After his military service he joined N. V. Philips' Gloeilampenfabrieken, Eindhoven, where he worked in the fields of electronic design of broadcast radio sets, integrated circuits and optical character recognition. Since 1970 he has been at the Philips Research Laboratories, where he is now working on digital communication systems.

Mr. Roza is a member of the Netherlands Electronics and Radio Society.

Peter W. Millenaar was born in Jutphaas, The Netherlands, on September 6, 1946. He received the degree in electrical engineering at the School of Technology, Rotterdam, in 1967.

In 1967 he joined the Philips Research Laboratories, Eindhoven, The Netherlands, where he worked in the field of data transmission and on the synchronization-aspects of a videophone system. Since 1972 he has been engaged in system design and electronics of high speed digital transmission.

Concise Papers

A Fast Computational Algorithm for the Discrete Cosine Transform

WEN-HSIUNG CHEN, C. HARRISON SMITH, AND S. C. FRALICK

Abstract—A Fast Discrete Cosine Transform algorithm has been developed which provides a factor of six improvement in computational complexity when compared to conventional Discrete Cosine Transform algorithms using the Fast Fourier Transform. The algorithm is derived in the form of matrices and illustrated by a signal-flow graph, which may be readily translated to hardware or software implementations.

INTRODUCTION

The Discrete Cosine Transform (DCT) has been successfully applied to the coding of high resolution imagery [1-5]. The conventional method of implementing the DCT utilized a double size Fast Fourier Transform (FFT) algorithm employing complex arithmetic throughout the computation [1]. Use of the DCT in a wide variety of applications has not been as extensive as its properties would imply due to the lack of an efficient algorithm. This report describes a more efficient algorithm involving only real operations for computing the Fast Discrete Cosine Transform (FDCT) of a set of N points. The algorithm can be extended to any desired value of $N = 2^m$, $m \geqslant 2$. The generalization consists of alternating cosine/sine butterfly matrices with binary matrices to reorder the matrix elements to a form which preserves a recognizable bit-reversed pattern at every other node. The generalization is not unique—several alternate methods have been discovered—but the method described herein appears to be the simplest to interpret. It is not necessarily the most efficient FDCT which could be constructed but represents one technique for methodical extension. The method takes $(3N/2)(\log_2 N - 1) + 2$ real additions and $N \log_2 N - 3N/2 + 4$ real multiplications: this is approximately six times as fast as the conventional approach using a double size FFT.

Paper approved by the Editor for Communication Theory of the IEEE Communications Society for publication without oral presentation. Manuscript received January 24, 1977.

W.-H. Chen was with the Western Development Laboratories Division, Ford Aerospace and Communications Corporation, Palo Alto, CA 94303. He is now with Compression Laboratories, Inc., Campbell, CA 95008.

C. H. Smith is with the Western Development Laboratories Division, Ford Aerospace and Communications Corporation, Palo Alto, CA 94303.

S. C. Fralick was with the Western Development Laboratories Division, Ford Aerospace and Communications Corporation, Palo Alto, CA 94303. He is now with Comtech Advanced Systems, Inc., Sunnyvale, CA 94086.

DISCRETE COSINE TRANSFORM

The discrete cosine transform of a discrete function $f(j)$, $j = 0, 1, \cdots, N-1$ is defined as [1]

$$F(k) = \frac{2c(k)}{N} \sum_{j=0}^{N-1} f(j) \cos\left[\frac{(2j+1)k\pi}{2N}\right]; \qquad k = 0, 1, \cdots, N-1 \tag{1}$$

and the inverse transform is

$$f(j) = \sum_{k=0}^{N-1} c(k)F(k) \cos\left[\frac{(2j+1)k\pi}{2N}\right]; \qquad j = 0, 1, \cdots, N-1 \tag{2}$$

where

$$c(k) = \frac{1}{\sqrt{2}} \quad \text{for } k = 0$$
$$= 1 \quad \text{for } k = 1, 2, \cdots, N-1.$$

The transform possesses a high energy compaction property which is superior to any known transform with a fast computational algorithm. [1-5] The transform also possesses a circular convolution-multiplication relationship which can readily be used in linear system theory. [6]

