A NEW IMPLEMENTATION OF ARBITRARY-LENGTH COSINE-MODULATED FILTER BANK

Xi-Qi Gao and Zhen-Ya He
Department of Radio Engineering
Southeast University
Nanjing,210096, P.R. China
Email: gxq@seu.edu.cn

Xiang-Gen Xia
Department of Electrical Engineering
University of Delaware
Newark, DE 19716, USA
Email: xxia@mail.eecis.udel.edu

ABSTRACT

In this paper, the fast implementation of cosine-modulated filter bank (CMFB) is revisited. A class of paraunitary CMFBs with arbitrary length is considered. By further reorganizing of the polyphase component matrix and using the linear-phase property of the prototype filter, we obtain a more efficient implementation structure for the CMFB, in which we use 2×2 lossless matrices instead of 2×1 ones. In the new implementation, the number of two-channel lossless lattices is reduced by a factor of two.

1. INTRODUCTION

Cosine-modulated filter banks (CMFB), which can be designed and implemented efficiently, have been investigated extensively in recent years [1]-[6]. While the design of CMFBs has been addressed by many researchers, we investigate the implementation of paraunitary CMFBs in this paper. Typically, the polyphase component matrix of a paraunitary CMFB can be expressed as the product of a polyphase part in terms of the polyphase components of the prototype filter and a modulation part, and the CMFB can be implemented through two-channel lossless lattices and fast discrete cosine/sine transform (DCT/DST) algorithms, see, for example, [1,4]. M two-channel lattices are used for an M-channel CMFB. Notice that only half the number of the lattices are required in the implementation of Malvar's CMFB called extended lapped transform (ELT) [2]. The motivation of this paper is to generalize the above Malvar's result to other paraunitary CMFBs. The arbitrary-length CMFB developed by Nguyen and Koilpillai in [3] is considered in this paper.

This paper is organized as follows. In section 2, we first review the arbitrary-length CMFB briefly. Then we show that the four filters in each pair of power complementary polyphase components of the prototype filter and its related one due to the linear phase property of the prototype filter form a 2×2 paraunitary matrix. In section 3, a new expression of the polyphase component matrix of the CMFB is developed. Based on it, a more efficient implementation structure is obtained by using the 2×2 lossless matrices instead of the 2×1 ones in the traditional implementation. The implementation complexity of the CMFB is discussed in section 4.

Notations: Bold letters indicate vectors and matrices. The functions $\lceil x \rceil$ and $\lfloor x \rfloor$ round the value of x to the nearest integers towards infinity and minus infinity respectively. $\mathbf{C}_N^{III}$ and $\mathbf{C}_N^{IV}$ stand for the standard DCT matrices as defined in [8]. $\mathbf{0}$ stands for matrix whose entries are all zeros. $\mathbf{I}_N$ and $\mathbf{J}_N$ are the $N \times N$ identity and reverse identity matrices, respectively.

2. THE ARBITRARY-LENGTH CMFB

2.1. A Review of The Arbitrary-Length CMFB

Let $h(n)$ denote a linear-phase low-pass prototype filter with length $N = 2m_0 M + m_1$, where m_0 and m_1 are integers, and $0 \le m_1 \le 2M-1$, then the M-channel arbitrary-length CMFB is defined as [3]:

$$h_k(n) = 2h(n)\cos\left(\frac{\pi(k+0.5)}{M}\left(n-\frac{N-1}{2}\right)+(-1)^k\frac{\pi}{4}\right) \quad (1a)$$

$$f_k(n) = 2h(n)\cos\left(\frac{\pi(k+0.5)}{M}\left(n-\frac{N-1}{2}\right)-(-1)^k\frac{\pi}{4}\right) \quad (1b)$$

$$0 \le k \le M-1,\ 0 \le n \le N-1$$

where $h_k(n)$ and $f_k(n)$ are the impulse responses of the kth analysis and synthesis filters. The CMFB is exactly the one investigated in [1] with an arbitrary length of the prototype filter $h(n)$.

