TABLE II
PERFORMANCE OF SEVERAL ORTHONORMAL SIGNAL DECOMPOSITION TECHNIQUES FOR GENERALIZED CORRELATION SOURCE MODEL GIVEN IN (27)

	M=2 G_{TC} (NER)	M=4 G_{TC} (NER)	M=8 G_{TC} (NER)
DCT	3.2027 (0.9733)	5.7332 (0.9273)	7.6714 (0.8503)
DST	3.2027 (0.9733)	3.8836 (0.8458)	4.8638 (0.7138)
MHT	3.2027 (0.9733)	3.7401 (0.8241)	4.4011 (0.5843)
WHT	3.2027 (0.9733)	5.0582 (0.9253)	5.8865 (0.8405)
Binomial-QMF(4tap)	3.7724 (0.9879)	6.5636 (0.9633)	8.0087 (0.9123)
Binomial-QMF(6tap)	3.9241 (0.9913)	7.0011 (0.9728)	8.6688 (0.9332)
Binomial-QMF(8tap)	3.9915 (0.9929)	7.1851 (0.9773)	8.9449 (0.9439)
Smith-Barnwell(8tap)	4.0297 (0.9939)	7.2831 (0.9805)	9.0903 (0.9517)
Most Regular(6tap)	3.9055 (0.9909)	6.9465 (0.9716)	8.5847 (0.9306)
Optimal QMF (8tap) *	4.0489 (0.9945)	7.3171 (0.9822)	9.1279 (0.9558)
Optimal QMF (8tap) **	4.0455 (0.9945)	7.2954 (0.9825)	9.0869 (0.9567)
Optimal QMF (6tap) *	3.9699 (0.9924)	7.1231 (0.9762)	8.8521 (0.9418)
Optimal QMF (6tap) **	3.9680 (0.9925)	7.1113 (0.9764)	8.8318 (0.9425)
Optimal QMF (4tap) *	3.7895 (0.9883)	6.6200 (0.9646)	8.0960 (0.9152)
Optimal QMF (4tap) **	3.7894 (0.9883)	6.6205 (0.9647)	8.0970 (0.9154)
Ideal Filter Bank	4.1643 (1.0000)	7.5980 (1.0000)	9.4757 (1.0000)

*Optimal QMF based on energy compaction[7].
**Optimal QMF based on minimized aliasing energy[7].

VI. CONCLUSIONS

A new objective performance measure for orthonormal signal decomposition is defined in this correspondence. The performance of several known decomposition techniques are compared and the results are interpreted. It is shown that the new measure NER complements the widely used energy compaction measure G_{TC} and is consistent with the experimental results.

REFERENCES

[1] F. Mintzer and B. Liu, "Aliasing error in the design of multirate filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 26, pp. 76–88, Feb. 1978.

[2] A. N. Akansu, R. Haddad, and H. Caglar, "The binomial QMF-wavelet transform for multiresolution signal decomposition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 41, pp. 13–19, Jan. 1993.

[3] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets. II: Variations on a Theme." preprint.

[4] —— private communication.

[5] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[6] M. Smith, and T. P. Barnwell, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 434–441, June 1986.

[7] H. Caglar, Y. Liu, and A. N. Akansu, "Statistically optimized PR-QMF design," in *Proc. SPIE Visual Commun. Image Processing*, Nov. 1991, pp. 86–94.

[8] R. A. Haddad and A. N. Akansu, "A new orthogonal transform for signal coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1404–1411, Sept. 1988.

[9] R. J. Clarke, *Transform Coding of Images*. London: Academic, 1985.

[10] A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands and Wavelets*. London: Academic, 1992.

Mixed-Radix Discrete Cosine Transform

Yuk-Hee Chan and Wan-Chi Siu

Abstract—**This note presents two new fast discrete cosine transform computation algorithms: a radix-3 and a radix-6 algorithm. These two new algorithms are superior to the conventional radix-2 algorithm as they i) require less computational complexity in terms of the number of multiplications per point, ii) provide a wider choice of the sequence length for which the DCT can be realized and, iii) support the prime-factor-decomposed computation algorithm to realize the 2^m3^n-point DCT. Furthermore, a mixed-radix algorithm is also proposed such that an optimal performance can be achieved by applying the proposed radix-3 and radix-6 and the well-developed radix-2 decomposition techniques in a proper sequence.**

Manuscript received November 21, 1991; revised January 4, 1993.
The authors are with the Department of Electronic Engineering, Hong Kong Polytechnic, Hung Hom, Kowloon, Hong Kong.
IEEE Log Number 9211527.

