DIGITAL FILTERING BASED ON

THE CONVOLUTION INTEGRAL

BY

RICHARD CARNEGIE, B.ENG., B.SC.

DIGITAL FILTERING BASED ON THE CONVOLUTION INTEGRAL

BY

RICHARD CARNEGIE, B.ENG., B.SC.

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Master of Engineering

McMaster University
November 1969

MASTER OF ENGINEERING (1969) McMASTER UNIVERSITY
(Electrical Engineering) Hamilton, Ontario.

TITLE: Digital Filtering Based on the Convolution Integral

AUTHOR: Richard Thomas Carnegie, B.Eng. (McMaster University), B.Sc. (University of Toronto)

SUPERVISOR: Professor S.S. Haykim

NUMBER OF PAGES: ix, 106

ABSTRACT:

A new method for realizing linear, time-invariant digital filters is developed and demonstrated. The result is based on the convolution integral. It is assumed that the specifications of the filter are known and from these, an appropriate analog filter is chosen. The properties of this filter are then retained by the digital filter after transformation. The behaviour of lowpass, highpass bandpass and bandstop digital filters is investigated in both the frequency and time domains, for both cascade and parallel structures. Based on these results, it is evident that the parallel structure is superior for lowpass and bandpass digital filters, and that the cascade structure is superior for highpass and bandstop digital filters.

PREFACE

This thesis has the purpose of developing a simple but soundly based transformation for digital filters. Since linear, time-invariant filters are the main concern, the convolution integral is evidently a suitable starting point. Inevitably, we are faced with compromise between accuracy and a useable result, but the experimental results show that the trade is a good one. In fact, the method developed here is shown to be the basis of another transformation, the impulse invariant. From this basis it is easy to understand the shortcomings of this method, and the advantages of the method which is developed in this thesis.

The theory has been extensively tested on a large general-purpose digital computer. We have not yet duplicated the arithmetic functions and delays on a laboratory-built special-purpose computer.

ACKNOWLEDGEMENTS

The constant interest and enthusiasm shown by the project supervisor, Dr. S. S. Haykim, has helped immensely in the day-to-day work which has gone into this thesis.

Several fellow students, too many to be named individually, also deserve my thanks for their critical listening and often their constructive help.

Finally, the generosity of IBM in supplying the typewriter used for this thesis is greatly appreciated.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Chapter 1 The Basic Ideas of Digital Filtering

1.1 Introduction

Without the availability of a large, fast digital computer such as the CDC 6400, the following work on digital filters would have remained a fruitless and in fact, a useless idea. The computer's ability to perform simple arithmetic operations many thousands of times per second, and to repeat such operations tirelessly is the key to the development of digital filtering.

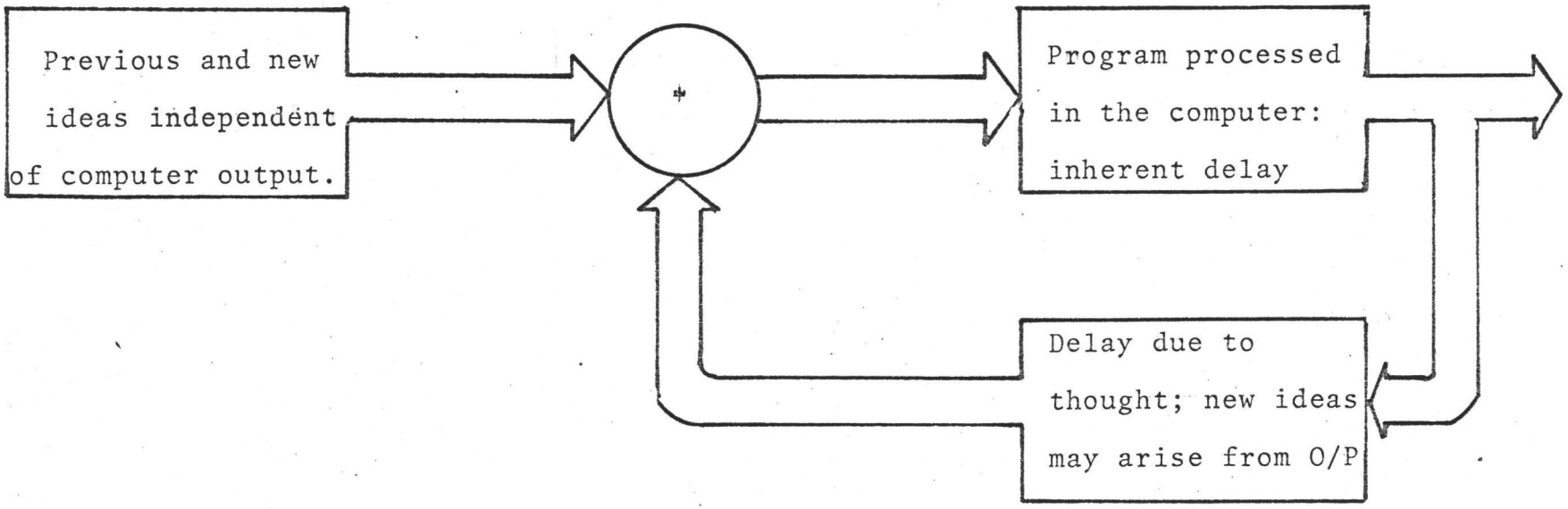
Throughout the investigations, it became evident, although the only visible accomplishment was on paper (i.e. printouts and several decks of computer program cards), that the computer was both a sophisticated laboratory instrument which could display results graphically or digitally, and an actual, physical realization of the particular digital filter specified in the instructions. The role of the deck of computer cards was to carry information as to how the available arithmetic functions should be "wired" together to realize the filter. The ease and speed with which a program could be altered was an important feature which allowed investigation of many details of the filter's behaviour.

The computer-human relationship in this project can be portrayed as a feedback system where the computer is the

forward path element which does only what it is told to do. The human is the summing device where the relevance of previous output information is considered and combined with his previous ideas and any new ones so as to alter the instructions to the computer. From the initial transient period, where errors plague the program, to the steady state where finally the program is debugged, the feedback relationship is an obvious one. Yet it provides the basis for an idea which is important here: simply, it is that the parts of the system are not in constant communication, but transmit information to one another only at discrete times. Explicitly, there is an inherent time delay between the discovery of an error or improvement and its implementation into a new program, and also a time delay before the new output is available. The system may therefore be described by Figure (1.1). The notion that the flow of information within a system is not necessarily continuous as in the familiar analog case, but that it may be discrete or sampled, is basic to the following work on digital filters.

1.2 Digital Filters

So that the meaning of the term "digital filter" is clear, let us consider what is meant by both words individually. First, the term filter can be applied to many input-output devices whether they are primarily electrical, mechan-



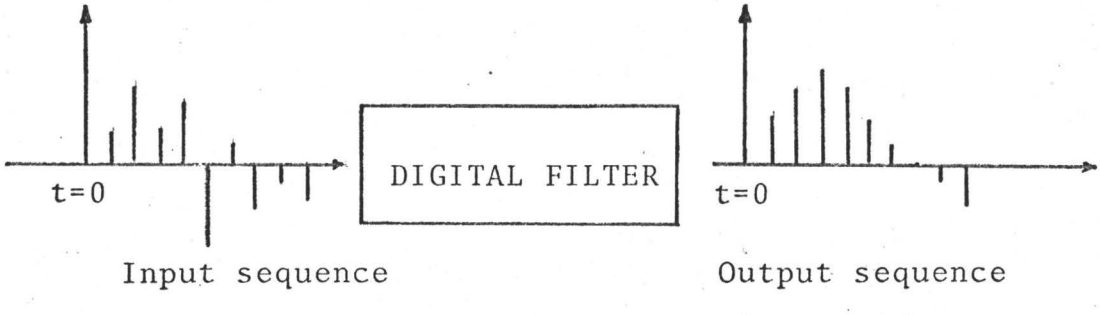
Feedback Mechanism

Figure (1.1)

ical or whatever in nature. The choice of the word filter over network or system implies that the frequency response of the device is of major importance.

By the term "digital" we mean that the input signal to the filter is a sequence of numbers which is fed into the filter at a regular rate, called the sampling frequency. Originally, this sequence may have been derived from a continuous voltage signal via an analog-to-digital converter, or it may be the original form of some information. In any case, the numbers of the input sequence are processed by the filter one-by-one and each time a new number enters the filter, a new number appears at its output. For large filters composed of several elementary filters in cascade, the numbers in the original input sequence are transformed several times before appearing at the external output. Within the filter, though, we require that each internal input and output sequence remain in synchronism. In Figure(1.2) we illustrate the meaning of digital inputs and outputs.

Figure (1.2)



1.3 Mathematical Description of Digital Filters

So as to make the mathematical description of linear, time-invariant digital filters readily acceptable, we shall draw a comparison with the more familiar description of linear, time-invariant analog filters. Corresponding to linear differential equations, Laplace transforms and the convolution integral, there are linear difference equations, Z-transforms and the convolution sum to characterize linear digital filters. As was suggested in Section 1.1, the calculation of each number in the output sequence of a linear digital filter is limited to some linear combination of past inputs, past outputs and the present input, if this is to be a causal filter. Thus, if the "present" is assigned the discrete time parameter nT (where T is the sampling period and n is an integer), then the present input and output values are defined as x(nT) and y(nT), respectively. Furthermore, both input and output sequences are defined only at instants of time corresponding to n=0,1,2,3,... , so that the previous input and output sequences which are available are respectively, x(nT) , x(nT- T), x(nT-2T),.., x(0), and {y(nT-mT)} , where m=1,2,3,..n. The most general linear input-output relation must therefore be the difference equation (1.1),

$$y(nT) = \sum_{i=0}^{n} a_i x(nT-iT) - \sum_{i=1}^{n} b_i y(nT-iT) \qquad 1.1$$

where the a_i and b_i are constant coefficients. If all the

b_i are zero, then there is no dependence between the output at time nT and all previous output values; in other words, there is no feedback and the filter is non-recursive and absolutely stable. Otherwise, the filter is recursive and can be made unstable due to the feedback. It should be emphasized that in either case the filter is completely characterized by the a_i and the b_i.

In a second description of the output sequence {y(nT)}, we assume that the digital filter is completely characterized by its discrete impulse response, {h(nT)} instead of the coefficients a_i and b_i. For example, in one realization, the digital equivalent of a first order analog lowpass filter responds to the input sequence {1,0,0,0,...} with an output sequence $\{e^{-nT/\tau}\}$ for n=0,1,2,.. . Thus the impulse response of this filter is $h(nT) = e^{-nT/\tau}$. Suppose now the input is an arbitrary sequence of numbers. Each one of these numbers will cause the filter to generate a scaled-up or scaled-down version of its unit impulse response, with the origin of each output sequence shifted in time by T seconds from the previous one. Since we have stipulated that only linear filters are being considered, then the output is determined by superimposing the effects of each impulse response every T seconds. This relationship between input, output and impulse response sequences is conveniently described by the convolution sum

$$y(nT) = \sum_{i=0}^{n} h(nT-iT)x(iT) \qquad 1.2$$

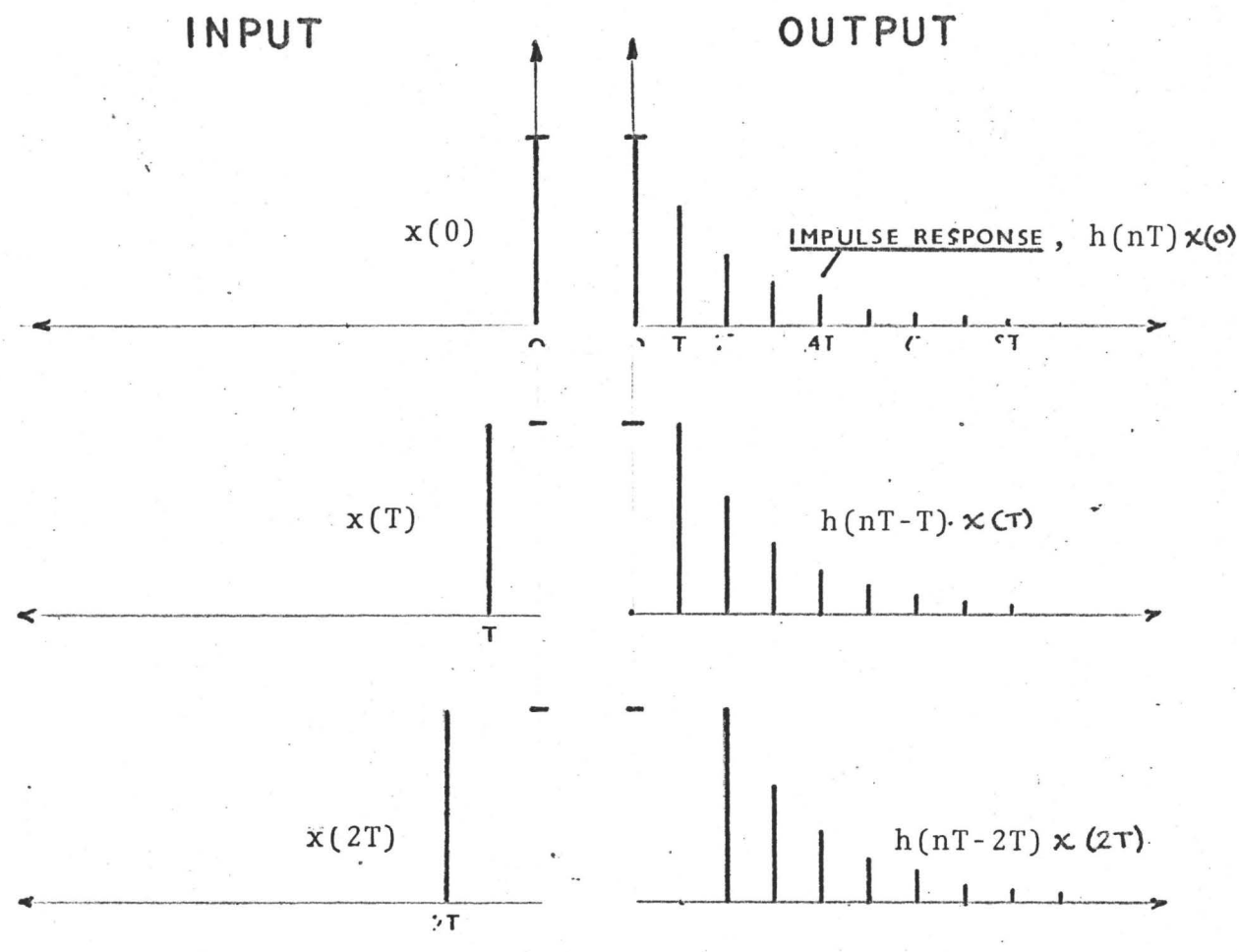
Figure (1.3) illustrates the meaning of this expression for three equal samples at the input.

The frequency domain properties of a linear system whose differential equation is known, are determined by Laplace transform techniques. In a similar way, we can operate on linear difference equations using Z-transform techniques to determine the frequency characteristics of the digital model. Just as the operator s is interpreted as the time differentiator in the continuous time domain, the operator z^{-1} is the unit delay in the discrete time domain. For example, if $X(z^{-1})$ is the Z-transform of a sequence $x(nT)$ then $(z^{-2}+z^{-1}+1)X(z^{-1})$ is directly recognizable as the Z-transform of $x(nT-2T) + x(nT-T) + x(nT)$. We can use this technique to transform Eqn. (1.1) and thereby determine the discrete transfer function $H(z^{-1})$ as follows:

$$\begin{array}{ccc} y(nT) & = \sum_{i=0}^{n} a_i x(nT-iT) & - \sum_{i=1}^{n} b_i y(nT-iT) \\ \downarrow & \downarrow & \downarrow \\ Y(z^{-1}) & = \sum_{i=0}^{n} a_i z^{-i} X(z^{-1}) & - \sum_{i=1}^{n} b_i z^{-i} Y(z^{-1}) \end{array}$$

Since the $Y(z^{-1})$ and $X(z^{-1})$ in the right hand terms are independent of the parameter i, they can be removed from within the summation signs to obtain the transfer function

$$H(z^{-1}) = \frac{Y(z^{-1})}{X(z^{-1})} = \frac{\sum_{i=0}^{n} a_i z^{-i}}{1 + \sum_{i=1}^{n} b_i z^{-i}}$$



Superposition of the Above Responses

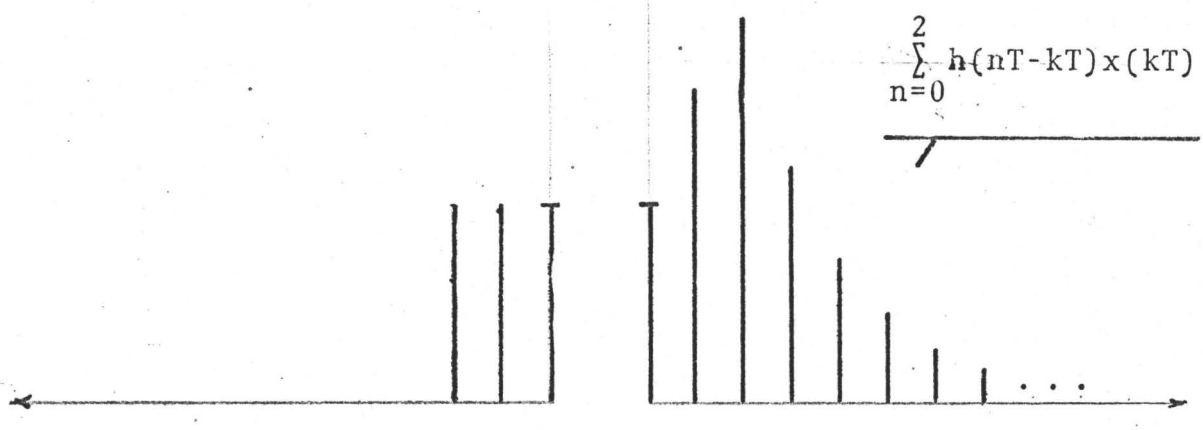


Figure (1.3)

It is apropos at this point to mention that the digital computer's memory makes it a natural device for the solution of difference equations because of the ease of realizing unit delays.

1.4 Review of Established Methods for Realizing Digital Filters

The following example will illustrate how a digital filter can be programmed on a general purpose digital computer. We start with a specific first order, lowpass analog filter described by

$H(s) = \frac{1}{s + 1}$,

or in the time domain by both $\frac{dy}{dt} + y = x$

and $y(t) = e^{t}\int_{0}^{t} e^{-\tau}x(\tau)d\tau$.

This is in fact how most digital filters are designed: an analog filter is first fitted to the required specifications (bandwidth, cutoff frequency, etc) and then a transformation from the s to z^{-1} planes is carried out. The two most commonly referred to are the impulse-invariant and the bilinear transformations. The first simply requires that the impulse response sequence of the digital filter $[H(z^{-1})]$ be identical with the sampled impulse response of the analog filter $[H(s)]$. The bilinear method maps the left half s-plane onto the interior of the unit circle in the z^{-1} plane via the substitution $s = \frac{2(1 - z^{-1})}{T(1 + z^{-1})}$. We shall use both to illustrate the implementation of a digital filter.

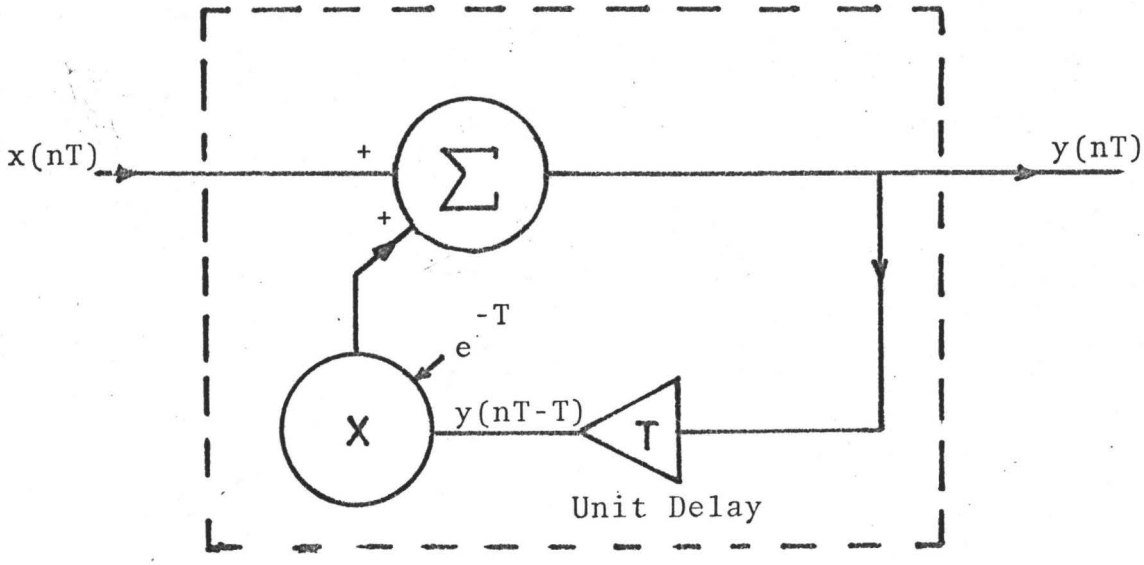
Since the impulse response of $H(s) = \frac{1}{s + 1}$ is $h(t) = e^{-t}$, then the impulse response sequence of the digital filter is constrained to be $\{e^{-nT}\}$, $n=0,1,2,..$ for the impulse invariant transformation. Directly, the Z-transform of this sequence is $\sum_{n=0}^{\infty} e^{-nT} z^{-n} = \frac{1}{1 - e^{-T}z^{-1}} = \frac{Y(z^{-1})}{X(z^{-1})}$ and finally, the difference equation to be incorporated into a computer program is $y(nT) = x(nT) + e^{-T}y(nT-T)$. A block diagram representing the action of this algorithm[Figure (1.4)] and an actual Fortran program follow:

```
      YY=0.0              initial conditions
      DO 10 M=1,500       500 iterations used
      EM=M
      X=SIN(EM*T)         input sequence is a sampled sine
      Y=X+EXP(-T)*YY      the algorithm of the filter
   10 YY=Y                the delay or memory
```

The bilinear transformation involves algebraic replacement of the Laplace variable s with $\frac{2(1 - z^{-1})}{T(1 + z^{-1})}$, so that the above filter transforms to $H(z^{-1}) = \frac{T(1 + z^{-1})}{(T + 2) + (T - 2)z^{-1}}$ and its block diagram is shown in Figure (1.5).

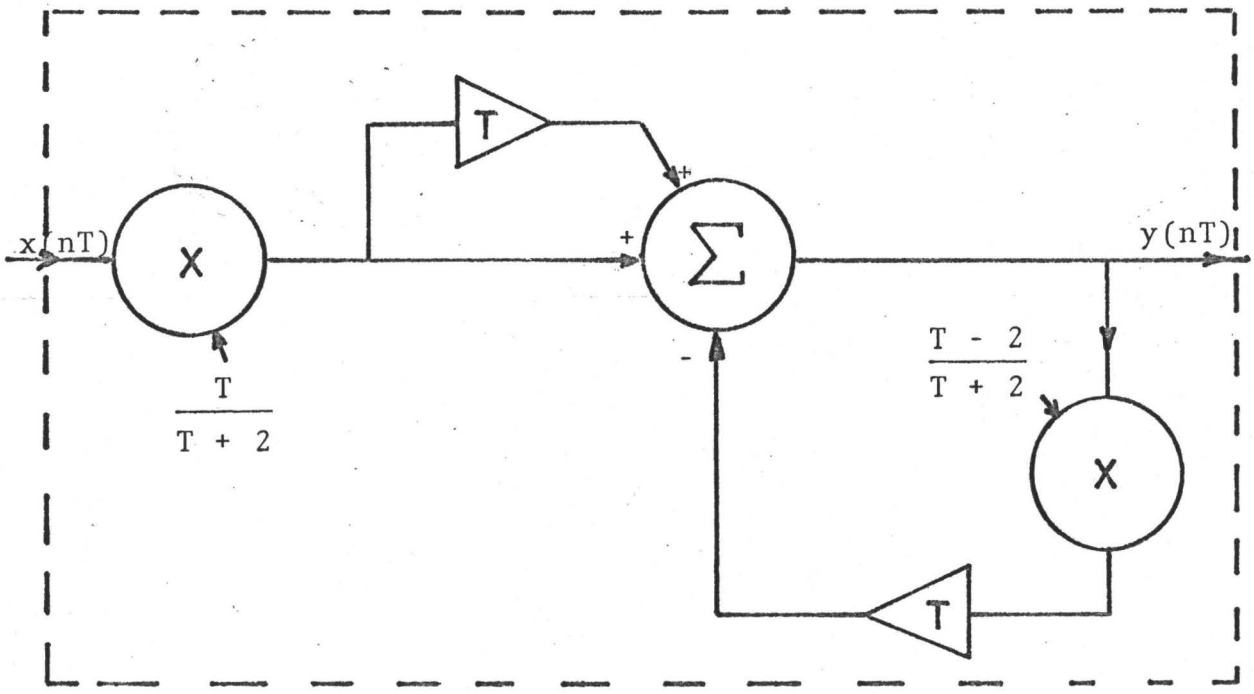
Each of these transformations is limited to the realization of only certain types of digital filters, and even in their useable range care must be taken to assure decent results. Consider first the impulse invariant method. Filters derived with it include an unexpected gain factor not present in the

Figure (1.4)



A Simple Impulse Invariant
Digital Filter

Figure (1.5)



Bilinear Version of Figure (1.4)

original analog filter, $H(s) = \frac{a}{s + a}$. The corresponding discrete transfer function is $H(z^{-1}) = \frac{a}{1 - \exp(-aT)z^{-1}}$

and its dc gain, found by setting $z^{-1} = \exp(-j\omega T)\Big|_{\omega=0} = 1$, is $\frac{a}{1 - \exp(-aT)}$; for the case that $|aT| \ll 1$, this gain factor reduces to $\frac{1}{T}$, the sampling frequency and this is very large in many filters[4]. Beside the above difficulty, a basic limitation of this method is that it is not suitable for realizing highpass or bandstop digital filters. To see why, consider the highpass filter $H(s) = \frac{s}{s + 1}$ which is transformed to

$$H(z^{-1}) = 1 - \frac{1}{1 - \exp(-T)z^{-1}} = \frac{-\exp(-T)z^{-1}}{1 - \exp(-T)z^{-1}}$$

As its pole-zero diagram indicates [Figure (1.6a)] the transfer function lacks a zero near $z^{-1}=1$, i.e. near $\omega=0$; this zero is an essential property for highpass filters. The inability of this method to realize highpass and bandstop filters is explained in more physical terms in Chapter 4.

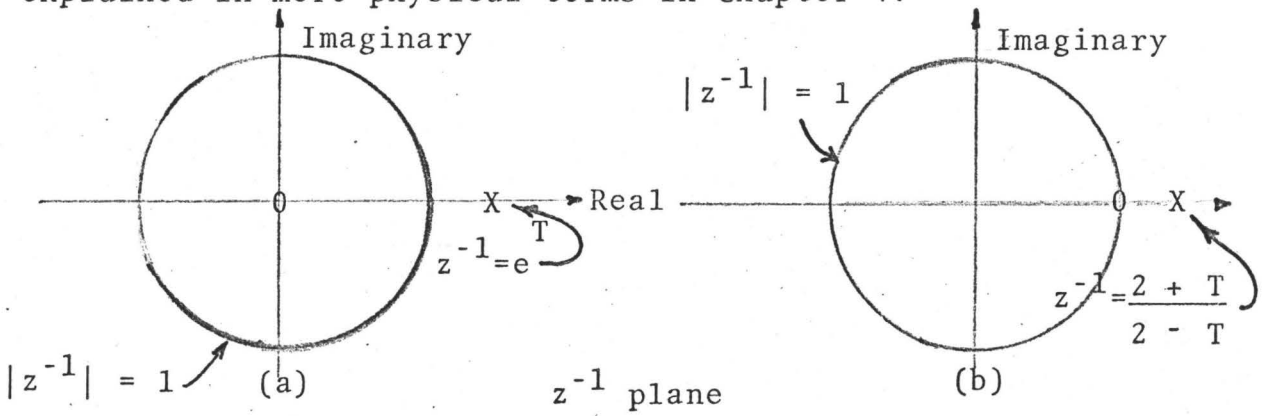


Figure (1.6)

The bilinear transformation overcomes this difficulty by introducing a zero at zero frequency: [see Figure (1.6b)]

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

However, the bilinear transformation which crowds the whole real frequency axis ($0 \leq \omega < \infty$) onto the perimeter of the unit circle in the z^{-1} plane, causes frequency warping[3,4] . This is especially important to compensate for in filters which have significant response near the Nyquist frequency, $\frac{1}{2T}$ hz. Essentially, frequency warping means that the specifications (e.g. centre frequency) of the analog filter, are changed by the transformation to undesired values. As a result, a compensating process called prewarping must be applied in many cases.

1.5 Nature of a Hardware Realization

We can take the physical realization of digital filters a practical step beyond its present form of a Fortran program in a general purpose computer, if we consider building a special purpose computer[5]. The essential functions which are determined by inspection of either Figure (1.4) or the program, are adders, multipliers and unit delays. Now however, we become much more intimately involved in the actual computing process: factors such as the number of bits used in the arithmetic, the sampling rate of the A/D converter, and the

precise interconnection of sub-filters for good economy and performance become very important for a successful realization. A block diagram of the basic filter is shown in Figure (1.7), where the clock synchronizes the operation of the filter.

We have assumed N bits to represent the fractional part of the digital input. Both registers, the adder-subtracter, and the A/D converter are N-bit devices (exclusive of the sign bit) which function in synchronism with a pulse sequence from the clock at the frequency $\frac{1}{T}$ hz. A single clock pulse causes the registers to shift out the N bits in parallel, and after a small delay (<<T sec), to be loaded again from the preceding section. The output, y(nT), thus appears as a collection of N 0's and 1's which are the coefficients of the N bit binary approximation to y(nT).

1.6 Comparison of Digital and Analog Filters

The accelerating interest in digital filters is partly derived from their several distinct advantages over conventional analog filters [4]. Until much higher speed computers are available, the realm of importance for digital filters is below about 1000 hz for real time use. In this range however, analog filters require capacitors and inductors of increasing physical size as the frequency decreases. Furthermore, these L's and C's are subject to variations as they age, and with temperature changes, whereas the coefficients entered into a register remain stable.

