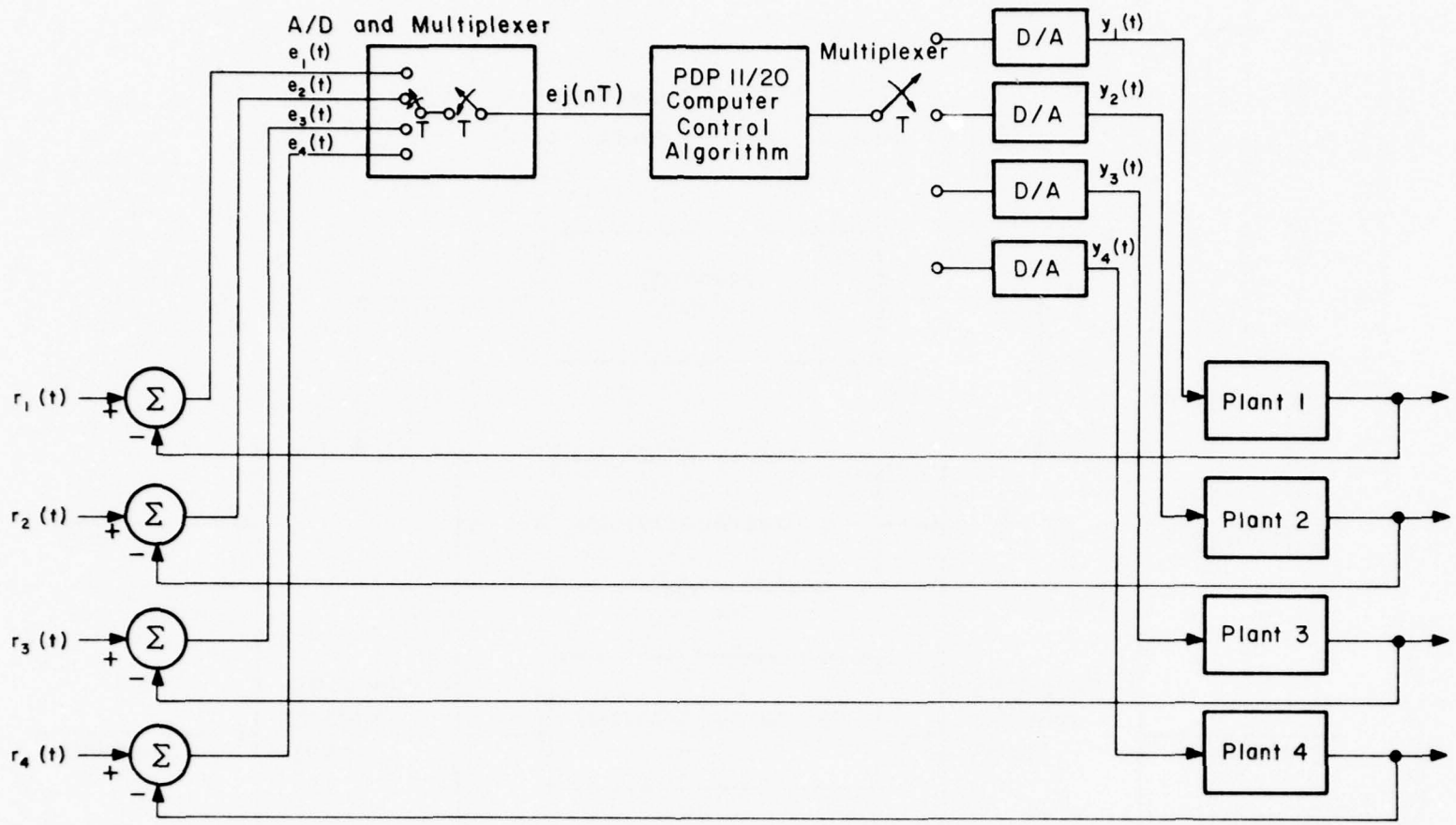


Fig. 5. Model of a multiplexed digital control system implemented on the PDP 11/20 computer system

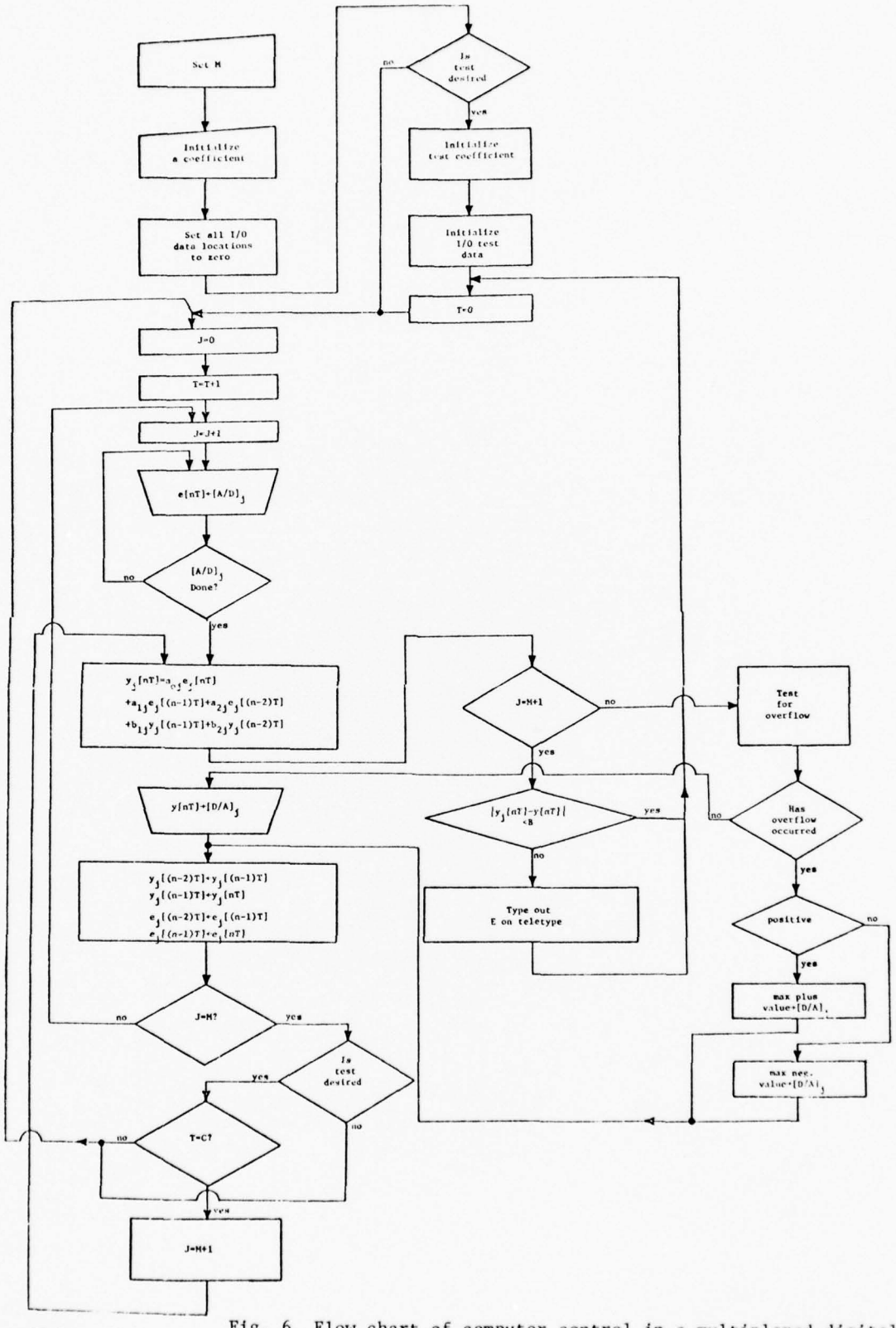


Fig. 6 Flow chart of computer control in a multiplexed digital control system implemented on the PDP 11/20.

REFERENCES

1. T. A. Brubaker and D. L. Harper, "Time Domain Synthesis of Recursive Digital Filters," International Journal of Control, Vol. 18, No. 4, pp. 721-730, 1973.

2. N. Morrison, Introduction to Sequential Smoothing and Prediction, McGraw-Hill Book Company, 1969.

3. L. R. Rabiner, B. Gold and C. McGonegal, "An Approach to the Approximation Problem for Nonrecursive Digital Filters," IEEE Trans. on Audio and Electrocoustics, Vol. AU-10, pp. 85-106, June 1970.

4. L. R. Rabiner, "The Design of Finite Impulse Response Digital Filters Using Linear Programming Techniques," Bell System Technical Journal Vol. 51, pp. 1177-1192, July-Aug. 1972.

5. D. C. Farden and L.L. Scharf, "Statistical Design of Nonrecursive Digital Filters," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-22, pp. 188-195, June 1974.

6. L. R. Rabiner, N.Y. Graham, and D. H. Helms, "Linear Programming Design of IIR Digital Filters with Arbitrary Magnitude Function," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-22, pp. 117-123, April 1974.

7. B. Gold and C. M. Rader, Digital Processing of Signals, McGraw-Hill Book Company, 1969.

8. W. R. Bennett, "Spectra of Quantized Signals," Bell System Technical Journal, Vol. 27, No. 3, 1942.

9. B. Widrow, "Statistical Analysis of Amplitude Quantized Sampled-Data Systems," AIEE Trans. Part 2, pp. 555-562, 1961.

10. J. E. Bertram, "The Effect of Quantization in Sampled-Data Feedback Systems," AIEE Trans., Vol. 77, pp. 177-182, 1958.

11. J. B. Slaughter, "Quantization in Digital Control Systems," IEEE Trans. on Automatic Control, Vol. AC-9, pp. 70-74, 1964.

12. B. D. Liu, "Effects of Finite Word Length on the Accuracy of Digital Filters," IEEE Trans. on Circuit Theory, Vol. CT-10, Nov. 1971.

13. T. A. Brubaker and J. N. Gowdy, "Limit Cycles in Digital Filters," IEEE Trans. on Automatic Control, Vol. Ac-17, No. 5, pp. 675-677, October 1972.

14. B. E. Elliott, "Multiplexed Digital Control Systems," M.S. Thesis, Department of Electrical Engineering, Colorado State University, Fort Collins, Colo. 80521.

APPENDIX A

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_1 respectively, the pre-warped upper (WDU), lower (WDL), and center (WDM) frequencies and the bandwidth between WDU and WDL are found by

$$WDU = \frac{2}{T}\tan\left(\frac{\omega_u T}{2}\right)$$

$$WDL = \frac{2}{T}\tan\left(\frac{\omega_1 T}{2}\right)$$

$$WDM = \frac{2}{T}\tan\left[\frac{\sqrt{\omega_u \omega_1}\,T}{2}\right] \qquad (3)$$

$$WB = WDU - WDL$$

ω_u and ω_1 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

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

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

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

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

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

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.

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

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

$$S_{\pm 1} = \omega_c \underline{/\pm 135^\circ}$$

$$\theta = 45^\circ$$

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 the poles lie in the left-half plane because poles in the left-half plane are stable. Putting $\theta = 45^\circ$ 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:

$$B, A = \frac{1}{2}((\sqrt{\varepsilon^{-2} + 1} + \varepsilon^{-1})^{1/N} \pm (\sqrt{\varepsilon^{-2} + 1} + \varepsilon^{-1})^{-1/N}) \tag{26}$$

where

$$\varepsilon = \left[\frac{1}{(1 - \delta)^2} - 1\right]^{1/2} \tag{27}$$

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

$$\text{Real Part} = A\cos\theta$$
$$\text{Imaginary Part} = \beta\sin\theta \tag{28}$$

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

$$H(S) = \frac{K_2\left[\frac{1}{\sqrt{1+\varepsilon^2}}\right]^{2/N}}{S^2 + K_8 S \quad K_2} . \tag{29}$$

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

$$K_8 = 2A\cos\theta \tag{30}$$
$$K_2 = A^2\cos^2\theta + B^2\sin^2\theta \quad . \tag{31}$$

The substitution of the low-pass to band-pass transformation, equation (14), into equation (29) yields

$$H(S) = \frac{S^2 WB^2 K_2\left[\frac{1}{\sqrt{1+\varepsilon^2}}\right]^{2/N}}{S^4 + S^3 K_8 WB + S^2(2WDM^2 + K_2 WB^2) + SK_8 WDM^2 WB + WDM^4} . \tag{32}$$

