A STUDY OF DIGITAL FILTER DESIGN TECHNIQUES

by

BUGGA RAO VUJJENI

B.E., Osmania University, Andhra Pradesh, India, 1973

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1976

Approved by:

N. Ahmed

Major Professor

TABLE OF CONTENTS

3'

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

This report is primarily concerned with the study of some of the more important design methods that are normally used in the design of digital filters. These methods can be divided into two broad classes, depending upon whether the digital filter of interest is of the FIR (finite impulse response) or the IIR (infinite impulse response) type.

FIR filters can be designed using three well-known methods, which are: (i) the window method, (ii) the frequency sampling method, and (iii) the optimization method. In this report, the window method has been discussed in detail and various design examples are presented. The other two methods are discussed relatively briefly.

Again, two design methods are considered in the case of IIR filters. These are: (i) the impulse invariance method, and (ii) the bilinear transformation method. More attention has been given to the bilinear transformation method since it is used frequently in the practice. Several illustrative examples are presented, including one related to bandpass filters in the low frequency region; 4-8 Hz and 8-12 Hz. These types of filters can be used in the area of EEG (electroencephalogram) signal processing.

This report consists of five chapters. Chapters II and III discuss design techniques of FIR and IIR digital filters respectively. Chapter IV provides a description of some experimental results related to EEG signal processing using IIR filters which are designed via the bilinear transformation technique. Some concluding remarks are included in Chapter V.

CHAPTER II

DESIGN TECHNIQUES FOR FIR DIGITAL FILTERS

The main objective of this chapter is to discuss the methods available for the design of FIR digital filters. There are basically three well-known FIR digital filter design methods; they are: (i) the window method (ii) the frequency sampling method and (iii) the optimization method.

The next sections discuss each of these methods in some detail.

2.1. The Window Method

Since the frequency response $H(e^{jw})$ of a digital filter is periodic with period 2π, it can be expressed in the form of a Fourier series expansion as follows:

$$H(e^{jw}) = \sum_{n=-\infty}^{\infty} h(n)e^{-jwn} \qquad (2.1\text{-}1)$$

where
$$h(n) = \frac{1}{2\pi}\int_0^{2\pi} H(e^{jw})\ e^{jwn}dw \qquad (2.1\text{-}2)$$

For designing FIR filters, there are two difficulties with respect to Equation (2.1-1). First, the impulse response h(n) is infinite in duration; second, the filter is unrealizable because the impulse response $h(n) \neq 0$ for $n < 0$. Hence one possible way of obtaining an FIR filter would be to truncate the above infinite Fourier series at $n = \pm m$. However, direct truncation could lead to undesirable effects such as the Gibbs phenomenon. A better way of obtaining an FIR filter is to use a finite duration weighting sequence w(n), called a window, to appropriately modify the Fourier coefficients h(n) in Eq. (2.1-1).

Effective windows are those that consist of a central lobe which contains most of the energy, and side lobes that decrease rapidly as w tends to π.

To obtain an FIR approximation to a given $H(e^{jw})$, we merely need to form the weighted sequence $\hat{h}(n) = h(n)\cdot w(n)$ where h(n) is the Fourier coefficient of frequency response $H(e^{jw})$; w(n) is the weighting sequence whose Fourier transform is $W(e^{jw})$.

The above discussion leads to the question of desirable window characteristics and how closely they are attained in practice. To this end some aspects of three window functions are described in what follows.

Types of windows. Several types of windows are available, e.g. rectangular, generalized Hamming, Kaiser, Dolph-Chebyshev, triangular, and Blackman windows. The windows discussed here included the rectangular, Hamming, and Kaiser windows.

(i) Rectangular window: The weighting function for an N-point rectangular window is defined as

$$w_R(n) = \begin{cases} 1.0 & -(\frac{N-1}{2}) \leq n \leq (\frac{N-1}{2}) \\ 0.0 & \text{elsewhere} \end{cases} \tag{2.1-3}$$

Again, the frequency response of the above window is given by

$$W_R(e^{jw}) = \sum_{n=-(N-1)/2}^{(N-1)/2} e^{-jwn} = \frac{\sin(wN/2)}{\sin(w/2)} \tag{2.1-4}$$

A sketch of Eq. (2.1-4) is as shown in Fig. 2.1-1, from which it is evident that considerable energy could be spread over the side lobes. This is the main disadvantage of rectangular windows.

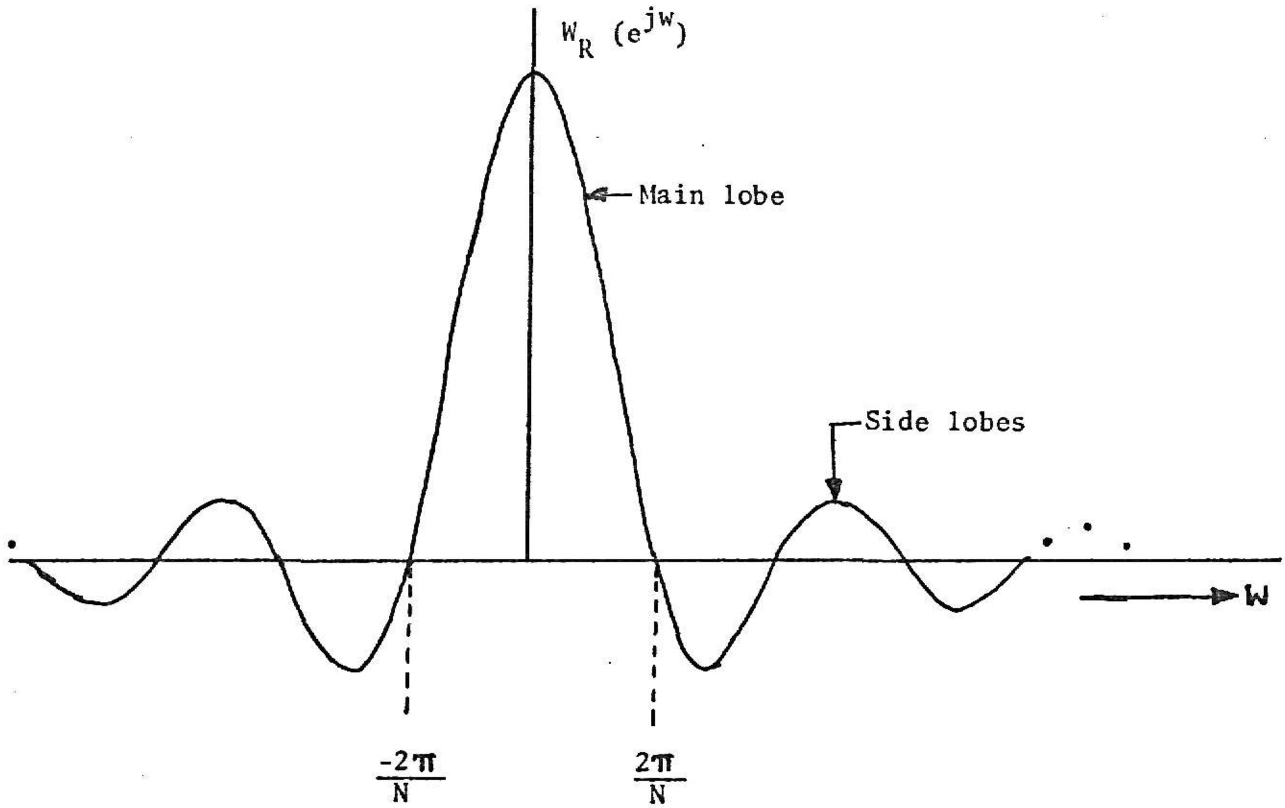


Fig. 2.1-1. Frequency response of a rectangular window.

(ii) Generalized Hamming window: The weighting function for this type of window is defined as

$$w_H(n) = \begin{cases} \alpha + (1-\alpha)\cos\left(\frac{2\pi n}{N}\right) & -\left(\frac{N-1}{2}\right) \le n \le \left(\frac{N-1}{2}\right) \\ 0.0 & \text{elsewhere} \end{cases} \qquad (2.1\text{-}5)$$

where α is in the range $0 \le \alpha \le 1.0$. If $\alpha = 0.54$ the window is called a Hamming window; again if $\alpha = 0.5$, it is called a hanning window.

We can also write Eq. (2.1-5) in the form

$$w_H(n) = w_R(n)\ [\ \alpha + (1-\alpha)\ \cos\ (\frac{2\pi n}{N})],$$

from which it is apparent that the frequency response of the generalized Hamming window is the convolution of the frequency response of the rectangular window $W_R(e^{jw})$, with an impulse train as follows:

$$W_H(e^{jw}) = W_R(e^{jw}) * [\alpha\delta(w) + \frac{(1-\alpha)}{2}\delta(w-\frac{2\pi}{N}) + \frac{(1-\alpha)}{2}\delta(w+\frac{2\pi}{N})].$$

That is

$$W_H(e^{jw}) = \alpha W_R(e^{jw}) + \frac{(1-\alpha)}{2} W_R[e^{j(w-\frac{2\pi}{N})}] + \frac{(1-\alpha)}{2} W_R[e^{j(w+\frac{2\pi}{N})}] \tag{2.1-6}$$

A graphical interpretation of Eq. (2.1-6) is shown in Fig. 2.1-2, along with the resulting frequency response $W_H(e^{jw})$ of a Hamming window; i.e. for $\alpha = 0.54$

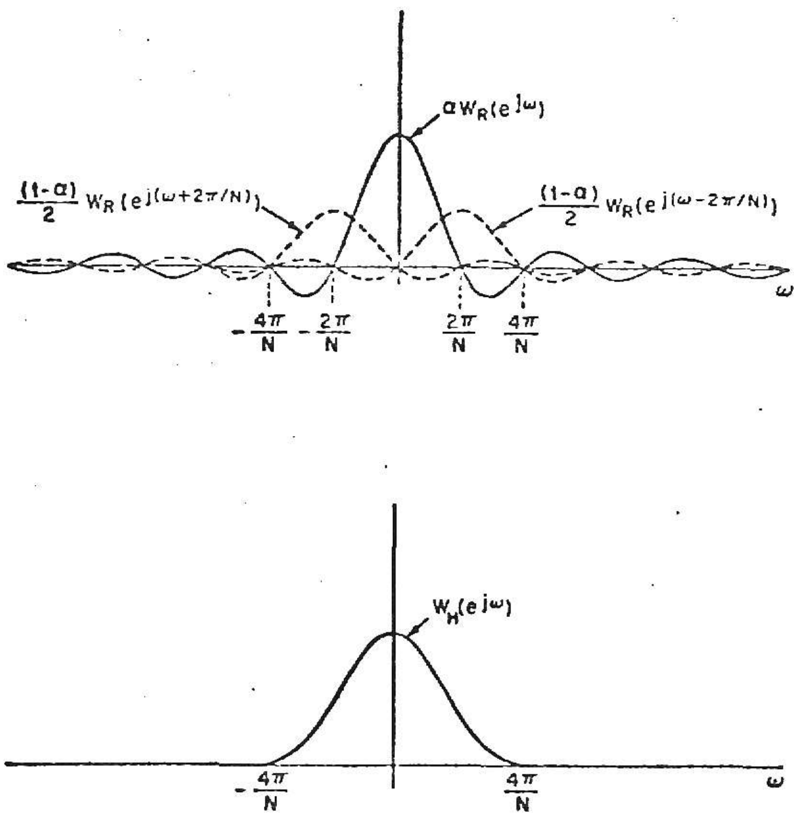


Fig. 2.1-2. Frequency response of a Hamming window; see Eq. (2.1-6).

(iii) Kaiser window: The weighting function for the Kaiser window is defined as

$$W_K(n) = \frac{I_0(\beta\sqrt{1-[2n/(N-1)]^2})}{I_0(\beta)}, \quad -(\frac{N-1}{2}) \leq n \leq (\frac{N-1}{2}) \qquad (2.1\text{-}7)$$

where β is a constant that specifies a frequency response trade-off between the peak height of the side lobe ripples and the width of the main lobe; $I_0(x)$ is the zeroth-order Bessel function.

It has also been shown that in the continuous-time case, the frequency response of a Kaiser window is proportional to

$$\frac{\sin[\beta\sqrt{(w/w_\beta)^2 - 1}]}{\sqrt{(w/w_\beta)^2 - 1}} \qquad (2.1\text{-}8)$$

where w_β is the spectral width of the central lobe of the frequency response. Details related to the Kaiser window are available elsewhere [1].

Next, three illustrative examples which employ the window method are presented.

Example 2.1-1. Design an FIR filter which provides an approximation to an ideal lowpass filter with zero phase. The impulse response of such a filter can be shown to be as follows [1]

$$h(n) = \frac{\sin(2\pi F_c n)}{\pi n} \qquad -N < n < N \qquad (2.1\text{-}9)$$

where F_c is the normalized cutoff frequency. Use the rectangular and Hamming windows with $F_c = 0.2$ and $N = 128$. Assume a sampling frequency of 64 Hz.

Solution: The steps involved in the design of an above FIR lowpass filter are as follows:

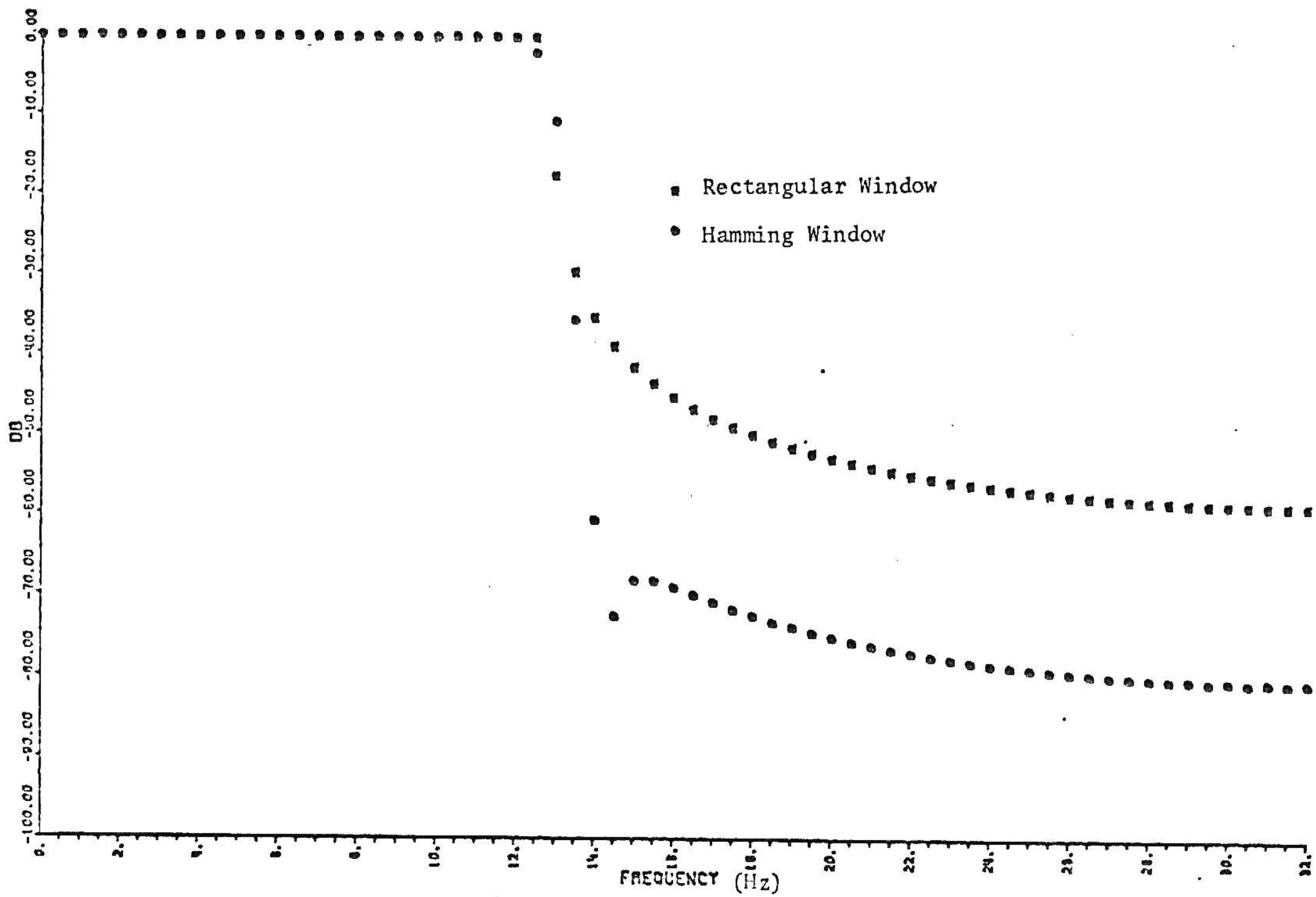


Fig. 2.1-3. Magnitude Characteristics of an FIR Lowpass Digital Filter, N = 128

Step 1. Generate the impulse response $h(n)$, $n = 0,1,\ldots,128$ using Eq. (2.1-9)

Step 2. Multiply $h(n)$ with 128 values of a Hamming window which are generated using Eq. (2.1-5) with $\alpha = 0.54$. Then the resulting sequence is

$$\hat{h}(n) = h(n) \cdot w_H(n), \quad n = 0,1,\ldots,128$$

which yields the impulse response of the desired FIR digital filter.

A program to implement the above two steps is given in Appendix II.

Fig. 2.1-3 shows the corresponding magnitude characteristic plotted with rectangular and Hamming windows. We note that the Hamming window characteristic is substantially superior in the vicinity of 12.8 Hz which is the cut off frequency.

A Fortran program that enables one to plot the magnitude characteristic for the above filter is given in Appendix III.

Example 2.1-2. Design an FIR digital filter which approximates an ideal bandpass filter with the following specifications:

phase = zero

Lower cut off frequency $F_l = 4$ Hz

Upper cut off frequency $F_h = 8$ Hz

Number of data points, $N = 128$

Sampling frequency $F_s = 64$ sps.

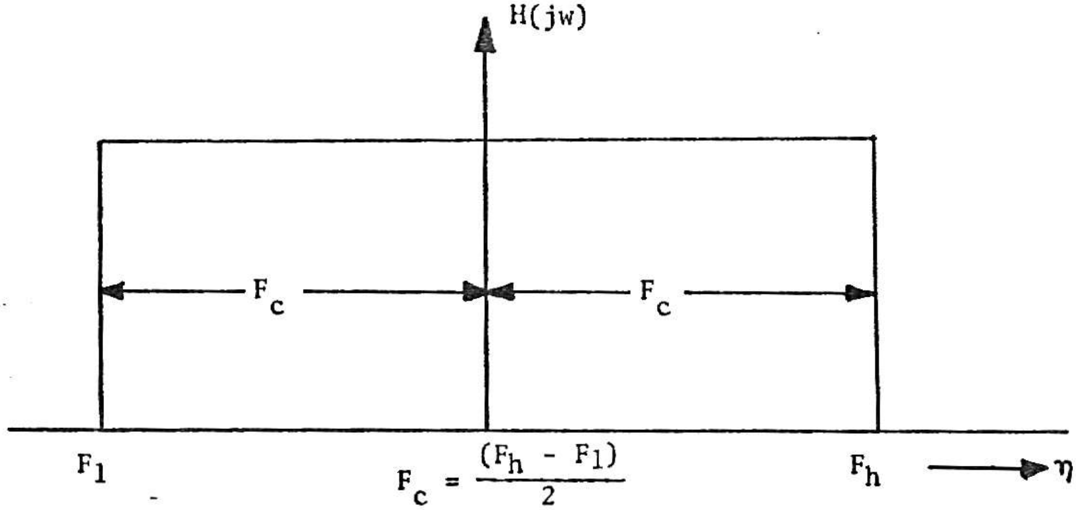
Use the Hamming window.