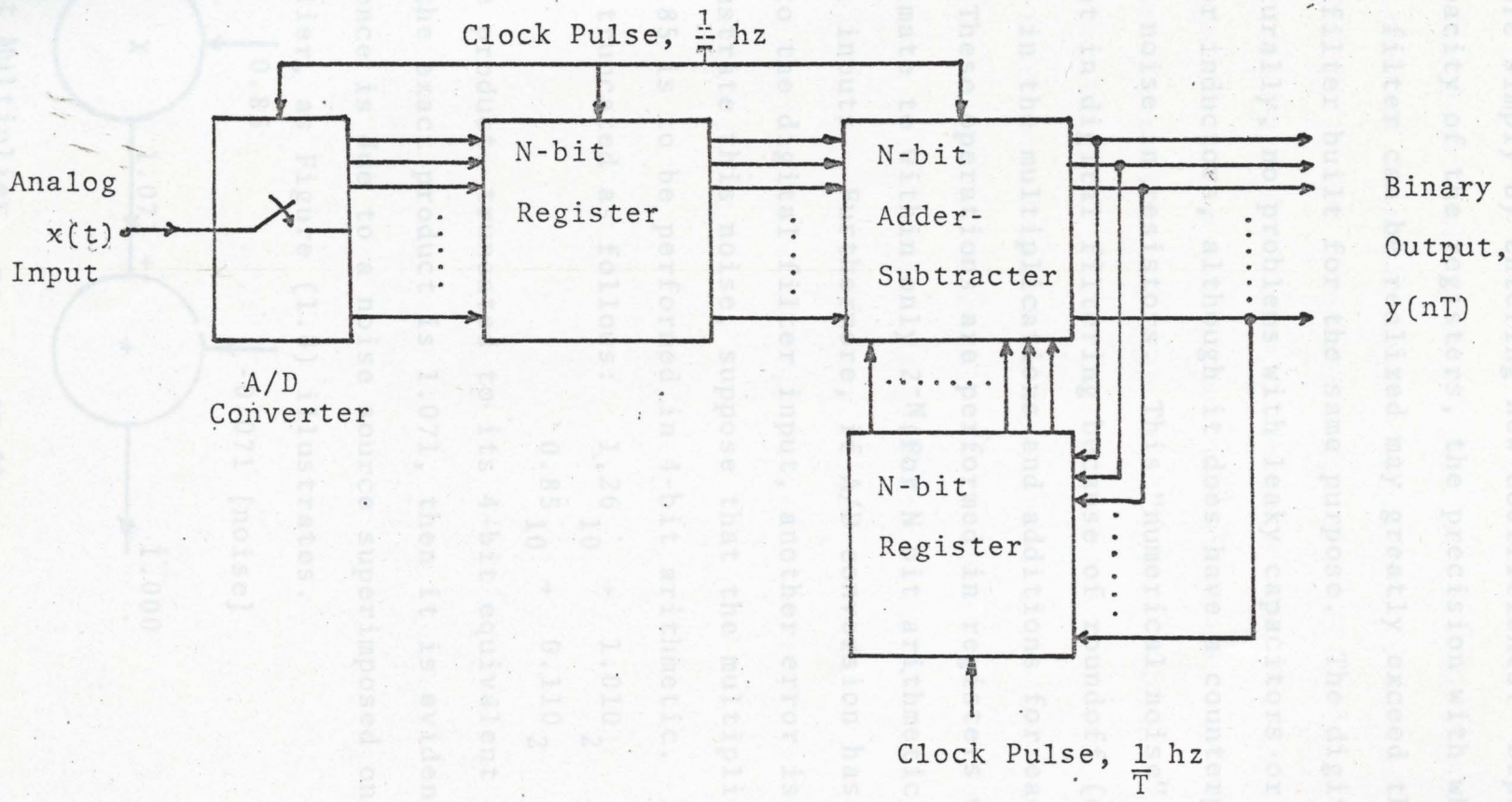
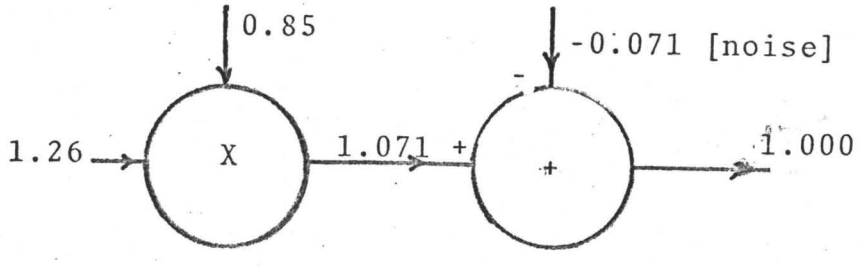

Clock Pulse, $\frac{1}{T}$ hz
Analog
x(t)
Input
A/D
Converter
N-bit
Register
N-bit
Adder-
Subtracter
Binary
Output,
y(nT)
N-bit
Register
Clock Pulse, $\frac{1}{T}$ hz

Figure (1.7)

Conversely, one can intentionally change the filter's structure simply by entering new coefficients. Depending on the capacity of the registers, the precision with which a digital filter can be realized may greatly exceed that of an analog filter built for the same purpose. The digital filter has, naturally, no problems with leaky capacitors or pickup in leads or inductors, although it does have a counterpart to thermal noise in resistors. This "numerical noise" [4] is inherent in digital filtering because of roundoff (or truncation) in the multiplications and additions for each iteration. These operations are performed in registers which can approximate to within only 2^{-N} (for N bit arithmetic) their digital inputs. Furthermore, if A/D conversion has been used prior to the digital filter input, another error is incurred. To illustrate this noise, suppose that the multiplication 1.26x0.85 is to be performed in 4-bit arithmetic. Each number is truncated as follows:

$$1.26_{10} \rightarrow 1.010_2$$
$$0.85_{10} \rightarrow 0.110_2$$

and the product, truncated to its 4-bit equivalent is 1.000. Since the exact product is 1.071, then it is evident that the difference is due to a noise source superimposed on an exact multiplier, as Figure (1.8) illustrates.



Exact Multiplier Figure (1.8)

1.7 Nature of this Thesis

Our work deals with a basic problem in digital filtering---that of determining the coefficients of the difference equation (1.1) for a specific filter. If we know the a_i and b_i then the frequency response of the filter is fixed. Although there are useful methods which do this (Section 1.4) in some circumstances, each has its own limitations. The following work describes the derivation and implementation of another method based on the convolution integral which has its own useful aspects. As we indicated in Section (1.4), the common transformations (impulse invariant and bilinear) are based on quite different criteria, neither of which is related to the basic description of linear systems, the convolution integral.

Chapter 2 Development of the Theory

2.1 Introduction

In this chapter we pursue development of both a cascade and parallel model for an Nth order linear digital filter. A new transformation from the s to z^{-1} plane is derived, and it leads to a flexible method of realizing lowpass, bandpass, highpass and bandstop digital filters, which is readily adapted to programming on a digital computer.

2.2 Direct Formulation

It is a fact in digital filter theory, as in analog filter theory, that a given transfer function can be realized as a physical filter by more than one configuration. These basic structures are the direct, parallel and cascade filters and in digital work, they are by no means equivalent in their performance. Kaiser[1] demonstrates that the direct realization will be the most susceptible to instability as a result of approximating the coefficients a_i and b_i in Equation (1.1) by a finite number of bits in a digital computer. Furthermore, because of the difficult or tedious work required to apply the established transformations (Section 1.4) from the s to z^{-1} planes, the direct form of synthesis is ruled out. Finally, we refer to Appendix I where we demonstrate that the direct realization, derived by the same method which is used for the cascade and parallel structures in this chapter, is cumbersome

and not adaptable to a general computer program.

2.3 Cascade Realization

Consider a linear, time-invariant, lowpass filter of order N, with a transfer function of the form

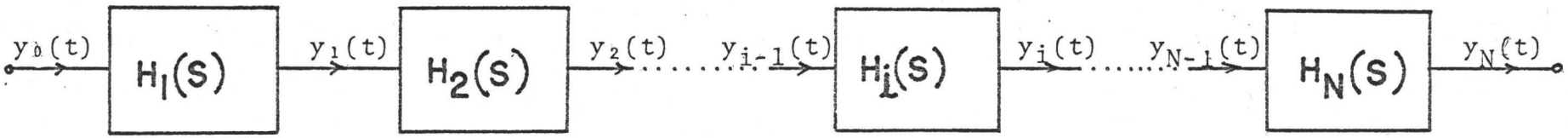
$$H(s) = \frac{K}{s^N + a_{N-1}s^{N-1} + \ldots + a_1 s + a_0} \qquad (2.1)$$

where s is the complex frequency variable, K is a scale factor, and a_0 to a_{N-1} are positive, real constants. Expressing H(s) in its factored form, we have

$$H(s) = K \prod_{i=1}^{N} \frac{1}{s + p_i} = K \prod_{i=1}^{N} H_i(s) \qquad (2.2)$$

where p_i defines the location of the i^{th} pole and the $H_i(s)$ are the elementary first order transfer functions making up H(s). For convenience in analysis, we shall assume that the poles are all simple: they may, however, be complex. As is well known, Eqn (2.2) can be represented by the cascade configuration shown in Figure (2.1). The variables $y_1(t)$ to $y_N(t)$ make up a set of state variables of the filter and except for the input $y_0(t)$ and the output $y(t) = y_N(t)$, the intermediate variables $y_1(t)$ to $y_{N-1}(t)$ are complex in general.

It is clear from Figure (2.1) that we have reduced the filter to a cascade of N first-order sections, each of which has



General Cascade Structure

Figure (2.1)

a transfer function of the form

$$H_i(s) = \frac{Y_i(s)}{Y_{i-1}(s)} = \frac{1}{s + p_i} \tag{2.3}$$

Assuming the network is initially relaxed, we may apply the convolution integral to express the response of the i^{th} section in the time domain as follows:

$$y_i(t) = e^{-p_i t} \int_0^t e^{p_i \tau} y_{i-1}(\tau) d\tau \tag{2.4}$$

Now, since "time" on the digital computer is necessarily discrete, in Equation (2.4) we may successively set t = nT and t = nT-T (with T denoting the sampling period, and n an integer) to obtain:

$$y_i(nT) = e^{-p_i nT} \int_0^{nT} e^{p_i \tau} y_{i-1}(\tau) d\tau \tag{2.5}$$

and

$$y_i(nT-T) = e^{-p_i(nT-T)} \int_0^{nT-T} e^{p_i \tau} y_{i-1}(\tau) d\tau \tag{2.6}$$

These two equations can be combined by multiplying (2.6) by $e^{-p_i T}$, and then subtracting the result from (2.5). The result is

$$y_i(nT) = e^{-p_i T} y_i(nT-T) + e^{-p_i nT} \int_{nT-T}^{nT} e^{p_i \tau} y_{i-1}(\tau) d\tau \tag{2.7}$$

The exact evaluation of the integral requires an analytic expression for the input $y_{i-1}(\tau)$ over the interval $nT-T \leq \tau < nT$; however, we can proceed to an approximate result by assuming a Taylor expansion of the input over each interval T. In the following derivations we have used the simplest expansion, namely that the input $y_{i-1}(\tau)$ of the i^{th} section remains constant over $nT-T \leq \tau < nT$. The remaining integration is simple and leads to a final difference equation:

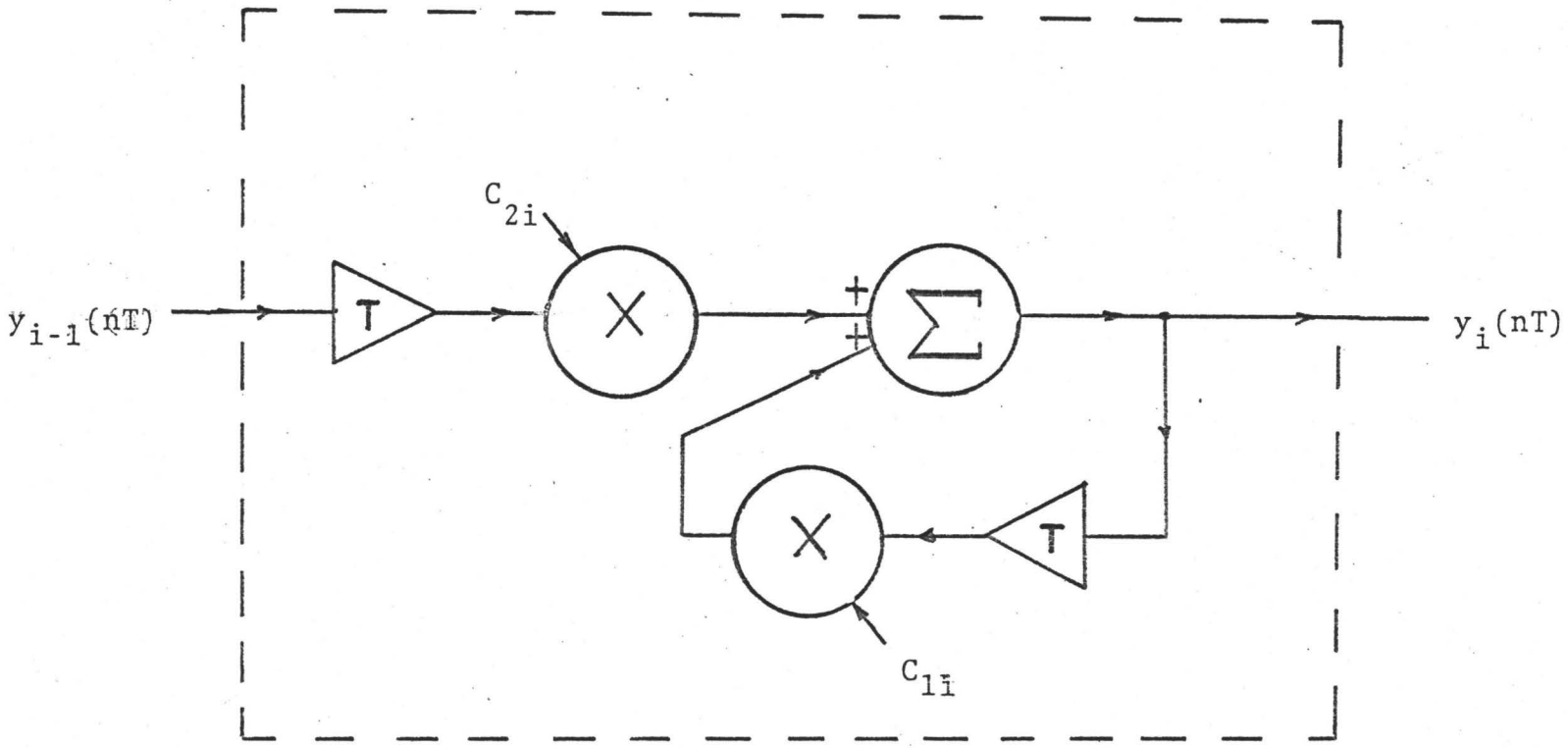
$$y_i(nT) = e^{-p_i T}\, y_i(nT-T) + \frac{1}{p_i}(1 - e^{-p_i T})\, y_{i-1}(nT-T) \quad (2.8a)$$

$$\triangleq [C_{1i}]\, y_i(nT-T) + [C_{2i}]\, y_{i-1}(nT-T) \quad (2.8b)$$

Equation (2.8b) corresponds to the signal flow diagram shown in Figure (2.2).

The cascade realization of the digital filter in the time domain consists of the tandem connection of N sections of which the i^{th} is defined by Equation (2.8a). Thus, for the complete digital filter we may write the following set of difference equations:

$$\begin{aligned}
y_1(nT) &= e^{-p_1 T}\, y_1(nT-T) + \frac{1}{p_1}(1 - e^{-p_1 T})\, y_o(nT-T) \\
y_2(nT) &= e^{-p_2 T}\, y_2(nT-T) + \frac{1}{p_2}(1 - e^{-p_2 T})\, y_1(nT-T) \\
&\vdots \\
y_N(nT) &= e^{-p_N T}\, y_N(nT-T) + \frac{1}{p_N}(1 - e^{-p_N T})\, y_{N-1}(nT-T)
\end{aligned} \quad (2.9)$$



Convolution-Approximation Version of

the Basic Lowpass Digital Filter. Figure (2.2)

The output $y_N(nT)$ of the N^{th} section constitutes the output of the complete filter. Equations (2.9) correspond to the signal flow diagram of Figure (2.3), where the C_{1i} and C_{2i} are defined in Equation (2.8b).

2.4 Discrete Transfer Function of the Cascade Lowpass Filter

From the difference equation (2.8a), we find directly that the discrete transfer function $H_i(z^{-1})$ corresponding to $H_i(s)$ of Equation (2.3) is

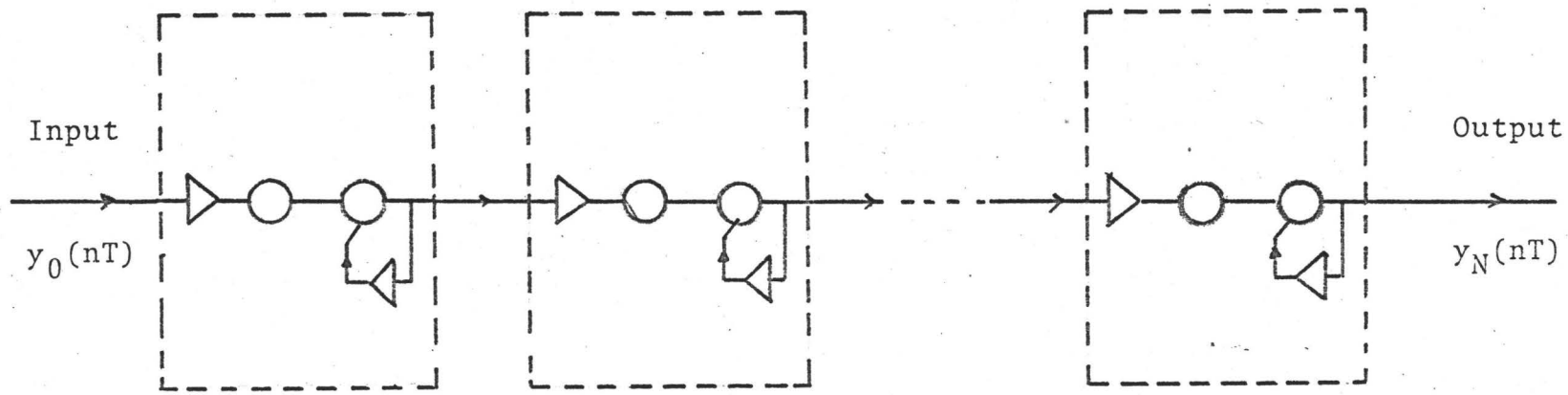
$$H_i(z^{-1}) = \frac{Y_i(z^{-1})}{Y_{i-1}(z^{-1})} = \frac{1}{p_i}(1 - e^{-p_iT})\frac{z^{-1}}{1 - e^{-p_iT}z^{-1}} \qquad (2.10)$$

where $z^{-1} = e^{-sT}$. The sequence of steps leading to the formulation of the discrete transfer function from the continuous transfer function is presented concisely in Figure (2.4)

Using Equation (2.10) along with Equation (2.2), we can express the discrete transfer function of the cascade form of the digital filter as

$$H(z^{-1}) = K \prod_{i=1}^{N} \frac{1}{p_i}(1 - e^{-p_iT}) \frac{z^{-1}}{1 - e^{-p_iT}z^{-1}}$$

$$= K \prod_{i=1}^{N} \left(\frac{C_{2i}\, z^{-1}}{1 - C_{1i}z^{-1}} \right) \qquad (2.11)$$



Nth Order Lowpass Digital Filter

Figure (2.3)

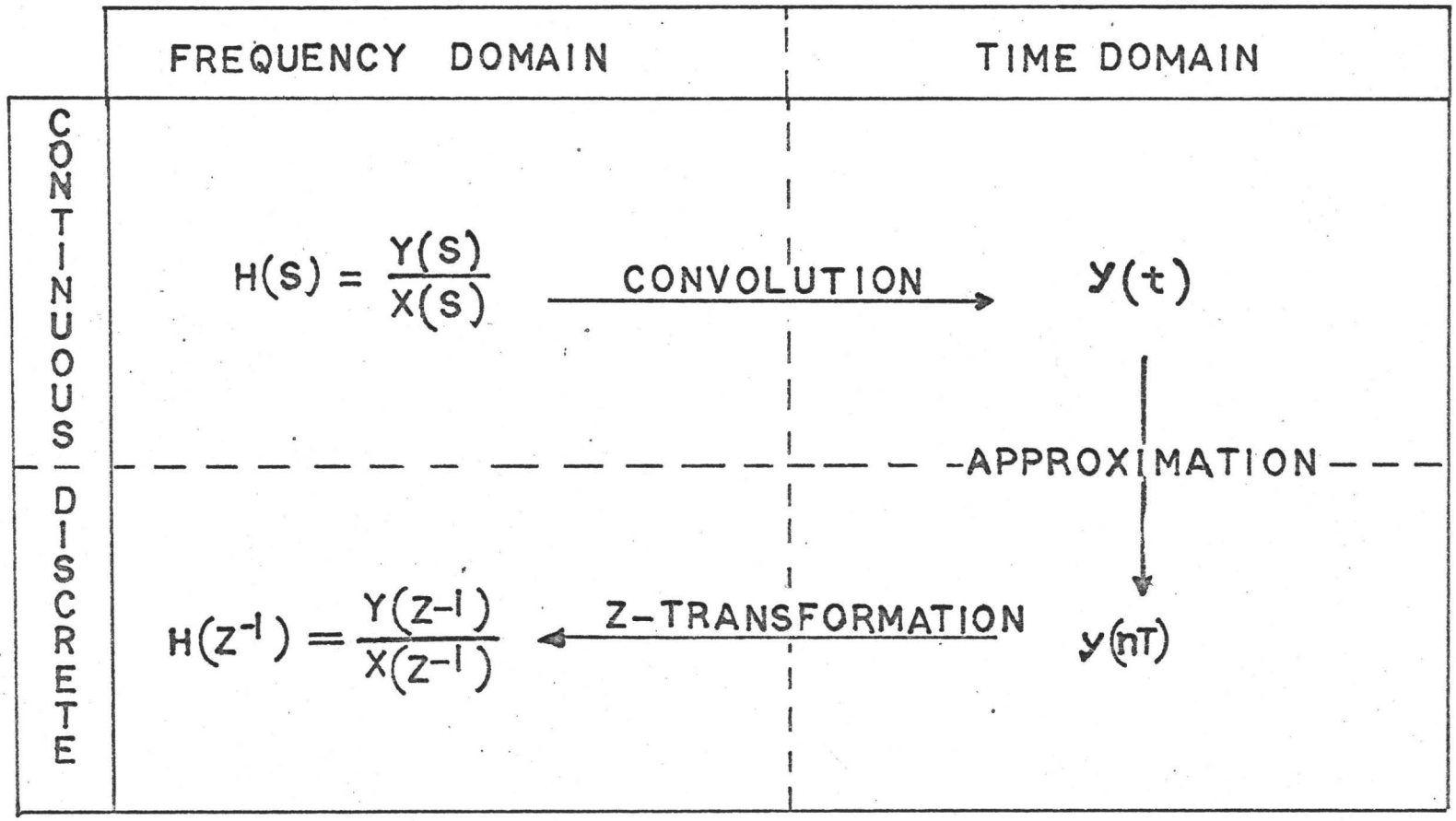

FREQUENCY DOMAIN
TIME DOMAIN
CONTINUOUS
DISCRETE
$H(S) = \frac{Y(S)}{X(S)}$
CONVOLUTION
$y(t)$
APPROXIMATION
$H(Z^{-1}) = \frac{Y(Z^{-1})}{X(Z^{-1})}$
Z-TRANSFORMATION
$y(nT)$

Figure (2.4)

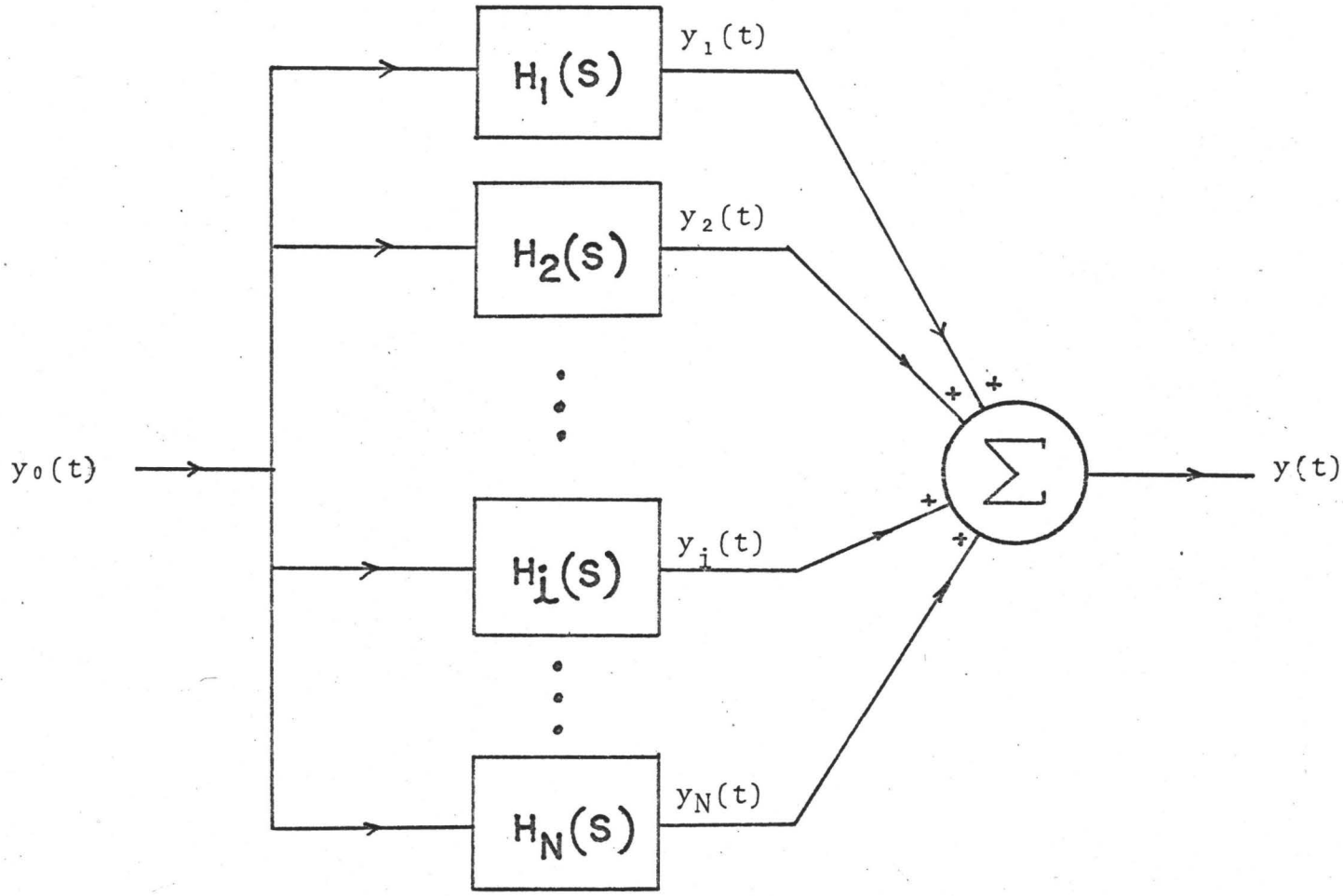
Equation (2.11) allows us to determine the frequency response of the filter simply by setting $z^{-1} = e^{-j\omega T}$ and letting ω vary. Since z^{-1} is a periodic function of ωT then $H(z^{-1})$ is also periodic. This is an important departure from the behaviour of the original H(s) which we will illustrate and discuss in Chapters 3 and 4.

2.5 Parallel Realization

Another useful realization of a digital filter in the time domain is obtained by expanding H(s) of Equation (2.1) into partial fractions; thus we may write

$$H(s) = \sum_{i=1}^{N} \frac{k_i}{s + p_i} \qquad (2.12)$$

where the residues, k_i, are complex constants in general. Equation (2.12) corresponds to the parallel configuration shown in Figure (2.5) where the $y_i(t)$, representing another set of state variables, are also complex, but their sum, the output y(t), is real. Here again we see that the filter has been reduced to the parallel connection of N complex first order sections, the i^{th} one having a transfer function of the form $k_i/(s + p_i)$. Thus, proceeding in a manner similar to that described in Section 2.3, we find that the parallel form of digital filter is defined by the set of



General Parallel Structure

Figure (2.5)

difference equations:

$$
\begin{aligned}
y_1(nT) &= e^{-p_1T}\, y_1(nT-T) + \frac{k_1}{p_1}(1 - e^{-p_1T})y_0(nT-T) \\
y_2(nT) &= e^{-p_2T}\, y_2(nT-T) + \frac{k_2}{p_2}(1 - e^{-p_2T})y_0(nT-T) \\
&\vdots \\
y_N(nT) &= e^{-p_NT}\, y_N(nT-T) + \frac{k_N}{p_N}(1 - e^{-p_NT})y_0(nT-T)
\end{aligned}
\tag{2.13}
$$

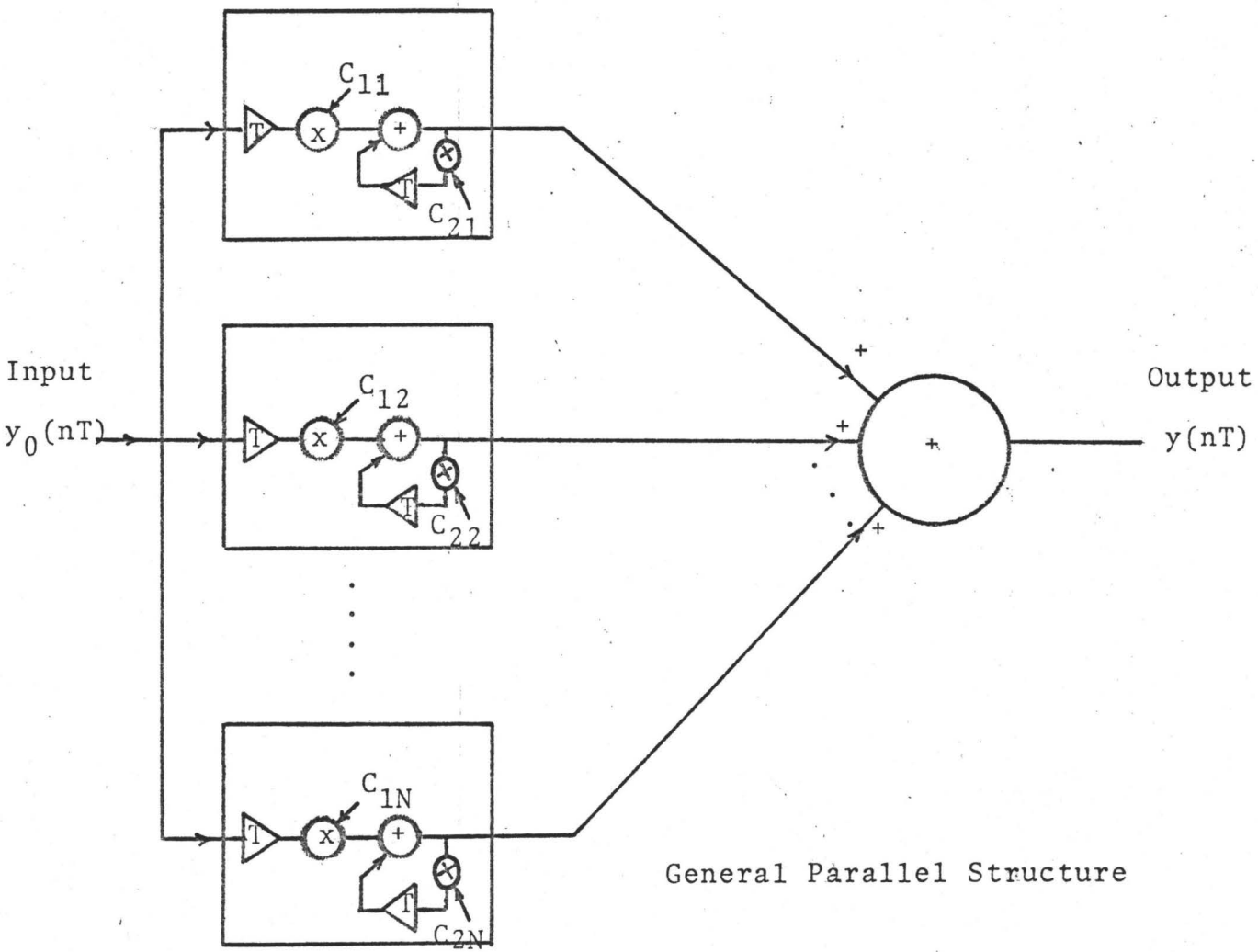
For the output of the whole filter, we have

$$
y(nT) = \sum_{i=1}^{N} y_i(nT) \tag{2.14}
$$

Equations (2.13) and (2.14) correspond to the signal flow diagram of Figure (2.6).

