Fast and accurate direct MDCT to DFT conversion with arbitrary window functions

Shuhua Zhang, Laurent Girin

Fast and accurate direct MDCT to DFT conversion with arbitrary window functions

Shuhua Zhang* and Laurent Girin

***Abstract*—In this paper, we propose a method for direct conversion of MDCT coefficients to DFT coefficients, without passing through time signal reconstruction. In contrast to previous work, this method is valid for any pair of MDCT and DFT window functions. It is based on the decomposition of the MDCT-to-DFT conversion matrices into a Toeplitz part plus a Hankel part. The latter is split, then mirrored and combined with the former to construct a global Toeplitz matrix. This leads to a fast FIR filtering implementation of the conversion process. The filter taps are DFT coefficients of window functions products, and concentrate most of their energy in a few low-frequency taps. The conversion can thus be efficiently approximated by keeping only a few most significant taps, as confirmed by numerical experiments: For example, for frame size of 2048, Hanning-windowed DFT is obtained from KBD-windowed MDCT with SNR over 60 dB when keeping only 20 taps.**

***Index Terms*—MDCT, DFT, window function, Toeplitz matrix, FIR filtering.**

I. Introduction

The Modified Discrete Cosine Transform (MDCT) [1] is a time-frequency (TF) transform that is widely used in audio processing, especially in perceptual audio coding algorithms. This is the case for, e.g., MPEG-2/4 Advanced Audio Coding (AAC) [2] and Ogg Vorbis. The MDCT belongs to the family of Lapped Transforms (LT) which are critically sampled, even with overlap between adjacent frames of input signal, and assume perfect reconstruction (for both time→TF→time and TF→time→TF) [3], [4]. Those properties are much appreciated in audio coding, since even with quantization of MDCT coefficients, the MDCT ensures smooth transitions between frames and good signal reconstruction.

However, the MDCT is poorly appropriate for spectral analysis and signal manipulation in the TF domain, for several reasons [4], [5], [6]: Its basis vectors are not shift-invariant, it does not conserve the energy, and MDCT coefficients, which are real-valued, cannot be easily interpreted in terms of magnitude and phase. All this contrasts with the widely used Discrete Fourier Transform (DFT) or Short-Term Fourier Transform (STFT)[1]. In the same line, linear time-invariant filtering is generally not equivalent to product in the MDCT domain, except for very specific filter shapes [7]. For all those reasons, the DFT/STFT are used in most audio/speech (TF) processing systems.

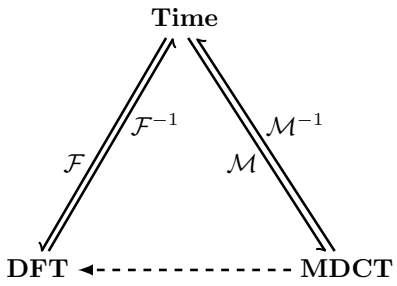

Fig. 1. Connections between the time, DFT, and MDCT domains. $\mathcal{F}$ denotes the DFT operation, and $\mathcal{M}$ denotes the MDCT operation.

Therefore, if one wants to apply some TF-domain signal processing on signals that are coming from perceptual audio decoders, one has the following two possibilities: 1) develop (often tricky and over-specific) MDCT-domain processing (e.g., [8] for instantaneous frequency estimation; see [6] for a review of several other examples of MDCT-domain processing), or 2) go to the DFT domain. The latter possibility is the more general, and currently, most audio processing systems that are cascaded with audio coders consider this solution.

The plain way to go from the MDCT to the DFT is to first go from the MDCT back to time using the inverse MDCT (IMDCT) and then go from time to the DFT, or the "IMDCT + DFT" scheme (Fig. 1). But there are two related drawbacks with this indirect method: Nonlocality and complexity. The "IMDCT + DFT" scheme works on complete spectra, even if only a subband conversion is intended. In other words, we need to apply the IMDCT on the whole MDCT spectra before DFT calculation, and this is true even if a limited number of DFT coefficients are intended. However, both the MDCT and the DFT decompose time signals into orthogonal trigonometric functions with evenly spaced frequencies. Thus, MDCT spectra and DFT spectra of the same time signal should look alike roughly, and a DFT coefficient should mainly depend on a few MDCT coefficients at nearby frequencies. Therefore, it is legitimate to look for a direct MDCT-to-DFT transform that would exploit such local relationship between MDCT and DFT coefficients. This would allow efficient calculation of specific DFT coefficients from a reduced set of MDCT coefficients at nearby frequencies, hence reduce complexity in subband conversion, and possibly reduce complexity for the fullband

S. Zhang and L. Girin are with the Grenoble Laboratory of Images, Speech, Signals, and Automation (GIPSA-lab), Grenoble Institute of Technology, Grenoble, France (see http://www.gipsa-lab.grenoble-inp.fr).

This work is supported by the French National Research Agency (ANR) as part of the DReaM project — ANR CONTINT 2009-006.

Manuscript submitted May 11, 2012.

[1]DFT refers to the discrete version of the Fourier transform as applied on a given frame of signal, whereas STFT refers to a set of DFTs applied on successive (generally ovelapping) signal frames. There is no such literary distinction for the MDCT: this term can refer to a given MDCT frame — as for the proposed MDCT-to-DFT conversion which is a frame-wise process — or it can refer to the overall set of MDCT frames, depending on the context.

conversion either.

In [11], the so-called mapping methods were proposed for the MDCT-to-DFT conversion, but this was only to save memory usage, not computational complexity. In [12], a method for directly converting the MDCT to the Modified Discrete Sine Transform (MDST) was proposed, which can be easily extended to direct conversion from the MDCT to the Modified Complex Lapped Transform (MCLT) [13], [14] — a special shifted DFT. However, the window functions for the MDCT and the MDST are required to be identical. This is sufficient when the MCLT is intended, but not when the DFT or a general shifted DFT is intended. Indeed, in practice different window functions are very often used for the MDCT (e.g., the Kaiser-Bessel-Derived (KBD) window [15] or the sine window that both ensure the perfect reconstruction property, for coding [1]) and the DFT (e.g., Hamming or Hanning windows, for spectral analysis and processing). In [6], an intermediate transform called Circulant Lapped Transform (CLT) has been proposed to convert the MDCT to the DFT, i.e., MDCT-to-CLT followed by CLT-to-DFT. The overall process is shown to be efficiently approximated by a complex-valued Finite Impulse Response (FIR) filtering applied on MDCT coefficients. But the conversion is limited to the DFT with the rectangular window (and the MDCT with an arbitrary symmetric window).

In the present paper, we propose a new direct MDCT-to-DFT conversion process that has the following advantages. Most importantly, it overcomes the limitation of the above methods concerning the window functions: It is valid for any arbitrary pair of MDCT and DFT windows. Also, this process is more efficient in the sense that it does not rely on an intermediate representation (such as the CLT) while also leading to a fast and accurate FIR implementation. This FIR implementation inherently allows locality of the MDCT-to-DFT conversion, i.e., it can be applied on a subband basis, and has a low complexity, even for full-band conversion. Fast algorithms for the MDCT [9] and the DFT [10] all have computational complexity of $\mathcal{O}(M \log_2 M)$ (M being the number of MDCT coefficients for a single transform, or $1/2$ of the frame size), while the proposed direct conversion method only has computational complexity of $\mathcal{O}(M)$. Moreover, unlike fast MDCT or FFT that depends on complicated bit-reverse indexing and butterfly operations, the direct method needs only vector scaling and vector addition, thus it is much more simpler to implement and also memory efficient. The resulting conversion process can be plugged on the output of any perceptual audio coder based on MDCT representation to provide DFT coefficients corresponding to any arbitrary window, hence ready-to-use for a large set of audio/speech processing applications.

The rest of the paper is organized as follows. In Section II, a general form of matrix transformation from MDCT to DFT vectors is presented. In Section III, the specific structure of the conversion matrices is investigated. The fast FIR implementation is derived from this specific structure in Section IV. Section V presents the accurate low-order approximation of the FIR-based conversion, numerical simulations, and an example of application that validate this approach. Section VI concludes the paper.

II. Matrix transformation from MDCT to DFT

Assuming that the MDCT is calculated with a window function $w_c(n)$ that satisfies the perfect reconstruction condition [1], [3], then MDCT coefficients can be transformed back to time samples, which can be further transformed to DFT coefficients. Therefore, in this section, we first provide the expression of the DFT coefficients (of a given signal frame) as a linear transformation of the MDCT coefficients (of the same frame and neighboring frames). For this aim, let us express the DFT matrix $\mathbf{F}$ and the MDCT matrix $\mathbf{P}$ as trigonometric matrices. Here a trigonometric matrix is the product of a real diagonal matrix (window function part) and a matrix whose entries are of the form of $W_N[\alpha] \equiv \exp[-j\frac{2\pi}{N}\alpha]$ or its real part. Let M be the size of the MDCT (the number MDCT coefficients for a single transform). The matrix $\mathbf{F}$ is of size $N \times N$ with $N = 2M$, $\mathbf{P}$ is of size $N \times M$, and we have[2]:

$$\begin{cases} \mathbf{F}(n,k) = w_f(n) W_{2M}[-nk], \\ \mathbf{P}(n,l) = C w_c(n) \mathrm{Re}\big\{W_{2M}[(n+\frac{1}{2}+\frac{M}{2})(l+\frac{1}{2})]\big\}, \end{cases} \tag{1}$$

where $C = \sqrt{2/M}$ for energy normalization. Note that we can have different arbitrary window functions $w_f(n)$ for $\mathbf{F}$ and $w_c(n)$ for $\mathbf{P}$ (but remind that $w_c(n)$ must satisfy the perfect reconstruction condition).

