Quantization Errors in the Computation of the Discrete Hartley Transform

AVIDEH ZAKHOR AND ALAN V. OPPENHEIM, FELLOW, IEEE

Abstract—**The principal objective of this paper is the study of the arithmetic roundoff error characteristics of several discrete Hartley transform (DHT) algorithms. We first summarize a variety of efficient DHT algorithms including Bracewell's original decimation-in-time radix-2 algorithm. Statistical models for fixed- and floating-point arithmetic roundoff errors are then used as the basis for a theoretical study of roundoff noise characteristics of a number of the DHT algorithms. The results of a detailed experimental study of roundoff noise are compared to the theoretical predictions. In fixed-point implementation of the decimation-in-time and frequency radix-2 algorithms, it is found that the noise-to-signal ratio increases approximately 1.1 bits per stage. For the floating-point implementation, the number of bits of rms noise-to-signal ratio for all the algorithms increase as $\sqrt{\log_2 N}$, so that doubling the number of points produced a mild increase in the output noise.**

I. Introduction

THE continuous-time Hartley transform, originally proposed by Hartley [2] in 1942, has many properties similar to those of the Fourier transform. An important distinction is that the Hartley transform of a real-valued function is also real valued, and furthermore, its evaluation does not involve complex functions. This is a potential advantage if the transform is to be explicitly computed.

As defined by Bracewell, the discrete Hartley transform (DHT) of a finite length sequence and its inverse are:

$$H(k) = \sum_{n=0}^{N-1} x[n] \operatorname{cas}\left(\frac{2\pi nk}{N}\right) \qquad 0 \le k \le N-1 \qquad (1.1a)$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} H(k) \operatorname{cas}\left(\frac{2\pi nk}{N}\right) \qquad 0 \le n \le N-1 \qquad (1.1b)$$

where

$$\operatorname{cas} \alpha \equiv \cos \alpha + \sin \alpha.$$

The DHT and the DFT are closely related. Specifically, the even and odd parts of the DHT correspond to the real and negative of the imaginary parts of the DFT. Consequently, the DHT can be used to obtain the DFT, and vice versa. In addition, as Bracewell points out, a variety of algorithms which traditionally utilize the DFT, such as convolution and correlation, and spectral analysis, can just as effectively be carried out with the DHT, which for real sequences requires only real-valued computation.

The DHT has fast algorithms similar in style to the FFT, the first of these proposed by Bracewell [4]. The principal objective of this paper is the study of the arithmetic roundoff error characteristics of several DHT algorithms. In Section II we summarize a variety of efficient DHT algorithms including Bracewell's original decimation-in-time radix-2 algorithm. In Sections III and IV, statistical models for fixed- and floating-point arithmetic roundoff errors and linear system noise theory are used as the basis for a theoretical study of the roundoff noise characteristics of a number of the DHT algorithms. The results of a detailed experimental study of roundoff noise are compared to the theoretical predictions.

II. DHT Algorithms

Bracewell proposed a decimation-in-time radix-2 algorithm for performing the discrete Hartley transform of a data sequence of N real elements in a time proportional to $N \log_2 N$ [4]. In the remainder of this paper we shall refer to this algorithm as DT1, where DT stands for decimation-in-time. In this section, we will review Bracewell's decomposition, propose a minor modification to the DT1 algorithm, and discuss the decimation-in-frequency version of the DT1 algorithm [6].

The DT1 algorithm can be derived by separating $x(n)$ into two $(N/2)$-point sequences consisting of even and odd points in $x(n)$. Thus, we obtain

$$H(k) = H_1(k) + H_2(k) \qquad (2.1)$$

where $H_1(k)$ is the $(N/2)$-point DHT of the even part of $x(n)$ and $H_2(k)$ which is given by

$$H_2(k) = \sum_{n=0}^{N/2-1} x(2n+1) \operatorname{cas}\left(\frac{2\pi(2n+1)k}{N}\right) \qquad (2.2)$$

and can be written in terms of $H_3(k)$, the $(N/2)$-point DHT of the odd part of $x(n)$.

The butterfly for the last stage of an N-point DHT using the above decomposition is shown in Fig. 1. It should be clear that computing any four points $H(k)$, $H((N/2) - k)$, $H((N/2) + k)$, and $H(N - k)$ in Fig. 1 with $0 \le k < (N/4)$ requires the computation of the two interme-

Manuscript received March 3, 1986; revised May 12, 1987. This work was supported in part by the Advanced Research Projects Agency monitored by the ONR under Contract N00014-81-K-0742, and in part by the National Science Foundation under Grant ECS-8407285. The work of A. Zakhor was supported by a Hertz Foundation fellowship.

The authors are with the Massachusetts Institute of Technology, Research Laboratory of Electronics, Department of Electrical Engineering & Computer Science, Cambridge, MA 02139.

IEEE Log Number 8716500.

0096-3518/87/1100-1592$01.00 © 1987 IEEE

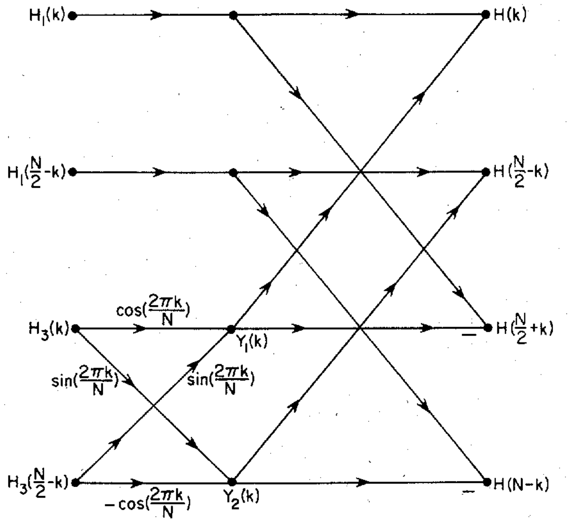


Fig. 1. Flow graph of the kth butterfly of the last stage of an N-point DHT computation using the DT1 algorithm.

diate quantities:[1]

$$Y_1(k) = H_3(k)\cos\left(\frac{2\pi k}{N}\right) + H_3\left(\left(\frac{N}{2}-k\right)\right)_{N/2} \cdot R_{N/2}(k)\sin\left(\frac{2\pi k}{N}\right) \quad (2.3a)$$

$$Y_2(k) = H_3(k)\sin\left(\frac{2\pi k}{N}\right) - H_3\left(\left(\frac{N}{2}-k\right)\right)_{N/2} \cdot R_{N/2}(k)\cos\left(\frac{2\pi k}{N}\right). \quad (2.3b)$$

Instead of using 4 multiplies and 2 adds, $Y_1(k)$ and $Y_2(k)$ can be computed with three of each in the following manner:

$$Y_1(k) = \left[\sin\left(\frac{2\pi k}{N}\right) + \cos\left(\frac{2\pi k}{N}\right)\right] \cdot H_3(k) + \sin\left(\frac{2\pi k}{N}\right) \cdot \left[H_3\left(\left(\frac{N}{2}-k\right)\right)_{N/2} R_{N/2}(k) - H_3(k)\right] \quad (2.4a)$$

$$Y_2(k) = \left[\sin\left(\frac{2\pi k}{N}\right) - \cos\left(\frac{2\pi k}{N}\right)\right] \cdot H_3\left(\frac{N}{2}-k\right) - \sin\left(\frac{2\pi k}{N}\right) \cdot \left[H_3\left(\left(\frac{N}{2}-k\right)\right)_{N/2} R_{N/2}(k) - H_3(k)\right]. \quad (2.4b)$$

The above implementation will be referred to as the MDT1 algorithm where MDT stands for modified decimation-in-time. Although the total operation count for the original algorithm (DT1) and the modified algorithm are the same, the error properties of the MDT1 are different from the DT1 algorithm.