Directly from the above two sets of equations, we obtain the parallel form of the discrete transfer function:

$$
\begin{aligned}
H(z^{-1}) &= \sum_{i=1}^{N} \frac{k_i}{p_i}(1 - e^{-p_iT})\frac{z^{-1}}{1 - e^{-p_iT}z^{-1}} \\
&\triangleq \sum_{i=1}^{N} \frac{[C_{2i}]z^{-1}}{1 - [C_{1i}]z^{-1}}
\end{aligned}
\tag{2.15}
$$



General Parallel Structure

Figure (2.6)

2.6 Extensions of the Convolution Approximation Method

As evidenced by Equations (2.9) and (2.13), the structure of an elementary lowpass digital filter is basically the same whether it is intended for cascade or parallel realizations. In the following extensions of the convolution-approximation (C-A) method , this elementary lowpass section is transformed into elementary bandpass, highpass and bandstop sections which likewise can be used in either cascade or parallel configurations.

(a) Bandpass

Given the lowpass transfer function, the corresponding bandpass transfer function is obtained by replacing s with $(s^2 + \omega_0^2)/Bs$, where ω_0 and B are its centre frequency and bandwidth respectively. The transfer function of the simple lowpass section, given by Equation (2.3) thus modifies to

$$H_{i,bp}(s) = \frac{Bs}{s^2 + Bp_i s + \omega_0^2} \qquad (2.16)$$

which is readily reduced to a difference of two simple lowpass filters by a partial fraction expansion:

$$H_{i,bp}(s) = \frac{r_i'}{s + p_i'} - \frac{r_i''}{s + p_i''} \qquad (2.17)$$

where

$$p_i' \ , \ p_i'' = \frac{Bp_i}{2} \pm \sqrt{\left(\frac{Bp_i}{2}\right)^2 - \omega_0^2}$$

$$r_i' = \frac{Bp_i'}{p_i' - p_i''} \qquad \text{and} \qquad r_i'' = \frac{Bp_i''}{p_i' - p_i''}$$

It thus follows that the discrete transfer function corresponding to the $H_{i,bp}(s)$ of Equation (2.16) is:

$$H_{i,bp}(z^{-1}) = \frac{r_i'}{p_i'}(1 - e^{-p_i'T})\frac{z^{-1}}{1 - e^{-p_i'T} z^{-1}}$$

$$- \frac{r_i''}{p_i''}(1 - e^{-p_i''T})\frac{z^{-1}}{1 - e^{-p_i''T} z^{-1}} \qquad (2.17a)$$

$$= \frac{[C_{2i}'] \ z^{-1}}{1 - [C_{1i}'] \ z^{-1}} - \frac{[C_{2i}''] \ z^{-1}}{1 - [C_{1i}''] \ z^{-1}} \ .$$

The difference equations pertaining to a single bandpass section (the i^{th}) are therefore (using the cascade configuration):

$$y_i'(nT) = e^{-p_i'T}\, y_i'(nT-T) + \frac{r_i'}{p_i'}(1 - e^{-p_i'T})\, y_{i-1}(nT-T)$$

$$y_i''(nT) = e^{-p_i''T}\, y_i''(nT-T) + \frac{r_i''}{p_i''}(1 - e^{-p_i''T})\, y_{i-1}(nT-T)$$

$$y_i(nT) = y_i'(nT) - y_i''(nT) \qquad (2.18)$$

The corresponding signal flow diagram is shown in Figure (2.7). Such blocks are connected in cascade to form the complete filter.

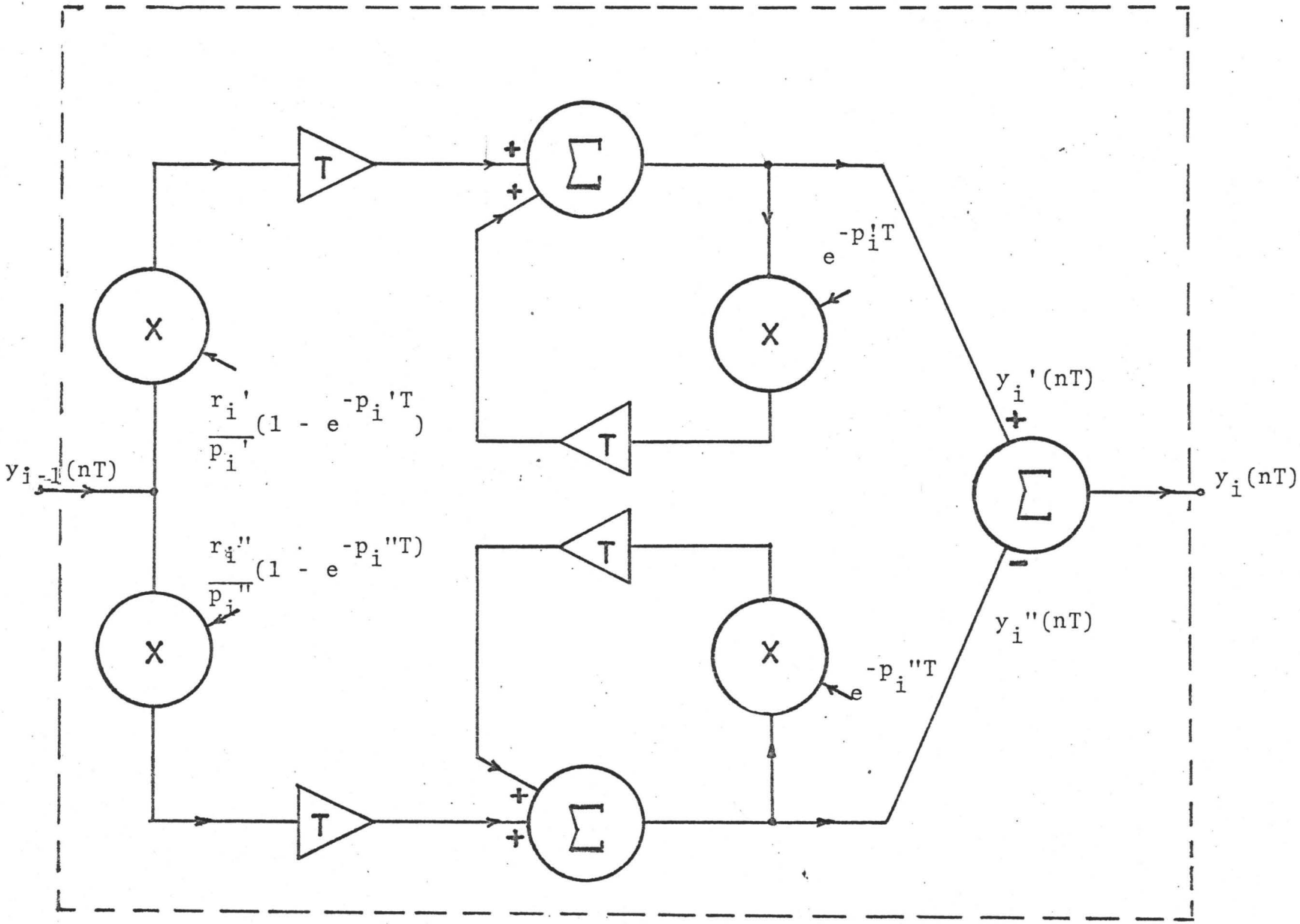
(b) Highpass

The lowpass to highpass transformation requires replacing s in the lowpass transfer function with ω_0/s, where ω_0 is the the desired cutoff frequency of the highpass filter. Thus, Equation (2.3) becomes

$$H_{i,hp}(s) = \frac{s}{p_i s + \omega_0} = \frac{1}{p_i}\left(\frac{s}{s + \omega_0/p_i}\right) \qquad (2.19)$$

This expression can be separated into a "direct link" and a parallel lowpass filter, by division:

$$H_{i,hp}(s) = \frac{1}{p_i} - \frac{\omega_0/p_i^2}{s + \omega_0/p_i} \triangleq \frac{Y_i'(s) - Y_i''(s)}{Y_{i-1}(s)}$$



Bandpass Filter Section

Figure (2.7)

Figure (2.8a) illustrates the direct link version of an analog highpass filter.

It is clear that at any instant of time t, the input $y_{i-1}(t)$ has an immediate influence on both channel outputs, $y_i'(t)$ and $y_i''(t)$. However, in moving from the s to z^{-1} domain, time is no longer continuous so that an input signal applied at time nT cannot affect the output of the lower channel in Figure (2.8b) until some finite time later. On the other hand, in the case of the direct link from input to output, the effect is still immediately observable at the output. It is evident therefore that if we transform the lowpass link into an equivalent discrete transfer function and leave the constant term as it is, the outputs of the direct and lowpass paths will not be in the correct phase, and the filter response will therefore deteriorate.

This problem may be overcome by introducing a compensating delay in the direct link. The size of this delay was first deduced from the experimental curves of Figure (2.9). These are the frequency response curves for an arbitrarily chosen tenth order bandstop filter with zero delay and unit delay(T) in the direct link. Such a precise reversal of form strongly suggests that the optimum delay lies very near the middle of the range [0,T] seconds; in fact, the experimental results which follow vindicate the choice of T/2 seconds.

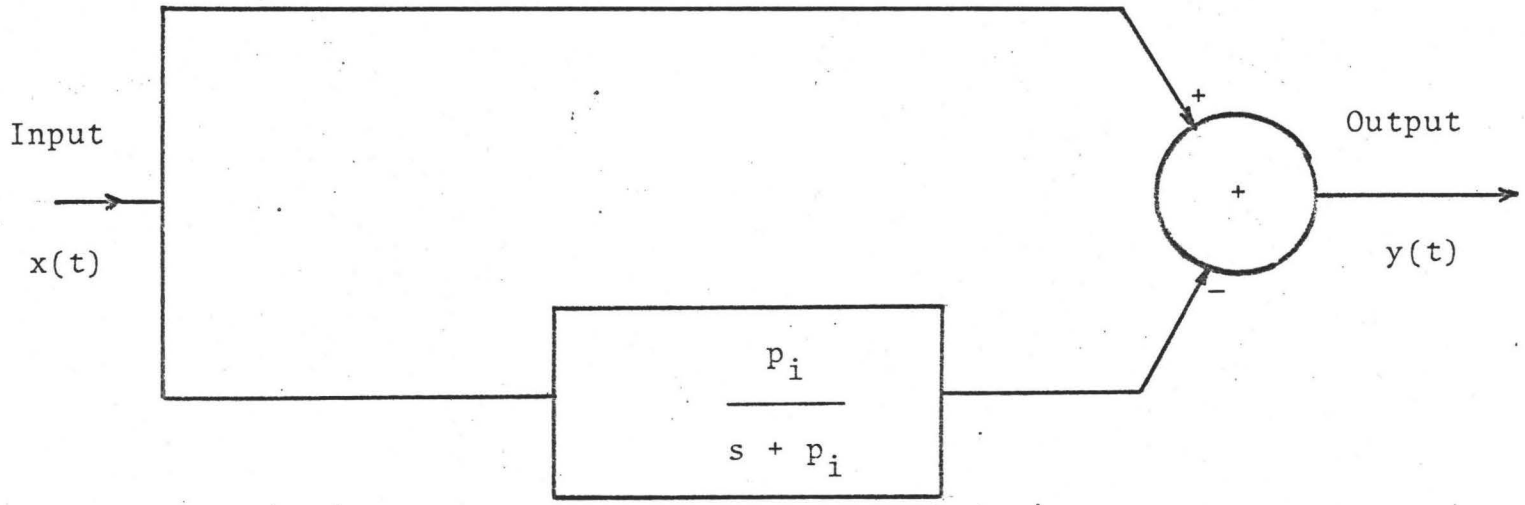


Figure (2.8)

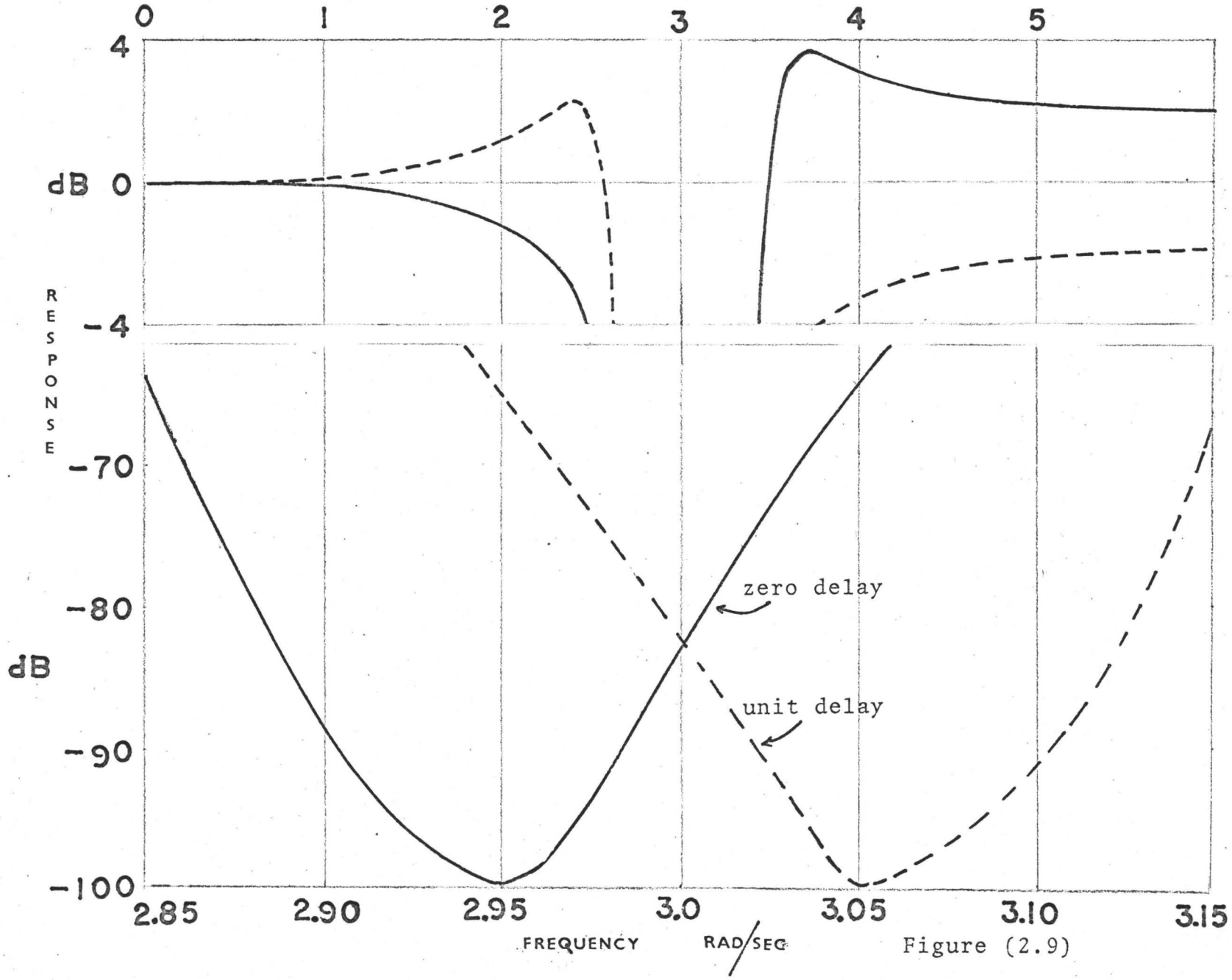

0
1
2
3
4
5
4
dB 0
-4
RESPONSE
-70
-80
dB
-90
-100
2.85
2.90
2.95
3.0
3.05
3.10
3.15
FREQUENCY
RAD/SEC
zero delay
unit delay

Figure (2.9)

The same conclusion can be derived theoretically by determining the steady-state delay across the lowpass function which is in parallel with the direct link. The discrete transfer function for the lowpass section is

$$H(z^{-1}) = \frac{[C_{2i}]z^{-1}}{1 - [C_{1i}]z^{-1}}, \quad \text{where } C_{1i} = e^{-p_i T},$$

and if $|p_i T| \ll 1$, [which can readily be arranged by adjusting the sampling rate], then $C_{1i} \simeq 1$ and $H(z^{-1}) \simeq [C_{2i}]\frac{z^{-1}}{1 - z^{-1}}$.

The complex constant C_{2i} has a constant phase angle (β say), so that the transfer function becomes

$$H(e^{-j\omega T}) = \frac{C_{2i}\, e^{-j[\omega T - \beta]}}{1 - e^{-j\omega T}}.$$

This quantity has a phase angle of $\phi = -[\omega T - \beta] + \tan^{-1}[\cot(\omega T/2)]$

and therefore the delay is $-\frac{\partial \phi}{\partial \omega} = T/2$ sec.
This result is valid for all transfer functions of the form of $H(z^{-1})$ above when β is independent of frequency, and where $|p_i T| \ll 1$. Such a delay (T/2) corresponds to " $z^{-1/2}$ ". However, for $H(z^{-1})$ to be realizable as a linear digital filter, it must be rational in z^{-1}. It follows that a further transformation, $s \to 2s$, is necessary; then, z^{-1} (which

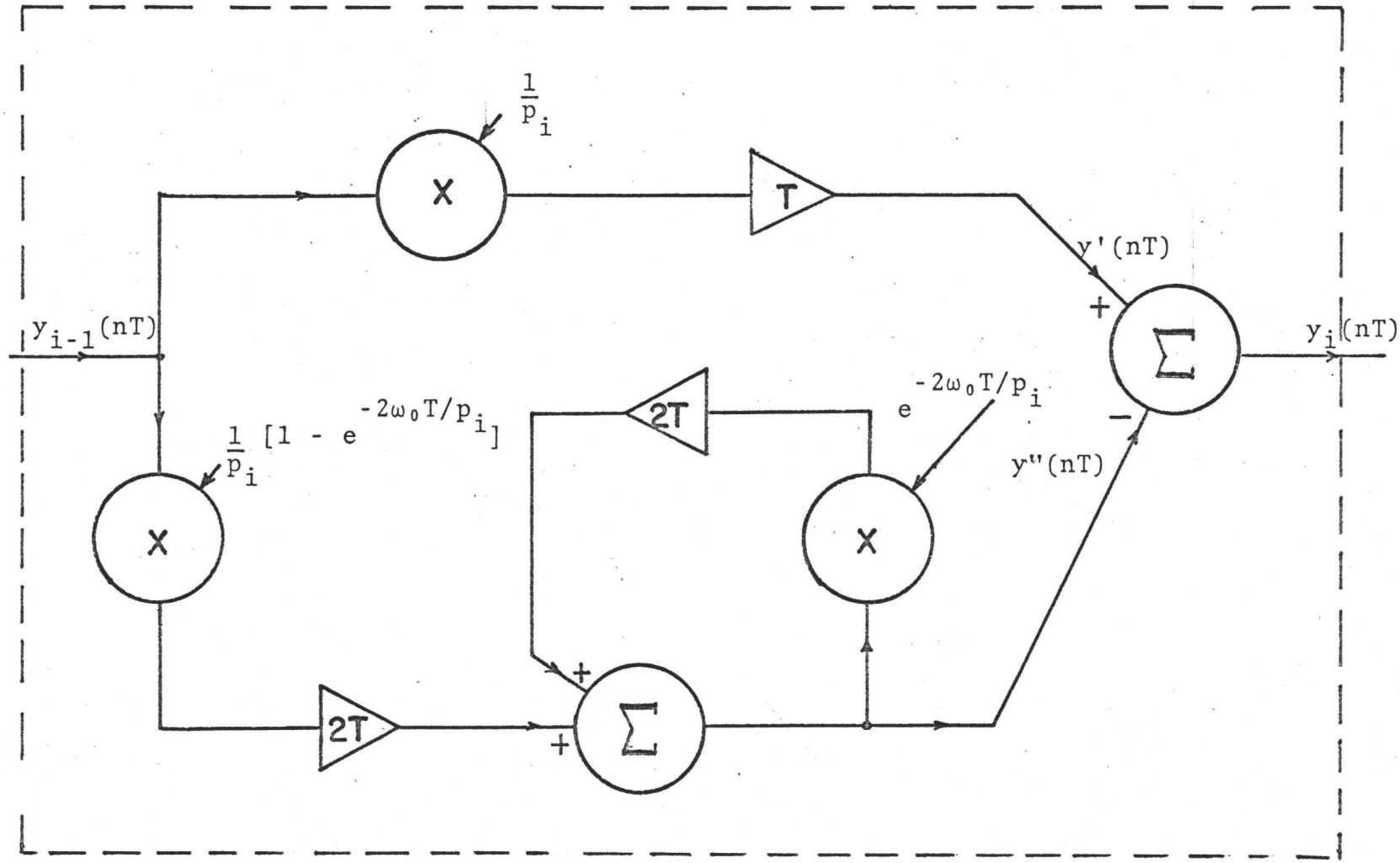
equals e^{-sT}) is transformed into z^{-2}, and $z^{-1/2}$ into z^{-1}. Of course now, the cutoff frequency, centre frequency and bandwidth must also be doubled before proceeding with the synthesis if the desired values are to be achieved. Thus corresponding to the highpass transfer function of Equation (2.18), we obtain the discrete counterpart:

$$H_{i,hp}(z^{-1}) = \frac{1}{p_i} z^{-1} - \left\{ \frac{1}{p_i} \left(1 - e^{-2\omega_0 T/p_i}\right) \right\} \times \left(\frac{z^{-2}}{1 - e^{-2\omega_0 T/p_i} z^{-2}} \right) \qquad (2.20)$$

Although this transformation introduces no new poles or zeroes into the analog transfer function, it does introduce both new poles and new zeroes into the discrete transfer function. Especially noteworthy is the new pole introduced near the Nyquist frequency (1/2T hz) as indicated in Figure (2.9). It is evident that the transformed highpass filter will not be useable at frequencies near 1/2T hz because of the gain introduced by the nearby pole. This effect will be illustrated in Chapter 3. Finally, if Equation (2.20) is rewritten as

$$H_{i,hp}(z^{-1}) = \frac{z^{-1}}{p_i} - \frac{[C_{2i}]z^{-2}}{1 - [C_{1i}]z^{-2}}$$

then Figure (2.10) shows its signal flow diagram.



Convolution-Approximation Version of a Basic Highpass Digital Filter; Figure (2.10)

We can write down the basic highpass difference equations directly from Equation (2.20) to match Figure (2.10) :

$$y'(nT) = \frac{1}{p_i} y_{i-1}(nT-T)$$

$$y''(nT) = \frac{1}{p_i}\left[1 - e^{-2\omega_0 T/p_i}\right] y_{i-1}(nT-2T) + e^{-2\omega_0 T/p_i} y''(nT-2T)$$

$$y_i(nT) = y'(nT) - y''(nT) \qquad (2.21)$$

(c) Bandstop

The lowpass to bandstop transformation requires replacing s in the lowpass transfer function by $Bs/(s^2 + \omega_0^2)$. Thus Equation (2.3) becomes, after partial fraction expansion,

$$H_{i,bs}(s) = \frac{1}{p_i}\left(1 - \frac{r_i'}{s + p_i'} + \frac{r_i''}{s + p_i''}\right) \qquad (2.22)$$

where

$$p_i', p_i'' = \frac{B}{2p_i} \pm \sqrt{\left(\frac{B}{2p_i}\right)^2 - \omega_0^2}$$

$$r_i' = \frac{B}{p_i}\left(\frac{p_i'}{p_i' - p_i''}\right) \quad \text{and} \quad r_i'' = \frac{B}{p_i}\left(\frac{p_i''}{p_i' - p_i''}\right)$$

To transform Equation (2.22) into a discrete transfer function, we proceed in the same way as was done for the highpass case. Here again, the centre frequency and bandwidth of the filter must be doubled before proceeding with the synthesis. Also, as a result of the transformation, new poles and zeroes are introduced into the z^{-1} plane around the Nyquist frequency, and again the filter will not be useable in this region because of high gains. The discrete transfer function for a cascade bandstop filter is:

$$H_{i,bs}(z^{-1}) = \frac{1}{p_i} z^{-1} - \frac{r_i' [1 - e^{-p_i'T}] z^{-2}}{p_i p_i' [1 - e^{-p_i'T} z^{-2}]} + \frac{r_i'' [1 - e^{-p_i''T}] z^{-2}}{p_i p_i'' [1 - e^{-p_i''T} z^{-2}]} \qquad (2.23)$$

$$\triangleq \frac{1}{p_i} z^{-1} - \frac{C_{2i}' z^{-2}}{1 - C_{1i}' z^{-2}} + \frac{C_{2i}'' z^{-2}}{1 - C_{2i}'' z^{-2}}$$

where

$$C_{1i}' = e^{-p_i'T} \quad \text{and} \quad C_{2i}' = \frac{r_i' [1 - e^{-p_i'T}]}{p_i p_i'} \quad \text{etc.}$$

The difference equations for the basic bandstop digital filter can be written down directly from the reduced form of Equation (2.23):

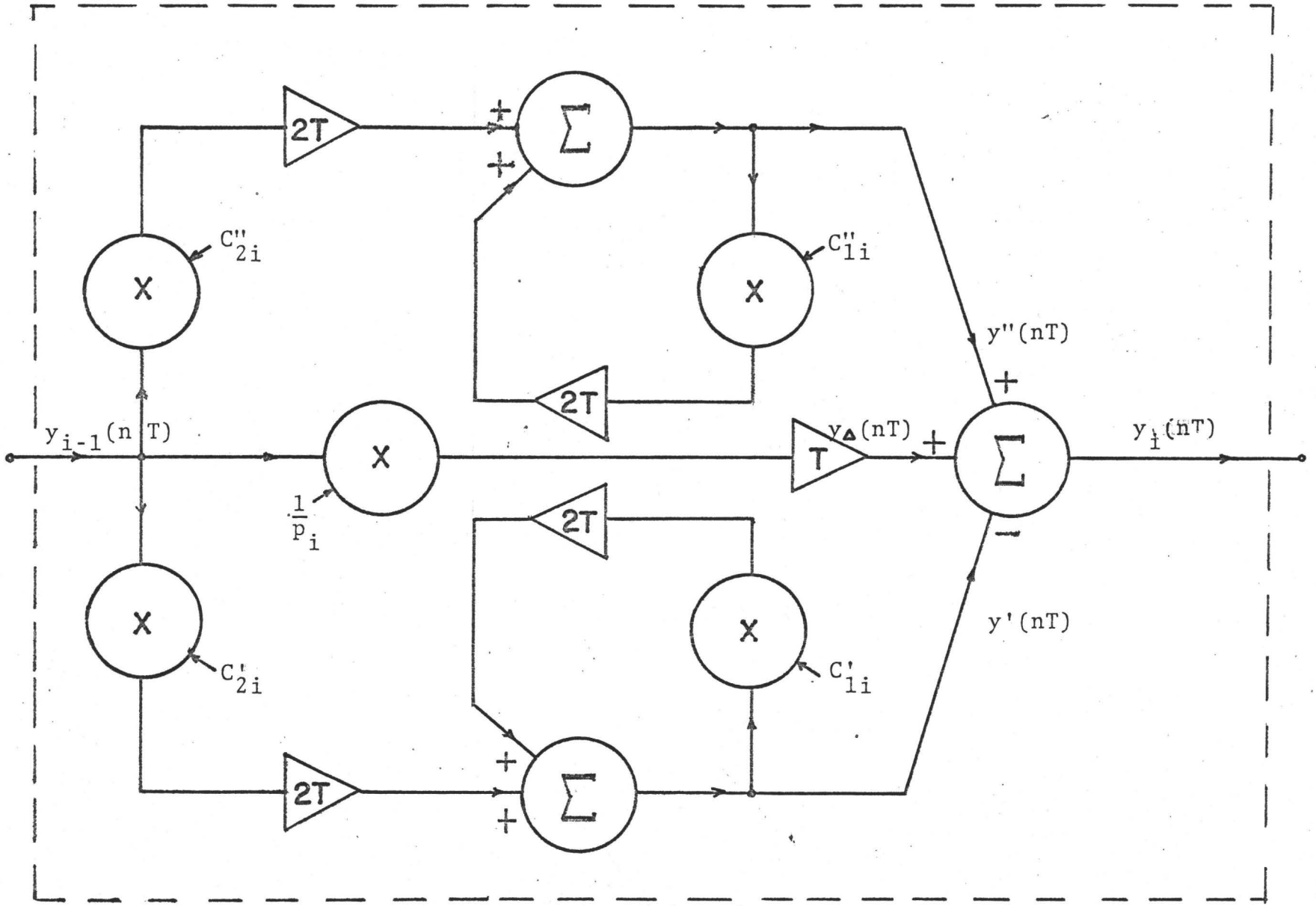
$$\left.\begin{aligned} y_{\Delta}(nT) &= \frac{1}{p_i}\, y_{i-1}(nT-T) \\ y'(nT) &= C'_{2i}\, y_{i-1}(nT-2T) + C'_{1i}\, y'(nT-2T) \\ y''(nT) &= C''_{2i}\, y_{i-1}(nT-2T) + C''_{1i}\, y''(nT-2T) \\ y_i(nT) &= y_{\Delta}(nT) - y'(nT) + y''(nT) \end{aligned}\right\} \quad (2.24)$$

Figure (2.11) shows the relationship of the various parameters.

2.7 Summary

In this chapter we have developed a basic transformation from the analog to digital domains, given by Equations (2.3) and (2.10) for the lowpass case. By transforming the factor $1/(s + p_i)$ to a bandpass factor and then reducing this to a combination of lowpass terms, the original transformation can be used again, to obtain Equation (2.17a).

Further, in the case of highpass or bandstop filters, we have determined that a second frequency transformation is necessary [$s \rightarrow 2s$] before we can realize the discrete



Convolution-Approximation Version of a Basic Bandstop Digital Filter; Figure (2.11)

transfer function as differende equations. In this case, the lowpass terms are transformed as follows:

$$\frac{1}{s + p_i} = \frac{\frac{1}{p_i}(1 - e^{-p_i T})z^{-2}}{1 - e^{-p_i T} z^{-2}}$$

and a direct link is transformed to z^{-1} , the unit delay.

Finally, it is important that the approximations involved in the theory be listed so that the limitations of the method are clear:

(1) For low sampling rates, the approximation of the convolution integral (Equation (2.4)) may produce an insufficiently accurate filter. In this event, either the sampling rate or the order of the approximation of $x(\tau)$ should be increased. In this work, we have used only the zero order approximation since it gave good results and led to the simplest algorithms.

(2) The method was extended to highpass and bandstop filters by introducing a delay of T/2 sec in the direct link. Only when the coefficient $e^{-p_i T}$ can be well approximated by unity is this the correct delay [Section 2.6] . In the examples done here, the magnitude of the exponent was no greater than 0.1, so that the coefficient was less than 0.90.