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

$$A_0 = 1$$

$$A_1 = 0$$

$$A_2 = -1$$

$$K_{1i} = \frac{2WB}{T} \cdot \frac{K_3}{G_i} \tag{34}$$

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 equai 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 $1/(1 + \epsilon^2)^{\frac{1}{2}} = 1 - \delta$ 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	212	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) = -9.99788916   IMAGINARY(2) = -52.14095930
REAL(3) = -4.55076557   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 = -0.19810630E+01  B2 =  0.98760851E+00
```

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.

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

$$K_8 = 2A\cos(\theta)$$

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

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.

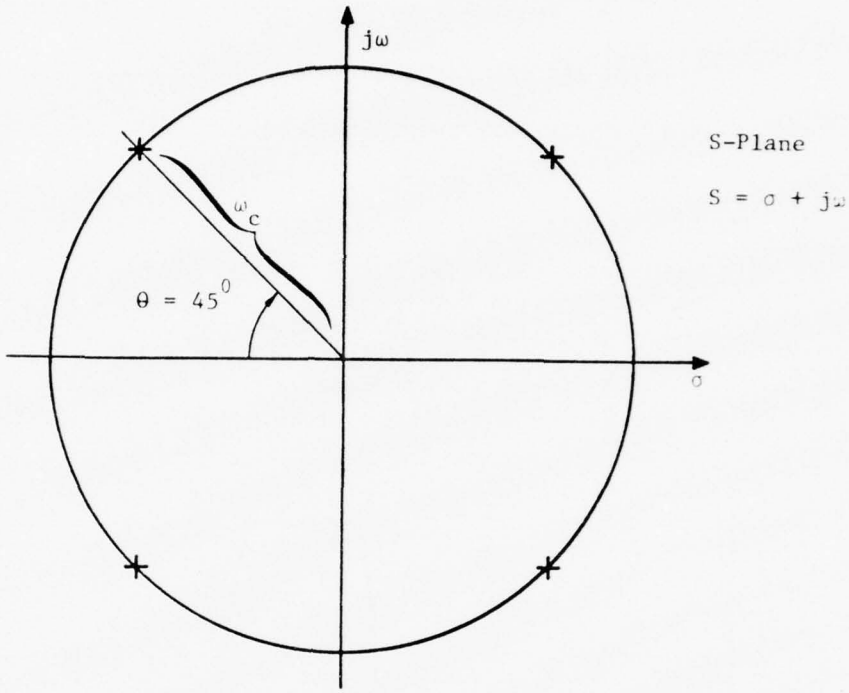


Figure 1

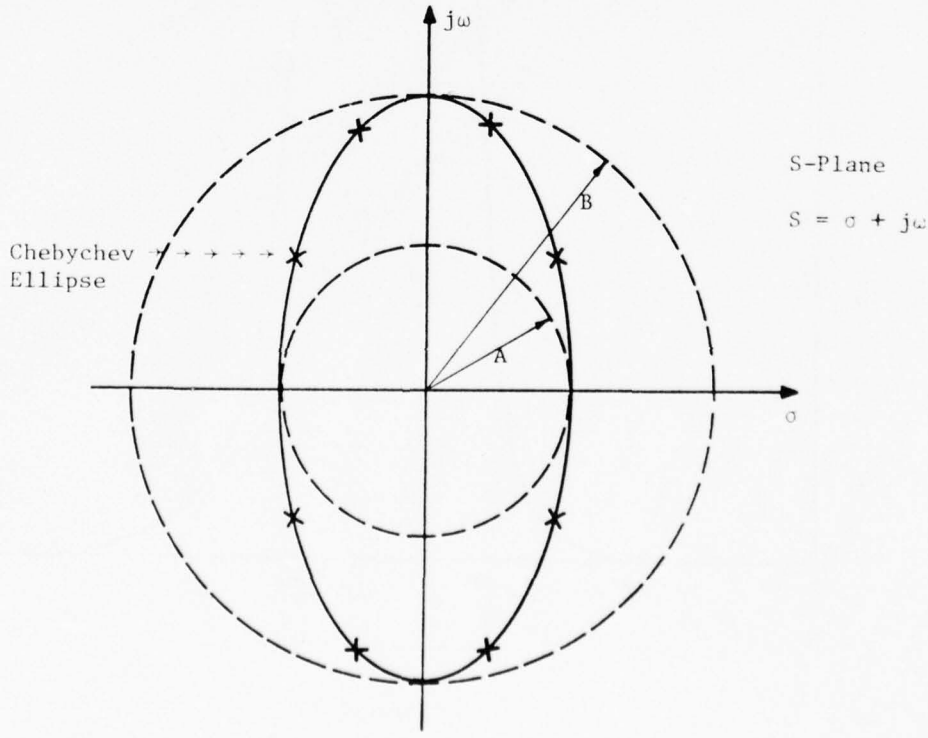


Figure 2

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_1 = 40 radians

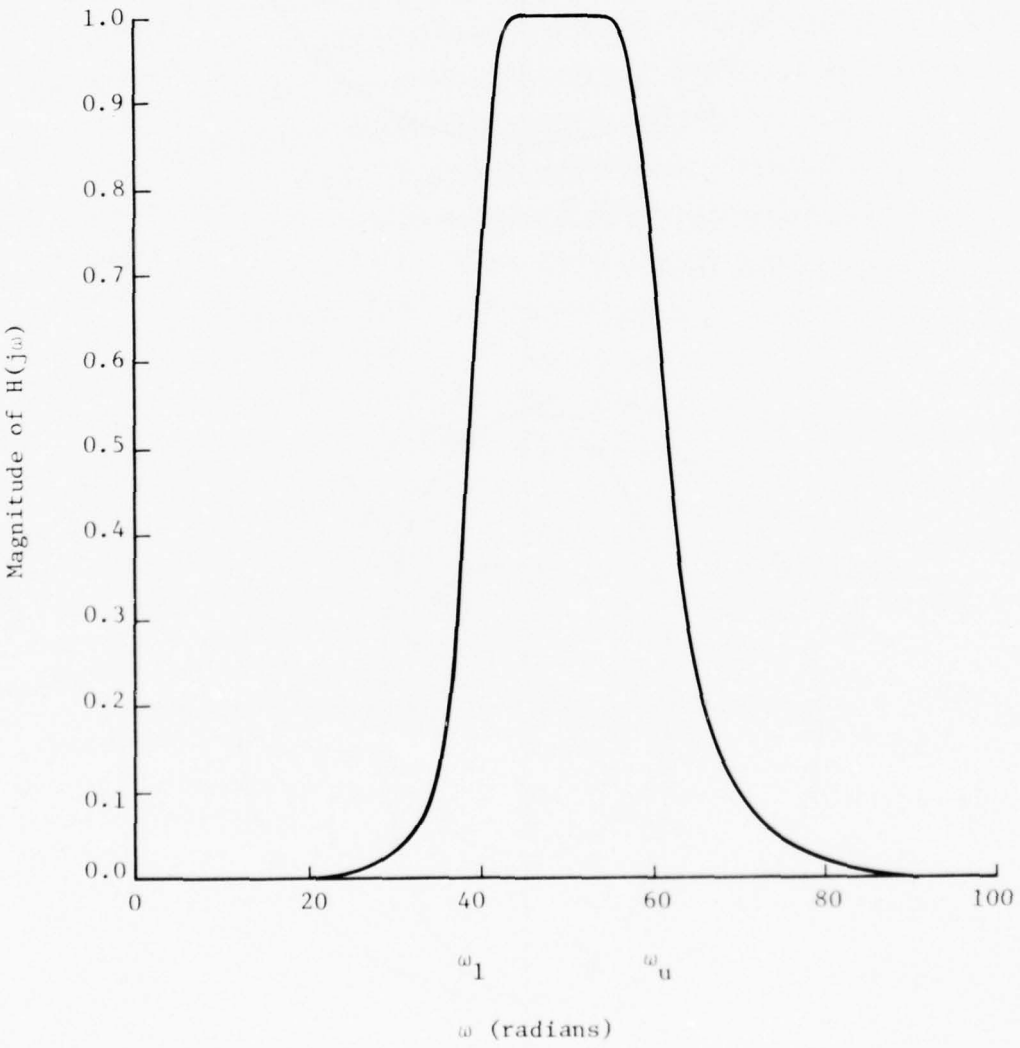


Figure 3

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

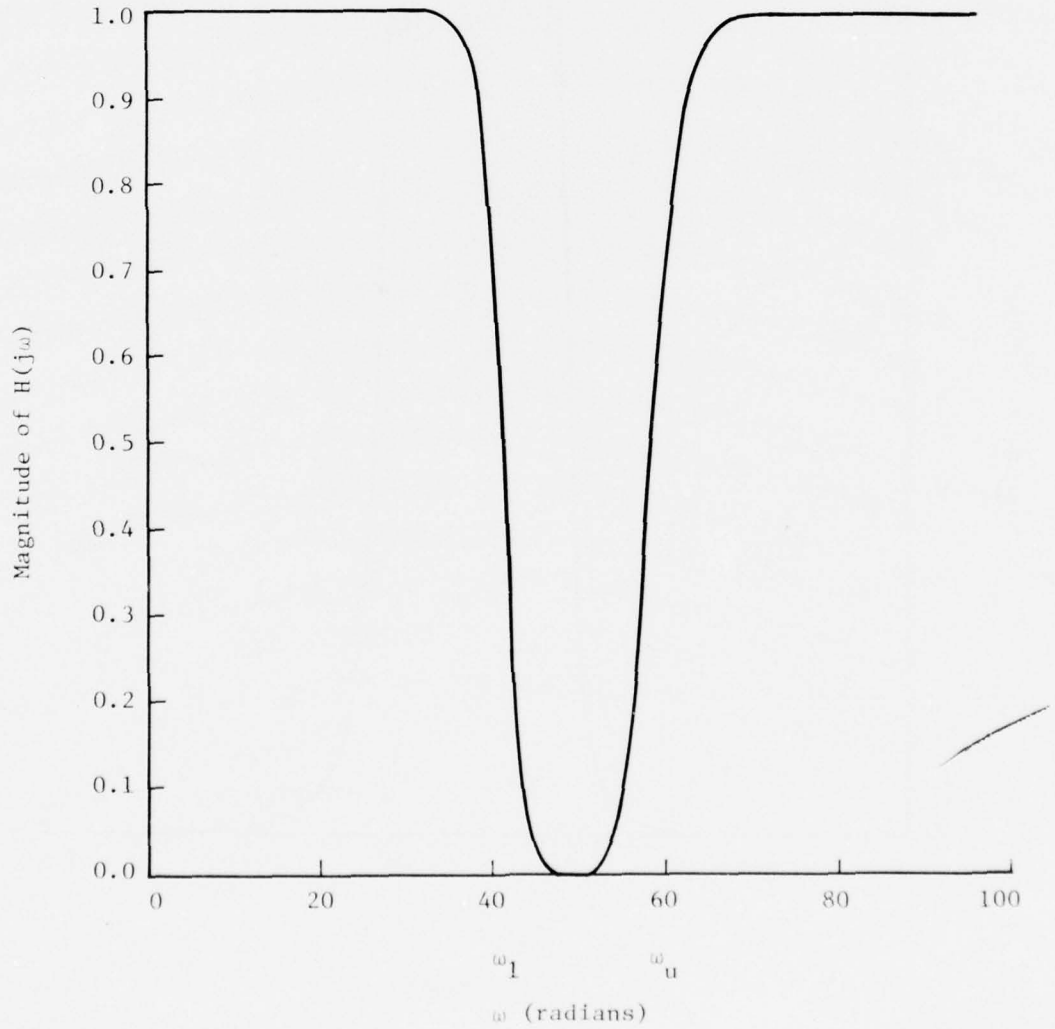


Figure 4

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

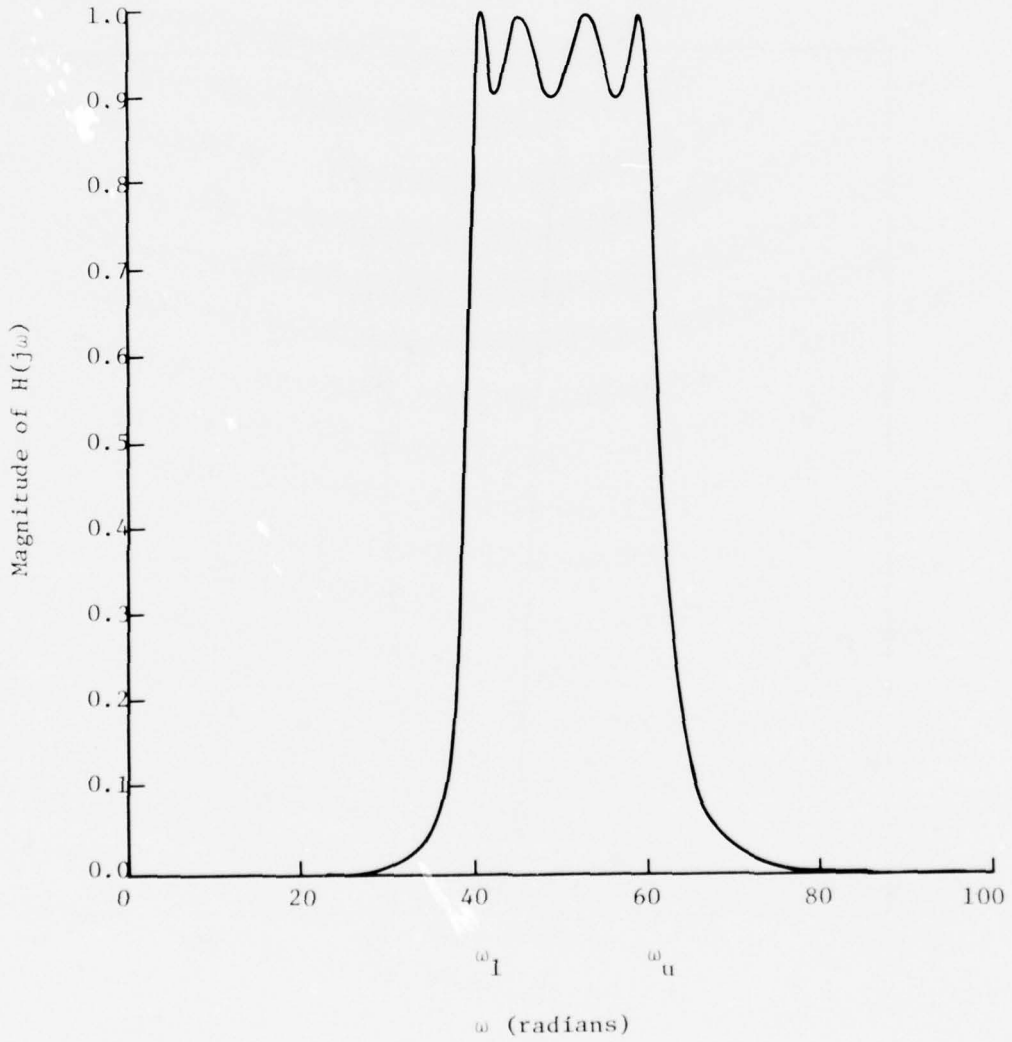


Figure 5

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

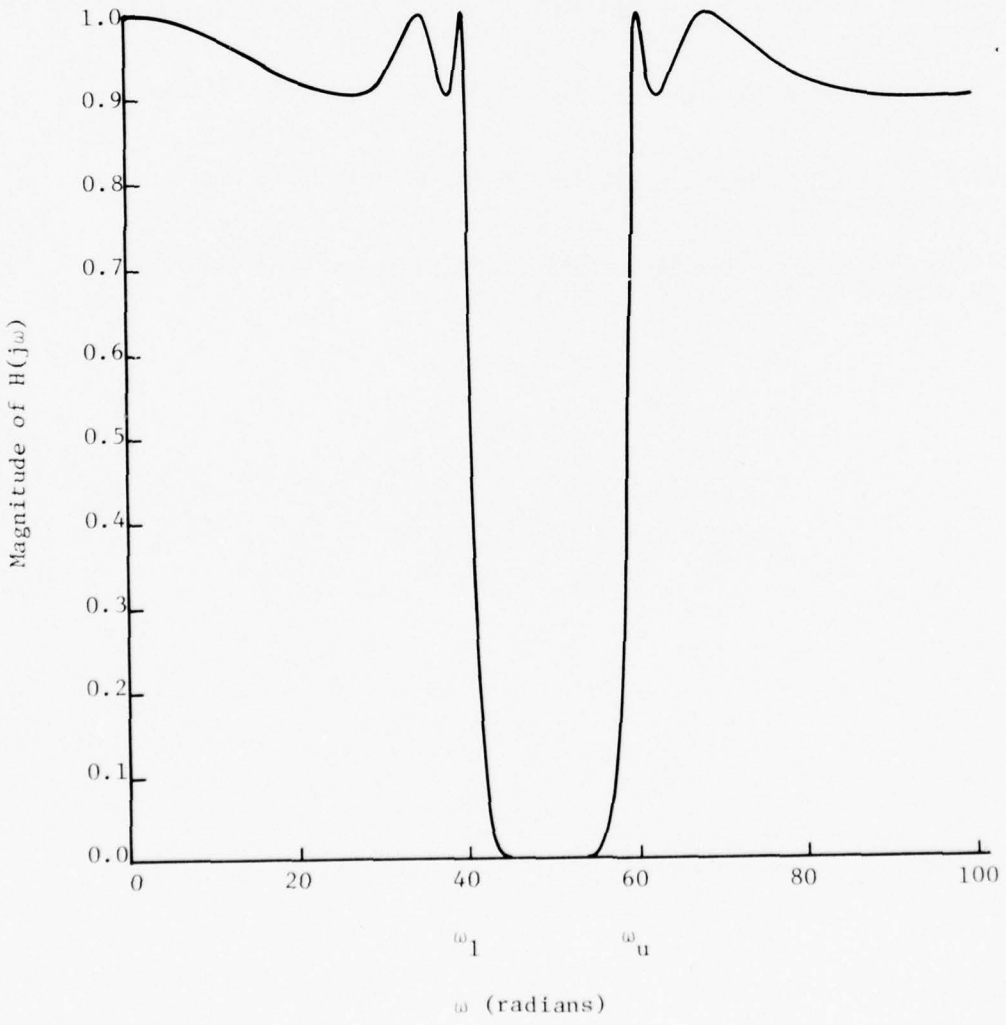


Figure 6

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 B

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:

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

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

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

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

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

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

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

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.

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

Substituting equations (14) into equation (11) yields

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

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

$$S = -\omega_c \ e^{\pm j\theta} = \omega_c[-\cos\theta \pm j\sin\theta]. \tag{16}$$

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.

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

$$S_{\pm 1} = \omega_c \angle \pm 45°$$

$$S_{\pm 2} = \omega \angle \pm 135°$$

$$\theta = 45°$$

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 poles lie in the left-half plane since poles in the left-half plane are stable. Putting $\theta = 45°$ 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)

$$A_0 = A_2 = 1$$

$$A_1 = 2$$

$$G_i = \frac{4}{T^2} + \frac{4}{T} \text{ WDC } \cos\theta_i + \text{WDC}^2 \tag{21}$$

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

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

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

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

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

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

$$H(Z) = \frac{K_2 aWDC^2(Z^2+2Z+1)}{\frac{4}{T^2}(Z^2-2Z+1) + \frac{2}{T}K_8WDC(Z^2-1) + WDC^2K_2(Z^2+2Z+1)} \ . \tag{31}$$

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, Butterworth)
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.

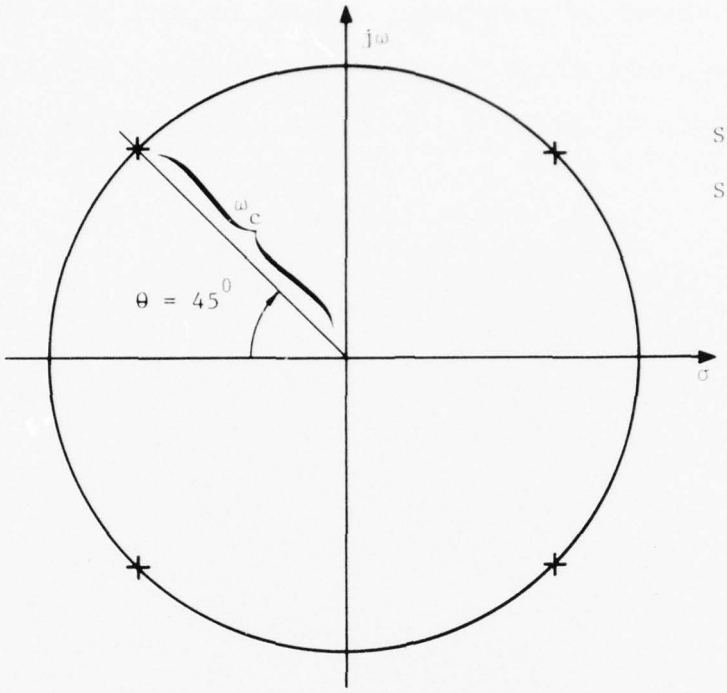
H1 is the magnitude of the digital transfer function (1^{st} stage).

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.

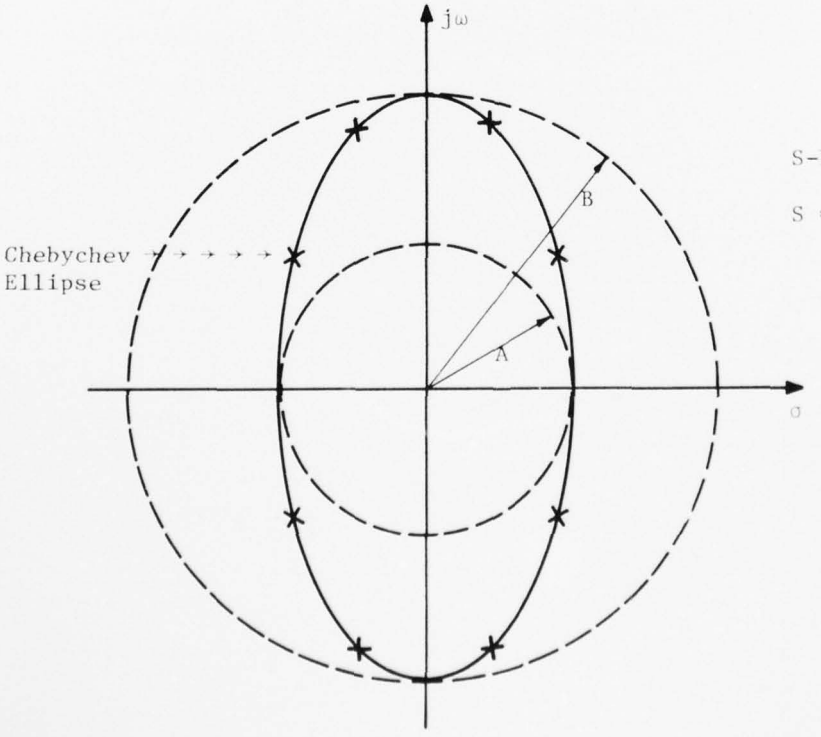
See Figure 8.



S-Plane

S = σ + jω

Figure 1



S-Plane

S = σ + jω

Figure 2

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

2ND ORDER BUTTERWORTH LOW-PASS FILTER

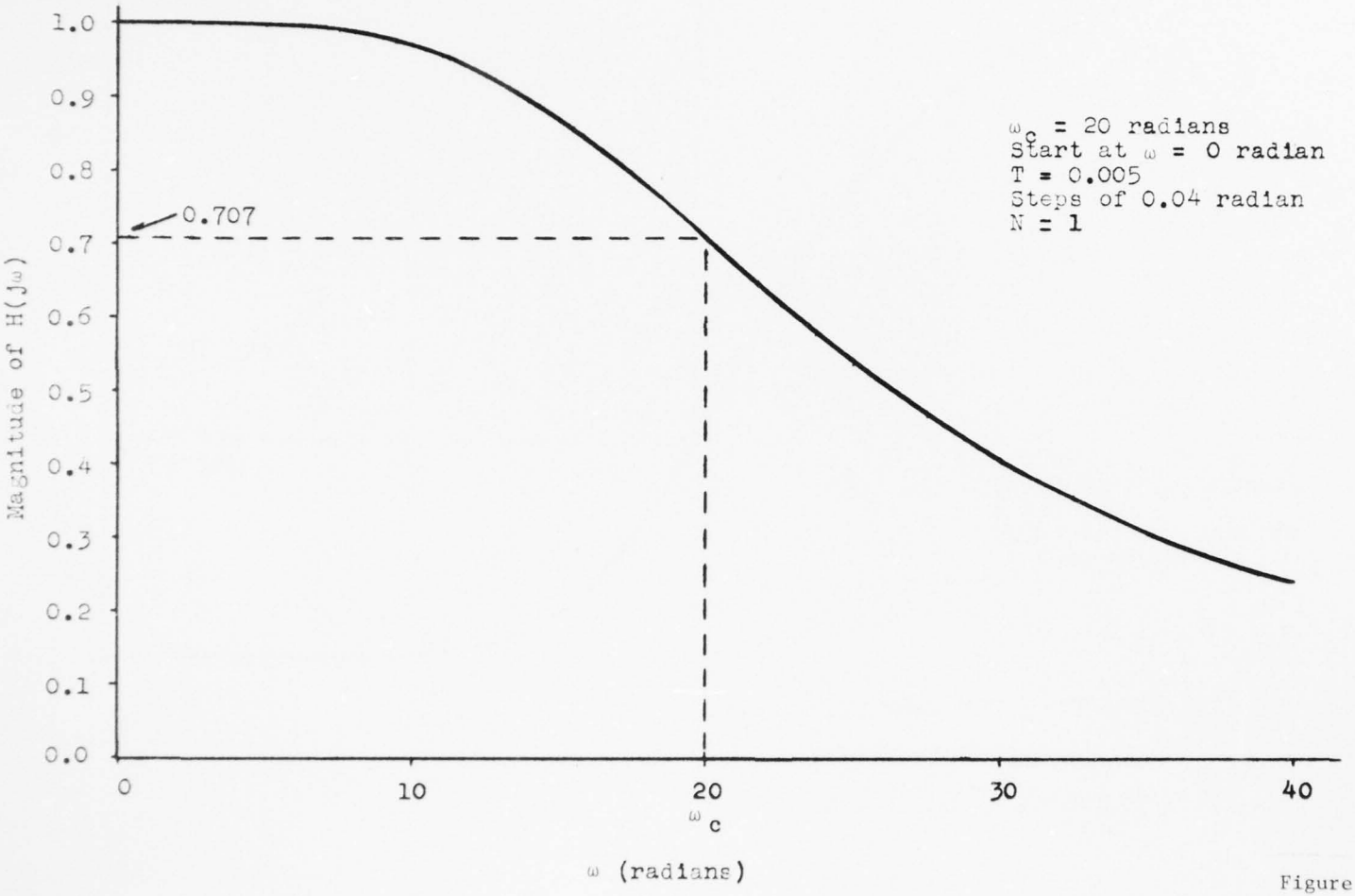
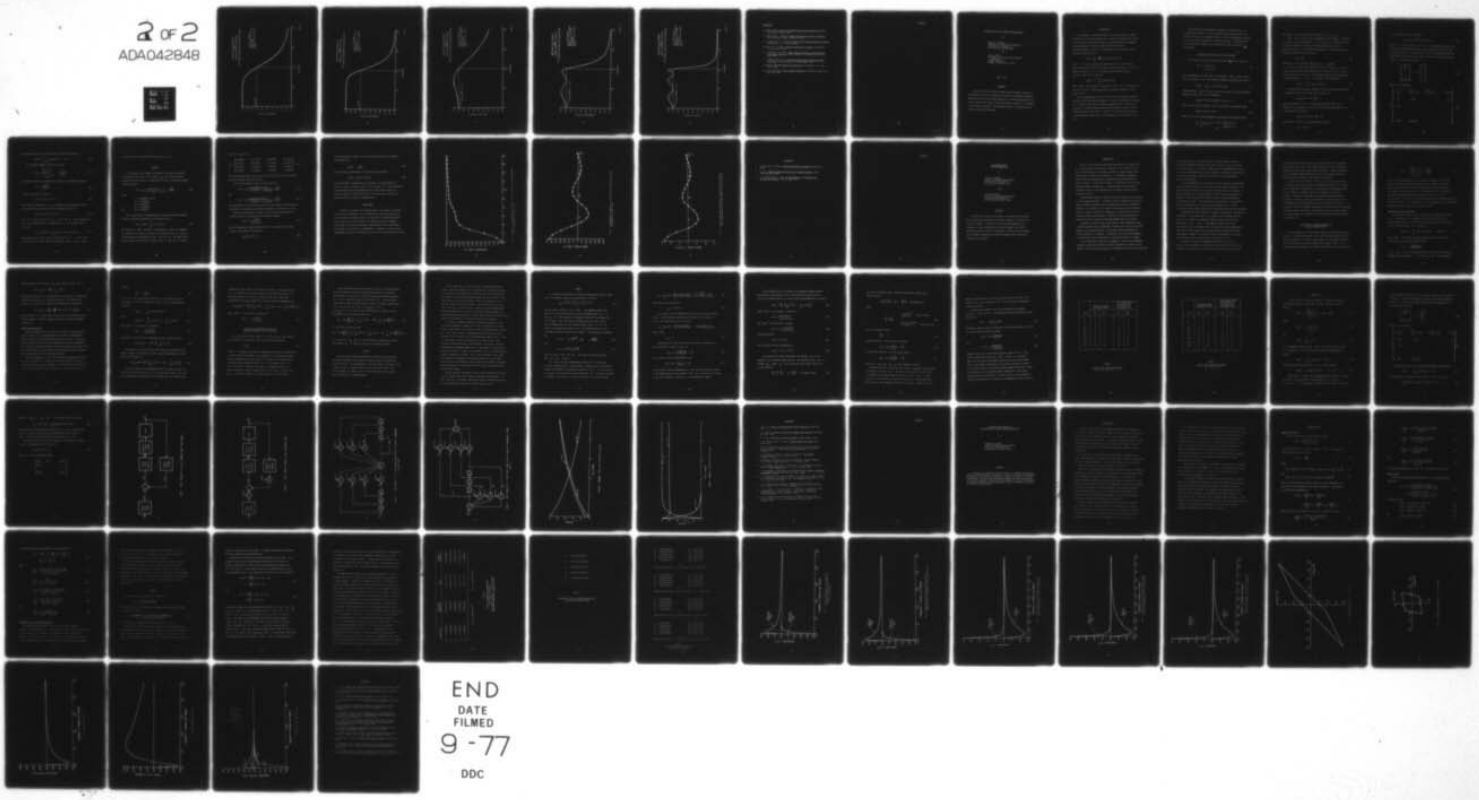


Figure 3

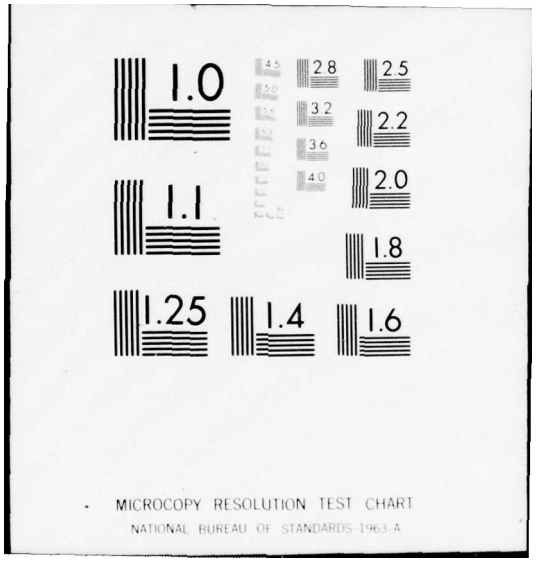
AD-A042 848 COLORADO STATE UNIV FORT COLLINS DEPT OF ELECTRICAL --ETC F/G 9/5
DIGITAL FILTER DESIGN AND IMPLEMENTATION METHODS.(U)
JUL 77 T A BRUBAKER

UNCLASSIFIED F33615-73-C-1253
AFAL-TR-75-211 NL

2 OF 2
ADA042848

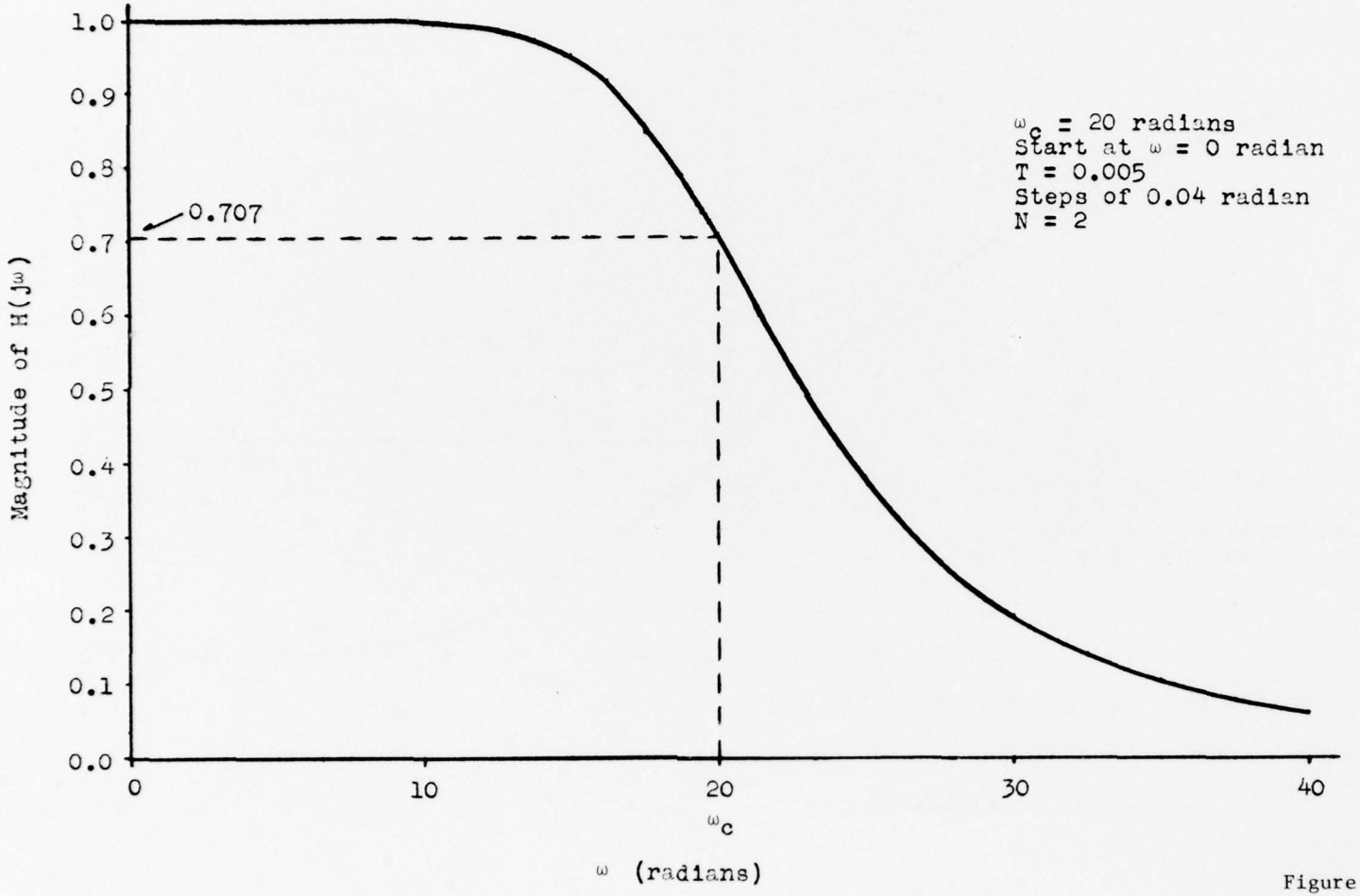


END
DATE FILMED
9-77
DDC


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

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

4^{TH} ORDER BUTTERWORTH LOW-PASS FILTER



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

Figure 4

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

6^{TH} ORDER BUTTERWORTH LOW-PASS FILTER

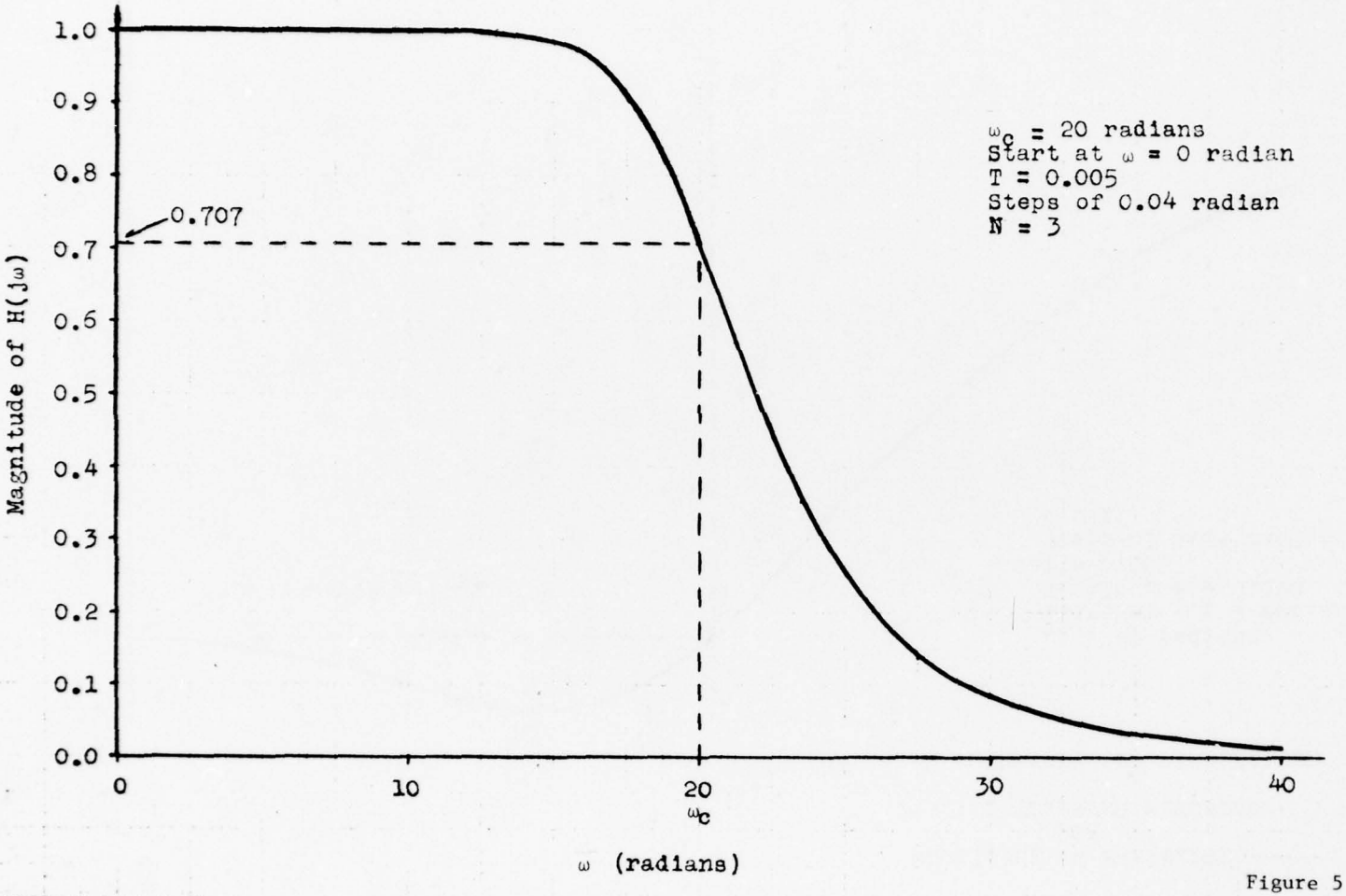


Figure 5

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

2ND ORDER CHEBYCHEV LOW-PASS FILTER

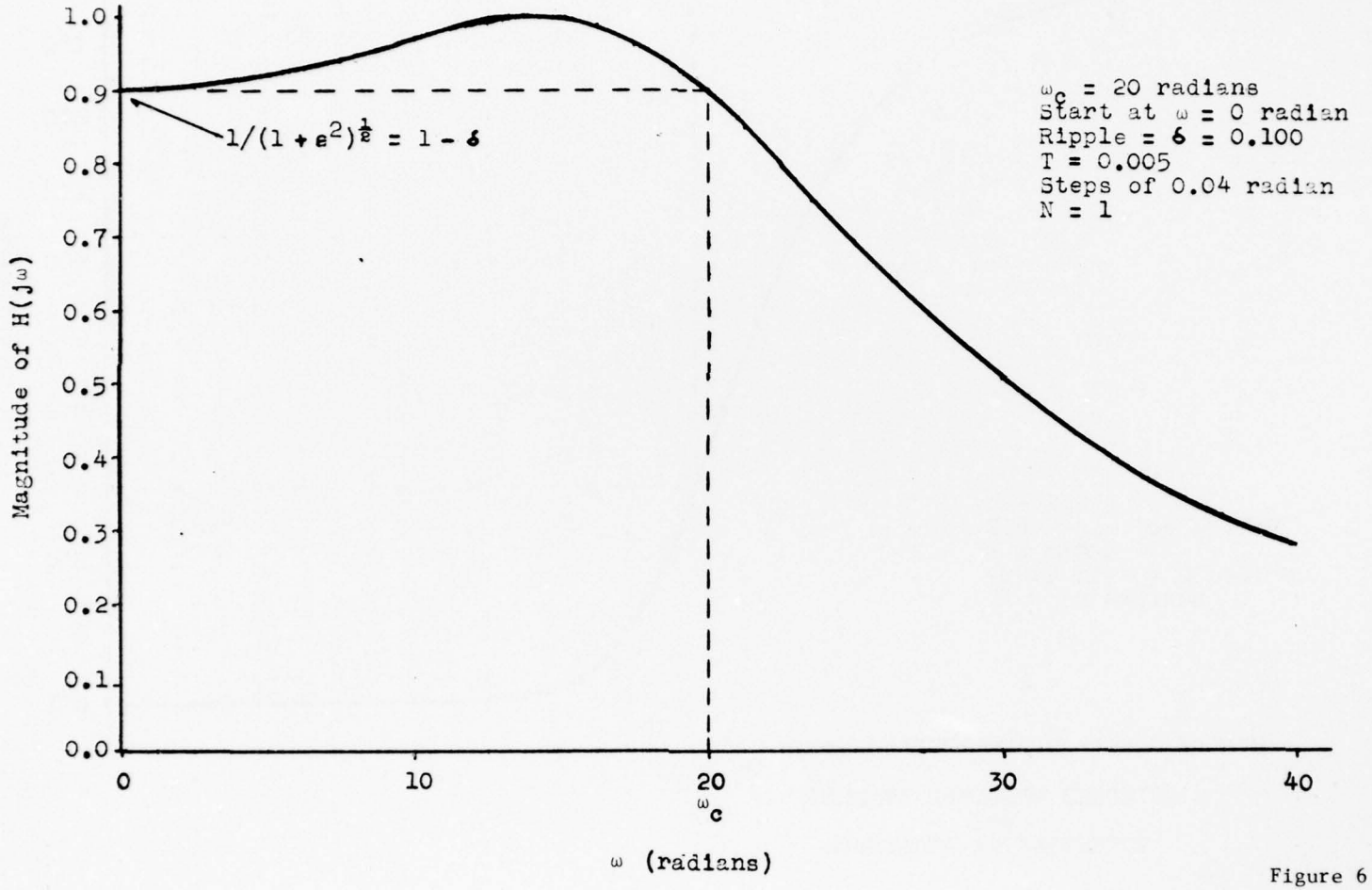
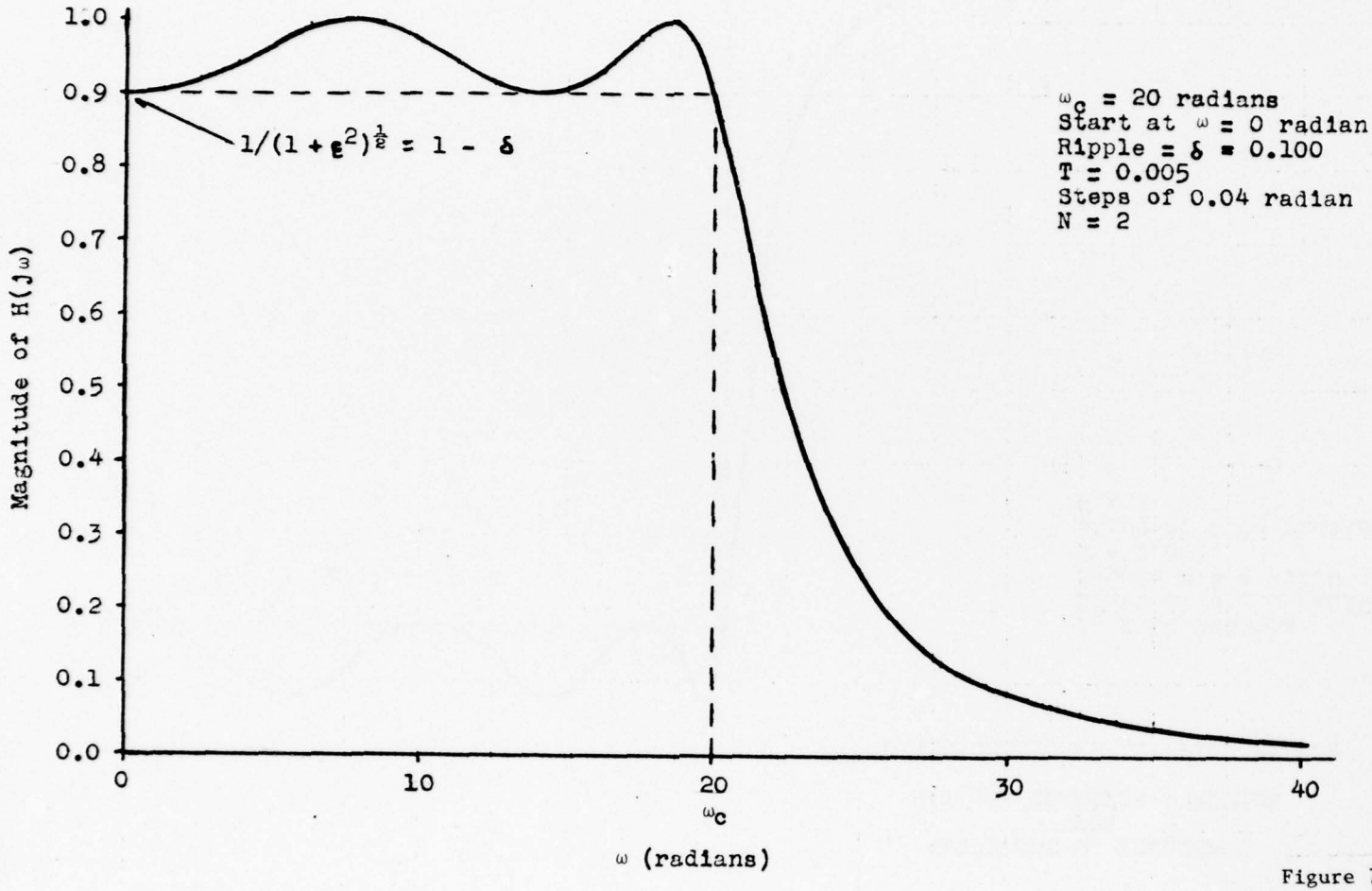


Figure 6

MAGNITUDE VS FREQUENCY
FOR
DIGITAL TRANSFER FUNCTION

4^{TH} ORDER CHEBYCHEV LOW-PASS FILTER

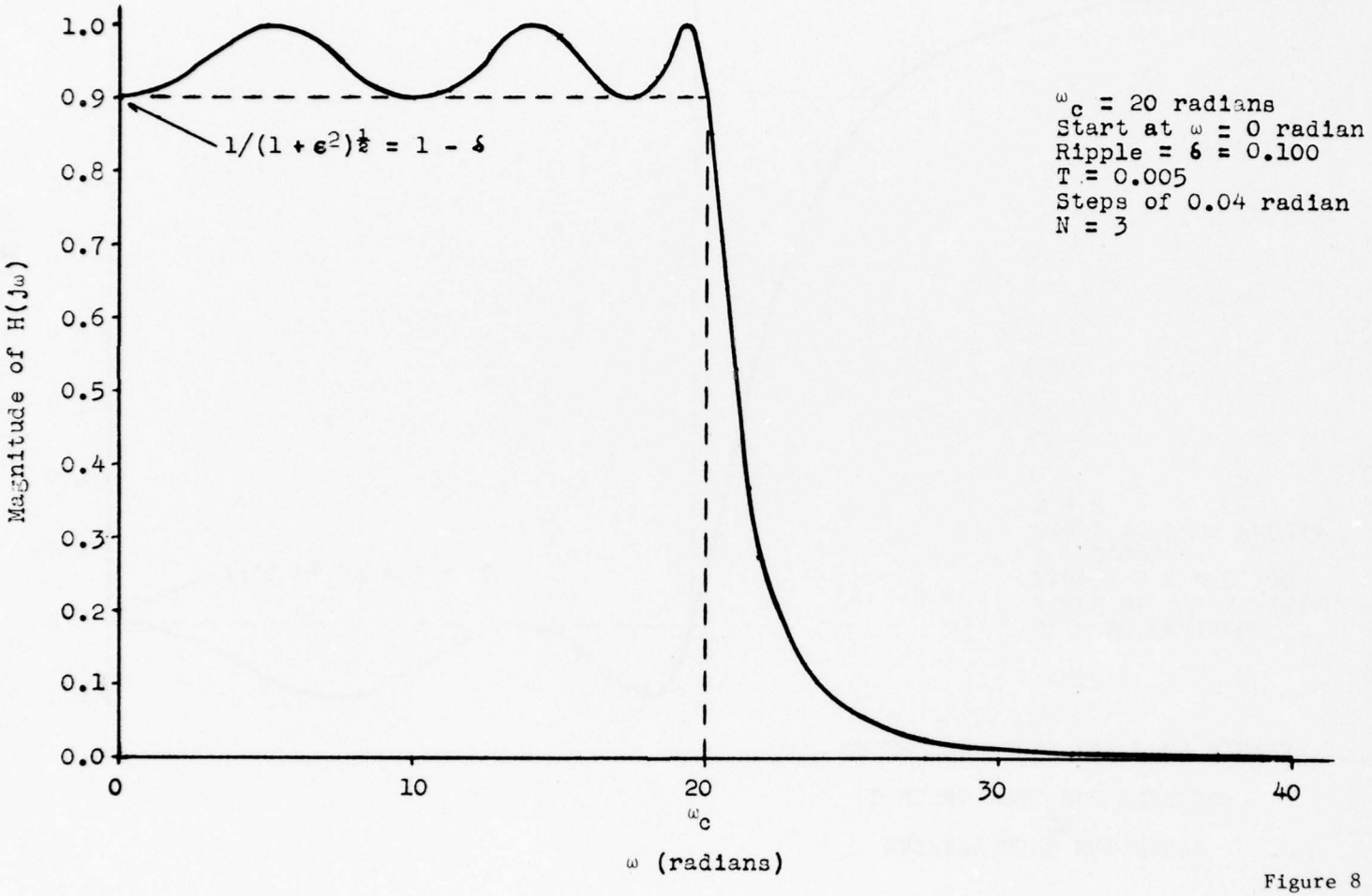


ω_c = 20 radians
Start at ω = 0 radian
Ripple = δ = 0.100
T = 0.005
Steps of 0.04 radian
N = 2

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 C

COMPUTATION OF THE DISCRETE AUTOCOVARIANCE

by

Thomas A. Brubaker
Department of Electrical Engineering
Colorado State University
Fort Collins, Colorado 80521

and

John N. Gowdy
Department of Electrical and Computer
Engineering
Clemson University
Clemson, South Carolina 29631

May 1, 1973

ABSTRACT

Given an Nth order discrete system the autocovariance sequence is formulated as a set of recursive and nonrecursive difference equations. Application of the final value theorm for a z transform then permits the first N terms of the steady state autocovariance sequence to be computed via one matrix inversion. The remaining terms are calculated using a simple recursive relationship.

INTRODUCTION

In the design of discrete systems it is often necessary to compute the variance or the autocovariance due to input and/or quantization noise. In many instances these noise sources can be represented as independent zero mean sequences [1] which allows the steady state autocovariance sequence $\{R[kT]\}$ to be represented by the discrete Wiener-Khinchine relationship [1,2]

$$R[kT] = \frac{\sigma^2}{2\pi j} \oint H[z]\, H[z^{-1}]\, z^{k-1}\, dz \quad . \tag{1}$$

In (1) σ^2 is the variance of the noise and $H[z]$ is the discrete transfer function from the noise source to the system output. For a noise sequence starting at $t=0$, the transient autocovariance sequence $\{R_n[kT]\}$ time $t=nT$ is given by

$$R_n[kT] = \sigma^2 \sum_{i=0}^{n-k} h[(i+k)T]\, h[iT] \tag{2}$$

where $h[nT]$ is the inverse z transform of $H[z]$. As m becomes large, each term $R_m[kT]$ in the autocovariance sequence converges to $R[kT]$ given by (1).

From a computational point of view, (1) can be evaluated using the residue theorm from complex variables. If only the steady state variance is of interest, Jury [2] has derived an expression for $R[OT]$ that is the ratio of two determinents whose terms are simple functions of the transfer function coefficients. The use of (2) requires the evaluation of the weighting sequence and a series summation for all desired values of k.

In this paper the autocovariance sequence is formulated as a set of difference equations. If only the steady state autocovariance is of interest, the use of z transforms leads to a set of simultaneous equations that can be easily evaluated via one matrix inversion. Remaining terms in the autovariance sequence are found using a simple nonrecursive relationship.

DIFFERENCE EQUATIONS FOR THE AUTOCOVARIANCE

First consider an Nth order discrete transfer function of the form

$$H[z] = \frac{1}{1 + \sum_{j=1}^{N} b_j z^{-j}} \quad . \tag{3}$$

For an independent zero mean input noise sequence $\{w[nT]\}$ with variance σ_w^2 that starts at $t=0$, the difference equation describing the system is

$$y[nT] = - \sum_{j=1}^{N} b_j \, y[(n-j)T] + w[nT] \quad . \tag{4}$$

The mean value of $y[nT]$ is zero for all n and terms in the autocovariance sequence at time $t=nT$ are now defined as

$$R_n[kT] = E\{y[nT] \; y[(n-k)T]\} \quad k=0,1,2,\ldots n \quad . \tag{5}$$

From (4) and (5) the variance of $y[nT]$ is given by the quadratic form

$$R_n[0T] = \bar{b}^t \, Q_n \, \bar{b} + \sigma_w^2 \tag{6}$$

where Q_n is a time varying symmetric autocovariance matrix with terms

$$\begin{aligned} q_{ij} &= R_{n-\min(i,j)}\,[|i-j|T] && n-\min(i,j) \geq 0 \\ &= 0 && n-\min(i,j) < 0 \end{aligned} \quad . \tag{7}$$

The vector $\bar{b}^t$ is defined as the row matrix $[-b_1 \; -b_2 \; \ldots -b_n]$.