Given a time vector $\boldsymbol{x}$ size of $2M$, whose first and second halves are $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$, respectively, the corresponding MDCT coefficient vector $\boldsymbol{X}$ size of M is

$$\boldsymbol{X} = \mathbf{P}'\boldsymbol{x} \equiv \mathbf{P}'_0\boldsymbol{x}_a + \mathbf{P}'_1\boldsymbol{x}_b, \tag{2}$$

where $\mathbf{P}_0, \mathbf{P}_1$ are the first and last M rows of $\mathbf{P}$, respectively. Let us denote by $\boldsymbol{x}_u$, $u = 0, 1, 2, 3$, four consecutive sample vectors of size M, and denote by $\boldsymbol{x}_{u,u+1}$, $u = 0, 1, 2$, the concatenation of $\boldsymbol{x}_u$ and $\boldsymbol{x}_{u+1}$ (see Fig. 2). Then MDCT coefficient vector $\boldsymbol{X}_{u,u+1} \equiv \mathbf{P}'\boldsymbol{x}_{u,u+1} = \mathbf{P}'_0\boldsymbol{x}_u + \mathbf{P}'_1\boldsymbol{x}_{u+1}$ from (2). By the inverse MDCT (IMDCT) and the overlap-add operation, time samples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ can be recovered from the MDCT coefficients, but from the last, current, and next frames:

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_1 & \mathbf{P}_0 \end{bmatrix} \begin{bmatrix} \boldsymbol{X}_{01} \\ \boldsymbol{X}_{12} \\ \boldsymbol{X}_{23} \end{bmatrix}, \tag{3}$$

if $\mathbf{P}_1\mathbf{P}'_0 = \mathbf{P}_0\mathbf{P}'_1 = \mathbf{0}$ and $\mathbf{P}_0\mathbf{P}'_0 + \mathbf{P}_1\mathbf{P}'_1 = \mathbf{I}$, which is ensured by the perfect reconstruction condition of the window function $w_c(n)$.

Similarly, for the DFT, let us denote by $\mathbf{F}_0$ and $\mathbf{F}_1$ the first and last M rows of $\mathbf{F}$. Thus the DFT carries the time vector $\boldsymbol{x}_{u,u+1}$ to DFT coefficient vector $\boldsymbol{Z}_{u,u+1} \equiv \mathbf{F}'\boldsymbol{x}_{u,u+1} = \mathbf{F}'_0\boldsymbol{x}_u + \mathbf{F}'_1\boldsymbol{x}_{u+1}$ for $u = 0, 1, 2$. Combining this latter equation for $u = 1$ with (3), the DFT coefficient vector $\boldsymbol{Z}_{12}$ for the time

[2]Note that for clarity, we adopt the "C convention" for vector/matrix entries indexing, i.e., all indexes go from 0 to number of terms minus one. Vectors are column oriented if not specified otherwise. The symbol $^\top$ denotes transpose, * denotes conjugate, and $'$ denotes transpose and conjugate.

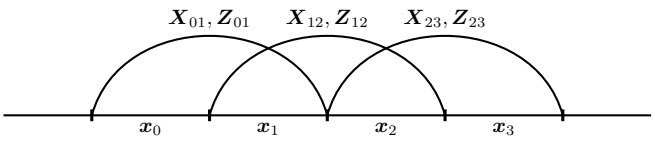

Fig. 2. Overlap and add in the time domain and corresponding TF-domain vectors. The time vectors $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2$ and $\boldsymbol{x}_3$ all have the same size M.

vector $\boldsymbol{x}_{12}$ is given by

$$\begin{aligned}\boldsymbol{Z}_{12} &= \mathbf{F}'\boldsymbol{x}_{12} \\ &= \begin{bmatrix}\mathbf{F}'_0 & \mathbf{F}'_1\end{bmatrix}\begin{bmatrix}\mathbf{P}_1 & \mathbf{P}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_1 & \mathbf{P}_0\end{bmatrix}\begin{bmatrix}\boldsymbol{X}_{01} \\ \boldsymbol{X}_{12} \\ \boldsymbol{X}_{23}\end{bmatrix} \\ &\equiv \begin{bmatrix}\mathbf{T}_{01} & \mathbf{T}_{12} & \mathbf{T}_{23}\end{bmatrix}\begin{bmatrix}\boldsymbol{X}_{01} \\ \boldsymbol{X}_{12} \\ \boldsymbol{X}_{23}\end{bmatrix}, \end{aligned} \tag{4}$$

where

$$\begin{cases}\mathbf{T}_{01} \equiv \mathbf{F}'_0\mathbf{P}_1 \\ \mathbf{T}_{12} \equiv \mathbf{F}'\mathbf{P} = \mathbf{F}'_0\mathbf{P}_0 + \mathbf{F}'_1\mathbf{P}_1 \\ \mathbf{T}_{23} \equiv \mathbf{F}'_1\mathbf{P}_0\end{cases} \tag{5}$$

are called conversion matrices, size of $(2M)\times M$. Thereby, the DFT coefficient vector of a given frame is obtained from the MDCT coefficient vectors of the previous, current and next frames, using three conversion matrices $\mathbf{T}_{01}$, $\mathbf{T}_{12}$ and $\mathbf{T}_{23}$, which depend only on the window functions $w_f(n)$ and $w_c(n)$, and share a specific structure that we shall see in the next section.

In the following, time samples are supposed to be real, thus their DFT spectra are conjugate symmetric. Therefore, we only consider the first $M+1$ rows of $\mathbf{T}_{01}, \mathbf{T}_{12}, \mathbf{T}_{23}$, hence the conversion matrices are reduced from the original size $2M \times M$ to truncated size $(M+1)\times M$.

III. Structure of the Conversion Matrices

Each entry of the three conversion matrices $\mathbf{T}_{01}, \mathbf{T}_{12}$, and $\mathbf{T}_{23}$ is an inner product between one DFT basis vector and one MDCT basis vector. This gives a specific structure to the matrices regardless of the MDCT window function $w_c(n)$ and the DFT window function $w_f(n)$[3].

A. Phase shift, Toeplitz and Hankel matrices

For the purpose of generality, i.e., deriving common properties for $\mathbf{T}_{01}, \mathbf{T}_{12}$, and $\mathbf{T}_{23}$, let us define two trigonometric matrices $\mathbf{U}$ and $\mathbf{V}$ given by

$$\begin{cases}\mathbf{U}(n,k) = w_0(n)W_{2M}[-(n+n_0)k], \\ \mathbf{V}(n,l) \;\; = Cw_1(n)\mathrm{Re}\{W_{2M}[(n+n_1)(l+\frac{1}{2})]\},\end{cases} \tag{6}$$

where $w_0(n), w_1(n)$ are real window functions, and n_0, n_1 are time shifts. We first study the product of $\mathbf{U}'$ and $\mathbf{V}$, and we will apply the results to $\mathbf{T}_{01}, \mathbf{T}_{12}$, and $\mathbf{T}_{23}$ in Section IV with specific settings of w_0, w_1, n_0 and n_1.

Remark 1. In the first equation of (6), by changing $[-(n+n_0)k]$ to $[-(n+n_0)(k+k_0)]$, where k_0 is a constant frequency shift, we can derive conversion from the MDCT to the shifted DFT, including the MCLT. Since the real part of the MCLT is simply the MDCT and the imaginary part is the MDST, we can derive the MDCT-to-MDST conversion. But for clarity of notations and mathematical development (at the cost of minor generality), we restrict our development to the conversion from the MDCT to the standard DFT.

The product of $\mathbf{U}'$ and $\mathbf{V}$ has the form

$$\begin{aligned}\mathbf{T}(k,l) &\equiv [\mathbf{U}'\mathbf{V}](k,l) \\ &= \phi(k)[h(k-l-1) + h(k+l)],\end{aligned} \tag{7}$$

where

$$\phi(k) = W_{2M}[(n_0-n_1)k], \tag{8}$$

$$h(l) = \frac{C}{2}\sum_{n=0}^{N-1} W_{2M}[(n+n_1)(l+\tfrac{1}{2})]w_0(n)w_1(n). \tag{9}$$

Here $\phi(k)$ is frequency-dependent phase shift, and $h(l)$ is the frequency response of $w_0(n)w_1(n)$ with time and frequency shift. See Appendix A for detailed derivation. Both $w_0(n)$ and $w_1(n)$ are real, thus frequency response $h(l)$ is conjugate symmetric about $l=-\frac{1}{2}$ and $2M$-periodic except for a phase term $\mu \equiv e^{-j2\pi n_1}$:

$$\begin{cases}h(-l-\frac{1}{2}) \;= h(l-\frac{1}{2})^*, \\ h(l+2M) = \mu h(l).\end{cases} \tag{10}$$

From (7), we see that matrix $\mathbf{T}$ can be factored into two matrices, the first one is $\mathbf{\Psi} \equiv \mathrm{diag}\{\phi(k)\}$ for phase shift and the second one is a sum of a Toeplitz matrix [16] and a Hankel matrix:

$$\begin{cases}\mathbf{T}_{\mathrm{toep}}(k,l) \equiv h(k-l-1), \\ \mathbf{T}_{\mathrm{hank}}(k,l) \equiv h(k+l).\end{cases} \tag{11}$$

This way (7) can be written as

$$\mathbf{T} = \mathbf{U}'\mathbf{V} = \mathbf{\Psi}(\mathbf{T}_{\mathrm{toep}} + \mathbf{T}_{\mathrm{hank}}). \tag{12}$$

Note that this structure is shared by the conversion matrices $\mathbf{T}_{01}, \mathbf{T}_{12}$, and $\mathbf{T}_{23}$ since $\mathbf{F}_0, \mathbf{F}, \mathbf{F}_1$ are special cases of $\mathbf{U}$ and $\mathbf{P}_0, \mathbf{P}, \mathbf{P}_1$ are special cases of $\mathbf{V}$ by (4) and (1).