Chapter 3 Experimental Results

3.1 Introduction

The theory developed in the previous chapter was written into the Fortran programs of Appendix II. Those programs are the basic ones from which we obtained the magnitude vs frequency characteristics, and the time domain behaviour of digital filters. In this chapter, we present a variety of results to illustrate the validity of our approach to digital filters.

3.2 Frequency Domain --- Narrowband Response

The filters considered were all derived from Butterworth lowpass poles, orders five to ten, and the other pertinent parameters were assigned the values shown in Table 3.1. These were arbitrary except that the centre frequency, cutoff frequency and bandwidth were constrained to be less than the Nyquist frequency, which is one-half the sampling frequency, or 10π rad/sec for all our filters. As a result, a sine wave which happens to have a frequency equal to the Nyquist rate is sampled only twice per cycle and these samples are not necessarily the maximum value of the waveform. Figure (3.1) illustrates the situation where, depending on the relative phase of the analog sig-

Parameter	Value
SAMPLING PERIOD	0.1 sec
CENTRE FREQUENCY	3.0 rad/sec
BANDWIDTH	1.0 rad/sec
CUTOFF FREQUENCY	3.0 rad/sec

Table 3.1

nal and the sampler, the input to the digital filter would be any sequence of the form $\{(\pm A)^n\}$, n=1,2,3,.., where $0<A<A_{max}$ If, for example, the filter did not attenuate at the Nyquist rate (e.g. a highpass filter), the output sequence would be exactly the same as the input sequence whatever the value of A. Consequently, we should ensure that the highest

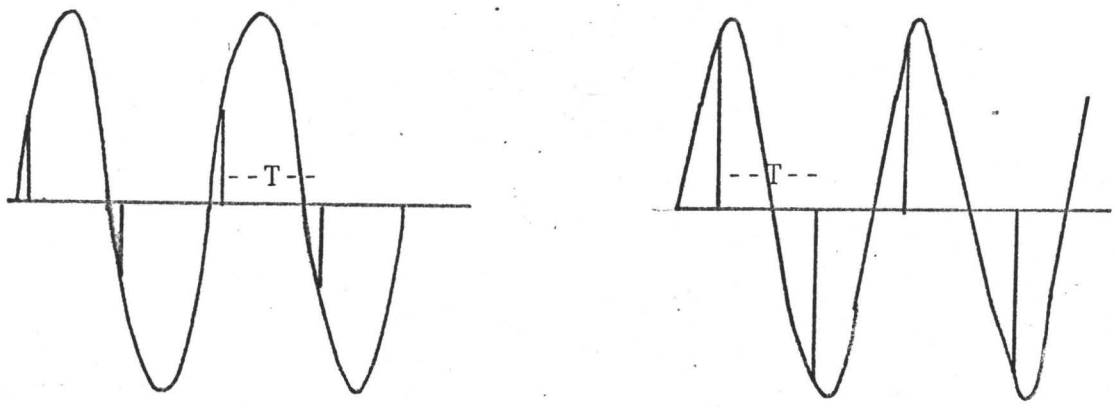


Figure (3.1)

frequency in the analog waveform will be sampled several times per cycle. This will provide satisfactory resolution at the output[2] and in our investigations of the time domain behaviour, no less than ten samples per cycle were used.

The basic magnitude vs frequency curves of these discretized Butterworth filters are shown in Figures (3.2)-(3.5) and (3.7)-(3.10). The first set is derived from the cascade structure and it is evident that [1] the -6N dB/octave slopes and cutoff frequency for the Nth order lowpass and highpass filters are preserved over the range $1<\omega<7$ rad/sec; [2] the centre frequency and bandwidth of the bandstop and bandpass

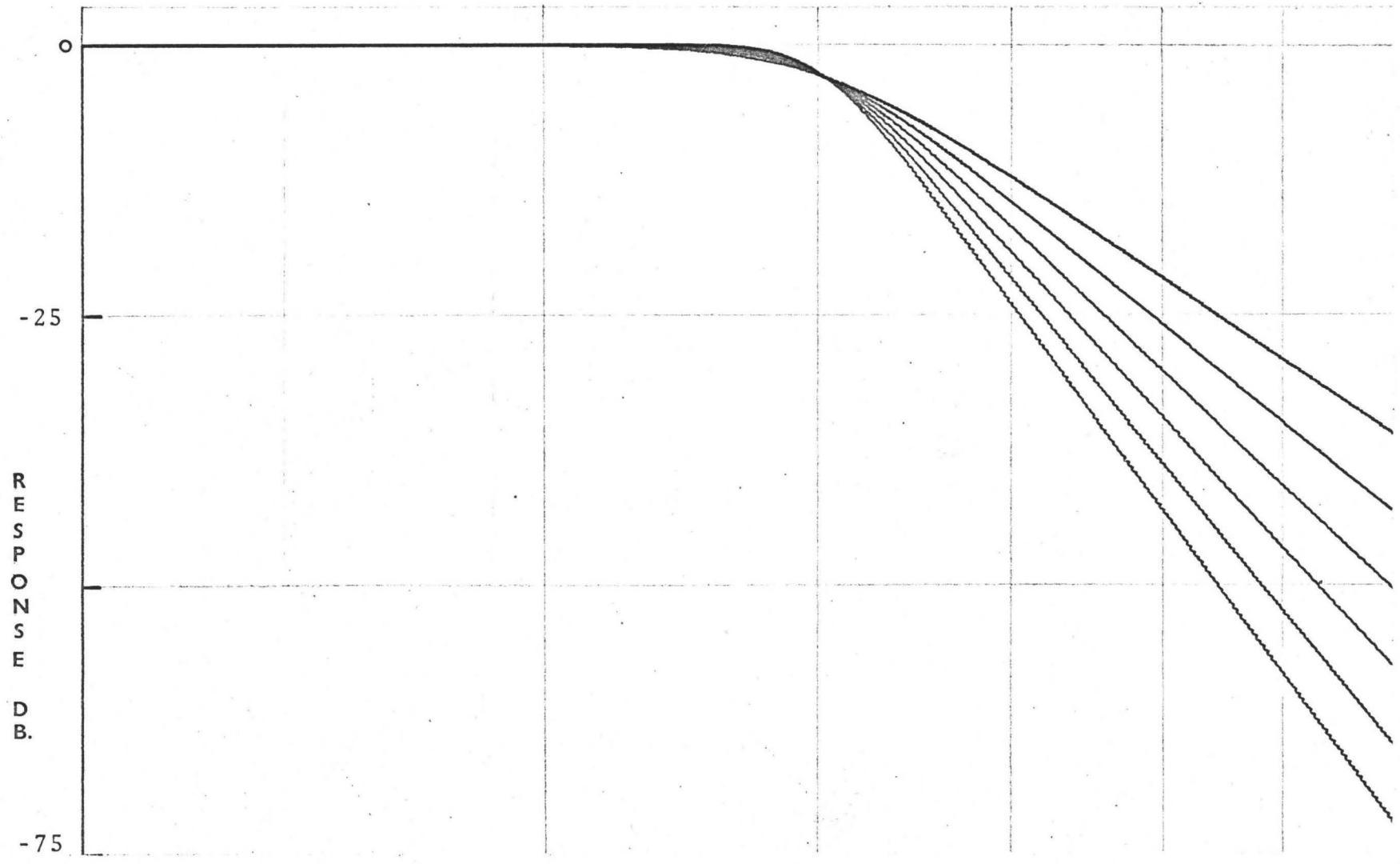


Figure (3.2)

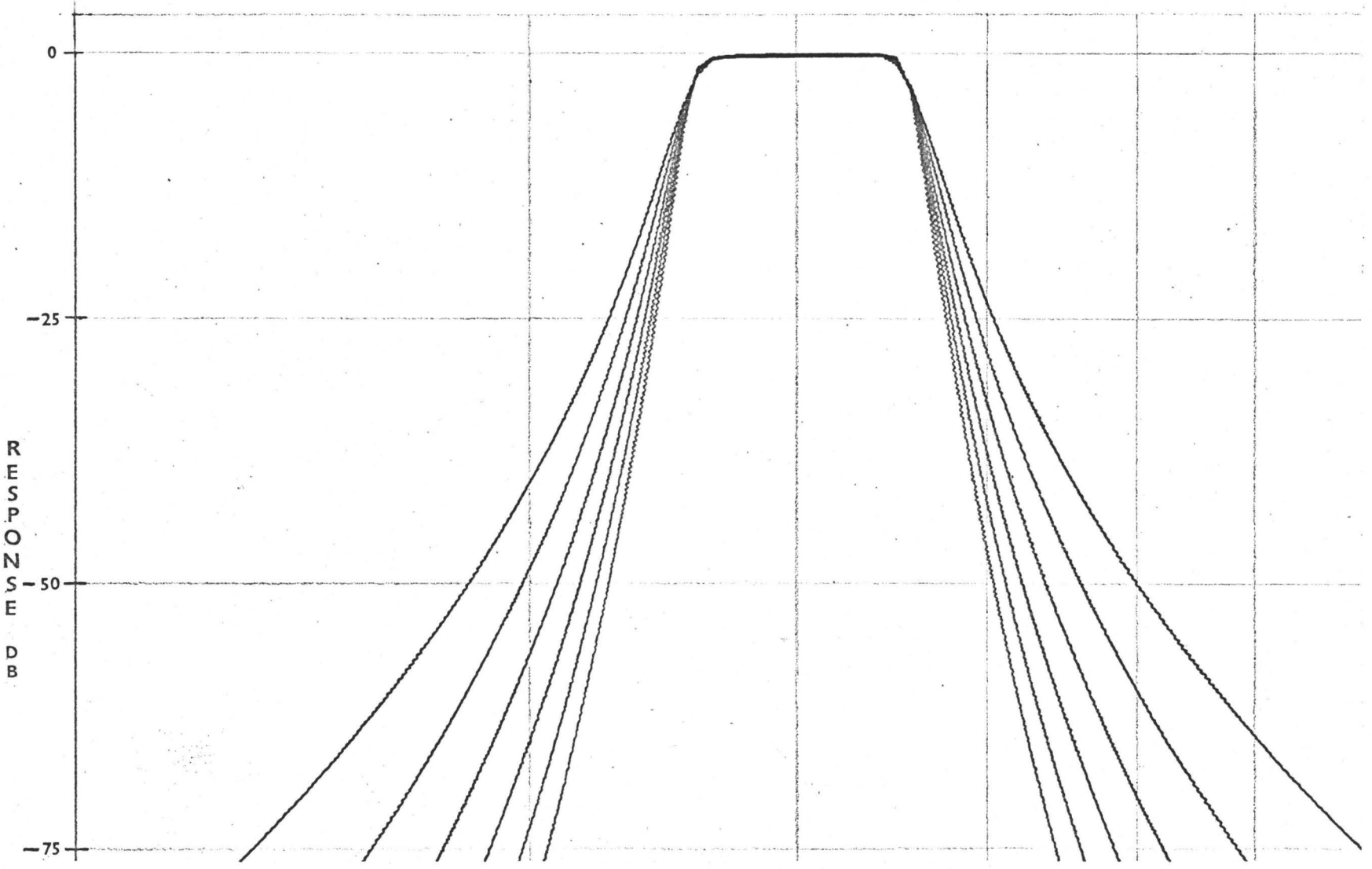


Figure (333)

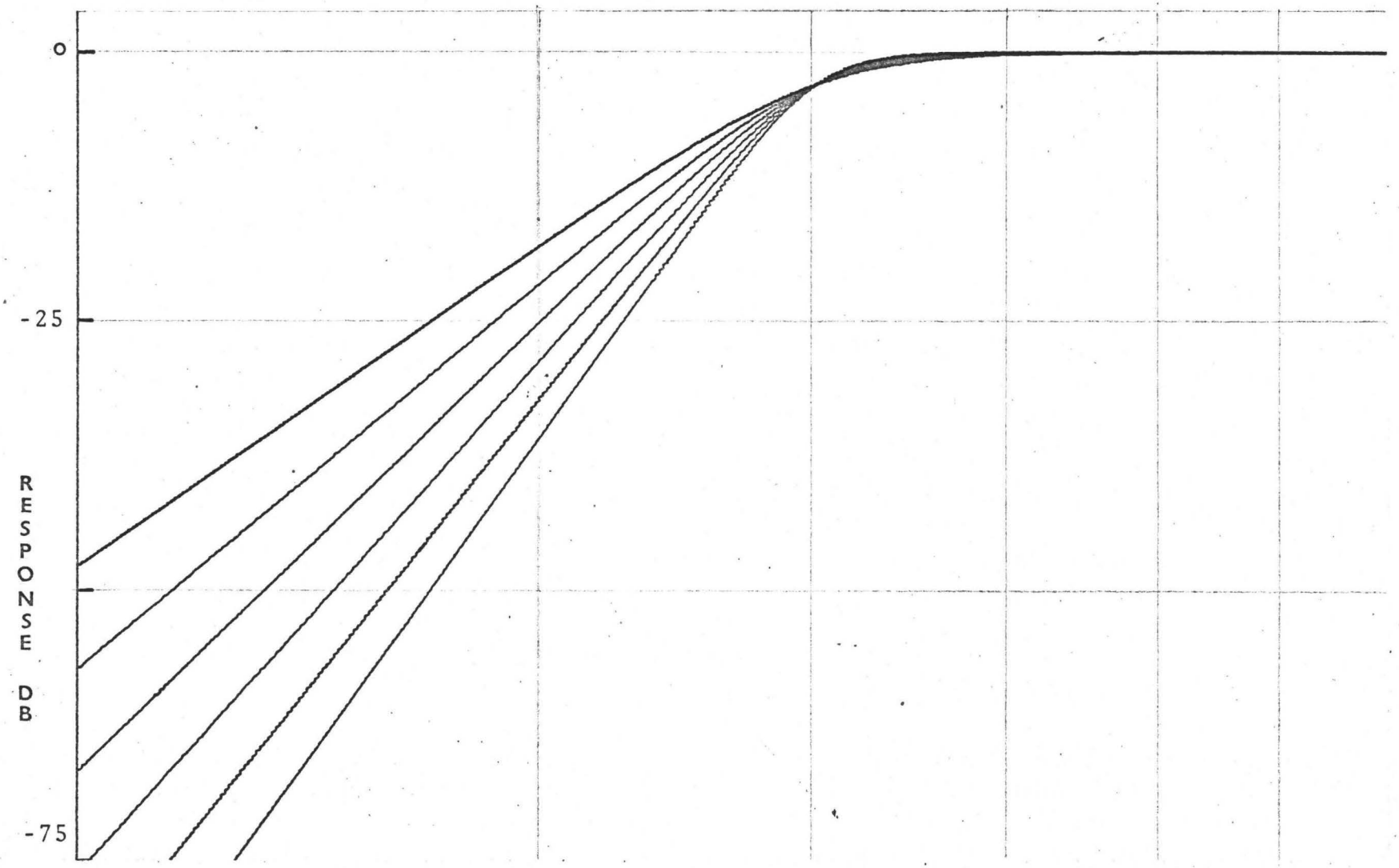


Figure (3.4)

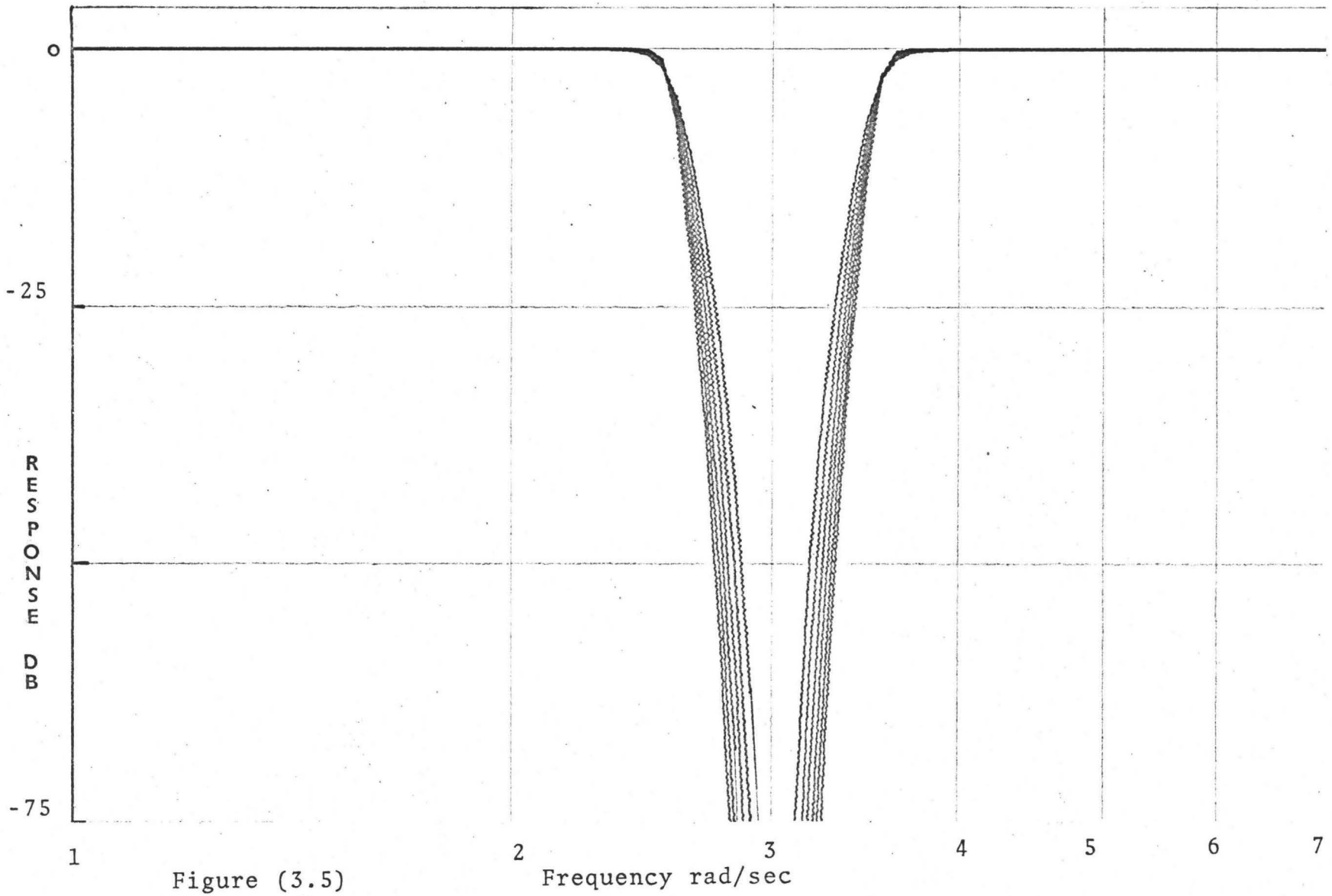


Figure (3.5)

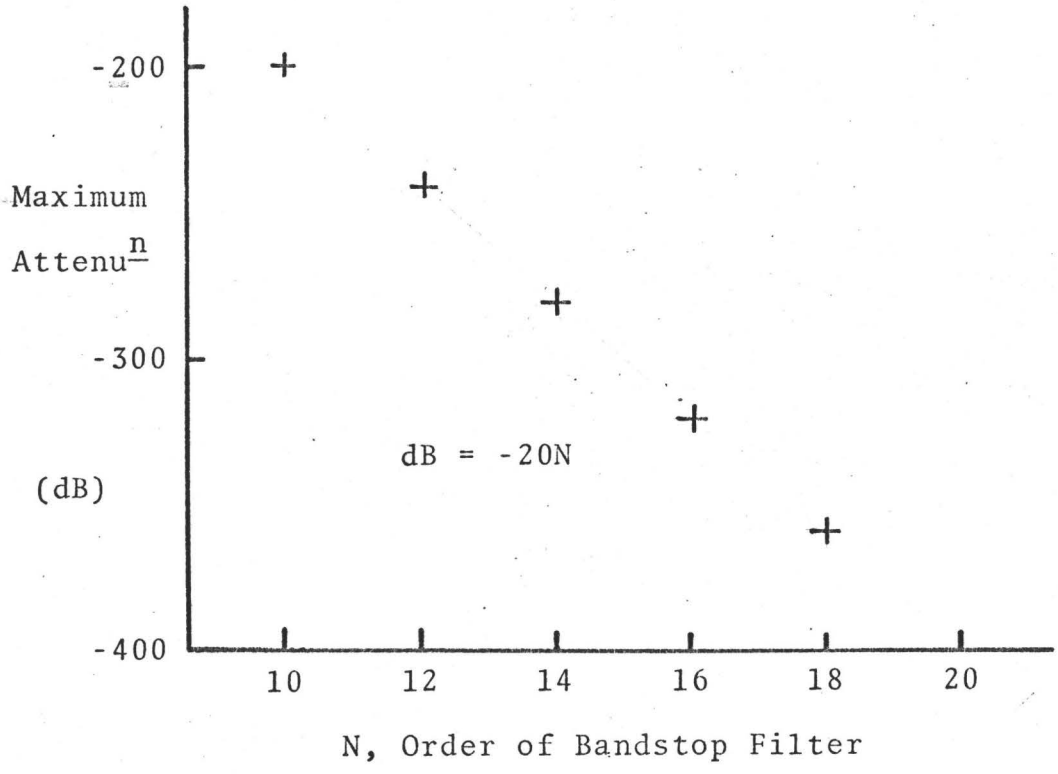


Figure (3.6)

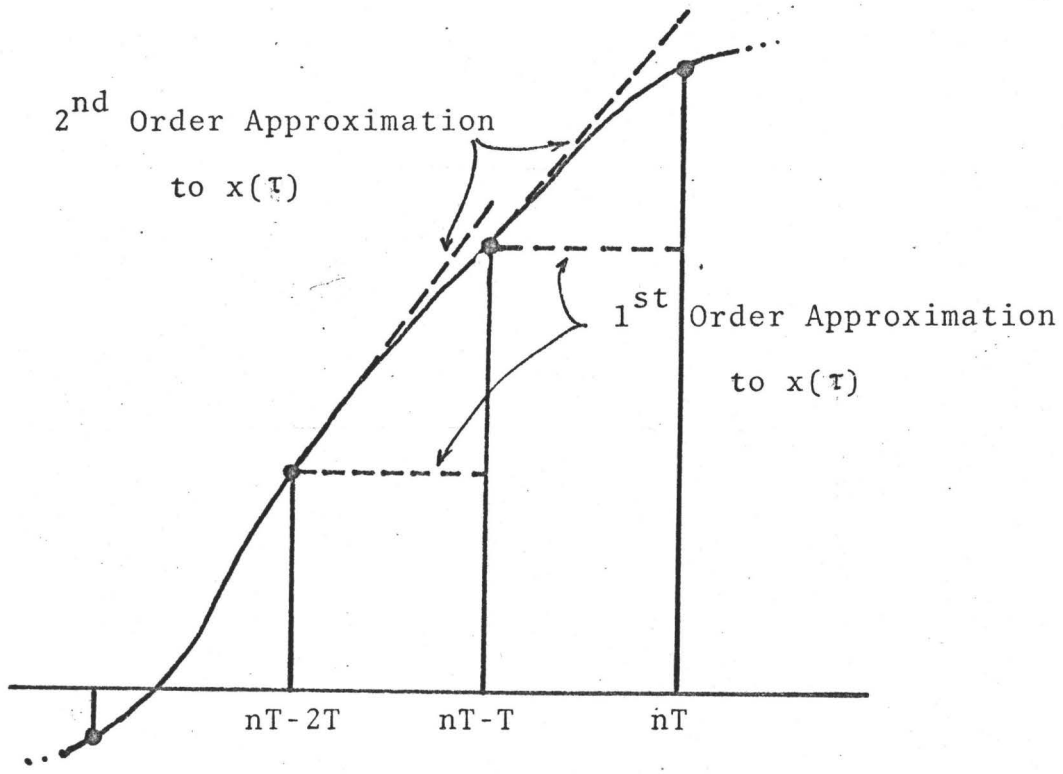


Figure (4.2)

filters are preserved also**.

A feature of these bandstop filters worth noting is the very large attenuations achieved in the stopband. Figure (3.6) summarizes the variation of this maximum attenuation as the order of the filter varies. Recall that the original analog transfer functions required a zero of transmission at the centre frequency and this is approximated well by these filters since -200 dB (e.g.) is 1 part in 10^{10} at the output. The small discrepancy is the result of using only seven decimal accuracy poles, and of the two approximations in the convolution integral (Section 2.2). Further reference will be made in Section 3.3 concerning the conditions for actually realizing such large attenuations.

The second set of characteristics[(3.7-3.10)] is based on the same parameter values as the first set, but now the parallel structure is used. The lowpass and bandpass filters are very similar to the corresponding cascade filters, but in general, the parallel versions are closer to the analog characteristics, as the printout of Table (3.2) shows. It is a portion of the frequency vs response (in dB) data around the centre frequency of an 6th order bandpass digital filter, for each realization. The 0.03 dB attenuation (i.e. 99.7% of the signal is passed) is significantly better than 0.2 dB for the

** The centre frequency for all filters was not exactly 3.0 rad/sec, but was 2.996 rad/sec.

BP

FREQ. R/S	MAGN.	PH
PARALLEL		
1.20951	-95.35638	I
1.38230	-85.20661	I
1.55509	-75.19872	I
1.72788	-65.01277	I
1.90066	-54.31073	I
2.07345	-42.67316	I
2.24624	-29.49828	I
2.41903	-13.93374	I
2.59181	-.87942	I
2.76460	-.02853	I
2.93739	-.03125	I
3.11018	-.03504	I
3.28296	-.04180	I
3.45575	-.63160	I
3.62854	-8.00704	I
3.80133	-18.88574	I
3.97411	-28.00726	I
4.14690	-35.57400	I
4.31969	-42.00936	I
4.49248	-47.59892	I
4.66527	-52.53505	I
4.83805	-56.95235	I
5.01084	-60.94848	I
5.18363	-64.59635	I
5.35642	-67.95180	I
5.52920	-71.05848	I
5.70199	-73.95110	I
5.87478	-76.65772	I
6.04757	-79.20128	I
6.22035	-81.60081	I
6.39314	-83.87225	I
6.56593	-86.02903	I
6.73872	-88.08262	I
6.91150	-90.04287	I
7.08429	-91.91828	I
7.25708	-93.71627	I
7.42987	-95.44332	I
7.60265	-97.10513	I
7.77544	-98.70675	+
7.94823	*00.25265	+

PASSBAND

CASCADE
-95.712
-85.551
-75.530
-65.329
-54.611
-42.956
-29.762
-14.177
-1.101
-.226
-.204
-.181
-.159
-.719
-8.063
-18.909
-27.996
-35.527
-41.925
-47.476
-52.371
-56.747
-60.699
-64.302
-67.610
-70.669
-73.512
-76.167
-78.658
-81.003
-83.218
-85.317
-87.312
-89.211
-91.024
-92.759
-94.420
-96.015
-97.548
-99.024

6th Order

FTable (3.2)

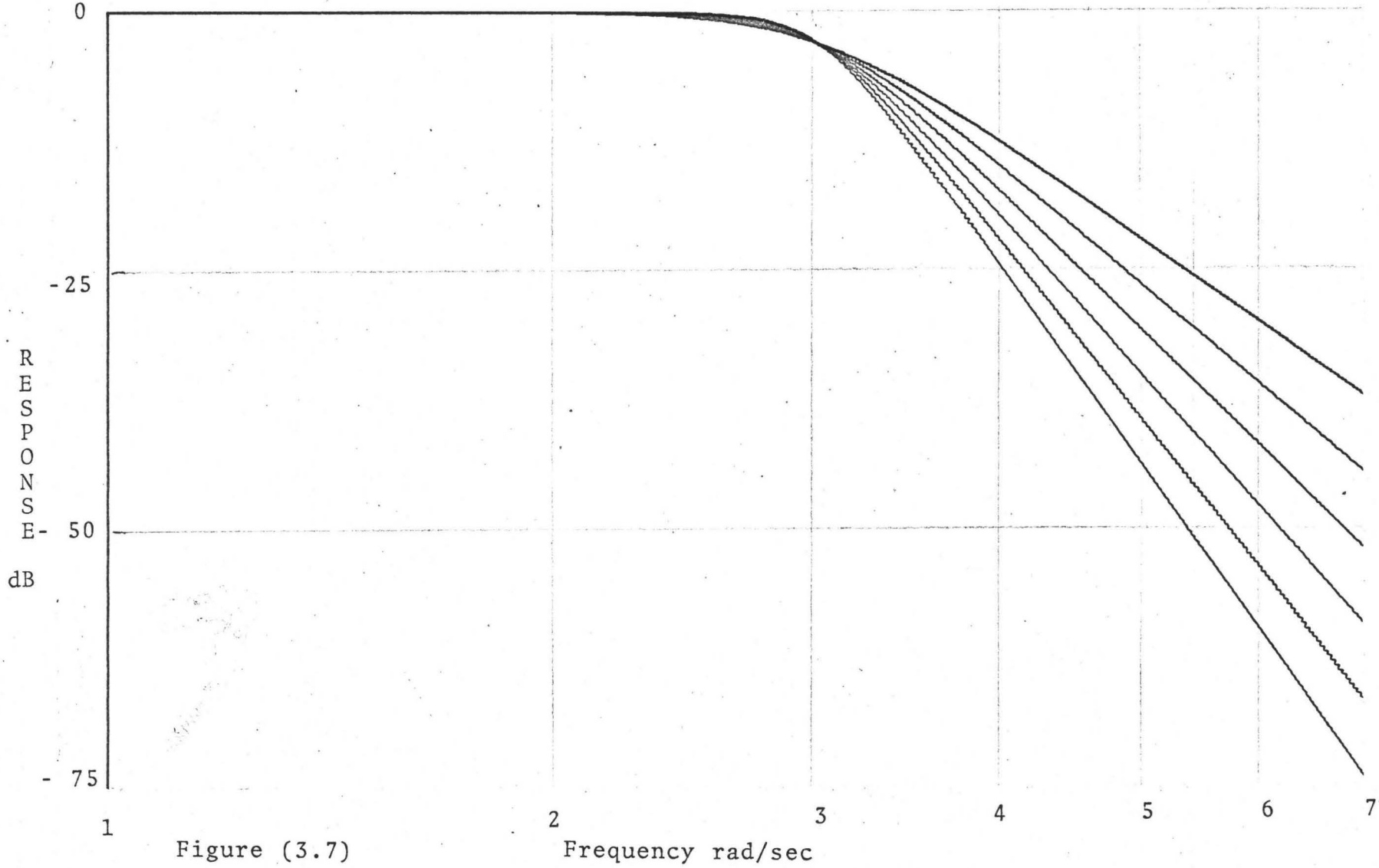


Figure (3.7)