Since $w[nT]$ is taken from an independent noise sequence, $y[(n-k)T]$ and $w[nT]$ are uncorrelated for $k \geq 1$ and the next N terms in the autocovariance sequence are found by multiplying both sides of (4) by $y[(n-k)T]$, $k=1,2,\ldots N$ and taking the expectation. This yields a set of equations described in vector form as

$$\bar{R}_n = Q_n \, \bar{b} \tag{8}$$

where $\bar{R}_n^{\,t}$ is the row matrix $\left[R_n[T] \; R_n[2T] \; \ldots \; R_n[NT]\right]$.

The $N+1$ equations described by (6) and (8) consist of a set of $N/2$ coupled recursive difference equations for N even and $(N+1)/2$ coupled recursive equations for N odd. The remaining equations are nonrecursive difference equations. For $k>N$ the $R_n[kT]$ terms are given by

$$R_n[kT] = - \sum_{i=1}^{N} b_i \, R_{n-i} \, [(k-i)T] \; . \tag{9}$$

A closed form steady state solution is derived by first defining the z transform of the autocovariance term $R_{n-i}[kT]$ as

$$Z\{R_{n-i}[kT]\} = z^{-i} \, R_z[kT] \tag{10}$$

where the subscript $(n-i)$ represents the discrete time $(n-i)T$. Taking the z transform of (6) and (8) now yields the expression for the z transform of the variance

$$R_z[0T] = \bar{b}^t \, Q[z] \, \bar{b} + \frac{z}{z-1} \, \sigma w^2 \tag{11}$$

and the next N terms of the autocovariance sequence

$$\bar{R}_z = Q[z] \, \bar{b} \quad . \tag{12}$$

In (11) the matrix $Q[z]$ has terms

$$q_{ij}[z] = z^{-\min(i,j)} R_z[|i-j|T] \tag{13}$$

where $|i-j|$ takes on values 0,1,2,...N-1. Expanding the quadratic form given by (11) and applying the final value theorm to this and the first N-1 equations given by (12) now yields a set of N simultaneous equations whose solution forms the first N terms of the steady state autovariance sequence. In matrix form we have

$$\begin{bmatrix} R[OT] \\ R[T] \\ . \\ . \\ R[(N-1)T] \end{bmatrix} = M^{-1} \begin{bmatrix} \sigma w^2 \\ 0 \\ . \\ . \\ 0 \end{bmatrix} \tag{14}$$

where M is the matrix

$$M = \begin{bmatrix} 1 - \sum_{i=1}^{N} b_i^2 & -2\sum_{i=1}^{N-1} b_i b_{i+1} & -2\sum_{i=1}^{N-2} b_i b_{i+2} & . & . & -2b_1 b_N \\ (b_1) & (1+b_2) & . & & & b_N \\ (b_2) & . & . & & . & \\ . & . & . & & . & \\ (b_N) & (b_{N-2}+b_N) & . & & & 1 \end{bmatrix} \tag{15}$$

The remaining terms in the autocovariance sequence are given by

$$R[kT] = - \sum_{i=1}^{N} b_j \, R[(k-i)T] \qquad k \geq N. \tag{16}$$

For a system transfer function of the form

$$H[z] = \frac{\sum_{i=0}^{V-1} a_i \, z^{-i}}{1 + \sum_{j=0}^{N} b_j \, z^{-j}} = \frac{N[z^{-1}]}{D[z^{-1}]} \tag{17}$$

the transfer function is separated by defining an intermediate operation

$$x[z^{-1}] = \frac{W[z^{-1}]}{D[z^{-1}]} \tag{18}$$

so that the response is given by

$$y[z^{-1}] = N[z^{-1}] \; x[z^{-1}] \quad . \tag{19}$$

The autovariance sequence for x_n is found using the preceeding results. The terms in the output autocovariance sequence are given by

$$E\{y[nT] \; y[(n-k)T]\} = \bar{a}^{t} \, C_n \, \tilde{a} \tag{20}$$

where $\bar{a}^{t}$ is the row matrix $[a_0 \; a_1 \; \ldots \; a_{V-1}]$ and C_n is the appropriate V by V time varying matrix. In steady state C_n is a constant matrix with terms

$$C_{ij} = R[(k-j+i)T] = \lim_{n \to \infty} E\{x[nT] \; x[(n-(k-j+i))T]\} \; . \tag{21}$$

When computing the steady state autocovariance for $K < V$ certain terms in the C matrix are found from the relationship $R[kT] = R[-kT]$. Thus,

a fixed nonrecursive expression exists only for $K \geq V$.

EXAMPLE

To illustrate the procedure, we consider a low pass fourth-order Butterworth filter with a -3db point at $w_0 = 10$. The filter was designed using the bilinear z transform with $T=0.01$ and has the general transfer function

$$H[z] = \frac{K[z+1]^4}{z^4 + b_1 z^3 + b_2 z^2 + b_3 z + b_4} = \frac{N[z]}{D[z]} \tag{22}$$

where

$$k = 4.69832343 \times 10^{-3},$$
$$b_1 = -2.53346973,$$
$$b_2 = 2.65559567,$$
$$b_3 = -1.28757608,$$

and

$$b_4 = 0.24062331.$$

For a direct form of implementation [1] with nine product rounding errors the difference equation for the output error is

$$e[nT] = p[nT] - \sum_{j=1}^{4} b_j \, e[(n-j)T] \quad . \tag{23}$$

The variance of $p[nT]$ is $9q^2/12$. The variance of $e[nT]$ was computed as a function of n using (6) and (8) and is plotted in Fig. 1. This was checked using the form given by (2) for $k=0$ and $n=\infty$. The steady state autocovariance for the filter is shown in Fig. 2. Here the M^{-1} matrix

from (14) is given by

$$M^{-1} = \begin{bmatrix} 64.818991 & -213.413104 & 129.627082 & -27.676674 \\ 57.510375 & -188.201279 & 115.638784 & -24.832401 \\ 38.381329 & -122.830831 & 79.495504 & -17.239467 \\ 14.134665 & -40.901684 & 33.388940 & -6.391421 \end{bmatrix} . \quad (24)$$

The multiplication shown in (14) gives the first four terms of the sequence with the remaining terms given by (16).

For two second order sections in series given by

$$H_1[z] = \frac{(5.78776100)10^{-2}\ (z+1)^2}{z^2 - 1.07350061\ z + 0.30805006} = \frac{N_1[z]}{D_1[z]} \quad (25)$$

and

$$H_2[z] = \frac{(7.99359506)10^{-2}(z+1)^2}{z^2 - 1.45996913\ z + 0.77971293} = \frac{N_2[z]}{D_2[z]} \quad (26)$$

the implementation of each section in direct form requires the following two calculations to compute the autovariance sequence at the output of $H_2[z]$.

First the autocovariance $R_{12}[kT]$ due to the noise in section one is found using the coefficients in the transfer function

$$H_{12}[z] = \frac{N_2[z]}{D_1[z]\ D_2[z]} \quad . \quad (27)$$

Thus the autovariance sequence shown in Fig. 2 is modified by the zeros of $H_2[z]$. From (20) this is given by

$$R_{12}[kT] = \bar{a}^t\ C\ \bar{a} \quad . \quad (28)$$

The autocovariance $R_{22}[kT]$ due to the noise in section two using the coefficients from

$$H_{22}[z] = \frac{1}{D_2[z]} \quad . \tag{29}$$

The steady state autovariance of the filter is now given by

$$R_y[kT] = R_{12}[kT] + R_{22}[k] \tag{30}$$

since the product rounding errors are assumed to be uncorrelated [1]. A plot of $R_y[kT]$ is shown in Fig. 3 as a function of k. This illustrates the well-known result that implementation of high order filters as a series or parallel combination of first and second order sections can reduce the effect of product rounding errors.

CONCLUSIONS

A method for computing the autocovariance at the output of a digital filter has been presented. For the steady state case, the inversion of an N by N matrix is required where N is the order of the system from the noise source to the filter output. The remaining terms are computed using simple nonrecursive expressions involving the filter coefficients. The procedure is suitable for programming on a computer or on one of the programmable scientific calculators that now exist in many design facilities.

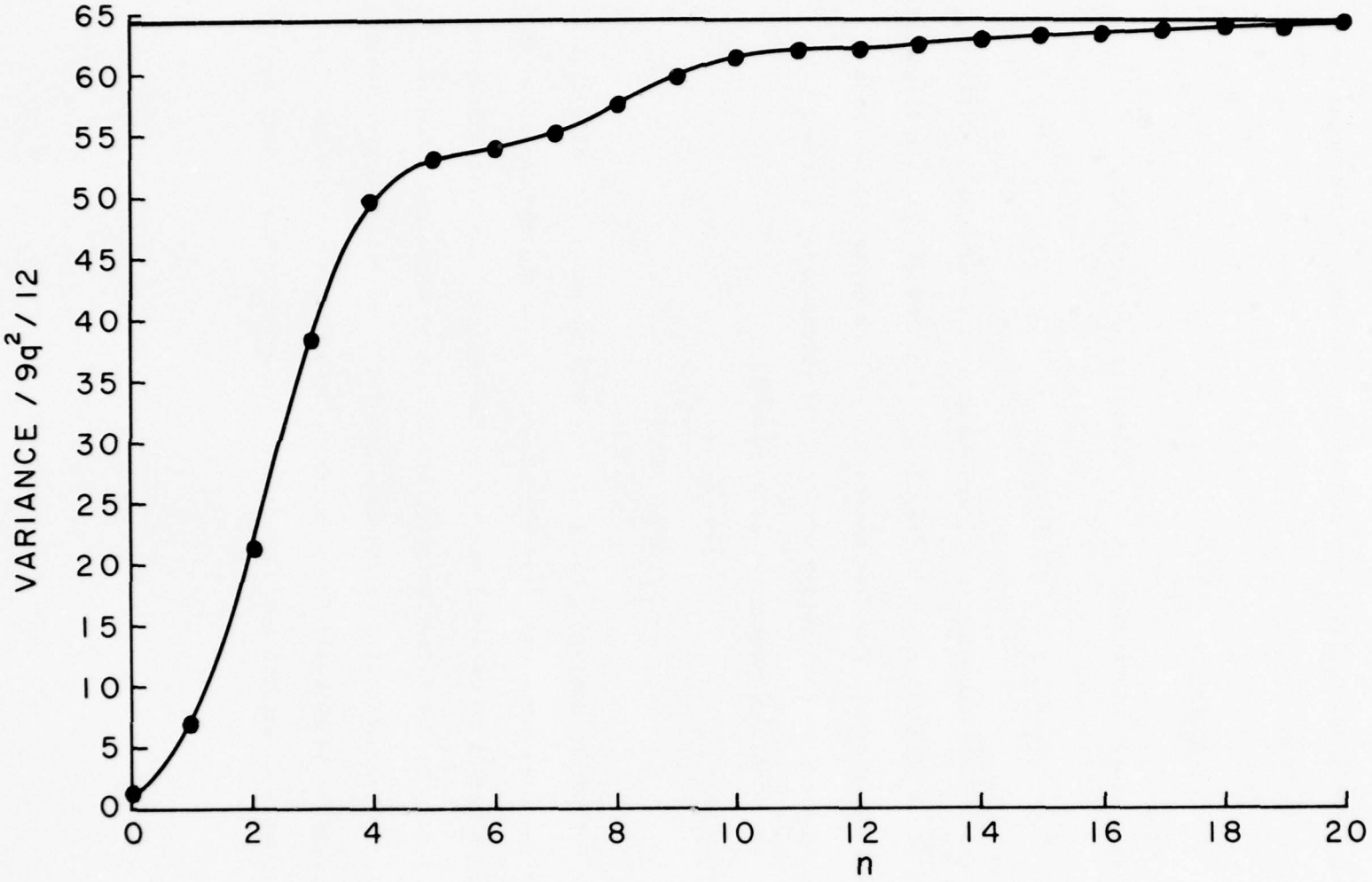


FIG. 1 Variance of y_n plotted as a function of n for a fourth order Butterworth filter implemented in direct form.

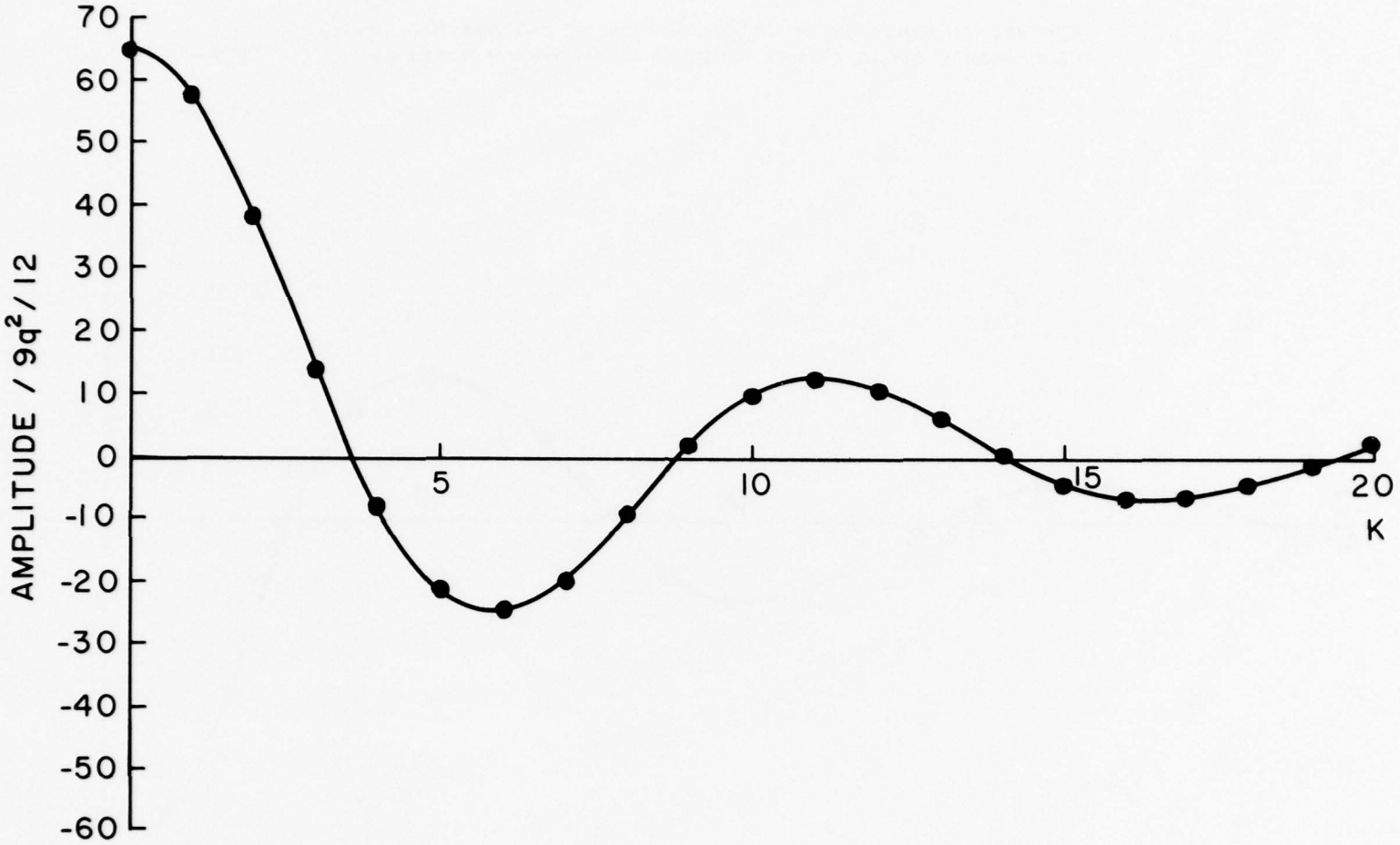


FIG. 2 Autocovariance sequence for a fourth order Butterworth filter implemented in direct form

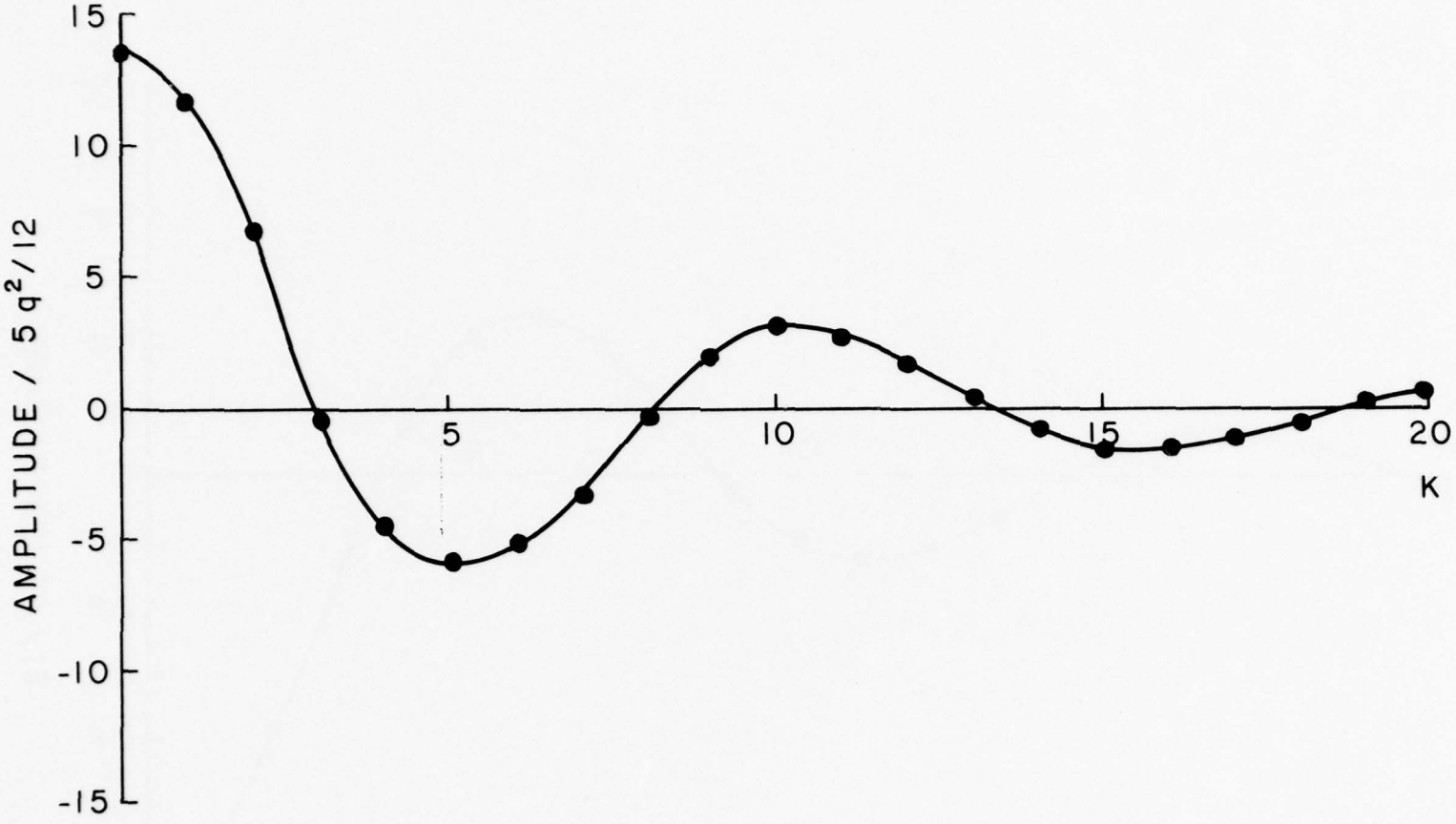


FIG. 3 Steady state autocovariance for a fourth order Butterworth filter implemented as two second order sections in cascade

REFERENCES

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

2. E. Jury, Theory and Application of the z-Transform Method, John Wiley and Sons, 1964.

3. J. L. Melsa and A. P. Sage, An Introduction to Probability and Stochastic Processes, Prentice Hall, 1972.

APPENDIX D

IMPLEMENTATION OF DIGITAL CONTROLLERS

by

Thomas A. Brubaker
Electrical Engineering Department
Colorado State University
Fort Collins, Colorado 80521

and

William Loendorf
Electrical Engineering Department
Colorado State University
Fort Collins, Colorado 80521

ABSTRACT

The variance of error at the output of a digital control system due to input data quantization and product rounding in a digital controller is derived. Two forms of controller implementations are considered. Then an expression for the word length that yields a specified variance of error is determined. Scaling is then discussed and an example consisting of a discrete integral plus proportional controller is presented.

INTRODUCTION

The use of discrete data processing algorithms as an integral part of a control system is well known. Books by Tou [1], Monroe [2] and Kuo [3] describe many of the design concepts for these systems called sampled data or digital control systems. In a somewhat independent effort design and implementation procedures have been developed for digital filters. A collection of papers in this area has recently been edited by Rabiner and Rader [4]. A tutorial paper that provides an overview of errors in the implementation of digital filters has been written by Liu [5].

