Durham E-Theses

Design of a programmable digital filter

Sansern Visuwan

How to cite:

Visuwan, Sansern (1973) Design of a programmable digital filter. Masters thesis, Durham University.

Use policy

DESIGN OF A PROGRAMMABLE DIGITAL FILTER

by

SANSERN VISUWAN

B.Sc. (CHULALONGKORN UNIVERSITY).

A thesis submitted for the degree of master

of science in the University of Durham.

AUGUST 1973.

DEPARTMENT OF APPLIED PHYSICS AND ELECTRONICS.

Acknowledgment.

ABSTRACT.

In recent years, digital filters for the processing of signals have developed rapidly. The aim of this project was to study the design and implementation of digital filters in terms of hardware, rather than by software programming of a general purpose digital computer. A digital filter has been designed and built with 8 bit accuracy intended as part of a high order filter. It is intended to perform as a lowpass, highpass or bandpass filter by multiplexing and frequency transformation.

DESIGN OF A PROGRAMMABLE DIGITAL FILTER

by

SANSERN VISUWAN

B. Sc. (CHULALONGKORN UNIVERSITY).

A thesis submitted for the degree of master

of science in the University of Durham.

AUGUST 1973.

DEPARTMENT OF APPLIED PHYSICS AND ELECTRONICS.

Acknowledgment.

This study was financed by the Columbo plan, the British

Council, and the Prince of Songkhla University, to them I would

like to express my sincerest thanks.

I wish to express my gratitude to my supervisor, Dr. B. J.

Stanier, for his encouragement, kindly help and invaluable advice.

I should also like to give my thanks to Professor D. A. Wright,

who made available to me the facilities of the Applied Physics and

Electronics Department, allowing me to carry out my research.

Mr. P. Friend provided useful technical discussion and information

and Mr. T. Nancarrow friendly help during both the construction of

the filter and the preparation of the thesis. To both of these I

give my deepest thanks.

Mrs. D. C. Nancarrow deciphered my writing and typed the thesis,

for which I thank her kindly.

I would also like to thank all my associates in the Applied

Physics and Electronics Department who have made me feel so welcome

and help me so much during the past years.

ABSTRACT.

In recent years, digital filters for the processing of signals have developed rapidly. The aim of this project was to study the design and implementation of digital filters in terms of hardware, rather than by software programming of a general purpose digital computer. A digital filter has been designed and built with 8 bit accuracy intended as part of a high order filter. It is intended to perform as a lowpass, highpass or bandpass filter by multiplexing and frequency transformation.

Design of a programmable digital filter.

Chapter 1 : Introduction.

Chapter 2 : Digital Filtering.

Chapter 3 : Basic Design of Digital Filters.

Chapter 1.

Introduction.

1.1. Introduction to digital filtering.

A digital filter is defined (1) as the computational process or algorithm by which a sampled signal or sequence of number, acting as an input, is transformed into a second sequence of number, termed the output, using digital-components as the basic elements for implementation. The process may be that of lowpass filtering, highpass filtering or bandpass filtering, etc., and if assumed to be linear may be designed by a difference equation. This equation defines the output signal as a function of the present input signal and any number of past input and output signals.

If the output signal is a function of only the present and past input signals, the filter is called nonrecursive.

If the past output signals are included as well, the filter is called recursive.

As digital filtering is a computational process, it can also be performed by a software program on a digital computer. Either a general-purpose computer or special-purpose computer can be used with a special interface. To realize a digital filter in real time, the digital technique must be fast enough to complete the computational process within the available time.

Digital filters have wide applications and these have been discussed in the literature. Kaiser (2) has classified the applications of digital filters in three major areas of activity as

follows :

 - Used as a signal or data processing element in the reduction of experimental test data on the performance of communication and speech processing system.

 - Used as an element in the simulation of speech and in signal processing systems, to study new techniques by detailed performance analysis, and to develop promising system configurations by determining the best choice of parameters.

 - Used as an integral part of a communication or signal processing system which is realized as actual hardware.

1.2. History and development of digital filters.

The study of digital filtering started in the early 1600's from the work of mathematicians in the forms of the processing of discrete systems by classical numerical analysis techniques. Until recently, the development of digital filtering was limited to the mathematics of discrete systems. The advent of digital computers accelerated this development making it possible to perform rapidly more extensive computation procedures on more complex problems. Kaiser has investigated the early development of digital filtering and several references were given in reference (1).

Various synthesis methods have been discussed and published in the literature. In general, digital filters can be synthesized from the time domain or the frequency domain. Rader and Gold (3) have investigated and published various frequency-domain techniques. Also R. M. Golden (4) (5) has synthesized digital filters by sampled-data transformation, taking advantage of the well-known design techniques developed for continuous filters. Many articles have also been published on synthesis in the time domain. T. C. Hsia (6) has proposed a method for synthesizing digital filters by assuming that the pulsed transfer function of a digital filter is the ratio of two rational polynomials. The coefficients can then be determined by least-square fitting the digital filters to the corresponding continuous filter's sampled input and output data.

A recent trend in the synthesis of digital filters has been by the method of frequency selectivity, where the design is not by reference to continuous filters, but by synthesis from the desired

frequency response. This method may be called the frequency sampling technique. Sonderegger (7) has demonstrated this technique, using a nonrecursive approach to develop a linear phase high quality bandpass filter. L. R. Rabiner has published the design techniques and realization of frequency sampling filters. For example; see references (8).

Most practical digital filters have been realized by computer programs. These can be used to work as on-line digital filters (9) for which a special interface is needed transferring the data to a remote general-purpose computer. White and Nagle (10) have also proposed the realization of digital filters by a special-purpose computer.

However, the realization of digital filters in hardware has also developed rapidly, the growth of large scale integrated circuit (LS1) cuts down the cost and increases the speed of the components. The components and techniques required for the high speed implementation of digital filters have been discussed in the literature. (11) (12).

The cost of a digital filter depends on the word length of the coefficients. But a small word length will cause inaccuracy in the digital filtering. A suitable structure to realize a digital filter with a reduced word length but without losing accuracy has been investigated by Avenhaus. (13) (14). By studying the density of allowable root positions of the polynomial in the transfer function of a digital filter and plotting them into the Z-plane, he has been able to realize a digital filter with the optimal word length.

Digital filters have now become an accepted tool in signal

processing. Tables and graphs for designing digital Chebyshev filters and digital Butterworth filters are now available (15) (16).

With the continuing advance of digital technology and availability of large scale integrated circuits, one can predict that a digital filter will soon be available on only one chip. This will reduce the cost of digital filters until they can replace many of the continuous filters used in communication systems.

1.3. Digital filters and continuous filters.

The aims of digital filtering are the same as continuous filtering, but the physical realization is different. Linear-continuous filter theory is based on linear differential equations and the Laplace transform while linear digital filter theory is based on linear difference equations and the Z-transform. However, digital filters are mathematically equivalent to continuous filters with sampled data inputs and outputs.

By using digital techniques in implementation, digital filters have several advantages over continuous filters. Some of these advantages are as follows :

- the absence of impedance-matching problems.

- the frequency response can be changed easily by varying the proper stored coefficients.

- any type of filter can be performed with the same hardware by using multiplexing and frequency transformation.

- potentially small-sized integrated circuit implementation.

- there are no drift problems which arise in the realization

of stable filters with very high Q's.

- the digital technology make it possible to produce a filter meeting the exact design requirements.

However, digital filters also have limitations. The limited word length leads to a Quantization error. Round off error, overflow and underflow problems in the computational process also have effects on the system operation. Much work has been published on attempts to overcome these problems, see for example, reference (3).

Chapter 2.

Digital Filtering.

In this chapter, some fundamental principles of digital filtering will be discussed to provide a basis for the design of a digital filter. We will begin with an introduction of the Z-domain or Z-transform which is the basic mathematical tool of digital filtering.

2.1. The Z-transform.

The Z-transform is a transformation that operates on a sequence of numbers producing a function of the complex variable Z. It is used as a tool for the solution of linear constant coefficient difference equations, just as the Laplace transform is appropriate for the solution of linear constant coefficient differential equations. In otherwords, the Z-transform is the Laplace transform of a sampled function.

Let a function of time $X(t)$ be sampled by the switch, represented schematically in Fig. 2.1. This switch can be realized as a Dirac delta function, (or impulse), $\delta(t)$, defined by

$$\delta(t-nT) = \begin{cases} \infty & t=nT \\ \\ 0 & t \neq nT \end{cases}$$
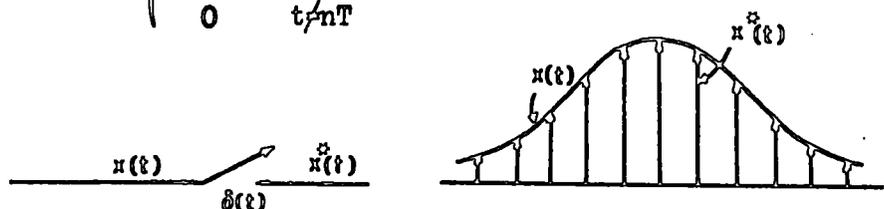
Fig. 2.1.

Where n and T are the nth sampling time and the time interval κ respectively, the area under $\delta(t-nT)$ is unity.

If x(t) is sampled during the interval $[0, \infty]$, the sampled function $x^*(t)$ is a time sequence of approximately weighted impulses and can be written as

$$x^*(t) = x(t) \sum_{n=0}^{\infty} \delta(t-nT) \qquad 2.1.2$$

$$= \sum_{n=0}^{\infty} x(nT) \delta(t-nT) \qquad 2.1.3$$

Taking the Laplace transform of equation 2.1.3, and recalling that $\mathcal{L}\{\delta(t-nT)\}$ is e^{-snT}, we obtain

$$\mathcal{L}\{x^*(t)\} = X^*(S) = \sum_{n=0}^{\infty} x(nT) e^{-snT}$$

or

$$= \frac{1}{T} \sum X(S + jn\omega_0) \qquad 2.1.4$$

where ω_0 is a sampling frequency.

It is seen that the inverse Laplace of this equation can be found - neither in the available table of Laplace transforms, nor can it be expanded in partial fractions. Thus a new procedure has been developed for - simplifying the inverse transformation procedure.

If we define,

$$Z = e^{sT} \qquad 2.1.5$$

Therefore, equation 2.1.4 can be expressed as a function of z, and referred to as the z-transform.

ie

$$X(Z) = \sum_{n=0}^{\infty} x(nT) Z^{-n} \qquad 2.1.6$$

The operation of taking the z-transform of a sequence may be denoted by

$$X(Z) = Z\{Xn\} \qquad 2.1.7$$

Because the function is being sampled during the interval $[0, \infty]$ equation 2.1.6 can be called the one sided z-transform. But if the function is being sampled during the interval $[-\infty, \infty]$, the two sided

Z-transform would result, and the transformation becomes

$$X(Z) = \sum_{n=0}^{\infty} x(nT) Z^{-n} \qquad\qquad 2.1.8.$$

It should be noted that the one sided Z-transform is a power series in the variable Z^{-1}, so it has all the properties associated with power series.

For example ; To find the Z-transform of the exponential function, $f(t) = e^{-aT}$

$$f(nT) = e^{-anT}$$

From equation 2.1.4.,

$$F(Z) = \sum_{n=0}^{\infty} e^{-anT} Z^{-n}$$

$$= \sum \left[\frac{e^{-aT}}{Z}\right]^{n}$$

The right-hand term is a geometric series having the first term 1 and geometric ratio of $\dfrac{e^{-aT}}{Z}$. Therefore,

$$F(Z) = \frac{1}{1 - e^{-aT} Z^{-1}}$$

The essential properties of the Z-transform are given below :.

1. Linearity : $Z\left\{af_n + bg_n\right\} = aZ\left\{f_n\right\} + bZ\left\{g_n\right\}$

where a and b are constants.

2. Shifting (delay) : $Z\left\{f_{n-k}\right\} = Z^{-k} Z\left\{f_n\right\}$

eg. $Z\left\{x(n-k)T\right\} = Z^{-k} X(Z)$

3. Convolution : $Z\left\{f_{k-n} \circ g_n\right\} = Z\left\{f_n\right\} \circ Z\left\{g_n\right\}$

eg. $Z\left\{\sum_{n=0}^{\infty} f[(k-n)T] \cdot g(nT)\right\} = F(Z) \cdot G(Z)$

4. Inverse Z-transform : $x(nT) = \dfrac{1}{2\pi j} \oint_C X(Z) Z^{n-1} dZ$

where c is a closed curve in the Z-plane which .loses all the poles of X(Z) and the origin.

Further details of the Z-transform can be found in the literature (17).

Relationship between the Z-domain and the S-domain.

Consider the complex variable Z, from the definition

$$Z = e^{sT}$$

$$= e^{\sigma T} e^{j\omega T}$$

where $S = \sigma + j\omega$

σ and ω are the real part and imaginary part of S respectively, therefore, Z has magnitude, $\quad |Z| = e^{\sigma T}$

and phase angle $\quad \Psi = \omega T$

It can be seen that :

$\sigma > 0 \implies |Z| > 1$; the entire left half of the S-plane can be mapped into the interior of the unit circle of the Z-plane

$\sigma = 0 \implies |Z| = 1$; the imaginary axis of the S-plane can be mapped on the circumference of the unit circle of the Z-plane.

$\sigma < 0 \implies |Z| < 1$; the entire right half of the S-plane can be mapped into the exterior of the unit circle of the Z-plane.

It should be noted that points on the S-plane do not map into the Z-plane in a one-to-one manner, because the unit circle on the Z-plane repeats at interval of the sampling frequency ω_s. For example, the points $.3\omega_0, 1.3\omega_0, 2.3\omega_0, \ldots\ldots$ etc will map to $.3\omega_0$ as shown in Fig. 2.2
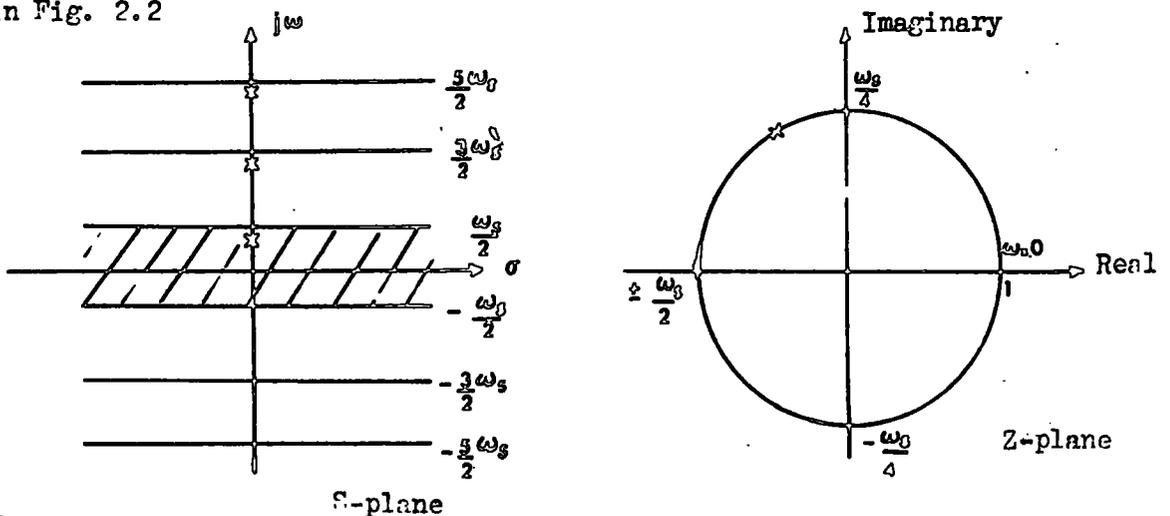
FIG. 2.2

2.2. Difference equations and digital transfer functions.

Difference equations (17) are a mathematical language used to represent a linear discrete system in time domain, and are the counterpart of the differential equations in linear continuous systems.

For a linear discrete system having input $x(nT)$ and output $y(nT)$ a Nth order difference equation can be written, with constant coefficients, as

$$y(nT) + b_1 y[(n-1)T] + \ldots + b_M y[(n-M)T] = a_0 x(nT) + \ldots + a_N x[(n-N)T]$$

$$2.2.1$$

Taking the z-transform, yields

$$Y(Z) + b_1 Z^{-1} Y(Z) + \ldots + b_M Z^{-M} Y(Z) = a_0 X(Z) + a_1 Z^{-1} X(Z) + \ldots + a_N Z^{-N} X(Z)$$

$$2.2.2$$

$$Y(Z)\left[1 + \sum_{i=1}^{M} b_i Z^{-i}\right] = X(Z) \sum_{i=0}^{N} a_i Z^{-i} \qquad 2.2.3$$

or

$$Y(Z) = H(Z) X(Z) \qquad 2.2.4$$

where

$$H(Z) = \frac{\sum_{i=0}^{N} a_i Z^{-i}}{1 + \sum_{i=1}^{M} b_i Z^{-i}} \qquad 2.2.5$$

$H(Z)$ is a proper rational function of Z^{-1} and may be called the digital transfer function of the system. Therefore, if the input signal $X(Z)$ and the digital transfer function $H(Z)$ are known, the output signal $Y(Z)$ can be determined. So it is obvious that the first stage in the design of a digital filter is to find the coefficients of the transfer function

Equation 2.2.1 can be rearranged as

$$Y(nT) = a_0 x(nT) + a_1 x[(n-1)T] + \ldots + a_N x[(n-N)T]$$

$$- b_1 Y[(n-1)T] - b_2 Y[(n-2)T] - \ldots - b_M Y[(n-M)T] \qquad 2.2.6$$

and it is seen that if the input signal x(t) is sampled, then the
filtering reduces to a computational process, which may be done with
a digital computer.

2.3. Frequency response of digital filters.

In continuous filtering, the frequency response is determined
by the values of the transfer function of the filters on the
imaginary axis. Similarly, the frequency response of a digital
filter can be obtained from values of $H(Z)$ on the unit circle
$\left[\ |Z| = 1\ \right]$

To determine the amplitude and phase characteristics, the poles
and zeros of the digital transfer function are plotted in the Z-
plane, as shown in Fig. 2.3. The magnitude response at a particular
frequency ωr can be determined from the product of the magnitudes
of the zero vectors, drawn to ωr on the unit circle, divided by the
product of the magnitudes of the pole vectors.

Let $\vec{Z_1}, \vec{Z_2}, \ldots \ldots \vec{Z_n}$ be the zero vectors, and

$\vec{P_1}, \vec{P_1}, \ldots \ldots \vec{P_m}$ be the pole vectors.

Therefore, the magnitude response

$$|H(Z)| = \frac{\prod\limits_{i=0}^{n} \vec{Z_i}}{\prod\limits_{i=0}^{m} \vec{P_i}}$$

for any ωr
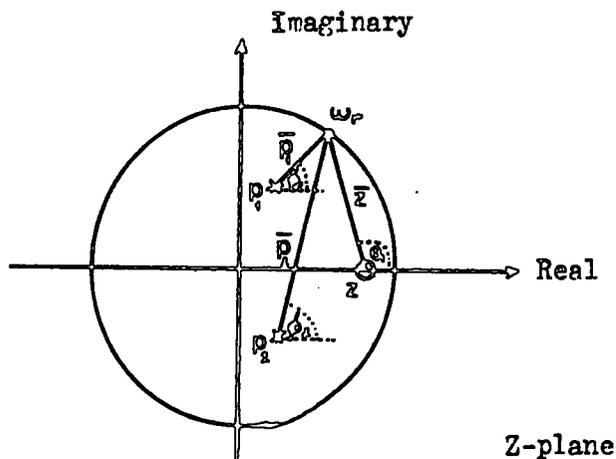
Fig. 2.3.

The phase response is obtained by combining the angles which the pole and zero vectors make with respect to the positive real axis. The total phase angle at a frequency ωr is the summation of the zero angles minus the summation of the pole angles.

Let $\Theta_1, \Theta_2, \ldots \ldots \Theta_N$ be the zero angles, and

$\Phi_1, \Phi_2, \ldots \ldots \Phi_M$ be the pole angles.

Therefore, the total phase angle $\Psi = \sum_{i=1}^{N} \Theta_i - \sum_{i=1}^{M} \Phi_i$

It is also possible to obtain the magnitude and phase response directly from the transfer function with the following substitution.

$$Z^{-i} = e^{-ji\omega T} = \cos i\omega T - j \sin i\omega T$$

Note that

1. The magnitude response characteristic is seen to repeat after reaching $\omega_0, 2\omega_0$ and so on, and is symmetrical about $\frac{\omega_s}{2}$.

2. The phase response characteristic may repeat after reaching $\omega_0, 2\omega_0 \ldots$ but this is often not the case.

Further details of the amplitude and phase characteristics can be found from Nowak and Schmid. (18).

2.4. Realization of digital filters.

2.4.1. Terminology.

The terminology shown in Fig. 2.4. is recommended (19) to be used in the block diagrams of realizations of digital filters.

unit delay

$$Y(nT) = x[(n-1)T]$$
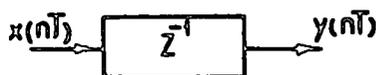
Adder/subtractor

$$r(nT) = x(nT) \pm Y(nT)$$
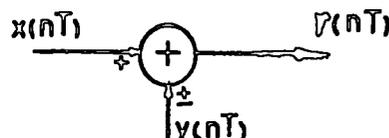
constant multiplier

$$Y(nT) = k \, x(nT)$$
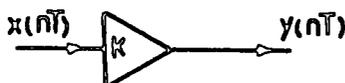
Branching operation

Fig. 2.4. THE RECOMMENDED TERMINOLOGY USED IN DIGITAL FILTERING.

2.4.2. Realization of recursive digital filters.

Assume that a digital filter has been designed having the transfer function.

$$H(Z) = \frac{\sum\limits_{i=0}^{n} a_i Z^{-i}}{1 + \sum\limits_{i=1}^{m} b_i Z^{-i}}$$ \hfill 2.4.1.

Thus, in the z-domain, we have

$$Y(Z) = \sum\limits_{i=0}^{n} a_i Z^{-i} X(Z) - \sum\limits_{i=1}^{m} b_i Z^{-i} Y(Z)$$ \hfill 2.4.2.

or in the time domain

$$Y(nT) = \sum\limits_{i=0}^{n} a_i x(nT-iT) - \sum\limits_{i=1}^{m} b_i Y(nT-iT)$$ \hfill 2.4.3.

A recursive digital filter, $(b_i \neq 0)$, can be realized in the following forms.

1. Direct form 1.

Equation 2.4.3. can be realized directly as in Fig. 2.5.
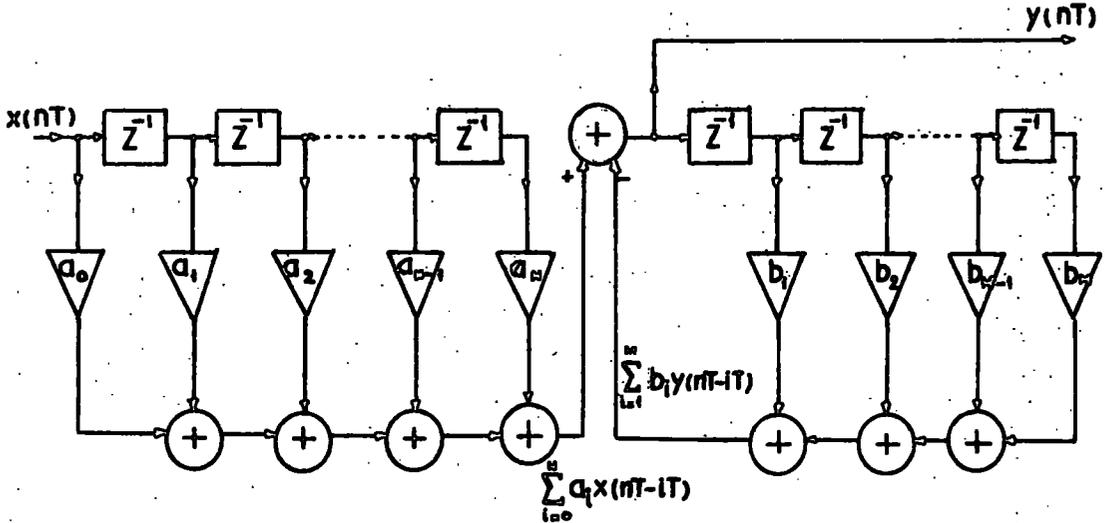
Fig. 2.5 Block diagram representation of direct form 1 for an Nth order filter.

If we define the intermediate state $W(Z)$

$$W(Z) \; = \; \frac{X(Z)}{1+\sum_{i=1}^{M} b_i Z^{-i}}$$

2.4.4.

Therefore, equation 2.4.2. can be written as

$$Y(Z) \; = \; \sum_{i=0}^{N} a_i Z^{-i} W(Z)$$

2.4.5.

or in the time domain

$$w(nT) \; = \; X(nT) - \sum_{i=1}^{M} b_i W(nT-iT)$$

2.4.6.

$$Y(nT) \; = \; \sum_{i=0}^{N} a_i W(nT-iT)$$

2.4.7.

Therefore equation 2.4.3. can also be realized as shown in Fig. 2.6.
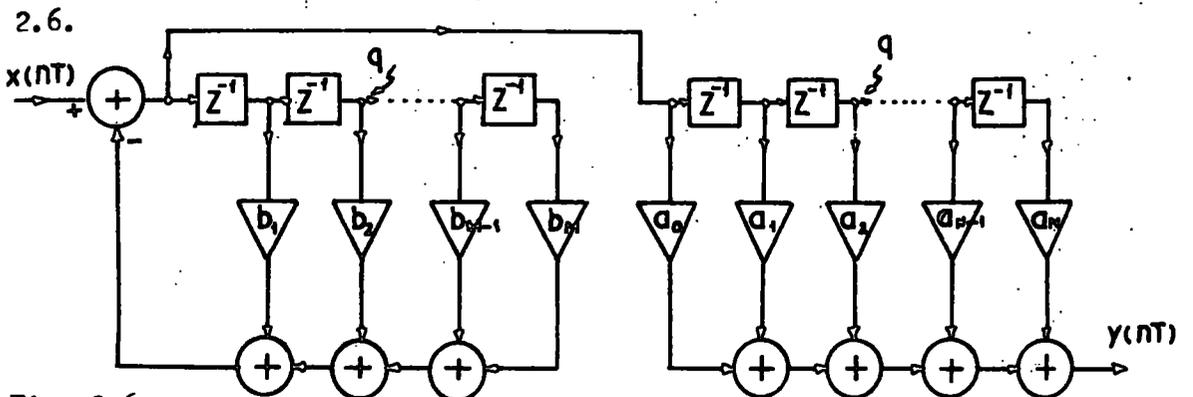
Fig. 2.6.

2. Direct form II of the canonic form.

From the Fig. 2.6, it is seen that signals at the points marked q are identical. Therefore, Fig. 2.6 may be simplified to Fig. 2.7.
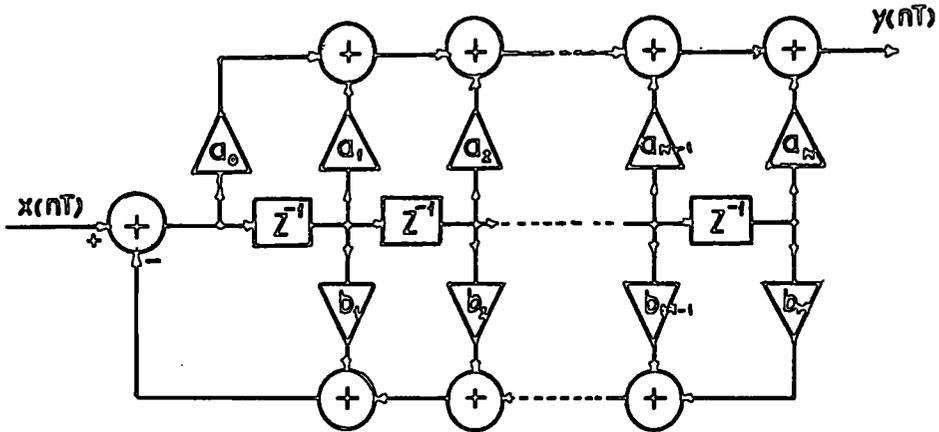
Fig. 2.7 Block diagram representation of direct form II for an Nth order filter.

Since this form has the minimum number of multipliers, adders and delay elements, it is called canonic.

3. Parallel form.

A digital transfer function $H(Z)$ can be written in the form of equation 2.4.1 , for example,

$$H(Z) \;=\; C + H_1(Z) + H_2(Z) + \cdots + H_k(Z)$$
$$=\; C + \sum_{i=1}^{k} H_i(Z)$$

where $H_i(Z)$ is ratio of polynomials.

In this form, the output $Y(nT)$ is the summation of the outputs of several subfilters. Each of these subfilters can be realized in either of the direct forms and second-order filters are recommended (24) If the canonic forms are used, it will be called the parallel canonic form. This form tends to be less sensitive to quantization effect than the direct form (24), and can be realized as in Fig. 2.8.
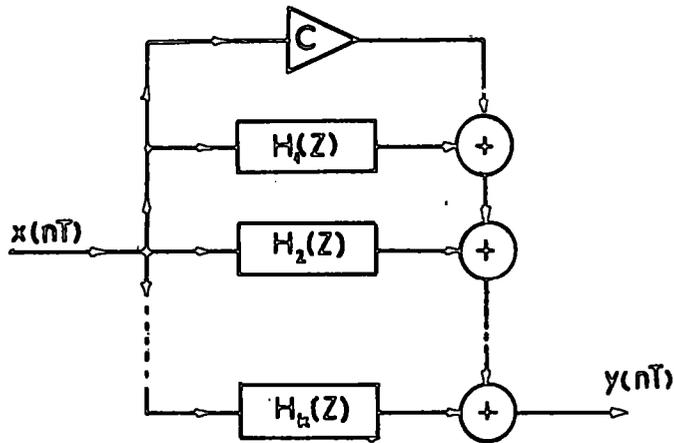
Fig. 2.8 Block diagram representation of the parallel form.