Solution: Consider the ideal bandpass filter characteristic which is as shown below. It can be shown that the impulse response of an ideal digital filter that yields this characteristic is given by [2]

$$h(n) = \frac{2\sin(2\pi F_c n)}{\pi n} \cdot \cos(2\pi F_0 n), \quad -\infty < n < \infty \qquad (2.1\text{-}10)$$

where $F_c = (F_h - F_1)/2$

and $F_0 = \sqrt{F_h F_1}$



Evaluation of $F_c = (F_h - F_1)/2\, F_s$, $F_1 = F_1/F_s$, $F_h = F_h/f_s$ and $F_0 = \sqrt{F_h F_1}$ results in $F_1 = 4/64$, $F_h = 8/64$, $F_c = 2/64$ and $F_0 = \sqrt{32}/64$. The the Eq. (2.1-10) becomes

$$h(n) = \frac{2 \sin (\frac{2\pi n}{32})}{\pi n} \cos (\frac{2\pi \sqrt{32} n}{64}) \qquad (2.1\text{-}11)$$

The solution again consists of two steps.

<u>Step 1</u>. Generate the sequence $h(n)$, $n = 0,1,\ldots,128$ using Eq. (2.1-11).

<u>Step 2</u>. Form the weighted sequence $\hat{h}(n) = h(n) \cdot w_H(n)$, $n = 0,1,\ldots,128$.

Then $\hat{h}(n)$, $n = 0,1,\ldots,128$ represents the impulse response of the desired digital filter.

Fig. 2.1-4 shows the magnitude characteristic for the above filter with a Hamming window for comparison, the corresponding characteristic with

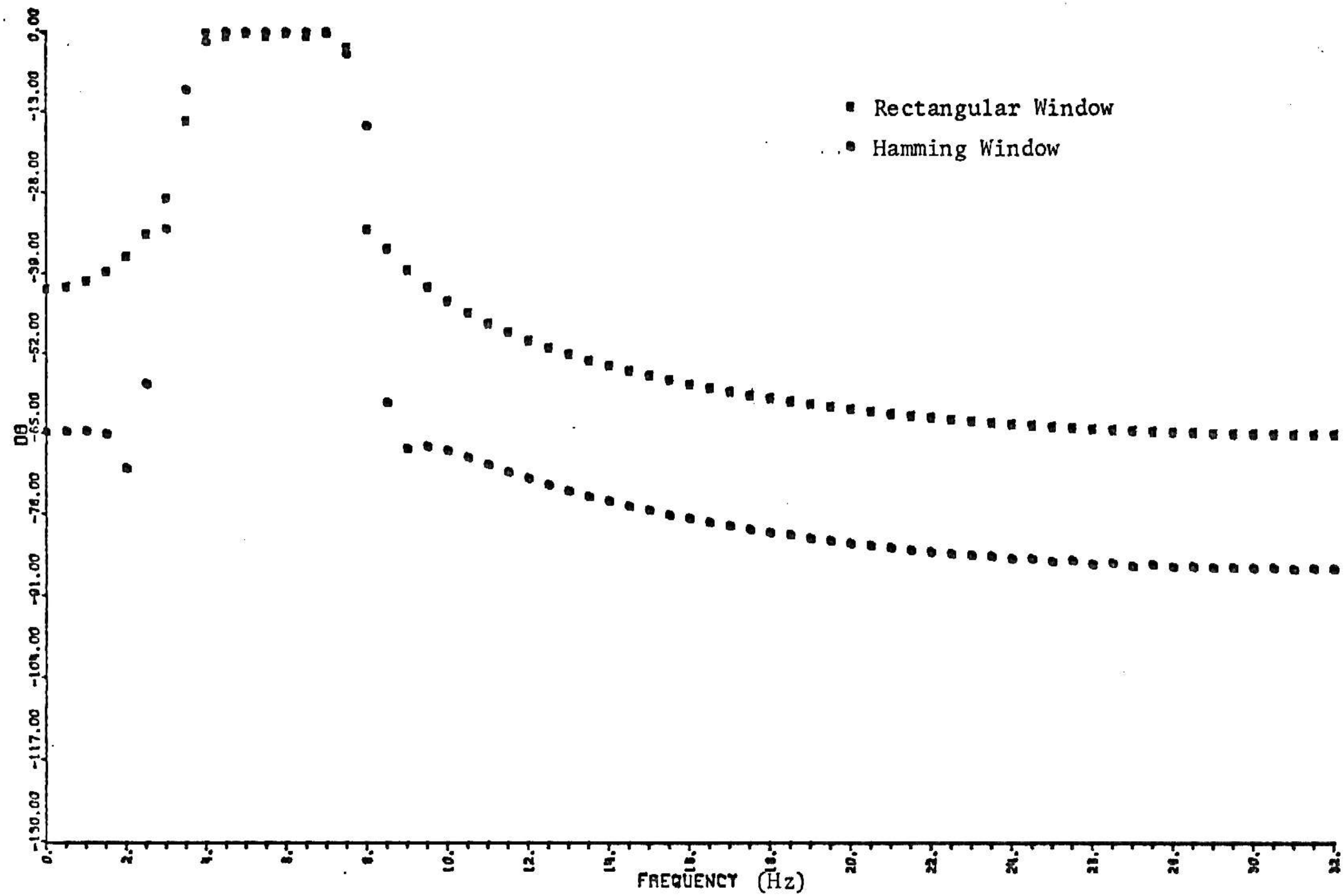


Fig. 2.1-4. Magnitude Characteristics of an FIR Bandpass Digital Filter, N = 128

a rectangular window is also included. From Fig. 2.1-4 it is again apparent that the windowing process yields more attenuation in the vicinities of the cut off frequencies.

Example 2.1-3. Design an FIR digital filter which provides an approximation to an ideal bandpass filter with the following specifications.

Zero phase

Lower cut off frequency F_l = 8Hz

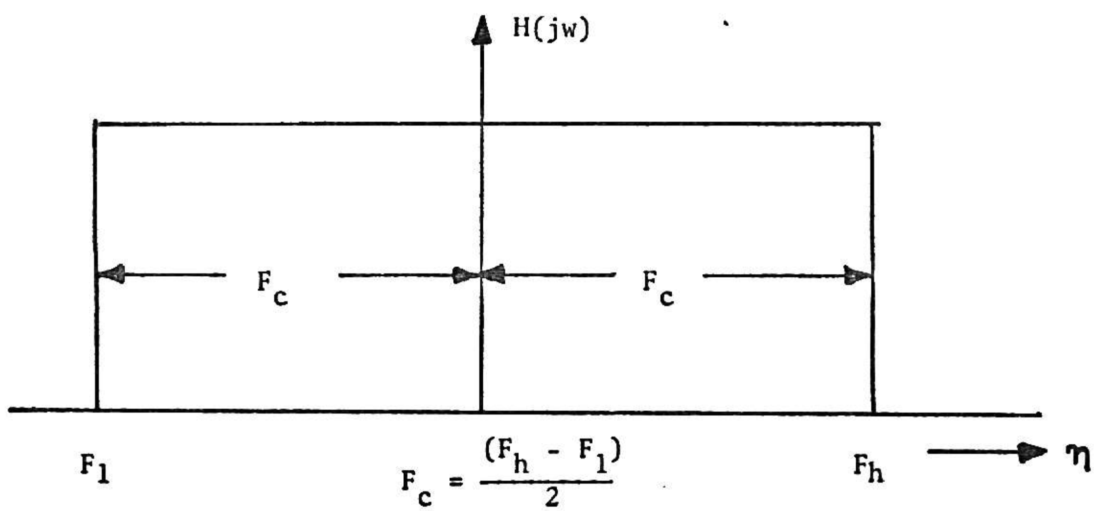
Upper cut off frequency F_h = 12 Hz

Number of data points, N = 128

Sampling frequency F_s = 64 sps

Use the Hamming window.

Solution: The impulse response of an ideal bandpass filter is given by Eq. (2.1-10).



Evaluating $F_c = (F_h - F_l)/2F_s$, $F_l = F_l/F_s$, $F_h = F_h/F_s$ and $F_0 = \sqrt{F_h F_l}$, we obtain $F_l = 8/64$, $F_h = 12/64$, $F_c = 2/64$ and $F_0 = \sqrt{96}/64$.

Then the Eq. (2.1-10) becomes

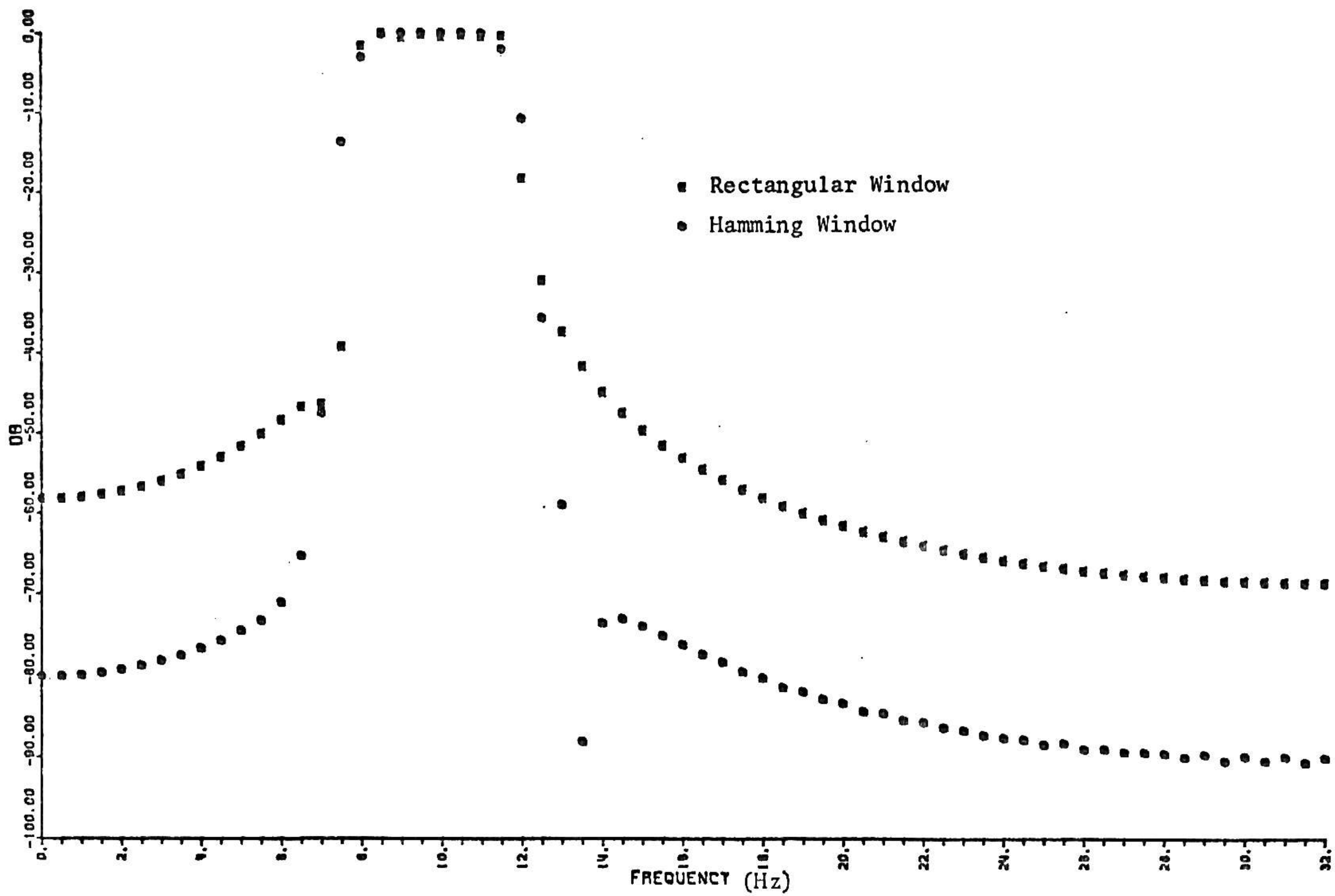


Fig. 2.1-5. Magnitude Characteristics of an FIR Bandpass Digital Filter, N = 128

$$h(n) = \frac{2\sin(2\pi n/32)}{\pi n} \cos\left(\frac{2\pi\sqrt{96}n}{64}\right) \qquad (2.1\text{-}12)$$

The solution again consists of the following steps.

Step 1. Generate the impulse response $h(n)$, $n = 0,1,\ldots,128$ using Eq. (2.1-12).

Step 2. Multiply $h(n)$ with 128 values of Hamming window which are generated using Eq. (2.1-5) with $\alpha = 0.54$. The resulting weighted sequence is

$$\hat{h}(n) = h(n)\cdot w_H(n), \quad n = 0,1,\ldots,128$$

which yields the impulse response of the desired FIR digital filter.

Fig. 2.1-5 shows the corresponding magnitude characteristic of the above filter plotted with rectangular and Hamming windows; as in the previous examples, it is evident that the Hamming window provides better attenuation characteristics relative to the rectangular window.

In conclusion, it is remarked that the reason for the superior performance of the Hamming window is that more of the total energy is included in the main lobe of its frequency spectrum relative to the frequency spectrum of the rectangular window.

2.2. The Frequency Sampling Method

The main idea of the frequency sampling design method is that a desired frequency response can be approximated by sampling it at N equispaced points and then obtaining an interpolated frequency response that passes through the frequency samples.

Since an FIR filter can be specified by either the impulse response coefficients $\{h(n)\}$ or the corresponding DFT coefficients $\{H(K)\}$, these sequences are related by the DFT and IDFT definitions as follows:

$$H(k) = \sum_{n=0}^{N-1} h(n)\ e^{-j(2\pi/N)nK} \qquad \text{DFT} \qquad (2.2\text{-}1)$$

and

$$h(n) = \frac{1}{N}\sum_{K=0}^{N-1} H(K)e^{j(2\pi/N)nK} \qquad \text{IDFT} \qquad (2.2\text{-}2)$$

However, for an FIR sequence, the DFT sample H(K) can be regarded as sample of the filter's z-transform, which is evaluated at N equally spaced points around the unit circle; that is

$$H(K) = H(z)\Big|_{z = e^{j(2\pi/N)K}} \qquad (2.2\text{-}3)$$

where

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}$$

Hence the z-transform of an FIR filter can be easily expressed in terms of its DFT coefficients by substituting Eq. (2.2-2) into the Eq. (2.2-3); i.e.

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{K=0}^{N-1} H(K)e^{j(2\pi/N)nK} \right] z^{-n} \qquad (2.2\text{-}4)$$

We can also write Eq. (2.2-4) in the form

$$H(z) = \sum_{K=0}^{N-1} \frac{H(K)}{N} \sum_{n=0}^{N-1} [e^{j(2\pi/N)K}\ z^{-1}]^n$$

$$= \sum_{K=0}^{N-1} \frac{H(K)}{N}\ \frac{(1 - e^{j2\pi K}\ z^{-N})}{[1-e^{j(2\pi/N)K}z^{-1}]} \qquad (2.2\text{-}5)$$

Since $e^{j2\pi K} = 1$, and K is an integer, Eq. (2.2-5) simplifies to yield

$$H(z) = \frac{(1 - z^{-N})}{N} \sum_{K=0}^{N-1} \frac{H(K)}{[1 - z^{-1}\ e^{j(2\pi/N)K}]} \qquad (2.2\text{-}6)$$

The interpretation of Eq. (2.2-6) is that, to approximate any continuous frequency response, we could sample in frequency at N equispaced points around the unit circle and evaluate the continuous frequency response as an interpolation of the sampled frequency response.

Now, substitution of $z = e^{jw}$ in (2.2-6) results in

$$H(e^{jw}) = \frac{e^{-jw(N-1)/2}}{N} \sum_{K=0}^{N-1} \frac{H(K)e^{-j(\pi K/N)}\sin(wN/2)}{\sin(w/2 - \pi K/N)} \qquad (2.2\text{-}7)$$

Equation (2.2-7) shows the frequency response of the desired filter to be a linear combination of the frequency samples H(K) with frequency interpolating functions that are of the form [1]

$$A(w,K) = e^{-j(\pi K/N)} \frac{\sin(wN/2)}{\sin(w/2 - \pi K/N)}$$

$$= \pm e^{-j(\pi K/N)} \frac{\sin[N(w/2 - \pi K/N)]}{\sin(w/2 - \pi K/N)} \qquad (2.2\text{-}8)$$

Thus, every frequency sample with value H(K) produces a frequency response proportional to a constant times $\sin(Nw/2)/\sin(w/2)$, displaced by $\pi K/N$ *in frequency.* For filters with <u>reasonably smooth frequency responses</u>, the interpolation error is generally small. When this is not the case, optimization techniques are used. For example, in the case of bandselect filters, the desired frequency response changes radically across the bands. Hence the frequency samples which occur in transition bands are made to be unspecified variables whose values are chosen by an optimization algorithm which minimizes some function of the approximation error of the filter. Linear programming techniques are employed to perform the necessary minimization.

There are two types of frequency sampling filters, depending on where the initial frequency sample occurs. The first set of frequencies are determined by the relation

$$F_K = \frac{K}{N} \qquad K = 0,1,\ldots, N-1; \ \underline{\text{Type 1}}.$$

Again, there is a second set of uniformly spaced frequencies for which a frequency sampling structure can conveniently be obtained. This set of frequencies is determined by the relation

$$F_K = \frac{(K + 1/2)}{N} \qquad K = 0,1,\ldots, N-1; \ \underline{\text{Type 2}}.$$

The frequency response of both types of linear phase frequency sampling filter can be obtained by using an optimization algorithm. More details related to the frequency sampling filters for types 1 and 2 are available elsewhere [1].

<u>Frequency Responses</u>. For Type 1 frequency sampling filters, the frequency response is given by [1]

$$H(e^{jw}) = e^{-jw(N-1)/2} \left(\frac{|H(0)|}{N} \frac{\sin(wN/2)}{\sin(w/2)} + \sum_{K=1}^{(\frac{N}{2})-1} \frac{|H(K)|}{N} \right.$$
$$\left. \times \left\{ \frac{\sin[N(w/2 - \pi K/N)]}{\sin(w/2 - \pi K/N)} + \frac{\sin[N(w/2 + \pi K/N)]}{\sin(w/2 + \pi K/N)} \right\} \right) \qquad (2.2\text{-}9)$$

The term inside the outermost parentheses of Eq. (2.2-9) is the magnitude characteristic $\left|H(e^{jw})\right|$ for N even, Type 1, linear phase frequency sampling designs.

For Type 1, N odd, linear phase designs, the magnitude characteristic is given by

$$\left|H(e^{jw})\right| = \left(\frac{H(0)}{N} \frac{\sin(wN/2)}{\sin(w/2)} + \sum_{K=1}^{(N-1)/2} \frac{|H(K)|}{N} \right.$$
$$\left. \times \left\{ \frac{\sin[N(w/2 - \pi K/N)]}{\sin(w/2 - \pi K/N)} + \frac{\sin[N(w/2 + \pi K/N)]}{\sin(w/2 + \pi K/N)} \right\} \right) \qquad (2.2\text{-}10)$$

Again, the corresponding magnitude characteristics of Type 2 linear phase frequency sampling designs are as follows:

$$\left|H(e^{jw})\right| = \left[\sum_{K=0}^{(\frac{N}{2})-1} \frac{|H(K)|}{N} \left(\frac{\sin\{N[w/2-(\pi/N)(K+1/2)]\}}{\sin[w/2-(\pi/N)(K+1/2)]} + \frac{\sin\{N[w/2+(\pi/N)(K+1/2)]\}}{\sin[w/2+(\pi/N)(K+1/2)]}\right)\right]; \quad N \text{ even} \tag{2.2-11}$$

$$\left|H(e^{jw})\right| = \left\{ \frac{|H[(N-1)/2]|}{N} \frac{\sin(wN/2)}{\sin(w/2)} + \sum_{K=0}^{(N-3)/2} \left[\frac{|H(K)|}{N} \left(\frac{\sin\{N[w/2-(\pi/N)(K+1/2)]\}}{\sin[w/2-(\pi/N)(K+1/2)]} + \frac{\sin N[w/2+(\pi/N)(K+1/2)]}{\sin[w/2+(\pi/N)(K+1/2)]}\right)\right]\right\}; \quad N \text{ odd} \tag{2.2-12}$$

Thus any of the above four equations [Eqs. (2.2-9), (2.2-10), (2.2-11), (2.2-12)] can be used in an optimization approach to design linear phase frequency sampling FIR digital filters. The related optimization procedures are discussed in detail in [1].

