UNIVERSITY OF TORONTO

DEPARTMENT OF ELECTRICAL ENGINEERING

COURSE 1509X

DIGITAL SIGNAL PROCESSING

INSTRUCTOR: A. N. VENETSANOPOULOS

TERM ASSIGNMENT

THE PROBLEM OF ROUND OFF NOISE IN DIGITAL FILTERS

STUDENT: J. M. COSTA

JANUARY 6, 1972

THE PROBLEM OF ROUND OFF NOISE IN DIGITAL FILTERS

0. ABSTRACT

Digital filters can be realized using fixed-point, floating-point, or block-floating-point arithmetic. In this paper we consider the round off noise problem with regard to the mode of arithmetic used to implement the digital filter. We present structures for fixed-point, floating-point and block-floating-point arithmetic, compare these techniques on the basis of their output noise-to-signal ratio, and plot curves representing this comparison.

= ♀ = ♀ = ♀ = ♀ = ♀ =

1. INTRODUCTION

In spite of the advantages offered by digital networks, there is an inherent accuracy problem associated with the implementation and operation of digital filters, since each number is represented by a finite number of bits and the arithmetic operations must be carried out with an accuracy limited by this finite word length. The specific sources of quantization error are as follows:

1) Coefficient truncation -caused by the fact that the multiplying constants must be quantized to some finite number of bits [1], [2], [3].

2) Input quantization -caused by the quantization of the input signal into a set of discrete levels [4], [5], [6].

3) Round off noise -caused by the accumulation of errors committed at each arithmetic operation because these operations are carried out with only finite bit accuracy.

In this paper we consider only the third error source (that is, the round off noise), and present an approach to

the analysis of this problem from the point of view of the mode of arithmetic employed, that is, fixed-point [7], [8], [9], [10], [11], [12], [13], floating-point [10], [14], [15], [16], or block-floating-point [17].

All the contents of this paper have been covered in the given references. The purpose of this paper is to give a unified presentation of the effect of the mode of arithmetic on the output round off noise.

In Section 2 we consider the round off noise problem and the variables in the filter implementation which determine the level and character of the round off noise. A structure for fixed-point, floating-point and block-floating-point is presented in Section 3. In Section 4 we compare these modes of arithmetic on the basis of their output noise-to-signal ratio, and present curves representing this comparison.

= ♀ = ♀ = ♀ = ♀ = ♀ =

2. THE PROBLEM OF ROUND OFF NOISE IN DIGITAL FILTERS

Round off noise as well as input quantization may be considered as noise introducing processes, very similar in nature since both involve quantization of the data, but the former differs in two respects: 1) The data to be quantized is already digital in form, and 2) the rounding or truncation of the data takes place at various points within the filter, not just at its input.

The content and complexity of any analysis of round off noise are determined to a large extent by the assumed correlation between round off errors. The analyses appearing in the literature concerning round off noise in digital filters usually employ the simplifying and often reasonable assumption of uncorrelated round off errors from sample to sample and from one error source (multi-

plier or other rounding point) to another [7], [8], [15]. The advantage of assuming uncorrelated errors from one sample to another is that the noise injected into the filter by each rounding operation is then "white" [4]; while the advantage of assuming uncorrelated error sources is that the output noise power spectrum may then be computed as simply the superposition of the (filtered) noise spectra due to the separate error sources [7], [11]. As long as the assumption of uncorrelated errors can be made, the results of an analysis of round off noise are applicable to the case of truncation as well as rounding, with the error variance for truncation being four times that for rounding. However, as the input signals become less "random", the uncorrelated-error assumption tends to break down for truncation more readily than for rounding. In the case of correlated-noise (for instance, when the signal is constant) we encounter the deadband effect and can have limit cycles, [9], [18], [19], [20]. Nevertheless round off limit cycles are negligible with floating-point arithmetic [14].

There are three variables in the filter implementation which determine the level and character of the round off noise for a given input signal:

1) The number of bits used to represent the data within the filter.

2) The mode of arithmetic employed.

3) The circuit configuration of the digital filter.

The number of bits in the data may be thought of as determining either the quantization step size or the dynamic range of the filter (that is, a maximum value or set of maximum values for the magnitudes of these data). If we choose the latter interpretation we will have the same step size for all filters, and therefore, the number of data bits will not affect the level of the round off noise directly, but rather it will limit the maximum

allowable signal level and hence the realizable signal-to--noise ratio. Data within the filter must, of course, be properly "scaled" if the maximum signal-to-noise ratio is to be maintained without exceeding the dynamic range limitations.

There are three modes of arithmetic which can be employed in the implementation of a digital filter: fixed--point, floating-point and block-floating-point. Since the analysis of round off noise in this paper is from this point of view we study this topic in detail in the two following sections.

The third variable in the implementation of a digital filter, that of circuit configuration, [9], [12], [13], [16], is the principal factor determining the character (spectrum) of the output round off noise and, along with mode of the arithmetic, ultimately determines the number of data digits required to satisfy the performance specifications. There are a multitude of equivalent circuit configurations for any given linear discrete filter (whose transfer function is expressible as a rational fraction in z); but in the implementation of the corresponding digital filter, these configurations are no longer equivalent, in general, because of the effects of coefficient truncation and round off noise. Assuming that the coefficients for the configurations under consideration have been (or can be) quantized satisfactorily, the choice between these configurations is then determined by the level and character of their output round off noise.