An alternative radix-2 algorithm with fewer multiplies can be obtained by using the identity

$$2\cos(\beta)\,\text{cas}(\alpha) = \text{cas}(\alpha+\beta) + \text{cas}(\alpha-\beta). \quad (2.5)$$

Specifically, letting $\alpha = 2\pi(2n+1)k/N$ and $\beta = 2\pi k/N$ and multiplying both sides of (2.2) by $\cos(2\pi k/N)$, $H_2(k)$ of (2.2) can be written as:

$$H_2(k) = \begin{cases} \dfrac{1}{2\cos\left(\frac{2\pi k}{N}\right)} \displaystyle\sum_{n=0}^{N/2-1} [x(2n+1) + x(2n-1)]\,\text{cas}\left(\frac{2\pi nk}{N/2}\right) & 0 \le k < \dfrac{N}{2},\quad k \ne \dfrac{N}{4} \\ \displaystyle\sum_{n=0}^{N/2-1} x(2n+1)(-1)^n & k = \dfrac{N}{4} \end{cases} \quad (2.6a)$$

and

$$H_2(k) = -H_2\left(k - \frac{N}{2}\right) \qquad \frac{N}{2} \le k < N \quad (2.6b)$$

where

$$x(-1) \equiv x(N-1).$$

Equation (2.6a) shows that $H_2(k)$ can also be computed via an $(N/2)$-point DHT. By repeating the above process we can decompose the DHT further. Although this algorithm would potentially be faster than Bracewell's algorithm, as shown in [18], its error characteristics are less favorable due to the factor

$$\frac{1}{2\cos(2\pi k/N)}$$

in (2.6a). This is because for values of k close to $(N/4)$,

[1] Throughout the paper, sequences and the associated DHT are considered implicitly to be finite length. Following the notation in [1] $((\cdot))_N$ is used to denote the argument modulo N so that, for example, $H((k))_N$ is $H(k \bmod N)$ and, in particular, is $H(k)$ periodically repeated with period N. The function $R_N(k)$ is the rectangular gating function given by

$$R_N(k) = \begin{cases} 1 & 0 \le k \le N-1 \\ 0 & \text{otherwise.} \end{cases}$$

the above factor becomes very large resulting in magnification of error at the corresponding output frequency point.

As is the case with FFT, the idea behind the DT1 algorithm can be extended to the decimation-in-frequency algorithm which we will refer to as the DF1 algorithm [6].

III. Roundoff Error Analysis of Fixed-Point Implementation of DT1 and DF1 Algorithms

In this section, the effects of fixed-point roundoff errors in computing the DHT algorithms of Section II will be explored. Our approach is to model the error sources statistically, with experiments used to test their validity.

In fixed-point arithmetic, rounding errors occur only when multiplications are performed. Fixed-point additions are free of errors provided no overflows occur. With no loss of generality, we consider fixed-point numbers to be represented as $(b + 1)$-bit binary fractions, with the binary point just to the right of the highest order bit. We will also assume that two's complement representation of negative numbers is used, and that the roundoff error in multiplying two fixed-point b-bit numbers has a uniform probability density function in the interval $(-\frac{1}{2}2^{-b}, \frac{1}{2}2^{-b})$ with variance of $\sigma_\epsilon^2 = \frac{1}{12}2^{-2b}$. Furthermore, the roundoff errors due to multiplications are assumed to be uncorrelated with each other and with the input. Based on these assumptions, we model roundoff noise by inserting additive signal independent white noise source generators at every multiplier that appears in the flow graph of a specific algorithm and then analyze the effects of the noise sources on the output.

In Section III-A and B we will derive the statistical error properties of fixed-point implementation of the DT1 and DF1 algorithms. Roundoff noise characteristics of the MDT1 algorithm using fixed-point arithmetic is almost identical to that of the DT1 algorithm and is described in [18] in detail. In Section III-C, the experimental results are compared to the theoretical predictions of Section III-A and B.

A. Roundoff Noise Analysis of the DT1 Algorithm for Fixed-Point Arithmetic

The simplified roundoff noise analysis of the DT1 algorithm with fixed-point arithmetic resembles that of the FFT. Taking into account the fact that the first two stages of the DT1 algorithm are error free, we can show that [12], [18]

$$\sigma^2(N, k) = \sum_{n=3}^{\nu} 2^{\nu-n+1}\sigma_\epsilon^2 = (2^{\nu-1} - 2)\sigma_\epsilon^2 \quad (3.1)$$

where $\sigma^2(N, k)$ denotes the variance of error at the kth point of an N-point DHT and $N = 2^\nu$.

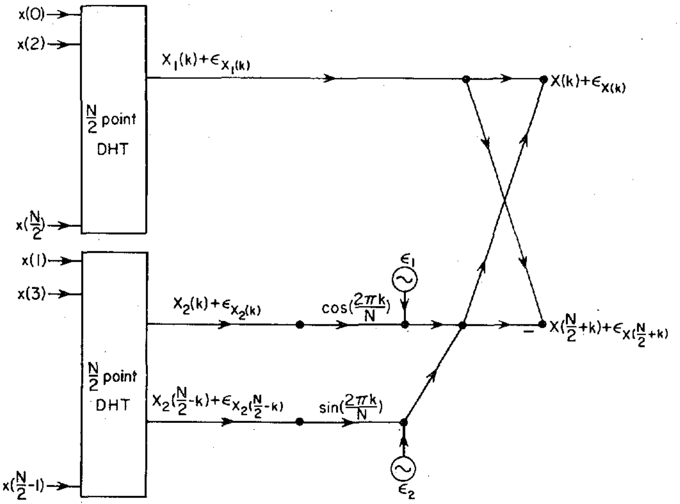


Fig. 2. Statistical model for fixed-point roundoff noise in a flow graph of the decimation-in-time decomposition of an N-point DHT computation into two $N/2$-point DHT computations using the DT1 algorithm.

In order to simplify the analysis leading to (3.1), the fact that multiplications by unity and zero can be performed noiselessly is neglected. In order to take into account this special case, an alternative way of finding the output noise variance is suggested. Fig. 2 shows the DT1 decomposition of an N-point sequence $x(n)$ into two $(N/2)$-point sequences $x_1(n)$ and $x_2(n)$ with $X_1(k)$ and $X_2(k)$ denoting their DHT's, respectively. Let $\epsilon_{X_i(k)}$ denote the error in $X_i(k)$. Then by inspection of Fig. 2 we have

$$\begin{aligned}\epsilon_{X(k)} &= \epsilon_{X_1(k)} + \epsilon_{X_2(k)}\cos\left(\frac{2\pi k}{N}\right)\\ &\quad + \epsilon_{X_2((N/2)-k)}\sin\left(\frac{2\pi k}{N}\right) + \epsilon_1 + \epsilon_2\end{aligned} \quad (3.2a)$$

$$\begin{aligned}\epsilon_{X(k+(N/2))} &= \epsilon_{X_1(k)} - \epsilon_{X_2(k)}\cos\left(\frac{2\pi k}{N}\right)\\ &\quad - \epsilon_{X_2((N/2)-k)}\sin\left(\frac{2\pi k}{N}\right) - \epsilon_1 - \epsilon_2\end{aligned} \quad (3.2b)$$

where ϵ_i denotes the roundoff error due to multiplications by sine and cosine. If we know the output noise variance distribution for $(N/2)$-point sequences, using (3.2) we can find it for N-point sequences. More specifically, we get

$$\sigma^2(N, k) = \begin{cases} \sigma^2\left(\frac{N}{2}, k\right) + \cos^2\left(\frac{2\pi k}{N}\right)\sigma^2\left(\frac{N}{2}, k\right) \\ \qquad 0 < k < \frac{N}{2}, \quad k \neq \frac{N}{4} \\ + \sin^2\left(\frac{2\pi k}{N}\right)\sigma^2\left(\frac{N}{2}, \frac{N}{2} - k\right) + 2\sigma_\epsilon^2 \\ 2\sigma^2\left(\frac{N}{2}, k\right) \qquad k = 0, \frac{N}{4} \end{cases} \quad (3.3a)$$

