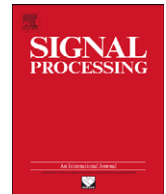




ELSEVIER

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Fast computational structures for an efficient implementation of the complete TDAC analysis/synthesis MDCT/MDST filter banks

Vladimir Britanak^{a,*}, Huibert J. Lincklaen Arriëns^b

^a Institute of Informatics, Slovak Academy of Sciences, Dubravská cesta 9, 845 07 Bratislava, Slovak Republic

^b Delft University of Technology, Department of Electrical Engineering, Mathematics and Computer Science, Mekelweg 4, 2628 CD Delft, The Netherlands

ARTICLE INFO

Article history:

Received 27 May 2008

Received in revised form

8 January 2009

Accepted 10 January 2009

Keywords:

Modified discrete cosine transform

Modified discrete sine transform

Modulated lapped transform

Modulated complex lapped transform

Fast computational structure

MPEG audio coding

ABSTRACT

A new fast computational structure identical both for the forward and backward modified discrete cosine/sine transform (MDCT/MDST) computation is described. It is the result of a systematic construction of a fast algorithm for an efficient implementation of the complete time domain aliasing cancellation (TDAC) analysis/synthesis MDCT/MDST filter banks. It is shown that the same computational structure can be used both for the encoder and the decoder, thus significantly reducing design time and resources. The corresponding generalized signal flow graph is regular and defines new sparse matrix factorizations of the discrete cosine transform of type IV (DCT-IV) and MDCT/MDST matrices. The identical fast MDCT computational structure provides an efficient implementation of the MDCT in MPEG layer III (MP3) audio coding and the Dolby Labs AC-3 codec. All steps to derive the computational structure are described in detail, and to put them into perspective a comprehensive list of references classified into categories is provided covering new research results achieved in the time period 1999–2008 in theoretical and practical developments of TDAC analysis/synthesis MDCT/MDST filter banks (general mathematical, symmetry and special properties, fast MDCT/MDST algorithms and efficient software/hardware implementations of the MDCT in MP3).

Crown Copyright © 2009 Published by Elsevier B.V. All rights reserved.

1. Introduction

The complete time domain aliasing cancellation (TDAC) analysis/synthesis MDCT (modified discrete cosine transform) filter banks [1,2] are the fundamental processing blocks in the current international audio coding standards. Well known modern information technologies for high-quality compression and decompression of digital audio signals in consumer electronics are e.g., the MPEG family: MPEG-1 layer III ISO/IEC 11172-3 [5], MPEG-2 layer III ISO/IEC 13818-3 [6] known as MP3, MPEG-2 AAC [8], MPEG-4 audio ISO/IEC 14496-2 [9] and the recently

developed MPEG-4 high-efficiency AAC (HE-AAC) [10], proprietary digital audio compression algorithms such as Sony ATRAC (adaptive transform acoustics coding)/ATRAC2/SDDS (Sony dynamic digital sound), AT&T perceptual audio coder (PAC) or Lucent Technologies PAC/Enhanced PAC/Multichannel PAC [4,13], and Dolby Labs AC-3/Dolby Digital/Dolby SR.D [7]. A general overview of audio codecs can be found in [3,4,12,13]. In addition, the complete TDAC analysis/synthesis MDCT filter banks are also used in the non-proprietary, patent-and-royalty-free Ogg Vorbis codec [11]. Thus, an efficient implementation of the TDAC MDCT processor has become the key technology to realize low-cost audio decoders in (portable) MP3 players and digital multimedia systems in particular.

Generally, in all audio codecs the size of a data block transformed by the MDCT is variable ($N = 12, 36, 128, 256, 512, 2048$). The computation of the complete analysis/

* Corresponding author. Tel.: +421 2 59411 175; fax: +421 2 54773 271.

E-mail addresses: britanak@savba.sk (V. Britanak), h.j.lincklaenarriens@tudelft.nl (H.J.L. Arriëns).

synthesis MDCT filter banks is the most time-consuming operation, and therefore, the existence of fast, efficient algorithms with a simple and regular structure is very important. The MDCT is equivalent to the modulated lapped transform (MLT) [43] which belongs to the class of lapped transforms. Basis vectors of the MDCT and the corresponding modified discrete sine transform (MDST) form a complex extension of the MLT, the so-called modulated complex lapped transform (MCLT) [47]. The real part of an MCLT corresponds to the MDCT or MLT, and its imaginary part corresponds to the MDST. In the last decade a number of fast algorithms for the efficient computation of MDCT [20–42,52–58], MLT [43–46] and MCLT [47–51] have been proposed/modified/improved. Almost all existing fast algorithms developed up to now employ other discrete sinusoidal unitary transforms such as the discrete Fourier transform (DFT) or discrete cosine/sine transforms of type II and IV (DCT-II/DST-II and DCT-IV/DST-IV) of lower size, or they are based on recursive filter structures [52–58]. Particularly, a proposed efficient MDCT implementation in MP3 audio coding [22] based on the fast algorithm derived in [25] has been sequentially improved and optimized in terms of arithmetic complexity and structural simplicity [24,30–32,34,36]. Recently, MP3 audio decoders for real-time processing have been realized on high-performance programmable DSP processors [60–62,65,70], universal RISC-based ARM processors [63,64,67,69], and implemented into VLSI full-custom ASIC [59,68] or semi-custom circuits (FPGA) [34,66].

In this paper, a new fast computational structure identical both for the forward and backward MDCT/MDST computation is described. It is the result of a systematic construction of a fast algorithm for an efficient implementation of the complete TDAC analysis/synthesis MDCT/MDST filter banks. Consequently, the same computational structure can be used both for the encoder and decoder, thus reducing design time and resources. The corresponding generalized signal flow graph is regular and defines new sparse matrix factorizations of DCT-IV and MDCT/MDST matrices. First, the definitions of complete TDAC analysis/synthesis MDCT and MDST filter banks are presented. Then, the MDCT and MDST as the block transforms applied to a single data block are considered and their general mathematical, symmetry and special properties are discussed in detail. In particular, matrix representations of the MDCT and MDST block transforms, their properties and consequences from the viewpoint of terminology used in the literature are emphasized. The systematic construction of fast analysis/synthesis MDCT filter banks is described in Section 3. In Section 4 the fast MDCT computational structure is derived. Finally, in Section 5 the fast MDCT computational structure is compared with existing fast algorithms and its important characteristics are discussed in detail. This fast MDCT computational structure provides an efficient implementation of the MDCT in MP3 audio coding and the AC-3 codec. It is important to note that the paper is intended to have a tutorial value. For a potential reader a comprehensive list of references is provided covering new research results achieved in the time period 1999–2008 in theoretical and practical developments TDAC analysis/

synthesis MDCT/MDST filter banks (general mathematical, symmetry and special properties, fast MDCT/MDST, MLT and MCLT algorithms and efficient software/hardware implementations of the MDCT in MP3).

2. Definitions, basic facts and notations

This section consists of several subsections. Besides definitions, basic facts and notations being used in the paper, this section covers both known and new theoretical research results referring to the MDCT and MDST filter banks. First, definitions of complete TDAC analysis/synthesis MDCT and MDST filter banks are presented (Section 2.1). When investigating general mathematical properties of the MDCT and MDST, and when developing fast computational structures for their efficient implementation, they are frequently considered as block transforms applied to a single data block. Defining them as block transforms enable us to investigate additional properties such as the periodicity and anti-periodicity of MDCT/MDST transform kernels, symmetry properties of MDCT/MDST basis vectors and to determine a relation between the MDCT and MDST (Section 2.2). An alternative way to represent the MDCT and MDST block transforms is in matrix–vector form. Matrix representations are very powerful tools to analyze MDCT/MDST characteristics of the single data block both in time and frequency domains [76,77]. In particular, on the basis of matrix representations the concept of TDAC and special MDCT/MDST properties having an impact on audio coding performance are better understood (Section 2.3).

2.1. Complete TDAC analysis/synthesis MDCT filter banks

The complete TDAC analysis and synthesis MDCT filter banks are, respectively, defined as [1,2,71]

$$c_k^{(t)} = \sqrt{\frac{4}{N}} \sum_{n=0}^{N-1} w_n x_n^{(t)} \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right],$$

$$k = 0, 1, \dots, \frac{N}{2} - 1, \quad (1)$$

$$\hat{x}_n^{(t)} = \sqrt{\frac{4}{N}} w_n \sum_{k=0}^{(N/2)-1} c_k^{(t)} \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right],$$

$$n = 0, 1, \dots, N - 1, \quad (2)$$

where w_n is a windowing function, and a superscript t denotes the data-block number. N , being the length of the data block is assumed to be an even integer. In the analysis filter bank given by (1), for the t -th data block, N windowed time domain samples $\{x_n^{(t)}\}$ are used to calculate $N/2$ unique transform coefficients $\{c_k^{(t)}\}$. Vice versa, the t -th block of $N/2$ transform coefficients $\{c_k^{(t)}\}$ is used to calculate N windowed time domain aliased samples $\{\hat{x}_n^{(t)}\}$ with the synthesis filter bank given by (2). The complete TDAC analysis/synthesis MDCT filter bank provides critical sampling, overlapping of adjacent data blocks by $N/2$ samples, possesses energy-packing capability and allows perfect reconstruction. To achieve critical sampling in combination with overlapping data

blocks, a subsampling in the frequency domain is performed by the analysis MDCT filter bank. This subsampling introduces aliasing in the time domain, but this can be cancelled by overlapping and adding of two adjacent recovered data blocks by the synthesis filter bank. This procedure is known as TDAC [1,2,71]. It is important to note that the time domain aliased data sequence $\{\hat{x}_n^{(t)}\}$ does not correspond to the original data sequence $\{x_n^{(t)}\}$.

Two succeeding data blocks t and $t + 1$ are overlapped by $N/2$ samples so that for each data block $N/2$ new time domain samples are processed. For a smooth block overlapping a windowing function w_n is applied to $\{x_n^{(t)}\}$. By applying analysis and synthesis MDCT filter banks a time domain aliasing error is introduced which is independent for each half of the data block. This leads to the realization of adaptive block-size switching (processing with variable-block size) [12]. Flexible dynamic block-size switching is an important concept to reduce pre-echo effects in MDCT-based audio codecs. The aliasing error is cancelled (or perfect reconstruction is accomplished) by adding outputs of the synthesis MDCT filter bank of two succeeding data blocks t and $t + 1$ in the overlapped part as follows [71]:

$$x_n^{(t+1)} = x_{(N/2)+n}^{(t)} = \hat{x}_{(N/2)+n}^{(t)} + \hat{x}_n^{(t+1)}, \quad n = 0, 1, \dots, \frac{N}{2} - 1. \quad (3)$$

To ensure TDAC, the windowing functions of two succeeding data blocks have to satisfy the so-called perfect-reconstruction conditions in their overlapped part. A sufficient condition for TDAC is given by

$$\begin{aligned} w_n^2 + w_{(N/2)+n}^2 &= 1, \\ w_n &= w_{N-1-n}, \quad \text{or} \quad w_{(N/2)+n} = w_{(N/2)-1-n}, \\ n &= 0, 1, \dots, \frac{N}{2} - 1. \end{aligned} \quad (4)$$

Note that the perfect-reconstruction conditions given by (4) do not imply that the windowing functions for two succeeding overlapped data blocks must be identical [12]. A frequently used windowing function, for example in MP3 [5,6] and MPEG-2 AAC [8], satisfying perfect-reconstruction conditions (4) and producing aliasing cancellation, being symmetrical and identical both for the analysis and synthesis filter banks is given by [3,12,13,43]

$$w_n = \sin\left[\frac{\pi}{2N}(2n + 1)\right], \quad n = 0, 1, \dots, N - 1. \quad (5)$$

A plot of this sine windowing function for $N = 2048$ is shown in Fig. 1.

As another example, the Ogg Vorbis audio compression algorithm (fully open, non-proprietary and patent-royalty-free distributed) uses the windowing function given by [11,17]

$$w_n = \sin\left[\frac{\pi}{2} \sin^2\left(\frac{\pi}{2N}(2n + 1)\right)\right], \quad n = 0, 1, \dots, N - 1. \quad (6)$$

A plot of the Ogg Vorbis windowing function for $N = 2048$ is shown in Fig. 2.