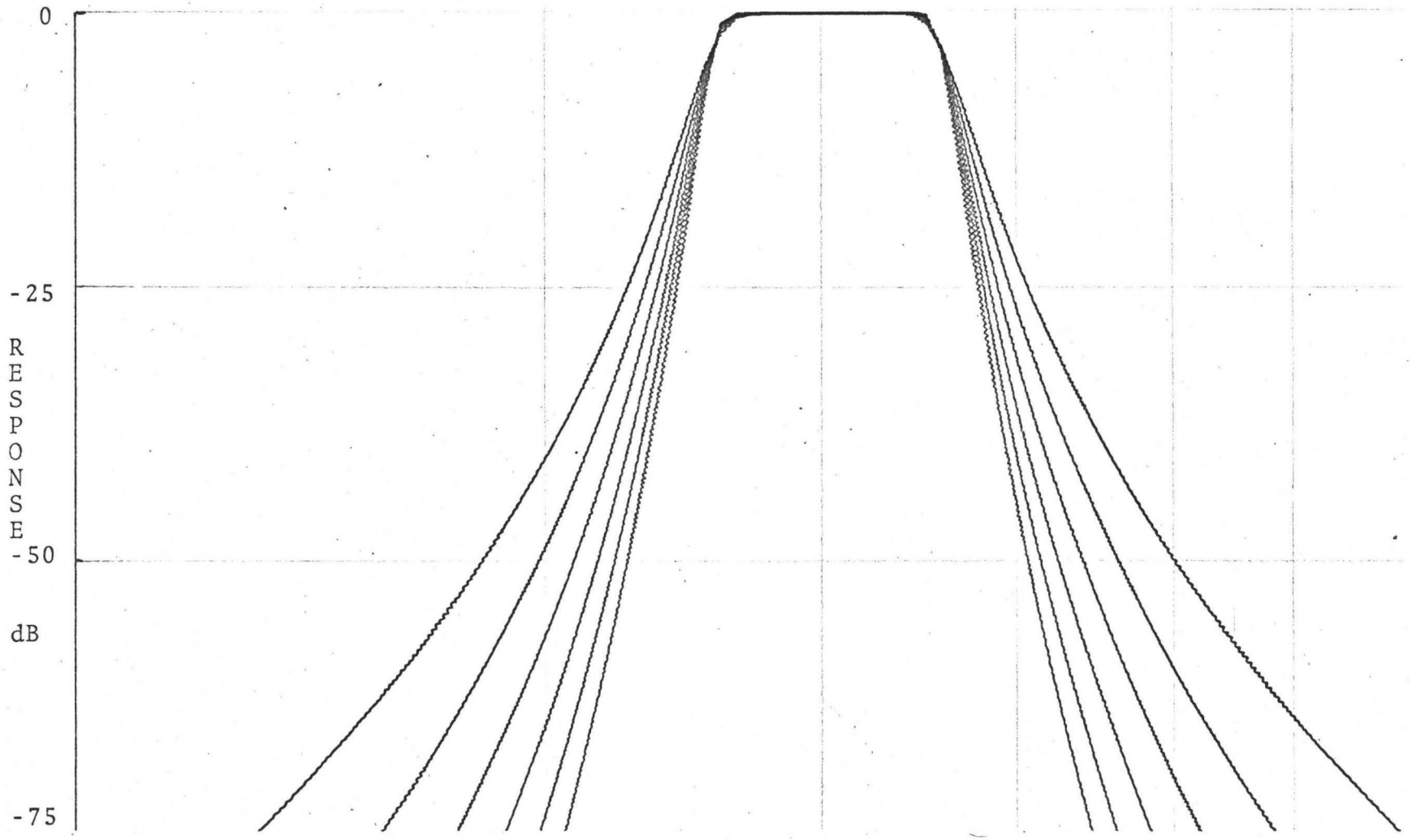


Figure (3.8)

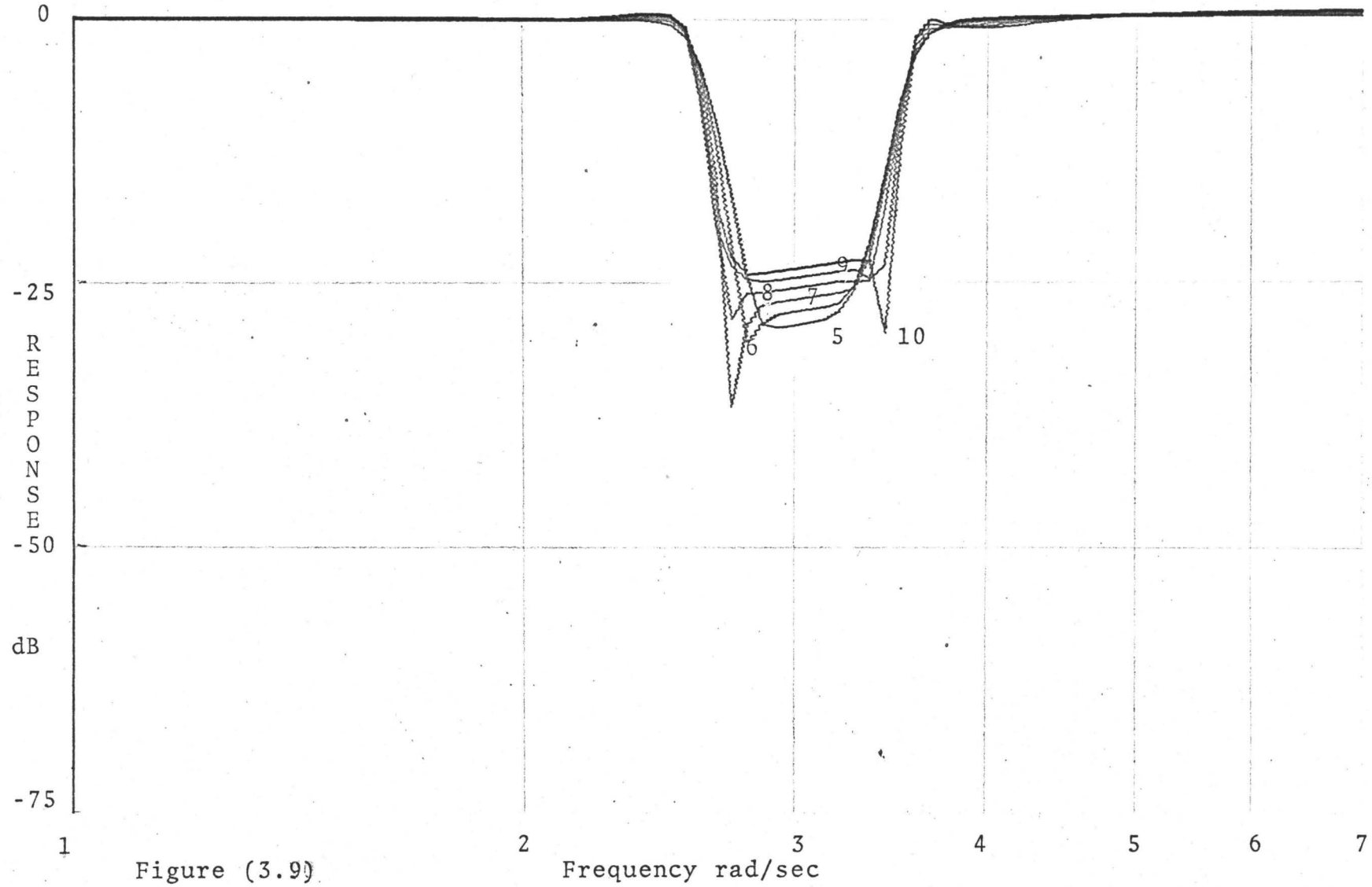


Figure (3.9)

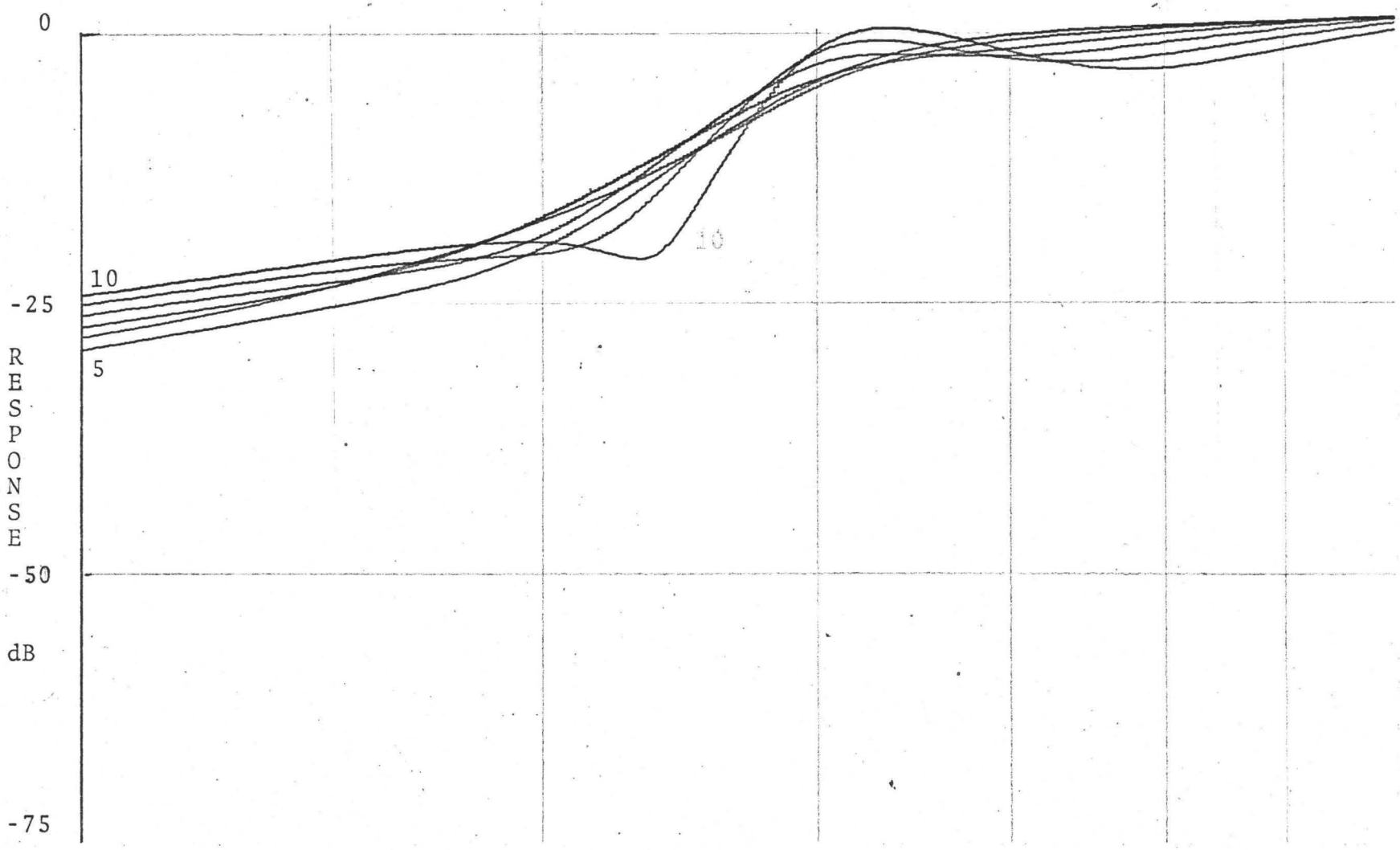


Figure (3.10)

cascade filter. As well, the parallel transformation preserves the centre frequency and bandwidth of bandpass filters, and the cutoff frequency and -6N dB/octave slopes of the lowpass filters.

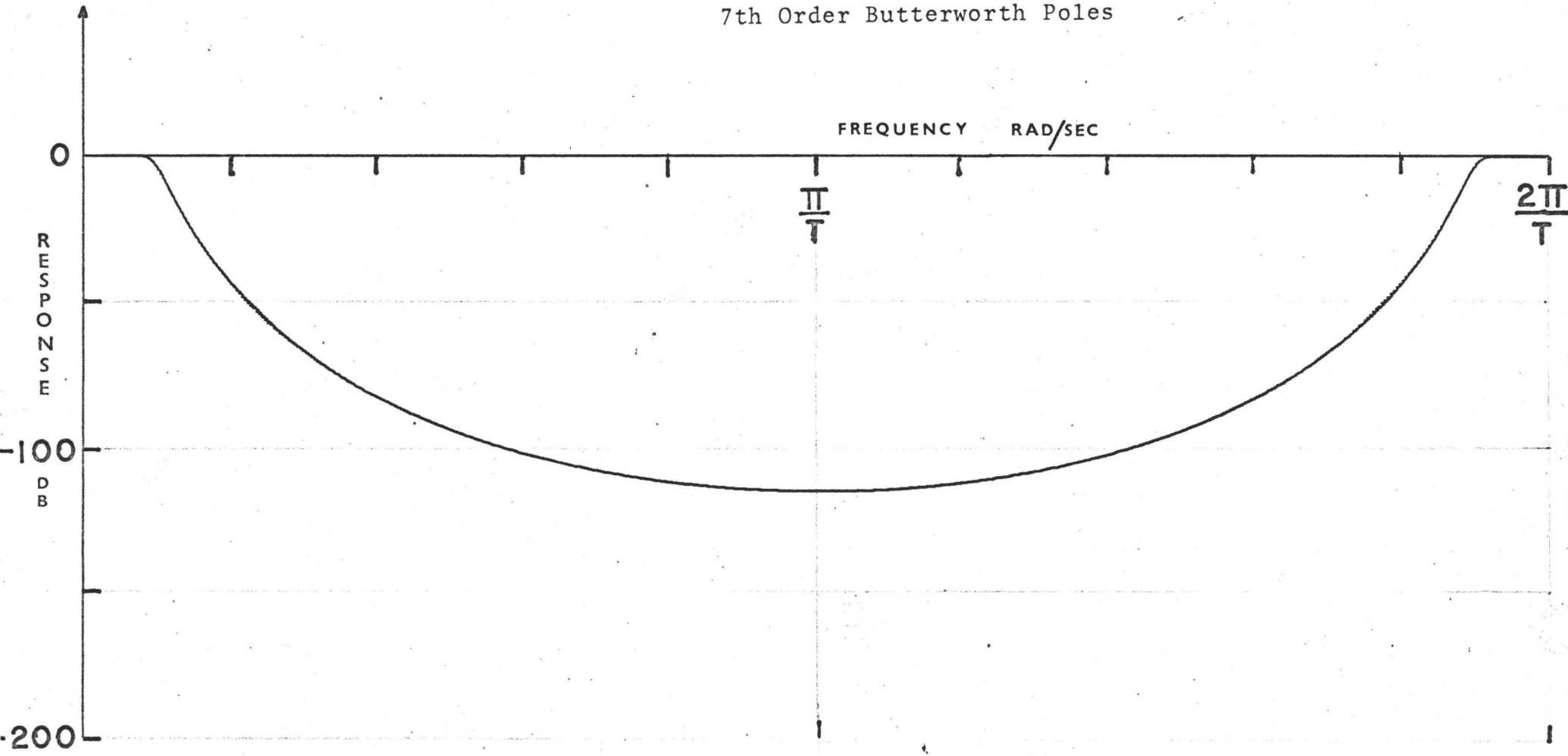
The major difference in performance between the parallel and cascade structures shows up in the bandstop and highpass filters. The bandstop characteristic [Figure(3.9)] exhibits about -25 dB of attenuation in the stopband, a startling change from the -280 dB for the cascade case. In terms of the percent of the signal that is rejected, the cascade filter stops essentially 100% of the input, while the parallel filter rejects about 95%. Otherwise, the characteristics of the parallel bandstop filter are as good as those of the cascade version; both the centre frequency and bandwidth exhibit only small(1%) deviations from their specified values. The worst performance of all is the parallel highpass filter. The characteristics of these filters [orders 5-10] are shown in Figure (3.10), and it is evident that they are completely unsuitable as highpass filters. As well as exhibiting slopes which are much less than +6N dB/octave, and shifting the cutoff frequency significantly, they all have gains at higher frequencies.

3,3 Frequency Domain --- Wideband Response

Since the above two sets of frequency curves are limited to the range $1 \leq \omega \leq 7$ rad/sec, and since the Nyquist frequency is 10π rad/sec, the wideband behaviour of the filters is not

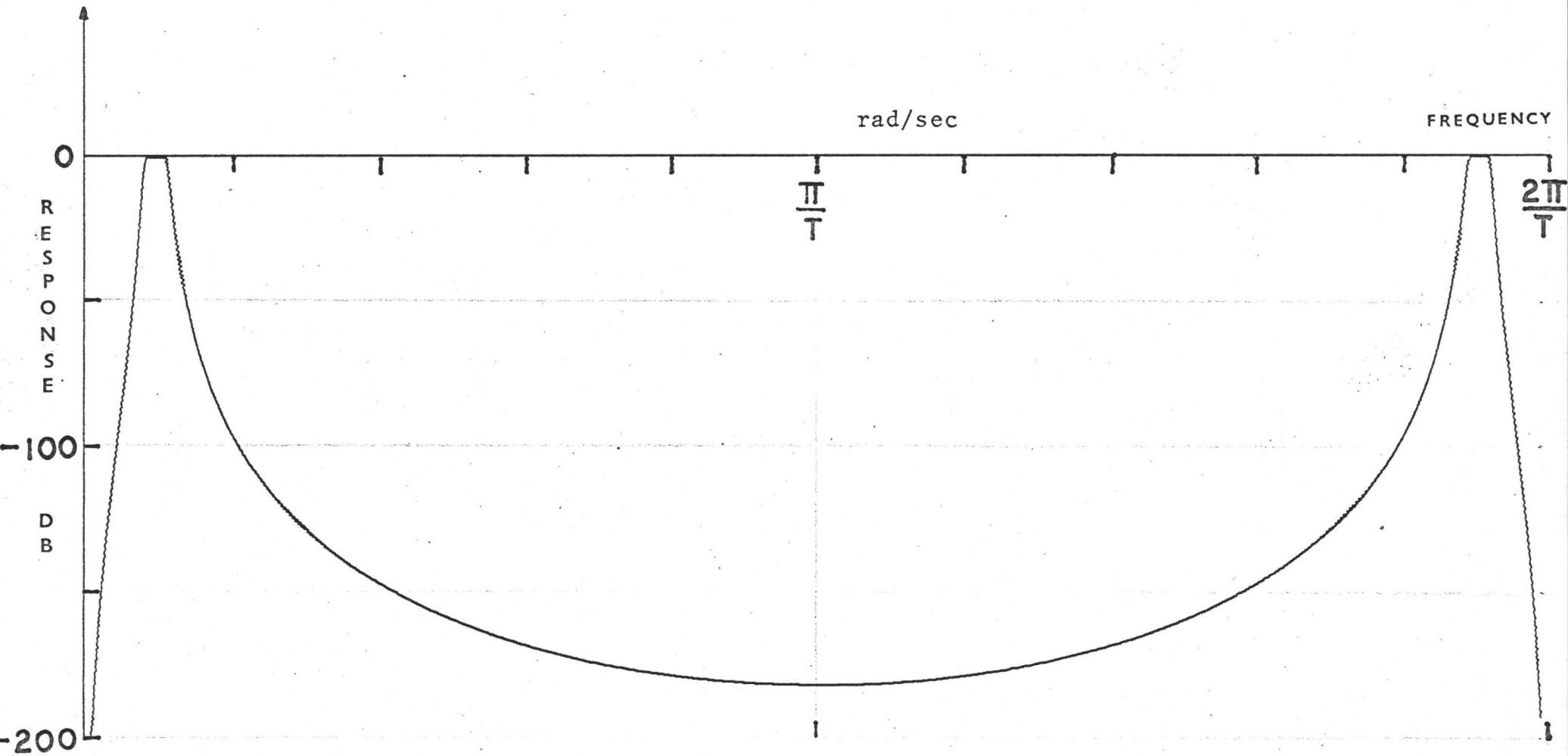
evident. Recall that we predicted in Section (2.2), on the basis of pole-zero patterns, that there should be a deterioration at higher frequencies in the response curves of bandstop and highpass filters realized by the C-A method. This is because the process of eliminating the "half-delay" introduces a pole near the Nyquist frequency. Figures (3.11) - (3.14) show the wideband response curves for filters derived from seventh order Butterworth poles using the cascade structure only. The effect of the unwanted poles is quite evident although the gainband occurs sufficiently close to the Nyquist frequency so that the filters can still be considered "wideband", i.e., useful over 70% or more of the Nyquist range [1]. In fact, all the curves remain at the 0 dB level for about 80% of this range.

A second form of distortion (compared to the analog responses) which is inherent to digital filtering, is illustrated by all four curves in Figures (3.11) - (3.14). Since $z^{-1} = e^{-j\omega T}$ has a period of 2π, then the transfer functions $H(z^{-1})$ also repeat with the same period. Consequently, the -6N dB/octave slopes of a lowpass filter cannot continue indefinitely as the frequency increases, but must eventually become flat at the Nyquist rate. In many cases however, the attenuation reaches large enough values to cause no practical problem (e.g. the 5th order lowpass filter attains a minimum of -82 dB).



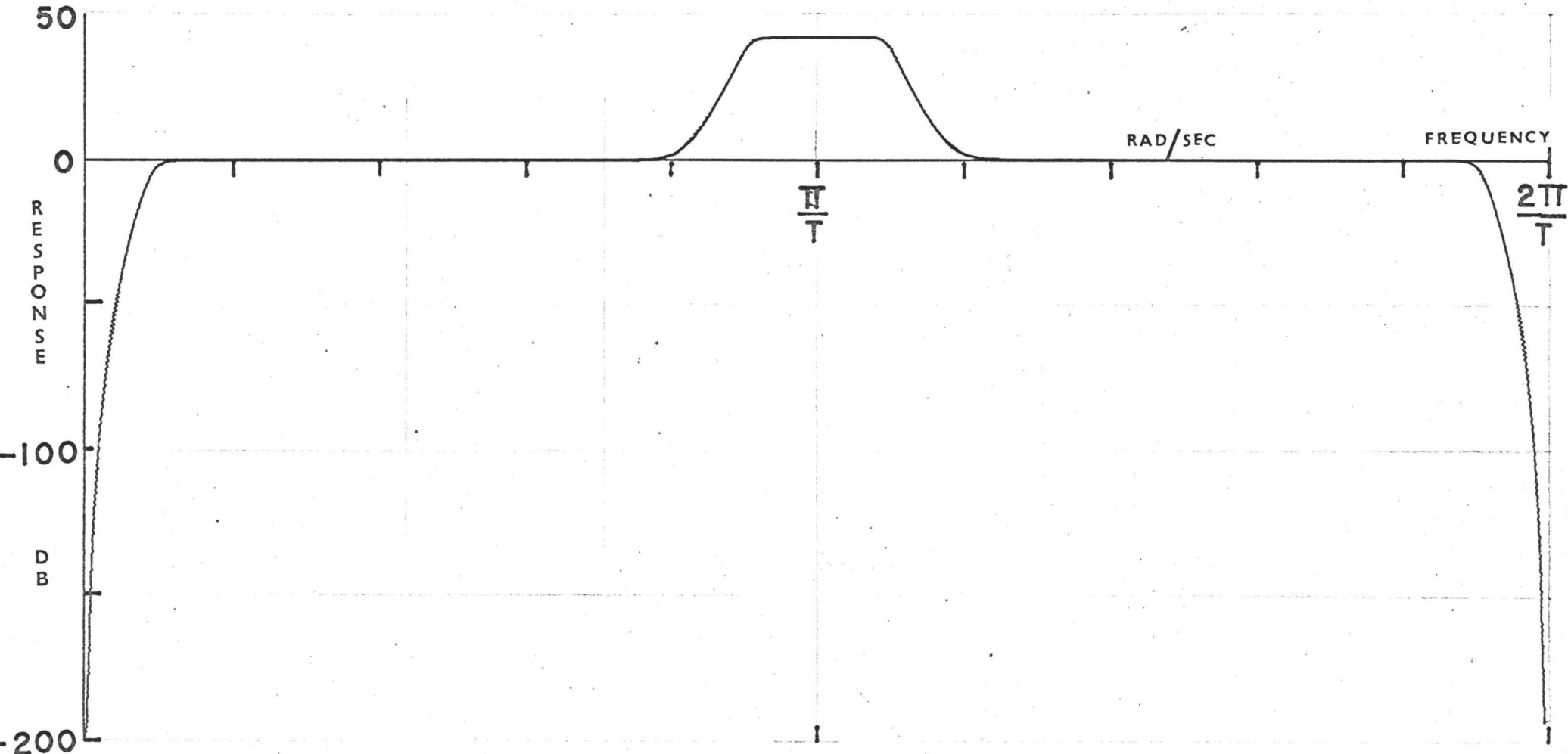
Lowpass Wideband Characteristic

Figure (3.11)



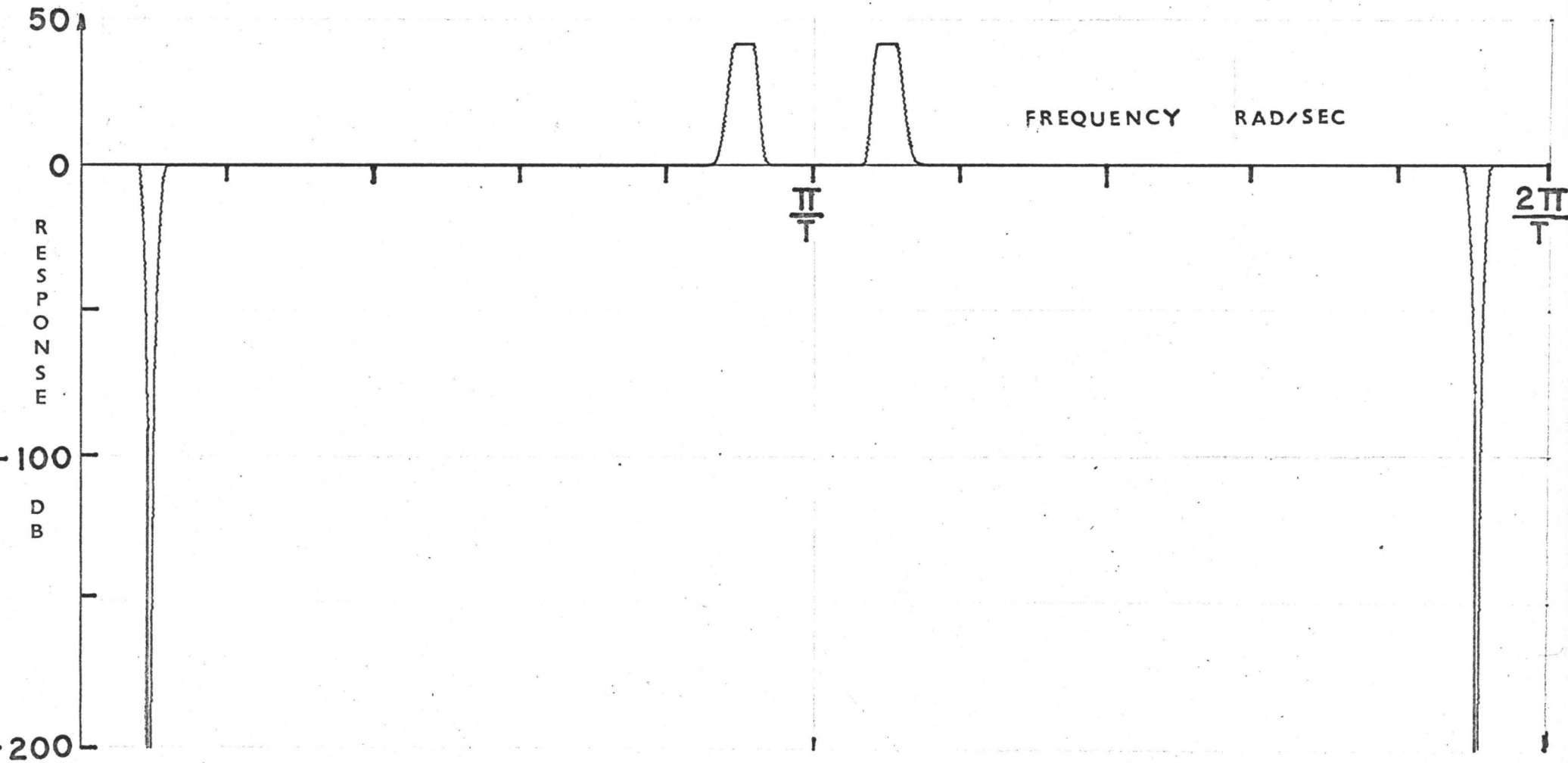
Bandpass Wideband Characteristic

Figure (3.12)



Highpass Wideband Characteristic

Figure (3.13)



Bandstop Wideband Characteristic

Figure (3.14)

The final investigation in the frequency domain was to obtain the magnitude and phase characteristics for digital allpass filters. These filters pass all frequencies unattenuated, but alter the phase of the output relative to the input, as the frequency varies. They can therefore, be used to compensate for phase distortion in another network [8]. This type of filter has poles and zeroes which are mirror images of each other across the jω axis in the complex frequency plane, and therefore, has a transfer function of the form

$$H(s) = \prod_{i=1}^{N} \left(\frac{s - p^*_i}{s + p_i} \right)$$

where p^*_i and p_i are a conjugate pair of poles. The pole-zero pattern for a 5th order allpass filter is shown in Figure (3.15). It is more convenient however, to regroup the poles and zeroes so that

$$H(s) = \prod_{i=1}^{N} \left(\frac{s - p_i}{s + p_i} \right) = \prod_{i=1}^{N} \left(1 - \frac{2p_i}{s + p_i} \right) \qquad (3.2)$$

Application of the above theory to Equation (3.2) leads to the discrete transfer function

$$H(z^{-1}) = \prod_{i=1}^{N} \left(z^{-1} - \frac{2[1 - e^{-p_i T}]z^{-2}}{1 - e^{-p_i T} z^{-2}} \right)$$

which is identical in form to the $H(z^{-1})$ for a highpass digital filter derived by the same method, so that some gain is expected near the Nyquist frequency. Both the magnitude and phase characteristics for fifth and tenth order allpass filters are presented in Figures (3.16) and (3.17), along with the corresponding results based on the bilinear transformation.