The difference between a digital filter and a digital control system is in the system structure. A digital filter is inherently an open loop signal processing component that is described entirely by a difference equation. In a digital control system, the discrete control algorithm, which is a digital filter, is a component in a feedback loop that also usually contains analog elements. Analysis of the mixed system then requires the use of artificial sampling elements within the system.

Currently, discrete control algorithms are often implemented using a digital computer that is programmed to handle one or more control paths. However, for some applications, the real-time speed requirements can tax the capacity of a computer. Furthermore, the real-time programming of many parallel control algorithms is costly and difficult to manage.

As an alternative, the recent development of small microprogrammable processors now permits a different design philosophy for digital control systems. The use of these processors to implement discrete control algorithms

as independent units or under control of a central computer leads to a distributed computing structure that is well suited for use in a variety of control systems. In a typical system, each microprogrammable processor is initially programmed from a central computer to perform a system function. If operating conditions change the processors can be reprogrammed to meet new requirements.

In the implementation of any discrete control algorithm, the designer should consider the effect of errors due to finite length arithmetic. The errors that are introduced are caused by the quantization of the input data, quantization of the algorithm coefficients and the quantization of products. When floating point hardware is used, sums and differences also introduce errors. For satisfactory system operation, the algorithm should be implemented in a form that minimizes the effect of these errors and the word length needed for the implementation should be determined.

Quantization has been studied by a number of researchers. Bennett [6] and Widrow [7] have shown that for a rounding form of quantization, the errors can be modeled as independent random variables with mean zero and variance $q^2/12$. Here q represents the value of the least significant bit in the computer word. This model has been experimentally verified for input signals that transverse several quantization levels from sample to sample. The truncation form of quantization is similar to rounding except that the mean value for each error is $q/2$.

A method for finding the maximum upper bound on the error in a digital control system due to quantization was introduced by Bertram [8]. Slaughter [9] determined an upper bound on the steady-state system error

by replacing each error source by a step function equal to the maximum value of the error source. The steady state error is then evaluated by applying the final value theorem from z-transforms. Knowles and Edwards [10, 11] assumed random quantization errors and developed statistical upper bounds for the error at the output of the system.

In this paper, the variance of the error at the system output due to rounding of the input data and the intermediate products in a digital control algorithm is evaluated for two common algorithm implementations. A unity feedback is assumed, however, the procedure can be easily extended. Fixed point arithmetic is also assumed since this form of arithmetic is common in small microprocessors. Because scaling is also a consideration, scaling procedures are described and their effect is discussed. Finally, an example consisting of a discrete form of an integral plus proportional controller is investigated and the word length needed for a given variance is compared to that obtained using methods described in reference [9].

EVALUATION OF THE ERROR VARIANCE IN DIGITAL CONTROL SYSTEMS

A block diagram for a unity feedback digital control system is shown in Fig. 1. Here the input signal $x(t)$ and the return signal from the plant are assumed to be analog. These signals are converted into digital form and subtracted to form the discrete error signal $e[nT]$. This signal serves as the input to the digital controller with transfer function

$$C(z) = \frac{\sum_{k=0}^{M-1} a_k z^{-k}}{1 + \sum_{j=1}^{R} b_j z^{-j}} = \frac{C_N[z]}{C_D(z)} \quad . \tag{1}$$

Errors in the implementation of (1) are due to rounding of the analog data by the analog-to-digital converter and the rounding of products. Coefficient quantization errors are deterministic and under control of the designer. These errors will not be considered here. For simplicity, the least significant quantization level will be taken as $\pm q/2$ for all errors. This is not a serious restriction and different quantization levels can be used. Each rounding error is also assumed to be an independent random variable with mean zero and variance $q^2/12$.

Quantization of the Input Data

For the system shown in Fig. 1, two input data quantization errors occur. These errors are represented by noise sources shown in the block diagram of Fig. 2. The error at the system output due to the two errors at discrete times $t=nT$ is given by

$$e_{y1}[nT] = \sum_{k=0}^{n} h[kT] \{e_1[(n-k)T] - e_2[(n-k)T]\} \tag{2}$$

where $h[kT]$ is the system weighting sequence given by the inverse z transform of the system transfer function

$$H[z] = \frac{C[z]\, G[z]}{1 + C[z]\, G[z]} \quad . \tag{3}$$

In (3), $G[z]$ is the z transform equivalent of the data reconstruction element and the fixed plant. If $e_1[kT]$ and $e_2[kT]$ are independent

random variables, the variance at the system output at time $t=nT$ is

$$\text{Var}\ \{e_{y1}[nT]\} = \frac{2q^2}{12} \sum_{k=0}^{n} h^2[kT] \quad . \tag{4}$$

The expression given by (4) is monotonically increasing as n increases and approaches steady state as n approaches infinity. The steady state variance is also given by the discrete Wiener-Khinchine relationship

$$\text{Var}\ \{e_{y1}[\infty]\} = \left(\frac{1}{2\pi j} \oint H[z]\ H[z^{-1}]\ z^{-1}\ dz\right) \frac{2q}{12} \quad . \tag{5}$$

The evaluation of the steady state variance and/or the steady-state auto-variance sequence is easily accomplished using the procedure developed in Appendix 1.

Product Rounding Errors

Product rounding errors occur within the digital control algorithm and their effect on the system depends on the form of implementation. Typical forms are the direct form one and direct form two. To minimize product rounding errors, high order algorithms are usually implemented as a cascade or parallel connection of first and second order sections [5]. The error analysis for these cases follows the general results presented in this paper and will not be considered further.

The block diagram for a third order algorithm implemented in direct form one is shown in Fig. 3 where each multiply operation consists of a rounded product plus a noise component. For this form of implementation, the transfer function from each error source to the controller output is

given by

$$C_1[z] = \frac{1}{C_D[z]} \quad . \tag{6}$$

For an Nth order algorithm described by (1) the resulting error at the output of the control system for a direct form one implementation is given by

$$e_{y2}[nT] = \sum_{k=0}^{n} h_1[kT]\, p[(n-k)T] \tag{7}$$

where

$$p[(n-k)T] = \sum_{k=0}^{M-1} \varepsilon_k[(n-k)T] + \sum_{j=1}^{R} \delta_j[(n-k)T] \tag{8}$$

and $h_1[kT]$ is the inverse z transform of

$$H_1[z] = \frac{C_1[z]\, G[z]}{1 + C[z]\, G[z]} \quad . \tag{9}$$

The variance of the error for independent product rounding errors is

$$\mathrm{Var}\{e_{y2}[nT]\} = (M+R)\, \frac{q^2}{12} \sum_{k=0}^{n} h_1^{\,2}[kT] \quad . \tag{10}$$

The variance of the total error due to input data quantization and product rounding for the digital control algorithm implemented in direct form one is now given by

$$\mathrm{Var}\{e_{yD1}[nT]\} = \frac{q^2}{12} \left\{ 2 \sum_{k=0}^{n} h^2[kT] + (M+R) \sum_{k=0}^{N} h_1^{\,2}[kT] \right\} \quad . \tag{11}$$

The other common form of implementation for a digital controller is the direct form two shown in Fig. 4 for a third order system. Here, the errors generated by multiplying with the denominator coefficients can be

considered as input errors to the digital controller. The errors due to the a_i coefficients are simply additive at the controller output. The resulting variance of error at the system output due to input data quantization and product rounding for a direct form two implementation is