/1. Cascade form.

Another form of H(Z) can be written as

$$H(Z) = \frac{C(Z-Z_1)(Z-Z_2)\ldots\ldots(Z-Z_u)}{(Z-P_1)(Z-P_2)\ldots\ldots(Z-P_u)}$$

$$= C \prod_{i=1}^{u} H_i(Z)$$

Where $H_i(Z)$ represents a subfilter, and again recommended to be second-order. (24). Because H(Z) is the product of the subfilter functions, this is called the cascade form. Again, if canonic forms are used as subfilters, this is referred to as the cascade-canonic form. This form can be depicted in Fig. 2.9.
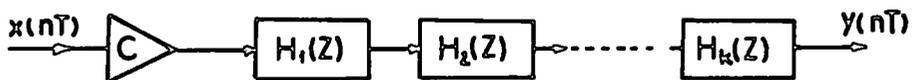
Fig. 2.9 Block diagram representation of the cascade form.

2.4.3. Realization of nonrecursive digital filters.

For a nonrecursive filter, the transfer function H(Z) is a polynomial in Z^{-1} rather than a ratio of polynomials, (all b_i's in the equation 2.4.1. are zero). Therefore, the output is a simple linear weighting of the present and previous samples of the input, and the filter can be realized as in Fig. 2.10.
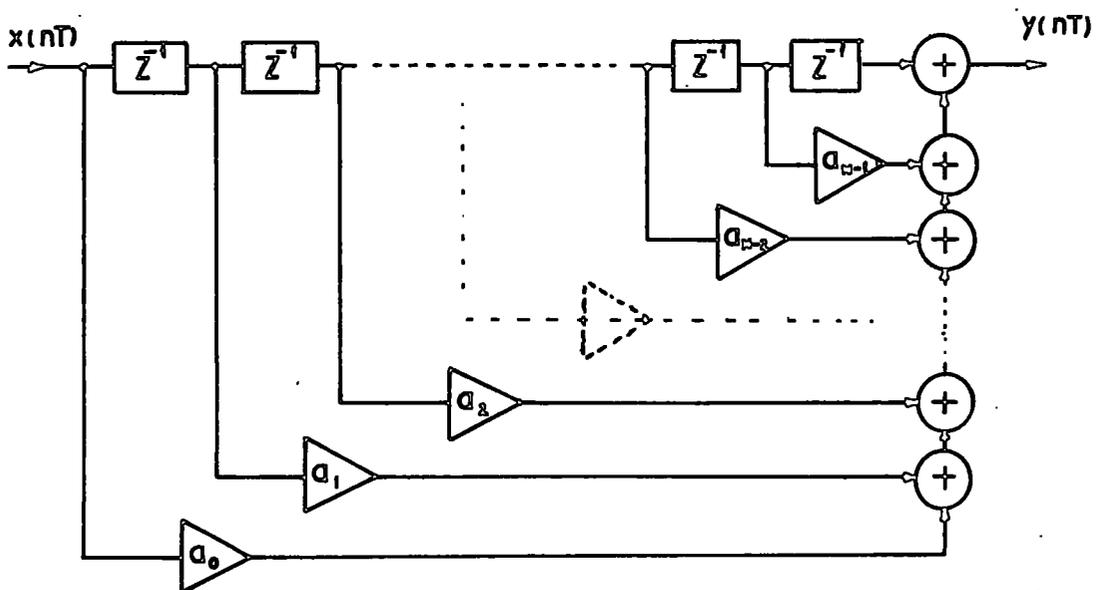
Fig. 2.10 Block diagram representation of a nonrecursive filter.

Each of the forms discussed above has its own advantages and limitations, and the choice depends on the requirement. But, the direct forms are not recommended for a high-order filter because of their sensitivity to quantization effects. Instead the parallel form or the cascade form is preferable. This topic has been discussed and illustrated in the literatures, see for examples (1), (7), (25).

2.5. Frequency transformation for digital filters.

Frequency transformation is a method of synthesising a desired filter from a given normalized filter. The idea of the transformation is to change the transfer function of a given filter to the desired filter. A. C. Constantinides has developed a theory of these transformations for digital filters on the Z-plane, without reference to the frequency transformations for analogue filters. The transformations from a normalized lowpass filter to any other type of filter (20) and at any cut off frequency (21) are given, and can be summerized as follows :

	Lowpass-to-highpass	Lowpass-to-bandpass	Lowpass-to-bandstop
Transformation	$z^{-1} \Rightarrow -z^{-1}$	$z^{-1} \Rightarrow \dfrac{-z^{-1}(z^{-1}-\alpha)}{1-\alpha z^{-1}}$	$z^{-1} \Rightarrow \dfrac{z^{-1}(z^{-1}-\alpha)}{1-\alpha z^{-1}}$
cut off frequencies	$W_{CL} = \dfrac{W_s}{2} - W_{CH}$	$W_a = W_2 - W_1$	$W_{CL} = \dfrac{W_s}{2} - (W_2 - W_1)$
centre frequencies	$\dfrac{W_s}{2}$	W_o	W_o

where $\alpha = \cos\left(\dfrac{2\pi W_o}{W_s}\right)$, and

W_{CL} is the cut off frequency of lowpass filter

W_{CH} is the cut off frequency of highpass filter

W_s is the centre frequency

W_1 is the lower cut off frequency

W_2 is the upper cut off frequency.

2.6 Quantization effects.

On programming a digital filter with a limited word-length, all data and parameters are quantized to a finite set of allowable values resulting in the introduction of an error which is referred to as the quantization effect. Here, the quantization has been defined (3) as the replacement of the exact value of a quantity by the value of the nearest of a set of levels, (the quantization levels), differing by steps of the width $2^{-(q-1)}$, where q is the word-length. This width may be considered as the width of quantization.

The results of the quantization effect are quantization noises, - degeneration in performance and perhaps instabilitity of the filter. As an approximation, the noises may be assumed statistically independent with zero mean, and will be treated as random quantities.

Consider the difference equation of a digital filter

$$Y(nT) = \sum_{k=0}^{\omega} a_k x(n-k)T - \sum_{k=1}^{\omega} b_k Y(n-k)T \qquad 2.6.1.$$

It is seen that the sources of error due to the limited word-length can be

1. the inaccuracy of input signal $x(nT)$, referred to as input quantization noise ;

2. the effect of rounding and truncation in arithmetic operations, referred to as roundoff noise ;

3. the inaccuracy of the coefficients, a_k's and b_k's , referred to as coefficient rounding effect.

Input quantization noise.

The input quantization noise is introduced by the analogue-to-digital converter. It is regarded as white noise, (having infinite bandwidth and its amplitude can assume a completely different value in an infinitesimally short period of time) , and being independent on the input signals. The mean-square value used in assessing this noise in the A/D converter is readily shown (22) as $\frac{\Delta^2}{12}$, where Δ is the width of quantization, and its mean-square value at the output of the filter can be determined by using the discrete form of Parseval's theorem (17) and the properties of random signal. For example, in the canonical formed filter, it can be deduced as (see section 4.7)

$$\frac{1}{2\pi j} \oint H(Z) H(Z^{-1}) \frac{\Delta^2}{12} \frac{dZ}{Z} \qquad \qquad 2.6.2.$$

Where $H(Z)$ is the transfer function of the filter.

Round off noise.

The round off noise is generated in the arithmetic unit when multiplication is performed, and the term "round off" has been used to include the case of truncation as well as rounding. It is assumed to be uncorrelated from sample to sample and from multiplier to multiplier. From this assumption, it follows that the mean-square values of this noise produced at each multiplication are $\frac{\Delta^2}{12}$ and $\frac{\Delta^2}{3}$, for rounded operation and truncated operation respectively.

Liu (23) has analysed the round off noise for each form of

realization, and, for a fixed-point filter, a general expression of the mean-square value of this noise at the output of the filter has been given as

$$\frac{1}{2\pi j} \oint \Phi_{ee}(Z) \frac{dZ}{Z} \qquad 2.6.3.$$

where $\Phi_{ee}(Z)$ is the power spectrum density of the round off noise term for each form of realization, (see also section 4.7).

It has been shown that the accuracy obtained by a direct form of realization of a high-order filter is considerably less than with either cascade or parallel forms of the same filter. Therefore, the first or second-order subfilters are recommended to be used as basic building blocks for higher-order filters.

Coefficient rounding effect.

Because the coefficients determine the performance of a filter, inaccuracy of the coefficients can cause degeneration in the frequency response of the filter. For example, the cut off frequency, attenuation rates and ripple may all be affected. Fortunately, the stability of the filter is hardly affected by the coefficient rounding. The loss of stability in a realization will normally occur only after the deviation between the actual and ideal frequency responses has become intolerable.

Knowles and Olcayto (25) have given a measure of the degeneration in filter performance due to this effect by following statistical mean-square convergence criterion :

$$\overline{\sigma_\omega^2} = \frac{T}{2\pi} \int_0^{\frac{2\pi}{T}} \left| F^*(j\omega) \right|^2 d\omega \qquad 2.6.4.$$

where $F^*(j\omega)$ represents the difference of the frequency responses of the actual and ideal filters and T is the sampling period.

In the z-domain, eg. 2.6.4., can be rewritten as

$$\overline{\sigma_\omega^2} = \overline{\frac{1}{2\pi j} \oint F(Z)F(Z^{-1}) \frac{dZ}{Z}} \qquad 2.6.5.$$

To calculate the minimum word-length, the expression

$$3\sqrt{\overline{\sigma_\omega^2}} \leqslant \left| \text{Acceptable Gain Fluctuation} \right| \qquad 2.6.$$

must be satisfied, provided that the output noise due to the other errors is also acceptable with this word-length, and the minimum word-length can be found from graphs plotted between $\sqrt{\overline{\sigma_\omega^2}}$ and word-length for each form of realization. (25) For example, a filter with 1 dB ripple in the passband and -12 dB attenuation in the rejection band is to be designed. For a worst case design, 2 dB fluctuation is allowable in the rejection band. Using the Knowles and Olcayto's method, for a parallel form of realization, this specification can be programmed with a minimum 8 bit word.

Knowles and Olcayto have also indicated that the actual degeneration in performance of a filter for the cascade form of realization is less than that of the direct form but greater than that of the parallel form.

The other sources of error that may cause a serious degeneration in performance and instability of the filter are overflow and underflow. Their effects are obvious, and can be cured by using an appropriate number representation, (the signed 2's complement representation is recommended), the type of arithmetic used, and, if necessary, appropriate scaling factors.

It can be said that quantization effects are an important

consideration in designing a filter. Many workers using different

approaches, have analysed and discussed these effects in the

literature, see for example (23) (25) (29).

Chapter 3.

Basic Design of Digital Filters.

3.1. Approximation of Digital Filters.

As it is a computational process defined by linear difference equations with constant coefficients, the design of digital filters is essentially the process of determining the value of these coefficients.

The design of digital filters can be approached from two directions : First, they can be referred to the well-known analogue filters and second, they can be designed directly from the specification without reference to analogue filters by the method of frequency selectivity. Both methods have their own advantages and both have several different approaches. See for example, Rader and Gold (3). The recent designs of digital filters have mostly used the second method to meet a high specification. However, in this project, the first method is chosen. The digital filter is referred to the well-known Chebyshev prototype.

The sampled-data transformation is used to design digital filters from analogue filters. A digital filter obtained from this transformation can be called a sampled-data filter and can be defined as a filter which accepts input and produces outputs only at specific instants of time called sample points. The sampled-data method is widely used in linear discrete systems and control systems.

In this chapter, the sampled-data transformation will be introduced and briefly discussed. Further details of this transformation can be found in the literature. (1) (5).

3.2. Sampled-data transformation.

Sampled-data transformation is an important tool in the design

of digital filters. It changes the analogue filter function, described

in terms of the Laplace transform complex variable s, to a digital

filter function, described in term of the unit delay variable z^{-1}.

This transformation is usually used in the case of recursive

realization and at least three different transformations are widely

used. They are ;   the standard z-transformation, the bilinear

z-transformation and the matched z-transformation. Each of these

transformations has its own advantages and limitations and it is not

immediately clear from the literature which is the best. The

selection depends on the requirements and the type of the selected

prototype analogue filter. However, for any particular design

requirement, one of these transformations can be used successfully.

See for example, R. M. Golden (4) (5).

In this section, the standard z-transformation and the matched

z-transformation will be briefly introduced and discussed but the

bilinear z-transformation will be discussed in detail in the

succeeding section.

The standard z-transformation.

The standard z-transformation, (or impulse invarient

transformation), is based on the fact that the discrete responses

of the derived digital filter to an impulse function will be the

samples of the continuous impulse response of the corresponding

continuous (analogue) filter. In other words, a system with sampled inputs is equivalent to a continuous system with sampled outputs.

Consider for example, a given continuous transfer function, having a simple pole,

$$H(S) = \frac{A}{S + a} \qquad 3.1.$$

This can be transformed to a digital transfer function, by taking the inverse Laplace transform of $H(S)$ to obtain the impulse response and then applying the Z-transformation, (see chapter 2), to both sides of the equation, gives

$$H(Z) = \frac{A}{1 - e^{aT} Z^{-1}} \qquad 3.2.$$

It should be noted here that the gain A in equation 3.2. is not compensated to account for the gain reduction, $\frac{1}{T}$, in the Fourier transform.

A limitation of the standard Z-transformation is the aliasing-effect, caused by a signal which is not bandlimited. This can be shown from the equation,

$$F(Z) = \sum_{n=-\infty}^{\infty} f(nT) \, e^{-snT}$$

$$= \frac{1}{T} \sum_{k=-\infty}^{\infty} F(S + j\omega_s k)$$

$F(Z)$ repeats itself every sampling frequency ω_s , and the spectra of a band limited signal and a non-band limited signal

can be depicted in Fig. 3.1.

(a) A bandlimited signal.

F(S)

Fig. 3.1.

(b) Spectrum of a
bandlimited signal.

F(Z)

$-\omega_s$ $-\dfrac{\omega_s}{2}$ 0 $\dfrac{\omega_s}{2}$ ω_0

(c) A non-bandlimited signal.

F(S)

(d) Spectrum of a non-
bandlimited signal.

F(Z)

$-\omega_0$ $-\dfrac{\omega_s}{2}$ 0 $\dfrac{\omega_s}{2}$ ω_0
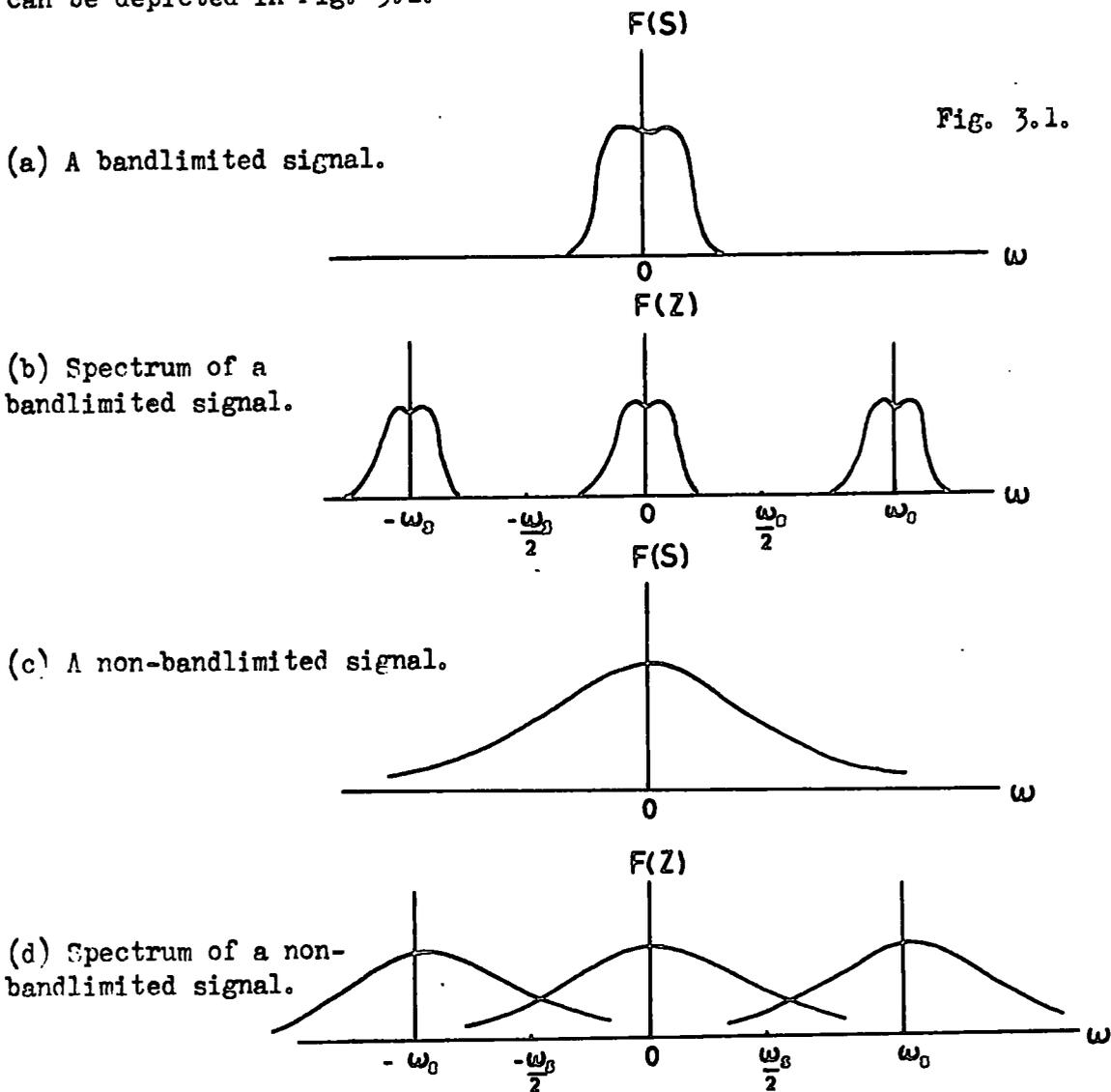
Thus, the standard Z-transformation is only satisfactory when $F(Z)$
is bandlimited. However, this problem can be avoided by passing the

input signal through a band limiting lowpass filter, (sometimes

called a guard filter).

An advantage of this transformation is that it preserves the

shape of the impulse-time response. The use of this transformation

is demonstrated for exampled by R. M. Golden (5).

The matched Z-transformation.

The matched Z-transformation is a transformation generating

a digital transfer function with poles and zeros matched to those

of the continuous transfer function. The mapping transformation

for the poles and zeros of the continuous function is given by

$$S \longrightarrow e^{sT} = Z$$

For example, a real pole or zero could be transformed according to

$$S - U \longrightarrow 1 - Z^{-1} e^{uT}$$

It should be noted that the poles of $H(Z)$ are identical to

those obtained by the standard Z-transformation, but the zeros do

not correspond. This transformation preserves the shape of the

frequency response characteristic and can be used for all types

of filters, but may require modification by insertion of

additional zeros, $(1 + Z^{-1})^N$, at the half-sampling frequency,

where N is the order of the half-sampling frequency zero desired

(5).

3.3. The Bilinear Z-transformation.

The disadvantage of the standard Z-transformation producing **the**

aliasing effect led to the development of the bilinear Z-

transformation. As we are going to use this transformation, it

is worthwhile considering it in detail.

In deriving this transformation, it should be recalled that a

digital filter is defined by a set of difference equations with

constant coefficients and a recursive filter is one whose output

is a function of the inputs and the previous outputs.

Consider the trapesoidal rule integrator for example, which

approximates the integral of the input signal by a summation of

trapesoids whose width is equal to the sampling interval T , as

shown in Fig. 3.2.
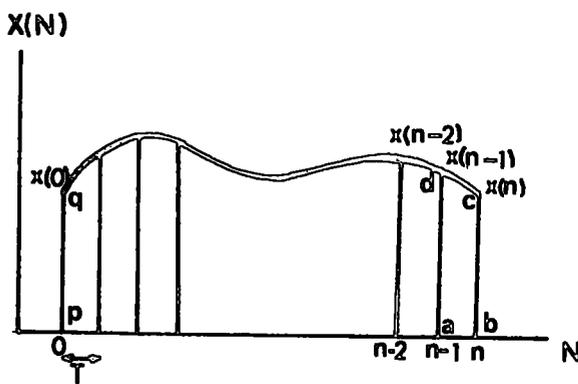
Fig. 3.2.

area of trapesoid abcd

$$= x(n).T + \frac{1}{2}.T.\left[x(n-1) - x(n)\right]$$

$$= \frac{T}{2}\left[x(n) + x(n-1)\right]$$

If $Y(n)$, the output, is defined as an area pbcq , and

$Y(n-1)$, a previous output, is defined as an area padq ,

a difference equation can be

$$Y(n) = \frac{T}{2}\left[x(n) + x(n-1)\right] + Y(n-1) \qquad 3.3.1.$$

Taking the Z-transform of both sides of the equation, yields

$$Y(Z) = \frac{T}{2}\left[X(Z) + Z^{-1}X(Z)\right] + Z^{-1}Y(Z) \qquad 3.3.2.$$

or

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{T}{2}\left[\frac{1 + Z^{-1}}{1 - Z^{-1}}\right] \qquad 3.3.3.$$

Since this equation is a good approximation to an integrator, $\frac{1}{j\omega}$

or $\frac{1}{s}$, it follows that (30)

$$\frac{1}{H(Z)} = \frac{2}{T}\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right]$$

or $$S \longrightarrow \frac{2}{T}\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right] = \frac{2}{T}\tanh\left(\frac{S_i T}{2}\right) \qquad 3.3.4.$$

is a good approximation for S or $j\omega$.

Thus, the bilinear Z-transformation is given by equation (3.3.4.), in which S in a continuous transfer function is replaced by $\frac{2}{T}\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right]$ to obtain the digital transfer function.

Now consider equation (3.3.4.), it can be written as

$$\omega = \frac{2}{T}\tan\left(\frac{\omega_i T}{2}\right) \qquad 3.3.5.$$
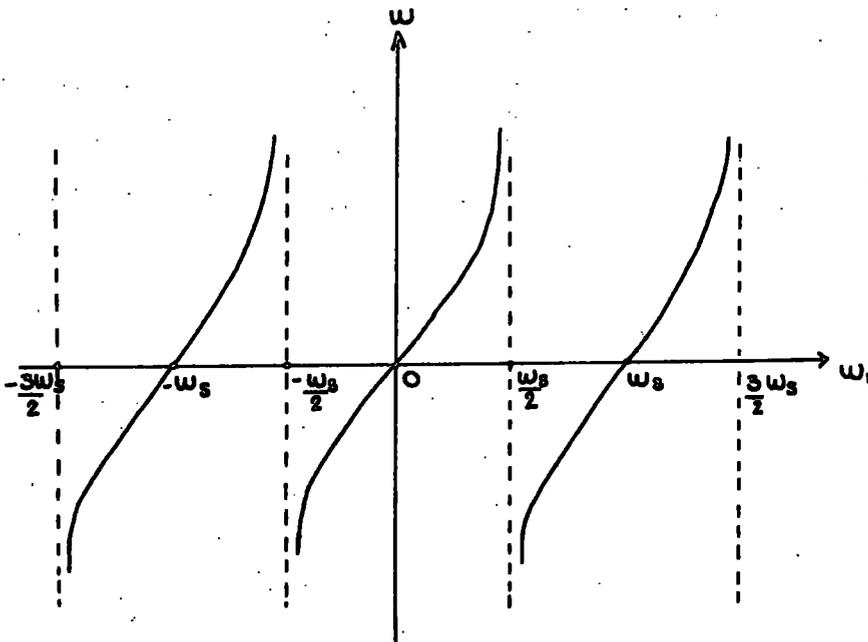
and being depicted in Fig. 3.3.2.

Fig. 3.3.2.

Since a transfer function $H(Z)$ must also be periodic in ω

period ω_s , this transformation will cause $H(S)$ to be mapped

identically in each of the other vertical strips bounded by the

line $\omega_1 = \left(n-\frac{1}{2}\right)\omega_s$ and $\omega_1 = \left(n+\frac{1}{2}\right)\omega_s$ in ω_1 plane, where n

is an integer. In other words, in term of Z^{-1} , this transformation

will uniquely map the left half of the S-plane into the exterior

of the unit circle in the Z^{-1} plane, or into the interior of the

unit circle in Z-plane. Then aliasing effects are eliminated,

since no folding occurs.

This transformation is an algebraic transformation which is

easy to use and can preserve a flat magnitude gain-frequency

response characteristics. It is suitable for all filter types

especially wide bandwidth filter and can be realized in either

parallel or serial form.

However, this transformation has the disadvantage that there

is distortion on the frequency axis when the critical frequencies

are near the half-sampling frequency. This can be seen from

equation (3.3.5.) and in Fig. 3.3.2.

However, this problem can be eliminated by correcting the

critical frequencies or prewarping the continuous filter design

in the opposite sense such that when we apply the Z-transformation

the critical frequencies will be shifted back to the desired

values. This prewarping process can be applied by the equation

$$\omega_c = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right) \qquad 3.3.6.$$

where ω_c is a computed cut off frequency, and

ω_d is a desired cut off frequency

or

$$f_c = \frac{2}{T} \tan\left(\frac{f_d T}{2}\right) \qquad 3.3.7.$$

It should be noted that R. M. Golden has shown (5) that the digital filter designed using the bilinear Z-transformation applied to a warped continuous bandstop filter yields a steeper attenuation slope on the high-frequency side of the bandstop and also changes the frequencies at which the minima and maxima of the in-band and out-of-band ripple occur.

3.4. Determination of the coefficients.

Consider a continuous filter of order N, having transfer function

$$H(S) = \prod_{i=1}^{k} \frac{1}{(S-(a_i \pm j b_i))} \qquad 3.4.1.$$

where $(a_i \pm j b_i)$ are the ith complex and its conjugate pole positions, having a_i as the real part and b_i as the imaginary part, and k is the integer part of $\frac{N+1}{2}$.

In parallel form, equation (3.4.1.) can be written as

$$H(S) = \sum_{i=1}^{k} \frac{(c_i \pm j d_i)}{(S-(a_i \pm j b_i))} \qquad 3.4.2.$$

where $(c_i \pm j d_i)$ are the residues evaluated at the ith pole and its conjugate, having c_i as the real part and d_i as the

imaginary part. Alternatively, $H(S)$ can be expressed as a parallel combination of second order subfilters, as suggested by Gold and Rader (24).

$$H(S) = C_0 + \sum_{i=1}^{k} H_i(S) \qquad 3.4.3.$$

where $H_i(S)$ is a second order subfilter, and C_0 is constant.

Such a second order subfilter, referring to equation (3.4.2.), can be written as

$$H_i(S) = \frac{(c_i + jd_i)}{(S-(a_i+jb_i))} + \frac{(c_i-jd_i)}{(S-(a_i-jb_i))} \qquad 3.4.4.$$

Using the bilinear Z-transformation, S will be replaced by $\frac{2}{T}\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right]$ and this gives the digital filter transfer function, (see Appendix A),

$$H_i(Z) = C_0 + \frac{\alpha_0 + \alpha_1 Z^{-1}}{1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}} \qquad 3.4.5.$$

where

$$C_0 = \frac{A_1}{\beta_2}$$

$$\alpha_0 = A_0 - \frac{A_1}{\beta_2}$$

$$\alpha_1 = A_0 + A_1\left(1 - \frac{\beta_1}{\beta_2}\right)$$

$$A_0 = \frac{cT}{D} - \frac{acT^2}{2D} - \frac{bdT^2}{2D} = \frac{T\left[c\left(1-\frac{aT}{2}\right) - d\left(\frac{bT}{2}\right)\right]}{D}$$

$$A_1 = -\frac{cT}{D} - \frac{acT^2}{2D} - \frac{bdT^2}{2D} = \frac{-T\left[c\left(1+\frac{aT}{2}\right) + d\left(\frac{bT}{2}\right)\right]}{D}$$

$$\beta_1 = \frac{-2\left[1-\left(\frac{aT}{2}\right)^2-\left(\frac{bT}{2}\right)^2\right]}{D}$$

$$\beta_2 = \frac{\left(1+\frac{aT}{2}\right)^2+\left(\frac{bT}{2}\right)^2}{D}$$

$$D = \left(1-\frac{aT}{2}\right)^2+\left(\frac{bT}{2}\right)^2$$

3.4.6.

Note that this has a normalized frequency ω of 1 , and S has to be replaced by $\frac{S}{\omega_c}$, where ω_c is the desired cut off frequency. Then equation 3.4.5. will be

$$H(Z) = \omega_c\left[C_0 + \frac{\alpha_0+\alpha_1 Z^{-1}}{1+\beta_1 Z^{-1}+\beta_2 Z^{-2}}\right]$$

and all a's and b's in equation 3.4.6. are replaced by a''s and b''s , where

$$a' = a.\omega_c \qquad\qquad b' = b.\omega_c$$

3.5. Number representation.

An understanding of number representation is necessary in the design of digital filters, not only because it enters into the implementation, but also because of quantization effects. Thus, it is worthwhile introducing the basic ideas of number representation here. For more detail see, for example, Y. Chu

(31)

A number N can be represented by an expression

$$N = \sum_{i=-m}^{n-1} d_i r^i \qquad 0 \leqslant d_i \leqslant (r-1)$$

where d_i is the ith digit,

r is the radix or base,

n is the number of digits, and

m is the number of fractional digits.