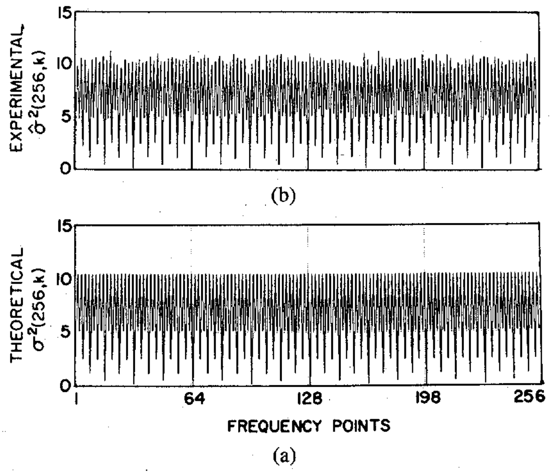
$$\sigma^2(N, k) = \sigma^2\left(N, k - \frac{N}{2}\right) \quad \frac{N}{2} \le k < N. \tag{3.3b}$$

Note that for $k = 0, N/4, N/2, (3N)/4$, the coefficients of the butterflies of Fig. 2 become 0 or 1. By taking care of these special cases, we have incorporated the noiselessness of these multipliers in our theoretical predictions.[2] Fig. 3(a) shows the distribution of variance of error for 256-point sequences using (3.3).

To obtain a formula for output noise-to-signal ratio, we next consider the dynamic range constraint. It can easily be shown that the magnitude of the output of the Hartley transform is less than 1 provided the magnitude of all the input points are less than $1/N$. Using this fact for both the half-length and full-length transform, one can show that no overflow can occur internally either.[3] Thus, assuming that the input is white and uniformly distributed in the interval $(-1/N, 1/N)$, the output signal variance is given by

$$\sigma^2_{\text{out}} = \frac{1}{3N}.$$

In Fig. 4, the average output noise-to-signal ratio using (3.3) as a function of transform size is plotted. In Section III-C, the experimental results confirming the theoretical predictions of Figs. 2 and 3(a) will be presented.



(b)

(a)

Fig. 3. Distribution of the output noise variance for 256-point sequences for fixed-point realization of the DT1 algorithm: (a) theoretical; (b) experimental.

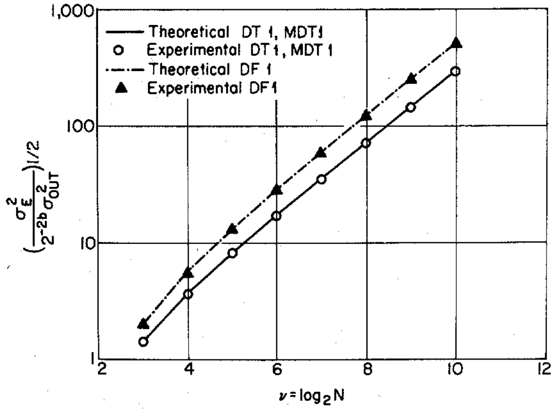


Fig. 4. Output noise-to-signal ratio for fixed-point realization of DT1, MDT1, and DF1 algorithms.

B. Roundoff Noise Analysis of the DF1 Algorithm Using Fixed-Point Arithmetic

The simplified theoretical analysis for the DF1 algorithm is very similar to that of the DT1 algorithm and the FFT [12], [18]. Taking into account the fact that the last two stages of the DF1 algorithm are error free, we can show that [12], [18]

$$\sigma^2(N, k) = \sum_{m=1}^{\nu-2} 2^{\nu-m+1}\, \sigma_\epsilon^2 = (2^{\nu+1} - 8)\, \sigma_\epsilon^2. \tag{3.4}$$

The analyses which led to (3.4) and (3.1) suggest that the output noise variance for an N-point sequence is proportional to $N/2$ for the decimation-in-time algorithm, and $2N$ for the decimation-in-frequency algorithm. This difference in the two algorithms can be explained intuitively by noting that the butterflies with zero and unity coefficients form the first two stages in the decimation-in-time implementation, and form the last two stages in the decimation-in-frequency implementation. Therefore, in the decimation-in-time algorithm, the presence or absence of these stages does not affect the output noise variance at all. However, in the decimation-in-frequency implementation, the variance of error due to the first $(\nu - 2)$ stages of the algorithm double as they propagate through each of these last two stages, resulting in a factor of four difference in the output noise variance for the DF1 algorithm as compared to the DT1 algorithm.

In order to simplify the analysis leading to (3.4), the fact that multiplications by unity and zero can be performed noiselessly is neglected. To improve the accuracy of our analysis, an alternative way of finding the output noise variance is suggested. Fig. 5 shows the decomposition of an N-point sequence $x(n)$ into two $(N/2)$-point sequences $x_1(n)$ and $x_2(n)$ using the DF1 decomposition. Let $\epsilon_{x_i(n)}$ denote the error in $x_i(n)$. By inspection of Fig. 5 we have

$$\epsilon_{x_1(n)} = 0 \tag{3.5a}$$

[2] In deriving (3.3) from (3.2), we used the fact that $\epsilon_{X_2(k)}$ and $\epsilon_{X_2((N/2)-k)}$ are uncorrelated. In general, we can use induction to show that if at the ith stage

$$\text{Var}\,[\epsilon_{X_i((N/2^i)-k)}] = \text{Var}\,[\epsilon_{X_i(k)}]$$

and

$$E[\epsilon_{X_i((N/2^i)-k)}\, \epsilon_{X_i(k)}] = 0,$$

then the above equations also hold at the $(i + 1)$st stage.

[3] Note that the input is not usually scaled by the full amount N, but the variables are scaled by 2 at each stage in order to get a better dynamic range.

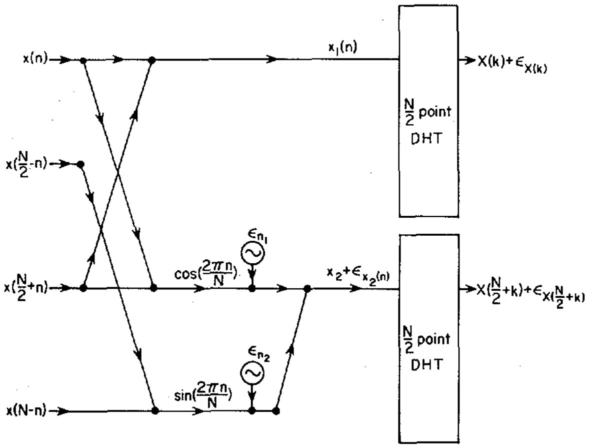


Fig. 5. Statistical model for fixed-point roundoff noise in a flow graph of the decimation-in-frequency decomposition of an N-point DFT computation into two $N/2$-point DHT computations using the DF1 algorithm.