I. INTRODUCTION

Many fast algorithms [1]–[8] for the computation of the discrete cosine transform (DCT) have been proposed since its first introduction in [9]. However, most algorithms were proposed for the computation of a 2^m-point DCT. Recently, Yang and Narasimha [10], and Lee [11], discussed a prime factor decomposed computation algorithm such that one can deal with DCT with lengths other than 2^m and therefore have a wider choice of the sequence length for which the DCT can be realized.

In this note, a new radix-3 and a new radix-6 algorithm are first presented to compute a length-3^m and a length-6^m DCT respectively. Further analyses are then made on using the prime factor-decomposed computation algorithm and a suggested mixed-radix algorithm for the fast computation of the DCT.

II. RADIX-3 DISCRETE COSINE TRANSFORM

The DCT [9] of a real data sequence $\{x(i): i = 0, 1, \cdots N - 1\}$ is defined by

$$X(k) = \sum_{i=0}^{N-1} x(i) \cos\left(\frac{\pi(2i+1)k}{2N}\right) \quad \text{for } k = 0, 1, \cdots N-1. \tag{1}$$

If $N = 3^m$, where m is a positive integer, we can realize the following three formulations to obtain the DCT result of the sequence $\{x(i)\}$ instead of realizing (1) directly.

$$\begin{aligned}
A(k) &= X(3k) = \sum_{i=0}^{N/3-1} \{a_i + b_i + c_i\} \cos\left(\frac{3\pi(2i+1)k}{2N}\right)\\
B(k) &= X(3k+1) + X(3k-1)\\
&= \sum_{i=0}^{N/3-1} \{(2a_i - b_i - c_i)\cos\alpha_i + (c_i - b_i)\sqrt{3}\sin\alpha_i\}\\
&\quad \cdot \cos\left(\frac{3\pi(2i+1)k}{2N}\right)\\
C(k) &= X(3k+2) + X(3k-2)\\
&= \sum_{i=0}^{N/3-1} \{(2a_i - b_i - c_i)\cos 2\alpha_i + (b_i - c_i)\sqrt{3}\sin 2\alpha_i\}\\
&\quad \cdot \cos\left(\frac{3\pi(2i+1)k}{2N}\right) \quad \text{for } k = 0, 1, \cdots N/3 - 1
\end{aligned} \tag{2}$$

where

$$a_i = x(i),\quad b_i = x(2N/3 + i),\quad c_i = x(2N/3 - i - 1),$$

$$\alpha_i = \frac{\pi(2i+1)}{2N} \quad \text{and} \quad X(-i) = X(i) \quad \text{for } i = 0, 1, \cdots N/3 - 1. \tag{3}$$

1053-587X/93$03.00 © 1993 IEEE

Note that $A(k)$, $B(k)$ and $C(k)$ are all $N/3$-point DCT's. As $B(0) = 2X(1)$ and $C(0) = 2X(2)$, one can obtain the sequence $\{X(k): k = 0, 1 \cdots N-1\}$ from $\{A(k): k = 0, 1, \cdots N/3 - 1\}$, $\{B(k): k = 0, 1, \cdots N/3 - 1\}$ and $\{C(k): k = 0, 1, \cdots N/3 - 1\}$ with $2N/3 - 2$ additions. Hence, one can realize an N-point DCT via the realization of three $N/3$-point DCT's. The overhead of this process involves the formation of input sequences of the three $N/3$-point DCT's. Specifically, to obtain the sequence $\{(2a_i - b_i - c_i) \cos \alpha_i + (c_i - b_i)\sqrt{3} \sin \alpha_i: i = 0, 1 \cdots N/3 - 1\}$, two multiplications are required for each i. For the computation of the sequence $\{(2a_i - b_i - c_i) \cos 2\alpha_i + (b_i - c_i)\sqrt{3} \sin 2\alpha_i: i = 0, 1 \cdots N/3 - 1\}$, since it can be rewritten as $\{2 \cos \alpha_i ((2a_i - b_i - c_i) \cos \alpha_i + (b_i - c_i)\sqrt{3} \sin \alpha_i) - (2a_i - b_i - c_i): i = 0, 1 \cdots N/3 - 1\}$, only one additional multiplication is required for each i. Hence, generally, three multiplications are required for each i to obtain all three input sequences. However, when $i = 1/2(N/3 - 1)$, we have $\alpha_i = \pi/6$. In such case, two more multiplications can be saved during the computation of these items.