In digital computation, the radix 2 corresponds to the binary system is used in the implementation. When the sign of a binary number is defined by one of its digits, (usually the most significant digit), it is called a signed binary number.

Signed binary numbers can be represented in 3 forms as follows :

1. Signed magnitude representation. Here the number digits represent the magnitude or absolute value of the number.

2. Signed 2's complement representation. Here the number digits are in 2's complement form when the number is negative.

3. Signed 1's complement representation. As above, except that the number digits are in 1's complement form.

It should be noted that if the number is positive, the three representations are identical.

In general, if X is the given binary number of $m+n$ digits, the three representations can be written as below :

Signed magnitude representation

$$X = (-1) X_n \left\{ \sum_{i=-m}^{n-1} X_i 2^i \right\}$$

where X_n is a signed digit, 0 for a positive number and 1 for a

negative number.

Signed - 2's - complement representation

$$X = 0_x 2^n + \sum_{i=-m}^{n-1} X_i 2^i \qquad \text{when X is positive}$$

$$X = -1_x 2^n + \sum_{i=-m}^{n-1} \overline{X}_i 2^i + 2^{-m} \qquad \text{when X is negative,}$$

$$\text{and } \overline{X}_i = 1 - X_i$$

Signed - 1's - complement representation

$$X = 0_x 2^n + \sum_{i=-m}^{n-1} X_i 2^i \qquad \text{when X is positive}$$

$$X = -1_x 2^n + \sum_{i=-m}^{n-1} \overline{X}_i 2^i \qquad \text{when X is negative}$$

An example of three representations of signed binary number is

Representation	Number (+10)	Number (-6)
Signed magnitude	0,1010	1,0110
Signed-2's-complement	0,1010	1,1010
Signed-1's-complement	0,1010	1,1001

3.5.1. **Signed 2's complement representation.**

The 2's complement representation of a number is preferable in digital filter implementation using serial arithmetic because in addition and subtraction the signed bit can be treated as a number

bit and there is no need for advance knowledge of the signs or
relative magnitudes of the numbers being operated. Using this
representation, the overflow problems in the addition of more than
two numbers having the total sum in the range $-1 < X < 1$ can be
ignored. (11). In other words, no information is lost if overflow
or underflow occur in any of the partial sums, and the correct
total sum will be obtained.

However, although the signed bit can be treated as a number bit,
the convention of signed bit for this representation should be noted.
The negative and positive signed bits are 1 and 0 respectively.
In shifting processes, such as occur in multiplication, the signed
bit must not be altered. This means that when shifting a positive
number, all the added bits will be 0's . But, for a negative
number, shifted to the left, added bits are 0's ; and when shifted
to the right, added bits must be 1's .
Example, for a given negative number

$$X = 1,0011$$

$$X2^{1} = 1,0110 \qquad \text{shifted to the left.}$$

$$X2^{2} = 1,1100 \qquad \text{shifted to the left.}$$

$$X2^{-1} = 1,1001 \qquad \text{shifted to the right.}$$

$$X2^{-2} = 1,1100 \qquad \text{shifted to the right.}$$

As the numbers used in digital filters are fractional numbers
in the range $-1 < X < 1$, the smallest number we can represent
using an 8 bit word length is 0.0078125 (decimal), and the
largest is 0.99218750 (decimal).

Chapter 4.

Design of a programmable digital filter.

4.1. Specification.

A prototype programmable digital filter is to be designed and constructed with an 8 bit word-length. It should perform as a lowpass, highpass or bandpass filter, and having cut off frequencies at anywhere in the Nyquist interval. Here, the design of a bandpass filter will be considered as our limiting example with the following specification ;

1. the lower cut off frequency at 100 HZ and the upper cut off frequency at 10 KHz ,

2. the nominal slopes above and below the cut off frequencies are to be 12 dB per octave,

3. maximum ripple in the passband of 1 dB .

4. A sampling frequency of 40 KHz .

4.2. Digital bandpass filter obtained by multiplexing.

In continuous filter design, a bandpass filter can be achieved by cascading a lowpass filter and a highpass filter provided they are adequately isolated. In digital filtering, the use of time sharing techniques or multiplexing makes it possible to perform this operation with the same hardware used as a lowpass filter or a highpass filter alternately provided that the available time is long enough.

When digital bandpass filtering is performed, the arithmetic unit must represent both a lowpass and a highpass filter apparently at the same time. For a filter having the lower cut off frequency at f_1 and the upper cut off frequency at f_2 , a lowpass filter with cut off frequency f_2 and a highpass filter with cut off frequency f_1 will be required. The output signal of the lowpass filter is stored and fed back to become the input of the high pass filter by multiplexing. In other words, the data stream going to the arithmetic unit must be at a clock frequency of twice that for the sampling frequency, or at least four times that for the signal frequency. This technique is depicted in Fig. 4.1.
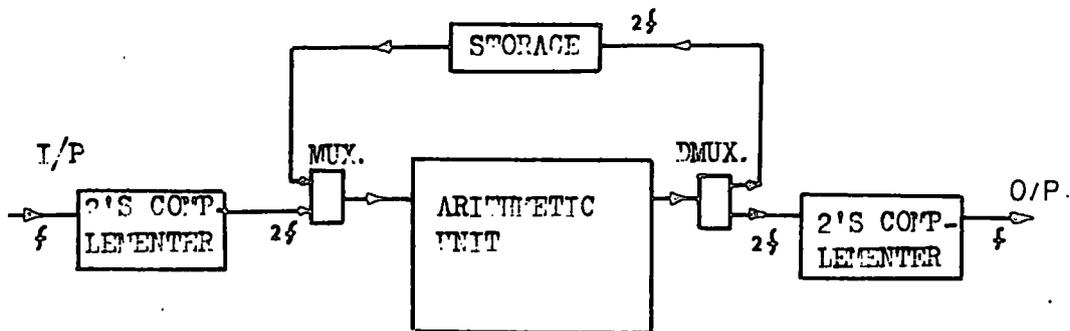

FIG. 4.1. THE DIAGRAM SHOWN THE MULTIPLEXING TECHNIQUE.

Thus far, it is obvious that a digital bandpass filter is obtained from lowpass and highpass filters of the same order. But using the frequency transformation, a highpass filter can be designed from a lowpass filter. Therefore, the design of a digital bandpass filter becomes simply the design of two digital lowpass filters.

For this example, the specification can be met with the type 1 Chebyshev filters.

4.3. Design of a digital lowpass filter.

Consider a digital lowpass filter with cut off frequency at 10 KHz , which will be the upper cut off frequency of the desired digital bandpass filter. Its specification must be as given in section 4.1. and can be depicted in Fig. 4.2.
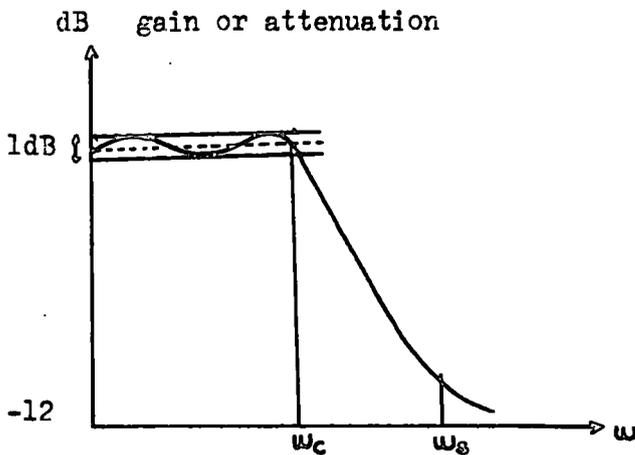
dB gain or attenuation

Fig. 4.2.

According to the prewarping process, the critical frequencies are transformed to corresponding analogue frequencies by the relation

$$W_A = \frac{2}{T} \tan \frac{W_p T}{2} \qquad 4.3.1.$$

So for the desired cut off frequencies, we have

$$W_c = 49421904 \qquad \text{radians/second} \qquad 4.3.2.$$

and

$$W_s = 2 W_c \qquad 4.3.3.$$

For a Chebyshev filter with ripple coefficient ϵ , the ripple is given by

$$Ripple = \frac{Peak\ magnitude}{Valley\ magnitude} \qquad 4.3.4.$$

or

$$= 20 \log_{10}(1 + \epsilon^2)^{\frac{1}{2}} \qquad dB \qquad 4.3.5.$$

Thus, 1 dB ripple corresponds to $\epsilon = \pm 0.509$

To find the required order, consider the Chebyshev filter transfer function

$$|H(S)|^2 = \frac{1}{1 + \epsilon^2 V_n^2\left(\frac{\omega}{\omega_c}\right)} \qquad \qquad 4.3.6.$$

For high frequencies, $\epsilon^2 V_n^2\left(\frac{\omega}{\omega_c}\right) \gg 1$, and then the gain in the stop band can be written as

$$H(S) \simeq \frac{1}{\epsilon V_n\left(\frac{\omega}{\omega_c}\right)} \qquad \qquad 4.3.7.$$

For large value of X , the Chebyshev polynomial

$$V_n(X) \simeq 2^{n-1} X^n \qquad \qquad 4.3.8.$$

Then
$$H(S) \simeq \frac{1}{\epsilon 2^{n-1}\left(\frac{\omega}{\omega_c}\right)^n} \qquad \qquad 4.3.9.$$

The gain in the stopband is approximately

$$A_V = -20 \log_{10}\left[\epsilon 2^{n-1}\left(\frac{\omega}{\omega_c}\right)^n\right] \quad dB \qquad \qquad 4.3.10.$$

Therefore, $\quad -12 = -20 \log_{10}\left[0.509 \times 2^{n-1}\left(\frac{\omega_s}{\omega_c}\right)^n\right]$

$$n = 2$$

So a second order filter will satisfy the specification and the corresponding transfer function is

$$H(S) = \frac{1}{[S-(a+jb)][S-(a-jb)]} \qquad \qquad 4.3.11.$$

where
$$a = 0.5488672$$
$$b = 0.8951286 \qquad \qquad 4.3.12.$$

Using the bilinear Z-transformation, and referring to procedure in Appendix A.

$$H(Z) = C_0 + \frac{\alpha_0 + \alpha_1 Z^{-1}}{1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}} \qquad 4.3.13.$$

where

$$C_0 = \frac{A}{\beta_2}$$

$$A = \frac{\omega_c^2 T^2}{4D}$$

$$D = \left[1 - \frac{a'T}{2}\right]^2 + \left[\frac{b'T}{2}\right]^2$$

$$\alpha_0 = A\left[1 - \frac{1}{\beta_2}\right]$$

$$\alpha_1 = A\left[2 - \frac{\beta_1}{\beta_2}\right]$$

$$\beta_1 = \frac{-2\left[1 - \left(\frac{a'T}{2}\right)^2 - \left(\frac{b'T}{2}\right)^2\right]}{D} \qquad 4.3.14.$$

$$\beta_2 = \frac{\left[1 + \frac{a'T}{2}\right]^2 + \left[\frac{b'T}{2}\right]^2}{D}$$

$$a' = a.\omega_c$$

$$b' = b.\omega_c$$

Thus, from the equation (4.3.2.), (4.3.12.) and (4.3.14), the coefficients are

$$C_0 = 0.99520$$
$$\alpha_0 = -0.68274$$
$$\alpha_1 = 0.56123 \qquad 4.3.15.$$
$$\beta_1 = 0.06399$$
$$\beta_2 = 0.31396$$

and the desired lowpass filter transfer function is

$$H(Z) = 0.99520 \diamond \left[\frac{-0.68274 + 0.56123\,Z^{-1}}{1 + 0.06399\,Z^{-1} + 0.31396\,Z^{-2}}\right]$$

4.4. Design of a digital highpass filter.

A digital highpass filter is to be designed with cut off
frequency at 100 Hz , which is the lower cut off frequency of the
desired bandpass filter, and having the same specification as given
in section 4.1.

From the frequency transformation, a digital highpass filter
transfer function can be obtained by changing Z^{-1} to $-Z^{-1}$ in a lowpass
filter transfer function, and the resulting cut off frequency is given
by

$$f_{cH} = \frac{f_s}{2} - f_{cL}$$

According to the specification, the original digital lowpass
filter must have the cut off frequency at

$$f_{cL} = \frac{40}{2}\,\text{KHz} - 100\,\text{Hz} = 19.9\,\text{KHz}$$

and the corresponding prewarping critical frequency

$$\omega_c = 10124405 \qquad\qquad 4.3.16.$$

Using the equations (4.3.16. -), (4.3.13.) and (4.3.13.),
the coefficients can be written as

$$C_0 = 0.91416$$

$$\alpha_0 = -0.01427$$

$$\alpha_1 = -0.01406 \qquad\qquad 4.3.17.$$

$$\beta_1 = 1.98416$$

$$\beta_2 = 0.98438$$

and the original digital lowpass filter transfer function is

$$H(Z) = 0.91416 + \left[\frac{-0.01427 - 0.01406\,Z^{-1}}{1 + 1.98416\,Z^{-1} + 0.98438\,Z^{-2}} \right]$$

Therefore, the desired digital highpass filter transfer function is

$$H(Z) = 0.91416 + \left[\frac{-0.01427 + 0.01406\,Z^{-1}}{1 - 1.98416\,Z^{-1} + 0.98438\,Z^{-2}} \right]$$

The frequency response of these filters can be depicted in Fig. 4.3. and Fig. 4.4.
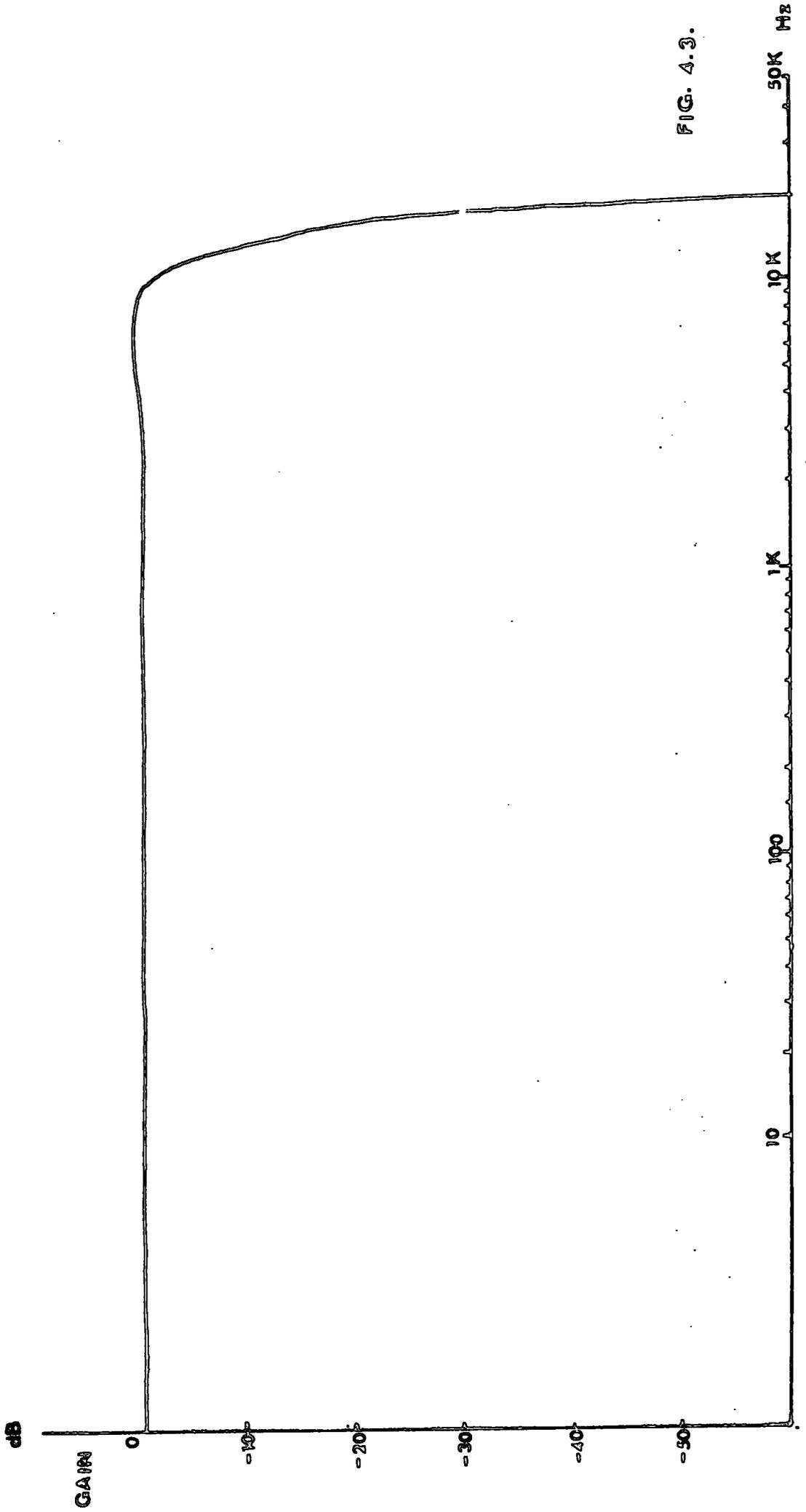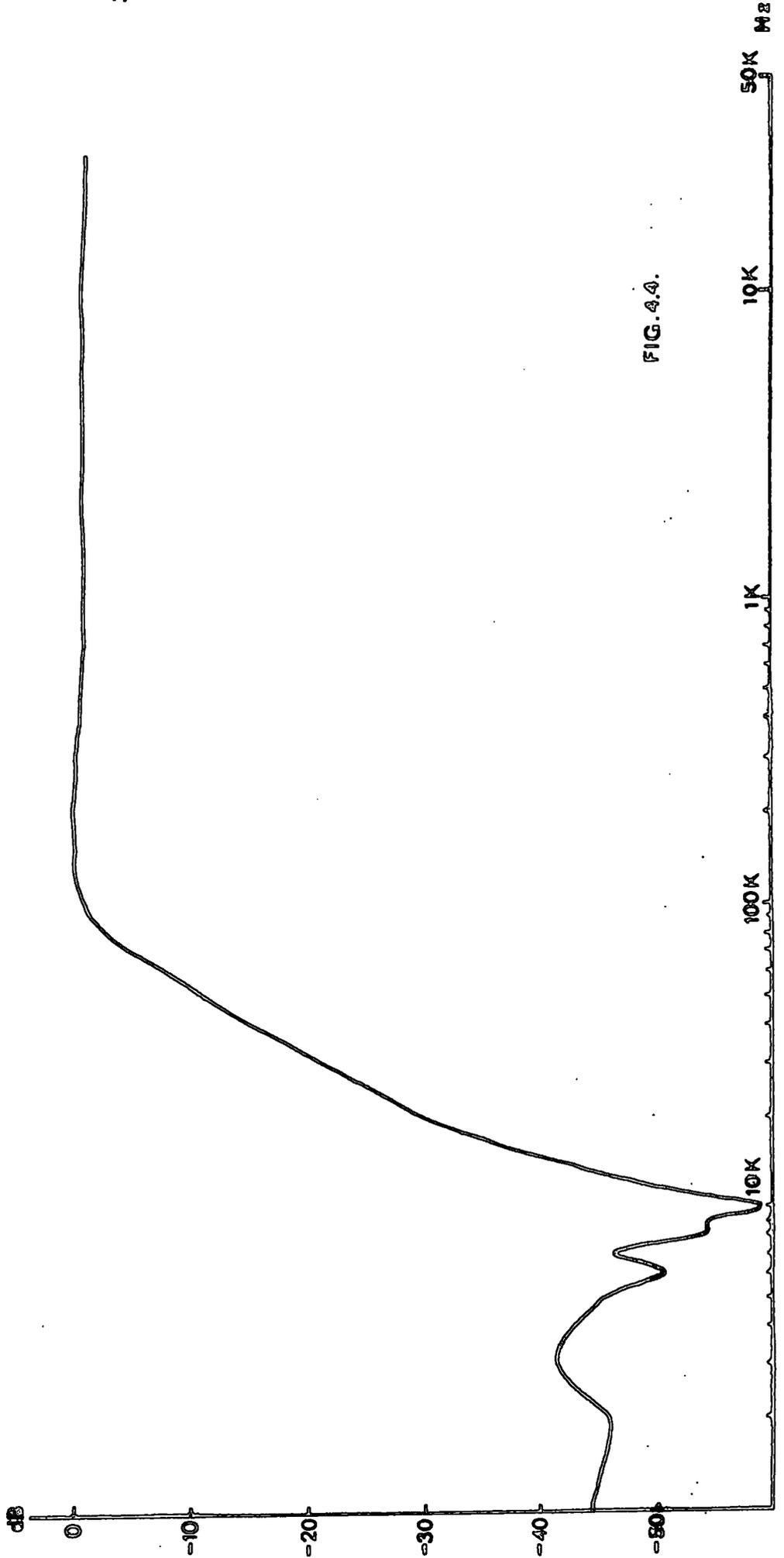
FIG. 4.3.

FIG. 4.4.

4.5. <u>Table of the coefficients at different cut off frequencies.</u>

For the Chebyshev digital lowpass filter of order 2, 1dB ripple and the sampling rate is 40 KHz

f_{cL} Hz	ω_c	C_o	α_o	α_1	P_1	P_2
20	0.12566×10^3	0.2471×10^{-5}	-0.85053×10^{-8}	0.98604×10^{-5}	-1.99654	0.99655
100	0.62834×10^3	0.62215×10^{-4}	-0.10634×10^{-5}	0.24565×10^{-3}	-1.98263	0.98290
10K	0.79990×10^5	0.99520	-0.68274	0.56123	0.06399	0.31396
15K	0.19311×10^6	1.22044	-0.64202	-0.15745	1.07689	0.47303
19K	0.10159×10^7	0.97821	-0.14150	-0.12120	1.83458	0.85534
19.8K	0.50773×10^7	0.92126	-0.02845	-0.02763	1.96822	0.96911
19.90K	0.10124×10^8	0.91416	-0.01427	-0.01406	1.98416	0.98438
19.98K	0.49420×10^8	0.09084	-0.02923	-0.02914	1.99677	0.99678

4.6. <u>Realization of a second order digital filter.</u>

From the designed example, the second order filter has a transfer

function

$$H(Z) = C_0 + \frac{\alpha_0 + \alpha_1 Z^{-1}}{1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}} = \frac{Y(Z)}{X(Z)} \qquad 4.6.1.$$

By introducing

$$W(Z) = \frac{X(Z)}{1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}}$$

$$= X(Z) - \beta_1 Z^{-1} W(Z) - \beta_2 Z^{-2} W(Z)$$

$$4.6.2.$$

then we have

$$Y(Z) = C_0 X(Z) + (\alpha_0 + \alpha_1 Z^{-1}) W(Z) \qquad 4.6.3.$$

and this can be realized with block diagrams as shown in Fig. 4.5.
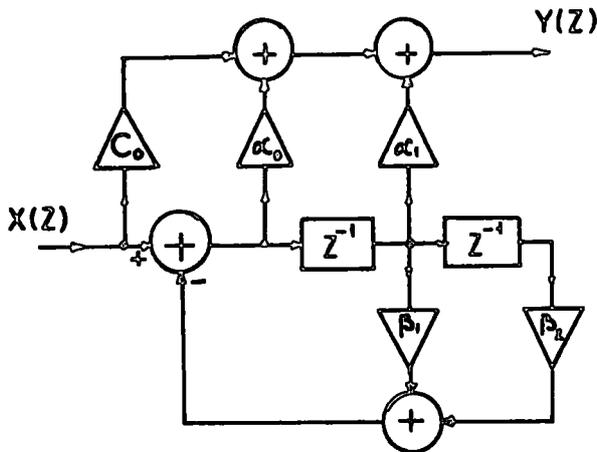

Fig. 4.5.

This circuit needs five multiplications, five coefficients and

four adders. However, for a single second order filter, one

multiplication can be reduced by algebraical modification. Equation

4.6.1. can be rewritten as

$$H(Z) = \frac{[(C_0 + \alpha_0) + (C_0\beta_1 + \alpha_1)Z^{-1} + C_0\beta_2 Z^{-2}]}{1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}} \qquad 4.6.4.$$

Therefore, $\quad Y(Z) = (C_0 + \alpha_0)W(Z) + (C_0\beta_1 + \alpha_1)Z^{-1}W(Z) + C_0\beta_2 Z^{-2}W(Z)$

or $\quad \dfrac{1}{C_0} Y(Z) = \alpha_0' W(Z) + \alpha_1' Z^{-1}W(Z) + \beta_2 Z^{-2}W(Z) \qquad 4.6.5.$

where $\quad \alpha_0' = \dfrac{C_0 + \alpha_0}{C_0}$

and $\quad \alpha_1' = \dfrac{C_0\beta_1 + \alpha_1}{C_0}$

The realization of the equation (4.6.5.) can be depicted in Fig. 4.6.

In practical use, the designed values of α_1' and β_1 may be more than 1 , but the number representation used is restricted to the range $-1 \leqslant X < 1$, see chapter 3.5. However, this problem can be solved by scaling the coefficients α_1' and β_1 by $\frac{1}{2}$, and then doubling the results of the multiplier. These multiplications by 2 can be done easily without using extra components, see chapter 5.2.
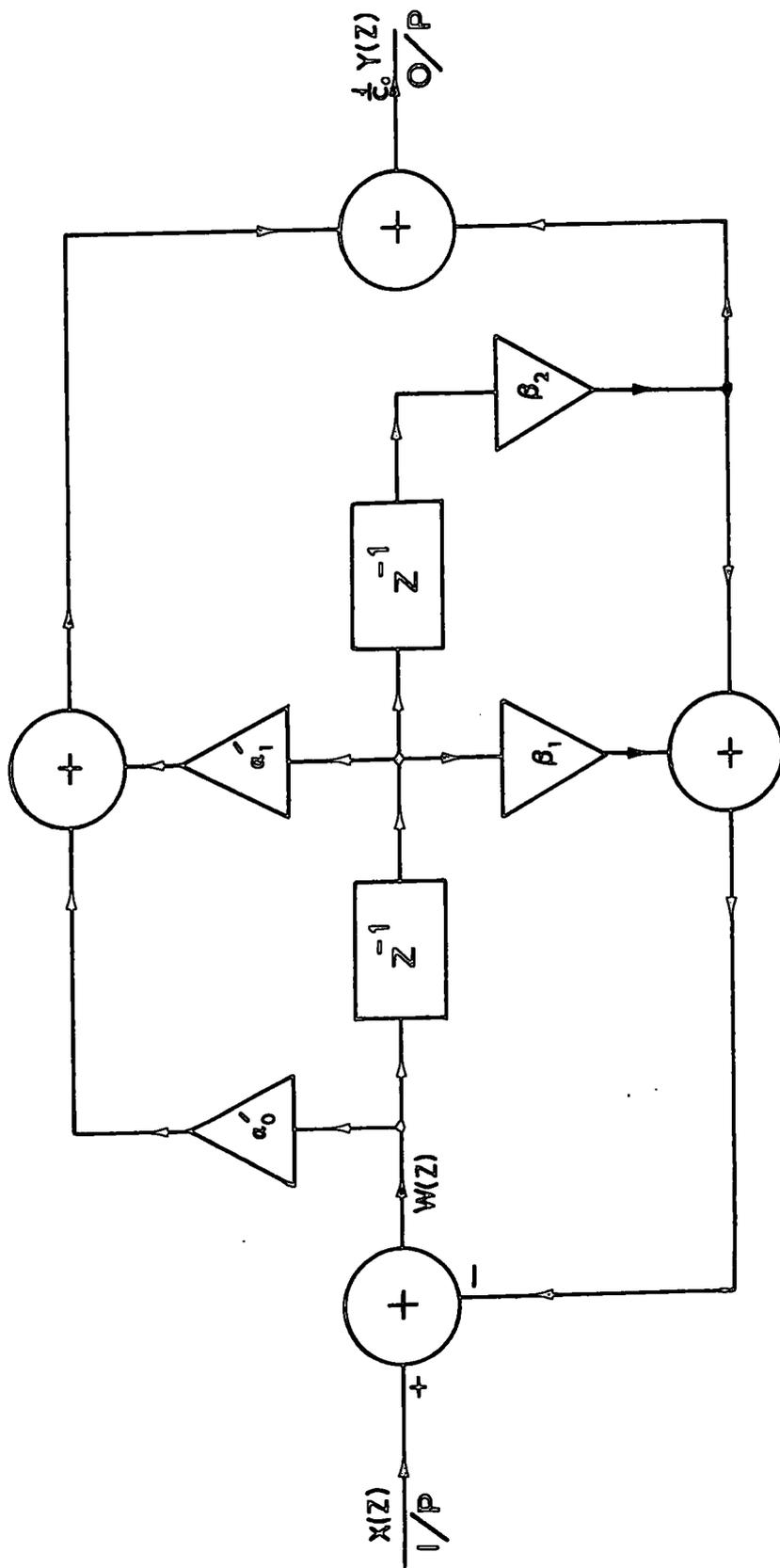
FIG. 4.6.   THE BLOCK DIAGRAM OF A SECOND ORDER FILTER.