$$\epsilon_{x_2(n)} = \begin{cases} \epsilon_{n_1} + \epsilon_{n_2} & 0 < n < \frac{N}{2}, \quad n \neq \frac{N}{4} \\ 0 & n = 0, \frac{N}{4} \end{cases} \tag{3.5b}$$

where ϵ_{n_i} denotes the roundoff error due to multiplications in fixed-point arithmetic. Its variance is denoted by σ_ϵ^2 and was defined earlier. The variance of error at the input $x_2(n)$ is given by

$$\text{Var}\left[\epsilon_{x_2(n)}\right] = \begin{cases} 2\sigma_\epsilon^2 & 0 < n < \frac{N}{2}, \quad n \neq \frac{N}{4} \\ 0 & n = 0, \frac{N}{4}. \end{cases}$$

If $\epsilon_{x_2(n)}$ was the only source of error in $X_2(k)$ (i.e., the $(N/2)$-point DHT was done with infinite precision), the output noise variance at $X_2(k)$ due to the error in the input would have been given by

$$\text{Var}'\left[\epsilon_{X_2(k)}\right] = \sum_{n=0}^{N/2-1} \text{Var}\left[\epsilon_{x_2(n)}\right] \cdot \text{cas}^2\left(\frac{2\pi kn}{N/2}\right) = (N-4)\,\sigma_\epsilon^2. \tag{3.6}$$

The error in $X_2(k)$ can be considered to be due to error in $x_2(n)$ and due to the errors introduced in the $(N/2)$-point DHT computation of $x_2(n)$. The variance of error due to the noise in $x_2(n)$ is given in (3.6), and the variance of error due to computations in the $(N/2)$-point DHT is $\sigma^2((N/2), k)$. Since these two errors are independent of each other, in order to find the variance of $\epsilon_{X_2(k)}$ we add these two variances to obtain

$$\text{Var}\left[\epsilon_{X_2(k)}\right] = (N-4)\,\sigma_\epsilon^2 + \sigma^2\left(\frac{N}{2}, k\right). \tag{3.7a}$$

Since no error has been introduced in computing $x_1(n)$, the variance of $\epsilon_{X_1(k)}$ is only due to the $(N/2)$-point DHT computation and, therefore,

$$\text{Var}\left[\epsilon_{X_1(k)}\right] = \sigma^2\left(\frac{N}{2}, k\right). \tag{3.7b}$$

At the output of the algorithm we have

$$\epsilon_{X(2k)} = \epsilon_{X_1(k)} \tag{3.8a}$$

$$\epsilon_{X(2k+1)} = \epsilon_{X_2}(k). \tag{3.8b}$$

Combining (3.7) and (3.8) we get

$$\sigma^2(N, k) = \begin{cases} \sigma^2\left(\frac{N}{2}, k\right) & k \text{ even} \\ \sigma^2\left(\frac{N}{2}, k\right) + (N-4)\,\sigma_\epsilon^2 & k \text{ odd.} \end{cases} \tag{3.9}$$

Fig. 6(a) shows the distribution of the variance of error for 256-point sequences using (3.9).

If we average $\sigma^2(N, k)$ of (3.9) over the frequency points, we can obtain a difference equation for the average output noise variance. Specifically,

$$\langle \sigma^2(2^\nu, k) \rangle = \langle \sigma^2(2^{\nu-1}, k) \rangle + \tfrac{1}{2}(2^\nu - 4)\,\sigma_\epsilon^2 \tag{3.10}$$

where $\langle \cdot \rangle$ is used to show averaging over frequency points. Since 2- and 4-point DHT's are essentially error free,

$$\langle \sigma^2(2, k) \rangle = \langle \sigma^2(4, k) \rangle = 0.$$

Solving (3.10) for $\nu > 2$, we obtain the following closed-form expression for the mean noise variance:

$$\langle \sigma^2(2^\nu, k) \rangle = (2^\nu - 2\nu)\,\sigma_\epsilon^2. \tag{3.11}$$

Equation (3.11) suggests that the output noise variance of an N-point sequence is approximately proportional to N as opposed to $2N$ which was derived in our simplified analysis of (3.4). For large values of N, the output noise variance for the decimation-in-frequency algorithm is double the corresponding quantity for the decimation-in-time algorithm.

To obtain a formula for the output noise-to-signal ratio, we next consider the dynamic range constraint. Similar to the DT1 algorithm, we can show that by keeping the magnitude of the input signal below $1/N$, we can avoid overflow. Assuming the input is white with uniform probability distribution function in the interval $(-1/N, 1/N)$, the output signal variance is $(1/3N)$. Fig. 4 shows the average noise-to-signal ratio for the decimation-in-time and frequency algorithms using (3.3) and (3.11). As anticipated, the DT1 algorithm has more favorable error properties than the DF1 algorithm. These results will be verified experimentally in Section III-C.

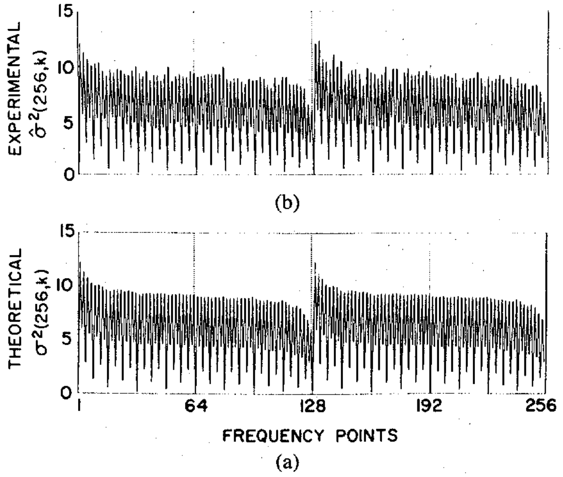


Fig. 6. Distribution of the output noise variance for 256-point sequences for fixed-point realization of the DF1 algorithm: (a) theoretical; (b) experimental.

C. *Experimental Verification of Fixed-Point Error Properties of the DT1 and DF1 Algorithms*

The experiments for roundoff noise analysis consisted of two parts. In the first part, zero mean white input sequences were generated using a random number generator routine. The probability density function (pdf) for each point of these sequences was uniformly distributed around zero. The width was chosen in such a way as to guarantee no overflows in the output or in intermediate computations.

In the second part, the generated test sequences were transformed twice; once using rounded fixed-point arithmetic with a word length of 15 bits (excluding the sign bit); the second time using double precision floating-point arithmetic with 55 bits of mantissa (excluding the sign and the hidden bit). The double precision computation was assumed to be exact in comparison to the fixed-point computation.

The above procedure was repeated with 1000 independent input sequences in order to obtain a stable estimate of the variance of error for each frequency point of an N-point transform. The estimator used is the sample mean $\hat{m}_\epsilon(N, k)$ and sample variance $\hat{\sigma}^2(N, k)$ at each frequency. In order to obtain an estimate of the mean output noise variance, we average $\hat{\sigma}^2(N, k)$ over the frequency points k. That is,

$$\hat{\sigma}_E^2 = \frac{1}{N} \sum_{k=0}^{N-1} \hat{\sigma}^2(N, k). \tag{3.12}$$

Since the input signal is zero mean and white, and its probability distribution function is uniformly distributed, we can easily find the output signal variance. Thus, we can obtain an experimental estimate of the mean noise-to-signal ratio for various algorithms using (3.12).

Fig. 4 shows the experimental and the theoretical noise-to-signal ratio for the fixed-point realization of the DT1, MDT1, and DF1 algorithms. Clearly, there is excellent agreement between the predicted and actual values of noise-to-signal ratio. We can fit the following equations to the data shown in Fig. 4.

$$\sqrt{\frac{\sigma_E^2}{2^{-2b}\sigma_{\text{out}}^2}} = 0.15N^{1.10} \qquad \text{DT1} \tag{3.13a}$$

$$\sqrt{\frac{\sigma_E^2}{2^{-2b}\sigma_{\text{out}}^2}} = 0.28N^{1.08} \qquad \text{DF1.} \tag{3.13b}$$

Although the multiplication count for the MDT1 algorithm is two-thirds that of the DT1 algorithm, as shown in Fig.4, the noise-to-signal ratios for the two algorithms are almost identical.