= ϱ = ϱ = ϱ = ϱ = ϱ =

3. THE MODES OF ARITHMETIC

There are three modes of arithmetic which can be employed in the implementation of a digital filter: fixed -point, floating-point, and block-floating-point. Their

structures follow.

In fixed-point arithmetic each number x must satisfy certain inequalities such as $-1 \leq x \leq 1$. In general each number will be allowed a fixed number t of bits for its representation, and we shall say that the digital filter works with words of t bits [22].

In floating-point arithmetic each number x is represented by its sign and an ordered pair m and e such that $x = (\text{sgn})\ m\ 2^e$, where m and e each have a fixed number of bits. The number e, called the exponent, index, or characteristic, is an integer, and the number m, called the mantissa or the fractional part, is between $\frac{1}{2}$ and 1. Usually zero is also permitted for m. It is assumed that enough bits are allowed for the exponent so that no computed number will lie outside the permissable range.

When two floating-point machine numbers x and y each with a t-bit mantissa are multiplied, their exact product in floating-point representation has in general a fractional part of 2t or (2t-1) bits. To represent the product in the computer, only the most significant t bits of the mantissa can be retained. This can usually be achieved either by truncation or by rounding. Similarly truncation or rounding is needed in general before or after the addition of two floating-point numbers.

In block-floating-point arithmetic the input and filter states (i.e., the outputs of the delay registers) are jointly normalized before the multiplication and adds are performed using fixed-point arithmetic. The scale factor obtained during the normalization is then applied to the final output to produce a fixed-point result. To illustrate, consider a first order filter described by the difference equation

$$y_n = x_n + K\, y_{n-1}. \tag{1}$$

For convenience we will treat all numbers as fixed-point

fractions. To perform the computation in a block-floating-point manner, we define

$$A_n = \frac{1}{IP\left[\max\left\{|x_n|, |y_{n-1}|\right\}\right]} \tag{2}$$

where IP[M] is used to denote the integer power of two such that $M \leq IP[M] \leq 2M$, that is, with M written as $M = m\ 2^e$ with m between $\frac{1}{2}$ and 1, $IP[M] = 2^e$. For M a fraction, 2^e is less than or equal to unity so that A_n is greater than or equal to unity. Thus A_n represents the power-of-two scaling which will jointly normalize x_n and y_{n-1}. Thus with block-floating-point we can compute y_n as

$$y_n = \frac{1}{A_n}\left[A_n x_n + K A_n y_{n-1}\right] \tag{3}$$

where the multiplications and addition in (3) are carried out in a fixed-point manner.

Because of the recursive nature of the computation for a digital filter, it is advantageous to modify (3) as

$$\hat{y}_n = A_n x_n + K \Delta_n w_{1n} \tag{4}$$

with

$$w_{1n} = A_{n-1} y_{n-1}$$

$$y_n = \frac{1}{A_n} \hat{y}_n$$

and

$$\Delta_n = A_n / A_{n-1}.$$

The difference between (3) and (4) is meant to imply that the number $A_n y_n$ rather than y_n is stored in the delay register of the filter. Because of (2), $A_n y_n$ is always more accurate (or as accurate) as y_n since multiplication by A_n corresponds to a left shift of the register.

A disadvantage with (4) is that y_{n-1} must be

available to compute A_n, and Δ_n must then be obtained from A_n and A_{n-1}. An alternative is represented by the set of equations

$$\hat{y}_n = \Delta_n \hat{x}_n + K \Delta_n w_{1n} \tag{5a}$$

with

$$\hat{x}_n = A_{n-1} x_n \tag{5b}$$

and

$$\Delta_n = \frac{A_n}{A_{n-1}} = \frac{1}{IP\left[\max\left\{|\hat{x}_n|, |w_{1n}|\right\}\right]}. \tag{5c}$$

In this case, we first scale x_n by A_{n-1} to form $\hat{x}_n$ and then determine the incremental scaling using (5c). As in (4), the scaled value $\hat{y}_n$ is stored in the delay register and the output value y_n is determined from $\hat{y}_n$. If we consider the general case of an Nth order filter of the form

$$y_n = x_n + K_1 y_{n-1} + K_2 y_{n-2} + \cdots + K_N y_{n-N} ,$$

then the block-floating-point realization corresponding to (5) and represented in the direct form is depicted in Fig. 1. For the general case,

$$\Delta_n = \frac{1}{IP\left[\max\left\{|\hat{x}_n|, |w_{1n}|, |w_{2n}|, \cdots, |w_{Nn}|\right\}\right]} \tag{6}$$

and

$$A_n = \frac{1}{IP\left[\max\left\{|x_n|, |y_{n-1}|, |y_{n-2}|, \cdots, |y_{n-N}|\right\}\right]} \tag{7}$$

$$= A_{n-1} \Delta_n .$$

As an additional consideration, we note that because of the block normalization, there is the possibility of over-flow in the addition, which cannot be avoided by an attenuation of the input. This possibility of overflow can be

avoided by decreasing the normalitation constant A_n by a fixed amount. Thus we modify (6) and (7) as

$$\Delta_n = \frac{1}{r\,IP\left[\max\left\{|\hat{x}_n|, |w_{1n}|, |w_{2n}|, \cdots, |w_{Nn}|\right\}\right]} \qquad (6')$$

and

$$A_n = \frac{1}{r\,IP\left[\max\left\{|x_n|, |y_{n-1}|, |y_{n-2}|, \cdots, |y_{n-N}|\right\}\right]} \qquad (7')$$

where r is a constant that may be changed depending on the filter to be implemented. In a first order filter, for example, r need never be greater than two.