2.3. The Optimization Method

Since the optimal linear phase FIR filter design problem can be formulated as a Chebyshev approximation problem, it is possible to derive a set of conditions for which it can be proved that the solution is optimal and unique. The desired filter coefficients of the optimal filter can be computed using standard optimization procedures including linear programming. Details pertaining to this method are available in a recent book by Rabiner and Gold [1].

In this section, we merely present an example, the solution for which is obtained by using a computer program available in [1]. By virtue of this program it is possible to readily design optional FIR filters with linear phase.

Example 2.3-1. Design a 128-point bandpass filter with normalized stopband cutoff frequencies of 0.12 and 0.2, and normalized passband cutoff frequencies of 0.128 and 0.192. Ripple weights of 10 and 1 in the stopbands and passband respectively are required. List the impulse response and the corresponding magnitude characteristic.

Solution: The impulse response of the desired filter is obtained by using the Fortran program which is given in Appendix IV. The specific card input to the program is as follows:

Card 1: 128, 001, 003, 001, 016

Card 2: 0, 0.12, 0.128, 0.192, 0.2, 0.5

Card 3: 0, 1, 0

Card 4: 10, 1, 10

Table 2.3-1 provides a list of the impulse response values, as they are obtained from the program in the form of an output. Again, the corresponding magnitude characteristic is as shown in Fig. 2.3-1. The Fortran program which yields the magnitude characteristic values is listed below; it uses an FFT subroutine which is also included.

```
      DIMENSICN RX(128),R(128),IM(128),Y(128),MAG(128),DB(128)
      COMPLEX X(512),CMPLX
      REAL IM,MAG
      RMAX=0.0091962
      N=128
      READ 100, (RX(I),I=1,64)
  100 FORMAT (5E16.8)
      DO 1C5 I=1,64
  105 RX(129-I)=RX(I)
      DO 110 I=1,128
  110 X(I)=CMPLX(RX(I),0.0)
      PRINT 18C,(I,RX(I),I=1,128)
  180 FCRMAT(' ',30X,'X(',I3,')=',E16.8)
      CALL FFT (X,N,0)
      DO 12C I=1,128
      R(I)=REAL(X(I))
      IM(I)=AIMAG(X(I))
      MAG(I)=CABS(X(I))
      IF (MAG(I).EQ.0.0) MAG(I)=1.0CE-10
  120 DB(I)=20.0*ALOG10(MAG(I)/RMAX)
      PRINT 140
  14C FORMAT ('1',T50,'FFT COEFFICIENTS')
      PRINT 15C,(I,R(I),IM(I),MAG(I),DB(I),I=1,128)
  150 FORMAT(' ',30X,'CX(',I3,')=',F9.6,'+',F9.6,'I',1CX,'MAGNITUDE=',
     XF9.7,1CX,'DB=',F9.4)
      STOP
      END
```

Table 2.3-1. A Listing of the Impulse Response Values.

```
**********************************************************************

        FINITE IMPULSE RESPONSE (FIR)
        LINEAR PHASE DIGITAL FILTER DESIGN
        REMEZ EXCHANGE ALGORITHM

        BANDPASS FILTER

 FILTER LENGTH = 128

 ***** IMPULSE RESPONSE *****
      H(  1) = 0.75493306E-02 = H( 128)
      H(  2) =-0.35247589E-02 = H( 127)
      H(  3) =-0.33030920E-02 = H( 126)
      H(  4) = 0.12743641E-02 = H( 125)
      H(  5) = 0.76503716E-02 = H( 124)
      H(  6) = 0.80869980E-02 = H( 123)
      H(  7) =-0.10590032E-02 = H( 122)
      H(  8) =-0.12503546E-01 = H( 121)
      H(  9) =-0.13790924E-01 = H( 120)
      H( 10) =-0.12264239E-02 = H( 119)
      H( 11) = 0.13909370E-01 = H( 118)
      H( 12) = 0.15981570E-01 = H( 117)
      H( 13) = 0.19938369E-02 = H( 116)
      H( 14) =-0.14280036E-01 = H( 115)
      H( 15) =-0.10742736E-01 = H( 114)
      H( 16) =-0.41005970E-02 = H( 113)
      H( 17) = 0.96721255E-02 = H( 112)
      H( 18) = 0.11487542E-01 = H( 111)
      H( 19) = 0.30545495E-02 = H( 110)
      H( 20) =-0.40918216E-02 = H( 109)
      H( 21) =-0.36072107E-02 = H( 108)
      H( 22) =-0.80437877E-03 = H( 107)
      H( 23) =-0.28356655E-02 = H( 106)
      H( 24) =-0.71933977E-02 = H( 105)
      H( 25) =-0.47939271E-02 = H( 104)
      H( 26) = 0.60519151E-02 = H( 103)
      H( 27) = 0.14507901E-01 = H( 102)
      H( 28) = 0.90677532E-02 = H( 101)
      H( 29) =-0.68135150E-02 = H( 100)
      H( 30) =-0.16597769E-01 = H(  99)
      H( 31) =-0.11034880E-01 = H(  98)
      H( 32) = 0.35866290E-02 = H(  97)
      H( 33) = 0.10964226E-01 = H(  96)
      H( 34) = 0.60644224E-02 = H(  95)
      H( 35) =-0.89042354E-03 = H(  94)
      H( 36) =-0.47803856E-04 = H(  93)
      H( 37) = 0.40042326E-02 = H(  92)
      H( 38) =-0.44908291E-03 = H(  91)
      H( 39) =-0.13012860E-01 = H(  90)
      H( 40) =-0.17540032E-01 = H(  89)
      H( 41) =-0.24366314E-02 = H(  88)
      H( 42) = 0.20802289E-01 = H(  87)
      H( 43) = 0.26748106E-01 = H(  86)
      H( 44) = 0.61235943E-02 = H(  85)
      H( 45) =-0.20751640E-01 = H(  84)
      H( 46) =-0.26275784E-01 = H(  83)
      H( 47) =-0.77669397E-02 = H(  82)
      H( 48) = 0.10771818E-01 = H(  81)
      H( 49) = 0.10283232E-01 = H(  80)

      H( 50) = 0.62211789E-03 = H(  79)
      H( 51) = 0.40146150E-02 = H(  78)
      H( 52) = 0.19224723E-01 = H(  77)
      H( 53) = 0.17239630E-01 = H(  76)
      H( 54) =-0.19526465E-01 = H(  75)
      H( 55) =-0.57484791E-01 = H(  74)
      H( 56) =-0.46217659E-01 = H(  73)
      H( 57) = 0.25034785E-01 = H(  72)
      H( 58) = 0.93190193E-01 = H(  71)
      H( 59) = 0.79084933E-01 = H(  70)
      H( 60) =-0.21540657E-01 = H(  69)
      H( 61) =-0.11736792E 00 = H(  68)
      H( 62) =-0.10786615E 00 = H(  67)
      H( 63) = 0.81793070E-02 = H(  66)
      H( 64) = 0.12204480E 00 = H(  65)

                         BAND  1         BAND  2         BAND  3
 LOWER BAND EDGE      0.0             0.128499985     0.19999998
 UPPER BAND EDGE      0.119999945     0.192199945     0.50000000
 DESIRED VALUE        0.0             1.000000000     0.0
 WEIGHTING           10.000000000     1.000000000    10.00000000
 DEVIATION            0.017836068     0.178360641     0.017836068
 DEVIATION IN DB    -34.973999023   -14.974018097   -34.973999023

 EXTREMAL FREQUENCIES
    0.0029297   0.0141602   0.0229492   0.0312500   0.0395508
    0.0473633   0.0556641   0.0639649   0.0717773   0.0800781
    0.0883789   0.0961914   0.1040039   0.1113281   0.1171875
    0.1199999   0.1285000   0.1333711   0.1426601   0.1519375
    0.1612148   0.1700039   0.1787930   0.1870937   0.1921999
    0.2000000   0.2074414   0.2078125   0.2146484   0.2214344
    0.2292969   0.2366211   0.2444336   0.2522461   0.2600586
    0.2678711   0.2756836   0.2834961   0.2913086   0.2991211
    0.3069336   0.3147461   0.3225586   0.3303594   0.3386719
    0.3464844   0.3542969   0.3621094   0.3699219   0.3777344
    0.3855469   0.3938476   0.4016601   0.4094726   0.4172851
    0.4250976   0.4329101   0.4407226   0.4485351   0.4568359
    0.4646484   0.4724609   0.4802734   0.4880859   0.4958984

**********************************************************************
```

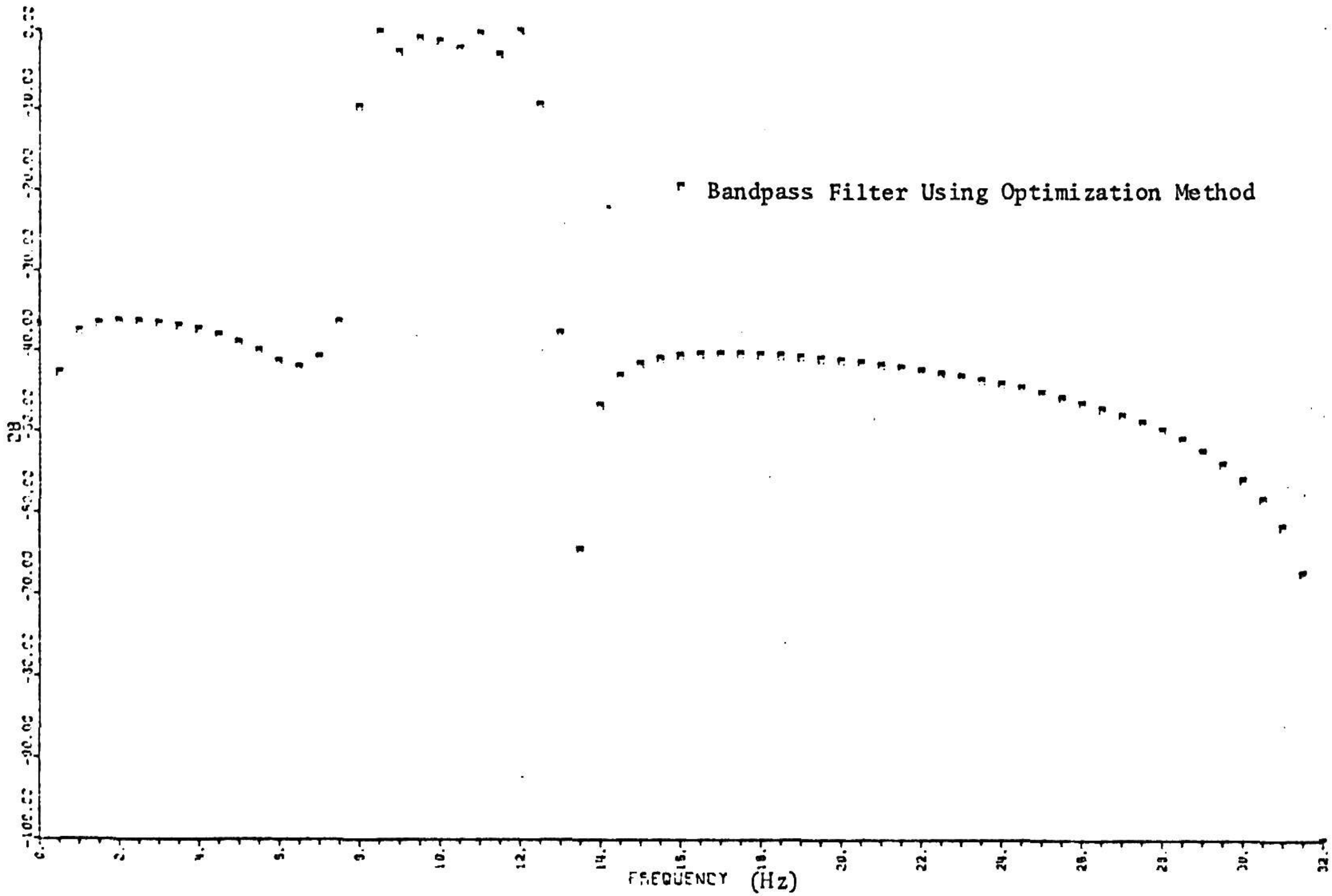


Fig. 2.3-1. Magnitude Characteristics of a 128-point Optional Bandpass Filter

CHAPTER III

DESIGN TECHNIQUES FOR IIR DIGITAL FILTERS

The main objective of this chapter is to discuss the methods available for the design of infinite impulse response (IIR) digital filters. There are two most widely used IIR filter design methods. They are as follows:

(i) The bilinear transformation method, and

(ii) The impulse invariant transformation method.

Some aspects of the above methods are discussed in the sections that follow. Results related to the design of Butterworth digital filters using the bilinear transformation method are included.

3.1. The Bilinear Transformation Method

The bilinear transformation is viewed as a transformation from the s-plane to the z-plane, which allows the conversion of analog poles and zeros into digital poles and zeros. That is, the bilinear transformation provides a simple conformal mapping from the s-plane to the z-plane which is defined as

$$s = \frac{z-1}{z+1} \tag{3.1-1}$$

From Eq. (3.1-1) it follows that the inverse mapping from the z-plane to the s-plane is given by

$$z = \frac{1+s}{1-s} \tag{3.1-2}$$

The nature of the mapping is illustrated in Fig. 3.1-1. We observe that the entire jw axis in the s-plane is mapped onto the unit circle; the left-half s-plane is mapped <u>inside</u> the unit circle in the z-plane and the right half s-plane is mapped outside the z-plane unit-circle. These properties readily follow from Eq. (3.1-2).

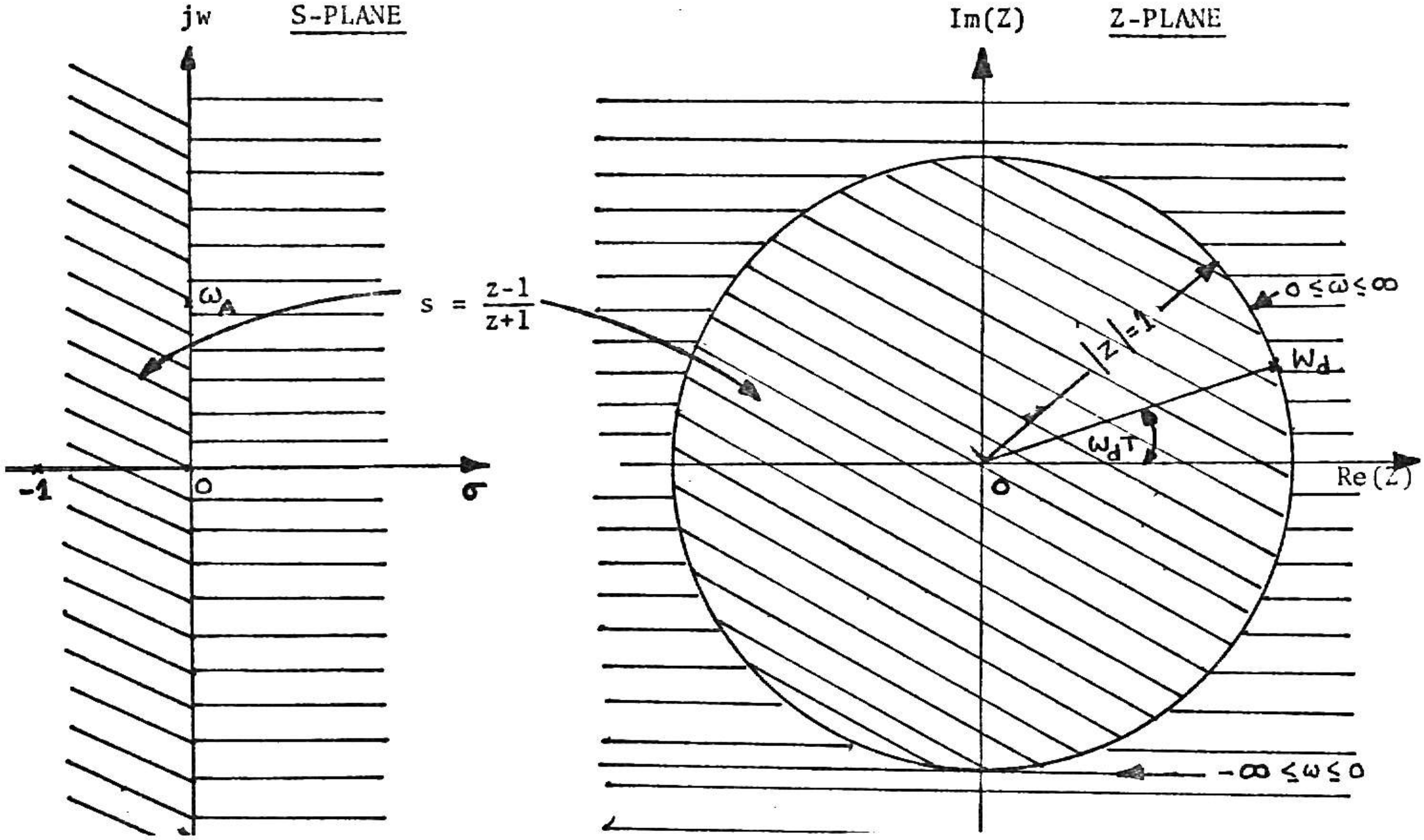


Fig. 3.1-1. The mapping from the s-plane to the z-plane corresponding to the bilinear transformation.

The transfer function of the digital filter H(z) is obtained from the bilinear transformation by making the algebraic substitution of Eq. (3.1-1); i.e.,

$$H(z) = H(s)\Big|_{s = \frac{z-1}{z+1}} \qquad (3.1\text{-}3)$$

For designing IIR filters via the bilinear transformation, it is important to consider the relationship between analog and digital frequencies. This relationship is nonlinear in nature, some aspects of which are discussed in the next section.

3.2. Relationship Between Analog and Digital Frequencies

Let w_A be an analog frequency variable along the jw-axis and let w_D be the corresponding digital frequency variable on the unit circle $|z| = 1$. Now, substitution of $s = jw_A$ and $z = e^{jw_D T}$ in Eq. (3.1-1) yields

$$jw_A = \frac{e^{jw_D T} - 1}{e^{jw_D T} + 1},$$

which can be written as

$$jw_A = \frac{e^{jw_D T} - e^{jw_D T/2} \cdot e^{-jw_D T/2}}{e^{jw_D T} + e^{jw_D T/2} \cdot e^{-jw_D T/2}}$$

That is

$$jw_A = \frac{e^{jw_D T/2} - e^{-jw_D T/2}}{e^{jw_D T/2} + e^{-jw_D T/2}} \tag{3.2-1}$$

From Eq. (3.2-1) it follows that

$$w_A = \tan\left(\frac{w_D T}{2}\right) \tag{3.2-2}$$

or

$$w_D T = 2 \tan^{-1} w_A \tag{3.2-3}$$

Equation (3.2-2) or Eq. (3.2-3) enables a one-to-one correspondence between points on the jw-axis in the s-plane and the unit circle in the z-plane.