$$\mathrm{Var}\{e_{yD2}[nT]\} = \frac{q^2}{12}\left\{2\sum_{k=0}^{n} h^2[kT] + R\sum_{k=0}^{n} h^2[kT] + (M)\sum_{k=0}^{n} h_2{}^2[kT]\right\} \quad (12)$$

where $h_2[kT]$ is the inverse z transform of

$$H_2[z] = \frac{G[z]}{1 + C[z]\,G[z]} \quad (13)$$

WORD LENGTH REQUIREMENTS USING THE VARIANCE OF ERROR AT THE SYSTEM OUTPUT

For a given analog signal range of $\pm$ A, the level of least quantization referred to the algorithm input is given by the inequality

$$q \geq \frac{A}{2^N - 1} \quad . \quad (14)$$

In (14), N represents the number of magnitude bits that are combined with a sign bit within the computer. The inequality is used because the least level of quantization is often chosen so the range of the scaled binary variables slightly exceeds the value of A. For example, with a q of 10 mv and a ten bit and sign word length, the maximum value of the binary variables in terms of signal units is 10.23 volts. Carefully note that q is given in terms of the analog range at the system input.

Given a specification on the variance of error at the system output, the word length that allows the specification to be met is found by substituting (14) into (11) or (12) and solving the resulting inequality for N. For a specified steady state variance σ_s^2, the word length requirement to meet or exceed the specification is found by substituting (14) into (11) or (12) with $\mathrm{Var}\{e_{yD1}[\infty]\} = \sigma_s^2$ or $\mathrm{Var}\{e_{yD2}[\infty]\} = \sigma_s^2$. This gives the word length for a direct form one implementation of the digital control as

$$N_{D1} \geq \log_2 \left\{ \left(\frac{A^2}{12\,\sigma_s^2} \left[2 \sum_{k=0}^{\infty} h^2[kT] + (M+R) \sum_{k=0}^{\infty} h_1^2[kT] \right] \right)^{1/2} + 1 \right\} . \qquad (15)$$

For the direct form two we have

$$N_{D2} \geq \log_2 \left\{ \left(\frac{A^2}{12\,\sigma_s^2} \left[2 \sum_{k=0}^{\infty} h^2[kT] + R \sum_{k=0}^{\infty} h^2[kT] + (M) \sum_{k=0}^{\infty} h_2^2[kT] \right] \right)^{1/2} + 1 \right\} . \qquad (16)$$

In (15) and (16) N_{D1} and N_{D2} are the smallest integers that satisfy the inequalities. In addition a sign bit is normally required.

SCALING

Along with word length considerations for input data quantization and product rounding, the designer of a digital control algorithm must consider scaling. For the algorithm to operate properly, the output $r(nT)$ as shown in Fig. 1, cannot overflow for any given input signal $x(t)$. If this happens, the control loop is effectively open and the system will not function in a desired manner.

For the common case of fixed point two's complement arithmetic, Jackson, Kaiser and McDonald [11] have shown that intermediate overflow at the output of any multiplier or adder will not alter the algorithm performance as long as the algorithm response does not overflow. This means that for the filter form shown in Fig. 3 the response $r(nT)$ is the only variable that must be scaled. For the direct form two, both $r(nT)$ and $r_1(nT)$ must be scaled. Scaling for these implementations has been investigated by Jackson [12] who used Holders inequality to establish steady state bounds for various digital filter forcing functions. However, these bounds do not include the effect of transients.

In practice, analytical methods for determining bounds on $r(nT)$ often gives pessimistic results and the use of simulation for a given class of input signals can be useful. Once the maximum value or a bound for $r(nT)$ and/or $r_1(nT)$ is determined the actual scaling is not difficult. For the direct form one implementation shown in Fig. 3, each a_i coefficient is reduced in magnitude by a common factor. This effectively reduces the gain of the algorithm. For a direct form two implementation, the input signal $e(nT)$ is multiplied by a scale factor so that the values of $r_1(nT)$ do not overflow. The a_i coefficients are reduced in magnitude to prevent $r(nT)$ from overflowing. Thus, again the scaling is accomplished by effectively reducing the filter gain. This gain may be reintroduced at other points in the system if desirable. The error analysis for the scaled filters or control algorithms follows from previous results.

Another important alternative is the use of an adaptive scale factor that is utilized only when an overflow is detected in the system. If this occurs and if sufficient computational time is available the filter can be automatically scaled until overflow ceases to exist.

EXAMPLE

To illustrate the procedure consider the system shown in Fig. 1 when C(z) is a discrete integral plus proportional controller

$$D(z) = \frac{\left(k_i\ T/2 + k_p\right)z + k_i\ T/2 - k_p}{z - 1} \tag{17}$$

and the plant is given by $G(s) = 2/s+5$. The sampling interval was chosen as .01 and the constants were first assigned values $K_i = .75$ and $K_p = .25$. This results in a sluggish system with no overshoot for a step input. For a step input the maximum value of $e(nT)$ is one and the maximum value of the controller output $r(nT)$ is 5/2. To obtain a controller with a gain of one, a multiplying factor of 5/2 is placed in the data reconstruction element. The resulting z transforms are

$$G'(z) = \frac{5}{2}\ \frac{1-e^{-jsT}}{s}\ \frac{2}{s+5} = \frac{.04877}{z-.95123} \tag{18}$$

and

$$D'(z) = \frac{.1015\ z - .0985}{z - 1} \quad . \tag{19}$$

Plots of $e(nT)$, $r(nT)$ and $y(t)$ are shown for the scaled system with $x(t) = \mu(t)$ in Fig. 5.

For a direct form one implementation shown in Fig. 6 there are two input rounding errors. The two product rounding errors are introduced between the numerator and denominator portions of D'(z). If the errors are each assumed to be step functions with magnitude q/2, the method developed by Slaughter [9]] results in a steady state error at the system output

$$E_{ss} = \lim_{n\to\infty} e[nT] = \left\{\left[\frac{q\ z\ G'(z)\ D'(z)}{1 + G'(z)\ D'(z)}\right] + \left[\frac{q\ z\ \frac{G'(z)}{z-1}}{1 + G'(z)\ D'(z)}\right]\right\}_{z=1} , \qquad (20)$$

which gives upon substitution

$$E_{ss} = (334.33)q \quad . \qquad (21)$$

For a direct form two implementation there are two input rounding errors. The two product roundings are introduced at the output of D'(z). The steady state error is given by

$$E_{ss} = \lim_{n\to\infty} e[nT] = \left\{\frac{q\ z\ G'(z)\ D'(z)}{1 + G'(z)\ D'(z)} + \frac{q\ z\ G'(z)}{1 + G'(z)\ D'(z)}\right\}_{z=1} \qquad (22)$$

which yields

$$E_{ss} = q \quad . \qquad (23)$$

Substituting (14) into (21) and (23) now gives an expression for the word length in terms of E_{ss}/A as

$$N_{df1} \geq \log_2 \left\{\frac{(334.33)}{(E_{ss}/A)} + 1\right\} \qquad (24)$$

for the direct form one implementation and

$$N_{df2} \geq \log_2 \left\{\frac{1}{(E_{ss}/A)} + 1\right\} \qquad (25)$$

for the direct form two implementation. Note that the variable A refers to the maximum value of the variable e(nT) for a controller with a gain of one and no overflow. Values of N are tabulated in Table 1.

If the rounding errors are modeled as independent random variables with mean zero and variance $q^2/12$, the steady state variance of the error at the system output for a direct form one implementation is given by

$$\text{Var}(e) = \frac{2q^2}{12} \left\{ \sum_{k=0}^{\infty} h^2(kT) + \sum_{k=0}^{\infty} h_1^{\,2}(kT) \right\} \tag{26}$$

where $h(kT)$ is the inverse z transform of

$$H'(z) = \frac{D'(z)\, G'(z)}{1 + D'(z)\, G'(z)} \tag{27}$$

and $h_1(kT)$ is the inverse z transfer

$$H_1'(z) = \frac{G'(z)/(z-1)}{1 + G'(z)\, D'(z)} \quad . \tag{28}$$

Substituting gives

$$\text{Var}(e) = (25.3)q^2 \quad . \tag{29}$$

For the direct form two implementation

$$\text{Var}(e) = (4.07 \times 10^{-3})q^2 \quad . \tag{30}$$

To establish word length requirements the variable e/A is now assumed to be a Gaussian random variable. The probability that e/A is between $\pm E_{ss}$, where E_{ss} is the previously given steady state error, is now defined as

$$\text{Prob} \left\{ -\frac{E_{ss}}{A} \leq \frac{e}{A} \leq \frac{E_{ss}}{A} \right\} = \text{confidence level.} \tag{31}$$

For a given confidence level, normalized distribution tables can be used by writing

$$\text{Prob}\left\{-\frac{E_{ss}}{A\sigma} \leq \frac{e}{A\sigma} \leq \frac{E_{ss}}{A\sigma}\right\} = \text{confidence level}$$

where

$$\sigma = \left\{\text{Var}\left(\frac{e}{A}\right)\right\} \geq \begin{cases} \left(\dfrac{25.3}{2^N - 1}\right)^{1/2} & \text{direct form one} \\ \left(\dfrac{4.07 \times 10^{-3}}{2^N - 1}\right)^{1/2} & \text{direct form two} \end{cases} \tag{32}$$