Figs. 3(b) and 6(b) show experimental verification of (3.3) and (3.9) for 256-point sequences using the DT1 and DF1 algorithms, respectively. The more accurate analysis of the MDT1 algorithm and its corresponding theoretical and experimental distribution of output noise variance are included in [18].

IV. Roundoff Error Analysis of Floating-Point Implementation of DT1 and DF1 and MDT1 Algorithms

In this section, the error properties of a floating-point implementation of the DT1, MDT1, and DF1 algorithms are discussed. We shall consider floating-point numbers with mantissas represented as $(b + 1)$-bit binary fractions. With x denoting the exact result of an addition or multiplication, the rounded value of x is $x(1 + \epsilon)$, where ϵ is the relative error. For the case of two's complement rounding, ϵ is assumed to be in the interval $(-2^{-b}, 2^{-b})$ with variance

$$\sigma_\epsilon^2 = \alpha 2^{-2b} \tag{4.1}$$

where, for a given algorithm, α is a constant which depends on the number of multiplies and additions and the order in which they are performed in that algorithm [1]. The factor α can be determined by matching the theoretical and experimental noise-to-signal ratio curves.

In Section IV-A and B, we will derive the statistical error properties of floating-point implementations of the DT1, MDT1, and DF1 algorithms. Unlike the fixed-point case, the error characteristics of the floating-point implementation of the MDT1 algorithm are different from those of the DT1 algorithm. In Section IV-C, the experimental results are compared to the theoretical predictions of Section IV-A and B.

A. *Roundoff Noise Analysis of the DT1 Algorithm Using Floating-Point Arithmetic*

The simplified roundoff noise analysis of the DT1 algorithm using floating-point arithmetic is similar to that of the FFT. Using the simplified analysis, it can be shown that the variance of error at the ouptut of an N-point DHT is given by $2\nu N\sigma_\epsilon^2\sigma_{\text{in}}^2$, where $\nu = \log_2 N$ and the input is assumed to be white with variance σ_{in}^2 [11], [18].

In order to derive the above result, some details have been neglected. Equal variance noise sources have been associated with all multipliers, including when the coefficients are zero or one. In order to take care of the special cases mentioned above, an alternative way of finding the output noise variance is suggested. Recall that an N-point DHT can be decomposed into two $(N/2)$-point DHT's, as shown in Fig. 7. By inspection of Fig. 7, the output noise for the kth point of an N-point transform denoted by $\epsilon_{X(k)}$ can be written in terms of $\epsilon_{X_1(k)}$, $\epsilon_{X_2(k)}$, and $\epsilon_{X_2((N/2)-k)}$. Since $X_1(k)$ and $X_2(k)$ are $N/2$-point DHT's of white sequences, the variance of $\epsilon_{X_1(k)}$ and $\epsilon_{X_2(k)}$ is given by $\sigma^2((N/2), k)$, and the variance of $\epsilon_{X_2((N/2)-k)}$ is given by $\sigma^2((N/2), (N/2) - k)$. Using this, and by inspection of Fig. 7, we get

$$\sigma^2(N, k) = \begin{cases} \sigma^2\left(\dfrac{N}{2}, k\right) + \cos^2\left(\dfrac{2\pi k}{N}\right)\sigma^2\left(\dfrac{N}{2}, k\right) & 0 < k < \dfrac{N}{2}, \quad k \neq \dfrac{N}{4} \\ \quad + \sin^2\left(\dfrac{2\pi k}{N}\right)\sigma^2\left(\dfrac{N}{2}, \dfrac{N}{2} - k\right) & \\ \quad + 2N\sigma_\epsilon^2\sigma_{\text{in}}^2 & \\ 2\sigma^2\left(\dfrac{N}{2}, k\right) + N\sigma_\epsilon^2\sigma_{\text{in}}^2 & k = 0, \dfrac{N}{4} \end{cases} \tag{4.2a}$$

$$\sigma^2(N, k) = \sigma^2\left(N, k - \frac{N}{2}\right) \qquad \frac{N}{2} \le k < N. \tag{4.2b}$$

Note that for $k = 0, N/4, N/2, (3N)/4$, the coefficients of the butterflies of Fig. 7 become 0 or 1. By taking care of these special cases, we have incorporated the noiselessness of these multipliers in our theoretical predictions. Fig. 8 shows the average output noise-to-signal ratio using the analysis shown in (4.2); and Fig. 9(a) shows the distribution of variance of error among the frequency points of a white 256-point sequence.

B. Roundoff Noise Analysis of the MDT1 Algorithm Using Floating-Point Arithmetic

Fig. 10 shows the decomposition used for an N-point sequence $x(n)$ with the noise sources injected for the multipliers and the adders. Referring to Fig. 10, the following expressions for $w_1(k)$ and $w_2(k)$ can be obtained:

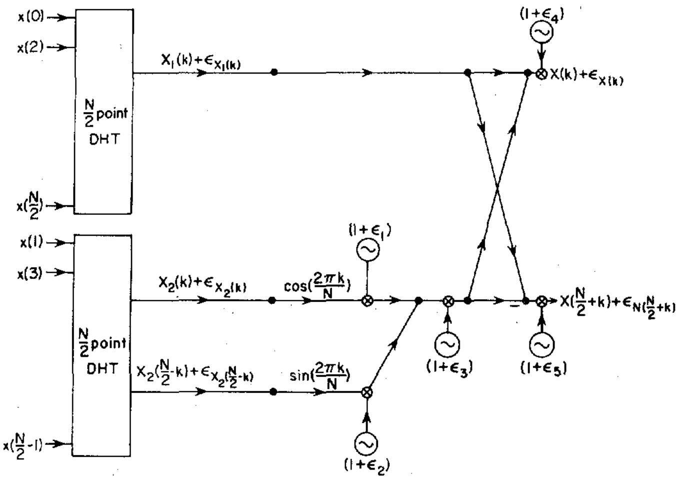


Fig. 7. Statistical model for floating-point roundoff noise in a flow graph of the decimation-in-time decomposition of an N-point DHT computation into two $N/2$-point DHT computations using the DT1 algorithm.

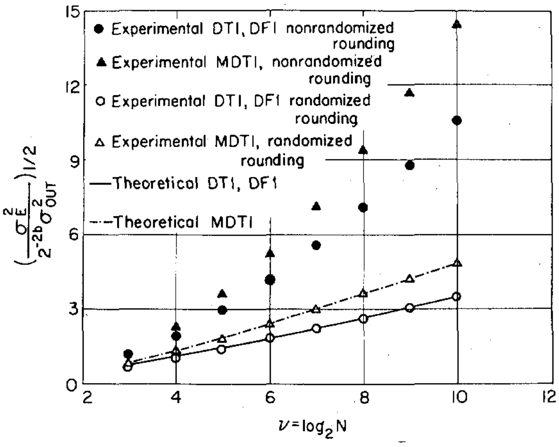


Fig. 8. Output noise-to-signal ratio for floating-point realization of DT1, MDT1, and DF1 algorithms.

