TABLE II

<table>
<thead>
<tr>
<th>TECHNIQUE</th>
<th>$G_{ec}$ (V.E.R.)</th>
<th>$G_{ec}$ (V.E.R.)</th>
<th>$G_{ec}$ (V.E.R.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>WST</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>Biorthogonal QTF (16tap)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>Biorthogonal QTF (64tap)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>Biorthogonal QTF (256tap)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>Multiwavelet (16tap)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>Optimal RBF</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>Optimal QTF</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>Optimal QTF</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
<tr>
<td>Real Filter Bank</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
<td>1.3202 (0.9735)</td>
</tr>
</tbody>
</table>

*Optimal QTF based on energy comparison.

**Optimal QTF based on minimum aliasing energy.

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 comparison measure $G_{ec}$ and is consistent with the experimental results.

REFERENCES


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 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*3-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.

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-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 2n 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-3n and a length-6n 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

$$X(k) = \sum_{i=0}^{N-1} x(i) \cos \left(\frac{\pi (2i + 1)k}{2N}\right)$$

for $k = 0, 1, \cdots, N - 1$. (1)

If $N = 3^n$, 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.

$$A(k) = X(3k) = \sum_{i=0}^{N/3-1} \{a_i + b_i + c_i\} \cos \left(\frac{3\pi (2i + 1)k}{2N}\right)$$

$$B(k) = X(3k + 1) + X(3k - 1) = \sum_{i=0}^{N/3-1} \{(2a_i - b_i - c_i) \cos a_i + (c_i - b_i) \sqrt{3} \sin a_i\}$$

$$C(k) = X(3k + 2) + X(3k - 2) = \sum_{i=0}^{N/3-1} \{(2a_i - b_i - c_i) \cos 2a_i + (b_i - c_i) \sqrt{3} \sin 2a_i\}$$

$$X(k) = \frac{3\pi (2i + 1)k}{2N}$$

for $k = 0, 1, \cdots, N/3 - 1$ (2)

where

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

$$a_i = \frac{\pi (2i + 1)}{2N}$$

for $i = 0, 1, \cdots, N/3 - 1$. (3)
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 $\{(2a_i - b_i - c_i)\cos(\alpha_i + (b_i - c_i)\sqrt{3}\sin(\alpha_i); i = 0, 1 \cdots N/3 - 1\}$, 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 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 $N (= 3^n)$-point DCT to be realized by this new algorithm is given by the following set of equations:

\[
M(N-DCT) = N \log_2 N - N + 1
\]

\[
A(N-DCT) = 3N \log_2 N - \frac{3}{2}(N - 1)
\]

for $N = 3^m$, $m > 0$. \hspace{1cm} (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.

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

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

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

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

\[
eq \sum_{i=0}^{N/6-1} \left\{[(2a_i - b_i - c_i - d_i - e_i + 2f_i) \cos\left(\frac{6\pi(2i+1)}{2N}\right) + \cos\left(\frac{6\pi(2i+1)}{2N}\right)\right\}
\]

\[
F(k) = X(6k + 5) + X(6k - 5) = \sum_{i=0}^{N/6-1} \left\{[(2a_i + b_i + c_i - d_i - e_i - 2f_i) \cos\left(\frac{6\pi(2i+1)}{2N}\right) + \cos\left(\frac{6\pi(2i+1)}{2N}\right)\right\}
\]

where

\[
a_i = x(i), b_i = x(N/3 - i - 1),
\]

\[
c_i = x(N/3 + i), d_i = x(2N/3 - i - 1),
\]

\[
e_i = x(N - i - 1),
\]

\[
\theta_i = \frac{\pi(2i+1)}{2N} \quad \text{and} \quad X(-i) = X(i)
\]