Product of a Toeplitz matrix and a vector is equivalent to FIR filtering applied to the vector; Product of a Hankel matrix and a vector is equivalent to FIR filtering applied to the reversed vector. Therefore, applying matrix $\mathbf{T}$ of (12) to a MDCT vector $\boldsymbol{X}$ is equivalent to two FIR filtering processes, one applied to $\boldsymbol{X}$ in the order of bin 0 to bin $M-1$ and the other applied to the same vector $\boldsymbol{X}$ but in the order of bin $M-1$ down to bin 0. However, neither $\mathbf{T}_{\mathrm{toep}}$ nor $\mathbf{T}_{\mathrm{hank}}$ is ready for vectorized implementation of FIR filtering, because different rows have different sets of non-zero entries (see the top two matrices in Fig. 3), thus modification of filter taps are needed for different DFT bins. In the following, we shall see that it is possible to reorganize the entries of matrix

[3]As already mentioned in the introduction, this is a notable extension to the previous work [6] where a rectangular window function was considered for the DFT. Note also that a specific matrix structure was also exploited in [6] but this was within the MDCT-to-CLT conversion, although we consider here direct MDCT-to-DFT conversion matrices.

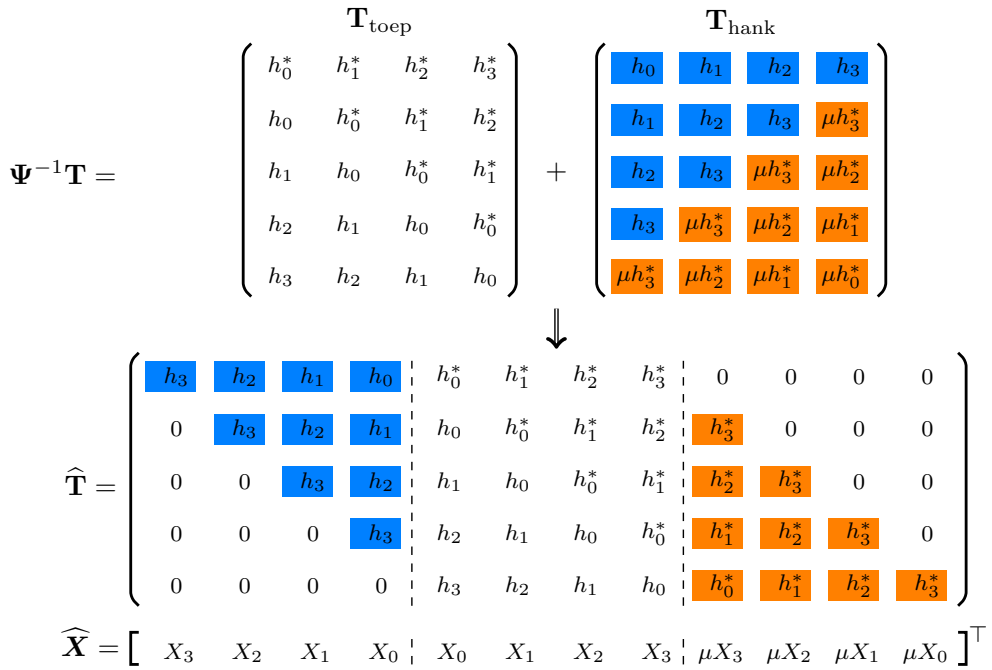

Fig. 3. Splitting and mirroring of the Hankel matrix $\mathbf{T}_{\text{hank}}$, and composition of the extended Toeplitz matrix $\widehat{\mathbf{T}}$. Here $M = 4$, $X_l = \boldsymbol{X}(l)$, and $h_l = h(l)$ for illustration.

$\mathbf{T}_{\text{toep}} + \mathbf{T}_{\text{hank}}$ into a global Toeplitz matrix, where each row has the same sequence of non-zero entries, leading to a single FIR filtering process ready for vectorized implementation.

B. Splitting and mirroring of the Hankel matrix

Equation (11) shows that the Toeplitz and the Hankel parts are connected by the frequency response function $h(l)$, whose properties (10) in turn lead to explicit relations between the two parts:

$$\mathbf{T}_{\text{toep}}(k, -l-1) = \mathbf{T}_{\text{hank}}(k, l), \tag{13}$$
$$\mathbf{T}_{\text{toep}}(k, M+l) = \mu^* \mathbf{T}_{\text{hank}}(k, M-l-1), \tag{14}$$

provided *negative* column indexing is permitted, which is equivalent to extending matrices to the left. Observe that the column indexes in (13) add up to -1 thus can be seen as mirrored values about $l = -\frac{1}{2}$. Similarly, the column indexes in (14) add up to $2M-1$ thus can be seen as mirrored values about $l = M - \frac{1}{2}$. This mirror symmetry allows the matrix $\mathbf{T}_{\text{hank}}$ to be split and mirrored, and then combined with the matrix $\mathbf{T}_{\text{toep}}$ to form an extended matrix $\widehat{\mathbf{T}}$:

$$\widehat{\mathbf{T}}(k,l) \equiv \begin{cases} \mathbf{T}_{\text{hank}}(k, -l-1), & -M+k \le l < 0, \\ \mathbf{T}_{\text{toep}}(k,l), & 0 \le l < M, \\ \mu^* \mathbf{T}_{\text{hank}}(k, 2M-l-1), & M \le l < M+k, \end{cases} \tag{15}$$

for $k = 0, 1, \ldots, M$. This process is illustrated in Fig. 3. The splitting of $\mathbf{T}_{\text{hank}}$ is along $k + l = M - \frac{1}{2}$, then the upper left part is mirrored to the left about $l = -\frac{1}{2}$ and the lower right part is mirrored to the right about $l = M - \frac{1}{2}$. $\mathbf{T}_{\text{toep}}$ is then inserted between the two mirrored parts of $\mathbf{T}_{\text{hank}}$. The resulting extended matrix $\widehat{\mathbf{T}}$ is a Toeplitz matrix size of $(M+1) \times 3M$:

$$\widehat{\mathbf{T}}(k,l) = \begin{cases} h(k-l-1), & \text{if } -M+1 \le k-l \le M, \\ 0, & \text{else.} \end{cases} \tag{16}$$

If $\mathbf{T}_{\text{toep}} + \mathbf{T}_{\text{hank}}$ is replaced with $\widehat{\mathbf{T}}$, the vector $\boldsymbol{X}$ needs to be replaced with an extended version $\widehat{\boldsymbol{X}}$ that echos (15) so that by (15) and (12), we have

$$\boldsymbol{Y} \equiv \boldsymbol{\Psi}^{-1}\mathbf{T}\boldsymbol{X} = (\mathbf{T}_{\text{toep}} + \mathbf{T}_{\text{hank}})\boldsymbol{X} = \widehat{\mathbf{T}}\widehat{\boldsymbol{X}}. \tag{17}$$

The extended vector $\widehat{\boldsymbol{X}}$ is given by (see Appendix B for the details)

$$\widehat{\boldsymbol{X}}(l) \equiv \begin{cases} \boldsymbol{X}(-l-1), & -M \le l < 0, \\ \boldsymbol{X}(l), & 0 \le l < M, \\ \mu \boldsymbol{X}(2M-l-1), & M \le l < 2M, \end{cases} \tag{18}$$

which can also be viewed as padding for the finite length input $\boldsymbol{X}$. As can be seen in Fig. 3, the vector $\boldsymbol{X}^\top$ is mirrored about $l = -\frac{1}{2}$ to the left and about $l = M - \frac{1}{2}$ to the right (with multiplication by μ). Finally, the rightmost term of (17) is a vectorized FIR filtering process applied on $\boldsymbol{X}$, as detailed in the next section.

IV. Symmetric FIR filtering of MDCT coefficients

A. Basic implementation

Let us now apply the developments of Section III to the MDCT-to-DFT conversion problem. Eq. (17) can be written

as

$$\begin{aligned}\boldsymbol{Y}(k) &= \sum_{l=-M}^{2M-1} \widehat{\mathbf{T}}(k,l)\widehat{\boldsymbol{X}}(l) \\ &= \sum_{l=k-M}^{k+M-1} h(k-l-1)\widehat{\boldsymbol{X}}(l) \\ &= \sum_{l=-M}^{M-1} h(l)\widehat{\boldsymbol{X}}(k-l-1) \\ &= \sum_{l=0}^{M-1} \{h(l)\widehat{\boldsymbol{X}}(k-l-1) + h^*(l)\widehat{\boldsymbol{X}}(k+l)\}, \end{aligned} \tag{19}$$

where the last equation is due to the conjugate symmetry of $h(l)$ in (10). Therefore, by extending $\mathbf{T}_{01}$, $\mathbf{T}_{12}$, and $\mathbf{T}_{12}$ in the way of extending $\mathbf{T}$ by (15), and extending $\boldsymbol{X}_{01}$, $\boldsymbol{X}_{12}$, and $\boldsymbol{X}_{23}$ in the way of extending $\boldsymbol{X}$ by (18), the matrix form of direct MDCT-to-DFT conversion (4) can be equivalently represented as FIR filtering.

Remark 2. Note that in (19), the FIR filtering, or convolution, is applied in the MDCT domain. This is not the usual fast implementation of time-domain convolution by frequency-domain point-wise product. Also in (19), for each $k = 0, 1, \ldots, M$, the filtering process requires the same sequence of filter taps $h(-M), h(-M+1), \ldots, h(M-1)$, thus can be easily implemented in a vectorized manner. In [12], direct MDCT-to-MDST conversion with the same window function was derived through trigonometric manipulations, resulting in a FIR filtering similar to (19). However, a key difference is that in [12], outputs at different bins require different segments of a filter taps sequence, which complicates implementation.

