R. Bregović, Y. J. Yu, A. Viholainen, and Y. C. Lim, "Implementation of linear-phase FIR nearly perfect-reconstruction cosine-modulated filterbanks utilizing the coefficient symmetry," *IEEE Trans. Circuits Syst. I*, vol. 57, pp. 139–151, Jan. 2010.

Implementation of Linear-Phase FIR Nearly Perfect-Reconstruction Cosine-Modulated Filterbanks Utilizing the Coefficient Symmetry

Robert Bregović, *Member, IEEE*, Ya Jun Yu, *Member, IEEE*, Ari Viholainen, *Member, IEEE*, and Yong Ching Lim, *Fellow, IEEE*

***Abstract*—Analysis and synthesis part of a cosine-modulated *M*-channel filterbank (FB) contains two sections, a modulation block and a prototype filter implemented in a polyphase structure. Although, in many cases a linear-phase prototype filter is used, the coefficient symmetry of this filter is not utilized when using the existing polyphase structure. In this paper a method is proposed for implementing a linear-phase prototype filter building a nearly perfect-reconstruction cosine-modulated FB in such a way that it enables one to partially utilize the coefficient symmetry, thereby reducing the number of required multiplications in the implementation. The proposed method can be applied for implementing FBs with an arbitrary filter order and number of channels. Moreover, it is shown that in all cases under consideration, the cosine-modulation part of the FB can be implemented by using a fast discrete cosine transform. The efficiency of the proposed implementation is evaluated by means of examples.**

***Index terms*—Multirate system, cosine-modulated filterbank, nearly perfect-reconstruction, FIR filter, linear-phase, fast DCT**

I. INTRODUCTION

During the last three decades, multirate systems have been used in various applications, e.g., adaptive signal processing, compression, denoising, data transmission [1]–[3]. One of the basic building blocks of a multirate system is a uniform *M*-channel critically-sampled filterbank (FB) shown in Figure 1. This FB consists of an analysis part (analysis FB), containing filters with the transfer functions $H_k(z)$ for $k = 0, 1, \ldots, M-1$ followed by down-samplers by *M*, and a synthesis part (synthesis FB), containing filters with transfer functions $F_k(z)$ for $k = 0, 1, \ldots, M-1$ preceded by up-samplers by *M*. Moreover, in this paper it is assumed that the processing unit does not modify the subband signals.

When synthesizing a FB, the goal is to generate a system that has either a perfect-reconstruction (PR) or a nearly perfect-reconstruction (NPR) property. In the PR case, the output signal $y[n]$ is a delayed version of the input signal $x[n]$, that is, $y[n] = x[n-D]$ with D being the FB delay. In the NPR case, this relation is only approximately satisfied, that is, $y[n] \approx x[n-D]$. By giving up on the PR property, FBs with better overall performance can be designed, for example, the same channel selectivity can be achieved by having filters of a lower order [4]. Therefore, for most applications, systems satisfying the NPR property are a better choice as long as the distortions introduced by the FB are smaller than the changes caused by the application.

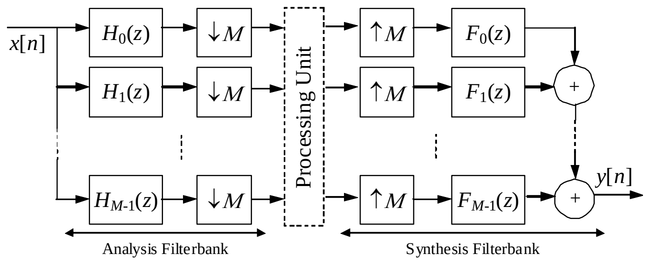


Figure 1. *M*-channel critically-sampled filterbank.

Among different types of *M*-channel FBs, the most commonly used ones are modulated FBs. In a modulated FB, the filter transfer functions $H_k(z)$ and $F_k(z)$ are generated by properly modulating one prototype filter[1]. In this case only one filter has to be designed and the implementation consists of a polyphase implementation of the prototype filter and a modulation block, thereby simplifying the design and implementation of *M*-channel FBs.

This paper considers the implementation of cosine-modulated FBs[2] with linear-phase FIR prototype filters. Cosine-modulated FBs are modulated FBs with modulation matrices based on cosine functions. In PR cosine-modulated FBs with linear-phase prototype filters of order $N = 2KM-1$, with K being an integer, the prototype filters can be efficiently

Manuscript received August 05, 2008; revised November 12, 2008. This work was supported in parts by Temasek Laboratories@NTU, Singapore, and the Academy of Finland, project No. 207019 (postdoctoral research grant) and 213462 (Finnish centre of Excellence program (2006–2011)).

R. Bregović is with the Department of Signal Processing, Tampere University of Technology, Finland. e-mail: bregovic@cs.tut.fi. Phone: +358 3 3115 3842, Fax: +358 3 3115 3817.

Y. J. Yu and Y. C. Lim are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. e-mail: eleyuyj@pmail.ntu.edu.sg and elelimyc@pmail.ntu.edu.sg.

A. Viholainen is with the Department of Communications Engineering, Tampere University of Technology, Finland. e-mail: ari.viholainen@tut.fi.

[1] In some modulated FBs different prototype filters are used for generating the analysis and synthesis FB. In this paper, without loss of generality, FBs with only one prototype filter are considered.

[2] For processing real-valued signals, cosine-modulated FBs are one of the more frequently used types of modulated FBs.

implemented by utilizing a lattice structure [1]–[3]. In such FBs the implementation of the prototype filter requires $(N+1)/2$ multiplications per M input samples. However, this implementation can not be used for NPR FBs. For NPR cosine-modulated FBs with an arbitrary N and M, the currently most efficient implementation for the prototype filter is achieved by using a polyphase structure. In this case the implementation of the prototype filter requires $N+1$ multiplications per M input samples. The drawback of this implementation is that the coefficient symmetry of a linear-phase prototype filter is not utilized, that is, the implementation complexity of the prototype filter is the same for linear-phase as well as non linear-phase filters.

It has been shown in [5] that for NPR FBs with prototype filters of order $N = 2KM-1$ the coefficient symmetry of the prototype filters can be partially utilized. However, this order selection is very restrictive due to the fact that in the case of NPR FBs, prototype filters of any order can be used compared to the PR case where only filters of order $N = 2KM-1$ result in good FBs [4] [6], [7] (this is also illustrated by means of an example in Section VI). Therefore, in this paper a method is proposed for implementing a linear-phase prototype filter of an arbitrary order in such a way that it enables one to partially utilize the coefficient symmetry, thereby reducing the number of required multiplications in the implementation. Furthermore, it is shown that in all cases under consideration the cosine modulation part of the FB can be implemented by using a fast discrete cosine transform (DCT). In order to simplify the discussion, in this paper only FBs with even number of channels and odd filter orders are considered. Similar principles presented in this paper can be applied to other cases.

The outline of this paper is as follows: Section II reviews the basic relations and properties of cosine-modulated FBs. This section also shows how the cosine-modulation part of the FB can be efficiently implemented by utilizing a fast DCT (the proof is given in the Appendix A). The implementation of the polyphase part of the FB is discussed in Section III. The proposed implementation method is given in Section IV with the expressions for evaluating the implementation complexity given in Section V. In Section VI a comparison is performed between the proposed implementation method and the polyphase one. Finally, some concluding remarks are given in Section VII.

II. Cosine-Modulated Filterbanks

This section reviews the basic properties and implementation structures for cosine-modulated FBs with synthesis and analysis filters derived by modulating one linear-phase FIR prototype filter. Moreover, the emphasis is put on FBs with even number of channels and odd filter orders. It is also shown that for all cases under consideration, the cosine-modulation part can be implemented by using a fast DCT.

For a linear-phase prototype filter of order N with the transfer function

$$H(z) = \sum_{n=0}^{N} h[n] z^{-n}\ , \tag{1}$$

where $h[N-n] = h[n]$ for $n = 0, 1, ..., N$, the impulse-response coefficients of filters with the transfer functions $H_k(z)$ and $F_k(z)$ building an M-channel cosine-modulated FB are generated by using the following modulation functions [8]–[10]:

$$h_k[n] = 2h[n]\cos\left[\left(k+\frac{1}{2}\right)\frac{\pi}{M}\left(n-\frac{D}{2}\right)+(-1)^k\frac{\pi}{4}\right] \tag{2a}$$

$$f_k[n] = 2h[n]\cos\left[\left(k+\frac{1}{2}\right)\frac{\pi}{M}\left(n-\frac{D}{2}\right)-(-1)^k\frac{\pi}{4}\right], \tag{2b}$$

for $k = 0, 1, \ldots, M-1$ and $n = 0, 1, \ldots, N$. In the above equation, D denotes the FB delay. For a cosine-modulated FB with a linear-phase prototype filter, the FB delay is equal to the filter order, that is, $D = N$. Moreover, as the emphasis in this paper is put on FBs with even number of channels and odd filter orders, the filter order will be represented as

$$N = 2K_E M + 2\Delta - 1, \tag{3a}$$

with K_E being an even integer, Δ being an integer, and $0 \le \Delta \le 2M-1$. Consequently,

$$K_E = 2\left\lfloor \frac{N+1}{4M} \right\rfloor \tag{3b}$$

and

$$\Delta = \frac{N+1}{2} - K_E M\ . \tag{3c}$$

Parameters K_E and Δ will be used later on when deriving the implementation structures.