\hspace{1cm} \text{for } i = 0, 1, \cdots N/6 - 1. \hspace{1cm} (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 $\{A(k): k = 0, 1 \cdots N/6 - 1\}$, $\{C(k): k = 0, 1 \cdots N/6 - 1\}$, $\{D(k): k = 0, 1 \cdots N/6 - 1\}$, $\{E(k): k = 0, 1 \cdots N/6 - 1\}$ and $\{C(k): k = 0, 1 \cdots N/6 - 1\}$ with $5N/6$ 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 $\{(2a_i - b_i - c_i - d_i - e_i + 2f_i) \cos\left(\frac{6\pi(2i+1)}{2N}\right) + \cos\left(\frac{6\pi(2i+1)}{2N}\right)\}$, it can be rewritten as $2\cos\left(\frac{6\pi(2i+1)}{2N}\right) - 2\cos\left(\frac{6\pi(2i+1)}{2N}\right)$.

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:

\[
M(N-DCT) = \frac{1}{4} N \log_2 N - \frac{N - 1}{6} N
\]

\[
A(N-DCT) = 4N \log_2 N - \frac{N}{12} N + 1 \hspace{1cm} \text{for } N = 6^m, m > 1.
\]

\hspace{1cm} \text{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.

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

Fig. 4. Computation 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 \( N = 2^m \cdot 3^n \) (where \( m, n > 0 \)). Note that the decomposition of 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 \( N = 6 \times 3^n \) (where \( m > 0 \), then from (3) and (6), we have \( 2B_i = \pi/6 \) 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

\[ A_i \cos 4\theta_i - B_i \sqrt{3} \sin 4\theta_i = \frac{1}{2} (A_i - 3B_i) \]

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
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

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

where $\{x^m y^n\}$ means that the DCT is realized through the following procedures: i) to perform the 1-to-X decomposition technique recursively on the length-$X^m Y^n$ DCT to obtain $X^m$ length-$Y^n$ DCT's, ii) to perform the 1-to-Y decomposition technique recursively on all length-$Y^n$ 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 I

<table>
<thead>
<tr>
<th>Decomposition Algorithm</th>
<th>Multiplication Overhead</th>
<th>Addition Overhead</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-to-2</td>
<td>$2^{m-n}6^n$</td>
<td>$6^{m-n}3^n$</td>
<td>$N = 2^m 3^n$</td>
</tr>
<tr>
<td>1-to-3</td>
<td>$2^{m-n}6^n$</td>
<td>$6^{m-n}3^n$</td>
<td>$N = 2^m 3^n$</td>
</tr>
<tr>
<td>1-to-6</td>
<td>$2^{m-n}6^n$</td>
<td>$6^{m-n}3^n$</td>
<td>$N = 2^m 3^n$</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS

In this note, we first present a new radix-3 and a new radix-6 algorithm to compute a length-3$^n$ and a length-6$^n$ 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 $\{x_k; i = 0, 1, \cdots, 5\}$ is defined as

$$X_l = \sum_{i=0}^{5} x_i \cos \frac{\pi}{12} (2i + 1)k$$

for $k = 0, 1, \cdots, 5$. (A1)

The relation

$$X_{2k} = \sum_{i=0}^{5} (x_{i} + x_{5-i}) \cos \frac{\pi}{6} (2i + 1)k$$

for $k = 0, 1, 2$ (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:

$$\begin{bmatrix}
I_0 \\
I_1 \\
I_2
\end{bmatrix} = \begin{bmatrix}
X_0 \\
X_1 \\
X_2
\end{bmatrix} - \begin{bmatrix}
X_3 \\
X_4 \\
X_5
\end{bmatrix}$$

$$I_1 = \sqrt{6} (I_0 + I_2)$$
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 sineto-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.

REFERENCES


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) at the primary input \( u(k) \) 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 SBR 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 \( \alpha \) 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 \( \mu \). 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 \( u(k) \) and the filter output \( y(k) \) as the reference input to the adaptive filter; i.e., \( 0 < \alpha < 1 \). By varying the feedback constant \( \alpha \), 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

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