A FAST COMPUTATIONAL ALGORITHM

The discrete cosine transform of an $N \times 1$ data vector $[f]$ can be expressed in a matrix form as

$$[F] = \frac{2}{N}[A_N][f] \tag{3}$$

where $[A_N] = [c(k) \cos (2j+1)k\pi/2N]$; $j, k = 0, 1, \cdots, (N-1)$ as defined in equation (1) and $[F]$ is the $N \times 1$ transformed vector. The fast computational algorithm to be presented here is based upon the matrix decomposition of the $[A_N]$ matrix. As shown below, this matrix can first be written into the following recursive form:

$$[A_N] = [P_N]\left[\begin{array}{c|c} A_{N/2} & 0 \\ \hline 0 & R_{N/2} \end{array}\right][B_N] \tag{4}$$

where $[B_N]$ is defined in equation (7),

$$[R_{N/2}] = \left[c(k) \cos \frac{(2j+1)(2k+1)\pi}{2N}\right]; \quad j, k = 0, 1, \cdots, \frac{N}{2} - 1$$

and $[P_N]$ is an $N \times N$ permutation matrix which permutes the transformed vector from a bit reversed order to a natural order. As in all unitary transforms the 2×2 DCT can be written as

$$[A_2] = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{5}$$

It can be seen from the recursive nature of equation (4) that $[A_2]$ can be extended into higher order matrices as long as there is a generalized method of decomposing the $[R_{N/2}]$ matrix.

The following discussion presents one systematic way of decomposing the $[R_{N/2}]$ matrix. It is emphasized that this method of decomposition is not unique and is not optimum. Several methods have been found which require fewer computational steps but with no apparent generalization to larger sizes.

The $[R_{N/2}]$ matrix is decomposed into $(2 \log_2 N - 3)$ matrices in the following manner:

$$[R_{N/2}] = [M1][M2][M3][M4] \cdots [M(2 \log_2 N - 3)]. \tag{6}$$

The matrices are of four distinct types.

- Type 1: $[M1]$, the first matrix
- Type 2: $[M(2 \log_2 N - 3)]$, the last matrix
- Type 3: $[Mq]$, the remaining odd numbered matrices $[M3]$, $[M5]$, etc.
- Type 4: $[Mp]$, the even numbered matrices $[M2]$, $[M4]$, etc.

Before describing the four types of matrices in detail, the following definitions are provided for notational efficiency:

$$B_N = \begin{bmatrix} I_{N/2} & \bar{I}_{N/2} \\ \bar{I}_{N/2} & -I_{N/2} \end{bmatrix}$$

$$B_N^* = \begin{bmatrix} -I_{N/2} & \bar{I}_{N/2} \\ \bar{I}_{N/2} & I_{N/2} \end{bmatrix} \tag{7}$$

where

$[I_{N/2}]$ is an identity matrix of order $\frac{N}{2}$

$[\bar{I}_{N/2}]$ is the opposite diagonal identity matrix

$$[S_i^k] = \sin\frac{k\pi}{i}\,[I_{N/2i}]$$

$$[\bar{S}_i^k] = \sin\frac{k\pi}{i}\,[\bar{I}_{N/2i}]$$

$$[C_i^k] = \cos\frac{k\pi}{i}\,[I_{N/2i}]$$

$$[\bar{C}_i^k] = \cos\frac{k\pi}{i}\,[\bar{I}_{N/2i}]. \tag{8}$$

In equations (8) above, the identity matrices $[I_{N/2i}]$ specify the order of the diagonal sine or cosine matrices $[S_i^k]$, $[\bar{S}_i^k]$, $[C_i^k]$, $[\bar{C}_i^k]$ with the condition that $[I_{N/2i}] \equiv 1$ for $i > N/2$.

The four types of matrices may now be described in detail with reference to the right hand side of equation (9).

TYPE 1

The first matrix $[M1]$ is formed by concatenating $S_{2N}^{a_j}$ matrices (of order 1) along the upper left to middle of the main diagonal and $C_{2N}^{a_j}$ matrices along the middle to lower right. The opposite diagonal is formed by $\bar{C}_{2N}^{a_j}$ matrices along the upper right to middle and $\bar{S}_{2N}^{a_j}$ matrices along the middle to lower left. For this type matrix the values of a_j are the binary bit-reversed representation of $N/2 + j - 1$ for $j = 1, 2, \cdots, N/2$.