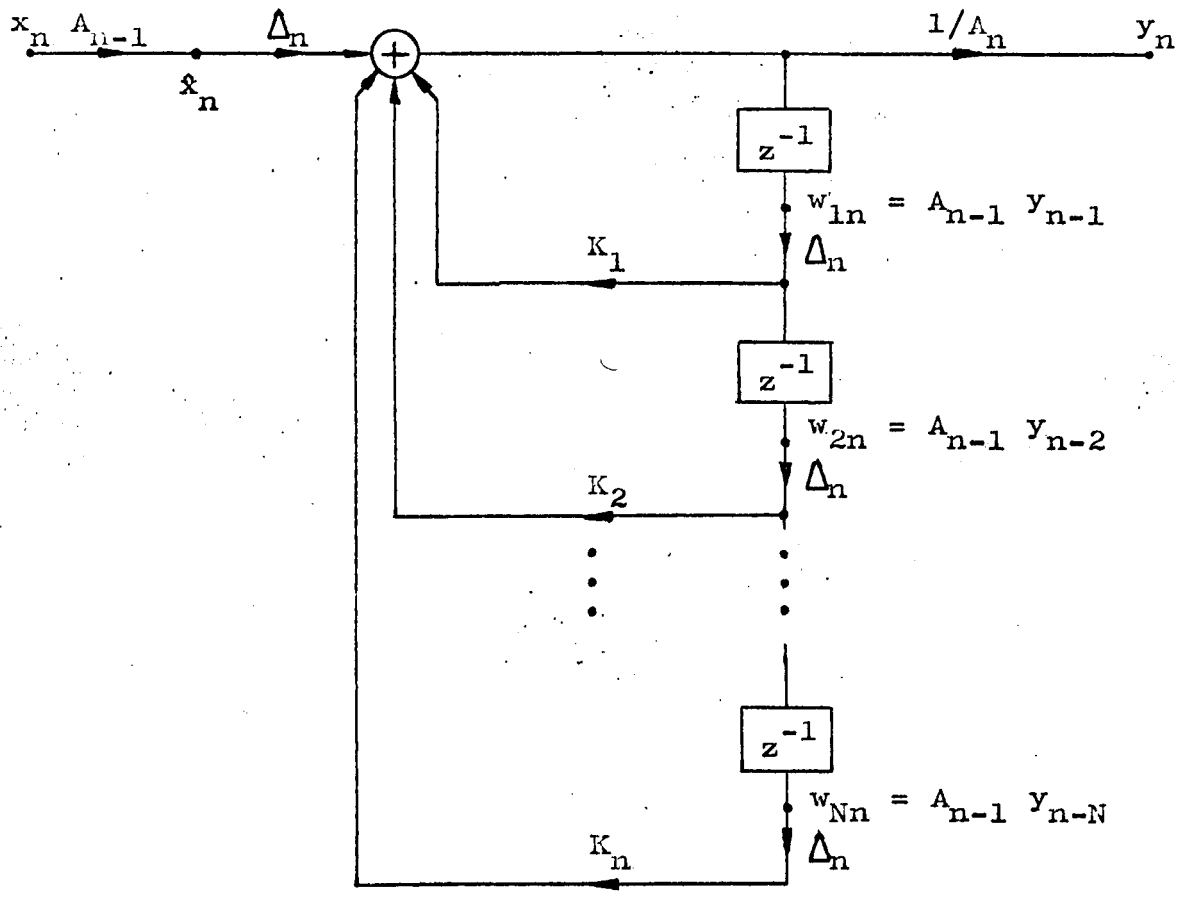


Figure 1. Network for block-floating-point realization of an Nth order filter.

= ♀ = ♀ = ♀ = ♀ = ♀ =

4. A COMPARISON OF ROUND OFF NOISE IN FIXED-POINT, FLOATING-POINT, AND BLOCK-FLOATING-POINT REALIZATIONS

There are basic differences concerning fixed-point and floating-point error estimation problems. Some of them stem from the fact that the modulus of every individual arithmetic error in the fixed-point mode is bounded by a constant determined by the machine, whereas the maximum modulus of the error in forming, for example, the floating-point sum of two floating-point numbers is proportional to the magnitude of the true sum.

Floating-point arithmetic has a larger dynamic range than fixed-point, but the latter is more accurate when the full register length can be utilized. Because of the limited dynamic range of fixed-point arithmetic, for high-gain filters, the input signal must be attenuated to prevent overflow in the output. Thus, for sufficiently high-gain, floating-point arithmetic leads to lower noise-to-signal ratio than fixed-point. On the other hand, floating-point arithmetic implies a more complex hardware structure than fixed-point arithmetic.

Block-floating-point is an alternative realization that provides a simplified form of automatic scaling of the filter data, and it lies somewhere between those of fixed-point and of floating-point.