Suppose that the low-pass prototype filter is symmetry, then the synthesis filter $f_k(n)$ is the time-reversed and shifted version of the analysis filter $h_k(n)$. This relation means that the CMFB is paraunitary if and only if it has perfect reconstruction property [9]. Let $G_k(z)$, $k=0,1,\ldots,2M-1$, denote the type I polyphase component filters of $H(z)$ [9]. Due to the symmetry property of the prototype filter, $G_k(z)$ satisfies the following conditions:

$$\widetilde{G}_k(z)=\begin{cases} z^{m_0} G_{m_1-1-k}, & k \le m_1-1 \\ z^{m_0-1} G_{2M+m_1-1-k}, & k \ge m_1 \end{cases} \quad (2)$$

where $\widetilde{G}_k(z)=G_k(z^{-1})$. It has been shown in [3] that the necessary and sufficient conditions on the polyphase filters $G_k(z)$ for perfect reconstruction are the following:

$$\tilde{G}_k(z)G_k(z)+\tilde{G}_{M+k}(z)G_{M+k}(z)=\frac{1}{2M}, \quad 0\le k\le M-1. \quad (3)$$

This means that appropriate pairs of the polyphase filters are power complementary. Depending on the lengths of the two filters $G_k(z)$ and $G_{M+k}(z)$ and the relationship between them, four classes of power complementary pairs can be distinguished in the general case for arbitrary length prototype filters. The conditions given by (3) can be satisfied by four different modes [3]. In *mode a* and *c*, the two filters have the same length. If they are related by equ. (2), they are under *mode c*, otherwise *mode a*. In *mode c*, both the two filters must be some delays. In *mode b* and *d*, $G_k(z)$ and $G_{M+k}(z)$ have different lengths. If they are related to themselves by equ. (2), they are under *mode d*, otherwise *mode b*. In *mode d*, one of the two filters must be a delay, and all coefficients of the other one must be zeros.

2.2. Lattice Structure for a Power Complementary Pair and Its Related One

The power complementary filter pair $G_k(z)$ and $G_{M+k}(z)$ satisfying (3) can be completely factored as the following two-channel lossless lattice:

$$\sqrt{2M}\begin{bmatrix} G_k(z) \\ G_{M+k}(z)\end{bmatrix}=\mathbf{R}_{k,m}\Lambda(z)\mathbf{R}_{k,m-1}\Lambda(z)\cdots\mathbf{R}_{k,1}\Lambda(z)\begin{bmatrix} c_{k,0} \\ s_{k,0}\end{bmatrix} \quad (4)$$

where

$$\mathbf{R}_{k,l}=\begin{bmatrix} c_{k,l} & s_{k,l} \\ s_{k,l} & -c_{k,l}\end{bmatrix}, \qquad \Lambda(z)=\begin{bmatrix} 1 & 0 \\ 0 & z^{-1}\end{bmatrix},$$

$c_{k,l}=\cos\theta_{k,l}$, $s_{k,l}=\sin\theta_{k,l}$, $l=0,1,2,\ldots m$, and m depends on the lengths of the two filters. For the case $N=2mM$, all the polyphase components $G_k(z)$ have the same length, and there is no restriction on any angle parameter $\theta_{k,l}$. For the general case when the prototype filter has arbitrary length, there are different constraints on the angles of the lossless lattices corresponding to the four modes (See [3] for details). For each power complementary filter pair we can find a related one from equ. (2). The four filters in the two pairs define a 2×2 system. We define the following three types of 2×2 systems in terms of different m_1 and k:

i). $m_1\le M$, $k\le m_1-1$ or $m_1>M$, $m_1-M\le k\le M-1$

$$\mathbf{L}_k^{(1)}(z)\overset{\Delta}{=}\sqrt{2M}\begin{bmatrix} G_k(z) & G_{M+m_1-1-k}(z) \\ -z^{-1}G_{M+k}(z) & G_{m_1-1-k}(z)\end{bmatrix} \quad (5a)$$

ii). $m_1\le M$, $m_1\le k\le M-1$

$$\mathbf{L}_k^{(2)}(z)\overset{\Delta}{=}\sqrt{2M}\begin{bmatrix} G_k(z) & G_{M+m_1-1-k}(z) \\ G_{M+k}(z) & -G_{2M+m_1-1-k}(z)\end{bmatrix} \quad (5b)$$

iii). $m_1>M$, $k\le m_1-M-1$

$$\mathbf{L}_k^{(3)}(z)\overset{\Delta}{=}\sqrt{2M}\begin{bmatrix} G_k(z) & G_{m_1-M-1-k}(z) \\ G_{M+k}(z) & -G_{m_1-1-k}(z)\end{bmatrix}. \quad (5c)$$

