Fast Algorithms for Low-Delay TDAC Filterbanks in MPEG-4 AAC-ELD

Ravi K. Chivukula, Yuriy A. Reznik, *Senior Member, IEEE*, Yanyan Hu, Venkat Devarajan, *Senior Member, IEEE*, and Mythreya Jayendra-Lakshman

***Abstract*—The MPEG committee has completed development of a new audio coding standard called "MPEG-4 advanced audio coding–enhanced low delay" (AAC-ELD). AAC-ELD uses low delay spectral band replication (LD-SBR) technology together with a low delay time domain alias cancellation (LD TDAC) filterbank in the encoder to achieve both high coding efficiency and low algorithmic delay. In this paper, we present fast algorithms for implementing LD-TDAC filterbanks in AAC-ELD. Two types of fast algorithms are presented. In the first, we map LD-TDAC analysis and synthesis filterbanks to modified discrete cosine transform (MDCT) and inverse modified discrete cosine transform (IMDCT), respectively. Since MDCT/IMDCT are already extensively used in AAC and they have many fast algorithms, this mapping not only provides a fast implementation but also allows a common implementation of the filterbanks in AAC Low Complexity (AAC-LC), AAC Low Delay (AAC-LD) and AAC-ELD codecs. In the second algorithm, we provide a mapping to discrete Cosine transform of type II. The mapping to DCT-II allows the merger of the matrix operations with the windowing stage that precedes or follows them. This further reduces the number of multiplications and leads to an algorithm with the lowest known arithmetic complexity. For filterbanks of lengths 1024 and 960, we also present a new fast factorization of 15-point DCT-II that requires only 14 irrational multiplications, 3 dyadic rational multiplications and 67 additions.**

***Index Terms*—AAC, audio coding, DCT, factorization, fast algorithms, filterbanks, low delay, MDCT, MPEG, speech coding, time domain alias cancellation.**

I. Introduction

TRADITIONALLY speech and audio coding paradigms have been significantly different. Speech coding is primarily based on source modeling [1], and low round trip algorithmic delay for full-duplex communications can be achieved [3]. However, most speech codecs are only efficient in encoding single-speaker material and are unsuitable for generic audio content [8]. On the other hand, audio coding is based on modeling the psychoacoustics of human auditory system [2]. The codecs are intended for perceptually transparent reproduction of generic music material. The delay of these codecs is generally high due to long frame lengths and the use of orthogonal filterbanks such as Modified Discrete Cosine Transform (MDCT) whose delay depends on the length of the window [4], [22]. Hence they are unsuitable for full-duplex communication. MPEG-4 AAC Low Complexity (AAC-LC) [5] is an example of this type of codec.

MPEG-4 AAC Low Delay (AAC-LD) [5] codec reduces algorithmic delay by halving the frame length from 1024/960 to 512/480, by removing block switching and by minimizing the use of bit reservoir in the encoder. AAC-LD could reduce the delay down to 20 ms but it still required bit rates close to 64 kbps per channel to deliver satisfactory audio quality [8].

To overcome these issues, MPEG standardized AAC Enhanced Low Delay (AAC-ELD) codec [6], [8], [9], [10] and [11]. This codec addresses the drawbacks of AAC-LD by incorporating a low-delay spectral band replication (LD-SBR) tool and a new low-delay TDAC filterbank. The LD-SBR tool improves coding efficiency and also has minimal delay [8], [9], [21]. The delay of the new TDAC filterbank is independent of window length [4], [8] and hence, a window with multiple overlap can be used for good frequency selectivity. Parts of the window that access future input values are zeroed out, thus reducing the delay further. AAC-ELD achieves an algorithmic delay of only 31 ms with good audio quality at low bit rates of 32 kbps per channel [9]. The entire AAC-ELD encoder and decoder are shown in Fig. 1.

Fast algorithms for the filterbanks are important because significant computational complexity of both encoders and decoders are dependent on these signal processing blocks. Especially in mobile devices, there is a need to reduce computational complexity to reduce battery power consumption. In [12] we presented fast algorithms for the LD-SBR filterbanks in AAC-ELD. In this paper, we present two fast algorithms for AAC-ELD TDAC filterbanks. In the first algorithm, we map the Low delay analysis and synthesis filterbanks to the well known MDCT and IMDCT respectively [16]. The mapping involves only permutations, sign changes and additions. Since many fast algorithms exist for MDCT, this mapping essentially provides a fast algorithm to implement the new filterbanks. Since LC and LD profiles also use MDCT filterbanks, the mapping also provides a common framework for the joint implementation of filterbanks in all three profiles (LC, LD and ELD). We also present a very efficient algorithm for a 15-point DCT-II implementation useful for frame lengths of 960. This algorithm requires only 14 irrational multiplications, 3 dyadic

Manuscript received November 11, 2013; revised May 21, 2014; accepted July 27, 2014. Date of publication August 08, 2014; date of current version August 23, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wai-Yip Geoffrey Chan.

R. K. Chivukula is with Indian School of Business, Punjab, 140306, India (e-mail: ravikiranc@gmail.com).

Y. A. Reznik is with InterDigital, Inc., San Diego, CA 92121 USA (e-mail: yreznik@gmail.com).

V. Devarajan and Y. Hu are with the Department of Electrical Engineering, The University of Texas at Arlington, Arlington, TX 76019 USA (e-mail: venkat@uta.edu; yanyan.hu@mavs.uta.edu).

M. Jayendra-Lakshman is with Qualcomm Inc, San Diego, CA 92121 USA (e-mail: mjayendr@qti.qualcomm.com).

Digital Object Identifier 10.1109/TASLP.2014.2346314

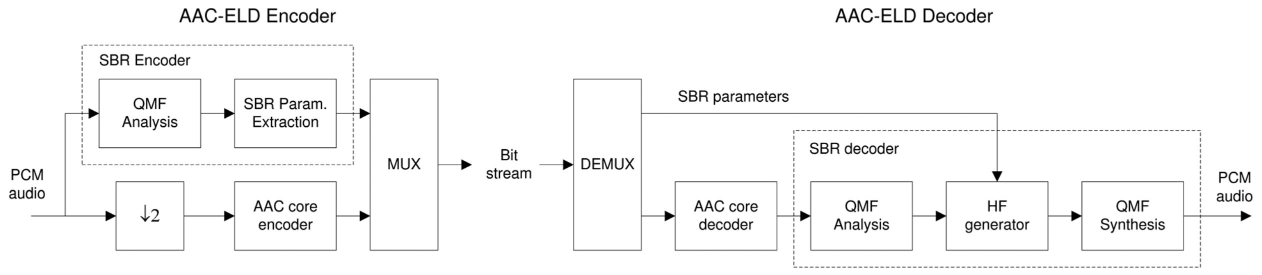


Fig. 1. Structure of AAC-ELD encoder and decoder.

rational multiplications and 67 additions. Complexity analysis of the AAC-ELD core coder filterbanks is provided at the end.

II. Definitions

In this paper time domain sequences are denoted by lower case italic letters such as $x(n)$, frequency domain sequences are denoted by upper case italic letters such as $X(k)$. Vectors are denoted by bold-face lower case letters such as $\mathbf{x}$, matrices are denoted by bold-face upper case letters such as $\mathbf{D}$. An element at lth row and mth column of a matrix $\mathbf{D}$ is denoted by $\mathbf{D}(l,m)$. The row and column indices start from 0. All sequences and matrix elements are real-valued. Also, in the subsequent sections, we assume N is a multiple of 4.

A. LD-TDAC Filterbanks in AAC-ELD

The matrix-vector product operation in the LD-TDAC analysis filterbank of AAC-ELD encoder is defined as follows [6] (with constants, sign '-' and scaling factors ignored):

$$X(k) = \sum_{n=-N}^{N-1} x(n) \cos\left(\frac{\pi(2n+n_0)(2k+1)}{2N}\right), 0 \le k < \frac{N}{2} \tag{1}$$

The matrix-vector product operation in the LD-TDAC synthesis filterbank of AAC-ELD decoder is defined as follows [6] (with constants, sign '-' and scaling factors ignored):

$$\hat{x}(n) = \sum_{k=0}^{\frac{N}{2}-1} X(k) \cos\left(\frac{\pi(2n+n_0)(2k+1)}{2N}\right), 0 \le n < 2N \tag{2}$$

where n_0 is defined as:

$$n_0 = -\frac{N}{2} + 1 \tag{3}$$

In the above equations, the sequence $x(n)$ denotes the windowed input data samples, $X(k)$ denotes subband coefficients and $\hat{x}(n)$ denotes reconstructed samples prior to alias cancellation. N is 1024 or 960.

B. MDCT Filterbanks

The MDCT and IMDCT are very similar to the LD-TDAC filterbanks. Below is the matrix-vector product operation of MDCT analysis filterbank (with constants and scaling factors ignored):

$$X'(k) = \sum_{n=0}^{N-1} z(n) \cos\left(\frac{\pi(2n+p_0)(2k+1)}{2N}\right), 0 \le k < \frac{N}{2} \tag{4}$$

The matrix-vector product operation of IMDCT filterbank is defined as:

$$x'(n) = \sum_{k=0}^{\frac{N}{2}-1} X'(k) \cos\left(\frac{\pi(2n+p_0)(2k+1)}{2N}\right), 0 \le n < N \tag{5}$$

where p_0 is defined as:

$$p_0 = \frac{N}{2} + 1 \tag{6}$$

In the above equations, the sequence $z(n)$ denotes the windowed input data samples, $X'(k)$ denotes MDCT spectrum coefficients, and $x'(n)$ denotes reconstructed samples prior to alias cancellation. N is the length of the input sequence.