In both the fixed-point and block-floating-point cases, the dynamic range for the output is constrained by the register length. Consequently, as the filter gain increases, the input must be scaled down to prevent the output from overflowing the register length. If h_n denotes the impulse response of the filter, then the output is given by

$$y_n = \sum_{j=o}^{\infty} h_j \, x_{n-j}$$

and it is bounded by

$$\max(|y_n|) = \max(|x_n|) \sum_{j=o}^{\infty} |h_j| . \qquad (8)$$

For a first order filter with impulse response $h_n = K^n$, this bound is

$$\max(|y_n|) = \max(|x_n|) \ (1/1-K). \qquad (9)$$

Interpreting the fixed-point numbers x_n and y_n as signed fractions, we require for no overflows that $|y_n| \leq 1$, restricting x_n to the range

$$-(1-K) \leq x_n \leq (1-K). \qquad (10)$$

Next we present statistical analyses of the effects of round off noise in first order filters implemented in fixed-point, floating-point, and block-floating-point. In each case we compute the output noise-to-signal ratio using experimental results given in the references [16], [17], and curves representing the output noise-to-signal ratio as a function of pole position are presented. Similar analyses for second order filters can be found in the references.

4.1. Fixed-Point Filter

For a first order filter (Fig. 2) of the form

$$y_n = K\, y_{n-1} + x_n \qquad (11)$$

with x_n white and uniformly distributed between the limits in (10), the variance of the input signal is

$$\sigma_x^2 = \frac{4(1-K)^2}{12} = \frac{(1-K)^2}{3}$$

and the variance of the output signal is

$$\sigma_y^2 = K^2 \sigma_y^2 + \sigma_x^2 = K^2 \sigma_y^2 + \frac{(1-K)^2}{3}$$

hence

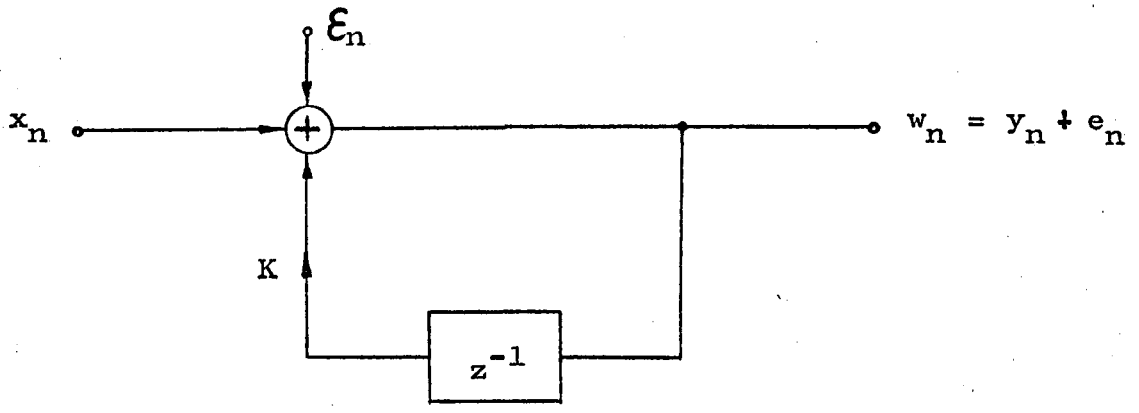


Figure 2. Noise model for first order filter.

$$\sigma_y^2 = \frac{(1-K)^2}{3(1-K^2)}$$

Since the noise ε_n is injected at the input of the filter (Fig. 2), we use (11) and get

$$\sigma_e^2 = K^2 \sigma_e^2 + \sigma_\varepsilon^2$$

where

$$\sigma_\varepsilon^2 = \frac{E_o^2}{12} = \frac{(2^{-t})^2}{12} = \frac{2^{-2t}}{12}.$$

Hence the variance of the output error is

$$\sigma_e^2 = \frac{2^{-2t}}{12(1-K^2)}. \qquad (13)$$

We note that the output noise is independent of the output signal variance. Dividing (13) and (12) we find the output noise-to-signal ratio for a first order filter implemented with fixed-point arithmetic as

$$\frac{\sigma_e^2}{\sigma_y^2} = \frac{2^{-2t}}{4(1-K)^2} \qquad (14)$$

4.2. Floating-Point Filter

For a first order filter of the form

$$y_n = K\, y_{n-1} + x_n \tag{15}$$

in floating-point arithmetic the computed output w_n is

$$w_n = \left[K\, w_{n-1}\, (1 + \varepsilon_n) + x_n \right] (1 + \delta_n) \tag{16}$$

Since the errors ε_n and δ_n arise from round off due to the floating-point multiply and add, respectively, we will assume that they are random variables and are independent of x_n and y_n. Furthermore we shall assume that the errors are independent from sample to sample (white). In Appendix I is shown that in the case of rounding, these errors are bounded by

$$-2^{-t} \leq \varepsilon_n, \delta_n \leq 2^{-t}$$

The error at the output at the n^{th} sample is the difference between the actual output and the ideal output.

$$e_n = w_n - y_n \tag{17}$$

Substracting (15) from (16) and neglecting second order terms in e, ε, and δ, we obtain a difference equation for the error e_n as

$$e_n - K\, e_{n-1} = K\, y_{n-1}\, (\varepsilon_n + \delta_n) + x_n\, \delta_n = u_n \tag{18}$$

With the assumptions above, u_n is white noise with variance dictated by the statistics of x_n and the variances σ_ε^2 and σ_δ^2 of ε_n and δ_n. The variance σ_e^2 of the output noise e_n is obtained easily from the variance σ_u^2 of u_n as

$$\sigma_e^2 = \sigma_u^2 \sum_{n=o}^{\infty} h_n^2 = \frac{1}{1 - K^2}\, \sigma_u^2 \tag{19}$$

where $h_n = K^n$ is the filter impulse response.