$\mathbf{L}_k^{(1)}$ is for the polyphase filters under *mode b* and *mode d*. $\mathbf{L}_k^{(2)}$ and $\mathbf{L}_k^{(3)}$ are for the polyphase filters under *mode a* and *mode c*. It can be shown that these systems are paraunitary and can be expressed as the following 2×2 lattices:

$$\mathbf{L}_k^{(1)}(z)=\begin{cases}\begin{bmatrix} 0 & 1 \\ -1 & 0\end{bmatrix}\mathbf{R}_{k,m_0-1}\Lambda(z)\cdots\mathbf{R}_{k,1}\Lambda(z)\mathbf{R}_{k,0}\begin{bmatrix} z^{-1} & 0 \\ 0 & (-1)^{m_0}\end{bmatrix}, & \theta_{k,m_0}=\frac{\pi}{2} \\ \Lambda(-z)\mathbf{R}_{k,m_0}\Lambda(z)\cdots\mathbf{R}_{k,2}\Lambda(z)\mathbf{R}_{k,1}\begin{bmatrix} 1 & 0 \\ 0 & (-1)^{(m_0+1)}\end{bmatrix}, & \theta_{k,0}=0.\end{cases} \quad (6a)$$

$$\mathbf{L}_k^{(2)}(z)=\mathbf{R}_{k,m_0-1}\Lambda(z)\mathbf{R}_{k,m_0-2}\Lambda(z)\cdots\mathbf{R}_{k,1}\Lambda(z)\mathbf{R}_{k,0}\begin{bmatrix} 1 & 0 \\ 0 & (-1)^{m_0}\end{bmatrix} \quad (6b)$$

$$\mathbf{L}_k^{(3)}(z)=\mathbf{R}_{k,m_0}\Lambda(z)\mathbf{R}_{k,m_0-1}\Lambda(z)\cdots\mathbf{R}_{k,1}\Lambda(z)\mathbf{R}_{k,0}\begin{bmatrix} 1 & 0 \\ 0 & (-1)^{m_0+1}\end{bmatrix} \quad (6c)$$

In *mode c* and *mode d*, the corresponding lattices are trivial [10].

In practice, a two-channel lossless lattice can be implemented by using the two-multiplier structure for each section[2,9]. A 2×1 lossless lattice with m free angle parameters and others set to be zero or $\pi/2$ can be implemented by using $2m$ multipliers and $2(m-1)$ adders. For the corresponding 2×2 system, both the two numbers are $2m+1$. For $\mathbf{L}_k^{(1)}$ and $\mathbf{L}_k^{(2)}$, there are m_0 free angle parameters, and hence $2m_0+1$ multipliers and $2m_0+1$ adders are required. For $\mathbf{L}_k^{(3)}$, there are m_0+1 free angle parameters, and hence $2m_0+3$ multipliers and $2m_0+3$ adders are required.

3. FAST IMPLEMENTATION OF THE ARBITRARY-LENGTH CMFB

In this section, we consider the implementation of the CMFB. Considering the relationship between the synthesis bank and the analysis bank, we only deal with the later. The polyphase component matrix of the analysis bank can be expressed as [3]:

$$\mathbf{E}(z)=\hat{\mathbf{C}}\begin{bmatrix}\mathbf{g}_0(-z^2) \\ z^{-1}\mathbf{g}_1(-z^2)\end{bmatrix} \quad (7)$$

where

$$[\hat{\mathbf{C}}]_{k,l}=2\cos\left(\frac{\pi(k+0.5)}{M}\left(l-\frac{N-1}{2}\right)+(-1)^k\frac{\pi}{4}\right)$$

