PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark ✓.

1. Glossy photographs _____
2. Colored illustrations _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Print shows through as there is text on both sides of page _____
6. Indistinct, broken or small print on several pages _____ throughout _____
7. Tightly bound copy with print lost in spine _____
8. Computer printout pages with indistinct print ✓
9. Page(s) _____ lacking when material received, and not available from school or author _____
10. Page(s) _____ seem to be missing in numbering only as text follows _____
11. Poor carbon copy _____
12. Not original copy, several pages with blurred type _____
13. Appendix pages are poor copy _____
14. Original copy with light type _____
15. Curling and wrinkled pages _____
16. Other _____

University Microfilms International

300 N. ZEEB RD., ANN ARBOR, MI 48106 (313) 761-4700

THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ERROR ANALYSIS OF DIGITAL FILTERS WITH LOGARITHMIC NUMBER SYSTEM

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

BY

TOMIO KUROKAWA

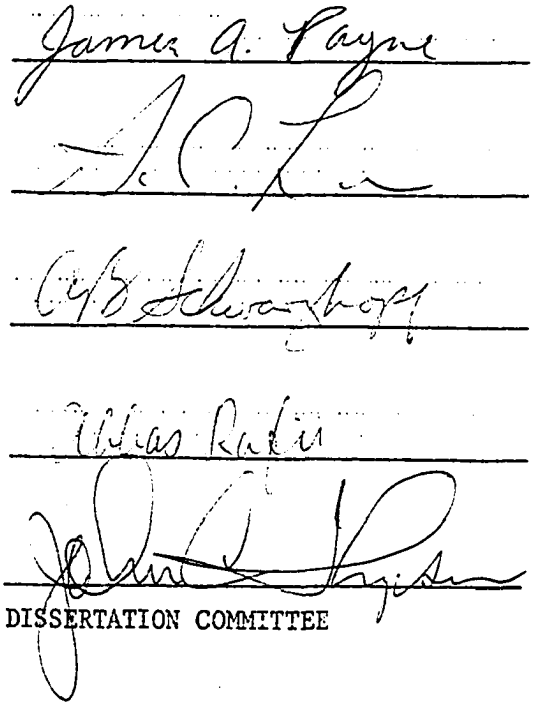
Norman, Oklahoma

1978

ERROR ANALYSIS OF DIGITAL FILTERS

WITH LOGARITHMIC NUMBER SYSTEM

APPROVED BY



DISSERTATION COMMITTEE

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Dr. James A. Payne, for his help, guidance, and patience. With his patient assistance, I could reach to the completion of this dissertation.

Thanks also go to Dr. Samuel C. Lee, Dr. Albert B. Schwarzkopf, Dr. John C. Thompson, Dr. Abbas Rafii, Dr. Stefan Feyock, and Dr. Teofilo Gonzalez for their valuable comments to the research and their guidance on my entire study of computing science.

Perhaps the deepest appreciation must go to Mrs. Betty Sudduth for her patient typing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

ABSTRACT

This study considers the use of logarithmic number systems in digital filters. Specifically it seeks to determine the capability and limitations in implementing filters with microcomputers, using these number systems.

Addition error in logarithmic number system is shown theoretically and experimentally to be the sum multiplied by a random variable which has an approximately uniform probability distribution. Formula is derived for logarithmic filter's accumulated roundoff errors for the case of stochastic input. It is shown theoretically that a logarithmic filter's accumulated roundoff error is smaller than that of a floating point filter, given the same number of bits and equal range for both of the number systems.

Several filters are experimentally tested for the comparison between the logarithmic and the floating point filter errors (coefficient quantization, input quantization, and accumulated roundoff errors together). Good agreement is made between the theory and the experiments. All the results show that the logarithmic filter errors are smaller.

A design procedure is given for digital filters. The logarithmic number system is specified which satisfies the requirements of the error to signal ratio, the memory utilization, the speed and the range specifications.

CHAPTER I

INTRODUCTION

Since the introduction of a micro-processor in 1971, its application has grown into vast fields. Some examples are games, process controllers, and sales terminals. One of the major applications is to the process controllers in which a digital filter plays an important role.

In an analog process control, the input-output control relations are expressed more or less by differential equations or Laplace transform notations which are also the analog filter notations. At the realization of the analog filter, the mathematical meaning of the Laplace transform is replaced by the combinations of various characteristics of electronic elements - resistance, inductance, etc. While in the digital process control, those input-output control relations are expressed in difference equations and in z-transform notations which are digital filter notations. There are several methods already established to do the transformation from analog filters to digital filters and vice versa: bilinear transformation, impulse invariant transformation, and mapping of differentials. Once the transformation is done, the analog filter is replaced by the corresponding digital filter which can be realized by a digital computer preferably of low cost and easy handling, a micro-processor.

A micro-processor is a very small and low cost LSI chip. It has all

the important elements of a conventional computer central processing unit, arithmetic, logic, and control. The micro-processor operates on memory in which the digital filter can be programmed. The space utilization depends on the program and the data size. For the logarithmic arithmetic, the size of the data becomes quite large as the number of bits in the fraction part of a word gets large, as shown in section 6.2 of Chapter VI. Note that the fraction part is defined in (2.6) as ℓ part in section 2.2. 16-bit FOCUS [6] uses 2305 bytes; but it can be covered by 10 chips (data only, 256 bytes for one chip). All this means that a digital device of complex functions is available with the cost comparable to that of an analog controller; and it has more advantages: flexibility, easy to change the program or filter characteristics; reliability, almost noise free transmission; etc.

Despite the various advantages of a micro-processor, demands always exceed its capability in speed, accuracy, cost, etc. As signals are processed by a digital machine, it is a natural tendency to have more sophisticated algorithms (higher order filter or spectral analysis) which produce more error in computations and consume more processing time or to have more dynamic ranges of input and output signals. Installation of a floating point arithmetic by hardware has been one way to solve the accuracy and dynamic range problems at the expense of speed and cost.

As various advantages of logarithmic number system over fixed or floating point number systems are stated in previous sections, the installation of the logarithmic arithmetic in micro-computers should provide the better solutions. FOCUS system [6] programmed in a micro-computer which employs logarithmic arithmetic can provide tremendous speed and accuracy in the basic arithmetic operations of addition, subtraction,

multiplication and division.

Yet it is not known how much better it is to use logarithmic arithmetic in digital filtering. It is the purpose of this research to investigate the accuracy questions in the digital filter with logarithmic arithmetic.

This dissertation includes the following:

1. Addition (only one arithmetic operation which produces error in a logarithmic number system) error is analyzed in terms of the probability distribution. Multiplication has no error.
2. Logarithmic digital filter's accumulated roundoff error is analyzed for the case of stochastic input.
   a) Theoretical error to signal ratio is shown in terms of filter coefficients and the input spectrum for both rounding and truncation.
   b) Experimental computation is done for some filters for the case of rounding and is compared with the above theoretical results.
   c) Theoretical error to signal ratio of a logarithmic filter is compared with that of a floating point filter.
3. Several filters are tested experimentally for deterministic inputs in order to compare the overall error to signal ratios (input quantization, coefficient quantization, and accumulated rounding errors together) between logarithmic filters and floating point filters.
4. Comments are given for the design of logarithmic filter considering the speed, the error to signal ratio, memory utilization, and the range of the number system.

CHAPTER II

BACKGROUND

2.1 A Digital Filter

A digital filter is defined by the computational algorithm:

$$w_n = \sum_{k=0}^{M} b_k x_{n-k} - \sum_{k=1}^{L} a_k w_{n-k} \tag{2.1}$$

where $\{x_n\}$ is the input sequence, $\{w_n\}$ is the output sequence, and b_k and a_k are some constants. To meet a specific requirement such as low pass, band pass, etc., is to determine the constants a_k and b_k with the consideration of the interval of the input sequence [1], [2]. There are abundant applications in many areas such as speech processing [1] process control, etc. An example of a very simple low pass filter is given below to show the relation between an analog filter and the corresponding digital filter.

A low pass filter

$$H(s) = 1/(1+s)$$

of continuous type is desired to be simulated as a digital filter. Using the bilinear transformation, the transfer function of the digital filter will be

$$H(z) = [T/(T+2)(1+z^{-1})]/[1+(T-2)/(T+2)z^{-1}]$$

where T is the sampling period. By H(s), the cutoff frequency is supposed to be 1 rad/s. Then $1 < \frac{\pi}{T}$. Since H(s) is not very sharp, T should be much less than π. The computational algorithm of H(z) is given by

$$w_n = \frac{T}{T+2}(x_n + x_{n-1}) - \frac{T-2}{T+2} w_{n-1}$$

2.2 Error Classification and Number System

Since a digital filter can be realized in a digital network, i.e., a computer, there are two obvious advantages which are flexibility due to the simplicity of changing filter characteristics (coefficients a_k and b_k) and reliability due to the computational algorithm. There is, however, an inherent limitation on accuracy because of the finite word length of the computer. In this section, the accuracy problem will be classified.

There are three factors of accuracy problems which are all due to the finite word length of the computer in some way. First is called common sources of error. Second is due to the form of the computational realization of the equation (2.1). And third is due to the type of number system used in the computations.

First: There are three common sources which are

1) the input quantization error caused by the quantization step of input signal $\{x_n\}$ into a finite number of bits;
2) the representation of the coefficients a_k and b_k into a finite number of bits;
3) Accumulated roundoff errors committed at each computational operation.

Second: There are four commonly used forms of equivalent realization of the digital filter (2.1) if in an ideal situation. They are 1) direct form; 2) canonical form; 3) parallel form; and 4) cascade form. Those forms in some way affect the accuracy of the filter output [3].

Third: There are three types of number systems to be considered for the digital filter. They are a fixed point number system, a floating point number system, and a logarithmic number system. Although the first

two number systems are well-known and widely used, all of the three will be defined in binary 2's complement form for the comparison and later use.

1) A fixed point number is defined by

$$g_0 \cdot g_1 g_2 \cdots g_t \tag{2.2}$$

where g_i are 1 or 0 and (.) represent the binary point. (See Appendix 2.1 for represented number)

$$\text{Range} = \frac{\text{largest positive value}}{\text{smallest positive value}} = \frac{1-2^{-t}}{2^{-t}} = 2^t - 1 \tag{2.3}$$

2) A floating point number is defined in this dissertation by

$$g_0 \cdot g_1 g_2 \cdots g_f e_0 e_1 \cdots e_h \tag{2.4}$$

where g_i and e_i are 1 or 0, and g part is the fractional part and e part is the exponent. (2.4) has to be always normalized, i.e., $g_0 \neq g_1$ so that the fractional part is always

$$\frac{1}{2} \leq | \text{ fractional part } | \leq 1$$

Note: When the value is negative, g_0=1 and g_1=0, because it is in 2's complement. The base of the exponent is 2 and the exponent is an integer (See Appendix 2.1 for more rigorous number representation).

$$\text{Range} = \frac{(1-2^{-f})2^{(2^h-1)}}{\frac{1}{2}\ 2^{-2^h}} = 2(1-2^{-f})2^{(2^{h+1}-1)} = (1-2^{-f})2^{2^{h+1}} \tag{2.5}$$

Note: h and f are specified in the way of (2.4).

3) A logarithmic number is defined by

$$S\ d_0 d_1 \cdots d_\alpha \cdot\ \ell_1\ \ell_2 \cdots \ell_\beta \tag{2.6}$$

where S, d_i and ℓ_i are 0 or 1; S is the sign of the value and d part and ℓ part combined represent the exponent of the number; the base is assumed to be a positive constant a; (.) represent the binary point of the exponent. Therefore, (2.6) represent the number

$$\pm_a [\text{d part} \;.\; \ell \text{ part}]$$

(See Appendix 2.1 for more rigorous number representation for (2.6).

Some examples are given below for the base a = 2, α = 3, and β = 3.

$$00010.010 = 2^{(2+\frac{1}{4})}$$

$$01110.100 = 2^{-(1+\frac{1}{2})}$$

$$10001.111 = -2^{(1+\frac{1}{2}+\frac{1}{4}+1/8)}$$

$$11101.001 = -2^{-(2+\frac{1}{2}+\frac{1}{4}+1/8)}$$

$$\text{Range} = \frac{a^{2^\alpha - 2^{-\beta}}}{a^{-2^\alpha}} = a^{(2^{\alpha+1} - 2^{-\beta})}$$

Note: α and β are specified in this way in (2.6).

A simple comparison can be made for the ranges of the three number systems. It depends on, however, how many bits are assigned for each part of the numbers (2.2), (2.4) and (2.6). But let us do some assignment: 8 bits are given for each number system. For the fixed point number t = 7; for the floating point number f + h = 6, let f = 3 and h = 3; for the logarithmic number $\alpha + \beta = 6$, let α = 3 and β = 3; and a = 2.

$$\text{Range} = 2^7 - 1 = 127 \text{ for the fixed point case;}$$

$$\text{Range} = (1 - 2^{-3})2^{2^4} \approx 57344 \text{ for floating point case;}$$

$$\text{Range} = 2^{2^4 - 2^{-3}} \approx 60097 \text{ for logarithmic case}$$

The logarithmic number can provide the largest range in the above comparison. The usual assignment for the floating point case is that f is three or four times larger than h; then comparable range of the logarithmic case is more advantageous.

2.3 Foundation of Roundoff Error Analysis

In section 2.2, the error of a digital filter is classified. The difference in the second category will not be clear until the analysis goes deep. So in this section somewhat clear differences at the starting point of the digital filter analysis will be shown for the roundoff error of each number system. The other two, i.e., input quantization error and the filter coeffieient quantization error can be handled in a similar fashion.

Assume that x and y are quantized in a computer according to the number system in section 2.2 and $(.)_t$ represent quantized result of the operation (.).

1) Fixed point case

$$(x + y)_t = x + y + e \quad : \quad e = 0 \tag{2.8}$$

$$(xy)_t = xy + e \qquad \begin{cases} |e| \le 2^{-t-1} & \text{for rounding} \\ |e| \le 2^{-t} & \text{for truncation} \end{cases} \tag{2.9}$$

2) Floating point case

$$(x+y)_t = (x+y)(1+e) \qquad |e| \le 2^{-h} \text{ for rounding} \tag{2.10}$$

$$(xy)_t = xy(1+e) \qquad |e| \le 2^{-h+1} \text{ for truncation}$$

3) Logarithmic case

$$(x+y)_t = (x+y)(1+e) \qquad \begin{cases} a^{-2^{-\beta-1}} - 1 \le e \le a^{2^{-\beta-1}} - 1 & \text{for rounding} \\ a^{-2^{-\beta}} - 1 \le e \le 0 & \text{for truncation which is done so that magnitude of represented number gets smaller.} \end{cases} \tag{2.11}$$

$$(xy)_t = (x+y)(1+e) \qquad e = 0 \qquad (2.12)$$

Derivation of (2.11) and the distribution of e are discussed in Chapter II, and it has been tested experimentally that e has the uniform probability distribution (See Fig. 2.1 for rounding).

In comparison between 1), 2) and 3) above, the error magnitude is bounded by a constant for the fixed point case; while for the floating point case and logarithmic case, it is directly proportional to the true result of the operation. One important special case is that the error of the multiplication for logarithmic case is zero. The proportionality shown above suggests that the error analysis method which can be applied to floating point case may be applied to the logarithmic case. Several error analysis [4], [3] have been reported for the floating point case. Next is the comparison of the error range of single arithmetic operation between 2) and 3). The fixed point case cannot be compared directly because the error is bounded by a constant. The same sample value used in section 2.2 will be used, i.e., h = 3, β = 3, a = 2. Only the rounding case is shown:

$|e| \leq 2^{-3} = 0.125$ for floating point case

$2^{-2^{-4}} - 1 \leq e \leq 2^{2^{-4}} - 1$

$|e| \leq 0.045$ for logarithmic case

The advantage of the logarithmic case over floating point case was again shown.

Next consideration has to be made to the availability of the logarithmic arithmetic. It is immediately obvious that the multiplication of logarithmic numbers is just the fixed point number addition. It implies a tremendous speed gain. For the logarithmic addition, an easy method was suggested [7],

though memory usage is quite large, depending on the number of bits used for ℓ-part. For a micro-processor case, two very fast and accurate logarithmic arithmetic operation programs were realized for 8-bit and 16-bit logarithmic numbers [6].

2.4 Recent History

In this section, the recent history of floating point number arithmetic error analysis with a digital filter and the logarithmic number arithmetic will be briefly outlined.

1) Digital Filter error analysis

Wilkinson's book [5] showed, among many things, the error analysis for floating point arithmetic shown in section 2.3 and that of successive arithmetic operations. Sandberg [4] applied the idea of the book to the digital filter and obtained the deterministic absolute bound of the digital filter error. Then Liu and Kaneko [3], in light of the ideas of Sandberg, pursued the error analysis of the same type of digital filter (2.1), as a stochastic input model. N. G. Kingsbury and P. J. W. Rayner [7] used a digital filter to show the effectiveness on accuracy in logarithmic arithmetic.

2) Logarithmic arithmetic realization

Mitchell [9] and Dean [8] computed $\log_2 X$ by approximation methods. Kingsbury and Rayner [7] introduced the logarithmic addition formula with the read-only-memory method. Lee and Edgar [6] realized the very efficient program of logarithmic arithmetic in a micro-computer and made general error analysis of the arithmetic.

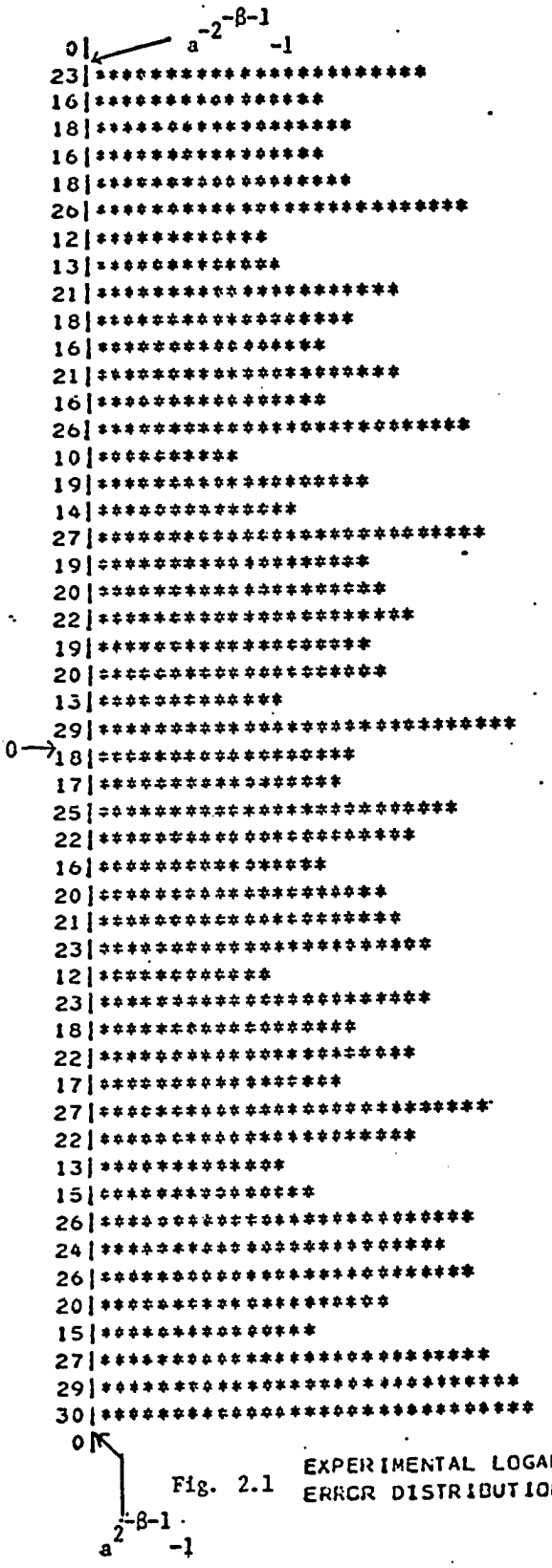


Fig. 2.1 EXPERIMENTAL LOGARITHMIC ADDITION ERROR DISTRIBUTION FOR ROUNDING

APPENDIX 2.1

1) Fixed point case

(2.2) represents the number

$$\sum_{i=1}^{t} 2^{-i} \cdot g_i \qquad \text{when } g_0 = 0,$$

$$-\left(\sum_{i=1}^{t} 2^{-i}\,\overline{g}_i + 2^{-t}\right) \qquad \text{when } g_0 = 1$$

2) Floating point case

(2.4) represents the number

$$\left(\sum_{i=1}^{t} 2^{-i}\, g_i\right) \cdot 2^{E_h} \qquad \text{when } g_0 = 0$$

$$-\left(\sum_{i=1}^{f} 2^{-i}\,\overline{g}_i + 2^{-f}\right) \cdot 2^{E_h} \qquad \text{when } g_0 = 1$$

where

$$E_h = \sum_{i=1}^{h} 2^{h-i}\, e_i \qquad \text{when } e_0 = 0$$

$$= -\left(\sum_{i=1}^{h} 2^{h-i}\,\overline{e}_i + 1\right) \qquad \text{when } e_0 = 1$$

3) Logarithmic case

(2.6) represents the number

$$a\,L_{\alpha\beta} \qquad \text{when} \quad S = 0$$

$$-a\,L_{\alpha\beta} \qquad \text{when} \quad S = 1$$

where $L_{\alpha\beta}$ is defined by

$$L_{\alpha\beta} = \sum_{i=1}^{\alpha} 2^{\alpha-i}\, d_i + \sum_{i=1}^{\beta} 2^{-i}\, \ell_i \qquad \text{when } d_0 = 0$$

$$L_{\alpha\beta} = -\left(\sum_{i=1}^{\alpha} 2^{\alpha-i}\, \bar{d}_i + \sum_{i=1}^{\beta} 2^{-i}\, \bar{\ell}_i + 2^{-\beta} \right) \qquad \text{when } d_0 = 0$$

CHAPTER III

THEORETICAL DEVELOPMENT

In this chapter, a theoretical error analysis of a digital filter for the case of accumulated roundoff errors committed at each computation and stochastic input will be given.

Section 3.1 discusses the probability distribution of addition error in logarithmic arithmetic. Several kinds of expected values related to the addition error, e, will be given:

$$m_e = E[e]$$
$$q^2 = E[(e-m_e)^2]$$
$$R = E[e+1]$$
$$T = E[(e+1)^2]$$

In section 3.2, the error sequence $\{e_n\}$ will be analyzed in terms of the spectrum. With the definition of the computed output sequence $\{y_n\}$ and the error sequence $\{e_n\}$, how errors are introduced and propagated in the digital filter will be shown. Then under the assumption of zero mean and wide sense stationary input sequence, the spectral density function of $\{e_n\}$ will be given in terms of the filter specification, the input spectrum, m_e, and q^2. Section 3.3 uses the results of the section 3.2 and calculates the error to signal ratio:

$$\frac{E[e_n^2]}{E[w_n^2]},$$

and its maximum bound.

3.1 Rounding and truncation error in the logarithmic number system.

We consider a number x which has a true value represented in infinite numbers of bits and x_t is the machine version of x. We define e_1 and e by

$$e_1 = x_t - x = x(x_t/x - 1) \tag{3.1}$$

$$e = \frac{x_t}{x} - 1 \tag{3.2}$$

3.1.1 Rounding

Rounding in the logarithmic number system in this paper means that the exponent is rounded in the usual way. We assume x is uniformly distributed over $[x_1, x_2]$, $0 < x_1 < x_2$ where x_1 and x_2 are defined by

$$x_1 = x_t a^{-2^{-\beta-1}} \tag{3.3}$$

$$x_2 = x_t a^{2^{-\beta-1}} \tag{3.4}$$

The relation between x_1, x_t, and x_2 are depicted in Fig. 3.1.

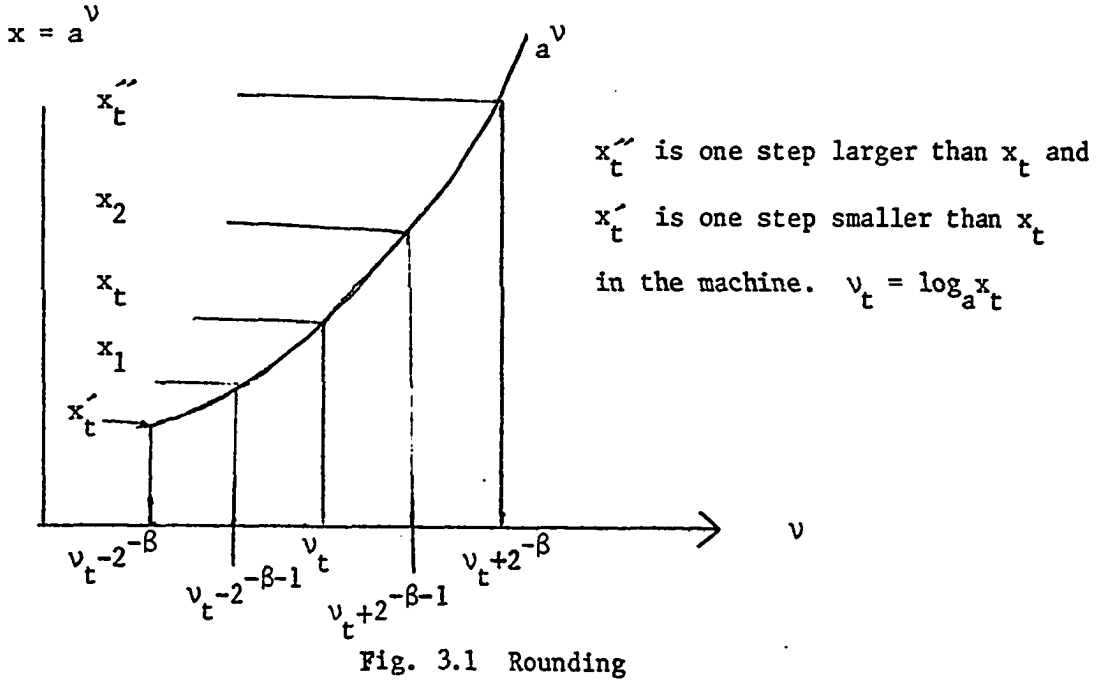


Fig. 3.1 Rounding

Note: Suppose x is a result of a computation (in logarithmic number system, addition only) and it falls between x_1 and x_2 ($|x_1-x_2|$ is small), then it is natural to assume that x will be uniformly distributed. Then e will be distributed over $[\frac{x_t}{x_2} - 1, \frac{x_t}{x_1} - 1]$.

We define f(x), the probability density function of x, by

$$f(x) = \frac{1}{x_2-x_1} \qquad x_1 \le x \le x_2 \tag{3.5}$$

$$= 0 \qquad \text{otherwise}$$

Using the rule of transformation of variables [10] in the probability density function, we obtain

$$g(e) = f(w(e))|w'(e)| = \frac{1}{x_2-x_1}\frac{x_t}{(e+1)^2} \qquad (\frac{x_t}{x_2} - 1 \le e \le \frac{x_t}{x_1} - 1) \tag{3.6}$$

$$= 0 \qquad \text{(otherwise)}$$

where $x = w(e) = \frac{x_t}{e+1}$

That is

$$g(e) = \frac{1}{a^{2^{-\beta-1}} - a^{-2^{-\beta-1}}} \frac{1}{(e+1)^2} \qquad (a^{-2^{-\beta-1}}-1 \le e \le a^{2^{-\beta-1}}-1) \tag{3.7}$$

$$= 0 \qquad \text{(otherwise)}$$

The case of x negative also gives the density function (3.7)

Set $A = 2^{-\beta-1}$ then e is distributed over $[a^{-A}-1, a^{A}-1]$. Since a^A is very close to 1, if β is not too small and a is not too large, we have

$$g(a^A-1) = \frac{1}{a^A-a^{-A}} \frac{1}{a^{2A}} \approx \frac{1}{a^A-a^{-A}} \tag{3.8}$$

$$g(0) = \frac{1}{a^A-a^{-A}} \tag{3.9}$$

$$g(a^{-A}-1) = \frac{1}{a^A-a^{-A}} \frac{1}{a^{-2A}} \approx \frac{1}{a^A-a^{-A}} \tag{3.10}$$

$$\int_{a^{-A}-1}^{a^A-1} \frac{1}{a^A-a^{-A}}\, de = 1 \tag{3.11}$$

(3.8) ~ (3.11) tell that e is of approximate uniformity.

We define m_e and q^2 to be the mean and the variance of e, then

$$m_e = \frac{a^A + a^{-A} - 2}{2} \tag{3.12}$$

$$q^2 = \frac{(a^A - a^{-A})^2}{12} \tag{3.13}$$

And we also define R and T by

$$R = E[e + 1] \tag{3.14}$$

$$T = E[(e+1)^2] \tag{3.15}$$

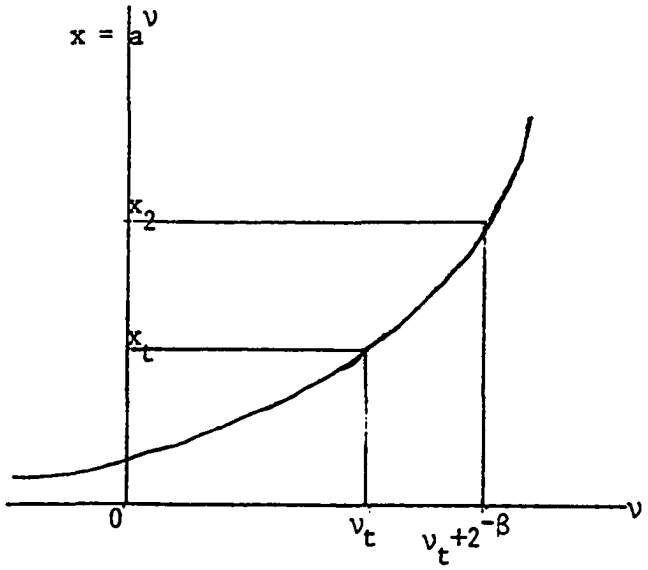
Then

$$R = \frac{a^A + a^{-A}}{2} \tag{3.16}$$

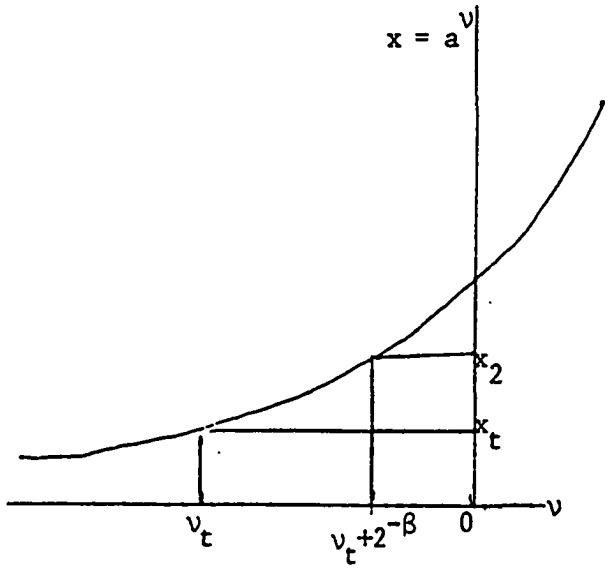
$$T = R^2 + q^2 \tag{3.17}$$

3.1.2 Truncation

The truncation in the logarithmic number system in this paper means that the exponent of the number is floored, i.e. the exponent will be the largest representable number which does not exceed the true exponent of the resulting value and that the sign will be the same as that of the true value. The four cases of the truncation are depicted in Figure 3.2.

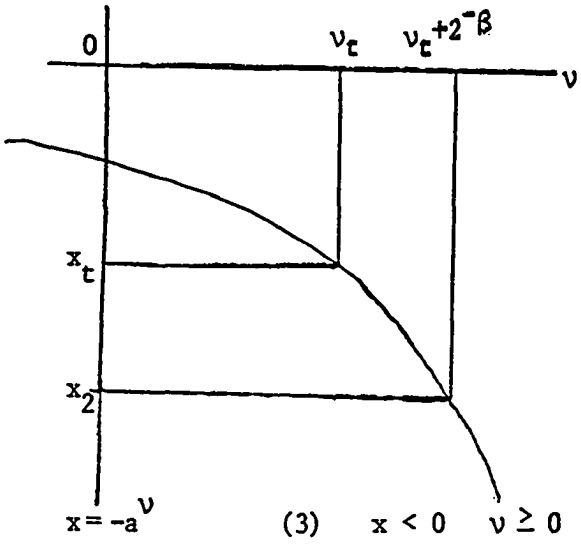


(1) $x > 0$, $\nu \geq 0$

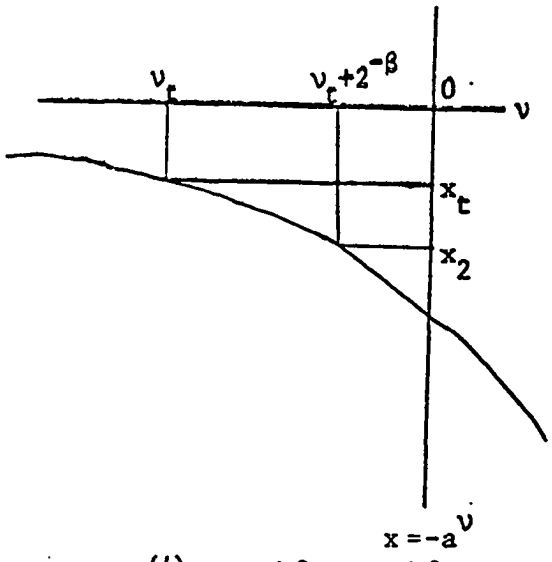


(2) $x > 0$, $\nu < 0$

Figure 3.2 Truncation



(3) $x < 0$ $\nu \geq 0$



(4) $x < 0$ $\nu < 0$

Figure 3.2 (continued)

In the figures the relation of x_2 and x_t is defined as

$$x_2 = x_t . a^{2^{-\beta}} \tag{3.18}$$

and the truncation is done so that any number between x_t and x_2 (or x_2 and x_t) will be quantized to be x_t. If the truncation is done in such a way that the magnitude of the exponent is truncated, then the error (e) range will be approximately twice larger (see Appendix 3.1).

First assume that x is uniformly distributed over $[x_t, x_2]$ $x_t < x < x_2$ (the cases of Fig 3.2 (1) and (2)). Then $e = (\frac{x_t}{x} - 1)$ will be distributed over $[\frac{x_t}{x_2} - 1, 0]$. Define f(x), the probability density function of x, by

$$f(x) = \frac{1}{x_2 - x_t} \qquad x_t \leq x < x_2 \tag{3.19}$$

$$= 0 \qquad \text{otherwise}$$

using the same procedure as in the rounding case, we get

$$g(e) = \frac{1}{x_2 - x_t} \frac{x_t}{(e+1)^2} \qquad (\frac{x_t}{x_2} - 1 \leq e < 0) \tag{3.20}$$

$$= 0 \qquad \text{otherwise}$$

where g(e) is the probability density function of e. That is

$$g(e) = \frac{1}{a^{2^{-\beta}} - 1} \frac{1}{(e+1)^2} \quad (a^{-2^{-\beta}} - 1 \leq e < 0) \tag{3.21}$$

As in the rounding case, we can assume e is uniform over $[a^{-2^{-\beta}}-1, 0]$.

The case of $x < 0$ (the cases of Fig 3.2 (3) and (4)) also gives the equation (3.21).

With the same procedure as in the rounding case, we have the following results in the truncation case.

$$m_e = E[e] = \frac{a^{-2A}-1}{2} \tag{3.22}$$

$$q^2 = V[e] = \frac{(1-a^{-2A})^2}{12} \tag{3.23}$$

$$R = E[e+1] = \frac{1+a^{-2A}}{2} \tag{3.24}$$

$$T = E[(e+1)^2] = R^2+q^2 \tag{3.25}$$

e has the uniform distribution over $[a^{-2A}-1,0]$.

Thus, the error e_1 which results from rounding or truncation has the form

$$e_1 = xe \tag{3.26}$$

where x is considered to be the true result of a computation and e has the uniform distribution over the range discussed above. Since multiplication is exact in the logarithmic number system, consideration has to be made only to addition. Let $x = y_1+y_2$ then from (3.1) and (3.26)

$$(y_1+y_2)_t = (y_1+y_2) + (y_1+y_2)e = (y_1+y_2)(1+e) \tag{3.27}$$

This is a well known formula often employed in the floating point arithmetic error analysis.

3.2 Digital filter and its error spectrum

The form of the digital filter is defined by

$$w_n = \sum_{i=0}^{M} b_i x_{n-i} - \sum_{i=1}^{L} a_i w_{n-i} \tag{3.28}$$

where $\{x_n\}$ is the input sequence and $\{w_n\}$ is the output sequence and in the z transform, the equation (3.28) becomes

$$W(z) = H(z)\ X\ (z) = \frac{N(z)}{D(z)}\ X\ (z) \tag{3.29}$$

where

$$N(z) = \sum_{i=0}^{M} b_i\ z^{-i} \tag{3.30}$$

$$D(z) = \sum_{i=0}^{L} a_i\ z^{-i} \tag{3.31}$$

and $a_o = 1$ without loss of generality.

The stochastic input error analysis method which was applied for the floating point case by Liu and Kaneko [3] is used for the logarithmic number case.

The quantities a_n, b_n, x_n are machine numbers. Assume that the input sequence $\{x_n\}$ is of zero mean, and is wide-sense stationary with autocorrelation function $R_{xx}(n)$ and power spectrum $\Phi_{xx}(z)$. We define e_n by

$$e_n = y_n - w_n \tag{3.32}$$

where $\{y_n\}$ is the machine version output sequence.

$$y_n = \left(\sum_{k=0}^{M} b_k\ x_{n-k} - \sum_{k=1}^{L} a_k y_{n-k} \right)_t \tag{3.33}$$

Assume the computation is made in the following fashion:

$$(\cdots((b_0x_n+b_1x_{n-1})+b_2x_{n-2})+\ldots+b_Mx_{n-M})-(\ldots(a_1y_{n-1}+a_2y_{n-2})+a_3y_{n-3})+\ldots+a_Ly_{n-L}) \tag{3.34}$$

The errors will be introduced as in the Fig 3.3

The equation (3.33) could be written in the form

$$y_n = \sum_{k=0}^{M} b_k\theta_{n,k}x_{n-k} - \sum_{k=1}^{L} a_k\phi_{n,k}y_{n-k} \tag{3.35}$$

where

$$\theta_{n,0} = \theta_{n,1}$$

$$\theta_{n,j} = (1+\xi_n) \prod_{i=j}^{M} (1+\zeta_{n,i}) \qquad j=1,2,\ldots,M \tag{3.36}$$

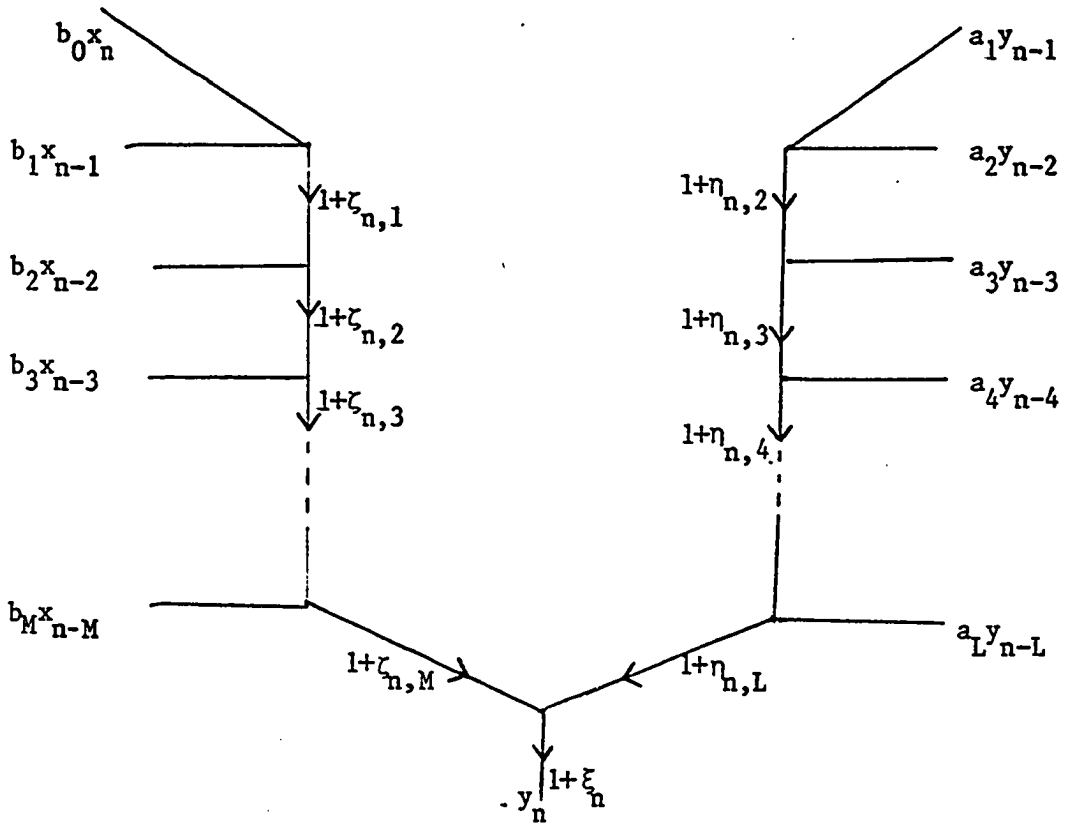


Fig 3.3 Flow Graph

$$\phi_{n,1} = \phi_{n,2}$$

$$\phi_{n,j} = (1+\xi_n) \prod_{i=j}^{L} (1+\eta_{n,i}) \qquad j = 2,3, \cdots, L \qquad (3.37)$$

where ξ_n, $\zeta_{n,k}$, $\eta_{n,k}$ are error variables caused by rounding or truncation at each addition step and are assumed to be identically distributed independent random variables as is e.

To solve (3.35), define y_n', y_n'', $\cdots$, $y^{(P)}$

$$\sum_{k=0}^{L} a_k\bar{\phi}_k y'_{n-k} = \sum_{k=0}^{M} b_k\bar{\theta}_k x_{n-k} \qquad (3.38)$$

$$\sum_{k=0}^{L} a_k\bar{\phi}_k y''_{n-k} = \sum_{k=0}^{M} b_k(\theta_{n,k}-\bar{\theta}_k)x_{n-k} - \sum_{k=0}^{L} a_k(\phi_{n,k}-\bar{\phi}_k)y'_{n-k} \qquad (3.39)$$

$$\sum_{k=0}^{L} a_k\bar{\phi}_k y^{(P)}_{n-k} = -\sum_{k=0}^{L} a_k(\phi_{n,k} - \bar{\phi}_k)\, y^{(P-1)}_{n-k} \qquad (3.40)$$

$$P = 3,4,5,---$$

where

$$\bar{\phi}_k = E[\phi_{n,k}] \text{ and } \bar{\theta}_k = E[\theta_{n,k}] \tag{3.41}$$

From (3.38), (3.39), (3.40) and (3.35) and the definition of $\phi_{n,o} = 1$,

$$y_n = y_n' + y_n'' + y_n''' + \cdots \tag{3.42}$$

Since in (3.39) $(\theta_{n,k} - \bar{\theta}_k)$ and $(\phi_{n,k} - \bar{\phi}_k)$ are very small (see Appendix 3.2) if β is not too small, the magnitude of y_n''' is expected to be much smaller than that of y_n' and x_n. In general, the magnitude of $y_n^{(P)}$ is expected to be much smaller than that of $y_n^{(P-1)}$. Therefore, only the $\{y_n'\}$ and $\{y_n''\}$ sequences are significant. The error e_n is given by

$$\begin{aligned} e_n &= y_n - w_n \\ &= y_n' + y_n'' - w_n \end{aligned} \tag{3.43}$$

Squaring (3.38) and taking the expected value of both sides we get

$$\sum_{k=0}^{L}\sum_{i=0}^{L} a_k \bar{\phi}_k a_i \bar{\phi}_i R_{y'y'}(n-i, n+m-k) = \sum_{k=0}^{M}\sum_{i=0}^{M} b_k \bar{\theta}_k b_i \bar{\theta}_i R_{xx}(n-i,\ n+m-k) \tag{3.44}$$

The right side of (3.44) is constant, and (3.44) is true for all n. Then

$$\sum_{k=0}^{L}\sum_{i=0}^{L} a_k \bar{\phi}_k\ a_i \bar{\phi}_i\ R_{y'y'}(n-i,\ \ n+m-k) = \sum_{k=0}^{L}\sum_{i=0}^{L} a_k\ \bar{\phi}_k\ \ a_i \bar{\phi}_k R_{y'y'}(n'-i, n'+m-k) \tag{3.45}$$

$$\sum_{k=0}^{L}\sum_{i=0}^{L} a_k\ \bar{\phi}_k\ a_i\ \bar{\phi}_i\ [R_{y'y'}(n-i\ n+m-k) - R_{y'y'}(n'-i, n'+m-k)] = 0 \tag{3.46}$$

Then

$$R_{y'y'}(n-i, n+m-k) = R_{y'y'}(n'-i,\ n'+m-k) \tag{3.47}$$

holds for any n, n'. Then $R_{y'y'}(n-i, n+m-k) = R_{y'y'}(0, m+i-k)$ only depends on $m+i-k$. Taking the expected value of (3.38), we get

$$\sum_{k=0}^{L} a_k\ \bar{\phi}_k\ E[y'_{n-k}] = \sum_{k=0}^{M} b_k\ \bar{\theta}_k\ E[x_{n-k}] = 0 \tag{3.48}$$

(3.48) holds for all n. Then

$$E[y'_{n-k}] = 0 \tag{3.49}$$

The above discussion tells that y'_n is wide sense stationary and zero mean.

$R_{y'y'}(n-i, n+m-k)$ could be written as $R_{y'y'}(m+i-k)$.