As briefly mentioned in the introduction, cosine-modulated FBs have the following two main properties: First, instead of designing M analysis and M synthesis filters, only one prototype filter has to be designed, thereby significantly simplifying the FB design. Second, the FB can be implemented as shown in Figure 2. This implementation consists of the prototype filter implemented in its polyphase form and a cosine-modulation matrix. The polyphase terms are generated by decomposing the prototype filter given by (1) into $2M$ polyphase components as

$$H(z) = \sum_{\xi=0}^{2M-1} z^{-\xi} G_\xi(z^{2M})\,, \tag{4a}$$

where $G_\xi(z)$ for $\xi = 0, 1, \ldots, 2M-1$ is the ξth polyphase component defined as

$$G_\xi(z) = \sum_{n=0}^{N_\xi} h[\xi + 2Mn] z^{-n} \tag{4b}$$

with N_ξ being the order of the ξth polyphase component defined as

$$N_\xi = \left\lceil \frac{N+1-\xi}{2M} \right\rceil - 1\,. \tag{4c}$$

The elements of the cosine-modulation matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are given by the following equations [8]–[10]:

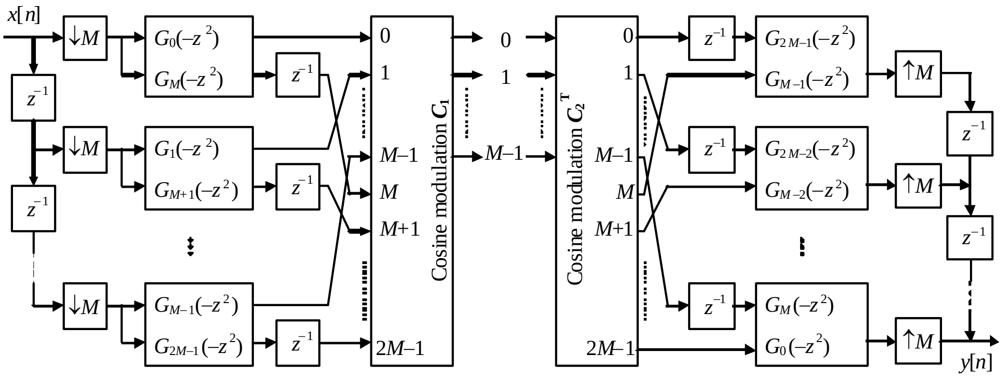


Figure 2. Polyphase structure for implementing a cosine-modulated filterbank.

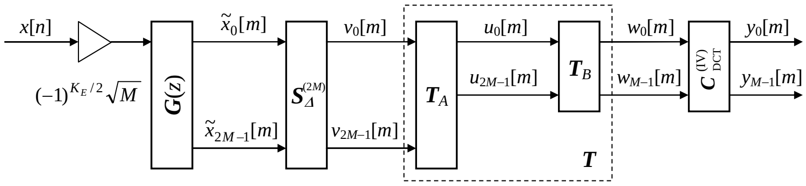


Figure 3. Efficient polyphase structure for implementing an analysis filterbank with $N = 2K_EM+2\Delta-1$.

$$[\boldsymbol{C}_1]_{kl} = 2\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(l-\frac{N}{2}\right)+(-1)^k\frac{\pi}{4}\right) \tag{5a}$$

$$[\boldsymbol{C}_2]_{kl} = 2\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(2M-1-l-\frac{N}{2}\right)-(-1)^k\frac{\pi}{4}\right) \tag{5b}$$

for $k=0, 1, \ldots, M-1$ and $l=0, 1, \ldots, 2M-1$. Due to the fact that the synthesis FB is implemented in a similar way as the analysis one, in the rest of this paper only the analysis FB is considered.

For FBs with prototype filters of order $N = 2K_EM+2\Delta-1$, with K_E and M being even integers and $0 \le \Delta \le 2M-1$, the implementation structure given in Figure 2 can be further simplified to the one shown in Figure 3. In this figure $\boldsymbol{G}(z)$ represents the polyphase implementation of the prototype filter as given in Figure 4 and the cosine-modulation matrix $\boldsymbol{C}_1$ from Figure 2 has been decomposed into four parts as

$$\boldsymbol{C}_1 = \lambda \boldsymbol{C}_{DCT}^{(IV)} \boldsymbol{T}\, \boldsymbol{S}_{\Delta}^{(2M)}. \tag{6a}$$

Here, λ is the scaling factor defined as

$$\lambda = (-1)^{K_E/2}\sqrt{M}. \tag{6b}$$

$\boldsymbol{C}_{DCT}^{(IV)}$ is the DCT-IV transform defined as [1], [11]

$$[\boldsymbol{C}_{DCT}^{(IV)}]_{k,l} = \sqrt{\frac{2}{M}}\cos\left[\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(l+\frac{1}{2}\right)\right] \tag{6c}$$

for $k, l=0, 1, \ldots, M-1$. Matrix $\boldsymbol{T}$ of size M by $2M$ is defined as

$$\boldsymbol{T} = \left[\boldsymbol{I}_M - \boldsymbol{J}_M \quad -(\boldsymbol{I}_M + \boldsymbol{J}_M)\right], \tag{6d}$$

with $\boldsymbol{I}_M$ and $\boldsymbol{J}_M$ being the identity and counter-identity matrices of size M by M. Moreover, matrix $\boldsymbol{T}$ is further decomposed as

$$\boldsymbol{T} = \boldsymbol{T}_B \boldsymbol{T}_A, \tag{6e}$$

with

$$\boldsymbol{T}_A = \begin{bmatrix} \boldsymbol{I}_{M/2} & -\boldsymbol{J}_{M/2} & \boldsymbol{0}_{M/2} & \boldsymbol{0}_{M/2} \\ \boldsymbol{0}_{M/2} & \boldsymbol{0}_{M/2} & \boldsymbol{J}_{M/2} & \boldsymbol{I}_{M/2} \end{bmatrix} \tag{6f}$$

$$\boldsymbol{T}_B = \begin{bmatrix} \boldsymbol{I}_{M/2} & -\boldsymbol{J}_{M/2} \\ -\boldsymbol{J}_{M/2} & -\boldsymbol{I}_{M/2} \end{bmatrix} \tag{6g}$$

and $\boldsymbol{0}_{M/2}$ being a zero matrix of size $M/2$ by $M/2$. Finally, matrix $\boldsymbol{S}_{\Delta}^{(2M)}$ of size $2M$ by $2M$ has the following form:

$$\boldsymbol{S}_{\Delta}^{(2M)} = \begin{bmatrix} \boldsymbol{0}_{2M-\Delta,\Delta} & \boldsymbol{I}_{2M-\Delta} \\ -\boldsymbol{I}_{\Delta} & \boldsymbol{0}_{\Delta,2M-\Delta} \end{bmatrix}, \tag{6h}$$

with $\boldsymbol{0}_{2M-\Delta,\Delta}$ being a zero matrix of size $2M-\Delta$ by Δ and Δ being defined by (3c). The proof for (6a)–(6h) can be found in Appendix A.

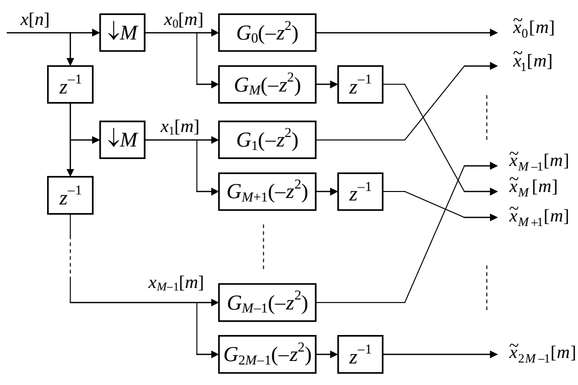


Figure 4. Polyphase implementation of the prototype filter.

TABLE I EQUATIONS DEFINING QUADRUPLETS USED FOR IMPLEMENTING THE FILTERBANK

Δ	Quadruplets	
$0 \le \Delta \le \frac{M}{2}-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} G_{\mu+\Delta}(-z^2) & -G_{M-\mu+\Delta-1}(-z^2) \\ z^{-1}G_{\mu+\Delta+M}(-z^2) & z^{-1}G_{2M-\mu+\Delta-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta}(z) \\ X_{M-\mu+\Delta-1}(z) \end{bmatrix}$ for $\mu = \Delta, \Delta+1, ..., \frac{M}{2}-1$	(9a)
$1 \le \Delta \le M-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} G_{\mu+\Delta}(-z^2) & -z^{-1}G_{M-\mu+\Delta-1}(-z^2) \\ z^{-1}G_{\mu+\Delta+M}(-z^2) & -G_{\Delta-\mu-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta}(z) \\ X_{\Delta-\mu-1}(z) \end{bmatrix}$ for $\mu = 0,1,..., \frac{M}{2}-1-\left\lvert \Delta - \frac{M}{2} \right\rvert$	(9b)
$\frac{M}{2}+1 \le \Delta \le \frac{3M}{2}-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} z^{-1}G_{\mu+\Delta}(-z^2) & -z^{-1}G_{M-\mu+\Delta-1}(-z^2) \\ -G_{\mu+\Delta-M}(-z^2) & -G_{\Delta-\mu-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta-M}(z) \\ X_{\Delta-\mu-1}(z) \end{bmatrix}$ for $\mu = \lvert \Delta - M \rvert, \lvert \Delta - M \rvert + 1, ..., \frac{M}{2}-1$	(9c)
$M+1 \le \Delta \le 2M-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} z^{-1}G_{\mu+\Delta}(-z^2) & G_{\Delta-\mu-M-1}(-z^2) \\ -G_{\mu+\Delta-M}(-z^2) & -z^{-1}G_{\Delta-\mu-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta-M}(z) \\ X_{\Delta-\mu-1}(z) \end{bmatrix}$ for $\mu = 0,1,..., \frac{M}{2}-1-\left\lvert \Delta - \frac{3M}{2} \right\rvert$	(9d)
$\frac{3M}{2}+1 \le \Delta \le 2M-1$	$\begin{bmatrix} U_\mu(z) \\ U_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} -G_{\mu+\Delta-2M}(-z^2) & G_{\Delta-\mu-M-1}(-z^2) \\ -z^{-1}G_{\mu+\Delta-M}(-z^2) & -z^{-1}G_{\Delta-\mu-1}(-z^2) \end{bmatrix} \begin{bmatrix} X_{\mu+\Delta-2M}(z) \\ X_{\Delta-\mu-M-1}(z) \end{bmatrix}$ for $\mu = 2M-\Delta, 2M-\Delta+1, ..., \frac{M}{2}-1$	(9e)