TYPE 2

The last matrix $[M(2 \log_2 N - 3)]$ is formed by concatenating $I_{N/8}$, $-C_4^1$, C_4^1, $I_{N/8}$ matrices along the upper left to lower right of the main diagonal and concatenating $O_{N/8}$, $\bar{C}_4^1$, $\bar{C}_4^1$, $O_{N/8}$ matrices along the upper right to lower left of the opposite diagonal.*

TYPE 3

The remaining odd matrices $[Mq]$ are formed by repeated concatenation of the matrix sequence $I_{N/2i}$, $-C_i^{k_j}$, $-S_i^{k_j}$ and $I_{N/2i}$ where $i = N/(2^{(q-1)/2})$ for $j = 1, 2, \cdots, i/8$ along the upper left to middle of the main diagonal and the matrix sequence $I_{N/2i}$, $C_i^{k_j}$, $S_i^{k_j}$ and $I_{N/2i}$ for $j = i/8 + 1, \cdots, i/4$ along the middle to lower right. The opposite diagonal is formed similarly, using the matrix sequence $O_{N/2i}$, $\bar{S}_i^{k_j}$, $-\bar{C}_i^{k_j}$, $O_{N/2i}$ along the upper right to middle and the matrix sequence $O_{N/2i}$, $-\bar{S}_i^{k_j}$, $\bar{C}_i^{k_j}$, $O_{N/2i}$ along the middle to lower left. Repeated concatenation of a matrix sequence along a diagonal is clearly illustrated in equation (9), where for clarity the k_j's have been replaced by b_j's and c_j's, etc., because the value of k_j depends on the matrix index q. For this type matrix, the values of the k_j are the binary bit-reversed variables $(i/4) + j - 1$.

* 0_j is a null matrix of order j.

$$
\left[R_{\frac{N}{2}}\right] = \begin{bmatrix}
S_{2N}^{a_1} & & & & & & & \bar{C}_{2N}^{a_1} \\
& S_{2N}^{a_2} & & & & & \bar{C}_{2N}^{a_2} & \\
& & \ddots & & & \cdot^{\cdot^{\cdot}} & & \\
& & & S_{2N}^{a_{N/4}} & \bar{C}_{2N}^{a_{N/4}} & & & \\
& & & -\bar{S}_{2N}^{a_{N/4+1}} & C_{2N}^{a_{N/4+1}} & & & \\
& & \cdot^{\cdot^{\cdot}} & & & \ddots & & \\
& -\bar{S}_{2N}^{a_{N/2-1}} & & & & & C_{2N}^{a_{N/2-1}} & \\
-\bar{S}_{2N}^{a_{N/2}} & & & & & & & C_{2N}^{a_{N/2}}
\end{bmatrix}
\begin{bmatrix}
B_2 & & & & & & \\
& B_2^* & & & & & \\
& & B_2 & & & & \\
& & & B_2^* & & & \\
& & & & \ddots & & \\
& & & & & B_2 & \\
& & & & & & B_2^*
\end{bmatrix}
$$

$$
\begin{bmatrix}
1 & & & & & & & & 0 \\
& -C_{N/2}^{b_1} & & & & & & \bar{S}_{N/2}^{b_1} & \\
& & -S_{N/2}^{b_1} & & & & -\bar{C}_{N/2}^{b_1} & & \\
& & & 1 & \ddots \quad \cdot^{\cdot^{\cdot}} & 0 & & & \\
& & & 0 & \cdot^{\cdot^{\cdot}} \quad \ddots & 1 & & & \\
& & -\bar{S}_{N/2}^{b_{N/8}} & & & & C_{N/2}^{b_{N/8}} & & \\
& \bar{C}_{N/2}^{b_{N/8}} & & & & & & S_{N/2}^{b_{N/8}} & \\
0 & & & & & & & & 1
\end{bmatrix}
\begin{bmatrix}
B_4 & & & & & & \\
& B_4^* & & & & & \\
& & B_4 & & & & \\
& & & B_4^* & & & \\
& & & & \ddots & & \\
& & & & & B_4 & \\
& & & & & & B_4^*
\end{bmatrix}
\begin{bmatrix}
I_2 & & & & & & & & 0 \\
& -C_{N/4}^{c_1} & & & & & & \bar{S}_{N/4}^{c_1} & \\
& & -S_{N/4}^{c_1} & & & & -\bar{C}_{N/4}^{c_1} & & \\
& & & I_2 & \ddots \quad \cdot^{\cdot^{\cdot}} & 0 & & & \\
& & & 0 & \cdot^{\cdot^{\cdot}} \quad \ddots & I_2 & & & \\
& & -\bar{S}_{N/4}^{c_{N/16}} & & & & C_{N/4}^{c_{N/16}} & & \\
& \bar{C}_{N/4}^{c_{N/16}} & & & & & & S_{N/4}^{c_{N/16}} & \\
0 & & & & & & & & I_2
\end{bmatrix}
$$