C. Discrete Cosine and Sine Transforms

In order to accelerate computation of LD-TDAC filterbanks, we map them to several standard transforms allowing fast computation [19]. The Discrete Cosine Transforms of types II - IV of a sequence $d(n)$ of length N are defined as follows ($0 \le k < N$):

$$D_{C2}(k) = \sum_{n=0}^{N-1} d(n) \cos\left(\frac{\pi(2n+1)k}{2N}\right) \tag{7}$$

$$D_{C3}(k) = \sum_{n=0}^{N-1} d(n) \cos\left(\frac{\pi(2k+1)n}{2N}\right) \tag{8}$$

$$D_{C4}(k) = \sum_{n=0}^{N-1} d(n) \cos\left(\frac{\pi(2k+1)(2n+1)}{4N}\right) \tag{9}$$

For simplicity, we omit standard normalization factors in all these definitions [19].

Discrete Sine Transform of type IV (DST-IV), $D_{S4}(k)$, of a sequence $d(n)$ is defined as follows [19]:

$$D_{S4}(k) = \sum_{n=0}^{N-1} d(n) \sin\left(\frac{\pi(2k+1)(2n+1)}{4N}\right), 0 \le k < N \tag{10}$$

It is known that DST-IV is related to DCT-IV by means of sign changes and reversals of the input and output sequences [19]:

$$D_{S4}(N-1-k) = \sum_{n=0}^{N-1} (-1)^n d(n) \cos\left(\frac{\pi(2k+1)(2n+1)}{4N}\right), 0 \le k < N \tag{11}$$

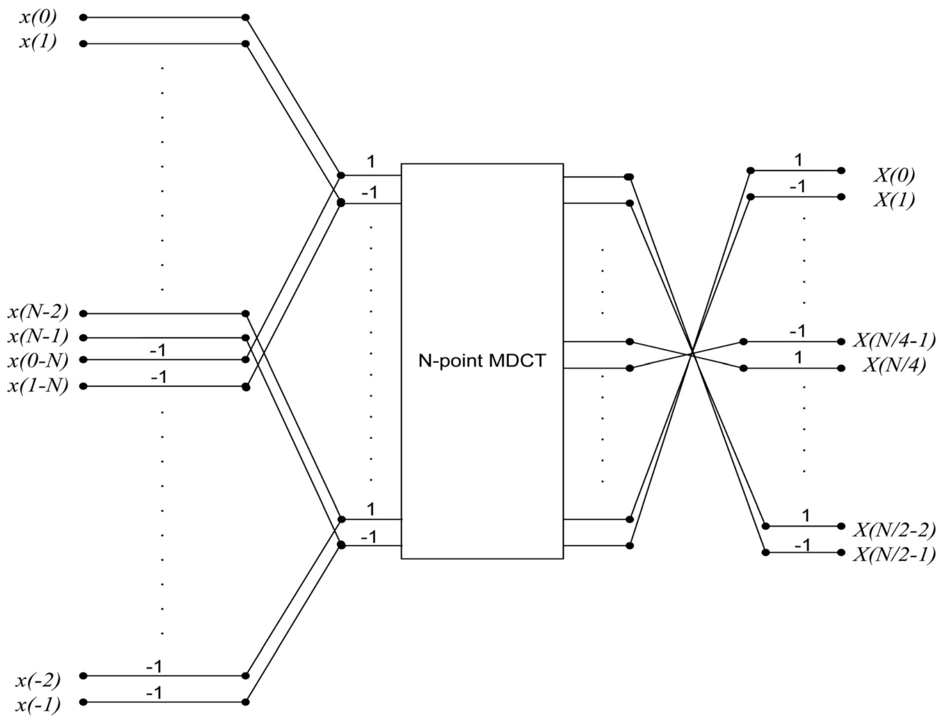


Fig. 2. Mapping LD-TDAC Analysis Filterbanks to MDCT.

III. MDCT-Based Fast Algorithms for the LD-TDAC Filterbanks

In this section we present how the LD-TDAC analysis and synthesis filterbanks map to MDCT and IMDCT respectively. This leads to fast algorithms since there are many fast algorithms for MDCT and IMDCT. These mappings are adopted in the reference software for AAC-ELD standard [7]. We present only the propositions and the flowgraphs; proofs are provided in the Appendix A.

I) Proposition 1: The analysis LD-TDAC filterbank can be mapped to MDCT as follows:

$$X(\frac{N}{2}-1-k) = (-1)^{\frac{N}{2}-1-k}\sum_{n=0}^{N-1}\left((-1)^{(n+\frac{N}{4})}\left(x(n)-x(n-N)\right)\right) \times \cos\left(\frac{\pi(2n+p_0)(2k+1)}{2N}\right), 0 \le k < \frac{N}{2} \quad (12)$$

II) Proposition 2: The synthesis LD-TDAC filterbank can be mapped to IMDCT as follows:

$$\hat{x}(n) = (-1)^{n+\frac{N}{4}}\sum_{k=0}^{\frac{N}{2}-1}\left((-1)^{(\frac{N}{2}-1-k)}X(\frac{N}{2}-1-k)\right)\cdot \times \cos\left(\frac{\pi(2n+p_0)(2k+1)}{2N}\right), 0 \le n < N \quad (13)$$

where $\hat{x}(n)$ has the following symmetry property:

$$\hat{x}(n+N) = -\hat{x}(n), 0 \le n < N \quad (14)$$

The implementations are shown in Figs. 2 and 3. A number of MDCT/IMDCT fast algorithms have been proposed in the literature for TDAC [13]–[15]. In the following, we propose a novel fast algorithm for LD-TDAC implementation by merging the windowing operation with DCT implementation.

IV. Mapping LD-TDAC Filterbanks to DCT-II

In this section, we present optimization of the real-valued LD-TDAC analysis filterbank by mapping it to DCT-II. This mapping uses DCT-IV as an intermediate step.

Definition 1:

$$s(n) \triangleq \begin{cases} \left\{x\left(n-\frac{N}{4}\right) - x\left(-\frac{N}{4}-1-n\right) + x\left(\frac{3N}{4}-1-n\right) -x\left(\frac{3N}{4}+n\right)\right\}, \text{for } 0 \le n < \frac{N}{4} \\ \left\{x\left(n-\frac{N}{4}\right) - x\left(-\frac{N}{4}-1-n\right) + x\left(\frac{3N}{4}-1-n\right) -x\left(n-\frac{5N}{4}\right)\right\}, \text{for } \frac{N}{4} \le n < \frac{N}{2} \end{cases} \quad (15)$$

Proposition 3:

$$X\left(\frac{N}{2}-1-k\right) = (-1)^{k+1}\sum_{n=0}^{\frac{N}{2}-1}(-1)^n s(n)\cos\left\{\frac{\pi(2n+1)(2k+1)}{4\left(\frac{N}{2}\right)}\right\}, \quad \text{for } 0 \le k < \frac{N}{2} \quad (16)$$

The proof of Proposition 3 is presented in the Appendix B.

The Proposition 3 presents the fast algorithm of mapping LD-TDAC to DCT-IV. To get the matrix description, let $\mathbf{x}$ denote the vector $x(n)$ for $-N \le n < N$ and let $\mathbf{s}$ denote the vector of samples $s(n)$ for $0 \le n < \frac{N}{2}$, i.e.,

$$\mathbf{x} = \left[\, x(-N) \quad x(-N+1) \quad \dots \quad x(N-1) \,\right]^T \quad (17)$$

$$\mathbf{s} = \left[\, s(0) \quad s(1) \quad \dots \quad s\left(\frac{N}{2}-1\right) \,\right]^T \quad (18)$$

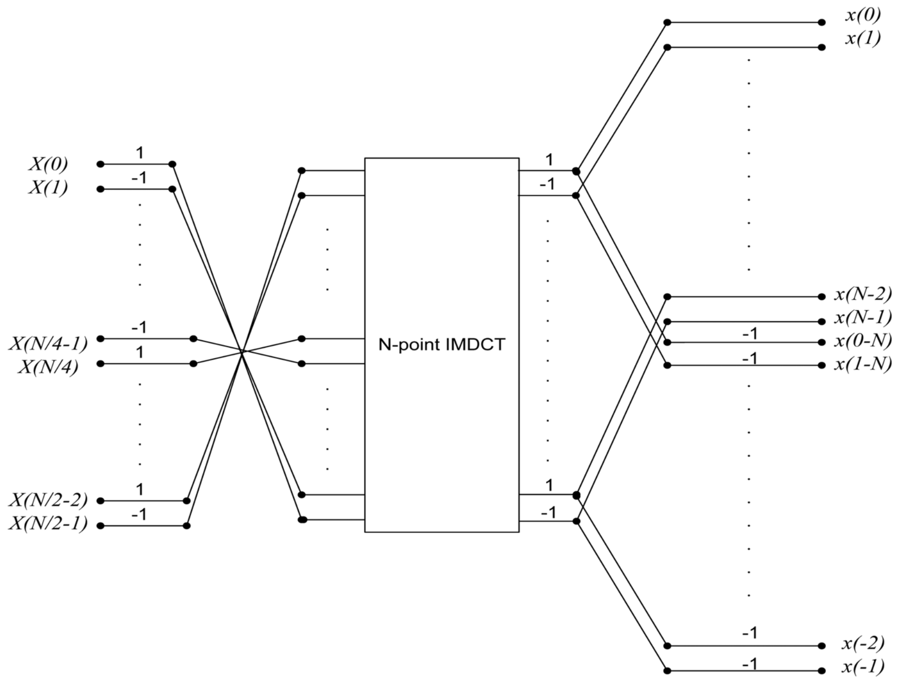


Fig. 3. Mapping LD-TDAC Synthesis Filterbanks to IMDCT.