Based on (6h), it can be seen that matrix $\boldsymbol{S}_\Delta^{(2M)}$ is only a cross-connection network that connects outputs of the polyphase filters with the inputs of matrix $\boldsymbol{T}$ on one-to-one basis. Therefore, it is straightforward to implement matrix $\boldsymbol{S}_\Delta^{(2M)}$ as shown in Figure 5. Consequently, the relation between $\tilde{x}_k[m]$ and $v_k[m]$ is given by

$$v_k[m] = \begin{cases} \tilde{x}_{k+\Delta}[m] & \text{for } k = 0,...,2M-\Delta-1 \\ -\tilde{x}_{k+\Delta-2M}[m] & \text{for } k = 2M-\Delta,...,2M-1. \end{cases} \quad (7)$$

For FBs with $N = 2K_E M - 1$ with K_E and M being even integers, $v_k[m] = \tilde{x}_k[m]$ for $k = 0, 1, \dots, 2M-1$, that is, $\boldsymbol{S}_\Delta^{(2M)} = \boldsymbol{I}$.

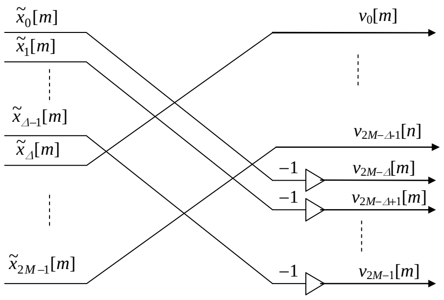


Figure 5. Implementation of the cross-connection matrix $\boldsymbol{S}_\Delta^{(2M)}$.

The matrix $\boldsymbol{T}$ given by (6d) combines outputs of four polyphase filters by adding them or subtracting them in order to build two inputs to the DCT-IV. As there are $2M$ polyphase filters, there are altogether $M/2$ such sets. Each of these sets is defined by the following equation:

$$\begin{bmatrix} w_\mu[m] \\ w_{M-\mu-1}[m] \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} u_\mu[m] \\ u_{M-\mu-1}[m] \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_\mu[m] \\ v_{M-\mu-1}[m] \\ v_{\mu+M}[m] \\ v_{2M-\mu-1}[m] \end{bmatrix} \quad (8)$$

for $\mu = 0, 1, \dots, M/2-1$. The implementation of (8) is given in Figure 6.

There are four important observations related to the modulation part that can be seen from (6a)–(6h), as well as Figure 6. First, matrix $\boldsymbol{T}$ (as well as $\boldsymbol{T}_A$ and $\boldsymbol{T}_B$) contains only additions (subtractions) and as such can be implemented straightforwardly. Second, the DCT-IV transform given by (6c) is of size M by M. This is half the original modulation matrix of size $2M$ by M. Third, matrices $\boldsymbol{T}$ and $\boldsymbol{C}_{DCT}^{(IV)}$ depend only on the number of channels and do not depend on the filter order and consequently FB delay. Therefore, when deriving an efficient implementation for the FB, the part related to the DCT-IV does not have to be considered as it is identical for all FBs having the same number of channels. Fourth, the DCT-IV is a well known transform that can be efficiently implemented by using a fast DCT [11], [12].

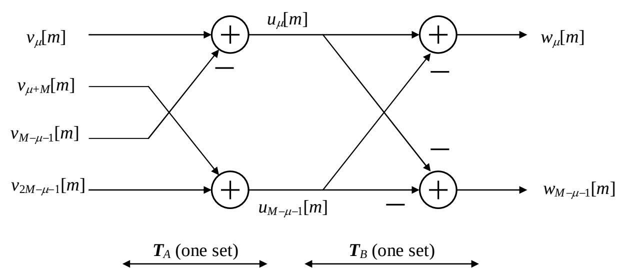


Figure 6. Implementation for one set of matrix $\boldsymbol{T}$.

III. Implementation of the Polyphase Section of the Filterbank

By combining the equations given in the previous section, for FBs with even number of channels M and odd filter orders N, the relations between two inputs of the modulation matrix $\boldsymbol{C}_{DCT}^{(IV)}$ and two inputs of the corresponding four polyphase components, can be expressed in the z domain, depending on Δ, by the equations given in Table I,[3] with

$$\begin{bmatrix} W_{\mu}(z) \\ W_{M-\mu-1}(z) \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} U_{\mu}(z) \\ U_{M-\mu-1}(z) \end{bmatrix} \qquad (10)$$

and Δ given by (3c). Here, $X(z)$, $U(z)$, and $W(z)$ are the z-transforms of $x[m]$, $u[m]$, and $w[m]$, respectively. In the rest of the paper, each of these sets of four polyphase components is referred to as a quadruplet. It should be pointed out that in a FB, quadruplets from one or two of the above sets have to be used. This is illustrated in Table II.

TABLE II SELECTION OF EQUATIONS (9A)–(9E) DEPENDING ON THE VALUE OF Δ.

Δ	(9a)	(9b)	(9c)	(9d)	(9e)
$\Delta=0$	+				
$1\le\Delta\le M/2-1$	+	+			
$\Delta=M/2$		+			
$M/2+1\le\Delta\le M-1$		+	+		
$\Delta=M$			+		
$M+1\le\Delta\le 3M/2-1$			+	+	
$\Delta=3M/2$				+	
$3M/2+1\le\Delta\le 2M-1$				+	+

In order to illustrate the relations (9a)–(9e), as an example, the implementation of a FB with $M=8$ and $N=33$ is considered. In this case, according to (3b) and (3c), $\Delta=1$ and $K_E=2$. According to (9a)–(9e), there are the following four sets of equations:

$$\Delta=1, \mu=0 \rightarrow \text{(9b)} \rightarrow$$
$$\rightarrow \begin{bmatrix} U_0(z) \\ U_7(z) \end{bmatrix} = \begin{bmatrix} G_1(-z^2) & -z^{-1}G_8(-z^2) \\ z^{-1}G_9(-z^2) & -G_0(-z^2) \end{bmatrix} \begin{bmatrix} X_1(z) \\ X_0(z) \end{bmatrix} \qquad (11a)$$

$$\Delta=1, \mu=1 \rightarrow \text{(9a)} \rightarrow$$
$$\rightarrow \begin{bmatrix} U_1(z) \\ U_6(z) \end{bmatrix} = \begin{bmatrix} G_2(-z^2) & -G_7(-z^2) \\ z^{-1}G_{10}(-z^2) & z^{-1}G_{15}(-z^2) \end{bmatrix} \begin{bmatrix} X_2(z) \\ X_7(z) \end{bmatrix} \qquad (11b)$$

$$\Delta=1, \mu=2 \rightarrow \text{(9a)} \rightarrow$$
$$\rightarrow \begin{bmatrix} U_2(z) \\ U_5(z) \end{bmatrix} = \begin{bmatrix} G_3(-z^2) & -G_6(-z^2) \\ z^{-1}G_{11}(-z^2) & z^{-1}G_{14}(-z^2) \end{bmatrix} \begin{bmatrix} X_3(z) \\ X_6(z) \end{bmatrix} \qquad (11c)$$

$$\Delta=1, \mu=3 \rightarrow \text{(9a)} \rightarrow$$
$$\rightarrow \begin{bmatrix} U_3(z) \\ U_4(z) \end{bmatrix} = \begin{bmatrix} G_4(-z^2) & -G_5(-z^2) \\ z^{-1}G_{12}(-z^2) & z^{-1}G_{13}(-z^2) \end{bmatrix} \begin{bmatrix} X_4(z) \\ X_5(z) \end{bmatrix}. \qquad (11d)$$

The implementations for one quadruplet given by (9a), (9b), (9c), (9d), and (9e) are shown in Figures 7(a), 7(b), 7(c), 7(d), and 7(e), respectively. As every FB under consideration can be decomposed by using these relations, the goal now is to derive efficient implementations for these five types of quadruplets. This is shown in the next section.