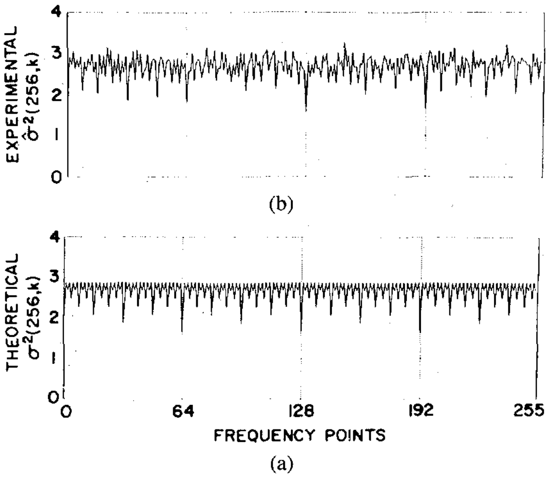


Fig. 9. Distribution of the output noise variance of 256-point white sequences for floating-point implementation of the DT1 algorithm: (a) theoretical; (b) experimental.

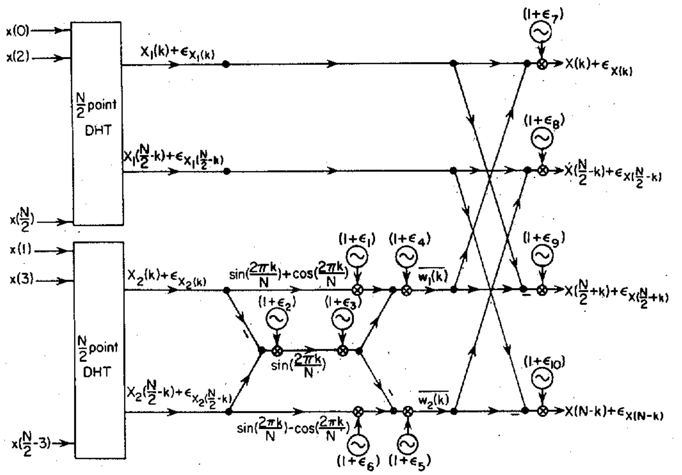

x(0)
x(2)
$x(\frac{N}{2})$
x(1)
x(3)
$x(\frac{N}{2}-3)$
$\frac{N}{2}$ point DHT
$\frac{N}{2}$ point DHT
$X_1(k)+\epsilon_{X_1(k)}$
$X_1(\frac{N}{2}-k)+\epsilon_{X_1(\frac{N}{2}-k)}$
$X_2(k)+\epsilon_{X_2(k)}$
$X_2(\frac{N}{2}-k)+\epsilon_{X_2(\frac{N}{2}-k)}$
$\sin(\frac{2\pi k}{N})+\cos(\frac{2\pi k}{N})$
$\sin(\frac{2\pi k}{N})$
$\sin(\frac{2\pi k}{N})-\cos(\frac{2\pi k}{N})$
$(1+\epsilon_1)$
$(1+\epsilon_2)$
$(1+\epsilon_3)$
$(1+\epsilon_4)$
$(1+\epsilon_5)$
$(1+\epsilon_6)$
$(1+\epsilon_7)$
$(1+\epsilon_8)$
$(1+\epsilon_9)$
$(1+\epsilon_{10})$
$\overline{w_1(k)}$
$\overline{w_2(k)}$
$X(k)+\epsilon_{X(k)}$
$X(\frac{N}{2}-k)+\epsilon_{X(\frac{N}{2}-k)}$
$X(\frac{N}{2}+k)+\epsilon_{X(\frac{N}{2}+k)}$
$X(N-k)+\epsilon_{X(N-k)}$

Fig. 10. Statistical model for floating-point roundoff noise in a flow graph of the decimation-in-time decomposition of an N-point DHT computation into two $N/2$-point DHT computations using the MDT1 algorithm.

$$\begin{aligned}\overline{w_1(k)} = (1+\epsilon_4)\Bigg\{&\left[\cos\left(\frac{2\pi k}{N}\right)+\sin\left(\frac{2\pi k}{N}\right)\right]\\ &\cdot\left[X_2(k)+\epsilon_{X_2(k)}\right](1+\epsilon_1)\\ &+(1+\epsilon_3)(1+\epsilon_2)\sin\left(\frac{2\pi k}{N}\right)\\ &\cdot\left[\left(X_2\left(\frac{N}{2}-k\right)+\epsilon_{X_2((N/2)-k)}\right)\right.\\ &\left.-\left(X_2(k)+\epsilon_{X_2(k)}\right)\right]\Bigg\}\end{aligned} \tag{4.3}$$

$$\begin{aligned}\overline{w_2(k)} = (1+\epsilon_5)\Bigg\{&\left[\sin\left(\frac{2\pi k}{N}\right)-\cos\left(\frac{2\pi k}{N}\right)\right]\\ &\cdot\left(X_2\left(\frac{N}{2}-k\right)+\epsilon_{X_2((N/2)-k)}\right)(1+\epsilon_6)\\ &-(1+\epsilon_3)(1+\epsilon_2)\sin\left(\frac{2\pi k}{N}\right)\\ &\cdot\left[\left(X_2\left(\frac{N}{2}-k\right)+\epsilon_{X_2((N/2)-k)}\right)\right.\\ &\left.-\left(X_2(k)+\epsilon_{X_2(k)}\right)\right]\Bigg\}.\end{aligned}$$

The computed output points are given by

$$\overline{X(k)} = \left[X_1(k)+\epsilon_{X_1(k)}+\overline{w_1(k)}\right](1+\epsilon_7) \tag{4.4a}$$

$$X\left(\frac{N}{2}-k\right) = \left[X_1\left(\frac{N}{2}-k\right)\epsilon_{X_1((N/2)-k)} + \overline{w_2(k)}\right](1+\epsilon_8) \tag{4.4b}$$

$$X\left(\frac{N}{2}-k\right) = \left[X_1(k)+\epsilon_{X_1(k)}-\overline{w_1(k)}\right](1+\epsilon_9) \tag{4.4c}$$

$$\overline{X(N-k)} = \left[X_1\left(\frac{N}{2}-k\right)+\epsilon_{X_1((N/2)-k)} - \overline{w_2(k)}\right](1+\epsilon_{10}). \tag{4.4d}$$

Since $X_1(k)$ and $X_2(k)$ are $N/2$-point DHT's of white sequences, the variance of $\epsilon_{X_1(k)}$ and $\epsilon_{X_2(k)}$ is simply $\sigma^2((N/2), k)$. Using (4.3) and (4.4), the variance of error for an N-point sequence is given by

$$\sigma^2(N,k) = \begin{cases} \left[2N + N\sin\left(\frac{2\pi k}{N}\right)\left(\cos\left(\frac{2\pi k}{N}\right) + 2\sin\left(\frac{2\pi k}{N}\right)\right)\right]\sigma_\epsilon^2\sigma_{\text{in}}^2 \\ \quad + \left(1+\cos^2\left(\frac{2k}{N}\right)\right)\sigma^2\left(\frac{N}{2},k\right) \\ \quad + \sin^2\left(\frac{2k}{N}\right)\sigma^2\left(\frac{N}{2},\frac{N}{2}-k\right) \\ \quad + \sin\left(\frac{4\pi k}{N}\right)\text{cov}_\epsilon\left(\frac{N}{2},k\right) & 0 < k < \frac{N}{4} \\ 2\sigma^2\left(\frac{N}{2},k\right) + N\sigma_\epsilon^2\sigma_{\text{in}}^2 & k = 0 \\ 2\sigma^2\left(\frac{N}{2},k\right) + 2.5N\,\sigma_\epsilon^2\sigma_{\text{in}}^2 & k = \frac{N}{4} \end{cases} \tag{4.5a}$$