4.7. Noise consideration.

In the following section, the noises of our system due to the quantization effect will be considered. Here, we assume that the degeneration of the filter performance due to the coefficient rounding effect is tolerable. Therefore, the only noises to be considered are the input quantization noise and the round off noise.

Input quantization noise.

Let $E(Z)$ be the input quantization noise due to the A-D converter injected to the filter as shown in Fig. 4.7. , and having the steady-state mean squared value of $\dfrac{\Delta^2}{12}$
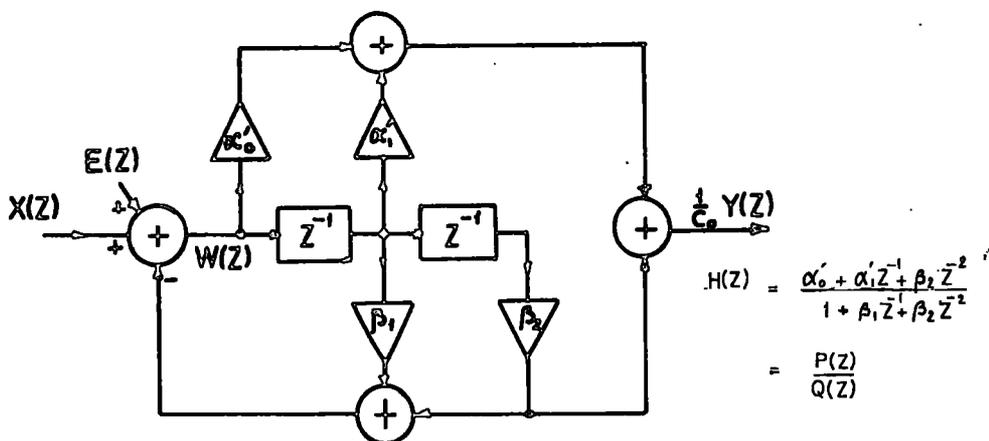

$$H(Z) = \frac{\alpha_0' + \alpha_1' Z^{-1} + \beta_2' Z^{-2}}{1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}}$$

$$= \frac{P(Z)}{Q(Z)}$$

Fig. 4.7. THE MODEL SHOWING THE INPUT QUANTIZATION NOISE.

It is seen that,

$$W(Z) = X(Z) - \beta_1 Z^{-1} W(Z) - \beta_2 Z^{-2} W(Z) + E(Z)$$

$$= \frac{X(Z) + E(Z)}{1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}}$$

$$= \frac{X(Z) + E(Z)}{Q(Z)}$$

and

$$\frac{1}{C_0} Y(Z) = \alpha_0' W(Z) + \alpha_1' Z^{-1} W(Z) + \beta_2 Z^{-2} W(Z)$$

$$= P(Z) W(Z)$$

$$= P(Z) \left[\frac{X(Z) + E(Z)}{Q(Z)} \right]$$

$$= \frac{P(Z)}{Q(Z)} X(Z) + \frac{E(Z) P(Z)}{Q(Z)}$$

$$= H(Z) X(Z) + E(Z) H(Z)$$

Therefore, the noise term is $E(Z) H(Z)$, and the mean squared value of this noise at the output of the filter will be (3)

$$MSONI = \frac{1}{2\pi j} \oint H(Z) H(Z^{-1}) E(Z) E(Z^{-1}) \frac{dZ}{Z}$$

It has been shown (3) that $E(Z) E(Z^{-1})$ is equal to $\frac{\Delta^2}{12}$

Thus,

$$MSONI = \frac{1}{2\pi j} \oint H(Z) H(Z^{-1}) \frac{\Delta^2}{12} \frac{dZ}{Z}$$

or,

$$= \frac{\Delta^2}{12} \left[\frac{1}{2\pi j} \oint H(Z) H(Z^{-1}) \frac{dZ}{Z} \right]$$

For the designed example, the transfer function in the equation (4.3.17.) we have, (28)

$$\frac{1}{2\pi j} \oint H(Z) H(Z^{-1}) \frac{dZ}{Z} = 1118.5463$$

and then the root mean squared value of this noise is plotted versus the number of bits as shown in Fig. 4.8.

It should be noted that the value of the noise in Fig. 4.8. is a normalized value to a unit output signal level. However, B. Gold and Rader (3) have given a useful relation, for canonic form, as
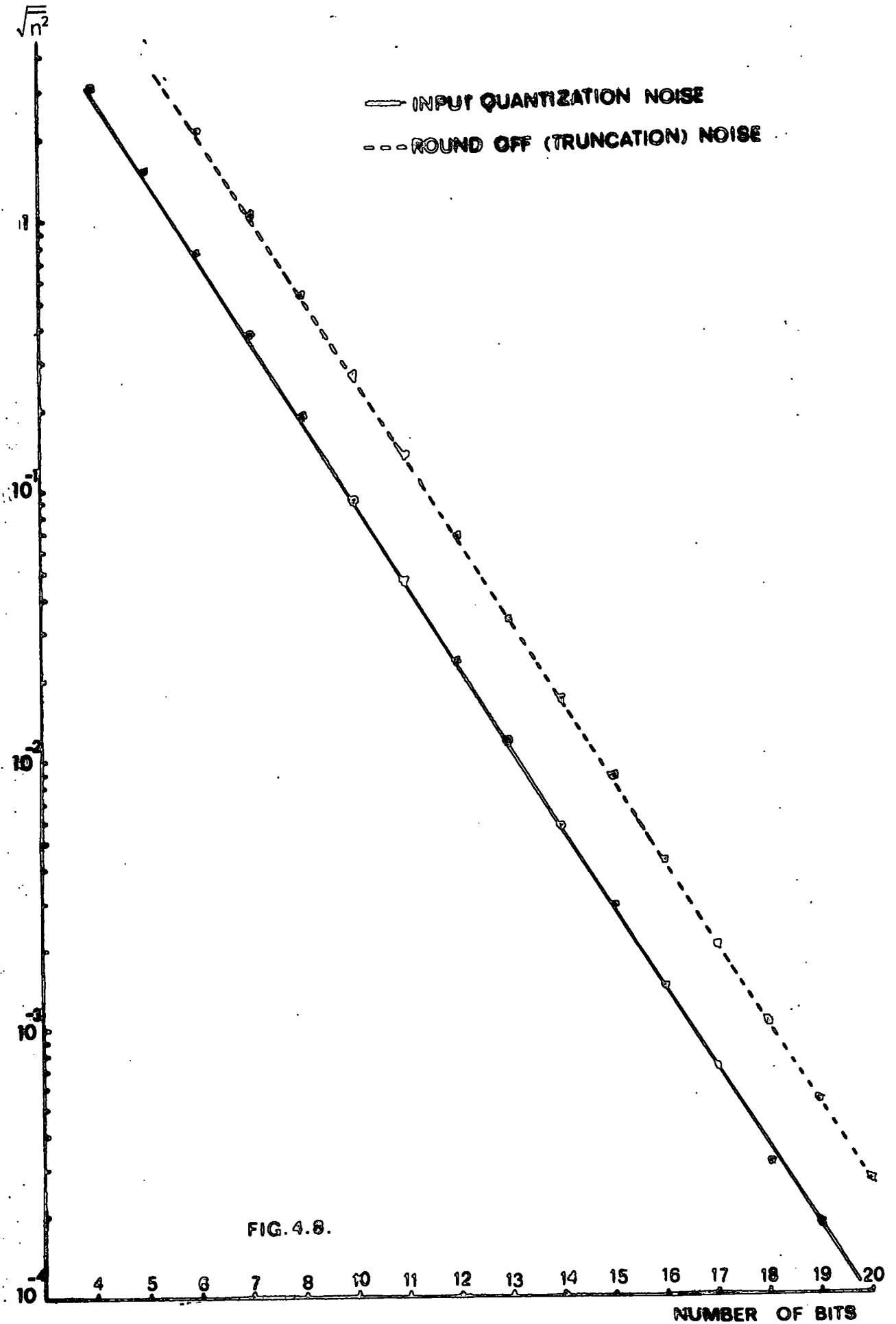
FIG. 4.8.

A GRAPH SHOWING THE QUANTIZATION EFFECTS.

$$B_{-1} = -0.166F - 1.79 + 1.66 \log_{10}\left[\frac{1}{2\pi j} \oint H(Z)H(Z^{-1})\frac{dZ}{Z}\right]$$

where, B_{-1} is the number of bits which must be retained below the
unit signal level ; an additional 3 to 5 bits should be retained
above the unit signal level to protect against overflow, and

F (dB) is the mean-squared output noise below a unit output
signal level.

Therefore, for the designed example, this relation can be
plotted as shown in Fig. 4.9.

Round off noise.

Let $V(Z)$ be the round off noise generated at each multiplication.
For unrounded operations, Knowles (26) has given that the steady-
state mean squared value of this noise is $\frac{\Delta^2}{3}$ per multiplication.
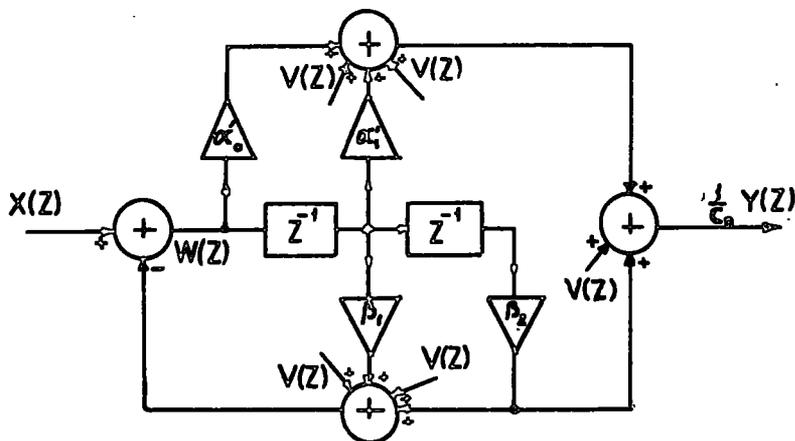This noise is injected into the filter as the model shown in Fig.
4.10.

Fig. 4.10. THE MODEL SHOWING THE ROUND OFF NOISE.

It is seen that,

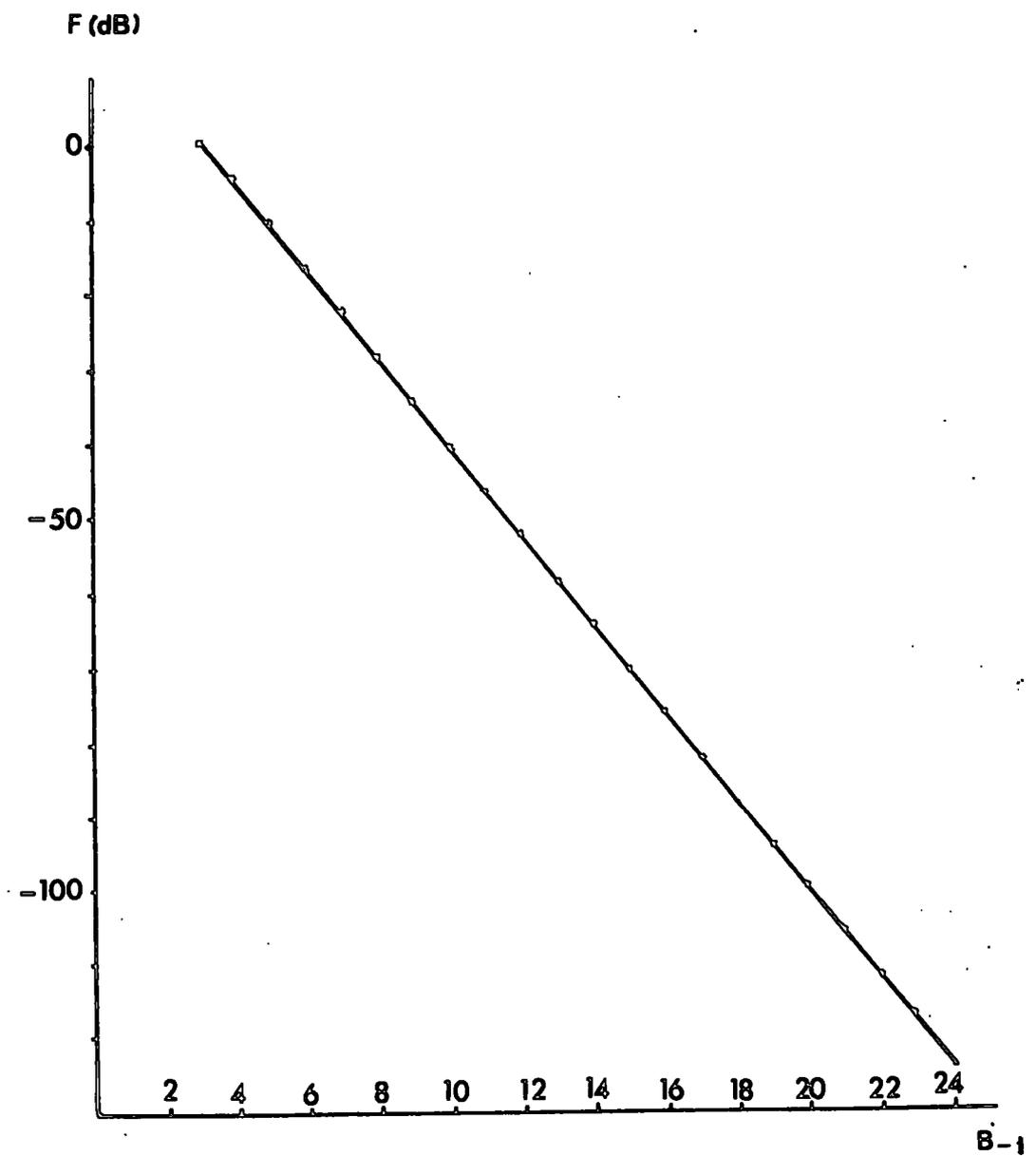$$W(Z) = X(Z) - (\beta_1 Z^{-1} W(Z) + V(Z)) - (\beta_2 Z^{-2} W(Z) + V(Z))$$

FIG 4.9 A GRAPH SHOWING THE INPUT QUANTIZATION NOISE IN dB.

$$W(Z) = X(Z) - \beta_1 Z^{-1} W(Z) - \beta_2 Z^{-2} W(Z) - 2V(Z)$$

$$= \frac{X(Z) - 2V(Z)}{1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}}$$

$$= \frac{X(Z) - 2V(Z)}{Q(Z)}$$

and, $\frac{1}{C_0} Y(Z) = \alpha_0' W(Z) + \alpha_1' Z^{-1} W(Z) + \beta_2 Z^{-2} W(Z) + V(Z) + V(Z) + V(Z)$

$$= P(Z)W(Z) + 3V(Z)$$

$$= P(Z)\left[\frac{X(Z) - 2V(Z)}{Q(Z)}\right] + 3V(Z)$$

$$= X(Z)H(Z) - \left[2V(Z)H(Z) - 3V(Z)\right]$$

Therefore, the mean squared output found off noise is given by

$$MSORN = \frac{1}{2\pi j} \oint 2V(Z)V(Z^{-1}) H(Z)H(Z^{-1}) \frac{dZ}{Z} - \frac{1}{2\pi j} \oint 3V(Z)V(Z^{-1}) \frac{dZ}{Z}$$

Because $V(Z)V(Z^{-1}) = \frac{\Delta^2}{3}$, and

$$\oint \frac{dZ}{Z} = 2\pi j$$

Thus,

$$MSORN = \frac{2\Delta^2}{3}\left[\frac{1}{2\pi j} \oint H(Z)H(Z^{-1}) \frac{dZ}{Z}\right] - \Delta^2$$

$$= \frac{2\Delta^2}{3}\left[\frac{1}{2\pi j} \oint H(Z)H(Z^{-1}) \frac{dZ}{Z} - \frac{3}{2}\right]$$

But $\frac{1}{2\pi j} \oint H(Z)H(Z^{-1}) \frac{dZ}{Z} \gg \frac{3}{2}$

Therefore, $MSORN = \frac{2\Delta^2}{3}\left[\frac{1}{2\pi j} \oint H(Z)H(Z^{-1}) \frac{dZ}{Z}\right]$

For the designed example, equation (4.1.17.), this noise can be plotted against the number of bits as shown, by the dotted line, in Fig. 4.8.

Chapter 5.

Implementation.

5.1. Input and output units.

In section 3.5.1. , it has been pointed out that the filter will

operate on a binary number in the signed 2's complement representation.

But a binary number in a digital system is usually in the signed

magnitude representation. Therefore, 2's complementers are required

as input and output devices to encode and decode the number

respectively.

5.1.1. 2's complementers.

A 2's complementer can be implemented with a sequential circuit

following the algorithm as below :

Starting from the least significant bit of a number,

1. for a positive number, transmit unchanged all bits,

2. for a negative number, transmit unchanged all bits up to and

including the first "1" and then invert all subsequent number bits,

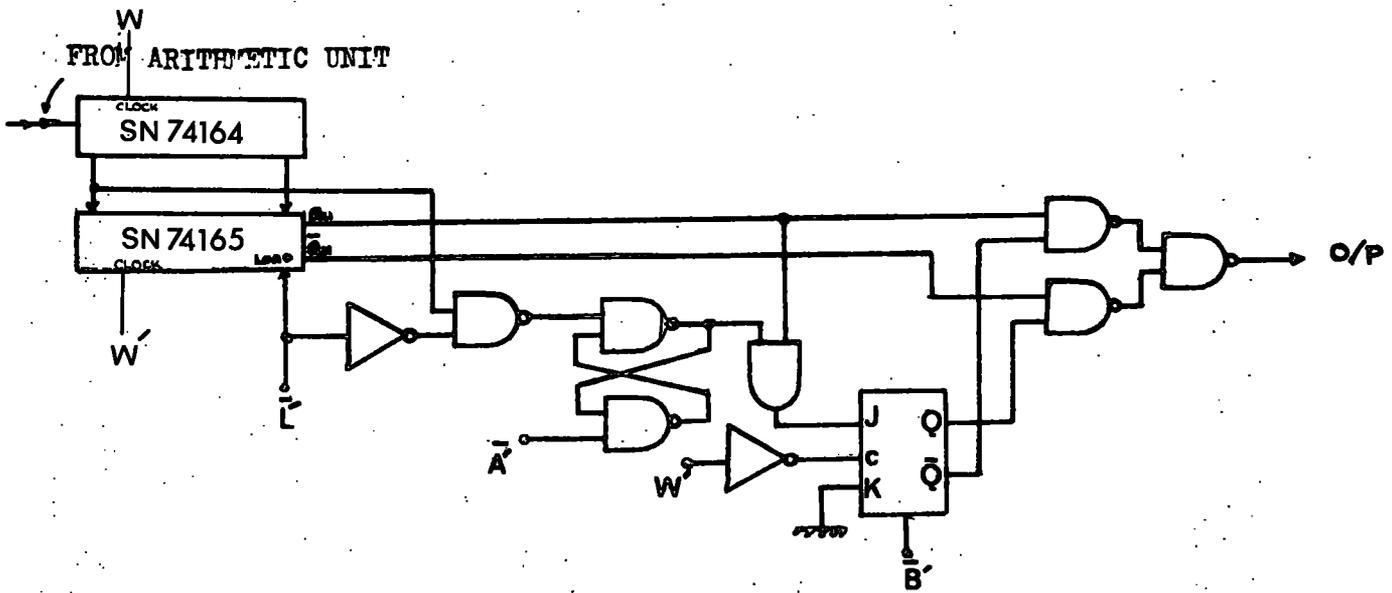3. transmit the sign bit unchanged.

The circuits and time table of the input 2's complementer and

the output 2's complementer can be depicted in Fig. 5.1.

The J-K flip-flop 6 and the cross-coupled gates 3 and 4 are

initially reset closing gates 5 and 8. If a positive number is

loaded, there is no change in these gates, and the data passes
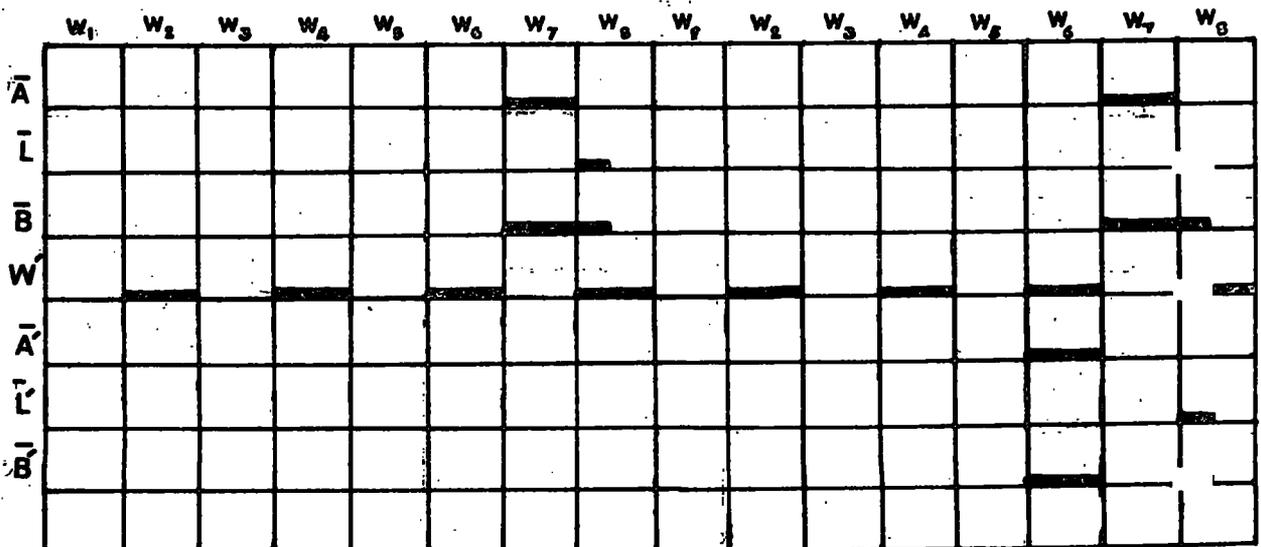
FIG. 5.1.
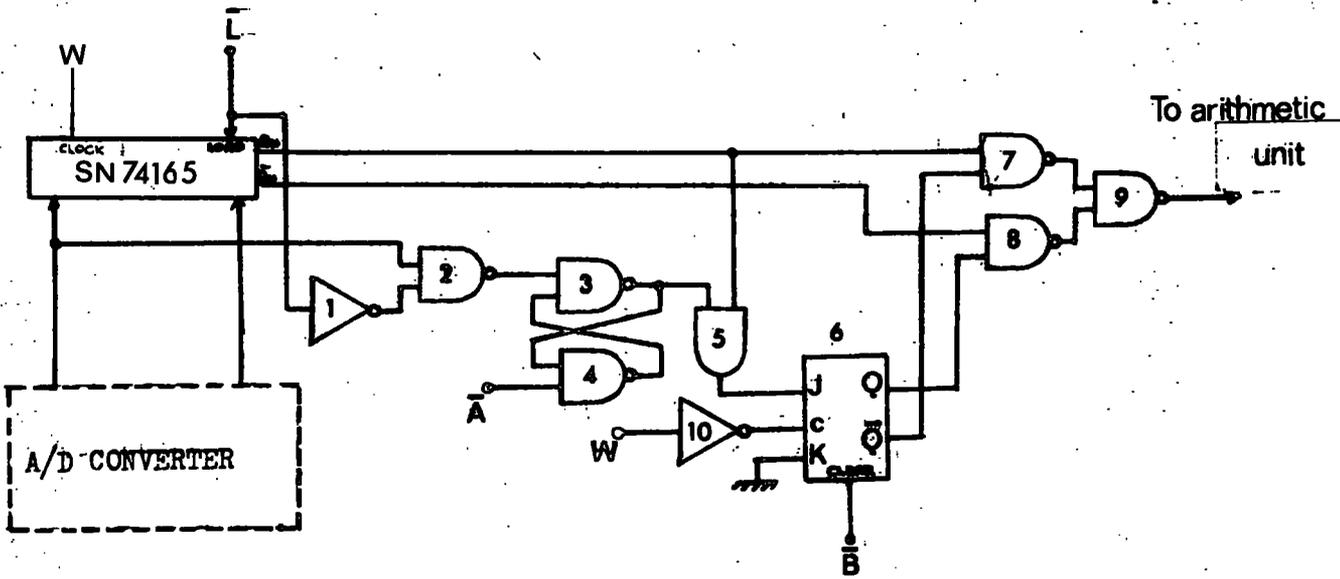
INPUT 2'S COMPLEMENTER.

OUTPUT 2'S COMPLEMENTER.

TIME DIAGRAM FOR THE 2'S COMPLEMENTERS.

unchanged through gates 7 and 9 to the output. When a negative number is loaded, the least significant bit passes unchanged to the output. In the meantime, the sign bit has opened gate 5. If the least significant bit is "0" , the flip-flop 6 remains at the low level leading to the succeeding bit passing unchanged to the output. But, if the least significant bit is "1" , the flip-flop 6 changes to the high level closing gate 7 and opening gate 8. Then all the succeeding number bits are complemented and a number in signed representation is obtained at the output. (or a number in signed magnitude representation is achieved if the complementer operates on a number in signed 2's complement representation.)

To keep a sign bit unchanged, the J-K flip-flop 6 must be reset at the last number bit, w_7 . The J-K flip-flop 6 operates at the negative-going edge of clock pulses. Therefore, an inverter, gate 10, is added.

It should be observed here that the control signals and clock frequencies for the input and output 2's complementers differ only when multiplexing is performed.

5.2. Arithmetic unit.

The implementation of the arithmetic unit for a digital filter consists of the interconnection of delays, adders, subtractors and multipliers in the way described in chapter 4. Parallel arithmetic may be used for the fast operations but the price is rather high. Considerable economy can be achieved by using serial arithmetic with the appropriate techniques. But the operations in the

arithmetic unit have to be done within a given sampling interval, and the use of serial arithmetic with multiplexing results in a demand for high speed digital components.

In the following sections, the individual components in the arithmetic unit and their circuits will be discussed.

5.2.1. Unit sample delays.

The use of serial arithmetic allows the easy implementation of sample delays as serial-in serial-out shift registers. For example, a **SN 7491** can be used to implement a unit sample delay Z^{-1}, as shown in Fig. 5.2. (a). When multiplexing is included, a unit sample delay can be implemented as a cascade of two serial-in serial-out shift registers and a multiplexer, as shown in Fig. 5.2. (b). Alternatively, a parallel form could be used.
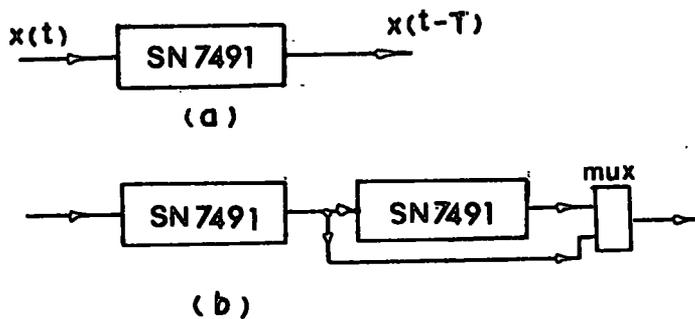

Fig. 5.2. Unit delay elements.

5.2.2. Serial adders and subtractors.

A serial adder for a number in signed 2's complement representation is identical to that for signed magnitude representation. It

consists of a full binary adder and a delay flip-flop, which is initially reset, to transfer the carry output back to the carry input. A circuit of a serial adder is depicted in Fig. 5.3. (a).
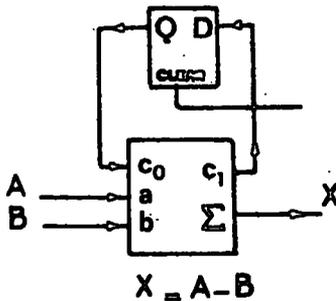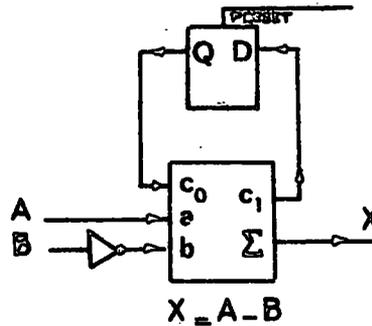

Fig. 5.3(a)A SERIAL ADDER.   Fig.5.3(b)A SERIAL SUBTRACTOR.

A serial subtractor for a number in signed 2's complement representation can be implemented basically as a serial adder. But the subtrahend must be inverted and the carry delay flip-flop is initially preset to the high level, according to the process of subtraction of a number in this representation. A circuit of a subtractor is depicted in Fig. 5.3. (b).

It should be noted that, using serial arithmetic, the processing rate is normally limited only by the speed of the full adders and not the carry delay flip-flop. Therefore, the high speed full adders, **SN 74H183** with a propagation delay time of **18 ns** , is used for both adders and subtractors, and the D-type flip-flop, **SN 7474** , can be utilised as the carry delay element.

5.2.3. Serial multipliers.

The Booth's method of multiplication for number in signed 2's complement representation is chosen because it gives the correct product, whatever the sign of the numbers, without any correction.

This method was suggested by A. D. Booth and K. H. V. Booth (32) (31) and has the scheme of operation as follows :

Let a_i and b_i be the ith bits of the number A and B , see section 3.5.1. , respectively. Then the algorithm for the product $A \times B$ is :

Starting from the least significant end, for all number bits, $i \not= 0$.