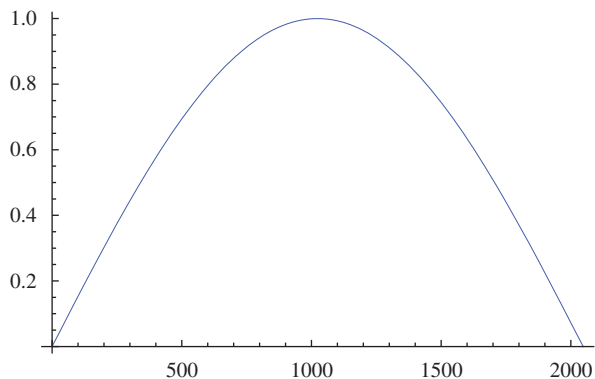


Fig. 1. Plot of sine windowing function for $N = 2048$.

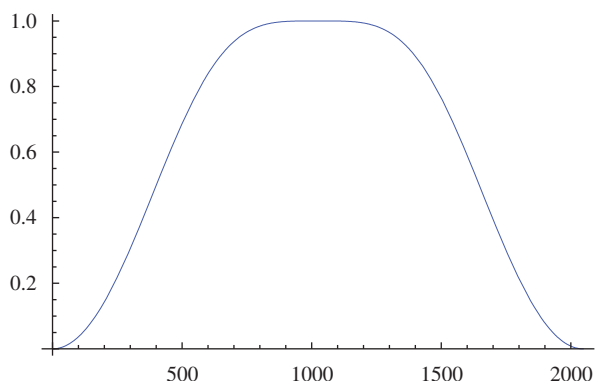


Fig. 2. Plot of the Ogg Vorbis windowing function for $N = 2048$.

Note 1. The corresponding complete TDAC analysis/synthesis MDST filter banks [25,47] can be defined in a similar way.

2.2. MDCT and MDST as the block transforms

The input data sequence $\{x_n\}$ is assumed to be windowed by a windowing function satisfying Eq. (4) before its transformation. Then, the forward and backward MDCT block transforms are, respectively, specified in simplified form as

$$\begin{aligned} c_k &= \sqrt{\frac{4}{N}} \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{2N}\left(2n + 1 + \frac{N}{2}\right)(2k + 1)\right], \\ k &= 0, 1, \dots, \frac{N}{2} - 1, \end{aligned} \quad (7)$$

$$\begin{aligned} \hat{x}_n^{MDCT} &= \sqrt{\frac{4}{N}} \sum_{k=0}^{(N/2)-1} c_k \cos\left[\frac{\pi}{2N}\left(2n + 1 + \frac{N}{2}\right)(2k + 1)\right], \\ n &= 0, 1, \dots, N - 1. \end{aligned} \quad (8)$$

It is noted that the MDCT is equivalent to the MLT [43]. The original definition of the MLT is obtained by substituting $N = 2M$ into Eqs. (7) and (8).

Similarly, the corresponding forward and backward MDST block transforms are, respectively, specified in

simplified form as [25,47]

$$s_k = \sqrt{\frac{4}{N}} \sum_{n=0}^{N-1} x_n \sin \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right],$$

$$k = 0, 1, \dots, \frac{N}{2} - 1, \quad (9)$$

$$\hat{x}_n^{MDST} = \sqrt{\frac{4}{N}} \sum_{k=0}^{(N/2)-1} s_k \sin \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right],$$

$$n = 0, 1, \dots, N - 1. \quad (10)$$

The MDCT and MDST basis vectors form the MCLT [47], whose real part corresponds to the MDCT or MLT and imaginary part is the MDST. In contrast to discrete unitary transforms, the MDCT and MDST have a special property: the original input data sequence cannot be perfectly reconstructed from a single block of MDCT and MDST coefficients.

MDCT/MDST sequences $\{c_k\}/\{s_k\}$ possess even anti-symmetry/symmetry properties given by [18,33]

$$c_{N-k-1} = -c_k, \quad s_{N-k-1} = s_k, \quad k = 0, 1, \dots, \frac{N}{2} - 1, \quad (11)$$

while the time domain aliased data sequences $\{\hat{x}_n^{MDCT}\}/\{\hat{x}_n^{MDST}\}$ recovered by the backward MDCT/MDST possess the following symmetries [18,33]:

$$\hat{x}_n^{MDCT} = -\hat{x}_{(N/2)-1-n}^{MDCT}, \quad \hat{x}_{(N/2)+n}^{MDCT} = \hat{x}_{N-1-n}^{MDCT},$$

$$\hat{x}_n^{MDST} = \hat{x}_{(N/2)-1-n}^{MDST}, \quad \hat{x}_{(N/2)+n}^{MDST} = -\hat{x}_{N-1-n}^{MDST},$$

$$n = 0, 1, \dots, \frac{N}{4} - 1. \quad (12)$$

The symmetry properties (11) and (12) can be simply verified by the proper substitution into Eqs. (7)–(10). We note that the symmetry properties (11) and (12) also hold if the input data sequence $\{x_n\}$ is windowed. From (11) it follows that only $N/2$ coefficients are unique in the MDCT and MDST sequences. Further, it can be easily seen that the time domain aliased data sequences $\{\hat{x}_n^{MDCT}\}/\{\hat{x}_n^{MDST}\}$ exhibit two local symmetries (odd/even symmetry in the first half and even/odd symmetry in the second half). From an algorithmic point of view this means that it is sufficient to compute only the time domain aliased samples \hat{x}_n^{MDCT} and $\hat{x}_{(N/2)+n}^{MDCT}$ for $n = 0, 1, \dots, (N/4) - 1$ by the backward MDCT.

2.2.1. Periodicity and anti-periodicity of MDCT/MDST transform kernels

Special kinds of data sequences, the periodic and anti-periodic sequences, are fundamental notions in harmonic analysis, convolution and correlation of signals. Now we recall the definitions of periodic and anti-periodic sequences [19].

Definition 1. A data sequence $\{y_n\}$ is called a periodic sequence if $y_{n+M} = y_n$, where $M > 0$ is called the period of periodic sequence $\{y_n\}$.

Definition 2. A data sequence $\{y_n\}$ is called an anti-periodic sequence if $y_{n+M} = -y_n$, where $M > 0$ is the period of anti-periodic sequence $\{y_n\}$.

An anti-periodic sequence $\{y_n\}$ may be treated as a periodic sequence with period $2M$ because $y_{n+2M} = -y_{n+M} = y_n$. However, the properties of $\{y_n\}$ depend only upon its values in one period M . The periodicity and anti-periodicity of sequences are closely related to their symmetry and anti-symmetry properties, respectively. Properties (sums and products) of periodic and anti-periodic sequences with a common period can be found in [19].

Denoting the MDCT and MDST transform kernels, respectively, as

$$t_{k,n}^{(c)} = \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right],$$

$$t_{k,n}^{(s)} = \sin \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right], \quad (13)$$

and substituting $n + N$, and then $n + 2N$ for n into (13) we obtain

$$t_{k,n}^{(c)} = -t_{k,n+N}^{(c)}, \quad t_{k,n}^{(c)} = t_{k,n+2N}^{(c)},$$

$$t_{k,n}^{(s)} = -t_{k,n+N}^{(s)}, \quad t_{k,n}^{(s)} = t_{k,n+2N}^{(s)}, \quad \forall k. \quad (14)$$

Eq. (14) implies that the MDCT and MDST transform kernels are anti-periodic sequences with period N , and periodic sequences with period $2N$.

2.2.2. Symmetry properties of MDCT/MDST basis vectors

For a given N , consider the MDCT and MDST transform kernels given by (13). One can observe that the MDCT and MDST basis vectors exhibit the following local symmetries:

$$t_{k,(N/2)-1-n}^{(c)} = -t_{k,n}^{(c)}, \quad t_{k,(N/2)+n}^{(c)} = t_{k,N-1-n}^{(c)},$$

$$t_{k,(N/2)-1-n}^{(s)} = t_{k,n}^{(s)}, \quad t_{k,(N/2)+n}^{(s)} = -t_{k,N-1-n}^{(s)}, \quad \forall k,$$

$$n = 0, 1, \dots, \frac{N}{4} - 1. \quad (15)$$

The symmetry properties of the MDCT and MDST basis vectors (15) can be simply verified by a proper substitution into Eq. (13). Note that they are quite similar to those of the time domain aliased data sequences $\{\hat{x}_n^{MDCT}\}/\{\hat{x}_n^{MDST}\}$ given by (12).

2.2.3. Relation between MDCT and MDST

There exists a simple relation between the MDCT and MDST given in [25]

$$c_{(N/2)-k-1} = (-1)^{N/4} \sum_{n=0}^{N-1} (-1)^n x_n \sin \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right],$$

$$k = 0, 1, \dots, \frac{N}{2} - 1. \quad (16)$$

Also

$$s_{(N/2)-k-1} = (-1)^{N/4} \sum_{n=0}^{N-1} (-1)^n x_n \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right],$$

$$k = 0, 1, \dots, \frac{N}{2} - 1, \quad (17)$$

and this fact results in a simple method to compute the MDST using a fast MDCT computational structure.

Note 2. The simple relation between the MDCT and the MDST allows us to concentrate on the investigation of a fast MDCT computational structure only.

2.3. Matrix representations of the MDCT/MDST

Consider the forward and backward MDCT/MDST defined by Eqs. (7), (8)/(9), (10), respectively. Based on symmetry properties of the MDCT and MDST coefficients, only $N/2$ rows of the MDCT and MDST matrices are linearly independent. Therefore, let $C_{(N/2) \times N}$ and $S_{(N/2) \times N}$ be the $(N/2) \times N$ MDCT and MDST matrices, respectively. Then $[C_{(N/2) \times N}]^T = C_{N \times (N/2)}$ and $[S_{(N/2) \times N}]^T = S_{N \times (N/2)}$, where T denotes transposition. Next, let $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]$, $\mathbf{c} = [c_0, c_1, \dots, c_{(N/2)-1}]$ and $\mathbf{s} = [s_0, s_1, \dots, s_{(N/2)-1}]$ be row vectors and \mathbf{x}^T , \mathbf{c}^T and \mathbf{s}^T their corresponding column representations. Then Eqs. (7) and (9) for the forward MDCT and MDST can be, respectively, written in the equivalent matrix-vector form as [33]

$$\mathbf{c}^T = C_{(N/2) \times N} \mathbf{x}^T, \quad \mathbf{s}^T = S_{(N/2) \times N} \mathbf{x}^T, \quad (18)$$

while Eqs. (8) and (10) for the backward MDCT and MDST can be, respectively, written as

$$\begin{aligned} [\hat{\mathbf{x}}^{MDCT}]^T &= C_{N \times (N/2)} \mathbf{c}^T = [C_{(N/2) \times N}]^T \mathbf{c}^T, \\ [\hat{\mathbf{x}}^{MDST}]^T &= S_{N \times (N/2)} \mathbf{s}^T = [S_{(N/2) \times N}]^T \mathbf{s}^T. \end{aligned} \quad (19)$$

Based on the matrix representation of the MDCT and MDST it was shown in [18,33] that the transposed MDCT and MDST matrices are pseudoinverses of their corresponding forward transform matrices. Hence, the forward and backward MDCT/MDST are actually pseudoinverse pairs.