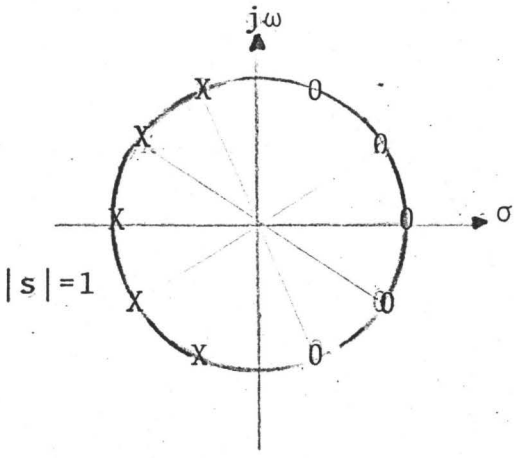
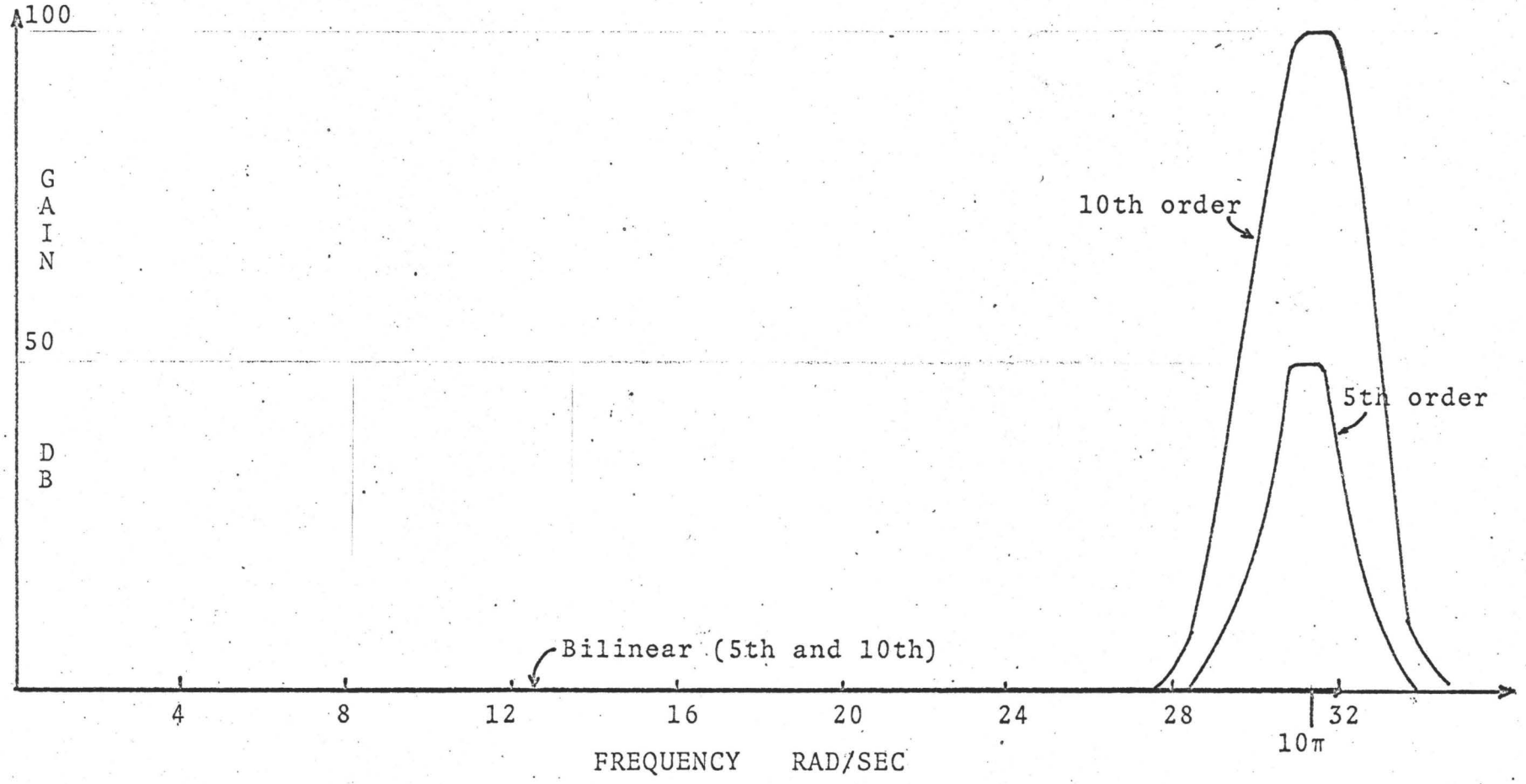


Figure 3.15

The magnitude curves show that the bilinear characteristic is perfectly flat over the total frequency range, while the C-A characteristic is flat [$\leq$0.05 dB] for 94% of the Nyquist interval. Centred about the Nyquist frequency is a gainband with a peak gain of about 10N dB, where N is the order of the filter. Based on this curve alone, our realization is definitely inferior; the phase characteristics however, possess a surprising property.

For both realizations, the phase curves are irregular near ω = 0.0 rad/sec . Past about 2 rad/sec, the bilinear curves start to change monotonically from 0° to -360° while the C-A phase curves decrease <u>almost linearly</u> over 86% of the Nyquist interval, the rate increasing with the order, N. The linear phase vs. frequency curves mean that the delay, $-\frac{\partial\phi}{\partial\omega}$ is constant over $2<\omega<27$ rad/sec for each order of filter.



Bilinear and Convolution Gain

Characteristics

Figure (3.16)

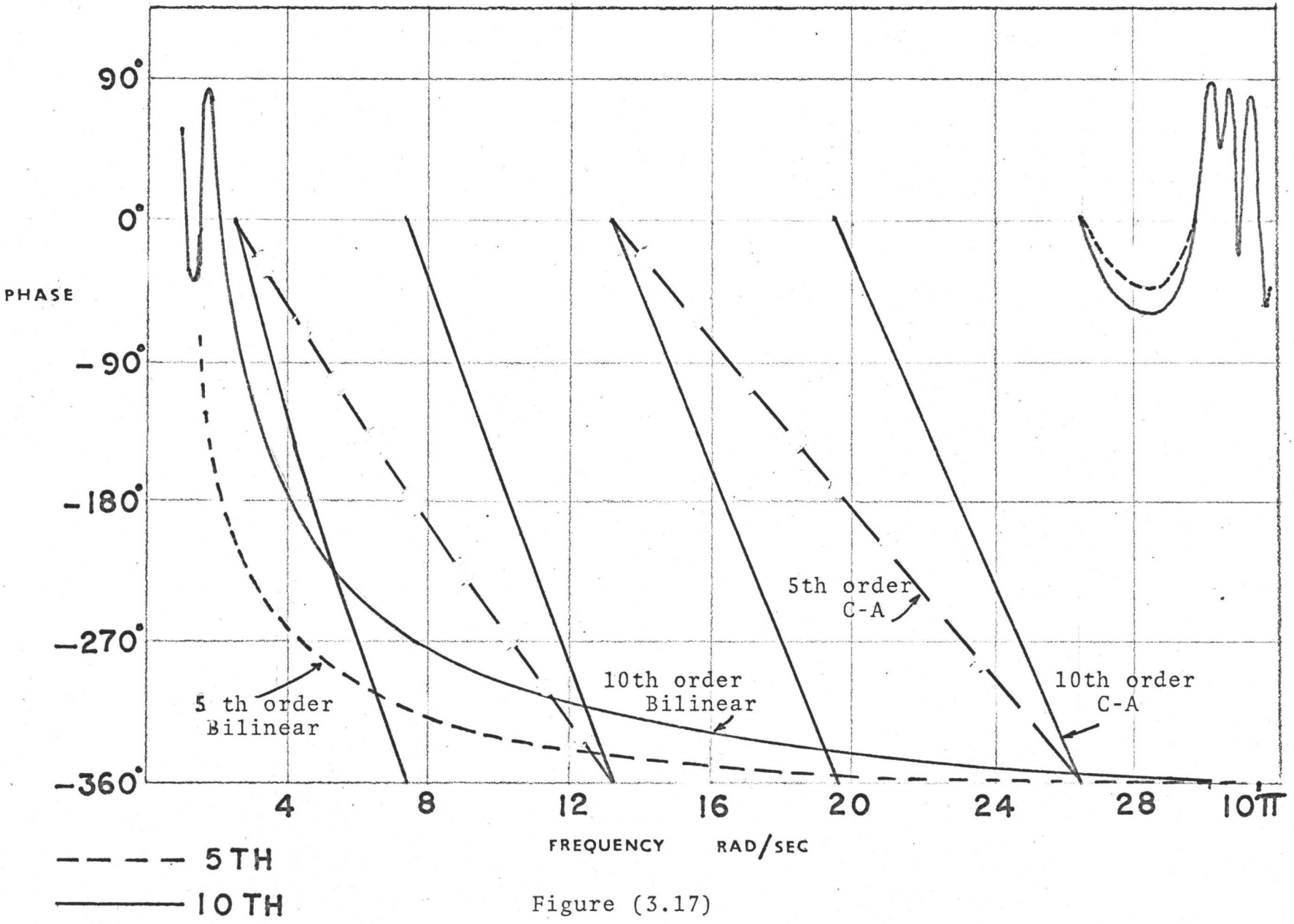

90°
0°
-90°
-180°
-270°
-360°
PHASE
4
8
12
16
20
24
28
10π
FREQUENCY RAD/SEC
5th order
C-A
10th order
Bilinear
10th order
C-A
5 th order
Bilinear
5TH
10TH

Figure (3.17)

The delays derived from these curves were 0.6, 0.8 and 1.1 sec for the 5th, 7th and 10th order filters. Evidently then, the delay will be about 1.1NT where N is the order and T is the sampling period. This unique phase characteristic which allows this filter to function as a fixed delay element , is a result of the particular transformation from analog to discrete domains.

3.4 Time Domain

The equations (2.8), (2.18), (2.21) and (2.24) , which are the fundamental difference equations for the four cascade filters, allow us to investigate the time-domain behaviour of these digital filters. In particular, the transient response can be determined and the steady state response can be verified [the steady state magnitude and phase are known from $H(z^{-1})$]. It is in the form of difference equations, naturally, that digital filters are implemented to process a time sequence, since they are the time-domain expression of the particular filter.

Seventh order poles were used to set up the difference equations for a 14th order bandstop digital filter. (Figure (3.1) summarizes the particular values of parameters) The cascade realization had seven blocks of the form of Figure (2.7) and the input to the first block was a sampled sine wave with a frequency of 1.0 rad/sec; the output of the seventh block is shown in Figure (3.18) where it is

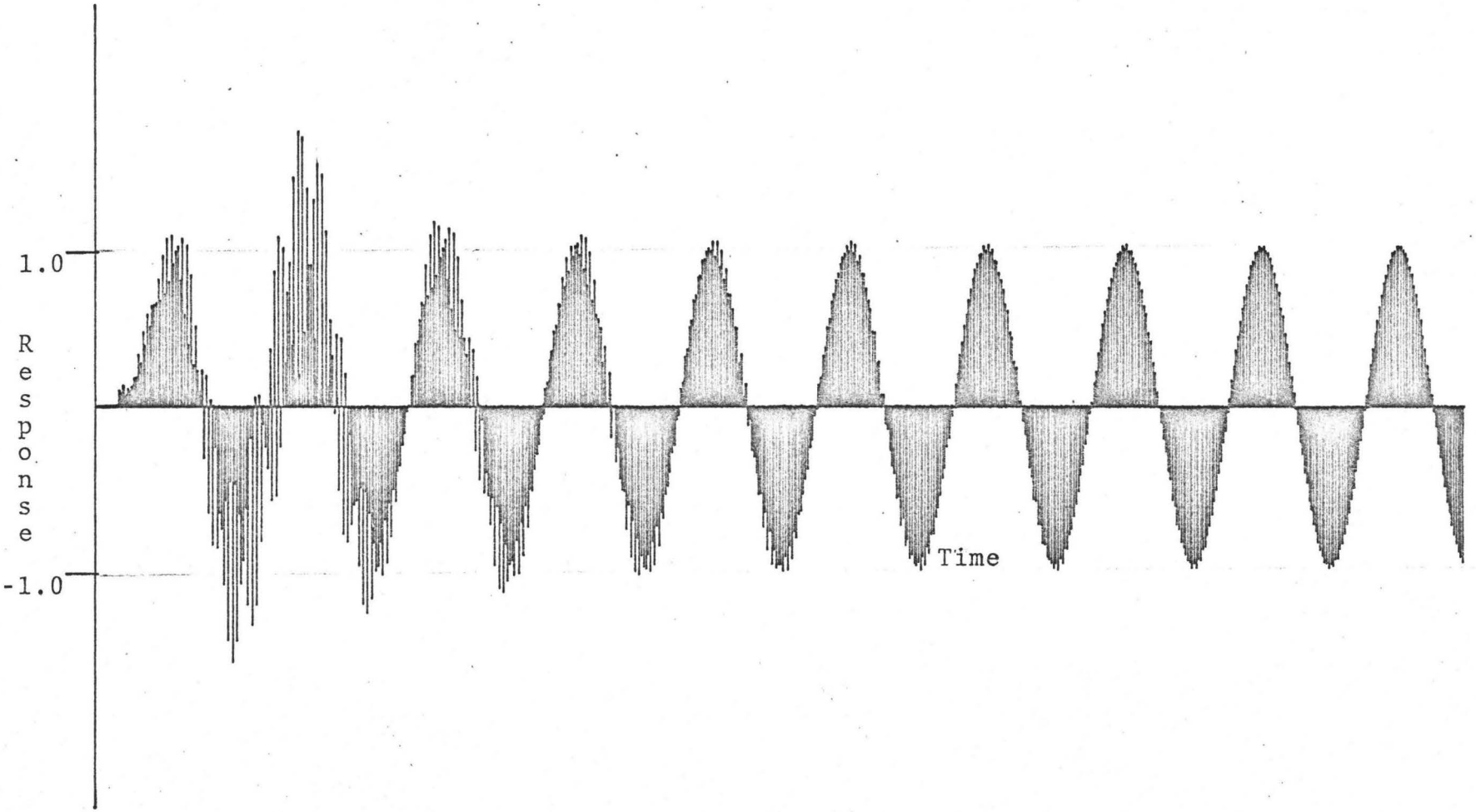


Figure (3.18)

evident that the system has settled to about the 0 dB level after about nine cycles. A high frequency component [10π rad/sec] gives the waveform a ragged appearance up to this time. Note also the delay at the beginning, which is the finite time taken for the signal to propagate through the filter. It is simply the sum of seven delays each T second long, or 0.7 sec in this case.

Using the same bandstop filter, we changed the input to a sampled sine wave with a frequency equal to the centre frequency of the bandstop filter, 3.0 rad/sec. Figure (3.19) shows the characteristic response for high order bandpass or bandstop filters at their centre frequencies. To approach the attenuation predicted by the frequency analysis for this filter [-280 dB], the filter requires a delay exceeding 100 cycles. Even then there is evidence which suggests that this level cannot be achieved without increased precision in pole locations at least. The evidence is this: since the input is a sequence of real numbers, the output must also be real, regardless of the complex operations involved in the intermediate stages. It has been found experimentally, however, that the output <u>does</u> contain a steady state imaginary component with an rms value on the order of 3.0×10^{-13} for unity amplitude input. (Figure(3.20a)) Such a residual value which must be present also at the real output, is the result of the seven decimal approximation of the poles locations and also the approximation in the convolution integral (Section 2.3). This effect, analogous to crosstalk in a [6]

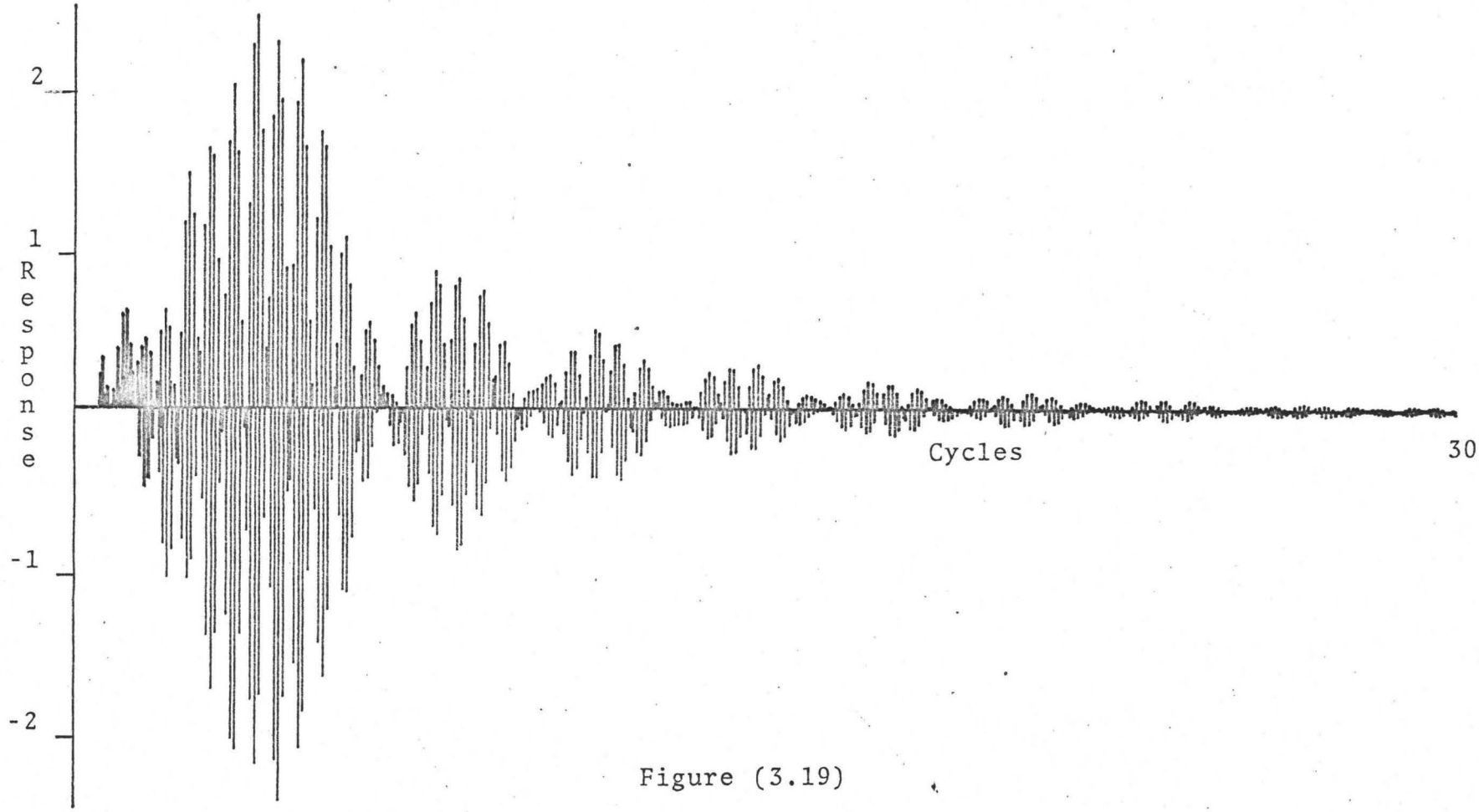


Figure (3.19)

Real Output Sequence	Imaginary Output Sequence
-7.39318E-01	-1.11022E-13
-7.86034E-01	1.15463E-13
-8.54981E-01	-8.08242E-14
-8.98393E-01	6.03961E-14
-9.35449E-01	-1.15463E-14
-9.74623E-01	-4.26326E-14
-9.79624E-01	1.17240E-13
-1.00981E+00	-1.82965E-13
-9.87361E-01	2.37144E-13
-1.00153E+00	-2.74447E-13
-9.58957E-01	2.88658E-13
-9.50202E-01	-2.81553E-13
-8.95034E-01	2.45137E-13
-8.59022E-01	-1.82077E-13
-7.96833E-01	9.59233E-14
-7.33278E-01	1.24345E-14
-6.66726E-01	-1.21680E-13
-5.79534E-01	2.29594E-13
-5.08688E-01	-3.20188E-13
-4.04840E-01	3.91687E-13
-3.28473E-01	-4.25215E-13
-2.16249E-01	4.30767E-13
-1.33409E-01	-3.83582E-13
-2.06647E-02	3.08864E-13
6.81391E-02	-2.00451E-13
1.75070E-01	7.27196E-14
2.67444E-01	6.19504E-14
3.64034E-01	-1.97620E-13
4.56042E-01	3.21076E-13

Figure (3.20a)

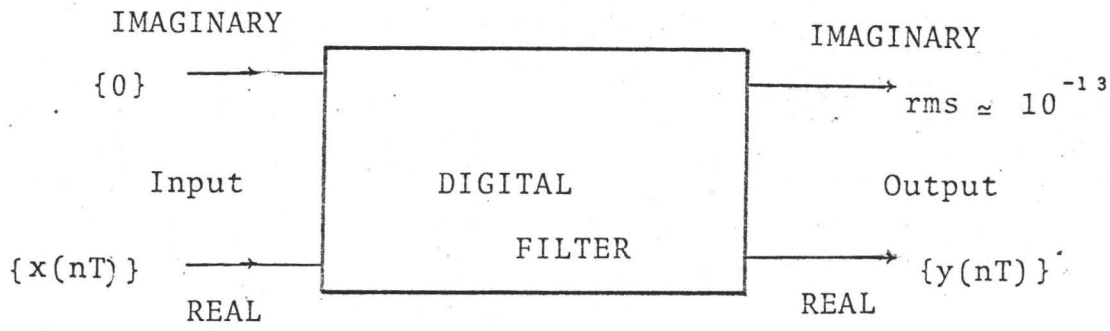


Figure (3.20b)

two channel system, prevents output signals below about -260 dB, unless greater precision is used as noted above.

To illustrate the behaviour of these digital filters further in the time domain, we chose as input, a sampled sine wave at the centre of the passband contaminated with band-limited Gaussian noise. The filter used was a tenth order Butterworth bandpass filter with a variable bandwidth. Figure (3.21) shows the total input signal where the first numbers to enter the filter are nearest the origin. Bandwidths of 0.5, 1.0, 5.0 and 10.0 rad/sec were used successively to determine if the digital filter would perform like its analog counterpart. These four outputs are shown in Figures (3.22) - (3.25). As the bandwidth decreases, the spectral purity of the output increases, but so do the rise time (T_R) and the delay (T_D) across the filter. Figures (3.26a,b) summarize the inverse relationships between each of these times and the bandwidth, B. Although both curves are hyperbolas, the delay curve is asymptotic to 1.0 sec since there is an inherent 1.0 sec delay across a 10th order filter, regardless of the bandwidth. Note that the rise time is defined as the time for the envelope of the transient to rise from 10% to 90% of its steady state value [compare with (8)], and the delay is defined as the time taken for the envelope of the response to rise to 50% of its steady state value. The observed relation $T_R \propto \frac{1}{B}$ is further evidence that the digital filter functions in basically the same way as the

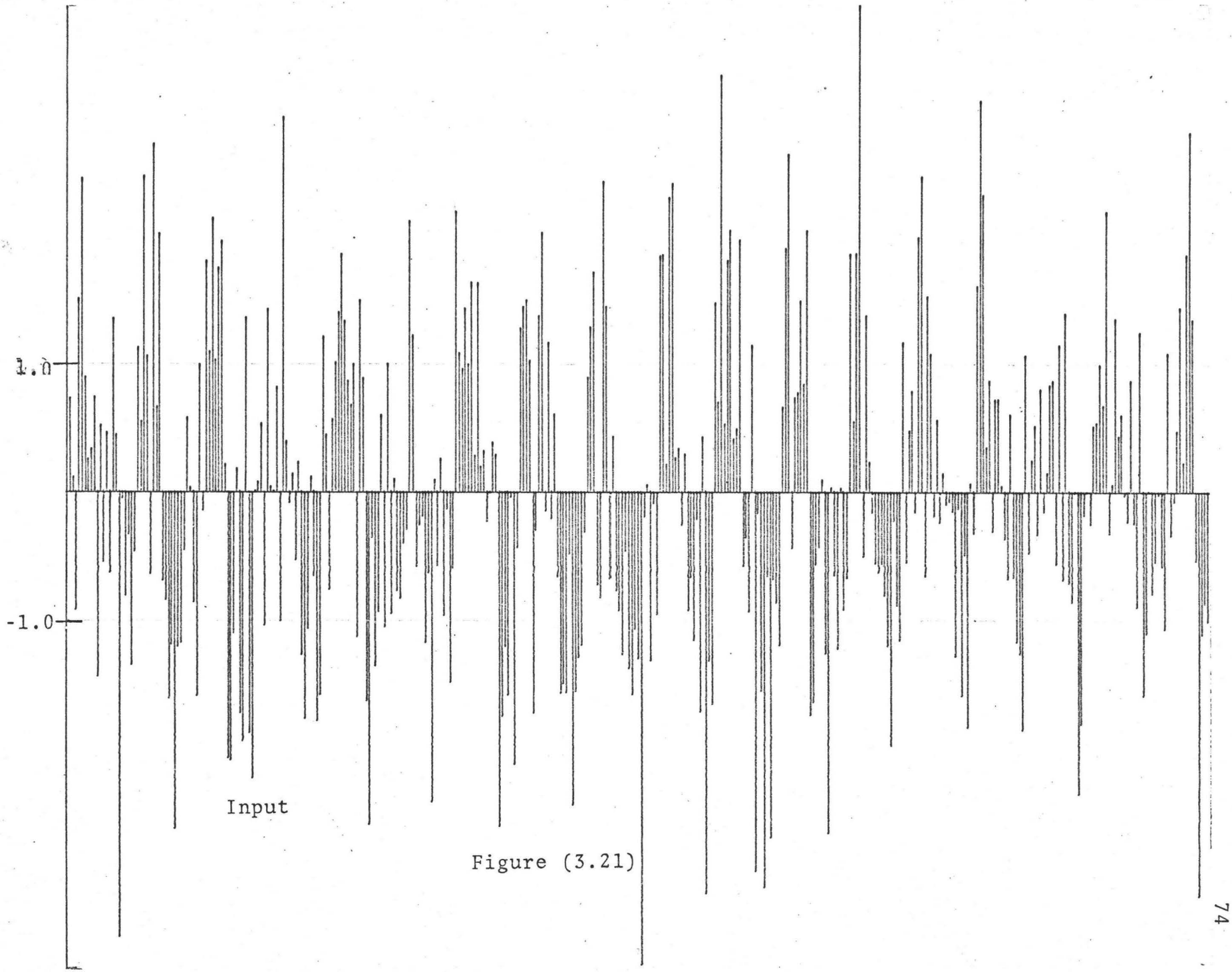


Figure (3.21)

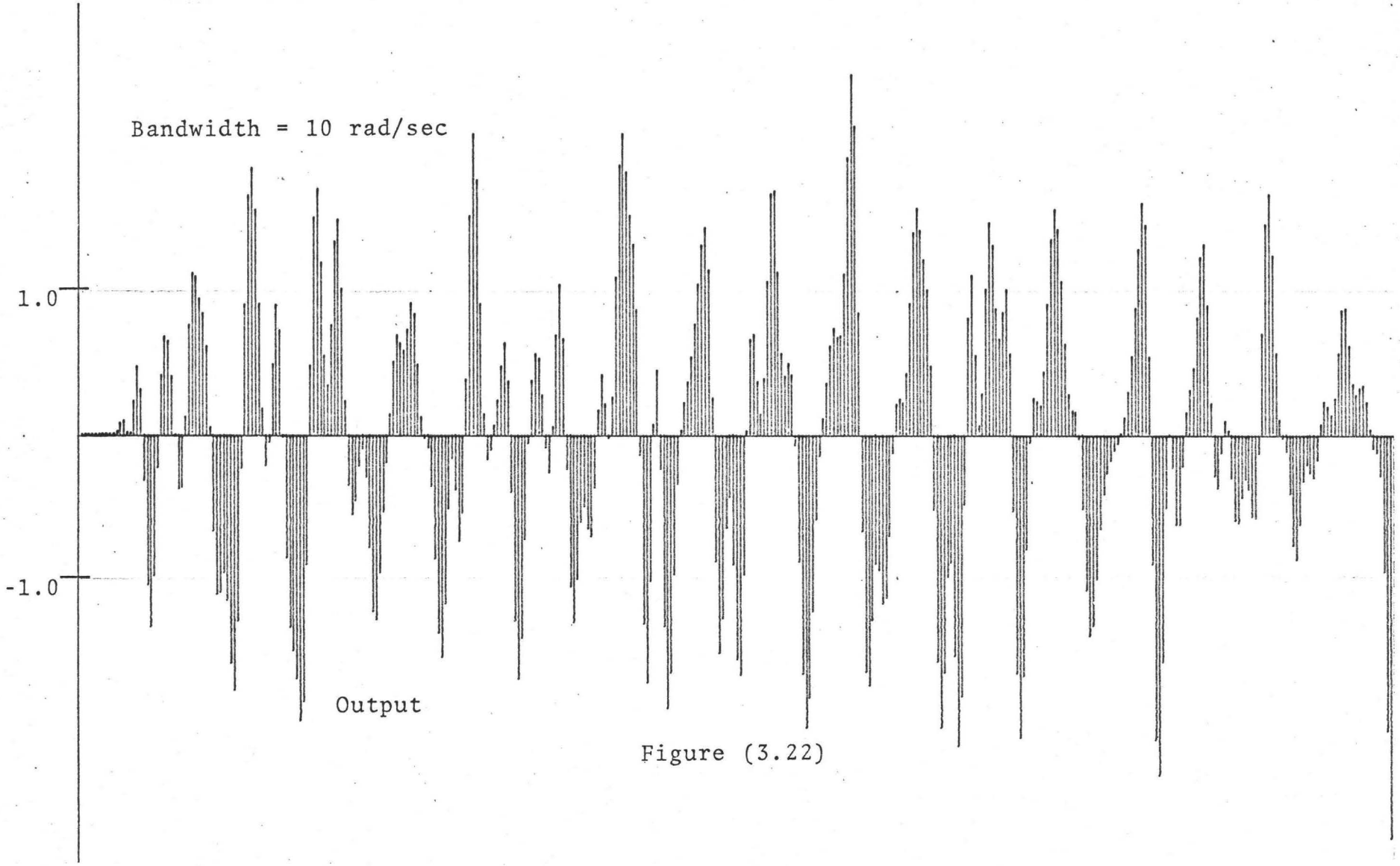


Figure (3.22)

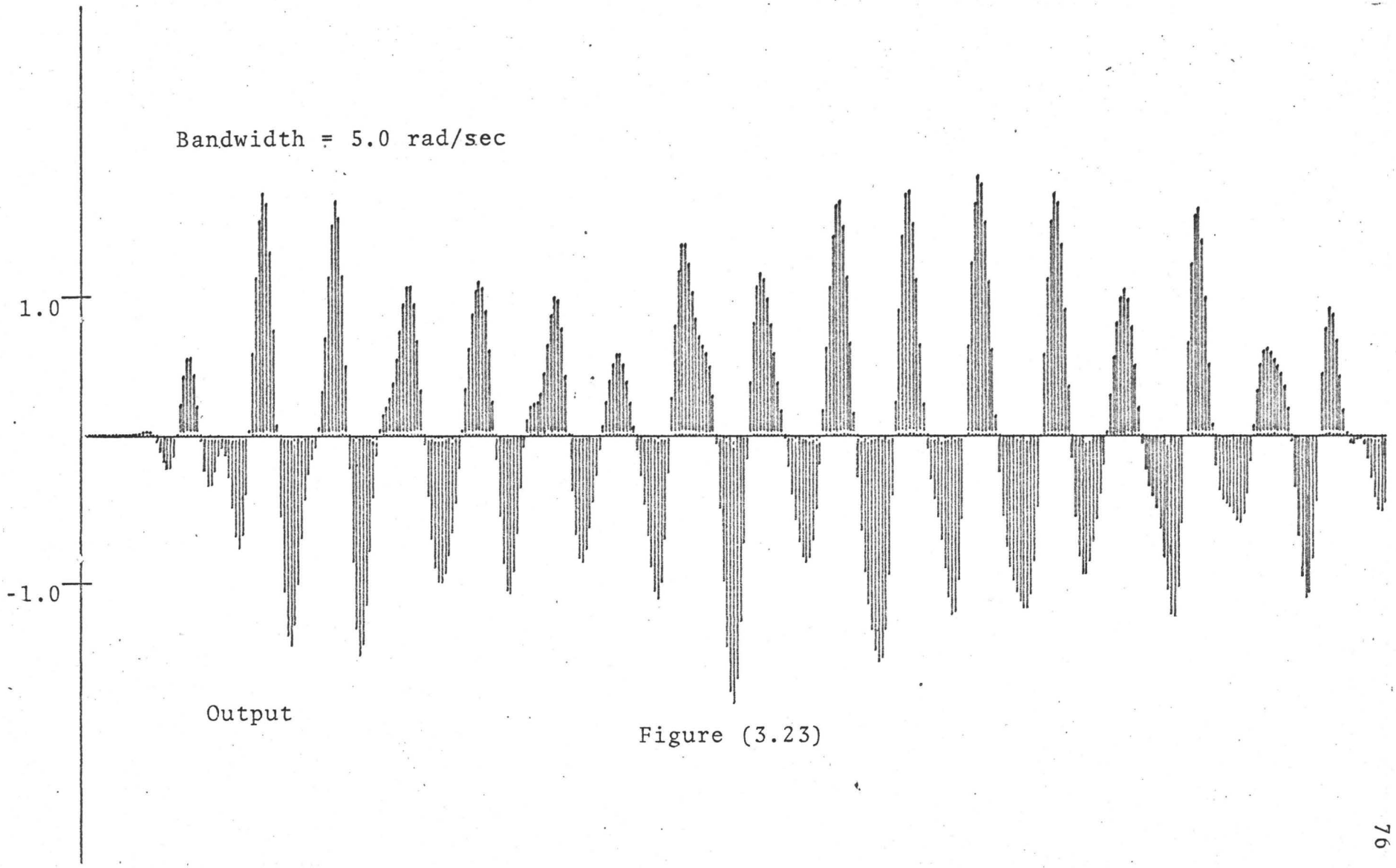


Figure (3.23)