In summary, the mathematical complexity of an $N(=3^m)$-point DCT to be realized by this new algorithm is given by the following set of equations:

$$M(N\text{-}DCT) = N \log_3 N - N + 1$$
$$A(N\text{-}DCT) = 3N \log_3 N - \tfrac{5}{2}(N-1) \quad \text{for } N = 3^m,\ m > 0. \tag{4}$$

III. Radix-6 Discrete Cosine Transform

If $N = 6^m$, where m is a positive integer, we can realize the following six formulations to obtain the DCT result of the sequence $\{x(i)\}$ instead of realizing (1) directly.

$$A(k) = X(6k) = \sum_{i=0}^{N/6-1} \{a_i + b_i + c_i + d_i + e_i + f_i\} \cdot \cos\left(\frac{6\pi(2i+1)k}{2N}\right)$$

$$B(k) = X(6k+1) + X(6k-1) = \sum_{i=0}^{N/6-1} \{(2a_i + b_i + c_i - d_i - e_i - 2f_i) \cdot \cos \theta_i + (b_i - c_i + d_i - e_i)\sqrt{3} \sin \theta_i\} \cdot \cos\left(\frac{6\pi(2i+1)k}{2N}\right)$$

$$C(k) = X(6k+2) + X(6k-2) = \sum_{i=0}^{N/6-1} \{(2a_i - b_i - c_i - d_i - e_i + 2f_i) \cdot \cos 2\theta_i + (b_i - c_i - d_i + e_i)\sqrt{3} \sin 2\theta_i\} \cdot \cos\left(\frac{6\pi(2i+1)k}{2N}\right)$$

$$D(k) = X(6k+3) + X(6k-3) = \sum_{i=0}^{N/6-1} \{2(a_i - b_i - c_i + d_i + e_i - f_i) \cos 3\theta_i\} \cdot \cos\left(\frac{6\pi(2i+1)k}{2N}\right)$$

$$E(k) = X(6k+4) + X(6k-4) = \sum_{i=0}^{N/6-1} \{(2a_i - b_i - c_i - d_i - e_i + 2f_i) \cdot \cos 4\theta_i - (b_i - c_i - d_i + e_i)\sqrt{3} \sin 4\theta_i\} \cdot \cos\left(\frac{6\pi(2i+1)k}{2N}\right)$$

$$F(k) = X(6k+5) + X(6k-5) = \sum_{i=0}^{N/6-1} \{(2a_i + b_i + c_i - d_i - e_i - 2f_i) \cdot \cos 5\theta_i - (b_i - c_i + d_i - e_i)\sqrt{3} \sin 5\theta_i\} \cdot \cos\left(\frac{6\pi(2i+1)k}{2N}\right) \quad \text{for } k = 0, 1, \cdots N/6 - 1 \tag{5}$$

where

$$a_i = x(i),\ b_i = x(N/3 - i - 1),$$
$$c_i = x(N/3 + i),\ d_i = x(2N/3 - i - 1),$$
$$e_i = x(2N/3 + i),\ f_i = x(N - i - 1),$$
$$\theta_i = \frac{\pi(2i+1)}{2N} \quad \text{and} \quad X(-i) = X(i) \quad \text{for } i = 0, 1, \cdots N/6 - 1. \tag{6}$$