$$
\begin{bmatrix}
B_8 & & & & & & \\
& B_8^* & & & & & \\
& & B_8 & & & & \\
& & & B_8^* & & & \\
& & & & \ddots & & \\
& & & & & B_8 & \\
& & & & & & B_8^*
\end{bmatrix}
\cdots
\begin{bmatrix}
I_{\frac{N}{16}} & & & & & & & 0 \\
& -C_8^1 & & & & & \bar{S}_8^1 & \\
& & -S_8^1 & & & -\bar{C}_8^1 & & \\
& & & I_{\frac{N}{16}} & 0 & & & \\
& & & 0 & I_{\frac{N}{16}} & & & \\
& & -\bar{S}_8^3 & & & C_8^3 & & \\
& \bar{C}_8^3 & & & & & S_8^3 & \\
0 & & & & & & & I_{\frac{N}{16}}
\end{bmatrix}
$$

$$
\left[\begin{array}{c|c}
B_{\frac{N}{4}} & \\
\hline
& B_{\frac{N}{4}}^*
\end{array}\right]
\left[\begin{array}{c|c|c|c}
I_{\frac{N}{8}} & & & \\
\hline
& -C_4^1 & \bar{C}_4^1 & \\
\hline
& \bar{C}_4^1 & C_4^1 & \\
\hline
& & & I_{\frac{N}{8}}
\end{array}\right] \tag{9}
$$

TYPE 4

The even numbered matrices $[Mp]$ are binary matrices formed by alternating B_l and B_l^* matrices along the upper left to lower right of the main diagonal. The subscript l indicates the order of the B or B^* matrix, and takes on the value of $2^{P/2}$.

A specific example of $[R_{N/2}]$ for $N = 16$ is shown in Eq. (10).

$$
\left[R_{\frac{16}{2}}\right] =
\begin{bmatrix}
\sin\frac{\pi}{32} & & & & & & & \cos\frac{\pi}{32} \\
 & \sin\frac{9\pi}{32} & & & & & \cos\frac{9\pi}{32} & \\
 & & \sin\frac{5\pi}{32} & & & \cos\frac{5\pi}{32} & & \\
 & & & \sin\frac{13\pi}{32} & \cos\frac{13\pi}{32} & & & \\
 & & & -\sin\frac{3\pi}{32} & \cos\frac{3\pi}{32} & & & \\
 & & -\sin\frac{11\pi}{32} & & & \cos\frac{11\pi}{32} & & \\
 & -\sin\frac{7\pi}{32} & & & & & \cos\frac{7\pi}{32} & \\
-\sin\frac{15\pi}{32} & & & & & & & \cos\frac{15\pi}{32}
\end{bmatrix}
\begin{bmatrix}
1 & 1 & & & & & & \\
1 & -1 & & & & & & \\
 & & -1 & 1 & & & & \\
 & & 1 & 1 & & & & \\
 & & & & 1 & 1 & & \\
 & & & & 1 & -1 & & \\
 & & & & & & -1 & 1 \\
 & & & & & & 1 & 1
\end{bmatrix}
$$