$$\sigma^2\left(N,\frac{N}{2}-k\right) = \begin{cases} \left[2N + N\sin\left(\frac{2\pi k}{N}\right)\left(2\sin\left(\frac{2\pi k}{N}\right) - \cos\left(\frac{2\pi k}{N}\right)\right)\right]\sigma_\epsilon^2\sigma_{\text{in}}^2 \\ \quad + \left(1+\cos^2\left(\frac{2\pi k}{N}\right)\right)\sigma^2\left(\frac{N}{2},\frac{N}{2}-k\right) \\ \quad \cdot\sin^2\left(\frac{2\pi k}{N}\right)\sigma^2\left(\frac{N}{2},k\right) \\ \quad - \sin\left(\frac{4\pi k}{N}\right)\text{cov}_\epsilon\left(\frac{N}{2},k\right) & 0 < k < \frac{N}{4},\quad k \neq \frac{N}{8} \\ 1.5\sigma^2\left(\frac{N}{2},\frac{N}{2}-k\right) + \frac{1}{2}\sigma^2\left(\frac{N}{2},k\right) \\ \quad + 1.5N\sigma_\epsilon^2\sigma_{\text{in}}^2 - \text{cov}_\epsilon\left(\frac{N}{2},k\right) & k = \frac{N}{8} \end{cases} \tag{4.5b}$$

$$\sigma^2(N, k) = \sigma^2\left(N, k - \frac{N}{2}\right) \qquad \frac{N}{2} \le k < N \qquad (4.5c)$$

where $\text{cov}_\epsilon(N, k)$ denotes the covariance of the error between the kth and $(N - k)$th point of the DHT of an N-point white sequence. Note that the special cases of $k = 0, N/4, N/8$ in (4.5) take account of noiseless multiplications by unity or zero. Equation (4.5) states that $\sigma^2(N, k)$ can be obtained from $\sigma^2(N/2, k)$ and $\text{cov}_\epsilon(N/2, k)$. In order to complete the recursion, we have to be able to obtain $\text{cov}_\epsilon(N, k)$ from $\sigma^2(N/2, k)$ and $\text{cov}_\epsilon(N/2, k)$. This can be done by finding the error in $X(k)$ and $X(N - k)$, and by evaluating the expected value of their products. The final result is

$$\text{cov}_\epsilon(N, k) = \begin{cases} \cos\left(\frac{2\pi k}{N}\right)\sin\left(\frac{2\pi k}{N}\right) \\ \quad \cdot \left[\sigma^2\left(\frac{N}{2}, \frac{N}{2} - k\right) - \sigma^2\left(\frac{N}{2}, k\right)\right] \\ \qquad 0 < k < \frac{N}{4}, \quad k \ne \frac{N}{8} \\ +2\cos^2\left(\frac{2\pi k}{N}\right)\text{cov}_\epsilon\left(\frac{N}{2}, k\right) \\ +\,2N\sin^2\left(\frac{2\pi k}{N}\right)\sigma_\epsilon^2\sigma_{\text{in}}^2 \\ 2\,\text{cov}_\epsilon\left(\frac{N}{2}, k\right) - N\sigma_\epsilon^2\sigma_{\text{in}}^2 \qquad k = \frac{N}{4} \end{cases} \qquad (4.6a)$$

$$\begin{aligned} \text{cov}_\epsilon(N, k) &= \text{cov}_\epsilon\left(N, \frac{N}{2} - k\right) \\ &= \text{cov}_\epsilon\left(N, \frac{N}{2} + k\right) \\ &= \text{cov}_\epsilon(N, N - k) \\ &\qquad 0 < k \le \frac{N}{4}. \end{aligned} \qquad (4.6b)$$

In the above equation, $\text{cov}_\epsilon(N, k)$ is nonzero as a result of the fact that the error sources ϵ_2 and ϵ_3 of Fig. 10 both contribute to the error at the kth and $(N - k)$th output points. This is clearly not the case in the DT1 algorithm. In fact, ignoring the $\text{cov}_\epsilon(N/2, k)$ term, (4.5) and (4.2) look somewhat similar.

Fig. 11(a) shows the distribution of output noise variance for 256-point white sequences using the MDT1 algorithm. Fig. 8 shows the noise-to-signal ratio of the DHT's of white sequences using the MDT1 and the DT1 algorithms. Although DT1 requries more multiplications than MDT1, its noise-to-signal ratio is lower than for the MDT1 algorithm. The experimental results supporting the theoretical predictions in Fig. 8 and 11(a) are presented in Section IV-C.

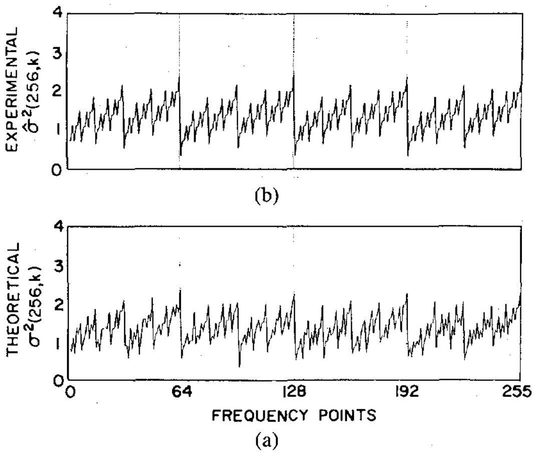


Fig. 11. Distribution of the output noise variance of 256-point white sequences for floating-point implementation of the MDT1 algorithm: (a) theoretical; (b) experimental.

The roundoff noise analysis of the DF1 algorithm using floating-point arithmetic is similar to that of the DT1 algorithm. Using the same approach as Section III-B, the output noise-to-signal ratio for an $N = 2^\nu$-point white input sequence can be shown to be $2\nu\sigma_\epsilon^2$ [18]. The more detailed analysis of the DF1 algorithm is also included in [18].

C. The Experimental Verification of Roundoff Noise for the Floating-Point Implementation of the DT1, MDT1, and DF1 Algorithms

The experiments for roundoff analysis using floating-point arithmetic paralleled those of fixed-point arithmetic. Specifically, zero mean white input test sequences were generated and transformed twice; once using floating-point arithmetic with 23 bits of mantissa (excluding the hidden bit and the sign bit), and the second time using double precision floating-point arithmetic with 55 bits of mantissa (excluding the sign bit and the hidden bit). The double precision computation was assumed to be exact in comparison to the former one. This procedure was repeated 1000 times in order to obtain a stable estimate of the sample mean and sample variance at each frequency.

The convention used to round the results of floating-point additions and multiplications was as follows. The results were rounded to the closest binary number (for a b-bit mantissa). If a result of an addition or a multiplication was midway between two binary numbers, a situation which can occur frequently, randomized rounding was used, i.e., a random choice was made as to whether to round up or down. Always rounding up (or down), rather than randomly up or down, in this situation introduces a correlation between roundoff error and signal sign. This contradicts the assumption that roundoff errors are signal independent.

Fig. 8 shows the experimental and theoretical noise-to-signal ratio for the floating-point implementation of the DT1, MDT1, and DF1 algorithms, including randomized rounding and nonrandomized rounding. As expected, the theoretical curves match the experimental results only if randomized rounding is used.

We can fit the following equations to the data shown in Fig. 8:

$$\frac{\sigma_E^2}{\sigma_{\text{out}}^2 2^{-2b}} = 0.40\,\nu - 0.53 \qquad \text{DT1} \qquad (4.7a)$$

$$\frac{\sigma_E^2}{\sigma_{\text{out}}^2 2^{-2b}} = 0.40\,\nu - 0.58 \qquad \text{DF1} \qquad (4.7b)$$

$$\frac{\sigma_E^2}{\sigma_{\text{out}}^2 2^{-2b}} = 0.59\,\nu - 1.09 \qquad \text{MDT1.} \qquad (4.7c)$$

Unlike the fixed-point case, the noise-to-signal ratio for the MDT1 algorithm shown in Fig. 8 is higher than that of the DT1 algorithm. As explained earlier, this is due to the correlations between the error at the inputs of the butterflies. As is shown in Fig. 8, the results for the DF1 algorithm are almost identical to that of the DT1 algorithm; unlike the fixed-point case, the dynamic range issues do not exist in floating-point implementations.