Then,

$$\mathbf{s} = -\mathbf{B}_{\frac{N}{2}\times 2N}\mathbf{x} \tag{19}$$

where, $\mathbf{B}_{\frac{N}{2}\times 2N}$ is a $\frac{N}{2}\times 2N$ overlap-add matrix, whose elements can be obtained from (15) as,

$$\begin{aligned}
&\mathbf{B}_{\frac{N}{2}\times 2N}\left(n, n+\frac{3N}{4}\right) = -1, \text{for } 0 \le n < \frac{N}{2}\\
&\mathbf{B}_{\frac{N}{2}\times 2N}\left(n, \frac{3N}{4}-1-n\right) = 1, \text{for } 0 \le n < \frac{N}{2}\\
&\mathbf{B}_{\frac{N}{2}\times 2N}\left(n, \frac{7N}{4}-1-n\right) = -1, \text{for } 0 \le n < \frac{N}{2}\\
&\mathbf{B}_{\frac{N}{2}\times 2N}\left(n, \frac{7N}{4}+n\right) = 1, \text{for } 0 \le n < \frac{N}{4}\\
&\mathbf{B}_{\frac{N}{2}\times 2N}\left(n, n-\frac{N}{4}\right) = 1, \text{for} \frac{N}{4} \le n < \frac{N}{2}\\
&\mathbf{B}_{\frac{N}{2}\times 2N}(n,k) = 0, \quad \text{for all other combinations of } n \text{ and } k
\end{aligned} \tag{20}$$

Let $\mathbf{C}^{\mathbf{IV}}_{\frac{N}{2}}$ denote the $\frac{N}{2}\times\frac{N}{2}$ DCT-IV matrix with elements:

$$\mathbf{C}^{\mathbf{IV}}_{\frac{N}{2}}(k,n) = \cos\left(\frac{\pi(2k+1)(2n+1)}{2N}\right), \begin{array}{l} 0 \le n < \frac{N}{2},\\ 0 \le k < \frac{N}{2}. \end{array} \tag{21}$$

Let $\mathbf{A}_{\frac{N}{2}}$ denote an $\frac{N}{2}\times\frac{N}{2}$ diagonal matrix that inverts signs of odd-indexed elements:

$$\mathbf{A}_{\frac{N}{2}}(n,n) = (-1)^n, 0 \le n < \frac{N}{2}, \tag{22}$$

and $\mathbf{J}_{\frac{N}{2}}$ is a $\frac{N}{2}\times\frac{N}{2}$ matrix that reverses a vector of length $\frac{N}{2}$.

From (16) we can show that,

$$\mathbf{X} = \mathbf{J}_{\frac{N}{2}}\mathbf{A}_{\frac{N}{2}}\mathbf{C}^{\mathbf{IV}}_{\frac{N}{2}}\mathbf{A}_{\frac{N}{2}}\mathbf{B}_{\frac{N}{2}\times 2N}\mathbf{x} \tag{23}$$

Using [17], [20], [22] we can show that,

$$\mathbf{X} = \mathbf{J}_{\frac{N}{2}}\mathbf{A}_{\frac{N}{2}}\mathbf{L}_{\frac{N}{2}}\mathbf{C}^{\mathbf{II}}_{\frac{N}{2}}\mathbf{D}_{\frac{N}{2}}\mathbf{A}_{\frac{N}{2}}\mathbf{B}_{\frac{N}{2}\times 2N}\mathbf{x} \tag{24}$$

where, $\mathbf{C}^{\mathbf{II}}_{\frac{N}{2}}$ is the $\frac{N}{2}\times\frac{N}{2}$ matrix of DCT-II with elements:

$$\mathbf{C}^{\mathbf{II}}_{\frac{N}{2}}(k,n) = \cos\left(\frac{\pi(2n+1)k}{N}\right), \begin{array}{l} 0 \le n < \frac{N}{2},\\ 0 \le k < \frac{N}{2}, \end{array} \tag{25}$$

$\mathbf{L}_{\frac{N}{2}}$ is a $\frac{N}{2}\times\frac{N}{2}$ recursive addition matrix defined as follows (assuming $\frac{N}{2}$ is even):

$$\mathbf{L}_{\frac{N}{2}} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & \dots & 0\\ -\frac{1}{2} & 1 & 0 & 0 & \dots & 0\\ \frac{1}{2} & -1 & 1 & 0 & \dots & 0\\ \dots & \dots & \dots & \dots & \dots & \dots\\ -\frac{1}{2} & 1 & -1 & 1 & \dots & 1 \end{bmatrix} \tag{26}$$

and $\mathbf{D}_{\frac{N}{2}}$ is a $\frac{N}{2}\times\frac{N}{2}$ diagonal matrix of factors

$$\mathbf{D}_{\frac{N}{2}}(n,n) = 2\cos\left(\frac{\pi(2n+1)}{2N}\right), 0 \le n < \frac{N}{2}. \tag{27}$$

Because $\mathbf{D}_{\frac{N}{2}}$ and $\mathbf{A}_{\frac{N}{2}}$ are diagonal matrices we can have

$$\mathbf{X} = \mathbf{J}_{\frac{N}{2}}\mathbf{A}_{\frac{N}{2}}\mathbf{L}_{\frac{N}{2}}\mathbf{C}^{\mathbf{II}}_{\frac{N}{2}}\mathbf{A}_{\frac{N}{2}}\mathbf{D}_{\frac{N}{2}}\mathbf{B}_{\frac{N}{2}\times 2N}\mathbf{x} \tag{28}$$

We can notice from (20) that each column of $\mathbf{B}_{\frac{N}{2}\times 2N}$ has exactly one non-zero element. Further the magnitude of every non-zero element is the same (in this case, 1). Hence it can be easily shown that,

$$\mathbf{D}_{\frac{N}{2}}\mathbf{B}_{\frac{N}{2}\times 2N} = \mathbf{B}_{\frac{N}{2}\times 2N}\mathbf{D}'_{2N} \tag{29}$$

where, $\mathbf{D}'_{2N}$ is a $2N\times 2N$ diagonal matrix defined as:

$$\begin{aligned}&\text{if } \mathbf{B}_{\frac{N}{2}\times 2N}(n,k) = 1 \text{ or } -1, \text{then } \mathbf{D}'_{2N}(k,k) = \mathbf{D}_{\frac{N}{2}}(n,n),\\ &\quad \text{for } 0 \le n < \frac{N}{2}, 0 \le k < 2N\end{aligned} \tag{30}$$

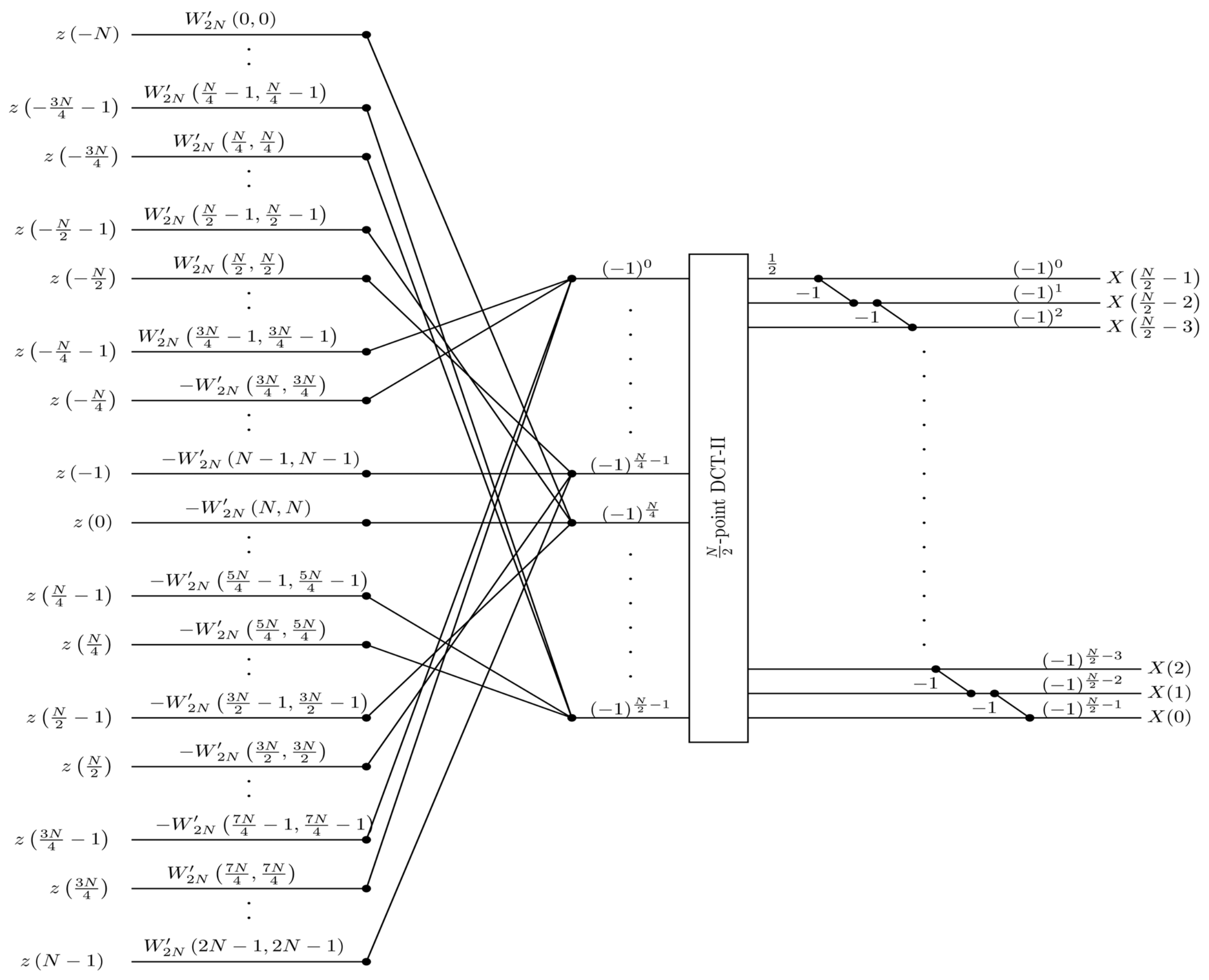


Fig. 4. Proposed Factorization of the LD-TDAC Analysis Filterbank.