We get z transform of (3.44):

$$\sum_{m=-\infty}^{\infty} \sum_{k=0}^{L} \sum_{i=0}^{L} a_k \bar{\phi}_k a_i \bar{\phi}_i R_{y'y'}(m+i-k) z^{-m} = \sum_{m=-\infty}^{\infty} \sum_{k=0}^{M} \sum_{i=0}^{M} b_k \bar{\theta}_k b_i \bar{\theta}_i R_{xx}(m+i-k) z^{-m} \tag{3.50}$$

Let $m + i - k = \ell$ then $-m = -\ell + i - k$

$$\sum_{k=0}^{L} \sum_{i=0}^{L} a_k \bar{\phi}_k a_i \bar{\phi}_i z^{i-k} \sum_{\ell=-\infty}^{\infty} R_{y'y'}(\ell) z^{-\ell} = \sum_{k=0}^{M} \sum_{i=0}^{M} b_k \bar{\theta}_k b_i \bar{\theta}_i z^{i-k} \sum_{\ell=-\infty}^{\infty} R_{xx}(\ell) z^{-\ell} \tag{3.51}$$

Let

$$\Phi_{y'y'}(z) = \sum_{\ell=-\infty}^{\infty} R_{y'y'}(\ell) z^{-\ell} \quad ; \quad \Phi_{xx}(z) = \sum_{\ell=-\infty}^{\infty} R_{xx}(\ell) z^{-\ell} \tag{3.52}$$

$$\sum_{k=0}^{L} a_k \bar{\phi}_k z^{-k} = D'(z) \; ; \sum_{k=0}^{M} b_k \bar{\theta}_k z^{-k} = N'(z) \tag{3.53}$$

Then, (3.51) will be

$$D'(z)\, D'(z^{-1})\, \Phi_{y'y'}(z) = N'(z)\, N'(z^{-1})\, \Phi_{xx}(z) \tag{3.54}$$

$$\Phi_{y'y'}(z) = \frac{N'(z)N'(z^{-1})}{D'(z)D'(z^{-1})} \Phi_{xx}(z) \tag{3.55}$$

From (3.39) and (3.43) we have

$$\sum_{k=0}^{L} a_k \bar{\phi}_k e_{n-k} = \sum_{k=0}^{L} a_k \bar{\phi}_k (y'_{n-k} - w_{n-k}) + u_n \tag{3.56}$$

where

$$u_n = \sum_{k=0}^{M} b_k (\theta_{n,k} - \bar{\theta}_k) x_{n-k} - \sum_{k=0}^{L} a_k (\phi_{n,k} - \bar{\phi}_k) y'_{n-k} \tag{3.57}$$

The statistics of $\theta_{n,k}$ and $\phi_{n,k}$ are given in Appendix 3.3. Using those results and (3.57) we obtain the following:

$$E[u_n] = E\left[\sum_{k=0}^{M} b_k (\theta_{n,k} - \bar{\theta}_k)\, x_{n-k} - \sum_{k=0}^{L} a_k (\phi_{n,k} - \bar{\phi}_k) y'_{n-k}\right] = 0 \tag{3.58}$$

$$E[u_n u_m] = E[(\sum_{k=0}^{M} b_k(\theta_{n,k}-\bar{\theta}_k)\, x_{n-k} - \sum_{k=0}^{L} a_k(\phi_{n,k}-\bar{\phi}_k)\, y'_{n-k}) \cdot (\sum_{i=0}^{M} b_i(\theta_{m,i}-\bar{\theta}_i) x_{m-i} - \sum_{i=0}^{L} a_i(\phi_{m,i}-\bar{\phi}_i)\, y'_{m-i})]$$

$$= E[\sum_{k=0}^{M}\sum_{i=0}^{M} b_k b_i(\theta_{n,k}-\bar{\theta}_k)(\theta_{m,i}-\bar{\theta}_i) x_{n-k}x_{m-i} - \sum_{k=0}^{M}\sum_{i=0}^{L} b_k a_i(\theta_{n,k}-\bar{\theta}_k)(\phi_{m,i}-\bar{\phi}_i) x_{n-k}y'_{m-i}$$

$$- \sum_{k=0}^{L}\sum_{i=0}^{M} a_k b_i(\phi_{n,k}-\bar{\phi}_k)(\theta_{m,i}-\bar{\theta}_i) y'_{n-k}x_{m-i} + \sum_{k=0}^{L}\sum_{i=0}^{L} a_k a_i(\phi_{n,k}-\bar{\phi}_k)(\phi_{m,i}-\bar{\phi}_i) y'_{n-k}y'_{m-i}] \tag{3.59}$$

Since $\theta_{u,k}$, $\phi_{n,k}$ are independent from y'_n and x_n and the expected values of $(\theta_{n,k}-\bar{\theta}_k)(\theta_{m,i}-\bar{\theta}_i)$, $(\theta_{n,k}-\bar{\theta}_k)(\phi_{m,i}-\bar{\phi}_i)$, $(\phi_{n,k}-\bar{\phi}_k)(\theta_{m,i}-\bar{\theta}_i)$, and $(\phi_{n,k}-\bar{\phi}_k)(\phi_{m,i}-\bar{\phi}_i)$ are zero when $n \neq m$, then

$$E[u_n u_m] = 0 \qquad n \neq m \tag{3.60}$$

When $n = m$, we obtain the following equations:

$$E[u_n^2] = E[\sum_{k=0}^{M}\sum_{i=0}^{M} b_k b_i(\theta_{n,k}-\bar{\theta}_k)(\theta_{n,i}-\bar{\theta}_i)\, x_{n-k}\, x_{n-i}$$

$$- \sum_{k=0}^{M}\sum_{i=0}^{L} b_k a_i(\theta_{n,k}-\bar{\theta}_k)(\phi_{n,i}-\bar{\phi}_i)\, x_{n-k}\, y'_{n-i}$$

$$- \sum_{k=0}^{L}\sum_{i=0}^{M} a_k b_i(\phi_{n,k}-\bar{\phi}_k)(\theta_{n,i}-\bar{\theta}_i)\, y'_{n-k}\, x_{n-i}$$

$$+ \sum_{k=0}^{L}\sum_{i=0}^{L} a_k a_i(\phi_{n,k}-\bar{\phi}_k)(\phi_{n,i}-\bar{\phi}_i)\, y'_{n-k}\, y'_{n-i}]$$

$$= \sum_{k=0}^{M}\sum_{i=0}^{M} b_k b_i E[(\theta_{n,k}-\bar{\theta}_k)(\theta_{n,i}-\bar{\theta}_i)] R_{xx}(k-i)$$

$$+ \sum_{k=0}^{L}\sum_{i=0}^{L} a_k a_i E[(\phi_{n,k}-\bar{\phi}_k)(\phi_{n,i}-\bar{\phi}_i)] R_{y'y'}(k-i)$$

$$- \sum_{k=0}^{M}\sum_{i=0}^{L} b_k a_i E[(\theta_{n,k}-\bar{\theta}_k)(\phi_{n,i}-\bar{\phi}_i)] E[x_{n-k}y'_{n-i}]$$

$$- \sum_{k=0}^{M}\sum_{i=0}^{L} a_i b_k E[(\phi_{n,i}-\bar{\phi}_i)(\theta_{n,k}-\bar{\theta}_k)] E[x_{n-k}y'_{n-i}] \tag{3.61}$$

Since $E[x_n y'_{n+m}] = R_{xy'}(m)$ only depends on m, which could be proved in the similar procedure that the $\{y'_n\}$ is stationary is proved, so $E[x_{n-k} y'_{n-i}] = R_{xy'}(k-i)$.

$$E[u_n^2] = q^2 \sum_{k=0}^{M} \sum_{i=0}^{M} b_k b_i B_{k,i} R_{xx}(k-i) + q^2 \sum_{k=1}^{L} \sum_{i=1}^{L} a_k a_i A_{k,i} R_{y'y'}(k-i) - 2q^2 \sum_{k=0}^{M} \sum_{i=1}^{L} b_k a_i C_{k,i} R_{xy'}(k-i) \quad (3.62)$$

where

$$B_{k,i} = E[(\theta_{n,k} - \bar{\theta}_k)(\theta_{n,i} - \bar{\theta}_i)] / q^2 \quad (3.63)$$

$$A_{k,i} = E[(\phi_{n,k} - \bar{\phi}_k)(\phi_{n,i} - \bar{\phi}_i)] / q^2 \quad (3.64)$$

$$C_{k,i} = E[(\theta_{n,k} - \bar{\theta}_k)(\phi_{n,i} - \bar{\phi}_i)] / q^2 \quad (3.65)$$

where

$$q^2 = \frac{(a^{A} - a^{-A})^2}{12} \quad \text{for rounding}$$

$$q^2 = \frac{(1 - a^{-2A})^2}{12} \quad \text{for truncation}$$

$A_{k,i}$, $B_{k,i}$, and $C_{k,i}$ all depend on the statistics of $\phi_{n,i}$ and $\theta_{n,i}$ given in Appendix 3.3. Using the relation $(1+x)^k \approx 1+kx$ when x is small, we obtain the following equations.

$$\begin{aligned} A_{i,k} &\approx L && \text{for } k=i=1 \\ &\approx L + 2 - \max(i,k) && \text{otherwise} \\ B_{i,k} &\approx M + 1 && \text{for } k=i=0 \\ &\approx M + 2 - \max(i,k) && \text{otherwise} \\ C_{i,k} &\approx 1 \end{aligned} \quad (3.66)$$

Thus $\{u_n\}$ is w.s. stationary and zero mean. $R_{uu}(m)$, the autocorrelation of $\{u_n\}$ could be written as:

$$R_{uu}(m) = E[u_n^2] \qquad \text{for } m = 0 \tag{3.67}$$

$$= 0 \qquad \text{for } m \neq 0$$

$\Phi_{uu}(z)$ the z transform of $R_{uu}(m)$ is as follows:

$$\Phi_{uu}(z) = q^2 \sum_{m=-\infty}^{\infty} \sum_{k=0}^{M} \sum_{i=0}^{M} b_k b_i B_{k,i} R_{xx}(m+k-i) z^{-m}$$

$$+ q^2 \sum_{m=-\infty}^{\infty} \sum_{k=1}^{L} \sum_{i=1}^{L} a_k a_i A_{k,i} R_{y'y'}(m+k-i) z^{-m}$$

$$-2q^2 \sum_{m=-\infty}^{\infty} \sum_{k=0}^{M} \sum_{i=1}^{L} b_k a_i C_{k,i} R_{xy'}(m+k-i) z^{-m} \tag{3.68}$$

$$\Phi_{uu}(z) = q^2 \left\{ |B(z)|^2 \Phi_{xx}(z) + |A(z)|^2 \Phi_{y'y'}(z) - 2C(z) \Phi_{xy'}(z) \right\} \tag{3.69}$$

Then

$$R_{uu}(0) = \sigma_u^2 = q^2 \frac{1}{2\pi j} \oint \left[|B(z)|^2 \Phi_{xx}(z) + |A(z)|^2 \Phi_{y'y'}(z) - 2C(z) \Phi_{xy'}(z) \right] \frac{dz}{z} \tag{3.70}$$

where $$|A(z)|^2 = \sum_{k=1}^{L} \sum_{i=1}^{L} a_k a_i A_{k,i} z^{k-i} \tag{3.71}$$

$$|B(z)|^2 = \sum_{k=0}^{M} \sum_{i=0}^{M} b_k b_i B_{k,i} z^{k-i} \tag{3.72}$$

$$C(z) = \sum_{k=0}^{M} \sum_{i=1}^{L} b_k a_i C_{k,i} z^{k-i} \tag{3.73}$$

From (3.38) we obtain

$$\Phi_{xy'}(z) = \frac{N'(z)}{D'(z)} \Phi_{xx}(z) \tag{3.74}$$

Putting (3.74) and the following equation (3.76) into (3.70), we get

$$\sigma_u^2 = \frac{q^2}{2\pi j} \oint \left[|B(z)|^2 + \left| A(z) \frac{N'(z)}{D'(z)} \right|^2 - 2C(z) \frac{N'(z)}{D'(z)} \right] \Phi_{xx}(z) \frac{dz}{z} \tag{3.75}$$

From (3.38) and (3.28) we obtain

$$\Phi_{y'y'}(z) = \left| \frac{N'(z)}{D'(z)} \right|^2 \Phi_{xx}(z) \tag{3.76}$$

$$\Phi_{ww}(z) = \left|\frac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z) \tag{3.77}$$

From (3.56), $\{e_n\}$ is wide sense stationary and zero mean and using that $\{u_n\}$ is zero mean and white, we obtain

$$\Phi_{ee}(z) = \Phi_{y'y'}(z) + \Phi_{ww}(z) - 2\,\Phi_{wy'}(z) + \frac{\sigma_u^2}{|D'(z)|^2}$$

$$= \left|\frac{N'(z)}{D'(z)} - \frac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z) + \frac{\sigma_u^2}{|D'(z)|^2} \tag{3.78}$$

(3.78) could be reduced as follows (see details in Appendix 3.4):

$$\Phi_{ee}(z) = \frac{m_e}{|D(z)|^2}\left|B'(z) - \frac{N(z)A'(z)}{D(z)}\right| \Phi_{xx}(z) + \frac{\sigma_u^2}{|D(z)|^2} \tag{3.79}$$

where $m_e = \dfrac{a^{A}+a^{-A}-2}{2}$ for rounding case

$m_e = \dfrac{a^{-2A}-1}{2}$ for truncation case

$$A'(z) = \sum_{k=1}^{L} a_k \alpha_k z^{-k} \tag{3.80}$$

$$B'(z) = \sum_{k=0}^{M} b_k \beta_k z^{-k} \tag{3.81}$$

$$\alpha_k = L \qquad \text{for } k = 1 \tag{3.82}$$

$$\alpha_k = L + 2 - k \qquad \text{for } k \geq 2$$

$$\beta_k = M + 1 \qquad \text{for } k = 0 \tag{3.83}$$

$$\beta_k = M + 2 - K \qquad \text{for } K \geq 1$$

3.3 Error to Signal Ratio

From (3.79), the expected value of e_n^2 is given by

$$E[e_n^2] = \frac{1}{2\pi j}\oint \Phi_{ee}(z)\,\frac{dz}{z}$$

$$= \frac{\sigma_u^2}{2\pi j}\oint \frac{dz}{|D(z)|^2 z} + \frac{1}{2\pi j}\oint \frac{m_e^2}{|D(z)|^2}\left| B'(z) - \frac{N(z)A'(z)}{D(z)}\right|^2 \Phi_{xx}(z)\frac{dz}{z} \tag{3.84}$$

and the expected value of w_n^2 is given by

$$E[w_n^2] = \frac{1}{2\pi j}\oint \Phi_{ww}(z)\frac{dz}{z} = \frac{1}{2\pi j}\oint \left|\frac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z)\frac{dz}{z} \tag{3.85}$$

Then the error to signal ratio $E[e_n^2]/E[w_n^2]$ is given by

$$\frac{E[e_n^2]}{E[w_n^2]} = \frac{\dfrac{\sigma_u^2}{2\pi j}\oint \dfrac{dz}{|D(z)|^2 z}}{\dfrac{1}{2\pi j}\oint \left|\dfrac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z)\dfrac{dz}{z}} + \frac{\dfrac{1}{2\pi j}\oint \dfrac{m_e^2}{|D(z)|^2}\left|B'(z) - \dfrac{N(z)A'(z)}{D(z)}\right|^2 \Phi_{xx}(z)\dfrac{dz}{z}}{\dfrac{1}{2\pi j}\oint \left|\dfrac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z)\,\dfrac{dz}{z}}$$

$$= \frac{\dfrac{1}{2\pi j}\oint \dfrac{dz}{|D(z)|^2 z}\cdot \dfrac{q^2}{2\pi j}\oint \left[|B(z)|^2 + \left|\dfrac{A(z)N'(z)}{D'(z)}\right|^2 - 2C(z)\dfrac{N'(z)}{D'(z)}\right]\Phi_{xx}(z)\,\dfrac{dz}{z}}{\dfrac{1}{2\pi j}\oint \left|\dfrac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z)\,\dfrac{dz}{z}}$$

$$+ \frac{\dfrac{m_e^2}{2\pi j}\oint \dfrac{1}{|D(z)|^2}\left|B'(z) - \dfrac{N(z)}{D(z)}\,A'(z)\right|^2 \Phi_{xx}(z)\,\dfrac{dz}{z}}{\dfrac{1}{2\pi j}\oint \left|\dfrac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z)\,\dfrac{dz}{z}} \tag{3.86}$$

Since $\bar{\phi}_k$, and $\bar{\theta}_k$ are very close to 1, $N'(z)$, and $D'(z)$ can be replaced by $N(z)$, and $D(z)$ respectively.

(3.86) will be

$$\frac{E[e_n^2]}{E[w_n^2]} = \frac{\dfrac{1}{2\pi j}\oint \dfrac{dz}{|D(z)|^2 z}\cdot \dfrac{q^2}{2\pi j}\oint \left[|B(z)|^2 + \left|A(z)\,\dfrac{N(z)}{D(z)}\right|^2 - 2C(z)\,\dfrac{N(z)}{D(z)}\right]\Phi_{xx}(z)\,\dfrac{dz}{z}}{\dfrac{1}{2\pi j}\oint \left|\dfrac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z)\dfrac{dz}{z}}$$

$$+\frac{\frac{m_e^2}{2\pi j}\oint \frac{1}{|D(Z)|^2}\left|B'(z)-\frac{N(z)}{D(z)}\cdot A'(z)\right|^2 \Phi_{xx}(z)\,\frac{dz}{z}}{\frac{1}{2\pi j}\oint \left|\frac{N(z)}{D(z)}\right|^2 \Phi_{xx}(z)\,\frac{dz}{z}} \qquad (3.87)$$

To get the maximum of (3.87)

$$\frac{E[e_n^2]}{E[w_n^2]} \le \frac{q^2}{2\pi j}\oint \frac{dz}{|D(z)|^2 z}\cdot \max_{|z|=1}\left[\left|\frac{D(z)}{N(z)}\right|^2 \left(|B(z)|^2+\left|A(z)\frac{N(z)}{D(z)}\right|^2-2C(z)\frac{N(z)}{D(z)}\right)\right]$$

$$+ m_e^2 \cdot \max_{|z|=1}\left[\frac{1}{|N(z)|^2}\left|B'(z)-\frac{N(z)}{D(z)}A'(z)\right|^2\right] \qquad (3.88)$$

This is the same formula shown by Liu and Kaneko [3] for the case of floating point numbers.

Appendix 3.1 Truncation of the Magnitude of Exponent

If the truncation is done in the way that the magnitude of the exponent is truncated, the relationship between x_t and x_2 is depicted in the figure A.3.1.

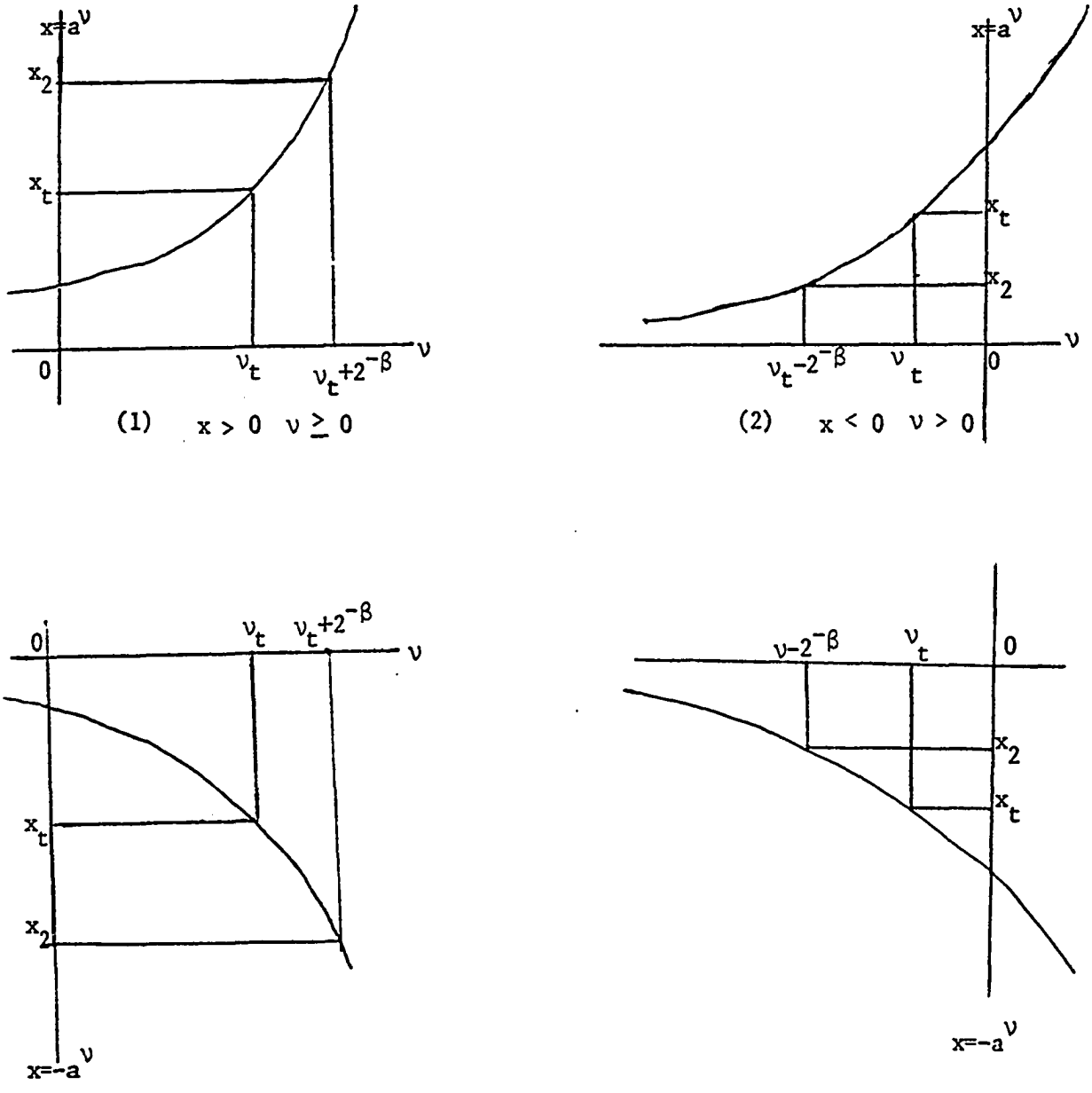


Fig. A.3.1 Truncation of the Exponent's Magnitude

For Fig. A. 3.1 (1) and (3) we have

$$x_2 = x_t \cdot a^{2^{-\beta}} \qquad (A.1)$$

For Fig. A. 3.1 (2) and (4) we have

$$x_2 = x_t \cdot a^{-2^{-\beta}} \qquad (A.2)$$

For the equation (A.2), if x is assumed to be distributed over $[x_2, x_t]$ $x_2 < x_t$ (Fig. A. 3.1(2), we have the error equation as

$$e = \frac{x_t - x}{x} = \frac{x_t}{x} - 1 \qquad (A.3)$$

then e. is distributed over $[0 , a^{2^{-\beta}} -1]$ and for the equation (A.1) as previously shown e is distributed over $[a^{-2^{-\beta}} -1, 0]$. Then e is distributed over $[a^{-2^{-\beta}} -1, a^{2^{-\beta}} -1]$

Appendix 3.2 Differences of Accumulated Errors and their Expected Values

A. Rounding

$$\max(\theta_{n,k}) = (a^{A})^{M+1}$$

$$\min(\theta_{n,k}) = (a^{-A})^{M+1}$$

$$\max(\bar{\theta}_k) = \left(\frac{a^{A}+a^{-A}}{2}\right)^{M+1}$$

$$\min(\bar{\theta}_k) = \left(\frac{a^{A}+a^{-A}}{2}\right)^{2}$$

$$\max(\phi_{n,k}) = (a^{A})^{L}$$

$$\min(\phi_{n,k}) = (a^{-A})^{L}$$

$$\max(\bar{\phi}_k) = \left(\frac{a^{A}+a^{-A}}{2}\right)^{L}$$

$$\min(\bar{\phi}_k) = \left(\frac{a^{A}+a^{-A}}{2}\right)^{2}$$

Then, since $A = 2^{-\beta-1}$, if β is not too small, $(\theta_{n,k}-\bar{\theta}_k)$ and $(\phi_{n,k}-\bar{\phi}_k)$ are very small.

B. Truncation

$$\max(\theta_{n,k}) = 1$$

$$\min(\theta_{n,k}) = (a^{-2A})^{M+1}$$

$$\max(\bar{\theta}_k) = ((1+a^{-2A})/2)^{2}$$

$$\min(\bar{\theta}_k) = ((1+a^{-2A})/2)^{M+1}$$

$$\max(\phi_{n,k}) = 1$$

$$\min(\phi_{n,k}) = (a^{-2A})^{M+1}$$

$$\max(\bar{\phi}_k) = ((1+a^{-2A})/2)^{2}$$

$$\min(\bar{\phi}_k) = ((1+a^{-2A})/2)^{L}$$

Then, since $A = 2^{-\beta-1}$, if β is not too small, $(\theta_{n,k}-\bar{\theta}_k)$ and $(\phi_{n,k}-\bar{\phi}_k)$ are also very small.

Appendix 3.3 Statistics of Accumulated Errors

Using the equations of (3.36) and (3.37), the following results are obtained.

A. Rounding Case

$$E[\theta_{n,j}] = R^{M+1} \quad \text{for } j = 0$$
$$= R^{M+2-j} \quad j \geq 1$$

$$E[\phi_{n,j}] = R^{L} \quad \text{for } j = 1$$
$$= R^{L+2-j} \quad j \geq 2$$

$$E[\theta^2_{n,j}] = T^{M+1} \quad \text{for } j = 0$$
$$= T^{M+2-j} \quad j \geq 1$$

$$E[\theta_{n,j}\theta_{n,k}] = R^{j-1}T^{M+2-j} \quad \text{for } M \geq j > k = 0$$
$$= R^{j-k}T^{M+2-j} \quad M \geq j > k = 1$$

$$E[\phi^2_{n,j}] = T^{L} \quad \text{for } j = 1$$
$$= T^{L+2-j} \quad j \geq 2$$

$$E[\phi_{n,j}\phi_{n,k}] = R^{j-2}T^{L+2-j} \quad \text{for } L \geq j > k = 1$$
$$= R^{j-k}T^{L+2-j} \quad L \geq j > k \geq 2$$

$$E[\theta_{n,j}\phi_{n,k}] = TR^{M+L-1} \quad \text{for } k = 1 \quad j = 0$$
$$= TR^{M+L+1-k} \quad \text{for } k \geq 2 \quad j = 0$$
$$= TR^{M+L-j} \quad k = 1 \quad j \geq 1$$
$$= TR^{M+L+2-k-j} \quad k \geq 2 \quad j \geq 1$$

where $R = E[(1+e)] = \frac{A+a^{-A}}{2}$, $T = E[(1+e)^2] = R^2 + q^2$

where $q^2 = \left(\frac{A-a^{-A}}{12}\right)^2$ and $A = 2^{-\beta-1}$

B. Truncation case.

$$
\begin{aligned}
E[\theta_{n,j}] &= R^{M+1} && \text{for } j = 0 \\
&= R^{M+2-j} && j \geq 1 \\
E[\phi_{n,j}] &= R^{L} && \text{for } j = 1 \\
&= R^{L+2j} && j \geq 2 \\
E[\theta^2_{n,j}] &= T^{M+1} && \text{for } j = 0 \\
&= T^{M+2-j} && j \geq 1 \\
E[\theta_{n,j}\theta_{n,k}] &= R^{j-1}T^{M+2-j} && \text{for } M \geq j > k = 0 \\
&= R^{j-k}T^{M+2-j} && M \geq j > k \geq 1 \\
E[\phi^2_{n,j}] &= T^{L} && \text{for } j = 1 \\
&= T^{L+2-j} && j \geq 2 \\
E[\phi_{n,j}\phi_{n,k}] &= R^{j-2}T^{L+2-j} && \text{for } L \geq j > k = 1 \\
&= R^{j-k}T^{L+2-j} && L \geq j > k \geq 2 \\
E[\theta_{n,j}\phi_{n,k}] &= TR^{M+L-1} && \text{for } k = 1,\ j = 0 \\
&= TR^{M+L+1-k} && k \geq 2,\ j = 0 \\
&= TR^{M+L-j} && k = 1,\ j \geq 1 \\
&= TR^{L+M+2-k-j} && k \geq 2,\ j \geq 1
\end{aligned}
$$

where $R = E[(1+e)] = \frac{1+a^{-2A}}{2}$, $T = E[(1+e)^2] = R^2+q^2$

where $q^2 = \frac{(1-a^{-2A})^2}{12}$, $A = 2^{-\beta-1}$

Appendix 3.4 Approximation Procedure of Error Spectrum

According to Appendix 3.3 $\bar{\phi}_k$ and $\bar{\theta}_k$ are very close to 1 then

$$D'(z) \approx D(z)$$

A. Rounding

$$\bar{\theta}_k = \left(\frac{a^{A}+a^{-A}}{2}\right)^{M+1} = \left(1 + \frac{a^{A}+a^{-A}-2}{2}\right)^{M+1} \approx 1 + m_e(M+1) \qquad \text{for } k = 0$$

$$\bar{\theta}_k = \left(\frac{a^{A}+a^{-A}}{2}\right)^{M+2-k} \approx 1 + m_e(M+2-k) \qquad \text{for } k \geq 1$$

Then we obtain

$$\bar{\theta}_k = 1 + m_e\beta_k$$

$$N'(z) = \sum_{k=0}^{M} b_k(1+m_e\beta_k)z^{-k} = N+m_eB'(z)$$

$$\bar{\phi}_k = 1 \qquad \text{for } k = 0$$

$$\bar{\phi}_k = \left(\frac{a^{A}+a^{-A}}{2}\right)^{L} \approx 1+m_eL \qquad \text{for } k = 1$$

$$\bar{\phi}_k = \left(\frac{a^{A}+a^{-A}}{2}\right)^{L+2-k} \approx 1+m_e(L+2-k) \qquad \text{for } k \geq 2$$

Then we obtain

$$\bar{\phi}_k = 1 + m_e\alpha_k$$

$$D'(z) = a_0\bar{\phi}_0 + \sum_{k=1}^{L} a_k(1+m_e\alpha_k)z^{-k} = D + m_eA'(z)$$

B. Truncation case

For the truncation case, with the same procedure we obtain the same result except

$$m_e = \frac{a^{-2A}-1}{2}$$

In either case, m_e is very small so we get

$$\left|\frac{N}{D} - \frac{N'}{D'}\right|^2 = \left|\frac{ND' - DN'}{DD'}\right|^2$$

$$= \left|\frac{N(D+m_eA') - D(N+m_eB')}{D(D+m_eA')}\right|^2$$

$$= \frac{m_e^2}{|D|^2}\left|\frac{NA'-DB'}{D+m_eA'}\right|^2$$

$$= \frac{m_e^2}{|D|^2}\left|\frac{DB'-NA'}{D}\right|^2 = \frac{m_e^2}{|D|^2}\left|B' - \frac{NA'}{D}\right|^2$$

where $D = D(z)$, $D' = D'(z)$, $N = N(z)$, $N' = N'(z)$

$A' = A'(z)$, $B' = B'(z)$

CHAPTER IV

EVALUATION OF THE THEORY

In this chapter the methods of computing the theoretical and experimental error to signal ratios will be shown under the assumption of the input sequence $\{x_n\}$ of $\Phi_{xx}(z) = 1$ and zero mean, unless otherwise stated. The theoretical computation is given in section 4.1 and the experimental computation is given in section 4.2. The error to signal ratios are computed only for rounding for both of the theoretical and experimental values except for one case. Example 3 of section 4.1 gives theoretical ratio computation for truncation also. The reason is that truncation obviously produces more error than rounding and is consequently of no practical use. The theoretical and experimental computational results are compared to each other for some number of bit assignments for a number of short words (8 bit to 16 bit words) and are also compared with a floating point arithmetic case. Input sequence with zero mean but $\Phi_{xx}(z) \neq 1$ is tested in section 4.3 and some higher Q filters are tested in section 4.4.

4.1 Theoretical error to signal ratio computation

From the equation (3.87) given in Chapter III, the error to signal ratio is given by

$$\frac{E[e_n^2]}{E[w_n^2]} = \frac{q^2 s_1 s_3 + m_e^2 s_4}{s_2} \tag{4.1}$$

where

$$s_1 = \frac{1}{\pi} \int_0^{\pi} \frac{d\omega}{|D(e^{j\omega})|^2}$$

$$s_2 = \frac{1}{\pi} \int_0^{\pi} \left| \frac{N(e^{j\omega})}{D(e^{j\omega})} \right|^2 \Phi_{xx}(e^{j\omega}) d\omega$$

$$s_3 = \frac{1}{\pi} \int_0^{\pi} \left[|B(e^{j\omega})|^2 + \left| A(e^{j\omega}) \frac{N(e^{j\omega})}{D(e^{j\omega})} \right|^2 - 2 \text{ real}\{C(e^{j\omega}) \frac{N(e^{j\omega})}{D(e^{j\omega})}\} \right] \Phi_{xx}(e^{j\omega}) d\omega$$

$$s_4 = \frac{1}{\pi} \int_0^{\pi} \frac{1}{|D(e^{j\omega})|^2} \left| B'(e^{j\omega}) - \frac{N(e^{j\omega})}{D(e^{j\omega})} A'(e^{j\omega}) \right|^2 \Phi_{xx}(e^{j\omega}) \, d\omega$$

Note:

The imaginary part of $C(e^{j\omega}) \frac{N(e^{j\omega})}{D(e^{j\omega})}$ is odd.

$\Phi_{xx}(e^{j\omega})$ does not have to be a constant one; in sections 4.1, 4.2 and 4.4 $\Phi_{xx}(e^{j\omega}) = 1$ is used for computations.

From the inequality (3.88) given in Chapter III, the maximum bound of the ratio is given by

$$\frac{E[e_n^2]}{E[w_n^2]} \leq q^2 s_1 m_1 + m_e^2 m_2 \tag{4.2}$$

where

$$m_1 = \max_{0 \leq \omega \leq \pi} \left\{ \left| \frac{D(e^{j\omega})}{N(e^{j\omega})} \right|^2 \left[|B(e^{j\omega})|^2 + \left| A(e^{j\omega}) \frac{N(e^{j\omega})}{D(e^{j\omega})} \right|^2 - 2 \text{ real}\{C(e^{j\omega}) \frac{N(e^{j\omega})}{D(e^{j\omega})}\} \right] \right.$$

$$m_2 = \max_{0 \leq \omega \leq \pi} \frac{1}{|N(e^{j\omega})|^2} \left| B'(e^{j\omega}) - \frac{N(e^{j\omega})}{D(E^{j\omega})} A'(e^{j\omega}) \right|^2$$

s_1, s_2, s_3 and s_4 can be computed by numerical integration given the coefficients of the digital filter. A computer program written in Fortran which uses Simpson's rule of numerical integration is given in Appendix 4.1. The program computes the ratio:

$$\{E[e_n^2] \; / \; E[w_n^2]\}^{\frac{1}{2}} \tag{4.3}$$

and its maximum bound for a number of β for each of the logarithmic bases 2 and 10. Three sample filters: two second order filters and a sixth order filter are used for the computation.

Example 1.

The filter used is given by:

$$w_n = x_n - (a_1 w_{n-1} + a_2 w_{n-2}) \tag{4.4}$$

where $a_1 = -\sqrt{2}\rho$, $a_2 = \rho^2$ and $\rho = 0.9$.

The results are shown in Table 4.1. The computation was done by the program in Appendix 4.1 which uses the numerical integration method. The theoretical maximums are a little less than twice the theoretical values. In this example, the following are also computed and compared:

$$RRR = \frac{\left[\sqrt{\frac{E[e_n^2]}{E[w_n^2]}}\right]_{base=2}}{\left[\sqrt{\frac{E[e_n^2]}{E[w_n^2]}}\right]_{base=10}} \quad , \quad QQR = \sqrt{\frac{q^2_{base=2}}{q^2_{base=10}}}$$

The results are shown in Table 4.2. RRR $\approx$ QQR $\approx$ 0.3. This is anticipated because in equation (4.1) $m_e^2 \approx 0$. The error ratio of base = 2 is about 30% of that of base = 10.

Example 2.

The digital filter used is designed by the following specifications:

1. The filter is a Butterworth low pass digital filter.
2. The passband magnitude is constant to within a dB for frequencies below $\Omega_p\pi$.
3. The stopband attenuation is greater than b dB for frequencies between $\Omega_s\pi$ and π. At $\Omega_s\pi$ it is exactly b dB.

4. Use of bilinear transformation with prewarping.

5. Sampling period is unity.

The following is the design procedure [2] for the above specifications.

Squared magnitude function of a continuous Butterworth filter is given by

$$|H(j\Omega)|^2 = \frac{1}{1 + (\frac{j\Omega}{j\Omega_c})^{2N}} \tag{4.5}$$

According to the specifications we obtain

$$20 \log_{10}|H(j2 \tan (\frac{\Omega_p \pi}{2}))| \leq -a \tag{4.6}$$

$$20 \log_{10}|H(j2 \tan (\frac{\Omega_p \pi}{2}))| \leq -b$$

Taking the equalities, we obtain

$$1 + \left(\frac{2 \tan (\frac{\Omega_p \pi}{2})}{\Omega_c} \right)^{2N} = 10^{\frac{a}{10}} \tag{4.7}$$

$$1 + \left(\frac{2 \tan (\frac{\Omega_s \pi}{2})}{\Omega_c} \right)^{2N} = 10^{\frac{b}{10}}$$

Then

$$N = \frac{1}{2} \frac{\log[(10^{\frac{a}{10}}-1)/(10^{\frac{b}{10}}-1)]}{\log[\tan \frac{\Omega_p \pi}{2} / \tan \frac{\Omega_s \pi}{2}]} \tag{4.8}$$

In order to meet the specifications N has to be an integer greater than or equal to the above equation's value.

At $\Omega_s \pi$ the attenuation is exactly b dB. Then

$$\Omega_c = 2 \tan \frac{\Omega_s \pi}{2} / (10^{\frac{b}{10}}-1)^{\frac{1}{2N}} \tag{4.9}$$

Using the above Ω_c, we can get the poles of the Butterworth filter of continuous type by the following set S_p.

$S_p = \{s(k) \mid$ real part of $s(k)$ is negative$\}$

where $s(k) = e^{j(\frac{2k+1}{2n})\pi}$ $\quad k = 0,1,\text{---} 2N-1$

Consequently, there are N elements in S_p. Renaming the elements of S_p by

$$S_p = \{s_p(i) \mid i = 0, 1, \text{---} N-1\}, \tag{4.10}$$

we have the Butterworth filter of continuous type by

$$H(s) = \frac{\prod_{i=0}^{N-1} s_p(i)}{\prod_{i=0}^{N-1} (-s_p(i)+s)} \tag{4.11}$$

By the z transform with $s = \frac{2(1-z^{-1})}{1+z^{-1}}$, the resulting digital filter is

$$H(z) = \frac{\sum_{i=0}^{N} b_i z^{-i}}{\sum_{i=0}^{N} a_i z^{-i}} \quad ; \quad a_0 = 1 \tag{4.12}$$

The above procedure is used in the computer program given in Appendix 4.2 to obtain the digital filter coefficients. In this example with a = 1, b = 15, $\Omega_p = 0.2$, and $\Omega_s = 0.3$, the digital filter coefficients become as in Table 4.3. With the coefficients of the digital filter, the theoretical error to signal ratios are computed by the program in Appendix 4.1. The results are given also in Table 4.3. Since the filter has a zero at π, the maximum values are computed for $0 \leq \omega \leq \pi/2$. The ratios of this filter are much larger than those of a simpler filter of Example 1. RRR and QQR for this example have almost the same values as those of Example 1.

Example 3.

Two similar types of filters of equation (4.4) are used in this example. One is with $\rho = 0.9$ which is exactly the same as that of Example 1 and the other is with $\rho = 0.999$. There are two purposes in this example. One is to assure that the numerical integration method program of Appendix 4.1 which is used in

Example 1 and 2 is correct by computing the same filter's error to signal ratio by a different method. The residue theorem of complex variables is used for this example. The other purpose is to compare the error to signal ratios of the logarithmic filter and those of the floating point filter.

To compute the error to signal ratio of the equation (3.87) and the inequality (3.88), we have to have the following:

$$N(z) = 1$$

$$D(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$$

$$|A(z)|^2 = 2(a_1^2 + a_2^2) + 2a_1 a_2 (z+z^{-1})$$

$$|B(z)|^2 = 1$$

$$C(z) = a_1 z^{-1} + a_2 z^{-2}$$

$$B'(z) = 1$$

$$A'(z) = 2(a_1 z^{-1} + a_2 z^{-2})$$

With $\Phi_{xx}(z) = 1$, the equation (3.87) of Chapter III becomes

$$\frac{E[e_n^2]}{E[w_n^2]} = \frac{q^2}{2\pi j} \oint \left[1 - \frac{2(a_1 z^{-1} + a_2 z^{-2})}{|D(z)|^2}\right] \frac{dz}{z}$$

$$+ \frac{\dfrac{m_e^2}{2\pi j} \oint \dfrac{|1 - a_1 z^{-1} - a_2 z^{-2}|^2}{|D(z)|^4} \dfrac{dz}{z}}{\dfrac{1}{2\pi j} \oint \dfrac{dz}{|D(z)|^2 z}} \qquad (4.13)$$

By $D(z) = 1 + a_1 z^{-1} + a_2 z^{-2} = 1 - \sqrt{2}\,\rho z^{-1} + \rho^2 z^{-2}$

$$= \frac{1}{z^2}\left(z - \frac{(1+j)\rho}{\sqrt{2}}\right)\left(z - \frac{(1-j)\rho}{\sqrt{2}}\right)$$

and the residue theorem of complex variables we have

$$T_1 = \frac{1}{2\pi j} \oint \frac{2a_1 z^{-1}}{|D(z)|^2 z}\, dz = \frac{2\sqrt{2}\, a_1 \rho}{(1-\rho^2)(\rho^4+1)} \tag{4.14}$$

$$T_2 = \frac{1}{2\pi j} \oint \frac{2a_2 z^{-2}}{|D(z)|^2 z}\, dz = \frac{2a_2 \rho^2}{\rho^4+1} \tag{4.15}$$

$$T_3 = \frac{1}{2\pi j} \oint \frac{dz}{|D(z)|^2 z} = \frac{\rho^2+1}{(1-\rho^2)(\rho^4+1)} \tag{4.16}$$

$$\frac{1}{2\pi j} \oint \frac{|1-a_1 z^{-1}-a_2 z^{-2}|^2}{|D(z)|^4 z}\, dz$$

$$= \frac{1}{2\pi j}\left(-\frac{1}{\rho^2}\right) \oint \frac{zdefg}{(habc)^2}\, dz = -\frac{1}{\rho^2} QQ$$

where

$$h = z - \frac{(1+j)\rho}{\sqrt{2}}\ ;\quad a = z - \frac{(1-j)\rho}{\sqrt{2}};\quad b = z - \frac{1+j}{\sqrt{2}\rho}\ ;$$

$$c = z - \frac{1+j}{\sqrt{2}\rho}\ ;\quad d = z + \frac{(1-\sqrt{3})\rho}{\sqrt{2}}\ ;\quad e = z + \frac{(1+\sqrt{3})\rho}{\sqrt{2}}\ ;$$

$$f = z - \frac{1+\sqrt{3}}{\sqrt{2}\rho}\ ;\quad g = z - \frac{1-\sqrt{3}}{\sqrt{2}\rho}\ \text{ and}$$

$$QQ = \frac{1}{2\pi j} \oint \frac{zdefg}{(habc)^2}\, dz \tag{4.17}$$

By use of the residue theorem we have

$$QQ = Q_1 + Q_2 \tag{4.18}$$

where

$$Q_1 = \frac{d}{dz}\left(\frac{zdefg}{(abc)^2}\right)\Bigg|_{z = \frac{(1+j)\rho}{\sqrt{2}}} \tag{4.19}$$

$$= \frac{(abc)^2(defg+zefg+zdfg+zdeg+zdef) - zdefg[2abc(bc+ac+ab)]}{(abc)^4}\Bigg|_{z=\frac{(1+j)\rho}{\sqrt{2}}}$$

$$Q_2 = \frac{d}{dz}\left(\frac{zdefg}{(hbc)^2}\right)\Bigg|_{z=\frac{(1-j)\rho}{\sqrt{2}}} \tag{4.20}$$

$$= \frac{(hbc)^2(defg+zefg+zdfg+zdeg+zdef) - zdefg\ 2hbc(bc+hc+hb)}{(hbc)^4}\Bigg|_{z=\frac{(1-j)\rho}{\sqrt{2}}}$$

Then we have

$$\frac{E[e_n^2]}{E[w_n^2]} = q^2(1-T_1-T_2) - m_e^2\, QQ/(T_3\rho^2) \tag{4.21}$$

with

$$\left|\frac{D(z)}{N(z)}\right|^2 \left[\,|B(z)|^2 + \left|A(z)\,\frac{N(z)}{D(z)}\right|^2 - 2C(z)\,\frac{N(z)}{D(z)}\,\right]$$

$$= 1 + a_1^2 + a_2^2 + a_1a_2(z+z^{-1}) \qquad \text{and}$$

$$\frac{1}{|N(z)|^2}\left|\,B'(z) - \frac{N(z)}{D(z)}\,A'(z)\right|^2 = \frac{|1-a_1z^{-1}-a_2z^{-1}|^2}{|D(z)|^2}$$

we have

$$\frac{E[e_n^2]}{E[w_n^2]} \le q^2 \cdot T_3 \cdot QQ_1 + m_2^3 \cdot QQ_2 \tag{4.22}$$

where

$$QQ_1 = \max_{|z|=1}[\,1+a_1^2+a_2^2+a_1a_2(z+z^{-1})] \tag{4.23}$$

$$QQ_2 = \max_{|z|=1}\left[\frac{|1-a_1z^{-1}-a_2z^{-2}|^2}{|1+a_1z^{-1}+a_2z^{-2}|^2}\right] \tag{4.24}$$

The above procedure is used in the computer program given in Appendix 4.3. The results for $\rho = 0.9$ and $\rho = 0.999$ are given in Tables 4.4 and 4.5 respectively.

The results for $\rho = 0.9$ agree with those of Example 1. The ratio of the filter for $\rho = 0.999$ are shown in [3] for the IBM 7094's floating point case of 36 bit word (1 bit of sign, 8 bits of characteristic, 27 bits of fraction, and base = 2) [13]. The theoretical ratio is 2.1×10^{-7} and the maximum bound is 3.3×10^{-7}. Those of the logarithmic case of base = 2 and 27 bits of fraction are 4.7×10^{-8} and 8.7×10^{-8}.

This is anticipated since the variance of q_ℓ^2 of the logarithmic case and the corresponding variance of q_f^2 of the floating point case are given by

$$q_\ell^2 = \frac{(2^{2^{-\beta-1}} - 2^{-2^{-\beta-1}})^2}{12} \quad ; \quad q_f^2 = \frac{2^{-2h}}{3}$$

and the multiplication of logarithmic numbers produces no error.

Note: m_e is very small for the logarithmic number system, and can be ignored. The graph of q_ℓ^2 and q_f^2 is shown in Fig 4.1.

$$\frac{q_f}{q_\ell} \approx 2.89$$

Then the error to signal ration of $\sqrt{E[e_n^2]/E[w_n^2]}$ of a floating point filter is at least 2.89 times the ratio of the logarithmic filters given the same number of bit for h and β. "At least" means no error for logarithmic multiplication.

Note: If the same number of bits are given for both of the number systems and the fractional parts have the same number of bits (h and β are equal), then the ratios of each number system are almost equal.

Note: The computation for the convergence takes quite a bit of computer time for the filter of $\rho = 0.999$ if it is done by the program of numerical integration method given in Appendix 4.1.

4.2 Experimental error to signal ratio computation

In order to test the theory developed in Chapter III and the theoretical error to signal ratios computed in section 4.1 some experiments are done. The experimental program written in PL/I is given in Appendix 4.4.

4.2.1 General view of the experimental program

The general flowchart is shown in Fig. 4.2. Two filters are operated in the program. One uses logarithmic number system and the other uses long floating point number system so that the error of the logarithmic filter can be computed. The symbol explanation for Fig. 4.2 is given below:

a_i $i = 0---L$: coefficients of the long floating point filter

b_i $i = 0---M$: coefficients of the long floating point filter

ℓ_{a_i} $i = 0---L$: coefficients of the logarithmic filter

ℓ_{b_i} $i = 0---M$: coefficients of the logarithmic filter

x_{n-i} $i = 0---M$: previous inputs of the long floating point filter

w_{n-i} $i = 0---L$: previous outputs of the long floating point filter

$\ell_{x_{n-i}}$ $i = 0---M$: previous inputs of the logarithmic filter

$\ell_{y_{n-i}}$ $i = 0---L$: previous outputs of the logarithmic filter

x'_n : new input of the long floating point filter

$\ell_{x'_n}$: new input of the logarithmic filter

e_n : the error of the logarithmic filter

w_n : all most true output (output of long floating point filter)

Since the long floating point number system has more accuracy, the coefficients, initial filter values, and new inputs are initially given in long floating point number and converted to the logarithmic numbers for the logarithmic filter and those logarithmic numbers are converted back to the long floating point numbers for the long floating point filter.

In the box number 2 of Fig. 4.2, the initial filter values of 0.5 are arbitrarily chosen instead of zeros, because there is no zero in a logarithmic number system and the nearest number to zero is a sort of extreme number in the number system. In the program given in Appendix 4.4, the following considerations which are not in the Fig. 4.2 are given below:

1. Any combination of bit assignments α and β can be tested.
2. Numbers of overflow and underflow in logarithmic arithmetic are counted.
3. Underflow of long floating point arithmetic is checked.
4. Waiting time for statistics collection is given; in the sixth order filter given in example 2, the impulse response dies after about 100 inputs.

4.2.2 Sub-procedures

There are several sub-procedures which are not especially clear in the flowchart of Fig. 4.2.

1. Floating point to logarithmic number conversion: CVBL. The following method is used for the conversion

$$\ell_x = \text{ROUND}(\log_a x) \tag{4.25}$$

where x is a floating point number, ℓ_x is the logarithmic number and a is the logarithmic base. ROUND(r) converts r to the fixed point number, then rounds it to a required precision (see the detail in the procedure CVBL in in Appendix 4.4).