Note that $A(k)$, $B(k)$, $C(k)$, $D(k)$, $E(k)$, and $F(k)$ are all $N/6$-point DCT's. Similar to the above section, one can obtain the sequence $\{X(k): k = 0, 1 \cdots N-1\}$ from $\{A(k): k = 0, 1, \cdots N/6 - 1\}$, $\{B(k): k = 0, 1, \cdots N/6 - 1\}$, $\{C(k): k = 0, 1, \cdots N/6 - 1\}$, $\{D(k): k = 0, 1, \cdots N/6 - 1\}$, $\{E(k): k = 0, 1, \cdots N/6 - 1\}$ and $\{F(k): k = 0, 1, \cdots N/6 - 1\}$ with $5N/6 - 5$ additions. In other words, one can realize an N-point DCT via the realization of six $N/6$-point DCT's. The overhead of this 1-to-6 decomposition process involves the formation of input sequences of the six $N/6$-point DCT's. For the computation of the sequence $\{(2a_i - b_i - c_i - d_i - e_i + 2f_i) \cos 4\theta_i - (b_i - c_i - d_i + e_i)\sqrt{3} \sin 4\theta_i: i = 0, 1 \cdots N/6 - 1\}$, as it can be rewritten as $\{2 \cos 2\theta_i ((2a_i - b_i - c_i - d_i - e_i + 2f_i) \cos 2\theta_i - (b_i - c_i - d_i + e_i)\sqrt{3} \sin 2\theta_i) - (2a_i - b_i - c_i - d_i - e_i + 2f_i): i = 0, 1 \cdots N/6 - 1\}$, only one additional multiplication is required for each i. Hence, in general, eight multiplications are required for each i to obtain all six input sequences. However, when $N = 36$, we have $4\theta_1 = \pi/6$ and $2\theta_4 = \pi/4$. In this case, we can further save one multiplication and three additions during the computation of these items.

Note that the realization of a 6-point DCT module requires four nontrivial multiplications and 16 additions (see Appendix). Hence, the mathematical complexity of the proposed algorithm is given by the following equations:

$$M(N\text{-}DCT) = \tfrac{4}{3} N \log_6 N - \tfrac{25}{36} N$$
$$A(N\text{-}DCT) = 4N \log_6 N - \tfrac{19}{12} N + 1 \quad \text{for } N = 6^m,\ m > 1. \tag{7}$$

IV. Comparison Among Radix-2, 3, and 6 Algorithms

Figs. 1 and 2 show the computational effort per point required for the realization of the DCT with different radix algorithms. Our radix-3 algorithm requires smaller numbers of multiplications/ad-

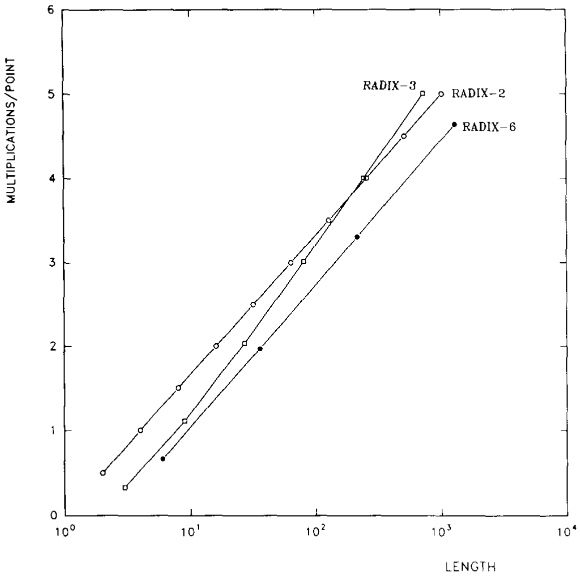

Fig. 1. Comparison of the numbers of multiplications per point among the radix-2, radix-3 and radix-6 algorithms.