The pseudoinverse matrix [76,77] and its properties provide an elegant mathematical tool to characterize the MDCT/MDST in a matrix representation. If we consider the forward MDCT/MDST given by (18) to be overdetermined systems of linear equations, then the time domain aliased data sequences $\{\hat{x}_n^{MDCT}\}/\{\hat{x}_n^{MDST}\}$ in (19) for given MDCT/MDST coefficients can be interpreted as least squares solutions, i.e., solutions with minimum norm. For the MDCT/MDST matrices $C_{(N/2) \times N}/S_{(N/2) \times N}$ and their transposed versions $C_{N \times (N/2)}/S_{N \times (N/2)}$ or their pseudoinverses, the following relations hold [18]:

$$\begin{aligned} C_{(N/2) \times N} [C_{(N/2) \times N}]^T &= C_{(N/2) \times N} C_{N \times (N/2)} = S_{(N/2) \times N} [S_{(N/2) \times N}]^T \\ &= S_{(N/2) \times N} S_{N \times (N/2)} = 2I_{(N/2)}, \end{aligned} \quad (20)$$

where $I_{N/2}$ is the identity matrix of order $N/2$, and

$$\begin{aligned} [C_{(N/2) \times N}]^T C_{(N/2) \times N} &= C_{N \times (N/2)} C_{(N/2) \times N} \\ &= \begin{pmatrix} I_{N/4} & -J_{N/4} & 0 & 0 \\ -J_{N/4} & I_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & J_{N/4} \\ 0 & 0 & J_{N/4} & I_{N/4} \end{pmatrix}, \end{aligned} \quad (21)$$

$$\begin{aligned} [S_{(N/2) \times N}]^T S_{(N/2) \times N} &= S_{N \times (N/2)} S_{(N/2) \times N} \\ &= \begin{pmatrix} I_{N/4} & J_{N/4} & 0 & 0 \\ J_{N/4} & I_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & -J_{N/4} \\ 0 & 0 & -J_{N/4} & I_{N/4} \end{pmatrix}, \end{aligned} \quad (22)$$

where $I_{N/4}$ is the identity matrix, and $J_{N/4}$ is the reverse ordered identity matrix, both of order $N/4$.

Substituting Eqs. (18) into (19) (i.e., performing the forward and backward MDCT/MDST) and using relations (21), (22) we have

$$\begin{aligned} [\hat{\mathbf{x}}^{MDCT}]^T &= C_{N \times (N/2)} \mathbf{c}^T = C_{N \times (N/2)} C_{(N/2) \times N} \mathbf{x}^T \\ &= \begin{pmatrix} I_{N/4} & -J_{N/4} & 0 & 0 \\ -J_{N/4} & I_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & J_{N/4} \\ 0 & 0 & J_{N/4} & I_{N/4} \end{pmatrix} \mathbf{x}^T, \end{aligned} \quad (23)$$

$$\begin{aligned} [\hat{\mathbf{x}}^{MDST}]^T &= S_{N \times (N/2)} \mathbf{s}^T = S_{N \times (N/2)} S_{(N/2) \times N} \mathbf{x}^T \\ &= \begin{pmatrix} I_{N/4} & J_{N/4} & 0 & 0 \\ J_{N/4} & I_{N/4} & 0 & 0 \\ 0 & 0 & I_{N/4} & -J_{N/4} \\ 0 & 0 & -J_{N/4} & I_{N/4} \end{pmatrix} \mathbf{x}^T. \end{aligned} \quad (24)$$

From Eqs. (23) and (24) it follows that the time domain aliased data samples $\{\hat{x}_n^{MDCT}\}$ can be derived explicitly in terms of the original data samples $\{x_n\}$ as

$$\begin{aligned} \hat{x}_n^{MDCT} &= x_n - x_{(N/2)-1-n}, & \hat{x}_{(N/2)-1-n}^{MDCT} &= -\hat{x}_n^{MDCT}, \\ \hat{x}_{(N/2)+n}^{MDCT} &= x_{(N/2)+n} + x_{N-1-n}, & \hat{x}_{N-1-n}^{MDCT} &= \hat{x}_{(N/2)+n}^{MDCT}, \\ n &= 0, 1, \dots, \frac{N}{4} - 1, \end{aligned} \quad (25)$$

while the time domain aliased data sequence $\{\hat{x}_n^{MDST}\}$ can be found as

$$\begin{aligned} \hat{x}_n^{MDST} &= x_n + x_{(N/2)-1-n}, & \hat{x}_{(N/2)-1-n}^{MDST} &= \hat{x}_n^{MDST}, \\ \hat{x}_{(N/2)+n}^{MDST} &= x_{(N/2)+n} - x_{N-1-n}, & \hat{x}_{N-1-n}^{MDST} &= -\hat{x}_{(N/2)+n}^{MDST}, \\ n &= 0, 1, \dots, \frac{N}{4} - 1. \end{aligned} \quad (26)$$

In Eqs. (25) and (26) we can clearly observe the recovered time domain aliased data sequences including their forms both in terms of original data samples and their symmetry properties given by (12). With Eqs. (25) and (26) in mind we can compute exactly the results of applying the forward and backward MDCT and MDST.

2.3.1. Perfect-reconstruction conditions and windowing operation

Let $\{x_n^{(t)}\}$ and $\{x_n^{(t+1)}\}$ be two adjacent overlapped data blocks. Transform these data blocks by the forward MDCT followed by the backward MDCT. Assuming that no changes are made in data blocks during the transformation, we find that the original signal in the overlapped part is perfectly reconstructed even without windowing. Indeed, according to the overlap and add procedure given by (3) and using Eq. (25), whereby in the overlapped part the following relations hold: $x_n^{(t+1)} = x_{(N/2)+n}^{(t)}$ and $x_{(N/2)-1-n}^{(t+1)} = x_{N-1-n}^{(t)}$, we have

$$\begin{aligned} \hat{x}_{(N/2)+n}^{(t)} + \hat{x}_n^{(t+1)} &= (x_{(N/2)+n}^{(t)} + x_{N-1-n}^{(t)}) + (x_n^{(t+1)} - x_{(N/2)-1-n}^{(t+1)}) \\ &= 2x_{(N/2)+n}^{(t)}, \quad n = 0, 1, \dots, \frac{N}{4} - 1, \end{aligned} \quad (27)$$

and the second half of the original data sequence $\{2x_{N-1-n}^{(t)}\}$ in the overlapped part can be obtained from the symmetry properties of $\{x_{(N/2)+n}^{(t)}\}$ and $\{x_n^{(t+1)}\}$.

The main objective in audio compression applications is to represent the transformed audio signal by fewer bits, while keeping the audio quality at an acceptable level. In modern encoders, signal parts that are not audible are removed resulting in a loss of information. Consequently, after coding and decoding the transformed signal is changed slightly and we can no longer expect a perfect reconstruction compared with the original audio signal. Of course, the remaining information that is passed should be analyzed and synthesized without errors. A serious problem here is that when data blocks are obtained using rectangular windows, sudden signal changes (“discontinuities”) at the block boundaries are to be expected, that will affect the coded data in an unrecoverable way. To eliminate these so-called “blocking artifacts”, each data block is multiplied by a windowing function such that the data block ends smoothly at both boundaries, while keeping overall gain constant during the block transition. In order to accomplish this while keeping the perfect-reconstruction property for the analysis/synthesis process, the windowing functions are applied to both the input and output of the transform procedure as follows.

Let $w_n^{(0)}$ and $w_n^{(1)}$ be the windowing functions applied to the input data blocks t and $t + 1$, respectively. Assume the windowed data blocks $\{w_n^{(0)}x_n^{(t)}\}$, $\{w_n^{(1)}x_n^{(t+1)}\}$, $n = 0, 1, \dots, N - 1$ are transformed by the forward MDCT. Let $h_n^{(0)}$ and $h_n^{(1)}$ be the windowing functions applied to the outputs after performing the backward MDCT. Then, from Eq. (27) we have

$$\begin{aligned} & h_{(N/2)+n}^{(0)}(w_{(N/2)+n}^{(0)}x_{(N/2)+n}^{(t)} + w_{N-1-n}^{(0)}x_{N-1-n}^{(t)}) \\ & + h_n^{(1)}(w_n^{(1)}x_{(N/2)+n}^{(t)} - w_{(N/2)-1-n}^{(1)}x_{N-1-n}^{(t)}) \\ & = (h_{(N/2)+n}^{(0)}w_{(N/2)+n}^{(0)} + h_n^{(1)}w_n^{(1)})x_{(N/2)+n}^{(t)} \\ & + (h_{(N/2)+n}^{(0)}w_{N-1-n}^{(0)} - h_n^{(1)}w_{(N/2)-1-n}^{(1)})x_{N-1-n}^{(t)}. \end{aligned}$$

In order to recover the original data sequence $\{x_{n+(N/2)}^{(t)}\}$, $n = 0, 1, \dots, (N/2) - 1$, we need the coefficient of $x_{(N/2)+n}^{(t)}$ to be one and the coefficient of $x_{N-1-n}^{(t)}$ to be zero. In other words, the perfect-reconstruction conditions for windowing functions have to be satisfied as follows:

$$\begin{aligned} & h_{(N/2)+n}^{(0)}w_{(N/2)+n}^{(0)} + h_n^{(1)}w_n^{(1)} = 1, \\ & h_{(N/2)+n}^{(0)}w_{N-1-n}^{(0)} - h_n^{(1)}w_{(N/2)-1-n}^{(1)} = 0, \\ & n = 0, 1, \dots, \frac{N}{2} - 1. \end{aligned} \quad (28)$$

Usually, the same windowing functions are used for all data blocks, so we can drop the superscripts, i.e., $w_n = w_n^{(0)} = w_n^{(1)}$ and $h_n = h_n^{(0)} = h_n^{(1)}$. If $w_n = h_n$ is identical both for the analysis and synthesis MDCT filter banks, then we obtain the perfect-reconstruction conditions given by (4). The reader is referred to a draft document [17] where mathematical properties of the MDCT are proved using basic trigonometry only.

2.3.2. Special properties of the MDCT/MDST

Studies of the MDCT and its implications to audio coding and error concealment are presented in [15,16]. In these, characteristics of the MDCT of a single data block both in time and frequency domains are analyzed (symmetry and non-orthogonal properties, energy-compaction capability and the concept of TDAC), and their impact on audio coding performance is discussed. In particular, based on Fourier frequency analysis the authors formulate conditions in which MDCT coefficients become zero even with non-zero time domain samples in the single data block. Also, conditions are derived in which the original time domain samples can be perfectly reconstructed by performing a forward and backward MDCT without even an applying an overlap and add procedure. These conditions constitute special or the so-called peculiar properties of the MDCT. In this paper, they are presented and verified based on the matrix representation compared to [15,16], where a relationship between the MDCT and a generalized DFT is exploited.

Claim 1. If the input data sequence $\{x_n\}$ exhibits a local symmetry such that

$$x_n = x_{(N/2)-1-n}, \quad x_{(N/2)+n} = -x_{N-1-n}, \quad n = 0, 1, \dots, \frac{N}{4} - 1, \quad (29)$$

then the MDCT coefficients are degenerated to zero, i.e., $c_k = 0$ for $k = 0, 1, \dots, (N/2) - 1$.

In order to prove this property, let us split the forward N -point MDCT given by (7) into two sums as

$$\begin{aligned} c_k & = \sum_{n=0}^{(N/2)-1} x_n \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right] \\ & + \sum_{n=(N/2)}^{N-1} x_n \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right] \\ & = \sum_{n=0}^{(N/2)-1} x_n \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right] \\ & + \sum_{n=0}^{(N/2)-1} x_{(N/2)+n} \cos \left[\frac{\pi}{2N} \left(2n + 1 + 3\frac{N}{2} \right) (2k + 1) \right], \end{aligned}$$

and substituting $(N/2) - 1 - n$ for n into both sums we get

$$\begin{aligned} c_k & = \sum_{n=0}^{(N/4)-1} (x_n - x_{(N/2)-1-n}) \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right] \\ & + \sum_{n=0}^{(N/4)-1} (x_{(N/2)+n} + x_{N-1-n}) \\ & \times \cos \left[\frac{\pi}{2N} \left(2n + 1 + 3\frac{N}{2} \right) (2k + 1) \right] \\ & = \sum_{n=0}^{(N/4)-1} (x_n - x_{(N/2)-1-n}) \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right] \\ & + \sum_{n=0}^{(N/4)-1} -(x_{(N/2)+n} + x_{N-1-n}) \\ & \times \cos \left[\frac{\pi}{2N} \left(2n + 1 - \frac{N}{2} \right) (2k + 1) \right]. \end{aligned} \quad (30)$$

Now, the special property stated by Claim 1 immediately follows from Eq. (30). The property also illustrates that the

MDCT does not fulfill Parseval's theorem [15]. Note that the N -point MDCT has been decomposed into two $N/4$ -point MDCT transforms.

Claim 2. If the input data sequence $\{x_n\}$ exhibits a local symmetry such that

$$x_n = -x_{(N/2)-1-n}, \quad x_{(N/2)+n} = x_{N-1-n}, \quad n = 0, 1, \dots, \frac{N}{4} - 1, \quad (31)$$

then, by the forward and backward MDCT, the original input data sequence will be perfectly reconstructed without the need for an overlap and add procedure, i.e., $\hat{x}_n = x_n$ for $n = 0, 1, \dots, N-1$.