2. $\Phi_{xx}(z) = 1$ and zero mean pseudo-random number generation: UTP

$\Phi_{xx}(z) = 1$ gives the following:

$$R_{xx}(m) = \frac{1}{2\pi j}\oint \Phi_{xx}(z) z^{m-1} dz \qquad (4.26)$$

$$= \frac{1}{2\pi j}\oint z^{m-1} dz = 1 \qquad (m = 0)$$

$$= 0 \qquad (m \neq 0)$$

Then, with mean $m_x = 0$, we have

$$\sigma_x^2 = E[x_n^2] = R_{xx}(0) = 1 \qquad (4.27)$$

The above means that $\{x_n\}$ is a sequence of random numbers with zero mean, variance is one, and no autocorrelation. In the program in Appendix 4.4, uniform random number generation method is used to meet the above requirement. Assume U is a uniform random number distributed over [0,1], then by the following relation, x has the uniform distribution over [-a, a].

$$U = \int_{-a}^{x} \frac{1}{2a}\, dx = \frac{x+a}{2a} \qquad (4.28)$$

which is

$$x = a(2U-1)$$

$$\text{since } \sigma_x^2 = \frac{a^2}{3} = 1, \quad a = \sqrt{3}$$

U is generated in the program by the mixed congruential generation [11].

$$U = z_i/m \; ; \; z_i = a\, z_{i-1} + c \pmod m \qquad (4.29)$$

where $m = 2^{14}$, $a = 129$, $c = 8085$ and $z_0 = 10825$

3. Logarithmic addition: LADD

Logarithmic addition is given in the following:

$$z = x + y \tag{4.30}$$

where $z = s_z a^{e_z}$, $x = s_x a^{e_x}$, $y = s_y a^{e_y}$;

s_z, s_x, and s_y are signs of z, x, and y respectively. a is the base.

Given s_x, e_x, s_y and e_y we like to ascertain s_z and e_z. The equation (4.30) above can be written by

$$s_z a^{e_z} = s_x a^{e_x} + s_y a^{e_y} \tag{4.31}$$

s_z and e_z can be computed in the following way:

1. when $s_x = s_y$, then $s_z = s_x$

$$a^{e_z} = a^{e_x} + a^{e_y} = a^{e_x}(1 + a^{e_y - e_x}) = a^{e_y}(1 + a^{e_x - e_y})$$

Taking the logarithm of base a

$$e_z = e_x + \log_a(1 + a^{e_y - e_x}) = e_y + \log_a(1 + a^{e_x - e_y})$$

Let $e_m = \max(e_x, e_y)$ and $e_f = -|e_x - e_y|$.

$$e_z = e_m + \log_a(1 + a^{e_f}) \tag{4.32}$$

2. When $s_x \neq s_y$, the sign of the result will be

$s_z = s_x$ if $e_x \geq e_y$

$s_z = s_y$ if $e_y < e_x$

and (4.31) will be

$$a^{e_z} = (a^{e_x} - a^{e_y}) \qquad \text{when } e_x \geq e_y$$

$$a^{e_z} = (a^{e_y} - a^{e_x}) \qquad \text{when } e_y > e_x$$

Let $e_m = \max(e_x, e_y)$ and $e_f = -|e_x - e_y|$.

Then

$$e_z = e_m + \log_a(1 - a^{e_f}) \tag{4.33}$$

The above procedure has been shown [7]. Although the FOCUS [6] uses tables produced by

$$F(e_f) = \log_a (1 + a^{e_f}) \qquad \text{and}$$

$$F(e_f) = \log(1 - a^{e_f}) \quad ,$$

the logarithmic addition procedure in this simulation uses the built-in logarithmic functions of PL/I (F) supplied by IBM. A sample look-up table for base = 2, 8 bit word of 3 bit fractional part produced by the procedure is given in Table 4.7. It is exactly the same as the look-up table used by 8 bit FOCUS [6]. For the overflow and the underflow of the logarithmic addition and also multiplication, see the procedures LADD and LMUL in Appendix 4.4.

4.2.3 Examples

The two filters used in the theoretical ratio computation are used in the experiments. Although α, defined in (2.6) in Chapter II, is infinite in the theoretical error to signal ratio computation, it is a small integer in the real situation. A number of combinations of α and β are tested.

Example 1.

This example uses the same filter used in Example 1 of section 4.1. The results are shown in Table 4.8. Table 4.9 shows the comparison between theoretical and experimental ratios for base = 2 which are taken from Table 4.1 and

Table 4.8. Both ratios agree fairly well under the condition that β has the same value and that α is not too small. When α becomes too small in the experiments, overflow or underflow occurs quite often, consequently the error ratio becomes large. For the given filter, base = 2, and the given inputs, the bit assignments of smallest ratios are that $\alpha = 2$ and $\beta = 4$ for 8 bit words, and $\alpha = 3$ and $\beta = 11$ for 16 bit words. For base = 10 in Table 4.8, those of smallest ratios are that $\alpha = 0$ and $\beta = 6$ for 8 bit words, and $\alpha = 1$ and $\beta = 13$ for 16 bit words. The theoretical ratio of $\beta = 7$ and base = 2 is 5.06670×10^{-3} and the experimental ratio of $\alpha = \beta = 7$ is 6.08334×10^{-3} for 400-150 = 250 inputs. The difference is about 20%. Table 4.6, however, shows the experimental ratio for the large number of inputs (8000-150 = 7850). The difference is about 0.8%. The bit combination of $\alpha = 3$ and $\beta = 3$ equivalent of FOCUS.8 [6] produced the ratio of 9.22432×10^{-2}.

Example 2.

This example uses the same filter used in Example 2 of section 4.1. The results are shown in Table 4.10. For this higher order filter, α and β should not be too small. Other than that, the experimental results agree with the theoretical results in Table 4.3. The bit assignments of smallest ratios are that $\alpha = 4$ and $\beta = 10$ for base = 2 and 16 bit words, and $\alpha = 2$ and $\beta = 12$ for base = 10 and 16 bit words. 8 bit words cannot handle this higher order filter. The equivalent of FOCUS.16[6] (base = 10, $\alpha = 5$ and $\beta = 9$) produced 4.71781×10^{-2} of the ratio and that of FOCUS.10[12] (base = 10, $\alpha = 4$, and $\beta = 10$) produced 3.07841×10^{-2}.

4.3 Test for Input Sequence with Spectrum other than Constant One

An easy way to get a zero mean wide sense stationary sequence with the spectrum other than constant one is to get the output sequence of a filter with input sequence of white noise with zero mean. The following filter is

used to get the input sequence.

$$x_n = \frac{1}{3}(g_n + g_{n-1}) + \frac{1}{3} x_{n-1}$$

where $\{g_n\}$ is the input sequence to this filter.

The transfer function of the filter is

$$\hat{H}(z) = \frac{\frac{1}{3}(1+z^{-1})}{1-\frac{1}{3}z^{-1}}$$

Then the spectral density of the sequence $\{x_n\}$ is

$$\Phi_{xx}(z) = |\hat{H}(z)|^2 \Phi_{gg}(z)$$

where $\Phi_{gg}(z)$ is the spectral density of the sequence $\{g_n\}$.

If $\Phi_{gg}(z) = 1$ then $\Phi_{xx}(z)$ is $|\hat{H}(z)|^2$

The sequence of $\{x_n\}$ is applied to the filter of table 4.11 which is the same as that of table 4.1. Theoretical error to signal ratios are shown in table 4.11 and those of experiments are in table 4.12. Like the input sequence of zero mean and $\Phi_{xx}(z) = 1$, the theoretical values agree with the experimental values.

Set t_2 and t_3 to be

$$t_2(\omega) = \left|\frac{N(e^{j\omega})}{D(e^{j\omega})}\right|^2$$

$$t_3(\omega) = |B(e^{j\omega})|^2 + |A(e^{j\omega}) \frac{N(e^{j\omega})}{D(e^{j\omega})}|^2 - \text{real } \{C(e^{j\omega}) \frac{N(e^{j\omega})}{D(e^{j\omega})}\}$$

Since m_e in the equation (4.1) is very small, the equation (4.1) can be considered as

$$\frac{E[e_n^2]}{E[w_n^2]} = \frac{q^2 s_1 s_3}{s_2}$$

Then the above error to signal ratio can be changed by changing the spectrum:

$$\Phi_{xx}(e^{j\omega})$$

of the input sequence if $t_2(\omega)$ and $t_3(\omega)$ have different shapes. But the example of the filter of Table 4.11 and 4.12 does not have observable difference in $t_2(\omega)$ and $t_3(\omega)$ as shown in Fig 4.3. In conclusion, the error to signal ratio of

$$E[e_n^2]/E[w_n^2]$$

does not depend on the input sequence unless $t_2(\omega)$ and $t_3(\omega)$ have different shapes.

4.4 Test for high Q filters

The filters of the form of equation (4.4)

$$w_n = x_n - (a_1 w_{n-1} + a_2 w_{n-2})$$

where $a_1 = -\sqrt{2}\rho$ and $a_2 = \rho^2$

has Q = 39.3 when $\rho = 0.99$ and Q = 393 when $\rho = 0.999$. These two filters are tested with the input sequence of zero mean and $\Phi_{xx}(z) = 1$. The results for the case of $\rho = 0.99$ are shown in Table 4.13 and 4.14.

Table 4.5 and Table 4.15 are the results for the case $\rho = 0.999$. The theoretical and experimental results agree well for $\rho = 0.99$. There are slight disagreements between the theoretical and experimental results for $\rho = 0.999$.

When Q gets high like the filter of Table 4.15, a slight change of the filter coefficients affect the filter characteristics. This slight change has happened in the computation of the results of Table 4.5 and 4.15. The filter of $\rho = 0.999$ of Table 4.5 is the same as that of Table 4.15 and the filter coefficients are given in the long floating point numbers. But the coefficients of the filter are converted to logarithmic number coefficients which differ a little from the original long floating point coefficients. The experimental results are computed with those logarithmic number coefficients.

But the theoretical results of Table 4.5 are computed with the original long floating point coefficients which are slightly different from the long floating point coefficients. The theoretical computation program of Appendix 4.5 computes the theoretical error to signal ratios with the converted logarithmic number coefficients. Since it takes quite a great amount of computing time, only one case is done. It is the case for $\alpha = 5$, $\beta = 9$ and base = 2 shown in Table 4.16. The result of 1.50198×10^{-2} is very much closer to the experimental value of 1.80897 for $\alpha = 5$, $\beta = 9$ and base = 2 of Table 4.15. A large number of inputs are applied for the case of $\alpha = 5$, $\beta = 9$, and the result is 1.69132×10^{-2} which is shown in Table 4.17.

Note: Q of a digital filter in this dissertation is defined by

$$Q = \frac{\theta}{2\sigma}$$

where σ is the distance from the pole $(a + bj)$ of the filter to the unit circle in z plane; the pole $(a + bj)$ is the nearest pole of the filter to the unit circle; θ is defined by

$$\theta = \tan^{-1}\left|\frac{b}{a}\right| \qquad \text{for } a > 0$$

$$= \pi - \tan^{-1}\left|\frac{b}{a}\right| \qquad \text{for } a < 0$$

$$= \frac{\pi}{2} \qquad \text{for } a = 0$$

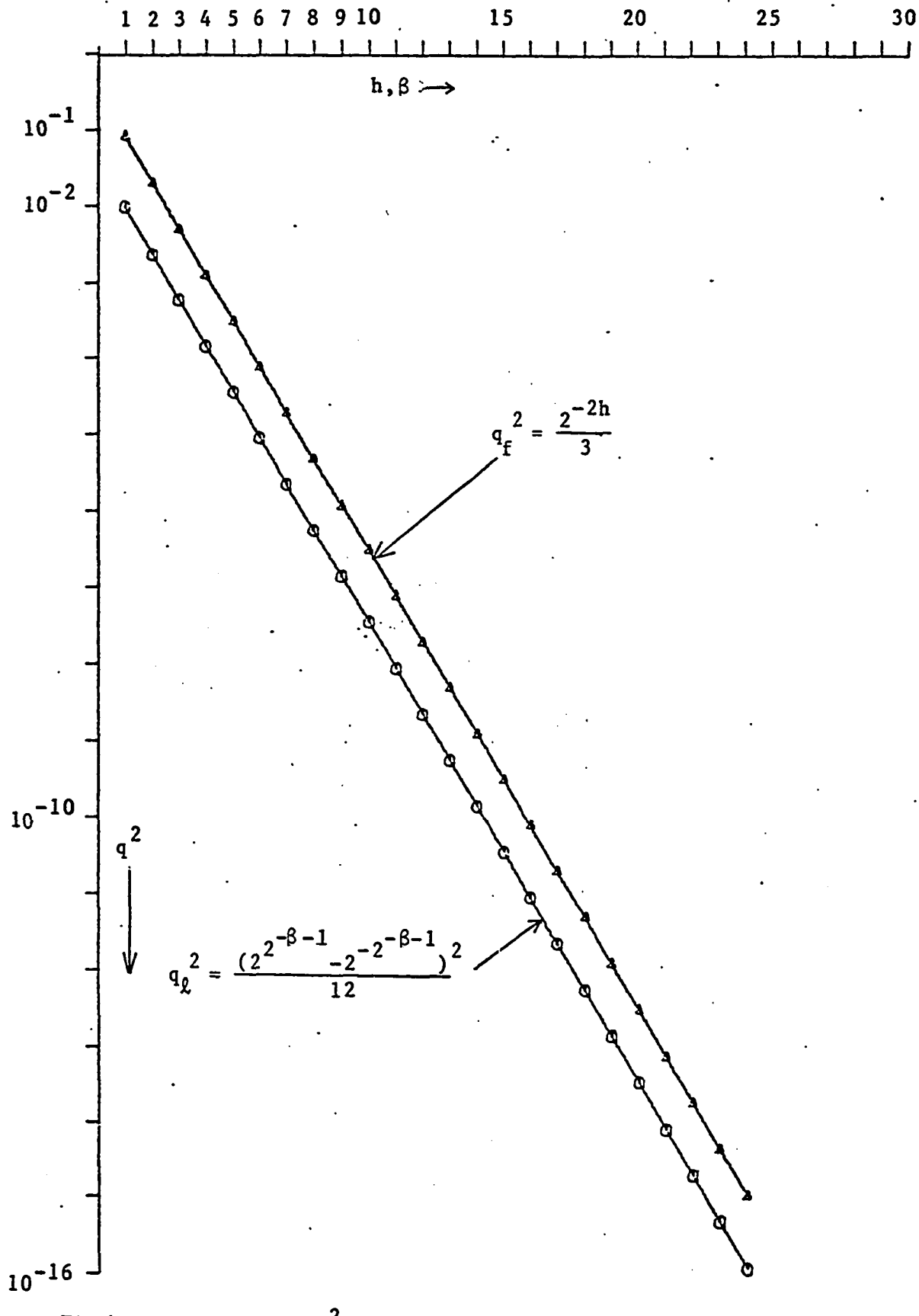


Fig 4.1 Comparison of q^2 of a floating point number system and a logarithmic number system (base = 2)

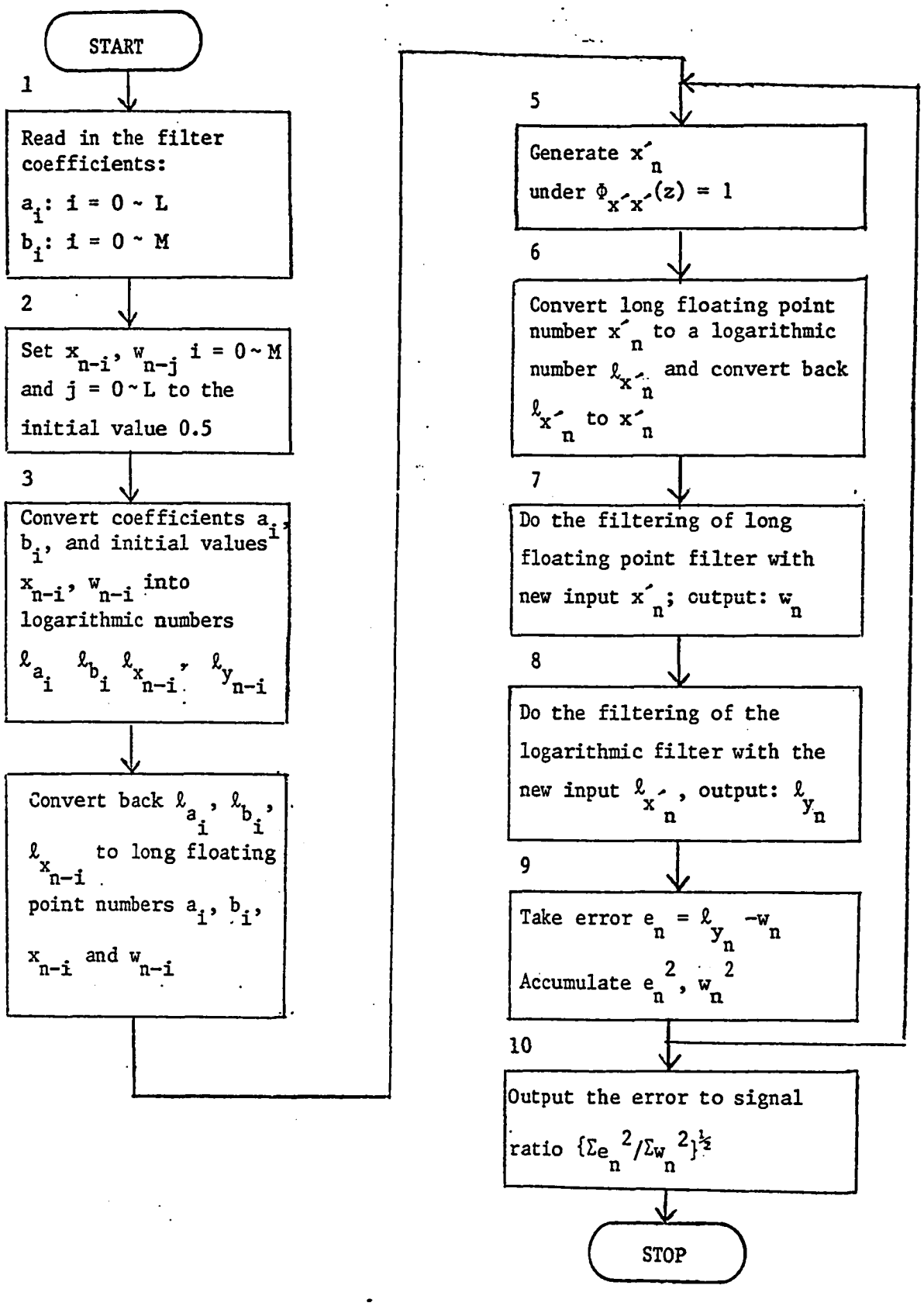


Fig. 4.2 General Flow Chart

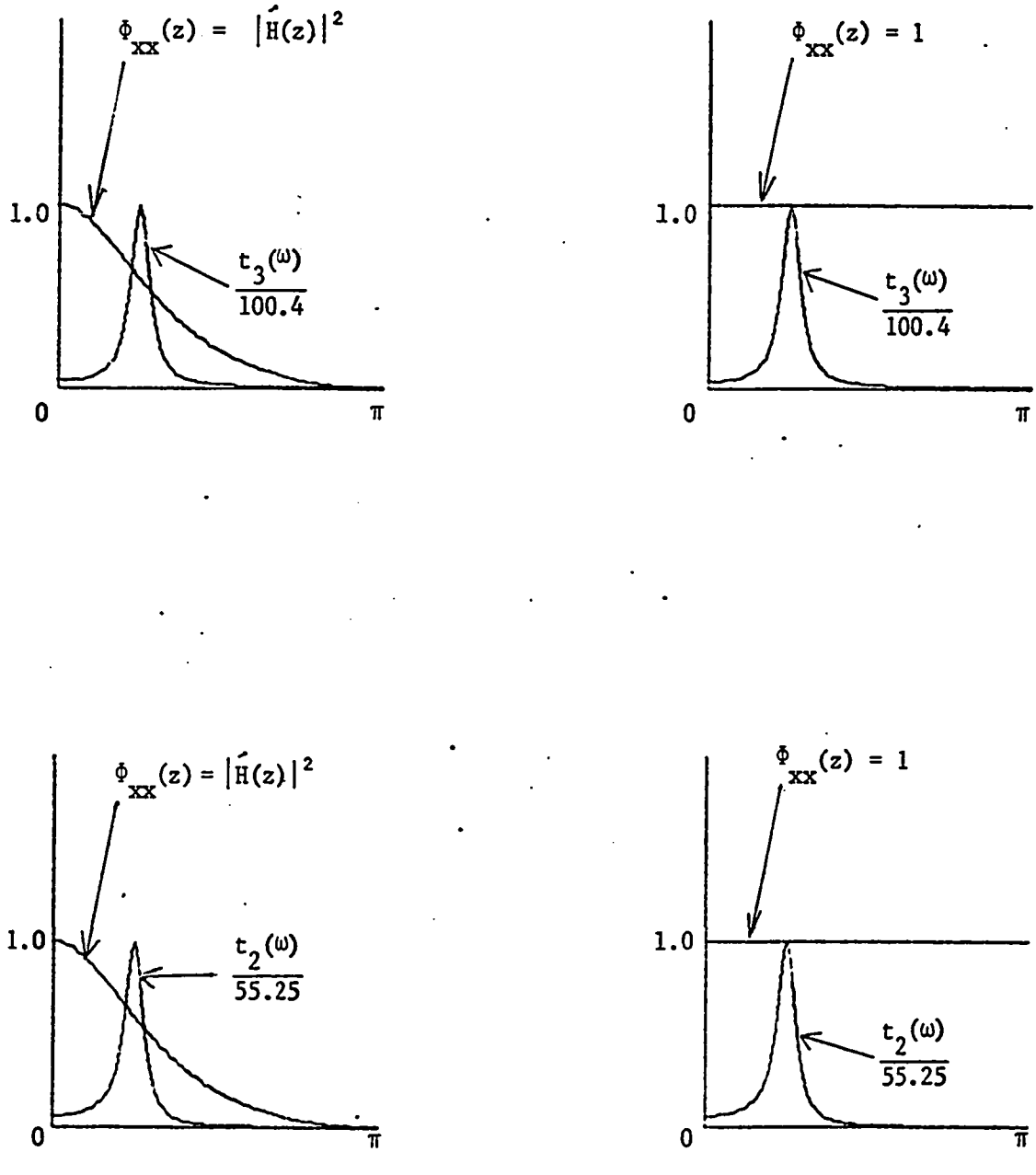


Fig 4.3 Spectrums

ERROR TO SIGNAL RATIO.
INPUT COEFFICIENTS

L= 2 M= 0
A(0)= 1.0000000000000000D 00
A(1)= -1.2727922061357860D 00
A(2)= 8.1000000000000000D-01
B(0)= 1.0000000000000000D 00

RESULTS

	BASE= 2		BASE= 10	
BETA	THEORETICAL	THEORETICAL MAX	THEORETICAL	THEORETICAL MAX
3	8.16463D-02	1.39274D-01	2.89956D-01	4.84720D-01
4	4.06053D-02	6.93833D-02	1.37295D-01	2.33295D-01
5	2.02753D-02	3.46599D-02	6.76563D-02	1.15490D-01
6	1.01343D-02	1.73260D-02	3.37032D-02	5.75998D-02
7	5.06670D-03	8.66249D-03	1.68360D-02	2.87817D-02
8	2.53330D-03	4.33118D-03	8.41602D-03	1.43886D-02
9	1.26664D-03	2.16558D-03	4.20777D-03	7.19400D-03
10	6.33320D-04	1.08279D-03	2.10385D-03	3.59696D-03
11	3.16660D-04	5.41395D-04	1.05192D-03	1.79848D-03
12	1.58330D-04	2.70698D-04	5.25961D-04	8.99238D-04
13	7.91650D-05	1.35349D-04	2.62980D-04	4.49619D-04
14	3.95825D-05	6.76744D-05	1.31490D-04	2.24809D-04
15	1.97912D-05	3.38372D-05	6.57451D-05	1.12405D-04
16	9.89562D-06	1.69186D-05	3.28725D-05	5.62024D-05
17	4.94781D-06	8.45930D-06	1.64363D-05	2.81012D-05
18	2.47390D-06	4.22965D-06	8.21813D-06	1.40506D-05
19	1.23695D-06	2.11482D-06	4.10907D-06	7.02530D-06
20	6.18476D-07	1.05741D-06	2.05453D-06	3.51265D-06
21	3.09238D-07	5.28706D-07	1.02727D-06	1.75632D-06
22	1.54619D-07	2.64353D-07	5.13633D-07	8.78162D-07
23	7.73095D-08	1.32177D-07	2.56817D-07	4.39081D-07
24	3.86548D-08	6.60883D-08	1.28408D-07	2.19540D-07
25	1.93274D-08	3.30441D-08	6.42042D-08	1.09770D-07
26	9.66369D-09	1.65221D-08	3.21021D-08	5.48851D-08
27	4.83185D-09	8.26103D-09	1.60510D-08	2.74426D-08

Table 4.1 Theoretical Error to Signal Ratios

RELATION OF BASE=2 AND BASE=10

BETA	RRR	QQR
3	2.81582D-01	3.00087D-01
4	2.95752D-01	3.00794D-01
5	2.99681D-01	3.00971D-01
6	3.00691D-01	3.01015D-01
7	3.00945D-01	3.01026D-01
8	3.01009D-01	3.01029D-01
9	3.01025D-01	3.01030D-01
10	3.01029D-01	3.01030D-01
11	3.01030D-01	3.01030D-01
12	3.01030D-01	3.01030D-01
13	3.01030D-01	3.01030D-01
14	3.01030D-01	3.01030D-01
15	3.01030D-01	3.01030D-01
16	3.01030D-01	3.01030D-01
17	3.01030D-01	3.01030D-01
18	3.01030D-01	3.01030D-01
19	3.01030D-01	3.01030D-01
20	3.01030D-01	3.01030D-01
21	3.01030D-01	3.01030D-01
22	3.01030D-01	3.01030D-01
23	3.01030D-01	3.01030D-01
24	3.01030D-01	3.01030D-01
25	3.01030D-01	3.01030D-01
26	3.01030D-01	3.01030D-01
27	3.01030D-01	3.01030D-01

$$RRR = \frac{\sqrt{\frac{E[e_n^2]}{E[w_n^2]}}\Big|_{\text{base}=2}}{\sqrt{\frac{E[e_n^2]}{E[w_n^2]}}\Big|_{\text{base}=10}}$$

$$QQR = \sqrt{\frac{q^2_{\text{base}=2}}{q^2_{\text{base}=10}}}$$

Table 4.2 Relation of the Error to Signal Ratios of Base = 2 and Base = 10

ERROR TO SIGNAL RATIO
INPUT COEFFICIENTS

L= 6 M= 6
A(0)= 1.0000000000000000D 00
A(1)= -3.1835917495472570 00
A(2)= 4.6222373189078950 00
A(3)= -3.7794774195233480 00
A(4)= 1.8136046877680020 00
A(5)= -4.7999750020915760-01
A(6)= 5.4445138161848080-02
B(0)= 7.3781993059347660-04
B(1)= 4.4269195835608600-03
B(2)= 1.1067298958902150-02
B(3)= 1.4756398611869530-02
B(4)= 1.1067298958902150-02
B(5)= 4.4269195835608600-03
B(6)= 7.3781993059347660-04

RESULTS

	BASE= 2		BASE= 10	
BETA	THEORETICAL	THEORETICAL MAX	THEORETICAL	THEORETICAL MAX
3	1.07209D 00	3.50163D 00	3.76608D 00	1.17665D 01
4	5.33680D-01	1.74930D 00	1.79901D 00	5.82788D 00
5	2.66543D-01	8.74459D-01	8.88729D-01	2.90700D 00
6	1.33235D-01	4.37206D-01	4.43008D-01	1.45263D 00
7	6.66127D-02	2.18600D-01	2.21334D-01	7.26206D-01
8	3.33058D-02	1.09300D-01	1.10646D-01	3.63090D-01
9	1.66528D-02	5.46498D-02	5.53203D-02	1.81543D-01
10	8.32640D-03	2.73249D-02	2.76598D-02	9.07713D-02
11	4.16320D-03	1.36624D-02	1.38299D-02	4.53856D-02
12	2.08160D-03	6.83122D-03	6.91492D-03	2.26928D-02
13	1.04080D-03	3.41561D-03	3.45746D-03	1.13464D-02
14	5.20400D-04	1.70780D-03	1.72873D-03	5.67320D-03
15	2.60200D-04	8.53902D-04	8.64365D-04	2.83660D-03
16	1.30100D-04	4.26951D-04	4.32183D-04	1.41830D-03
17	6.50500D-05	2.13476D-04	2.16091D-04	7.09151D-04
18	3.25250D-05	1.06738D-04	1.08046D-04	3.54575D-04
19	1.62625D-05	5.33689D-05	5.40228D-05	1.77288D-04
20	8.13124D-06	2.66844D-05	2.70114D-05	8.86438D-05
21	4.06562D-06	1.33422D-05	1.35057D-05	4.43219D-05
22	2.03281D-06	6.67111D-06	6.75285D-06	2.21610D-05
23	1.01641D-06	3.33556D-06	3.37643D-06	1.10805D-05
24	5.08203D-07	1.66778D-06	1.68821D-06	5.54024D-06
25	2.54101D-07	8.33889D-07	8.44107D-07	2.77012D-06
26	1.27051D-07	4.16944D-07	4.22053D-07	1.38506D-06
27	6.35253D-08	2.08472D-07	2.11027D-07	6.92530D-07

* THEORETICAL MAX IS TAKEN FOR 0<=W<=PI/2

Table 4.3 Theoretical Error to Signal Ratios

P= 0.900D 00

BETA	ROUND	ROUND MAX	TRUNC	TRUNC MAX
2	1.667D-01	2.826D-01	8.200D-01	1.153D 00
3	8.165D-02	1.393D-01	4.277D-01	6.015D-01
4	4.061D-02	6.938D-02	2.185D-01	3.073D-01
5	2.028D-02	3.466D-02	1.104D-01	1.553D-01
6	1.013D-02	1.733D-02	5.551D-02	7.807D-02
7	5.067D-03	8.662D-03	2.783D-02	3.914D-02
8	2.533D-03	4.331D-03	1.393D-02	1.960D-02
9	1.267D-03	2.166D-03	6.972D-03	9.806D-03
10	6.333D-04	1.083D-03	3.487D-03	4.904D-03
11	3.167D-04	5.414D-04	1.744D-03	2.453D-03
12	1.583D-04	2.707D-04	8.720D-04	1.226D-03
13	7.916D-05	1.353D-04	4.360D-04	6.132D-04
14	3.958D-05	6.767D-05	2.180D-04	3.066D-04
15	1.979D-05	3.384D-05	1.090D-04	1.533D-04
16	9.896D-06	1.692D-05	5.451D-05	7.666D-05
17	4.948D-06	8.459D-06	2.725D-05	3.833D-05
18	2.474D-06	4.230D-06	1.363D-05	1.916D-05
19	1.237D-06	2.115D-06	6.813D-06	9.582D-06
20	6.185D-07	1.057D-06	3.407D-06	4.791D-06
21	3.092D-07	5.287D-07	1.703D-06	2.396D-06
22	1.546D-07	2.644D-07	8.517D-07	1.198D-06
23	7.731D-08	1.322D-07	4.258D-07	5.989D-07
24	3.865D-08	6.609D-08	2.129D-07	2.994D-07
25	1.933D-08	3.304D-08	1.065D-07	1.497D-07
26	9.664D-09	1.652D-08	5.323D-08	7.486D-08
27	4.832D-09	8.261D-09	2.661D-08	3.743D-08
28	2.416D-09	4.131D-09	1.331D-08	1.872D-08
29	1.208D-09	2.065D-09	6.654D-09	9.358D-09
30	6.040D-10	1.033D-09	3.327D-09	4.679D-09

THEORETICAL ERROR TO SIGNAL RATIOS BASE = 2

Table 4.4

P= 0.999D 00

BETA	ROUND	ROUND MAX	TRUNC	TRUNC MAX
2	4.076D 00	6.064D 00	7.957D 01	1.125D 02
3	1.228D 00	1.974D 00	4.150D 01	5.870D 01
4	4.599D-01	8.023D-01	2.120D 01	2.999D 01
5	2.063D-01	3.745D-01	1.072D 01	1.516D 01
6	9.997D-02	1.838D-01	5.387D 00	7.619D 00
7	4.958D-02	9.145D-02	2.701D 00	3.820D 00
8	2.474D-02	4.567D-02	1.352D 00	1.913D 00
9	1.236D-02	2.283D-02	6.766D-01	9.569D-01
10	6.181D-03	1.141D-02	3.384D-01	4.786D-01
11	3.090D-03	5.706D-03	1.692D-01	2.393D-01
12	1.545D-03	2.853D-03	8.462D-02	1.197D-01
13	7.726D-04	1.427D-03	4.231D-02	5.984D-02
14	3.863D-04	7.133D-04	2.116D-02	2.992D-02
15	1.931D-04	3.566D-04	1.058D-02	1.496D-02
16	9.657D-05	1.783D-04	5.289D-03	7.481D-03
17	4.829D-05	8.916D-05	2.645D-03	3.740D-03
18	2.414D-05	4.458D-05	1.322D-03	1.870D-03
19	1.207D-05	2.229D-05	6.611D-04	9.351D-04
20	6.036D-06	1.115D-05	3.306D-04	4.676D-04
21	3.018D-06	5.573D-06	1.653D-04	2.338D-04
22	1.509D-06	2.786D-06	8.264D-05	1.169D-04
23	7.545D-07	1.393D-06	4.132D-05	5.844D-05
24	3.772D-07	6.966D-07	2.066D-05	2.922D-05
25	1.886D-07	3.483D-07	1.033D-05	1.461D-05
26	9.431D-08	1.741D-07	5.165D-06	7.306D-06
27	4.716D-08	8.707D-08	2.583D-06	3.653D-06
28	2.358D-08	4.354D-08	1.291D-06	1.826D-06
29	1.179D-08	2.177D-08	6.457D-07	9.132D-07
30	5.894D-09	1.088D-08	3.228D-07	4.566D-07

THEORETICAL ERROR TO SIGNAL RATIOS BASE = 2

Table 4.5

```
ERROR TO SIGNAL RATIO
EXPERIMENTAL
   N=  8000   STATISTICS COLLECTED AFTER    150
   L=   2     M=   0
A(  0)=    1.000000000000000E+00
A(  1)=   -1.272792206135786E+00
A(  2)=    8.099999999999998E-01
B(  0)=    1.000000000000000E+00
```

NO.	ALPHA	BETA	BASE= 2 RATIO	OV	UN	BASE= 10 RATIO	OV	UN
1	7	7	5.10727E-03	0	27	1.69877E-02	0	66

* THE RATIO OF 1.00000E+03. IF ANY. INDICATES THAT THE LOGARITHMIC FILTER PRODUCED TOO MUCH ERROR.

* THOSE UNDER OV OR UN INDICATE THE NUMBER OF TIMES THAT OVERFLOW OR UNDERFLOW OCCURRED IN THE LOGARITHMIC FILTER.

* #. IF ANY. INDICATES THAT UNDERFLOW OCCURRED IN THE LONG FLOATING POINT FILTER.
WITH THE RESULT OF ZERO OPERATION CONTINUED.

Table 4.6 Experimental Error to Signal Ratios

LOGARITHMIC ADDITION TABLE

BASE= 2 8-BIT WORD 3-BIT FRACTIONAL PART

8	8	7	7	6	6	5	5	5	4	4	4	3	3	3	3
3	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
127	29	21	17	14	12	10	9	8	7	6	6	5	5	4	4
3	3	3	2	2	2	2	2	2	1	1	1	1	1	1	1
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.7 Sample Look-up Table

ERROR TO SIGNAL RATIO
EXPERIMENTAL
N= 400 STATISTICS COLLECTED AFTER 150
L= 2 M= 0
A(0)= 1.000000000000000E+00
A(1)= -1.272792206135786E+00
A(2)= 8.099999999999998E-01
B(0)= 1.000000000000000E+00

NO.	ALPHA	BETA	BASE= 2 RATIO	OV	UN	BASE= 10 RATIO	OV	UN
1	4	2	1.64805E-01	0	24	1.00000E+03	0	0
2	3	3	9.22432E-02	0	12	3.34434E-01	0	45
3	2	4	4.62585E-02	0	32	2.19483E-01	0	20
4	1	5	6.52919E-01	81	88	8.43133E-02	0	8
5	0	6	8.19413E-01	257	239	4.06916E-02	0	29
6	3	4	4.03250E-02	0	11	2.19483E-01	0	20
7	4	4	4.03088E-02	0	9	2.19483E-01	0	20
8	5	4	4.03088E-02	0	9	2.19483E-01	0	20
9	5	5	1.86052E-02	0	2	8.47985E-02	0	10
10	5	6	1.07065E-02	0	1	3.79105E-02	0	4
11	5	7	6.08334E-03	0	0	1.67491E-02	0	2
12	5	8	2.08043E-03	0	0	7.52678E-03	0	2
13	8	6	1.07065E-02	0	1	3.79105E-02	0	4
14	7	7	6.08334E-03	0	0	1.67491E-02	0	2
15	6	8	2.08043E-03	0	0	7.52678E-03	0	2
16	5	9	1.18924E-03	0	0	4.15961E-03	0	0
17	4	10	5.12106E-04	0	1	2.18760E-03	0	3
18	3	11	3.15692E-04	0	0	9.42029E-04	0	0
19	2	12	5.43300E-03	0	23	5.58489E-04	0	0
20	1	13	6.80944E-01	83	99	4.26464E-04	0	3
21	0	14	8.13813E-01	248	232	1.58354E-02	0	35
22	7	23	7.66544E-08	0	0	2.94870E-07	0	0

* THE RATIO OF 1.00000E+03. IF ANY. INDICATES THAT THE LOGARITHMIC FILTER PRODUCED TOO MUCH ERROR.

* THOSE UNDER OV OR UN INDICATE THE NUMBER OF TIMES THAT OVERFLOW OR UNDERFLOW OCCURRED IN THE LOGARITHMIC FILTER.

* #. IF ANY. INDICATES THAT UNDERFLOW OCCURRED IN THE LONG FLOATING POINT FILTER.
WITH THE RESULT OF ZERO OPERATION CONTINUED.

Table 4.8 Experimental Error to Signal Ratios

Theoretical		Experimental					
β	ratio		α	β	ratio	OV	UN
3	8.16463×10^{-2}	8 bit word	4	2	1.64805×10^{-1}	0	24
4	4.06053×10^{-2}		3	3	9.22432×10^{-2}	0	12
5	2.02753×10^{-2}		2	4 *	4.62585×10^{-2}	0	32
6	1.01343×10^{-2}		1	5	6.52919×10^{-1}	81	88
7	5.06670×10^{-3}		0	6	8.19413×10^{-1}	257	239
8	2.53330×10^{-3}		3	4	4.03250×10^{-2}	0	11
9	1.26664×10^{-3}		4	4	4.03088×10^{-2}	0	9
10	6.33320×10^{-4}		5	4	4.03088×10^{-2}	0	9
11	3.16660×10^{-4}		5	5	1.86052×10^{-2}	0	2
12	1.58330×10^{-4}		5	6	1.07065×10^{-2}	0	1
13	7.91650×10^{-5}		5	7	6.08334×10^{-3}	0	0
14	3.95825×10^{-5}		5	8	2.08043×10^{-3}	0	0
23	7.73095×10^{-8}	16 bit word	8	6	1.07065×10^{-2}	0	1
			7	7	6.08334×10^{-3}	0	0
			6	8	2.08043×10^{-3}	0	0
			5	9	1.18924×10^{-3}	0	0
			4	10	5.12106×10^{-4}	0	1
			3	11 *	3.15692×10^{-4}	0	0
			2	12	5.43300×10^{-3}	0	23
			1	13	6.80944×10^{-1}	83	99
			0	14	8.13813×10^{-1}	248	232
			7	23	7.66544×10^{-8}	0	0

Table 4.9 Comparison between theoretical and experimental ratios for Base = 2

ERROR TO SIGNAL RATIO
EXPERIMENTAL
N= 400 STATISTICS COLLECTED AFTER 150
L= 6 M= 6

```
A( 0)=   1.000000000000000E+00
A( 1)=  -3.183591749547257E+00
A( 2)=   4.622237318907895E+00
A( 3)=  -3.779477419523348E+00
A( 4)=   1.813604687768002E+00
A( 5)=  -4.799975002091575E-01
A( 6)=   5.444513816184809E-02
B( 0)=   7.378199305934766E-04
B( 1)=   4.426919583560859E-03
B( 2)=   1.106729895890215E-02
B( 3)=   1.475639861186953E-02
B( 4)=   1.106729895890215E-02
B( 5)=   4.426919583560859E-03
B( 6)=   7.378199305934766E-04
```

NO.	ALPHA	BETA	BASE= 2			BASE= 10		
			RATIO	OV	UN	RATIO	OV	UN
1	3	3	1.00000E+03	0	43	1.00000E+03	0	0
2	4	4	1.00000E+03	0	2	1.00000E+03	0	0
3	4	5	1.00000E+03	0	2	1.00000E+03	0	2
4	4	6	1.96056E-01	0	15	1.00000E+03	0	0
5	4	7	7.48502E-02	0	13	2.31887E-01	0	11
6	4	8	2.93780E-02	0	9	1.29876E-01	0	7
7	4	9	1.67517E-02	0	8	4.71781E-02	0	1
8	8	6	1.96056E-01	0	7	1.00000E+03	0	0
9	7	7	7.48502E-02	0	4	2.31887E-01	0	11
10	6	8	2.93780E-02	0	1	1.29876E-01	0	7
11	5	9	1.67517E-02	0	0	4.71781E-02	0	1
12	4	10	7.58597E-03	0	10	3.07841E-02	0	2
13	3	11	1.30671E-01	0	1467	1.37965E-02	0	0
14	2	12	1.00000E+03	22	79	7.17134E-03	0	82
15	1	13	1.00000E+03	6	32	3.20201E-01	0	2026
16	0	14	1.00000E+03	5	25	1.00000E+03	17	60
17	7	23	9.22511E-07	0	0	3.62876E-06	0	0

* THE RATIO OF 1.00000E+03, IF ANY, INDICATES THAT THE LOGARITHMIC FILTER PRODUCED TOO MUCH ERROR.

* THOSE UNDER OV OR UN INDICATE THE NUMBER OF TIMES THAT OVERFLOW OR UNDERFLOW OCCURRED IN THE LOGARITHMIC FILTER.

* #, IF ANY, INDICATES THAT UNDERFLOW OCCURRED IN THE LONG FLOATING POINT FILTER.
WITH THE RESULT OF ZERO OPERATION CONTINUED.

Table 4.10 Experimental Error to Signal Ratios

ERROR TO SIGNAL RATIO
INPUT COEFFICIENTS

L= 2 M= 0
A(0)= 1.0000000000000000D 00
A(1)= -1.2727922061357860D 00
A(2)= 8.1000000000000000D-01
B(0)= 1.0000000000000000D 00

RESULTS

	BASE= 2		BASE= 10	
BETA	THEORETICAL	THEORETICAL MAX	THEORETICAL	THEORETICAL MAX
3	7.92522D-02	1.09420D-01	2.81921D-01	3.90521D-01
4	3.54090D-02	5.43546D-02	1.33312D-01	1.84177D-01
5	1.96773D-02	2.71577D-02	6.56685D-02	9.06548D-02
6	9.83522D-03	1.35739D-02	3.27097D-02	4.51466D-02
7	4.91719D-03	6.78634D-03	1.63393D-02	2.25506D-02
8	2.45854D-03	3.39309D-03	8.16768D-03	1.12725D-02
9	1.22926D-03	1.69654D-03	4.08360D-03	5.63588D-03
10	6.14631D-04	8.48268D-04	2.04177D-03	2.81790D-03
11	3.07315D-04	4.24134D-04	1.02088D-03	1.40894D-03
12	1.53658D-04	2.12067D-04	5.10440D-04	7.04471D-04
13	7.68288D-05	1.06033D-04	2.55220D-04	3.52235D-04
14	3.84144D-05	5.30167D-05	1.27610D-04	1.76118D-04
15	1.92072D-05	2.65083D-05	6.38050D-05	8.80588D-05
16	9.60360D-06	1.32542D-05	3.19025D-05	4.40294D-05
17	4.80180D-06	6.62709D-06	1.59512D-05	2.20147D-05
18	2.40090D-06	3.31354D-06	7.97562D-06	1.10074D-05
19	1.20045D-06	1.65677D-06	3.98781D-06	5.50368D-06
20	6.00225D-07	8.28386D-07	1.99390D-06	2.75184D-06
21	3.00113D-07	4.14193D-07	9.96952D-07	1.37592D-06
22	1.50056D-07	2.07096D-07	4.98476D-07	6.87960D-07
23	7.50281D-08	1.03548D-07	2.49238D-07	3.43980D-07
24	3.75141D-08	5.17741D-08	1.24619D-07	1.71990D-07
25	1.87570D-08	2.58871D-08	6.23095D-08	8.59950D-08
26	9.37852D-09	1.29435D-08	3.11548D-08	4.29975D-08
27	4.68926D-09	6.47176D-09	1.55774D-08	2.14987D-08

* THEORETICAL MAX IS TAKEN FOR 0<=W<=PI/2

Table 4.11 Theoretical Error to Signal Ratios with Input Sequence of the Spectrum other than the Constant One

ERROR TO SIGNAL RATIO
EXPERIMENTAL
N= 400 STATISTICS COLLECTED AFTER 150
L= 2 M= 0
A(0)= 1.000000000000000E+00
A(1)= -1.272792206135786E+00
A(2)= 8.099999999999998E-01
B(0)= 1.000000000000000E+00

NO.	ALPHA	BETA	BASE= 2 RATIO	OV	UN	BASE= 10 RATIO	OV	UN
1	4	2	1.42442E-01	0	24	1.00000E+03	0	15
2	3	3	7.70268E-02	0	13	3.35838E-01	0	29
3	2	4	4.43382E-02	0	17	1.51679E-01	0	19
4	1	5	4.68571E-01	44	102	7.23629E-02	0	11
5	0	6	7.43865E-01	205	261	3.61641E-02	0	35
6	3	4	4.35654E-02	0	4	1.51678E-01	0	19
7	4	4	4.35647E-02	0	4	1.51678E-01	0	19
8	5	4	4.35647E-02	0	4	1.51678E-01	0	19
9	5	5	1.83349E-02	0	1	7.65309E-02	0	6
10	5	6	1.03715E-02	0	0	3.42846E-02	0	7
11	5	7	5.51634E-03	0	2	1.48421E-02	0	3
12	5	8	2.40542E-03	0	0	8.48998E-03	0	6
13	8	6	1.03715E-02	0	0	3.42846E-02	0	7
14	7	7	5.51634E-03	0	2	1.48421E-02	0	3
15	6	8	2.40542E-03	0	0	8.48998E-03	0	6
16	5	9	1.40099E-03	0	0	4.27003E-03	0	0
17	4	10	6.07245E-04	0	0	1.91977E-03	0	0
18	3	11	2.71471E-04	0	2	9.65395E-04	0	0
19	2	12	7.75585E-03	0	27	5.61917E-04	0	0
20	1	13	4.82850E-01	44	104	2.96075E-04	0	5
21	0	14	7.45153E-01	207	261	1.60810E-02	0	44
22	7	23	9.15156E-08	0	0	2.28407E-07	0	0

* THE RATIO OF 1.00000E+03, IF ANY, INDICATES THAT THE LOGARITHMIC FILTER PRODUCED TOO MUCH ERROR.

* THOSE UNDER OV OR UN INDICATE THE NUMBER OF TIMES THAT OVERFLOW OR UNDERFLOW OCCURRED IN THE LOGARITHMIC FILTER.

* #, IF ANY, INDICATES THAT UNDERFLOW OCCURRED IN THE LONG FLOATING POINT FILTER,
WITH THE RESULT OF ZERO OPERATION CONTINUED.