3.3 Design Procedure

The following are the steps involved in the design of IIR digital filters using the bilinear transformation.

<u>Step 1.</u> Transfer all the digital filter frequency specifications; i.e., (F_{D1}, F_{D2}, . . ., etc.) on to the jw-axis in the s-plane using the Eq. (3.2-2), i.e.,

$$W_A = \tan\left(\frac{W_D T}{2}\right)$$

<u>Step 2.</u> Design a filter which meets the specifications using the W_A information; that is, determine $\hat{H}(s)$ which is the scaled version of the desired $H(s)$.

<u>Step 3.</u> Substitute $s = \frac{z-1}{z+1}$ in $\hat{H}(s)$ to obtain the corresponding $\hat{H}(z)$.

<u>Step 4.</u> Implement $\hat{H}(z)$.

3.4 Impulse Invariant Transformation

The characteristic property preserved by this transformation is that the impulse response of the digital filter is a sampled version of the impulse response of an analog filter; i.e., the frequency response of the digital filter is an aliased version of the frequency response of the corresponding analog filter, which is discussed next.

3.5 Demonstration of How an Analog Filter is Digitized Using the Impulse Invariant Transformation

To demonstrate how an analog filter is digitized using the impulse invariant transformation, we could assume that the coefficients of an analog filter with Laplace transform (i.e. transfer function) is [1]

$$H(s) = \frac{\sum_{i=0}^{M} b_i s^i}{\sum_{i=1}^{N} a_i s^i} = \frac{\prod_{i=1}^{M} (s + c_i)}{\prod_{i=1}^{N} (s + d_i)} \qquad (3.5\text{-}1)$$

In its partial fraction expansion, the above equation (3.5-1) can be written as

$$H(s) = \sum_{i=1}^{N} \frac{c_i}{s+d_i} \tag{3.5-2}$$

where $c_i = H(s) \cdot (s+d_i)\Big|_{s=-d_i}$ (3.5-3)

and d_i is the location of the ith pole. In writing Eq. (3.5-2) in the given form, we have assumed that in Eq. (3.5-1) the order of numerator M is less than the order of the denominator N and that all the poles of H(s) are simple. The assumption that $M<N$ must be valid for the system to be digitized, otherwise the aliasing in the digital system would be intolerable.

Now, we can write the impulse response of the analog filter h(t) corresponding to Eq. (3.5-1) is of the form

$$h(t) = \sum_{i=1}^{N} c_i \, e^{-d_i t}, \quad t \geq 0 \tag{3.5-4}$$

Hence the corresponding digital impulse response h(nT) can be written as the sampled version of Eq. (3.5-4); i.e.

$$h(nT) = \sum_{i=1}^{N} c_i \, e^{-d_i nT}, \quad n \geq 0 \tag{3.5-5}$$

where T is the sampling interval. Taking the z-transform of Eq. (3.5-5), we obtain

$$H(z) = \sum_{n=0}^{\infty} h(nT) \, z^{-n},$$

Substitution of Eq. (3.5-5) in the above equation results in

$$H(z) = \sum_{n=0}^{\infty} \sum_{i=1}^{N} c_i \, e^{-d_i nT} \, z^{-n} \tag{3.5-6}$$

Now, interchanging the order of summation in Eq. (3.5-6) we get

$$H(z) = \sum_{i=1}^{N} c_i \sum_{n=0}^{\infty} (e^{-d_i T} z^{-1})^n$$

That is

$$H(z) = \sum_{i=1}^{N} \frac{c_i}{1 - e^{-d_i T} z^{-1}} \quad (3.5\text{-}7)$$

Now, by comparing Eq. (3.5-7) and (3.5-1) it is seen that H(z) is obtained from H(s) by using the following mapping relation which is valid *for the case of simple poles:*

$$\frac{1}{s+d_i} \longrightarrow \frac{1}{1 - z^{-1} e^{-d_i T}} \quad (3.5\text{-}8)$$

The case when d_i is complex-valued in relatively more tedious to analyze. The desired mapping can be shown to be as follows [1]:

$$H_1(s) = \frac{s+\sigma}{s^2+2\sigma s+\sigma^2+w^2} \longrightarrow \frac{1 - z^{-1} e^{-\sigma T} \cos wT}{1 - 2z^{-1} e^{-\sigma T} \cos wT + z^{-2} e^{-2\sigma T}} \quad (3.5\text{-}9)$$

and

$$H_2(s) = \frac{w}{s^2+2\sigma s+\sigma^2+w^2} \longrightarrow \frac{z^{-1} e^{-\sigma T} \sin wT}{1-2z^{-1} e^{-\sigma T} \cos wT + z^{-2} e^{-2\sigma T}} \quad (3.5\text{-}10)$$

where the impulse responses corresponding to $H_1(s)$ and $H_2(s)$ are given by

$$h_1(t) = e^{-\sigma t} \cos wt \quad (3.5\text{-}11)$$

and

$$h_2(t) = e^{-\sigma t} \sin wt, \quad t \geq 0 \quad (3.5\text{-}12)$$

3.6 Analog Butterworth Filters

In this section we review some fundamentals of lowpass, highpass and bandpass filters.

Lowpass Filters

We consider for two cases as follows:

(i) Cutoff frequency $w_c = 1$ rad/sec, and

(ii) Cutoff frequency $w_c \neq 1$

Case 1: The transfer function $H_n(s)$ of the n-th order lowpass Butterworth filter is such that

$$\left|H_n(jw)\right|^2 = \frac{1}{1 + w^{2n}}, \quad n = 1,2,3,. . . . \tag{3.6-1}$$

A plot of $\left|H_n(jw)\right|^2$ is shown in Fig. (3.8-1) from which it is evident that the bandwidth of all filters is 1 regardless of n.

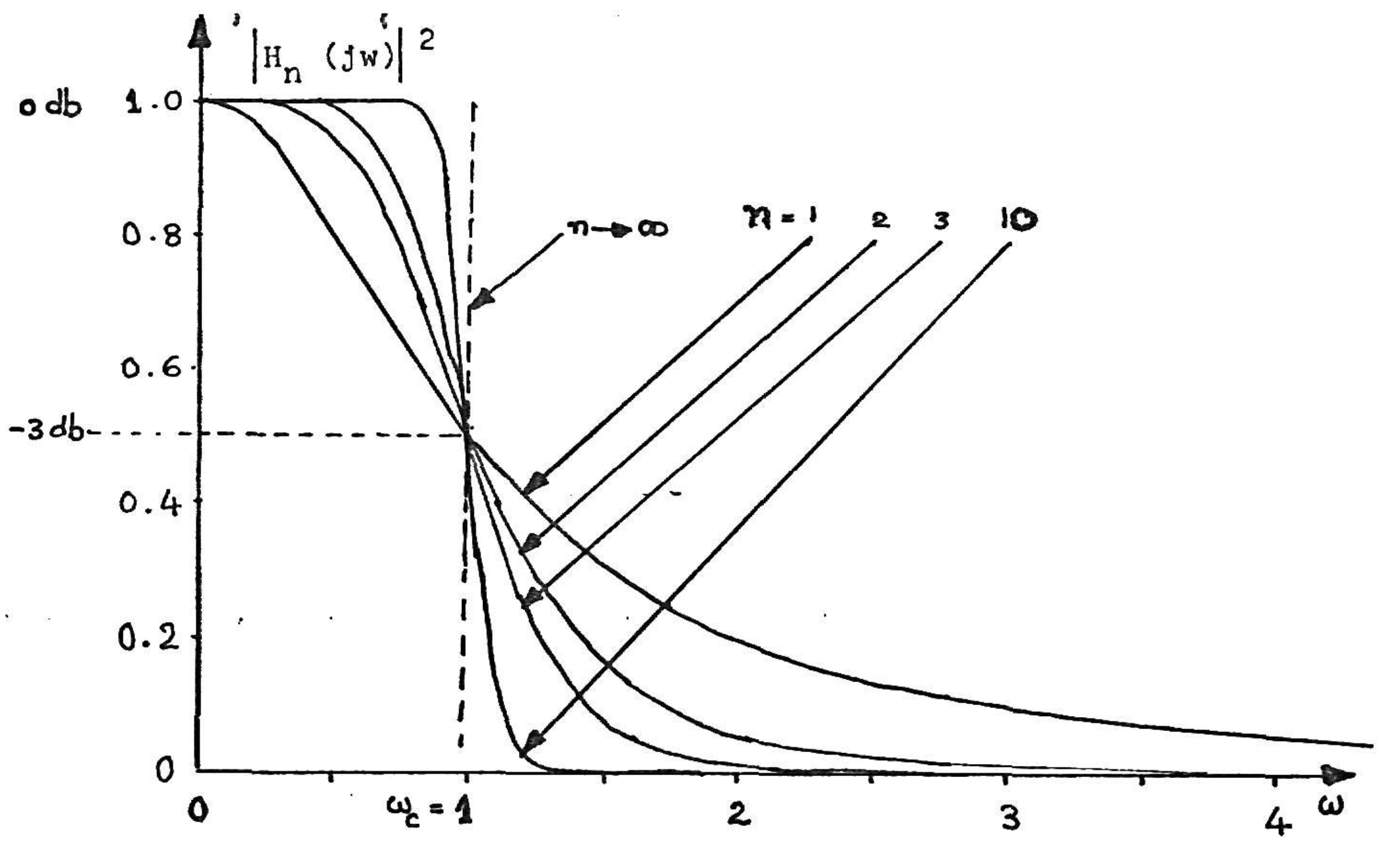


Fig. 3.6-1. Magnitude characteristic of a Butterworth lowpass filter with cutoff frequency $w_c = 1$.

For the purposes of illustration, the transfer functions corresponding to n = 1,2,3 and 4 are summarized in Table 3.6-1.

Case 2: In this case we merely replace s by s/w_c which corresponds to a frequency scaling process.

Table 3.6-1. Transfer functions of some lowpass filters.

n	$H_n(s)$
1.	$1/s+1$
2.	$1/s^2 + \sqrt{2}\, s + 1$
3.	$1/s^3 + 2s^2 + 2s + 1$
4.	$1/s^4 + 2.61\, s^3 + 3.41s^2 + 2.61s + 1$

Highpass Filters. Again we consider for two cases, which correspond to $w_c = 1$ and $w_c \neq 1$ respectively.

Case 1: The transformation from lowpass to highpass filters is achieved by replacing $\frac{1}{s}$ by s in the expression for $H_n(s)$ in Table 3.6-1. We denote the corresponding highpass transfer function by $G_n(s)$; i.e.,

$$G_n(s) = H_n(\tfrac{1}{s}) \qquad (3.6\text{-}2)$$

Fig. 3.6-2 shows the corresponding magnitude characteristics for n = 1,2,3, etc.

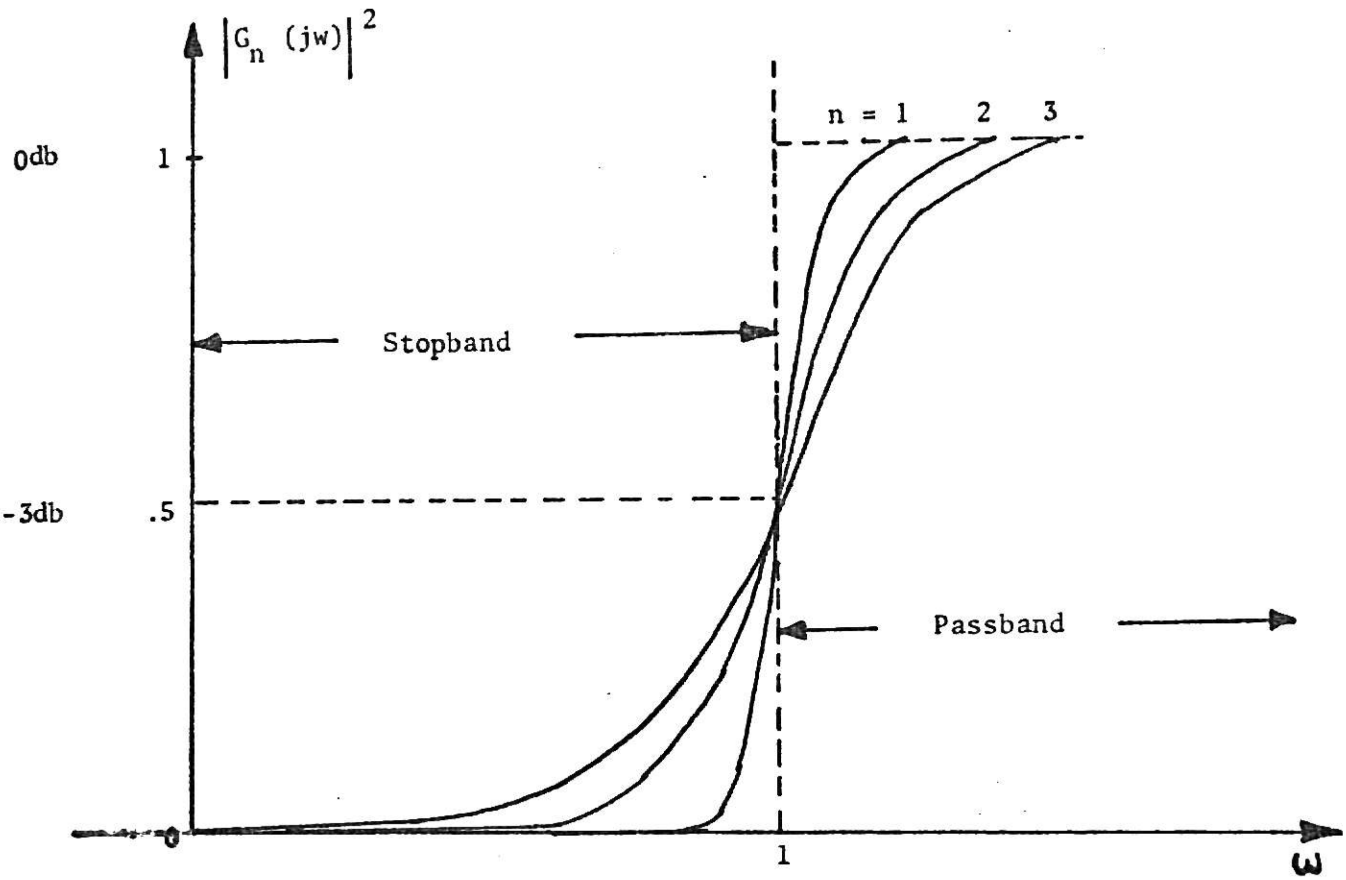


Fig. 3.6-2. Magnitude characteristic of a Butterworth highpass filter with cutoff frequency $w_c = 1$.

Case 2: Here, we replace Eq. (3.6-2) by $G_n\left(\frac{s}{w_c}\right)$ to obtain $\hat{G}_n(s)$; i.e.,

$$\hat{G}_n(s) = G_n(s)\Big|_{s = \frac{s}{w_c}} \tag{3.6-3}$$

Bandpass Filters: The two cases of interest with respect to bandpass filters are these where the filter bandwidth $(w_h - w_1) = 1$ and $(w_h - w_1) \neq 1$ where w_h and w_1 are the upper and lower cutoff frequencies respectively.

To convert lowpass filter to bandpass filter, we replace s by $\frac{(s^2 + w_0^2)}{s}$; where w_0 is the radian center frequency of the bandpass filter, and is given by $w_0 = \pm\sqrt{w_h\, w_1}$.

If $L_n(s)$ denotes the corresponding transfer function of an nth-order Butterworth bandpass filter, then it is given by

$$L_n(s) = H_n(s)\Big|_{s = \frac{s^2 + w_0^2}{s}} \tag{3.6-4}$$

Fig. 3.6-3 shows the corresponding magnitude characteristic of the above bandpass filter

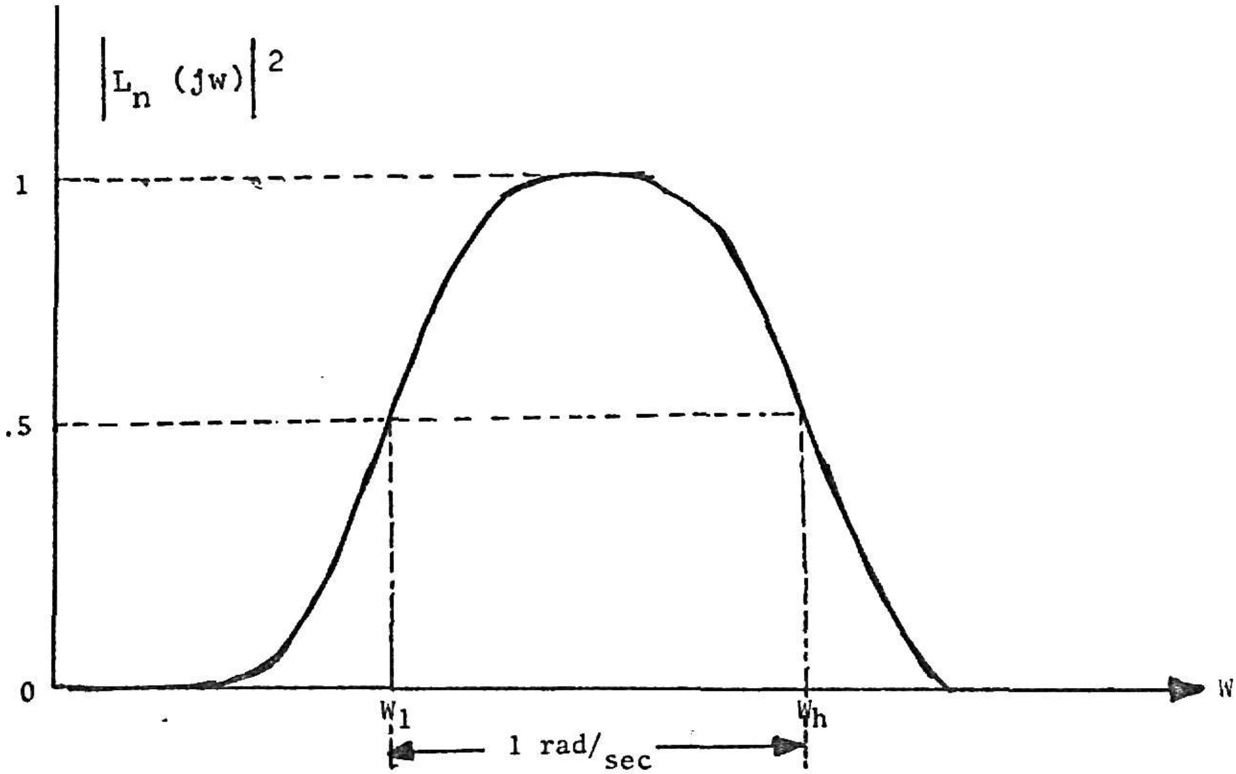


Fig. 3.6-3. Magnitude characteristic of a Butterworth bandpass filter

Case 2: Here we let $w_h - w_1 = B$, where B is the filter bandwidth. In this case we replace $L_n(s)$ by $\hat{L}_n(s)$ where

$$\hat{L}_n(s) = L_n(s)\Big|_{s = \frac{s}{B}} \tag{3.6-5}$$

Next, we present a design example.

3.7. A Design Example.

(a) Starting with the third-order Butterworth transfer function

$$H(s) = \frac{1}{s^3+2s^2+2s+1}$$

design a digital bandpass filter with the following specifications.

Lower cutoff frequency, $F_{Dl} = 4$ Hz

Upper cutoff frequency, $F_{Dh} = 8$ Hz

Sampling frequency, $F_s = 62.5$ sps

(b) Obtain the difference equations which enable a software implementation of the above bandpass filter.

(c) Plot the corresponding magnitude characteristic.