For a 95% confidence level

$$\frac{E_{ss}}{A\sigma} = 2 \tag{33}$$

and substitution of (32) into (33) now gives

$$N_{df1} \geq \log_2 \left\{\frac{10}{E_{ss}/A} + 1\right\} \tag{34}$$

for the direct form one. For the direct form two

$$N_{df2} \geq \log_2 \left\{\frac{.128}{E_{ss}/A} + 1\right\} \quad . \tag{35}$$

The value of N_{df1} and N_{df2} are also tabulated in Table 1.

Referring to Table 1, the form that is used to implement the controller is obviously a critical factor. The use of the variance of error results in a somewhat shorter word length for a given value of E_{ss}/A. However, in practice a very short word length will give rise to other undesirable effects due to large deadbands caused by the rounding. Therefore, for the

example, a word length of ten bits plus sign might be utilized. This gives an error E_{ss}/A less than 10^{-3} for the controller implemented in direct form two.

To improve the response a new control algorithm was designed using $K_p = 3$ and $K_i = 15$. The resulting discrete controller transfer function is

$$D'[z] = \frac{1.23\ z - 1.18}{z - 1} \quad . \tag{36}$$

Simulation, however, shows the controller output will overflow so a value of 1.23 was factored from (36) giving

$$D''[z] = \frac{z - .9593495935}{z - 1} \tag{37}$$

and

$$G''[z] = \frac{.0599878078}{z - .9512294245} \quad . \tag{38}$$

Response curves for the resulting system are shown in Fig. 6. Note that the steady state output for $r(nT)$ is about .8 units which means the full range of the digital controller is not being utilized fully. Word length requirements for the system are given in Table 2. For the direct form one the improved system requires fewer bits. While this result may not be expected careful examination of the system structure shows the new system to be less critical to noise generated within the system when the controller is implemented in direct form one.

E_{ss}/A	Word Length Using Slaughters Method		Word Length Using The Variance of Error and a 95% Confidence Level	
	N_{df1}	N_{df2}	N_{df1}	N_{df2}
10	6	1	2	1
1	9	1	4	1
10^{-1}	12	4	7	2
10^{-2}	15	7	10	4
10^{-3}	19	10	14	7
10^{-4}	22	14	17	11
10^{-5}	26	17	20	14
10^{-6}	29	20	24	17

TABLE 1

TABLE OF WORD LENGTH REQUIREMENTS
$K_i = .75$ $K_p = .25$

E_{ss}/A	Word Length Using Slaughters Method		Word Length Using The Variance of Error and a 95% Confidence Level	
	N_{df1}	N_{df2}	N_{df1}	N_{df2}
10	2	1	1	1
1	4	1	2	1
10^{-1}	8	4	4	2
10^{-2}	11	7	8	5
10^{-3}	14	10	10	8
10^{-4}	18	14	14	8
10^{-5}	21	17	18	14
10^{-6}	24	20	21	18

TABLE 2

TABLE OF WORD LENGTH REQUIREMENTS
$K_i = 15 \quad K_p = 3$

APPENDIX A

In this section a closed form method for computing the discrete autocovariance function is derived. Consider a discrete transfer function

$$H[z] = \frac{\sum_{k=0}^{M} a_k z^{-k}}{1 - \sum_{j=1}^{N} b_j z^{-j}} = H_1[z] \; H_2[z] \tag{1a}$$

where

$$H_1[z] = \frac{1}{1 - \sum_{j=1}^{N} b_j z^{-j}} \tag{2a}$$

and

$$H_2[z] = \sum_{k=0}^{M} a_k z^{-k} \quad . \tag{3a}$$

If an independent noise sequence $\{e[jT]\}$ with mean zero and variance σ^2 is applied to $H_1[z]$ the difference equation representation is given by

$$x[nT] = \sum_{j=1}^{N} b_j \, x[(n-j)T] + e[nT] \quad . \tag{4a}$$

The discrete autovariance at time $t = nT$ is now defined as

$$R_n[kT] = E\{x[nT] \; x[(n-k)T]\} \quad k = 0, 1, 2, \ldots n \quad . \tag{5a}$$

Since $e[nT]$ is taken from an independent noise sequence $x[(n-k)T]$ and $e[nT]$ are uncorrelated. Multiplying (4a) by $x[(n-k)T]$, $k = 0, 1, 2, \ldots N-1$ and taking the expectation now results in a set of

coupled difference equations in the variables $\{R_n[0T], R_n[T] \; R_n[(N-1)T]\}$. Taking the z transform and applying the final value theorm now permits the first N terms of the steady state autovariance sequence to be written in matrix form as

$$\begin{bmatrix} R[0T] \\ R[T] \\ \cdot \\ \cdot \\ \cdot \\ R[(n-1)T] \end{bmatrix} = M^{-1} \begin{bmatrix} \sigma^2 \\ 0 \\ 0 \\ 0 \\ \\ 0 \end{bmatrix} \tag{6a}$$

where M is the matrix

$$M = \begin{bmatrix} 1 - \sum_{j=1}^{N} b_j^{\,2} & -2 \sum_{k=1}^{N-1} b_j \, b_{j+1} & -2 \sum_{j=1}^{N-2} b_j \, b_{j+2} & \cdot \; \cdot & -2b_1 \, b_N \\ (b_1) & (1 + b_2) & \cdot & & b_N \\ (b_2) & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ (b_N) & (b_{N-2} \; b_N) & \cdot & & 1 \end{bmatrix} \tag{7a}$$

The remaining terms in the autovariance sequence are found from

$$R[kT] = - \sum_{j=1}^{N} b_j \, R[(k-j)T] \quad k \geq N \; . \tag{8a}$$

The steady state autovariance at the output of $H[z]$ is given by

$$\lim_{n \to \infty} E\{y[nT] \; y[(n-k)T]\} = R_y[kT] = \bar{a}^{-t} \, c \, \bar{a} \tag{9a}$$

where $\bar{a}^t = [a_0, a_1 \cdots a_M]$ and c is a constant matrix with terms

$$c_{ij} = R[(k-j+i)T] = \lim_{n\to\infty} E[x[nT]\, x[n-(k-j+i)T] \quad . \tag{10a}$$

When computing the steady state autovariance for $k < M-1$ certain terms in the C matrix are found using the relationship $R[kT] = R[-kT)$. Thus, a fixed nonrecursive expression only exists for $k \geq M-1$.

The variance at the output of the system is found using the results of (6a). The expression at the output is

$$\lim E[y_n^2] = \bar{a}^t\, C_1\, \bar{a} \tag{11a}$$

where C_1 is the autovariance matrix

$$\begin{bmatrix} R[0T] & R[T] & . & R[MT] \\ R[T] & . & . & . \\ . & . & . & . \\ R[MT] & . & . & R[0T] \end{bmatrix} .$$

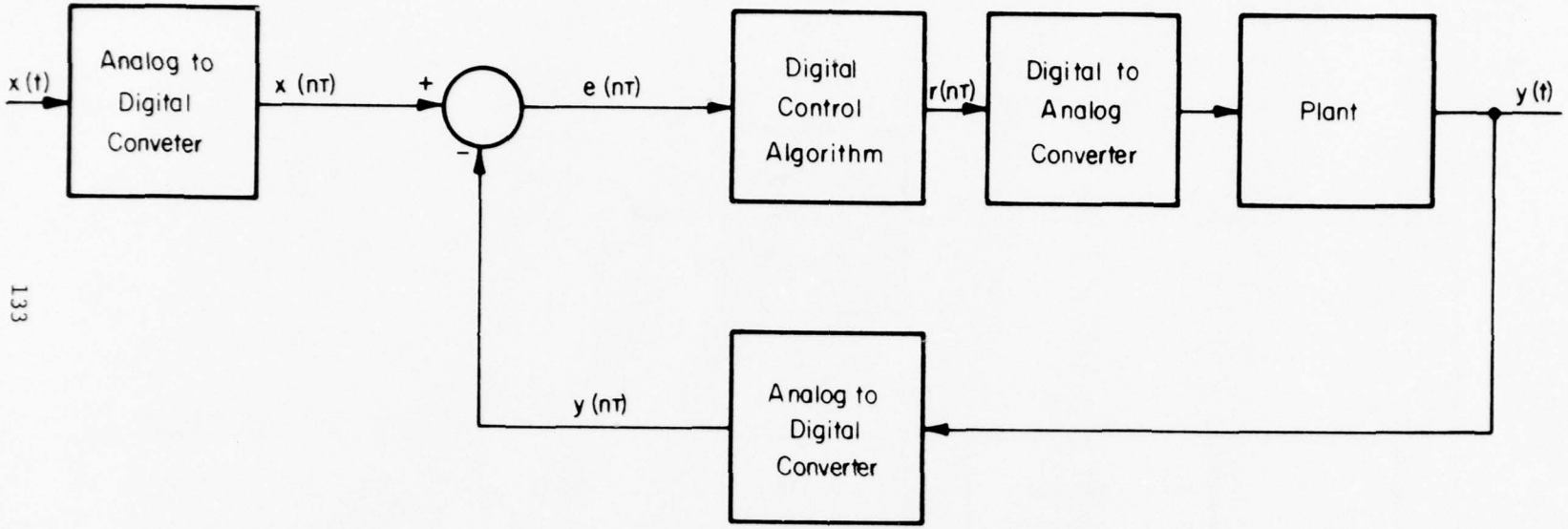


Figure 1 Block Diagram for a Unity Feedback Digital Control System

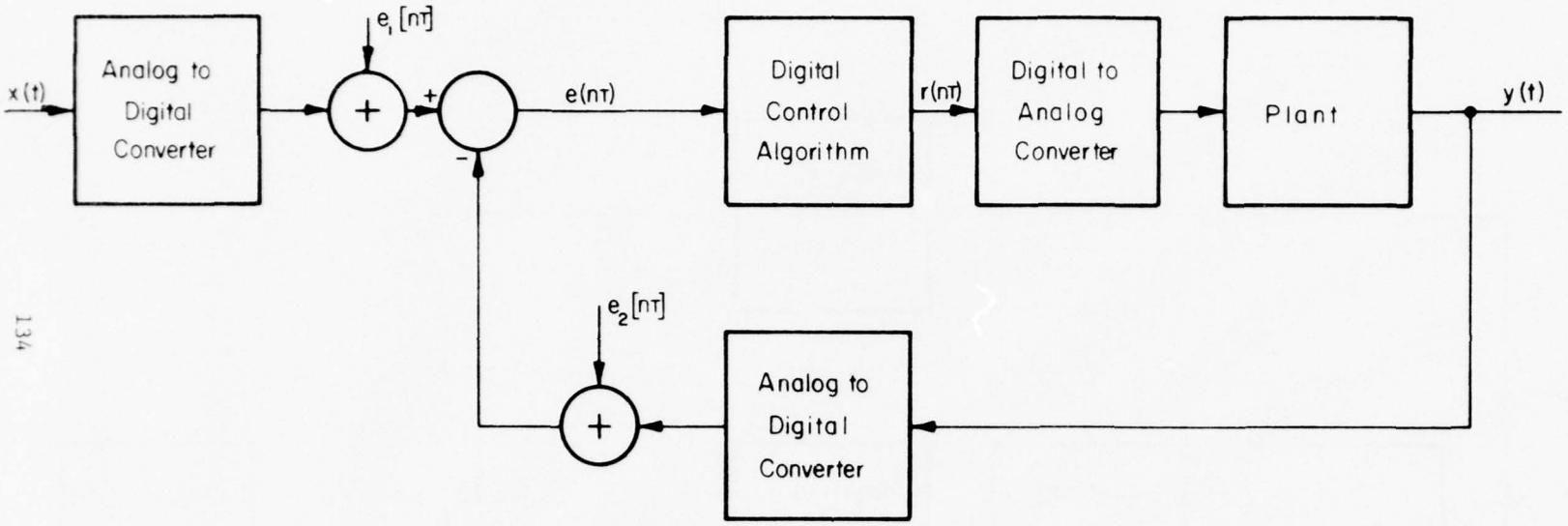


Figure 2 Block Diagram Showing Data Quantization Errors

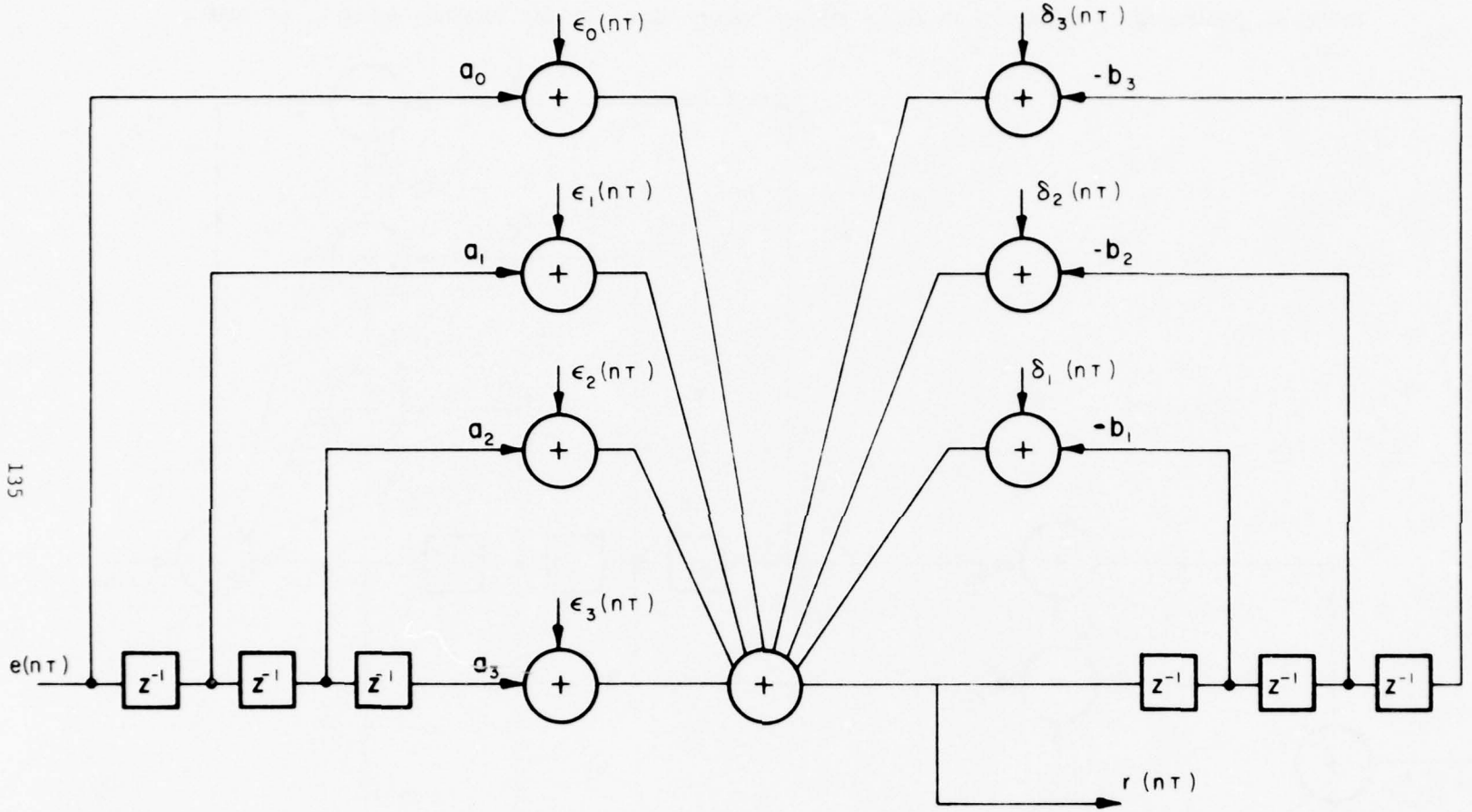


Figure 3 Block Diagram for a Third Order Digital Filter or Controller Implemented in Direct Form I

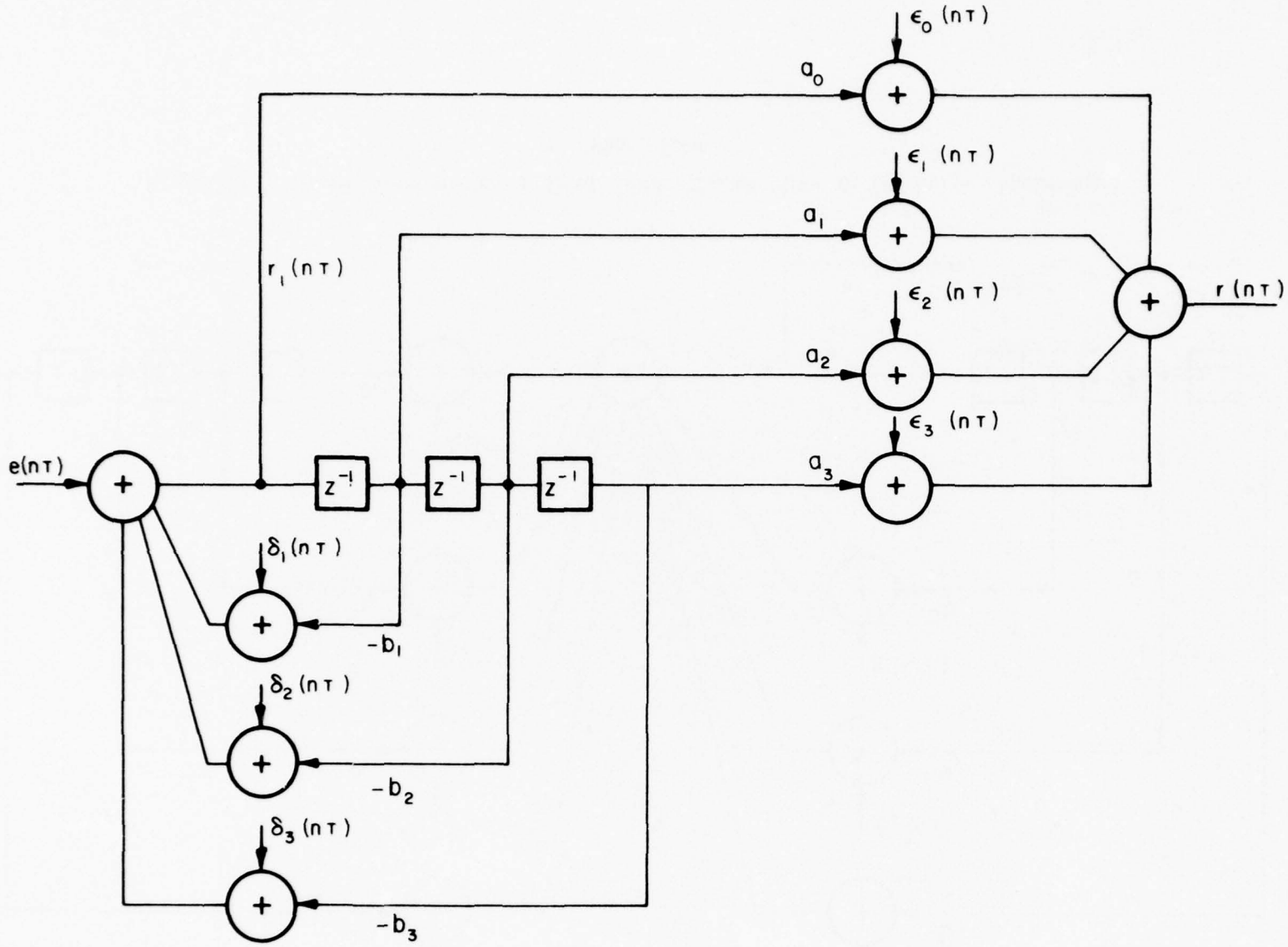


Figure 4 Block Diagram for a Third Order Digital Filter or Controller Implemented in Direct Form 2

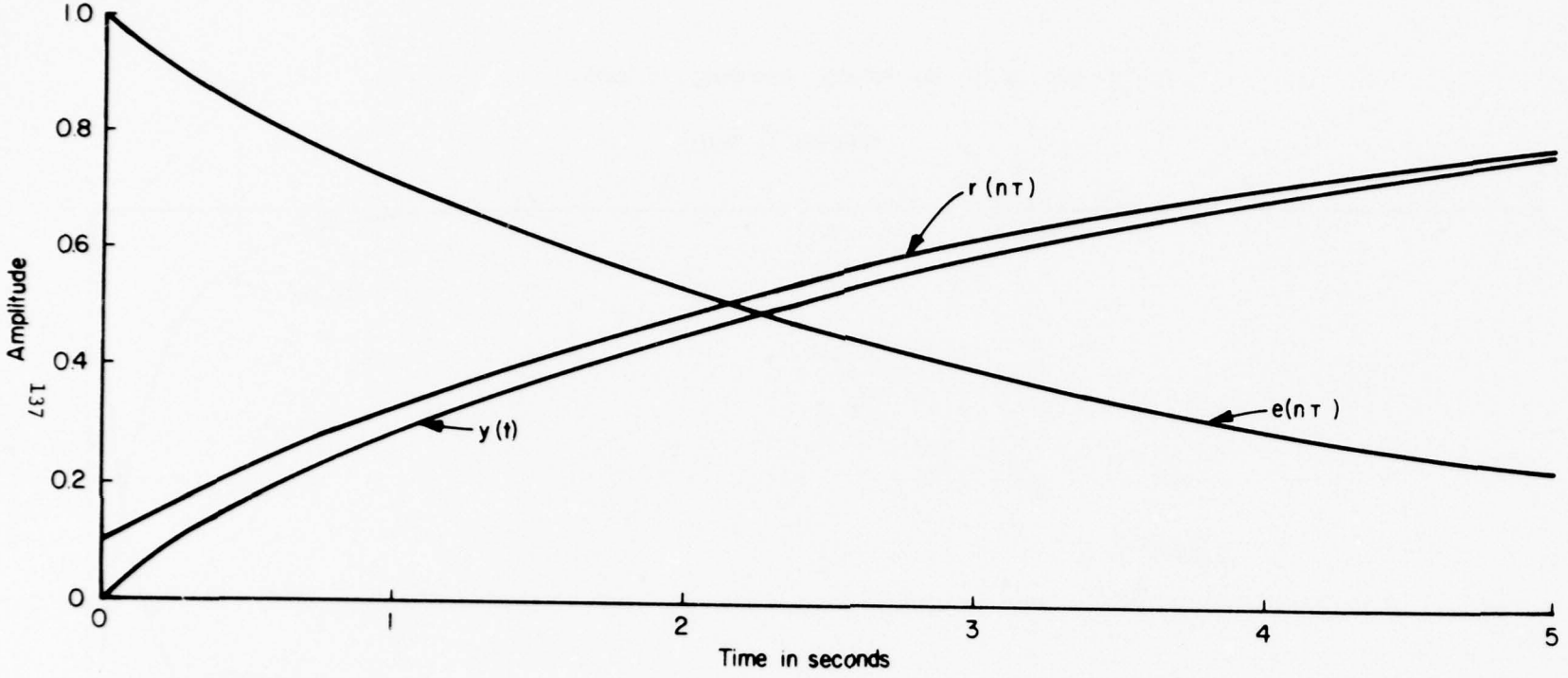


Figure 5 Response Curves for $K_i = 0.75$ and $K_p = 0.25$

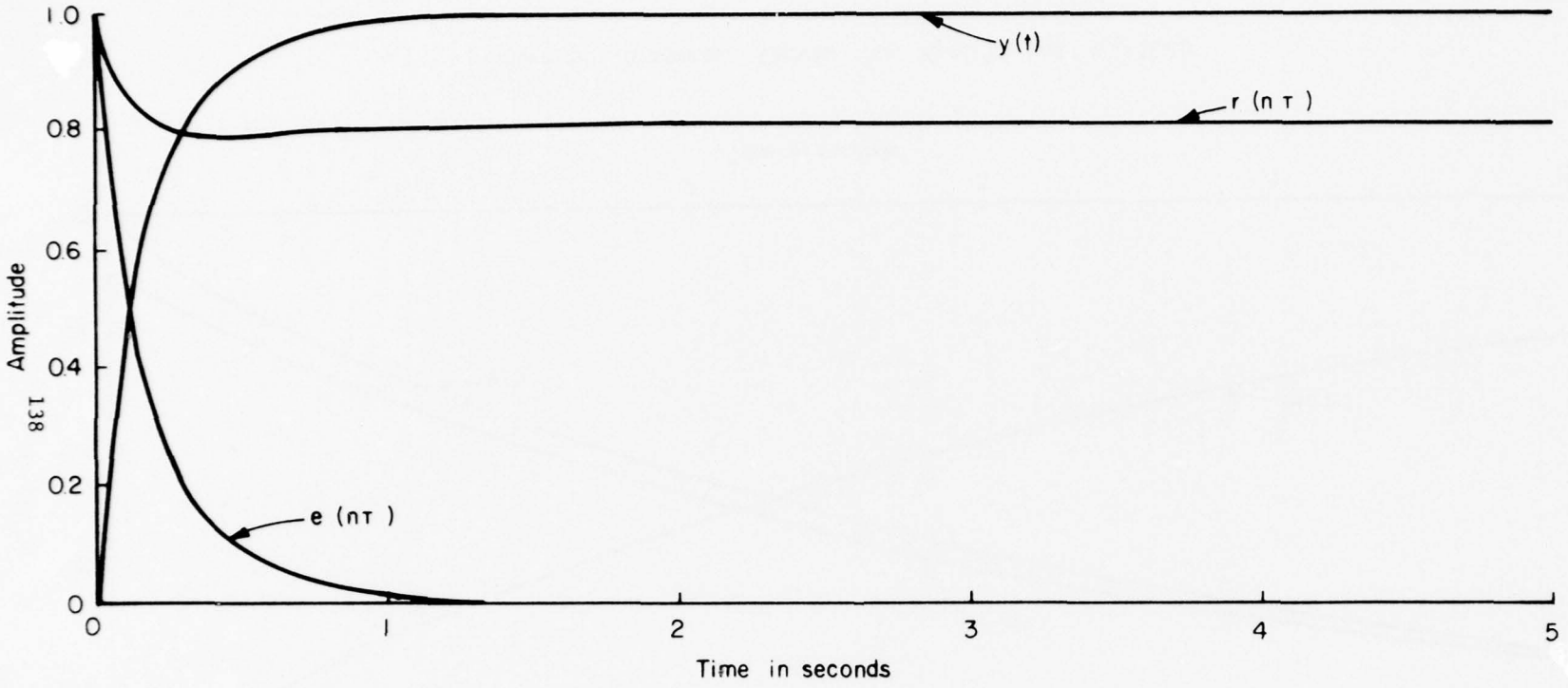


Figure 6 Response Curves for K_i=15 and K_p=3

REFERENCES

1. J. T. Tou, Digital and Sampled Data Control Systems, McGraw-Hill, 1959.

2. A. J. Monroe, Digital Processes for Sampled Data Systems, John Wiley and Sons, 1962.

3. B. C. Kuo, Discrete Data Control Systems, Prentice-Hall, 1970.

4. L. R. Rabinier and C. M. Rader, Digital Signal Processing, IEEE Press, 1972.

5. B. Liu, "Effects of Finite Word Length on the Accuracy of Digital Filters--A Review," IEEE Trans. on Circuit Theory, Vol. CT-18, November 1971.

6. W. R. Bennett, "Spectra of Quantized Signals," Bell System Technical Journal, Vol. 27, No. 3, 1948.

7. B. Widrow, "Statistical Analysis of Amplitude Quantized Sampled-Data Systems," AIEE Trans. Part 2, pp. 555-562, 1961.

8. J. E. Bertram, "The Effect of Quantization in Sampled-Data Feedback System," AIEE Transactions, Vol. 77, pp. 177-182, 1958.

9. J. B. Slaughter, "Quantization in Digital Control Systems, IEEE Trans. on Automatic Control, Vol. AC-9, pp. 70-74, 1964.

10. J. B. Knowles and R. Edwards, "Effect of a Finite Word Length Computer in a Sampled-Data Feedback System," Proceedings of the IEE, Vol. 112, No. 6, pp. 1197-1207, 1965.

11. J. B. Knowles and R. Edwards, "Computational Error Effects in a Direct Digital Control System," Automatica, Vol. 4, pp. 7-29, 1966.

12. L. B. Jackson, J. F. Kaiser and H. S. McDonald, "An Approach to the Implementation of Digital Filters," IEEE Trans. on Audio and Electroacoustics, AU-18, September 1968.

13. L. B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," Bell System Technical Journal, Vol. 49, February 1970.

APPENDIX E

A STRATEGY FOR COEFFICIENT QUANTIZATION IN DIGITAL CONTROL ALGORITHMS

by

Thomas A. Brubaker
Electrical Engineering Department
Colorado State University
Fort Collins, Colorado 80521

ABSTRACT

A strategy for quantizing the coefficients in a general second order digital control algorithm is presented. First, the errors in the magnitude and phase functions for the algorithm are derived in terms of the filter coefficients. By specifying a maximum allowable error for each function over a given frequency range,quantization regions can then be established. An example consisting of a lag-lead digital control algorithm is included to illustrate the procedure.

INTRODUCTION

The use of discrete data processing algorithms as an integral part of a control system is well known. Books to Tou [1], Monroe [2] and Kuo [3] describe many of the design concepts for these systems called sampled data or digital control systems. In a somewhat independent effort, design and implementation procedures have been developed for digital filters. A collection of papers in this area has recently been edited by Rabiner and Rader [4].