Table 4.12 Experimental Error to Signal Ratios with Input Sequence of the Spectrum other than the Constant One

P=. 0.9900 00

BETA	ROUND	ROUND MAX	TRUNC	TRUNC MAX
2	6.271D-01	1.063D 00	7.969D 00	1.128D 01
3	2.678D-01	4.789D-01	4.157D 00	5.885D 00
4	1.276D-01	2.324D-01	2.124D 00	3.006D 00
5	6.296D-02	1.153D-01	1.073D 00	1.519D 00
6	3.138D-02	5.754D-02	5.396D-01	7.638D-01
7	1.568D-02	2.876D-02	2.705D-01	3.833D-01
8	7.836D-03	1.438D-02	1.354D-01	1.917D-01
9	3.918D-03	7.188D-03	6.777D-02	9.593D-02
10	1.959D-03	3.594D-03	3.389D-02	4.798D-02
11	9.795D-04	1.797D-03	1.695D-02	2.400D-02
12	4.897D-04	8.985D-04	8.476D-03	1.200D-02
13	2.449D-04	4.492D-04	4.238D-03	6.000D-03
14	1.224D-04	2.246D-04	2.119D-03	3.000D-03
15	6.122D-05	1.123D-04	1.060D-03	1.500D-03
16	3.061D-05	5.615D-05	5.298D-04	7.500D-04
17	1.530D-05	2.808D-05	2.649D-04	3.750D-04
18	7.652D-06	1.404D-05	1.324D-04	1.875D-04
19	3.826D-06	7.019D-06	6.622D-05	9.375D-05
20	1.913D-06	3.510D-06	3.311D-05	4.687D-05
21	9.565D-07	1.755D-06	1.656D-05	2.344D-05
22	4.783D-07	8.774D-07	8.278D-06	1.172D-05
23	2.391D-07	4.387D-07	4.139D-06	5.859D-06
24	1.196D-07	2.194D-07	2.069D-06	2.930D-06
25	5.978D-08	1.097D-07	1.035D-06	1.465D-06
26	2.989D-08	5.484D-08	5.174D-07	7.324D-07
27	1.495D-08	2.742D-08	2.587D-07	3.662D-07
28	7.473D-09	1.371D-08	1.293D-07	1.831D-07
29	3.736D-09	6.855D-09	6.467D-08	9.155D-08
30	1.868D-09	3.427D-09	3.233D-08	4.578D-08

THEORETICAL ERROR TO SIGNAL RATIOS Base = 2

Table 4.13

ERROR TO SIGNAL RATIO
EXPERIMENTAL
N= 4000 STATISTICS COLLECTED AFTER 150
L= 2 M= 0
A(0)= 1.0000000000000000E+00
A(1)= -1.4000714267493b4E+00
A(2)= 9.8000999999999998E-01
B(0)= 1.0000000000000000E+00

NO.	ALPHA	BETA	BASE= 2 RATIO	CV	UN	BASE= 10 RATIO	CV	UN
1	7	7	1.66716E-02	0	16	5.05967E-02	0	39
2	6	8	7.94774E-03	0	6	2.61207E-02	0	13
3	5	9	3.52924E-03	0	0	1.63080E-02	0	8
4	4	10	2.17642E-03	0	0	6.23495E-03	0	3
5	7	23	2.56128E-07	0	0	6.91874E-07	0	0

* THE RATIO OF 1.00000E+03, IF ANY, INDICATES THAT THE LOGARITHMIC FILTER PRODUCED TOO MUCH ERROR.

* THOSE UNDER CV OR UN INDICATE THE NUMBER OF TIMES THAT OVERFLOW OR UNDERFLOW OCCURRED IN THE LOGARITHMIC FILTER.

* #, IF ANY, INDICATES THAT UNDERFLOW OCCURRED IN THE LONG FLOATING POINT FILTER, WITH THE RESULT OF ZERO OPERATION CONTINUED.

Table 4.14 Experimental Test for a Filter (Q = 39.3)

```
ERROR TO SIGNAL RATIO
EXPERIMENTAL
   N=  4000    STATISTICS COLLECTED AFTER    150
   L=   2      M=   0
A(  0)=    1.000000000000000E+00
A(  1)=   -1.412799348810722E+00
A(  2)=    9.800009999999998E-01
B(  0)=    1.000000000000000E+00
```

NO.	ALPHA	BETA	BASE= 2 RATIO	OV	UN	BASE= 10 RATIO	OV	UN
1	7	7	1.00000E+03	0	9	2.15885E-01	0	23
2	6	8	2.98476E-02	0	8	1.55473E-01	0	10
3	5	9	1.80897E-02	0	1	1.00000E+03	0	2
4	4	10	5.63633E-03	0	0	1.87013E-02	0	2
5	7	23	6.02181E-07	0	0	2.61476E-06	0	0

* THE RATIO OF 1.00000E+03, IF ANY, INDICATES THAT THE LOGARITHMIC FILTER PRODUCED TOO MUCH ERROR.

* THOSE UNDER OV OR UN INDICATE THE NUMBER OF TIMES THAT OVERFLOW OR UNDERFLOW OCCURRED IN THE LOGARITHMIC FILTER.

* #, IF ANY, INDICATES THAT UNDERFLOW OCCURRED IN THE LONG FLOATING POINT FILTER, WITH THE RESULT OF ZERO OPERATION CONTINUED.

Table 4.15 Experimental Test for a Filter (Q = 393)

```
ERROR TO SIGNAL RATIO
INPUT COEFFICIENTS


L=  2      M=  0
A( 0)=     1.0000000000000000D 00
A( 1)=    -1.4127993488107220 00
A( 2)=     9.9800099999999980-01
B( 0)=     1.0000000000000000 00

RESULTS

                BASE=  2

 ALPHA BETA  THEORETICAL  THEORETICAL MAX
     5    9  1.50198D-02       2.77275D-02
```

Table 4.16 Theoretical Error Ratio with Converted Filter Coefficients with Logarithmic Number

```
ERROR TO SIGNAL RATIO
EXPERIMENTAL
   N= 12000   STATISTICS COLLECTED AFTER   3000
   L=   2     M=   0
A(  0)=    1.0000000000000000E+00
A(  1)=   -1.4127993488107220E+00
A(  2)=    9.9800099999999980E-01
B(  0)=    1.0000000000000000E+00

NO. ALPHA BETA       BASE=  2                    BASE= 10
                      RATIO     OV   UN         RATIO     OV   UN
 1      5    9  1.69132E-02      0    5  1.38428E-01      0   25

* THE RATIO OF     1.00000E+03. IF ANY. INDICATES THAT THE
  LOGARITHMIC FILTER PRODUCED TOO MUCH ERROR.

* THOSE UNDER OV OR UN INDICATE THE NUMBER OF TIMES THAT
  OVERFLOW OR UNDERFLOW OCCURRED IN THE LOGARITHMIC FILTER.

* #. IF ANY. INDICATES THAT UNDERFLOW OCCURRED IN THE
  LONG FLOATING POINT FILTER.
  WITH THE RESULT OF ZERO OPERATION CONTINUED.
```

Table 4.17 Experimental Test for a Filter (Q = 393) for a Large Number of Inputs

APPENDIX 4.1

```
C
C PROGRAM FOR COMPUTING THE THEORETICAL VALUES AND MAXIMUM BOUNDS
C OF ERROR TO SIGNAL RATIO OF DIGITAL FILTERS USING THE SIMPSON'S
C RULE OF NUMERICAL INTEGRATION
C
C INPUT DATA CARDS
C CARD 1
C COLUMN 1--5: DEGREE OF D(Z), L
C COLUMN 6--10: DEGREE OF N(Z), M
C COLUMN 11--15  1---MAX IS COMPUTED FOR 0<=W<=PI
C                2---MAX IS COMPUTED FOR 0<=W<=PI/2
C CARDS 2--L+2 COLUMN 1--25: COEFFICIENTS, A(0)--A(L)
C CARDS L+3--L+M+3 COLUMN 1--25: COEFFICIENTS, B(0)--B(M)
C
C
C  L,M: DEGREES OF D(Z) AND N(Z) RESPECTIVELY
C  A(I),B(I):  COEFFICIENTS OF THE FILTER
C  PRM1: MAXIMUM VALUE OF FUNCTION R3
C  PRM2: MAXIMUM VALUE OF FUNCTION R4
C  S(1),S(2),S(3) AND S(4) WILL HAVE THE INTEGRATION VALUES OF
C  FUNCTIONS R1;R2,R3 AND R4
C  TH(1): ERROR TO SIGNAL RATIO FOR BASE=2
C  THM(1): MAXIMUM VALUE OF TH(1)
C  TH(2): ERROR TO SIGNAL RATIO FOR BASE=10
C  THM(2): MAXIMUM VALUE OF TH(2)
C  BETA: NUMBER OF BITS AFTER BINARY POINT
C  BS: BASE OF LOGARITHM
C  QS: VARIANCE OF ERROR
C  MES: SQUARED VALUE OF ERROR MEAN
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      REAL*8 S(10),DS(10),PS(10),TH(2),THM(2)
      REAL*8 BS(2),QS(2),MES(2),QQR(30),RRR(30)
      EXTERNAL R1,R2,R3,R4
      N2=2
      N10=10
      PRM1=0.0D0
      PRM2=0.0D0
      EPS=1.0D-3
      PI=3.14159265358979300
C
C READ COEFFICIENTS
C
      READ(5,50) L,M,ITYPE
   50 FORMAT(3I5)
      LP1=L+1
      MP1=M+1
      DO 100 I=1,LP1
      READ(5,101) A(I)
  100 WRITE(6,102) A(I)
```

```
  101 FORMAT(D25.16)
  102 FORMAT(1X,D25.16)
      DO 110 I=1,MP1
      READ(5,101) B(I)
  110 WRITE(6,102) B(I)
C
C DO THE INTEGRATIONS
C
      DO 530 I=1,4
      PS(I)=0.0D0
      N=50
      DO 510 J=1,10
      N=N*J*2
      IF (I.NE.1) GO TO 511
      S(I)=SIMPS(0.0D0,PI,N,R1)
      GO TO 520
  511 IF (I.NE.2) GO TO 512
      S(I)=SIMPS(0.0D0,PI,N,R2)
      GO TO 520
  512 IF (I.NE.3) GO TO 513
      S(I)=SIMPS(0.0D0,PI,N,R3)
      GO TO 520
  513 S(I)=SIMPS(0.0D0,PI,N,R4)
  520 S(I)=S(I)/PI
      DS(I)=S(I)-PS(I)
      WRITE(6,201) N,S(I),DS(I),PRM1,PRM2
  201 FORMAT(1X,I10,4D20.5)
C
C CHECK FOR CONVERGENCE
C
      IF (DABS(DS(I)/S(I)).LT.EPS) GO TO 521
      PS(I)=S(I)
  510 CONTINUE
  521 WRITE(6,202)
  202 FORMAT(' ','CONVERGE')
  530 CONTINUE
C
C COMPUTES THE RESULTS FOR EACH BETA AND OUTPUTS
C
      WRITE(6,888)
  888 FORMAT('1'/////' ')
      WRITE(6,601)
  601 FORMAT('0',5X,'ERROR TO SIGNAL RATIO'/6X,'INPUT COEFFICIENTS'/)
      WRITE(6,602) L,M
  602 FORMAT('0',5X,'L=',I3,5X,'M=',I3)
      DO 610 I=1,LP1
      I1=I-1
  610 WRITE(6,603) I1,A(I)
  603 FORMAT(1X,5X,'A(',I2,')=',1PD25.15)
      DO 620 I=1,MP1
      I1=I-1
  620 WRITE(6,604) I1,B(I)
  604 FORMAT(1X,5X,'B(',I2,')=',1PD25.15)
      WRITE(6,619)
```

```
  619 FORMAT('0',5X,'RESULTS')
      WRITE(6,605)  N2,N10
  605 FORMAT('0',17X,'BASE=',I3,26X,'BASE=',I3)
      WRITE(6,606)
  606 FORMAT('0',6X,'BETA',2X,'THEORETICAL',2X,'THEORETICAL MAX',
     12X,'THEORETICAL',2X,'THEORETICAL MAX')
      BS(1)=2.0D0
      BS(2)=10.0D0
      DO 160 I=3,27
      BETA=I
      AA=2.0D0**(-BETA-1.0D0)
      DO 155 K=1,2
      TEMP=BS(K)**AA-BS(K)**(-AA)
      QS(K)=TEMP**2/12.0D0
      TEMP=BS(K)**AA+BS(K)**(-AA)-2.0D0
      MES(K)=(TEMP/2.0D0)**2
      TH(K)=QS(K)*S(1)*S(3)/S(2)+MES(K)*S(4)/S(2)
      THM(K)=QS(K)*S(1)*PRM1+MES(K)*PRM2
      TH(K)=DSQRT(TH(K))
  155 THM(K)=DSQRT(THM(K))
      QQR(I)=DSQRT(QS(1)/QS(2))
      RRR(I)=TH(1)/TH(2)
      WRITE(6,701) I,TH(1),THM(1),TH(2),THM(2)
  701 FORMAT(1X,I10,2(1PD13.5,1PD17.5))
  160 CONTINUE
      IF (ITYPE.NE.2) GO TO 800
      WRITE(6,709)
  709 FORMAT('0',5X,'* THEORETICAL MAX IS TAKEN FOR 0<=W<=PI/2')
  800 WRITE(6,901)
  901 FORMAT('1'/////5X,'RELATION OF BASE=2 AND BASE=10')
      WRITE(6,903)
  903 FORMAT('0',6X,'BETA',12X,'RRR',12X,'QQR')
      DO 850 I=3,27
  850 WRITE(6,902) I,RRR(I),QQR(I)
  902 FORMAT(1X,5X,I5,2(1PD15.5))
      STOP
      END
C
C
      FUNCTION SIMPS(P,Q,N,F)
C
C THIS SUBROUTINE USES N APPLICATIONS OF SIMPSON'S RULE TO COMPUTE THE
C INTEGRAL OF F(X)*DX BETWEEN P AND Q
C
      IMPLICIT REAL*8(A-H,O-Z)
      FN=N
      TWOH=(Q-P)/FN
      H=TWOH/2.0D0
      SUMEND=0.0D0
      SUMMID=0.0D0
      DO 1 K=1,N
      FK1=K-1
      X=P+FK1*TWOH
      SUMEND=SUMEND+F(X)
```

```
    1 SUMMID=SUMMID+F(X+H)
      SIMPS=(2.0D0*SUMEND+4.0D0*SUMMID-F(P)+F(Q))*H/3.0D0
      RETURN
      END
C
C
      FUNCTION R1(W)
C
C THIS FUNCTION COMPUTES THE FUNCTIONAL VALUE INSIDE THE
C INTEGRATION OF S1 GIVEN IN THE EQUATION (4.1) IN SECTION 4.1
C FOR EACH VALUE OF W
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
      COMPLEX*16 Z,DZ
      LP1=L+1
      Z=CDEXP(DCMPLX(0.0D0,W))
      DZ=DCMPLX(0.0D0,0.0D0)
      DO 100 K=1,LP1
  100 DZ=DZ+A(K)*Z**(-(K-1))
      R1=CDABS(DZ)**2
      R1=1.0D0/R1
      RETURN
      END
C
C
      FUNCTION R2(W)
C
C THIS FUNCTION COMPUTES THE FUNCTIONAL VALUE INSIDE THE INTEGRATION
C OF S2 GIVEN IN THE EQUATION (4.1) IN SECTION 4.1 FOR EACH VALUE OF W
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
      COMPLEX*16 DZ,NZ,Z
      LP1=L+1
      MP1=M+1
      Z=CDEXP(DCMPLX(0.0D0,W))
      DZ=DCMPLX(0.0D0,0.0D0)
      NZ=DZ
      DO 100 K=1,LP1
  100 DZ=DZ+A(K)*Z**(-(K-1))
      DO 102 K=1,MP1
  102 NZ=NZ+B(K)*Z**(-(K-1))
      R2=CDABS(NZ/DZ)**2
      RETURN
      END
C
C
      FUNCTION R3(W)
C
C THIS FUNCTION COMPUTES THE FUNCTIONAL VALUE INSIDE THE INTEGRATION
C OF S3 GIVEN IN THE EQUATION (4.1) IN SECTION 4.1 FOR EACH VALUE OF W
```

```
C AND FINDS THE MAXIMUM VALUE M1 GIVEN IN THE INEQUALITY (4.2)
C AND STORES IT IN PRM1
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      COMPLEX*16 Z,DZ,NZ,CZ,BZ2,AZ2
      PI=3.14159265358979D0
      LP1=L+1
      MP1=M+1
      Z=CDEXP(DCMPLX(0.0D0,W))
      DZ=DCMPLX(0.0D0,0.0D0)
      NZ=DZ
      CZ=DZ
      BZ2=DZ
      AZ2=DZ
      DO 100 K=1,LP1
  100 DZ=DZ+A(K)*Z**(-(K-1))
      DO 102 K=1,MP1
  102 NZ=NZ+B(K)*Z**(-(K-1))
      DO 104 K=1,MP1
      DO 104 I=2,LP1
  104 CZ=CZ+B(K)*A(I)*Z**(K-I)
      DO 110 K=1,MP1
      DO 110 I=1,MP1
  110 BZ2=BZ2+B(K)*B(I)*BKI(K,I)*Z**(K-I)
      DO 112 K=2,LP1
      DO 112 I=2,LP1
  112 AZ2=AZ2+A(K)*A(I)*AKI(K,I)*Z**(K-I)
      R31=CDABS(BZ2)
      R321=CDABS(NZ/DZ)**2
      R32=CDABS(AZ2*R321)
      R33=DREAL(CZ*NZ/DZ)
      R3=R31+R32-2.0D0*R33
      RM1=R3/R321
      IF (ITYPE.NE.2) GO TO 115
      IF (W.GT.PI/2.0D0) RETURN
  115 IF (RM1.GT.PRM1) PRM1=RM1
      RETURN
      END
C
C
      FUNCTION R4(W)
C
C THIS FUNCTION COMPUTES THE FUNCTIONAL VALUE INSIDE THE INTEGRATION
C OF S4 GIVEN IN THE EQUATION (4.1) IN SECTION 4.1 FOR EACH VALUE OF W
C AND FINDS THE MAXIMUM VALUE M2 GIVEN IN THE INEQUALITY (4.2)
C AND STORE IT IN PRM2
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
```

```
      COMPLEX*16 Z,DZ,NZ,BPZ,APZ
      PI=3.14159265358979300
      LP1=L+1
      MP1=M+1
      Z=CDEXP(DCMPLX(0.0D0,W))
      DZ=DCMPLX(0.0D0,0.0D0)
      NZ=DZ
      BPZ=DZ
      APZ=DZ
      DO 100 K=1,LP1
  100 DZ=DZ+A(K)*Z**(-(K-1))
      DO 102 K=1,MP1
  102 NZ=NZ+B(K)*Z**(-(K-1))
      DO 106 K=1,MP1
  106 BPZ=BPZ+B(K)*BETA(K)*Z**(-(K-1))
      DO 108 K=2,LP1
  108 APZ=APZ+A(K)*ALPH(K)*Z**(-(K-1))
      R40=CDABS(NZ)**2
      R41=CDABS(DZ)**2
      R42=CDABS(BPZ-NZ*APZ/DZ)**2
      R4=R42/R41
      RM2=R42/R40
      IF (ITYPE.NE.2) GO TO 114
      IF (W.GT.PI/2.0D0) RETURN
  114 IF (RM2.GT.PRM2) PRM2=RM2
      RETURN
      END
C
C
      FUNCTION BKI(K,I)
C
C THIS FUNCTION COMPUTES B(K,I) OF EQUATION (3.66) IN CHAPTER 3
C GIVEN K AND I
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      IF(K.EQ.I.AND.I.EQ.1) GO TO 10
      BKI=FLOAT(M+2)-AMAX0(I-1,K-1)
      RETURN
   10 BKI=FLOAT(M+1)
      RETURN
      END
C
C
      FUNCTION AKI(K,I)
C
C THIS FUNCTION COMPUTES A(K,I) OF EQUATION (3.66) IN CHAPTER 3
C GIVEN K AND I
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
```

```
      COMMON RM1,PRM1,RM2,PRM2
      IF (K.EQ.1.AND.I.EQ.2) GO TO 100
      AKI=DFLOAT(L+2)-AMAX0(K-1,I-1)
      RETURN
  100 AKI=L
      RETURN
      END
C
C
      FUNCTION BETA(K)
C
C THIS FUNCTION COMPUTES BETA(K) OF EQUATION (3.83) IN CHAPTER 3.
C GIVEN K
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      IF (K.EQ.1) BETA=M+1
      IF (K.GE.2) BETA=M+3-K
      RETURN
      END
C
C
      FUNCTION ALPH(K)
C
C THIS FUNCTION COMPUTES ALPH(K) IN EQUATION (3.82) IN CHAPTER 3
C GIVEN K
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      IF (K.EQ.2) ALPH=L
      IF (K.GE.3) ALPH=L+3-K
      RETURN
      END
```

APPENDIX 4.2

```
C
C
C THIS PROGRAM COMPUTES THE DIGITAL FILTER COEFFICIENTS
C FROM BUTTERWORTH LOW PASS FILTER SPECIFICATION.
C
C INPUT DATA CARDS
C CARD 1
C COLUMN 1---10: HIGHEST FREQUENCY/PI OF PASS BAND: OP
C COLUMN 11---20: LOWEST FREQUENCY/PI OF STOP BAND: OS
C CARD 2
C COLUMN 1---10: LARGEST PASS BAND ATTENUATION IN DECIBEL: A
C COLUMN 11---20: SMALLEST STOP BAND ATTENUATION IN DECIBEL: B
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 RZ1(15),RZ2(15),RZ3(15),D(15),E(15),RC(15),FT(15)
      COMPLEX*16 P(15),Q(15),C(15),PL,Z,SP(15),C0,D0
      PI=3.14159265358979300
C
C READ HIGHEST FREQUENCY/PI OF PASS BAND: OP
C      LOWEST FREQUENCY/PI OF STOP BAND: OS
C
      READ(5,10) OP,OS
   10 FORMAT(2F10.0)
C
C READ PASS BAND ATTENUATION IN DECIBEL: A
C      STOP BAND ATTENUATION IN DECIBEL: B
C
      READ(5,10) A,B
      WRITE(6,947)
  947 FORMAT('1',////////10X,'DIGITAL FILTER'//)
      WRITE(6,826)
      WRITE(6,824) OP,OS
  824 FORMAT(' ',10X,'OP=',F10.5,5X,'OS=',F10.5)
      WRITE(6,826)
  826 FORMAT('0')
      WRITE(6,825) A,B
  825 FORMAT(' ',10X,'A=',F12.5,5X,'B=',F12.5)
      WRITE(6,826)
C
C SET CONSTANTS
C
      T0=0.0D0
      T1=1.0D0
      T10=10.0D0
      T2=2.0D0
C
C COMPUTE N, THE DEGREE OF BUTTERWORTH FILTER WITH PREWARPING FOR
C BILINEAR TRANSFORMATION
C
      OP=OP*PI
      OS=OS*PI
      OP=OP/T2
      OS=OS/T2
      A=A/T10
```

```
      B=B/T10
      FN=DLOG((T10**A-T1)/(T10**B-T1))
      FN=FN/DLOG(DTAN(CP)/DTAN(OS))
      FN=FN/T2
      N=FN+T1
C
C COMPUTE OC: OMEGAC, THE CUTOFF FREQUENCY OF BUTTERWORTH FILTER
C
      FN=N
      N2=N*2
      FN2=N2
      R1=T1/FN2
      OC=T2*DTAN(OS)/(T10**B-T1)**R1
C
C
C COMPUTE SP(J1): POLES OF THE FILTER OF CONTINUOUS TYPE, J1 POLES
C WILL BE OBTAINED
C
      J1=0
      DO 20 I=1,N2
      FI1=I-1
      Z=DCMPLX(T0,PI*(T2*FI1+T1)/FN2)
      PL=CDEXP(Z)*DCMPLX(T0,OC)
      IF (DREAL(PL).GE.T0) GO TO 20
      J1=J1+1
      SP(J1)=PL
   20 CONTINUE
      C0=DCMPLX(T0,T0)
      DO 60 I=1,15
      P(I)=C0
      Q(I)=C0
      C(I)=C0
      E(I)=0.0D0
   60 CONTINUE
C
C COMPUTE COEFFICIENTS OF DENOMINATOR OF TRANSFER FUNCTION OF
C CONTINUOUS TYPE
C LL WILL BE THE DEGREE
C
      NN=2
      MM=2
      P(1)=-SP(1)
      P(2)=DCMPLX(T1,T0)
      DO 40 K1=2,J1
      Q(1)=-SP(K1)
      Q(2)=DCMPLX(T1,T0)
      CALL ZMULT(P,NN,Q,MM,C,LL)
      DO 45 K2=1,LL
   45 P(K2)=C(K2)
      NN=LL
   40 CONTINUE
      DO 48 K2=1,LL
   48 RC(K2)=DREAL(C(K2))
C
```

```
C COMPUTE THE NUMERATOR OF THE TRANSFER FUNCTION OF CONTINUOUS TYPE: DR
C
      DO=DCMPLX(T1,T0)
      DO 53 I=1,J1
   53 DO=DO*SP(I)
      DR=DREAL(DO)
C
C DO THE Z TRANSFORM
C E(I)S WILL HAVE THE DENOMINATOR COEFFICIENTS
C FI(I)S WILL HAVE THE NUMERATOR COEFFICIENTS
C
      DO 234 K=1,LL
      CALL FACT(2.0D0,-2.0D0,K-1,RZ1,K3)
      CALL FACT(1.0D0,1.0D0,LL-K,RZ2,K4)
      CALL RMULT(RZ1,K3,RZ2,K4,D,K5)
      DO 133 I=1,K5
      D(I)=RC(K)*D(I)
  133 E(I)=E(I)+D(I)
  234 CONTINUE
      CALL FACT(1.0D0,1.0D0,LL-1,FT,K6)
      DO 182 I=1,K6
  182 FT(I)=DR*FT(I)
C
C OUTPUT THE COEFFICIENTS WITH CONSTANT TERM OF DENOMINATOR ONE
C
      DO 195 I=1,LL
      J=I-1
      CF=E(I)/E(1)
  195 WRITE(6,199) J,CF
  199 FORMAT(1X,9X,'A(',I3,')=',1PD25.15)
      DO 193 I=1,K6
      J=I-1
      CF=FT(I)/E(1)
  193 WRITE(6,194) J,CF
  194 FORMAT(1X,9X,'B(',I3,')=',1PD25.15)
      STOP
      END
C
C
C
      SUBROUTINE ZMULT(P,N,Q,M,C,L)
C
C THIS SUBROUTINE MULTIPLIES TWO POLYNOMIALS P AND Q OF COMPLEX
C COEFFICIENTS AND PLACES THE RESULTS IN C
C N: THE DEGREE OF P
C M: THE DEGREE OF Q
C L: THE DEGREE OF C
C THE COEFFICIENTS ARE PLACED FROM LOW TO HIGH
C
      COMPLEX*16 C(15),P(15),Q(15)
      COMPLEX*16 CO
      CO=DCMPLX(0.0D0,0.0D0)
      DO 10 I=1,15
   10 C(I)=CO
```

```
      DO 50 I=1,N
      DO 50 J=1,M
      K=I+J-1
   50 C(K)=C(K)+P(I)*Q(J)
      L=N+M-1
      RETURN
      END
C
C
C
      SUBROUTINE RMULT(P,N,Q,M,C,L)
C
C THIS SUBROUTINE MULTIPLIES POLYNOMIALS OF REAL COEFFICIENTS P AND Q
C AND PLACES THE RESULTS IN C
C N: THE DEGREE OF P
C M: THE DEGREE OF Q
C L: THE DEGREE OF C
C THE COEFFICIENTS ARE PLACED FROM LOW TO HIGH
C
      REAL*8 P(15),C(15),Q(15)
      DO 10 I=1,15
   10 C(I)=0.0D0
      DO 50 I=1,N
      DO 50 J=1,M
      K=I+J-1
   50 C(K)=C(K)+P(I)*Q(J)
      L=N+M-1
      RETURN
      END
C
C
C
      SUBROUTINE FACT(A,B,N,R,M)
C
C THIS SUBROUTINE COMPUTES BINOMIAL COEFFICIENTS OF (A+B*X)**N
C AND PLACES THE RERULTS IN R
C M: THE DEGREE OF R
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 R(15)
      NP1=N+1
      M=NP1
      DO 100 J=1,NP1
      I=J-1
  100 R(J)=BIN(N,I)*(B**I)*(A**(N-I))
      RETURN
      END
C
C
C
      FUNCTION BIN(N,I)
C
C THIS FUNCTION COMPUTES THE BINOMIAL COEFFICIENTS: C(N,I)
C
```

```
      REAL*8 BIN,FJ
      NI=N-I
      BIN=1.0D0
      IF(N.EQ.I.OR.I.EQ.0) RETURN
      DO 10 J=1,N
      FJ=J
   10 BIN=BIN*FJ
      DO 20 J=1,I
      FJ=J
   20 BIN=BIN/FJ
      DO 30 J=1,NI
      FJ=J
   30 BIN=BIN/FJ
      RETURN
      END
```

APPENDIX 4.3

```
C
C
C THIS PROGRAM COMPUTES THE THEORETICAL ERROR TO SIGNAL RATIOS OF
C THE FILTER:
C A(0)=1.0
C A(1)=-SQRT(2.0)*P
C A(2)=P*P
C B(0)=1.0
C WHERE ABSOLUTE VALUE OF INPUT P IS LESS THAN 1.0
C LOGARITHMIC BASE=2
C USAGE OF  THE RESIDUE THEOREM OF COMPLEX VARIABLES
C
C INPUT DATA CARDS
C CARD 1---N
C COLUMN 1---10: VALUE OF P
C
C BETA: NUMBER OF BITS OF THE FRACTIONAL PART
C
      IMPLICIT REAL*8(A-H,M-Z)
      N1=1.0D0
      N2=2.0D0
      N12=12.0D0
      S2=DSQRT(N2)
  100 READ(5,101,END=200) P
  101 FORMAT(F10.2)
      A1=-S2*P
      A2=P**2
      WRITE(6,105) P
  105 FORMAT(1H1///9X,'P=',D15.3)
      WRITE(6,109)
  109 FORMAT(1H0,5X,' BETA',10X,'ROUND',6X,'ROUND MAX',10X,'TRUNC',6X,
     1'TRUNC MAX'/)
C
C QQ,T1,T2,T3 ARE GIVEN IN EXAMPLE 3 OF SECTION 4.1
C
      CALL RES(P,QQ)
      T1=N2*S2*A1*P/((N1-P*P)*(P**4+N1))
      T2=N2*A2*P*P/(P**4+N1)
      CALL MAX1(P,QQ1)
      CALL MAX2(P,QQ2)
      T3=(P*P+N1)/((N1-P*P)*(P**4+N1))
      DO 160 I=2,30
      BETA=I
C
C   K=1 FOR ROUND, K=2 FOR TRUNC
C
      DO 150 K=1,2
C
C QS IS THE VARIANCE
C MES IS THE SQUARED MEAN
C
      A=N2**(-BETA-N1)
      IF (K.EQ.2) GO TO 2
      TEMP=N2**A-N2**(-A)
```

```
      QS=TEMP**2/N12
      TEMP=N2**A+N2**(-A)-N2
      MES=(TEMP/N2)**2
      GO TO 3
    2 TEMP=N1-N2**(-N2*A)
      QS=(TEMP**2)/N12
      MES=(TEMP/N2)**2
    3 RT=QS*(N1-T1-T2)-MES*QQ/(T3*P*P)
      RTM=QS*T3*QQ1+MES*QQ2
      IF (K.EQ.2) GO TO 150
      RR=RT
      RRM=RTM
  150 CONTINUE
C
C RR AND RT ARE THE ROOT SQUARES OF THE RATIOS FOR ROUNGING AND
C TRUNCATION RESPECTIVELY
C RRM AND RTM ARE THE MAXIMUM BOUNDS FOR ROUNDING AND TRUNCATION
C RESPECTIVELY
C
      RR=DSQRT(RR)
      RRM=DSQRT(RRM)
      RT=DSQRT(RT)
      RTM=DSQRT(RTM)
      WRITE(6,153) I,RR,RRM,RT,RTM
  153 FORMAT(1H ,5X,I5,4(1PD15.3))
  160 CONTINUE
      WRITE(6,201)
  201 FORMAT('0',9X,'THEORETICAL ERROR TO SIGNAL RATIOS')
      GO TO 100
  200 STOP
      END
C
C
C
      SUBROUTINE RES(P,QQR)
C
C THIS SUBROUTINE COMPUTES QQ GIVEN IN EXAMPLE 3 OF SECTION 4.1
C BY THE USE OF RESIDUE THEOREM
C QQR CORRESPONDS TO QQ OF EQUATION(4.18)
C Q1 AND Q2 ARE GIVEN IN THE EQUATIONS (4.19) AND (4.20)
C
      IMPLICIT COMPLEX*16(A-H,O-Z)
      COMPLEX*16 N1MJ,N1PJ
      REAL*8 QQR
      REAL*8 N1,N2,N3,S2,S3,P,D1,E1,F1,G1
      N1=1.0D0
      N2=2.0D0
      N3=3.0D0
      S2=DSQRT(N2)
      S3=DSQRT(N3)
      D1=(N1-S3)*P/S2
      E1=(N1+S3)*P/S2
      F1=(N1+S3)/(S2*P)
      G1=(N1-S3)/(S2*P)
```

```
      N1MJ=DCMPLX(N1,-N1)
      N1PJ=DCMPLX(N1,N1)
      B1=N1PJ/(P*S2)
      C1=N1MJ/(P*S2)
      DO 10 I=1,2
      IF (I.EQ.2) GO TO 2
    1 A1=N1MJ*P/S2
      Z=N1PJ*P/S2
      GO TO 3
    2 A1=N1PJ*P/S2
      Z=N1MJ*P/S2
    3 A=Z-A1
      B=Z-B1
      C=Z-C1
      D=Z+D1
      E=Z+E1
      F=Z-F1
      G=Z-G1
      ABC2=(A*B*C)**2
      ABC4=ABC2**2
      R1=D*E*F*G
      R2=Z*E*F*G
      R3=Z*D*F*G
      R4=Z*D*E*G
      R5=Z*D*E*F
      R6=Z*D*E*F*G*A*B*C*N2
      R7=B*C+A*C+A*B
      AA=ABC2*(R1+R2+R3+R4+R5)-R6*R7
      AA=AA/ABC4
      IF (I.EQ.2) GO TO 21
      Q1=AA
      GO TO 10
   21 Q2=AA
   10 CONTINUE
      QQ=Q1+Q2
      QQR=DREAL(QQ)
      RETURN
      END
C
C
C
      SUBROUTINE MAX1(P,QQ)
C
C THIS SUBROUTINE COMPUTES QQ1 GIVEN IN THE EQUATION (4.23) IN
C EXAMPLE 3 OF SECTION 4.1
C
      IMPLICIT REAL*8(A-H,N-Z)
      N1=1.0D0
      N2=2.0D0
      A1=-DSQRT(N2)*P
      A2=P*P
      QQ=N1+A1*A1+A2*A2-A1*A2*N2
      RETURN
      END
```

```
C
C
C
      SUBROUTINE MAX2(P,QQ)
C
C THIS SUBROUTINE COMPUTES QQ2 GIVEN IN THE EQUATION (4.23) IN
C EXAMPLE 3 OF SECTION 4.1
C
      IMPLICIT REAL*8(A-H,N-Z)
      COMPLEX*16 ITH,Z
      N1=1.0D0
      N2=2.0D0
      A1=-DSQRT(N2)*P
      A2=P*P
      W=3.1415927D0/4.0D0
      ITH=DCMPLX(0.0D0,W)
      Z=CDEXP(ITH)
      NUM=CDABS(N1-(A1*Z+A2*Z**2))**2
      DEN=CDABS(N1+A1*Z+A2*Z**2)**2
      QQ=NUM/DEN
      RETURN
      END
```

APPENDIX 4.4

```
MN: PROC OPTIONS (MAIN);
/*
PROGRAM TO COMPUTE THE ERROR TO SIGNAL RATIOS BY SIMULATION.  SAME
INPUT SEQUENCE GOES THROUTH TWO FILTERS: ONE USES LOGARITHMIC
ARITHMETIC AND THE OTHER USES LONG FLOATING POINT ARITHMETIC.
THE ERRORS ARE COMPUTED AS THE DIFFERENCES BETWEEN THE TWO
OUTPUTS.
*/
/*
INPUT DATA CARDS
CARD 1: NUMBER OF FILTERS TO SIMULATE
CARD 2: NUMBER OF RANDOM NUMBERS WHICH GO THROUGH FILTERS
        AND THE NUMBER AFTER WHICH STATISTICS WILL BE COLLECTED
CARD 3: DEGREES OF D(Z) AND N(Z)---L AND M
CARD 4---L+4: COEFFICIENTS A(I)
CARD L+5---L+M+5: COEFFICIENTS B(I)
CARD L+M+6: NUMBER OF COMBINATIONS OF ALPHA AND BETA TO SIMULATE: NAB
CARD L+M+7---L+M+NAB+6: ALPHA AND BETA
*/
/*
LA,LB: LOGARITHMIC FILTER COEFFICIENTS
LX,LY: LOGARITHMIC NUMBER INPUTS AND OUTPUT RESPECTIVELY
A,B: LONG FLOATING POINT FILTER COEFFICIENTS
X,W: LONG FLOATING POINT INPUTS AND OUTPUTS RESPECTIVELY
E: DIFFERENCE BETWEEN THE TWO OUTPUTS
ALP: NUMBER OF BITS BEFORE THE BINARY POINT OF EXPONENT MINUS ONE
BETA: NUMBER OF BITS AFTER THE BINARY POINT FOR LOGARITHMIC NUMBER
BS: BASE OF LOGARITHM
*/
   DCL 1 LA(0:10),
         2 SA BIN FIXED,
         2 EA BIN FIXED(31);
   DCL 1 LB(0:10),
         2 SB BIN FIXED,
         2 EB BIN FIXED(31);
   DCL 1 LY(0:10),
         2 SY BIN FIXED,
         2 EY BIN FIXED(31);
   DCL 1 LX(0:10),
         2 SX BIN FIXED,
         2 EX BIN FIXED(31);
   DCL 1 LX0,
         2 SX0 BIN FIXED,
         2 EX0 BIN FIXED(31);
   DCL 1 TLX(0:1),
         2 TSX BIN FIXED,
         2 TEX BIN FIXED(31);
   DCL 1 TLY(0:1),
         2 TSY BIN FIXED,
         2 TEY BIN FIXED(31);
   DCL (A,B,W,X) (0:10) BIN FLOAT(53);
   DCL (AV,BB) (0:10) BIN FLOAT(53);
   DCL (UT,US,FUN,FUN2,FUN10) CHAR(1);
   DCL (N0,N1,N2,N3,N6,HN1,S6,S3,S2) BIN FLOAT(53);
```

```
   DCL N10 BIN FLOAT(53);
   DCL (BIG,N50) BIN FLOAT(53);
   DCL UST BIN FLOAT(53);
   DCL Z BIN FIXED(31,0);
   DCL (AA,CC,MM) BIN FIXED(31,0);
   DCL (NEX,LNEX) DEC FIXED(5,0);
   DCL (I1,J1) BIN FIXED;
   DCL (K,N) BIN FIXED(31,0);
   DCL NSTR BIN FIXED(31,0);
   DCL (I,L,M,J) BIN FIXED(31,0);
   DCL (ALBT2,ALBT1) BIN FIXED(31);
   DCL (TFX,TFW) BIN FLOAT(53);
   DCL (Y0,TMW,E,TES,TWS,FN,REWR2,REWR10) BIN FLOAT(53);
   DCL TEMP BIN FLOAT(53);
   DCL (ALP,BETA,AL2,BT2,EMAX,XMIN,BS) BIN FLOAT(53);
   DCL (NABJ,NAB) BIN FLOAT(53);
   DCL (OV,OV2,OV10,UN,UN2,UN10) BIN FIXED(31);
/*
*/
   CVBL: PROC(X,L);
/*
THIS PROCEDURE CONVERTS FLOATING POINT NUMBER X TO LOGARITHMIC NUMBER L
*/
      DCL 1 L,
            2 SH BIN FIXED,
            2 H BIN FIXED(31);
      DCL (X,ABX) BIN FLOAT(53);
      SH=SIGN(X);
      ABX=ABS(X);
      IF ABX>XMIN
         THEN
            IF BS=N2
               THEN H=ROUND(FIXED(LOG2(ABX)*BT2,31,1),0);
               ELSE H=ROUND(FIXED(LOG10(ABX)*BT2 ,31,1),0);
         ELSE DO;
                 H=-ALBT2;
                 IF SH=0 THEN SH=1;
              END;
   END CVBL;
/*
*/
   UTP: PROC(UT,X);
/*
THIS PROCEDURE GENERATES RANDOM NUMBERS OF SPECTRUM=1 IN X
*/
      DCL UT CHAR(1);
      DCL (FU,X) BIN FLOAT(53);
      RZU: PROC(FU);
         DCL FU BIN FLOAT(53);
         DCL U BIN FIXED(31,31);
         Z=AA*Z+CC;
         Z=MOD(Z,MM);
         U=DIVIDE(Z,MM,31,31);
         FU=FLOAT(U,53);
```

```
      END RZU;
      CALL RZU(FU);
      IF UT='U'
         THEN X=S3*(N2*FU-N1);
         ELSE
            IF FU<HN1
               THEN X=S6*(SQRT(N2*FU)-N1);
               ELSE X=S6*(N1-SQRT(N2-N2*FU));
   END UTP;
/*
*/
   LMUL: PROC(X,Y,Z);
/*
THIS PROCEDURE DOES THE MULTIPLICATION BETWEEN X AND Y IN
LOGARITHMIC ARITHMETIC AND RETURNS THE RESULT IN Z
*/
      DCL 1 X,
            2 SX BIN FIXED,
            2 EX BIN FIXED(31);
      DCL 1 Y,
            2 SY BIN FIXED,
            2 EY BIN FIXED(31);
      DCL 1 Z,
            2 SZ BIN FIXED,
            2 EZ BIN FIXED(31);
      IF SX=SY
         THEN SZ=1;
         ELSE SZ=-1;
      EZ=EX+EY;
      IF EZ<-ALBT2
         THEN DO;
                 EZ=-ALBT2;
                 UN=UN+1;
              END;
         ELSE
            IF EZ>ALBT1
               THEN
                  DO;
                     EZ=ALBT1;
                     OV=OV+1;
                  END;
   END LMUL;
/*
*/
   LADD: PROC(X,Y,Z);
/*
THIS PROCEDURE DOES THE ADDITION BETWEEN X AND Y IN LOGARITHMIC
ARITHMETIC AND RETURNS THE RESULT IN Z
*/
      DCL 1 X,
            2 SX BIN FIXED,
            2 EX BIN FIXED(31);
      DCL 1 Y,
            2 SY BIN FIXED,
```

```
        2 EY BIN FIXED(31);
    DCL 1 Z,
        2 SZ BIN FIXED,
        2 EZ BIN FIXED(31);
    DCL (PX,EZ1,FEZ) BIN FIXED(31);
    DCL (TEMP,AT,FYMX) BIN FLOAT(53);
    PX=EX;
    FYMX=-ABS(FLOAT(EY-EX,53)/BT2);
    IF SX=SY
       THEN
          DO;
             SZ=SX;
             IF EX<EY
                THEN PX=EY;
             IF BS=N2
                THEN AT=LOG2(N1+BS**FYMX);
                ELSE AT=LOG10(N1+BS**FYMX);
          END;
       ELSE
          DO;
             IF EX>=EY
                THEN SZ=SX;
                ELSE
                   DO;
                      SZ=SY;
                      PX=EY;
                   END;
             IF FYMX¬=N0
                THEN DO;
                        IF BS=N2
                           THEN   AT=LOG2(N1-BS**FYMX);
                           ELSE   AT=LOG10(N1-BS**FYMX);
                     END;
                ELSE AT=-EMAX;
          END;
    EZ=PX+ROUND(FIXED(AT*BT2,31,1),0);
    IF EZ<-ALBT2
       THEN DO;
               EZ=-ALBT2;
               IF K>NSTR
                  THEN UN=UN+1;
            END;
    IF EZ>ALBT1
       THEN
          DO;
             EZ=ALBT1;
             IF K>NSTR
                THEN OV=OV+1;
          END;
 END LADD;
/*
*/
 LCOFF: PROC;
/*
```

```
THIS PROCEDURE DOES THE INITIAL SETTINGS FOR THE TWO FILTERS
*/
      DO I=0 BY 1 WHILE(I<=M);
         X(I)=HN1;
         CALL CVBL(X(I),LX(I));
         X(I)=FLOAT(SX(I),53)*BS**(FLOAT(EX(I),53)/BT2);
      END;
      DO I=0 BY 1 WHILE(I<=L);
         W(I)=HN1;
         CALL CVBL(W(I),LY(I));
         W(I)=FLOAT(SY(I),53)*BS**(FLOAT(EY(I),53)/BT2);
      END;
      DO I=1 BY 1 WHILE(I<=L);
         CALL CVBL(A(I),LA(I));
         A(I)=FLOAT(SA(I),53)*BS**(FLOAT(EA(I),53)/BT2);
      END;
      DO I=0 BY 1 WHILE(I<=M);
         CALL CVBL(B(I),LB(I));
         B(I)=FLOAT(SB(I),53)*BS**(FLOAT(EB(I),53)/BT2);
      END;
   END LCOFF;
/*
*/
   LSIM: PROC;
/*
THIS PROCEDURE DOES THE SIMULATION AND COMPUTES THE
ERROR TO SIGNAL RATIOS
*/
      TES=N0;
      TWS=N0;
      US=' ';
      GV=0;
      UN=0;
      FUN=' ';
      Z=10825;
/*
APPLY N INPUTS TO THE FILTERS
*/
      DO K=1 BY 1 TO N;
         DO I=M BY -1 WHILE(I>=1);
            X(I)=X(I-1);
         END;
/*
GENERATE A RANDOM NUMBER IN X(0)
*/
         CALL UTP(UT,X(0));
/*
MAKE CONVERSION
*/
         CALL CVBL(X(0),LX0);
         X(0)=FLOAT(SX0,53)*BS**(FLOAT(EX0,53)/BT2);
/*
DO THE FILTERING OF FLOATING POINT NUMBERS
*/
```

```
          ON UNDERFLOW
             BEGIN;
                IF K>NSTR
                   THEN FUN='#';
             END;
          DO I=L BY -1 WHILE(I>=1);
             W(I)=W(I-1);
          END;
          TFX=N0;
          DO I=0 BY 1 WHILE(I<=M);
             TFX=TFX+B(I)*X(I);
          END;
          TFW=N0;
          DO I=1 BY 1 WHILE(I<=L);
             TFW=TFW+A(I)*W(I);
          END;
          W(0)=TFX-TFW;
          ON UNDERFLOW SYSTEM;
/*
DO THE FILTERING OF LOGARITHMIC NUMBERS
*/
          DO I=M BY -1 WHILE(I>=1);
             LX(I)=LX(I-1);
          END;
          LX(0)=LX0;
          DO I=L BY -1 WHILE(I>=1);
             LY(I)=LY(I-1);
          END;
          CALL LMUL(LB(0),LX(0),TLX(0));
          DO I=1 BY 1 WHILE(I<=M);
             CALL LMUL(LB(I),LX(I),TLX(1));
             CALL LADD(TLX(0),TLX(1),TLX(0));
          END;
          CALL LMUL(LA(1),LY(1),TLY(0));
          DO I=2 BY 1 WHILE(I<=L);
             CALL LMUL(LA(I),LY(I),TLY(1));
             CALL LADD(TLY(0),TLY(1),TLY(0));
          END;
          TSY(0)=-TSY(0);
          CALL LADD(TLX(0),TLY(0),LY(0));
/*
COLLECT STATISTICS
*/
          Y0=FLOAT(SY(0),53)*BS**(FLOAT(EY(0),53)/BT2);
          E=Y0-W(0);
          IF ABS(E)>10.0E0
             THEN
                DO;
                   K=N;
                   US='*';
                END;
             ELSE
                DO;
                   IF K>NSTR
```

```
                    THEN
                        DO;
                            TES=TES+E*E;
                            TWS=TWS+W(0)*W(0);
                        END;
                END;
        END;                        /* END OF LOOP */
        IF US¬='*'
            THEN
                DO;
                    IF BS=N2
                        THEN REWR2=SQRT(TES/TWS);
                        ELSE REWR10=SQRT(TES/TWS);
                END;
            ELSE
                DO;
                    IF BS=N2
                        THEN REWR2=UST;
                        ELSE REWR10=UST;
                END;
    END LSIM;
/*
*/
/*
MAIN PROCEDURE OPERATION STARTS HERE
*/
    N0=0.0E0B;
    N1=1.0E0B;
    N2=10.0E0B;
    N3=11.0E0B;
    N6=110.0E0B;
    HN1=0.1E0B;
    N10=1010.0E0B;
    N50=50.0E0;
    N200=10;
    BIG=1.0E20;
    UST=1.0E3;
    AA=129;
    CC=8085;
    MM=16384;
    UT='U';
    S6=SQRT(N6);
    S3=SQRT(N3);
    S2=SQRT(N2);
/*
GET THE NUMBER OF FILTERS TO SIMULATE
*/
    GET LIST(NEX);
    DO LNEX=1 BY 1 TO NEX;
        PUT PAGE;
        PUT SKIP(10);
        PUT EDIT('ERROR TO SIGNAL RATIO') (X(10),A);
        PUT SKIP EDIT('EXPERIMENTAL') (X(10),A);
/*
```

```
GET THE NUMBER OF INPUTS FOR EACH SIMULATION
*/
      GET SKIP LIST(N,NSTR);
      PUT SKIP EDIT('N=',N,'STATISTICS COLLECTED AFTER ',NSTR)
                  (X(13),A,F(6),X(3),A,F(6));
      DO;
/*
GET  THE DEGREES OF D(Z), L AND N(Z), M
*/
         GET SKIP LIST(L,M);
         PUT SKIP EDIT('L=',L,'M=',M)
               (X(13),A,F(4),X(5),A,F(4));
/*
GET COEFFICIENTS A(I) AND B(I)
*/
         DO I=0 BY 1 WHILE(I<=L);
            GET SKIP LIST(AV(I));
         END;
         DO I=0 BY 1 WHILE(I<=M);
            GET SKIP LIST(BB(I));
         END;
         DO I=0 BY 1 TO L;
            PUT SKIP EDIT('A(',I,')=',AV(I))
                     (X(10),A,F(3),A,E(25,15));
         END;
         DO I=0 BY 1 TO M;
            PUT SKIP EDIT('B(',I,')=',BB(I))
                        (X(10),A,F(3),A,E(25,15));
         END;
         PUT SKIP(2) EDIT('NO.','ALPHA','BETA','BASE=',N2,'BASE=',
            N10)
         (X(5),A,X(1),A,X(1),A,2 (X( 7),A,F(3),X(13)));
         PUT SKIP EDIT('RATIO','OV','UN','RATIO','OV','UN')
                  (X(27),A,X(4),A,X(3),A,X( 8),A,X(4),A,X(3),A);
         GET SKIP LIST(NAB);
/*
DO FOR NAB CONBINATIONS OF ALPHA AND BETA
*/
         DO NABJ=N1 BY N1 TO NAB;
/*
GET THE ASSIGNED NUMBERS ALPHA AND BETA
*/
            GET SKIP LIST(ALP,BETA);
            AL2=N2**ALP;
            BT2=N2**BETA;
            EMAX=N2**(ALP+N1)-N2**(-BETA);
            ALBT2=FIXED(AL2*BT2,31,0);
            ALBT1=ALBT2-1;
/*
DO FOR BASE=2 AND BASE=10
*/
            DO BS=N2,N10;
               XMIN=BS**(-AL2);
               DO I=0 BY 1 WHILE(I<=M);
```

```
                B(I)=BB(I);
            END;
            DO I=0 BY 1 WHILE(I<=L);
                A(I)=AV(I);
            END;
            CALL LCOFF;
            CALL LSIM;
            IF BS=N2
                THEN DO;
                        OV2=OV;
                        UN2=UN;
                        FUN2=FUN;
                    END;
                ELSE DO;
                        OV10=OV;
                        UN10=UN;
                        FUN10=FUN;
                    END;
        END;.
        PUT SKIP EDIT(NABJ,ALP,BETA,REWR2,FUN2,OV2,UN2,
                                REWR10,FUN10,OV10,UN10)
        (F(7),F( 7),F(5),2 (E(13,5),A(1),2 F(5)));
    END;
    PUT SKIP(2) EDIT('* THE RATIO OF ',UST,
  ', IF ANY, INDICATES THAT THE',
        'LOGARITHMIC FILTER PRODUCED TOO MUCH ERROR.')
                    (X(5),A,E(14,5),A,SKIP,X(7),A);
    PUT SKIP(2) EDIT('* THOSE UNDER OV OR UN INDICATE THE',
            ' NUMBER OF TIMES THAT',
          'OVERFLOW OR UNDERFLOW OCCURRED IN THE ',
                      'LOGARITHMIC FILTER.')
                       (X(5),A,A,SKIP,X(7),A,A);
    PUT SKIP(2) EDIT('* #, IF ANY, INDICATES THAT UNDERFLOW ',
    'OCCURRED IN THE',
                    'LONG FLOATING POINT FILTER, ',
          'WITH THE RESULT OF ZERO OPERATION CONTINUED.')
                (X(5),A,A,SKIP,X(7),A,SKIP,X(7),A);
  END;
 END;
END MN;
```