The vector $\mathbf{x}$ is obtained by windowing a vector $\mathbf{z}$ of length $2N$, i.e.,

$$\mathbf{x} = \mathbf{W_{2N}z} \tag{31}$$

where $\mathbf{W_{2N}}$ is a $2N \times 2N$ diagonal matrix of constants defining the windowing operation. Hence, (28) becomes,

$$\mathbf{X} = \mathbf{J}_{\frac{\mathbf{N}}{2}}\mathbf{A}_{\frac{\mathbf{N}}{2}}\mathbf{L}_{\frac{\mathbf{N}}{2}}\mathbf{C}^{\mathbf{II}}_{\frac{\mathbf{N}}{2}}\mathbf{A}_{\frac{\mathbf{N}}{2}}\mathbf{B}_{\frac{\mathbf{N}}{2}\times\mathbf{2N}}\mathbf{D}'_{2N}\mathbf{W_{2N}z} \tag{32}$$

Since $\mathbf{D}'_{2N}$ and $\mathbf{W_{2N}}$ are diagonal matrices, we can have

$$\mathbf{W}'_{2N} = \mathbf{D}'_{2N}\mathbf{W_{2N}} \tag{33}$$

where, $\mathbf{W}'_{2N}$ is a $2N \times 2N$ diagonal matrix of constants. Using this in (32) we get,

$$\mathbf{X} = \mathbf{J}_{\frac{\mathbf{N}}{2}}\mathbf{A}_{\frac{\mathbf{N}}{2}}\mathbf{L}_{\frac{\mathbf{N}}{2}}\mathbf{C}^{\mathbf{II}}_{\frac{\mathbf{N}}{2}}\mathbf{A}_{\frac{\mathbf{N}}{2}}\mathbf{B}_{\frac{\mathbf{N}}{2}\times\mathbf{2N}}\mathbf{W}'_{2N}\mathbf{z} \tag{34}$$

By merging the multiplications given by $\mathbf{D}_{\frac{\mathbf{N}}{2}}$ in (27) with the windowing operation we further saved $\frac{N}{2}$ multiplications. Based on equation (34), the implementation is presented in Fig. 4.

V. Fast Algorithm for the LD-TDAC Synthesis Filterbank

The mapping of LD-TDAC analysis filterbank to MDCT and DCT-II is shown in previous figures. The synthesis filterbank is just a transpose of the analysis filterbank (up to a scale factor). Let $\mathbf{M}^{\mathbf{A}}_{\frac{\mathbf{N}}{2}\times\mathbf{2N}}$ be the analysis filterbank matrix without the scaling factor -2. Let $\mathbf{M}^{\mathbf{S}}_{\mathbf{2N}\times\frac{\mathbf{N}}{2}}$ be the synthesis filterbank matrix without the scaling factor $-2/N$, i.e., their elements are, respectively, given by

$$\mathbf{M}^{\mathbf{A}}_{\frac{\mathbf{N}}{2}\times\mathbf{2N}}(k,n+N) = \cos\left(\frac{\pi(2n+n_0)(2k+1)}{2N}\right),$$
$$\text{for } 0 \le k < \frac{N}{2}, -N \le n < N \tag{35}$$
$$\mathbf{M}^{\mathbf{S}}_{\mathbf{2N}\times\frac{\mathbf{N}}{2}}(n,k) = \cos\left(\frac{\pi(2n+n_0)(2k+1)}{2N}\right),$$
$$\text{for } 0 \le n < 2N, 0 \le k < \frac{N}{2} \tag{36}$$

It is easy to see that,

$$\mathbf{M}^{\mathbf{S}}_{\mathbf{2N}\times\frac{\mathbf{N}}{2}} = -\left[\mathbf{M}^{\mathbf{A}}_{\frac{\mathbf{N}}{2}\times\mathbf{2N}}\right]^T \tag{37}$$

Hence the algorithm for the synthesis filterbank can be obtained from a transposed version of the flow graph for the analysis filterbank, with the application of suitable scale factors. Transposing a flow graph consists of reversing the directions of the signals, replacing adders with tap-off points and replacing tap-off points with adders. The following proposition then becomes evident:

Proposition 4:

$$s(n) = (-1)^n \sum_{k=0}^{\frac{N}{2}-1} (-1)^{k+1} X\left(\frac{N}{2} - 1 - k\right) \times \cos\left(\frac{\pi(2n+1)(2k+1)}{4\left(\frac{N}{2}\right)}\right), \text{for } 0 \le n < \frac{N}{2} \quad (38)$$

$$\hat{x}(n) = s\left(n + \tfrac{N}{4}\right), \quad 0 \le n < \tfrac{N}{4} \quad (39)$$

$$\hat{x}(n) = s\left(\tfrac{3N}{4} - 1 - n\right), \quad \tfrac{N}{4} \le n < \tfrac{3N}{4} \quad (40)$$

$$\hat{x}(n) = -s\left(n - \tfrac{3N}{4}\right), \quad \tfrac{3N}{4} \le n < \tfrac{5N}{4} \quad (41)$$

$$\hat{x}(n) = -s\left(\tfrac{7N}{4} - 1 - n\right), \quad \tfrac{5N}{4} \le n < \tfrac{7N}{4} \quad (42)$$

$$\hat{x}(n) = s\left(n - \tfrac{7N}{4}\right), \quad \tfrac{7N}{4} \le n < 2N \quad (43)$$

Considering that the synthesis equation is simply a transpose of the analysis equation, the final expression for the synthesis filterbank optimization can be expressed as:

$$\hat{\mathbf{x}} = -\mathbf{B}^T_{\frac{\mathbf{N}}{\mathbf{2}}\times\mathbf{2N}} \mathbf{A}^T_{\frac{N}{2}} \mathbf{D}^T_{\frac{N}{2}} \mathbf{C}^{\mathbf{III}}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{L}^T_{\frac{N}{2}} \mathbf{A}^T_{\frac{N}{2}} \mathbf{J}^T_{\frac{N}{2}} \mathbf{X} \quad (44)$$

Since $\mathbf{D}_{\frac{N}{2}}$ and $\mathbf{A}_{\frac{N}{2}}$ are diagonal matrices, and $\mathbf{J}_{\frac{N}{2}}$ is an anti-diagonal matrix, the transposing operation has no effect on the matrices and therefore we can eliminate it. Further, we swap the order of $\mathbf{A}_{\frac{N}{2}}$ and $\mathbf{D}_{\frac{N}{2}}$ since it results in the same output. We combine the two operations and obtain the following result:

$$\hat{\mathbf{x}} = -\mathbf{B}^T_{\frac{\mathbf{N}}{\mathbf{2}}\times\mathbf{2N}} \mathbf{D}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{A}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{C}^{\mathbf{III}}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{L}^T_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{A}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{J}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{X} \quad (45)$$

We can use the equation (29) and the fact that $\mathbf{D}'_{\mathbf{2N}}$ is a diagonal matrix and derive the result:

$$\mathbf{B}^T_{\frac{\mathbf{N}}{\mathbf{2}}\times\mathbf{2N}} \mathbf{D}_{\frac{\mathbf{N}}{\mathbf{2}}} = \mathbf{D}'_{\mathbf{2N}} \mathbf{B}^T_{\frac{\mathbf{N}}{\mathbf{2}}\times\mathbf{2N}} \quad (46)$$

where, $\mathbf{D}'_{2N}$ is a $2N \times 2N$ diagonal matrix defined as:

$$\text{if } \mathbf{B}^T_{\frac{\mathbf{N}}{\mathbf{2}}\times\mathbf{2N}}(k,n) = 1 \text{or} - 1, \text{then} \mathbf{D}'_{2N}(k,k) = \mathbf{D}_{\frac{\mathbf{N}}{\mathbf{2}}}(n,n), \text{for } 0 \le n < \frac{N}{2}, 0 \le k < 2N \quad (47)$$

Hence, the final result for the synthesis filterbank is:

$$\hat{\mathbf{x}} = -\mathbf{D}'_{\mathbf{2N}} \mathbf{B}^T_{\frac{\mathbf{N}}{\mathbf{2}}\times\mathbf{2N}} \mathbf{A}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{C}^{\mathbf{III}}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{L}^T_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{A}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{J}_{\frac{\mathbf{N}}{\mathbf{2}}} \mathbf{X} \quad (48)$$

Based on equation (48), the implementation is presented in Fig. 5.

VI. Fast 15-point DCT-II Algorithm

As we have seen in the previous sections, DCT-II-based factorizations of the filterbanks lead to computationally-efficient algorithms. Generally, DCT-II is implemented by decimation in time or decimation in frequency strategies [19]. When N is equal to 1024 or 960, such strategies eventually lead to -point DCTs.