Solution: The first step is to transform the frequencies w_{Dl} and w_{Dh}, which in this problem are

$$w_{Dl} = 2\pi F_{Dl} = 8\pi$$

$$\text{i.e., } w_{Dl}T = 0.128\pi \text{ radians} \tag{3.7-1}$$

Similarly

$$w_{Dh} = 16\pi$$

and

$$w_{Dh}T = 0.256\pi \text{ radians} \tag{3.7-2}$$

The corresponding analog frequencies are [see Eq. (3.2-2)]

$$\hat{w}_{Al} = \tan[w_D T/2] = 0.2038 \tag{3.7-3}$$

and

$$\hat{w}_{Ah} = \tan[w_D T/2] = 0.4253 \tag{3.7-4}$$

Hence, the corresponding bandwidth is given by

$$B = \hat{w}_{Ah} - \hat{w}_{Al}$$

By substituting the values of $\hat{w}_{Ah}$ and $\hat{w}_{Al}$ in the above equation, we obtain

$$B = .2215 \tag{3.7-5}$$

Again, we know that

$$\hat{w}_{Al}\,\hat{w}_{Ah} = B^2\, w_{Al}\, w_{Ah}$$

or

$$\hat{w}_{Al}\,\hat{w}_{Ah} = B^2\, w_0^{\,2}$$

Thus, if $\hat{w}_0$ denote the center frequency, then

$$\hat{w}_0^{\,2} = \hat{w}_{Al}\,\hat{w}_{Ah},$$

which yields

$$\hat{w}_0 = .294408 \tag{3.7-6}$$

Next, the given Butterworth lowpass filter transfer function is transformed to obtain a corresponding bandpass filter transfer function via the frequency transformation

$$s \longrightarrow \left(\frac{s^2 + w_0^{\,2}}{s}\right).$$

Thus we have

$$L_3(s) = H_3(s)\Big|_{s = \frac{s^2 + w_0^{\,2}}{s}},$$

which yields

$$L_3(s) = \frac{s^3}{s^6+2s^5+s^4(2+3w_0^{\,2})+s^3(1+4w_0^{\,2})+s^2(3w_0^{\,4}+2w_0^{\,2})+2sw_0^{\,4}+w^6} \tag{3.7-7}$$

The normalized bandwidth of the filter corresponding to $L_3(s)$ is unity.

To account for a bandwidth B, frequency scaling is used; i.e., s in $L_3(s)$ is replaced by s/B to obtain

$$\hat{L}_3(s) = \frac{s^3B^3}{s^6+2s^5B+s^4B^2(2+3w_0^2)+s^3B^3(1+4w_0^2)+s^2B^4(3w_0^4+2w_0^2)+2sB^5w_0^4+w_0^6B^6} \tag{3.7-8}$$

Now, by substituting the Equations (3.7-5) and (3.7-6) into Eq. (3.1-8), we obtain

$$\hat{L}_3(s) = \frac{.010867s^3}{s^6+.443s^5+.358153s^4+.087662s^3+.031043s^2+.003328s+.000651} \tag{3.7-9}$$

The third step is to obtain $\hat{L}_3(z)$, corresponding to $\hat{L}_3(s)$ using the bilinear transformation; i.e.,

$$\hat{L}_3(z) = \hat{L}_3(s)\Big|_{s = \frac{z-1}{z+1}}$$

$$\hat{L}_3(z) = \frac{a_6z^6+a_4z^4a_2z^2+a_0}{b_6z^6+b_5z^5+b_4z^4+b_3z^3+b_2z^2+b_1z+b_0} \tag{3.7-10}$$

where

$a_6 = .0108$, $a_4 = -0.03261$, $a_2 = .03261$, $a_0 = -.01087$

$b_6 = 1.92384$, $b_5 = -8.40890$, $b_4 = 16.58922$, $b_3 = -18.6786$,

$b_2 = 12.6518$, $b_1 = -4.89162$ and $b_0 = .85586$

The last step is concerned with obtaining a software implementation of the transfer function $\hat{L}_3(z)$; i.e., Eq. (3.7-10), which defines the desired filter. It is straightforward to show that the following difference equations enable one to compute the output y(nT), n = 0,1,2,. . . of the filter corresponding to the input x(nT), n = 0,1,2,. . . .

$$y(nT) = Kw(nT) - 3Kw[(n-2)T] + 3Kw[(n-4)T] - Kw[(n-6)T]$$

$$n = 1,2,3,\ldots \qquad (3.7\text{-}11)$$

where

$$K = .01087$$

and

$$\begin{aligned} w(nT) = {} & 4.37089\, w[(n-1)T] - 8.62297\, w[(n-2)T] + 9.70902\, w[(n-3)T] \\ & -6.57637\, w[(n-4)T] + 2.54263\, w[(n-5)T] - .44417\, w[(n-6)T] \\ & + .51979\, x(nT) \end{aligned} \qquad (3.7\text{-}12)$$

Fig. 3.7-1 shows the block diagram representation of Eqs. (3.7-11) and (3.7-12). The magnitude and phase characteristics of the desired filter are obtained by using the FORTRAN program which is given in Appendix V. The specific card input to the program is as follows:

Card 1: 06, 06, 049, 10, 0.0, 0.5

Card 2: 0.01087, 0.0, -0.03261, 0.0, 0.03261, 0.0, -0.01087

Card 3: 1.92384, -8.40890, 16.58922, -18.6786, 12.65188, -4.89162, 0.85586

Table 3.7-1 provides a list of the magnitude and phase values, as they are obtained from the program in the form of an output.* Again, corresponding plots of these characteristics are shown in Fig. 3.7-2, from which it follows that the filter essentially has a linear phase in the passband (i.e., 4-8 Hz region), where the normalized frequency values corresponding to 4 and 8 Hz are .064 and .128 respectively.

*The variable "ETA" in Table 3.7-1 denotes normalized frequency; that is ETA = f/Fs.

Table 3.7-1. A Listing of the Magnitude and Phase Values.

ETA	MAGNITUDE	PHASE(RAD)
0.000000	0.719601E-04	0.000000
0.104167E-01	0.626866E-03	-1.62325
0.208333E-01	0.550677E-02	-1.91169
0.312500E-01	0.228114E-01	-2.14344
0.416667E-01	0.741712E-01	-2.44206
0.520833E-01	0.226257	-2.90564
0.625000E-01	0.627083	2.52554
0.729166E-01	0.969037	1.37027
0.833333E-01	1.00573	0.515555
0.937499E-01	1.00798	-0.156077
0.104167	1.00557	-0.781756
0.114583	0.965356	-1.44936
0.125000	0.784447	-2.16721
0.135417	0.517730	-2.76552
0.145833	0.321637	3.11611
0.156250	0.205920	2.85109
0.166667	0.138012	2.66564
0.177083	0.963208E-01	2.52793
0.187500	0.694444E-01	2.42071
0.197917	0.513635E-01	2.33416
0.208333	0.387601E-01	2.26229
0.218750	0.297141E-01	2.20125
0.229167	0.230623E-01	2.14847
0.239583	0.180721E-01	2.10212
0.250000	0.142654E-01	2.06090
0.260417	0.113208E-01	2.02385
0.270833	0.901648E-02	1.99021
0.281250	0.719596E-02	1.95941
0.291667	0.574640E-02	1.93101
0.302083	0.458504E-02	1.90464
0.312500	0.365018E-02	1.88000
0.322917	0.289517E-02	1.85686
0.333333	0.228424E-02	1.83500
0.343750	0.178967E-02	1.81426
0.354167	0.138970E-02	1.79449
0.364583	0.106710E-02	1.77558
0.375000	0.808078E-03	1.75740
0.385417	0.601486E-03	1.73987
0.395833	0.438241E-03	1.72289
0.406250	0.310361E-03	1.70640
0.416667	0.213128E-03	1.69031
0.427083	0.139816E-03	1.67455
0.437500	0.864844E-04	1.65901
0.447917	0.493098E-04	1.64355
0.458333	0.249510E-04	1.62781
0.468750	0.104386E-04	1.61062
0.479167	0.308261E-05	1.58560
0.489583	0.393357E-06	1.46779
0.500000	0.468572E-07	0.899586E-14

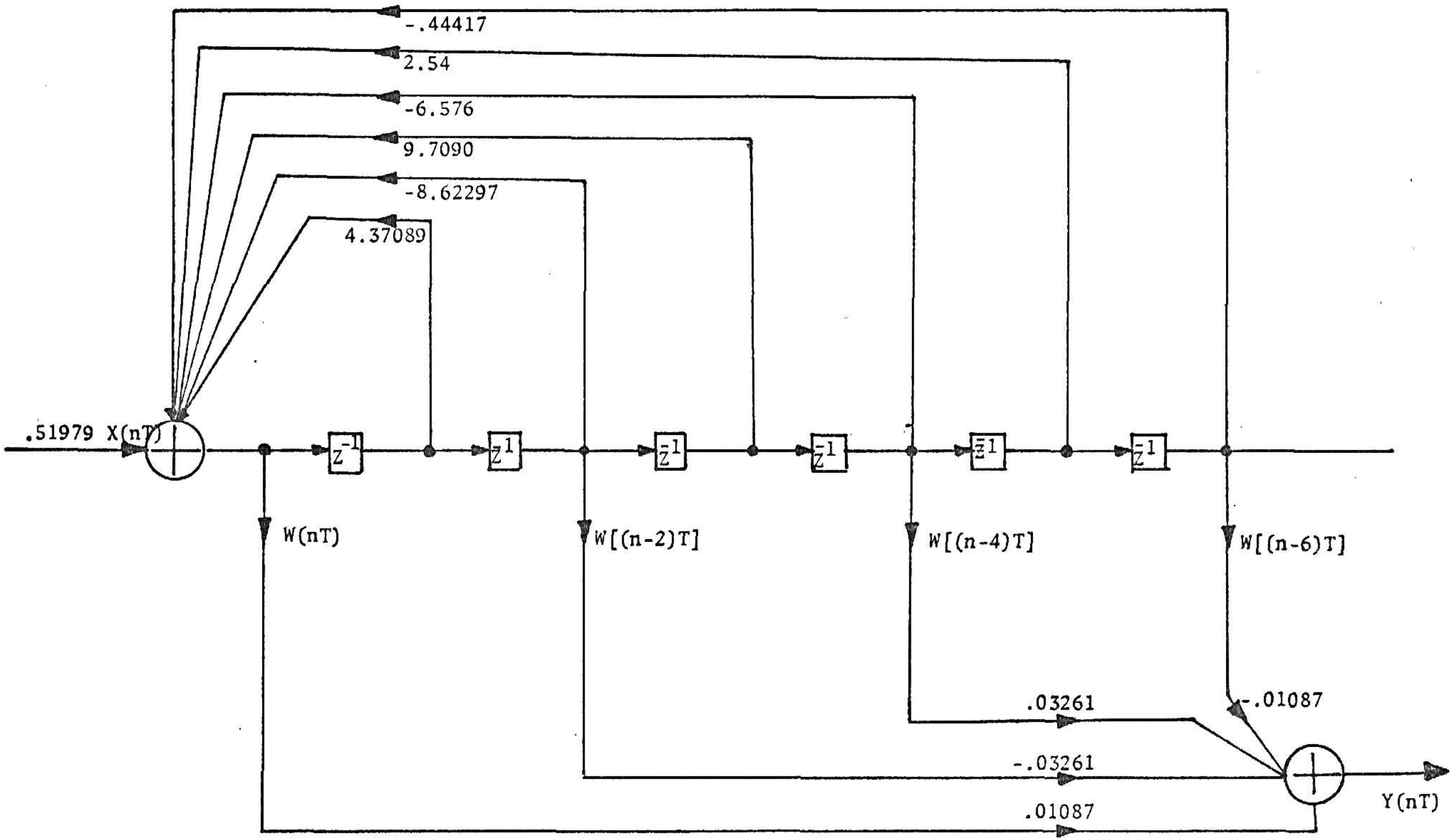


Fig. 3.7-1. Block diagram representation of the difference Equations in (3.7-11) and (3.7-12)

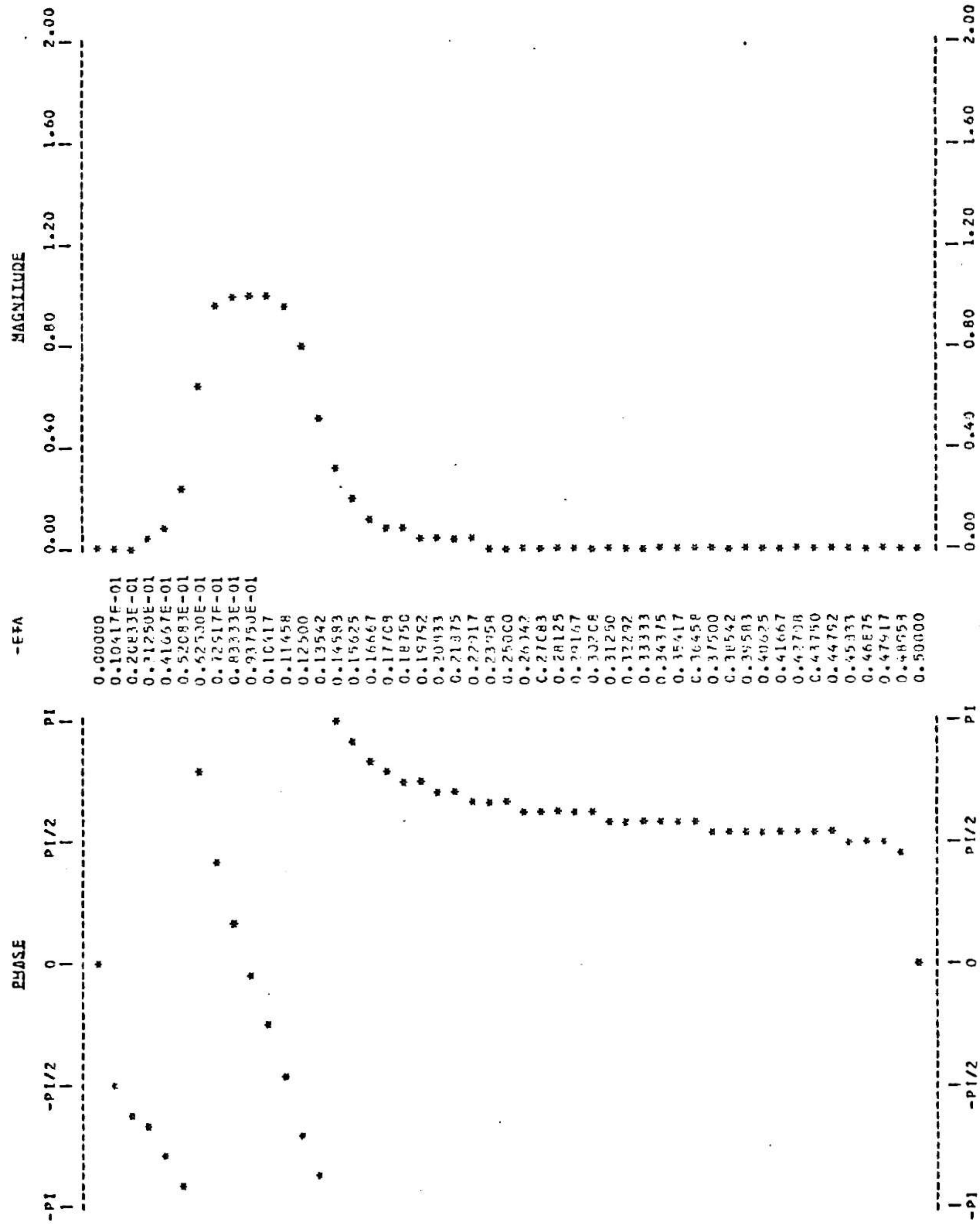


Fig. 3.7-2. Magnitude and Phase Characteristics of a Third-order 4-8 Hz Butterworth Digital Filter

CHAPTER IV

SOME EXPERIMENTAL RESULTS

The main objective of this chapter is to present some experimental results related to electroencephalograph (EEG) digital signal processing [3]. We domonstrate that the IIR filter design technique (via the bilinear transform) which was discussed in Chapter III can be used effectively to extrapolate the THETA and ALPHA components of EEG data. This is achieved by using Butterworth bandpass digital filters of the type discussed in Section 3.7. The passbands associated with the THETA and ALPHA components are 4-8Hz and 8-12 Hz respectively. In what follows, we refer to these filters as the THETA and ALPHA filters respectively.

4.1 Results

The desired THETA and ALPHA filters consist of a cascade of <u>three identical sixth-order bandpass Butterworth filters</u> whose transfer function are given as follows:

$$H(z) = \frac{a_6z^6+a_4z^4+a_2z^2+a_0}{b_6z^6+b_5z^5+b_4z^4+b_3z^3+b_2z^2+b_1z+b_0} \qquad (4.1\text{-}1)$$

The values of the a_i and b_i in Eq. (4.1-1) for the 4-8 Hz and 8-12 Hz filters are summarized in Table 4.1-1.

The overall magnitude characteristics of the THETA and ALPHA filters are shown in Fig. 4.1-1. We remark that these digital bandpass filters have an almost linear phase function in the passband. This desirable property is illustrated in Fig. 4.1-2, which shows plots of the magnitude

Table 4.1-1. Values of a_i and b_i for H(z) in Eq. (4.1-1).

	THETA	ALPHA
a_6	.01087	.0183
a_4	-.03261	-0.0549
a_2	.03261	0.0549
a_0	.01087	-0.0183
b_6	1.92384	3.24045
b_5	-8.40890	-9.215604
b_4	16.58922	15.941314
b_3	18.6786	-16.618736
b_2	12.65188	12.181134
b_1	-4.89162	-5.361084
b_0	.85586	1.441678

and phase characteristics associated with the transfer function H(z) in Eq. (4.1-1) for the 8-12 Hz case. In Fig. 4.1-2, the magnitude and phase characteristics have been plotted with respect to the normalized frequency variable η = F/Fs, where F_S denotes the sampling frequency, which was considered as 64 sps for designing the above filters.

A typical set of THETA and ALPHA filter outputs for the purposes of illustration are shown in Fig. 4.1-3. The amplitudes of the waveforms shown in Fig. 4.1-3 are expressed in microvolts, and were obtained by scaling the outputs of the THETA and ALPHA filters by a factor of 380×10^{-6}, which accounts for the overall processing gain.

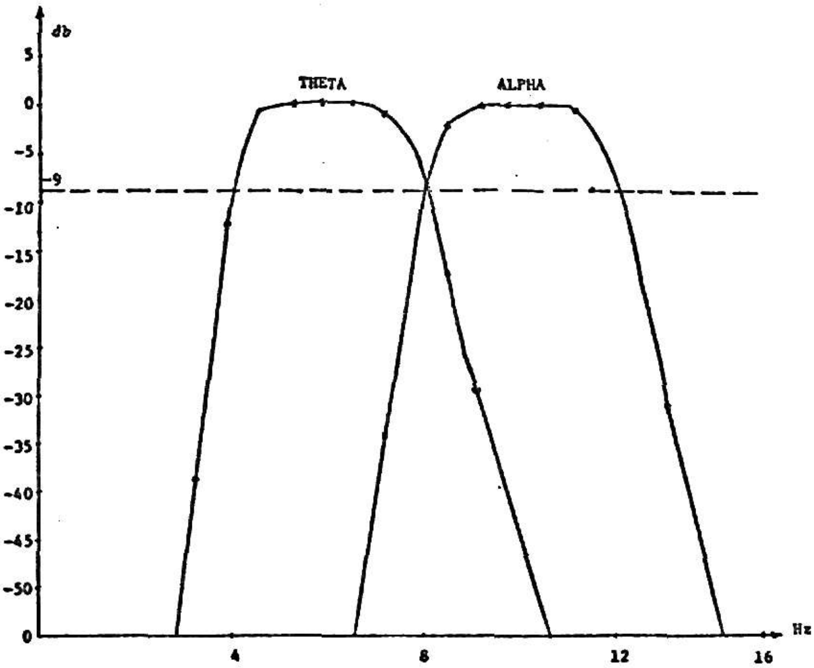


Fig. 4.1-1. Magnitude characteristics of THETA and ALPHA filters used for digital filtering.