APPENDIX 4.5

```
C
C PROGRAM FOR COMPUTING THE THEORETICAL VALUES AND MAXIMUM BOUNDS
C OF ERROR TO SIGNAL RATIO OF DIGITAL FILTERS USING THE SIMPSON'S
C RULE OF NUMERICAL INTEGRATION
C
C INPUT DATA CARDS
C CARD 1
C COLUMN 1--5: DEGREE OF D(Z), L
C COLUMN 6--10: DEGREE OF N(Z), M
C COLUMN 11--15  1---MAX IS COMPUTED FOR 0<=W<=PI
C                2---MAX IS COMPUTED FOR 0<=W<=PI/2
C CARDS 2--L+2 COLUMN 1--25: COEFFICIENTS, A(0)--A(L)
C CARDS L+3--L+M+3 COLUMN 1--25: COEFFICIENTS, B(0)--B(M)
C CARD L+M+4  COLUMN 1--5: NUMBER OF COMBINATIONS OF ALPHA AND BETA
C CARDS L+M+5---COLUMN 1--20: ALPHA
C               COLUMN 21--40: BETA
C
C
C  L,M: DEGREES OF D(Z) AND N(Z) RESPECTIVELY
C  A(I),B(I):  COEFFICIENTS OF THE FILTER
C  PRM1: MAXIMUM VALUE OF FUNCTION R3
C  PRM2: MAXIMUM VALUE OF FUNCTION R4
C  S(1),S(2),S(3) AND S(4) WILL HAVE THE INTEGRATION VALUES OF
C  FUNCTIONS R1,R2,R3 AND R4
C  TH(1): ERROR TO SIGNAL RATIO FOR BASE=2
C  THM(1): MAXIMUM VALUE OF TH(1)
C  TH(2): ERROR TO SIGNAL RATIO FOR BASE=10
C  THM(2): MAXIMUM VALUE OF TH(2)
C  BETA: NUMBER OF BITS AFTER BINARY POINT
C  BS: BASE OF LOGARITHM
C  QS: VARIANCE OF ERROR
C  MES: SQUARED VALUE OF ERROR MEAN
C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      REAL*8 S(10),DS(10),PS(10),TH(2),THM(2)
      REAL*8 BS(2),QS(2),MES(2),QQR(30),RRR(30)
      REAL*8 AL(10),BL(10)
      EXTERNAL R1,R2,R3,R4
      N2=2
      N10=10
      PRM1=0.0D0
      PRM2=0.0D0
      EPS=0.05D0
      PI=3.14159265358979D0
      PI1=24.0D0*PI/100.0D0
      PI2=26.0D0*PI/100.0D0
      JN1=300
      JN2=900
      N=2000
C
```

```
C READ COEFFICIENTS
C
      READ(5,50) L,M,ITYPE
   50 FORMAT(3I5)
      LP1=L+1
      MP1=M+1
      DO 100 I=1,LP1
      READ(5,101) A(I)
      AL(I)=A(I)
  100 WRITE(6,102) A(I)
  101 FORMAT(D25.16)
  102 FORMAT(1X,D25.16)
      DO 110 I=1,MP1
      READ(5,101) B(I)
      BL(I)=B(I)
  110 WRITE(6,102) B(I)
C
C COMPUTES THE RESULTS FOR EACH BETA AND OUTPUTS
C
      WRITE(6,888)
  888 FORMAT('1'/////' ')
      WRITE(6,601)
  601 FORMAT('0',5X,'ERROR TO SIGNAL RATIO'/6X,'INPUT COEFFICIENTS'/)
      WRITE(6,602) L,M
  602 FORMAT('0',5X,'L=',I3,5X,'M=',I3)
      DO 610 I=1,LP1
      I1=I-1
  610 WRITE(6,603) I1,A(I)
  603 FORMAT(1X,5X,'A(',I2,')=',1PD25.15)
      DO 620 I=1,MP1
      I1=I-1
  620 WRITE(6,604) I1,B(I)
  604 FORMAT(1X,5X,'B(',I2,')=',1PD25.15)
      WRITE(6,619)
  619 FORMAT('0',5X,'RESULTS')
      WRITE(6,605)  N2
  605 FORMAT('0',17X,'BASE=',I3)
      WRITE(6,606)
  606 FORMAT('0',2X,'ALPHA',1X,'BETA',2X,'THEORETICAL',2X,
     1'THEORETICAL MAX')
      BS(1)=2.0D0
      BS(2)=10.0D0
      READ(5,974) NAB
  974 FORMAT(I5)
      DO 160 IG=1,NAB
      READ(5,975) ALP,BETA
  975 FORMAT(2D20.5)
      I1=ALP
      I2=BETA
      AA=2.0D0**(-BETA-1.0D0)
      DO 155 K=1,1
      DO 963 J=1,LP1
  963 CALL CVBLF(AL(J),A(J),BS(K),ALP,BETA)
      DO 964 J=1,MP1
```

```
  964 CALL CVBLF(BL(J),B(J),BS(K),ALP,BETA)
C
C DO THE INTEGRATIONS
C
      DO 530 I=2,4
      PS(I)=0.0D0
      DO 510 J=1,1
      IF (I.NE.1) GO TO 511
      S(I)=SIMPS(0.0D0,PI,N,R1)
      GO TO 520
  511 IF (I.NE.2) GO TO 512
      SA1=SIMPS(0.0D0,PI1,JN1,R2)
      SA2=SIMPS(PI2,PI,JN2,R2)
      SA3=SIMPS(PI1,PI2,N,R2)
      S(I)=SA1+SA2+SA3
      GO TO 520
  512 IF (I.NE.3) GO TO 513
      SA1=SIMPS(0.0D0,PI1,JN1,R3)
      SA2=SIMPS(PI2,PI,JN2,R3)
      SA3=SIMPS(PI1,PI2,N,R3)
      S(I)=SA1+SA2+SA3
      GO TO 520
  513 CONTINUE
      SA1=SIMPS(0.0D0,PI1,JN1,R4)
      SA2=SIMPS(PI2,PI,JN2,R4)
      SA3=SIMPS(PI1,PI2,N,R4)
      S(I)=SA1+SA2+SA3
  520 S(I)=S(I)/PI
      DS(I)=S(I)-PS(I)
C
C CHECK FOR CONVERGENCE
C
      PS(I)=S(I)
  510 CONTINUE
  530 CONTINUE
      S(1)=S(2)
      TEMP=BS(K)**AA-BS(K)**(-AA)
      QS(K)=TEMP**2/12.0D0
      TEMP=BS(K)**AA+BS(K)**(-AA)-2.0D0
      MES(K)=(TEMP/2.0D0)**2
      TH(K)=QS(K)*S(1)*S(3)/S(2)+MES(K)*S(4)/S(2)
      THM(K)=QS(K)*S(1)*PRM1+MES(K)*PRM2
      TH(K)=DSQRT(TH(K))
  155 THM(K)=DSQRT(THM(K))
      WRITE(6,701) I1,I2,TH(1),THM(1)
  701 FORMAT(1X,I7,I5,2(1PD13.5,1PD17.5))
  160 CONTINUE
      IF (ITYPE.NE.2) GO TO 800.
      WRITE(6,709)
  709 FORMAT('0',5X,'* THEORETICAL MAX IS TAKEN FOR 0<=W<=PI/2')
  800 STOP
      END
C
C
```

```
      FUNCTION SIMPS(P,Q,N,F)
C
C THIS SUBROUTINE USES N APPLICATIONS OF SIMPSON'S RULE TO COMPUTE THE
C INTEGRAL OF F(X)*DX BETWEEN P AND Q
C
      IMPLICIT REAL*8(A-H,O-Z)
      FN=N
      TWOH=(Q-P)/FN
      H=TWOH/2.0D0
      SUMEND=0.0D0
      SUMMID=0.0D0
      DO 1 K=1,N
      FK1=K-1
      X=P+FK1*TWOH
      SUMEND=SUMEND+F(X)
    1 SUMMID=SUMMID+F(X+H)
      SIMPS=(2.0D0*SUMEND+4.0D0*SUMMID-F(P)+F(Q))*H/3.0D0
      RETURN
      END
C
C
      FUNCTION R1(W)
C
C THIS FUNCTION COMPUTES THE FUNCTIONAL VALUE INSIDE THE
C INTEGRATION OF S1 GIVEN IN THE EQUATION (4.1) IN SECTION 4.1
C FOR EACH VALUE OF W
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMPLEX*16 Z,DZ
      LP1=L+1
      Z=CDEXP(DCMPLX(0.0D0,W))
      DZ=DCMPLX(0.0D0,0.0D0)
      DO 100 K=1,LP1
  100 DZ=DZ+A(K)*Z**(-(K-1))
      R1=CDABS(DZ)**2
      R1=1.0D0/R1
      RETURN
      END
C
C
      FUNCTION R2(W)
C
C THIS FUNCTION COMPUTES THE FUNCTIONAL VALUE INSIDE THE INTEGRATION
C OF S2 GIVEN IN THE EQUATION (4.1) IN SECTION 4.1 FOR EACH VALUE OF W
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMPLEX*16 DZ,NZ,Z
      LP1=L+1
      MP1=M+1
      Z=CDEXP(DCMPLX(0.0D0,W))
```

```
      DZ=DCMPLX(0.0D0,0.0D0)
      NZ=DZ
      DO 100 K=1,LP1
  100 DZ=DZ+A(K)*Z**(-(K-1))
      DO 102 K=1,MP1
  102 NZ=NZ+B(K)*Z**(-(K-1))
      R2=CDABS(NZ/DZ)**2
      RETURN
      END
C
C
      FUNCTION R3(W)
C
C THIS FUNCTION COMPUTES THE FUNCTIONAL VALUE INSIDE THE INTEGRATION
C OF S3 GIVEN IN THE EQUATION (4.1) IN SECTION 4.1 FOR EACH VALUE OF W
C AND FINDS THE MAXIMUM VALUE M1 GIVEN IN THE INEQUALITY (4.2)
C AND STORES IT IN PRM1
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      COMPLEX*16 Z,DZ,NZ,CZ,BZ2,AZ2
      PI=3.14159265358979D0
      LP1=L+1
      MP1=M+1
      Z=CDEXP(DCMPLX(0.0D0,W))
      DZ=DCMPLX(0.0D0,0.0D0)
      NZ=DZ
      CZ=DZ
      BZ2=DZ
      AZ2=DZ
      DO 100 K=1,LP1
  100 DZ=DZ+A(K)*Z**(-(K-1))
      DO 102 K=1,MP1
  102 NZ=NZ+B(K)*Z**(-(K-1))
      DO 104 K=1,MP1
      DO 104 I=2,LP1
  104 CZ=CZ+B(K)*A(I)*Z**(K-I)
      DO 110 K=1,MP1
      DO 110 I=1,MP1
  110 BZ2=BZ2+B(K)*B(I)*BKI(K,I)*Z**(K-I)
      DO 112 K=2,LP1
      DO 112 I=2,LP1
  112 AZ2=AZ2+A(K)*A(I)*AKI(K,I)*Z**(K-I)
      R31=CDABS(BZ2)
      R321=CDABS(NZ/DZ)**2
      R32=CDABS(AZ2*R321)
      R33=DREAL(CZ*NZ/DZ)
      R3=R31+R32-2.0D0*R33
      RM1=R3/R321
      IF (ITYPE.NE.2) GO TO 115
      IF (W.GT.PI/2.0D0) RETURN
  115 IF (RM1.GT.PRM1) PRM1=RM1
```

```
      RETURN
      END
C
C
      FUNCTION R4(W)
C
C THIS FUNCTION COMPUTES THE FUNCTIONAL VALUE INSIDE THE INTEGRATION
C OF S4 GIVEN IN THE EQUATION (4.1) IN SECTION 4.1 FOR EACH VALUE OF W
C AND FINDS THE MAXIMUM VALUE M2 GIVEN IN THE INEQUALITY (4.2)
C AND STORE IT IN PRM2
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      COMPLEX*16 Z,DZ,NZ,BPZ,APZ
      PI=3.14159265358979300
      LP1=L+1
      MP1=M+1
      Z=CDEXP(DCMPLX(0.0D0,W))
      DZ=DCMPLX(0.0D0,0.0D0)
      NZ=DZ
      BPZ=DZ
      APZ=DZ
      DO 100 K=1,LP1
  100 DZ=DZ+A(K)*Z**(-(K-1))
      DO 102 K=1,MP1
  102 NZ=NZ+B(K)*Z**(-(K-1))
      DO 106 K=1,MP1
  106 BPZ=BPZ+B(K)*BETA(K)*Z**(-(K-1))
      DO 108 K=2,LP1
  108 APZ=APZ+A(K)*ALPH(K)*Z**(-(K-1))
      R40=CDABS(NZ)**2
      R41=CDABS(DZ)**2
      R42=CDABS(BPZ-NZ*APZ/DZ)**2
      R4=R42/R41
      RM2=R42/R40
      IF (ITYPE.NE.2) GO TO 114
      IF (W.GT.PI/2.0D0) RETURN
  114 IF (RM2.GT.PRM2) PRM2=RM2
      RETURN
      END
C
C
      FUNCTION BKI(K,I)
C
C THIS FUNCTION COMPUTES B(K,I) OF EQUATION (3.66) IN CHAPTER 3
C GIVEN K AND I
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      IF(K.EQ.I.AND.I.EQ.1) GO TO 10
```

```
      BKI=FLOAT(M+2)-AMAX0(I-1,K-1)
      RETURN
   10 BKI=FLOAT(M+1)
      RETURN
      END
C
C
      FUNCTION AKI(K,I)
C
C THIS FUNCTION COMPUTES A(K,I) OF EQUATION (3.66) IN CHAPTER 3
C GIVEN K AND I
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      IF (K.EQ.I.AND.I.EQ.2) GO TO 100
      AKI=DFLOAT(L+2)-AMAX0(K-1,I-1)
      RETURN
  100 AKI=L
      RETURN
      END
C
C
      FUNCTION BETA(K)
C
C THIS FUNCTION COMPUTES BETA(K) OF EQUATION (3.83) IN CHAPTER 3
C GIVEN K
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      IF (K.EQ.1) BETA=M+1
      IF (K.GE.2) BETA=M+3-K
      RETURN
      END
C
C
      FUNCTION ALPH(K)
C
C THIS FUNCTION COMPUTES ALPH(K) IN EQUATION (3.82) IN CHAPTER 3
C GIVEN K
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      IF (K.EQ.2) ALPH=L
      IF (K.GE.3) ALPH=L+3-K
      RETURN
      END
C
C
```

```
      SUBROUTINE CVBLF(X,Y,BS,ALP,BETA)
C
C THIS SUBROUTINE CONVERTS X TO THE FLOATING POINT NUMBER
C TO BE TESTED WITH SPECIFICATION OF ALP AND BETA, THEN
C CONVERTS IT BACK TO A LONG FLOATING POINT NUMBER IN Y
C
C BS: THE BASE OF THE FLOATING POINT NUMBER TO BE TESTED
C        IN THIS PROGRAM IT IS 2
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON L,M,ITYPE,IJU12
      COMMON A(10),B(10)
      COMMON RM1,PRM1,RM2,PRM2
      INTEGER*4 IALBT2,IE,IA
      T2=2.0D0
      T0=0.0D0
      AL2=T2**ALP
      BT2=T2**BETA
      IALBT2=AL2*BT2
      XMIN=BS**(-AL2)
      ABX=DABS(X)
      IF (ABX.GT.XMIN) GO TO 10
      IE=-IALBT2
      GO TO 20
   10 IF (BS.EQ.T2) GO TO 30
      IE=IDINT(DLOG10(ABX)*BT2*T2)
   40 IA=IABS(IE)
      IF (MOD(IA,2).NE.0) IA=IA+1
      IA=IA/2
      IF (IE.LT.0) GO TO 50
      IE=IA
      GO TO 20
   50 IE=-IA
      GO TO 20
   30 IE=IDINT(DLOG10(ABX)*BT2*T2/DLOG10(T2))
      GO TO 40
   20 DIE=IE
      Y=BS**(DIE/BT2)
      IF (X.LT.T0) Y=-Y
      RETURN
      END
```

CHAPTER V

EXPERIMENTAL COMPARISONS BETWEEN LOGARITHMIC FILTERS AND FLOATING POINT FILTERS

5.1 Direct Form Digital Filters

5.1.1 Method and Results

Four filters, each employing the logarithmic number system and the floating point number system, are tested. The Summary is given in Table 5.1 and the output graphs are shown in figures 5.2 to 5.9. The number systems used are as follows:

Logarithmic number system: $\alpha = 6$, $\beta = 8$ and base = 2 in the logarithmic number system definition of (2.6) of Chap. II

Note: the above specification of $\alpha = 6$, $\beta = 8$ and base = 2 is, by the equivalence relation of (6.1), between FOCUS .16 and FOCUS .10 [12].

Floating point number system: $f = 6$, $h = 8$ and base = 2 in the floating point number system definition of (2.5) of Chapter II

The reasons for choosing those specific number systems are as follows:

1. Both the logarithmic number system and the floating point number system use 16 bits which are convenient for micro-computers.
2. The range of the number system largely affects the accuracy. Then the ranges of both number systems should be equal. The above specification of $\alpha = 6$, $\beta = 8$, $f = 6$ and $h = 8$ gives almost equal ranges for both number systems. The range of the logarithmic number system is slightly larger than that of the floating point number system.

Note: the definition of α, β, f, and h are given in section 2.2.

The general procedure of the experiment is suggested in the Fig 5.1 and explained below:

The filter coefficients are originally given in the long floating point numbers and then converted to each number system. The conversion to the logarithmic number is explained in section 4.2 of Chapter IV, and the conversion to the floating point number is made by the procedure CVBF given in Appendix 5.1. The arithmetic in logarithmic number system is also explained in section 4.2, and the arithmetic in the floating point number system is made by the procedure FADML given in Appendix 5.1. The conversion and arithmetic use rounding instead of truncation. Like the logarithmic filter which is programmed in the PL/I program in Appendix 4.4, the floating point filter can be easily programmed. Examples are given below for a conversion to a floating number (of f = 4, h = 4 and base = 2), and a multiplication and an addition in the floating number system. The Procedures of CVBF and FADML of Appendix 5.1 can operate on many other combinations of f and h. See Chapter IV for the conversion and the arithmetic operations of logarithmic case.

a) conversion:

5.0 is to be converted to the form:

$$m \times 2^{e}$$

where m is a four bit fraction and e is an integer in a 5 bit number (the entire number of bits of the word is 10, see the floating point number definition in section 2.2)

$$\log_2 5.0 \approx 2.3219281$$

$$\approx 3-0.6780719$$

Then $5.0 \approx 2^{-0.6780719} \times 2^{3}$

$$\approx 0.625 \times 2^{3}$$

0.625 is 0.10100 in the binary form

3 is 00011 in the binary form.

Then m = 0.1010 (rounded to the 4th place)

e = 00011

b) multiplication:

Two numbers a and b given in the binary form is to be multiplied.

a and b are given by

$$a = 0.1101 \times 2^{00011}$$

$$b = 0.1001 \times 2^{00001}$$

Then $a \times b = 0.01110101 \times 2^{00100}$

$= 0.1110101 \times 2^{00011}$ (normalization)

$= 0.1111 \times 2^{00011}$ (rounding)

Then the fraction is 0.1111 and the exponent is 00011

c) addition:

Two numbers a and b given in the binary form is to be added.

a and b are

$$a = 0.1000 \times 2^{00010}$$

$$b = 0.1010 \times 2^{00000}$$

$a + b = (0.1000 + 0.001010) \times 2^{00010}$

$= 0.101010 \times 2^{00010}$

$= 0.1011 \times 2^{00010}$ (rounding)

Then the fraction is 0.1011 and the exponent is 00010.

According to the method explained above, the experimental error analysis for those of deterministic inputs involves all of the input quantization, the coefficients quantization, and the accumulated computation roundoff errors. The results are all in favor of the logarithmic filters, which give the smaller error to signal ratios. And the pictorial presentation of the outputs

of Fig. 5.2 to Fig. 5.9 also shows that the logarithmic filters produce the outputs which are closer to those of the long floating point filters than the floating point filters.

As suggested in section 4.4 of CHAPTER IV, given filter coefficients and the filter inputs in long floating point numbers it is impossible to have the same filter coefficients and the same filter inputs for both the logarithmic filter and the floating point filter to be tested. It is because the two number systems do not have the same representable numbers. It implies that a pure comparison on accumulated roundoff errors between the logarithmic filter and the floating point filter to be tested is difficult. So the overall error comparison is made between the two number system filters in this CHAPTER.

Note: The second order filter of Table 5.1 is

$$w_n = x_n - (a_1 w_{n-1} + a_2 w_{n-2})$$

where $a_1 = -\sqrt{2}\rho$, $a_2 = \rho^2$ and $\rho = 0.999$.

The fourth order filter of Table 5.1 is from the book [14]. It is a Butterworth digital filter with OMEGA = 0.5750 (OMEGA is a notation of [14]). It is re-arranged to be of direct form to become the filter of Table 5.2. The sixth order filter of Table 5.1 is from example 2 of section 4.1. The eighth order filter of Table 5.1 is also from the book [14]. It originally comes from the cascade of the fourth order Chebyschev lowpass digital filter with OMEGA = 1.6501 [14] and the fourth order Chebyschev highpass digital filter which is made highpass (by the method of [14]) from the Chebyschev lowpass filter with OMEGA = 1.8297 [14]. Then it has the poles as follows:

real part	imaginary part
0.2350490533164940 00	0.8237703442573550 00
0.2350490533164940 00	-0.8237703442573550 00
-0.8226449996012430-01	0.8480718731880190 00
-0.8226449996012430-01	-0.8480718731880190 00
0.2772630694390190 00	0.4036761522293090 00
0.2772630694390190 00	-0.4036761522293090 00
-0.1752496473943880 00	0.4207258224487310 00
-0.1752496473943880 00	-0.4207258224487310 00

The first four poles are changed to be

real part	imaginary part
0.2D0	0.95D0
0.2D0	-0.95D0
-0.1D0	0.955D0
-0.1D0	-0.955D0

Then the coefficients are computed again to become those of Table 5.3.

5.1.2 Comments

The plots of Fig. 5.2, 5.3, and 5.4 are the outputs of the high Q filter (Q = 393). Fig. 5.1 shows that the filter has a long starting transient state. Fig. 5.3 and 5.4 show that the floating point filters magnitude response is very much reduced compared to that of the logarithmic filter. This may suggest that the floating point filter is very much different from the original long floating point filter and that the logarithmic filter is between the two. This filter is further tested and examined in section 5.3.

Fig. 5.5 is the output of the 4th order lowpass filter and Fig. 5.6 is for the 6th order lowpass filter. Although some irregularities are shown in the logarithmic filter outputs, a great deal of irregularities are observed in the floating point filter outputs. The irregularities observed in the floating point filter output might come from the fact that a floating point number system does not have the quantization error which is proportional to the magnitude represented [12]. Or they might be caused by the zero which is

one of the representable numbers in the floating point number system. In the experiment, zero can be produced in the conversion to the floating point number or in the arithmetic if zero is the nearest representable number. It means that an underflow in the floating point arithmetic or the conversion to that number gives zero in the result. Further examination is not made for the irregularities. The magnitude of the response of the logarithmic filter or the floating point filter is not changed much except for that of the 6th order logarithmic filter of Fig. 5.6.

Fig. 5.7 and 5.8 are for the 8th order bandpass filter. If the waves are closely examined, the logarithmic filter output resembles that of the long floating point filter more than the floating point filter output. Fig. 5.9 is the unit sample response for the three filters.

5.1.3. Further experiments of a high Q filter

The bandpass filter of Q = 393 is further examined for several single frequency inputs. Table 5.4 shows that the error to signal ratios for both the logarithmic filter and the floating point filter and the references to the output plots of Fig. 5.10 to 5.21. The error to signal ratios of the Table 5.4 are all in favor of the logarithmic filter over the floating point filter except one case for the frequency of a_1'. But if the Fig. 5.14 is closely watched, a phase shift can be observed. Considering the phase shift, the error to signal ratios can be computed as

$$\text{Error to signal ratio} = \sqrt{\frac{\sum(\ell y_n - w_{n-1})^2}{\sum w_{n-1}^2}} \quad \text{for the logarithmic filter}$$

$$\text{Error to signal ratio} = \sqrt{\frac{\sum(f y_n - w_{n-1})^2}{\sum w_{n-1}^2}} \quad \text{for the floating point filter}$$

where w_n, ℓy_n and fy_n are the output of the long floating point, the floating point, and the logarithmic filters respectively. Then, the error to signal ratio for the input frequency of a_1' becomes as in the parenthesis.

Table 5.5 shows the very rough magnitude of responses for those 7 input frequencies for the three filters. Since the magnitudes for the frequencies of a_3 and a_3' are roughly equal for the three filters, the differences of the magnitude response for the input frequencies of a_2, a_1, 0.785, a_1', and a_2' indicates that the filter performances are different for the three filterings. And those magnitudes of the logarithmic filter are closer to those of the long floating point filter than those of the floating point filter. The logarithmic filter's closeness to the long floating point filter is also observed in all the figures of 5.10 to 5.21 and also of 5.2 to 5.9.

Since the filter tested in this section has very high Q, slight changes of coefficients which are expected to happen in the conversion process are supposed to cause great differences in the filter performance. Table 5.6 shows the actual value of the coefficients of the three filters after the coefficients conversion. Not very much difference is made as long as the numbers are concerned. But the squared magnitude frequency responses of the three filters are much different as shown in Fig. 5.22. The squared magnitude frequency responses for the filters are computed as

$$\left|\frac{N(z)}{D(z)}\right|^2 = \frac{N(z)N(z^{-1})}{D(z)D(z^{-1})}$$

where $$N(z) = \sum_{i=0}^{0} B(i)z^{-i}$$

$$D(z) = \sum_{i=0}^{2} A(i)z^{-i} \quad .$$

$z = e^{j\omega}$

A(i) and B(i) are in Table 5.6 for the three filters

It is again shown that the logarithmic filter is closer to the long floating point filter than the floating point filter. The experimental magnitude responses of Table 5.5 roughly agree with Fig. 5.22 if the values of Table 5.5 are squared.

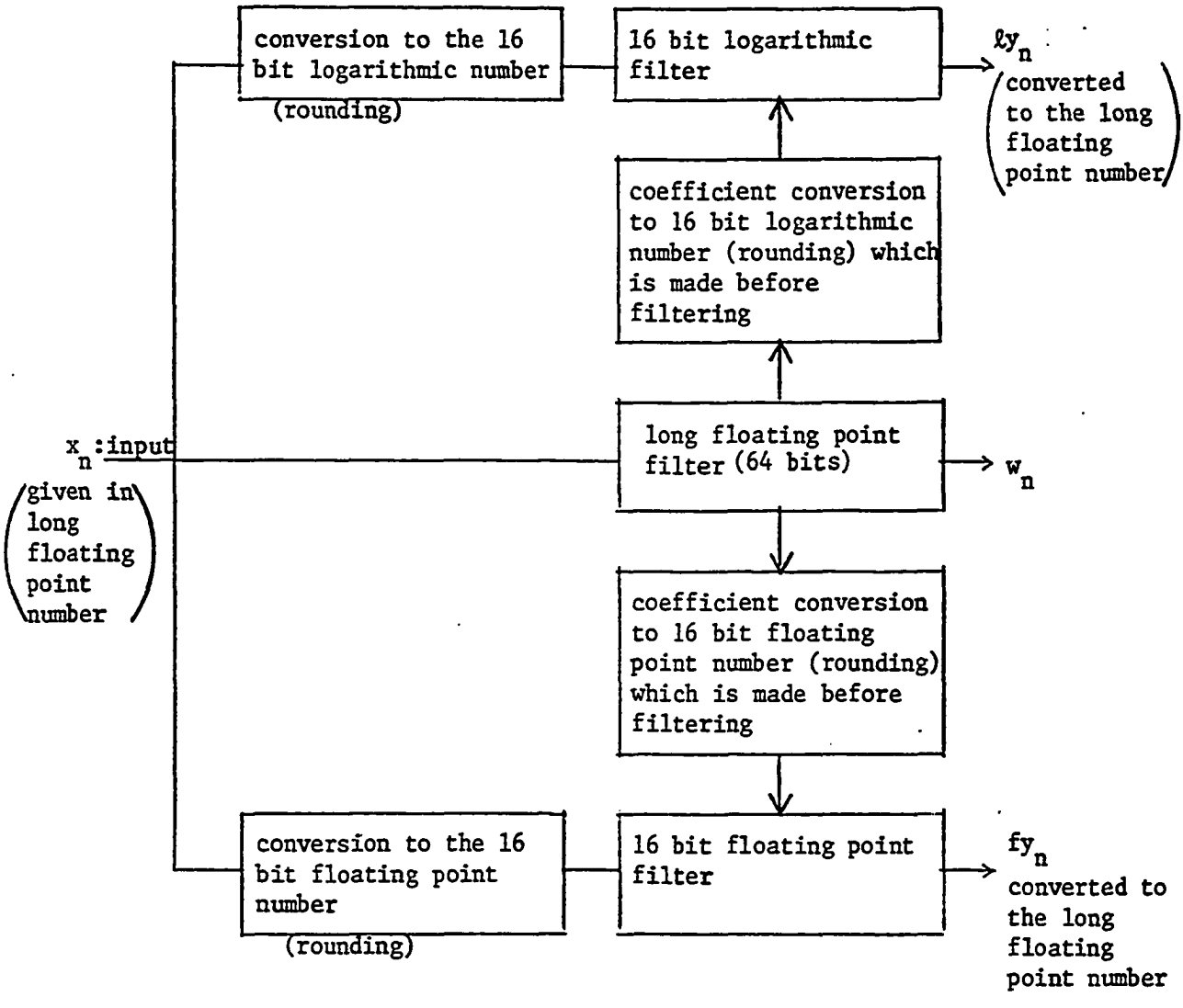
Although only one example is shown, it can be said that a filter with a very high Q is severely affected by the coefficient quantization, if the number of bits of a word is limited like 16.

5.2 Cascade and Parallel Form Digital Filters

Three filters: one is of cascade form; the other two are of parallel form. The method used is exactly the same as that of section 5.1 except for the filter forms. The filter forms used in this section are depicted in Fig. 5.23 where all arithmetic operations from input to output are made in each of the number systems described in section 5.1. The summary of the experiment is given in Table 5.7 and the output graphs are shown in Fig 5.24 to 5.26. For all three filters, the error to signal ratio of the logarithmic filters are smaller than those of the floating point filters. The output graphs of Fig 5.24 and 5.25 do not show much difference between those two number system filters. But the graph of Fig 5.26 shows the advantage of the logarithmic filters over the floating point filter, although there are observable differences between the logarithmic filter output and the ideal long floating point filter output. The filter used for the output of Fig 5.26 has 8th order. It has more irregularities in the output than the direct form sixth order filter output of Fig 5.6. It may suggest that the filter order is related to the irregularities in lowpass filters.

Note: The filters of Table 5.8 and 5.9 are sample Chebyschev digital filters

[2] on pages 223 and 221 respectively. The filter of Table 5.10 is a Butterworth digital filter [14] with OMEGA = 0.5750 (OMEGA is a notation of [14]).



Note: ℓy_n, w_n and fy_n are plotted in the same scales

$\ell e_n = \ell y_n - w_n$ for the logarithmic filter error

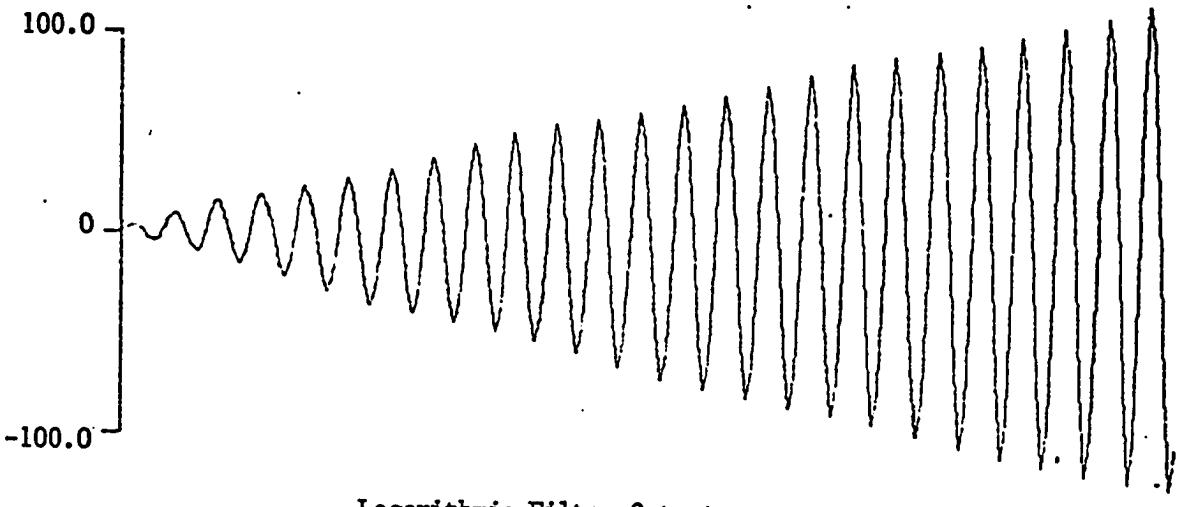
$fe_n = fy_n - w_n$ for the floating point filter error

The error to signal ratios in Table 5.1 are computed as

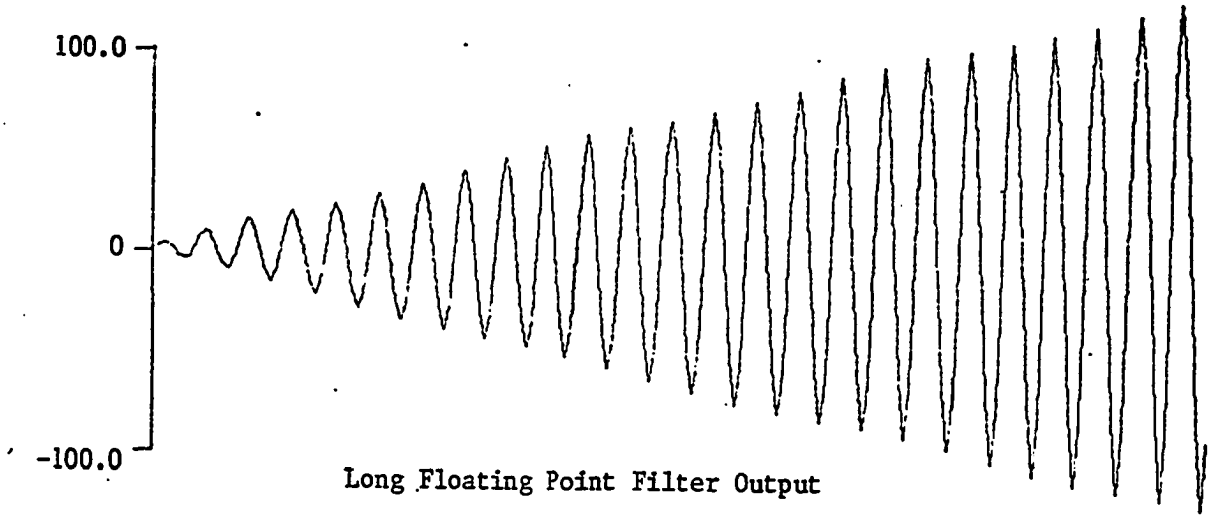
$\sqrt{\Sigma \ell e_n^2 / \Sigma w_n^2}$ for the logarithmic filter

$\sqrt{\Sigma fe_n^2 / \Sigma w_n^2}$ for the floating point filter

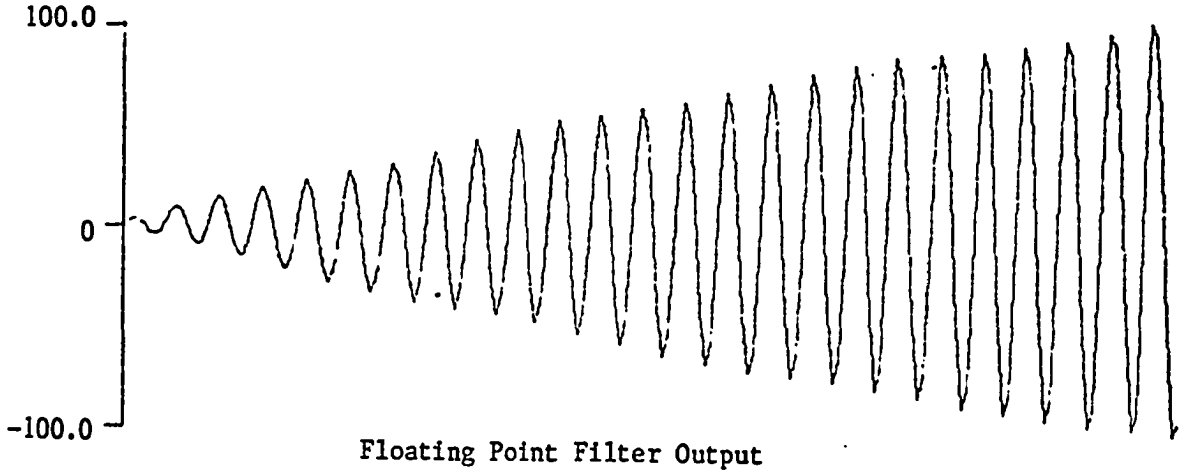
Fig 5.1 Experimental Comparison of Logarithmic and Floating Point Filters