In order to a compute 15-point DCT-II, we propose [18] to use a factorization shown in Fig. 6. This factorization involves multiplication by 17 factors. The factors $c_1, \ldots, c_{17}$ in this flow-graph are defined as follows ($u = \frac{-2\pi}{5}$ and $v = \frac{-2\pi}{3}$):

$$\begin{aligned} &c_1 = \frac{\cos u + \cos 2u}{2} - 1;\ c_2 = \frac{\cos u - \cos 2u}{2}; \\ &c_3 = \sin u + \sin 2u;\ c_4 = \sin 2u;\ c_5 = \sin u - \sin 2u; \\ &c_6 = \cos v - 1;\ c_7 = c_1 c_6;\ c_8 = c_2 c_6;\ c_9 = c_3 c_6; \\ &c_{10} = c_3 c_6;\ c_{11} = c_5 c_6;\ c_{12} = \sin v; \\ &c_{13} = c_1 c_{12};\ c_{14} = c_2 c_{12};\ c_{15} = -c_3 c_{12}; \\ &c_{16} = -c_4 c_{12};\ c_{17} = -c_5 c_{12}; \end{aligned} \quad (49)$$

Among the 17 factors, 3 are dyadic rational numbers:

$$c_1 = -\frac{5}{4}; c_6 = -\frac{3}{2}; c_7 = \frac{15}{8} \quad (50)$$

This factorization requires only 14 non-trivial multiplications and 67 additions. Details of this derivation are explained in Appendix C.

VII. Complexity Analysis

In this section, we discuss the computational complexity of the proposed algorithms. We present the complexity functions based on the DCT-II approach and report the numbers of addition and multiplication operations required to implement AAC-ELD TDAC filterbanks in Table I.

Due to the transpose-like relationship between analysis and synthesis filterbank, we note that when the lengths of analysis and synthesis filterbanks are equal, the required number of additions and multiplications for their computation are the same. Consequently, the presented complexity formulas are valid for both analysis and synthesis filterbanks. In all cases, N denotes the length of the filterbanks, whose value is either 1024 or 960. To reduce redundancy, we only show the complexity analysis in the case of $N = 1024$.

In the DCT-II-based approach, multiplications are contributed by the $\frac{N}{2}$-point DCT-II block and the windowing matrix $\mathbf{W}'_{2N}$ in (33). The number of multiplication contributed by the $\frac{N}{2}$-point DCT-II block are $\frac{N}{4}\log_2(\frac{N}{2})$. The matrix $\mathbf{W}'_{2N}$ contributes $2N$ multiplications, but $\frac{N}{8}$ samples of the window are actually zeros and hence, multiplications and additions involving these coefficients need not be counted. Therefore, the total number of multiplications is given by

$$M(N) = \frac{N}{4}\log_2\left(\frac{N}{2}\right) + \frac{15N}{8} \quad (51)$$

Additions are contributed by the $\frac{N}{2}$-point DCT-II block, the matrix $\mathbf{B}_{\frac{\mathbf{N}}{\mathbf{2}}\times\mathbf{2N}}$ in equation (20) and the matrix $\mathbf{L}_{\frac{\mathbf{N}}{\mathbf{2}}}$ in equation (26). The $\frac{N}{2}$-point DCT-II block contributes $\frac{3N}{4}\log_2(\frac{N}{2}) - \frac{N}{2} + 1$ additions. Matrix $\mathbf{B}_{\frac{\mathbf{N}}{\mathbf{2}}\times\mathbf{2N}}$ contributes $\frac{3N}{2}$ additions. Matrix $\mathbf{L}_{\frac{\mathbf{N}}{\mathbf{2}}}$ contributes $\frac{N}{2} - 1$ additions. Again, $\frac{N}{8}$ samples of the window are actually zeros and hence, additions involving these coefficients need not be counted. Therefore, the total number of additions is given by

$$A(N) = \frac{3N}{4}\log_2\left(\frac{N}{2}\right) + \frac{11N}{8} \quad (52)$$

In the case of $N = 960$, the 15-point DCT-II fast algorithm in Section V combined with [20] is implemented to calculate the $\frac{N}{2}$-point DCT-II blocks.

We would like to point out that compared with the well-known standard (high delay) TDAC filterbank, which needs for its fast algorithm, 3584 multiplications and 7680 additions for N = 1024 [22], our fast algorithm for LD-TDAC needs only 17% higher multiplications and 8% higher additions.

VIII. Conclusions

In this paper, we developed fast algorithms for the TDAC analysis and synthesis filterbanks of MPEG-4 AAC-ELD codec. This new algorithm is based on mapping the filterbanks to DCT-II, in which some multiplications are merged with the windowing stage. We also presented a fast algorithm for a 15-point DCT-II, which is based on Heideman's mapping and Winograd Fourier Transform algorithm. This is useful for all TDAC filterbanks, in which the input data length is of the form $2^m \times 15$. This new fast 15-point DCT-II algorithm requires only 14 irrational multiplications, 3 multiplications by dyadic rationals and 67 additions. Finally, we presented a computational complexity analysis of the proposed algorithm.

Appendix A
Proof of Proposition 1 and 2

To prove Proposition 1, equation (1) is reproduced below for convenience:

$$X(k) = \sum_{n=-N}^{N-1} x(n)\cos\left\{\frac{\pi(2n+n_0)(2k+1)}{2N}\right\}, 0 \le k < \frac{N}{2} \tag{53}$$

Splitting the summation in (53) into two parts, we get:

$$\begin{aligned}X(k) &= \sum_{n=-N}^{-1} x(n)\cos\left\{\frac{\pi(2n+n_0)(2k+1)}{2N}\right\} \\ &\quad + \sum_{n=0}^{N-1} x(n)\cos\left\{\frac{\pi(2n+n_0)(2k+1)}{2N}\right\} \\ &= \sum_{n=0}^{N-1} x(n-N)\cos\left\{\frac{\pi(2n-2N+n_0)(2k+1)}{2N}\right\} \\ &\quad + \sum_{n=0}^{N-1} x(n)\cos\left\{\frac{\pi(2n+n_0)(2k+1)}{2N}\right\} \\ &= \sum_{n=0}^{N-1} \{x(n)-x(n-N)\}\cos\left\{\frac{\pi(2n+n_0)(2k+1)}{2N}\right\}\end{aligned} \tag{54}$$

Define p_0 as follows:

$$p_0 \triangleq \left(\frac{N}{2}+1\right) \tag{55}$$

Then,

$$\begin{aligned}X(k) &= \\ &\sum_{n=0}^{N-1}\left\{(x(n)-x(n-N))\cos\left(\frac{\pi(2n+p_0-N)(2k+1)}{2N}\right)\right\} \\ &= (-1)^k \sum_{n=0}^{N-1}\{x(n)-x(n-N)\}\sin\left\{\frac{\pi(2n+p_0)(2k+1)}{2N}\right\}\end{aligned} \tag{56}$$

Finally,

$$\begin{aligned}X\left(\frac{N}{2}-1-k\right) &= (-1)^{\frac{N}{2}-1-k}\sum_{n=0}^{N-1}(x(n)-x(n-N)) \\ &\quad \times \sin\left(\frac{\pi(2n+p_0)(2(\frac{N}{2}-1-k)+1)}{2N}\right) \\ &= (-1)^{\frac{N}{2}-1-k}\sum_{n=0}^{N-1}(-1)^{(n+\frac{N}{4})}(x(n)-x(n-N)) \\ &\quad \times \cos\left(\frac{\pi(2n+p_0)(2k+1)}{2N}\right)\end{aligned} \tag{57}$$

We note that the summation on the RHS of (57) is an MDCT. Thus, Proposition 1 achieves in mapping LD-TDAC analysis filterbanks to MDCT.

Proposition 2 maps the LD-TDAC synthesis filterbanks to IMDCT. The proof follows the same concept as shown above. To avoid redundancy, the details of proof of Proposition 2 are not presented here.