This property immediately follows from Eqs. (25) and (30). The properties stated by Claims 1 and 2 are very special theoretical cases which are rarely occurring in real audio coding applications, especially after the proper windowing operation. Nevertheless, in both cases the time domain aliased samples can still be perfectly reconstructed by the overlap and add procedure. However, based on Claim 1 if the input signal is close to (29) then the MDCT spectrum will show an unwanted and unrecoverable behavior. For completeness, additional properties are presented in the following text as partial cases of Claim 2.

Claim 2.1. If the input data sequence $\{x_n\}$ exhibits a local symmetry such that

$$x_n = x_{(N/2)-1-n}, \quad x_{(N/2)+n} = x_{N-1-n}, \quad n = 0, 1, \dots, \frac{N}{4} - 1, \quad (32)$$

then, after applying the forward and backward MDCT, the first half of the recovered data sequence will be equal to zero, i.e., $\hat{x}_n = 0$, $n = 0, 1, \dots, (N/2) - 1$, and the second half of the original data sequence $\{x_{(N/2)+n}\}$ will be perfectly reconstructed without an overlap and add procedure, i.e., $x_{(N/2)+n} = \hat{x}_{(N/2)+n}$, $n = 0, 1, \dots, (N/2) - 1$.

Claim 2.2. If the input data sequence $\{x_n\}$ exhibits a local symmetry such that

$$x_n = -x_{(N/2)-1-n}, \quad x_{(N/2)+n} = -x_{N-1-n}, \quad n = 0, 1, \dots, \frac{N}{4} - 1, \quad (33)$$

then, after applying the forward and backward MDCT, the first half of the original input data sequence $\{x_n\}$ will be perfectly reconstructed without the need for an overlap and add procedure, i.e., $x_n = \hat{x}_n$, $n = 0, 1, \dots, (N/2) - 1$, and the second half of the recovered data sequence will be equal to zero, i.e., $\hat{x}_{(N/2)+n} = 0$, $n = 0, 1, \dots, (N/2) - 1$.

Properties stated by Claims 2.1 and 2.2 also follow from Eqs. (25) and (30).

Note 3. The special properties of the MDST can be derived in a similar way.

2.3.3. Consequences of MDCT/MDST matrix representations

From a viewpoint of terminology used in the literature the matrix representations of the MDCT and MDST

considered as the block transforms imply two very important facts:

1. In contrary to the discrete trigonometric transforms [75] which are represented by square invertible orthogonal/orthonormal matrices of order N (therefore, forward and inverse transform), the MDCT and MDST are represented by nonsquare $(N/2) \times N$ matrices for which matrix inversions are not defined (actually, there exist their pseudoinverses [18,76]). Consequently, the notion of "inverse MDCT/MDST", frequently used in the literature, is vague from a standpoint of matrix theory. The more comprehensive notion to be used is "backward MDCT/MDST" (preferred) or "reverse MDCT/MDST". The second argument supporting such a conclusion is that compared to discrete unitary transforms, the MDCT/MDST do not fulfill Parseval's theorem, i.e., its time domain energy is not equal to its frequency domain energy. There exist several published papers pointing out and discussing this topic [14–17].
2. Further, the discrete trigonometric transforms [75] for input data sequences of length N generate N unique frequency coefficients (therefore, N -point forward and inverse transform). Hence, the notion of " N -point" is naturally associated with the length of a input data sequence. In the case of the MDCT/MDST, the forward MDCT/MDST for input time domain sequences of lengths N generate $N/2$ unique frequency coefficients while the backward MDCT/MDST for $N/2$ input frequency domain coefficients generate N time domain aliased samples. Therefore, we would use the notions N -point forward MDCT/MDST and $N/2$ -point backward MDCT/MDST.

3. Fast TDAC analysis/synthesis MDCT filter banks

For a given N the complete TDAC analysis/synthesis MDCT filter bank given by (1)/(2) using the direct approach requires totally $N/2(N+2)$ multiplications and $N/2(N-1)$ additions. In the following sections we show that the total computational complexity can be significantly reduced [43,44].

3.1. Fast analysis MDCT filter bank

Consider the TDAC analysis MDCT filter bank given by (1). Following the same procedure which leads to Eq. (30) but with the windowing operation included, by splitting the analysis MDCT filter bank into two sums we have

$$\begin{aligned} c_k^{(t)} &= \sum_{n=0}^{(N/4)-1} (w_n x_n^{(t)} - w_{(N/2)-1-n} x_{(N/2)-1-n}^{(t)}) \\ &\quad \times \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right] \\ &\quad + \sum_{n=0}^{(N/4)-1} -(w_{(N/2)+n} x_{(N/2)+n}^{(t)} + w_{N-1-n} x_{N-1-n}^{(t)}) \\ &\quad \times \cos \left[\frac{\pi}{2N} \left(2n + 1 - \frac{N}{2} \right) (2k + 1) \right], \end{aligned}$$

or

$$c_k^{(t)} = \sum_{n=0}^{(N/4)-1} y_n^{(t)} \cos \left[\frac{\pi}{2N} \left(2n + 1 + \frac{N}{2} \right) (2k + 1) \right] + \sum_{n=0}^{(N/4)-1} y_n''^{(t)} \cos \left[\frac{\pi}{2N} \left(2n + 1 - \frac{N}{2} \right) (2k + 1) \right],$$

where

$$y_n^{(t)} = w_n x_n^{(t)} - w_{(N/2)-1-n} x_{(N/2)-1-n}^{(t)},$$

$$y_n''^{(t)} = -w_{(N/2)+n} x_{(N/2)+n}^{(t)} - w_{N-1-n} x_{N-1-n}^{(t)},$$

$$n = 0, 1, \dots, \frac{N}{4} - 1.$$

To eliminate the terms $\pm N/2$ in the cosine transform kernels of the first and second sum we substitute $n - (N/4)$ and $(N/4) - 1 - n$ for n , respectively, and we get

$$c_k^{(t)} = \sum_{n=N/4}^{(N/2)-1} y_{n-(N/4)}^{(t)} \cos \left[\frac{\pi}{2N} (2n + 1)(2k + 1) \right] + \sum_{n=(N/4)-1}^0 y_{(N/4)-1-n}''^{(t)} \cos \left[\frac{\pi}{2N} (2n + 1)(2k + 1) \right].$$

$\{y_n^{(t)}\}$ and $\{y_n''^{(t)}\}$ are two independent $N/4$ -point time domain aliased subsequences and concatenating them into one sum, the analysis MDCT filter bank is reduced to $N/2$ -point DCT of type IV (DCT-IV) of $\{y_n^{(t)}\}$ as

$$c_k^{(t)} = \sum_{n=0}^{(N/2)-1} y_n^{(t)} \cos \left[\frac{\pi}{4(N/2)} (2n + 1)(2k + 1) \right],$$

$$k = 0, 1, \dots, \frac{N}{2} - 1, \quad (34)$$

where

$$y_{(N/4)+n}^{(t)} = w_n x_n^{(t)} - w_{(N/2)-1-n} x_{(N/2)-1-n}^{(t)},$$

$$y_{(N/4)-1-n}^{(t)} = -w_{(N/2)+n} x_{(N/2)+n}^{(t)} - w_{N-1-n} x_{N-1-n}^{(t)},$$

$$n = 0, 1, \dots, \frac{N}{4} - 1. \quad (35)$$

Eqs. (34) and (35) define the fast analysis MDCT filter bank or fast MLT algorithm [43,44]. We note that the permutation (35) without windowing operation can be written in matrix-vector form as

$$\mathbf{y}^T = \begin{pmatrix} 0 & 0 & -J_{N/4} & -I_{N/4} \\ I_{N/4} & -J_{N/4} & 0 & 0 \end{pmatrix} \mathbf{x}^T. \quad (36)$$

3.2. Fast synthesis MDCT filter bank

Since the DCT-IV matrix is symmetric and involutory, i.e., being self-inverse, the synthesis MDCT filter bank is realized by the inverse $N/2$ -point DCT-IV of $\{c_k^{(t)}\}$ as

$$y_n^{(t)} = \sum_{k=0}^{(N/2)-1} c_k^{(t)} \cos \left[\frac{\pi}{4(N/2)} (2n + 1)(2k + 1) \right],$$

$$n = 0, 1, \dots, \frac{N}{2} - 1. \quad (37)$$

The time domain aliased data sequence $\{\hat{x}_n^{(t)}\}$ is obtained from $\{y_n^{(t)}\}$ by applying the inverse permutation to (35) and using the symmetry of the windowing function

as follows:

$$\hat{x}_n^{(t)} = w_n y_{(N/4)+n}^{(t)},$$

$$\hat{x}_{(N/2)-1-n}^{(t)} = -w_{(N/2)-1-n} y_{(N/4)+n}^{(t)},$$

$$\hat{x}_{(N/2)+n}^{(t)} = -w_{(N/2)+n} y_{(N/4)-1-n}^{(t)} = -w_{(N/2)-1-n} y_{(N/4)-1-n}^{(t)},$$

$$\hat{x}_{N-1-n}^{(t)} = -w_{N-1-n} y_{(N/4)-1-n}^{(t)} = -w_n y_{(N/4)-1-n}^{(t)},$$

$$n = 0, 1, \dots, \frac{N}{4} - 1, \quad (38)$$

or in matrix-vector form without the windowing operation as

$$\hat{\mathbf{x}}^T = \begin{pmatrix} 0 & I_{N/4} \\ 0 & -J_{N/4} \\ -J_{N/4} & 0 \\ -I_{N/4} & 0 \end{pmatrix} \mathbf{y}^T. \quad (39)$$

Eqs. (37) and (38) define the fast synthesis MDCT filter bank. Especially note that the transform kernels in (34) and (37) are identical and symmetric with respect to the time and frequency indices n and k , while both n and k are members of the same set of integer values $0, 1, \dots, (N/2) - 1$. This implies that *the identical computational structure* can be used both for the computation of the analysis and the synthesis MDCT filter banks. Although this conclusion seems rather obvious here, in the past many papers suggested a different structure for the synthesis filter bank based on a reversely computed analysis structure.

Note 4. Specifically, the windowing operation with the sine function given by (5) in the analysis MDCT filter bank (see Eq. (35)), can be converted to regular cascades of Givens–Jacobi rotations. This also holds for the windowing, overlapping and add procedure after the backward transformation of the transform coefficients in the synthesis MDCT filter bank (see Eqs. (3) and (38)). It is exactly the same procedure as employed in the fast MLT [43] and the efficient implementation of the analysis/synthesis MDCT filter banks in MPEG-2 AAC [71].

3.3. Comments on related existing fast MDCT (MLT) algorithms

Without loss of generality it is assumed that the input data sequence $\{x_n^{(t)}\}$ is windowed by a windowing function satisfying Eq. (4). In order to convert the analysis MDCT filter bank to a DCT-IV of half size, previously described fast MDCT algorithms [20,21,24,26,28,30–32,35,36] applied to $\{x_n^{(t)}\}$ the following permutation:

$$y_n^{(t)} = \begin{cases} -x_{(3N/4)+n}^{(t)}, & n = 0, 1, \dots, \frac{N}{4} - 1, \\ x_{n-(N/4)}^{(t)}, & n = \frac{N}{4}, \frac{N}{4} + 1, \dots, N - 1. \end{cases}$$

It can be easily verified that by substituting $(3N/4) + n$ and $n - (N/4)$ for n into (1) the analysis MDCT filter bank

is reduced to

$$\begin{aligned} c_k^{(t)} &= \sum_{n=0}^{N-1} y_n^{(t)} \cos\left[\frac{\pi}{2N}(2n+1)(2k+1)\right] \\ &= \sum_{n=0}^{(N/2)-1} (y_n^{(t)} - y_{N-1-n}^{(t)}) \cos\left[\frac{\pi}{4(N/2)}(2n+1)(2k+1)\right], \\ & \quad k = 0, 1, \dots, \frac{N}{2} - 1. \end{aligned}$$

Finally, by combining the equations above the analysis MDCT filter bank can be expressed as

$$\begin{aligned} c_k^{(t)} &= \sum_{n=0}^{(N/2)-1} y_n^{(t)} \cos\left[\frac{\pi}{4(N/2)}(2n+1)(2k+1)\right], \\ & \quad k = 0, 1, \dots, \frac{N}{2} - 1, \end{aligned} \quad (40)$$

where

$$y_n^{(t)} = \begin{cases} -x_{(3N/4)+n}^{(t)} - x_{(3N/4)-1-n}^{(t)}, & n = 0, 1, \dots, \frac{N}{4} - 1, \\ x_{n-(N/4)}^{(t)} - x_{(3N/4)-1-n}^{(t)}, & n = \frac{N}{4}, \frac{N}{4} + 1, \dots, \frac{N}{2} - 1. \end{cases} \quad (41)$$

Although the permutations (35) and (41) seem to be different, they generate exactly the same time domain aliased data sequence $\{y_n^{(t)}\}$.