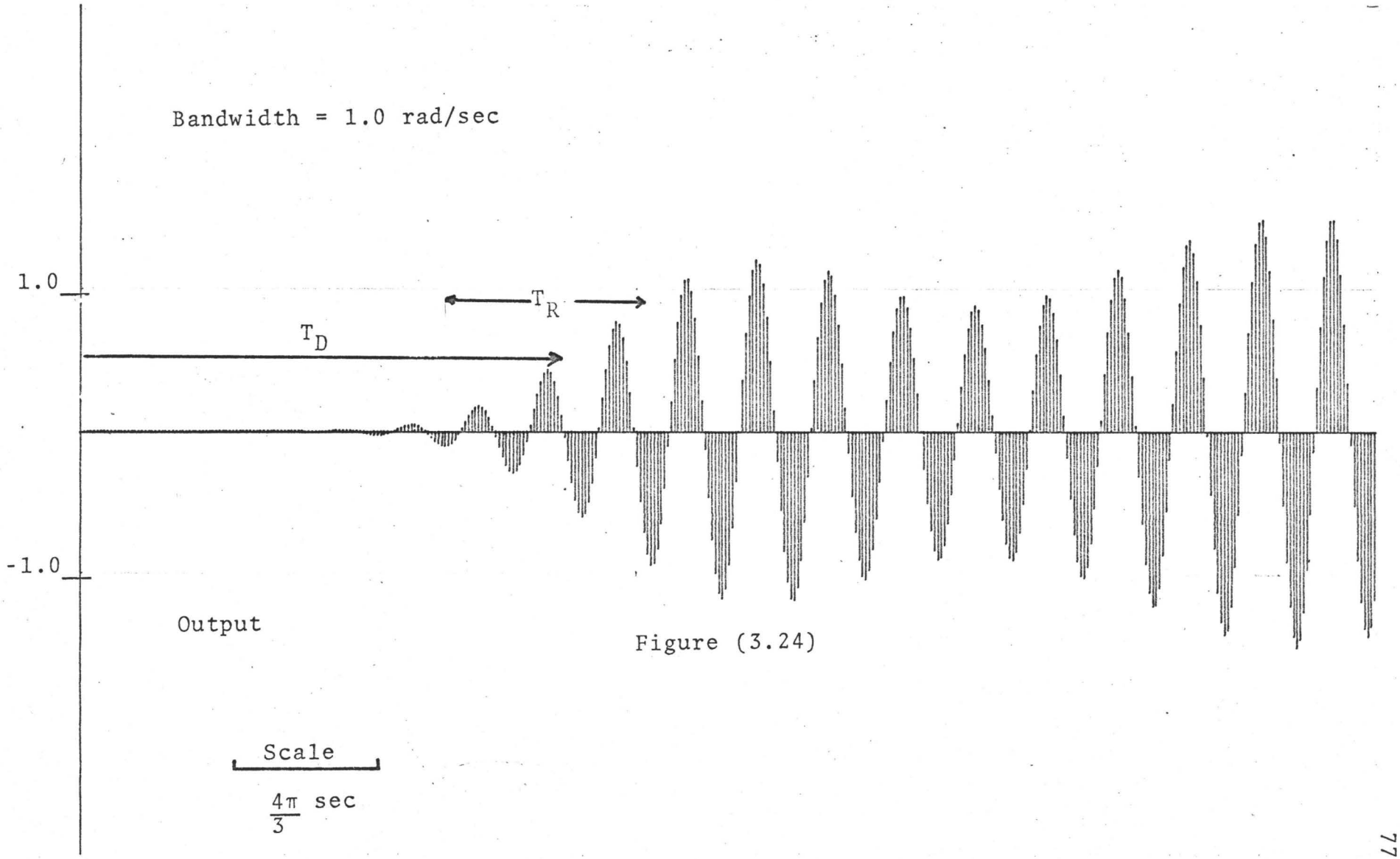


Figure (3.24)

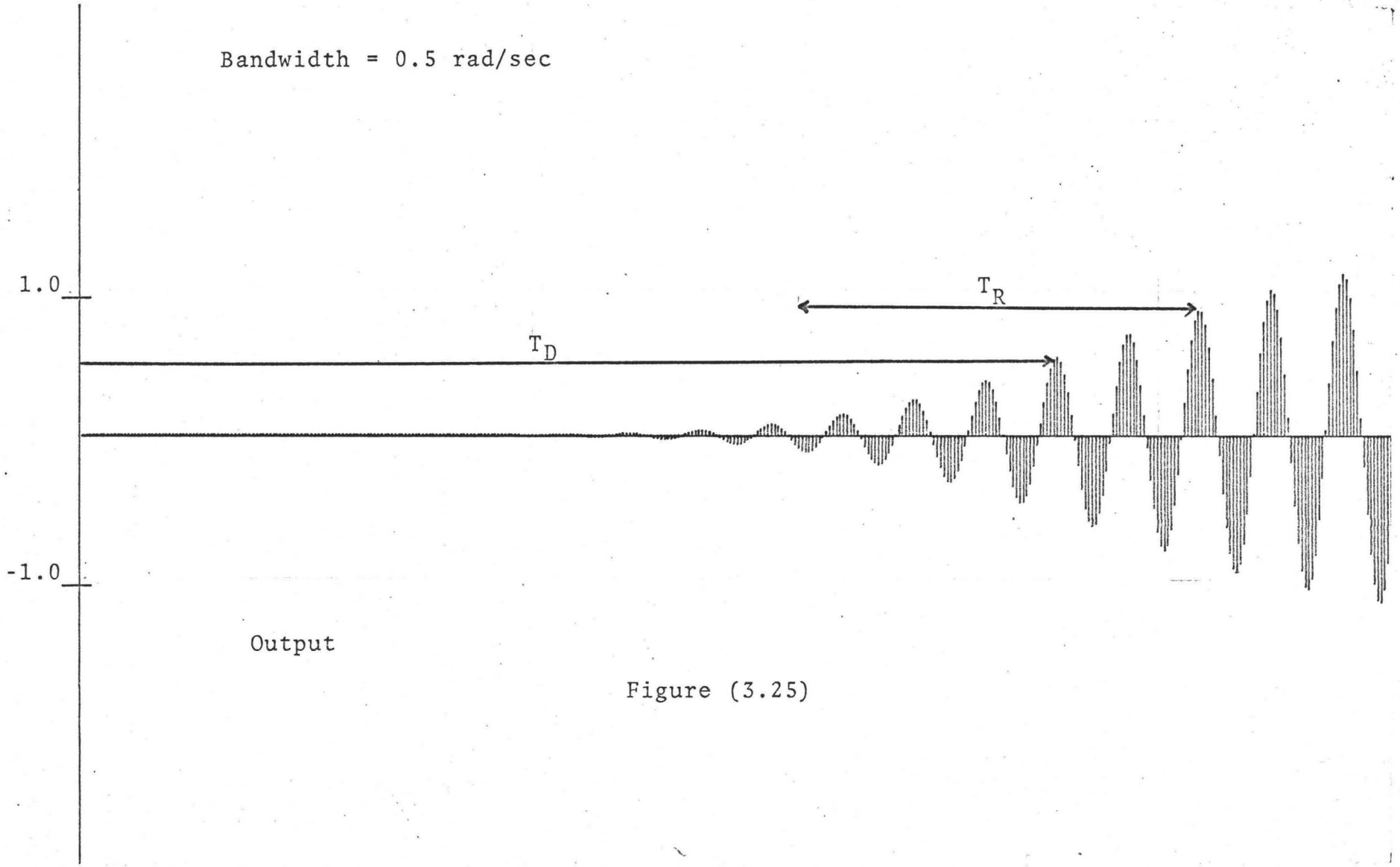


Figure (3.25)

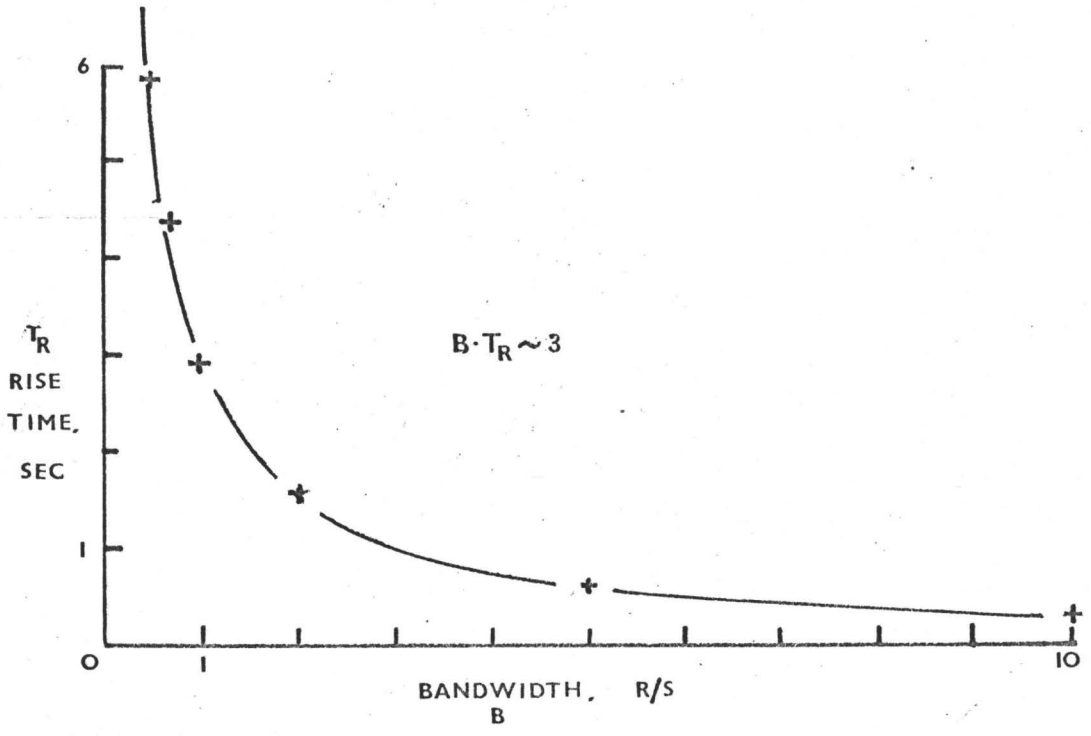


Figure 3.26(a)

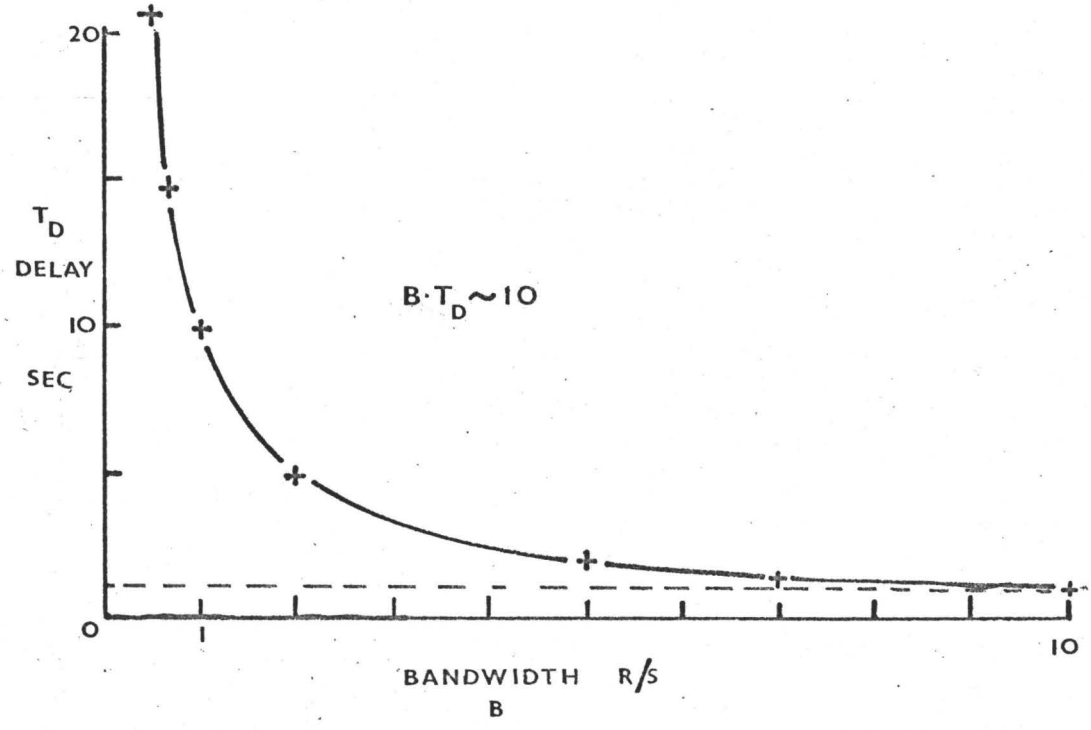


Figure (3.26b)

analog counterpart.

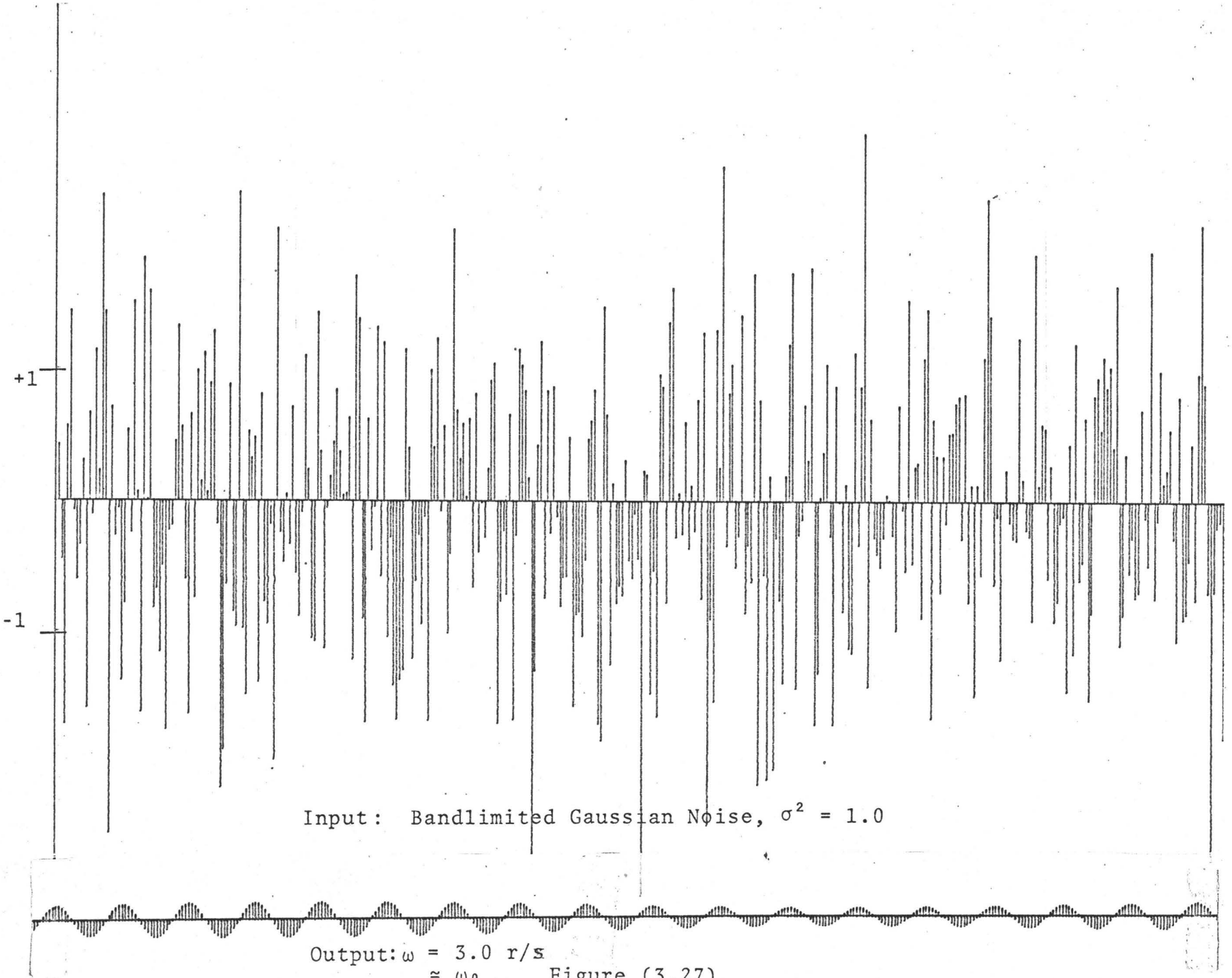
Figure 3.27 shows the output of the same bandpass filter [but using B = 0.2 rad/sec] for an input of bandlimited Gaussian noise only . Evidently, the filter has selectively passed only a small portion of the frequency spectrum of the noise, centred around 3.0 rad/sec resulting in an output which closely approximates a sinusoid of frequency 3.0 rad/sec. Again, the filter functions as a good approximation to its analog counterpart, so we can conclude that the basic transformation is valid. Since the power [i.e. the mean square value] of the random function is just the variance which was chosen to be $\sigma^2 = 1.0$, then the power associated with the output signal should be just the fraction of the total bandwidth which the filter passes [bounded by the 3 dB levels]. This is approximately

$$\frac{B}{\omega_{Nyquist}} \times \sigma^2 = \frac{0.2}{10\pi} \times 1.0$$

$$\simeq 0.0064 \quad \text{[watts]} ,$$

and hence, the rms value of the sine wave at the output is 0.08 [volts]. In comparison, the measured value is in the range 0.075 - 0.085, the variation resulting from the small but finite bandwidth of the filter.

There is a point to mention concerning the "noise" used in the above experiments. The computer subroutine produces normally distributed numbers ($\mu = 0$, $\sigma^2 = 1$)



Input: Bandlimited Gaussian Noise, $\sigma^2 = 1.0$

Output: $\omega = 3.0$ r/s
$\simeq \omega_0$

Figure (3.27)

whose autocorrelation function,

$$\phi(nT) = \frac{1}{2n + 1} \sum_{k=-n}^{+n} f_1(kT) f_1(nT + kT)$$

calculated over a few thousand samples, closely approximates that for "white" noise. However, the sampler which operates at a frequency $\frac{1}{T}$ hz, effectively bandlimits this white noise to $\frac{1}{2T}$ hz before the sequence enters the filter proper. The filter then shapes the spectrum of this new noise input.

3.5 Summary of Results

From the experimental results, we have observed the following points about the convolution-approximation method:

[1] the digital cascade realization provides very good approximations to all the basic analog filter characteristics, excepting the gainbands in the highpass and bandstop filters;

[2] the cascade realization is the only acceptable one for highpass digital filters, and is definitely superior to the parallel version, for bandstop filters;

[3] the parallel realization gives slightly better results when applied to lowpass and bandpass filters, than the cascade method does;

[4] in many cases, these digital filters perform in the same way that their analog counterparts do.

Chapter 4 Extensions and Improvements

4.1 Introduction

In this concluding chapter, we make several suggestions regarding improvements and extensions to this work. The most important result is that the impulse invariant transformation (Section 1.5) can be derived as a special case of the C-A method. Furthermore, the inability of the impulse invariant method to produce satisfactory highpass and bandstop filters is given a physical explanation. A first order approximation of the convolution integral and a consistent "impulse sequence" for digital filters are also derived.

4.2 Derivation of the Impulse Invariant Transformation

The original approximation of the convolution integral (Section (2.3)) has the effect of a zero-order hold, so that by the time the sampled value is used in a calculation, it is T seconds old; i.e. $x(\tau) = x(nT-T)$ is used at time nT. It is equally valid, and evidently more up-to-date to make the approximation $x(\tau) = x(nT)$ and to use this value immediately at time nT. Figure (4.1) shows the difference graphically. Doing this, we find that the basic lowpass discrete transfer function corresponding to $H(s) = 1/(s + p_i)$ is

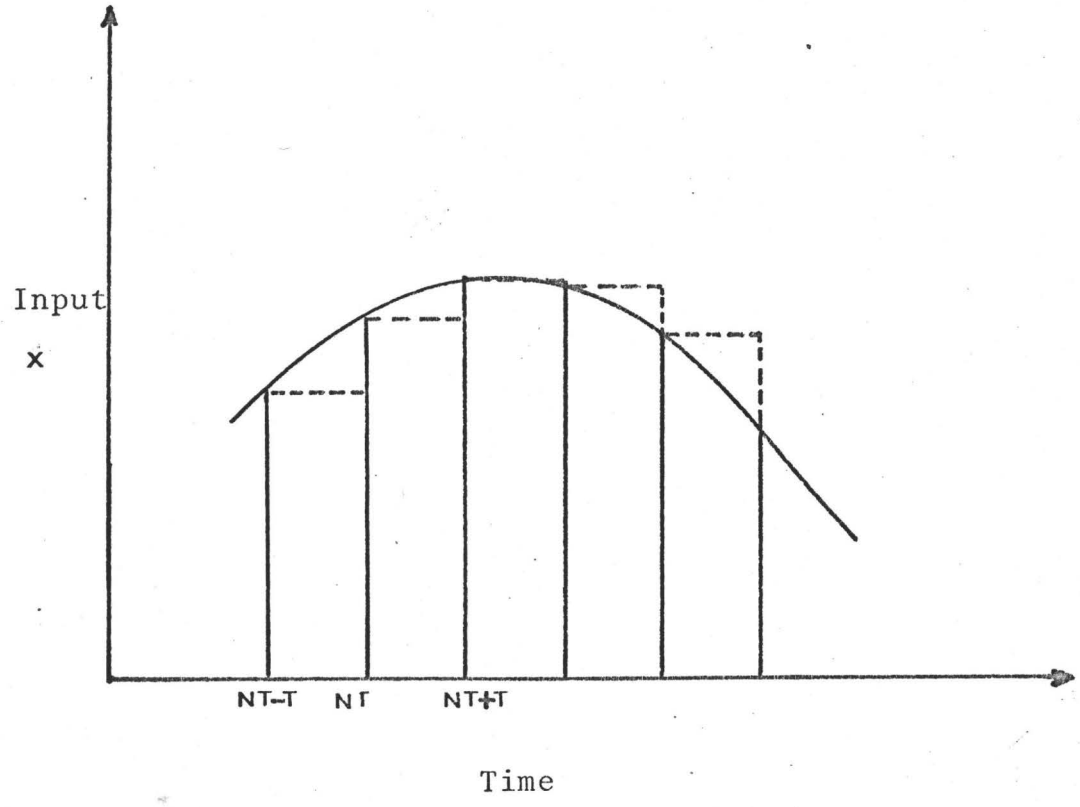
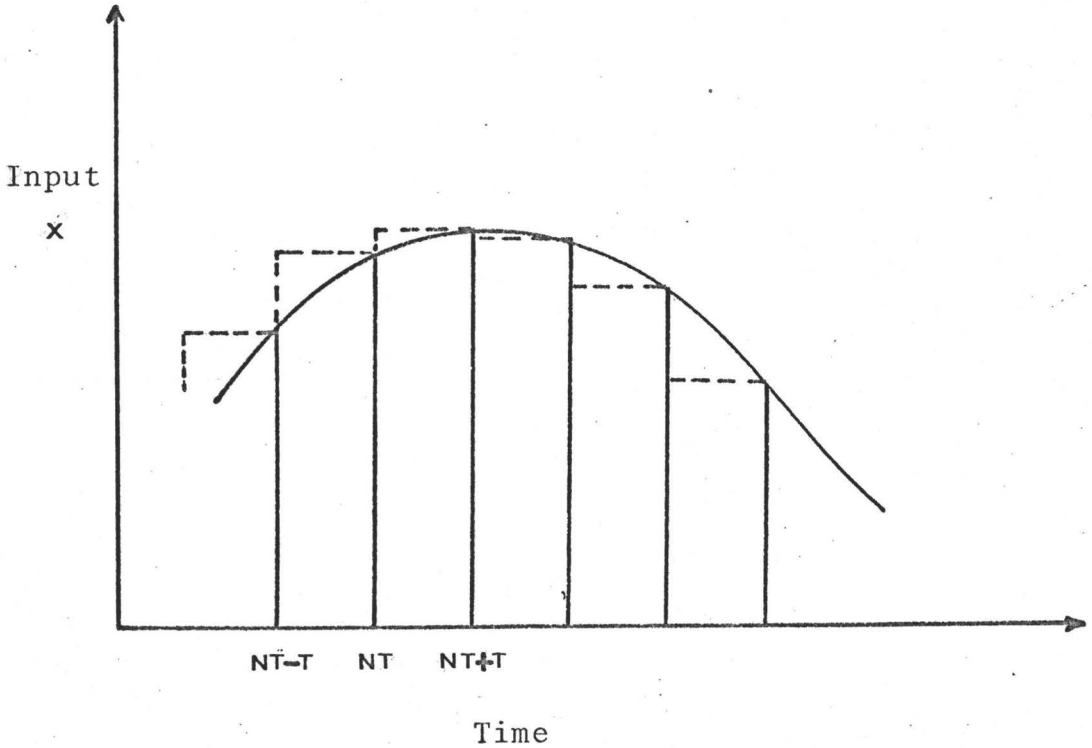


Figure (4.1)



$$H(z^{-1}) = \left(\frac{1 - e^{-p_i T}}{p_i}\right) \frac{1}{1 - e^{-p_i T} z^{-1}}$$

$$= \frac{K_i}{1 - e^{-p_i T} z^{-1}} \qquad (4.1)$$

The only difference between Equations (4.1) and (2.10) is the complex delay factor, z^{-1}, in the numerator of Equation (2.10). It is important to note that except for the constant K_i, Equation (4.1) is just the impulse invariant transform of $1/(s + p_i)$. Furthermore, we shall show in Section 4.3 that this constant overcomes the difficulty of large gains usually encountered in impulse invariant digital filters.[4]

The response characteristics derived from Equation (4.1) for lowpass and bandpass filters are virtually identical with those in Figures (3.2)and(3.3) and therefore they are not included here. However, with the highpass and bandstop filters we meet a problem which is similar to that illustrated by Figure (2.9). In that case, we showed that a half delay in the direct link led to useable highpass and bandstop characteristics. In this case, we find both experimentally and from theory that the solution is a half advance in the direct link; but such a solution is not realizeable as a difference equation (even after ridding the expression of $z^{1/2}$)since future information is necessary to calculate the output at

any instant. This restriction in the time domain complements the restriction in the frequency domain (Section 1.4).

It is evident now, that although the impulse-invariant and C-A transformation are closely related, the latter has the distinct advantage of being able to realize highpass and bandstop digital filters, as well as lowpass and bandpass.

4.3 First Order Approximation of the Convolution Integral

The accuracy of the convolution method can be increased by improving the approximation of the integral, $\int_{nT-T}^{nT} e^{p_i\tau} x(\tau)d\tau$. So far we have approximated $x(\tau)$ by the first term in its Taylor expansion, by setting $x(\tau) = x(nT-T)$; Figure (4.2) shows how the approximation can be extended by using the first two terms to obtain: [Figure (4.2) is on Page 52]

$$x(\tau) = x(nT-T) + \left\{\frac{x(nT-T) - x(nT-2T)}{T}\right\}[\tau - (nT-T)] \qquad (4.1)$$

The right side of Equation (4.1) which indicates that $x(\tau)$ is now approximated as a ramp, can be inserted into the integral and the expression reduced as was done in Equations (2.7) and (2.8a). The final form corresponding to Equation (2.10) is:

$$H(z^{-1}) = \frac{Y(z^{-1})}{X(z^{-1})} \qquad (4.2)$$

$$= \frac{z^{-1}e^{-p_iT}}{1 - e^{-p_iT}z^{-1}} + \frac{z^{-1}(1 - z^{-1})(e^{-p_iT} - [1 - p_iT])}{(1 - e^{-p_iT}z^{-1})p_i^2T}.$$

This form emphasizes that the effect of adding more terms to the Taylor expansion of $x(\tau)$ is to do the same to the discrete transfer function $H(z^{-1})$.

Obviously, this approach can be extended to higher order approximations only if the higher derivatives can be calculated accurately enough. These approximations involve more and more of the past inputs, but only the immediate past output explicitly. For example, we approximate the second derivative as

$$\frac{x(nT-T) - 2x(nT-2T) + x(nT-3T)}{T}.$$

4.4 The Transition from Analog to Discrete

We have already seen in Chapter 2 that there are important subtleties in the transition from analog to discrete descriptions of filters. To assure ourselves that results such Equation (2.10) or (4.2) are reasonable counterparts of the original, $1/(s + p_i)$, without actually calculating the response curves, a basic test is available.

The best test of a result in the digital domain is to see if it leads in the limit, as $T \to 0$, to the corresponding result in the analog domain. For example, the Nyquist frequency, $\frac{\pi}{T}$, which limits the useable range of any digital filter, recedes toward infinity as $T \to 0$. Thus in the limit, a non-repeating response characteristic is attained, identical to the analog characteristic. Of course, the input and output signals are now continuous too. Further, the basic C-A transformation (Equation (2.10)) bears out this idea: setting $z^{-1} = e^{-sT}$ and applying 1'Hôpital's rule, $H(z^{-1})$ reduces exactly to the original $H(s)$. However it is common in the literature [2] to find that the digital equivalent to a unit impulse is the sequence $\{1,0,0,0,\ldots\}$. This is obviously independent of T, the sampling period, so that there is no continuity between the digital and analog expressions of the unit impulse.

To rectify this inconsistency, consider the difference equation derived from Equation (2.10):

$$y(nT) = [e^{-p_i T}]y(nT-T) + [\frac{1}{p_i}(1 - e^{-p_i T})]x(nT-T) .$$

Table (4.1) shows the response of this digital filter to an input sequence

$$x(nT) = \{A,0,0,0,..\}$$

with all initial conditions assumed zero:

Time	Output y(nT)	Input x(nT)
0	0	A
T	$\frac{1}{p_i}(1 - e^{-p_iT})A$	0
2T	$\frac{1}{p_i}(1 - e^{-p_iT})e^{-p_iT}A$	0
3T	$\frac{1}{p_i}(1 - e^{-p_iT})e^{-2p_iT}A$	0
.	.	
.	.	