The accuracy of any digital control algorithm is dependent on the finite word length used to represent the input data, the filter coefficients and the intermediate products. A recent review paper on this topic that is concerned with open loop digital filters is by Liu [5].

Error due to deterministic coefficients in digital filters is described by Kaiser [6], Otnes and McNamee [7] and Mantey [8]. Gold and Rader [9,10] have investigated the effect of coefficient error on the filter pole locations and have developed filter forms that are less sensitive to error. Avenhaus [11] developed a procedure for finding the coefficient word length in a digital filter when rounding is employed. This procedure assumes an ideal magnitude function in the error formulation and results are developed only for the passband and stopband. A statistical approach to the problem is described by Knowles and Olcayto [12].

A general result from the references is that higher order digital filters and/or control algorithms should be implemented as a cascade or parallel connection of first and second order filter sections. When the filter sections have been determined, the filter coefficients must then be quantized in such a way that the specifications are satisfied.

The implementation of digital control algorithms with current digital hardware can also place constraints on the coefficient quantization. Using medium or large scale integrated circuits there are often constraints on the input and output pin connections. Computations done within the integrated circuit on the other hand can often utilize more bits. This means, for example, that sums of products can often be truncated only after all multiply-add operations have been completed. Since the coefficients represent a set of inputs to the control algorithm quantization should be carried out that allows a minimum number of bits to represent each coefficient.

When the control algorithm is implemented using fixed point arithmetic on a computer with a word length of n bits, it is desirable to quantize the coefficients to less than n bits. This allows flexibility in scaling when the programming is carried out.

In this paper a strategy for quantizing the coefficients in a second order control algorithm is developed. First, the errors in the magnitude and phase functions are expressed in terms of the filter coefficients. By comparing the derivatives of the magnitude and phase functions with respect to each coefficient, closed regions can be established for the quantization of the coefficients over a frequency range of interest. These regions are also useful when the coefficients are quantized using conventional rounding. The procedure is illustrated for a second order lag-lead compensator designed using the bilinear z transform.

ERROR ANALYSIS

Magnitude Function

Given a second order discrete transfer function

$$H(z) = \frac{a_0 z^2 + a_1 z + a_2}{z^2 + b_1 z + b_2} \tag{1}$$

the frequency function is given by letting $z = e^{j\omega T}$. The resulting magnitude function is written as

$$|H(\omega)| = \frac{A(\omega)}{B(\omega)} \tag{2}$$

where

$$A(\omega) = \left\{a_0^2 + a_1^2 + a_2^2 + (2a_0a_1 + 2a_1a_2)\cos\omega T + 2a_0a_2\cos 2\omega T\right\}^{\frac{1}{2}} \tag{3}$$

and

$$B(\omega) = \left\{1 + b_1^2 + b_2^2 + 2b_1(1 + b_2)\cos\omega T + 2b_2\cos\omega T\right\}^{\frac{1}{2}} . \tag{4}$$

When the coefficients are allowed to vary (2) can be considered as a function of five variables for any value of frequency. This leads to the differential approximation

$$\begin{aligned}\Delta|H(\omega)| = {} & \frac{\partial|H(\omega)|}{\partial b_1}\Delta b_1 + \frac{\partial|H(\omega)|}{\partial b_2}\Delta b_2 \\ & + \frac{\partial|H(\omega)|}{\partial a_0}\Delta a_0 + \frac{\partial|H(\omega)|}{\partial a_1}\Delta a_1 + \frac{\partial|H(\omega)|}{\partial a_2}\Delta a_2\end{aligned} \tag{5}$$

where each partial derivative is a function of frequency given by

$$\frac{\partial|H(\omega)|}{\partial b_1} = \frac{-A(\omega)\{b_1 + (1 + b_2)\cos\omega T\}}{[B(\omega)]^3} , \tag{6}$$

$$\frac{\partial |H(\omega)|}{\partial b_2} = \frac{- A(\omega)\{b_2 + b_1 \cos\omega T + \cos 2\omega T\}}{[B(\omega)]^3} \quad , \tag{7}$$

$$\frac{\partial |H(\omega)|}{\partial a_0} = \frac{a_0 + a_1 \cos\omega T + a_2 \cos 2\omega T}{A(\omega)\ B(\omega)} \quad , \tag{8}$$

$$\frac{\partial |H(\omega)|}{\partial a_1} = \frac{a_1 + (a_0 + a_2) \cos\omega T}{A(\omega)\ B(\omega)} \tag{9}$$

and

$$\frac{\partial |H(\omega)|}{\partial a_2} = \frac{a_2 + a_1 \cos\omega T + a_0 \cos 2\omega T}{A(\omega)\ B(\omega)} \quad . \tag{10}$$

In the above expressions $A(\omega)$ and $B(\omega)$ are given by (3) and (4).

Phase Function

Given the transfer function described by (1) the resulting phase function is

$$\theta(\omega) = \text{Tan}^{-1}\left\{\frac{a_0 \sin 2\omega T + a_1 \sin\omega T}{a_0 \cos 2\omega T + a_1 \cos\omega T + a_2}\right\} - \text{Tan}^{-1}\left\{\frac{\sin 2\omega T + b_1 \sin\omega T}{\cos 2\omega T + b_1 \cos\omega T + b_2}\right\} \quad . \tag{11}$$

For simplicity let

$$N_1(\omega) = a_0 \cos 2\omega T + a_1 \sin\omega T + a_2 \tag{12}$$

$$N_2(\omega) = a_0 \sin 2\omega T + a_1 \sin\omega T \tag{13}$$

$$D_1(\omega) = \cos 2\omega T + b_1 \cos\omega T + b_2 \tag{14}$$

and

$$D_2(\omega) = \sin 2\omega T + b_1 \sin\omega T \tag{15}$$

The differential phase approximation is now described by

$$\Delta\theta(\omega) = \frac{\partial\theta(\omega)}{\partial b_1}\Delta b_1 + \frac{\partial\theta(\omega)}{\partial b_2}\Delta b_2 + \frac{\partial\theta(\omega)}{\partial a_0}\Delta a_0 + \frac{\partial\theta}{\partial a_1}\Delta a_1 + \frac{\partial\theta}{\partial a_2}\Delta a_2 \tag{16}$$

where

$$\frac{\partial\theta(\omega)}{\partial b_1} = \frac{-\left\{D_1(\omega)\ \sin\omega T - D_2(\omega)\ \cos\omega T\right\}}{[D_1(\omega)]^2 + [D_2(\omega)]^2} \tag{17}$$

$$\frac{\partial\theta(\omega)}{\partial b_2} = \frac{D_2(\omega)}{[D_1(\omega)]^2 + [D_2(\omega)]^2}\ , \tag{18}$$

$$\frac{\partial\theta(\omega)}{\partial a_0} = \frac{N_1(\omega)\ \sin 2\omega T - N_2(\omega)\ \cos 2\omega T}{[N_1(\omega)]^2 + [N_2(\omega)]^2}\ , \tag{19}$$

$$\frac{\partial\theta(\omega)}{\partial a_1} = \frac{N_1(\omega)\ \sin\omega T - N_2(\omega)\ \sin\omega T}{[N_1(\omega)]^2 + [N_2(\omega)]^2} \tag{20}$$

and

$$\frac{\partial\theta(\omega)}{\partial a_2} = \frac{-\ N_2(\omega)}{[N_1(\omega)]^2 + [N_2(\omega)]^2} \tag{21}$$

Determination of Quantization Regions

The expressions described by (5) and (16) are linear algebraic equations in the variables Δb_1, Δb_2, Δa_0, Δa_1 and Δa_2 with coefficients that are functions of frequency. For specifications described by the bounds $-\varepsilon_1 \leq \Delta|H(\omega)| \leq \varepsilon_1$ and $-\varepsilon_2 \leq \Delta\theta(\omega) \leq \varepsilon_2$, (5) and (16) each form a five dimensional space and the strategy is to first find the two smallest

spaces by utilizing the proper combination of coefficients. The vector $\{\Delta b_1 \quad \Delta b_2 \quad \Delta a_0 \quad \Delta a_1 \quad \Delta a_2\}$ is then forced to be inside of each space by the proper quantization of each filter coefficient.

In practice, however, the quantization problem can often be treated using two dimensional regions. In the example that follows, one two dimensional region is sufficient for the specification on $\Delta|H(\omega)|$ and one region for the specification on $\Delta\theta(\omega)$. In other practical algorithms, the error in $\Delta H(\omega)$ and $\Delta\theta(\omega)$ can be distributed among the coefficients so that one and two dimension regions can be utilized for the actual quantization.

EXAMPLE

For an analog lag-lead compensator described by

$$C(s) = \frac{(s + .1)\ (s + 1)}{(s + .01)\ (s + 10)} \tag{22}$$

the equivalent discrete compensator designed using the extended bilinear z transform is given by

$$c(z) = \frac{0.8356618816z^2 - 1.626383584z + 0.7909250553}{z^2 - 1.592691562z + 0.592894916} \quad . \tag{23}$$

In the design, the sampling time was chosen to be $T = 0.05$.

Graphs of the partial derivatives described by (6) through (10) and (17) through (21) are shown in Figs. 1 through 5 as functions of frequency. Values for each partial derivative were computed over a range of zero to twenty radians, however, to illustrate the important features the frequency