For example, if we assume that x_n is stationary white noise of variance σ_x^2, and using (15), (18), and (19), we obtain

$$\sigma_e^2 = \frac{\sigma_\delta^2 + K^2 \sigma_\varepsilon^2}{(1 - K^2)^2} \sigma_x^2 = \frac{\sigma_\delta^2 + K^2 \sigma_\varepsilon^2}{(1 - K^2)} \sigma_y^2 \qquad (20)$$

Since both ε and δ are due to quantizing, it is reasonable to assume that they are uniformly distributed in the range $(-2^{-t}, 2^{-t})$ with variances $\sigma_\varepsilon^2 = \sigma_\delta^2 = 2^{-2t}/3$ [15], [16]. Actual measurements [10] of the noise due to a multiply and an add verified that the variances

$$\sigma_\varepsilon^2 = \sigma_\delta^2 = (0.23)\ (2^{-2t}) \qquad (21)$$

would better represent these noise sources. Using (20) and (21), we can compute the output noise-to-signal ratio as

$$\frac{\sigma_e^2}{\sigma_y^2} = (0.23)\ (2^{-2t})\ \frac{1 + K^2}{1 - K^2} \qquad (22)$$

4.3. Block-Floating-Point Filters

In evaluating the performance of the block-floating-point realization in the presence of round off noise [17], we will restrict attention to the implementation of (5) and Fig. 1 for the first order case. We will assume that no round off occurs in the computation of $\hat{x}_n$ from x_n and the subsequent multiplication by Δ_n. Since A_{n-1} and $A_{n-1}\Delta_n$ are always nonnegative powers of two, that is, they always correspond to a positive scaling, the above assumption corresponds to allowing more bits in the representation of the intermediate variable $\hat{x}_n$. This is reasonable if we take the attitude that it is primarily in the variables used in the arithmetic computations that the register length is important.

For the first order case, round off noise is

introduced in the multiplication of w_{1n} by Δ_n, the multiplication by K, and the final multiplication by $1/A_n$. The effects of multiplier round off will be modeled by representing the round off by additive white noise sources. We consider, for convenience, the fixed-point numbers in the registers to represent signed fractions, with the register length excluding sign denoted by t bits. Each of the round off noise generators is assumed to be white, mutually independent and independent of the input, and to have a variance σ_ε^2 equal to $(1/12)2^{-2t}$. The network for the first order filter including the noise sources representing round off error is presented in Fig. 3(A). In Fig. 3(B) an equivalent representation is shown, where the noise sources are at the filter input. If we consider the input to be a stationary random signal, then the noise source δ_n will be white stationary random noise with variance

$$\sigma_\delta^2 = \sigma_\varepsilon^2 \, (1 + K^2) \, C^2 \tag{23}$$

where C^2 denotes the expected value of $(1/A_n)^2$. Letting e_n denote the noise in the filter output due to the noise δ_n, the variance of the output noise e_n will be

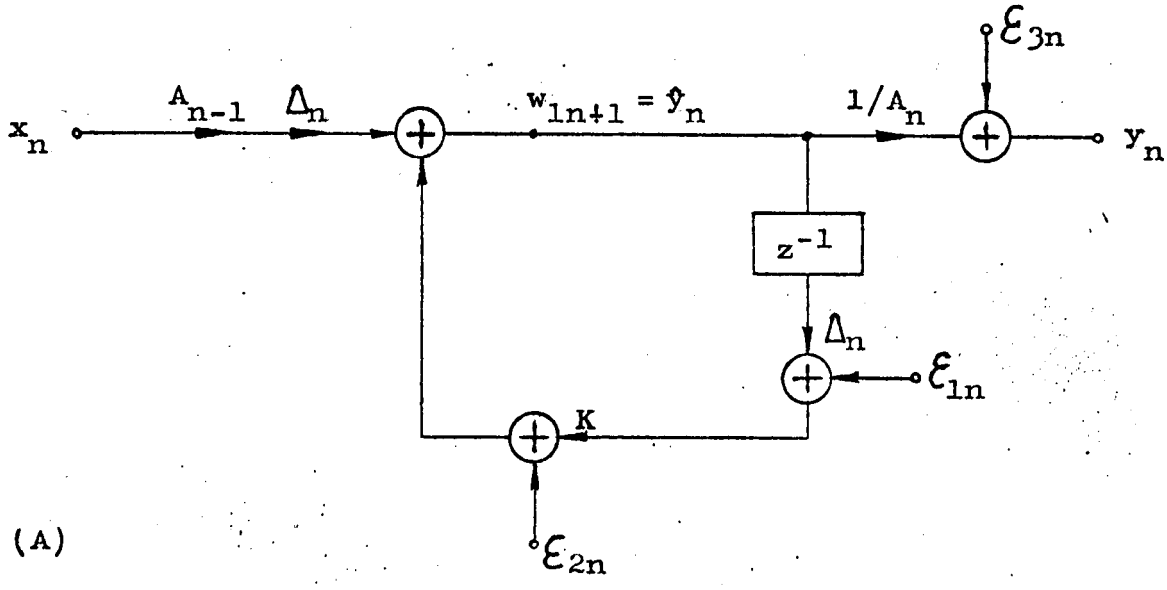
$$\sigma_e^2 = \sigma_\delta^2 \sum_{n=0}^{\infty} h_n^2 + (\sigma_{\varepsilon 3n})^2 = \sigma_\varepsilon^2 \left[1 + \frac{1 + K^2}{1 - K^2} \, C^2 \right]$$

$$= \frac{2^{-2t}}{12} \left[1 + \frac{1 + K^2}{1 - K^2} \, C^2 \right]. \tag{24}$$

This result is derived by observing that in Fig. 3(B) the transmission from the noise source δ_n to the output is that of a first order filter with unit sample response h_n given by $h_n = K^n$. An experimental verification of (24) is given in ref. [17]. We note that the expression (24) for the noise has a term independent of the signal and a term which depends on the signal through the factor C^2.