Figs. 9 and 11 show the theoretical and experimental distribution of variance of error for the DT1 and MDT1 algorithms, respectively. A careful analysis demonstrates a nonuniform distribution of variance of error among the output frequency points of the DF1 algorithm. The distribution of variance of error for this algorithm is derived analytically and verified experimentally in [18].

V. Summary and Conclusions

Statistical models were used to predict the output noise-to-signal ratio in fixed- and floating-point computation of various DHT algorithms. In fixed-point implementation of the DT1, MDT1, and DF1, it was found that the output noise-to-signal ratio increases by approximately 1.1 bits per stage. The output noise variance for the decimation-in-frequency algorithm was twice that of the decimation-in-time algorithm.

For the floating-point implementation, the number of bits of rms noise-to-signal ratio for DT1 and DF1 increased as $\sqrt{\log_2 N}$, so that doubling the number of points produced a mild increase in the output noise. For example, a 1-bit increase corresponds to quadrupling the transform size. The rate of increase of the noise-to-signal ratio was slightly higher for the MDT1 algorithm than the DT1 and DF1 algorithms. This was due to the special structure of the butterflies of the MDT1 algorithm.

Furthermore, for the floating-point implementations, the theoretical predictions checked closely with the experiments only when randomized rounding was used. Nonrandomized rounding seemed to introduce enough correlation between roundoff noise and signal to make the experimental results deviate from the predictions of our model which assumed signal and noise to be uncorrelated.

Finally, it is worthwhile to compare the error properties of the DT1 algorithm to the decimation-in-time version of the FFT. Comparing Fig. 4 to Fig. 12 of [13], and Fig. 8 to Fig. 13(a) of [12], we reach the conclusion that fixed- and floating-point error characteristics of the two algorithms are very similar. The similar butterfly structure that these two algorithms share is the main reason behind the above result.

References

[1] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[2] R. V. Hartley, "A more symmetrical Fourier analysis applied tc transmission problems," *Proc. IRE*, vol. 30, pp. 144-150, Mar. 1942.

[3] R. N. Bracewell, "The discrete Hartley transform," *J. Opt. Soc. Amer.*, vol. 73, pp. 1832-1835, Dec. 1983.

[4] —, "The fast Hartley transform," *Proc. IEEE*, vol. 72, no. 8, pp. 1010-1018, Aug. 1984.

[5] Z. W. Wang, "Fast algorithms for the discrete *W* transform and foi the discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signai Processing*, vol. ASSP-32, pp. 803-816, Aug. 1984.

[6] H. V. Sorenson, D. L. Jones, C. S. Burrus, and M. T. Heideman, "On computing the discrete Hartley transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1231-1238, Oct. 1985.

[7] P. Duhamel and H. Hollmann, "Split radix FFT algorithm," *Electron. Lett.*, vol. 32, no. 4, pp. 750-762, Aug. 1984.

[8] H. V. Sorensen, M. T. Heideman, and C. S. Burrus, "On computing the split-radix FFT," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 152-156, Feb. 1986.

[9] C. S. Burrus, "Index mapping for multidimensional formulation of the DFT and convolution," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 239-242, June 1977.

[10] J. Makhoul, "A fast cosine transform in one- and two-dimensions," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 27-34, Feb. 1980.

[11] C. J. Weinstein, "Roundoff noise in floating-point fast Fourier transform computation," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 209-215, Sept. 1969.

[12] A. V. Oppenheim and C. J. Weinstein, "Effects of finite registei length in digital filtering and the fast Fourier transform," *Proc. IEEE*, vol. 60, pp. 957-978, Aug. 1972.

[13] C. J. Weinstein, "Quantization effects in digital filters," Ph.D. dissertation, Dep. Elec. Eng., M.I.T., Cambridge, MA, 1969.

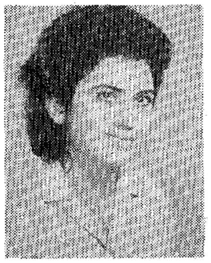
[14] W. H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004-1009, 1977.

[15] Z. Wang, "Reconsideration of 'A fast computational algorithm foi the discrete cosine transform,'" *IEEE Trans. Commun.*, vol. COM-31, pp. 121-123, 1983.

[16] L. R. Rabiner, R. W. Schafer, and C. M. Rader, "The chirp z-transform algorithm," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 86-92, June 1979.

[17] R. N. Bracewell, *The Fourier Transform and Its Applications*. New York: McGraw-Hill, 1975.

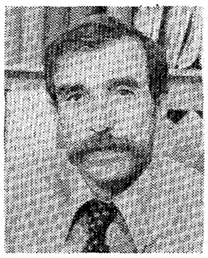
[18] A. Zakhor, "Error properties of Hartley transform algorithms," S.M. thesis, Dep. Elec. Eng. Comput. Sci., M.I.T., Cambridge, MA, June 1985.



Avideh Zakhor was born on May 11, 1963. She received the B.S. degree from the California Institute of Technology, Pasadena, and the S.M. degree from the Massachusetts Institute of Technology, Cambridge, both in electrical engineering, in 1983 and 1985, respectively. She is currently pursuing the Ph.D. degree in the Department of Electrical Engineering and Computer Science at M.I.T. under a Hertz fellowship. Her doctoral thesis is in the general area of reconstruction from partial information.

She has been a consultant to Rand Corporation, Santa Monica, CA, since 1984, where she conducted feasibility studies on adaptive beamforming algorithms in thinned arrays and various constant envelope digital phase modulation schemes. During the Summer of 1985 she was with Technology Service Corporation, Santa Monica, CA, working on various aspects of a spread spectrum ECCM radar.

Ms. Zakhor has been a Hertz Fellow, General Motors Scholar, and Atlantic College Scholar, and received the Henry Ford Engineering Award and Caltech Prize in 1983. She is also a member of Tau Beta Pi, Sigma Xi, and the ASSP Society of the IEEE.



Alan V. Oppenheim (S'57-M'65-SM'71-F'77) received the S.B. and S.M. degrees in 1961 and the Sc.D. degree in 1964, all in electrical engineering, from the Massachusetts Institute of Technology, Cambridge.

In 1964 he joined the Faculty at M.I.T. where he is currently a Professor of Electrical Engineering and Computer Science. From 1978 to 1980 he was Associate Head of the Data Systems Division at M.I.T. Lincoln Laboratory. Since 1977 he has also been a Guest Investigator at the Woods Hole Oceanographic Institution, Woods Hole, MA. His research interests are in the general area of signal processing and its application to speech, image, and seismic data processing. A current area of emphasis is knowledge-based signal processing. He is coauthor of the text *Digital Signal Processing*, coauthor of the text *Signal and Systems*, Editor of the book *Applications of Digital Signal Processing*, Co-Editor of the book *Advanced Topics in Signal Processing*, and Editor of the reprint book *Papers on Digital Signal Processing*. He has been Editor of the Prentice-Hall *Series on Signal Processing* since 1975. He is also author of two video tape lecture series and study guides on signal processing.

Dr. Oppenheim has been a Guggenheim Fellow, a Sackler Fellow, and has held the Cecil H. Green Distinguished Chair in Electrical Engineering and Computer Science. He has also received a number of awards for outstanding research and teaching, and is an elected member of the National Academy of Engineering. He is also a member of Tau Beta Pi, Eta Kappa Nu, and Sigma Xi.