1. If $b_i b_{i+1}$ are 00 or 11, shift partial product to the right 1 bit.

2. If $b_i b_{i+1}$ are 10, subtract A from the partial product, and then shift the latter to the right 1 bit.

3. If $b_i b_{i+1}$ are 01, add A into the partial product, and then shift the latter to the right 1 bit.

Repeat until i is the sign bit, $(i = 0)$. Here, the process is exactly as above except that the shift is omitted.

Note that for the least significant multiplier bit b_i , we must take b_{i+1} to be 0.

Booth has given the recursion for the above process as :

$$a_{i-j} = \tfrac{1}{2} a_{i-j+1} + (b_{i-j+1} - b_{i-j}) \cdot m$$

$$j = 1, 2, \ldots \ldots (i-1)$$

where a_{i-j+1} is the contents of the accumulator after the jth stage of the process, and m represents the multiplicand.

From the recursion formula, it is seen that the factor $(b_{k+1} - b_k)$ has the values :

0 if $b_k b_{k+1}$ are 00 aor 11

1 if $b_k b_{k+1}$ are 01

-1 if $b_k b_{k+1}$ are 10

as required by the algorithm described previously.

Further detail of this method, including a proof of its validity and examples of its use can be found in references (31) (32).

The circuit of a serial multiplier using the Booth's method and the time diagram are shown in Fig. 5.4. It consists of two registers, one full adder, three D-type flip-flops, two AND-OR-INVERT gates, two NAND gates, two AND gates and one multiplexer.

A coefficient is treated as a multiplicand and a data word as a multiplier. The two least significant bits, b_i and b_{i+1}, are shifted serially into the bistables M and E. The outputs of M and E control the operations through the AND-OR-INVERT gates and AND gates. If an addition is required, the contents of the multiplicand register are fed to the full adder with the carry delay flip-flop initially reset. If a subtraction is to be performed, the inverted outputs of the multiplicand register are used, the carry delay flip-flop is initially preset to the high level by gates 3 and 4. The contents of the circulating multiplicand register is shifted to the right by 8 places, and the accumulator is shifted by 9 places, for every bit time from w_1 to w_7, but at bit time w_0, both shift right 8 places.

A multiplexer is used at the most significant bit of the accumulator for the purpose of shifting the partial product to the right 1 bit. From the example in chapter 3.5.2. , it is seen that, when shifting to the right, the new most significant bit is the original sign bit, but the sign bit is retained unchanged.

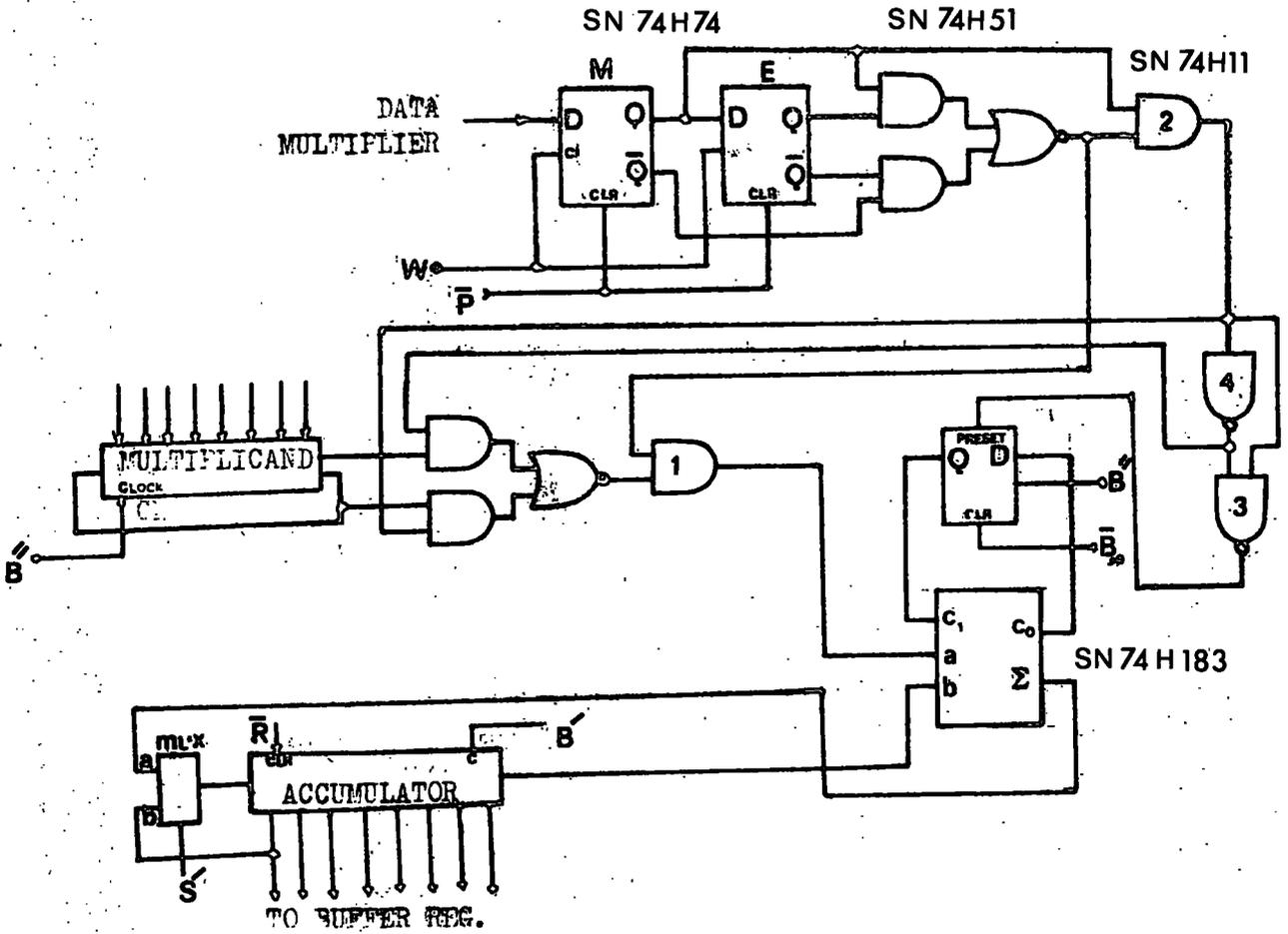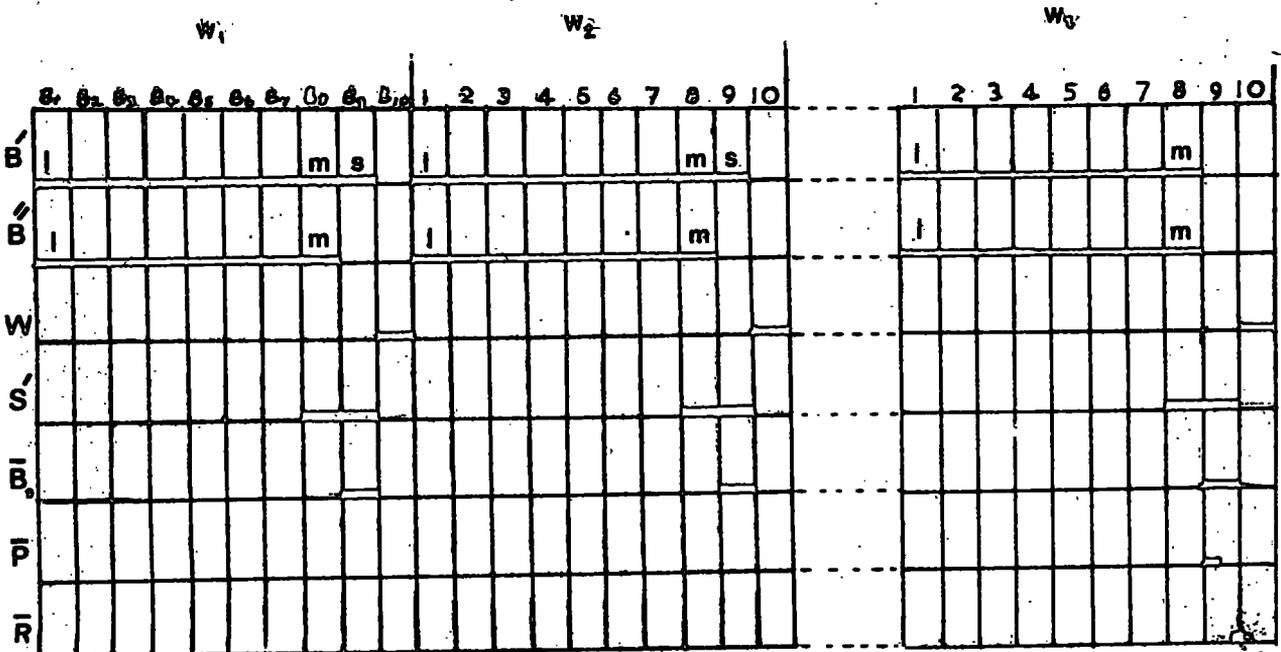The use of the high speed components, SN 74H183 , SN 74H51

FIG. 5.4.   A SERIAL MULTIPLIER USING THE BOOTH'S METHOD.

TIME DIAGRAM FOR THE MULTIPLIERS.

and SN74H11 , makes it possible to perform the multiplication of

two 8 bit words within 1 μs.

It should be observed that, in this multiplication, truncation

is used to eliminate the lower 8 least significant bit of the

product, retaining an 8 bit word throughout the arithmetic unit.
                          this
In implementation of the arithmetic unit, ~~because~~ the serial
                                        the operation on
multiplication gives the answer after the most significant bit has been

performed. This means the result is delayed by one word, which is

equivalent to a unit sample delay. Therefore, the realization

shown in Fig. 4.6. must be rewritten as in Fig. 5.5.(a). It is

seen, from Fig. 5.5.(a) , that the output will be delayed by one

word.

A circuit of the arithmetic unit can be depicted in Fig. 5.5.

(b). Note that two buffer registers, SN74165 and SN7491 , and

a data selector are required in each multiplier to buffer answer

from the multiplication to the next stage. When multiplexing is

performed, the output of this cascade is utilized, otherwise, the

content of SN74165 will be required. A multiplexer and a

demultiplexer are included at the input and output of the arithmetic

unit respectively, see Fig. 4.1. Because α_i' and β_i have been

scaled, the answer needs multiplying by 2 which can be done when

the contents of the accumulators are loaded to the buffer registers.

The first number bit of the accumulator is rejected and the second

number bit is loaded to the first number bit of the buffer register,

and so on, see Fig. 5.5.(b). Obviously, this process is equivalent
            the   by
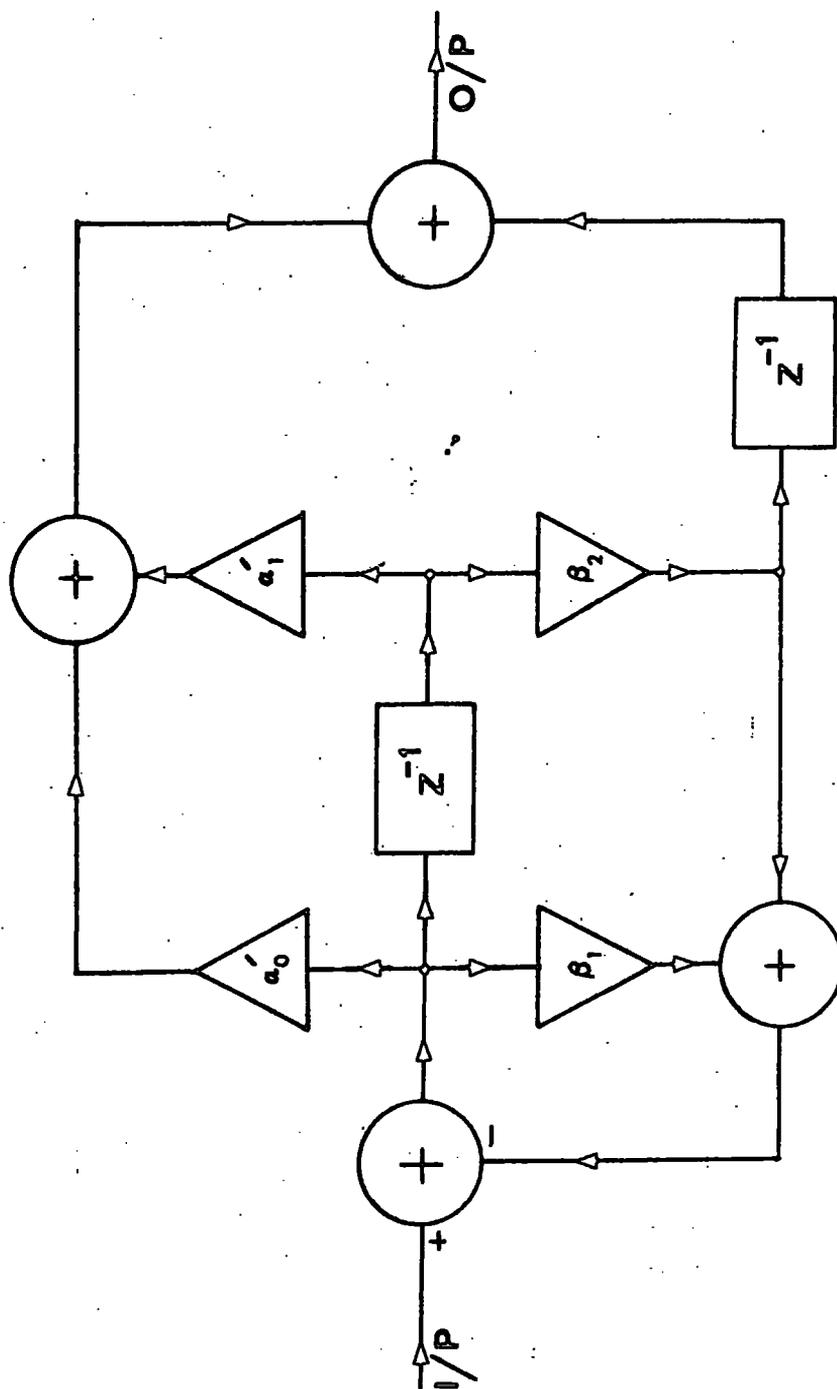to shifting contents in the buffer register to the left 1 bit.

FIG. 5.5 (a)   THE MODIFIED BLOCK DIAGRAM OF FIG. 4.6.

FIG. 5.5(b)   THE ARITHMETIC UNIT.

5.3. Coefficient storage.

Coefficient storage can be implemented by random access memories, (R. A. M.'s) and/or read only memories, (R. O. M's). But serial arithmetic with multiplexing needs either a very high speed memory or a set of R. A. M's with buffer registers. Both solutions are expensive at the moment, but they will become possible in the future. However, at present storage can be economically realized from bistable latches and multiplexers, which will be utilized as scratch pad memories. This form of memory is also easy to extend for a high-order filter.

For a second order digital filter, having eight coefficients, the coefficient storage consists of eight sets of bistable latches, 2-input data selectors, (multiplexers), and NOR gates. Its circuit is shown in Fig. 5.6.

For test purposes an eight bit word coefficient is written into the appropriate address in the storage from the data switches, D_i to D_0 , as soon as the "write" is operated. It is read into the arithmetic unit from the data selector by the selecting signal " S'' ". Thus, the delay time in-changing the coefficients is only the propogation delay time of the data selectors which is typically 18ns.

In practical use, a R. O. M. and/or a calculator chip may be incorporated to determine the values of the coefficients. Calculator chips are now available in a cheap form. Although they are not fast enough to be used in the arithmetic unit they can be used here.

FIG. 5.6. THE COEFFICIENT STORAGE.
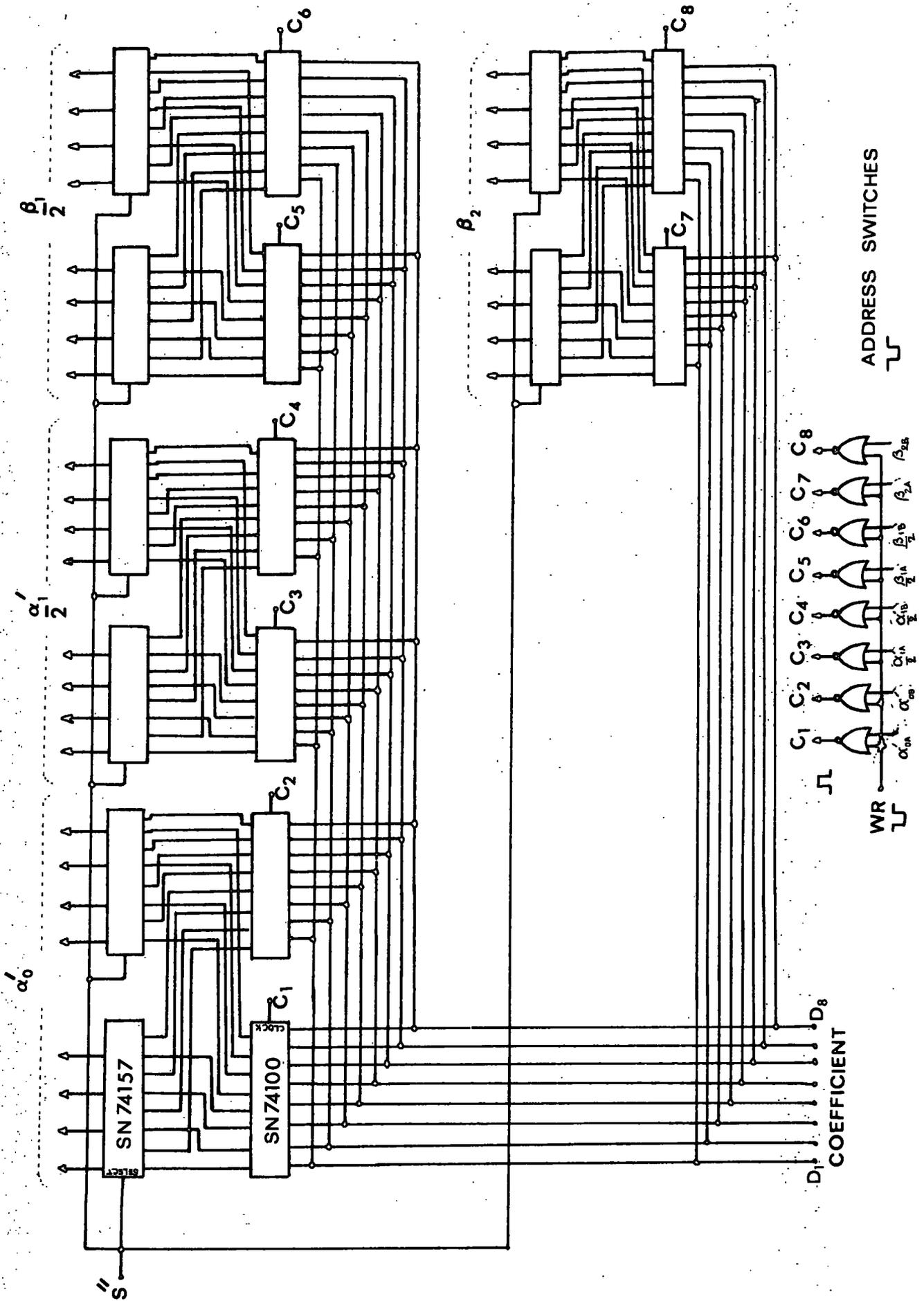
5.1. Control unit.

The control unit of the digital filter supplies all the synchronization and control pulses to the complementers and arithmetic unit. It consists of a fundamental clock, sample clock, binary counter, decade counter, D-type flip-flop, J-K flip-flop and timing level generator. A circuit of the control unit and a part of the timing diagram is shown in Fig. 5.6. , (see also the time diagram of the multiplier and the 2's complementers). The organization and function of each of these elements is self-explanatory. However, their functions can be summarized as follows :

Symbol	Polarity	Function.
CK	+	Fundamental clock.
B'	+	Shift the accumulators.
B''	+	Shift the coefficients.
$\overline{B}_9$	–	Clear the carry-delay flip-flops in the multiplications.
W	+	Shift the unit delay elements ; shift the buffer registers in the multiplication ; shift the buffer registers in the 2's complementers ; clock the multiplier bits ; clock the D-flip-flops in adders and subtractors ; clock the J-K flip-flop used to produce W' .
$\overline{W}$	–	Clock the D-flip-flop in the input 2's complementers.
L	–	Load the buffer register in the input 2's complementers.
$\overline{A}$	–	Clear the cross-coupled gates in the input 2's complementers.

| Symbol	Polarity	Function.
$\overline{B}$	—	Clear the J-K flip-flop in the input 2's complementer.
$W^{\prime}$	+	Shift the second buffer register in the output 2's complementer.
$\overline{W}^{\prime}$	—	Clock the J-K flip-flop in the output 2's complementer.
$\overline{L}^{\prime}$	—	Load the second buffer register in the output 2's complementer.
$\overline{A}^{\prime}$	—	Clear the cross-coupled gates in the output 2's complementer.
$\overline{B}^{\prime}$	—	Clear the J-K flip-flop in the output 2's complementer.
$\overline{K}$	—	Load the coefficients.
$\overline{P}$	—	Load the buffer registers in the multipliers.
$\overline{R}$	—	Clear the accumulators.
S		Multiplexing switch. S = 0 ; for bandpass filter S = 1 ; for lowpass or highpass filter.
$S^{\prime}$		Select the new sign bits of the accumulators.
$S^{\prime\prime}$		Select input data going to the arithmetic unit; select the coefficients.
X , Y		Select output data going out from the arithmetic unit. X = 1 , the ouput data goes to the feed back loop. Y = 1 , the output data goes to the output 2's complementer.
WR	—	Write the coefficients.
M	—	Manual clear.

Fig. 5.7 (a) THE CONTROL UNIT.
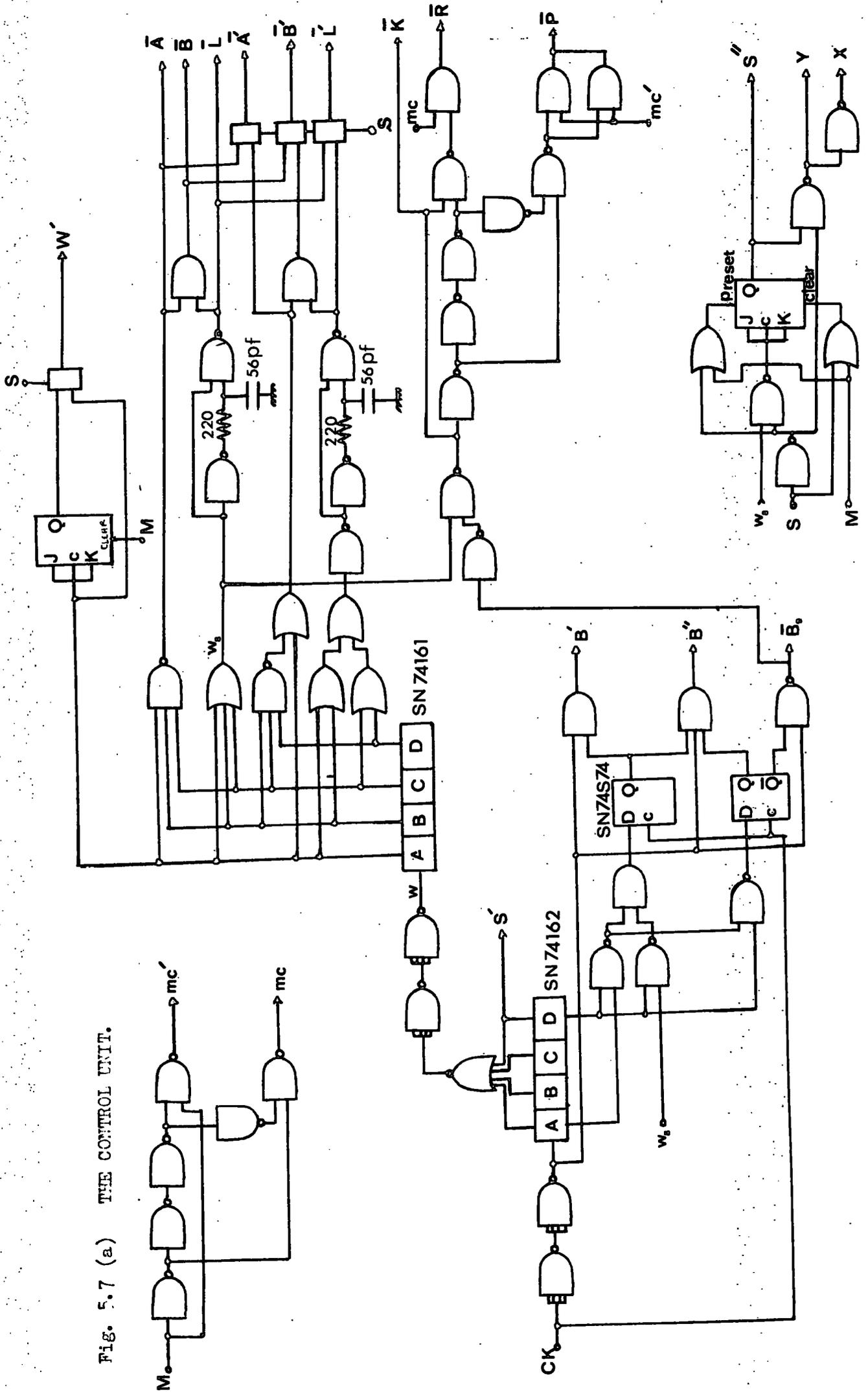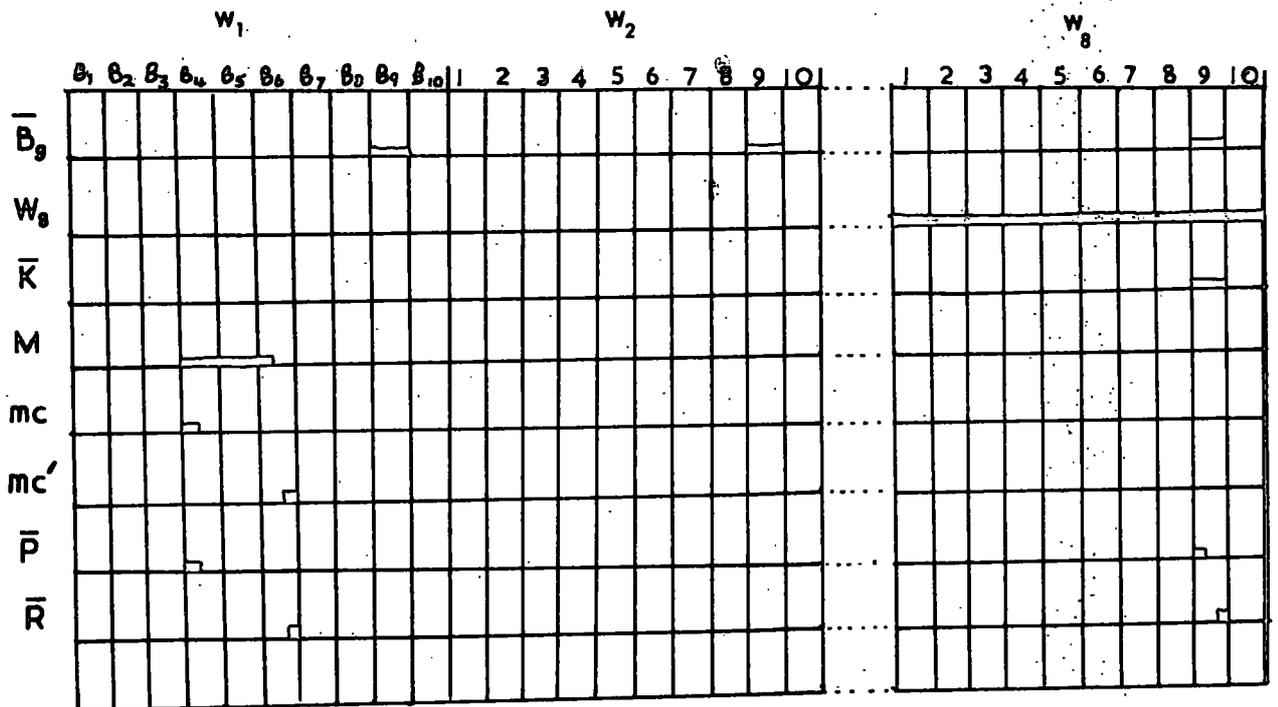
Fig. 5.7. (b)   A PART OF THE TIME DIAGRAM FOR THE CONTROL UNIT.

(SEE ALSO FIG. 5.1. AND FIG. 5.4.)

Chapter 6.

Performance of the Prototype filter.

For testing purpose, an A-D converter and a D-A converter, whose circuits are shown in Appendix B, have been built to allow the filter to be tested with analogue signals. A Brookdeal 471 signal generator was used to produce the input sine wave signal to the filter and the output was detected by a Marconi TF 2600 valve voltmeter. A pulse generator, Advance PG 58, was used as the fundamental clock generator.

6.1. Lowpass filters.

The filter has been set with a range of cut off frequencies, as given in Table 6.1. , and the frequency responses are shown in Fig. 6.1. to Fig. 6.6. The attenuation slopes obtained are approximately -10 dB per octave with 1 dB ripple in the passband instead of -12 dB per octave and 1 dB ripple as in the design, see also Chapter 4. This is probably the result of the quantization effects and instability of the fundamental clock generator used, (approximately ± 5%).