Logarithmic Filter Output

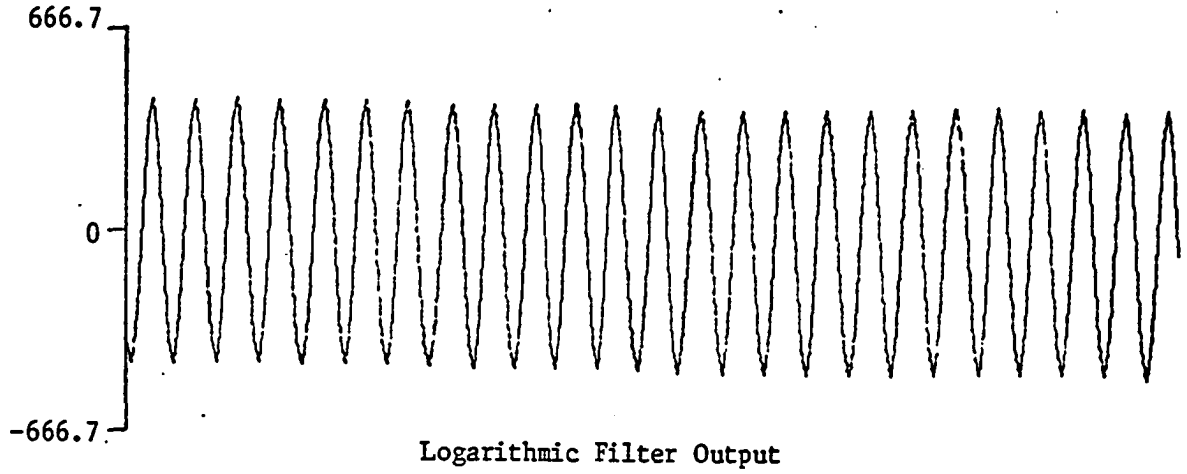


Long Floating Point Filter Output

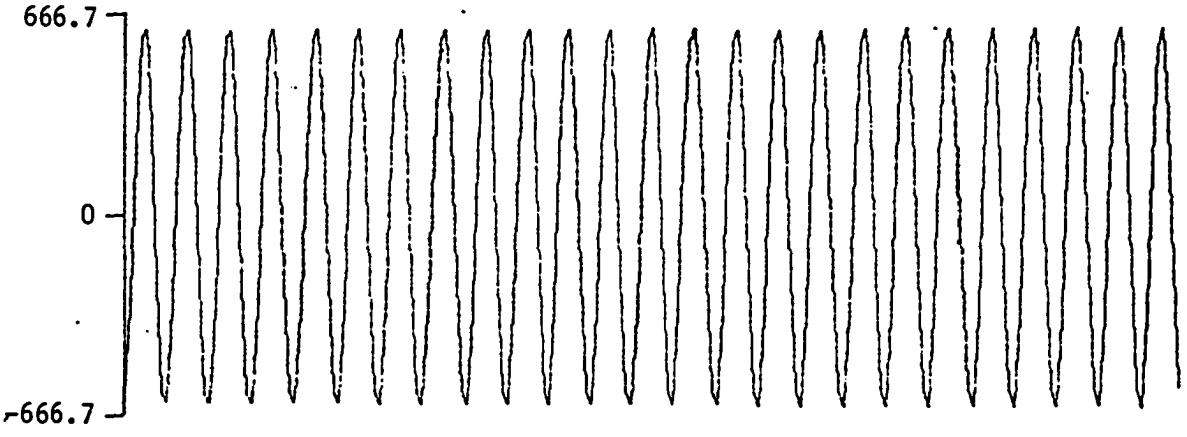


Floating Point Filter Output

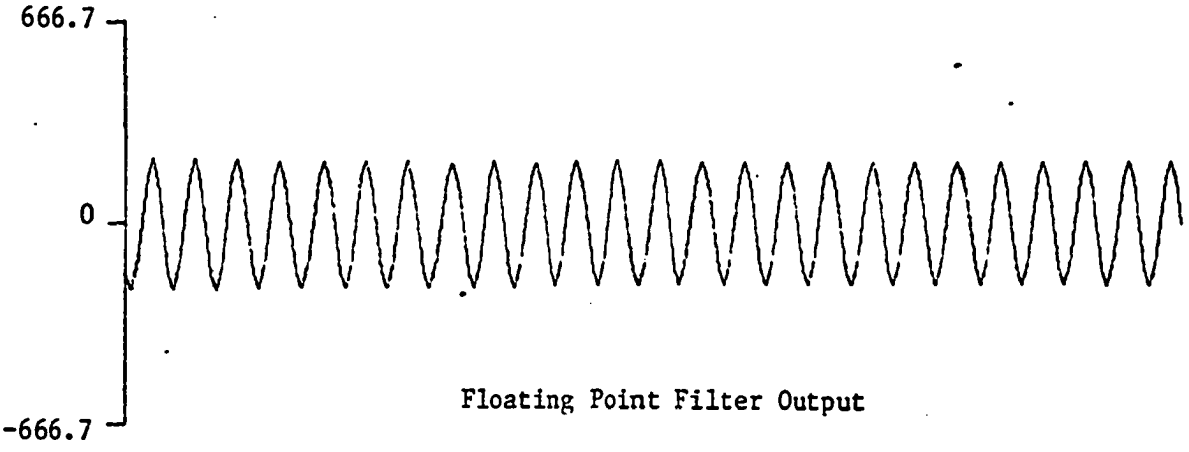
Fig 5.2



Logarithmic Filter Output

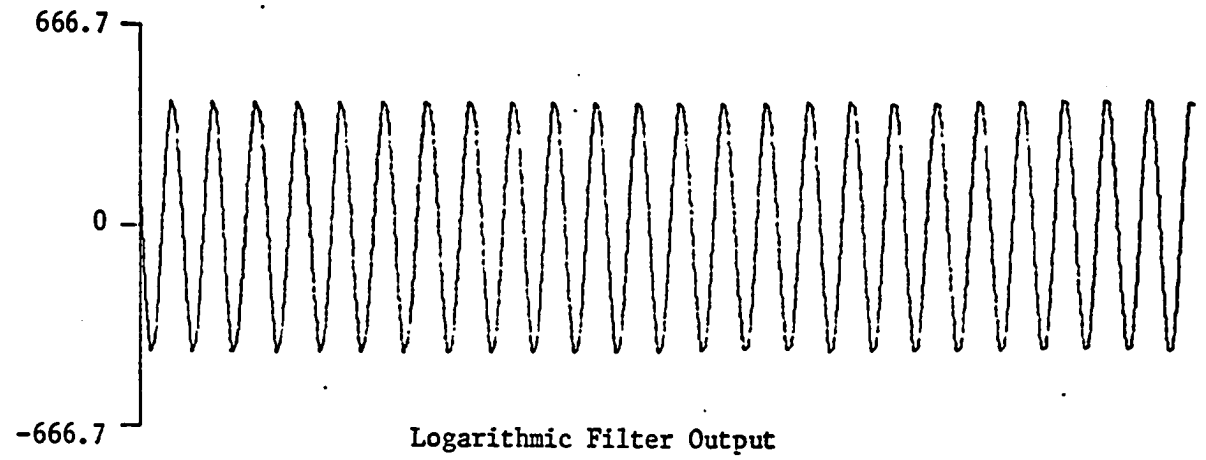


Long Floating Point Filter Output

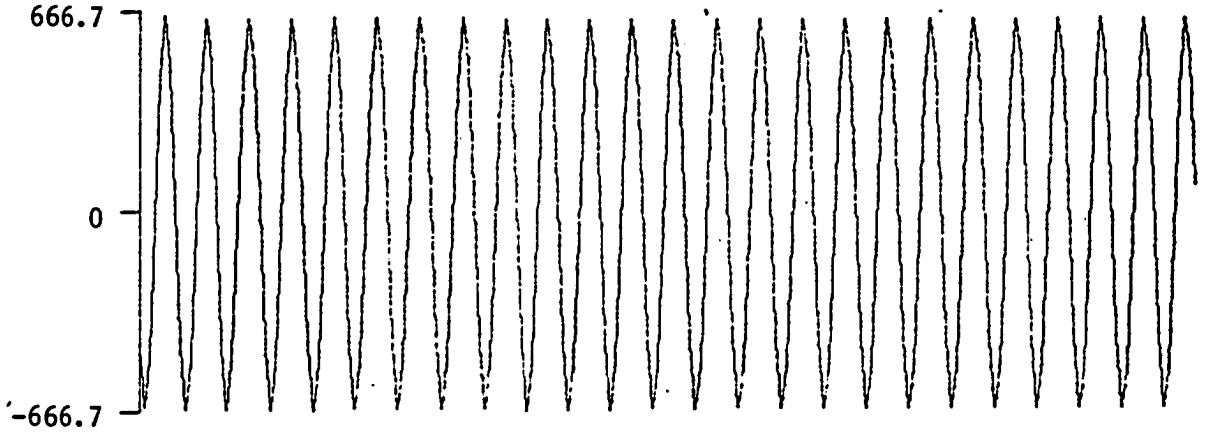


Floating Point Filter Output

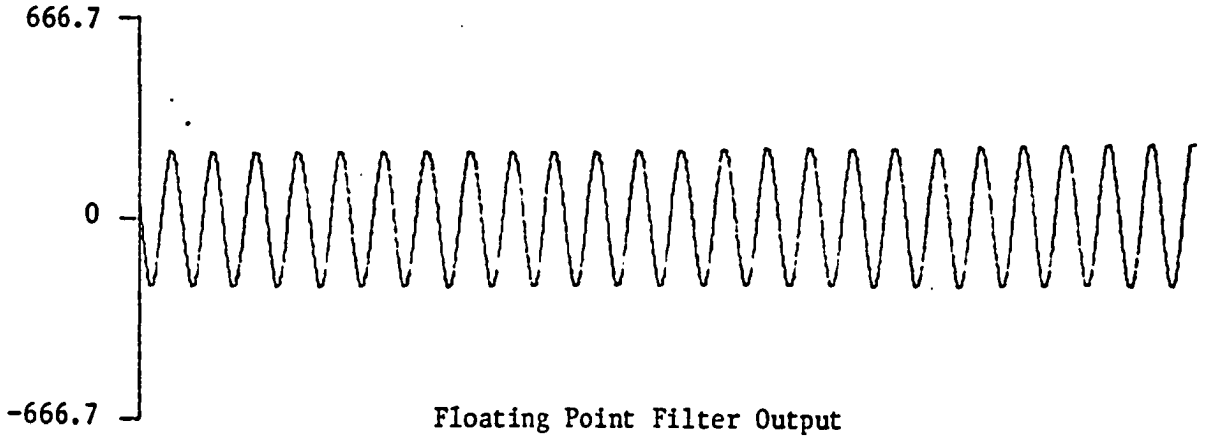
Fig 5.3



Logarithmic Filter Output

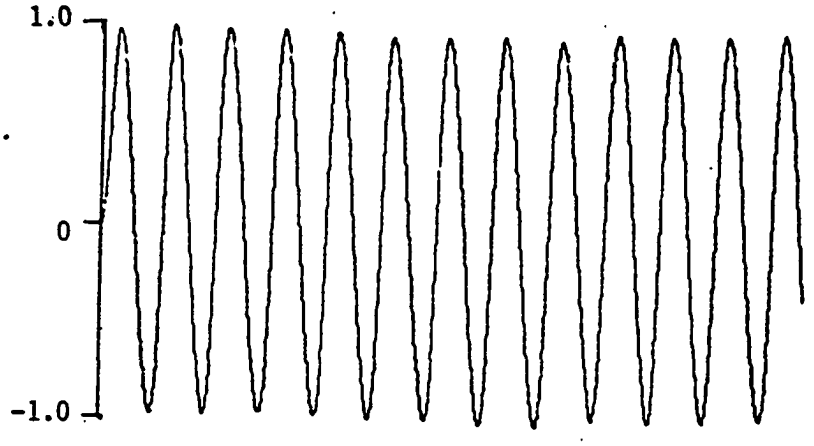


Long Floating Point Filter Output

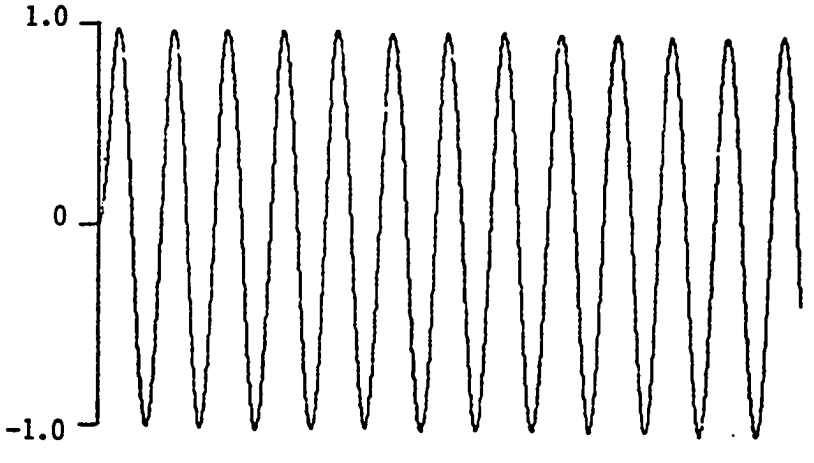


Floating Point Filter Output

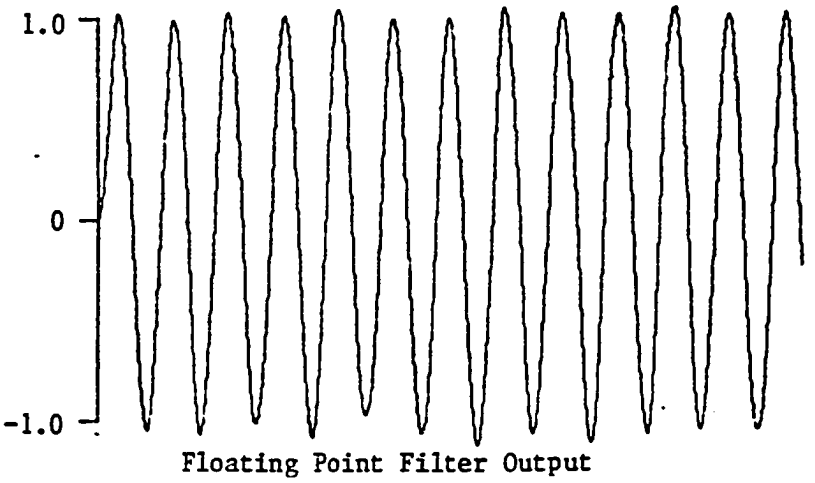
Fig 5.4



Logarithmic Filter Output

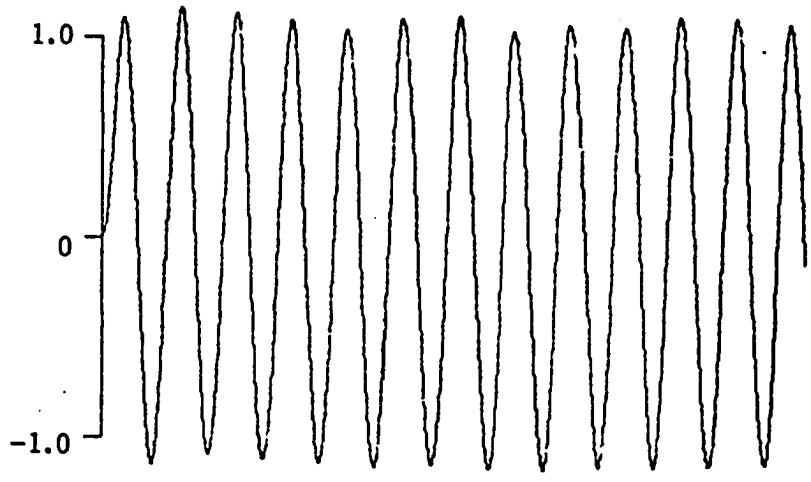


Long Floating Point Filter Output

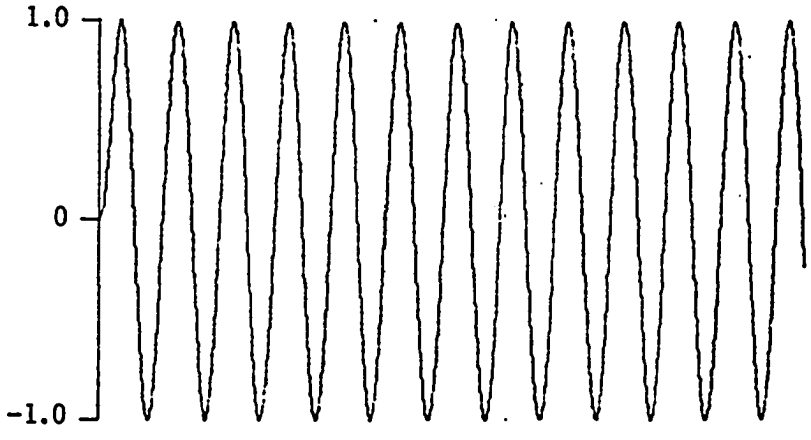


Floating Point Filter Output

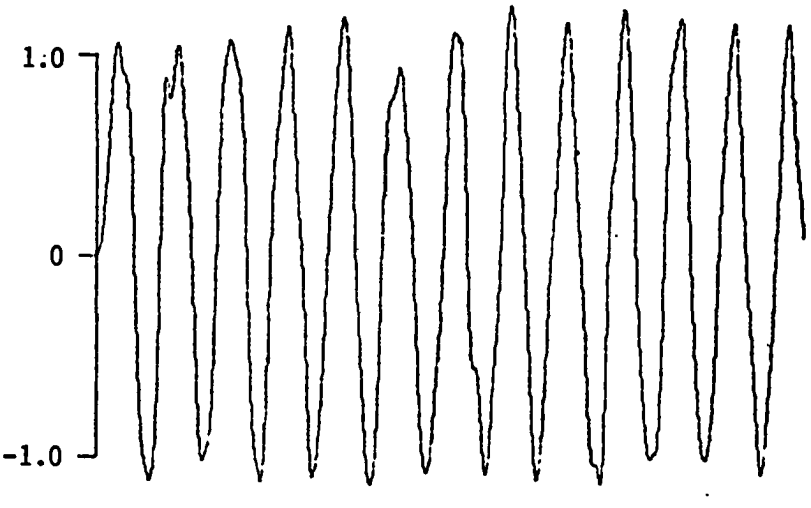
Fig 5.5



Logarithmic Filter Output

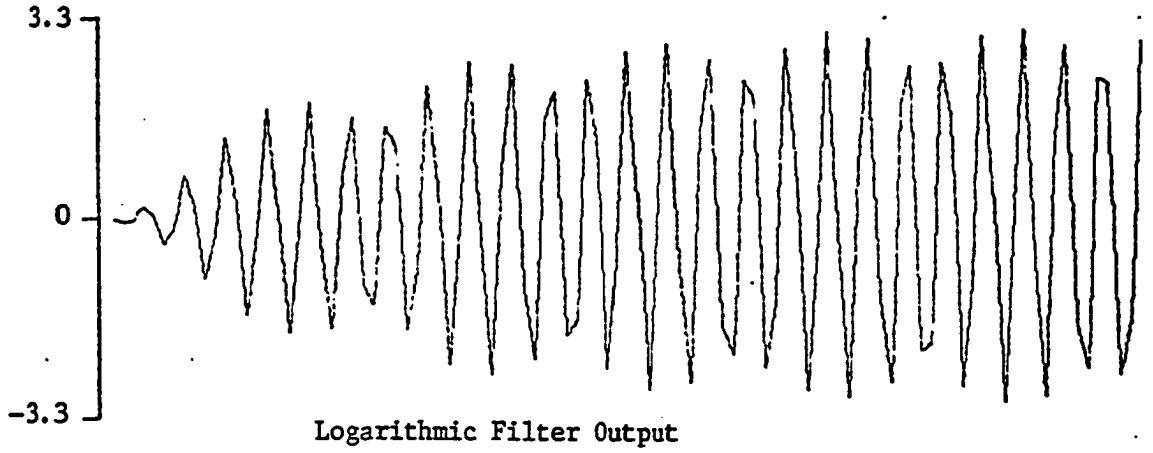


Long Floating Point Filter Output

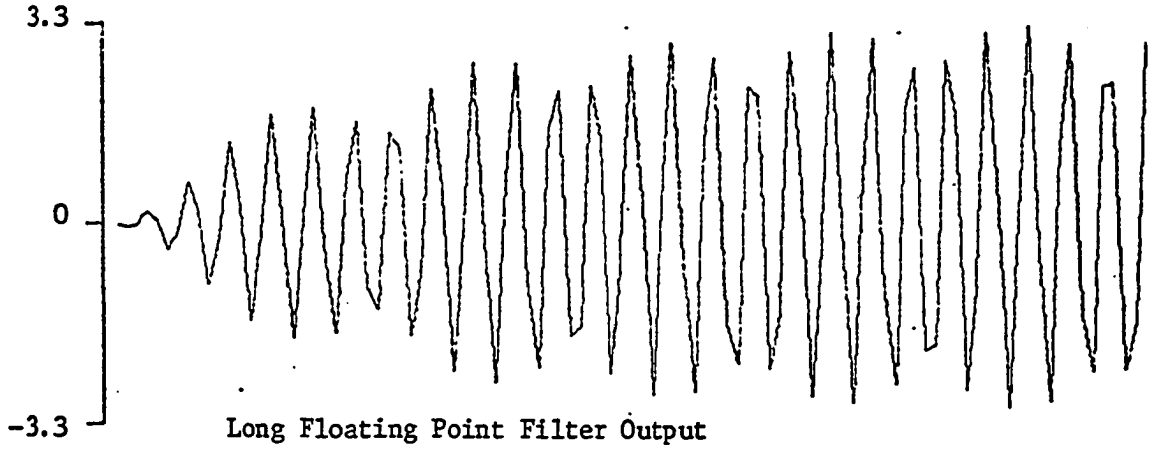


Floating Point Filter Output

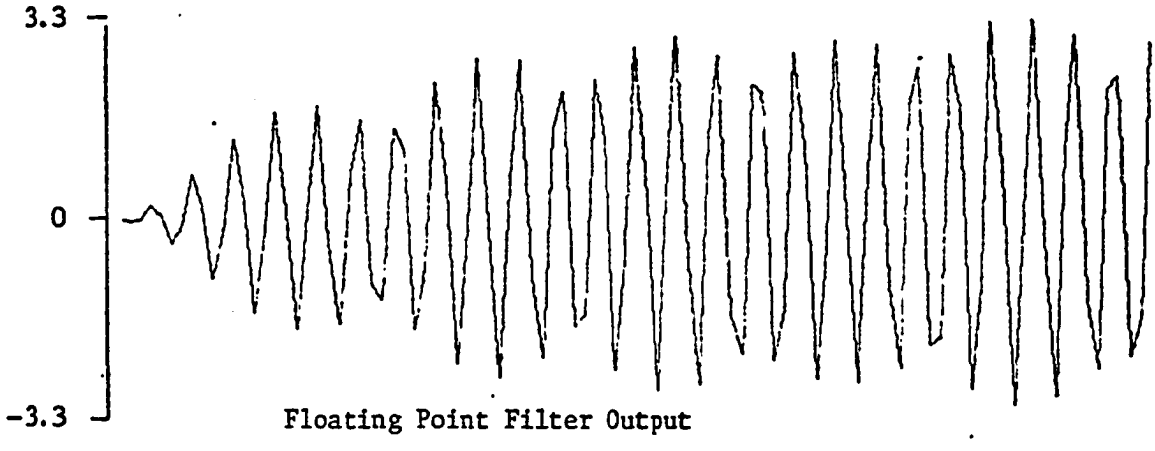
Fig 5.6



Logarithmic Filter Output

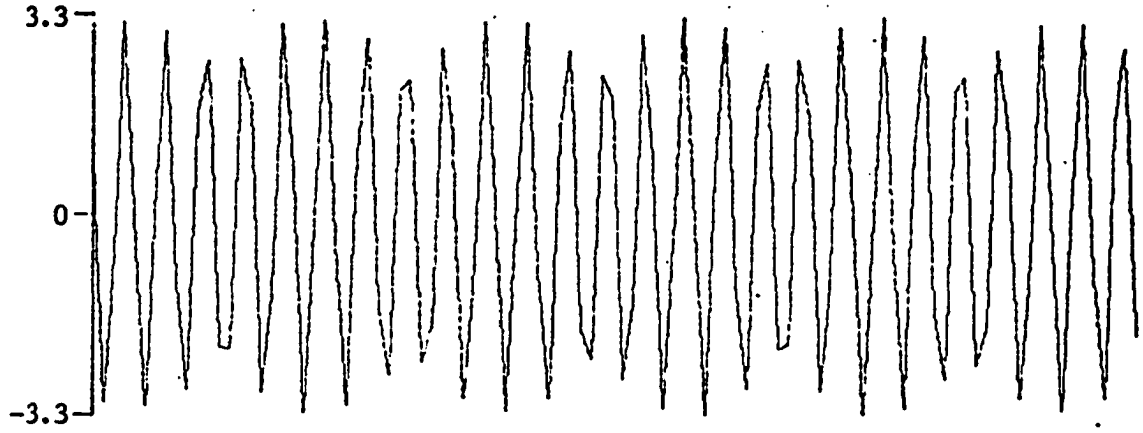


Long Floating Point Filter Output

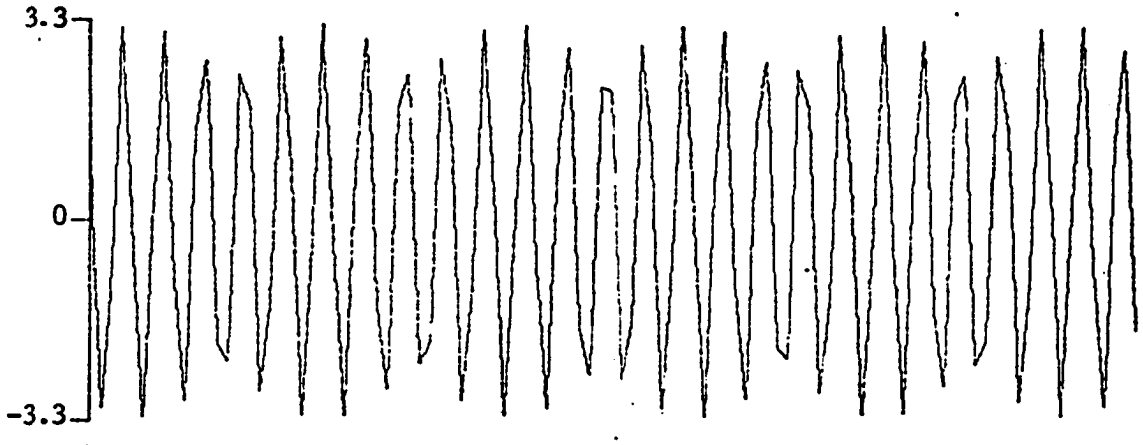


Floating Point Filter Output

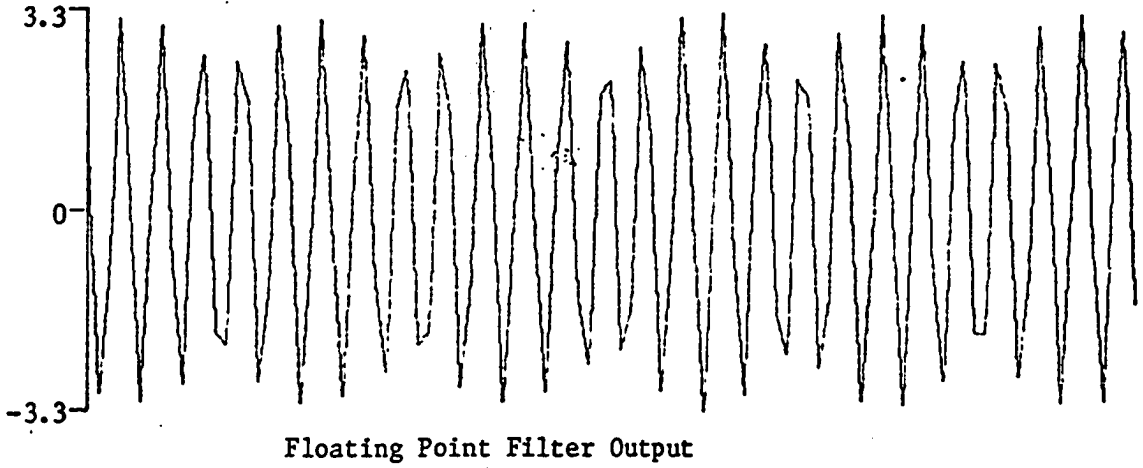
Fig 5.7



Logarithmic Filter Output

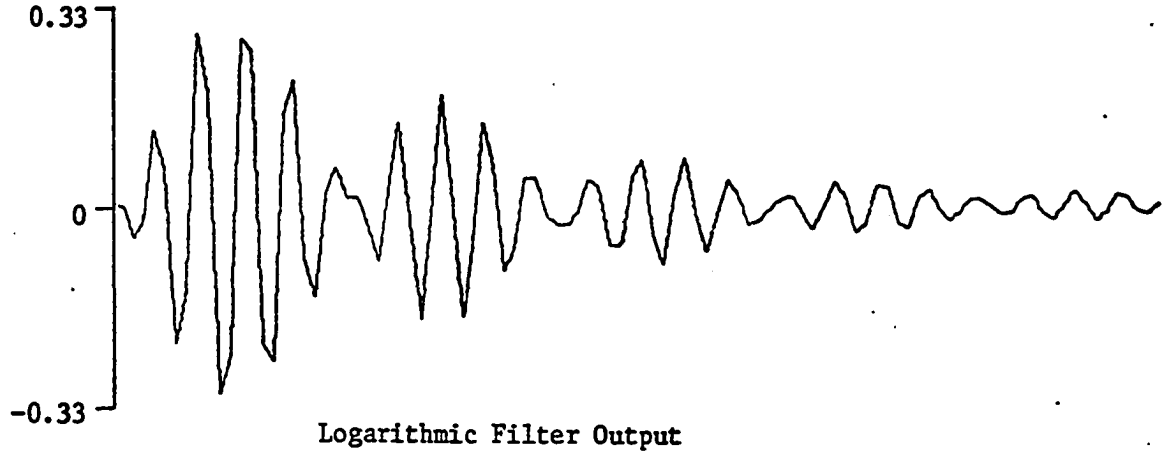


Long Floating Point Filter Output

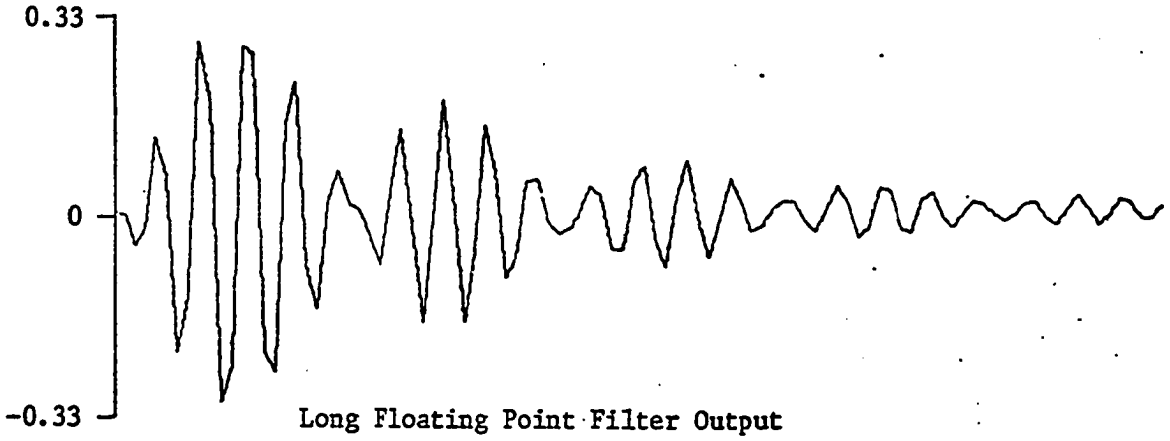


Floating Point Filter Output

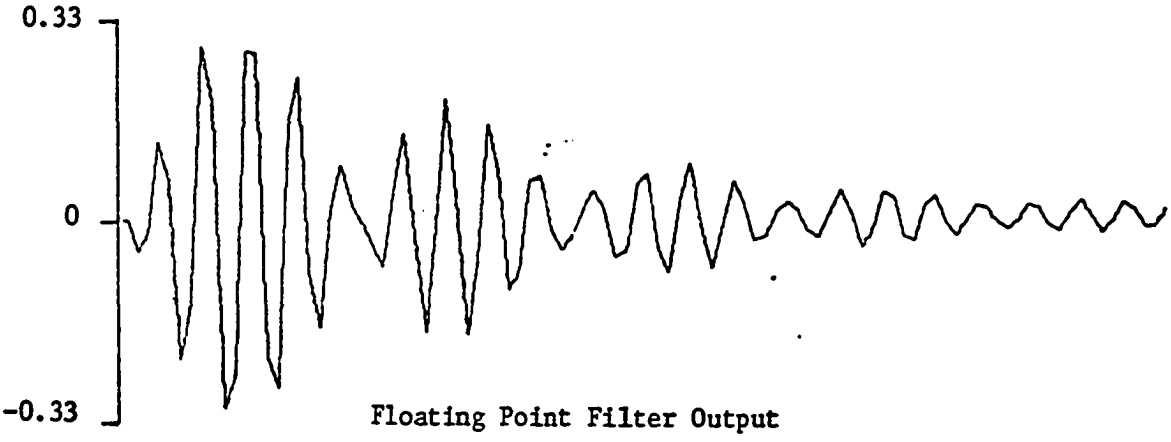
Fig 5.8



Logarithmic Filter Output

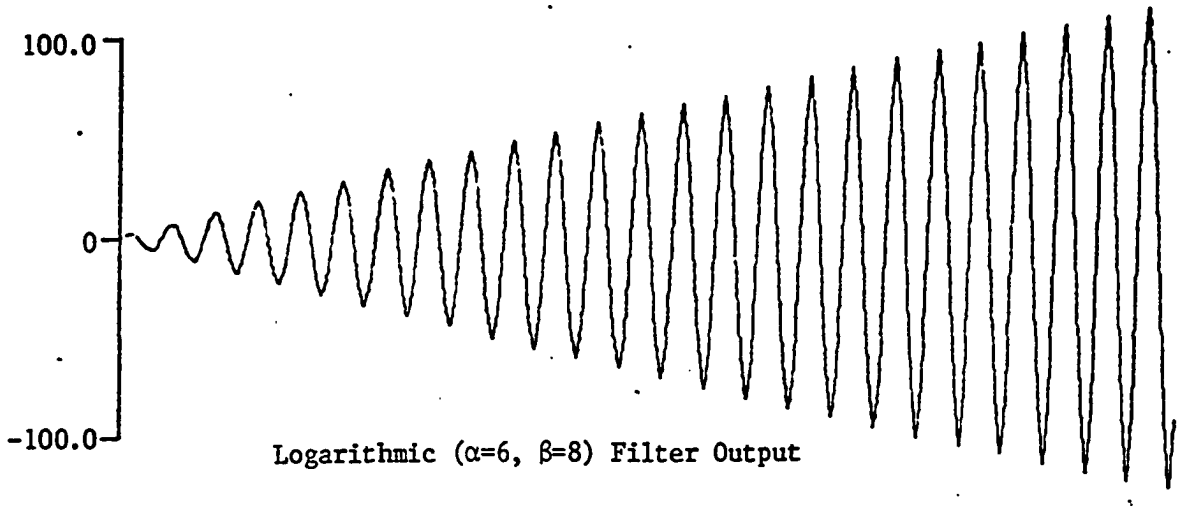


Long Floating Point Filter Output

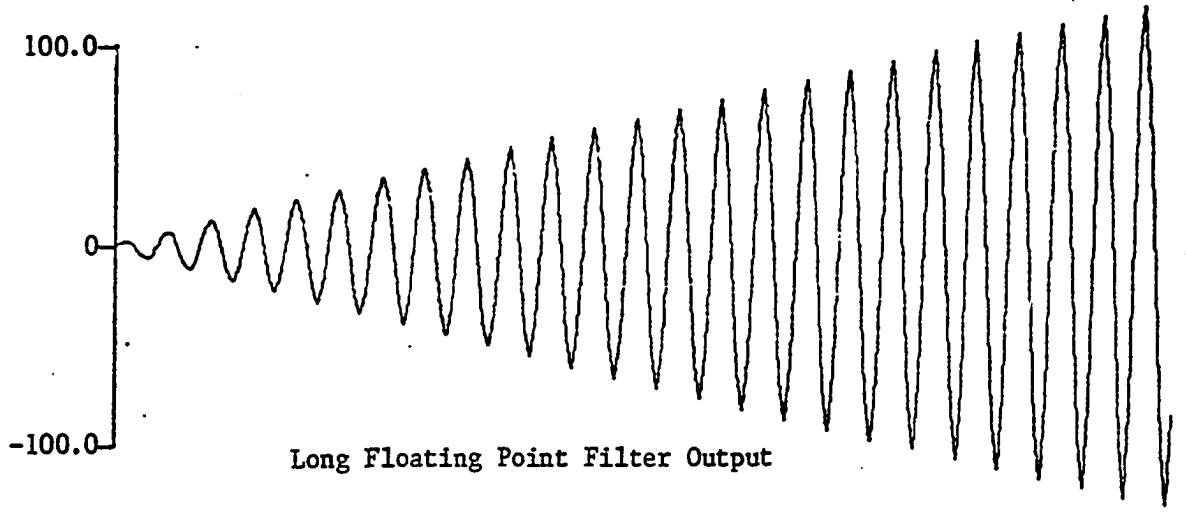


Floating Point Filter Output

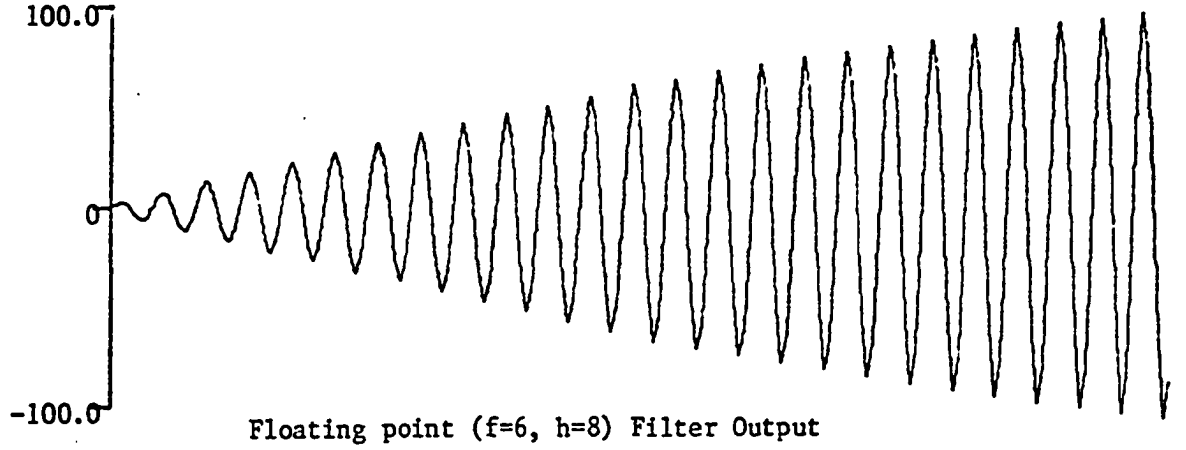
Fig 5.9



Logarithmic (α=6, β=8) Filter Output

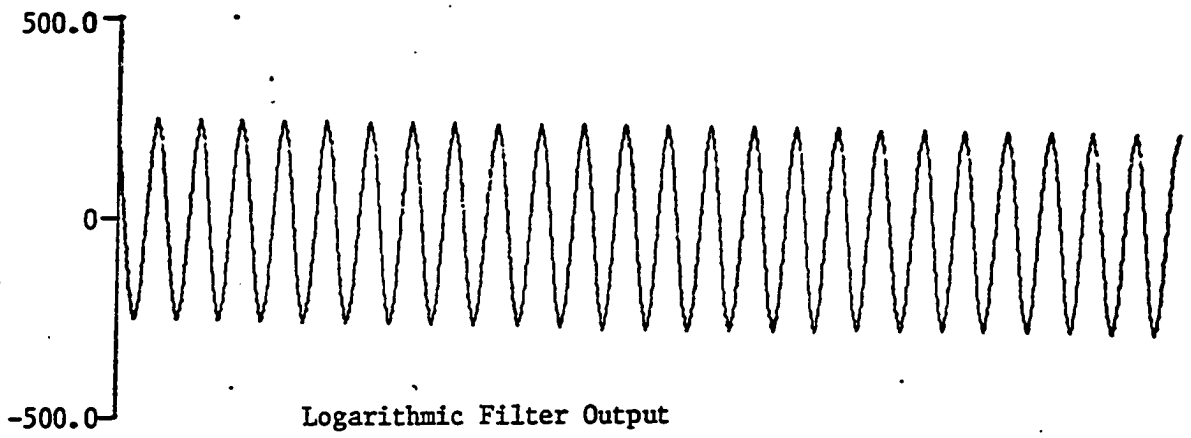


Long Floating Point Filter Output

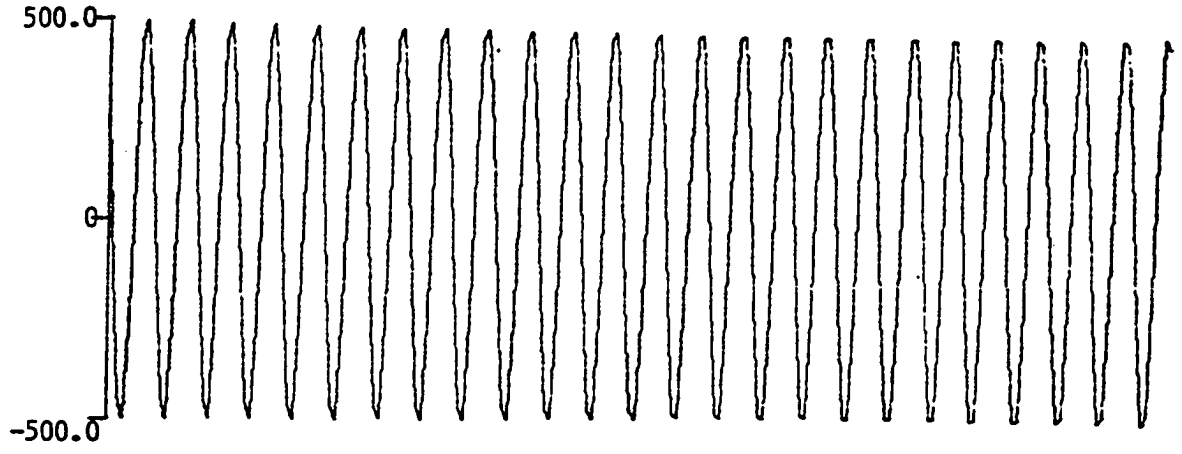


Floating point (f=6, h=8) Filter Output

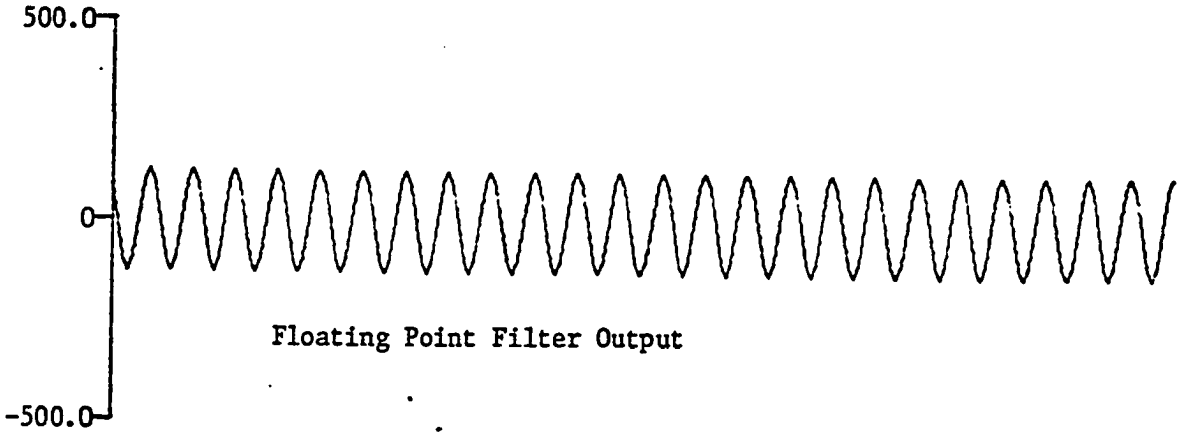
Fig 5.10



Logarithmic Filter Output

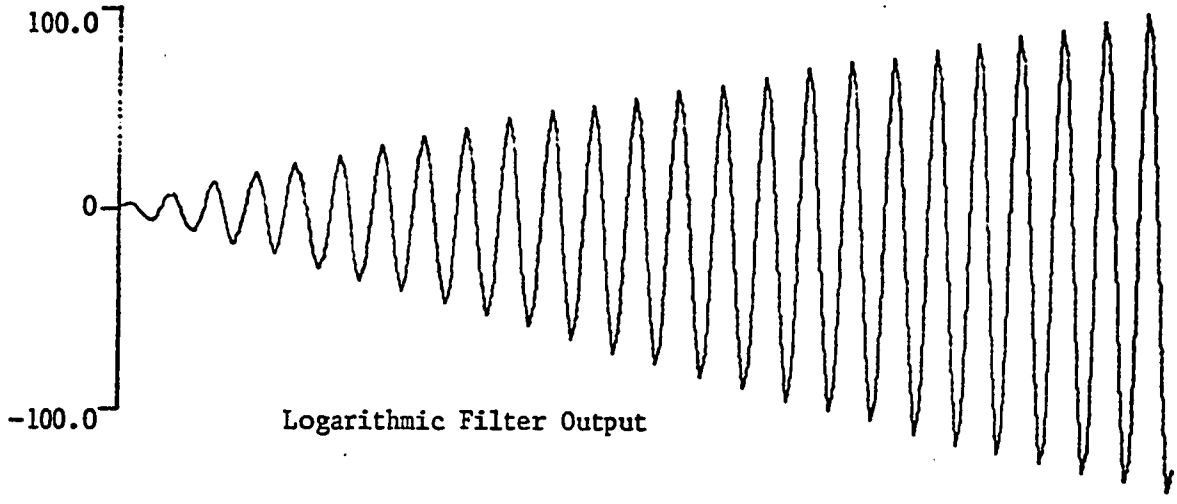


Long Floating Point Filter Output

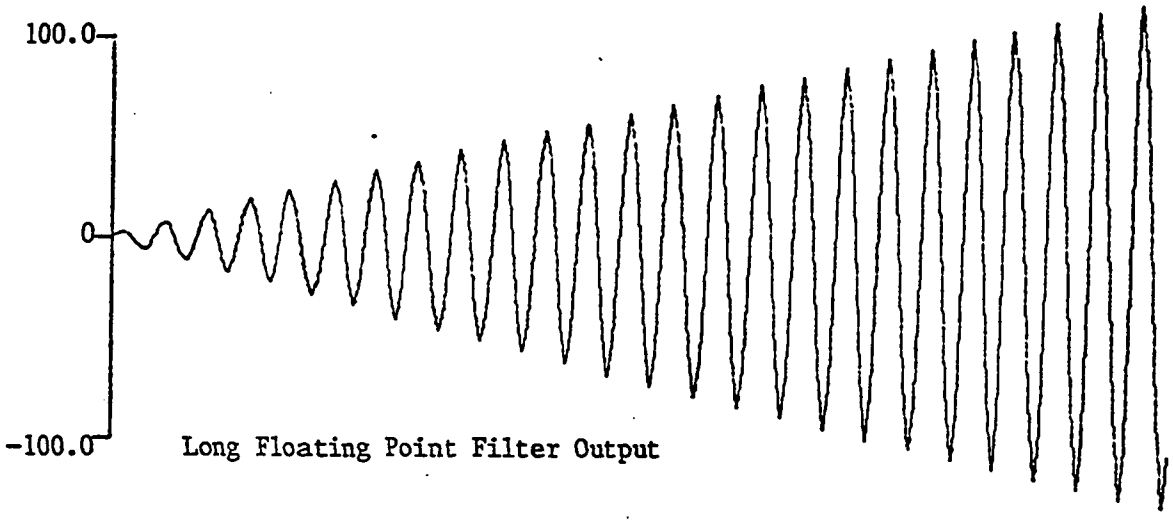


Floating Point Filter Output

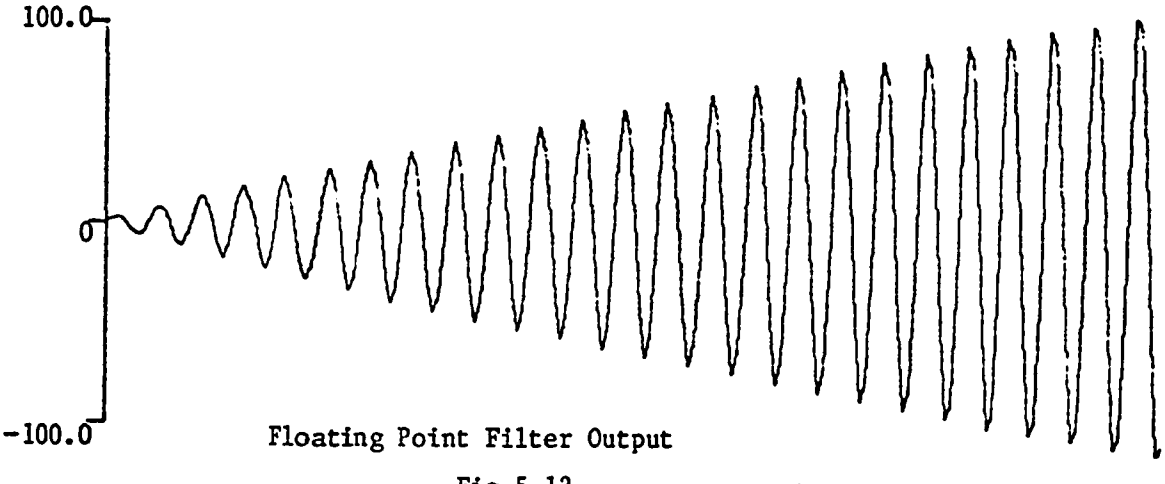