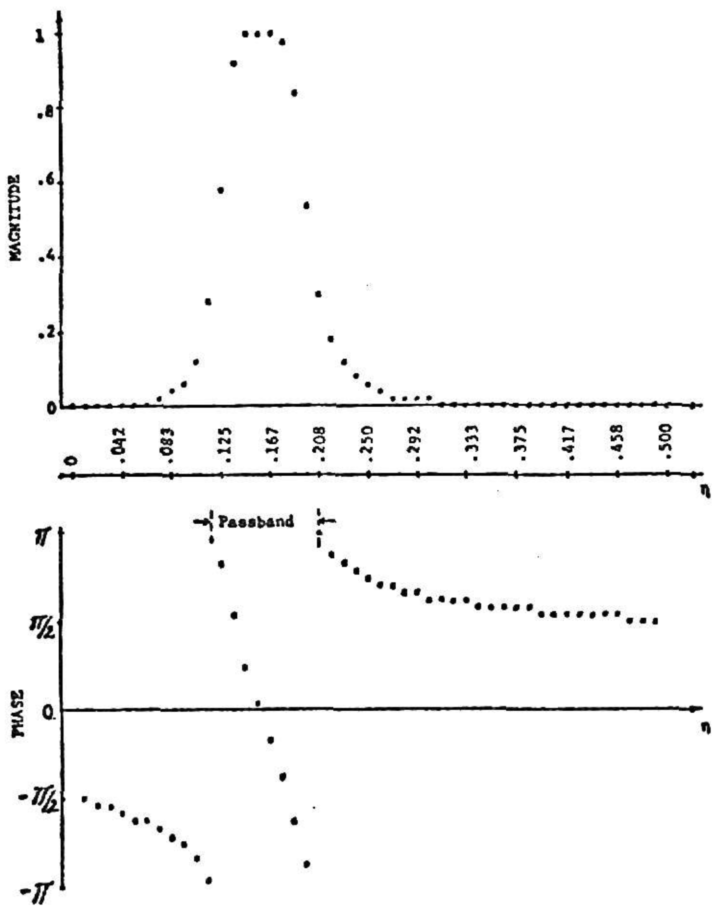


Fig. 4.1-2. Magnitude and phase characteristics of a sixth-order 8-12 Hz Butterworth digital filter.

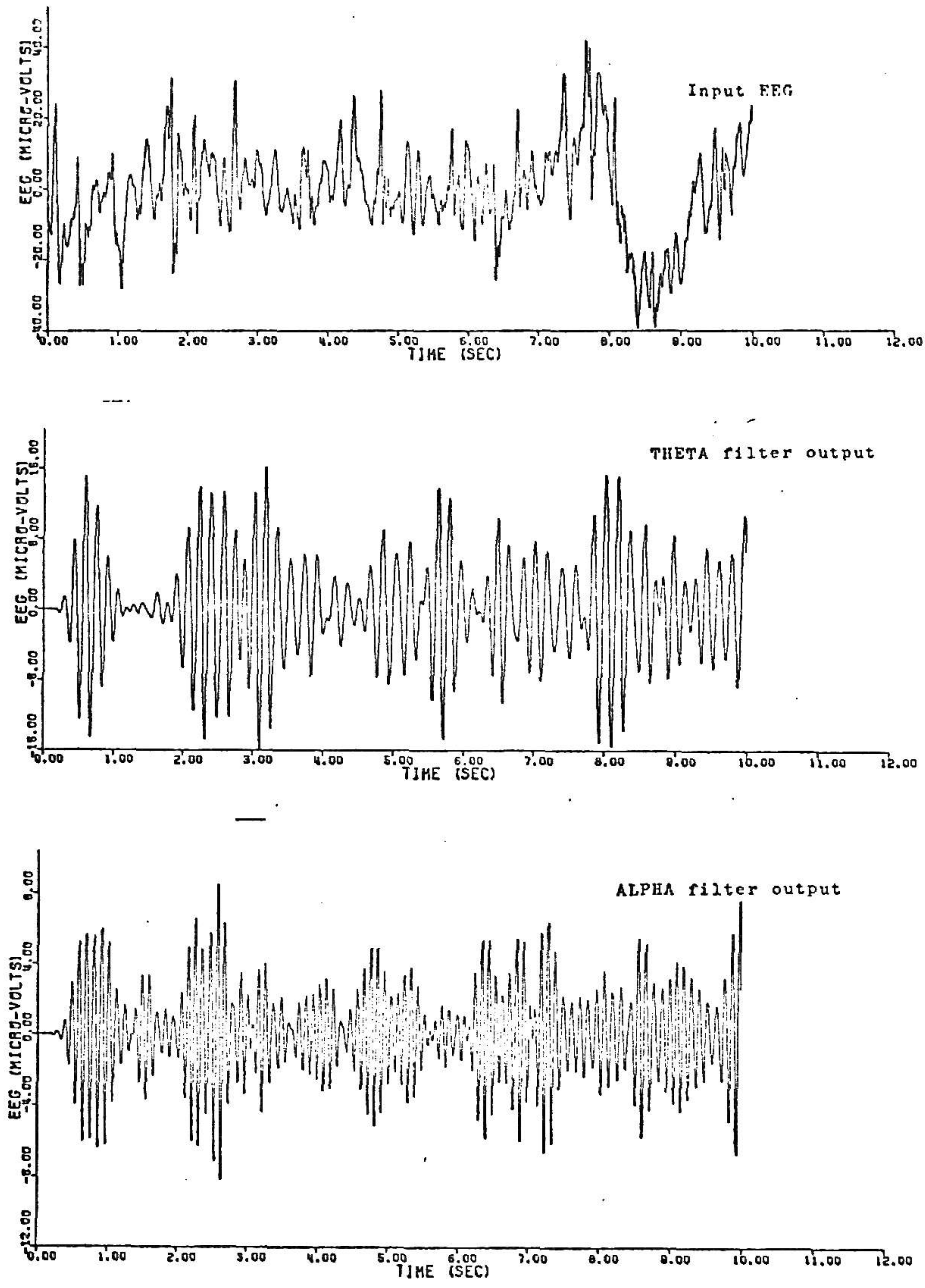


Fig. 4.1-3. A typical EEG segment and the corresponding THETA and ALPHA filter outputs.

CHAPTER V

SOME CONCLUDING REMARKS

Among the various methods that are available for the design of FIR digital filters, the window method is the most convenient for one to use in situations where approximations to ideal filters (i.e., lowpass, highpass, bandpass and bandstop) are desired. However, in cases where specific filtering characteristics are necessary, the optimization method is the most useful. A program that enables one to do so is given in Appendix IV, and was taken from a recent book by Rabiner and Gold [1].

In the case of FIR filters, the bilinear transformation method is most effective. This method has the advantage that all the work that has been done in connection with analog filters can be carried over to digital filters by virtue of the bilinear transformation.

It is interesting to note that until the advent of the fast Fourier transform (FFT) algorithm, FIR filters were generally felt to be computationally unfeasible because exceedingly long sequences were required to approximate sharp cutoff filters adequately. The computational efficiency of the FFT made the techniques of fast convolution feasible. Some currently available FIR filters are in fact competitive even with sharp cutoff IIR filter designs.

In conclusion, we remark that it is not possible for one to select either FIR or IIR design techniques and claim that it will be applicable in all cases of interest. Thus it is especially important that the designer fully understand the nature of the various design procedures and also be aware of their advantages and limitations.

REFERENCES

1. Lawrence R. Rabiner and Bernard Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., 1975.

2. Philip F. Panter, Modulation Noise, and Spectral Analysis, McGraw-Hill, Inc., 1965.

3. Nasir Ahmed, etc., "On the Digital Processing of EEG Data," Proc., 1976 International Symposium on Electromagnetic Compatibility, Washington; July 1976.

4. Samuel D. Stearns, Digital Signal Analysis, Hayden, Inc., 1975.

APPENDICES

Appendix I

```
      SUBROUTINE FFT(X,N,INV)
C***********************************************************************
C     THIS PROGRAM IMPLEMENTS THE FFT ALGORITHM TO COMPUTE THE DISCRETE
C     FOURIER COEFFICIENTS OF A DATA SEQUENCE OF N POINTS
C     CALLING SEQUENCE FROM THE MAIN PROGRAM:
C     CALL FFT(X,N,INV)
C           N:NUMBER OF DATA POINTS
C           DECLARE II AS-- COMPLEX X(512)
C         X:COMPLEX ARRAY CONTAINING THE DATA SEQUENCE.IN THE END DFT
C           COEFF. ARE RETURNED IN THE ARRAY.MAIN PROGRAM SHOULD
C        INV: FLAG FOR INVERSE
C          INV=0 FOR FORWARD TRANSFORM
C          INV=1 FOR INVERSE TRANSFORM
C***********************************************************************
      COMPLEX X(512),W,T,CMPLX
C
C     CALCULATE THE NO. OF ITERATIONS (LOG. N TO THE BASE 2)
C
      ITER=0
      IREM=N
   10 IREM=IREM/2
      IF (IREM.EQ.0) GO TO 20
      ITER=ITER+1
      GO TO 10
   20 CONTINUE
      SIGN=-1
      IF (INV.EQ.1) SIGN=1.
      NXP2=N
      DO 50 IT=1,ITER
C
C     COMPUTATION FOR EACH ITERATION
C     NXP:NUMBER OF POINTS IN A PARTITION
C     NXP2:NXP/2
C
      NXP=NXP2
      NXP2=NXP/2
      WPWR=3.141592/FLOAT(NXP2)
      DO 40 M=1,NXP2
C
C     CALCULATE THE MULTIPLIER
C
      ARG=FLOAT(M-1)*WPWR
      W=CMPLX(COS(ARG),SIGN*SIN(ARG))
      DO 40 MXP=NXP,N,NXP
C
C     COMPUTATION FOR EACH PARTITION
C
      J1=MXP-NXP+M
      J2=J1+NXP2
      T=X(J1)-X(J2)
      X(J1)=X(J1)+X(J2)
   40 X(J2)=T*W
   50 CONTINUE
C
C     UNSCRAMBLE THE BIT REVERSED DFT COEFFS.
C
      N2=N/2
      N1=N-1
      J=1
      DO 65 I=1,N1
      IF(I.GE.J) GO TO 55
      T=X(J)
      X(J)=X(I)
      X(I)=T
   55 K=N2
   60 IF(K.GE.J) GO TO 65
      J=J-K
      K=K/2
      GO TO 60
   65 J=J+K
      IF (INV.EQ.1) GO TO 75
      DO 70 I=1,N
   70 X(I)=X(I)/FLOAT(N)
   75 CONTINUE
      RETURN
      END
```

APPENDIX II

This Appendix provides a listing of a FORTRAN program which enables me to determine the magnitude and phase characteristics of a digital filter whose impulse response is specified. Provision is made for multiplying a given impulse response by a Hamming window sequence prior to computing its magnitude and phase characteristics. The FFT subroutine referred to in the program is listed in Appendix I.

```
$JOB
      DIMENSICN RX(128),R(128),IM(128),Y(128),MAG(128),DB(128)
      CCMPLEX X(512),CMPLX
      REAL IM,MAG
      PI=4.*ATAN(1.)
      N=128
      RMAX=C.CC782C9
      DO 50 I=1,128
      IF(I.EC.65.) GC TC 5C
      Y(I)=2.*SIN(2.*PI*(I-65.)/32.)*CCS(2.*PI*(I-65.)*(96.**.5)/64.)/
     X(PI*(I-65.))
C*****************************************************************************
C     THE NEXT THREE CARCS ARE FOR HAMMING WINDOWING
C     FOR RECTANGULAR WINDCWING REMOVE THESE THREE CARDS
C     ALPHA=C.54
C     W=ALPHA+(1.-ALPHA)*CCS(2*PI*(I-65)/N)
C     Y(I)=Y(I)*W
C*****************************************************************************
      ALPHA=C.54
      W=ALPHA+(1.-ALPHA)*CCS(2*PI*(I-65)/N)
      Y(I)=Y(I)*W
      X(I)=CMPLX(Y(I),0.0)
   50 CONTINUE
      Y(65)=0.125
      X(65)=CMPLX(C.125,C.C)
      PRINT 18C,(I,Y(I),I=1,128)
  180 FCRMAT(' ',30X,'Y(',I3,')=',F12.8)
      CALL FFT (X,N,0)
      CO 120 I=1,128
      R(I)=REAL(X(I))
      IM(I)=AIMAG(X(I))
      MAG(I)=CABS(X(I))
  120 CB(I)=20.0*ALCG1C(MAG(I)/RMAX)
      PRINT 14C
  140 FCRMAT ('1',T50,'FFT CCEFFICIENTS')
      PRINT 15C,(I,R(I),IM(I),MAG(I),CB(I),I=1,128)
      PUNCH 160,(CB(I),I=1,65)
  160 FCRMAT(8F10.5)
  150 FORMAT(' ',30X,'CX(',I3,')=',F9.6,'+',F9.6,'I',10X,'MAGNITUDE=',
     XF9.7,10X,'CB=',F9.4)
      STCP
      END
```

APPENDIX III

This Appendix provides a listing of a FORTRAN program used to obtain CALCOMP plots corresponding to given data. It has been used in this report to plot the various magnitude characteristics of the digital filters that have been discussed.

```
C       LINEAR PLOTTING
C       NP IS THE NUMBER OF PLOTS
C       N IS THE NUMBER OF POINTS
C       NC IS THE NUMBER OF CURVES
        DIMENSION IBUF(8000)
        DIMENSION X(67),Y(67),INTQ(8)
        SC=0.5
        CALL PLOTS(IBUF,8000)
        CALL FACTOR(0.5)
        READ 11,NP
     11 FORMAT(I3)
        READ 12,N,NC
     12 FORMAT (2I3)
        READ 10,(INTQ(I),I=1,NC)
     10 FORMAT(10I3)
        DO 100 IP=1,NP
        DO 13  I=1,N
     13 X(I)=FLOAT(I-1)*SC
        CALL PLOT(0.,-11.,3)
        CALL PLOT(0.,-10.,-3)
        X(N+1)=0.
        X(N+2)=2.*SC
        Y(N+1)=-130.
        Y(N+2)=13.
        CALL DAXIS(0.,0.,16.,0.,.25,-4)
        CALL SAXIS(0.,2.,0,+3,-4,'FREQUENCY',9)
        CALL AXIS(0.,0.,'DB',+2,10.,90.,-130.,13.)
        DO 14  IC=1,NC
        READ 15,(Y(I),I=1,N)
     15 FORMAT(8F10.6)
        LTYP=INTQ(IC)
        CALL LINE(X,Y,N,1,-1,LTYP)
     14 CONTINUE
        CALL SYMBOL(10.,9.07,.08,11,0.,-1)
        CALL SYMBOL(10.5,9.,.14,
      1'BANDPASS FILTER OF 4,8HZ WITH RECTANGULAR WINDOWING',0.,52)
        CALL SYMBOL(10.,8.57,.08,1,0.,-1)
        CALL SYMBOL(10.5,8.5,.14,
      1'BANDPASS FILTER OF 4,8HZ WITH HAMMIND WINDOWING',0.,49)
        CALL PLOT(22.,0.,-3)
    100 CONTINUE
        CALL PLOT(0.,0.,999)
        STOP
        END
```