Table (4.1)

The response has the form $K_i e^{-(n-1)p_iT}$. (4.3)

Now since both the real part of p_i and T are small, the condition $Re[p_i]T << 1$ holds and the constant $(1 - e^{-p_iT})/p_i$ reduces to T for all poles p_i. Equation (4.3) becomes

$$y(nT) = ATe^{-(n-1)p_iT}, \quad n > 1 \tag{4.4}$$

If the impulse response is to be the sampled version of the impulse response of the corresponding analog filter, then it is evident that the value of A is $\frac{1}{T}$, and the input sequence corresponding to a unit impulse should therefore be

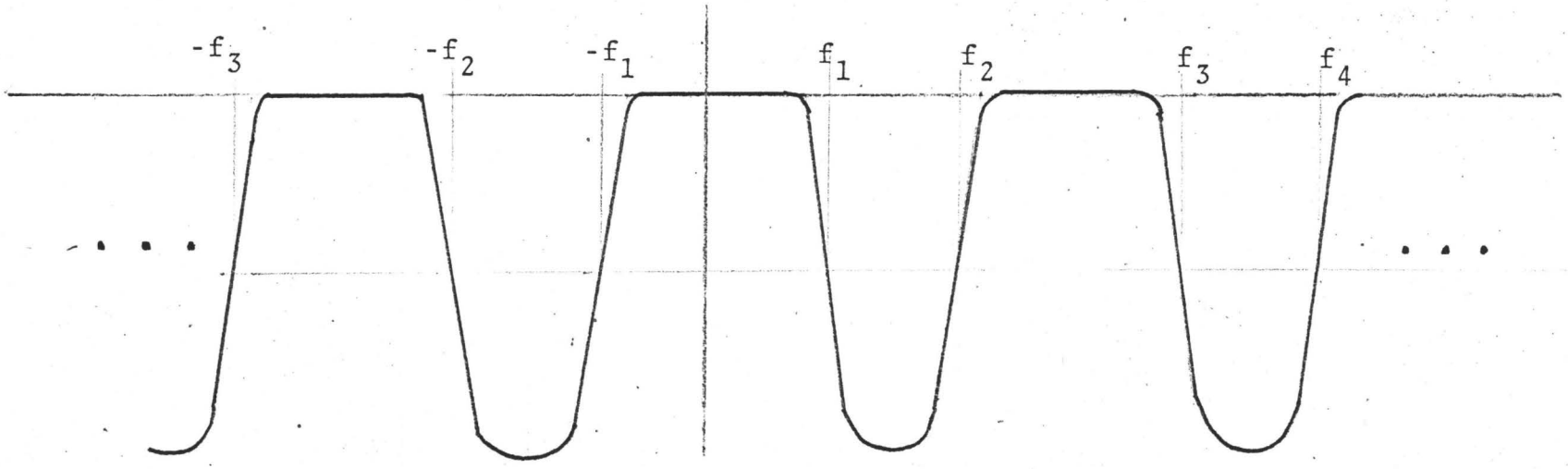
$x(nT) = \{\frac{1}{T}, 0, 0, \ldots\}$. Such a conclusion is also a direct result of our assumption that $x(\tau)$ is constant over

the sampling period T. Since the strength of an impulse, $\delta(t)$ is determined by the area integral $\int_{\infty}^{\infty} \delta(t)dt$, then the discrete equivalent of a unit impulse over T seconds is just $\frac{1}{T}$. Furthermore, there is the continuity we sought, since

$$\lim_{T \to 0} \{\frac{1}{T}, 0, 0, \ldots\} = \delta(t) .$$

4.5 The Periodic Nature of the Discrete Transfer Function

The fact that the discrete transfer function $H(z^{-1})$ is periodic can lead to misunderstanding of the digital filter's capabilities. To illustrate, suppose we have a (fictitious) analog filter with an infinitely repeating characteristic, just as a digital filter has (Figure 4.3). Depending on the period, certain regularly spaced frequencies in the analog input would be attenuated in exactly the same way, and these frequencies would then appear at the output. The difference in the digital case is that the frequency of a digital signal is determined by its envelope, so that the maximum frequency which can be represented unambiguously occurs when the samples are alternately positive and negative. In other terms, if at least two samples per cycle are taken (on the average), the original analog frequency is retained in the digital signal. However, when less than two samples per cycle are taken, the analog frequency is lost. In fact, the digital frequency which is

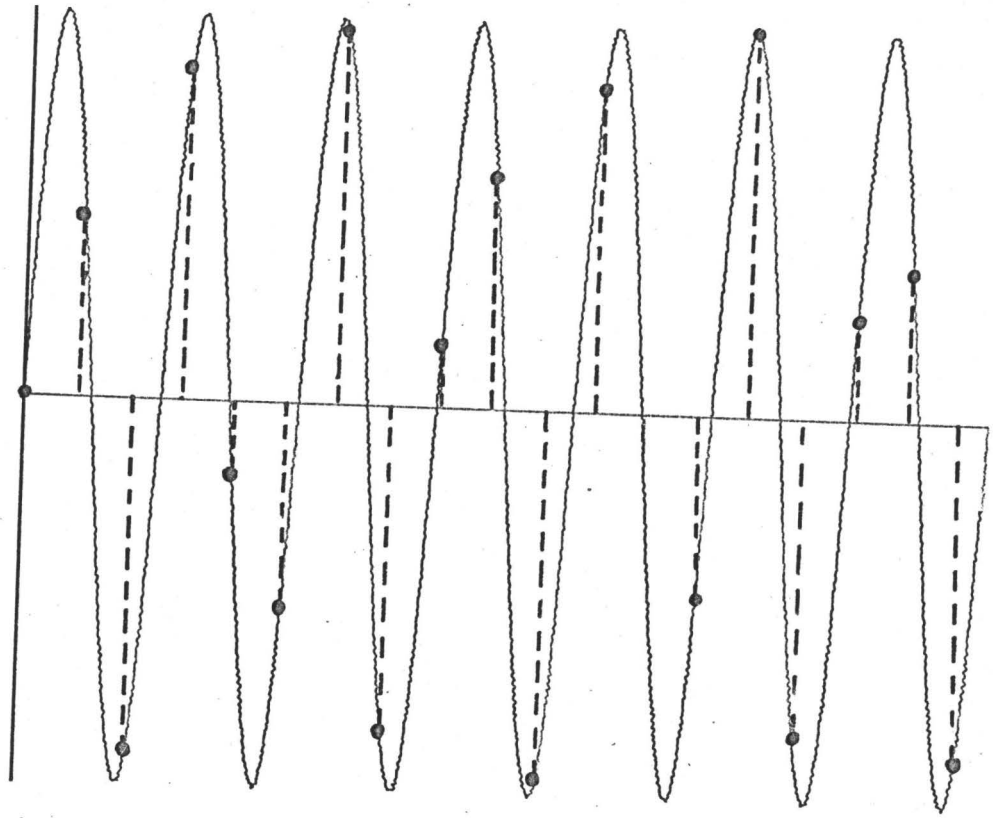


Infinitely Repeating Gain Characteristic

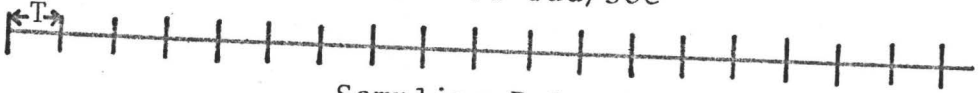
Figure (4.3)

produced is less than the Nyquist frequency by an amount equal to the difference between the frequency of the analog signal and the Nyquist rate. (A proof of this is given in Appendix III) For example, if the Nyquist rate is 10π rad/sec and the input analog signal has frequencies of 13π rad/sec and 7π rad/sec, then the input and output digital signals will have a frequency of 7π rad/sec only. Figure (4.4) illustrates the situation. The important point to observe is that the output digital signal is strictly limited to the baseband $0 - \frac{\pi}{T}$ rad/sec regardless of the analog input frequency. This effect is known as aliasing [1,2,3].

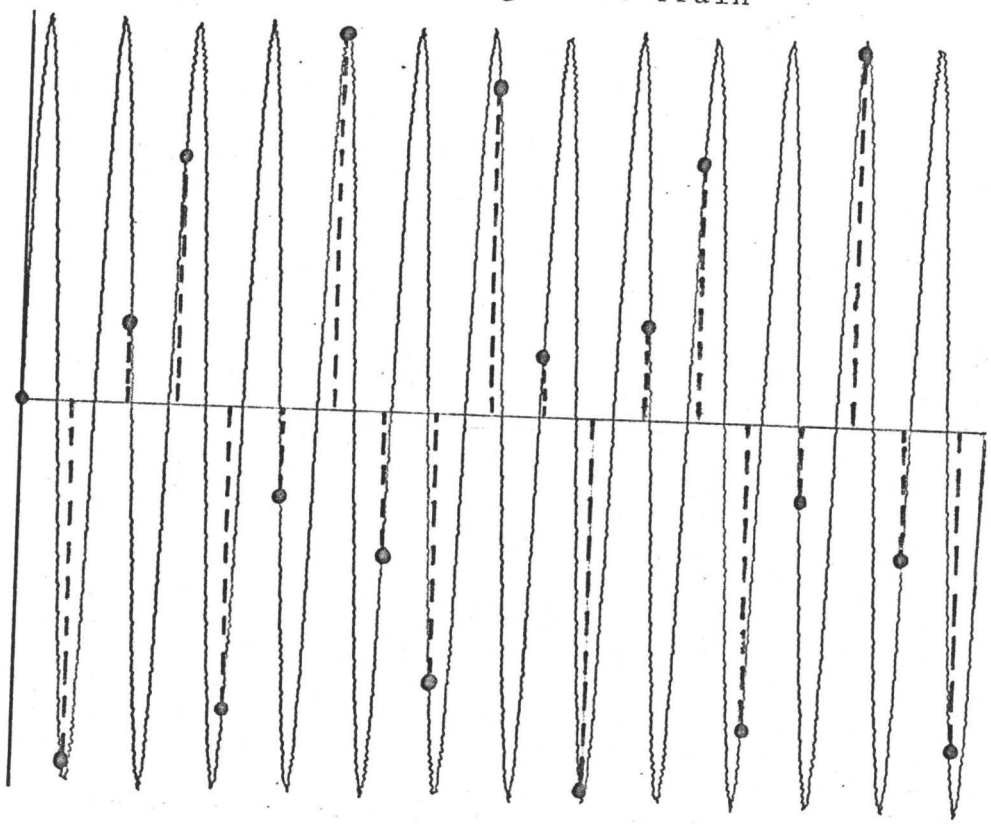
One straightforward use of this frequency shifting property of sampling is discrimination of an amplitude modulated signal. For example, suppose we have an A.M. signal with a carrier of 1000 khz and an audio spectrum from 100 to 7500 hz. Therefore, the sampling rate should be at least 15000 hz, but also 1000 khz should be an integer multiple of the exact sampling rate. Simple considerations show that a sampling rate of 15.15 khz satisfies both constraints, and the result is a sequence with frequencies in the baseband 100 to 7500 hz. Digital to analog conversion would then recover the original audio signal.



$\omega = 7\pi$ rad/sec



Sampling Pulse Train



$\omega = 13\pi$ rad/sec

Figure (4.4)

APPENDIX I Formulation of the Direct Method

In this Appendix, we consider the direct formulation of digital filters along the same lines as the parallel and cascade which this thesis has developed. The direct approach amounts to the conventional state-space description commonly used in control systems. We outline the mathematics and find that although the direct method is not suitable for our purposes, the analysis does lead to the useful reduction of the state-space which has been used in this thesis.

A transfer function of the form (2.1) leads to a vector differential equation [9]

$$\underline{x}(t) = e^{At}\,\underline{x}(0) + \int_0^t e^{A(t-\tau)}\,Bu(\tau)\,d\tau$$

where e^{At} is the transition matrix formed from the coefficients of Equation (2.1). This equation can be discretized in exactly the same way as done in Equations (2.7) and (2.8) to obtain a vector difference equation

$$\underline{x}(nT) = e^{AT}\,\underline{x}(nT-T) + [e^{AT} - I]A^{-1}Bu(nT)$$; I is the unit matrix.

In order to realize this equation as a digital filter, the matrix coefficients must be evaluated. This involves a series approximation of e^{AT} and it is a time-consuming calculation especially for high order systems; furthermore,

since any NxN A-matrix has $(N-1)^2$ zeroes, a significant amount of calculation time is wasted manipulating them. For example, a 10x10 A-matrix has 81 zeroes.

Further complications arise in this approach when we wish to describe other than lowpass systems. First, it becomes a tedious job of algebraic manipulation to obtain the analog transfer function for highpass, bandpass and bandstop filters, since the calculation of the a_i and b_i is not readily programmable for the general case. Secondly, to obtain a vector difference equation of the above form for these filters requires four transformations.[9] Finally, there is a difficulty which pertains to even the lowpass formulation using the Direct method. To illustrate it, we consider the expansion of the above vector difference equation where the elements of e^{AT} are defined as ϕ_{ii} and the elements of $[e^{AT}-I]A^{-1}B$ are defined as ζ_i :

$$x_1(nT) = \phi_{11}x_1(nT-T)+..\phi_{1N}x_N(nT-T) + \zeta_1 u(nT-T)$$

$$x_2(nT) = \phi_{21}x_1(nT-T)+..\phi_{2N}x_N(nT-T) + \zeta_2 u(nT-T)$$

$$\vdots$$

$$x_N(nT) = \phi_{N1}x_1(nT-T)+..\phi_{NN}x_N(nT-T) + \zeta_N u(nT-T)$$

If now we apply the Z-transformation to each equation, the result is:

$$X_1(z^{-1}) = z^{-1}\,[\phi_{11}X_1 + \phi_{12}X_2 + \ldots + \phi_{1N}X_N\,] + \zeta_1 z^{-1}U$$

$$X_2(z^{-1}) = z^{-1}\,[\phi_{21}X_1 + \phi_{22}X_2 + \ldots + \phi_{2N}X_N\,] + \zeta_2 z^{-1}U$$

$$\vdots$$

$$X_N(z^{-1}) = z^{-1}\,[\phi_{N1}X_1 + \phi_{N2}X_2 + \ldots + \phi_{NN}X_N\,] + \zeta_N z^{-1}U$$

Immediately, we see that if the discrete transfer function $H(z^{-1})$ is desired, a great deal of extra calculation is necessary to extract X_1/U from these N equations.

All the above difficulties can be avoided if we set N=1, i.e. so that there are no zeroes in the A-matrix. This amounts to reducing Equation (2.1) to a product of N first order factors and treating each as an individual transfer function, although they are complex.

****A paper based on the material developed in this thesis, has been accepted for publication in the Proceedings of the Institute of Electrical Engineers.

```
REQUEST,TAPE10,HI.        PLOTTER TAPE
RUN(S)
SETINDF.
LGO.
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE10)
C
C
C    THIS PROGRAM IS A DIGITAL FILTER BASED ON A MODIFICATION OF
C    STATE SPACE ANALYSIS.  THE INPUT INCLUDES THE POLES OF A
C    LOWPASS FILTER. IF A TRANSFORMATION IS TO BE MADE TO HIGHPASS
C    BANDPASS OR BANDSTOP THEN THE BANDWIDTH AND CENTRE FREQUENCY
C    (OR CUTOFF FREQUENCY) ARE ALSO REQUIRED. THE ANALYSIS USES
C    COMRLEX ARITHMETIC.  REGARDLESS OF THE ORDER OF THE LOWPASS
C    OR TRANSFORMED FUNCTION, IT IS REDUCED TO A SET OF  CASCADED
C    FIRST ORDER FILTERS.  EACH CASCADED BLOCK MAY BE MADE UP OF
C    ONE,TWO OR THREE PARALLEL FILTERS, DEPENDING ON THE TYPE.
C
C
C
C
C              C A S C A D E
C
C
C
      DIMENSION PLP(100),C1(100),C2(100),R(100),P(100),HT(100)
      DIMENSION PLOTR(101),T1(100)
      DIMENSION Y(100),YY(100),YYY(100)
      DIMENSION X(100),XX(100),XXX(100)
      COMPLEX X,XX,XXX
      COMPLEX Y,YY,YYY
      COMPLEX HT,OMT,R,P,PLP,C1,C2,GAMMA,CC,CP,T1,F1,F2,F3
      COMPLEX V1,V2
      REAL LP
      DIMENSION HPLOT(2000,2)
      PI=3.14159265
      P2=2.0*PI
      OM=P2/6.0
      T=0.1
C
C
C
C    THE FOUR FILTER TYPES ARE READ IN.
C
      READ(5,555) LP,BP,BS,HP,AP
  555 FORMAT(1X,5A2)
C
C    THE LOWPASS POLES, PLP(J), ARE READ IN AND COUNTED.
C
      NPOLES=0
      GAMMA=1.0
      DO 5 K=1,100
      READ(5,7) PLP(K)
      IF(PLP(K).EQ.1000.0) GO TO 8
      WRITE(6,7) PLP(K)
    7 FORMAT(1X,2F15.8)
      GAMMA=GAMMA*PLP(K)
    5 NPOLES=NPOLES+1
    8 WRITE(6,9) NPOLES
    9 FORMAT(1X,1I3)
      NBLOX=NPOLES
```

```
C
C
C    THE TYPE OF TRANSFORMATION IS CHOSEN.  IN ALL CASES THERE WILL BE
C    'NBLOX' CASCADED SECTIONS.  WITHIN EACH SECTION THERE WILL BE
C    ONE(LP),TWO(BP OR HP) OR THREE(BS) PARALLEL SUBFILTERS.
C    THE NUMBER OF PARALLEL BRANCHES IS CALLED 'NPATHS'.
C
C
C    BW IS THE BANDWIDTH OF THE BP OR BS FILTERS. WO IS THE CENTRE
C    FREQUENCY FOR BS AND BP, OR THE CUTOFF FREQUENCY FOR LP AND HP.
C
C
      NPOLES=NBLOX
      READ(5,557) TYPE
  557 FORMAT(1X,1A2)
      WRITE(6,557) TYPE
      WO=3.0
      BW=1.0
      IF(TYPE.EQ.BP) GO TO 165
      IF(TYPE.EQ.BS) GO TO 166
      IF(TYPE.EQ.HP) GO TO 167
      IF(TYPE.EQ.AP) GO TO 169
C
C    THIS BLOCK DOES THE LP TO LP TRANSFORMATION.
C
      DO 499 K=1,NBLOX
      P(K)=PLP(K)*WO
      R(K)=WO
      C1(K)=CEXP(-P(K)*T)
  499 C2(K)=(1.0-C1(K))/P(K)
      NPATHS=1
      FACTOR=1.0
      GO TO 2222
C
C    THIS BLOCK DOES THE LP TO BP TRANSFORMATION.
C
  165 DO 1010 I=1,NBLOX
      I2=2*I
      I21=I2-1
      T1(I)=BW*PLP(I)/2.0
      P(I21)=T1(I)+CSQRT(T1(I)**2-WO**2)
      P(I2 )=T1(I)-CSQRT(T1(I)**2-WO**2)
      F1=BW*P(I21)
      F2=BW*P(I2)
      F3=P(I21)-P(I2)
      R(I21)=F1/F3
 1010 R(I2)=F2/F3
      NPATHS=2
      FACTOR=1.0
      NPOLES=2*NBLOX
      DO 1013 I=1,NPOLES
      C1(I)=CEXP(-P(I)*T)
 1013 C2(I)=(1.0-C1(I))/P(I)
      GO TO 2222
C
C    THIS BLOCK DOES THE LP TO BS TRANSFORMATION.
C
  166 BW=BW*2.0
```

```
      WO=WO*2.0
      DO 1011 I=1,NBLOX
      I2=I*2
      I21=I2-1
      T1(I)=BW/(2.0*PLP(I))
      P(I21)=T1(I)+CSQRT(T1(I)**2-WO**2)
      P(I2 )=T1(I)-CSQRT(T1(I)**2-WO**2)
      F1=(BW/PLP(I)**2)*P(I21)
      F2=(BW/PLP(I)**2)*P(I2)
      F3=P(I21)-P(I2)
      R(I21)=F1/F3
 1011 R(I2)=F2/F3
      NPATHS=2
      FACTOR=2.0
      NPOLES=2*NBLOX
      DO 1014 I=1,NPOLES
      C1(I)=CEXP(-P(I)*T)
 1014 C2(I)=(1.0-C1(I))/P(I)
      GO TO 2222
C
C  THIS BLOCK DOES THE LP TO HP TRANSFORMATION.
C
  167 BW=BW*2.0
      WO=WO*2.0
      DO 1012 I=1,NBLOX
      P(I)=WO/PLP(I)
      R(I)=WO/PLP(I)**2
      C1(I)=CEXP(-P(I)*T)
 1012 C2(I)=(1.0-C1(I))/P(I)
      NPATHS=1
      FACTOR=2.0
      GO TO 2222
  169 DO 1015 I=1,NBLOX
      C1(I)=CEXP(-PLP(I)*T)
      C2(I)=(1.0-C1(I))/PLP(I)
 1015 R(I)=2.0*PLP(I)
      NPATHS=1
      FACTOR=2.0
C
C
C   THIS SECTION INCREMENTS THE FREQUENCY AND CALCULATES
C   THE RESPONSE.
C
C
 2222 CONTINUE
      NFREQ=2
      NF=FACTOR
      NB=NBLOX+1
C
C
C   =NPB= IS THE NUMBER OF NON-DIRECT PATHS CONNECTING
C   INPUT TO OUTPUT.
C
C
      NPB=NPATHS*NBLOX
      DO 601 KK=1,NFREQ
      READ(5,55) OM,CYCLE
      WRITE(6,55) OM,CYCLE
```

```
   55 FORMAT(1X,2F15.6)
      MM=CYCLE*(P2/OM)/T
C
C    THE SYSTEM IS INITIALLY RELAXED.
C
      DO 69 K=1,NB
      XXX(K)=0.0
   69 XX(K)=0.0
C
      DO 44 I=1,NPB
      YYY(I)=0.0
   44 YY(I)=0.0
      DO 602 M=1,MM
      EM=M
      H=T*EM
      X(1)=SIN(OM*H)
   26 IF(FACTOR.EQ.2.0) GO TO 71
C
C
C    =FACTOR= IS A PARAMETER WHICH DETERMINES THE STRUCTURE
C    OF THE FILTER.  IT TAKES THE VALUE 1.0 FOR BP AND LP,
C    AND THE VALUE 2.0 FOR BS AND HP.
C
C
      DO 72 I=2,NB
      IF(TYPE.EQ.LP) GO TO 80
      L=2*I-3
      GO TO 81
   80 L=I-1
   81 DO 70 J=1,NPATHS
      INDEX=NF+J
      V1=C1(L)*YY(L)
      V2=R(L)*C2(L)*XX(I-1)
      Y(L)=V1+((-1.0)**INDEX)*V2
   70 L=L+1
      X(I)=0.0
      DO 72 K=1,NPATHS
   72 X(I)=X(I)+Y(L-K)
      GO TO 38
   71 DO 76 I=2,NB
      IF(TYPE.EQ.HP.OR.TYPE.EQ.AP) GO TO 82
      L=2*I-3
      GO TO 83
   82 L=I-1
      IF(TYPE.EQ.AP) GO TO 84
   83 X(I)=XX(I-1)/PLP(I-1)
      GO TO 85
   84 X(I)=XX(I-1)
   85 DO 74 J=1,NPATHS
      INDEX=NF+J
      V1=C1(L)*YYY(L)
      V2=R(L)*C2(L)*XXX(I-1)
      Y(L)=V1+((-1.0)**INDEX)*V2
   74 L=L+1
      DO 76 K=1,NPATHS
   76 X(I)=X(I)+Y(L-K)
   38 WRITE(6,96) X(NB)
   96 FORMAT(1X, 2E15.5)
```

```
      HPLOT(M,KK)=REAL(X(NB))
C
C
C    THIS BLOCK SHIFTS THE X AND Y VALUES BACK ONE UNIT
C    IN TIME. THIS IS THE MEMORY.
C
C
   37 DO 600 K=1,NB
      XXX(K)=XX(K)
  600 XX(K)=X(K)
      DO 602 K=1,NPB
      YYY(K)=YY(K)
  602 YY(K)=Y(K)
  601 CONTINUE
      CALL NAME
      CALL PLOT(4.0,5.0,-3)
      DO 124 KJ=1,NFREQ
      CALL PLOT(8.5,0.0,2)
      CALL PLOT(0.0,-2.5,3)
      CALL PLOT(0.0,+2.5,2)
      CALL PLOT(0.0,0.0,3)
      EMM=EM
      DO 123 I=1,MM
      EM=I
      X=EM*8.5/EMM
      Y=HPLOT(I,KJ)
      CALL PLOT(X,0.0,3)
      CALL PLOT(X,Y,2)
  123 CONTINUE
  124 CALL PLOT(15.0,0.0,-3)
      CALL PLOT(0.0,0.0,999)
      STOP
      END
      SUBROUTINE NAME
      CALL PLOT(1.,4.,3)
      CALL  LETTER(14,10,90,1.,4.,14HCARNEGIE A4107)
      CALL PLOT(10.0,0.5,-3)
      RETURN
      END
         6400 END RECORD
 LPBPBSHPAP
  0.1564345        .9876883
  0.1564345       -.9876883
  .4539905         .8910065
  .4539905        -.8910065
  .7071068         .7071068
  .7071068        -.7071068
  .8910065         .4539905
  .8910065        -.4539905
  .9876883         .1564345
  .9876883        -.1564345
   1000.0          0.0
 BS
            3.0          40.0
       1.0               10.0
```

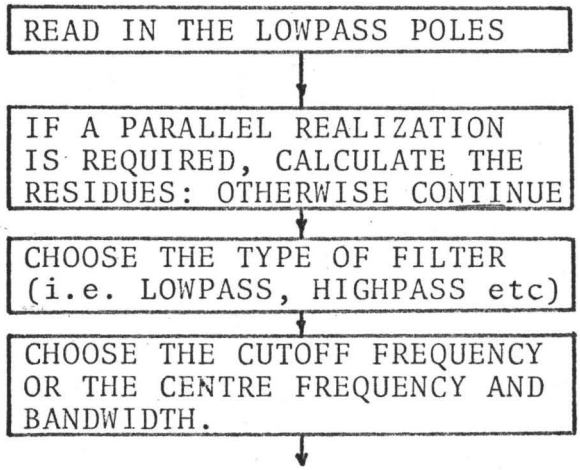
CD TOT 0288

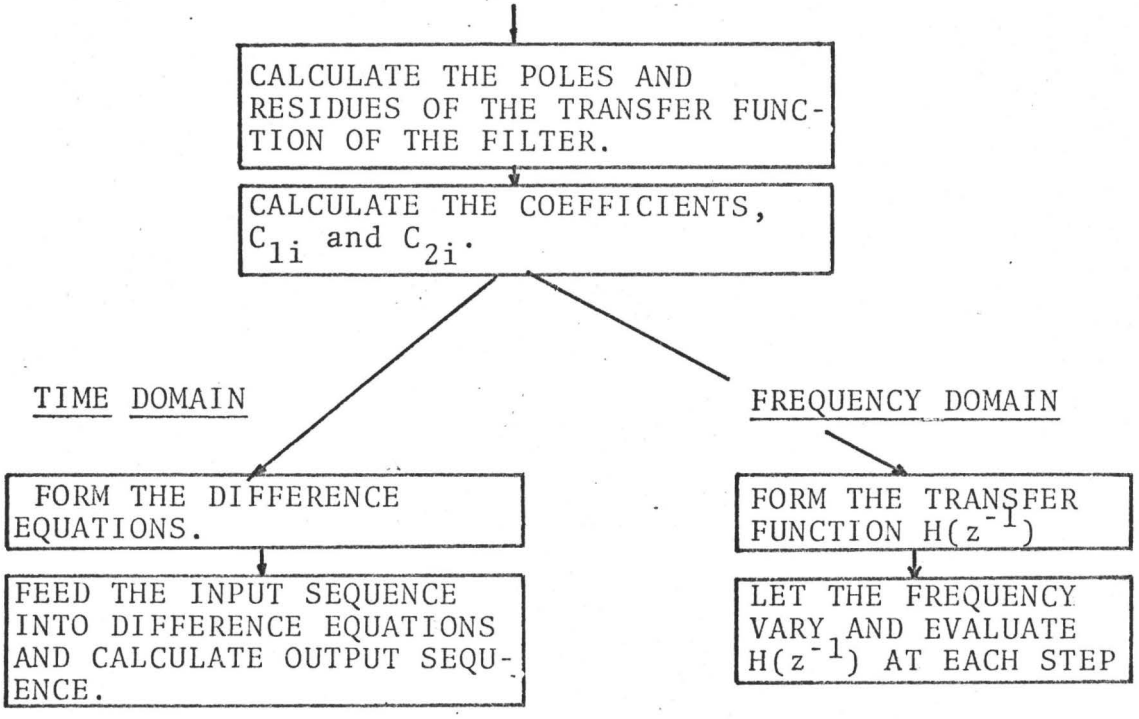
APPENDIX II Description of the Programs

The description of the actual programs which were used to illustrate the above theory will be carried out in two parts: first, we present a simple flow chart to establish the general nature of the programs with a minimum of complication; second, the actual programs, which are divided into blocks by explanatory COMMENT cards, will follow. They are written in Fortran IV.

The flow diagrams for both time and frequency domains are shown below. They have a major section of calculations in common, with only the last two blocks different. The only major difference between the cascade and parallel programs is the calculation of the residues of the original lowpass filter for the parallel case. Only the cascade programs are included.

Flowchart:





The results can then be accumulated and automatically graphed (see the actual program for details) or printed out.

APPENDIX III Basis of Folding Effect

Consider for ease, the analytic continuous signal

$$x(t) = e^{j\omega t}$$

which is sampled at the rate $\frac{1}{T}$ hz, to obtain a sequence

$$x(nT) = e^{j\omega nT} \quad , n = 0,1,2,... \qquad \text{AIII1}$$

Suppose that the frequency ω, is written as an integral multiple of the Nyquist frequency, plus a fraction:

$$\omega = \frac{m\omega_s}{2} \pm \Omega_x \quad , \text{where } \Omega_x \text{ is the remainder, m an integer.}$$

Substituting this into AIII1 we get:

$$x(nT) = e^{jnT\left[\frac{m\omega_s}{2} \pm \Omega_x\right]}$$

$$= e^{jk\pi} e^{j\Omega_x nT}$$

$$= \pm e^{j\Omega_x nT} \quad , \text{where } k=mn \text{ an integer and } \omega_s T = 2\pi \ .$$

It is obvious now that whatever the frequency of the analog input, the frequency of the digital sequence is limited to the baseband, i.e. $0 \leq \omega \leq \frac{\omega_s}{2}$

LIST OF REFERENCES

1 Kaiser, J.F. : "System Analysis by Digital Computer", Chapter 7, John Wiley and Sons, Inc., 1966.

2 Gold, B., and Rader, C.M.: "Digital Processing of Signals" Chapter 1, McGraw Hill Book Co., 1969.

3 Golden, R.M. and Kaiser, J.F. : "Design of Wide-band Sampled-Data Filters", Bell System Technical Journal, July, 1964, Page 1534.

4 Rader, C.M. and Gold, B. : "Digital Filter Design Techniques in the Frequency Domain", Proc. IEEE, Vol. 55, No. 2, Feb. 1967.

5 Jackson,L.B., Kaiser, J.F. and McDonald, H.S. : "An Approach to the Implementation of Digital Filters", IEEE Transactions on Audio and Electroacoustics, Vol. AU-16, No. 3, Sept. 1968.

6 Weinberg, L. : "Network Analysis and Synthesis", McGraw-Hill Book Co.

7 Golden, R.M. : "Digital Filter Synthesis by Sampled Data Transformation", IEEE Transactions on Audio and Electroacoustics, Vol. AU-16, No. 3, Sept. 1968.

8 Kuo, F.F. : "Network Analysis and Synthesis", 2nd Edition, John Wiley and Sons, Inc.

9 Ogata, K. : "State Space Analysis of Control Systems" , Prentice-Hall, Inc.