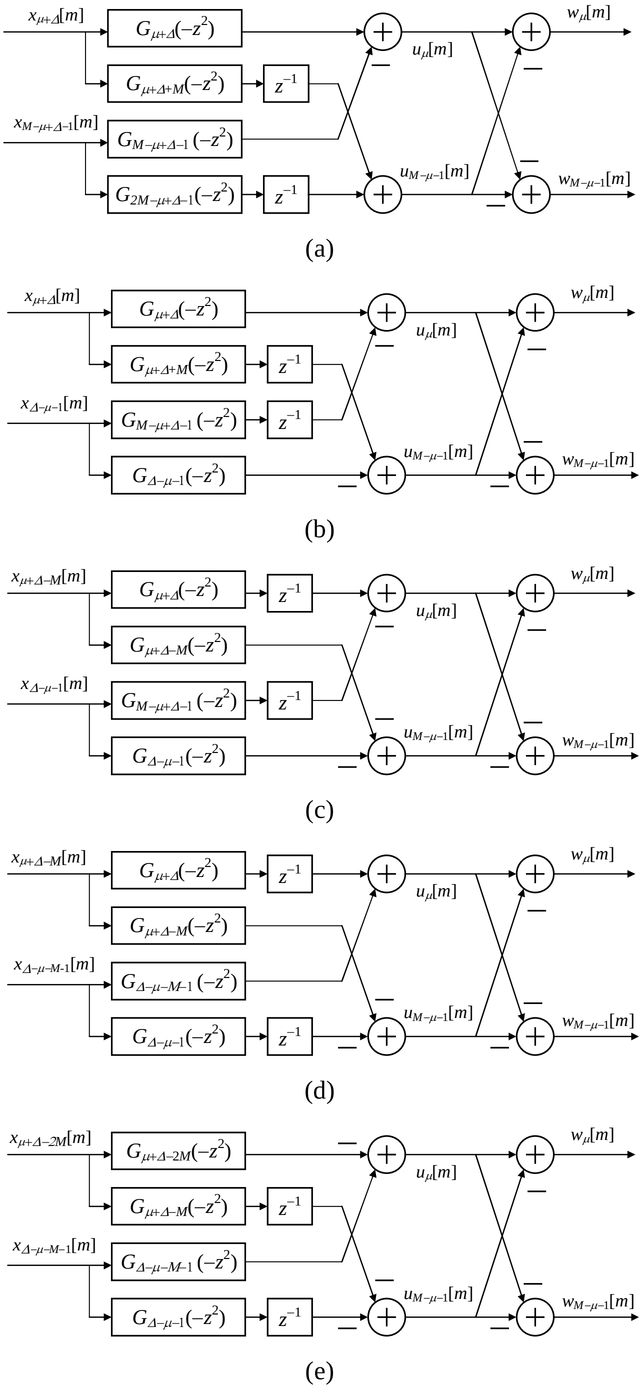


Figure 7. Implementation structure for quadruplets given by relations (9a)–(9e).

[3] It should be pointed out that the relations given by (9a)–(9e) are somehow similar to the ones derived in [9] for describing the PR requirements. However, the ones introduced here are more useful for deriving an efficient implementation because all signals are organized in such a way that they can be directly used as inputs into a DCT-IV. No further rearrangements are needed.

IV. PROPOSED IMPLEMENTATION METHOD

In order to derive an efficient implementation for the five quadruplets given in the previous section, and consequently derive an efficient implementation for the FBs under consideration, it should be observed from Figures 7(a)–7(e) that there are, in principle, only two different quadruplets to be implemented. Namely, the quadruplets shown in Figures 7(a), 7(c), and 7(e) are from the implementation point similar[4]. The same applies for the ones shown in Figures 7(b) and 7(d). Therefore, in this paper, only the implementation for quadruplets given in Figures 7(a) and 7(b) is considered.

In order to simplify the notations, in Figure 8,[5] the two quadruplets under consideration are depicted again with the transfer functions $G_\chi(-z^2)$ from Figures 7(a)–7(e) replaced by generic transfer functions $A(-z^2)$, $B(-z^2)$, $C(-z^2)$, and $D(-z^2)$ defined as:

$$A(z)=\sum_{n=0}^{R_0} a_n z^{-n} \tag{12a}$$

$$B(z)=\sum_{n=0}^{R_1} b_n z^{-n} \tag{12b}$$

$$C(z)=\sum_{n=0}^{R_1} c_n z^{-n} \tag{12c}$$

$$D(z)=\sum_{n=0}^{R_0} d_n z^{-n} \tag{12d}$$

with R_0 and R_1 being the filter orders of these generic transfer functions. These orders are calculated according to (4c) and are related to the parameter K_E, based on Δ, as shown in Table III. Parameter R given in the table will be used later on. The relation between orders R_0 and R_1 and parameters K_E and Δ could also be expressed by using equations but the table is more self-explanatory. Moreover, due to the linear-phase property of the prototype filter it turns out that for all cases under consideration

$$d_n = a_{R_0-n} \quad \text{for} \quad n=0,1,\ldots,R_0 \tag{13a}$$

$$c_n = b_{R_1-n} \quad \text{for} \quad n=0,1,\ldots,R_1 . \tag{13b}$$

A proof for the above relations is given in Appendix B. As an example, a FB with $M=8$ and $N=33$ is considered. For $\mu=0$, the quadruplet given by (9b) (see also (11a)) contains polyphase components $\boldsymbol{g}_0=[h_0\ h_{16}\ h_1]$, $\boldsymbol{g}_9=[h_9\ h_8]$, $\boldsymbol{g}_8=[h_8\ h_9]$, and $\boldsymbol{g}_1=[h_1\ h_{16}\ h_0]$ corresponding to filters with transfer functions $A(z)$, $B(z)$, $C(z)$, and $D(z)$, respectively. For this case the filter orders are $R_0=2$ and $R_1=1$. It is obvious that the symmetries given by (13a) and (13b) hold. Same can be shown for all other quadruplets.

[4] For example, the structure in Figure 7(e) is actually the structure in Figure 7(a) multiplied with –1. Consequently, the structure in Figure 7(e) can be implemented as the structure in Figure 7(a) with the input or output signals multiplied by –1.

[5] The part related to (10) is omitted from these figures as that part consists only from two additions.

In the next two sections it is shown, how to efficiently implement each of those quadruplets.

TABLE III FILTER ORDERS FOR GIVEN VALUES OF Δ.

<table>
<tr><th rowspan="2">Δ</th><th colspan="2">Type 1</th><th colspan="3">Type 2</th></tr>
<tr><th>R_0, R_1</th><th>R</th><th>R_0</th><th>R_1</th><th>R</th></tr>
<tr><td>$\Delta=0$</td><td rowspan="2">K_E-1</td><td rowspan="2">K_E-1</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td>$1\le\Delta\le M/2-1$</td><td rowspan="3">K_E</td><td rowspan="3">K_E-1</td><td rowspan="3">K_E</td></tr>
<tr><td>$\Delta=M/2$</td><td>×</td><td>×</td></tr>
<tr><td>$M/2+1\le\Delta\le M-1$</td><td rowspan="3">K_E</td><td rowspan="3">K_E</td></tr>
<tr><td>$\Delta=M$</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td>$M+1\le\Delta\le 3M/2-1$</td><td rowspan="3">K_E+1</td><td rowspan="3">K_E</td><td rowspan="3">K_E+1</td></tr>
<tr><td>$\Delta=3M/2$</td><td>×</td><td>×</td></tr>
<tr><td>$3M/2+1\le\Delta\le 2M-1$</td><td>K_E+1</td><td>K_E+1</td></tr>
</table>

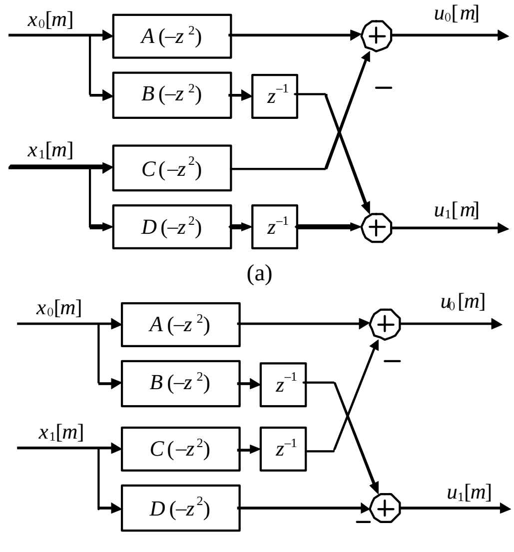


Figure 8. Implementation structure for quadruplets under consideration. (a) Type 1. (b) Type 2.

A. Type 1

A Type 1 quadruplet depicted in Figure 8(a) is defined by the following input-output relation:

$$\begin{bmatrix} U_0(z) \\ U_1(z) \end{bmatrix} = \begin{bmatrix} A(-z^2) & -C(-z^2) \\ z^{-1}B(-z^2) & z^{-1}D(-z^2) \end{bmatrix} \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}. \tag{14}$$

In the case under consideration (see Table III),[6]

$$R_0 = R_1 = K_E - 1 . \tag{15a}$$

In order to keep the equations compact, in the rest of this section, R will be used instead of K_E–1, that is,

$$R = K_E - 1 . \tag{15b}$$

Moreover, as K_E is even, R is in this case odd. By taking (15) into account, (14a) can be written in the time domain as (16a) (see next page, top). After applying the coefficient symmetries defined by (13a) and (13b), the time domain representation is given by (16b) (see next page, top), with

[6] As seen in Table III, for quadruplets shown in Figures (9c) and (9e), $R_0=R_1$ is equal to K_E and K_E+1, respectively.

$$\begin{bmatrix} u_0[m] \\ u_1[m] \end{bmatrix} = \begin{bmatrix} a_0 & 0 & -a_1 & 0 & \dots & 0 & -a_R & 0 & -c_0 & 0 & c_1 & 0 & \dots c_{R-1} & 0 \\ 0 & b_0 & 0 & -b_1 & \dots b_{R-1} & 0 & -b_R & 0 & d_0 & 0 & -d_1 & \dots & 0 & -d_R \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)}_{0,1,2R+1}[m] \\ \mathbf{x}^{(1)}_{0,1,2R+1}[m] \end{bmatrix} \quad (16a)$$

$$\begin{bmatrix} u_0[m] \\ u_1[m] \end{bmatrix} = \begin{bmatrix} a_0 & 0 & -a_1 & 0 & \dots & 0 & -a_R & 0 & -b_R & 0 & b_{R-1} & 0 & \dots b_0 & 0 \\ 0 & b_0 & 0 & -b_1 & \dots b_{R-1} & 0 & -b_R & 0 & a_R & 0 & -a_{R-1} & \dots & 0 & -a_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)}_{0,1,2R+1}[m] \\ \mathbf{x}^{(1)}_{0,1,2R+1}[m] \end{bmatrix} \quad (16b)$$

$$\mathbf{x}^{(t)}_{k,\eta,l}[m] = \begin{bmatrix} x_t[m-k] \\ x_t[m-k-\eta] \\ x_t[m-k-2\eta] \\ \vdots \\ x_t[m-k-l\eta] \end{bmatrix} \quad (17)$$

for $m, k, l, \eta \in \mathbb{Z}$, and $t = 0, 1$. In order to derive an efficient implementation, (16b) can be rewritten as

$$\begin{bmatrix} u_0[m] \\ u_1[m+1] \end{bmatrix} = \begin{bmatrix} a_0 - a_1 \dots a_{R-1} - a_R - b_R & b_{R-1} & \dots - b_1 & b_0 \\ b_0 - b_1 \dots b_{R-1} - b_R & a_R & -a_{R-1} \dots & a_1 & -a_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)}_{0,2,R}[m] \\ \mathbf{x}^{(1)}_{0,2,R}[m] \end{bmatrix}. \quad (18)$$

Here, the delays z^{-1} in (14) have been moved to the left side of the equation, based on the following set of identities:

$$U(z) = z^{-1}X(z) \Leftrightarrow u[m] = x[m-1] \Leftrightarrow u[m+1] = x[m]. \quad (19)$$

Therefore, in (18), $u_1[m+1]$ instead of $u_1[m]$ has been evaluated. Moreover, the columns containing only zeros have been removed.