$$
\cdot
\begin{bmatrix}
1 & & & & & & & 0 \\
 & -\cos\frac{\pi}{8} & & & & & \sin\frac{\pi}{8} & \\
 & & -\sin\frac{\pi}{8} & & & -\cos\frac{\pi}{8} & & \\
 & & & 1 & 0 & & & \\
 & & & 0 & 1 & & & \\
 & & -\sin\frac{3\pi}{8} & & & \cos\frac{3\pi}{8} & & \\
 & \cos\frac{3\pi}{8} & & & & & \cos\frac{3\pi}{8} & \\
0 & & & & & & & 1
\end{bmatrix}
\begin{bmatrix}
1 & & & 1 & & & & \\
 & 1 & 1 & & & & & \\
 & 1 & -1 & & & & & \\
1 & & & -1 & & & & \\
 & & & & -1 & & & 1 \\
 & & & & & -1 & 1 & \\
 & & & & & 1 & 1 & \\
 & & & & 1 & & & 1
\end{bmatrix}
$$

$$
\cdot
\begin{bmatrix}
1 & & & & & & & 0 \\
 & 1 & & & & & 0 & \\
 & & -\cos\frac{\pi}{4} & & & \cos\frac{\pi}{4} & & \\
 & & & -\cos\frac{\pi}{4} & \cos\frac{\pi}{4} & & & \\
 & & & \cos\frac{\pi}{4} & \cos\frac{\pi}{4} & & & \\
 & & \cos\frac{\pi}{4} & & & \cos\frac{\pi}{4} & & \\
 & 0 & & & & & 1 & \\
0 & & & & & & & 1
\end{bmatrix}
\tag{10}
$$

The computational steps required for $[F]$ of equation (3) can be found from equation (4) with the following recursive relations:

$$K_{A_N} = N + K_{A_{N/2}} + K_{R_{N/2}} \tag{11a}$$

$$K_{A_N}' = K_{A_{N/2}}' + K_{R_{N/2}}' \tag{11b}$$

where K_{A_i} and K_{A_i}' are the number of additions and multiplications for $[A_i]$, and K_{R_i} and K_{R_i}' are the number of additions and multiplications for $[R_i]$. The number of additions for $[R_{N/2}]$ can easily be determined from equation (9) by noting that $\log_2 N - 2$ stages of decomposed matrices are half occupied on the opposite diagonals and $\log_2 N -$ stages are fully occupied on both diagonals. Therefore

$$K_{R_{N/2}} = \frac{N}{4}(\log_2 N - 2) + \frac{N}{2}(\log_2 N - 1)$$

$$= \frac{3N}{4}\log_2 N - N \qquad N \geqslant 4. \tag{12a}$$

As for the number of multiplications, only the odd matrices consist of multiplicative terms. In these matrices the first matrix consists of N multipliers, the last matrix consists of $N/4$ multipliers, and the rest of the $\log_2 N - 3$ matrices each consists of $N/2$ multipliers. Thus