Because of problems with overflow at the intermediate stage w(z) resulting from the 8 bit word length, which will be discussed in section 6.4. , the cut off frequencies below 4 KHz could not be tested.
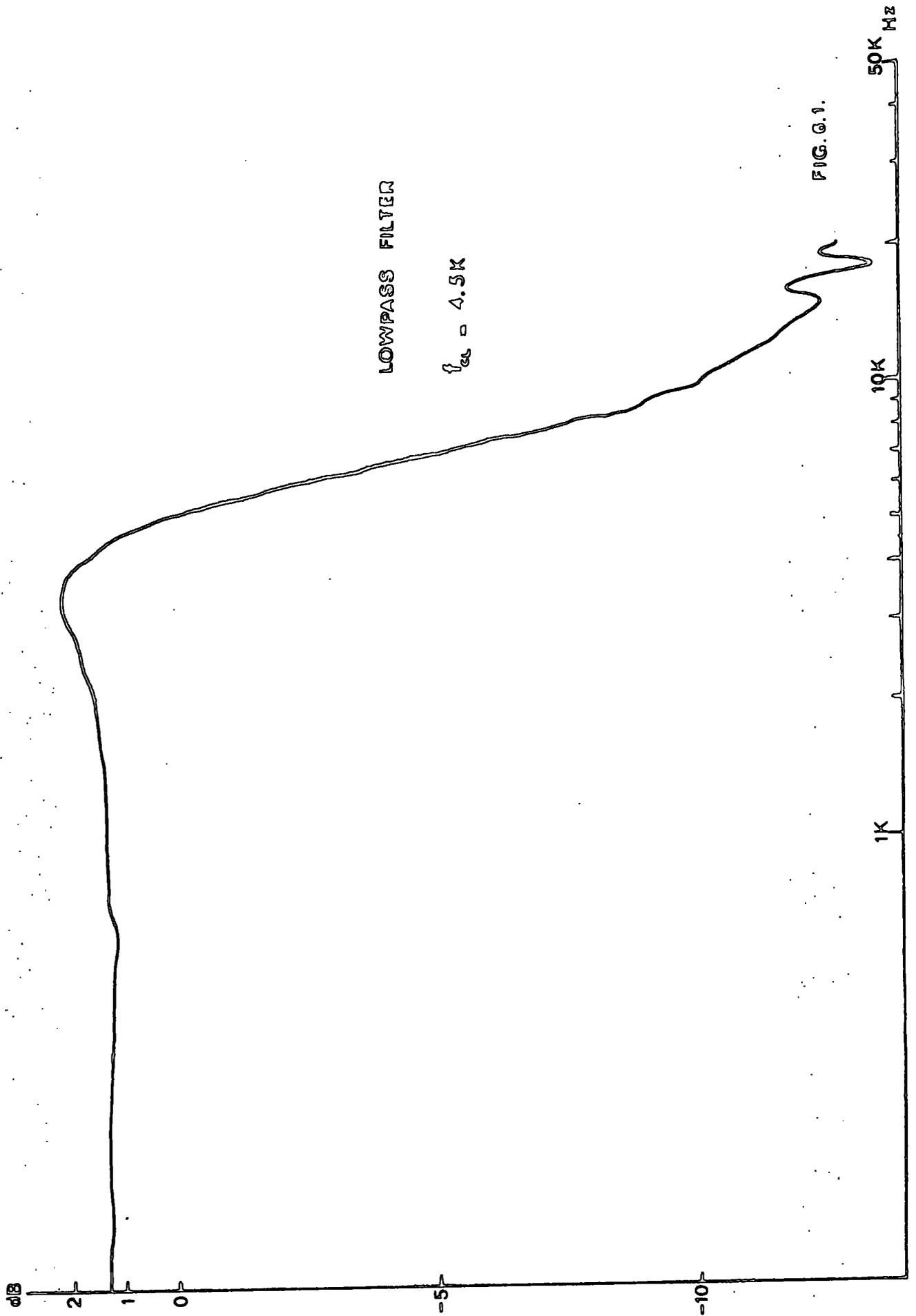
6.2. High pass filters.

For high pass filtering, the problem of overflow becomes serious, and, with the 8 bit word length, the filter could not be tested with the calculated coefficients. However, by modifying the coefficients, as shown in Table 6.2., and also reducing the **d.c** transient, to reduce the overflow problem, see also section 6.4., some results have been obtained and the frequency responses are depicted in Fig. 6.7. to Fig. 6.10. It is seen that the cut off frequencies are shifted and the attenuation slopes are reduced to approximately 6 dB per octave. The ripple in the pass band is about 1 dB. These degenerations are results from the modification of the coefficients.

6.3. Band pass filters.

Using the coefficients for low pass and high pass filters, band pass filters implemented via multiplexing were obtained, and the frequency responses are shown in Fig. 6.11. to Fig. 6.14. The coefficients used are given in Table 6.3. It is seen that the lower attenuation slopes and the upper attenuation slopes are those for the high pass and low pass filters used respectively as expected. The ripple in the pass band is approximately 1 dB.

Because the delay time is increased by the wiring, the arithmetic unit could not be used at the designed sampling frequency of **80 KHz**. Instead, a sampling frequency of **70.5 KHz** was used resulting in the shifting of the cut off frequencies as seen in Fig. 6.11. to Fig. 6.14. However, this problem can be solved by reducing the baseband
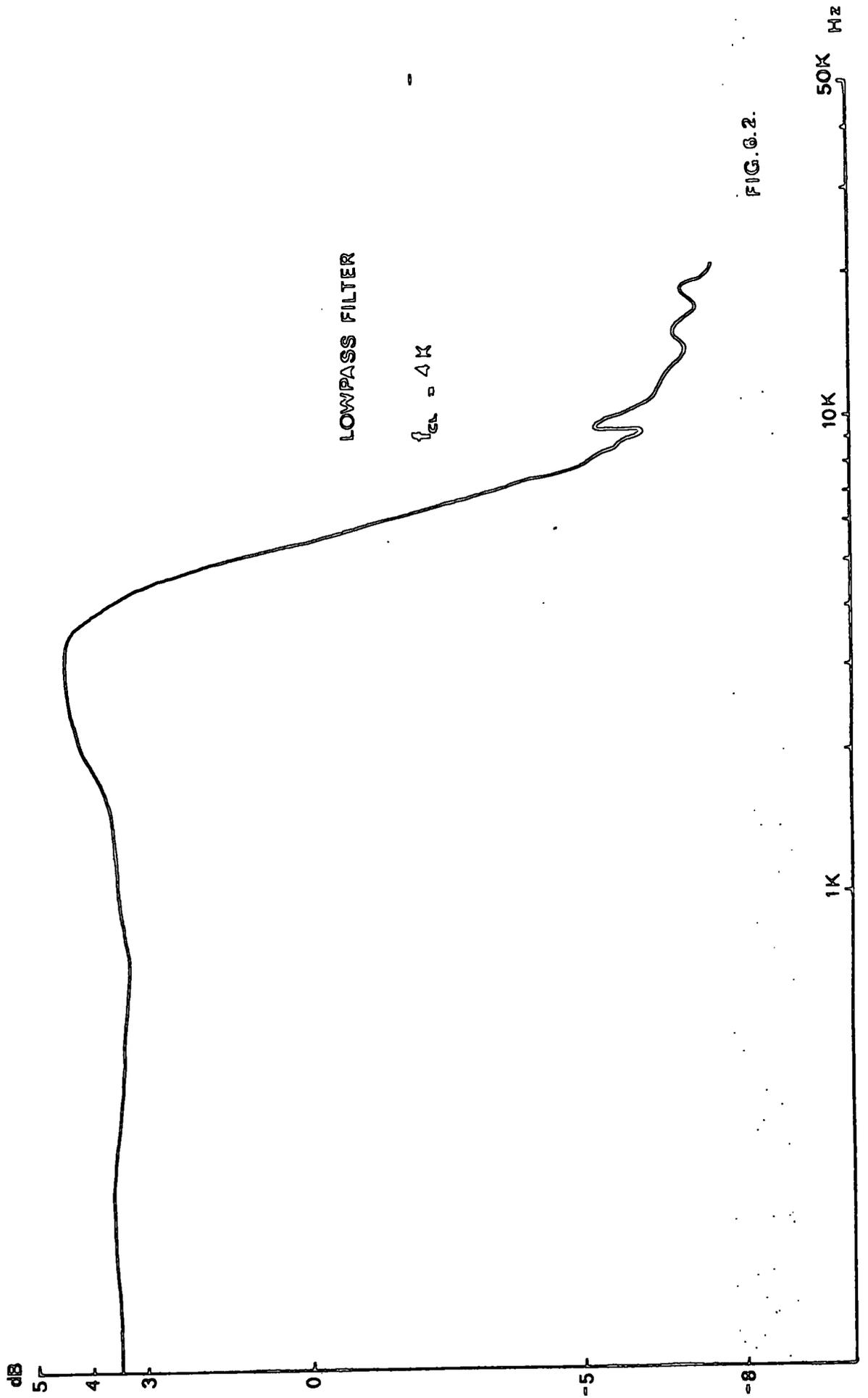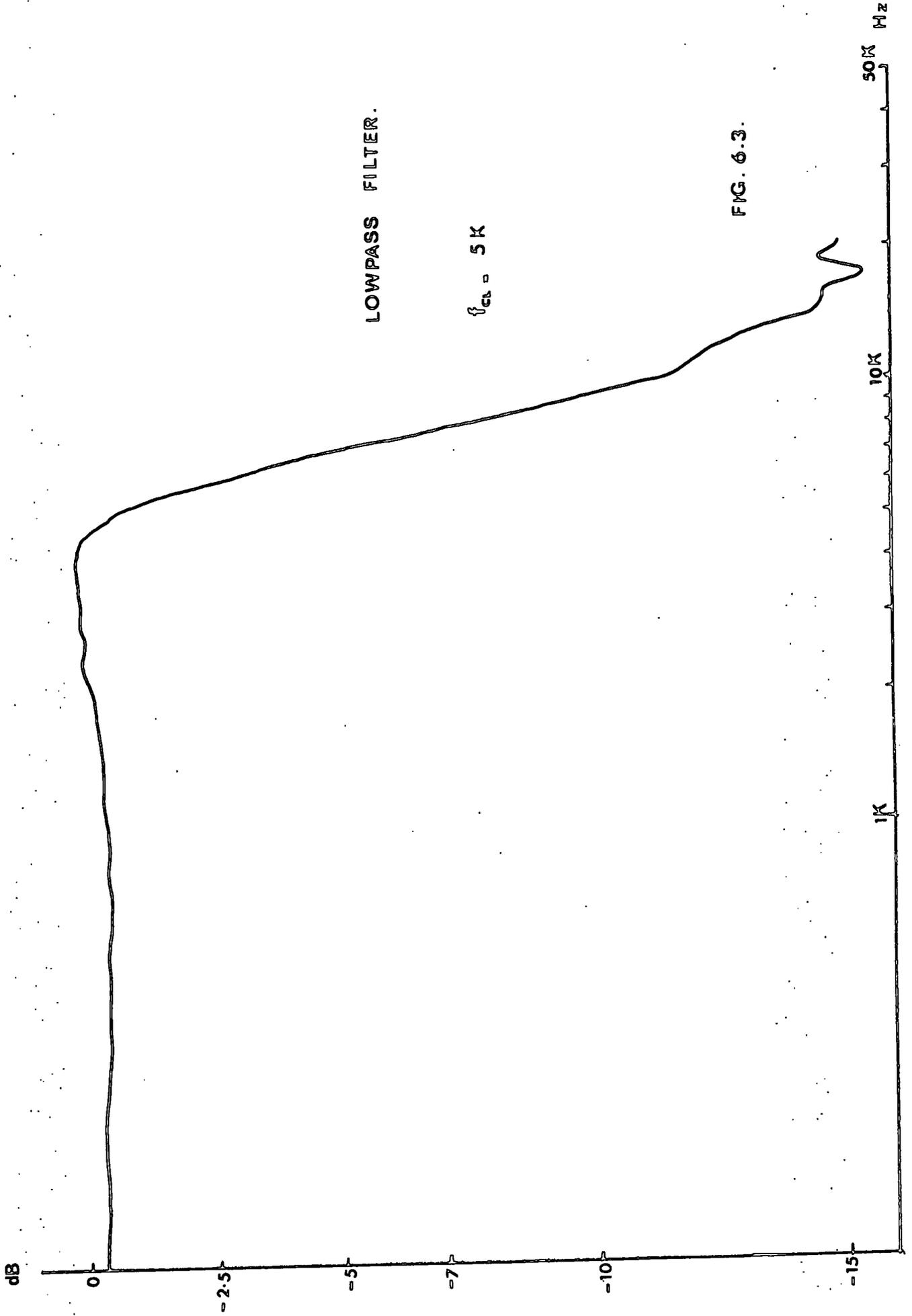
LOWPASS FILTER

$f_{GL} = 4.5K$

FIG. 0.1.

LOWPASS FILTER

$f_{cL} = 4K$

FIG. 8.2.

LOWPASS FILTER.

$f_{co} = 5K$

FIG. 6.3.

LOWPASS FILTER.

$f_{CL} = 8\text{K}$

FIG. 6.4.

80

LOWPASS FILTER

$f_{GL} = 10 K$

Fig 6.5

dB

-11½
-12

-15

-20
-21

1K   10K   50K   Hz

LOWPASS FILTER

$f_G = 16 K$

FIG. 6.6

HIGHPASS FILTER

$f_{CH} = 5K$

dB

-11
-12
-15
-18

1K   10K   50K   Hz

FIG 6.7

HIGHPASS FILTER

$f_{CM} = 8 K$

FIG.6.8.

HIGHPASS FILTER

$f_{cu} = 10K$

FIG. 6.9

HIGHPASS FILTER.

$f_{(co)} = 15 K$

dB

5

0

-1

1K   10K   50K   H z

F I G. 6.10.

BANDPASS FILTER

$f_{CH} = 5K$

$f_{CL} = 10K$

FIG. 6.11

BANDPASS FILTER

$f_{CH} = 5$ K

$f_{CL} = 15$ K

FIG. 6.12.

BANDPASS FILTER.

$f_{cH} = 5\ K$

$f_{cL} = 16\ K$
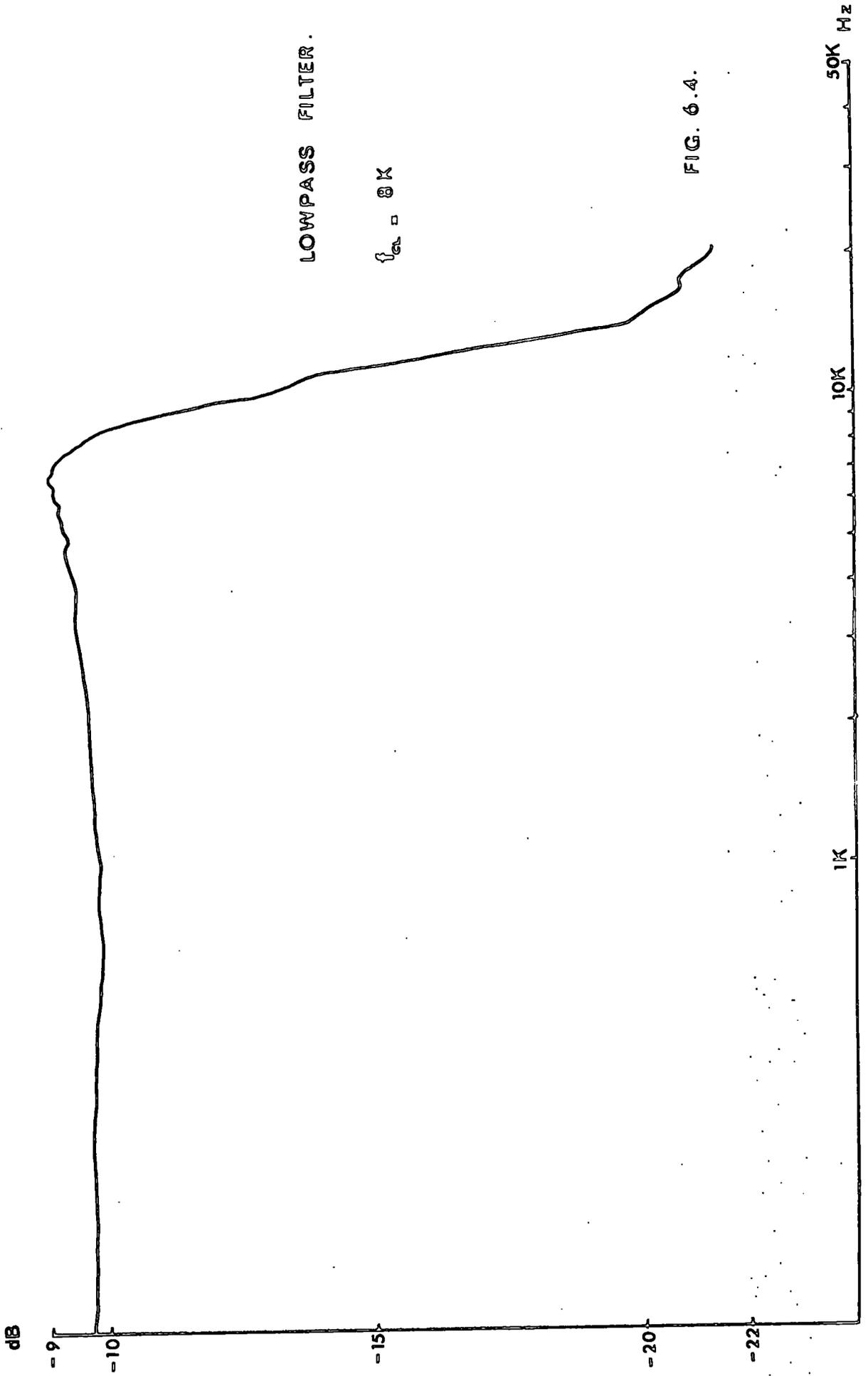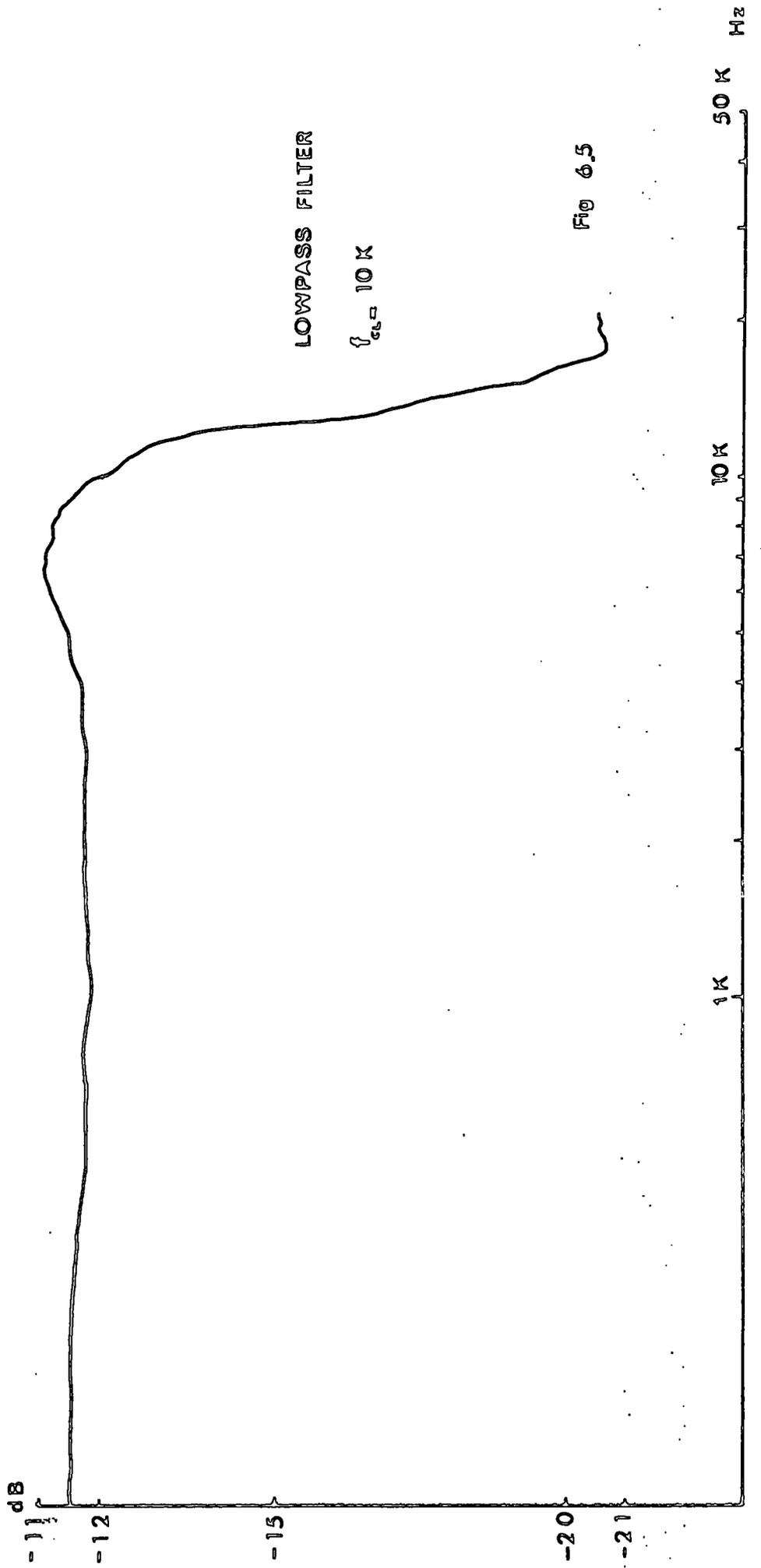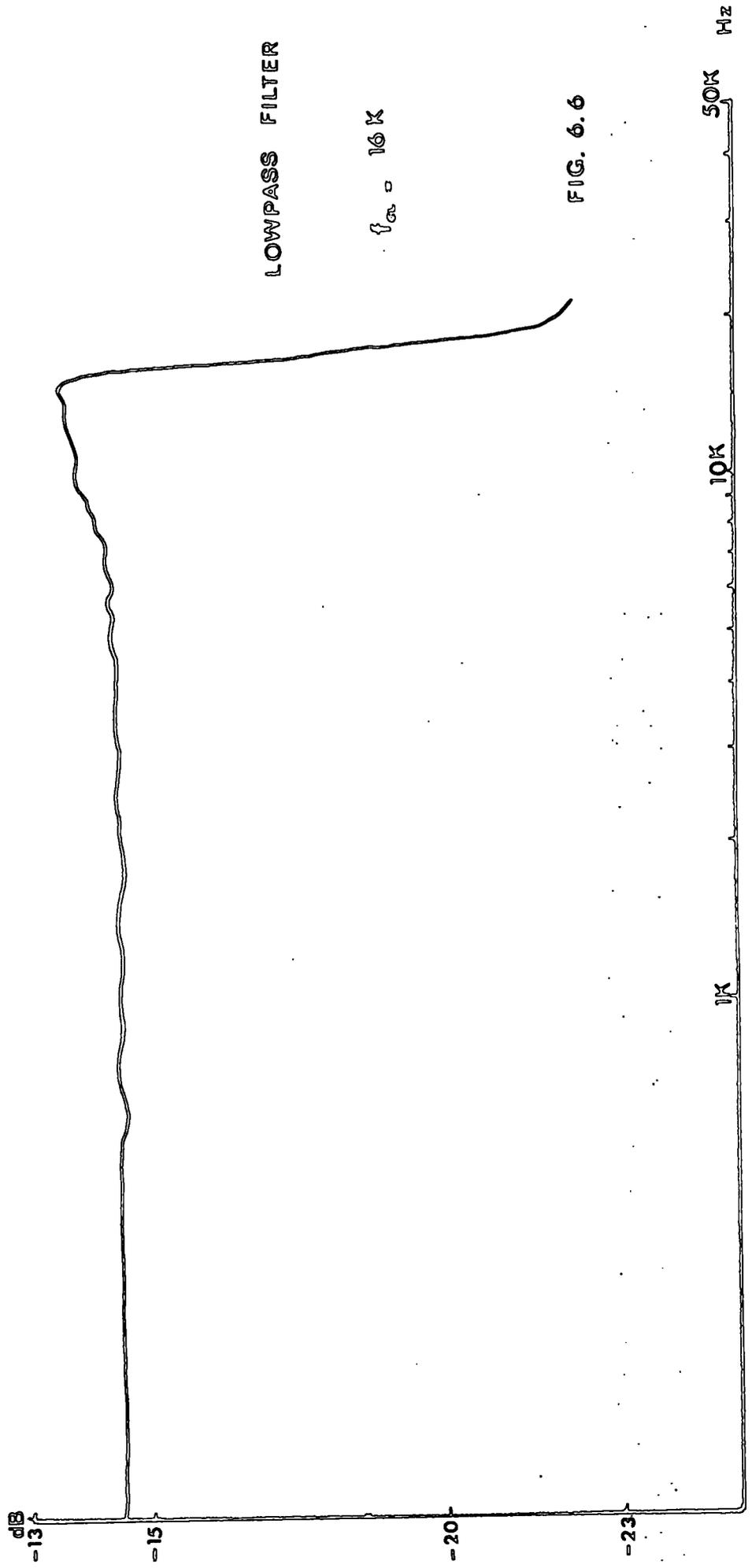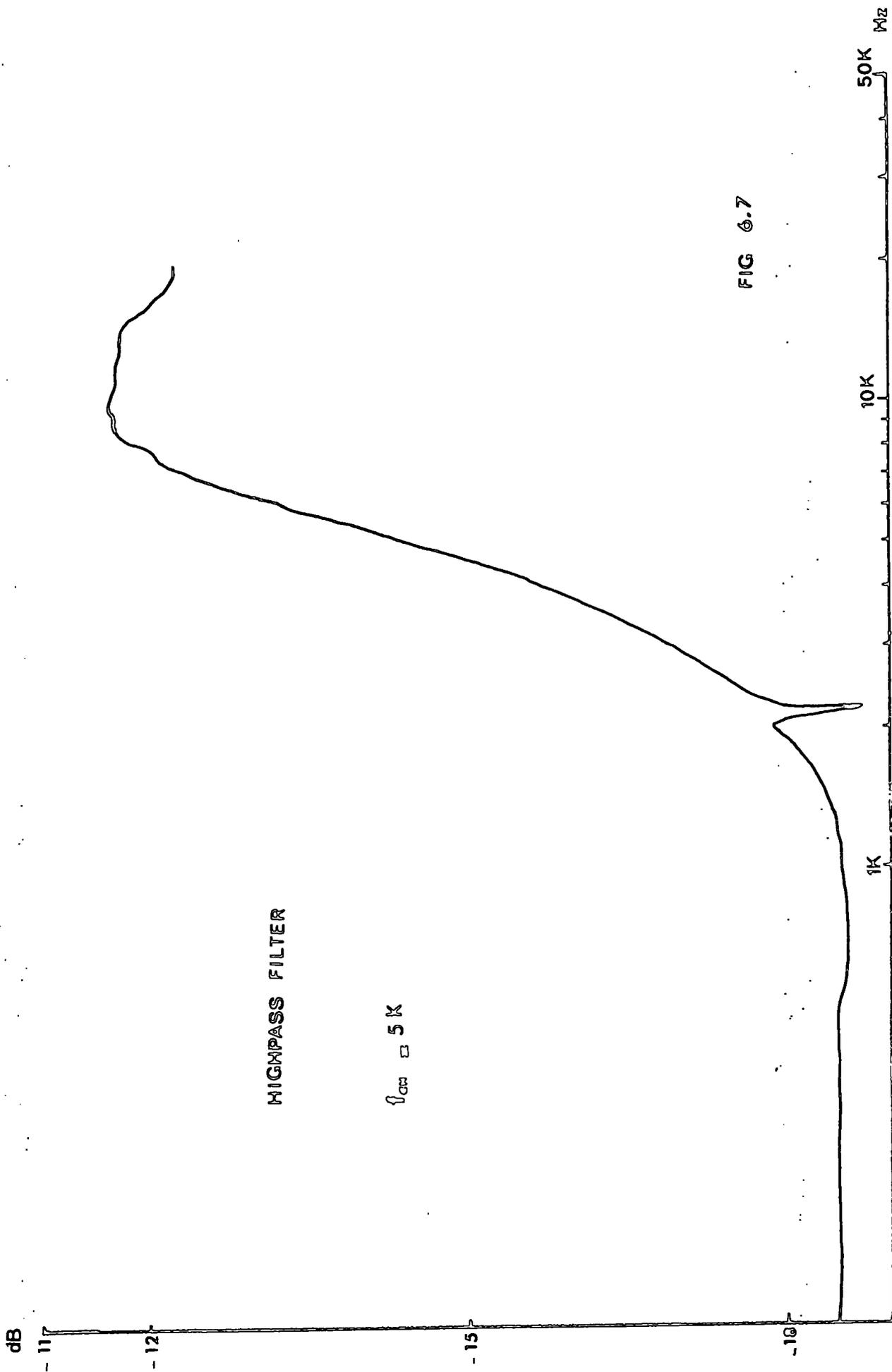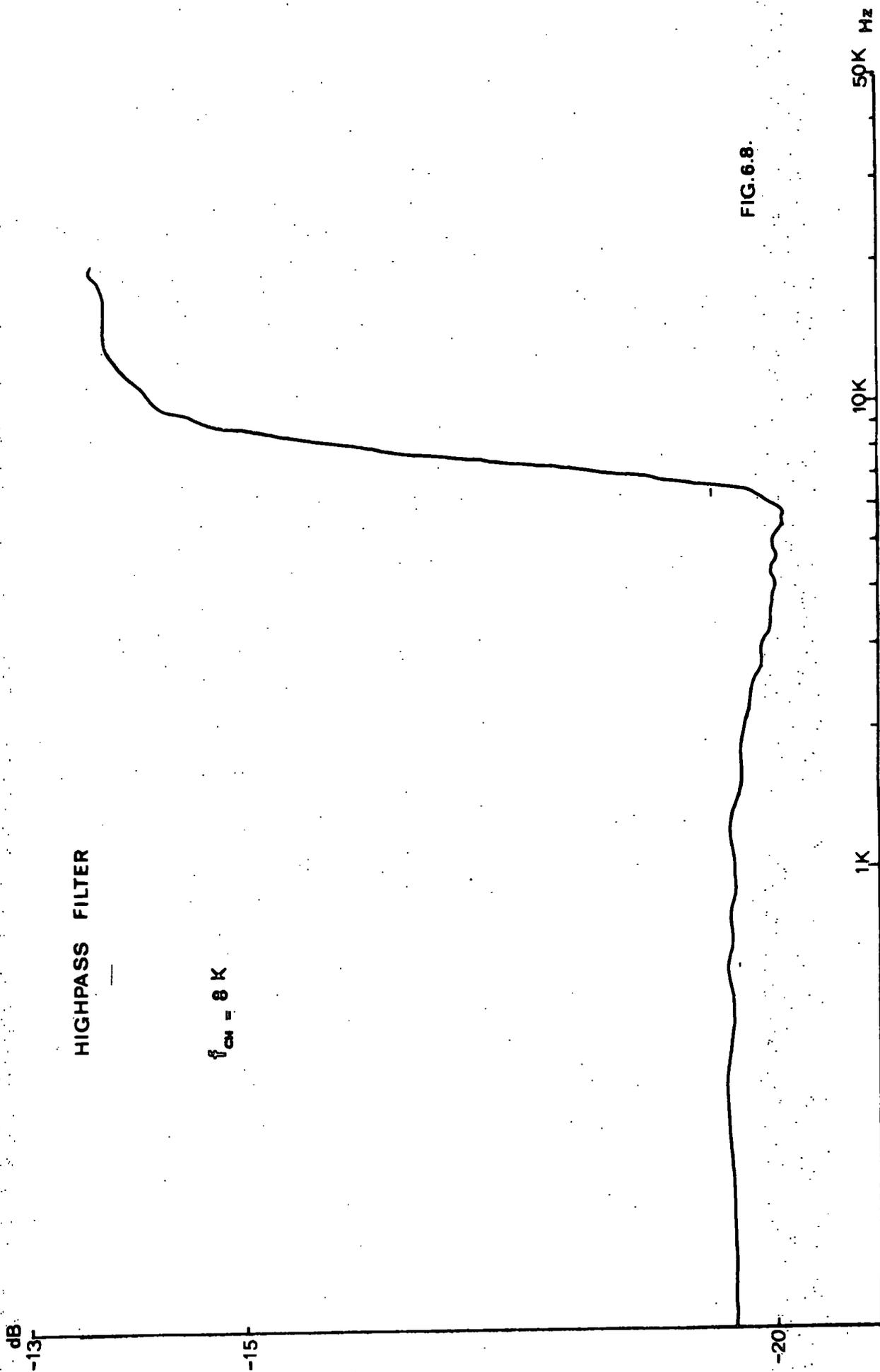
FIG. 6.13.