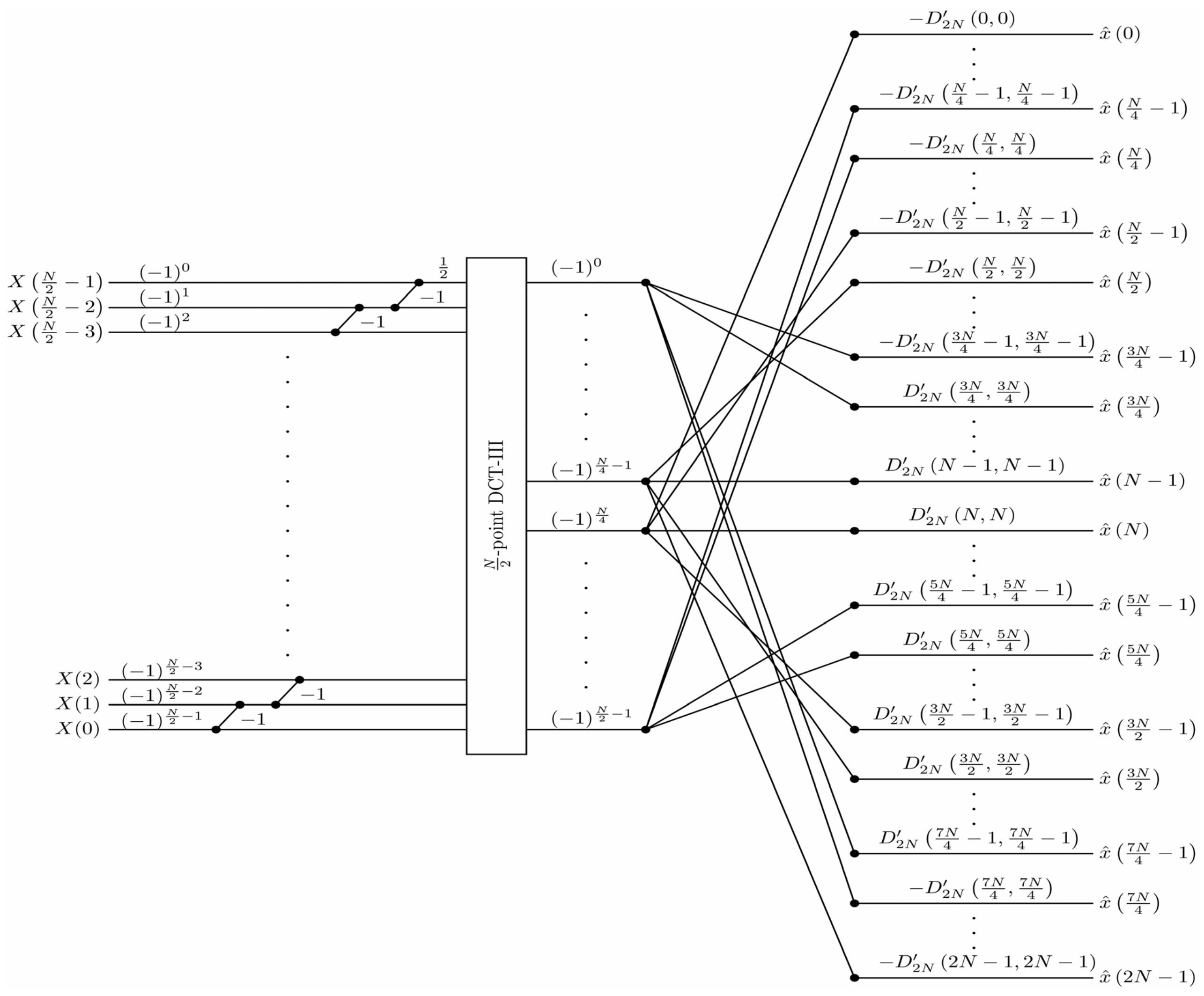
Appendix B
Proof of Proposition 3 and 4

To prove Proposition 3, we use the result of equation (56) and define a new sequence $r(n)$ as follows:

$$r(n) \triangleq \{x(n)-x(n-N)\}, 0 \le n < N \tag{58}$$

Using (58) and splitting the summation in (56) into four parts, we get:

$$\begin{aligned}X(k) = (-1)^{(k)} &\left[\sum_{n=0}^{\frac{N}{4}-1} r(n)\sin\left\{\frac{\pi(2n+p_0)(2k+1)}{2N}\right\}\right. \\ &+ \sum_{n=\frac{N}{4}}^{\frac{N}{2}-1} r(n)\sin\left\{\frac{\pi(2n+p_0)(2k+1)}{2N}\right\} \\ &+ \sum_{n=\frac{N}{2}}^{\frac{3N}{4}-1} r(n)\sin\left\{\frac{\pi(2n+p_0)(2k+1)}{2N}\right\} \\ &+ \left.\sum_{n=\frac{3N}{4}}^{N-1} r(n)\sin\left\{\frac{\pi(2n+p_0)(2k+1)}{2N}\right\}\right]\end{aligned} \tag{59}$$



$X\left(\frac{N}{2}-1\right)$
$X\left(\frac{N}{2}-2\right)$
$X\left(\frac{N}{2}-3\right)$
$(-1)^0$
$(-1)^1$
$(-1)^2$
$\frac{1}{2}$
-1
-1
$X(2)$
$X(1)$
$X(0)$
$(-1)^{\frac{N}{2}-3}$
$(-1)^{\frac{N}{2}-2}$
$(-1)^{\frac{N}{2}-1}$
-1
-1
$\frac{N}{2}$-point DCT-III
$(-1)^0$
$(-1)^{\frac{N}{4}-1}$
$(-1)^{\frac{N}{4}}$
$(-1)^{\frac{N}{2}-1}$
$-D'_{2N}(0,0)$
$-D'_{2N}\left(\frac{N}{4}-1,\frac{N}{4}-1\right)$
$-D'_{2N}\left(\frac{N}{4},\frac{N}{4}\right)$
$-D'_{2N}\left(\frac{N}{2}-1,\frac{N}{2}-1\right)$
$-D'_{2N}\left(\frac{N}{2},\frac{N}{2}\right)$
$-D'_{2N}\left(\frac{3N}{4}-1,\frac{3N}{4}-1\right)$
$D'_{2N}\left(\frac{3N}{4},\frac{3N}{4}\right)$
$D'_{2N}(N-1,N-1)$
$D'_{2N}(N,N)$
$D'_{2N}\left(\frac{5N}{4}-1,\frac{5N}{4}-1\right)$
$D'_{2N}\left(\frac{5N}{4},\frac{5N}{4}\right)$
$D'_{2N}\left(\frac{3N}{2}-1,\frac{3N}{2}-1\right)$
$D'_{2N}\left(\frac{3N}{2},\frac{3N}{2}\right)$
$D'_{2N}\left(\frac{7N}{4}-1,\frac{7N}{4}-1\right)$
$-D'_{2N}\left(\frac{7N}{4},\frac{7N}{4}\right)$
$-D'_{2N}(2N-1,2N-1)$
$\hat{x}(0)$
$\hat{x}\left(\frac{N}{4}-1\right)$
$\hat{x}\left(\frac{N}{4}\right)$
$\hat{x}\left(\frac{N}{2}-1\right)$
$\hat{x}\left(\frac{N}{2}\right)$
$\hat{x}\left(\frac{3N}{4}-1\right)$
$\hat{x}\left(\frac{3N}{4}\right)$
$\hat{x}(N-1)$
$\hat{x}(N)$
$\hat{x}\left(\frac{5N}{4}-1\right)$
$\hat{x}\left(\frac{5N}{4}\right)$
$\hat{x}\left(\frac{3N}{2}-1\right)$
$\hat{x}\left(\frac{3N}{2}\right)$
$\hat{x}\left(\frac{7N}{4}-1\right)$
$\hat{x}\left(\frac{7N}{4}\right)$
$\hat{x}(2N-1)$

Fig. 5. Proposed Factorization of the LD-TDAC Synthesis Filterbank.

Making the first summation go from $\frac{N}{4} \le n < \frac{N}{2}$, reversing the order of second summation, making the third and fourth summations go from $0 \le n < \frac{N}{4}$, we get:

$$
\begin{aligned}
X(k) = &\\
&(-1)^{(k)}\left[\sum_{n=\frac{N}{4}}^{\frac{N}{2}-1} r\left(n-\frac{N}{4}\right)\right.\\
&\times \sin\left\{\frac{\pi(2(n-N/4)+p_0)(2k+1)}{2N}\right\}\\
&+\sum_{n=\frac{N}{4}}^{\frac{N}{2}-1} r\left(\frac{3N}{4}-1-n\right)\\
&\times \sin\left\{\frac{\pi(2(3N/4-1-n)+p_0)(2k+1)}{2N}\right\}\\
&+\sum_{n=0}^{\frac{N}{4}-1} r\left(\frac{3N}{4}-1-n\right)\\
&\times \sin\left\{\frac{\pi(2(3N/4-1-n)+p_0)(2k+1)}{2N}\right\}\\
&+\sum_{n=0}^{\frac{N}{4}-1} r\left(\frac{3N}{4}+n\right)\\
&\left.\times \sin\left\{\frac{\pi(2(3N/4+n)+p_0)(2k+1)}{2N}\right\}\right]\\
&=(-1)^{(k)}\left[\sum_{n=\frac{N}{4}}^{\frac{N}{2}-1}\left[r\left(n-\frac{N}{4}\right)+r\left(\frac{3N}{4}-1-n\right)\right]\right.\\
&\times \sin\left\{\frac{\pi(2n+1)(2k+1)}{2N}\right\}\\
&+\sum_{n=0}^{\frac{N}{4}-1}\left[-r\left(\frac{3N}{4}+n\right)+r\left(\frac{3N}{4}-1-n\right)\right]\\
&\left.\times \sin\left\{\frac{\pi(2n+1)(2k+1)}{2N}\right\}\right] \qquad (60)
\end{aligned}
$$

Define a new sequence $s(n)$ as follows:

$$
s(n) \triangleq \begin{cases} -r\left(\frac{3N}{4}+n\right)+r\left(\frac{3N}{4}-1-n\right), \text{for } 0 \le n < \frac{N}{4} \\ r\left(n-\frac{N}{4}\right)+r\left(\frac{3N}{4}-1-n\right), \text{for } \frac{N}{4} \le n < \frac{N}{2} \end{cases} \qquad (61)
$$

Using (58), $s(n)$ can be expressed in terms of $x(n)$ as follows:

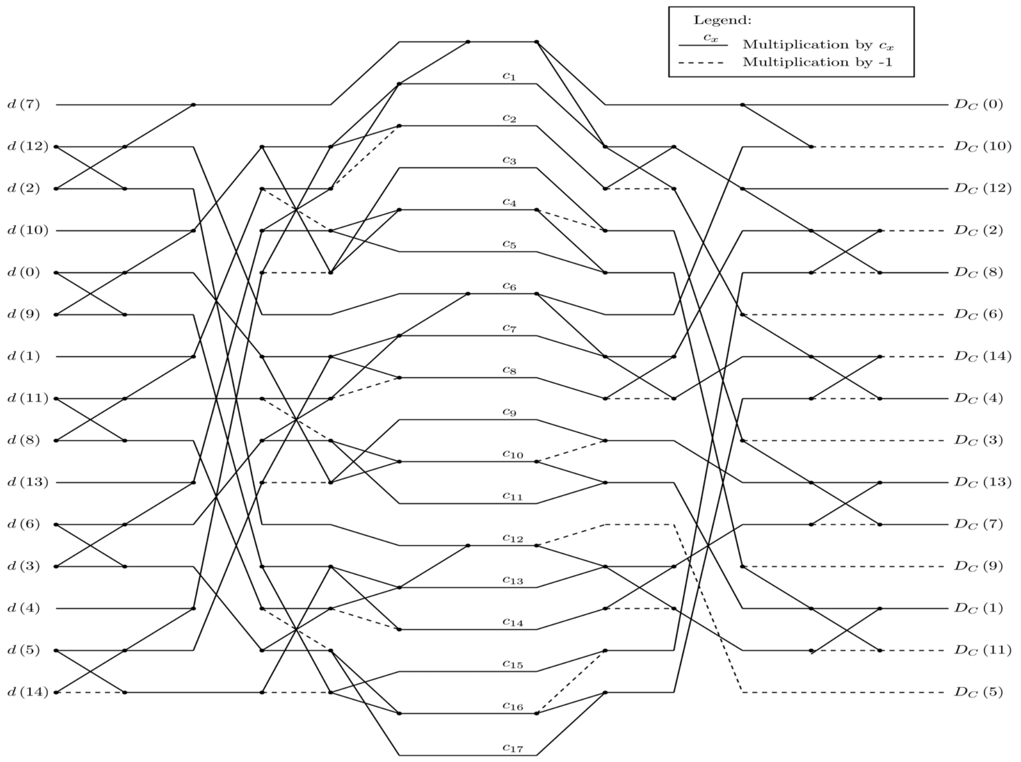


Fig. 6. Fast Factorization of 15-point DCT-II.