$$K_{R_{N/2}}' = N + \frac{N}{4} - \frac{N}{2}(\log_2 N - 3)$$

$$= \frac{N}{2}\log_2 N - \frac{N}{4}; \qquad N \geqslant 8. \tag{12b}$$

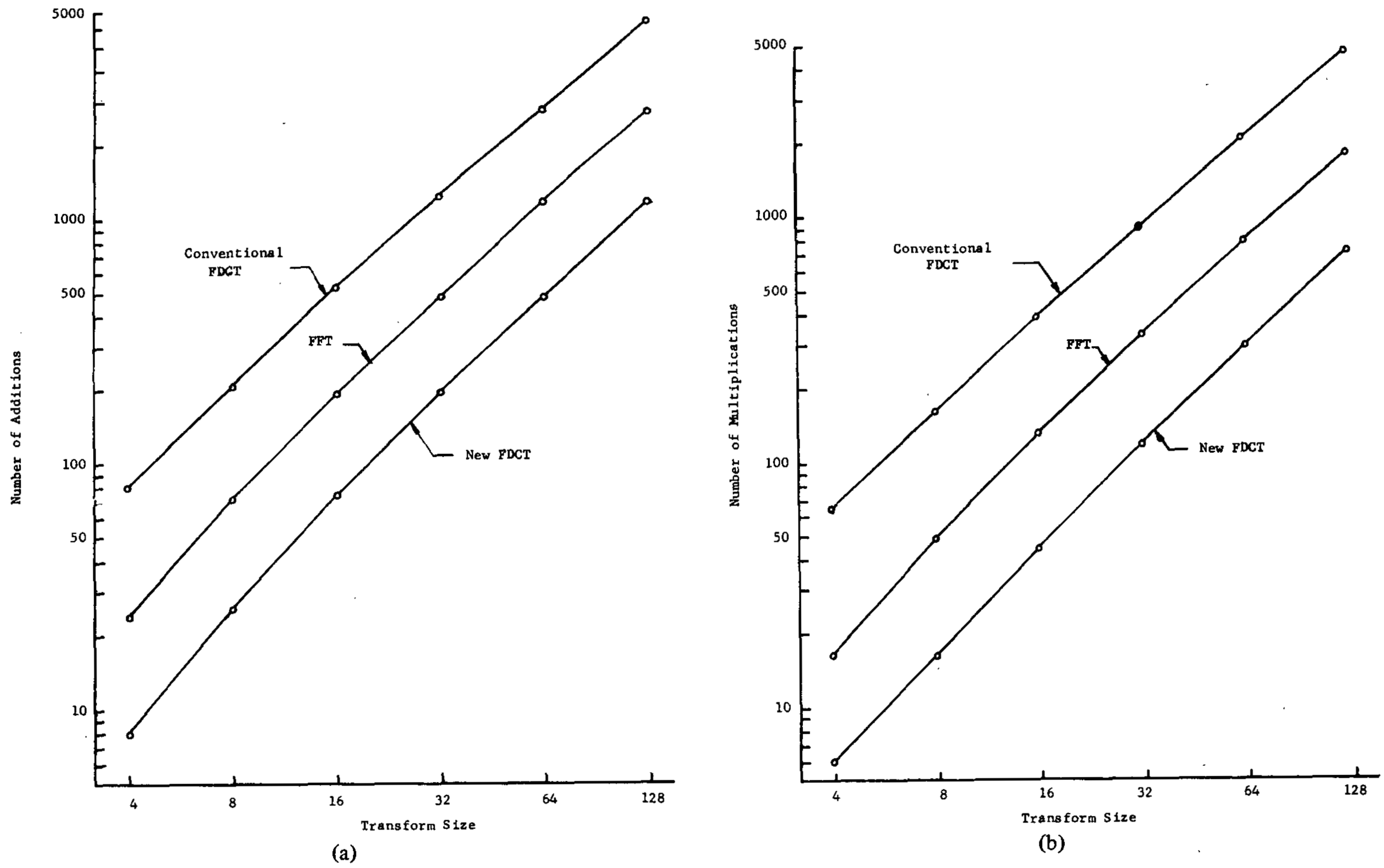

Figure 1. Comparison of Computational Steps for Conventional FDCT, FFT and FDCT. (a) Additions. (b) Multiplications.

For $N = 4$, $K_{R_2}{}' = 4$ since the only matrix involved is the first matrix. Now substituting equations (12a) and (12b) into equations (11a) and (11b), knowing (from equation 5) that

$$K_{A_2} = 2 \tag{13a}$$

$$K_{A_2}{}' = 2 \tag{13b}$$

one can obtain

$$K_{A_N} = \frac{3N}{2}(\log_2 N - 1) + 2 \tag{14a}$$

$$K_{A_N}{}' = N \log_2 N - \frac{3N}{2} + 4 \qquad N \geqslant 4. \tag{14b}$$

For purposes of comparison the conventional approach of computing the DCT utilizing an FFT takes $2N \log 2N$ complex additions and $N(\log 2N + 1)$ complex multiplications (which are equivalent to $6N \log 2N + 2N$ additions and $4N(\log 2N + 1)$ real multiplications). Figure 1 plots the number of computational steps versus the transform sizes for the conventional algorithm and the algorithm presented in this paper. It can easily be seen that the algorithm presented here takes less than 1/6 as many steps as the conventional algorithm. Also plotted in the figure are the computational steps for FFT. It can be seen that the new algorithm takes only 1/3 as many steps as the FFT.