Similarly, since the DCT-IV matrix is self-inverse, it follows from (40) that the synthesis MDCT filter bank can be realized by Eq. (37) while the time domain aliased data sequence $\{\hat{x}_n^{(t)}\}$ is obtained from Eq. (38).

Note 5. The symmetry property of time domain aliased data sequence $\{\hat{x}_n^{(t)}\}$ given by (12) can be written in an alternative, equivalent form as

$$\begin{aligned} \hat{x}_{(N/4)-1-n}^{(t)} &= -\hat{x}_{(N/4)+n}^{(t)}, & \hat{x}_{(3N/4)-1-n}^{(t)} &= \hat{x}_{(3N/4)+n}^{(t)}, \\ & n = 0, 1, \dots, \frac{N}{4} - 1. \end{aligned}$$

Compared to (12), now the points of symmetry are explicitly taken at $N/4$ and $3N/4$.

4. Fast MDCT computational structure

At this point, the fast analysis/synthesis MDCT filter bank is reduced to the windowing/windowing&overlap&add procedure and an $N/2$ -point DCT-IV. Now it becomes rewarding to specify a suitable fast DCT-IV algorithm with a simple and regular computational structure. In the following we derive a fast regular DCT-IV computational structure leading to the so-called fast MDCT computational structure.

Consider Eqs. (34)/(37) which define the fast analysis/synthesis MDCT filter banks. Since we will consider transformation of a single data block, we omit the data-block number t in (34) and (37). At first, we derive a fast DCT-IV computational structure for the fast analysis MDCT filter bank given by (34). Because the DCT-IV transform kernel in (34) and (37) is symmetric with respect to the time and frequency indices n and k , by applying the same procedure described below but exchanging the role of n and k , we obtain the identical fast

DCT-IV computational structure for the fast synthesis MDCT filter bank given by (37). In the first step, if we substitute $(N/2) - 1 - n$ for n into (34), we get

$$\begin{aligned} c_k &= \sum_{n=0}^{(N/4)-1} y_n \cos\left[\frac{\pi}{2N}(2n+1)(2k+1)\right] + (-1)^k y_{(N/2)-1-n} \\ & \quad \times \sin\left[\frac{\pi}{2N}(2n+1)(2k+1)\right], \\ & \quad k = 0, 1, \dots, \frac{N}{2} - 1. \end{aligned} \quad (42)$$

Now, let us introduce a two-to-one mapping defined as

$$k' = \begin{cases} \frac{k}{2} & \text{if } k = 2k' \text{ is even, } k \in \{0, 2, 4, \dots\}, \\ \frac{k+1}{2} & \text{if } k = 2k' - 1 \text{ is odd, } k \in \{1, 3, 5, \dots\}, \end{cases} \quad (43)$$

where k' is a new frequency index. Then, we are able to replace the cosine transform kernel term $2k+1$ by $4k'+1/4k'-1$ depending on whether k is even/odd (see Table 1), and Eq. (42) can be equivalently written for the new frequency index k' in the form of two sums

$$\begin{aligned} c_{2k'} &= \sum_{n=0}^{(N/4)-1} y_n \cos\left[\frac{\pi}{2N}(2n+1)(4k'+1)\right] \\ & \quad + y_{(N/2)-1-n} \sin\left[\frac{\pi}{2N}(2n+1)(4k'+1)\right], \\ & \quad k' = 0, 1, \dots, \frac{N}{4} - 1, \\ c_{2k'-1} &= \sum_{n=0}^{(N/4)-1} y_n \cos\left[\frac{\pi}{2N}(2n+1)(4k'-1)\right] \\ & \quad - y_{(N/2)-1-n} \sin\left[\frac{\pi}{2N}(2n+1)(4k'-1)\right], \\ & \quad k' = 1, 2, \dots, \frac{N}{4}. \end{aligned} \quad (44)$$

In the second step, substituting the following trigonometric identities into (44):

$$\begin{aligned} & \cos\left[\frac{\pi}{2N}(2n+1)(4k' \pm 1)\right] \\ &= \cos\left[\frac{\pi(2n+1)}{2N}\right] \cos\left[\frac{\pi(2n+1)k'}{N/2}\right] \\ & \quad \mp \sin\left[\frac{\pi(2n+1)}{2N}\right] \sin\left[\frac{\pi(2n+1)k'}{N/2}\right], \\ & \sin\left[\frac{\pi}{2N}(2n+1)(4k' \pm 1)\right] \\ &= \cos\left[\frac{\pi(2n+1)}{2N}\right] \sin\left[\frac{\pi(2n+1)k'}{N/2}\right] \\ & \quad \pm \sin\left[\frac{\pi(2n+1)}{2N}\right] \cos\left[\frac{\pi(2n+1)k'}{N/2}\right], \end{aligned}$$

Table 1
Two-to-one mapping defined by (43).

k/n	$2k+1/2n+1$	k'/n'	$4k' \pm 1/4n' \pm 1$
0	1	0	$4k' + 1/4n' + 1$
1	3	1	$4k' - 1/4n' - 1$
2	5	1	$4k' + 1/4n' + 1$
3	7	2	$4k' - 1/4n' - 1$
4	9	2	$4k' + 1/4n' + 1$
5	11	3	$4k' - 1/4n' - 1$
6	13	3	$4k' + 1/4n' + 1$
⋮	⋮	⋮	⋮

and after some algebraic manipulations the complete formulas constituting the fast DCT-IV computational structure for the fast analysis MDCT filter bank are obtained as

$$\begin{aligned}
 c_{2k} &= \sum_{n=0}^{(N/4)-1} a_n \cos \left[\frac{\pi(2n+1)k'}{2(N/4)} \right] + b_n \sin \left[\frac{\pi(2n+1)k'}{2(N/4)} \right], \\
 c_{2k-1} &= \sum_{n=0}^{(N/4)-1} a_n \cos \left[\frac{\pi(2n+1)k'}{2(N/4)} \right] - b_n \sin \left[\frac{\pi(2n+1)k'}{2(N/4)} \right], \\
 k' &= 1, 2, \dots, \frac{N}{4} - 1,
 \end{aligned} \tag{45}$$

where

$$\begin{aligned}
 a_n &= y_n \cos \frac{\pi}{2N} (2n+1) + y_{(N/2)-1-n} \sin \frac{\pi}{2N} (2n+1), \\
 b_n &= -y_n \sin \frac{\pi}{2N} (2n+1) + y_{(N/2)-1-n} \cos \frac{\pi}{2N} (2n+1), \\
 n &= 0, 1, \dots, \frac{N}{4} - 1.
 \end{aligned} \tag{46}$$

For $k' = 0$ in the first sum and for $k' = N/4$ in the second sum of (45), we, respectively, get

$$c_0 = \sum_{n=0}^{(N/4)-1} a_n, \quad c_{(N/2)-1} = - \sum_{n=0}^{(N/4)-1} (-1)^n b_n. \tag{47}$$

It can be easily seen that the $N/2$ -point DCT-IV is decomposed into the block of $N/4$ Givens–Jacobi rotations given by (46), an $N/4$ -point DCT-II and a corresponding $N/4$ -point DST-II. If we denote respectively in Eq. (45) the $N/4$ -point DCT-II of $\{a_n\}$ and the $N/4$ -point DST-II of $\{b_n\}$,

then we have

$$\begin{aligned}
 c_k^{\text{II}} &= \sum_{n=0}^{(N/4)-1} a_n \cos \left[\frac{\pi(2n+1)k'}{2(N/4)} \right], \\
 s_k^{\text{II}} &= \sum_{n=0}^{(N/4)-1} b_n \sin \left[\frac{\pi(2n+1)k'}{2(N/4)} \right] \\
 &= \sum_{n=0}^{(N/4)-1} (-1)^n b_n \cos \left[\frac{\pi(2n+1)k'}{2(N/4)} \right] \\
 &= \sum_{n=0}^{(N/4)-1} b'_n \cos \left[\frac{\pi(2n+1)k'}{2(N/4)} \right], \\
 k' &= 1, 2, \dots, \frac{N}{4} - 1,
 \end{aligned}$$

and the $N/4$ -point DST-II of $\{b_n\}$ is converted to an $N/4$ -point DCT-II of $\{b'_n\}$. Thus, Eqs. (45) and (47) can be rewritten in a simplified equivalent form as

$$\begin{aligned}
 c_{2k} &= c_k^{\text{II}} + s_k^{\text{II}}, \quad c_0 = c_0^{\text{II}}, \\
 c_{2k-1} &= c_k^{\text{II}} - s_k^{\text{II}}, \quad c_{(N/2)-1} = -s_{N/4}^{\text{II}}, \\
 k' &= 1, 2, \dots, \frac{N}{4} - 1,
 \end{aligned} \tag{48}$$

which corresponds to the butterfly stage.

Finally, by applying the above procedure (see the first and second step) but exchanging the role of n and k , we obtain the complete formulas constituting the identical fast DCT-IV computational structure for the fast synthesis

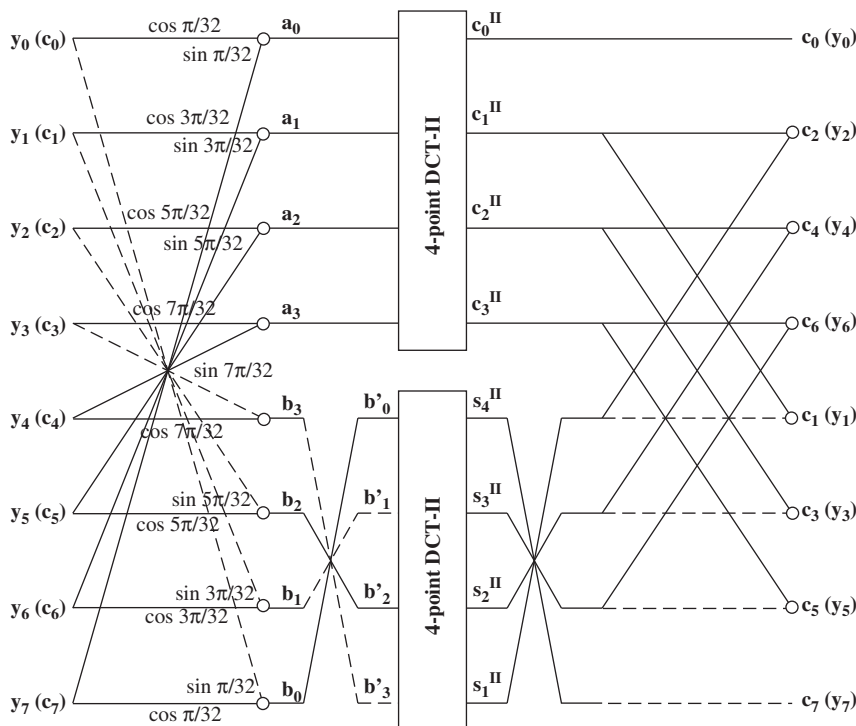


Fig. 3. The fast DCT-IV computational structure for $N/2 = 8$.