range for each derivative was reduced. For higher frequencies each partial derivative asymptotically approaches zero.

Referring to the graphs the following assumptions can be made. First, equations (6) and (7) are approximately the same as are (17) and (18). Secondly, equations (8), (9) and (10) are approximately the same as are (19), (20) and (21). From computer printout, the approximations are good to three decimal places. This implies that (5) and (16) can be rewritten as

$$\Delta|H(\omega)| \approx \frac{\partial|H(\omega)|}{\partial a_0}\{\Delta a_0 + \Delta a_1 + \Delta a_2\} + \frac{\partial|H(\omega)|}{\partial b_1}\{\Delta b_1 + \Delta b_2\} \tag{24}$$

and

$$\Delta\theta(\omega) \approx \frac{\partial\theta(\omega)}{\partial a_0}\{\Delta a_0 + \Delta a_1 + \Delta a_2\} + \frac{\partial\theta(\omega)}{\partial b_1}\{\Delta b_1 + \Delta b_2\} \tag{25}$$

Quantization regions are now established by letting $\Delta x_1 = \{\Delta a_0 + \Delta a_1 + \Delta a_2\}$ and $\Delta x_2 = \{\Delta b_1 + \Delta b_2\}$ and specifying values of $\Delta|H(\omega)|$ and $\Delta\theta(\omega)$. For $\Delta|H(\omega)| = \pm 0.1$ the minimum quantization region using (24) is shown in Fig. 6. This region is established by substituting various values of $\partial|H(\omega)|/\partial a_0$ and $\partial|H(\omega)|/\partial b_1$ into (24) to find the set of lines that enclose a minimum area. For $\theta(\omega) = \pm 1$ degree the minimum region is shown in Fig. 7. The actual coefficient quantization is done so that Δx_1 and Δx_2 lie inside of each quantization region. The quantized coefficients and quantization errors are shown in Table 1. Note that the quantization

was done to yield coefficients that can be represented by a minimum number of binary bits. The binary two's complement representation for each coefficient is also given in Table 1. Graphs of the error plotted as a function of frequency are shown in Figs. 8 and 9. These graphs show that the specifications are met. For higher frequencies the errors converge toward zero.

The reader should be aware that quantization leading to a binary representation for each coefficient is more difficult than quantizing a decimal number. To achieve a minimum word length it is necessary to carefully quantize each group of coefficients so that error cancellation takes place. The results obtained in Table 1 are now compared to those obtained using conventional rounding. First the coefficients in the lag-lead compensator were converted to 18 bits plus sign as shown in Table 2. Then conventional rounding was used to obtain fifteen bit plus sign to 12 bit plus sign representation for the coefficients. Rounding was done by looking at the most significant bit of the truncated portion of the coefficient. If this bit is a one, a one is added to the least significant bit of the truncated coefficient. If the bit is zero a zero is added to the least significant bit. The truncated coefficients are shown in Table 3. The magnitude function error curves are shown in Fig. 10. Here, with conventional rounding at least 14 bits plus sign are needed to satisfy the specification $\Delta|H(\omega)| = \pm .1$. Furthermore, for this example when 12 bits plus sign are used, conventional rounding gives coefficient values that result in a dc gain of 0.48828042×10^{6} rather than the desired dc gain of one. When the proper strategy is used, however, only 12 bits plus sign are needed for the coefficients as shown in Table 1.

COEFFICIENT	QUANTIZED COEFFICIENT	ERROR	BINARY EQUIVALENT
$b_1 = -1.592691562$	-1.5927734370	8.18750×10^{-5}	10.0110100001
$b_2 = .592894916$	.5930175782	-12.26622×10^{-5}	0.100101111101
$a_0 = .835661882$	.8356933593	-3.14773×10^{-5}	0.110101011111
$a_1 = -1.626383584$	-1.6264648440	8.12600×10^{-5}	10.01011111101
$a_2 = .790925055$	.7910156250	-9.05700×10^{-5}	0.110010101

$\Delta x_1 = -4.07876 \times 10^{-5}$ $\Delta x_2 = -4.07872 \times 10^{-5}$

TABLE 1

TABLE OF QUANTIZED COEFFICIENTS
AND BINARY EQUIVALENTS

EACH BINARY VARIABLE IS GIVEN IN TWO'S
COMPLEMENT FORM WITH A LEADING SIGN BIT

b_1	10.01101000010001011
b_2	0.100101111100011110
a_0	0.110101011110110110
a_1	10.010111111101001011
a_2	0.110010100111101000

TABLE 2

EIGHTEEN BIT PLUS SIGN BINARY EQUIVALENTS
FOR THE FILTER COEFFICIENTS

b_1	10.01101000010001	$\Delta b_1 = 2.08 \times 10^{-5}$
b_2	0.100101111100100	$\Delta b_2 = -5.92 \times 10^{-7}$
a_0	0.110101011110111	$\Delta a_0 = -9.60 \times 10^{-7}$
a_1	10.01011111101001	$\Delta a_1 = 2.02 \times 10^{-5}$
a_2	0.110010100111101	$\Delta a_2 = 9.83 \times 10^{-7}$

(a)

Fifteen Bits Plus Sign $\Delta x_1 = 2.02 \times 10^{-5}$ $\Delta x_2 = 2.02 \times 10^{-5}$

b_1	10.0110100001001	$\Delta b_1 = -4.02 \times 10^{-5}$
b_2	0.10010111110010	$\Delta b_2 = -5.92 \times 10^{-7}$
a_0	0.11010101111011	$\Delta a_0 = 2.96 \times 10^{-5}$
a_1	10.0101111110101	$\Delta a_1 = -4.08 \times 10^{-5}$
a_2	0.11001010011111	$\Delta a_0 = -2.95 \times 10^{-5}$

(b)

Fourteen Bits Plus Sign $\Delta x_1 = -4.07 \times 10^{-5}$ $\Delta x_2 = -4.08 \times 10^{-5}$

b_1	10.011010000100	$\Delta b_1 = 8.19 \times 10^{-5}$
b_2	0.1001011111001	$\Delta b_2 = -5.92 \times 10^{-7}$
a_0	0.1101010111110	$\Delta a_0 = -3.15 \times 10^{-5}$
a_1	10.010111111010	$\Delta a_1 = 8.13 \times 10^{-5}$
a_2	0.1100101001111	$\Delta a_2 = 3.15 \times 10^{-5}$

(c)

Thirteen Bits Plus Sign $\Delta x_1 = 8.13 \times 10^{-5}$ $\Delta x_2 = 8.13 \times 10^{-5}$

b_1	10.01101000010	$\Delta b_1 = 8.19 \times 10^{-5}$
b_2	0.100101111100	$\Delta b_2 = 12.15 \times 10^{-5}$
a_0	0.110101011111	$\Delta a_0 = -3.15 \times 10^{-5}$
a_1	10.01011111101	$\Delta a_1 = 8.13 \times 10^{-5}$
a_2	0.110010101000	$\Delta a_2 = -9.06 \times 10^{-5}$

(d)

Twelve Bits Plus Sign $\Delta x_1 = -4.08 \times 10^{-5}$ $\Delta x_2 = 20.34 \times 10^{-5}$

Table 3
Filter Coefficients Obtained Using
Conventional Rounding

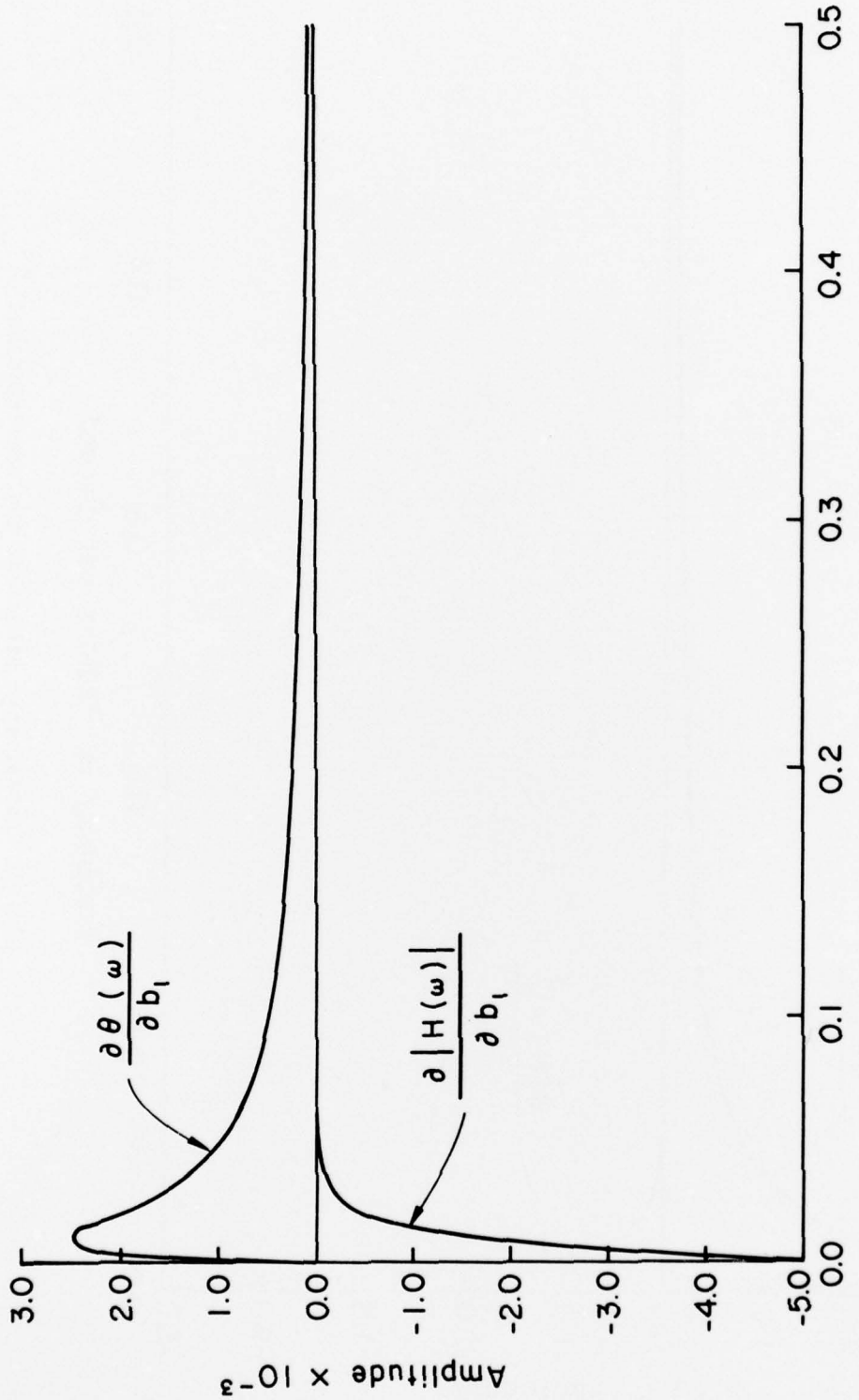


Fig. 1 Partial Derivatives of the Magnitude and Phase Function with Respect to the b_1 Coefficient

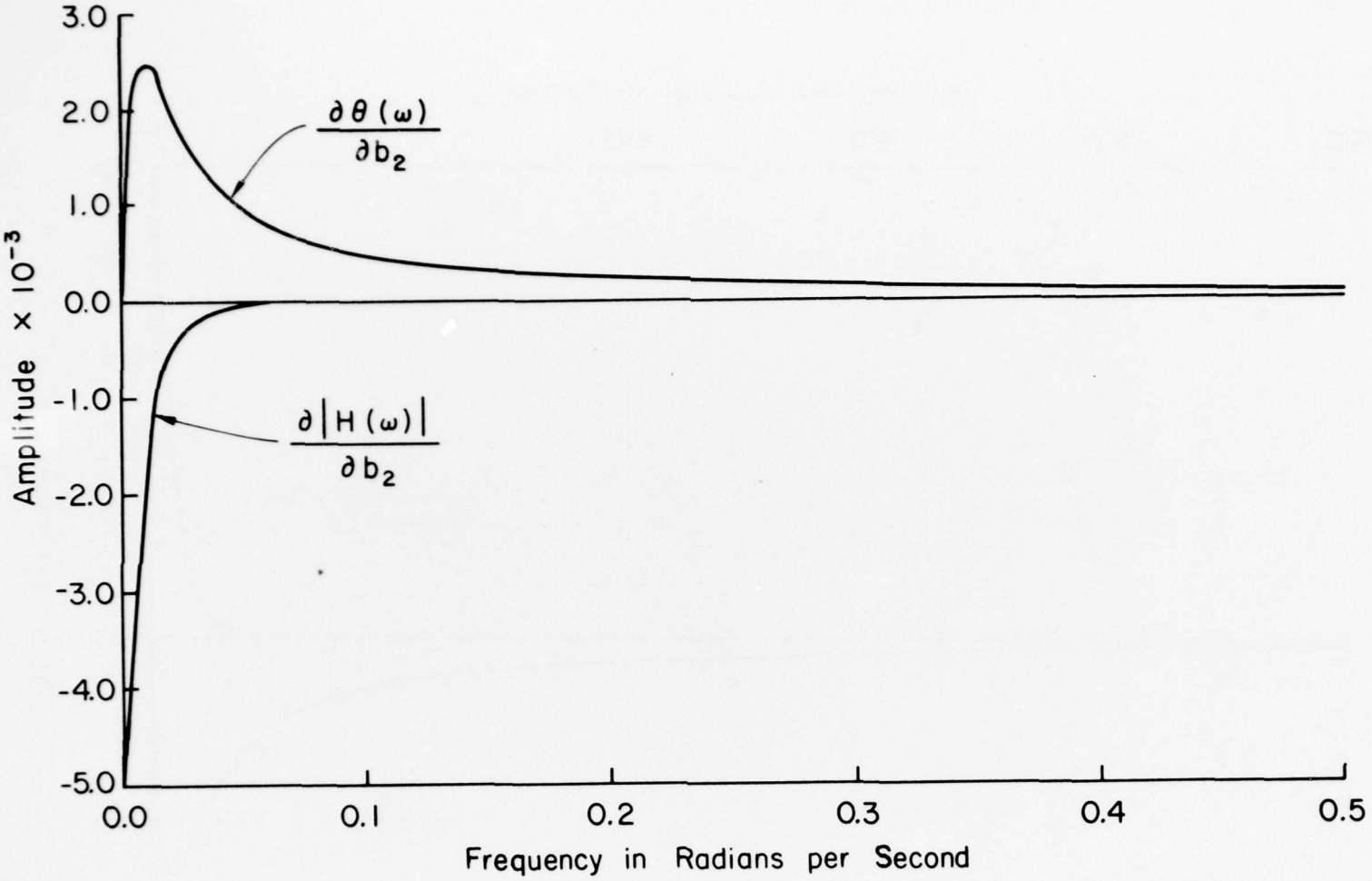


Fig. 2 Partial Derivatives of the Magnitude and Phase Function with Respect to the b_2 Coefficient

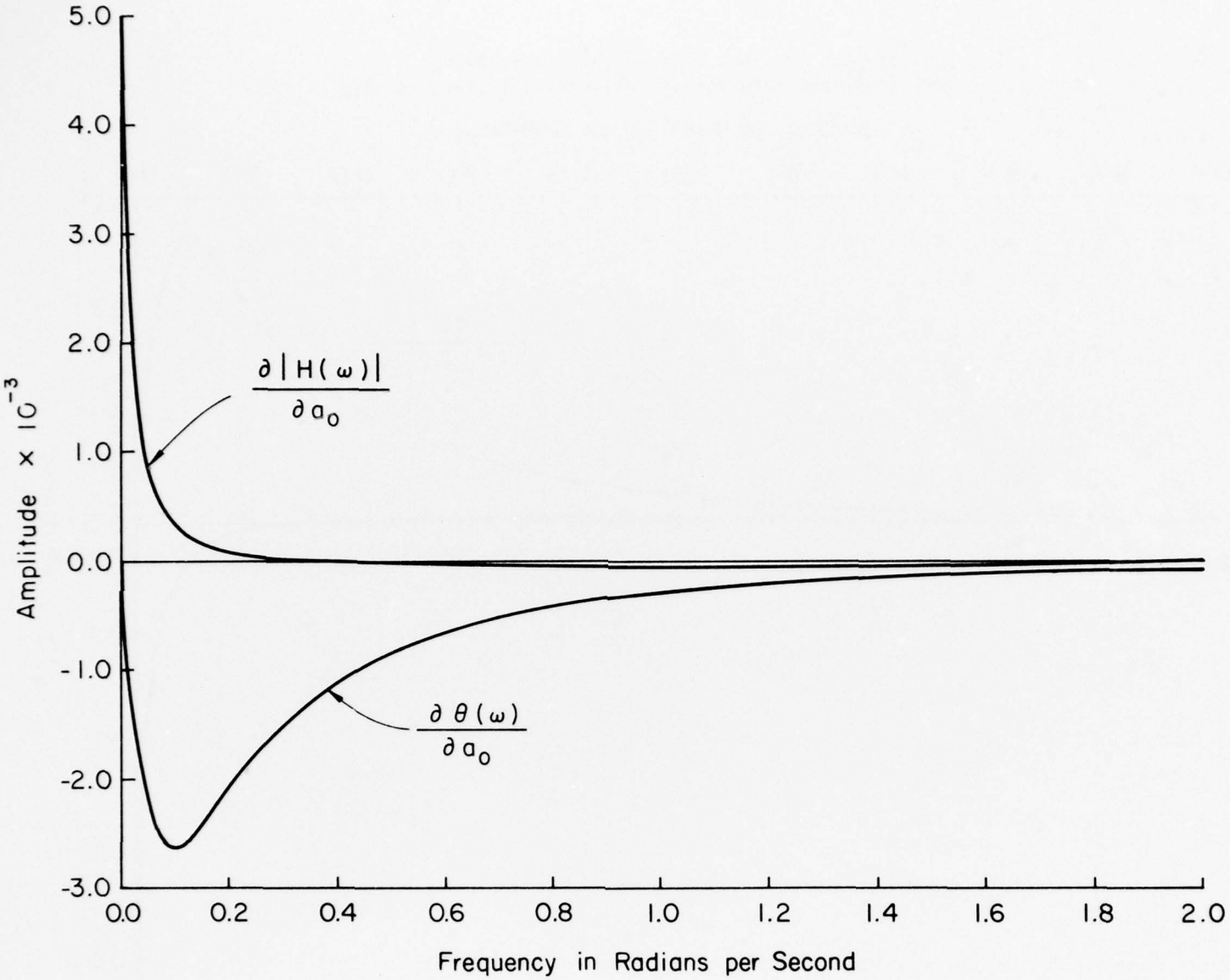


Fig. 3 Partial Derivative of the Magnitude and Phase Functions with Respect to the a_0 Coefficient

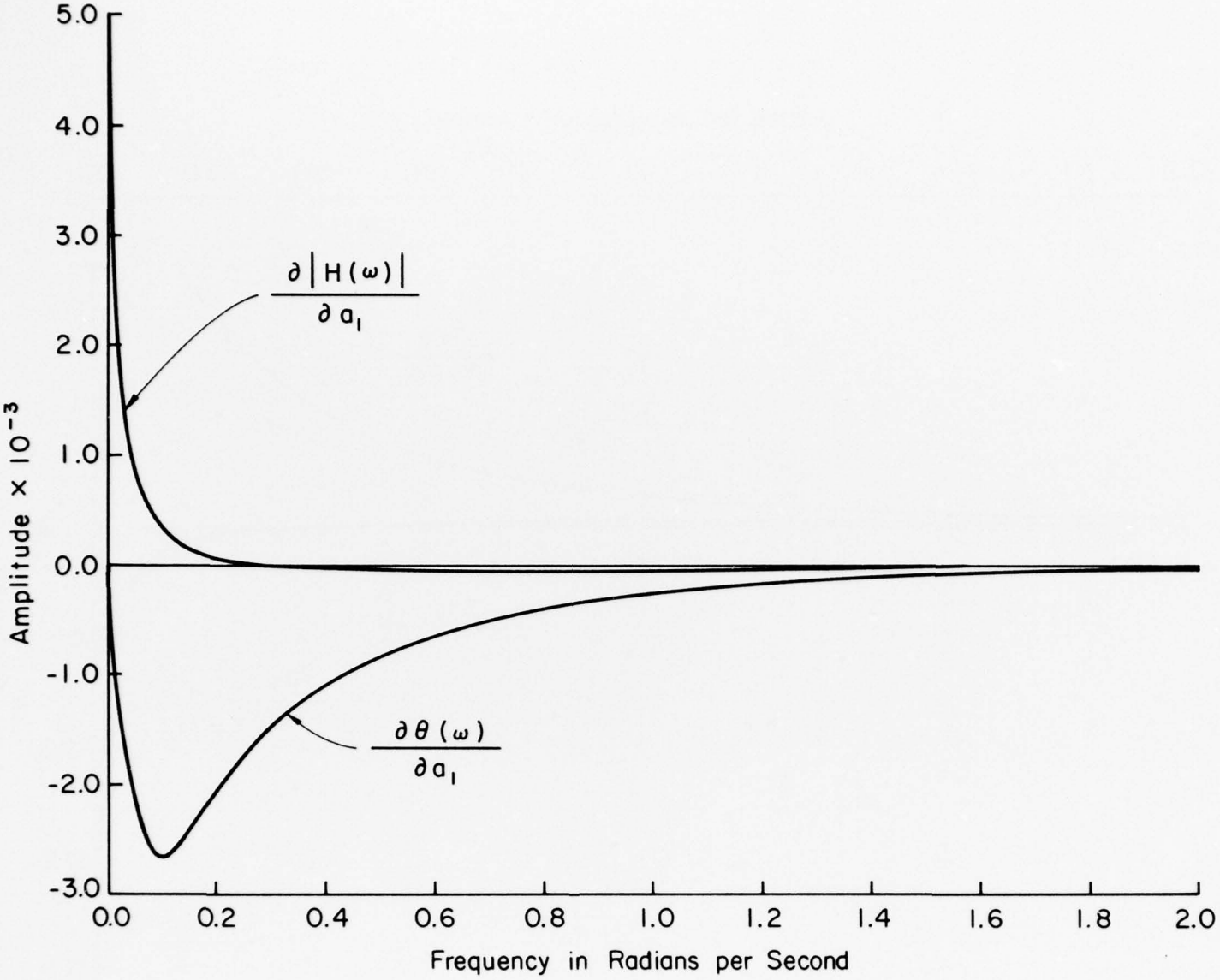


Fig. 4 Partial Derivatives of the Magnitude and Phase Function with Respect to the a_1 Coefficient

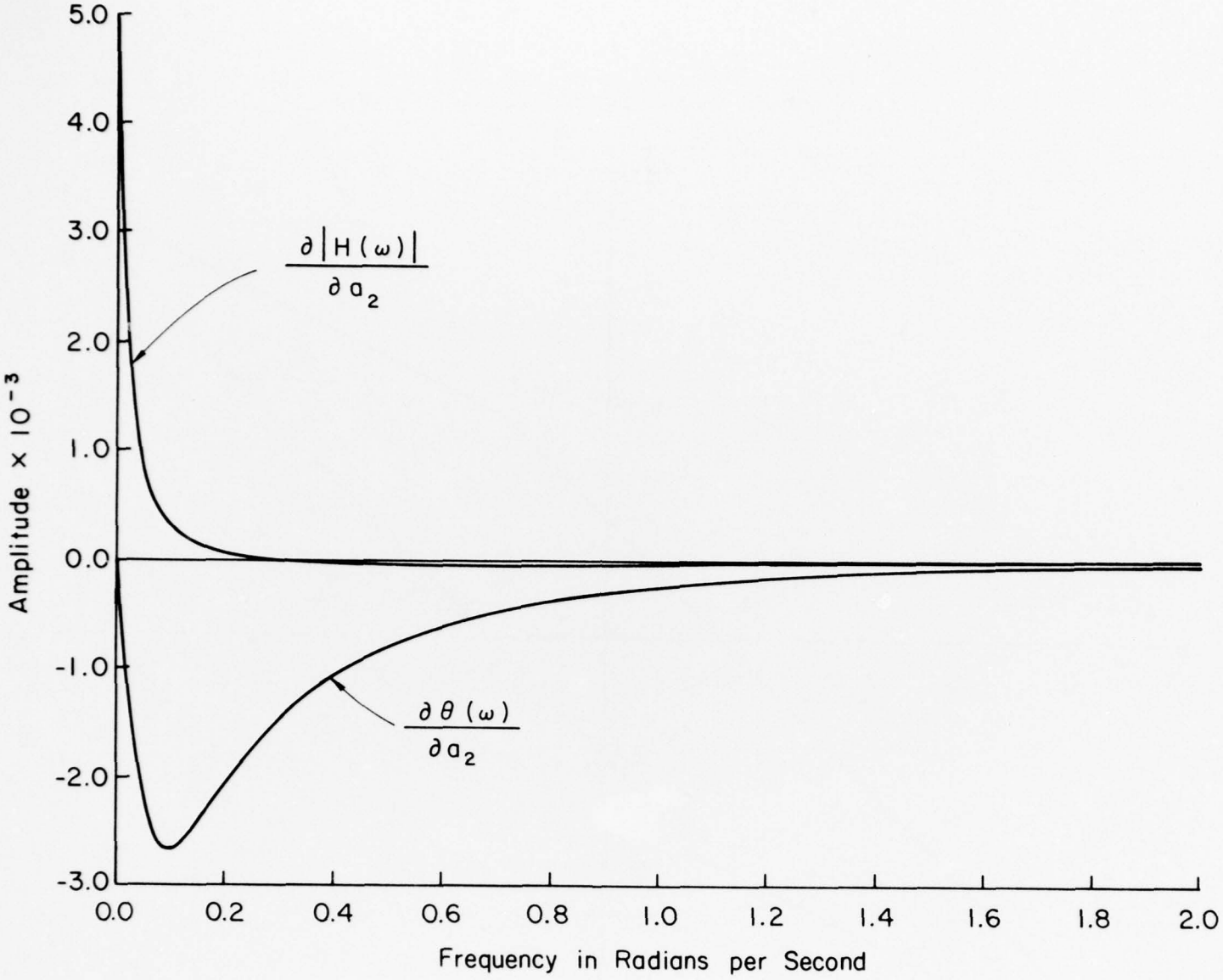


Fig. 5 Partial Derivatives of the Magnitude and Phase Function with Respect to the a_2 Coefficient

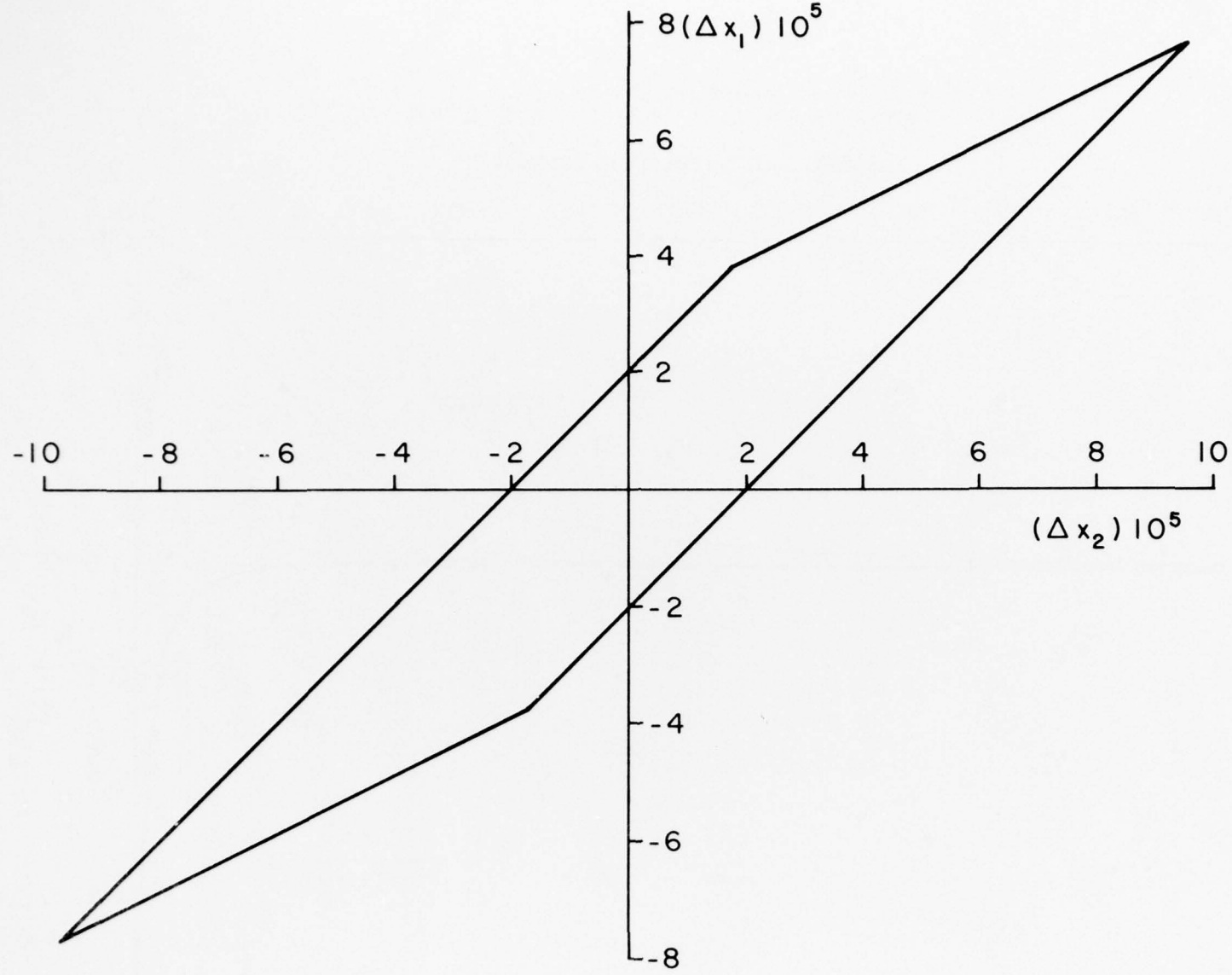


Fig. 6 Quantization Region for $\Delta|H(\omega)| = \pm.1$

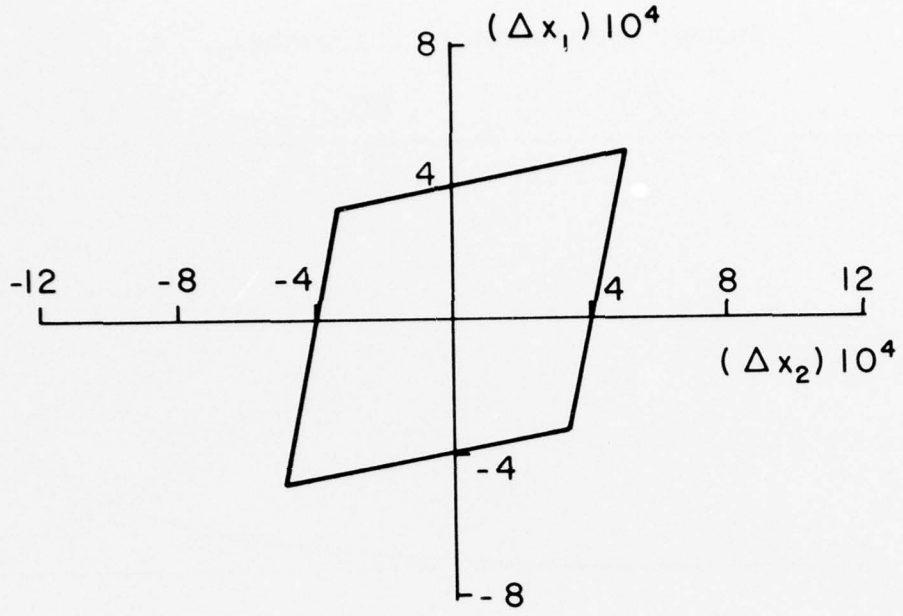


Fig. 7 Quantization Region for $\Delta\theta(\omega) = \pm 1$ Degree

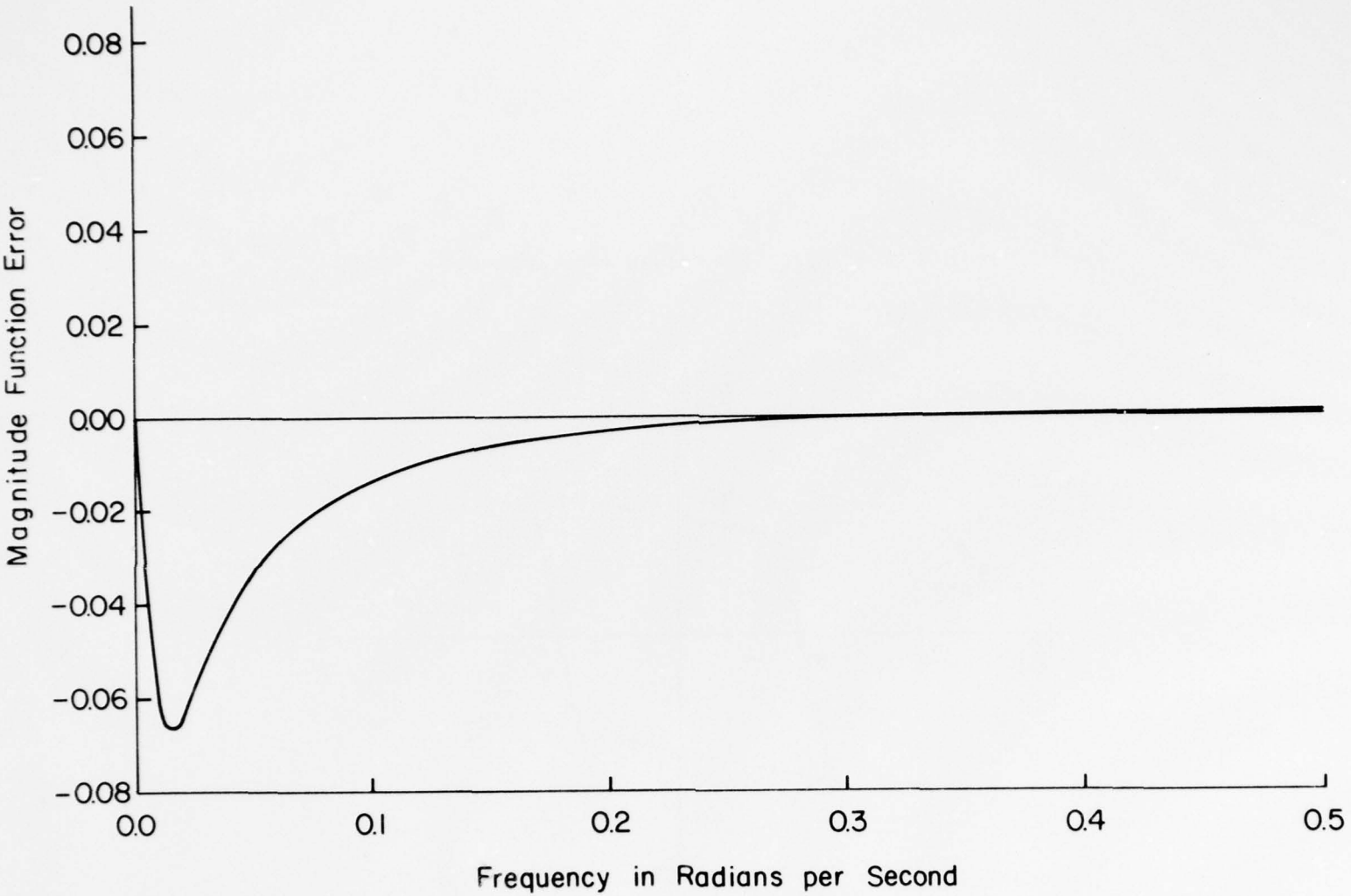


Fig. 8 Magnitude Function Error

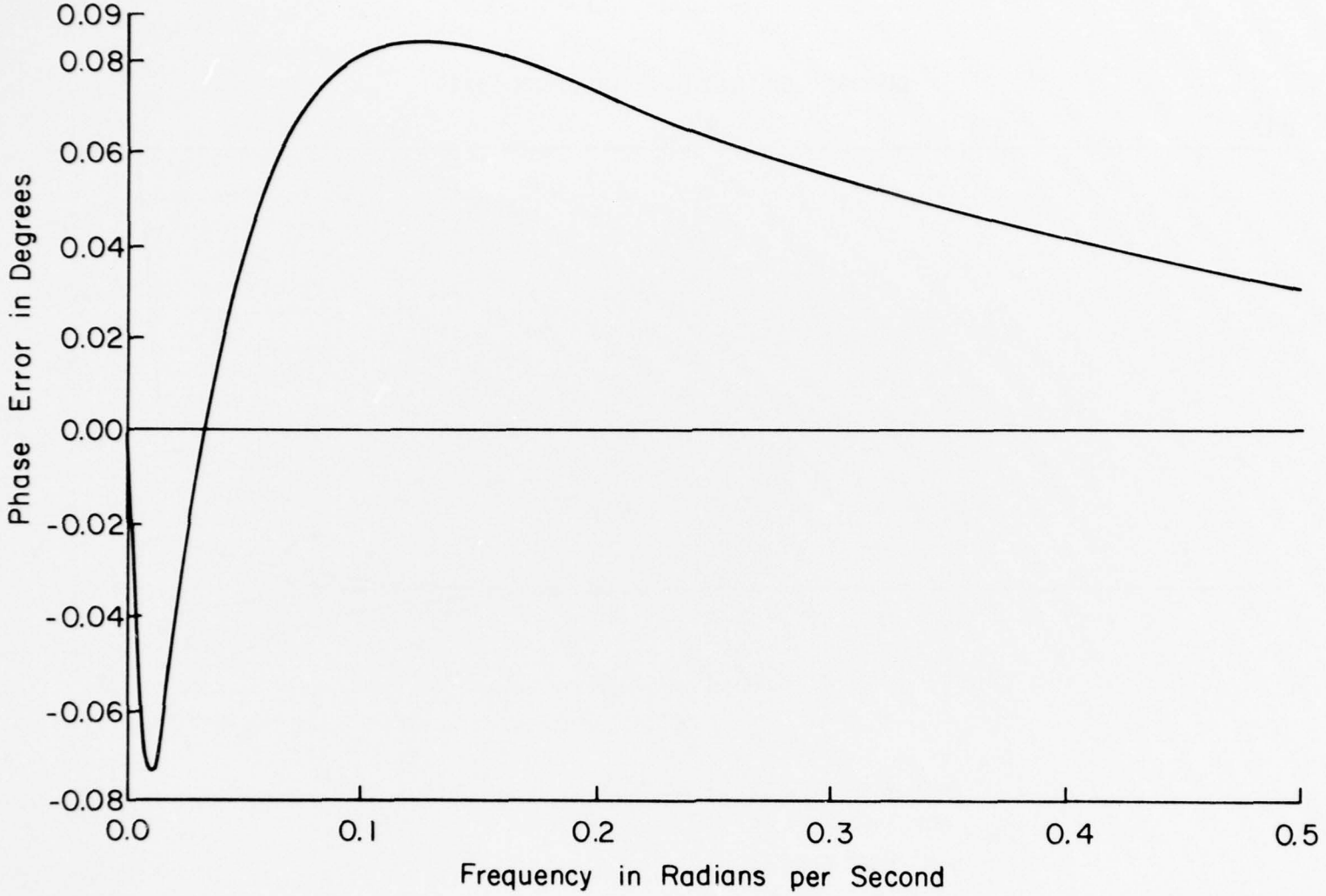


Fig. 9 Phase Function Error

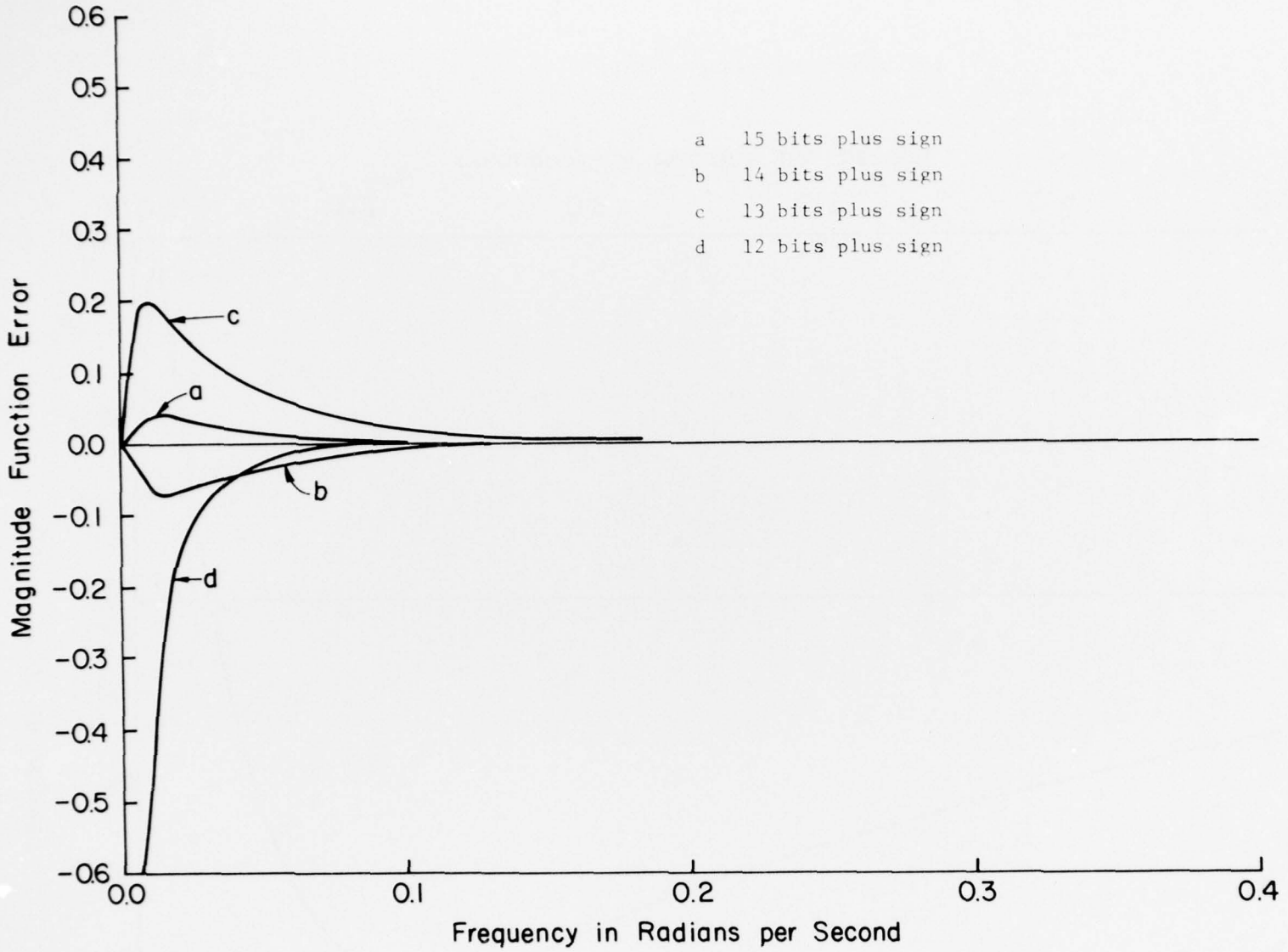


Fig. 10 Magnitude Function Error for Conventional Coefficient Rounding

REFERENCES

1. J. T. Tou, Digital and Sampled Data Control Systems, McGraw-Hill, 1959.

2. A. J. Monroe, Digital Processes for Sampled Data Systems, John Wiley and Sons, 1962.

3. B. C. Kuo, Discrete Data Control Systems, Prentice-Hall, 1970.

4. L. R. Rabiner and C. M. Rader, Digital Signal Processing, IEEE Press, 1972.

5. B. Liu, "Effects of Finite Word Length on the Accuracy of Digital Filters--A Review," IEEE Trans. on Circuit Theory, Vol. CT-18, November 1971.

6. J. F. Kaiser, "Some Practical Considerations in the Realization of Linear Digital Filters," Proc. Third Annual Allerton Conference on Circuit and System Theory, pp. 621-633, 1965.

7. R. K. Otnes and L. P. McNamee, "Instability Thresholds in Digital Filters Due to Coefficient Rounding," IEEE Trans. on Audio and Electroacoustics, Vol. AV-18, December 1970.

8. P. E. Mantey, "Eigenvalue Sensitivity and State-Variable Selection," IEEE Trans. on Automatic Control, Vol. AC-13, June 1968.

9. C. M. Rader and B. Gold, "Effects of Parameter Quantization on the Poles of a Digital Filter," Proc. IEEE, pp. 688-689, May 1967.

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

11. E. Avenhaus, "On the Design of Digital Filters with Coefficients of Limited Word Length," IEEE Trans. on Audio and Electroacoustics, August 1972.

12. J. B. Knowles and E. M. Olcayto, "Coefficient Accuracy and Digital Filter Response,"IEEE Trans. on Circuit Theory, Vol. CT-15, March 1968.

☆U.S.Government Printing Office: 1977 — 757-001/03