APPENDIX IV

This Appendix gives a FORTRAN program that is capable of designing a wide variety of optional (minimax) FIR filters including lowpass, highpass, bandpass, and bandstop filters as well as differentiators and Hilbert transformers. It has been taken from the recent text by Rabiner and Gold [1] and adapted to the IBM 370/158 computer.

```
C      PROGRAM FOR THE DESIGN OF LINEAR PHASE FINITE IMPULSE
C      RESPONSE (FIR) FILTERS USING THE REMEZ EXCHANGE ALGORITHM
C      JIM MCCLELLAN, RICE UNIVERSITY, APRIL 13, 1973
C      THREE TYPES OF FILTERS ARE INCLUDED--BANDPASS FILTERS
C      DIFFERENTIATORS, AND HILBERT TRANSFORM FILTERS
C
C      THE INPUT DATA CONSISTS OF 5 CARDS
C
C      CARD 1--FILTER LENGTH, TYPE OF FILTER.  1-MULTIPLE
C      PASSBAND/STOPBAND, 2-DIFFERENTIATOR, 3-HILBERT TRANSFORM
C      FILTER. NUMBER OF BANDS, CARD PUNCH DESIRED, AND GRID
C      DENSITY.
C
C      CARD 2--BANDEDGES, LOWER AND UPPER EDGES FOR EACH BAND
C      WITH A MAXIMUM OF 10 BANDS.
C
C      CARD 3--DESIRED FUNCTION (OR DESIRED SLOPE IF A
C      DIFFERENTIATOR) FOR EACH BAND.
C
C      CARD 4--WEIGHT FUNCTION IN EACH BAND. FOR A
C      DIFFERENTIATOR, THE WEIGHT FUNCTION IS INVERSELY
C      PROPORTIONAL TO F.
C
C      THE FOLLOWING INPUT DATA SPECIFIES A LENGTH 32 BANDPASS
C      FILTER WITH STOPBAND 0 TO 0.1 AND 0.425 TO 0.5, AND
C      PASSBAND FROM 0.2 TO 0.35 WITH WEIGHTING OF 10 IN THE
C      STOPBAND AND 1 IN THE PASSBAND. THE IMPULSE RESPONSE
C      WILL BE PUNCHED AND THE GRID DENSITY IS 32.
C
C      SAMPLE INPUT DATA SETUP
C      32,1,3,1,32
C      0,0.1,0.2,0.35,0.425,0.5
C      0,1,0
C      10,1,10
C
C      THE FOLLOWING INPUT DATA SPECIFIES A LENGTH 32 WIDEBAND
C      DIFFERENTIATOR WITH SLOPE 1 AND WEIGHTING OF 1/F. THE
C      IMPULSE RESPONSE WILL NOT BE PUNCHED AND THE GRID
C      DENSITY IS ASSUMED TO BE 16.
C
C      32,2,1,0,0
C      0,0.5
C      1.0
C      1.0
C
C
       COMMON PI2,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
       DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
       DIMENSION H(66)
       DIMENSION DES(1045),GRID(1045),WT(1045)
       DIMENSION EDGE(20),FX(10),WTX(10),DEVIAT(10)
       DOUBLE PRECISION PI2,PI
       DOUBLE PRECISION AD,DEV,X,Y
       PI2=6.283185307179586
       PI=3.141592653589793
C
C      THE PROGRAM IS SET UP FOR A MAXIMUM LENGTH OF 128, BUT
C      THIS UPPER LIMIT CAN BE CHANGED BY REDIMENSIONING THE
C      ARRAYS IEXT, AD, ALPHA, X, Y, H TO BE NFMAX/2 + 2.
C      THE ARRAYS DES, GRID, AND WT MUST DIMENSIONED
```

```
C       16(NFMAX/2 + 2).
C
        NFMAX=128
  100   CONTINUE
        JTYPE=0
C
C       PROGRAM INPUT SECTION
C
        READ 465,NFILT,JTYPE,NBANDS,JPUNCH,LGRID
  465   FORMAT(5I3)
        IF(NFILT.GT.NFMAX.OR.NFILT.LT.3) CALL ERROR
        IF(NBANDS.LE.0) NBANDS=1
C
C       GRID DENSITY IS ASSUMED TO BE 16 UNLESS SPECIFIED
C       OTHERWISE
C
        IF(LGRID.LE.0) LGRID=16
        JB=2*NBANDS
        READ 470,(EDGE(J),J=1,JB)
        READ 470,(FX(J),J=1,NBANDS)
        READ 470,(WTX(J),J=1,NBANDS)
  470   FORMAT(10F8.0)
        IF(JTYPE.EQ.0) CALL ERROR
        NEG=1
        IF(JTYPE.EQ.1) NEG=0
        NODD=NFILT/2
        NODD=NFILT-2*NODD
        NFCNS=NFILT/2
        IF(NODD.EQ.1.AND.NEG.EQ.0) NFCNS=NFCNS+1
C
C       SET UP THE DENSE GRID.  THE NUMBER OF POINTS IN THE GRID
C       IS (FILTER LENGTH +1)*GRID DENSITY/2
C
        GRID(1)=EDGE(1)
        DELF=LGRID*NFCNS
        DELF=0.5/DELF
        IF(NEG.EQ.0) GO TO 135
        IF(EDGE(1).LT.DELF) GRID(1)=DELF
  135   CONTINUE
        J=1
        L=1
        LBAND=1
  140   FUP=EDGE(L+1)
  145   TEMP=GRID(J)
C
C       CALCULATE THE DESIRED MAGNITUDE RESPONSE AND THE WEIGHT
C       FUNCTION ON THE GRID
C
        DES(J)=EFF(TEMP,FX,WTX,LBAND,JTYPE)
        WT(J)=WATE(TEMP,FX,WTX,LBAND,JTYPE)
        J=J+1
        GRID(J)=TEMP+DELF
        IF(GRID(J).GT.FUP) GO TO 150
        GO TO 145
  150   GRID(J-1)=FUP
        DES(J-1)=EFF(FUP,FX,WTX,LBAND,JTYPE)
        WT(J-1)=WATE(FUP,FX,WTX,LBAND,JTYPE)
        LBAND=LBAND+1
        L=L+2
        IF(LBAND.GT.NBANDS) GO TO 160
```

```
      GRID(J)=EDGE(L)
      GO TC 140
  160 NGRID=J-1
      IF(NEG.NE.NODD) GO TC 165
      IF(GRID(NGRID).GT.(0.5-DELF)) NGRID=NGRID-1
  165 CONTINUE
C
C     SET UP A NEW APPROXIMATICN PROBLEM WHICH IS EQUIVALENT
C     TO THE ORIGINAL PRCBLEM
C
      IF(NEG) 170,17C,18C
  170 IF(NODD.EC.1) GO TO 200
      DO 175 J=1,NGRID
      CHANGE=DCOS(PI*GRID(J))
      DES(J)=DES(J)/CHANGE
  175 WT(J)=WT(J)*CHANGE
      GO TO 200
  180 IF(NODD.EQ.1) GC TC 190
      DO 185 J=1,NGRID
      CHANGE=DSIN(PI*GRIC(J))
      DES(J)=DES(J)/CHANGE
  185 WT(J)=WT(J)*CHANGE
      GO TO 200
  190 DO 195 J=1,NGRID
      CHANGE=DSIN(PI2*GRID(J))
      DES(J)=DES(J)/CHANGE
  195 WT(J)=WT(J)*CHANGE
C
C     INITIAL GUESS FOR THE EXTREMAL FREQUENCIES--EQUALLY
C     SPACEC ALONG THE GRIC
C
  200 TEMP=FLCAT(NGRID-1)/FLCAT(NFCNS)
      CO 210 J=1,NFCNS
  210 IEXT(J)=(J-1)*TEMP+1
      IEXT(NFCNS+1)=NGRID
      NM1=NFCNS-1
      NZ=NFCNS+1
C
C     CALL THE REMEZ EXCHANGE ALGORITHM TO DO THE APPROXIMATION
C     PRCBLEM
C
      CALL REMEZ(EDGE,NBANDS)
C
C     CALCULATE THE IMPULSE RESPONSE.
C
      IF(NEG) 300,300,320
  300 IF(NODC.EQ.0) GO TO 31C
      DO 305 J=1,NM1
  305 H(J)=C.5*ALPHA(NZ-J)
      H(NFCNS)=ALPHA(1)
      GO TC 350
  310 H(1)=C.25*ALPHA(NFCNS)
      DO 315 J=2,NM1
  315 H(J)=C.25*(ALPHA(NZ-J)+ALPHA(NFCNS+2-J))
      H(NFCNS)=C.5*ALPHA(1)+C.25*ALPHA(2)
      GO TC 350
  320 IF(NODD.EQ.0) GC TC 33C
      H(1)=0.25*ALPHA(NFCNS)
      H(2)=0.25*ALPHA(NM1)
      DO 325 J=3,NM1
```

```
  325 H(J)=0.25*(ALPHA(NZ-J)-ALPHA(NFCNS+3-J))
      H(NFCNS)=0.5*ALPHA(1)-0.25*ALPHA(3)
      H(NZ)=0.0
      GO TO 350
  330 H(1)=0.25*ALPHA(NFCNS)
      DO 335 J=2,NM1
  335 H(J)=0.25*(ALPHA(NZ-J)-ALPHA(NFCNS+2-J))
      H(NFCNS)=0.5*ALPHA(1)-0.25*ALPHA(2)
C
C     PROGRAM OUTPUT SECTION.
C
  350 PRINT 360
  360 FORMAT(1H1, 70(1H*)//25X,'FINITE IMPULSE RESPONSE (FIR)'/
     125X,'LINEAR PHASE DIGITAL FILTER DESIGN'/
     225X,'REMEZ EXCHANGE ALGORITHM'/)
      IF(JTYPE.EQ.1) PRINT 365
  365 FORMAT(25X,'BANDPASS FILTER'/)
      IF(JTYPE.EQ.2) PRINT 370
  370 FORMAT(25X,'DIFFERENTIATOR'/)
      IF(JTYPE.EQ.3) PRINT 375
  375 FORMAT(25X,'HILBERT TRANSFORMER'/)
      PRINT 378,NFILT
  378 FORMAT(15X,'FILTER LENGTH = ',I3/)
      PRINT 380
  380 FORMAT(15X,'***** IMPULSE RESPONSE *****')
      DO 381 J=1,NFCNS
      K=NFILT+1-J
      IF(NEG.EQ.0) PRINT 382,J,H(J),K
      IF(NEG.EQ.1) PRINT 383,J,H(J),K
  381 CONTINUE
  382 FORMAT(20X,'H(',I3,') =',E15.8,' = H(',I4,')')
  383 FORMAT(20X,'H(',I3,') =',E15.8,' = -H(',I4,')')
      IF(NEG.EQ.1.AND.NODD.EQ.1) PRINT 384,NZ
  384 FORMAT(20X,'H(',I3,') = 0.0')
      DO 450 K=1,NBANDS,4
      KUP=K+3
      IF(KUP.GT.NBANDS) KUP=NBANDS
      PRINT 385,(J,J=K,KUP)
  385 FORMAT(/24X,4('BAND',I3,8X))
      PRINT 390,(EDGE(2*J-1),J=K,KUP)
  390 FORMAT(2X,'LOWER BAND EDGE',5F15.9)
      PRINT 395,(EDGE(2*J),J=K,KUP)
  395 FORMAT(2X,'UPPER BAND EDGE',5F15.9)
      IF (JTYPE.NE.2) PRINT 400,(FX(J),J=K,KUP)
  400 FORMAT(2X,'DESIRED VALUE',2X,5F15.9)
      IF(JTYPE.EQ.2) PRINT 405, (FX(J),J=K,KUP)
  405 FORMAT(2X,'DESIRED SLOPE',2X,5F15.9)
      PRINT 410,(WTX(J),J=K,KUP)
  410 FORMAT(2X,'WEIGHTING',6X,5F15.9)
      DO 420 J=K,KUP
  420 DEVIAT(J)=DEV/WTX(J)
      PRINT 425,(DEVIAT(J),J=K,KUP)
  425 FORMAT(2X,'DEVIATION',6X,5F15.9)
      IF (JTYPE.NE.1) GO TO 450
      DO 430 J=K,KUP
  430 DEVIAT(J)=20.0*ALOG10(DEVIAT(J))
      PRINT 435,(DEVIAT(J),J=K,KUP)
  435 FORMAT(2X,'DEVIATION IN DB',5F15.9)
  450 CONTINUE
      PRINT 455,(GRID(IEXT(J)),J=1,NZ)
```

```
  455 FORMAT(/2X,'EXTREMAL FREQUENCIES'/(2X,5F12.7))
      PRINT 460
  460 FORMAT(/1X,70(1H*)/1H1)
      IF(JPUNCH.NE.0) PUNCH 475,(H(J),J=1,NFCNS)
  475 FORMAT(5E16.8)
      IF(NFILT.NE.0) GO TO 100
      RETURN
      END
      FUNCTION EFF(TEMP,FX,WTX,LBAND,JTYPE)
C
C     FUNCTION TO CALCULATE THE DESIRED MAGNITUDE RESPONSE
C     AS A FUNCTION OF FREQUENCY.
C
      DIMENSION FX(5),WTX(5)
      IF(JTYPE.EQ.2) GO TO 1
      EFF=FX(LBAND)
      RETURN
    1 EFF=FX(LBAND)*TEMP
      RETURN
      END
      FUNCTION WATE(TEMP,FX,WTX,LBAND,JTYPE)
C
C     FUNCTION TO CALCULATE THE WEIGHT FUNCTION AS A FUNCTION
C     OF FREQUENCY.
C
      DIMENSION FX(5),WTX(5)
      IF(JTYPE.EQ.2) GO TO 1
      WATE=WTX(LBAND)
      RETURN
    1 IF(FX(LBAND).LT.0.0001) GO TO 2
      WATE=WTX(LBAND)/TEMP
      RETURN
    2 WATE=WTX(LBAND)
      RETURN
      END
      SUBROUTINE REMEZ(EDGE,NBANDS)
C
C     THIS SUBROUTINE IMPLEMENTS THE REMEZ EXCHANGE ALGORITHM
C     FOR THE WEIGHTED CHEBYCHEV APPROXIMATION OF A CONTINUOUS
C     FUNCTION WITH A SUM OF COSINES.  INPUTS TO THE SUBROUTINE
C     ARE A DENSE GRID WHICH REPLACES THE FREQUENCY AXIS, THE
C     DESIRED FUNCTION ON THIS GRID, THE WEIGHT FUNCTION ON THE
C     GRID, THE NUMBER OF COSINES, AND AN INITIAL GUESS OF THE
C     EXTREMAL FREQUENCIES.  THE PROGRAM MINIMIZES THE CHEBYCHEV
C     ERROR BY DETERMINING THE BEST LOCATION OF THE EXTREMAL
C     FREQUENCIES (POINTS OF MAXIMUM ERROR) AND THEN CALCULATES
C     THE COEFFICIENTS OF THE BEST APPROXIMATION.
C
      COMMON PI2,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
      DIMENSION EDGE(20)
      DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
      DIMENSION DES(1045),GRID(1045),WT(1045)
      DIMENSION A(66),P(65),Q(65)
      DOUBLE PRECISION PI2,DNUM,DDEN,DTEMP,A,P,Q
      DOUBLE PRECISION AD,DEV,X,Y
C
C     THE PROGRAM ALLOWS A MAXIMUM NUMBER OF ITERATIONS OF 25
C
      ITRMAX=25
      DEVL=-1.0
```

```
      NZ=NFCNS+1
      NZZ=NFCNS+2
      NITER=0
  100 CONTINUE
      IEXT(NZZ)=NGRID+1
      NITER=NITER+1
      IF(NITER.GT.ITRMAX) GO TO 400
      DO 110 J=1,NZ
      DTEMP=GRID(IEXT(J))
      DTEMP=DCOS(DTEMP*PI2)
  110 X(J)=DTEMP
      JET=(NFCNS-1)/15+1
      DO 120 J=1,NZ
  120 AD(J)=D(J,NZ,JET)
      DNUM=0.0
      DDEN=0.0
      K=1
      DO 130 J=1,NZ
      L=IEXT(J)
      DTEMP=AD(J)*DES(L)
      DNUM=DNUM+DTEMP
      DTEMP=K*AD(J)/WT(L)
      DDEN=DDEN+DTEMP
  130 K=-K
      DEV=DNUM/DDEN
      NU=1
      IF(DEV.GT.0.0) NU=-1
      DEV=-NU*DEV
      K=NU
      DO 140 J=1,NZ
      L=IEXT(J)
      DTEMP=K*DEV/WT(L)
      Y(J)=DES(L)+DTEMP
  140 K=-K
      IF(DEV.GE.DEVL) GO TO 150
      CALL OUCH
      GO TO 400
  150 DEVL=DEV
      JCHNGE=0
      K1=IEXT(1)
      KNZ=IEXT(NZ)
      KLOW=0
      NUT=-NU
      J=1
C
C     SEARCH FOR THE EXTREMAL FREQUENCIES OF THE BEST
C     APPROXIMATION
C
  200 IF(J.EQ.NZZ) YNZ=COMP
      IF(J.GE.NZZ) GO TO 300
      KUP=IEXT(J+1)
      L=IEXT(J)+1
      NUT=-NUT
      IF(J.EQ.2) Y1=COMP
      COMP=DEV
      IF(L.GE.KUP) GO TO 220
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 220
```

```
      COMP=NUT*ERR
  210 L=L+1
      IF(L.GE.KUP) GO TO 215
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 215
      COMP=NUT*ERR
      GO TO 210
  215 IEXT(J)=L-1
      J=J+1
      KLOW=L-1
      JCHNGE=JCHNGE+1
      GO TO 200
  220 L=L-1
  225 L=L-1
      IF(L.LE.KLOW) GO TO 250
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.GT.0.0) GO TO 230
      IF(JCHNGE.LE.0) GO TO 225
      GO TO 260
  230 COMP=NUT*ERR
  235 L=L-1
      IF(L.LE.KLOW) GO TO 240
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 240
      COMP=NUT*ERR
      GO TO 235
  240 KLOW=IEXT(J)
      IEXT(J)=L+1
      J=J+1
      JCHNGE=JCHNGE+1
      GO TO 200
  250 L=IEXT(J)+1
      IF(JCHNGE.GT.0) GO TO 215
  255 L=L+1
      IF(L.GE.KUP) GO TO 260
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 255
      COMP=NUT*ERR
      GO TO 210
  260 KLOW=IEXT(J)
      J=J+1
      GO TO 200
  300 IF(J.GT.NZZ) GO TO 320
      IF(K1.GT.IEXT(1)) K1=IEXT(1)
      IF(KNZ.LT.IEXT(NZ)) KNZ=IEXT(NZ)
      NUT1=NUT
      NUT=-NU
      L=0
      KUP=K1
      COMP=YNZ*(1.00001)
      LUCK=1
  310 L=L+1
```

```
      IF(L.GE.KUP) GO TO 315
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 310
      COMP=NUT*ERR
      J=NZZ
      GO TO 210
  315 LUCK=6
      GO TO 325
  320 IF(LUCK.GT.9) GO TO 350
      IF(COMP.GT.Y1) Y1=COMP
      K1=IEXT(NZZ)
  325 L=NGRID+1
      KLOW=KNZ
      NUT=-NUT1
      COMP=Y1*(1.00001)
  330 L=L-1
      IF(L.LE.KLOW) GO TO 340
      ERR=GEE(L,NZ)
      ERR=(ERR-DES(L))*WT(L)
      DTEMP=NUT*ERR-COMP
      IF(DTEMP.LE.0.0) GO TO 330
      J=NZZ
      COMP=NUT*ERR
      LUCK=LUCK+10
      GO TO 235
  340 IF(LUCK.EQ.6) GO TO 370
      DO 345 J=1,NFCNS
  345 IEXT(NZZ-J)=IEXT(NZ-J)
      IEXT(1)=K1
      GO TO 100
  350 KN=IEXT(NZZ)
      DO 360 J=1,NFCNS
  360 IEXT(J)=IEXT(J+1)
      IEXT(NZ)=KN
      GO TO 100
  370 IF(JCHNGE.GT.0) GO TO 100
C     CALCULATION OF THE COEFFICIENTS OF THE BEST APPROXIMATION
C     USING THE INVERSE DISCRETE FOURIER TRANSFORM
C
  400 CONTINUE
      NM1=NFCNS-1
      FSH=1.0E-06
      GTEMP=GRID(1)
      X(NZZ)=-2.0
      CN=2*NFCNS-1
      DELF=1.0/CN
      L=1
      KKK=0
      IF(EDGE(1).EQ.0.0.AND.EDGE(2*NBANDS).EQ.0.5) KKK=1
      IF(NFCNS.LE.3) KKK=1
      IF(KKK.EQ.1) GO TO 405
      DTEMP=DCOS(PI2*GRID(1))
      DNUM=DCOS(PI2*GRID(NGRID))
      AA=2.0/(DTEMP-DNUM)
      BB=-(DTEMP+DNUM)/(DTEMP-DNUM)
  405 CONTINUE
      DO 430 J=1,NFCNS
      FT=(J-1)*DELF
```

```
      XT=DCOS(PI2*FT)
      IF(KKK.EQ.1) GC TO 410
      XT=(XT-BB)/AA
      FT=ARCCS(XT)/PI2
  410 XE=X(L)
      IF(XT.GT.XE) GO TO 420
      IF((XE-XT).LT.FSH) GC TO 415
      L=L+1
      GO TO 410
  415 A(J)=Y(L)
      GO TO 425
  420 IF((XT-XE).LT.FSH) GO TO 415
      GRID(1)=FT
      A(J)=GEE(1,NZ)
  425 CONTINUE
      IF(L.GT.1) L=L-1
  430 CONTINUE
      GRID(1)=GTEMP
      DDEN=PI2/CN
      DO 51C J=1,NFCNS
      DTEMP=C.0
      DNUM=(J-1)*DDEN
      IF(NM1.LT.1) GO TO 5C5
      DO 500 K=1,NM1
  500 DTEMP=DTEMP+A(K+1)*DCOS(DNUM*K)
  505 DTEMP=2.0*DTEMP+A(1)
  510 ALPHA(J)=DTEMP
      DO 550 J=2,NFCNS
  550 ALPHA(J)=2*ALPHA(J)/CN
      ALPHA(1)=ALPHA(1)/CN
      IF(KKK.EC.1) GO TO 545
      P(1)=2.C*ALPHA(NFCNS)*BB+ALPHA(NM1)
      P(2)=2.0*AA*ALPHA(NFCNS)
      Q(1)=ALPHA(NFCNS-2)-ALPHA(NFCNS)
      DO 540 J=2,NM1
      IF(J.LT.NM1) GO TO 515
      AA=0.5*AA
      BB=0.5*BB
  515 CONTINUE
      P(J+1)=0.0
      DO 520 K=1,J
      A(K)=P(K)
  520 P(K)=2.0*BB*A(K)
      P(2)=P(2)+A(1)*2.0*AA
      JM1=J-1
      DO 525 K=1,JM1
  525 P(K)=P(K)+Q(K)+AA**A(K+1)
      JP1=J+1
      DO 530 K=3,JP1
  530 P(K)=P(K)+AA*A(K-1)
      IF(J.EC.NM1) GC TC 540
      DO 535 K=1,J
  535 C(K)=-A(K)
      Q(1)=C(1)+ALPHA(NFCNS-1-J)
  540 CONTINUE
      DO 543 J=1,NFCNS
  543 ALPHA(J)=P(J)
  545 CONTINUE
      IF(NFCNS.GT.3) RETURN
      ALPHA(NFCNS+1)=0.0
```

```
      ALPHA(NFCNS+2)=0.0
      RETURN
      END
      SUBROUTINE OUCH
      PRINT 1
    1 FORMAT('  ************ FAILURE TO CONVERGE **********'/
     1'OPROBABLE CAUSE IS MACHINE ROUNDING ERROR'/
     2'OTHE IMPULSE RESPONSE MAY BE CORRECT'/
     3'OCHECK WITH A FREQUENCY RESPONSE'/)
      RETURN
      END
      DOUBLE PRECISION FUNCTION D(K,N,M)
C
C     FUNCTION TO CALCULATE THE LAGRANGE INTERPOLATION
C     COEFFICIENTS FOR USE IN THE FUNCTION GEE.
C
      COMMON PI2,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
      DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
      DIMENSION DES(1045),GRID(1045),WT(1045)
      DOUBLE PRECISION AD,DEV,X,Y
      DOUBLE PRECISION Q
      DOUBLE PRECISION PI2
      D=1.0
      Q=X(K)
      DO 3 L=1,M
      DO 2 J=L,N,M
      IF(J-K)1,2,1
    1 D=2.0*D*(Q-X(J))
    2 CONTINUE
    3 CONTINUE
      D=1.0/D
      RETURN
      END
      DOUBLE PRECISION FUNCTION GEE(K,N)
C
C     FUNCTION TO EVALUATE THE FREQUENCY RESPONSE USING THE
C     LAGRANGE INTERPOLATION FORMULA IN THE BARYCENTRIC FORM
C
      COMMON PI2,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
      DIMENSION IEXT(66),AD(66),ALPHA(66),X(66),Y(66)
      DIMENSION DES(1045),GRID(1045),WT(1045)
      DOUBLE PRECISION P,C,D,XF
      DOUBLE PRECISION PI2
      DOUBLE PRECISION AD,DEV,X,Y
      P=0.0
      XF=GRID(K)
      XF=DCOS(PI2*XF)
      D=0.0
      DO 1 J=1,N
      C=XF-X(J)
      C=AD(J)/C
      D=D+C
    1 P=P+C*Y(J)
      GEE=P/D
      RETURN
      END
      SUBROUTINE ERROR
      PRINT 1
    1 FORMAT('  ************ ERROR IN INPUT DATA **********')
      STOP
      END
```