Fig 5.11



Logarithmic Filter Output

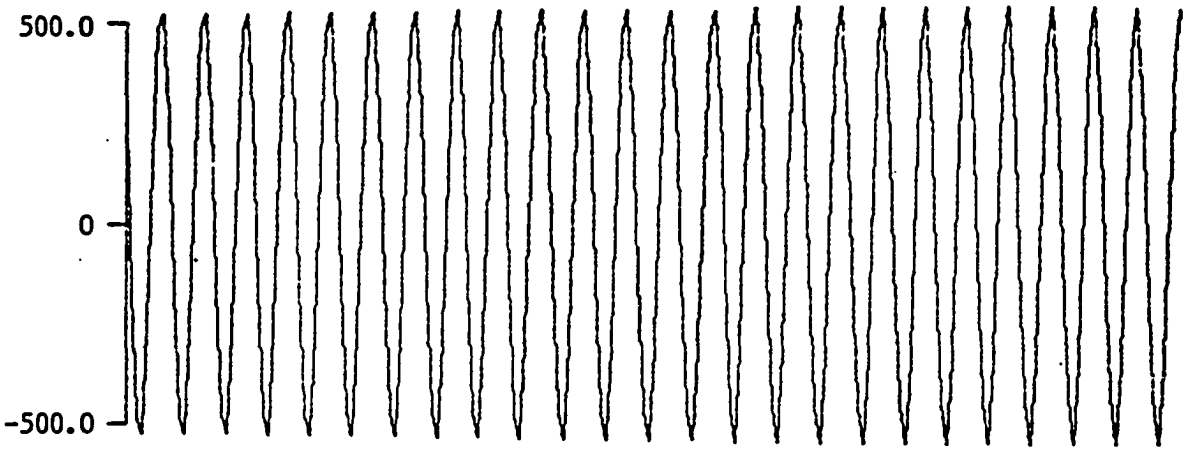


Long Floating Point Filter Output

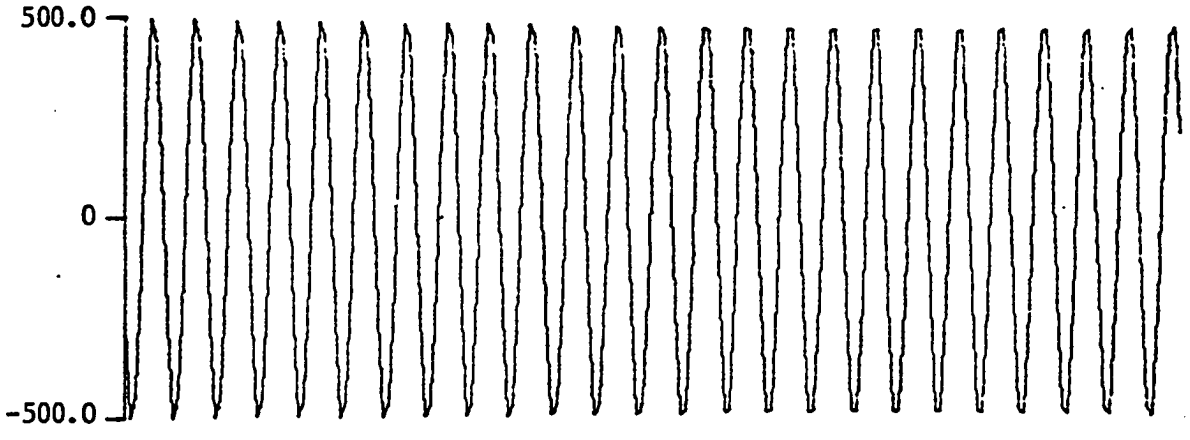


Floating Point Filter Output

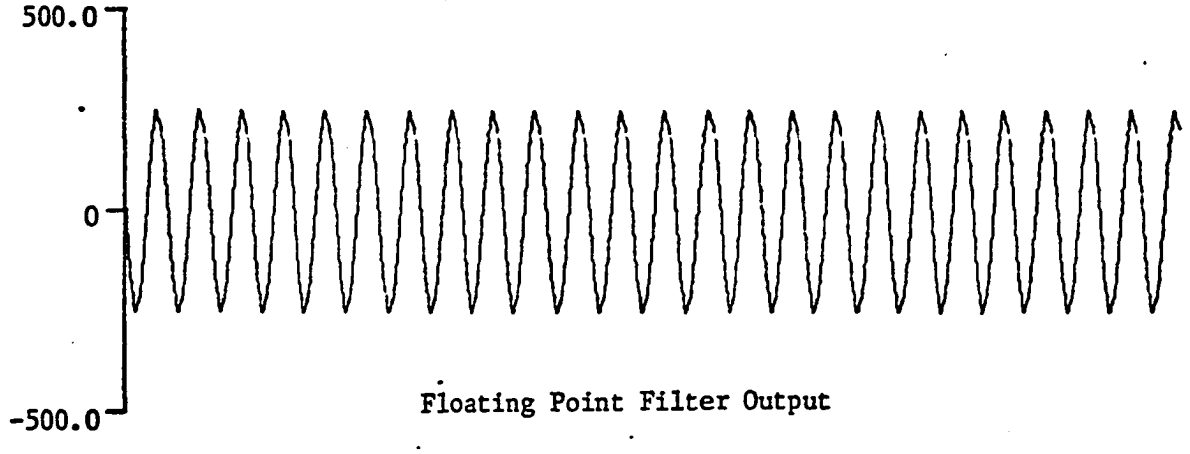
Fig 5.12



Logarithmic Filter Output

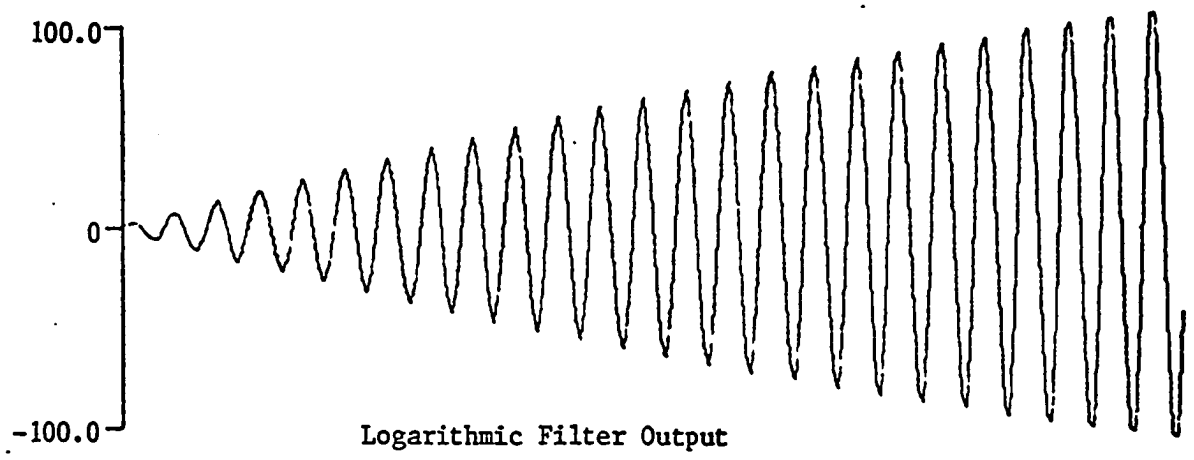


Long Floating Point Filter Output

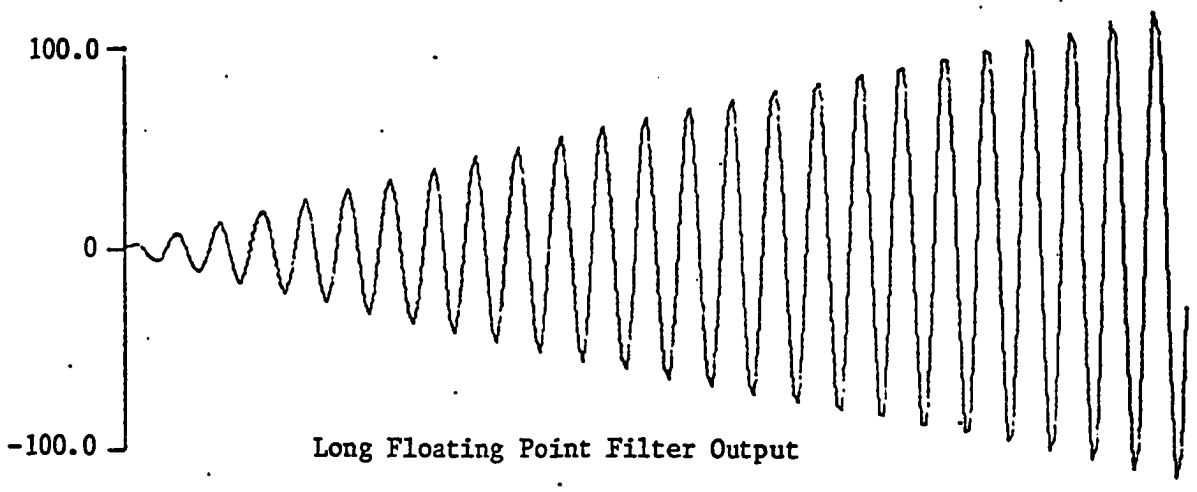


Floating Point Filter Output

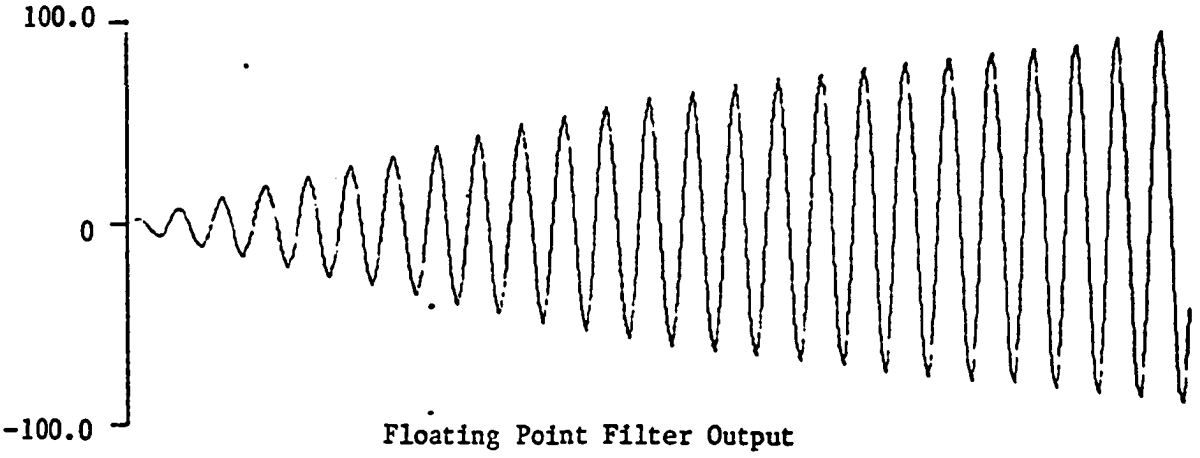
Fig 5.13



Logarithmic Filter Output

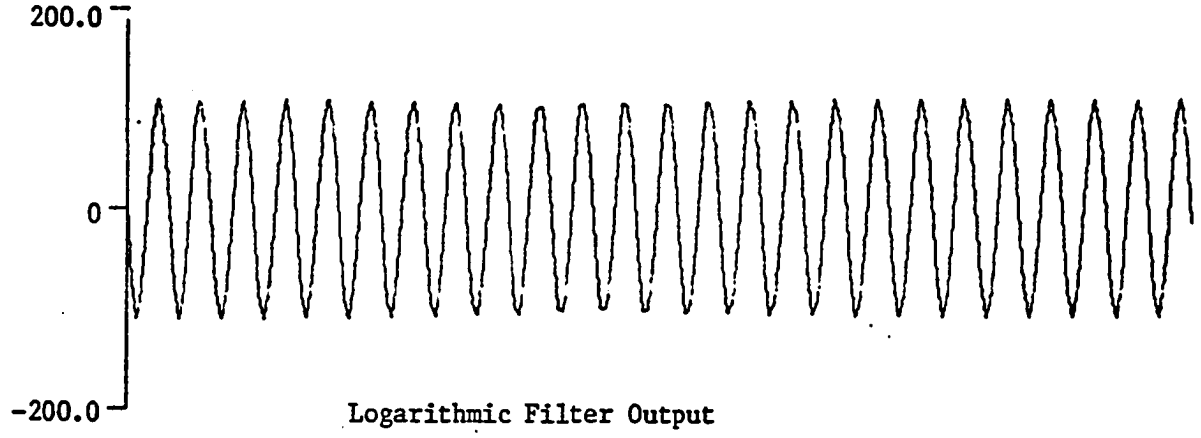


Long Floating Point Filter Output

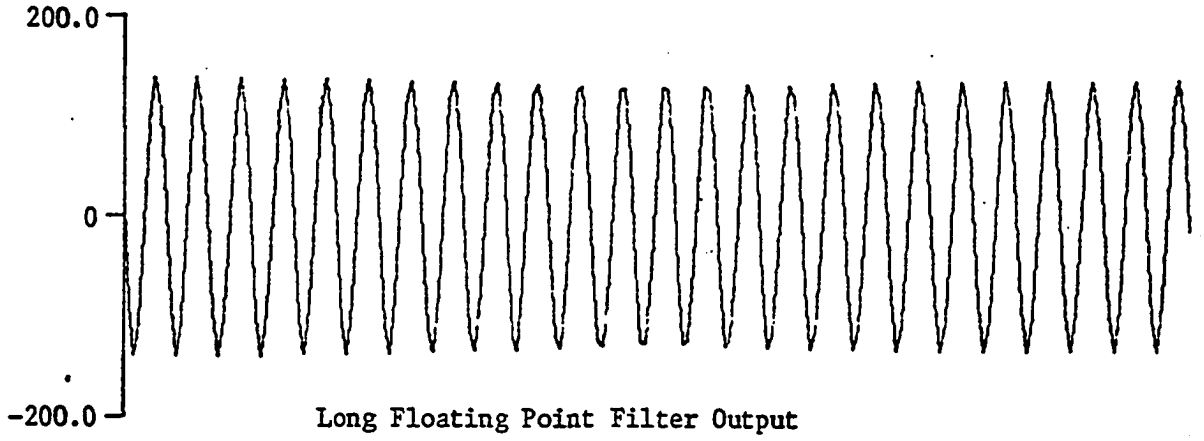


Floating Point Filter Output

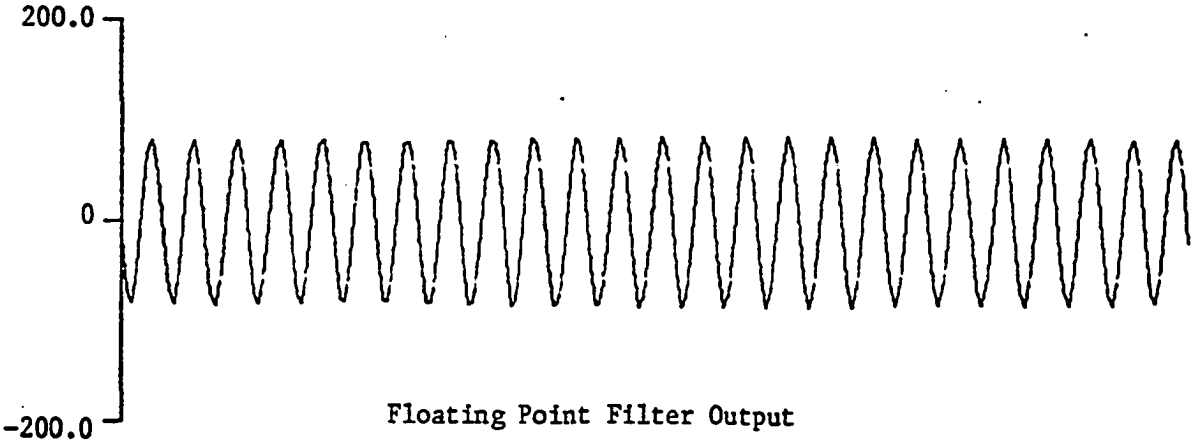
Fig 5.14



Logarithmic Filter Output

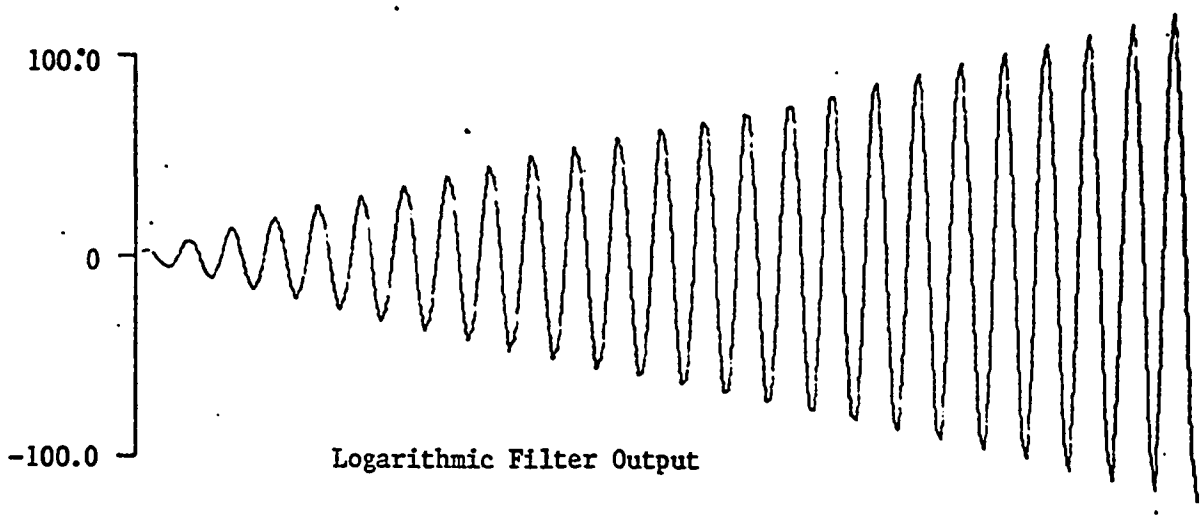


Long Floating Point Filter Output

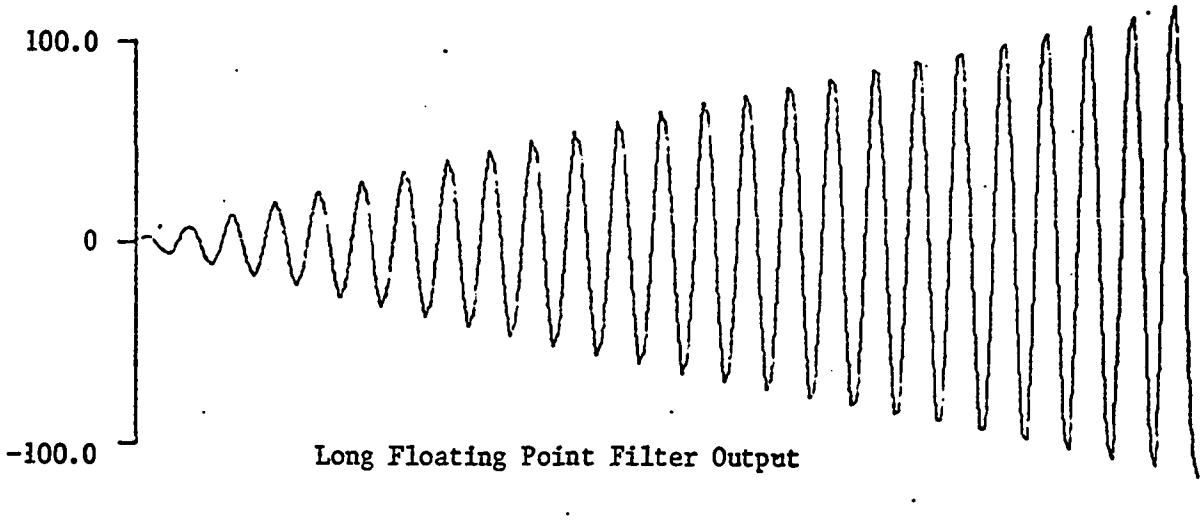


Floating Point Filter Output

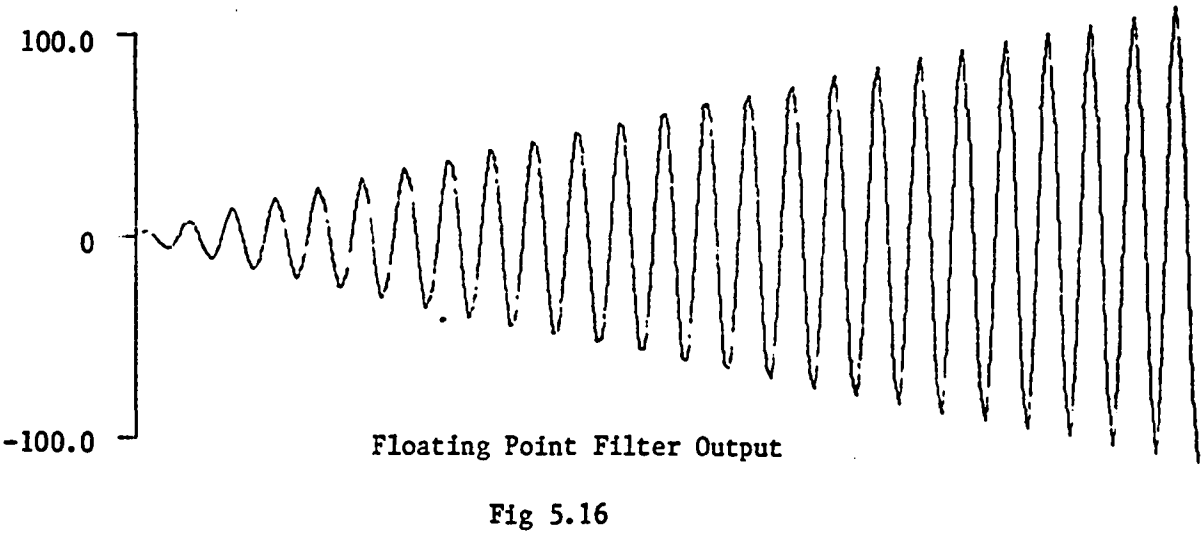
Fig 5.15



Logarithmic Filter Output

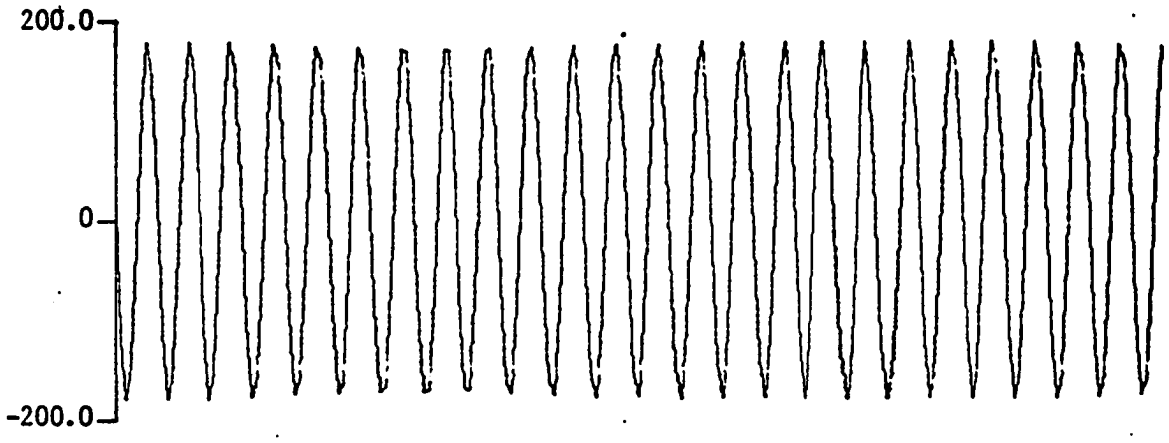


Long Floating Point Filter Output

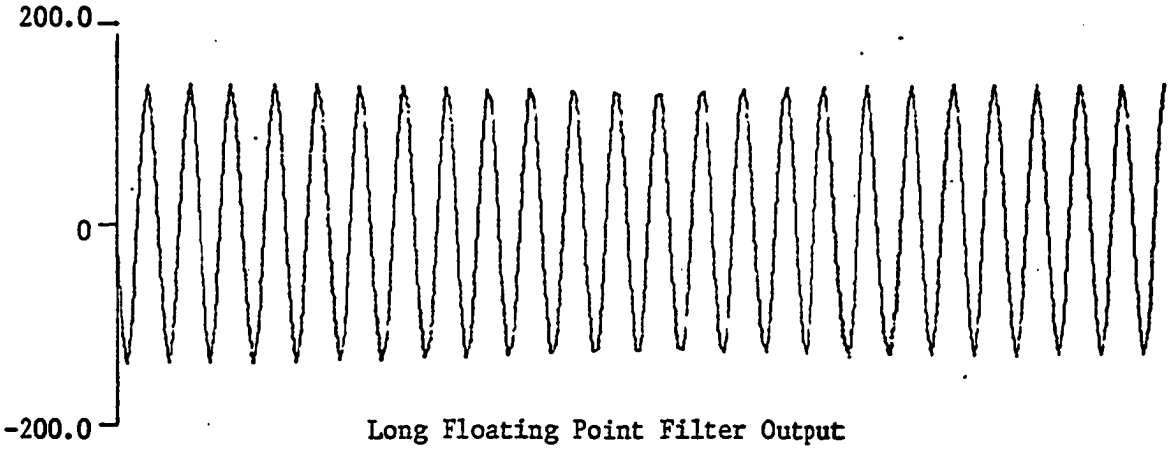


Floating Point Filter Output

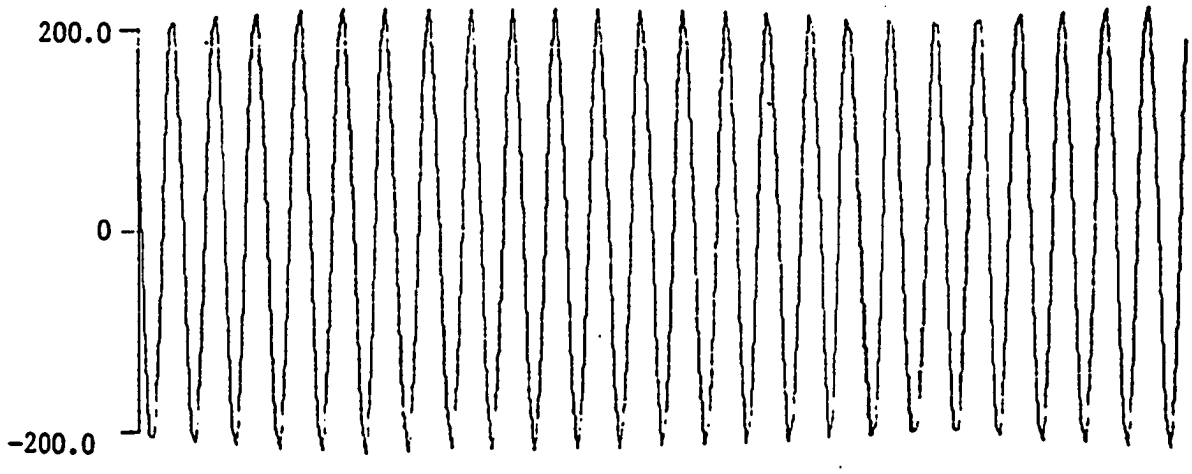
Fig 5.16



Logarithmic Filter Output

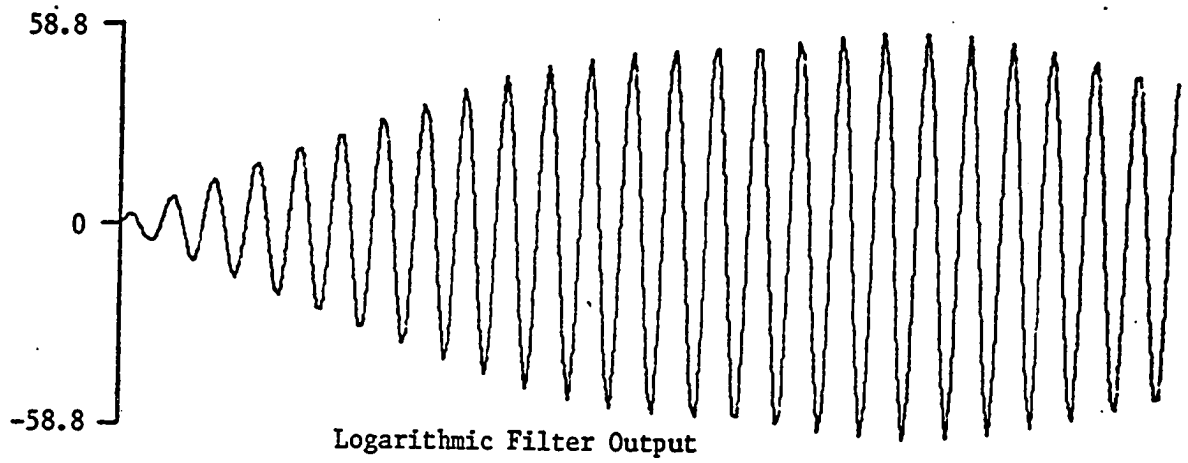


Long Floating Point Filter Output

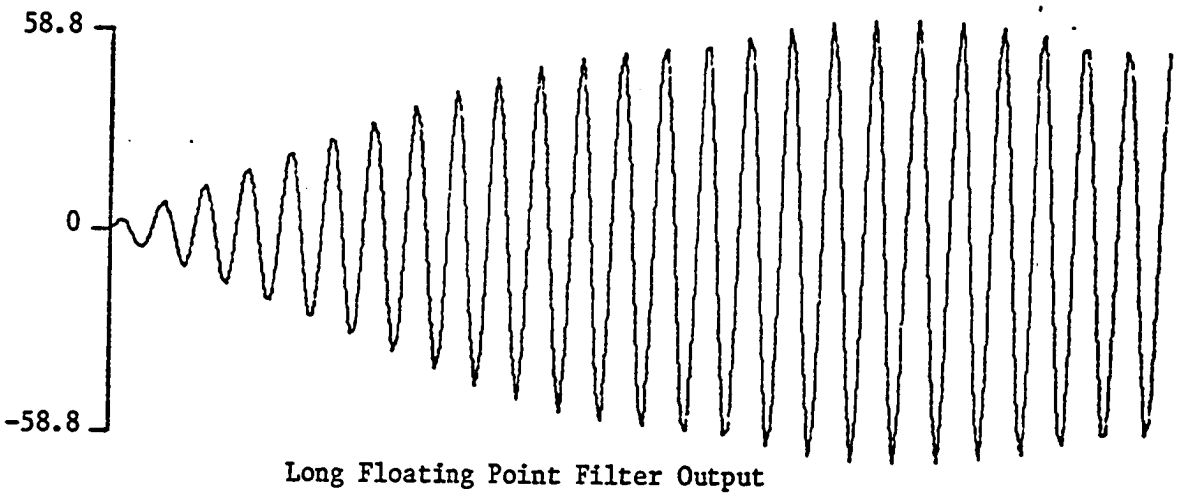


Floating Point Filter Output

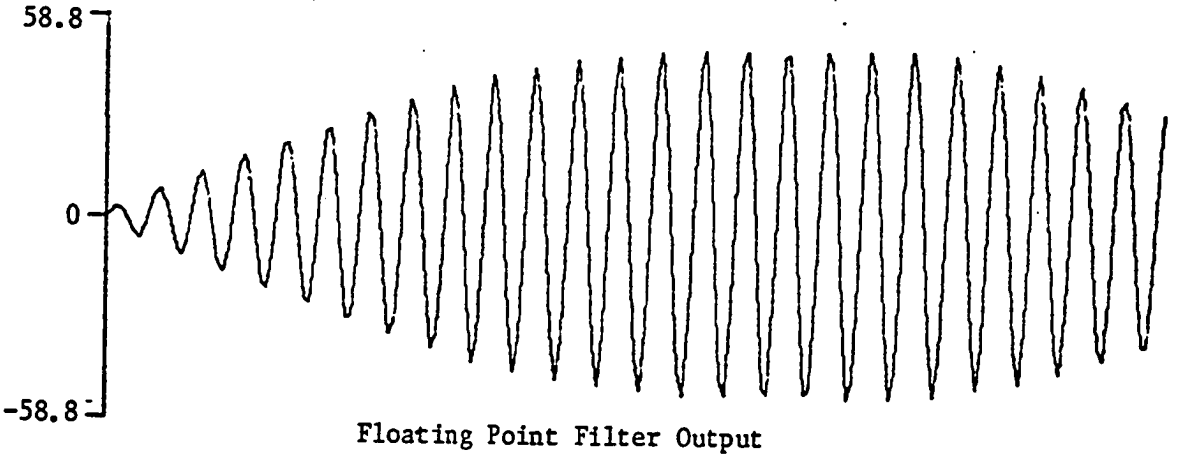
Fig 5.17



Logarithmic Filter Output

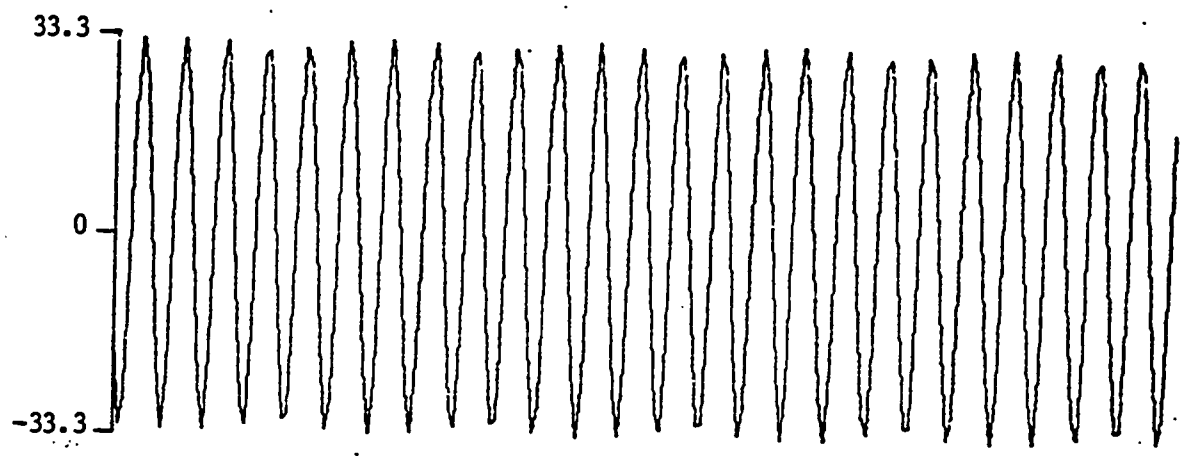


Long Floating Point Filter Output

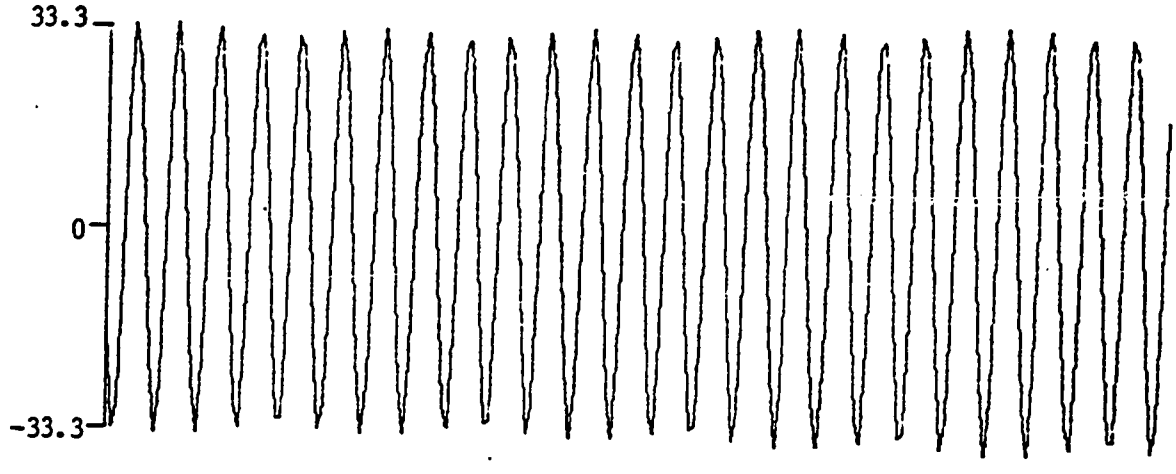


Floating Point Filter Output

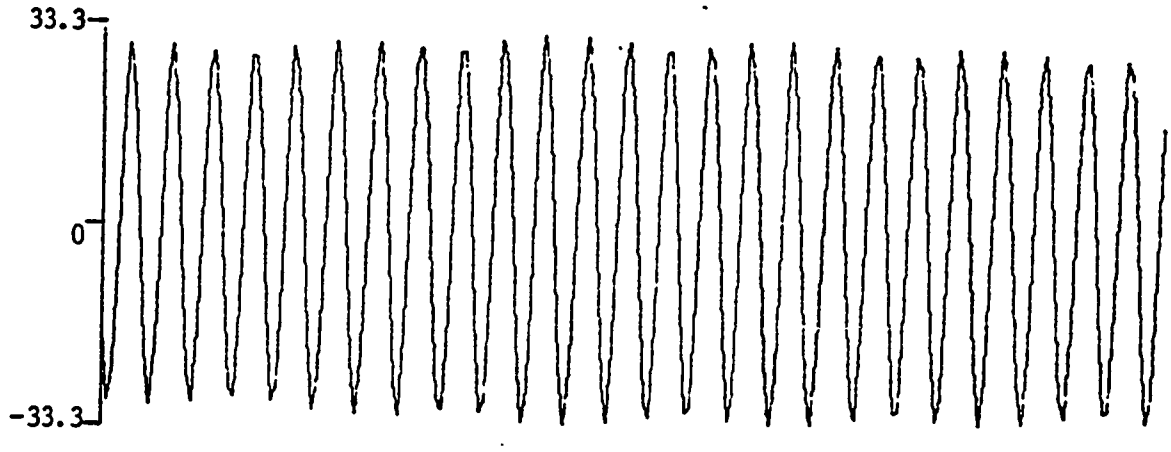
Fig 5.18



Logarithmic Filter Output

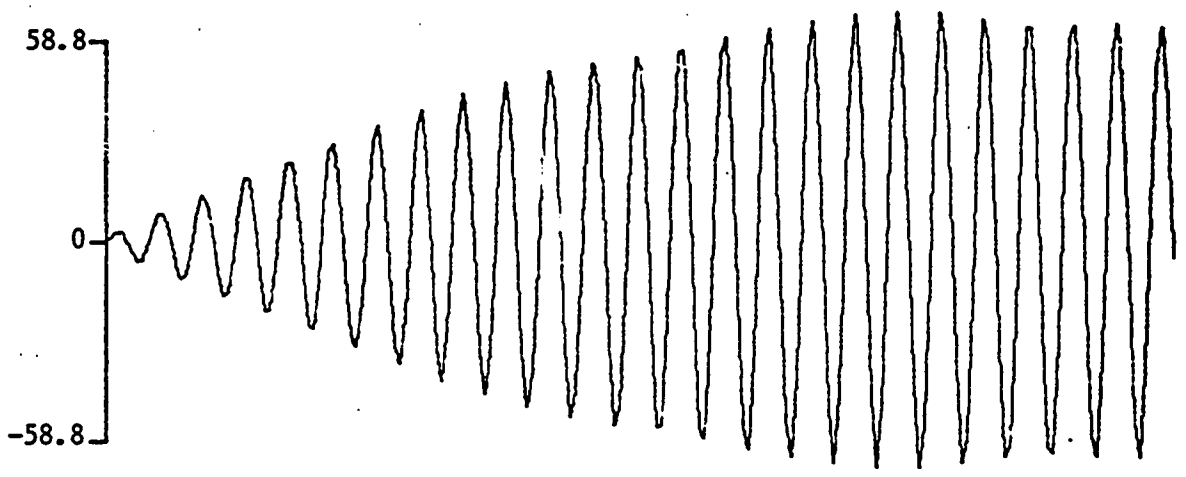


Long Floating Point Filter Output

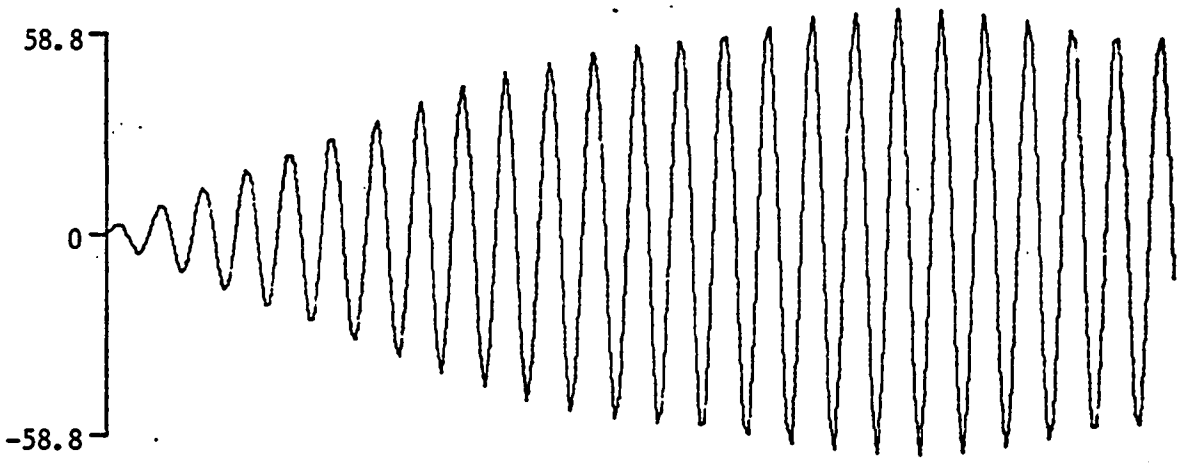


Floating Point Filter Output

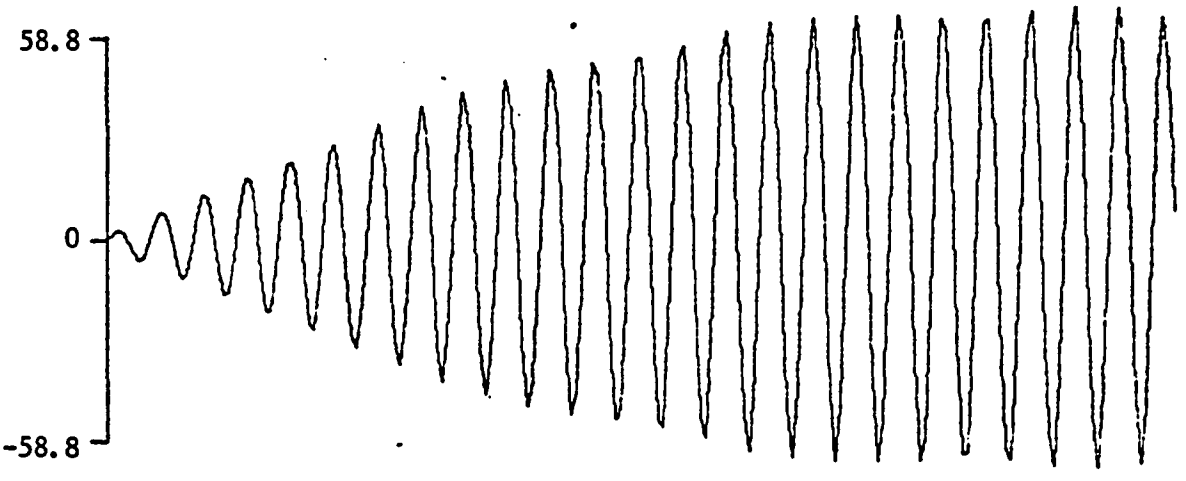
Fig 5.19



Logarithmic Filter Output

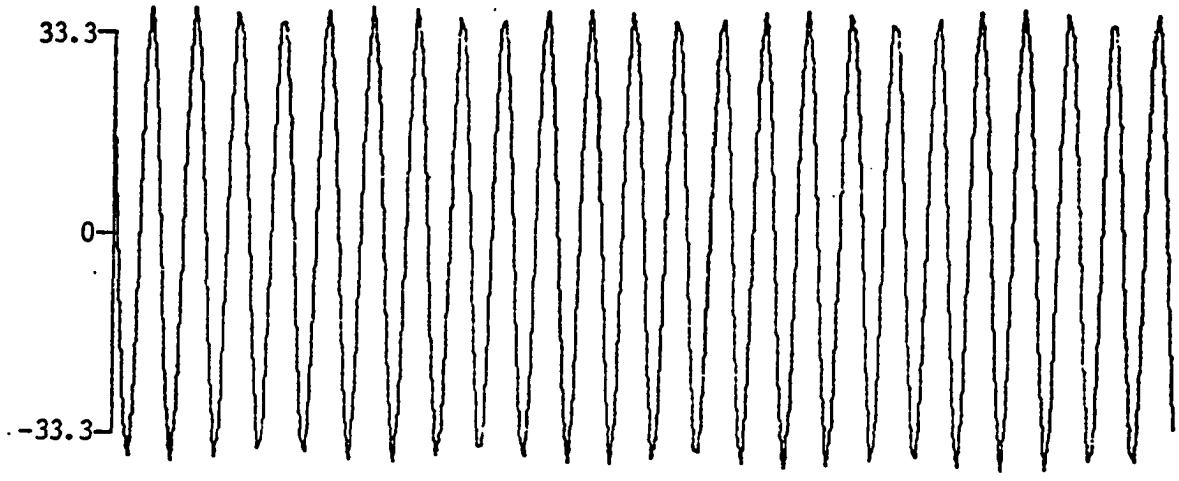


Long Floating Point Filter Output

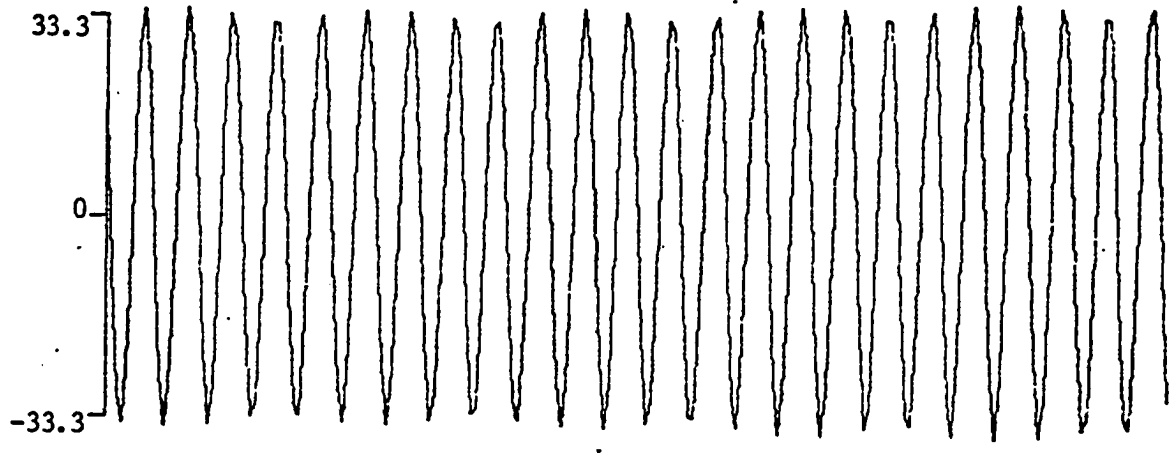


Floating Point Filter Output

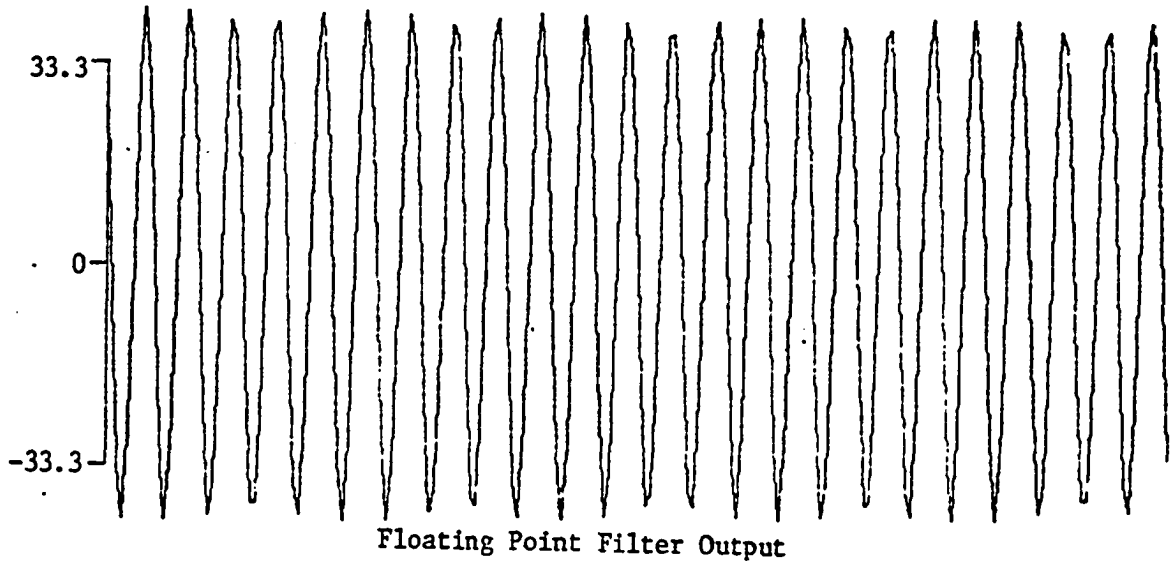
Fig 5.20



Logarithmic Filter Output



Long Floating Point Filter Output



Floating Point Filter Output

Fig 5.21

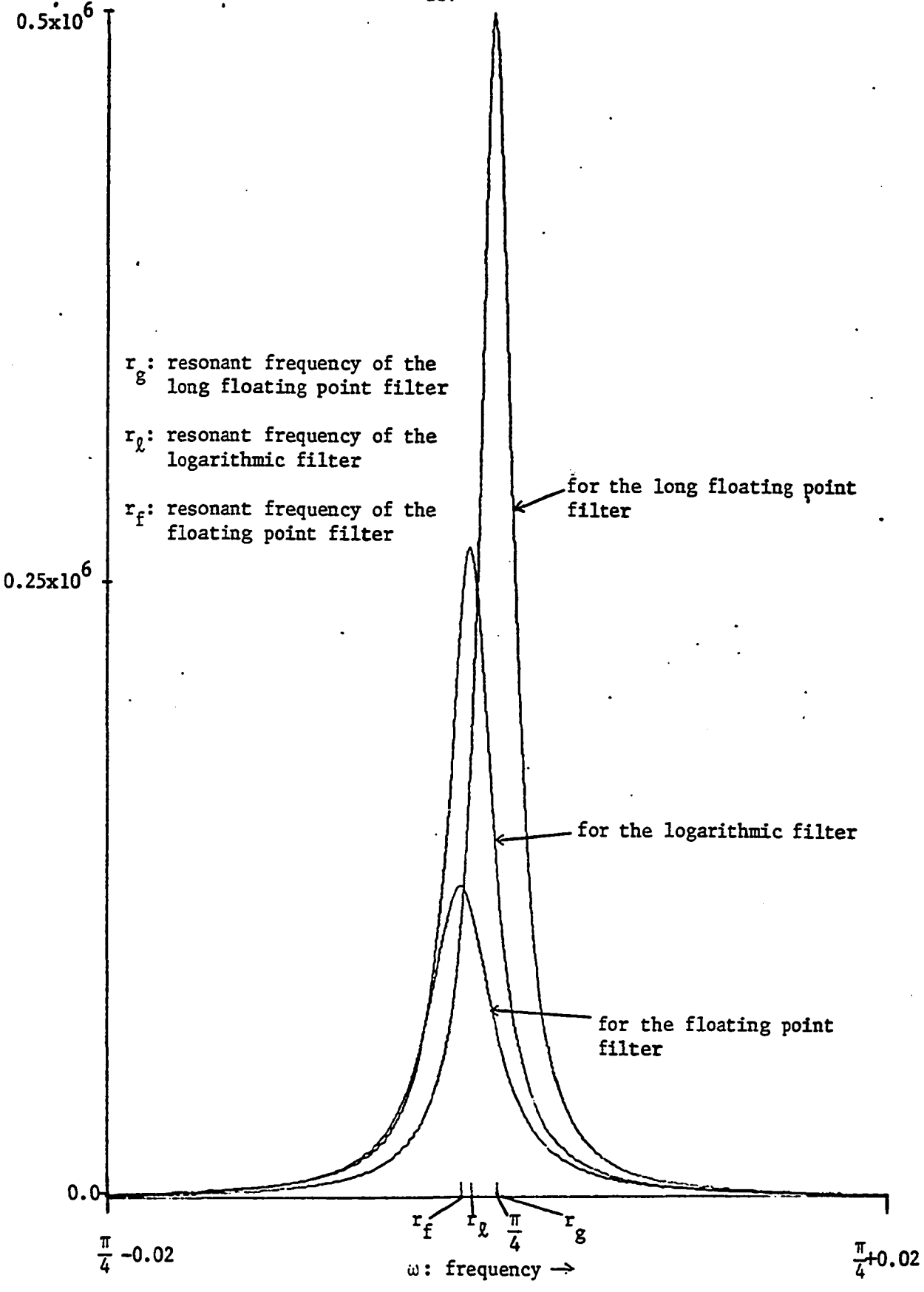
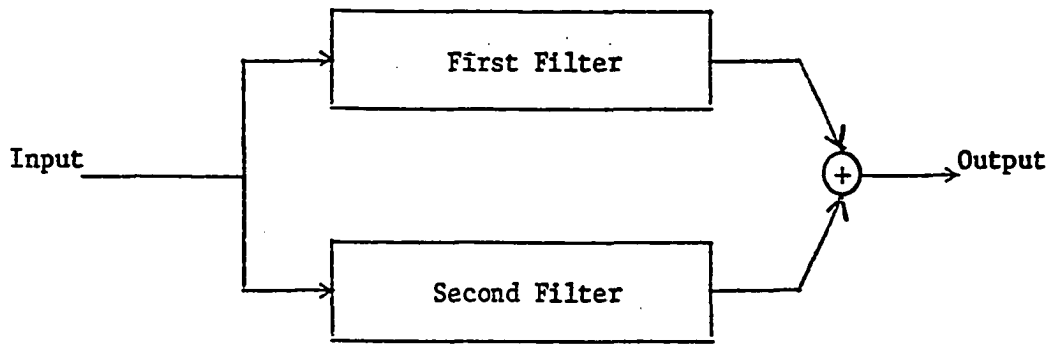