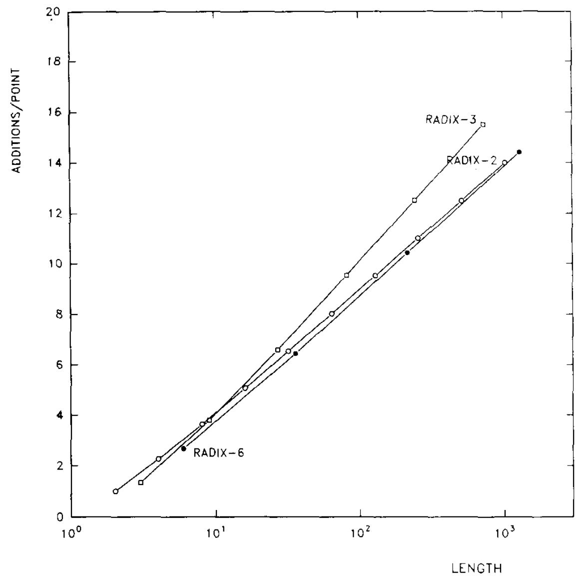

Fig. 2. Comparison of the numbers of additions per point among the radix-2, radix-3 and the prime-factor-decomposed algorithm (PFA) with the proposed radix-3 algorithm radix-6 algorithms.

ditions per point compared with the radix-2 algorithm when length, N, is small. In particular, the virtual breakeven point is at $N \approx 200$ when the number of multiplications per point is concerned. This symptom is due to the fact that the realization of the length-3 DCT module is more efficient than that of the length-2 DCT module while the decomposition overhead required for the radix-3 algorithm is larger than that for the radix-2 algorithm. On the other hand, our proposed radix-6 algorithm requires the least computational complexity among three algorithms whether the number of multiplications or additions per point is concerned. In fact, when N is sufficiently large, the radix-2 algorithm becomes more efficient than the radix-6 algorithm. However, this virtual breakeven point occurs at $N \approx 1.69 \times 10^{13}$ when the number of multiplications per point is concerned, which would imply cases that are far from the practical considerations.

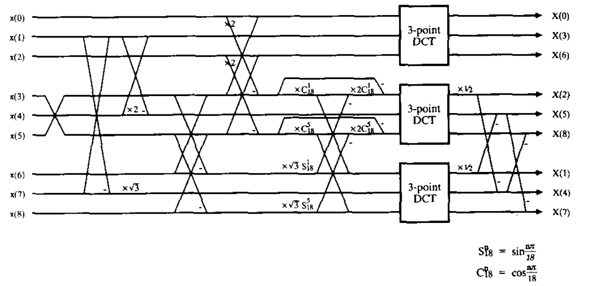

Fig. 3. Flowgraph of a 9-point DCT.

Basically, both radix-3 and radix-6 algorithms are decimation-in-frequency algorithms and therefore their computation structures are as simple as that of a radix-2 decimation-in-frequency algorithm such as Lee's algorithm [1]. Fig. 3 shows the flowgraph of the realization of a 9-point DCT. Similar flowgraphs can be obtained for the realization of the 6^m-point DCT's, which applies the radix-6 algorithm. Note, that the two proposed algorithms not only show their superiority in terms of the computational efficiency compared with the radix-2 algorithms, they also provide a wider choice of sequence lengths. Typically, a zero-padding technique has to be applied to realize a DCT of size $N \neq 2^n$. The existence of these algorithms provides a considerable reduction of this unnecessary computational effort. Fig. 4 shows the computational effort in terms of the number of the multiplications required to realize the DCT of different lengths when various algorithms are used.

V. Mixed-Radix Algorithm

By making use of the 1-to-3 and the 1-to-6 decomposition algorithms proposed in the above sections and those well-developed 1-to-2 decomposition techniques, an efficient mixed-radix algorithm can be developed to realize an N-point DCT for any $N = 2^m 3^n$ (where $m, n > 0$). Note that the decomposition overhead varies according to the transform length even though the same decomposition algorithm is applied. This variation in overhead is due to the fact that some additional saving in computation can be achieved during the decomposition of some special lengths. For example, if one performs a 1-to-6 decomposition on a DCT with length $N = 6 \times 3^m$ (where $m > 0$), then from (5) and (6), we have $2\theta_i = \pi/6$ if $i = 1/2(N/6 - 1)$. In this case, we can further save two multiplications and one addition compared with the normal case for $N = 6^k$ ($k > 2$) as we have the following:

$$A_i \cos 2\theta_i + B_i \sqrt{3} \sin 2\theta_i = \frac{\sqrt{3}}{2}(A_i + B_i)$$

and

$$A_i \cos 4\theta_i - B_i \sqrt{3} \sin 4\theta_i = \tfrac{1}{2}(A_i - 3B_i)$$

where

$$A_i = (2a_i - b_i - c_i - d_i - e_i + 2f_i), \quad B_i = (b_i - c_i - d_i + e_i).$$