Let us now calculate the phase term $\phi(k)$ and FIR taps $h(l)$ for each of $\mathbf{T}_{01}$, $\mathbf{T}_{12}$, and $\mathbf{T}_{23}$. For shorthand, let us denote here

$$\begin{cases} \varphi(k) \equiv W_{2M}[-(\frac{1}{2} - \frac{1}{2}M)k], \\ f(n,l) \equiv (n + \frac{1}{2} + \frac{1}{2}M)(l + \frac{1}{2}). \end{cases} \tag{20}$$

Then let us substitute time shifts of $\mathbf{F}_0$, $\mathbf{F}$, and $\mathbf{F}_1$ in place of n_0, and times shifts of $\mathbf{P}_0$, $\mathbf{P}$, and $\mathbf{P}_1$ in place of n_1 for matrices $\mathbf{U}$ and $\mathbf{V}$ defined in (6). From (8), we have the phase terms:

$$\phi_{01}(k) = (-1)^k \phi_{12}(k) = \phi_{23}(k) = \varphi(k), \tag{21}$$

and from (9), we have the filter taps determined by the window functions $w_f(n)$ and $w_c(n)$:

$$\begin{cases} h_{01}(l) = \frac{C}{2} \sum_{n=M}^{2M-1} W_{2M}[f(n,l)] w_f(n-M) w_c(n), \\ h_{12}(l) = \frac{C}{2} \sum_{n=0}^{2M-1} W_{2M}[f(n,l)] w_f(n) w_c(n), \\ h_{23}(l) = \frac{C}{2} \sum_{n=0}^{M-1} W_{2M}[f(n,l)] w_f(n+M) w_c(n). \end{cases} \tag{22}$$

See Appendix C for the details. Also, the vectors $\boldsymbol{X}_{01}, \boldsymbol{X}_{12}, \boldsymbol{X}_{23}$ are extended to $\widehat{\boldsymbol{X}}_{01}, \widehat{\boldsymbol{X}}_{12}, \widehat{\boldsymbol{X}}_{23}$, respectively, using (18) and here $\mu = (-1)^{M+1}$ ($n_1 = \frac{M}{2} + \frac{1}{2}$). Then applying (21) and (22) to (17), (16), and (4), we have

$$\begin{aligned}\boldsymbol{Z}_{12}(k) = \varphi(k) \sum_{l=-M}^{M-1} \Big[&(-1)^k h_{12}(l) \widehat{\boldsymbol{X}}_{12}(k-l-1) \\ &+ h_{01}(l) \widehat{\boldsymbol{X}}_{01}(k-l-1) + h_{23}(l) \widehat{\boldsymbol{X}}_{23}(k-l-1) \Big]. \end{aligned} \tag{23}$$

Therefore, each of h_{01}, h_{12}, h_{23} can be seen as a FIR filter with $2M$ taps and is conjugate symmetric by (10). The FIR filtering processing of (23) is represented as a flowchart diagram in Fig. 4(a).

B. Filter coefficients calculation using DFT and alternative implementation

One way to compute the filter taps is to use (22) directly. But it is also possible to compute the filter taps by the DFT, that is, DFT of element-wise window function product with appropriate pre- and post-twiddle. Let

$$w_f^{\pm}(n) = \begin{cases} w_f(n+M), & \text{for } 0 \le n < M, \\ \pm w_f(n-M), & \text{for } M \le n < 2M, \end{cases}$$

be two circular extensions of $w_f(n)$, which are different only in sign at their second halves. Let $h_0(l) \equiv h_{12}(l)$ and $h_{\pm}(l) \equiv h_{23}(l) \pm h_{01}(l)$. Then, we have

$$\begin{cases} h_0(l) = \frac{C}{2} \psi_{\text{post}}(l) \mathcal{F}_{2M}\{\psi_{\text{pre}}(n)[w_f(n) w_c(n)]\}, \\ h_+(l) = \frac{C}{2} \psi_{\text{post}}(l) \mathcal{F}_{2M}\{\psi_{\text{pre}}(n)[w_f^+(n) w_c(n)]\}, \\ h_-(l) = \frac{C}{2} \psi_{\text{post}}(l) \mathcal{F}_{2M}\{\psi_{\text{pre}}(n)[w_f^-(n) w_c(n)]\}, \end{cases} \tag{24}$$

where $\mathcal{F}_{2M}$ denotes DFT size of $2M$, and the pre- and post-twiddle factors are

$$\begin{cases} \psi_{\text{post}}(l) \equiv W_{2M}[(\frac{1}{2} + \frac{1}{2}M)l], \\ \psi_{\text{pre}}(n) \equiv W_{2M}[\frac{1}{2}(n + \frac{1}{2} + \frac{1}{2}M)], \end{cases} \tag{25}$$

for $n, l = 0, 1, \ldots, M-1$ (see Appendix C). Fig. 5 provides an illustration of window function products and Fig. 6 provides an illustration of the magnitude of the corresponding filter taps (discussed in Section V-A). Note that an important case is when both $w_f(n)$ and $w_c(n)$ are symmetric. Then the filter taps in (24) have equal real and imaginary parts except for sign (see Appendix C):

$$\begin{cases} \text{Im}\{h_0(l)\} = (-1)^l \text{Re}\{h_0(l)\}, \\ \text{Im}\{h_+(l)\} = (-1)^l \text{Re}\{h_+(l)\}, \\ \text{Im}\{h_-(l)\} = (-1)^{l+1} \text{Re}\{h_-(l)\}, \end{cases} \tag{26}$$

Obviously, the taps $h_{01}(l)$, $h_{12}(l)$, and $h_{23}(l)$ can be easily recovered from (24) with $h_{01}(l) = \frac{1}{2}(h_+(l) - h_-(l))$, $h_{12}(l) = h_0(l)$, and $h_{23}(l) = \frac{1}{2}(h_+(l) + h_-(l))$, and since this calculation is made only once, the FIR process can be then implemented with (23). Alternately, the coefficients of (24) can be used directly in the equivalent FIR filtering:

$$\begin{aligned}\boldsymbol{Z}_{12}(k) = \varphi(k) \sum_{l=-M}^{M-1} \Big[&(-1)^k h_0(l) \widehat{\boldsymbol{X}}_0(k-l-1) \\ &+ h_-(l) \widehat{\boldsymbol{X}}_-(k-l-1) + h_+(l) \widehat{\boldsymbol{X}}_+(k-l-1) \Big], \end{aligned} \tag{27}$$

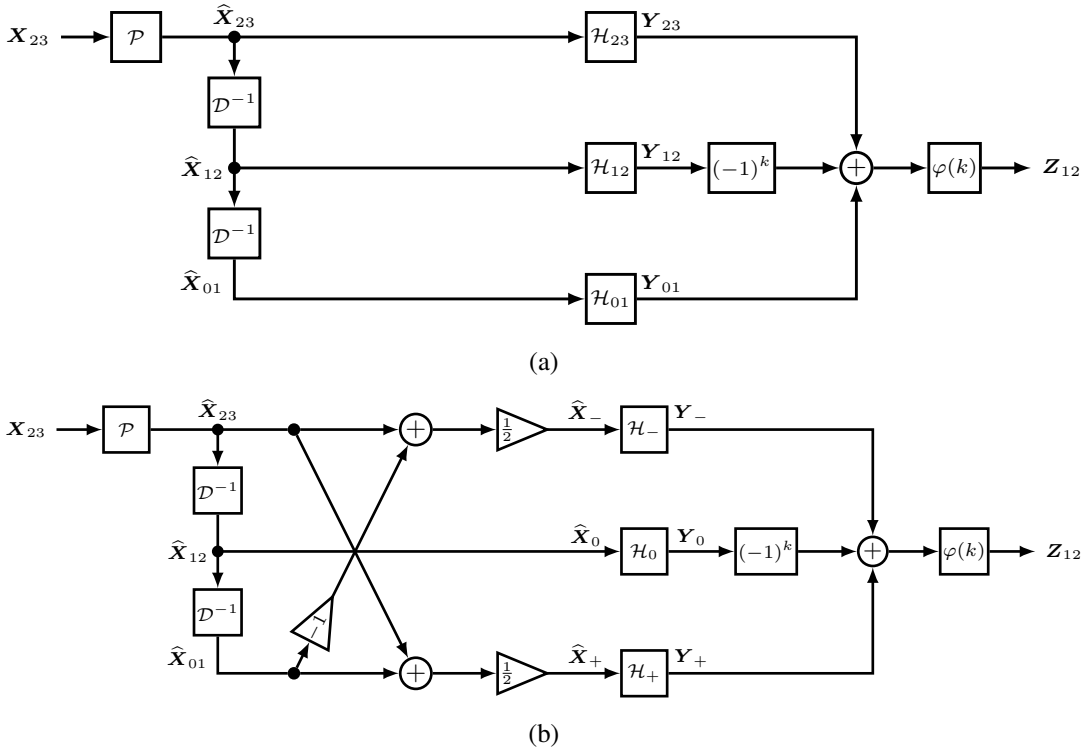

Fig. 4. (a) Flowchart of direct MDCT-to-DFT conversion by (23); (b) Flowchart of direct MDCT-to-DFT conversion by (27). Here $\mathcal{D}^{-1}$ denotes one frame delay of the MDCT spectrum, $\mathcal{P}$ denotes extension (padding) operation for the MDCT spectra by (18), $\mathcal{H}_{[.]}$ denotes convolution with the filter $h_{[.]}$.