TABLE I
COMPUTATIONAL COMPLEXITY OF THE PROPOSED ALGORITHM

Filterbank	N	Algorithm using DCT-II	
		Multiplications	Additions
AAC-ELD Core Encoder	1024	4224	8320
AAC-ELD Core Decoder	1024	4224	8320
AAC-ELD Core Encoder	960	3544	7512
AAC-ELD Core Decoder	960	3544	7512

$$s(n) \triangleq \begin{cases} \left\{x\left(n-\frac{N}{4}\right)-x\left(-\frac{N}{4}-1-n\right)+x\left(\frac{3N}{4}-1-n\right)\right. \\ \qquad \left.-x\left(\frac{3N}{4}+n\right)\right\}, \text{for } 0 \le n < \frac{N}{4} \\ \left\{x\left(n-\frac{N}{4}\right)-x\left(-\frac{N}{4}-1-n\right)+x\left(\frac{3N}{4}-1-n\right)\right. \\ \qquad \left.-x\left(n-\frac{5N}{4}\right)\right\}, \text{for } \frac{N}{4} \le n < \frac{N}{2} \end{cases} \tag{62}$$

Using (61) in (60), we get:

$$X(k) = (-1)^{(k)} \sum_{n=0}^{\frac{N}{2}-1} s(n) \sin\left\{\frac{\pi(2n+1)(2k+1)}{4\left(\frac{N}{2}\right)}\right\}, \quad \text{for } 0 \le k < \frac{N}{2} \tag{63}$$

We realize that the summation in (63) is an $\frac{N}{2}$-point DST-IV. As discussed in the previous sections, this DST-IV can be converted to DCT-IV. Also noting that $\frac{N}{2}$ is even, we finally get:

$$X\left(\frac{N}{2}-1-k\right) = (-1)^{k+1} \sum_{n=0}^{\frac{N}{2}-1} (-1)^n s(n) \times \cos\left\{\frac{\pi(2n+1)(2k+1)}{4\left(\frac{N}{2}\right)}\right\}, \text{for } 0 \le k < \frac{N}{2} \tag{64}$$

Proposition 4 maps the LD-TDAC synthesis filterbanks to DCT-IV. The proof follows the same concept as shown above. To avoid redundancy, the details of proof of Proposition 4 are not presented here.

APPENDIX C
DERIVATION OF FAST ALGORITHM FOR 15-POINT DCT-II

In this appendix, we present an efficient implementation of 15-point DCT-II by an equal-length real-valued FFT. The factorization needed for this algorithm was derived by using a combination of known techniques, such as Heideman mapping [23] and Winograd Fourier Transform Algorithm (WFTA) [26]–[28]. We first discuss Heideman's mapping of odd-length DCT-IIs to equal length real DFTs. This mapping involves only permutations at input and output and, sign changes at the output. 15-point DCT-II is implemented by applying Heideman's mapping to 15-point Winograd Fourier Transform Algorithm (WFTA). The resulting 15-point DCT-II algorithm requires just 14 irrational multiplications, 3 multiplications by dyadic rationals and 67 additions.

A. Heideman's Mapping

To implement DCT-II through DFT, we typically need to use DFT of twice or four times the length of DCT-II [23]. Hence, it is generally inefficient to compute DCT-II by FFT. However, [23] shows that an odd length DCT-II can be implemented by DFT of the same size with just input and output permutations and output sign changes. This is possible because, for odd length DCT-II the set of cosines is exactly the same as the set of sines and cosines in DFT of the same length. This is however not true if the length is even. Thus, an odd length DCT-II can be efficiently implemented by a real valued DFT. The mapping given by Heideman is as follows:

Let $d(n)$ represent the original sequence for which we would like to apply the DCT-II. Define a new sequence $\hat{d}(n)$ as follows:

$$\hat{d}(n) = \begin{cases} d\left((-1)^{n+(N+1)/2+1} n + \frac{N-1}{2}\right), \\ \qquad \text{for } 0 \le n < \frac{N+1}{2} \\ d\left((-1)^{n+(N+1)/2+1} (N-n) + \frac{N-1}{2}\right), \\ \qquad \text{for } \frac{N+1}{2} \le n < N \end{cases} \tag{65}$$

All indices are computed modulo N. $\hat{d}(n)$ is essentially a permutation of the input sequence $d(n)$. Let $D_C(k)$ be the DCT-II of the sequence $d(n)$. Let $\hat{D}_F(k)$ be the DFT of the sequence $\hat{d}(n)$. Then,

$$D_C(2k) = (-1)^k \mathrm{Re}\left[\hat{D}_F(k)\right], 0 \le k < \frac{N+1}{2} \tag{66}$$

$$D_C(N-2k) = (-1)^{k+1} \mathrm{Im}\left[\hat{D}_F(k)\right], 1 \le k < \frac{N+1}{2} \tag{67}$$

As can be seen from the above equations, we need to compute only the first $\frac{N+1}{2}$ points of the DFT. This is equal to the complexity of a DFT of a real-valued sequence of length N.

B. *Winograd Fourier Transform Algorithm (WFTA)*

Winograd short-N DFT modules are the building blocks for constructing the WFTA for longer lengths. The short-N modules are defined for prime lengths. Specifically, we need 3-point and 5-point DFT modules for the 15-point transform. The Winograd DFT modules are based on a fast cyclic convolution algorithm for prime lengths using the theoretically minimum number of multiplications [24], [25], [26]. Winograd mapped this optimum convolution algorithm to DFT using Rader's method [24] to give very efficient DFT modules for prime lengths.

Winograd's algorithm achieves a decomposition of the DFT matrix as shown below. Using the matrix notation in [27],

$$\mathbf{F}_N = \mathbf{S}_{N,J}\mathbf{C}_J\mathbf{T}_{J,N} \tag{68}$$

where, N is a prime number, $\mathbf{F}_N$ is the $N \times N$ DFT matrix, $\mathbf{S}_{N,J}$ is a $N \times J$ addition matrix having only 0's, 1's and -1's, $\mathbf{C}_J$ is a $J \times J$ diagonal matrix contributing to multiplications, $\mathbf{T}_{J,N}$ is a $J \times N$ addition matrix having only 0's, 1's and -1's. J is an integer generally close to N for small values of N. Moreover, the elements of $\mathbf{C}_J$ matrix are either purely real or purely imaginary, so for real input data, we will have just one real multiplication for each element of $\mathbf{C}_J$. Hence, the number of real multiplications will be J. For $N = 3$, J will be 3 and we have the following matrices in factorized form to optimize the number of additions [26]:

$$\mathbf{S}_{3,3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{69}$$

$$\mathbf{C}_3 = \mathrm{diag}\left[1, \quad \cos\left(-\tfrac{2\pi}{3}\right) - 1, \quad j\sin\left(-\tfrac{2\pi}{3}\right)\right] \tag{70}$$

$$\mathbf{T}_{3,3} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \tag{71}$$

For $N = 5$, J will be 6 and we have the following matrices in factorized form to optimize the number of additions:

$$\mathbf{S}_{5,6} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{72}$$

$$\mathbf{C}_6 = \mathrm{diag}\left[1, \quad \frac{\cos u + \cos 2u}{2} - 1, \quad \frac{\cos u - \cos 2u}{2}, \quad j(\sin u + \sin 2u), \quad j\sin 2u, \quad j(\sin u - \sin 2u)\right] \tag{73}$$

where u is defined as:

$$u = \frac{-2\pi}{5} \tag{74}$$

$$\mathbf{T}_{6,5} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{bmatrix} \tag{75}$$

The Winograd Fourier Transform Algorithm (WFTA) for a size $N = N_1N_2$ (N_1 and N_2 are primes) DFT is given by,

$$\mathbf{F}_N = \mathbf{P}_o^{-1}\mathbf{S}_{N,J}\mathbf{C}_J\mathbf{T}_{J,N}\mathbf{P}_i \tag{76}$$

where,

$$\mathbf{S}_{N,J} = (\mathbf{S}_{N_1,J_1} \otimes \mathbf{S}_{N_2,J_2}) \tag{77}$$

$$\mathbf{C}_J = (\mathbf{C}_{J_1} \otimes \mathbf{C}_{J_2}) \tag{78}$$

$$\mathbf{T}_{J,N} = (\mathbf{T}_{J_1,N_1} \otimes \mathbf{T}_{J_2,N_2}) \tag{79}$$

where, $\otimes$ represents Kronecker product. $J = J_1J_2$. $\mathbf{S}_{N,J}$, $\mathbf{T}_{J,N}$ again contain only 1's, -1's and 0's, i.e., they are addition matrices. $\mathbf{C}_J$ is a diagonal matrix contributing only multiplications. $\mathbf{P}_i$ and $\mathbf{P}_o$ are $N \times N$ input and output permutation matrices respectively. Details of the algorithms including definitions of the input and output permutation matrices can be found in [27], [28]. In general, the order of N_1 and N_2 affects the number of additions in the algorithms but the number of multiplications remains constant.