Table I shows the overhead involved for the decomposition when various decomposition algorithms are applied to decompose an N-point DCT. Typically, for a given length-$N(= 2^m 3^n)$ DCT, there

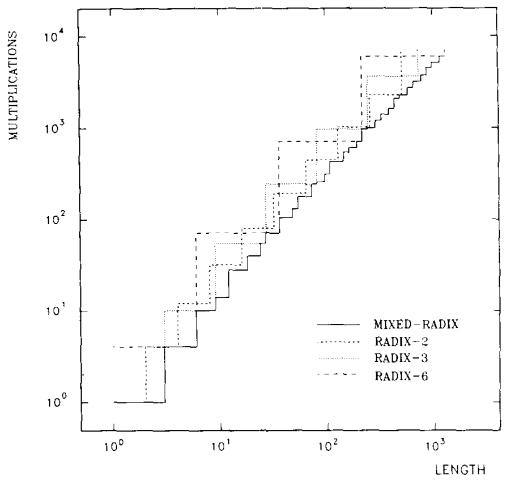

Fig. 4. Number of multiplications required for the computation of a DCT of specific length by making use of various algorithms.

TABLE I
ARITHMETIC OPERATIONS OVERHEAD FOR SELECTED DECOMPOSITION ALGORITHMS APPLIED TO AN N-POINT DCT

Decomposition Algorithm	Multiplication Overhead	Addition Overhead	Conditions
1-to-2	$N/2$	$3N/2 - 1$	$N = 2m;\ m > 1$
1-to-3	$N - 2$	$3N - 5$	$N = 3^m;\ m > 1$
	N	$3N - 2$	$N = 2^n \times 3^m;\ m,n > 0$
1-to-6	$4N/3 - 2$	$4N - 6$	$N = 6 \times 3^m;\ m > 0$
	$4N/3 - 1$	$4N - 8$	$N = 12 \times 3^m;\ m \geq 0$
	$4N/3$	$4N - 5$	$N = 2^n \times 6 \times 3^m;\ m \geq 0;\ n > 1$

are a number of approaches to realize this DCT when the mixed-radix algorithm is applied as there are many choices of decomposition sequences. From our analysis, it is found that the most efficient decomposition sequence of a length-$N(= 2^m 3^n)$ DCT is in the form of

$$\begin{cases} \{2^{m-n} 6^n\} & \text{if } m \geq n \\ \{6^m 3^{n-m}\} & \text{if } n > m \end{cases}$$

where $\{X^k Y^\lambda\}$ means that the DCT is realized through the following procedures: i) to perform the 1-to-X decomposition technique recursively on the length-$X^k Y^\lambda$ DCT to obtain X^k length-Y^λ DCT's, ii) to perform the 1-to-Y decomposition technique recursively on all length-Y^λ DCT's to obtain length-Y DCT's and iii) to realize all length-Y DCT modules.

Table II shows the comparison between our mixed-radix algorithm and the prime-factor-decomposed algorithm (PFA) [11]. Note that our radix-3 algorithm has already been applied to greatly enhance the performance of the PFA in [11]. It shows that the performance of the mixed-radix algorithm is always better than that of the PFA even though the PFA has already made use of the most efficient radix-3 and radix-2 algorithms. Specifically, the proposed mixed-radix algorithm always requires smaller numbers of both multiplications and total computational operations for the DCT realization. On the other hand, as our mixed-radix algorithm involves mainly a recursive decomposition, it is much more structural than that of the PFA. Complicated data management and data routing algorithms can be avoided.