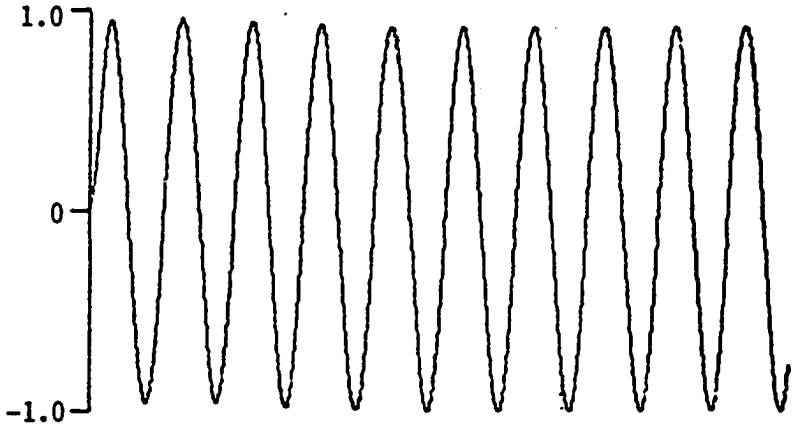
Fig 5.22 Squared Magnitude frequency response of the second order bandpass filter (Q = 393) of Table 5.6

(1) Cascade Form

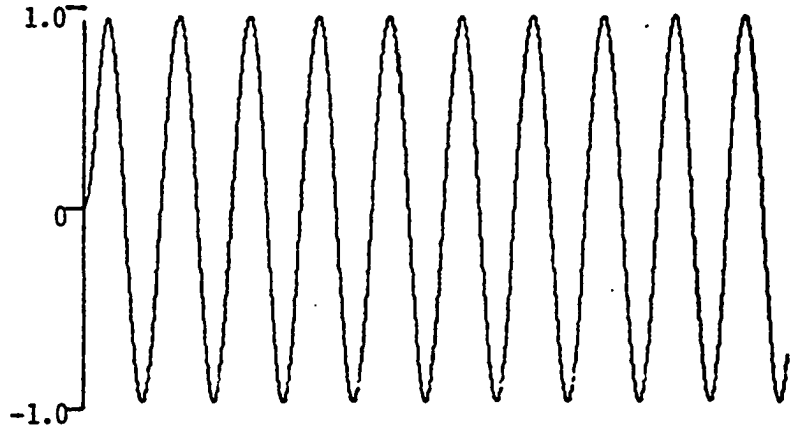


(2) Parallel Form

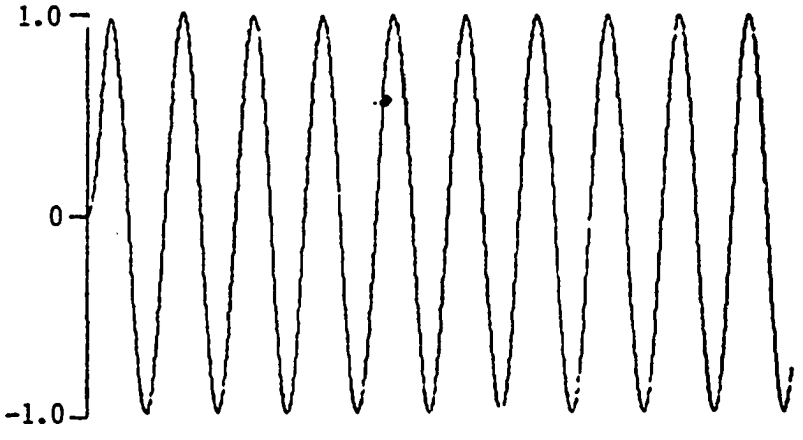
Fig 5.23 Cascade and Parallel Forms



Logarithmic Filter Output

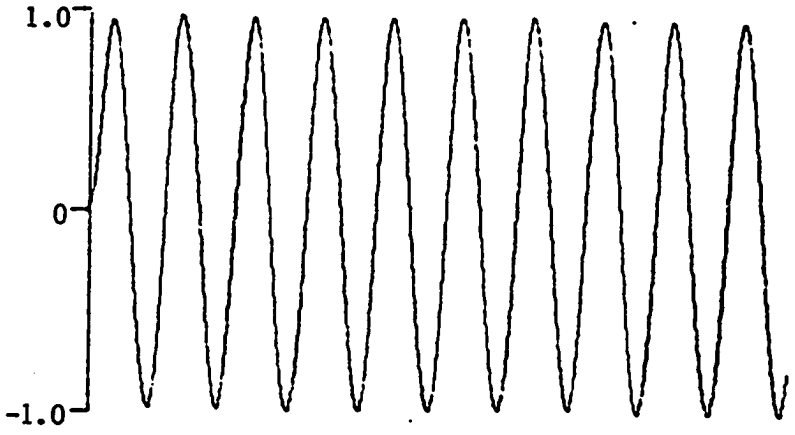


Long Floating Point Filter Output

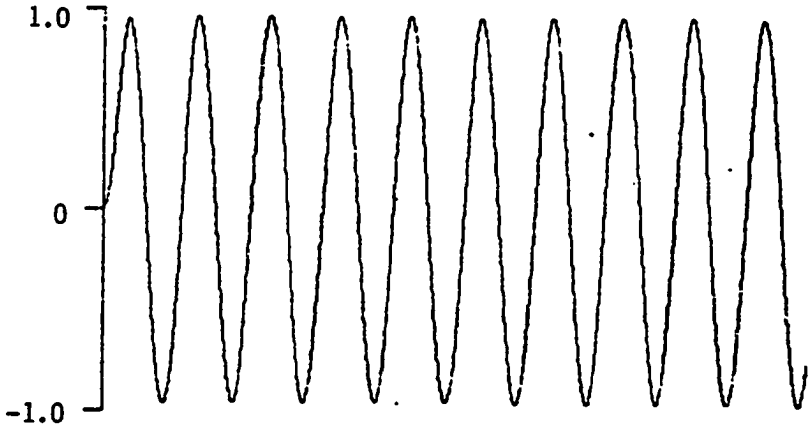


Floating Point Filter Output

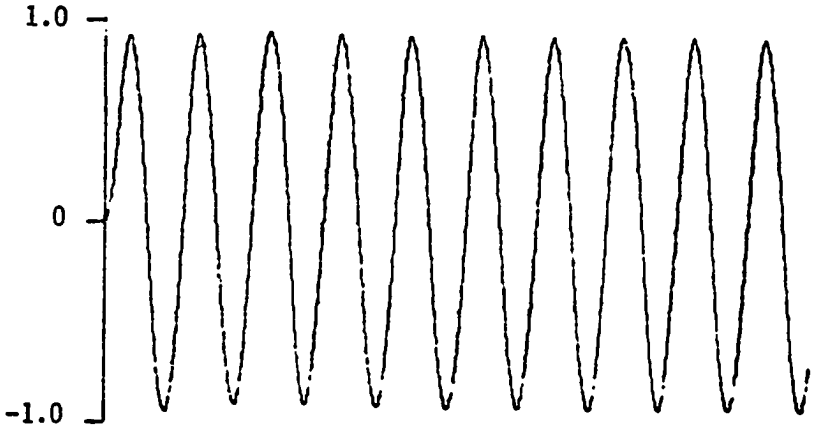
Fig 5.24



Logarithmic Filter Output

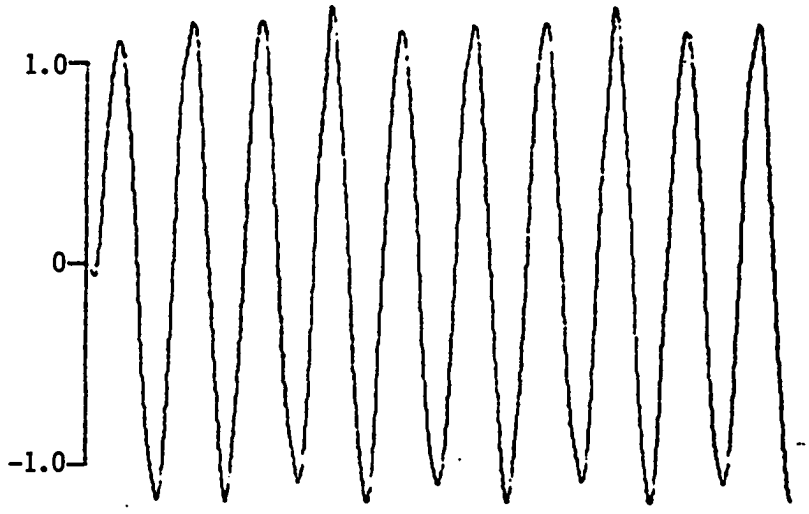


Long Floating Point Filter Output

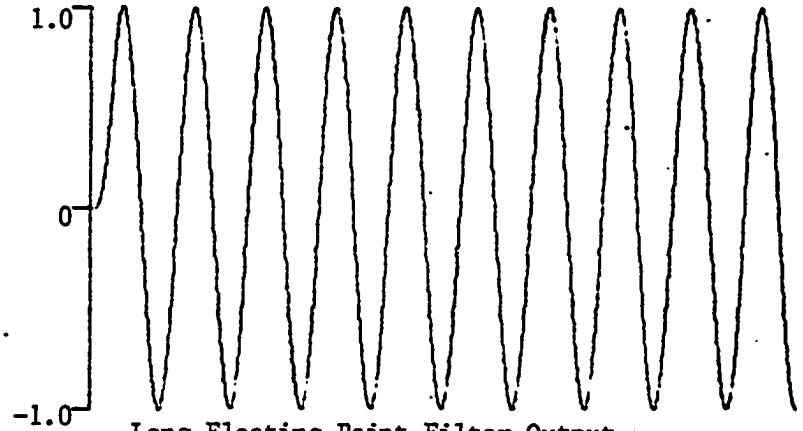


Floating Point Filter Output

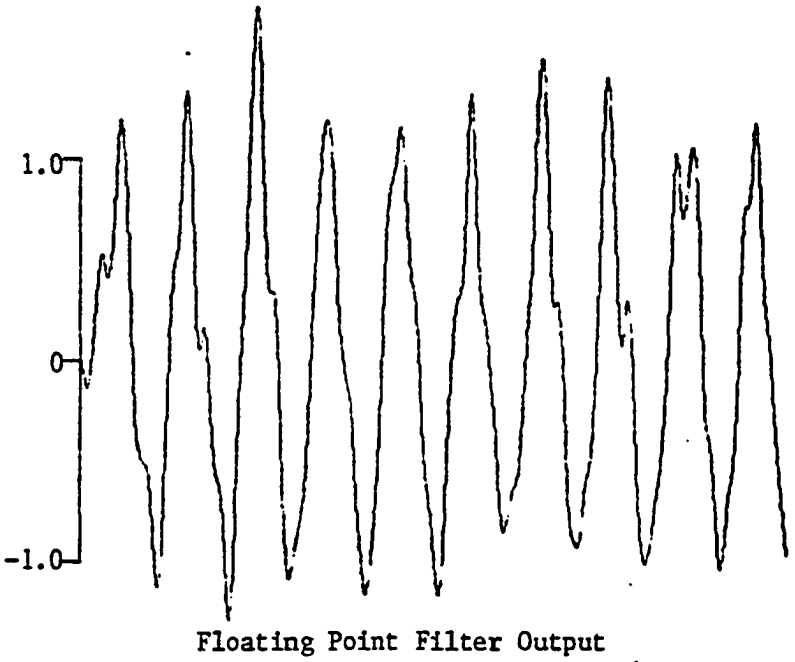
Fig 5.25



Logarithmic Filter Output



Long Floating Point Filter Output



Floating Point Filter Output

Fig 5.26

Filter coefficients	Order	Q	Cutoff Frequency	Input	Output	Error to Signal Ratio	N
Table 4.15 Bandpass	2	393	Lower $\pi/4-0.001$ Upper $\pi/4+0.001$	sin 0.785n + sin 0.1n	Fig 5.2 $1\leq n\leq 200$ Fig 5.3 $4001\leq n\leq 4200$ Fig 5.4 $7001\leq n\leq 7200$	L:1.98×10^{-1} F:4.18×10^{-1}	250
Table 5.2 Lowpass	2	1.4	0.58	sin 1.66n +sin 0.2n	Fig 5.5 $1\leq n\leq 400$	L:3.51×10^{-2} F:1.34×10^{-1}	250
Table 4.3 Lowpass	6	2.2	0.74	sin 1.66n + sin 0.2n	Fig 5.6 $1\leq n\leq 399$	L:1.68×10^{-1} F:2.11×10^{-1}	250
Table 5.3 Bandpass	8	23.4	Lower 1.402 Upper 1.706	sin 1.66n + sin 0.2n	Fig 5.7 $1\leq n\leq 100$ Fig 5.8 $101\leq n\leq 200$	L:5.54×10^{-2} F:7.12×10^{-2}	250
				Unit Sample	Fig 5.9 $1\leq n\leq 100$	L:2.71×10^{-1} F:5.26×10^{-1}	250

Note: L is for logarithmic filter of $\alpha = 6$, $\beta = 8$ and base = 2

F is for floating point filter of $f = 6$, $h = 8$

N is the number of output used for computation of error ratio

Q is defined in section 4.4

Table 5.1 Comparison of Logarithmic and Floating Point Filters

A(0)= 1.0000000000000000D 00
A(1)= -2.5064665409100000D 00
A(2)= 2.5387613839124500D 00
A(3)= -1.1924176783258200 00
A(4)= 2.1709534562567090-01
B(0)= 3.5607818936999980-03
B(1)= 1.4243127576507230-02
B(2)= 2.1364691363350580-02
B(3)= 1.4243127575588330-02
B(4)= 3.5607818939125030-03

A LOWPASS FILTER

Table 5.2

A(0)= 1.0000000000000000D 00
A(1)= -4.0402680635452270-01
A(2)= 2.0785207748413090 00
A(3)= -6.2616682052612300-01
A(4)= 1.4136648178100590 00
A(5)= -2.8846555948257450-01
A(6)= 3.1453907489776610-01
A(7)= -3.6032881587743760-02
A(8)= 4.3292239308357240-02
B(0)= 8.2425525653165570-03
B(1)= 3.7104173900015820-13
B(2)= -3.2970210261539810-02
B(3)= 3.8739064039949860-13
B(4)= 4.9455315391783530-02
B(5)= 7.7713009638546990-14
B(6)= -3.2970210261237170-02
B(7)= 2.0373459863609610-14
B(8)= 8.2425525653465690-03

A BANDPASS FILTER

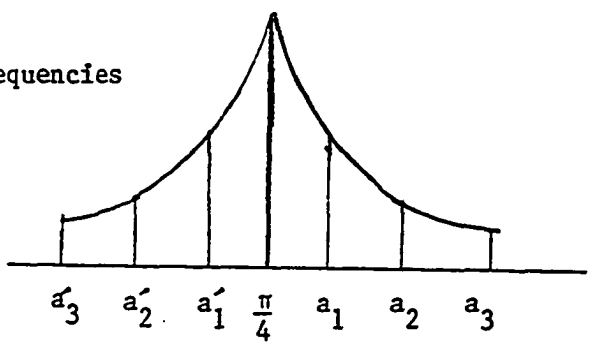
Table 5.3

Input	Output	Error to signal ratios	Number of Output used in ratio computation
sin $a_1 n$	Fig 5.10 ($1 \le n \le 200$) Fig 5.11 ($3401 \le n \le 3600$)	L: 0.578 F: 0.776	3450
sin $a_1' n$	Fig 5.12 ($1 \le n \le 200$) Fig 5.13 ($3401 \le n \le 3600$)	L: 0.837 (0.189) F: 0.647 (0.530)	3450
sin $a_2 n$	Fig 5.14 ($1 \le n \le 200$) Fig 5.15 ($3401 \le n \le 3600$)	L: 0.313 F: 0.485	3450
sin $a_2' n$	Fig 5.16 ($1 \le n \le 200$) Fig 5.17 ($3401 \le n \le 3600$	L: 0.480 F: 0.967	3450
sin $a_3 n$	Fig 5.18 ($1 \le n \le 200$) Fig 5.19 ($3401 \le n \le 3600$)	L: 0.251 F: 0.329	3450
sin $a_3' n$	Fig 5.20 ($1 \le n \le 200$) Fig 5.21 ($3401 \le n \le 3600$)	L; 0.263 F; 0.386	3450

Note: a_i and a_i' are as follows

$a_1 = \pi/4 + 0.001$
$a_1' = \pi/4 - 0.001$
} cutoff frequencies

$a_2 = \pi/4 + 0.005$
$a_2' = \pi/4 - 0.005$
$a_3 = \pi/4 + 0.02$
$a_3' = \pi/4 - 0.02$



parenthized error to signal ratio is the shifted version, L: for logarithmic filter; F: for floating point filter.

Table 5.4 Test of a high Q (=393) filter with several single frequency inputs

Input	Output		
	the logarithmic filter	the long floating point filter	the floating point filter
$\sin a_3 n$	≈ 33 from Fig 5.19	≈ 34 from Fig 5.19	≈ 30 from Fig 5.19
$\sin a_2 n$	≈100 from Fig 5.15	≈ 140 from Fig 5.15	≈ 80 from Fig 5.15
$\sin a_1 n$	≈ 250 from Fig 5.11	≈ 500 from Fig 5.11	≈ 125 from Fig 5.11
$\sin 0.785n$	≈ 447 from Fig 5.3	≈ 660 from Fig 5.3	≈ 200 from Fig 5.3
$\sin a'_1 n$	≈ 525 from Fig 5.13	≈ 500 from Fig 5.13	≈ 250 from Fig 5.13
$\sin a'_2 n$	≈ 180 from Fig 5.17	≈ 140 from Fig 5.17	≈ 210 from Fig 5.17
$\sin a'_3 n$	≈ 38 from Fig 5.21	≈ 34 from Fig 5.21	≈ 43 from Fig 5.21

Note: a_i and a'_i are given in Table 5.4

Table 5.5 Experimental Magnitude Response

```
A(  0)=    1.0000000000000000D 00
A(  1)=   -1.4142135623730950D 00
A(  2)=    9.9729605608547000-01
B(  0)=    1.0000000000000000D 00
```

The Logarithmic Filter Coefficients

```
A(  0)=    1.0000000000000000D 00
A(  1)=   -1.4127993488107220 00
A(  2)=    9.9800099999999980-01
B(  0)=    1.0000000000000000D 00
```

The Long Floating Point Filter Coefficients

```
A(  0)=    1.0000000000000000D 00
A(  1)=   -1.4140625000000000D 00
A(  2)=    9.9609375000000000-01
B(  0)=    1.0000000000000000D 00
```

The Floating Point Filter Coefficients

Table 5.6

Filter coefficients	Form	Cutoff Frequency	Input	Output	Error to signal ratio	N
Table 5.8 Lowpass	Cascade	0.650	$\sin(0.05\pi n)$ + $\sin(0.6\pi n)$	Fig 5.24 $1 \le n \le 200$	L:0.00736 F:0.0275	250
Table 5.9 Lowpass	Parallel	0.628	$\sin(0.05\pi n)$ + $\sin(0.6\pi n)$	Fig 5.25 $1 \le n \le 200$	L:0.0165 F:0.0582	250
Table 5.10 Lowpass	Parallel	0.580	$\sin(0.05\pi n)$ + $\sin(0.6\pi n)$	Fig 5.26 $1 \le n \le 200$	L:0.195 F:0.395	250

Note: L is for the logarithmic filter of $\alpha = 6$, $\beta = 8$ and base = 2;

F is for the floating point filter of $f = 6$ and $h = 8$;

N is the number of output used for computation of error ratio;

The ratio is computed in the same way as in section 5.1.

Table 5.7 Comparison of Logarithmic and Floating Point Filters

```
A1(  0)=    1.0000000000000000D 00
A1(  1)=   -1.4996000000000000D 00
A1(  2)=    8.4820000000000000D-01
B1(  0)=    4.2848569999999990D-02
B1(  1)=    8.5697140000000000D-02
B1(  2)=    4.2848569999999990D-02
```

FIRST FILTER

```
A2(  0)=    1.0000000000000000D 00
A2(  1)=   -1.5548000000000000D 00
A2(  2)=    6.4930000000000000D-01
B2(  0)=    4.2848569999999990D-02
B2(  1)=    8.5697140000000000D-02
B2(  2)=    4.2848569999999990D-02
```

SECOND FILTER

Table 5.8

```
A1(  0)=    1.0000000000000000D 00
A1(  1)=   -1.5658000000000000D 00
A1(  2)=    6.5490000000000000D-01
B1(  0)=    8.3270000000000000D-02
B1(  1)=    2.3900000000000000D-02
```

FIRST FILTER

```
A2(  0)=    1.0000000000000000D 00
A2(  1)=   -1.4934000000000000D 00
A2(  2)=    8.3920000000000000D-01
B2(  0)=   -8.3270000000000000D-02
B2(  1)=   -2.4600000000000000D-02
```

SECOND FILTER

Table 5.9

```
A1(  0)=    1.0000000000000000D 00
A1(  1)=   -2.8064379184800000D 00
A1(  2)=    3.3001038725798190 00
A1(  3)=   -1.8550118662687240 00
A1(  4)=    4.3314445468668630-01
B1(  0)=    8.6412423710000000-01
B1(  1)=   -2.4017364483117480 00
B1(  2)=    2.3467243449889860 00
B1(  3)=   -9.0388883878750220-01
```

FIRST FILTER

```
A2(  0)=    1.0000000000000000D 00
A2(  1)=   -2.2504320521100000 00
A2(  2)=    1.9467700170463740 00
A2(  3)=   -7.6471017709783410-01
A2(  4)=    1.1481330399870750-01
B2(  0)=   -8.6411121208862400-01
B2(  1)=    1.9214190571868820 00
B2(  2)=   -1.1895554277695530 00
B2(  3)=    2.3996254534211650-01
B2(  4)=    3.0070823792533060-05
```

SECOND FILTER

Table 5.10

APPENDIX 5.1

FADML and CVBF

1. FADML

AM = 'M' means multiplication

= 'A' means addition

Operation is done between X and Y, and the result is placed in Z.

XFE, YFE, and ZFE are the exponent of X, Y and Z respectively.

EFF, YFF, and ZFF are the fraction parts of X, Y and Z. The binary point is supposed to be after IBETAF bits from the right for each of XFF, YFF and ZFF.

IFBT2F : An integer which is 2^{h-2}

IBETAF: An integer which is h

IAL2F : An integer which is 2^{f}

IBT2F : An integer which is 2^{h}

HBT2F : An integer which is 2^{h-1}

IAL21F: An integer which is IAL2F-1

2. CVBF

A long floating point number in X is converted to the floating point number.

HPMINFF: A long floating point number which is $\frac{1}{2}2^{(-2^{h}-1)}$

BT2F : 2^{h}

IBT2F : an integer which is 2^{h}

IAL2F : an integer which is 2^{f}

NO : constant zero

N1 : constant one

N2 : constant two

```
/*
*/
   FADML: PROC(X,Y,Z,AM);
/*
THIS PROCEDURE DOES ADDITION AND MULTIPLICATION IN THE
FLOATING POINT NUMBER SYSTEM TO BE TESTED
*/
 DCL (P,PI,P2) BIN FLOAT(53);
    DCL (R1,R2,R3,R4,R5) BIN FLOAT(53);
      DCL 1 X,
         2 (XFE,XFF) BIN FIXED(31);
      DCL 1 Y,
         2 (YFE,YFF) BIN FIXED(31);
      DCL 1 Z,
         2 (ZFE,ZFF) BIN FIXED(31);
      DCL AM CHAR(1);
      DCL AD BIN FIXED(31,5);
    DCL (XE,XF,YE,YF,ZE,ZF,AF) BIN FIXED(31,5);
      XE=XFE;
      XF=XFF;
      YE=YFE;
      YF=YFF;
      IF AM='M'
       THEN DO;
              ZE=XFE+YFE;
              ZF=XFF*YFF/IFBT2F;
              ZE=ZE-2;
            END;
       ELSE
          DO;
             IF XE=YE
                THEN DO;
                       ZE=XE;
                       ZF=XF+YF;
                       END;
                ELSE
                   DO;
             AD=ABS(XE-YE);
                     IF XE>YE
                        THEN
                           DO;
                           ZE=XE;
                           IF AD>IBETAF
                              THEN ZF=XF;
                              ELSE ZF=XF+YF/(2**AD);
                           END;
                        ELSE
                           DO;
                           ZE=YE;
                           IF AD>IBETAF
                              THEN ZF=YF;
                              ELSE ZF=YF+XF/(2**AD);
                           END;
                     END;
```

```
            END;
         IF ZF=0
            THEN ZE=-IAL2F;
            ELSE
               DO;
                  AF=ABS(ZF);
                  IF AF>=IBT2F
                     THEN
                        DO WHILE(AF>=IBT2F);
                        AF=AF/2;
                        ZE=ZE+1;
                        END;
                     ELSE
                        DO WHILE(AF<HBT2F);
                        AF=AF*2;
                        ZE=ZE-1;
                        END;
                  IF ZF>0
                     THEN ZF=AF;
                     ELSE ZF=-AF;
               END;
         ZFE=ZE;
         ZFF=ROUND(ZF,0);
         IF ABS(ZFF)=IBT2F
            THEN
               DO;
               ZFF=ZFF/2;
               ZFE=ZFE+1;
               END;
       IF ZFE>IAL21F
            THEN
               DO;
               ZFE=IAL21F;
               ZFF=IBT2F-1;
               END;
         IF ZFE<=-IAL2F-1
            THEN
               DO;
               ZFE=-IAL2F;
               ZFF=0;
               END;
   END FADML;
/*
*/
   CVBF: PROC(X,Y);
/* THIS PROCEDURE CONVERTS A LONG FLOATING POINT NUMBER TO THE
PARTICULAR FLOATING POINT NUMBER TO BE TESTED
*/
      DCL X BIN FLOAT(53);
      DCL 1 Y,
         2 (YFE,YFF) BIN FIXED(31);
      DCL (AX,E,IE,F) BIN FLOAT(53);
      AX=ABS(X);
      IF AX>=HPMINFF
```

```
         THEN
            DO;
               E=LOG2(AX);
               IE=FLOOR(E+N1);
               F=N2**(E-IE);
               YFE=FIXED(IE,31,0);
               YFF=ROUND(FIXED(BT2F*F,31,1),0);
               IF ABS(YFF)=IBT2F
                  THEN
                     DO;
                     YFF=YFF/2;
                     YFE=YFE+1;
                     END;
               IF X<N0
                  THEN YFF=-YFF;
            END;
         ELSE
            DO;
               YFE=-IAL2F;
               YFF=0;
            END;
   END CVBF;
/*
*/
```

CHAPTER VI

COMMENTS TO FILTER DESIGN

6.1 Relation between base and bit assignment

A logarithmic number system of a digital machine is completely defined if the following are determined:

1) base: a

2) word length: ℓ

3) bit assignment: α or β

For a constant word length, a and β determine all the numbers in the system.

$$a^{2^{-\beta}} = b^{2^{-(\beta-\log_2(\log_b a))}} \tag{6.1}$$

Then, the number system with base = a and fractional part of β bits is equivalent to the system with base = b and the fractional part of

$$\beta - k$$

where $k = \log_2(\log_b a)$ (6.2)

In a machine k has to be an integer. Then, by the above equivalence, word length ℓ, fractional part β, and the base of a by

$$1 < \sqrt{b} < a \leq b$$

completely determine the number system. Or, word length ℓ and the base $a > 1$ completely determine the system if β is fixed.

6.2 Steps for filter design

An approach for the design of a logarithmic filter (given coefficients) concerning errors includes the consideration of the following:

1) Parameters of logarithmic number system:

   base: a

   word length: ℓ

   fractional part length: β

2) memory size

3) speed

4) input sequence $\{x_n\}$

For 1) parameters, the system is required to have

1. |intermediate value of the filter| < maximum representable number, or range > constant; and

2. $\sqrt{\frac{E[e_n^2]}{E[w_n^2]}} < \text{constant}$ or $\sqrt{\frac{e_n^2}{w_n^2}}\Bigg|_{max} < \text{constant}$

Let us assume that the range is required to be more than constant c_1. Then

$$\text{Range} = a^{2^{\alpha+1}-2^{-\beta}}$$

Assuming $\beta > 0$ and having some margin

$$\text{Range} \geq a^{2^{\alpha}} > c_1 \quad (c_1 \text{ constant}) \tag{6.3}$$

The range line $a^{2^{\alpha}} = c_1$ is depicted in Fig. 6.1.

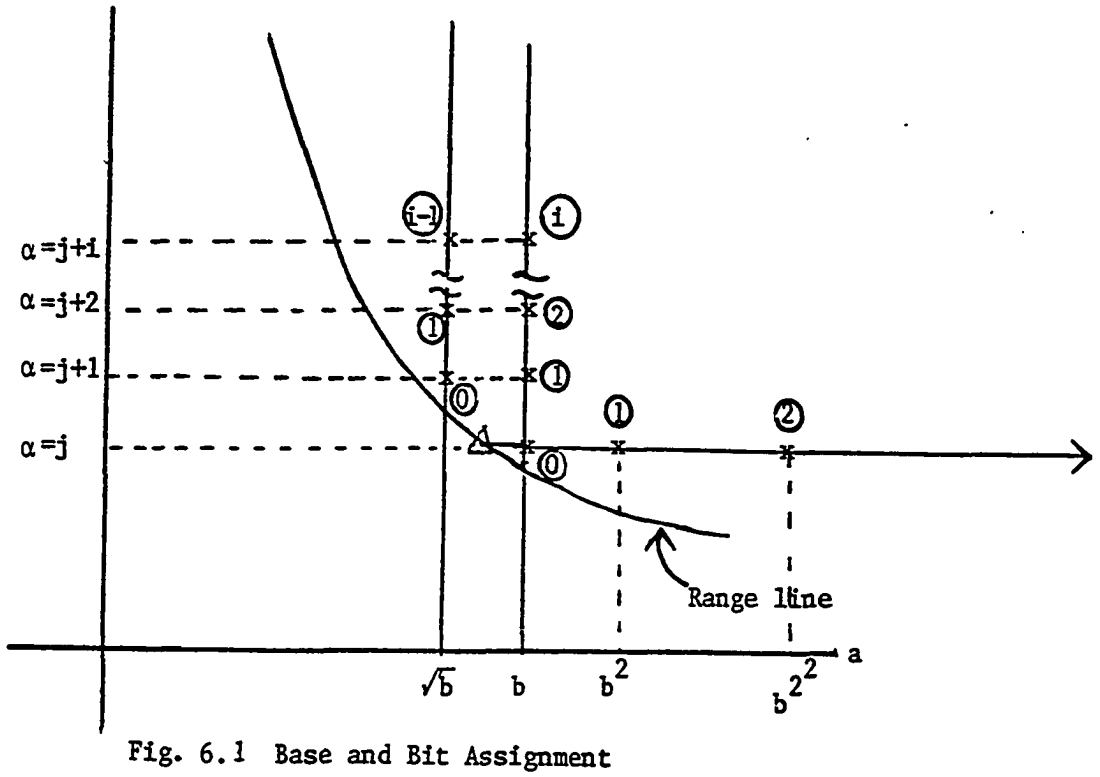


Fig. 6.1 Base and Bit Assignment

Above the range line is the acceptable region. Intersections of α = an integer and the range line give the optimal solutions for the range. The theoretical error to signal ratio given by the equation (4.1) is restated below:

$$\sqrt{\frac{E[e_n^2]}{E[w_n^2]}} = \sqrt{\frac{q^2 s_1 s_3 + m_e^2 s_4}{s_2}}$$

$$\approx \frac{a^{2^{-\beta-1}} - a^{-2^{-\beta-1}}}{2\sqrt{3}} \sqrt{\frac{s_1 s_3}{s_2}} \quad \text{(for rounding)} \quad (6.4)$$

In this section, the error to signal ratio means only that of rounding. $\sqrt{s_1 s_3 / s_2}$ is independent of the number system and $\Phi_{xx}(e^{j\omega})$ is not necessarily a constant one. If word length $\ell = \alpha + \beta + 2$ is fixed, as α or a increases, the ratio increases. In other words, as a increases (α is fixed), the ratio increases. So for a wide sense stationary input sequence $\{x_n\}$, the range line in Fig. 6.1 gives the smallest theoretical error to signal ratio. Since the theoretical ratios agreed with experimental ratios for not too small base a with fixed α and ℓ, there should be an optimal base a for the ratio within a range and with fixed α and ℓ. The conceptual relation of base a, the range, and the ratio is illustrated in Fig. 6.2

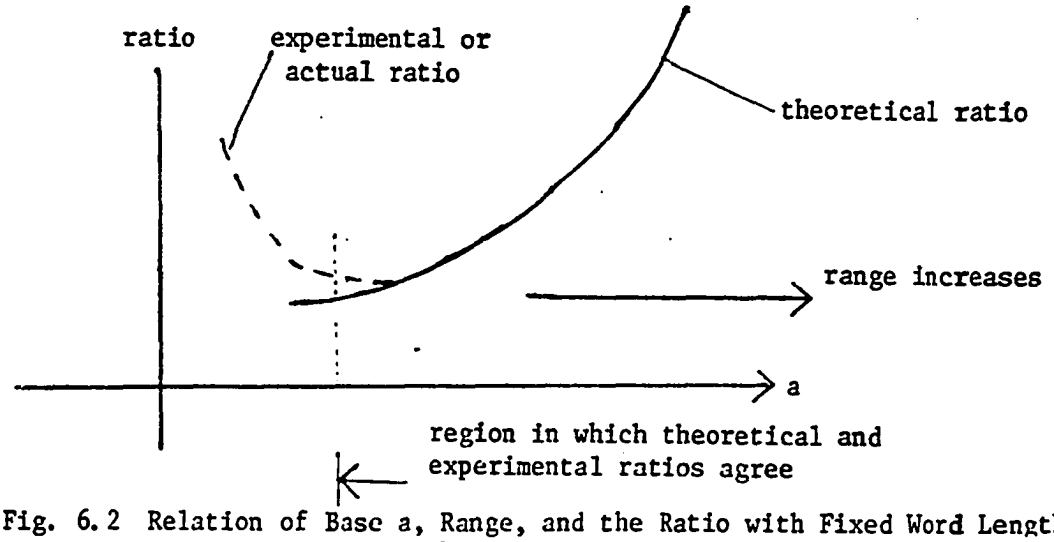


Fig. 6.2 Relation of Base a, Range, and the Ratio with Fixed Word Length ℓ and fractional part β.

The problem can be considered to minimize the ratio under the conditions:

$$a^{2^{\alpha}} > c_1 \qquad \text{(for range)}$$

$$a^{2^{\alpha}} > c_2 \qquad \text{(for agreement between theoretical and experimental ratios)}$$

Memory size of look-up table for logarithmic addition for 2 byte word case is given by

$$F + G + F' + G'$$

where

$$F = \text{Truncate}[-2^{\beta} \cdot \log_a (a^{(M-0.5)2^{-\beta}}) + 1]$$

$$G = \text{Truncate}[-2^{\beta} \cdot \log_a (1-a^{-(M-0.5)2^{-\beta}}) + 1]$$

$$F' = \text{Truncate}[-2^{\beta} \log_a (a^{2^{\ell-\beta}-2^{-\beta-1}} - 1) + 1]$$

$$\text{if } F' < 0 \quad \text{Then } F' = 0$$

$$G' = \text{Truncate}[-2^{\beta} \log_a (1-a^{-2^{\ell-\beta+2^{-\beta-1}}}) + 1]$$

Note: ℓ is the number of bits in a byte: usually 8.

Partial table for memory size with fixed ℓ and α is given in Table 6.1. M in the table is the speed parameter of logarithmic addition and Q is $|q|$ in the equation (6.4). When M increases the speed is lowered (see [6]), and when a increases, the memory size decreases.

An example of finding the base a with fixed word length ℓ and fractional part β under some restriction is given below:

Example: Restrictions are

$$1) \quad q = \frac{a^{2^{-8}} - a^{-2^{-8}}}{2\sqrt{3}} \leq 0.2 \times 10^{-2} \qquad \text{(for ratio)}$$

Note: $\sqrt{s_1 s_3 / s_2}$ can be computed for a wide sense stationary input sequence. $\Phi_{xx}(e^{j\omega})$ is not necessarily constant.

2) $a^{2^7} \geq 2^{2^7}$ (for range)

3) $a^{2^7} \geq 2^{2^4}$ (for theoretical and experimental ratios' agreement)

4) storage $\leq$ 2000 bytes

5) $M \leq 2$ (for speed)

Note: $\beta = 7$; $\alpha = 7$; $\ell = 16$

The restriction 2) means $a > 2$; the restriction 3) means $a > 1.09$. Then by the Table 6.1 a is restricted by

$$2.146 \leq a \leq 2.405 \quad \text{for } M = 1$$

$$? \leq a \leq 2.405 \quad \text{for } M = 2$$

To find α or β with fixed a and ℓ is another way of determining the number system. But it is not flexible because α and β are integers.

As long as logarithmic number system is concerned, the existing programs should be considered for the filter design. FOCUS.8, FOCUS.16 and FOCUS.10 [12] correspond to the cases of:

1) $\alpha = 3$, $\beta = 3$ and base = 2

2) $\alpha = 5$, $\beta = 9$ and base = 10

3) $\alpha = 4$, $\beta = 10$ and base = 10 respectively.

The experimental program of Appendix 3.4 can be used to check if the existing programs can meet a specific digital filter requirement. In case of general input sequence $\{x_n\}$ which may not have the zero mean wide sense stationarity, the procedure UTP has to be changed so that the input sequence resembles the real situation.

Since a logarithmic number system has a gap around zero, a very small magnitude input causes an accuracy problem. Note that a floating point number system also has the same sort of discontinuity problem around zero

although it has zero. The above implies that in a number system, floating or logarithmic, there are lower and upper magnitude bounds so that the error (quantization) is proportional to the magnitude of the signal. The largest and smallest representable magnitudes in the logarithmic number system are:

$$\text{Largest magnitude} = a^{2^{\alpha}-2^{-\beta}}$$

$$\text{Smallest magnitude} = a^{-2^{\alpha}}$$

and

$$\log_a \left(\frac{\text{largest magnitude}}{a^0} \right) = 2^{\alpha} - 2^{-\beta} \approx 2^{\alpha}$$

$$\log_a \left(\frac{a^0}{\text{smallest magnitude}} \right) = 2^{\alpha}$$

The above implies that there are equal ranges and equal number of representable numbers below one and above one in magnitude in the logarithmic number system. Then an analog input signal should be multiplied by a constant before digitization so that the mean of the logarithms of the absolute values of the signals becomes close to zero.

The design scheme so far described only includes the accumulated roundoff errors. Since input quantization and coefficient quantization errors cannot be ignored, they should be included in the design procedure. The theoretical analysis of those two errors are not done in this dissertation. If, however, those two errors have the same property as the accumulated roundoff error shown in Fig 6.2, or if overall error to signal ratio (of accumulated roundoff, input quantization, and coefficient quantization errors together) has a same sort of curve as in Fig 6.2, then the error to signal ratio stated in this chapter can be replaced by the overall error to signal ratio. The inclusion of the input quantization and coefficient quantization errors in the design scheme is left for future work.

N= BASE	1	2	3	4	5	6	7	8	0
2.000	2238	1833	1644	1520	1427	1353	1291	1238	0.156D-02
2.019	2201	1801	1615	1492	1401	1328	1267	1215	0.159D-02
2.039	2165	1770	1587	1466	1375	1303	1243	1192	0.161D-02
2.060	2129	1739	1558	1439	1350	1279	1220	1169	0.163D-02
2.081	2094	1710	1532	1414	1326	1256	1198	1148	0.165D-02
2.102	2059	1681	1505	1389	1302	1233	1175	1126	0.168D-02
2.124	2025	1651	1478	1364	1278	1210	1153	1104	0.170D-02
2.146	1992	1624	1452	1340	1255	1188	1132	1084	0.172D-02
2.169	1959	1595	1426	1315	1232	1166	1111	1063	0.175D-02
2.193	1927	1568	1402	1292	1210	1145	1090	1044	0.177D-02
2.217	1894	1541	1377	1269	1188	1123	1069	1023	0.180D-02
2.242	1863	1515	1353	1246	1166	1103	1050	1004	0.182D-02
2.267	1832	1488	1328	1223	1145	1082	1030	985	0.185D-02
2.293	1802	1463	1305	1202	1124	1062	1011	966	0.187D-02
2.320	1772	1438	1283	1180	1104	1043	992	948	0.190D-02
2.348	1743	1413	1260	1159	1084	1024	974	931	0.192D-02
2.376	1713	1388	1237	1138	1063	1004	955	912	0.195D-02
2.405	1685	1365	1216	1117	1044	985	937	895	0.198D-02
2.434	1657	1341	1194	1097	1025	967	919	878	0.201D-02
2.465	1630	1318	1173	1078	1006	949	902	861	0.203D-02
2.496	1603	1296	1153	1058	988	932	885	845	0.206D-02
2.528	1577	1273	1132	1039	970	915	868	829	0.209D-02
2.561	1550	1251	1112	1021	952	898	852	813	0.212D-02
2.595	1525	1230	1093	1002	935	881	836	798	0.215D-02
2.629	1499	1208	1073	984	917	864	820	782	0.218D-02
2.665	1474	1187	1054	966	901	848	804	767	0.221D-02
2.702	1451	1168	1036	950	885	833	790	753	0.224D-02
2.740	1427	1148	1018	932	868	817	775	739	0.227D-02
2.778	1403	1128	1000	915	852	802	760	724	0.230D-02
2.818	1379	1108	982	898	836	787	746	710	0.234D-02
2.859	1356	1088	964	882	821	772	731	696	0.237D-02
2.902	1334	1070	948	867	806	758	718	684	0.240D-02
2.945	1312	1052	931	851	791	744	704	670	0.244D-02
2.990	1290	1033	914	835	776	729	690	657	0.247D-02
3.036	1269	1016	898	821	763	716	678	645	0.250D-02
3.083	1248	998	882	805	748	703	665	632	0.254D-02
3.132	1228	981	867	791	735	690	652	620	0.257D-02
3.183	1207	964	851	777	721	677	640	608	0.261D-02
3.234	1187	948	836	763	708	664	628	597	0.265D-02
3.288	1167	931	821	749	695	651	616	585	0.268D-02
3.343	1147	914	806	735	681	639	603	573	0.272D-02
3.400	1129	899	792	722	669	627	592	562	0.276D-02
3.458	1111	884	779	709	657	616	581	552	0.280D-02
3.519	1092	868	764	696	645	604	570	541	0.284D-02
3.581	1074	853	751	683	633	593	559	530	0.288D-02
3.645	1056	838	737	670	621	581	548	520	0.292D-02
3.712	1038	824	724	659	610	570	538	510	0.296D-02
3.780	1022	810	712	647	599	560	528	500	0.300D-02
3.851	1004	795	698	635	587	549	517	490	0.304D-02
3.924	988	782	686	623	576	539	507	481	0.308D-02
4.000	971	768	673	611	565	528	497	471	0.313D-02

Table 6.1 REQUIRED STORAGE (BYTE) FOR
WORD LENGTH=16 BITS
FRACTIONAL PART=7 BITS

CHAPTER VII

CONCLUSION AND FUTURE WORK

It is shown theoretically and experimentally that a logarithmic addition error is the sum multiplied by a random variable which has an approximate uniformity of probability distribution. Formula is derived for logarithmic filter's accumulated roundoff errors (rounding and truncation) for the case of stochastic input. The error ratio to the output is expressed in terms of the filter coefficients and the input spectrum. Good agreement is made between the theory and the experiments except the extreme cases in which the logarithmic number system has too wide gap around zero. If the gap is wide, underflow occurs often. The accumulated roundoff error comparison is made theoretically between floating point filters and logarithmic filters. It is shown that the errors of the logarithmic filters are much smaller. Several filters are experimentally tested for deterministic inputs for overall error computations (coefficients quantization, input quantization, and accumulated roundoff errors together) for the comparison between logarithmic filters and the floating point filters, given the same number of bits of a word and equal ranges for both of the number systems. The results are all in favor of the logarithmic filters. A logarithmic filter design procedure is given by choosing a particular logarithmic number system while it satisfies the requirements of the accumulated roundoff error ratio, the memory utilization, the speed, and the range.

As the overall filter error of coefficient quantization, input quantization, and accumulated roundoff errors together is computed in Chapter V, the theoretical analysis should be made for each of the input quantization and the coefficient quantization for future work. And it should be included in the filter design procedure of Chapter VI.

REFERENCES

[1] L. R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing. Englewood Cliffs, N.J.: Prentice-Hall 1975.

[2] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing. Englewood Cliffs, N. J.: Prentice-Hall 1975.

[3] B. Liu and T. Kaneko, "Error Analysis of Digital Filters Realized with Floating-Point Arithmetic", Proc. IEEE Vol 57, Oct 1969, pp 1735-1747.

[4] I. W. Sandberg, "Floating-Point Roundoff Accumulation in Digital Filter Realization", Bell Sys. Tech J., 46, 1775-1791, Oct. 1967.

[5] J. H. Wilkinson, Rounding Errors in Algebraic Processes. Englewood Cliffs, N.J.: Prentice-Hall, 1963.

[6] S. C. Lee and A. D. Edgar "Focus Microcomputer Number System", Microcomputer Design and Applications, 1-40, NY. SF. London: Academic Press 1977.

[7] N. G. Kingsbury and P. J. W. Rayner "Digital Filtering Using Logarithmic Arithmetic", Electronic Letters Vol 7 No 2 28th Jan. 1971 pp 56-58.

[8] K. J. Dean "Binary Logarithms", Electron Engng., 1968, 40, pp 560-562.

[9] J. N. Mitchell "Computer Multiplication and Division Using Binary Logarithms" I.R.E. Trans. on Elect. Computers EC-11 512 (1962).

[10] Robert V. Hogg and Allen T. Craig, "Introduction to Mathematical Statistics" 3rd Edition, The MacMillan Company 1970.

[11] G. S. Fishman, "Concepts and Methods in Discrete Event Digital Simulation" John Wiley & Sons, Inc. 1973.

[12] A. D. Edgar, "Micro-optical Tomography", Ph.D. Thesis, University of Oklahoma, 1978.

[13] P. M. Sherman, "Programming and Coding the IBM 709-7090-7094 Computers" John Wiley & Sons, Inc. 1963.

[14] R. Genesio, A. Laurentini, V. Mauro, and A.R. Meo, Butterworth and Chebyschev Digital Filters: Tables For Their Design, Elsevier Scientific Publishing Company, New York 1973.