BANDPASS FILTER

$f_{CH} = 10\ K$
$f_{CL} = 15\ K$
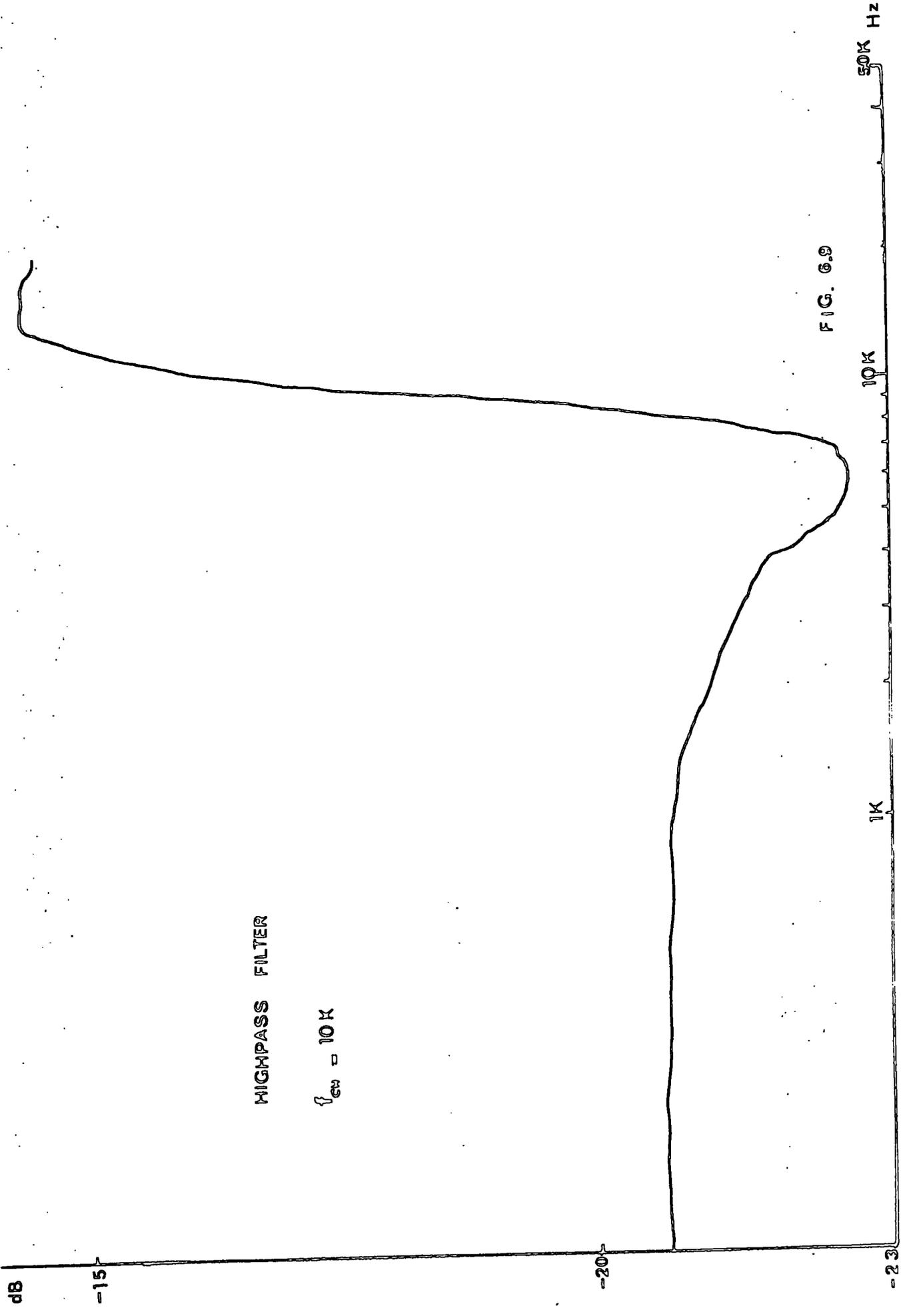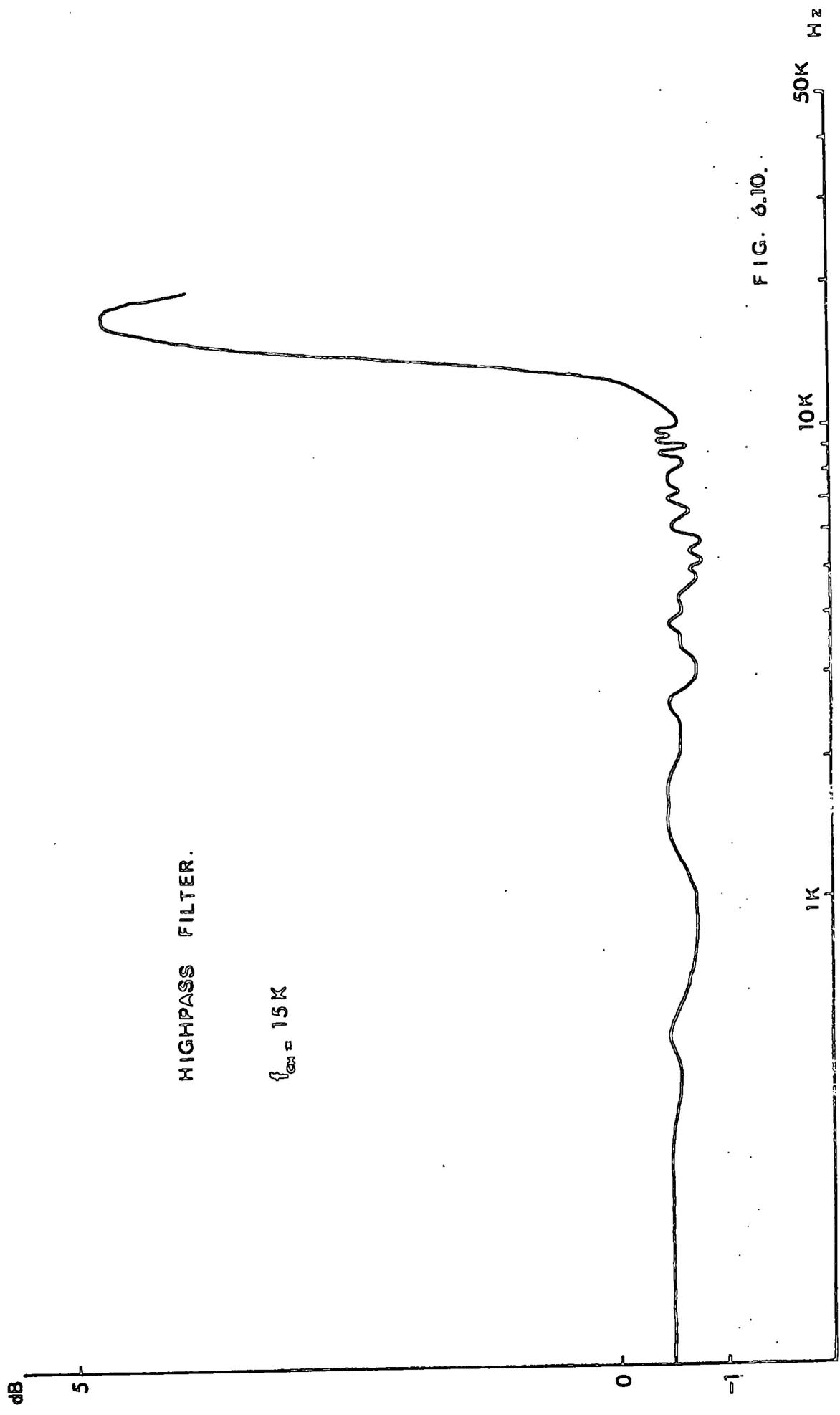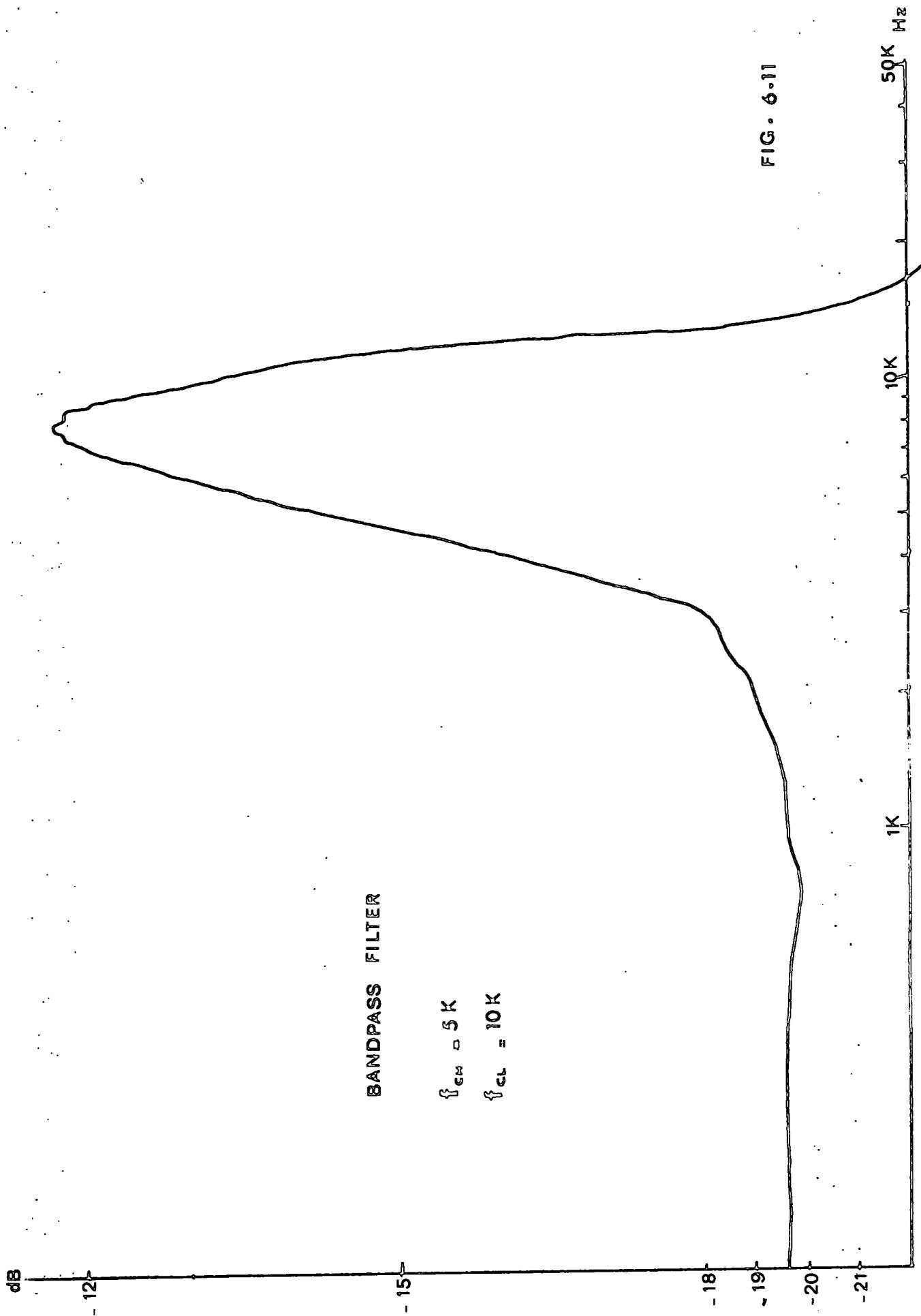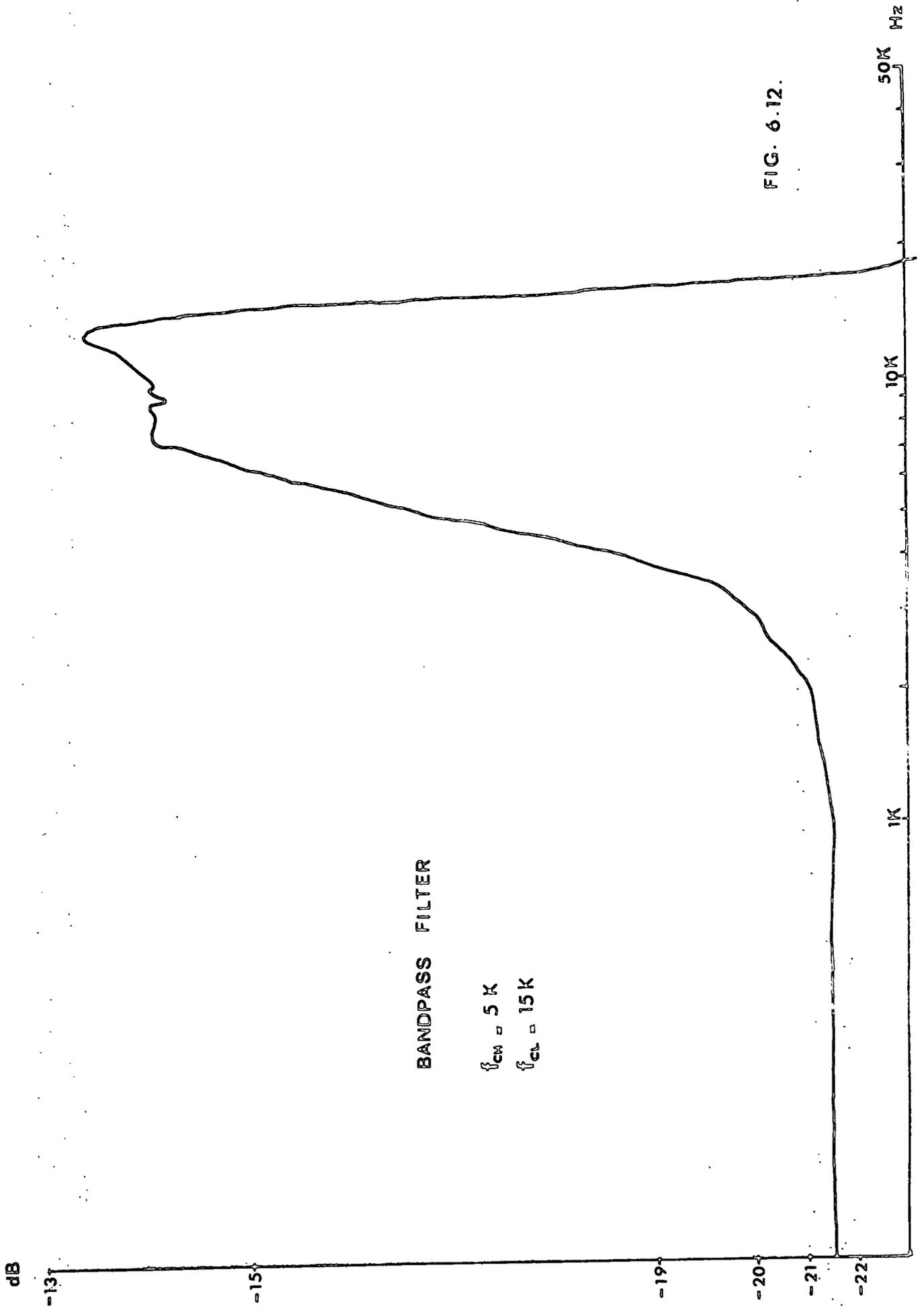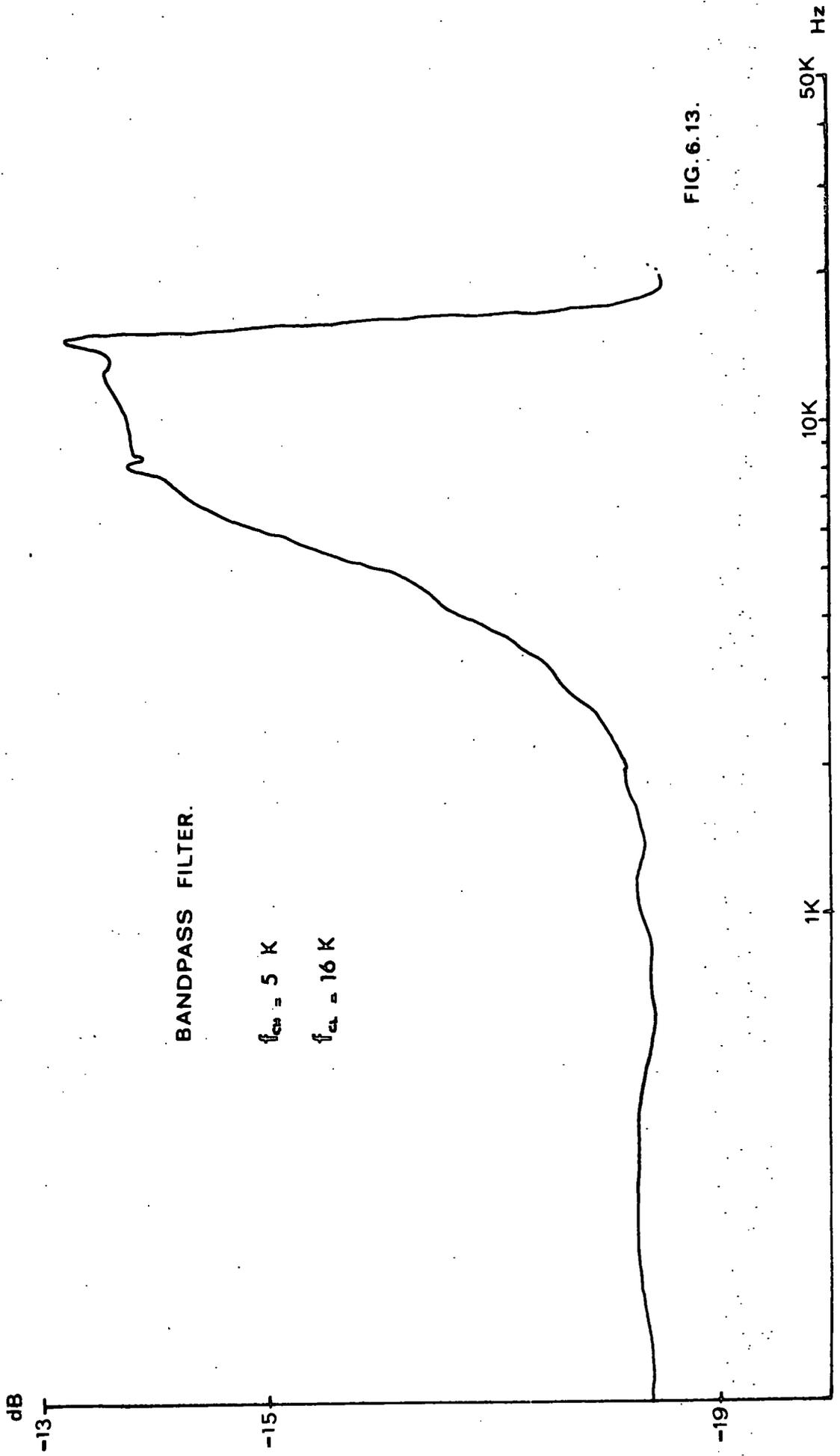
FIG. 6.14.
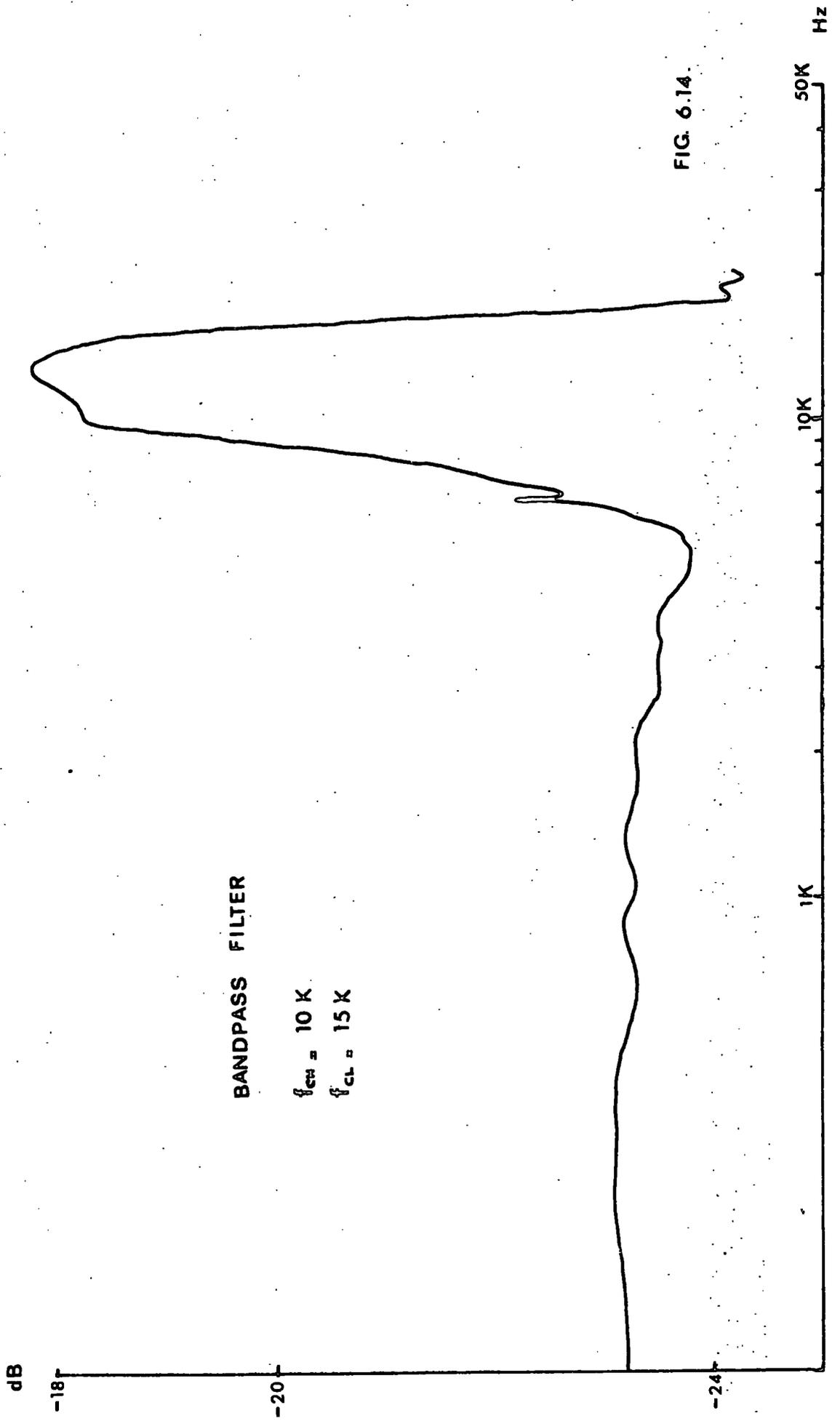
Table C.1. The Lowpass Filter Coefficients.

FIG.	F_s (Hz)	f_{CL} (Hz)	α'_0	$\dfrac{\alpha'_1}{2}$	$\dfrac{\beta_1}{2}$	β_2
6.1.	40K	4.0K	0.51562	0.51562	-0.59375	0.51562
6.2.	40K	4.5K	0.47656	0.47656	-0.54687	0.47656
6.3.	40K	5K	0.44531	0.44531	-0.49218	+0.44531
6.4.	40K	8K	0.32812	0.32812	-0.17187	0.32810
6.5.	40K	10K	0.31250	0.31250	0.03906	0.31250
6.6.	40K	16K	0.53906	0.53906	0.63281	0.53906

Table C.2. The Highpass Filter Coefficients.

FIG.	F_s (Hz)	f_{CH} (Hz)	α'_0	$\dfrac{\alpha'_1}{2}$	$\dfrac{\beta_1}{2}$	β_2
6.7.	40K	5K	0.46875	-0.46875	-0.28125	0.25000
6.8.	40K	8K	0.46875	-0.28906	-0.23437	0.36718
6.9.	40K	10K	0.31250	-0.18750	-0.03125	0.31250
6.10	40K	15K	0.44531	-1.0	0.49218	0.44531

Table 6.3.   The Bandpass Filter Coefficients.

FIG.	F_s (Hz)	f_1 (Hz)	f_2 (Hz)	α'_{OA}	α'_{OB}	$\dfrac{\alpha'_{1A}}{2}$	$\dfrac{\alpha'_{1B}}{2}$	$\dfrac{\beta_{1A}}{2}$	$\dfrac{\beta_{1B}}{2}$	β_{2A}	β_{2B}
6.11.	70.5K	5K	10K	0.31250	0.46875	0.31250	-0.46875	0.03906	-0.28125	0.31250	0.25000
6.12.	70.5K	5K	15K	0.47656	0.46875	0.47656	-0.46875	0.53906	-0.28125	0.47656	0.25000
6.13.	70.5K	5K	16K	0.53906	0.46875	0.53906	-0.46875	0.63281	-0.28125	0.53906	0.25000
6.14.	70.5K	10K	15K	0.47656	0.31250	0.47656	-0.18750	0.53906	-0.03125	0.47656	0.31250

and modifying the coefficients.

6.4. <u>Discussion.</u>

Computer simulation of the d.c responses of the low pass and high pass filters, (the programme is given in Appendix C), shows that overflow problems at the intermediate stage W(z) become a limiting factor in the operation of the filters. These repponses, which have been plotted for 80 sample points, are shown in Fig. 6.15. and Fig. 6.16. The d.c transient response become larger at the lower cut off frequencies. In other words, a filter to be used at the low range cut off frequencies must have a long word length. However, for midrange cut off frequencies and above, it is seen that a filter with limited word length can be used and this was confirmed by the results in the previous sections.

Some possible solutions may be suggested to solve the overflow problem. Firstly, a non recursive realization of the filter, without the intermediate stage W(z) , could be implemented. With the use of the signed 2's complement number representation, overflow problem should only appear at the output, see (11) , and would be eliminated by adding few extra bits. Secondly, because the overflow problem is the result of the d.c transient input, it could be decreased by reducing the d.c level. This can be done by the bipolar adjustment in the A-D converter. Limitation of the input signal would also assist. This was used for the high pass filters in section 6.2. But, this solution could be used only for simple signals. For practical audio input signals, in which transients

may occur, this solution may not be valid. Also a new A-D converter

capable of eliminating the **d.c** transient input must be used.