Figure 2 is a signal-flow graph for $N = 4, 8, 16, 32$ arranged in the fashion described. Note that the input samples are in natural order from top to bottom. For every N, the output transform coefficients are in bit-reversed order.

Note that as N increases the even coefficients of each successive transform are obtained directly from the coefficients of the prior transform by doubling the subscript of the prior coefficients.

It can be seen that extension of the signal-flow graph to the next power of 2 merely involves adding a set of ±1 butterflies to accommodate the new set of input samples and a series of alternating cosine/sine butterflies and ±1 butterflies to yield the new set of odd transform coefficients.

In Figure 2, the coefficients have not been normalized. To obtain the normalized N-point DCFT coefficients, the appropriate terminal points of the flow graph of Figure 2 should each be multiplied by $2/N$. This signal-flow graph represents the forward transform matrix $[A_N]$. The inverse transform matrix $[A_N]^{-1}$ is simply $(N/2)[A_N]^T$.

Thus, except for normalization factor these FDCT signal-flow graphs are bidirectional, i.e., the inverse transform may be computed by introducing the vector $[F]$ at the output and recovering the vector $[f]$ at the input. This follows from the fact that every butterfly pair in the signal-flow graph is a unitary matrix except for a normalization factor.

SUMMARY

A Fast Discrete Cosine Transform (FDCT) algorithm has been developed which may be extended to any desired value of $N = 2^m \geqslant 2$. The algorithm has been interpreted in the form of matrices and illustrated by a signal-flow graph. The