Dividing (24) by the variance of the output signal (12) we find the noise-to-signal ratio

$$\frac{\sigma_e^2}{\sigma_y^2} = \frac{2^{-2t}}{4}\left[\frac{1-K^2}{(1-K)^2} + \frac{1+K^2}{(1+K)^2}\,C^2\right] \qquad (25)$$



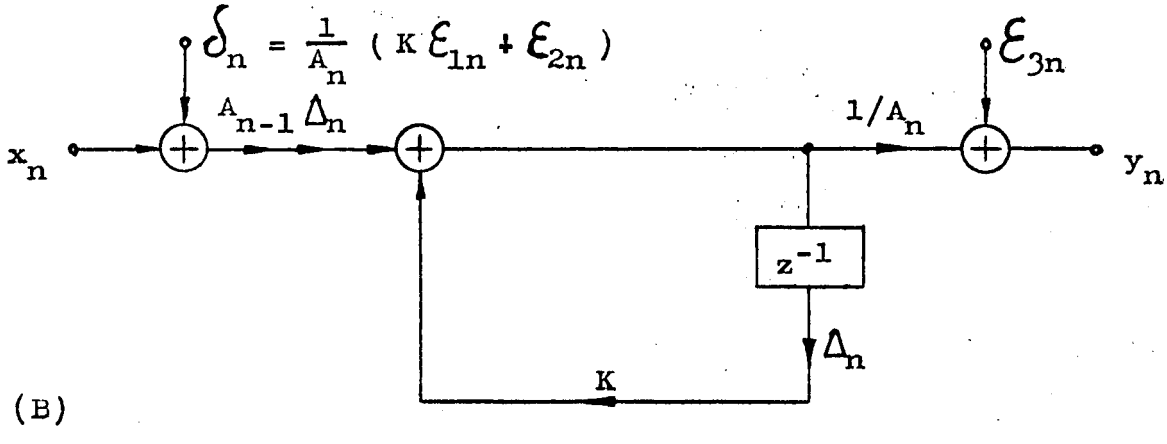


Figure 3. (A) Noise model for block-floating-point first order filter.

(B) Equivalent noise model.

4.4. Comparison of Noise-to-Signal Ratios

In Fig. 4, (14),(22), and (25) are compared [17]. The noise-to-signal ratios for first order filters are plotted in bits so that the difference between two of the curves reflects the number of bits that the mantissas should differ by to achieve the same noise-to-signal ratio. The difference between floating-point and block--floating-point is approximately constant (one bit) as the filter gain (or the proximity of the poles to the unit circle) increases. In contrast, the fixed-point noise-to--signal ratio increases at a faster rate than floating--point or block-floating-point, and for low gain is better and for high gain is worse than block-floating-point.

In evaluating the comparison between fixed-point, floating-point, and block-floating-point filter realizations, it is important to note that Fig. 4 is based only on the mantissa length and do not reflect the additional bits needed to represent the characteristic in either floating-point or block-floating-point arithmetic.

An additional consideration which is not reflected in these curves is that in both fixed-point and block--floating-point the noise-to-signal ratio is computed on the assumption that the input signal is as large as possible consistent with the requirement that the output fit within the register length. If the input signal is in fact smaller than permitted, then the noise-to-signal ratio for the fixed-point case will be proportionately higher. For block-floating-point, as the input signal decreases, C^2 decreases, thus reducing the output noise. From (24) we observe that as the input signal decreases the output noise variance asymtotically approaches σ_ε^2. For the case of high gain filters, (14),(22), and (25) can be approximated by asymptotic expressions which place in evidence the relationship between them [17].

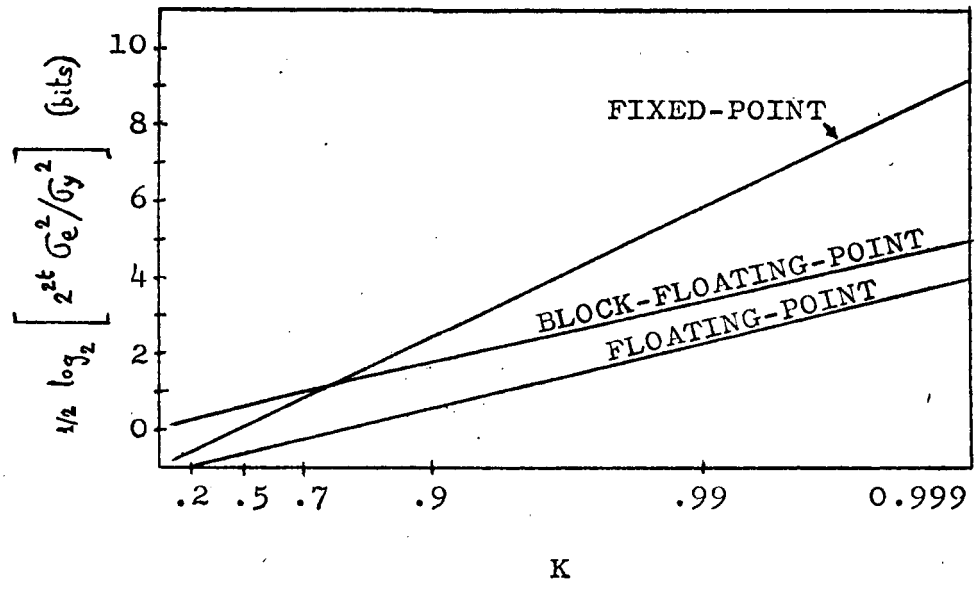


Figure 4. Comparison of noise-to-signal ratios for first order filter using fixed-point, floating-point, and block-floating-point arithmetic.