MDCT filter bank as

$$y_{2n'} = \sum_{k=0}^{(N/4)-1} a_k \cos \left[\frac{\pi(2k+1)n'}{2(N/4)} \right] + b_k \sin \left[\frac{\pi(2k+1)n'}{2(N/4)} \right],$$

$$y_{2n'-1} = \sum_{k=0}^{(N/4)-1} a_k \cos \left[\frac{\pi(2k+1)n'}{2(N/4)} \right] - b_k \sin \left[\frac{\pi(2k+1)n'}{2(N/4)} \right],$$

$$n' = 1, 2, \dots, \frac{N}{4} - 1, \quad (49)$$

or in the equivalent simplified form as

$$y_{2n'} = c_{n'}^{\text{II}} + s_{n'}^{\text{II}}, \quad y_0 = c_0^{\text{II}},$$

$$y_{2n'-1} = c_{n'}^{\text{II}} - s_{n'}^{\text{II}}, \quad y_{(N/2)-1} = -s_{N/4}^{\text{II}},$$

$$n' = 1, 2, \dots, \frac{N}{4} - 1, \quad (50)$$

where

$$a_k = c_k \cos \frac{\pi}{2N}(2k+1) + c_{(N/2)-1-k} \sin \frac{\pi}{2N}(2k+1),$$

$$b_k = -c_k \sin \frac{\pi}{2N}(2k+1) + c_{(N/2)-1-k} \cos \frac{\pi}{2N}(2k+1),$$

$$k = 0, 1, \dots, \frac{N}{4} - 1. \quad (51)$$

Similarly, for $n' = 0$ in the first sum and for $n' = N/4$ in the second sum of (49) we, respectively, get

$$y_0 = \sum_{k=0}^{(N/4)-1} a_k, \quad y_{(N/2)-1} = - \sum_{k=0}^{(N/4)-1} (-1)^k b_k. \quad (52)$$

The regular fast DCT-IV computational structure for $(N/2) = 8$ is shown in Fig. 3. Full lines represent transfer factors +1 while broken lines represent transfer factors -1. Symbol \circ represents addition.

From the signal flow graph shown in Fig. 3 a new sparse matrix factorization of the DCT-IV matrix denoted by C_N^{IV} can be directly extracted as

$$C_N^{\text{IV}} = P_N \begin{pmatrix} 1 & & & 0 \\ & I_{(N/2)-1} & I_{(N/2)-1} & \\ & I_{(N/2)-1} & -I_{(N/2)-1} & \\ 0 & & & -1 \end{pmatrix} \begin{pmatrix} C_{N/2}^{\text{II}} & 0 \\ 0 & J_{N/2} C_{N/2}^{\text{II}} \end{pmatrix} \times \begin{pmatrix} I_{N/2} & 0 \\ 0 & D_{N/2} J_{N/2} \end{pmatrix} G_N, \quad (53)$$

where P_N is a permutation matrix, the matrix product $J_{N/2} C_{N/2}^{\text{II}}$ denotes the $(N/2) \times (N/2)$ DCT-II matrix with reverse ordered rows. $D_{(N/2)} = \text{diag}\{1, -1, \dots, (-1)^{(N/2)+1}\}$ is the diagonal odd-sign changing matrix, and G_N is the rotation matrix (Givens-Jacobi rotations) defined by

$$G_N = \begin{pmatrix} \cos \frac{\pi}{4N} & & & 0 & & \sin \frac{\pi}{4N} \\ & \cos \frac{3\pi}{4N} & & & & \sin \frac{3\pi}{4N} \\ & & \cos \frac{(N-1)\pi}{4N} & \sin \frac{(N-1)\pi}{4N} & & \\ 0 & & -\sin \frac{(N-1)\pi}{4N} & \cos \frac{(N-1)\pi}{4N} & & 0 \\ & & & & & \\ & -\sin \frac{3\pi}{4N} & & & \cos \frac{3\pi}{4N} & \\ -\sin \frac{\pi}{4N} & & & 0 & & \cos \frac{\pi}{4N} \end{pmatrix}. \quad (54)$$

Since the DCT-IV matrix is self-inverse, by transposing (53) we obtain its alternative sparse matrix factorization. In [29], a slightly different sparse matrix factorization of the DCT-IV matrix has been proposed which was extracted from the fast MDCT algorithm represented by the sparse matrix factorization of the MDCT matrix [25].

Note 6. Based on a relation between the DCT-IV and DST-IV matrices given in [75], the fast DST-IV computational structure can be easily obtained.

The proposed fast DCT-IV computational structure with the properly appended permutations (35) and (38) results in the fast MDCT computational structure being identical for the efficient computing both the analysis and synthesis MDCT filter banks. For a given value of N , the common computational complexity of the fast MDCT computational structure consists of the complexity of the windowing operation and the $N/2$ -point fast DCT-IV computational structure. Additionally, the fast analysis MDCT filter bank requires $N/2$ additions for performing the permutation (35) while the fast synthesis MDCT bank requires $N/2$ additions for performing the overlap and add procedure (3). Because the fast MDCT computational structure is valid for any N divisible by 4, its computational complexity depends on a fast DCT-II algorithm to be adopted. Since $N/4$ may be even or odd, an efficient even-length or odd-length DCT-II algorithm is required. Further, taking into account Eqs. (36), (39), (53) and (54), we can derive sparse matrix factorizations of the MDCT matrices. Let $C_{(N/2) \times N}$ and $C_{N \times (N/2)}$ be MDCT matrices representing the forward and backward MDCT, respectively. Then, they can be factored as

$$C_{(N/2) \times N} = C_{N/2}^{\text{IV}} \begin{pmatrix} 0 & 0 & -J_{N/4} & -I_{N/4} \\ I_{N/4} & -J_{N/4} & 0 & 0 \end{pmatrix},$$

$$C_{N \times (N/2)} = \begin{pmatrix} 0 & I_{N/4} \\ 0 & -J_{N/4} \\ -J_{N/4} & 0 \\ -I_{N/4} & 0 \end{pmatrix} C_{N/2}^{\text{IV}}. \quad (55)$$

Recently, similar fast MDCT computational structures have been derived [30–32,34] with the aim to improve or optimize an efficient implementation of the MDCT in MP3 audio coding [22,25]. The last butterfly stage in Fig. 3 corresponds to Eqs. (48) and (50) or equivalently, it is represented by the product of the leftmost two matrices on the right-hand side of (53). If we denote the output coefficients $\{c_0^{\text{II}}, c_1^{\text{II}}, \dots, c_{(N/4)-1}^{\text{II}}, s_1^{\text{II}}, s_2^{\text{II}}, \dots, s_{N/4}^{\text{II}}\}$ of the two $N/4$ -point unnormalized DCT-II transforms by $\{z_k\}$, then the last butterfly stage in Fig. 3 can be reorganized and the final MDCT coefficients $\{c_k\}$ given by (48) are obtained in natural order in terms of the two-to-one mapping (43) as

$$c_k = \begin{cases} z_{k/2} + z_{(N/4)-1+(k/2)}, & k = 2, 4, \dots, \frac{N}{4} - 2 \text{ is even,} \\ c_0 = z_0, & \\ z_{(k+1)/2} - z_{(N/4)-1+((k+1)/2)}, & k = 1, 3, \dots, \frac{N}{4} - 1 \text{ is odd,} \\ c_{N/2-1} = -z_{N/2-1}. & \end{cases}$$

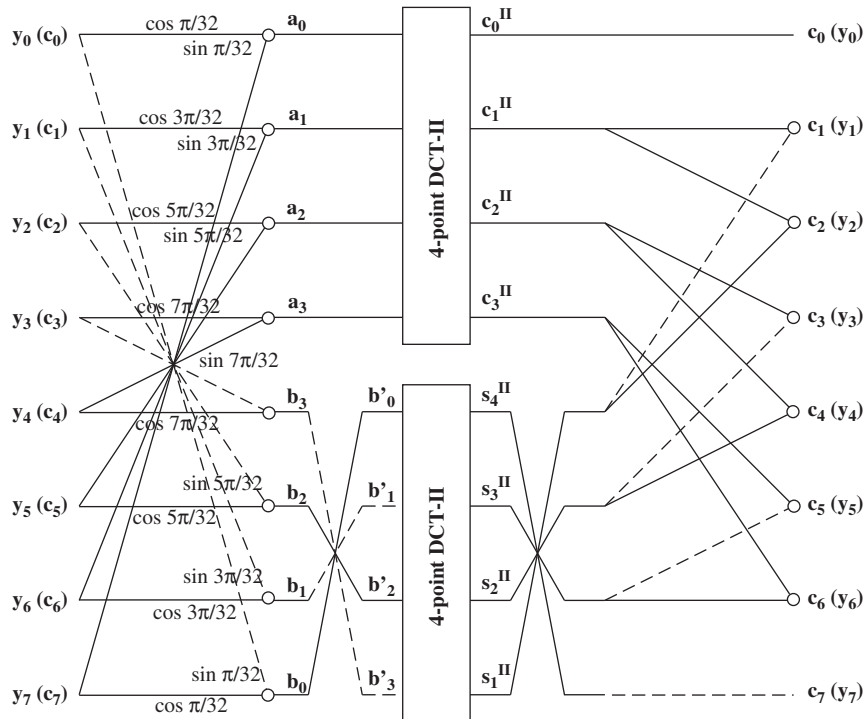


Fig. 4. The fast DCT-IV computational structure for $N/2 = 8$ with the reorganized last butterfly stage.

The corresponding fast DCT-IV computational structure for $N/2 = 8$ with the reorganized last butterfly stage is shown in Fig. 4, and it strongly resembles the ones proposed in [31,32,34] although these were derived using different procedures. The regular output of Fig. 4 is preferable to that of Fig. 3. When the $N/2$ -point fast DCT-IV computational structure in Fig. 4 is built into the encoder in a TDAC analysis filter bank, the input values $y_0, y_1, \dots, y_{(N/2)-1}$ are derived from the input samples according to (35) resulting in the MDCT coefficients $c_0, c_1, \dots, c_{(N/2)-1}$ in natural order. When it is built into the decoder in a TDAC synthesis filter bank, inputs $c_0, c_1, \dots, c_{(N/2)-1}$ are the MDCT coefficients in natural order, while the output values $y_0, y_1, \dots, y_{(N/2)-1}$ result in the reconstructed data sequence according to (38).

5. Discussion and comparison with existing fast algorithms

The fast MDCT computational structure relies on the proposed fast DCT-IV computational structure. An indirect fast DCT-IV algorithm which maps the DCT-IV into a DFT of half size and uses a recursive split-radix FFT can be found in [43]. Direct recursive fast DCT-IV algorithms based on recursive orthogonal sparse matrix factorizations of the DCT-IV matrix have been presented in [72–74]. Whereas in [72,73] the DCT-IV matrix is recursively factored into two DCT-IV matrices of half sizes with pre-permutation/butterfly and post-rotation stages, the DCT-IV matrix in [74] is recursively factored into two DCT-II

matrices of half sizes with pre-rotation and post-butterfly/permutation stages. Usually, for a recursive algorithm if $N = 2^n$, n successive regular stages are needed in its corresponding computational structure. Consequently, for higher values of N (typically in audio coding schemes where $N = 256, 512, 2048$), the structural complexity of the recursive fast DCT-IV algorithm increases. The main advantage of the proposed fast DCT-IV computational structure is its structural simplicity. It has a constant geometry for any $N = 2^n$, i.e., consists of three regular stages only. For the N -point DCT-IV computation with $N = 2^n$, all above mentioned fast algorithms (including the proposed fast DCT-IV computational structure) require exactly $(N/2)(n+2)$ multiplications and $(3N/2)n$ additions, i.e., totally $2Nn + N$ arithmetic operations, being so far (see Note below) the lowest achievable arithmetic complexity for the DCT-IV [43]. For the fast MDCT computation, the DCT-IV can be alternatively converted to the DCT-II of the same size at the cost of additional pre-multiplications and recursive post-additions [20,24,26,28,35,36,39,40,44].

Note 7. Recently, new recursive algorithms for the 2^n -length DCT-IV/DST-IV and MDCT computation have been presented [41] requiring fewer total real multiplications and additions than algorithms published up to now. They are based on a new improved FFT algorithm being actually the modified split-radix FFT with fewer arithmetic operations. For the N -point DCT-IV, $N = 2^n$, $n > 2$, the total number of arithmetic operations is asymptotically reduced from $2Nn + N$ to $\frac{17}{9}Nn + \frac{31}{27}N$. Since the DCT-IV and

MDCT are closely related, this improved DCT-IV algorithm immediately implies an improved MDCT algorithm.