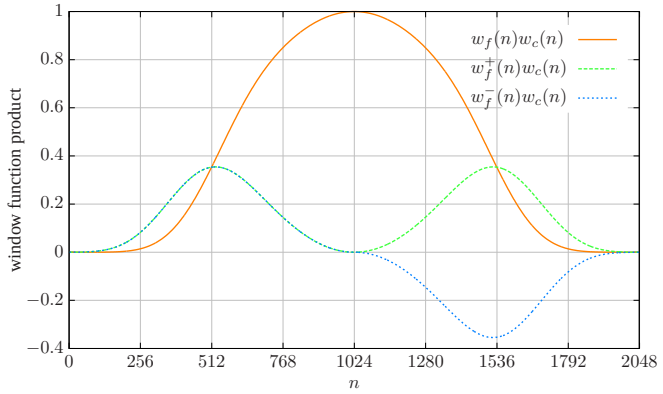

Fig. 5. Product of window functions. Here $M = 1024$, $w_f(n)$ is a Hanning window and $w_c(n)$ is a KBD window.

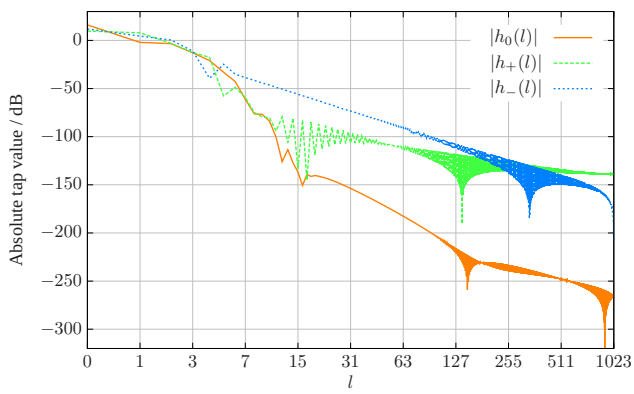

Fig. 6. Filter taps for MDCT to DFT conversion (log-magnitude). Here $M = 1024$, $w_f(n)$ is a Hanning window and $w_c(n)$ is a KBD window.

where $\widehat{\boldsymbol{X}}_0 \equiv \widehat{\boldsymbol{X}}_{12}$ and $\widehat{\boldsymbol{X}}_{\pm} \equiv \frac{1}{2}[\widehat{\boldsymbol{X}}_{23} \pm \widehat{\boldsymbol{X}}_{01}]$ for consistency. This alternative FIR filtering processing is represented as a flowchart diagram in Fig. 4(b). Although a little bit more complicated than the implementation of (23)/Fig. 4(a) because of the MDCT vectors addition/subtraction, we will consider this implementation in the next section since we shall see that (24) is not only fast way to compute filter taps but also plays a key role in addressing the problem of low-order FIR approximation and convergence of filter taps as $M \to \infty$. Because of the tight and simple links between both sets of filters, similar derivations and conclusions can be drawn from the simpler implementation (23).

V. Low-order FIR approximation

A. Energy concentration of filter taps

Both the MDCT and the DFT are Fourier-type spectral transforms with a grounded physical interpretation in terms of energy concentration around the spectral components of the transformed signal. Given a pure tonal signal $x(n) = \cos[\omega n]$ where the frequency $\omega \in [0, \pi)$, with appropriate smooth windowing functions, both its MDCT spectrum $\boldsymbol{X}(k)$ and its DFT spectrum $\boldsymbol{Z}(k)$ concentrate at $k \approx \omega M/\pi$. Moreover, $|\boldsymbol{X}(k)|^2$ and $|\boldsymbol{Z}(k)|^2$ should have similar shapes since they both approximate the power spectrum of $x(n)$ (for a discussion on the specific shape of MDCT spectra, see, e.g., [17] and

[5]). Therefore, only minor *local* modifications should be needed to go from $\boldsymbol{X}(k)$ to $\boldsymbol{Z}(k)$, i.e., each coefficient $\boldsymbol{Z}(k_0)$ should be fairly well reconstructed from $\boldsymbol{X}(k_0)$ and the neighboring coefficients, and therefore, we can expect some energy concentration of the three filters $h_0(l), h_-(l), h_+(l)$ in the coefficients around $l = 0$. Note that this echoes the discussion about the locality vs. non-locality of the conversion as begun in the introduction.

This is totally compliant with the fact that from the perspective of multirate filterbank, smooth window functions $w_f(n)$ and $w_c(n)$ are impulse responses of low-pass prototype FIR filters [18], and therefore, in the frequency domain both $w_f(n)$ and $w_c(n)$ are assumed to have a narrow mainlobe centered around 0 that concentrates most of the taps energy (one of the most important design goals of window functions [19]). This is, for example, the case for the Hamming or Hanning window used for the DFT, and the sine or KBD window used for the MDCT. Then, this will also be the case for the product functions $w_f(n)w_c(n)$ and $w_f^{\pm}(n)w_c(n)$. More specifically, suppose that the mainlobes of the frequency responses of $w_f(n)$ and $w_c(n)$ are within $|\omega| < \omega_f \ll \pi$ and $|\omega| < \omega_c \ll \pi$, respectively. Since time-domain pointwise product corresponds to frequency-domain convolution, by (24), most energy of $h_0(l)$ will be within

$$|l| < (\omega_f + \omega_c)\frac{\pi}{M} \ll M.$$

Similar results can be drawn for $h_-(l)$ and $h_+(l)$.

The energy concentration property of the conversion filters is illustrated in Fig. 6. Here, the DFT window $w_f(n)$ is a Hanning window (widely used in, e.g., spectral analysis) and the MDCT window $w_c(n)$ is a KBD window (the most frequently used window in AAC coding [2]), Let us recall that the filter taps are conjugate symmetric (about $l = -\frac{1}{2}$) so that we only represent here their magnitude for positive indexes l. It can be seen that for the three filters the power of the taps decreases to 0 very quickly with the index l (note the $\log_2$ scale of the l-axis). Taps with $|l| > 7$ (i.e., 99.2% of $M = 1024$ taps) are more than 50 dB below the tap at $l = 0$; in other words, most of the taps energy is concentrated at a few low-frequency bins. Of course, how much precisely of energy is concentrated at low frequency taps depends on the DFT and MDCT window functions, but similar results are obtained with other window combinations.

B. Low-order approximation

Based on the above discussion, it is possible to approximate (27) accurately by keeping only several most significant taps, that is, keeping the coefficients of the conversion filters for $-m \le l < m$, with $m \ll M$, and setting the other coefficients to zeros. Furthermore, when doing that, it may be desirable that the number of coefficients kept be not the same for the three filters, i.e., $m = m_0$ for $h_0(l)$, $m = m_+$ for $h_+(l)$, and $m = m_-$ for $h_-(l)$, resulting in an approximate FIR filtering[4]:

$$\begin{aligned}\boldsymbol{Z}_{12}(k) \approx \varphi(k)\Big\{&(-1)^k \\ &\times \sum_{l=0}^{m_0-1}\left[h_0(l)\widehat{\boldsymbol{X}}_0(k-l-1) + h_0^*(l)\widehat{\boldsymbol{X}}_0(k+l)\right] \\ &+ \sum_{l=0}^{m_--1}\left[h_-(l)\widehat{\boldsymbol{X}}_-(k-l-1) + h_-^*(l)\widehat{\boldsymbol{X}}_-(k+l)\right] \\ &+ \sum_{l=0}^{m_+-1}\left[h_+(l)\widehat{\boldsymbol{X}}_+(k-l-1) + h_+^*(l)\widehat{\boldsymbol{X}}_+(k+l)\right]\Big\}.\end{aligned} \tag{28}$$

Indeed, the three filters $h_0(l)$, $h_+(l)$ and $h_-(l)$ display the same general trends, but they generally do not have the same overall magnitude and decaying speed. As illustrated in Fig. 6, $h_0(l)$ usually decays faster to 0 than $h_\pm(l)$, due to their difference in the phasing of the window functions (Fig. 5). But on the other hand, $h_0(l)$ usually has larger total energy than $h_\pm(l)$, that is, $\sum_{n=0}^{M-1}|h_0(l)|^2 > \sum_{n=0}^{M-1}|h_+(l)|^2 = \sum_{n=0}^{M-1}|h_-(l)|^2$, by (24) and Parseval's theorem. Therefore, different values for m_0, m_+ and m_- in (28) can be set to obtain the best tradeoff between (high) conversion accuracy and (low) computational cost.

More specifically, the computational cost of the approximate FIR processing is proportional to the total number of kept taps $m_{\text{tot}} = m_0 + m_+ + m_-$. Its accuracy can be estimated in log signal-to-noise power terms as (see Appendix D):

$$\text{SNR} \approx 10\log_{10}\frac{1}{1-\sigma(m_0,m_+,m_-)/\sigma(M,M,M)}, \tag{29}$$

where

$$\sigma(a,b,c) \equiv \sum_{l=0}^{a-1}|h_0(l)|^2 + \sum_{l=0}^{b-1}|h_+(l)|^2 + \sum_{l=0}^{c-1}|h_-(l)|^2. \tag{30}$$

Eq. (29) can be used to predict approximation accuracy before filtering signals with given filters length, or to set m_{tot} and corresponding optimal values of m_0, m_+ and m_- given a target accuracy. For a given value of m_{tot}, one basic strategy to obtain m_0, m_+ and m_- is to sort out all coefficients $h_0(l)$, $h_-(l)$, $h_+(l)$ in decreasing order of their absolute values, then keep the first m_{tot} taps, and finally count out m_0, m_-, and m_+.