= ♀ = ♀ = ♀ = ♀ = ♀ =

5. SUMMARY

The variance of the round off noise as it appears in the output has been calculated for a first order filter implemented with fixed-point, floating-point, and block--floating-point arithmetic. In the fixed-point case the output noise is independent of the output signal variance, and in the floating-point case the output noise is proportional to the output signal variance. The expression for block-floating-point noise has a term independent of the signal and a term which depends on the signal.

In the case of first order filter with impulse response $h_n = K^n$ and the input signal constrained in the range $-(1 - K) \leq x_n \leq (1 - K)$, the noise-to-signal ratios have been computed and plotted in bits as a function of

pole position. The difference between floating-point and block-floating-point is approximately constant as the filter gain increases. In contrast, the fixed-point noise--to-signal ratio increases at a faster rate than floating--point or block-floating-point, and for low gain is better and for high gain is worse than block-floating-point.

= o = o = o = o = o =

6. APPENDIX

In this appendix we derive bounds for the round off errors due to the floating-point multiply and add.

Consider the number x represented in floating-point form, $x = (sgn)\ m\ 2^{e}$. The exponent is given by

$$e = \left[\log_2 x\right]$$

where the brackets $[\cdot]$ denote the smallest integer exceeding the quantity inside the brackets. The mantissa is therefore

$$m = x/2^{e}$$

For convenience x is taken to be positive. If only t bits is allowed to the mantissa, m must be so truncated or rounded, thus commiting an error e given by

$$e = x - m_t\ 2^{e} = 2^{e}\ (m - m_t) = 2^{e}\ e' \qquad (26)$$

where m_t is the truncated or the rounded version of m. We denote e' the difference $m - m_t$. It is clear that e' is bounded by

$-2^{-t-1} \leq e' \leq 2^{-t-1}$ for rounding, and

$-2^{-t} \leq e' \leq 0$ for truncation.

Therefore e is bounded by

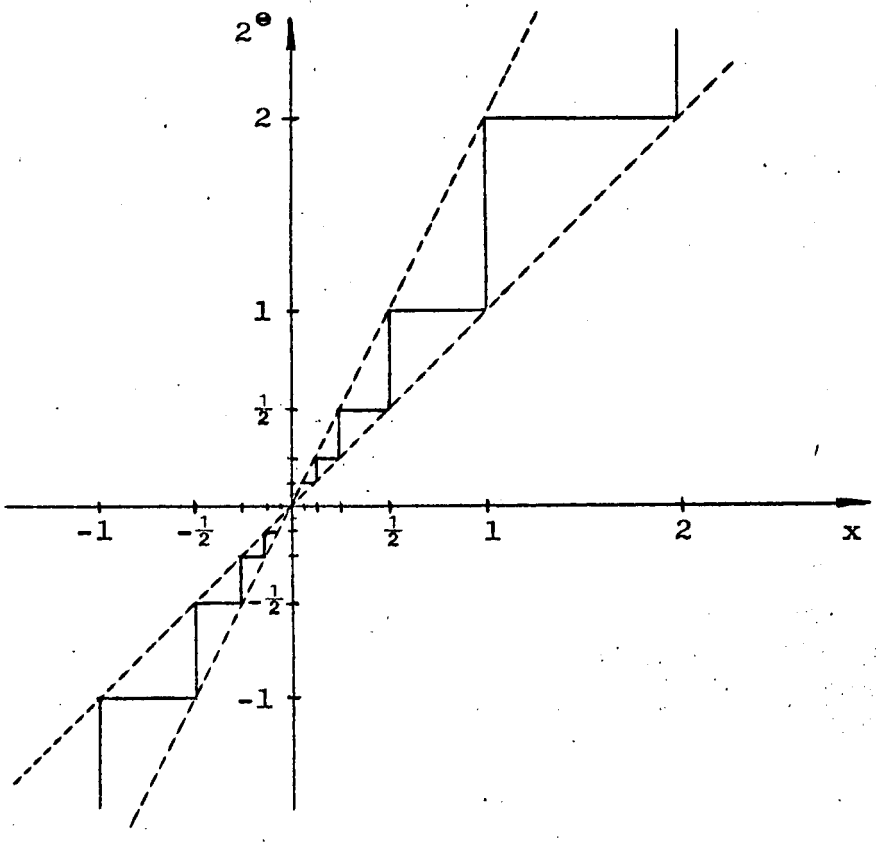


Figure 5. 2^e as a function of x.