APPENDIX V

This Appendix contains FORTRAN program for calculating and plotting the magnitude and phase characteristics corresponding to analog or digital transfer function as a function of w or respectively, where is the normalized frequency variable given by F/F_S.

```
C  PROGRAM DESCRIPTION.
C
C        MAGPHASE IS A GENERALIZED PROGRAM FOR CALCULATING AND PLOTTING
C     THE MAGNITUDE AND PHASE CHARACTERISTICS OF ANY ANALOG OR DIGITAL
C     TWO-PORT NETWORK AS A FUNCTION OF OMEGA OR ETA RESPECTIVELY,
C     GIVEN THE TRANSFER FUNCTION H(S) OR H(Z) AS A POLYNOMIAL
C     NUMERATOR AND DENOMINATOR.
C
C=======================================================================
C
C  ENVIRONMENT.
C
C        THIS PROGRAM WAS RUN AT KANSAS STATE UNIVERSITY ON AN IBM
C     370/158 USING BOTH WATFIV V1L4 AND FORTRAN IV G LEVEL 21.
C     PRINT WIDTH WAS HELD TO LESS THAN 120 COLUMNS TO EASE CONVERSION.
C     NON-STANDARD FORTRAN USE INCLUDES: DOUBLE PRECISION COMPLEX
C     NUMBERS, INITIALIZATION IN DECLARATION STATEMENTS, USE OF
C     END-OF-FILE TEST ON READ STATEMENTS, CERTAIN FUNCTIONS SUCH AS:
C     DATAN2,CDEXP,SQRT,ETC., THE USE OF A NON-PRINTING CHARACTER FOR
C     CONTINUATION, AND THE "G" FORMAT ON OUTPUT.  IN EACH CASE CLOSE
C     EXAMINATION SHOULD SUGGEST ALTERNATE APPROACHES.
C
C=======================================================================
C
C  INPUT.
C
C        THE FIRST CARD OF EACH NETWORK IS A CONTROL CARD.  ITS FIELDS
C     ARE:
C
C     1-2     - DEGREE OF NUMERATOR
C     3-4     - DEGREE OF DENOMINATOR
C     5-7     - NUMBER OF PLOTTED POINTS BETWEEN THE STARTING AND
C               STOPPING FREQUENCIES INCLUSIVE
C     8       - H(S) (=0) OR H(Z) (=1)
C     9       - FREQUENCY INCREMENT METHOD (0=LINEAR, 1=LOG)
C     11-20   - STARTING FREQUENCY
C     21-30   - FINAL FREQUENCY
C

C        THE FIRST FIVE FIELDS ARE READ WITH AN INTEGER FORMAT.  THE
C     REST OF THIS CARD AND THE FOLLOWING CARDS ARE READ WITH AN "F10.5"
C     FORMAT.
C
C        THE NEXT CARDS ARE FOR THE NUMERATOR AND DENOMINATOR.  VALUES
C     ARE READ 8 PER CARD AND ARE IN  *** DESCENDING ***  ORDER.  THE
C     DENOMINATOR BEGINS ON A NEW CARD: I.E. IT DOES NOT CONTINUE ON THE
C     NUMERATOR CARD.
C
C=======================================================================
C
C  POLES AND ZEROES.
C
C        POLES AND ZEROES ARE CHECKED FOR AND ARE FLAGGED IN THE OUTPUT.
C     THE CRITERION FOR A ZERO IS A ZERO IN THE NUMERATOR REGARDLESS OF
C     THE MAGNITUDE OF THE DENOMINATOR.  THE CRITERION FOR A POLE IS
C     AN ORDER OF MAGNITUDE INCREASE OVER THE PREVIOUS MAXIMUM VALUE.
C
C=======================================================================
```

```
      COMPLEX*16 PARM,CDEXP,DCMPLX,NTEMP,DTEMP
      REAL*8 FSTEP,FREQ,LABEL(2),LABEL1(2)/'  OMEGA ','   ETA  '/
      REAL*8 LABEL2(2)/' ----- ',' ---   '/,PI,PI2
      REAL*8 DATAN2,NREAL(2),DREAL(2),DBLE,DLOG,DEXP
      REAL FSTART,FSTOP,MAX,MAG(200),PHS(200),OMEGA(200),NUM(99)
      REAL DEN(99),MAGNUM,MAGDEN
      INTEGER TYPE,NUMPTS,NUMDIM,DENDIM,INCTYP,DIM/200/
      INTEGER POLE(200)
      EQUIVALENCE (NTEMP,NREAL(1)),(DTEMP,DREAL(1))
      PI=3.141592653589793
      PI2=PI*2.0
C                 READ IN NEXT NETWORK.
    5 READ(5,1000,END=900) NUMDIM,DENDIM,NUMPTS,TYPE,INCTYP,FSTART,FSTOP
      NUMDIM=NUMDIM+1
      DENDIM=DENDIM+1
C                 READ IN NUMERATOR AND DENOMINATOR POLYNOMIALS.
      READ(5,1002,END=902) (NUM(I),I=1,NUMDIM)
      READ(5,1002,END=902) (DEN(I),I=1,DENDIM)
C                 CALCULATE STEP SIZE TO BE USED.
      IF (NUMPTS.LT.2) NUMPTS=2
      IF (NUMPTS.GT.200) NUMPTS=200
      IF (INCTYP)                                 10,10,15
   10 FSTEP=(DBLE(FSTOP)-DBLE(FSTART))/(NUMPTS-1)
                                                  GO TO 20
   15 FSTEP=DEXP(DLOG(DBLE(FSTOP)/DBLE(FSTART))/(NUMPTS-1))
   20 FREQ=FSTART
      MAX=0.0
C                 SET UP PRINT LABELS.
      LABEL(1)=LABEL1(TYPE+1)
      LABEL(2)=LABEL2(TYPE+1)
      WRITE(6,1004) LABEL(1)
      DO 105 I=1,NUMPTS
C                 CALCULATE VALUE OF OMEGA OR ETA.
         IF (TYPE)                                25,25,30
   25    PARM=DCMPLX(0.0D0,FREQ)
                                                  GO TO 35
   30    PARM=CDEXP(DCMPLX(0.0D0,PI2*FREQ))
C                 EVALUATE NUMERATOR AND DENOMINATOR USING HORNER'S
C                 METHOD OF EVALUATING POLYNOMIALS.
   35    ISTOP=NUMDIM-1
         IF (ISTOP)                               40,40,45
   40    NTEMP=DCMPLX(DBLE(NUM(1)),0.0D0)
                                                  GO TO 55
   45    NTEMP=(0.0,0.0)
         DO 50 J=1,ISTOP
   50       NTEMP=(NTEMP+NUM(J))*PARM
         NTEMP=NTEMP+NUM(NUMDIM)
   55    ISTOP=DENDIM-1
         IF (ISTOP)                               60,60,65
   60    DTEMP=DCMPLX(DBLE(DEN(1)),0.0D0)
                                                  GO TO 75
   65    DTEMP=(0.0,0.0)
         DO 70 J=1,ISTOP
   70       DTEMP=(DTEMP+DEN(J))*PARM
         DTEMP=DTEMP+DEN(DENDIM)
   75    OMEGA(I)=FREQ
         MAGNUM=NREAL(1)*NREAL(1)+NREAL(2)*NREAL(2)
         MAGDEN=DREAL(1)*DREAL(1)+DREAL(2)*DREAL(2)
C                 CHECK FOR ZEROES.
         IF (MAGNUM.NE.0)                         GO TO 80
```

```
      PCLE(I)=0
      WRITE(6,1010) OMEGA(I)
                                                  GC TO 90
C                   CHECK FOR POLES.
   80 IF(MAGDEN.EC.0)                              GO TO 82
      IF (I.LE.2)                                 GO TO 85
      IF (MAGDEN.GT.MAGNUM/(1C.0*MAX))             GO TO 85
   82 PCLE(I)=1
      WRITE(6,1012) OMEGA(I)
                                                  GO TO 90
   85 PCLE(I)=-1
      MAG(I)=SCRT(MAGNUM/MAGDEN)
      IF (MAG(I).GT.MAX) MAX=MAG(I)
      PHS(I)=DATAN2(NREAL(2),NREAL(1))-DATAN2(DREAL(2),CREAL(1))
      IF (PHS(I).GT.PI) PHS(I)=PHS(I)-PI2
      IF (PHS(I).LT.-PI) PHS(I)=PHS(I)+PI2
      WRITE(6,1CC6) OMEGA(I),MAG(I),PHS(I)
C                   INCREMENT FREQUENCY.
   90 IF (INCTYP)                                 95,95,100
   95 FREC=FREC+FSTEP
                                                  GC TO 105
  100 FREC=FREC*FSTEP
  1C5 CONTINUE
      CALL GRAPH(MAG,PHS,OMEGA,NUMPTS,MAX,LABEL,PCLE)
                                                  GO TO 5
  9C2 WRITE(6,1008)
  9CC RETURN
 1000 FORMAT(2I2,I3,2I1,1X,2F1C.5)
 1002 FCRMAT(8F10.5)
 1CC4 FORMAT('1   ',A8,10X,'MAGNITUDE',9X,'PHASE(RAD)')
 1CC6 FORMAT(' ',3(G13.6,5X))
 1008 FORMAT(' ***** ERROR *****  END-OF-FILE REACHED WHILE READING ',
     X   'CCEFFICIENTS.')
 1C1C FORMAT(' ',G13.6,8X,'<<< ZERO ENCCUNTERED >>>')
 1012 FORMAT(' ',G13.6,8X,'<<< PCLE ENCCUNTERED >>>')
      END
      SUBROUTINE GRAPH(MAG,PHASE,CMEGA,NUMPTS,MAX,LABEL,POLE)
C                         SUBROUTINE GRAPH PRODUCES A MAGNITUDE AND PHASE
C                   RESPONSE GRAPH.  THE FREQUENCY,MAGNITUDE AND PHASE
C                   ARE ASSUMED TC BE IN THE ARRAYS OMEGA,MAG AND PHASE
C                   RESPECTIVELY.  NUMPTS IS THE NUMBER CF CATA POINTS
C                   MAX IS THE MAXIMUM VALUE IN MAG,AND LABEL IS THE
C                   TITLE DESIRED ABOVE THE FREQUENCY COLUMN.  THE ARRAY
C                   POLE CONTAINS A FLAG FCR PCLES OR ZERCES.
C                   THE DATA IS PLOTTED 5C PER PAGE.
      REAL*8 LABEL(2)
      REAL MAG(20C),PHASE(200),CMEGA(20C),MAX,SCALEV,RELBL(6)
      INTEGER NUMPTS,ILBL(6),EXP,PLCTP(49)/49*' '/,PLCTM(51)/51*' '/
      INTEGER XBLK/' '/,XAST/'*'/,LBLDIM/6/,IWIDTH/51/,ISTART,ISTOP
      INTEGER POLE(200)
      PI24=7.6394373
      CALL SCALE(MAX,SCALEV,RELBL,ILBL,EXP,LBLDIM,IWIDTH)
      ISTCP=0
    1 ISTART=ISTOP+1
      ISTOP=ISTOP+51
      IF (ISTOP.GT.NUMPTS) ISTOP=NUMPTS
      WRITE(6,1000) LABEL(1),LABEL(2)
      IF (ILBL(2).GT.C)                           GC TO 5
      WRITE(6,1002) (RELBL(I),I=1,6)
                                                  GO TO 10
```

```
    5 WRITE(6,1004) (ILBL(I),EXP,I=1,6)
   10 WRITE(6,1006)
      DO 30 I=ISTART,ISTOP
         IF (POLE(I))                                25,20,15
   15    WRITE(6,1012) OMEGA(I)
                                                     GO TO 30
   20    WRITE(6,1014) OMEGA(I)
                                                     GO TO 30
   25    J=PHASE(I)*PI24+25.5
         K=MAG(I)*SCALEV+1.5
         PLOTP(J)=XAST
         PLOTM(K)=XAST
         WRITE(6,1008) (PLOTP(L),L=1,49),OMEGA(I),(PLOTM(L),L=1,51)
         PLOTP(J)=XBLK
         PLOTM(K)=XBLK
   30    CONTINUE
      WRITE(6,1010)
      IF (ILBL(2).GT.0)                              GO TO 35
      WRITE(6,1002) (RELBL(I),I=1,6)
                                                     GO TO 40
   35 WRITE(6,1004) (ILBL(I),EXP,I=1,6)
   40 IF (ISTOP.GE.NUMPTS)                           RETURN
      ISTOP=ISTOP-1
                                                     GO TO 1
 1000 FORMAT('1',23X,'PHASE',26X,A8,25X,'MAGNITUDE'/
     X    '+',23X,'_____',26X,A8,25X,'_________'/
     X    '0-PI',8X,'-PI/2',9X,'0',10X,'PI/2',9X,'PI')
 1002 FORMAT('+',63X,6(F7.2,3X))
 1004 FORMAT('+',63X,6(I2,'E',I3,4X))
 1006 FORMAT('    ',5('|',11X),5X,5('|',9X),'|'/'    ',49('-'),16X,51('-'))
 1008 FORMAT('    ',49A1,2X,G12.5,2X,51A1)
 1010 FORMAT('    ',49('-'),16X,51('-')/'    ',5('|',11X),5X,5('|',9X),'|'/
     X    ' -PI',8X,'-PI/2',9X,'0',10X,'PI/2',9X,'PI')
 1012 FORMAT(15X,'<<< POLE ENCOUNTERED >>>',14X,G12.5,16X,
     X    '<<< POLE ENCOUNTERED >>>')
 1014 FORMAT(15X,'<<< ZERO ENCOUNTERED >>>',14X,G12.5,2X,'*')
      END
      SUBROUTINE SCALE(MAX,SCALEV,RELBL,ILBL,EXP,LBLDIM,IWIDTH)
C                          SUBROUTINE SCALE CALCULATES A SUITABLE SCALE
C                    FOR PLOT POINTS. IT PRODUCES THE NECESSARY LABELS
C                    AND SCALE FACTOR. INPUT TO THIS SUBROUTINE CONSISTS
C                    OF: MAX,MAXIMUM VALUE OF THE DATA;LBLDIM, THE NUMBER
C                    OF LABELS DESIRED; AND IWIDTH, THE WIDTH OF THE PLOT
C                    IN COLUMNS.  IT OUTPUTS: SCALEV, A SCALING CONSTANT
C                    THAT WILL SCALE THE DATA RANGING FROM ZERO TO MAX;
C                    RELBEL, THE AXIS LABELS IF MAX  IS SUCH TO ALLOW
C                    LABELS BETWEEN 800 AND./1 ; AND ILBL AND EXP, THE
C                    LABELS IF DATA IS OUTSIDE THE GIVEN RANGE.
      REAL RELBL(LBLDIM),SCALEW,SCALEV,MAX,X,Y
      INTEGER ILBL(LBLDIM),EXP,IWIDTH,K,I,J
      K=LBLDIM-1
      SCALEW=(IWIDTH-1)/FLOAT(K)
C                    TAKE THE LOG OF THE NUMBER AND SEPARATE THE
C                    CHARACTERISTIC FROM THE MANTISSA.
      Y=ALOG10(MAX/K)
      I=Y
      Y=Y-I
      IF (Y.GE.0)                                    GO TO 5
      Y=Y+1
      I=I-1
```

```
    5  IF(Y.GT..001)                                        GO TO 10
       X=1.0
                                                            GO TO 35
   10  IF (Y.GT..302)                                       GO TO 15
       X=2.0
                                                            GO TO 35
   15  IF (Y.GT..603)                                       GO TO 20
       X=4.0
                                                            GO TO 35
   20  IF (Y.GT..7)                                         GO TO 25
       X=5.0
                                                            GO TO 35
   25  IF (Y.GT..904)                                       GO TO 30
       X=8.0
                                                            GO TO 35
   30  X=1.0
       I=I+1
C                    FILL LABEL ARRAYS.
   35  IF (I.LT.-2.OR.I.GT.2)                               GO TO 45
       DO 40 J=1,K
   40     RELBL(J+1)=X*J*10.0**I
       ILBL(2)=-1
       RELBL(1)=0.0
       SCALEV=SCALEW/RELBL(2)
       RETURN
   45  DO 50 J=1,K
   50     ILBL(J+1)=X*J
       ILBL(1)=0
       EXP=I
       SCALEV=SCALEW/X/10.0**I
       RETURN
       END
```

ACKNOWLEDGMENTS

The author wishes to express his deep gratitude and sincere appreciation to his major professor, Dr. Nasir Ahmed, for his advice and encouragement during the preparation of this report. Appreciation and thanks are also due to Mr. T. Natarajan, David Hein, and D. Parikh for their assistance in the writing of this report. Furthermore, the author expresses his sincere gratitude to Drs. Donald Hummels and Doris Grosh, who served as his committee members.

The author also expresses his sincere appreciation for the financial assistance received from the Electrical Engineering Department, and is also grateful to Mrs. Patricia L. Stewart for typing this report.

A STUDY OF DIGITAL FILTER DESIGN TECHNIQUES

by

BUGGA RAO VUJJENI

B.E., Osmania University, Andhra Pradesh, India, 1973

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Electrical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1976

This report is primarily concerned with the study of some of the more important design methods that are normally used in the design of digital filters. These methods can be divided into two broad classes, depending upon whether the digital filter of interest is of the FIR (finite impulse response) or the IIR (infinite impulse response) type.

FIR filters can be designed using three well-known methods, which are: (i) the window method, (ii) the frequency sampling method, and (iii) the optimization method. In this report, the window method has been discussed in detail and various design examples are presented. The other two methods are discussed relatively briefly.

Again, two design methods are considered in the case of IIR filters. These are: (i) the impulse invariance method, and (ii) the bilinear transformation method. More attention has been given to the bilinear transformation method since it is used frequently in the practice. Several illustrative examples are presented.