Following this sorting strategy for a range of m_{tot} values, it is shown in Fig. 7 that the estimated SNR of (29) closely follows the actual SNR (resulting from numerical simulations) for both random and musical signals (5×10^6 samples, or 113 s of 44.1 kHz signal), hence validating (29). This is observed here for both the Hanning-KBD and the Hanning-sine window configurations. The Hanning-KBD configuration appears to be significantly more accurate than the Hanning-sine given the same $m_{\text{tot}} > 16$, which is consistent to the significantly lower sidelobes of the KBD window than those

[4]Note that (28) reveals locality between MDCT spectra and DFT spectra, as discussed in Section V-A: An output coefficient $\boldsymbol{Z}_{12}(k)$ depends mainly on the $2m_0$, $2m_+$ and $2m_-$ consecutive input coefficients in $\widehat{\boldsymbol{X}}_0$, $\widehat{\boldsymbol{X}}_+$ and $\widehat{\boldsymbol{X}}_-$ centered at bin k, respectively. This is a direct consequence of energy concentration of the filter taps.

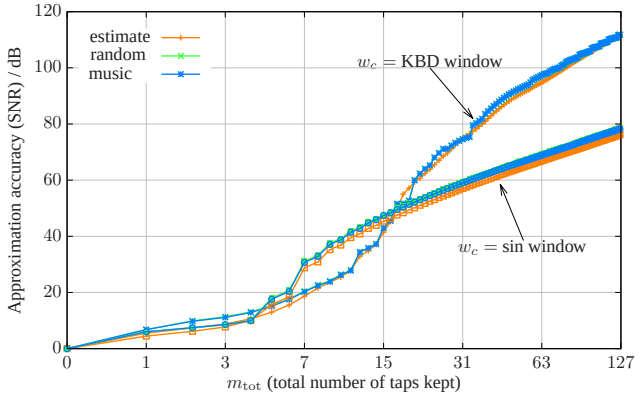

Fig. 7. Approximation accuracy versus the total number of taps kept. Here $M = 1024$, $w_f(n)$ is a Hanning window. Both the random and music signals have 5×10^6 samples.

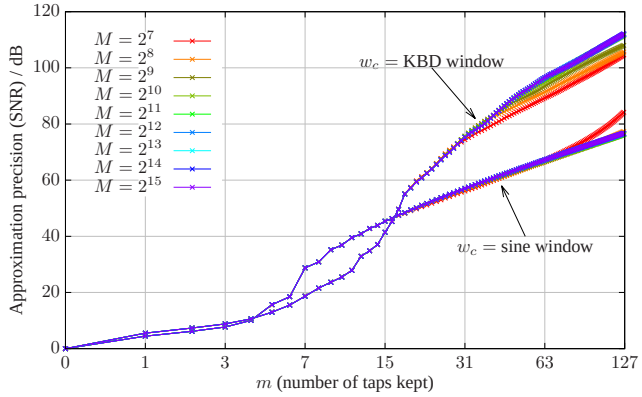

Fig. 8. Approximation accuracy versus the total number of taps kept over different M. Here $w_f(n)$ is always a Hanning window.

of the sine window[5]. We can also see from Fig. 7 that setting $m_{\rm tot} = 20$ is enough to keep SNR at about 60 dB for the Hanning-KBD configuration. Such a number of coefficients is very low compared to the MDCT size $M = 1024$. (Note that for the Hanning-KBD configuration, a reconstruction SNR about 100 dB is obtained with $m_{\rm tot} = 64$, i.e., $1/16$ of the number of MDCT coefficients.)

C. Asymptotic analysis of FIR approximation

As the MDCT size M increases, we may expect that, for a given accuracy, the required total number of taps $m_{\rm tot}$ increases too. But this is not the case: the required $m_{\rm tot}$ actually tends to *saturate* at some value even as $M \to \infty$, which guarantees $\mathcal{O}(M)$ complexity of the approximate FIR filtering process for a given accuracy. This saturation phenomenon is illustrated in Fig. 8, where the independence of $m_{\rm tot}$ w.r.t. M is visible for SNRs lower than, say, 60 dB with the Hanning-KBD configuration, and for SNRs lower than, say, 45 dB with the Hanning-sine configuration.

There are two reasons for this phenomenon: The first is that the total energy of the taps of each of the three filters in (27) is proportional to M; the second is that for a fixed l, the filter tap $h_0(l)$, or $h_-(l)$, or $h_+(l)$, when normalized (divided) by $\sqrt{2M}$, converges to a fixed value as $M \to \infty$. Thus, if first m_0, m_-, and m_+ taps of the three filters are to be kept, respectively, then the ratio of their energy to the total energy converges, and by (29), this implies that the accuracy converges too as $M \to \infty$. If this limit accuracy is no lower than the required accuracy, then $m_{\rm tot} = m_0 + m_- + m_+$ will ensure the required accuracy for any M, in other words, the needed $m_{\rm tot}$ saturates at some value as $M \to \infty$. Let $w(t)$ be a function defined on $[0, 1]$ whose periodical extension on $\mathbb{R}$ has a convergent Fourier series, and when discretely sampled, becomes the window function product with pre-twiddle $\psi_{\rm pre}(n)w_f(n)w_c(n)$ (or $\psi_{\rm pre}(n)w_f^{\pm}(n)w_c(n)$). By (24) and Parseval's theorem, as $M \to \infty$, we have

$$\begin{aligned}\frac{1}{2M}\sum_{l=0}^{M-1}|h_0(l)|^2 &= \sum_{n=0}^{2M-1}|\psi_{\rm pre}(n)w_f(n)w_c(n)|^2\frac{1}{4M}\\ &= \frac{1}{2}\sum_{n=0}^{2M-1}|w(n/(2M))|^2\frac{1}{2M}\\ &\to \frac{1}{2}\int_0^1|w(t)|^2\,dt. \qquad (31)\end{aligned}$$

Note that the post-twiddle factor, which does not change energy, is omitted here for simplicity, and

$$\begin{aligned}\frac{|h_0(l)|}{\sqrt{2M}} &= \Big|\sum_{n=0}^{2M-1} w(\tfrac{n}{2M})e^{-j\frac{\pi}{M}nl}\frac{1}{2M}\Big| \qquad (32)\\ &\to \Big|\int_0^1 w(t)e^{-j2\pi lt}\,dt\Big|. \qquad (33)\end{aligned}$$

The same is true for $h_-(l)$ and $h_+(l)$. Therefore, the two reasons mentioned above are valid and saturation of $m_{\rm tot}$ is proved.

D. Comparison with the plain MDCT-to-DFT conversion

The complexity of the direct MDCT-to-DFT conversion (filtering + phase-shifting) for a complete spectrum is $2m_{\rm tot}(M+1)+4(M+1)$ real multiplications and $2m_{\rm tot}(M+1)+2(M+1)$ real additions using (28), or totally $(4m_{\rm tot}+6)(M+1)$. On the other hand, the plain MDCT-to-DFT conversion scheme, that is, IMDCT + DFT, has the complexity of $(2+10\log_2 M)M+8$ of additions and multiplications (the fast IMDCT based on the FFT costs $2M\log_2(2M)$ [9]; the split-radix FFT costs $8M\log_2(2M)-12M+8$). Therefore, roughly, if $4m_{\rm tot}+6 < 2+10\log_2 M$, then, theoretically, the direct method will be faster than the plain method. For a typical $M = 1024$, we have $m_{\rm tot} < 24$.

Moreover, compared to the plain method, the proposed direct method has the two following major advantages. First, it works also locally, that is, conversion can be applied directly within a subband by (28). In contrast, calculating a reduced set of DFT coefficients with the plain scheme implies to calculate the complete IMDCT. Second, (28) is straightforward

[5]This is why the KBD window is used more often in AAC coding.

TABLE I
RUNNING TIME OF THE DIRECT MDCT-TO-DFT METHOD AND THE PLAIN METHOD (IMDCT + DFT), C IMPLEMENTATION.

	direct				plain
	m_{tot}=5	m_{tot}=10	m_{tot}=15	m_{tot}=20	
$M = 1024$	1.54	2.29	3.00	4.01	3.35
$M = 2048$	1.67	2.48	3.21	3.95	3.82
$M = 4096$	1.71	2.52	3.30	3.98	4.27
$M = 8192$	1.73	2.51	3.31	4.91	4.78

Note: Running time is measured in ms for a mono audio signal sampled at 44.1 kHz of 1 second long (CPU 1.83 GHz).

TABLE II
SNRS AND MOS OF THE PHASE VOCODER WITH THE APPROXIMATE MDCT-TO-DFT CONVERSION AGAINST THE PHASE VOCODER WITH THE EXACT MDCT-TO-DFT CONVERSION.

	m_{tot}=1	m_{tot}=5	m_{tot}=10	m_{tot}=15	m_{tot}=20
SNR ($r = 0.8$)	2.26	4.50	11.94	18.02	20.68
SNR ($r = 0.6$)	0.70	1.95	8.17	15.18	15.59
MOS ($r = 0.8$)	2.96	3.16	3.92	4.16	4.31
MOS ($r = 0.6$)	2.67	2.78	3.63	4.08	4.01

Note: Time scaling factor is denoted by r and $r < 1$ is for slowing down; SNR has the unit of dB; MOS is from 1 to 5 for bad to excellent quality.

to implement in Matlab, C, or assembly. Unlike the fast IMDCT and the FFT, it does not involve any bit-reversed addressing or complicated data flow control, and can be very efficient on systems supporting vector operations, for example, most modern DSPs.

We have implemented the direct method on Matlab using vector operations. It runs about 60X real-time on a 3.0 GHz CPU for 44.1 kHz mono signals with $m_{\text{tot}} = 20$ and $M = 1024$. We have also implemented both the direct method and the plain method in C. (The Matlab and C implementations are available at http://www.gipsa-lab.grenoble-inp.fr/~laurent.girin/demo/). Running speeds of both methods with different M and different m_{tot} are shown in Tab. I. Generally, the larger M or the smaller m_{tot}, the faster the direct method relatively, and the break-even m_{tot} is about 20, close to the above-mentioned value. It should be noted that the fast IMDCT and the FFT in the plain method use the renowned FFTW3 library (http://www.fftw.org) which is highly optimized, although the direct method is implemented without using any optimized vector library.