By utilizing the efficient way of doing the complex multiplications (see for example [13]), systems given by (18) can be efficiently implemented as

$$\begin{bmatrix} u_0[m] \\ u_1[m+1] \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \times$$

$$\begin{bmatrix} e_0\, e_1 \dots e_R & 0 & 0\;0 & 0 & 0\;0\;0\;0\;0 \\ 0\;0\;0\;0 & -b_0\, b_1 \dots -b_{R-1}\, b_R & 0\;0\;0\;0 \\ 0\;0\;0\;0 & 0 & 0\;0 & 0 & 0\; f_R \dots f_1\; f_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)}_{0,2,R}[m] \\ \mathbf{x}^{0,2,R,-}_{0,2,R}[m] \\ \mathbf{x}^{(1)}_{0,2,R}[m] \end{bmatrix} \quad (20a)$$

with

$$e_0 = a_0 + b_0,\ e_1 = -a_1 - b_1,\ \dots, e_R = -a_R - b_R \quad (20b)$$

$$f_0 = a_0 - b_0,\ f_1 = -a_1 + b_1,\ \dots,\ f_R = -a_R + b_R \quad (20c)$$

and

$$\mathbf{x}^{k_1,\eta_1,l_1,-}_{k_0,\eta_0,l_0}[m] = \begin{bmatrix} x_0[m-k_0] - x_1[m-k_1-l_1\eta_1] \\ x_0[m-k_0-\eta_0] - x_1[m-k_1-(l_1-1)\eta_1] \\ \vdots \\ x_0[m-k_0-l_0\eta_0] - x_1[m-k_1] \end{bmatrix}. \quad (20d)$$

The coefficients e_k and f_k depend only on the prototype filter coefficients and as such can be pre-calculated. Furthermore, it should be pointed out that for the case with $N = 2KM-1$, all quadruplets in the FB are of Type 1. Therefore, in this special case, that is, for $N = 2KM-1$, the proposed implementation is similar to the one reported in [5].

As an example, the implementation for a Type 1 quadruplet for $M = 8$ and $N = 33$ is considered. As can be seen from (9a)–(9e), there are 3 Type 1 quadruplets in this FB generated for $\mu = 1, 2, 3$ (see also (11b)–(11d)). The one for $\mu = 1$ is considered here. After applying the above derivation on the quadruplet given by (9a), the following system of equation is obtained:

$$\begin{bmatrix} u_1[m] \\ u_6[m+1] \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} e_0 & e_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -h_7 & h_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 & f_1 & f_0 \end{bmatrix} \begin{bmatrix} x_2[m] \\ x_2[m-2] \\ x_2[m] - x_7[m-2] \\ x_2[m-2] - x_7[m] \\ x_7[m] \\ x_7[m-2] \end{bmatrix}, \quad (21a)$$

with

$$e_0 = h_2 + h_7,\ e_1 = -h_{15} - h_{10} \quad (21b)$$

$$f_0 = h_2 - h_7,\ f_1 = -h_{15} + h_{10}. \quad (21c)$$

The implementation structure is shown in Figure 9. Similar implementation can also be derived for $\mu = 2$ and $\mu = 3$.

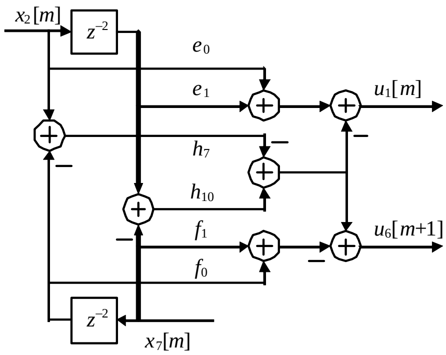


Figure 9. Proposed implementation structure for a Type 1 quadruplet for $\mu = 1$ that is part of a filterbank with M=8 and N=33.

B. Type 2

A Type 2 quadruplet depicted in Figure 8(b) is defined by the following input-output relation:

$$\begin{bmatrix} U_0(z) \\ U_1(z) \end{bmatrix} = \begin{bmatrix} A(-z^2) & -z^{-1}C(-z^2) \\ z^{-1}B(-z^2) & -D(-z^2) \end{bmatrix} \begin{bmatrix} X_0(z) \\ X_1(z) \end{bmatrix}. \quad (22)$$

In the case under consideration (see Table III),[7]

$$R_0 = K_E \tag{23a}$$

$$R_1 = K_E - 1\,. \tag{23b}$$

The difference between the orders R_0 and R_1 is due to a non-complete polyphase decomposition ($N+1 \neq 2KM$ with K being an integer), that is, some polyphase terms have more coefficients than others. Again, in this section, R instead of K_E will be used, that is,

$$R = K_E\,. \tag{23c}$$

By taking into account (23a)–(23c) as well as the coefficient symmetries given by (13a) and (13b), (22) can be written in the time domain as given by (24) (see page bottom), with $x^{(t)}_{k,\eta,l}$ defined by (17). In order to derive an efficient implementation, (24) can be rewritten after removing columns containing only zeros and by utilizing (19), as given by (25) (see page bottom).

Such system can be split into two parts as

$$\begin{bmatrix} u_0[m] \\ u_1[m+1] \end{bmatrix} = a_R \begin{bmatrix} x_0[m-2R] \\ -x_1[m+1] \end{bmatrix} + \begin{bmatrix} \tilde{u}_0[m] \\ \tilde{u}_1[m+1] \end{bmatrix}, \tag{26a}$$

with

$$\begin{bmatrix} \tilde{u}_0[m] \\ \tilde{u}_1[m+1] \end{bmatrix} = \begin{bmatrix} a_0 & -a_1 & a_2 & \dots & -a_{R-1} & -b_{R-1} & b_{R-2} & \dots & -b_1 & b_0 \\ b_0 & -b_1 & b_2 & \dots & -b_{R-1} & a_{R-1} & -a_{R-2} & \dots & a_1 & -a_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)}_{0,2,R-1}[m] \\ \mathbf{x}^{(1)}_{1,2,R}[m] \end{bmatrix}. \tag{26b}$$

The fist part can be implemented straightforwardly (using only two multiplications), whereas the second part, given by (26b), can be implemented by following the approach used for Type 1, that is,

$$\begin{bmatrix} \tilde{u}_0[m] \\ \tilde{u}_1[m+1] \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} e_0 \; e_1 \dots e_{R-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0\;0\;0 & 0 & b_0 - b_1 \dots b_{R-1} & 0 & 0 & 0 & 0 \\ 0\;0\;0 & 0 & 0 & 0 & 0 & 0 & f_{R-1} \dots f_1 \; f_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)}_{0,2,R-1}[m] \\ \mathbf{x}^{1,2,R-1,+}_{0,2,R-1}[m] \\ \mathbf{x}^{(1)}_{1,2,R-1}[m] \end{bmatrix}, \tag{27a}$$

with

$$e_0 = a_0 + b_0,\; e_1 = -a_1 - b_1,\; \dots,\; e_{R-1} = -a_{R-1} - b_{R-1} \tag{27b}$$

$$f_0 = a_0 + b_0,\; f_1 = -a_1 - b_1,\; \dots,\; f_{R-1} = -a_{R-1} - b_{R-1}\,, \tag{27c}$$

$\mathbf{x}^{(t)}_{k,\eta,l}$ defined by (17), and

$$\mathbf{x}^{k_1,\eta_1,l_1,+}_{k_0,\eta_0,l_0}[m] = \begin{bmatrix} x_0[m-k_0] + x_1[m-k_1-l_1\eta_1] \\ x_0[m-k_0-\eta_0] + x_1[m-k_1-(l_1-1)\eta_1] \\ \vdots \\ x_0[m-k_0-l_0\eta_0] + x_1[m-k_1] \end{bmatrix}. \tag{27d}$$

As in the case of Type 1, the coefficients e_k and f_k depend only on the prototype filter coefficients and as such can be pre-calculated.

As an example, the implementation for a Type 2 quadruplet for $M = 8$ and $N = 33$ is considered. As can be seen from (9b), there is one Type 2 quadruplets in this FB, that can be generated for $\mu = 0$ (see also (11a)). After applying the above derivation on the quadruplet given by (9b), following system of equation is obtained:

$$\begin{bmatrix} u_0[m] \\ u_7[m+1] \end{bmatrix} = h_1 \begin{bmatrix} x_1[m-4] \\ -x_0[m+1] \end{bmatrix} + \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} e_0 & e_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_8 & -h_9 & 0 & 0 \\ 0 & 0 & 0 & 0 & f_1 & f_0 \end{bmatrix} \begin{bmatrix} x_1[m] \\ x_1[m-2] \\ x_1[m] + x_0[m-3] \\ x_1[m-2] + x_0[m-1] \\ x_0[m-1] \\ x_0[m-3] \end{bmatrix}, \tag{28a}$$

with

$$e_0 = h_0 + h_8,\; e_1 = -h_{16} - h_9 \tag{28b}$$

$$f_0 = h_0 - h_8,\; f_1 = -h_{16} + h_9\,. \tag{28c}$$

The implementation is shown in Figure 10.