TABLE II
COMPARISON OF THE COMPUTATIONAL COMPLEXITY BETWEEN THE MIXED-RADIX ALGORITHM RADIX-3 ALGORITHM

N	M(DCT)		A(DCT)	
	PFA(+radix-3)	mixed-radix	PFA(+radix-3)	mixed-radix
6	5	4	16	16
12	16	14	49	49
18	29	28	94	90
24	44	40	133	133
36	76	71	241	232
48	112	104	337	337
54	137	130	436	414
72	188	178	589	571
96	272	256	817	817
108	328	311	1033	964
144	448	428	1393	1357
162	569	544	1786	1710
192	640	608	1921	1921
216	764	714	2389	2251
288	1040	1000	3217	3145
324	1300	1211	4057	3772
384	1472	1408	4417	4417
432	1744	1644	5425	5149
486	2189	2110	6808	6570
576	2368	2288	7297	7153
648	2924	2730	9085	8371
768	3328	3200	9985	9985
864	3920	3720	12145	11593
972	4864	4559	15073	14140
1152	5312	5152	16321	16033
1296	6496	6012	20113	18685

VI. CONCLUSIONS

In this note, we first present a new radix-3 and a new radix-6 algorithm to compute a length-3^m and a length-6^m DCT respectively. The number of multiplications per point of these new algorithms show their superiority in mathematical complexity compared with that of radix-2 algorithms when N is small. They also provide a wider choice of the sequence lengths for which the DCT can be realized and support the prime-factor-decomposed computation algorithm to reduce the computational complexity. A mixed-radix algorithm is also presented, which gives the optimal performance in terms of the number of operations and the data managing requirements.

APPENDIX

A 6-point DCT on input sequence $\{x_i: i = 0, 1 \cdots 5\}$ is defined as

$$X_k = \sum_{i=0}^{5} x_i \cos \frac{\pi}{12} (2i + 1)k \qquad \text{for } k = 0, 1 \cdots 5. \tag{A1}$$

The relation

$$X_{2k} = \sum_{i=0}^{2} (x_i + x_{5-i}) \cos \frac{\pi}{6} (2i + 1)k \qquad \text{for } k = 0, 1, 2 \tag{A2}$$

enables the even-indexed outputs to be obtained via a 3-point DCT and three extra additions. For odd items, namely, $\{X_k: k = 1, 3, 5\}$, three multiplications and nine additions are required for their realization as follows:

$$\begin{bmatrix} t_0 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} x_5 \\ x_4 \\ x_3 \end{bmatrix};$$

$$t_3 = \frac{\sqrt{6}}{4} (t_0 + t_2);$$

$$t_4 = t_0 - t_2;$$

$$t_5 = \frac{\sqrt{2}}{4}(t_4 + 2t_1);$$

$$X_1 = t_3 + t_5;$$

$$X_3 = \frac{\sqrt{2}}{2}(t_4 - t_1);$$

$$X_5 = t_3 - t_5.$$

Note that the realization of a 3-point DCT requires one multiplication and four additions. Hence, four multiplications and 16 additions are required for the realization of a 6-point DCT.

References

[1] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-32, Dec. 1984, pp. 1243-1245.

[2] H. S. Hou, "A fast recursive algorithm for computing the discrete transform," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-35, Oct. 1987, pp. 1455-1461.

[3] M. Vetterli and N. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operation," *Signal Processing*, vol. 6, Aug. 1984, pp. 267-278.

[4] Y. H. Chan and W. C. Siu, "A cyclic correlated structure for the realization of the discrete cosine transform," *IEEE Trans. Circuit Syst.*, vol. 39, Feb. 1992, pp. 109-112.

[5] W. Li, "A new algorithm to compute the DCT and its inverse," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 39, June 1991, pp. 1305-1313.

[6] P. Duhamel and H. H'mida, "New 2^n DCT algorithms suitable for VLSI implementation," in *ICASSP-87*, Dallas, Apr. 1987, pp. 1805-1808.

[7] Y. H. Chan and W. C. Siu, "Algorithm for prime length discrete cosine transform," *IEEE Electronics Letters*, vol. 26, Feb. 1990, pp. 206-208.

[8] M. J. Narasimha and A. M. Peterson, "On the computation of the discrete cosine transform," *IEEE Trans. Communication*, vol. COM-26, June 1978, pp. 934-946.

[9] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, Jan. 1974, pp. 90-93.