E. Example of application – Phase vocoder

As a straightforward example of application, we have combined our direct conversion method in Matlab with D. Ellis's vocoder [20]. A phase vocoder typically works in the DFT domain and speeds up or slows down audio signals by interpolating the amplitude and phase of DFT coefficients [21], [22]. With the direct MDCT-to-DFT method, we can construct a phase vocoder that accepts MDCT coefficients as input. This is applicable to audio signals compressed by MDCT-domain audio coders. For instance, the inputs are MDCT coefficients decoded from AAC bitstreams encoded at 32 kbps for a mono speech signal sampled at 16 kHz. The performance are tested in terms of SNR (against exact MDCT-to-DFT conversion, i.e., log power ratio between reconstructed signal and difference between reconstructed signals with the two methods) and Mean Opinion Score (MOS, given by the PESQ evaluation software [23], references are time-scaled signals with the exact conversion). In Tab. II, results with two time scaling factors ($r = 0.8, 0.6$, slowing down) and four different approximation orders ($m_{\text{tot}} = 1, 5, 15, 20$) are given. It is found that the SNRs are noticeably lower than the SNRs of the direct conversion itself (i.e., without time scaling) with the same approximation order. This is because the phase vocoder accumulates phase and thus accumulates phase error, and reconstruction SNRs are very sensitive to this problem. However, the quality in terms of MOS given by PESQ is quite good for $m_{\text{tot}} > 10$ (larger than 4, which means the difference between the approximate and exact methods is perceptually insignificant), and it remains fair (MOS ≈ 3) even for the extreme case $m_{\text{tot}} = 1$.

VI. CONCLUSION

In this paper, we have proposed a method for converting MDCT coefficients to DFT coefficients through conjugate symmetric FIR filtering, which can be effectively approximated by retaining only the few first (most significant) taps. This method is based on the observation that three MDCT-to-DFT conversion matrices are involved in this process, and that each of those three matrices can be separated into a Toeplitz matrix and a Hankel matrix, which can be combined into an extended Toeplitz matrix due to the $1/2$ frequency shift term in the MDCT. Also, we exploited the fact that the coefficients of the extended Toeplitz matrix are DFT coefficients of window functions product, resulting in an equivalent FIR filtering process with sharp concentration of energy at low-frequency taps for usual DFT and MDCT window functions.

Beyond the presented study, the low order FIR filtering for MDCT-to-DFT conversion reveals an intrinsic relationship between the MDCT and the DFT: the energy of an MDCT coefficient is projected locally, that is mostly to a few DFT coefficients of near frequencies. Therefore, using the presented conversion technique, it is possible to accurately estimate amplitude, phase or group delay of a signal in a subband using local MDCT coefficients and very few local computations (instead of relying on whole spectra IMDCT synthesis and DFT analysis). In a general manner, the proposed method allows direct chaining of MDCT-based perceptual audio decoders (e.g., AAC) and DFT-domain processing. As a straightforward example of application, we have plugged our conversion method between an AAC decoder and a phase vocoder. Another application of this method is for MDCT quantization in perceptual audio coders: the method may allow accurate and efficient control of quantization noise in the MDCT domain according to DFT domain psychoacoustic criteria. Finally, we plan to plug the presented method within the Informed Source Separation (ISS) system of [24], which is based on a DFT-domain Wiener filtering of source signals from mixture signals, to adapt it efficiently to AAC compressed mixture signals: source separation and generation of remix signal will then be possible without passing through mix signal time-domain reconstruction.

APPENDIX A

Note that $\mathrm{Re}\{W_{2M}[\alpha]\} = \frac{1}{2}\{W_{2M}[-\alpha] + W_{2M}[\alpha]\}$. Then, the entry of $\mathbf{T} = \mathbf{U}'\mathbf{V}$ at (k,l), by definition of matrix multiplication and (6), is

$$\begin{aligned}
\mathbf{T}(k,l) &= \sum_{n=0}^{2M-1} (\mathbf{U}(n,k))^* \mathbf{V}(n,l) \\
&= \frac{1}{2} \sum_{n=0}^{2M-1} w_0(n) w_1(n) W_{2M}[(n+n_0)k] \\
&\quad \times \left\{ W_{2M}[(n+n_1)(-l-\tfrac{1}{2})] + W_{2M}[(n+n_1)(l+\tfrac{1}{2})] \right\} \\
&= W_{2M}[(n_0-n_1)k] \frac{1}{2} \sum_{n=0}^{2M-1} w_0(n) w_1(n) W_{2M}[(n+n_1)k] \\
&\quad \times \left\{ W_{2M}[(n+n_1)(-l-\tfrac{1}{2})] + W_{2M}[(n+n_1)(l+\tfrac{1}{2})] \right\} \\
&= \phi(k) \Big\{ \frac{1}{2} \sum_{n=0}^{2M-1} w_0(n) w_1(n) W_{2M}[(n+n_1)(k-l-1+\tfrac{1}{2})] \\
&\qquad + \frac{1}{2} \sum_{n=0}^{2M-1} w_0(n) w_1(n) W_{2M}[(n+n_1)(k+l+\tfrac{1}{2})] \Big\} \\
&= \phi(k)[h(k-l-1) + h(k+l)],
\end{aligned}$$

as is stated in (7), (8), and (9).

APPENDIX B

To show that the extended vector $\widehat{\boldsymbol{X}}$ given by (18) indeed satisfies (17), we compare the kth element in $\widehat{\mathbf{T}}\widehat{\boldsymbol{X}}$ and the kth element in $(\mathbf{T}_{\mathrm{toep}} + \mathbf{T}_{\mathrm{hank}})\boldsymbol{X}$.

By (16) and (18), we have

$$\begin{aligned}
&\sum_{l=-M}^{2M-1} \widehat{\mathbf{T}}(k,l) \widehat{\boldsymbol{X}}(l) \\
&= \sum_{l=0}^{M-1} \widehat{\mathbf{T}}(k,l) \widehat{\boldsymbol{X}}(l) + \sum_{l=-M}^{-1} \widehat{\mathbf{T}}(k,l) \widehat{\boldsymbol{X}}(l) + \sum_{l=M}^{2M-1} \widehat{\mathbf{T}}(k,l) \widehat{\boldsymbol{X}}(l) \\
&= \sum_{l=0}^{M-1} \mathbf{T}_{\mathrm{toep}}(k,l) \boldsymbol{X}(l) \\
&\quad + \sum_{l=-M+k}^{-1} \mathbf{T}_{\mathrm{hank}}(k,-l-1) \boldsymbol{X}(-l-1) \\
&\quad + \sum_{l=M}^{M+k-1} \mu^* \mathbf{T}_{\mathrm{hank}}(k,2M-l-1) \mu \boldsymbol{X}(2M-l-1) \\
&= \sum_{l=0}^{M-1} \mathbf{T}_{\mathrm{toep}}(k,l) \boldsymbol{X}(l) \\
&\quad + \sum_{l=0}^{M-k-1} \mathbf{T}_{\mathrm{hank}}(k,l) \boldsymbol{X}(l) + \sum_{l=M-k}^{2M-1} \mathbf{T}_{\mathrm{hank}}(k,l) \boldsymbol{X}(l) \\
&= \sum_{l=0}^{M-1} \mathbf{T}_{\mathrm{toep}}(k,l) \boldsymbol{X}(l) + \sum_{l=0}^{M-1} \mathbf{T}_{\mathrm{hank}}(k,l) \boldsymbol{X}(l).
\end{aligned}$$

Therefore, the extended vector $\widehat{\boldsymbol{X}}$ given by (18) satisfies (17). (Note that the phase term verifies $\mu^*\mu \equiv 1$.)

APPENDIX C

From (1), we write explicitly the matrices $\mathbf{F}_1$ and $\mathbf{P}_1$ as

$$\begin{aligned}
\mathbf{F}_1(n,k) &= w_f(n+M) W_{2M}[-(n+M)k], \\
\mathbf{P}_1(n,l) &= C w_c(n+M) \mathrm{Re}\big\{ W_{2M}[(n+\tfrac{1}{2}+\tfrac{3M}{2})(l+\tfrac{1}{2})] \big\},
\end{aligned}$$

which implies their time shifts are M and $\frac{1}{2}+\frac{3M}{2}$, respectively. Obviously, both $\mathbf{F}_0$ and $\mathbf{F}$ have time shift 0 while both $\mathbf{P}_0$ and $\mathbf{P}$ have time shift $\frac{1}{2}+\frac{M}{2}$. Then replace n_0 with the time shifts of $\mathbf{F}_0, \mathbf{F}, \mathbf{F}_1$ and replace n_1 with time shifts of $\mathbf{P}_1, \mathbf{P}, \mathbf{P}_0$ respectively in (8), we have

$$\begin{aligned}
\phi_{01}(k) &= W_{2M}[(0-(\tfrac{1}{2}+\tfrac{3M}{2}))k] = W_{2M}[-\tfrac{1}{2}(1-M)k], \\
\phi_{12}(k) &= W_{2M}[(0-(\tfrac{1}{2}+\tfrac{M}{2}))k] = W_{2M}[-\tfrac{1}{2}(1+M)k], \\
\phi_{23}(k) &= W_{2M}[(M-(\tfrac{1}{2}+\tfrac{M}{2}))k] = W_{2M}[-\tfrac{1}{2}(1-M)k].
\end{aligned}$$

Note that $W_{2M}[\alpha + Mk] = (-1)^k W_{2M}[\alpha]$. To compute the filter taps h_{01}, h_{12}, h_{23} by (9), we have to replace w_0 with the window functions for $\mathbf{F}_0$, $\mathbf{F}$, and $\mathbf{F}_1$, and then replace w_1 with the window functions for $\mathbf{P}_1$, $\mathbf{P}$, and $\mathbf{P}_0$:

$$\begin{aligned}
&w_0(n) = w_f(n-M), \quad && w_1(n) = w_c(n), \quad && \text{for } h_{01}(l), \\
&w_0(n) = w_f(n), && w_1(n) = w_c(n), && \text{for } h_{12}(l), \\
&w_0(n) = w_f(n+M), && w_1(n) = w_c(n), && \text{for } h_{23}(l).
\end{aligned}$$

Note that n ranges from M to $2M-1$ for $h_{01}(l)$, from 0 to $2M-1$ for $h_{12}(l)$, and from 0 to $M-1$ for $h_{23}(l)$. Then (22) follows readily from the above.