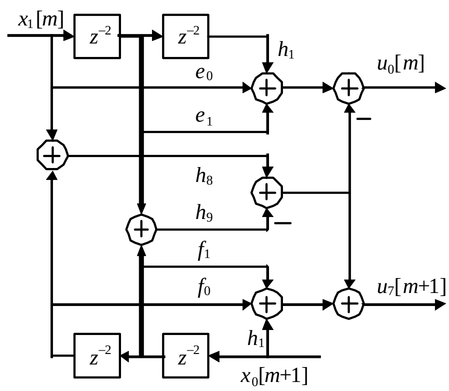


Figure 10. Proposed Implementation structure for one Type 2 quadruplet that is part of a filterbank with M=8 and N=33.

V. COMPLEXITY EVALUATION

The complexity of the proposed structures will be evaluated through the number of multiplications and number of

$$\begin{bmatrix} u_0[m] \\ u_1[m] \end{bmatrix} = \begin{bmatrix} a_0 & 0 & -a_1 & 0 & \dots & 0 & a_R & 0 & -b_{R-1} & 0 & \dots b_0 & 0 \\ 0 & b_0 & 0 & -b_1 & \dots & -b_{R-1} & 0 & -a_R & 0 & a_{R-1} & \dots 0 & -a_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)}_{0,1,2R} \\ \mathbf{x}^{(1)}_{0,1,2R} \end{bmatrix} \tag{24}$$

$$\begin{bmatrix} u_0[m] \\ u_1[m+1] \end{bmatrix} = \begin{bmatrix} a_0 & -a_1 & a_2 & \dots & -a_{R-1} & a_R & 0 & -b_{R-1} & b_{R-2} & \dots & -b_1 & b_0 \\ b_0 & -b_1 & b_2 & \dots & -b_{R-1} & 0 & -a_R & a_{R-1} & -a_{R-2} & \dots & a_1 & -a_0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^{(0)}_{0,2,R}[m] \\ \mathbf{x}^{(1)}_{-1,2,R}[m] \end{bmatrix} \tag{25}$$

[7] As seen in Table III, for quadruplet shown in Figure (9d), R_0 and R_1 are equal to K_E+1 and K_E, respectively.

additions per M input samples denoted as C^* and C^+, respectively. First the complexities for Type 1 and Type 2 structures are given, and then a generalization is presented for any combination of M and N. In the complexity evaluation, the given formulas take into account the prototype filter and matrices $\boldsymbol{S}_{\Delta}^{(2M)}$ and $\boldsymbol{T}$, as given in Figure 3. The implementation of DCT-IV is omitted from the evaluation as it is the same for all cases under consideration.

For a Type 1 quadruplet the implementation complexity can be evaluated by

$$C_{T1}^{*}(R) = 3R + 3 \tag{29a}$$

$$C_{T1}^{+}(R) = 4R + 5\,. \tag{29b}$$

As discussed in IV.A, R is equal to $K_E - 1$, K_E, or $K_E + 1$ for the structure given by Figure 7(a), Figure 7(c), or Figure 7(e), respectively.

For a Type 2 quadruplet the implementation complexity can be evaluated by

$$C_{T2}^{*}(R) = 3R + 2 \tag{30a}$$

$$C_{T2}^{+}(R) = 4R + 3\,. \tag{30b}$$

As discussed in IV.B, R is equal to K_E or $K_E + 1$ for the structure given by Figure 7(b) or Figure 7(d), respectively.

For an NPR cosine-modulated M-channel FB, with M being even, synthesized by using a linear-phase prototype filter of order N, with N being odd, the implementation complexity can be evaluated by using one of the following expressions:

$0 \le \Delta \le M - 1$

$$C^{x} = \left|\Delta - \frac{M}{2}\right| C_{T1}^{x}\left(K_E - 1 + \left\lfloor \frac{\Delta}{M/2} \right\rfloor\right) + \left(\frac{M}{2} - \left|\Delta - \frac{M}{2}\right|\right) C_{T2}^{x}(K_E) \tag{31a}$$

$M \le \Delta \le 2M - 1$

$$C^{x} = \left|\Delta - \frac{3M}{2}\right| C_{T1}^{x}\left(K_E + \left\lfloor \frac{\Delta}{3M/2} \right\rfloor\right) + \left(\frac{M}{2} - \left|\Delta - \frac{3M}{2}\right|\right) C_{T2}^{x}(K_E + 1) \tag{31b}$$

with K_E and Δ being related to the filter order by (3a). In the above equation, ‘x’ stands for ‘*’ or ‘+’, that is, the same equation is used for evaluating the number of required multiplications and additions.

For prototype filters with even order N, the above expression can be used to get a close enough estimation by evaluating them for order N+1. FBs with odd number of channels are not of interest in practice due to the fact that a FB with even number of channels, having one channel more than the FB with odd number of channels, can in most cases be implemented more efficiently than the one with odd number of channels. Therefore, complexity estimation for M odd can be evaluated by using the above equations for M+1 instead of M.

VI. Examples

This section shows by means of an example the benefits of the proposed implementation method. The section is divided into two parts. First, the proposed implementation is compared to the polyphase implementation, and, second, it is elaborated why having efficient implementation methods for NPR FBs with an arbitrary prototype filter order is beneficial. It should be pointed out that the complexity evaluation is performed only for the implementation of the prototype filter and matrices $\boldsymbol{S}_{\Delta}^{(2M)}$ and $\boldsymbol{T}$. In both cases, the proposed one and the polyphase one, in addition to the prototype filter, the DCT-IV has to be implemented. It is omitted from the following comparison because the implementation cost of the DCT-IV is equal for both approaches.

In order to compare the proposed method with the polyphase one, a family of FBs has been designed having following properties: $M = 32$, maximum allowable aliasing and amplitude distortions $\delta_a = \delta_d = 0.01$, stopband edge $\omega_s = \pi/M$, and $N = 95, 97, \ldots, 319$. The FBs have been designed by using the method presented in [6], that is, the stopband attenuation of the prototype filter is minimized subject to the given FB constraints. For these FBs, the achieved stopband attenuations are shown in Figure 11. It can be noticed that the attenuation is increasing in a monotone, continuous manner, when the filter order is increasing.[8] This is not the case for PR FBs where designs other than $N = 2KM–1$ result in FBs with poor performance [14].

The implementation complexities by using the proposed implementation and the polyphase one are shown in Figure 12 with the relative comparison given in Figure 13. These figures show that the number of required multiplications for the proposed implementation is always lower than that of the polyphase one. The number of additions in the proposed method is higher than that of the polyphase one by M/2 addition independently of the filter order. This is not a problem as the difference is small, particularly for filters of high order, compared to the overall number of addition and the fact that adders are less costly to implement than multipliers. The number of required multiplications and additions in the polyphase case has been evaluated by the following two expressions:

$$C_p^{*} = N + 1 \tag{32a}$$

$$C_p^{+} = N + 1\,. \tag{32b}$$

As already mentioned before, (32b) also includes the 2M additions required for implementing matrices $\boldsymbol{S}_{\Delta}^{(2M)}$ and $\boldsymbol{T}$ as given in Figure 3. In order to make the comparison more fair, it was assumed that both implementation structures, the polyphase one and the proposed one, are implemented according to Figure 3. The only difference is that the polyphase one goes one step further and utilizes the coefficient symmetries, as described in this paper.

By using the proposed implementation method for NPR FBs, systems with lower delays / complexity for a given FB

[8] Similar behavior was also observed for different values of M.

requirements can be achieved. This is important due to the fact that for many applications the delay introduced by the system is limited by standards. As an example, in Table IV, numerical data of some characteristic designs is given. As seen from the table, if a design with filters having 50 dB stopband attenuation is desired with parameters δ_a, δ_d, and ω_s as defined above, then a filter with order $N = 219$ is required. In this case 22% less multiplications are required by the proposed implementation compared to the polyphase one. In order to compare with the technique proposed in [5], where the efficient implementation applies only to filters with $N = 2KM–1$, the two nearest $N = 2KM–1$ cases are given in the table. As it can be seen, the order $N = 191$ has attenuation lower than 50 dB and as such it does not satisfy the requirements of this example. In order to meet the given specification, the technique proposed in [5] has to use an over-designed filter with order $N = 255$. However, such system has a 36 samples longer delay and a more complex implementation compared to the optimum, $N = 219$ case. Therefore, the proposed method gives more flexibility when selecting an appropriate FB for a problem at hand.