REFERENCES

[1] A. M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems*, 2nd ed. New York, NY, USA: Wiley, 2004.
[2] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–515, Apr. 2000.
[3] A. Spanias, "Speech coding: A tutorial review," *Proc. IEEE*, vol. 82, no. 10, pp. 1541–1582, Oct. 1994.
[4] G. D. T. Schuller and T. Karp, "Modulated filterbanks with arbitrary system delay: Efficient implementations and the time varying case," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 737–748, Mar. 2000.
[5] *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 3: Advanced Audio Coding (AAC), Subpart 4: General Audio Coding (GA) - AAC, TwinVQ, BSAC*, ISO/IEC 14496-3:2005, 2005.
[6] *Information Technology - Generic Coding of Moving Pictures and Associated Audio Information - Part 3: Advanced Audio Coding (AAC), Amendment 9: Enhanced Low Delay AAC*, ISO/IEC 14496-3:2005/Amd 9:2008, Jul. 2008.
[7] *Reference software for AAC-ELD*, ISO/IEC 14496-5:2001/Amd 24:2009, Oct. 2009.
[8] M. Schnell, R. Geiger, M. Schmit, M. Jander, M. Multrus, G. Schuller, and J. Herre, "Enhanced MPEG-4 low delay AAC - Low bitrate high quality communication," in *Proc. 122nd Conv. AES*, Vienna, Austria, May 2007, Preprint #6998.
[9] M. Schnell, R. Geiger, M. Schmidt, M. Multrus, M. Mellar, J. Herre, and G. Schuller, "Low delay filterbanks for enhanced low delay audio coding," in *Proc. IEEE Workshop Applicat. Signal Process, Audio, Acoust.*, Oct. 2007, pp. 235–238.
[10] M. Schnell, M. Schmidt, M. Jander, T. Albert, R. Geiger, V. Ruoppila, P. Ekstrand, M. Lutzky, and B. Grill, "MPEG-4 enhanced low delay AAC - A new standard for high quality communication," in *Proc. 125th AES Conv.*, San Francisco, CA, USA, Oct. 2008, Preprint #7503.
[11] M. Lutzky, M. L. Valero, M. Schnell, and J. Hilpert, "AAC-ELD v2 - The new state of the art in high quality communication audio coding," in *Proc. 131st AES Conv.*, New York, NY, USA, Oct. 2011, Preprint #8516.
[12] R. K. Chivukula, Y. A. Reznik, V. Devarajan, and M. Jayendra-Lakshman, "Fast Algorithms for Low-Delay SBR Filterbanks in MPEG-4 AAC-ELD," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 1022–1031, Mar. 2012.
[13] V. Britanak and K. R. Rao, "A new fast algorithm for the unified forward and inverse MDCT/MDST computation," *Signal Process.*, vol. 82, no. 3, pp. 433–459, Mar. 2002.
[14] V. Britanak and H. J. Lincklaen Arriens, "Fast computational structures for an efficient implementation of the complete TDAC analysis/synthesis MDCT/MDST filter banks," *Signal Process.*, vol. 89, no. 7, pp. 1379–1394, Jul. 2009.
[15] V. Britanak, "A survey of efficient MDCT implementations in MP3 audio coding standard: Retrospective and state-of-the-art," *Signal Process.*, vol. 91, no. 4, pp. 624–672, Apr. 2011.
[16] R. K. Chivukula, Y. A. Reznik, and V. Devarajan, "Efficient algorithms for MPEG-4 AAC-ELD, AAC-LD and AAC-LC filterbanks," in *Proc. Int. Conf. Audio, Lang., Image Process.*, Jul. 2008, pp. 1629–1634.
[17] R. K. Chivukula and Y. A. Reznik, "Efficient implementation of a class of MDCT/IMDCT filterbanks for speech and audio coding applications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2008, pp. 213–216.
[18] Y. A. Reznik and R. K. Chivukula, "Fast 15x15 transform for image and video coding applications," in *Proc. Data Compression Conf.*, Snowbird, UT, USA, Mar. 2009, p. 465.
[19] K. R. Rao, D. N. Kim, and J. J. Hwang, *Fast Fourier Transform - Algorithms and Applications*. New York, NY, USA: Springer, 2010.
[20] C. W. Kok, "Fast algorithm for computing discrete cosine transform," *IEEE Trans. Signal Process.*, vol. 45, no. 3, pp. 757–760, Mar. 1997.
[21] M. Dietz *et al.*, "Spectral band replication, a novel approach in audio coding," in *Proc. 12th Conv. AES*, Munich, Germany, Apr. 2002, Preprint #5553.
[22] H. S. Malvar, *Signal Processing with Lapped Transforms*. Boston, MA, USA: Artech House, 1992.
[23] M. T. Heideman, "Computation of an odd-length DCT from a real-valued DFT of the same length," *IEEE Trans. Signal Process.*, vol. 40, no. 1, pp. 54–61, Jan. 1992.
[24] C. S. Burrus and T. W. Parks, *DFT/FFT and Convolution Algorithms - Theory and Implementation*. New York, NY, USA: Wiley, 1985.
[25] H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. New York, NY, USA: Springer, 1981.
[26] S. Winograd, "On computing the discrete fourier transform," *Math. Comput.*, vol. 32, no. 141, pp. 175–199, Jan. 1978.
[27] H. F. Silverman, "An introduction to programming the winograd fourier transform algorithm (WFTA)," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-25, no. 2, pp. 152–165, Apr. 1977.
[28] B. D. Tseng and W. C. Miller, "Comments on 'An introduction to programming the winograd fourier transform algorithm (WFTA)'," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 3, pp. 268–269, Jun. 1978.

Ravi. K. Chivukula received his Bachelor of Technology degree in electronics and communication engineering from National Institute of Technology, Warangal, India, in 2002. He received the M.S. degree in electrical engineering from The University of Texas at Arlington in 2008. He is currently pursuing the post-graduate program in management at Indian School of Business, India.

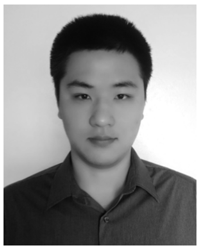
From 2002 to 2006, he was with Emuzed India Pvt. Ltd. working on reference design and optimization of speech and audio codecs. In 2007, he interned with Qualcomm Corporate R&D division on development of fast algorithms for several audio codecs. From 2008 to 2014, he was with Qualcomm as a Staff Engineer designing and developing various multimedia software products.

His interests include multimedia signal processing, information theory and embedded system programming.

Yuriy. A. Reznik (M'97–SM'07) received the Radio Engineer (honors) and graduate school diplomas from Kiev Polytechnic Institute, Kiev, Ukraine, in 1992 and 1995, respectively, and the Ph.D. degree in computer science from Kiev University, Kiev, in 2005.

He is currently a Director, CTO Office at InterDigital, Inc., San Diego, CA. In 2008, he was a Visiting Scholar with the Information Systems Laboratory at Stanford University, Stanford, CA. From 1998 to 2005, he was with RealNetworks, Inc., Seattle, WA, where he was a Principal Engineer and co-developer of RealVideo and RealAudio algorithms for Internet streaming. Since 2001, he was also involved with ITU-T SG16 and MPEG standards committees and made contributions to several speech, audio, and video coding standards, including MPEG-4 ALS, MPEG-4 AVC | ITU-T Rec. H.264, ITU-T Rec. G.718, and MPEG-C Parts 1, 2, and 4.

Dr. Reznik is a member of ACM, AES, and SPIE. He is a co-author of over 70 conference and journal papers, and co-inventor of over 50 (including 11 granted) U.S. patents.



Yanyan Hu received his B.S.E.E. from The University of Texas at Arlington (UTA) in 2012. He is currently pursuing the master degree in electrical engineering in UTA. From 2010 to 2011, he was with the Resonant Sensor Inc. as an undergraduate research assistant working on optimization and upgrading of hardware. From 2012 to 2014, he was with the Virtual Environment Laboratory working as a research student.

His interests include signal/image processing, pattern recognition and computer vision.

Venkat Devarajan (SM'02) studied electrical engineering at Indian Institute of Technology Madras, Chennai, India, where he received the B.S.E.E. and M.S.E.E. degrees in 1973 and 1975, respectively. He performed his doctoral research in the area of color television image compression and received the Ph.D. degree in 1980 from the University of Texas at Arlington.

He worked in the aerospace industry for over a decade. He helped develop photo-based visual systems technology, which has since found wide acceptance in the right simulation community. He was the chief architect of TOPSCENE, the mission rehearsal system used by the U.S. Navy, Air Force, and Army pilots to train their air-to-ground missions. His present research interests are image processing and virtual reality applications to bio-medicine, aerospace, and manufacturing. He is an associate fellow of AIAA.

Mythreya Jayendra Lakshman received his B.E. in electronics and communication engineering from SSN College of Engineering, which is affiliated to the Anna University in Chennai, India, in 2006. He received his M.S. in electrical engineering from the University of Texas at Dallas in 2008 and his Ph.D. in electrical engineering from the University of Texas, Arlington, in 2014. He is currently with Qualcomm Corporate R&D working on computer vision and image processing applications as a Senior Software Engineer.