$$-2^e\, 2^{-t-1} \leq e \leq 2^e\, 2^{-t-1} \qquad \text{for rounding, and}$$

$$-2^e\, 2^{-t} \leq e \leq 0 \qquad \text{for truncation.}$$

The function 2^e is a piecewise constant function of x and is sketched in Fig. 5. It is seen that

$$x \leq 2^e \leq 2x, \text{ or}$$

$$1 \leq (2^e/x) \leq 2.$$

The above characterization of e and e' is not easy to use in analysis because of the nonlinear dependence of 2^e on x. Equation (26) may be rewritten as

$$e = x\, e''$$

with

$$e'' = (m - m_t)\ (2^e/x) \qquad (27)$$

It is clear from (27) that

$$-2^{-t} \leq e'' \leq 2^{-t} \qquad \text{for rounding, and}$$

$$-2^{-t+1} \leq e'' \leq 0 \qquad \text{for truncation.}$$

Using the notation $fl(\cdot)$ to denote the machine number resulting from performing the arithmetic operation specified by the parenthesis, for multiplication and addition we have:

$$fl(x\,y) = x\,y\,(1+\delta)$$

$$fl(x+y) = (x+y)\,(1+\varepsilon)$$

where if the two floating-point numbers x and y have t-bit mantissa ε and δ are bounded by

$$-2^{-t} \leq \varepsilon, \delta \leq 2^{-t} \qquad \text{for rounding, and}$$

$$-2^{-t+1} \leq \varepsilon, \delta \leq 0 \qquad \text{for truncation.}$$

= ♀ = ♀ = ♀ = ♀ = ♀ =

7. REFERENCES

[1] J. F. Kaiser, "Some practical considerations in the realization of linear digital filters," Proc. 3rd. Ann. Allerton Conf. on Circuit and System Theory, pp. 621-633, October 1965.

[2] C. M. Rader and B. Gold, "Effects of parameter quantization on the poles of a digital filter," Proc. IEEE, vol. 55, pp. 688-689, May 1967.

[3] J. B. Knowles and E. M. Olcayto, "Coefficient accuracy and digital filter response," IEEE Trans. on Circuit Theory, vol. CT-15, pp. 31-41, March 1968.

[4] W. R. Bennett, "Spectra of quantized signals," Bell Sys. Tech. J., vol. 27, pp. 446-472, July 1948.

[5] B. Widrow, "Statistical analysis of amplitude-quantized sampled-data systems," AIEE Trans.

(Aplication and Industry), vol. 59, pp. 555-568, 1960 (January 1971 section).

[6] J. Katzenelson, "On errors introduced by combined sampling and quantization," IRE Trans. Automatic Control, vol. AC-7, pp. 58-68, April 1962.

[7] J. B. Knowles and R. Edwards, "Effect of a finite--word-length computer in a sampled-data feedback system," Proc. IEE (London), vol. 112, pp. 1197-1207, June 1965.

[8] B. Gold and C. M. Rader, "Effect of quantization noise in digital filters," 1966 Spring Joint Computer Conf., AFIPS Proc., vol. 28. Washington, D. C.: Spartan Books, 1966, pp. 213-219.

[9] ——, Digital Processing of Signals, New York: Mc Graw-Hill, 1969, pp. 1-130

[10] C. Weinstein and A. V. Oppenheim, "A comparison of roundoff noise in floating point and fixed point digital filters realizations," Proc. IEEE (Letters), vol. 57, pp. 1181-1183, June 1969.

[11] L. B. Jackson, "An analysis of roundoff noise in digital filters," Sc. D. Thesis, Stevens Institute of Technology, Hoboken, New Jersey, 1969.

[12] ——, "On the interaction of roundoff noise and dynamic range in digital filters," Bell Sys. Tech. J., vol. 49, pp. 159-184, February 1970.

[13] ——, "Roundoff-noise analysis for fixed point digital filters realized in cascade or parallel form," IEEE Trans. Audio and Electroacustics, vol. AU-18, pp. 107-122, June 1970.

[14] I. W. Sandberg, "Floating-point round-off accumulation in digital filter realization," Bell Sys. Tech. J., vol. 46, pp. 1775-1971, October 1967.

[15] T. Kaneko and B. Liu, "Round-off error of floating-point digital filters," Proc. 6th. Ann. Allerton Conf. on Circuit and System Theory, October 1968.

[16] B. Liu and T. Kaneko, "Error analysis of digital filters realized with floating-point arithmetic," Proc. IEEE, vol. 57, pp. 1735-1747, October 1969.

[17] A. V. Oppenheim, "Realization of digital filters using block-floating-point arithmetic," IEEE Trans. Audio and Electroacustics, vol. AU-18, pp. 130-136, June 1970.

[18] I. W. Sandberg, "A theorem concerning limit cycles in digital filters," Proc. 7th Ann. Allerton Conf. on Circuit and System Theory, October 1969.

[19] H. A. Ojongbed, "Limit-cycle constraints for recursive-digital-filter design," Electronic Letters, vol. 6, p. 698, 1970.

[20] R. B. Blackman, Linear Data-Smoothing and Prediction in Theory and in Practice. Reading, Mass.: Addison-Wesley, 1965, pp. 75-81.

[21] R. Edwards, J. Bradley, and J. B. Knowles, "Comparison of noise performance of programming methods in the realization of digital filters," Proc. of the Symposium on Computer Processing in Communications, XIX, PIB-MRI Symposia Series, 1969.

[22] J. H. Wilkinson, Rounding Errors in Algebraic Processes, Englewood Cliffs, New Jersey: Prentice-Hall, 1963.

[23] R. B. Blackman and J. W. Tukey, The Measurement of Power Spectra from the Point of view of Communication Engineering. New York: Dover, 1959.

= ₀ = ₀ = ₀ = ₀ = ₀ =

= ₀ = ₀ = ₀ =

= ₀ =