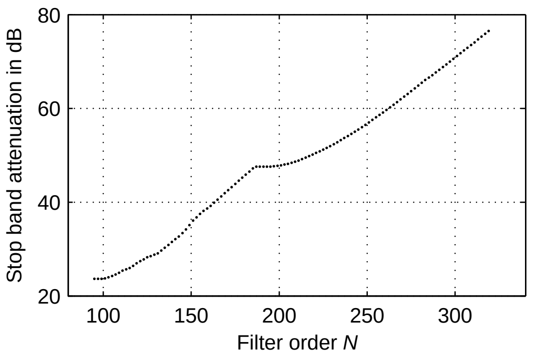
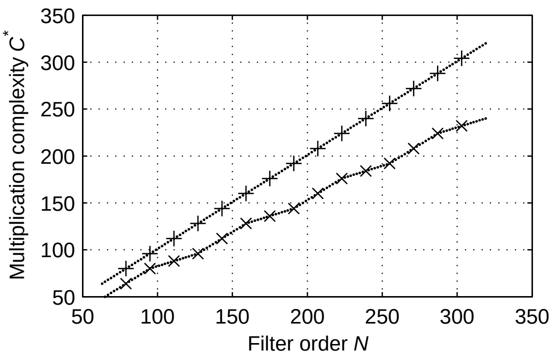
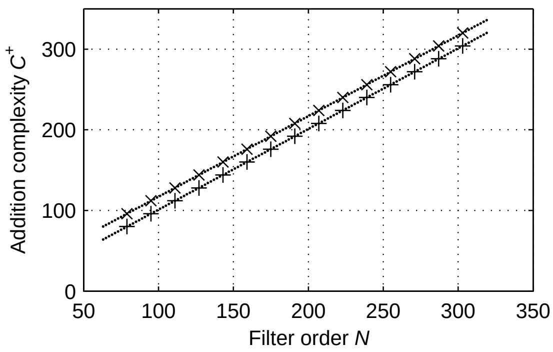


Figure 11. Stopband attenuations for 32-channel NPR cosine-modulated filterbanks with $\delta_a = \delta_d = 0.01$.

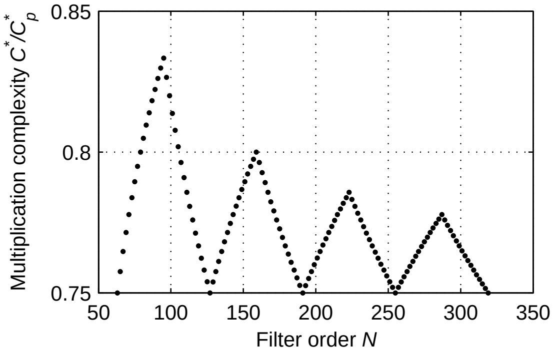


(a)

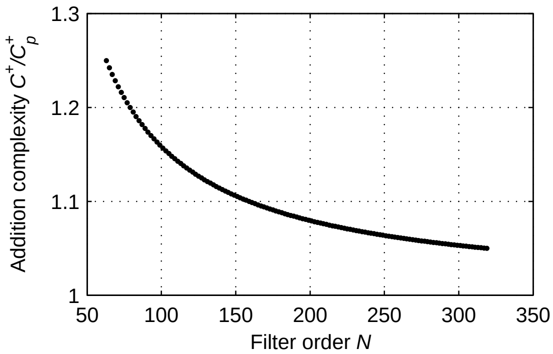


(b)

Figure 12. Implementation complexity for filterbanks shown in Figure 11 when using the proposed method (···×···) and the polyphase method (···+···). (a) Multiplication complexity. (b) Addition complexity.



(a)



(b)

Figure 13. Relative comparison between the proposed method and the polyphase method. (a) Multiplication complexity. (b) Addition complexity.

TABLE IV ATTENUATION AND IMPLEMENTATION COMPLEXITY FOR FILTERBANKS WITH $M = 32$ AND VARIOUS FILTER ORDERS IN NUMBER OF MULTIPLICATIONS (C^*) AND ADDITIONS (C^+) PER OUTPUT SAMPLE

		Proposed		Method in [5]		Polyphase		Comparison	
N	Att [dB]	C^*	C^+	$C^*_{[5]}$	$C^+_{[5]}$	C^*_p	C^+_p	C^*/C^*_p	C^+/C^+_p
2·3·32–1=191	47.6	144	208	144	108	192	192	0.75	1.08
219	50.2	172	236	–	–	220	220	0.78	1.07
2·4·32–1=255	58.1	192	272	192	272	256	256	0.75	1.06

VII. Concluding Remarks

In this paper an implementation method has been proposed that reduces the number of required multiplications when implementing a linear-phase prototype filter of an arbitrary order used for building an NPR cosine-modulated FB. Following remarks regarding the discussion presented in this paper should be made:

First, the method is useful only for NPR FBs. PR FBs can be implemented more efficiently by using a lattice structure. However, the PR FBs are restricted to orders $N=2KM-1$, whereas the proposed method can be used with an arbitrary filter order.

Second, although this paper mainly concentrates on odd filter orders and even number of channels, the method can also be used to implement prototype filters of even order as well as FBs with odd number of channels. In these cases, in addition to Type 1 and Type 2 quadruplets, some trivial relations have to be implemented.

Third, it is shown how to efficiently implement the cosine-modulated part of the FB for even number of channels and odd filter orders. This can be extended to other number of channels and filter orders. The paper concentrated on the above cases as those are the most useful ones from a practical viewpoint.

Fourth, the proposed implementation does not depend on the properties of the FB or the method how the FB has been designed. Therefore the proposed method can be used for any existing or newly designed FB having a linear-phase prototype filter.

Fifth, as shown in the example section, in some cases FBs with lower delay can be used without increasing the number of required multiplications. This is very important in many applications.

Sixth, in this paper analysis-synthesis systems have been considered. However, the proposed implementation method can also be used for synthesis-analysis systems, also known as transmultiplexers.

Appendix A

This appendix shows that the modulation part of a cosine-modulated FB with a linear-phase prototype filter of an odd order having even number of channels can always be implemented by using a fast DCT.[9] Although, this appendix concentrates on the above mention cases, a similar principle can be also applied for other filter orders and/or number of channels.

It has been shown in [1] that for FBs with even number of channels and filter orders equal to $N=2K_EM-1$, with K_E being even, cosine-modulation matrix given by (5a) can be implemented by using a DCT-IV transform as given by (6a)–(6g)[10] (DCT-IV can be implemented by a fast DCT [11], [12]). In this special case, the cross-connection matrix $\boldsymbol{S}_{\Delta}^{(2M)}$ becomes an identity matrix because $\Delta=0$. This special order will be denoted here as N_K, that is, $N_k=2K_EM-1$. The corresponding modulation matrix denoted as $[\boldsymbol{C}_{NK}]_{k,l}$ is given by

$$[\boldsymbol{C}_{NK}]_{k,l} = 2\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(l-\frac{N_K}{2}\right)+\theta^k\right) \tag{33a}$$

for $k=0,1,\ldots,M-1$ and $l=0,1,\ldots,2M-1$ and θ_k is

$$\theta_k = (-1)^k\frac{\pi}{4}. \tag{33b}$$

The goal now is to show that the cosine-modulation matrix $[\boldsymbol{C}_1]_{k,l}$ of a FB with filter order $N=2K_EM+2\Delta-1$ can be implemented as

$$[\boldsymbol{C}_1]_{k,l} = [\boldsymbol{C}_{NK}]_{k,l}\boldsymbol{S}_{\Delta}^{(2M)}, \tag{34}$$

with $\boldsymbol{S}_{\Delta}^{(2M)}$ given by (6h).

In order to prove this, first the implementation for $N=N_K+2$ is considered. For this case the modulation matrix is given by

$$[\boldsymbol{C}_1]_{k,l} = 2\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(l-\frac{N_K+2}{2}\right)+\theta^k\right) \tag{35}$$

for $k=0,1,\ldots,M-1$ and $l=0,1,\ldots,2M-1$. By comparing (33a) and (35), it can be observed that

$$l-\frac{N_K+2}{2} = l-\frac{N_K}{2}-1 = (l-1)-\frac{N_K}{2}, \tag{36a}$$

that is, matrices $[\boldsymbol{C}_1]_{k,l}$ and $[\boldsymbol{C}_{NK}]_{k,l}$ have identical columns

$$[\boldsymbol{C}_1]_{k,l+1} = [\boldsymbol{C}_{NK}]_{k,l} \tag{36b}$$

for $k=0,1,\ldots,M-1$ and $l=0,1,\ldots,2M-2$. The only different columns are the first column in matrix $[\boldsymbol{C}_1]_{k,l}$ given as

$$\begin{aligned}[\boldsymbol{C}_1]_{k,0} &= 2\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(0-\frac{N_K+2}{2}\right)+\theta^k\right) = \\ &2\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(-K_EM-\frac{1}{2}\right)+\theta^k\right)\end{aligned} \tag{37a}$$

and the last column in matrix $[\boldsymbol{C}_{NK}]_{k,l}$ given as

$$\begin{aligned}[\boldsymbol{C}_{NK}]_{k,2M-1} &= 2\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(2M-1-\frac{N_K}{2}\right)+\theta^k\right) = \\ &2\cos\left((2k+1)\pi+\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(-K_EM-\frac{1}{2}\right)+\theta^k\right).\end{aligned} \tag{37b}$$

By applying the cosine transformation $\cos(\alpha+\beta)=\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)$ on (37b) and noticing that

$$\cos((2k+1)\pi) = -1 \tag{38a}$$

$$\sin((2k+1)\pi) = 0, \tag{38b}$$

(37b) becomes

$$[\boldsymbol{C}_{NK}]_{k,2M-1} = -2\cos\left(\frac{\pi}{M}\left(k+\frac{1}{2}\right)\left(-K_EM-\frac{1}{2}\right)+\theta^k\right), \tag{39}$$

that is,

---

[9] A similar proof for a different type of cosine modulation functions than the one used in this paper can be found in [18].