Thirdly, extra word length might be provided for **W(Z)**, perhaps

incorporating scaling factors. But then all following arithmetic

operations will need this extra word length too. Finally, for

recursive filters, a new intermediate stage or a new number

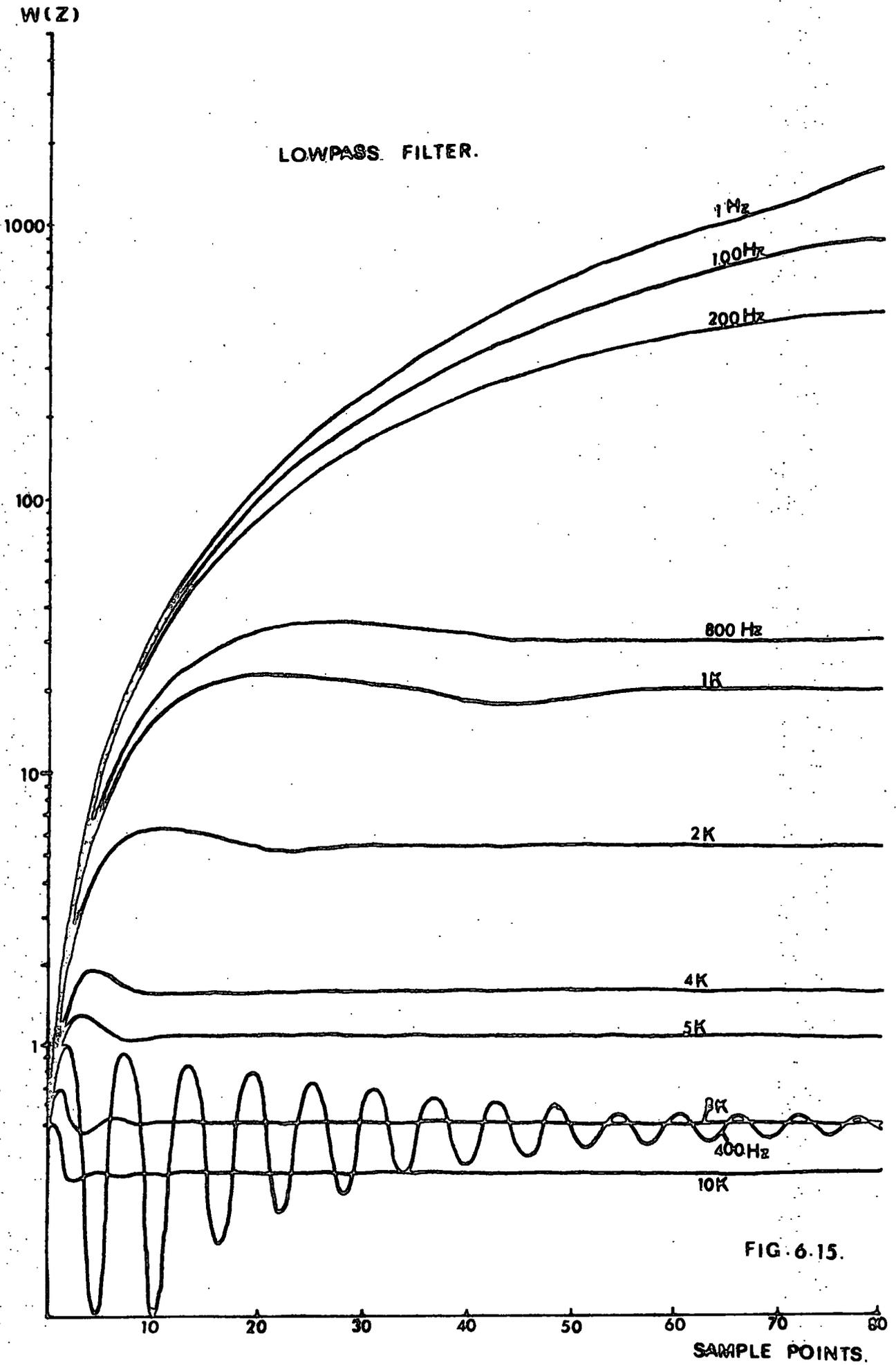representation might be defined to overcome this problem.

LOWPASS FILTER.

FIG. 6.15.

W (Z)

HIGHPASS FILTER.
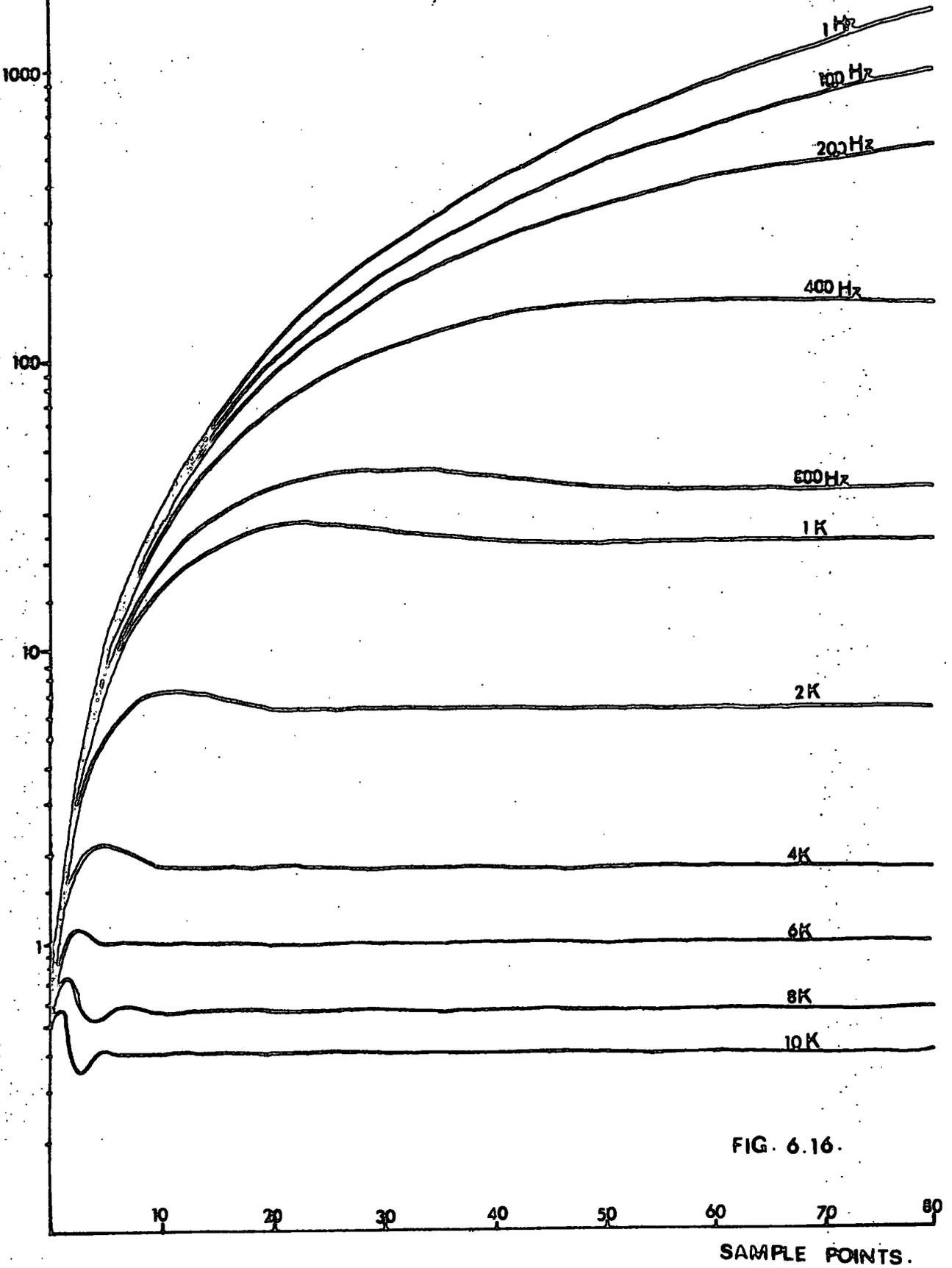
1000

100

10

1

1 Hz

100 Hz

200 Hz

400 Hz

800 Hz

1 K

2 K

4 K

6 K

8 K

10 K

FIG. 6.16.

10   20   30   40   50   60   70   80

SAMPLE POINTS.

Appendix A.

Determination of the coefficients for high-order digital filters.

The use of subfilters as basic building blocks for a high-order digital filter has been introduced in chapters 2 and 3. The parallel form is preferable for extension of our filter to a high-order filter, so we shall consider the determination of the subfilters for the parallel form.

It has been shown in chapter 3 that for an Nth order continuous filter having the transfer function

$$H(S) = \prod_{i=1}^{k} \frac{1}{\left[s - (a_i \pm jb_i)\right]} \qquad A.1.$$

A second order subfilter can be written as

$$H_i(S) = \frac{(c_i + jd_i)}{\left[s - (a_i + jb_i)\right]} + \frac{(c_i - jd_i)}{\left[s - (a_i - jb_i)\right]} \qquad A.2.$$

where $(a_i \pm jb_i)$ are the ith complex and its conjugate pole positions.

$(c_i \pm jd_i)$ are the residues evaluated at the ith pole and its conjugate, and

k is the integer part of $\frac{K + 1}{2}$

For convenience in writing, we will ignore the subscript i, and then rewrite equation (A.2.) as

$$H(S) = \frac{(c + jd)}{\left[s - (a + jb)\right]} + \frac{(c - jd)}{\left[s - (a - jb)\right]} \qquad A.3.$$

or
$$\pi(S) = 2 \; \frac{(cs-ac-bd)}{\left[(s-a)^2 + b^2\right]} \qquad \text{A.4.}$$

Using the bilinear Z-transformation, we replace s by $\frac{2}{T}\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right]$,

therefore, the numerator of equation (A.4.) becomes

$$2\left[cs-ac-bd\right] \;=\; 2\left[c\frac{2}{T}\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right] -ac-bd\right]$$

$$=\; 2\left[\frac{\dfrac{2c}{T} - \dfrac{2c}{T}Z^{-1} -ac-acZ^{-1}-bd-bdZ^{-1}}{1+Z^{-1}}\right]$$

$$=\; \frac{\left[\left(\dfrac{4c}{T} - 2ac - 2bd\right) - \left(\dfrac{4c}{T} - 2ac + 2bd\right)Z^{-1}\right]}{1+Z^{-1}} \qquad \text{A.5.}$$

and the denominator becomes

$$(s-a)^2 + b^2 \;=\; \left[\frac{2}{T}\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right] - a\right]^2 + b^2$$

$$=\; \left[\frac{2}{T}\right]^2\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right]^2 - 2\left[\frac{2}{T}\right]a\left[\frac{1-Z^{-1}}{1+Z^{-1}}\right] + a^2 + b^2$$

$$=\; \frac{\left[\dfrac{2}{T}\right]^2(1-Z^{-1})^2 -2a\dfrac{2}{T}(1-Z^{-1})(1+Z^{-1}) + a^2(1+Z^{-1})^2 + b^2(1+Z^{-1})^2}{(1+Z^{-1})^2}$$

$$=\; \frac{\dfrac{4}{T^2} - \dfrac{8}{T^2}Z^{-1}+\dfrac{4}{T^2}Z^{-2} -4a+4aZ^{-2}+a^2 +2a^2 Z^{-1}+a^2Z^{-2}+b^2 +2b^2Z^{-1}+b^2 Z^{-2}}{(1+Z^{-1})^2}$$

$$=\; \frac{\left(\dfrac{4}{T^2} - \dfrac{4a}{T}+a^2+b^2\right) +\left(2a^2+2b^2-\dfrac{8}{T^2}\right)Z^{-1} + \left(\dfrac{4}{T^2} + \dfrac{4a}{T} +a^2 +b^2\right)Z^{-2}}{(1+Z^{-1})^2}$$

$$= \frac{K\left(1 + \beta_1 Z^{-1} + \beta_2 Z^{-2}\right)}{(1+Z^{-1})^2} \qquad \text{A.6.}$$

Let $D = \left(1 - \frac{aT}{2}\right)^2 + \left(\frac{bT}{2}\right)^2$

Therefore, $K = \left(\frac{2}{T}\right)^2 D$

$$\beta_1 = \frac{2a^2 + 2b^2 - 8/T^2}{\left(\frac{2}{T}\right)^2\left[\left(1 - \frac{aT}{2}\right)^2 + \left(\frac{bT}{2}\right)^2\right]}$$

$$= \frac{-2\left[1 - \left(\frac{aT}{2}\right)^2 - \left(\frac{bT}{2}\right)^2\right]}{D}$$

$$\beta_2 = \frac{\frac{4}{T^2} + \frac{4a}{T} + a^2 + b^2}{\left(\frac{2}{T}\right)^2\left[\left(1 - \frac{aT}{2}\right)^2 + \left(\frac{bT}{2}\right)^2\right]}$$

$$= \frac{\left(1 + \frac{aT}{2}\right)^2 + \left(\frac{bT}{2}\right)^2}{D}$$

Substituting equation (A.5.) and (A.6.) into (A.4.), H(S)

becomes H(Z)

$$H(Z) = \frac{(1+Z^{-1})\left[\left(\frac{4c-2ac-2bd}{T}\right) - \left(\frac{4c+2ac+2bd}{T}\right)Z^{-1}\right]}{K(1+\beta_1 Z^{-1}+\beta_2 Z^{-2})}$$

$$= \frac{(1+Z^{-1})}{1+\beta_1 Z^{-1}+\beta_2 Z^{-1}}\left[\frac{(4c-2ac-2bd)}{T} - \frac{(4c+2ac+2bd)Z^{-1}}{T}\right]\frac{T^2}{4D}$$

$$= \frac{(1+Z^{-1})}{1+\beta_1 Z^{-1}+\beta_2 Z^{-2}}\left[\left(\frac{cT}{D} - \frac{acT^2}{2D} - \frac{bdT^2}{2D}\right) - \left(\frac{cT}{T} + \frac{acT^2}{2D} + \frac{bdT^2}{2D}\right)Z^{-1}\right]$$

$$= \frac{(1 + z^{-1})\ (A_0 + A_1 z^{-1})}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}$$

$$= \frac{A_0 + A_0 z^{-1} + A_1 z^{-1} + A_1 z^{-2}}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}$$

$$= \frac{A_0 + A_0 z^{-1} + A_1 z^{-1} + A_1 z^{-2} + \frac{A_1}{\beta_2} - \frac{A_1}{\beta_2} + \left(\frac{A_1 \beta_1}{\beta_2}\right) z^{-1} - \left(\frac{A_1 \beta_1}{\beta_2}\right) z^{-1}}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}$$

$$= \frac{\frac{A_1}{\beta_2}\left(1 + \beta_1 z^{-1} + \beta_2 z^{-2}\right) + \left(A_0 - \frac{A_1}{\beta_2}\right) + \left[A_0 + A_1\left(1 - \frac{\beta_1}{\beta_2}\right)\right] z^{-1}}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}$$

$$= \frac{A_1}{\beta_2} + \frac{\left(A_0 - \frac{A_1}{\beta_2}\right) + \left[A_0 + A_1\left(1 - \frac{\beta_1}{\beta_2}\right)\right] z^{-1}}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}$$

$$= C_0 + \frac{\alpha_0 + \alpha_1 z^{-1}}{1 + \beta_1 z^{-1} + \beta_2 z^{-2}}$$

A.7.

where $C_0 = \dfrac{A_1}{\beta_1}$

$\alpha_0 = A_0 - \dfrac{A_1}{\beta_2}$

$\alpha_1 = A_0 + A_1 \left(1 - \dfrac{\beta_1}{\beta_2}\right)$

$$A_0 = \frac{cT}{D} - \frac{acT^2}{2D} - \frac{bdT^2}{2D} = \frac{T\left[c\left(1 - \frac{aT}{2}\right) - d\left(\frac{bT}{2}\right)\right]}{D}$$

$$A_1 = -\frac{cT}{D} - \frac{acT^2}{2D} - \frac{bdT^2}{2D} = \frac{-T\left[c\left(1 + \frac{aT}{2}\right) + d\left(\frac{bT}{2}\right)\right]}{D}$$

$$\beta_1 = \frac{-2\left[1 - \left(\frac{aT}{2}\right)^2 - \left(\frac{bT}{2}\right)^2\right]}{D}$$

$$\beta_2 = \frac{\left(1 + \frac{aT}{2}\right)^2 + \left(\frac{bT}{2}\right)^2}{D} \qquad\qquad A.8.$$

It should be stressed here that the equation A.8. is valid only for subfilters of a high-order filter, ($N > 2$). But, if a second order continuous filter function is given, as in section 4 , equation (A.8.) can be derived directly from the pole positions without using the residues.

APPENDIX B : CIRCUITS OF A-D CONVERTER AND D-A CONVERTER.

A-D CONVERTER.
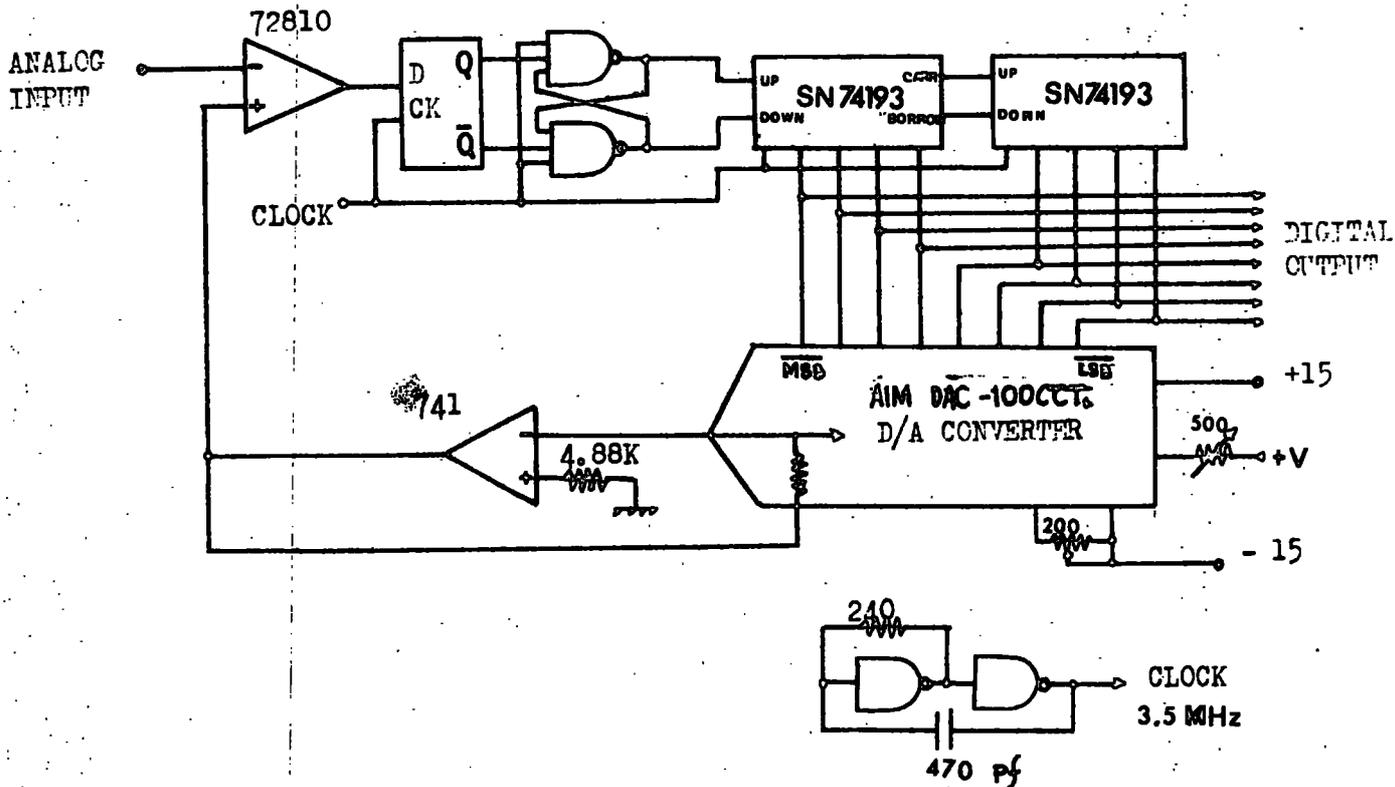
D-A CONVERTER.

V: BIPOLAR ADJUSTMENT.

Appendix C.
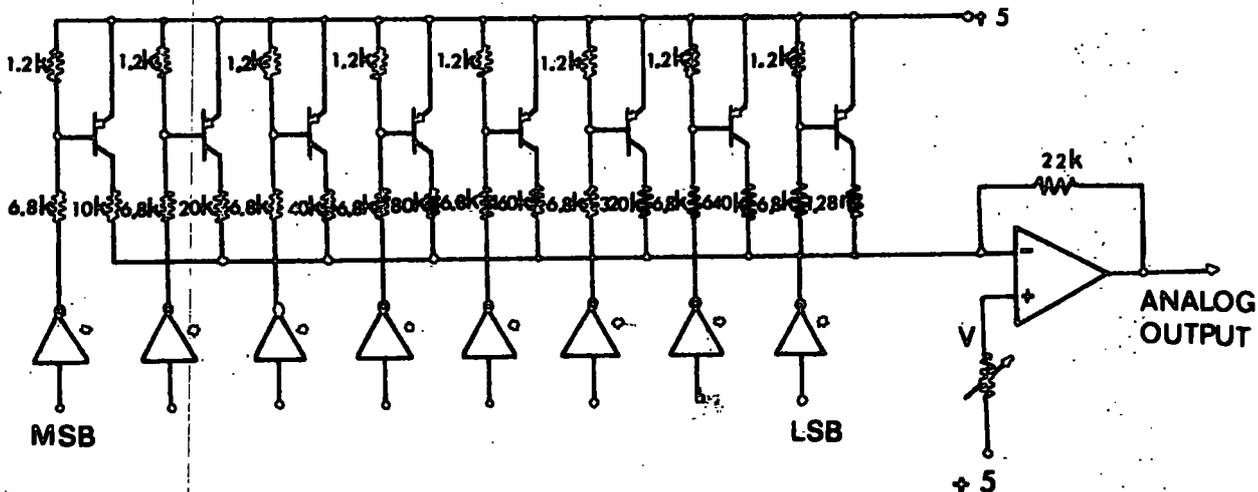
<u>A FORTRAN programme for determination of d&c: responses of the filters.</u>

```
      DIMENSION  WZ(150),A0(150),B1(150),B2(150),YZ(150)

      READ  100, ALPHAO, ALPHA1, BETA1, BETA2

100   FORMAT ( 4F13.10)

      PRINT 200, ALPHAO, ALPHA1, BETA1, BETA2

200   FORMAT (///5X,F13.10,5X,F13.10,5X,F13.10,5X,F13.10)

C

      WZ(1)=0

      WZ(2)=0

      WZ(3)=0

      XZ=0.5

C

C     XZ  IS THE D.C. INPUT.

C     WZ(I)  ARE THE INTERMEDIATE STAGE.

C     YZ(I)  ARE THE OUTPUTS.

C

      DO 1 I=4,104

      WZ(I)=XZ-BETA1 ⊙ WZ(I-1)-BETA2 * WZ(I-2)

      A0(I)=WZ(I) * ALPHAO

      A1(I)=WZ(I-1) * ALPHA1

      B1(I)=WZ(I-1) * BETA1

      B2(I)=WZ(I-2) * BETA2

      YZ(I)=A0(I)+A1(I)+B2(I)
```

```
    1   CONTINUE

        PRINT 300

  300   FORMAT (/////)

        PRINT 400,(XZ,WZ(I),A0(I),A1(I),B1(I),B2(I),YZ(I),I=4,104)

  400   FORMAT (2X,F10.5,2X,F10.5,2X,F10.5,2X,F10.5,2X,F10.5,2X,
       1F10.5,2X,F10.5)

        STOP

        END.
```

REFERENCES.

(1)   F. F. Kuo and J. F. Kaiser, System Analysis by Digital Computer, Wiley, 1967.

(2)   J. F. Kaiser, The Digital Filter and Speech Communication, IEEE. Trans. Audio and Electroacoustics, Vol. AU 16, June 1968.

(3)   R. M. Rader and B. Gold, Digital Filter Design Techniques in the Frequency domain, PROC. IEEE. Vol. 55, No. 2, Feb. 1967, pp 149-171.

(4)   R. M. Golden and J. F. Kaiser, Design of wideband sampled-data filter, Bell sys. Tech. J., Vol. 43, pt 2, pp 1533-1546.

(5)   R. M. Golden, Digital Filter Synthesis by sampled-data Transformation, IEEE. Trans. Audio. Electroacoustics, Vol. AU 16, No. 3, Sept. 1968, pp 321-329.

(6)   T. C. Hsia, A Technique for Synthesizing Digital Filters, IEEE. Trans. Instrumentation and Measurement, Vol. IM 18, No. 2, June 1969, pp 93-96.

(7)   Sonderegger, Digital Filter Realization of High Quality Bandpass Filter, IEEE. Computers and Communications Conf. Record, N.Y. 1969.

(8)   L. R. Rabiner and R. W. Schafer, Recursive and Nonrecursive Realizations of Digital Filters Designed by Frequency sampling Techniques, IEEE. Trans. Audio and Electroacoustics,

Vol. AU 19 No. 3, Sept. 1971, pp 200-207.

(9) A. Buttle, A. G. Constantinides, J. E. Brignell, On line
Digital Filtering, Electronic. Letter, Vol. 4, No. 12, June
1968, pp252-253.

(10) R. White, H. T. Nagle JR, Digital Filter Realizations Using
a special-purpose stored-program Computer, IEEE. Trans.
Audio and Electroacoustics, Vol. AU 20, No. 4, Oct. 1972,
pp 289-294.

(11) L. B. Jackson, J. F. Kaiser, H. S. McDonald, An Approach to
the Implementation of Digital Filters, IEEE. Trans. Audio
Electroacoustics, Vol. AU 16, No. 3, Sept. 1968, pp 413-
421.

(12) Y. Cho, F. G. Popp, A High Speed Arithmetic unit for
Digital Filter Implementation, NEREM Record, 1967, p 115.

(13) E. Avenhaus, A Proposal to Find Suitable Canonical Structures
for the Implementation of Digital Filters with Small
Coefficient word length, A report from the Institut. fur
Nachrichtentechnik, Universitat Erlangen-Nurnberg.

(14) E. Avenhaus, On the Design of Digital Filters with Coefficients
of Limited Word length, IEEE. Audio and Electroac., Vol.
AU 20, No. 3, Aug. 1972, pp 206-212.

(15) V. Mauro, A. R. Meo, Tables and graphs for Designing
Digital Chebyshev Filters, Electron. Lett., March 1970,
Vol. 6, No. 6, pp 155-157.

(16) R. Genesio, A. Laurentini, Tables for Butterworth-Digital

Filter Design, Electron.Lett, March 1970, Vol. 6, No. 6, pp 157-159.

(17) J. A. Cadzow and H. R. Martens, "Discrete-Time and Computer control System," Prentice-Hall, 1970.

(18) D. J. Nowak and P. E. Schmid, Introduction to Digital Filters, IEEE. Trans.Electro. Compatibility, Vol. EMC 10, No. 2, June 1968, pp 210-220.

(19) L. R. Rabiner. et,al, Terminology in Digital Siganl Processing, IEEE. Trans.Audio and Electroac, Vol. AU 20, No. 5, Dec. 1972, pp 322-337.

(20) A. G. Constantinides, Frequency Transformations for Digital Filters, Electron. Lett., Nov. 1967, Vol. 3, No. 11, pp 487-489.

(21) A. G. Constantinides, Frequency Transformations for Digital Filters, Electron.Lett., April 1968, Vol. 4, No. 7, pp 115-116.

(22) Betts, "Signal Processing, Modulation and Noise", The English Universities Press, 1970.

(23) Liu, Effect of Finite word length on the Accuracy of Digital Filter - A review, IEEE. Trans. Circuit Theory, Vol. CT 18, Nov. 1971, pp 670-677.

(24) B. Gold and C. M. Rader, Effects of Quantization Noise in Digital Filters, PROC. A FIPS 1966 Spring Joint Computer Conf., pp 213-219.

(25)   J. B. Knowles and E. M. Olcayto, Coefficient Accuracy and

Digital Filter Response, IEEE. Trans. Circuit Theory,

Vol. CT15, March 1968, pp 31-41.

(26)   J. B. Knowles and R. Edwards, Effect of a Finite-Word-Length

Computer in a Sampled-Data Feedback System, PROC. IEE,

Vol. 112, No. 6, June 1965, pp 1197-1207.

(27)   J. B. Knowles and R. Edwards, Finite Word-length Effects in

Multirate Direct Digital Control Systems., PROC. IEE, Vol.

112, No. 12, Dec. 1965, pp 2376-2384.

(28)   K. J. Astrom, E. I. Jury, and R. C. Agniel., A Numerical

Method for the Evaluation of Complex Integrals, IEEE. Trans.

Automat. Contr. (short paper)., Vol. AC 13, Aug. 1970,

pp 468-471.

(29)   J. K. Kaiser, Some practical considerations in the

realization of linear digital filters, PROC. 3rd Annu.

Allerton Conf. Circuit system theory, pp 621-633, 1965.

(30)   A. T. Anderson, Programmable Digital Filter Performs

Multiple Functions, Electronics(USA), Vol. 43, 1970,

pp 78-84.

(31)   Y. Chu, Digital Computer Design Fundamentals, McCraw-Hill,

1962.

(32)   Booth and Booth, Automatic Digital Calculator, 3rd Edition,

Butterworth, 1965.