For the DFT form (24), the modulation term is $W_{2M}[f(n,l)] = W_{2M}[(n+\frac{1}{2}+\frac{M}{2})(l+\frac{1}{2})]$ instead of the normal $W_{2M}[nl]$. But we have

$$\begin{aligned}
&W_{2M}[(n+\tfrac{1}{2}+\tfrac{M}{2})(l+\tfrac{1}{2})] \\
&\quad = W_{2M}[(\tfrac{1}{2}+\tfrac{M}{2})l] W_{2M}[nl] W_{2M}[\tfrac{1}{2}(n+\tfrac{1}{2}+\tfrac{M}{2})] \\
&\quad = \psi_{\mathrm{post}}(l) W_{2M}[nl] \psi_{\mathrm{pre}}(n),
\end{aligned}$$

which implies (24). The modulation term also has a special symmetry due to its phase:

$$\begin{aligned}
&f(n,l) + f(2M-1-n,l) = 3M(l+\tfrac{1}{2}) \\
&\Rightarrow W_{2M}[f(n,l)] \pm W_{2M}[f(2M-1-n,l)] \\
&= (1 \pm (-1)^l j)(\cos[f(n,l)] \pm \sin[f(n,l)]).
\end{aligned}$$

Now consider $w_f(n)$ and $w_c(n)$ are even symmetric (and real). Then window function products $w_f(n)w_c(n)$ and $w_f^+(n)w_c(n)$ are even symmetric too and $w_f^-(n)w_c(n)$ is odd symmetric, which, with the above symmetry of the modulation term, lead to (26).

APPENDIX D

Let us assume that in (28), coefficients $\widehat{\boldsymbol{X}}_0(l)$ are zero-mean uncorrelated white noises with the same variance equal to 1. Then, the variance of the error of approximating $\boldsymbol{Y}_0(k)$

(defined in (19), but with subscript $_0$) in (28) is

$$\begin{aligned}\mathrm{E}[|\delta \boldsymbol{Y}_0(k)|^2] =& \sum_{l=m}^{M-1} \Big\{ |h_0(l)|^2 \mathrm{E}[|\widehat{\boldsymbol{X}}_0(k-l-1)|^2] \\ &+ |h_0^*(l)|^2 \mathrm{E}[|\widehat{\boldsymbol{X}}_0(k+l)|^2] \Big\} \\ =& \sum_{l=m}^{M-1} \Big\{ |h_0(l)|^2 + |h_0^*(l)|^2 \Big\} \\ =& 2 \sum_{l=m}^{M-1} |h_0(l)|^2. \end{aligned} \tag{34}$$

And similarly, the variance of the exact filter output $\boldsymbol{Y}_0(k)$ is

$$\mathrm{E}[|\boldsymbol{Y}_0(k)|^2] = 2 \sum_{l=0}^{M-1} |h_0(l)|^2. \tag{35}$$

Similar results can be drawn for $\delta \boldsymbol{Y}_\pm(k)$ and $\boldsymbol{Y}_\pm(k)$. Let us further assume that $\widehat{\boldsymbol{X}}_0(l), \widehat{\boldsymbol{X}}_-(l), \widehat{\boldsymbol{X}}_+(l)$ in (28) are independent with each other (in addition to being each one a zero-mean 1-variance white noise). The function $\sigma(a,b,c)$ defined in (30) is the total energy of the first a, b, c taps of the three filters, respectively. Then, by (34), the variance of total approximation error of DFT coefficient $\boldsymbol{Z}_{12}(k)$ using (28) is

$$\mathrm{E}[|\delta \boldsymbol{Z}_{12}(k)|^2] = 2\sigma(M,M,M) - 2\sigma(m_0, m_-, m_+), \tag{36}$$

and by (35), the variance of the DFT coefficient $\boldsymbol{Z}_{12}(k)$ is

$$\mathrm{E}[|\boldsymbol{Z}_{12}(k)|^2] = 2\sigma(M,M,M). \tag{37}$$

Therefore, the approximation accuracy in term of log-SNR can be estimated as

$$\begin{aligned}\mathrm{SNR} \approx& 10 \log_{10} \frac{\mathrm{E}[|\boldsymbol{Z}_{12}(k)|^2]}{\mathrm{E}[|\delta \boldsymbol{Z}_{12}(k)|^2]} \\ \approx& 10 \log_{10} \frac{1}{1 - \sigma(m_0, m_+, m_-)/\sigma(M,M,M)}. \end{aligned} \tag{38}$$

Note that the white noise and independence assumptions of $\widehat{\boldsymbol{X}}_0(l)$, $\widehat{\boldsymbol{X}}_-(l)$, and $\widehat{\boldsymbol{X}}_+$ may not be valid in reality. Nevertheless, (29) still gives very close estimation of the approximation accuracy even for musical signals, as shown by experiments, illustrated in Fig. 7, and commented in Section V-B.

REFERENCES

[1] J. Princen and A. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 5, pp. 1153–1161, Oct. 1986.

[2] ISO/IEC JTC1/SC29/WG11 MPEG, "Coding of moving pictures and audio, part 7: Advanced Audio Coding," Tech. Rep. ISO/IEC 13818-7, 2005.

[3] H. Malvar, *Signal processing with lapped transforms*. Norwood, USA: Artech House, 1992.

[4] Y. Wang, L. Yaroslavsky, M. Vilermo, and M. Vaananen, "Some peculiar properties of the MDCT," in *Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on*, vol. 1, Aug. 2000, pp. 61–64.

[5] S. Zhang, W. Dou, P. Chi, and H. Yang, "MDCT spectrum separation: Catching the fine spectral structures for stereo coding," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, Mar. 2010, pp. 369–372.

[6] S. Zhang, W. Dou, and H. Yang, "DFT spectrum estimation from critically sampled lapped transforms," *Signal Processing*, vol. 91, no. 2, pp. 300–310, Feb. 2011.

[7] K. Suresh and T. Sreenivas, "Linear filtering in DCT IV/DST IV and MDCT/MDST domain," *Signal Processing*, vol. 89, no. 6, pp. 1081 – 1089, June 2009.

[8] S. Merdjani and L. Daudet, "Direct estimation of frequency from MDCT-encoded files," in *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx-03). DAFx-03*, Sept. 2003, pp. 1–4.

[9] P. Duhamel, Y. Mahieux, and J. Petit, "A fast algorithm for the implementation of filter banks based on 'time domain aliasing cancellation'," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, vol. 3, apr 1991, pp. 2209–2212.

[10] P. Duhamel, "Implementation of "split-radix" FFT algorithms for complex, real, and real-symmetric data," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 34, no. 2, pp. 285–295, Apr. 1986.

[11] M. Goodwin, "Efficient system and method for converting between different transform-domain signal representations," Sept. 2001, US Patent App. 09/948,053.

[12] C. Cheng, "Method for estimating magnitude and phase in the MDCT domain," in *116th AES Convention*. Audio Engineering Socitey, May 2004, p. Paper Number: 6091.

[13] H. Malvar, "A modulated complex lapped transform and its applications to audio processing," in *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*, vol. 3, Mar 1999, pp. 1421–1424.

[14] S. Chen, N. Xiong, J. Hyuk Park, M. Chen, and R. Hu, "Spatial parameters for audio coding: MDCT domain analysis and synthesis," *Multimedia Tools and Applications*, vol. 48, no. 2, pp. 225–246, 2010.

[15] M. Bosi and R. E. Goldberg, *Introduction to digital audio coding and standards*. Norwell, USA: Kluwer Academic Publishers, 2003.

[16] R. Gray, *Toeplitz and circulant matrices: A review*. Hanover, USA: Now Publishers Inc., 2006.

[17] L. Daudet and M. Sandler, "MDCT analysis of sinusoids: exact results and applications to coding artifacts reduction," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 3, pp. 302–312, May 2004.

[18] P. Vaidyanathan, *Multirate systems and filter banks*. Englewood Cliffs, USA: Prentice-Hall Inc., 1993.

[19] F. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, Jan. 1978.

[20] D. P. W. Ellis, "A phase vocoder in Matlab," 2002, web resource. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/pvoc/

[21] M. Portnoff, "Implementation of the digital phase vocoder using the fast Fourier transform," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 3, pp. 243–248, Jun. 1976.

[22] M. Dolson, "The phase vocoder: A tutorial," *Computer Music Journal*, vol. 10, no. 4, pp. 14–27, 1986. [Online]. Available: http://www.panix.com/~jens/pvoc-dolson.par

[23] ITU-T Rec. P.862, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Tech. Rep. ITU-T Rec. P.862, 2001.

[24] A. Liutkus, J. P. amd Roland Badeau, L. Girin, and G. Richard, "Informed source separation through spectrogram coding and data embedding," *Signal Processing*, vol. 19, no. 10, pp. 1–13, 2011.