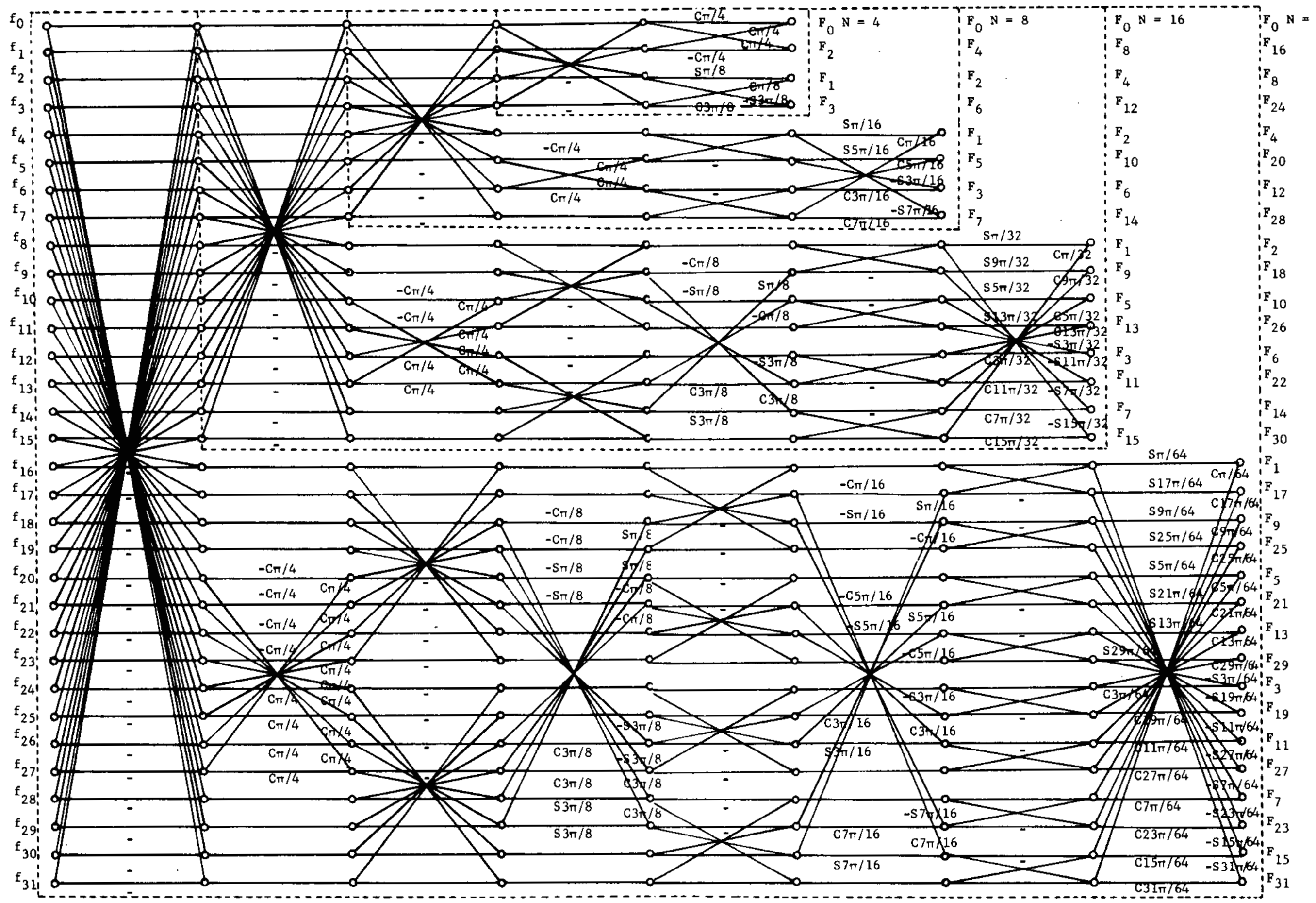

Figure 2. FDCT Flow Graph for $N = 4$, $N = 8$, $N = 16$ and $N = 32$; $Ci = \cos i$, $Si = \sin i$.

signal-flow graph may be readily translated to hardware (or software) implementation. The number of computational steps has been shown to be less than 1/6 of the conventional DCT algorithm employing a 2-sided FFT.

ACKNOWLEDGMENT

The authors are deeply indebted to Miss J. Del Mastro for her patience and skill in typing this paper.

REFERENCES

1. N. Ahmed, T. Natarjan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Transactions on Computer,* January 1974, pp. 90-93.
2. A. Habibi, "Hybrid Coding of Pictorial Data," *IEEE Transactions on Communications,* Vol. COM-22, No. 5, May 1974, pp. 614-624.
3. J. A. Rose, W. K. Pratt, G. S. Robinson, and A. Habibi, "Interframe Transform Coding and Predictive Coding Methods," *Proceedings of ICC 75,* IEEE Catalog No. 75CH0971-2GSCB, pp. 23.17-23.21.
4. W. Chen and C. H. Smith, "Adaptive Coding of Color Images Using Cosine Transform," *Proceedings of ICC 76,* IEEE Catalog No. 76CH1085-0 CSCB, pp. 47-7 to 47-13.
5. J. R. Persons and A. G. Tescher, "An Investigation of MSE Contributions in Transform Image Coding Schemes," *Proceedings of SPIE,* August 1975, pp. 196-206.
6. W. Chen and S. C. Fralick, "Image Enhancement Using Cosine Transform Filtering," *Proceedings of the Symposium on Current Mathematical Problems in Image Science,* Montery, Ca., November 1976, pp. 186-192.

Incoherent Adaptive Reception of Signal with Unknown Envelope

SOLOMON M. FLEISHER, MEMBER, IEEE

Abstract—This paper deals with the design of adaptive algorithms for reception of a narrowband signal with an unknown envelope in a noisy channel (Gaussian noise). We here consider a system (with a "real teacher") which learns from the samples classified by this self-learning system (decision directed adaptive receiver). By using those samples which are accepted as learning samples, the parameters of the unknown envelope are estimated. The envelope's parameters appear in the form of coefficients of the generalized Fourier series expansion of the signal (with respect to eigenfunctions of appropriate integral equation). It is possible to utilize any orthonormal set with respect to the interval $(0, T)$ under the usual assumption, that the complex autocovariance function is $R(\tau) = N\delta(\tau)$ (i.e., that the noise bandwidth is much greater than both $1/T$ and the signal bandwidth and N is the unilateral spectral density of the noise in the neighborhood of the signal spectrum). We present expressions that enable the upper bound estimates of the error probability to be found for the derived algorithms. The results obtained for the binary detection are readily generalized to the case of an M-ary signal.

Paper approved by the Editor for Communication Theory of the IEEE Communications Society for publication without oral presentation. Manuscript received June 1, 1976; revised January 30, 1977.

The author is with the School of Education, Center for Technological Education, Tel Aviv University, Holon, Israel.