[10] P. P. N. Yang and M. J. Narasimha, "Prime factor decomposition of the discrete cosine transform," in *ICASSP-85*, Tampa, FL, Mar. 1985, pp. 772-775.

[11] B. G. Lee, "Input and output index mappings for a prime-factor-decomposed computation of discrete cosine transform," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 37, Feb. 1989, pp. 237-244.

The Feedback Adaptive Line Enhancer: A Constrained IIR Adaptive Filter

Jue Chang and John R. Glover, Jr.

***Abstract*—A new adaptive line enhancer (ALE) structure, called the Feedback ALE (FALE), is presented and is shown to be a constrained IIR adaptive filter. Extensive simulations show that the FALE gives a higher sine-to-broadband ratio (SBR) gain and smaller sine estimation error than does an equal-order ALE; conversely, the order of the FALE can be much lower than the ALE to achieve equivalent performance.**

Manuscript received June 20, 1991; revised February 3, 1993.

The authors are with the Department of Electrical Engineering, University of Houston, Houston, TX 77204.

IEEE Log Number 9211539.

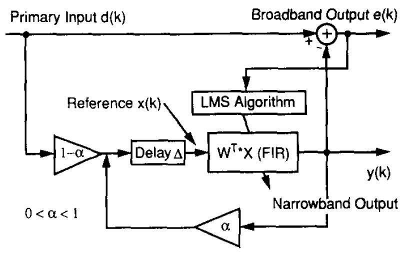

Fig. 1. The configuration of the FALE. When $\alpha = 0$, the FALE simplifies to the ALE.

I. Introduction

The adaptive line enhancer (ALE), as shown in Fig. 1 with $\alpha = 0$, is a well known configuration of the adaptive filter [1], [2]. Its purpose is to separate the sinusoidal component of its input from the broadband component without having a reference for either individually and without having *a priori* knowledge of the sinusoidal frequency. The bandwidth of the converged filter is determined by the number of weights in the filter. When the *sine-to-broadband ratio* (SBR)[1] at the primary input $d(k)$ (SBRin) is high, the gain of the resulting FIR filter at the sinusoidal frequency is very close to unity, making separation easy. It turns out, however, that the ALE gives poorer performance when the SBRin is low. In this case the gain of the FIR filter at the frequency of the sinusoid is much less than unity, resulting in poorer separation of the sinusoidal and broadband components.

Griffiths [3] proposed a modification which involves setting the coefficient α in Fig. 1 to unity after convergence has been achieved for the ALE. This configuration was tested as an *adaptive oscillator* to track the instantaneous frequency of the input signal. However, our simulations show that the output of the adaptive oscillator either has strong amplitude modulation or dies out under different values of μ. Moreover, it does not track well at all when the sinusoidal frequency drifts.

In this correspondence we present a new configuration which is a compromise between the original ALE and the adaptive oscillator [4]. As shown in Fig. 1, we use a weighted average of the primary input $d(k)$ and the filter output $y(k)$ as the reference input to the adaptive filter; i.e., $0 < \alpha < 1$. By varying the feedback constant α, we have a continuous transformation from the ALE ($\alpha = 0$) to the adaptive oscillator ($\alpha = 1$). The motivation behind the new configuration, called the feedback ALE (FALE), was to achieve some of the benefits of a noise canceller with a separate pure sinusoidal reference [5] in cases when a self-referencing ALE is necessary.

Simulations show considerable improvement in the sine estimation error and the SBR at $y(k)$ over those obtained with the ALE. At the same time, the stability problem of the adaptive oscillator is eliminated. Under some assumptions, predicted results fit the simulations quite well. In the following sections, we will describe the FALE in detail and provide simulation results, and we will offer

[1] In the sense of a *line enhancer*, the sinusoidal component is seen as signal, which is being enhanced over the broadband noise, producing the signal estimate at the filter output $y(k)$. Of course, viewed as a self-referencing *noise canceller* [1], the same configuration removes the sinusoidal component, now seen as interference, from the broadband signal, producing the signal estimate at the error output $e(k)$. Therefore, to avoid confusion we define and use a term called *sine-to-broadband ratio* (SBR).

1053-587X/93$03.00