The fast MDCT computational structure based on the proposed fast DCT-IV computational structure possesses the following important characteristics:

- It is efficient both in terms of the computational complexity and structural simplicity with identical regular computational blocks both for the analysis and synthesis filter banks, i.e., the same computational structure is used for the encoder and decoder in audio coding schemes. The computational complexity of the $N/2$ -point DCT-IV is given by the block of $N/4$ Givens–Jacobi rotations (requiring $3(N/4)$ multiplications and $3(N/4)$ additions), the complexity of two $(N/4)$ -point unnormalized DCT-II transforms (for a specific 2^n -length, $n > 2$, they require $(N/4)(n - 2)$ multiplications plus $(N/4)(3n - 8) + 2$ additions) and the last butterfly stage requiring $(N/2) - 2$ additions. Then, for a 2^n -length, provided that we use the most efficient DCT-II algorithm, the total computational complexity for the $N/2$ -point DCT-IV computation is $(N/4)(n + 1)$ multiplications and $(N/4)(3n - 3)$ additions.
- Since the identical fast MDCT computational structure is valid for any N divisible by 4, it provides an efficient implementation of the forward and backward MDCT block transforms in MP3 audio coding. The optimized 3-point DCT-II module (requiring one multiplication, four additions and one shift) and 9-point DCT-II module (requiring eight multiplications, 34 additions and two shifts) can be found in [22,34]. Then, the forward MDCT computation including the input data permutation given by (35) for $N = 12$ requires 11 multiplications, 27 additions and 2 shifts, whereas for $N = 36$ it requires 43 multiplications, 129 additions and four shifts. The backward MDCT computation without the overlap and add procedure given by (3) requires exactly $N/2$ less additions than that of the forward MDCT. A comparison in terms of arithmetic complexity of the proposed efficient MDCT implementation in MP3 codecs with representative important MDCT implementations [23,24,36] is shown in Table 2. Table 2 clearly shows that the structure proposed here compares favorably against previous ones, with the added advantage of using the same computational structure in the forward as well as in the backward

part. Moreover, the algorithms of [24,36] are based on recursion which makes them less suitable for fast hardware implementations.

The complete analysis/synthesis MDCT filter bank given by (1)/(2) using the direct approach requires totally 84 multiplications plus 66 additions for $N = 12$, and 684 multiplications plus 630 additions for $N = 36$. An efficient implementation of the complete analysis/synthesis MDCT filter bank consists of the $N/2$ -point fast DCT-IV computational structure and windowing/windowing&overlap&add procedures defined by (35)/(38) & (3) in the overlapped part of two adjacent data blocks. The fast DCT-IV computational structure itself, for $N = 12$ requires 11 multiplications, 21 additions and two shifts, whereas for $N = 36$ it requires 43 multiplications, 111 additions and four shifts. Since the windowing/windowing&overlap&add procedures (for the sine windowing function given by (5)) can be realized by $N/4$ Givens–Jacobi rotations, the complete analysis and synthesis MDCT filter banks require totally 20 multiplications, 30 additions and two shifts when $N = 12$, and 70 multiplications, 138 additions and four shifts when $N = 36$.

- It provides an efficient implementation of two variants of cosine/sine-modulated filter banks (called the first and the second short transforms) defined by the Dolby Labs AC-3 digital audio compression algorithm [7]. Let $\{x_n\}, n = 0, 1, \dots, N - 1$ be a windowed input data sequence. Since the short transforms are actually the DCT-IV and DST-IV of half sizes [25], the first short transform can be computed by taking $y_n^c = x_n - x_{N-1-n}$, $n = 0, 1, \dots, (N/2) - 1$, and the second short transform by taking $y_n^s = -x_{(N/2)+n} - x_{(N/2)-1-n}$, $n = 0, 1, \dots, (N/2) - 1$.
- The simple relation between the MDCT and the MDST given by (16) and (17), provides an efficient implementation of the MCLT. Compared to existing fast MCLT algorithms [47–51] its computational complexity is identical to that of [47].
- It defines new sparse matrix factorizations of the DCT-IV and MDCT/MDST matrices.

It is interesting to compare the new fast MDCT algorithm previously proposed in [25] and the fast MDCT computational structure derived here. Complete formulas and some corresponding computational blocks of the new fast MDCT algorithm [25] (see Eqs. (18) and (20)) and those of the fast MDCT computational structure here are quite similar though they have been derived by different procedures. This fact indicates that the computational structures are closely related. Indeed, the fast MDCT computational structure [25] can also be adopted for the efficient implementation of two short transforms defined in the AC-3 codec [7]. Consequently, the first and second butterfly stages in the fast MDCT computational structure [25] are reduced to permutations only (thus, saving $3N/2$ additions), and this observation has led to a new DCT-IV/DST-IV computational structure [29]. On the other hand, the fast MDCT computational structure here has been derived directly from the DCT-IV of half size. Comparing

Table 2
Comparison of the number of multiplications/additions.

	Length $N = 12$		Length $N = 36$	
	Forward	Backward	Forward	Backward
Our structure	1127	$\frac{11}{21}$	$\frac{43}{129}$	$\frac{43}{111}$
[36]	$\frac{11}{27}$	$\frac{11}{23}$	$\frac{43}{129}$	$\frac{43}{115}$
[24]	$\frac{11}{29}$	$\frac{11}{23}$	$\frac{43}{133}$	$\frac{43}{115}$
[23]—refined version of [22]	$\frac{11}{39}$	$\frac{11}{33}$	$\frac{43}{165}$	$\frac{43}{147}$

both computational structures in detail, the following essential differences are immediately evident: for the backward MDCT computation, the fast MDCT computational structure [25] should be used in reverse direction performing inverse operations, so increasing its structural complexity, while the fast MDCT computational structure derived here is identical both for the forward and backward MDCT computation. Moreover, the fast MDCT computational structure [25] includes additional scale factors $\sqrt{2}/2$ and final sign changes apparently caused by the existence of two subsequent butterfly stages at the beginning. Both fast MDCT computational structures contain Givens–Jacobi rotations which are of opposite type. Therefore, the fast MDCT computational structure derived here is generally more efficient in terms of computational complexity and structural simplicity compared to the one proposed in [25].

6. Conclusions

A new fast identical computational structure both for the forward and backward MDCT/MDST computation based on the proposed fast DCT-IV computational structure has been described. It is the result of a systematic construction of a fast algorithm for an efficient implementation of the complete TDAC analysis/synthesis MDCT/MDST filter banks. Thus, the same computational structure is to be used both in the encoder and in the decoder, which obviously will result in strongly reduced design times and the possibility of resource reduction. The corresponding generalized signal flow graph is regular and defines new sparse matrix factorizations of DCT-IV and MDCT/MDST matrices. In particular, the consequences of MDCT/MDST matrix representations from the viewpoint of terminology used in the literature have been emphasized. The fast MDCT computational structure is compared with existing fast algorithms and its important characteristics are discussed in detail. The identical fast MDCT computational structure provides an efficient implementation of the MDCT in MP3 audio coding and AC-3 codecs. All steps to derive our computational structure are described in detail, and to put them into perspective a comprehensive list of references is provided covering new research results achieved in the time period 1999–2008 in theoretical and practical developments of TDAC analysis/synthesis MDCT/MDST filter banks. For clarity, the list of references is classified into categories.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments and suggestions, which significantly improved the clarity of presentation in this paper. This work was partially supported by Slovak Scientific Grant Agency VEGA, Project no. 2/7097/27.

References

- [1] J.P. Princen, A.W. Johnson, A.B. Bradley, Subband/transform coding using filter bank designs based on time domain aliasing cancellation, in: Proceedings of the IEEE ICASSP'87 Dallas TX, April 1987, pp. 2161–2164.
- [2] A.W. Johnson, A.B. Bradley, Adaptive transform coding incorporating time domain aliasing cancellation, *Speech Communications* 6 (December 1987) 299–308.
- [3] K.R. Rao, J.J. Hwang, *Techniques and Standards for Digital Image/Video/Audio Coding*, Prentice-Hall, Upper Saddle River, NJ, 1996.
- [4] T. Painter, A. Spanias, Perceptual coding of digital audio, *Proceedings of the IEEE* 88 (4) (April 2000) 451–513.
- [5] Information technology—coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, Part 3: Audio, ISO/IEC JTC1/SC29/WG11 MPEG, International Standard 11172–3 (MPEG-1), 1992.
- [6] Information technology—generic coding of moving pictures and associated audio, Part 3: Audio, ISO/IEC JTC1/SC29/WG11 MPEG, International Standard 13818–3 (MPEG-2), 1994.
- [7] Digital audio compression (AC-3) ATSC Standard, Document of Advanced Television Systems Committee (ATSC), Audio specialist group T3/S7, December 1995.
- [8] Information technology—generic coding of moving pictures and associated audio information, Part 7: Advanced Audio Coding, ISO/IEC JTC1/SC29/WG11 MPEG, International Standard 13818–7 (MPEG-2 AAC), 1997.
- [9] Information technology—coding of audio–visual objects, Part 3: Audio, ISO/IEC JTC1/SC29/WG11 MPEG, International Standard 14496–3 (MPEG-4 audio), 1999.
- [10] J. Herre, M. Dietz, MPEG-4 high-efficiency AAC coding, *IEEE Signal Processing Magazine* 25 (3) (May 2008) 137–142.
- [11] [a] "0.8* 1.2Vorbis I specification, Document on web site: (<http://www.xiph.org/>), 2004, 68pp. *Books on digital audio coding*.
- [12] M. Bosi, R.E. Goldberg, *Introduction to Digital Audio Coding and Standards*, Springer Science+Business Media, Inc., New York, 2003.
- [13] A. Spanias, T. Painter, V. Atti, *Audio Signal Processing and Coding*, Wiley, NJ, 2007. *Mathematical and special (peculiar) properties of the MDCT/MDST*.
- [14] Y. Wang, L. Yaroslavsky, M. Vilermo, On the relationship between MDCT, in: Proceedings of the Fifth International Conference on Signal Processing (ICSP'2000), Beijing, China, August 2000, pp. 44–47.
- [15] Y. Wang, L. Yaroslavsky, M. Vilermo, M. Väänänen, Some peculiar properties of the MDCT, in: Proceedings of the Fifth International Conference on Signal Processing (ICSP'2000), August 2000, pp. 61–64.
- [16] Y. Wang, M. Vilermo, Modified discrete cosine transform—its implications for audio coding and error concealment, *Journal of Audio Engineering Society* 51 (1/2) (January/February 2003) 52–61.
- [17] K. Wright, Notes on Ogg Vorbis and the MDCT, Draft document on web site: (www.free-comp-shop.com/vorbis.html), May 2003, 7pp.
- [18] V. Britanak, A note on the MDCT/MDST and pseudoinverse matrix, *Computing and Informatics* 23 (3) (March 2004) 205–214.
- [19] Z. Wang, B.R. Hunt, The discrete W transform, *Applied Mathematics and Computation* 16 (1985) 19–48. *Fast MDCT/MDST algorithms and efficient MDCT implementations in MP3*.
- [20] D.-Y. Chan, J.-F. Yang, S.-Y. Chen, Regular implementation algorithms of time domain aliasing cancellation, *IEEE Proceedings—Vision, Image and Signal Processing* 143 (6) (December 1996) 387–392.
- [21] C.-M. Liu, W.-C. Lee, A unified fast algorithm for cosine-modulated filter banks in current audio coding standards, *Journal of the Audio Engineering Society* 47 (12) (December 1999) 1061–1075.
- [22] V. Britanak, K.R. Rao, An efficient implementation of the forward and inverse MDCT in MPEG audio coding, *IEEE Signal Processing Letters* 8 (2) (February 2001) 48–51 (Erratum, *IEEE Signal Processing Letters* 8(10) (October 2001) p. 279).
- [23] V. Britanak, The refined efficient implementation of the MDCT in MP3 and comparison with other methods, Technical Report II SAS-2002-02, Department of Numerical Methods and Algorithms, Institute of Informatics, Slovak Academy of Sciences, Bratislava, September 2002.
- [24] S.W. Lee, Improved algorithm for efficient computation of the forward and backward MDCT in MPEG audio coder, *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing* 48 (10) (October 2001) 990–994.
- [25] V. Britanak, K.R. Rao, A new fast algorithm for the unified forward and inverse MDCT/MDST computation, *Signal Processing* 82 (3) (March 2002) 433–459.