[10] Similar relations can be derived for $N=2K_OM-1$, with K_O being an odd integer. However, it has turned out that for the discussion in this paper, it is beneficial to use only the relations for K_E even.

$$[\boldsymbol{C}_{NK}]_{k,2M-1} = -[\boldsymbol{C}_1]_{k,0}. \tag{40}$$

This means that for $N = N_K+2$, all columns of matrix $[\boldsymbol{C}_1]_{k,l}$ are shifted version of matrix $[\boldsymbol{C}_{NK}]_{k,l}$ with one of the columns having a different sign. Therefore, by simply rearranging the input signals going into the cosine-modulation matrix, the matrix $[\boldsymbol{C}_{NK}]_{k,l}$ can be used for both of those cases. The required rearrangement can be expressed by the following matrix (see also Figure 3):

$$\boldsymbol{S}_1^{(2M)} = \begin{bmatrix} \boldsymbol{0}_{2M-1,1} & \boldsymbol{I}_{2M-\Delta} \\ -1 & \boldsymbol{0}_{1,2M-1} \end{bmatrix}, \tag{41a}$$

that is,

$$\boldsymbol{C}_1 = \boldsymbol{C}_{NK}\boldsymbol{S}_1^{(2M)}. \tag{41b}$$

Following the some idea stated above, it can be easily shown that the same principle can be applied for other filter orders up to $N = N_K+4M–2$. For example, for $N = N_K+4$,

$$[\boldsymbol{C}_1]_{k,l+2} = [\boldsymbol{C}_{NK}]_{k,l} \tag{42a}$$

for $k = 0, 1, \ldots, M–1$ and $l = 0, 1, \ldots, 2M–3$ and

$$[\boldsymbol{C}_{NK}]_{k,2M-2} = -[\boldsymbol{C}_1]_{k,0} \tag{42b}$$

$$[\boldsymbol{C}_{NK}]_{k,2M-1} = -[\boldsymbol{C}_1]_{k,1}. \tag{42c}$$

Consequently,

$$\boldsymbol{S}_2^{(2M)} = \begin{bmatrix} \boldsymbol{0}_{2M-2,2} & \boldsymbol{I}_{2M-2} \\ -\boldsymbol{I}_2 & \boldsymbol{0}_{2,2M-2} \end{bmatrix} \tag{42d}$$

and

$$\boldsymbol{C}_1 = \boldsymbol{C}_{NK}\boldsymbol{S}_2^{(2M)}. \tag{42e}$$

In the more general case, for $N = 2K_E M - 1 + 2\Delta$, the cross-connection matrix $\boldsymbol{S}_\Delta^{(2M)}$ becomes as given by (6h) thereby proving (34).

Finally it should be pointed out that for $N = N_K+4M$, $[\boldsymbol{C}_1]_{k,l} = -[\boldsymbol{C}_{NK}]_{k,l}$ for $k = 0, 1, \ldots, M–1$ and $l = 0, 1, \ldots, 2M–1$. Because all columns change sign, this can be implemented by changing sign of the input or output signals. In the implementation this is performed by the scaling factor λ. Therefore, for $N = N_K+4M$,

$$[\boldsymbol{C}_1]_{k,l} = [\boldsymbol{C}_{NK}]_{k,l} \tag{43a}$$

for $k = 0, 1, \ldots, M–1$ and $l = 0, 1, \ldots, 2M–1$ with

$$\lambda = -\lambda_{NK} \tag{43b}$$

and λ_{NK} corresponding to the $N = N_K$ case.

Here, only the analysis modulation matrix is considered. Same principle can be applied for deriving an implementation for the synthesis modulation matrix.

APPENDIX B

This appendix shows that the symmetry relations given by (13a) and (13b) hold for the quadruplet given by Figure 7(a), that is, (9a). This corresponds to Type 1 quadruplet given by Figure 8(a). For quadruplets given by Figures 7(b)–7(e), that is, (9b)–(9e), a proof similar to the one here can be done.

According to Figure 8(a), Figure 7(a), and (9a) it follows that

$$A(z) = G_{\mu+\Delta}(z) \tag{44a}$$

$$B(z) = G_{\mu+\Delta+M}(z) \tag{44b}$$

$$C(z) = G_{M-\mu+\Delta-1}(z) \tag{44c}$$

$$D(z) = G_{2M-\mu+\Delta-1}(z) \tag{44d}$$

for $0 \le \Delta \le M/2–1$ and $\mu = \Delta, \Delta+1, \ldots, M/2–1$ with Δ given by (3c). Combining the above relations with (13a) and (13b), it turns out that for the case under consideration the following relations have to be proven:

$$g_{\mu+\Delta}[n] = g_{2M-\mu+\Delta-1}[R_0 - n] \quad \text{for} \quad n = 0,1,\ldots,R_0 \tag{45a}$$

$$g_{M-\mu+\Delta-1}[n] = g_{\mu+\Delta+M}[R_1 - n] \quad \text{for} \quad n = 0,1,\ldots,R_1. \tag{45b}$$

According to (4c), the orders of the polyphase components can be derived as

$$N_\xi = N_A = \left\lceil \frac{N+1}{2M} \right\rceil - 1 \quad \text{for} \quad \xi = 0, 1, \ldots, P–1 \tag{46a}$$

$$N_\xi = N_B = \left\lfloor \frac{N+1}{2M} \right\rfloor - 1 \quad \text{for} \quad \xi = P, P+1, \ldots, 2M–1, \tag{46b}$$

where

$$P = N + 1 - 2M\left\lfloor \frac{N+1}{2M} \right\rfloor. \tag{46c}$$

This means that the first P polyphase filters are of order N_A and the rest $2M–P$ polyphase filters are of order N_B. By combining (3a) with (46c), it turns out that

$$P = 2K_E M + 2\Delta - 2M\left\lfloor K_E + \frac{\Delta}{2M} \right\rfloor = 2\Delta. \tag{47}$$

The last equality in (47) holds because $\Delta < M/2$.

As discussed in [15]–[17], the polyphase filter coefficients present mirror image symmetries for polyphase filter pairs. More specifically,

$$g_\zeta[n] = g_{P-1-\zeta}[N_A - n] \quad \text{for} \quad \zeta = 0,1,\ldots,P-1 \tag{48a}$$

$$g_\zeta[n] = g_{2M-1+P-\zeta}[N_B - n] \quad \text{for} \quad \zeta = P, P+1,\ldots,2M-1. \tag{48b}$$

For the case under consideration, $\mu \ge \Delta$. Therefore, $\mu + \Delta \ge 2\Delta = P$ and $R_0 = N_B = N_{KE} – 1$. Consequently, (48b) shows that $(\mu+\Delta)$-th polyphase filter is the reverse of the $(2M - 1 + P - (\mu+\Delta))$-th polyphase filter. Since

$$2M - 1 + P - (\mu+\Delta) = 2M - 1 + 2\Delta - \mu - \Delta = 2M - 1 - \mu + \Delta, \tag{49}$$

(45a) is proven. (45b) can be proven in the exactly same way, since $\mu + \Delta + M \ge 2\Delta + M \ge P$.

VIII. REFERENCES

[1] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[2] N. J. Fliege, *Multirate Digital Signal Processing*, Chicester: John Wiley and Sons, 1994.

[3] H. S. Malvar, *Signal Processing with Lapped Transforms*. Norwood: Artec House, 1992.

[4] R. Bregović and T. Saramäki, "An efficient approach for designing nearly perfect-reconstruction low-delay cosine-modulated filter banks," in *Proc. IEEE Int. Symp. Circuits Syst.*, Scottsdale, Arizona, May 2002, vol. 1, pp. 825–828.

[5] J. Alhava and A. Viholainen, "Coefficient quantization in nearly perfect-reconstruction cosine-modulated filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Istanbul, Turkey, June 2000, pp. 536–539.

[6] T. Saramäki and R. Bregović, "An efficient approach for designing nearly perfect-reconstruction cosine-modulated and modified DFT filter banks," in *Proc IEEE Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, Utah, May 2001, vol. VI, pp. 3617–3620.

[7] W.-S. Lu, T. Saramäki, and R. Bregović, "Design of practically perfect-reconstruction cosine-modulated filter banks: A second-order cone programming approach," *IEEE Trans. Circuits Syst. I*, vol. 51, pp. 552–563, March 2004.

[8] P. N. Heller, T. Karp, and T. Q. Nguyen, "A general formulation of modulated filter banks," *IEEE Trans. on Signal Processing*, vol. 47, pp. 986–1002, April 1999.

[9] T. Karp, A. Mertins, and G. Schuller, "Efficient biorthogonal cosine-modulated filter banks," *Signal Processing*, vol. 81, pp. 997–1016, May 2001.

[10] F. Cruz-Roldán and M. Monteagudo-Prim, "Efficient implementation of nearly perfect reconstruction FIR cosine-modulated filterbanks," *IEEE Trans. on Signal Processing*, vol. 52, pp. 2661–2664, Sept. 2004.

[11] K. R. Rao and P. Yip, *Discrete Cosine Transform, Algorithms, Advantage, Applications.* NY: Academic, 1990.

[12] C. W. Kok, "Fast algorithm for computing discrete cosine transform," *IEEE Trans. on Signal Processing*, vol. 45, pp. 757–760, March 1997.

[13] K. K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. NY: Wiley, 1999.

[14] T. Q. Nguyen and R. D. Koilpillai, "The theory and design of arbitrary-length cosine-modulated filter banks and wavelets, satisfying perfect reconstruction," *IEEE Trans. on Signal Processing*, vol. 44, pp. 473–483, March 1996.

[15] Z. J. Mou, "Symmetry exploitation in digital interpolators/decimators," *IEEE Trans. Signal Process.*, vol 44, pp. 2611–2615, Oct. 1996.

[16] Y. J. Yu, Y. C. Lim, and T. Saramäki, "Restoring coefficient symmetry in polyphase implementation of linear-phase filters," *Circuits, Systems, Signal Processing*, no. 2, pp. 253–264, 2006.

[17] R. Bregović, T. Saramäki, Y. J. Yu, and Y. C. Lim, "An efficient implementation of linear-phase FIR filters for a rational sampling rate conversion," in *Proc. IEEE Int. Symp. Circuits Syst.*, Island of Kos, Greece, May 2006, pp. 5395–5398.

[18] G. D. T. Schuller and T. Karp, "Modulated filter banks with arbitrary system delay: Efficient implementations and the time-varying case," *IEEE Trans. Signal Process.*, vol. 48, pp. 737–748, March 2000.