- [26] M.-H. Cheng, Y.-H. Hsu, Fast IMDCT and MDCT algorithms—a matrix approach, *IEEE Transactions on Signal Processing* 51 (1) (January 2003) 221–229.
- [27] N. Rama Murthy, M.N.S. Swamy, A parallel/pipelined algorithm for the computation of MDCT and IMDCT, in: *Proceedings of the International Symposium on Circuits and Systems (ISCAS'2003)*, vol. 4, Bangkok, Thailand, May 2003, pp. 540–543.
- [28] P.-S. Wu, Y.-T. Hwan, Efficient IMDCT core designs for audio signal processing, in: *Proceedings of the IEEE Workshop on Signal Processing Systems: Design and Implementation (SiPS'2003)*, Seoul, Korea, August 2003, pp. 275–280.
- [29] V. Britanak, The fast DCT-IV/DST-IV computation via the MDCT, *Signal Processing* 83 (8) (August 2003) 1803–1813.
- [30] V. Nikolajević, G. Fettweis, Improved implementation of MDCT in MP3 audio coding, in: *Proceedings of the IEEE 10th Asian-Pacific Conference on Communications and 5th International Symposium on Multi-Dimensional Mobile Communications (APCC/MDMC'2004)*, vol. 1, Tsinghua University, Beijing, China, August–September 2004, pp. 309–312.
- [31] H.-S. Kim, Y.-K. Cho, W.-P. Lee, A new optimized algorithm for computation of MDCT and its inverse transform, in: *Proceedings of the IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS'2004)*, Seoul, Korea, November 2004, pp. 528–530.
- [32] Y.-K. Cho, T.-H. Song, H.-S. Kim, An optimized algorithm for computing the modified discrete cosine transform and its inverse transform, in: *Proceedings of the IEEE Region 10 Conference TENCON'2004*, Chiang Mai, Thailand, November 2004, pp. 626–628.
- [33] V. Britanak, An efficient computing of oddly stacked MDCT/MDST via evenly stacked MDCT/MDST and vice versa, *Signal Processing* 85 (7) (July 2005) 1353–1374.
- [34] H.J. Lincklaen Arriens, Implementation of an 18-point IMDCT on FPGA, Internal Report no. CAS-LA05.1, Department of Electrical Engineering, Mathematical and Computer Science, Delft University of Technology, Delft, The Netherlands, July 2005, 22pp.
- [35] Y.-T. Hwang, S.-C. Lai, A novel MDCT/IMDCT computing kernel design, in: *Proceedings of the IEEE Workshop on Signal Processing Systems: Design and Implementation*, Athens, Greece, November 2005, pp. 526–531.
- [36] T.K. Truong, P.D. Chen, T.C. Cheng, Fast algorithm for computing the forward and inverse MDCT in MPEG audio coding, *Signal Processing* 86 (5) (May 2006) 1055–1060.
- [37] K.C. Anup, A.K. Bangla, A new efficient implementation of TDAC synthesis filterbank based on radix-2 FFT, in: *Proceedings of the 14th EUSIPCO Signal Processing Conference*, Florence, Italy, September 2006.
- [38] H. Shu, X. Bao, Ch. Toumoulin, L. Luo, Radix-3 algorithm for the fast computation of forward and inverse MDCT, *IEEE Signal Processing Letters* 14 (2) (February 2007) 93–96.
- [39] T. Li, R. Zhang, R. Yang, H. Huang, F. Lin, A unified computing kernel for MDCT/IMDCT in modern audio coding standards, in: *Proceedings of the IEEE International Symposium on Communication and Information Technologies (ISCIT'2007)*, Sydney, Australia, October 2007, pp. 546–550.
- [40] R.K. Chivukula, Y.A. Reznik, Efficient implementation of a class of MDCT/IMDCT filterbanks for speech and audio coding applications, in: *Proceedings of the IEEE ICASSP'2008*, Las Vegas, NV, March–April 2008, pp. 213–216.
- [41] X. Shao, S.G. Johnson, Type-IV DCT, DST, and MDCT algorithms with reduced numbers of arithmetic operations, *Signal Processing* 88 (6) (June 2008) 1313–1326.
- [42] J.S. Wu, H.Z. Shu, L. Senhadji, L.M. Luo, A fast algorithm for the computation of 2-D forward and inverse MDCT, *Signal Processing* 88 (6) (June 2008) 1313–1326.
Fast algorithms for the MLT computation.
- [43] H.S. Malvar, *Signal Processing with Lapped Transforms*, Artech House, Norwood, MA, 1992.
- [44] H.S. Malvar, Fast algorithms for orthogonal and biorthogonal modulated lapped transforms, in: *Proceedings of the IEEE Symposium on Advances in Digital Filtering and Signal Processing*, Victoria, Canada, June 1998, pp. 159–163.
- [45] C.-Y. Jing, H.-M. Tai, Fast algorithm for computing modulated lapped transform, *Electronics Letters* 37 (12) (June 2001) 796–797.
- [46] C.-Y. Jing, H.-M. Tai, Design and implementation of a fast algorithm for modulated lapped transform, *IEEE Proceedings: Vision Image and Signal Processing* 149 (1) (February 2002) 27–32.
Fast algorithms for the MCLT computation.
- [47] H.S. Malvar, A modulated complex lapped transform and its application to audio processing, in: *Proceedings of the IEEE ICASSP'99*, Phoenix, AR, May 1999, pp. 1421–1424.
- [48] H.-M. Tai, C.-Y. Jing, Design and efficient implementation of a modulated complex lapped transform processor using pipeline technique, *IEICE Transactions Fundamentals E84-A* (5) (May 2001) 1280–1287.
- [49] H.S. Malvar, Fast algorithm for the modulated complex lapped transform, *IEEE Signal Processing Letters* 10 (1) (January 2003) 8–10.
- [50] Q. Dai, X.-J. Chen, New algorithm for modulated complex lapped transform with symmetrical window function, *IEEE Signal Processing Letters* 12 (12) (December 2004) 925–928.
- [51] X.-J. Chen, Q. Dai, A novel DCT-based algorithm for computing the modulated complex lapped transform, *IEEE Transactions on Signal Processing* 54 (11) (November 2006) 4480–4484.
MDCT/MDST algorithms based on recursive filter structures.
- [52] C.-H. Chen, C.-B. Wu, B.-D. Liu, J.-F. Yang, Recursive architectures for the forward and inverse modified discrete cosine transform, in: *Proceedings of the IEEE Workshop on Signal Processing Systems: Design and Implementation (SiPS'2000)*, Lafayette, LA, October 2000, pp. 50–59.
- [53] V. Nikolajević, G. Fettweis, New recursive algorithms for the forward and inverse MDCT, in: *Proceedings of the IEEE Workshop on Signal Processing Systems: Design and Implementation (SiPS'2001)*, Antwerp, Belgium, September 2001, pp. 51–57.
- [54] C.-H. Chen, B.-D. Liu, J.-F. Yang, Recursive architectures for realizing modified discrete cosine transform and its inverse, *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing* 50 (1) (January 2003) 38–45.
- [55] V. Nikolajević, G. Fettweis, Computation of forward and inverse MDCT using Clenshaw's recurrence formula, *IEEE Transactions on Signal Processing* 51 (5) (May 2003) 1439–1444.
- [56] V. Nikolajević, G. Fettweis, New recursive algorithms for the unified forward and inverse MDCT/MDST, *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology* 34 (3) (July 2003) 203–208.
- [57] Z.-Y. Cheng, C.-H. Chen, B.-D. Liu, J.-F. Yang, Unified selectable fixed-coefficient recursive structures for computing DCT, in: *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'2004)*, Vol. 3, Vancouver, Canada, May 2004, pp. 557–560.
- [58] T.W. Fox, A. Carriera, Goertzel implementations of the forward and inverse modified discrete cosine transform, in: *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'2004)*, vol. 4, Niagara Falls, Canada, May 2004, pp. 2371–2374.
Hardware/software MDCT implementations in MP3.
- [59] X. Yang, S. Shi, A.K. Wong, Tradeoffs in modified discrete cosine transform implementations, in: *Proceedings of the Fourth International Conference on ASIC*, Shanghai, China, October 2001, pp. 370–373.
- [60] P. Sundareson, A re-evaluation of fundamental transform structures for efficient implementation on semi-parallel DSP architectures, in: *112th AES Convention*, Munich, Germany, May 2002, Preprint #5614.
- [61] K.H. Bang, J.S. Kim, Y.C. Park, D.H. Youn, Design optimization of a dual MP3/AAC decoder, in: *Proceedings of the IEEE ICASSP'2002*, vol. 4, Orlando, FL, May 2002, pp. 3776–3779.
- [62] X. Wang, W. Dou, Z. Hou, An improved audio encoding architecture based on 16-bit fixed-point DSP, in: *Proceedings of the IEEE International Conference on Communications, Circuits and Systems, and West Sino Expositions (ICCCAS and WeSino Expo'2002)*, Chengdu City, China, vol. 2, June 2002, pp. 918–921.
- [63] W. Lee, K. You, W. Sung, Software optimization of MPEG audio layer-III for a 32-bit RISC processor, in: *Proceedings of the IEEE Asia-Pacific Conference on Circuits and Systems*, vol. 1, Bali, Indonesia, October 2002, pp. 435–438.
- [64] W. Sung, H. Chang, W. Lee, S. Ryu, Speaking partner—An ARM7-based multimedia handheld device, in: *Proceedings of the IEEE Workshop on Signal Processing Systems: Design and Implementation (SiPS'2002)*, San Diego, CA, October 2002, pp. 218–221.
- [65] J.A.A. Cardona, S. Jayakumar, Real-time MP3 encoder implemented in DSP hardware, in: *Proceedings of the International Conference on Signal Processing and Global DSP Expo*, Dallas, TX, March–April 2003.
- [66] I. Fältman, M. Hast, A. Lundgren, S. Malki, E. Montnemery, A. Rangevall, J. Sandvall and M. Stamenkovic, A hardware implementation of an MP3 decoder, Digital IC-Project, LTH, Lund Institute of Technology, Sweden, May 2003.
- [67] V. Gurkhe, Optimization of an MP3 decoder on the ARM processor, in: *Proceedings of the IEEE Region 10 Asia-Pacific Conference TENCON'2003*, Bangalore, India, October 2003, pp. 1475–1478 (Poster 280).

- [68] T.-H. Tsai, Y.-C. Yang, Low power and cost effective VLSI design for an MP3 audio decoder using an optimized synthesis-subband approach, *Computers and Digital Techniques* 151 (3) (May 2004) 245–251.
- [69] Y. Yao, Q. Yao, P. Liu, Z. Xiao, Embedded software optimization for MP3 decoder implemented on RISC core, *IEEE Transactions on Consumer Electronics* 50 (4) (November 2004) 1244–1249.
- [70] W. Zhang, P. Liu, Z.-B. Zhai, A hardware/software co-optimization approach for embedded software of MP3 decoder, *Journal of Zhejiang University SCIENCE A* 8 (1) (January 2007) 49–56. *DCT/DST (block) matrix factorizations, matrix theory and linear algebra*.
- [71] R. Geiger, T. Sporer, J. Koller, K. Brandenburg, Audio coding based on integer transforms, in: 111th AES Convention, New York, September 2001, Preprint #5471.
- [72] R. Geiger, Y. Yokotani, G. Schuller, Improved integer transforms for lossless audio coding, in: *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2003, pp. 2119–2123.
- [73] R. Geiger, Y. Yokotani, G. Schuller, J. Herre, Improved integer transforms using multidimensional lifting, in: *Proceedings of the IEEE ICASSP'2004*, Montreal, Canada, May 2004, pp. 1005–1008.
- [74] G. Plonka, M. Tasche, Fast and numerically stable algorithms for discrete cosine transforms, *Linear Algebra and its Applications* 394 (1) (January 2005) 309–345.
- [75] V. Britanak, P. Yip, K.R. Rao, *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations*, Academic Press Inc. Elsevier Science, Amsterdam, 2007.
- [76] F.R. Gantmacher, *The Theory of Matrices*, second ed., Nauka, Moscow, 1966 (in Russian), (English translation: vols. 1 and 2, Chelsea, New York, 1959).
- [77] G.H. Golub, C.F. Van Loan, *Matrix Computations*, third ed., The Johns Hopkins University Press, Baltimore, 1996.