$$\mathbf{g}_0(z)=diag(G_0(z)\ G_1(z)\ \ldots\ G_{M-1}(z))$$

$$\mathbf{g}_1(z)=diag(G_M(z)\ G_{M+1}(z)\ \ldots\ G_{2M-1}(z))$$

Based on (7) and the power complementary condition in equ. (3) for perfect reconstruction, the filter bank can be implemented through a parallel bank of 2×1 lossless lattices cascaded by the modulation matrix. The number of the 2×1 lattices is equal to the number of subchannels M. The modulation part can be implemented by fast DCT algorithm. Such an implementation structure has been widely used in the CMFBs with $N=2mM$ [1,4]. The linear-phase property of the prototype filter is not exploited to reduce the complexity.

Let $m_1+M-1=2l_0-l_1$, where l_0 and l_1 are integers, and $l_1\in\{0,1\}$. For l_1 equal to zero, one of m_1 and M is odd, and the other even. For l_1 equal to one, both m_1 and M are odd or even. By using the properties of cosine function, the modulation matrix can be expressed as:

$$\hat{\mathbf{C}}=\sqrt{2M}\mathbf{DC}[\mathbf{A}\Lambda_0-\mathbf{B}\Lambda_1 \quad \mathbf{A}\Lambda_1+\mathbf{B}\Lambda_0] \tag{8}$$

where $\mathbf{D}$ is an $M\times M$ diagonal matrix with the kth diagonal component $[\mathbf{D}]_{k,k}=(-1)^{\lfloor k/2\rfloor}$, and

$$\mathbf{C}=\begin{cases}\mathbf{C}_M^{III}, & l_1=0\\ \mathbf{C}_M^{IV}, & l_1=1\end{cases}$$

$$\mathbf{A}=\begin{cases}\begin{bmatrix}\sqrt{2} & 0\\ 0 & \mathbf{I}_{M-1}\end{bmatrix}, & l_1=0\\ \mathbf{I}_M, & l_1=1,\end{cases} \qquad \mathbf{B}=\begin{cases}-\begin{bmatrix}0 & 0\\ 0 & \mathbf{J}_{M-1}\end{bmatrix}, & l_1=0\\ -\mathbf{J}_M, & l_1=1\end{cases}$$

$$\Lambda_0=\begin{cases}\begin{bmatrix}0 & \mathbf{I}_{M-l_0}\\ 0 & 0\end{bmatrix}, & l_0\le M\\ -\begin{bmatrix}0 & 0\\ \mathbf{I}_{l_0-M} & 0\end{bmatrix}, & l_0>M,\end{cases} \qquad \Lambda_1=\begin{cases}\begin{bmatrix}0 & 0\\ \mathbf{I}_{l_0} & 0\end{bmatrix}, & l_0\le M\\ \begin{bmatrix}0 & \mathbf{I}_{2M-l_0}\\ 0 & 0\end{bmatrix}, & l_0>M.\end{cases}$$

Substituting (8) into (7), we obtain the following expression of $\mathbf{E}(z)$:

$$\mathbf{E}(z)=\mathbf{DCG}(z) \tag{9}$$

where

$$\mathbf{G}(z)=\sqrt{2M}[(\mathbf{A}\Lambda_0-\mathbf{B}\Lambda_1)\mathbf{g}_0(-z^2)+z^{-1}(\mathbf{A}\Lambda_1+\mathbf{B}\Lambda_0)\mathbf{g}_1(-z^2)] \tag{10}$$

Based on this expression, the analysis bank can be implemented more efficiently. The diagonal matrix $\mathbf{D}$ only changes the signs of the output subband signals. The matrix $\mathbf{C}$ is the type III DCT and type IV DCT for l_1 to be zero and one respectively. It can be shown that the $M\times M$ matrix $\mathbf{G}(z)$ can be implemented through a parallel bank of 2×2 lossless lattices, which are related to $\mathbf{L}_k^{(i)}$ defined in Section 2, and some delays. To see this clearly, eight cases can be distinguished as the following:

Case 1: $l_1=0$, $l_0<M$, and $m_1=0$

Case 2: $l_1=0$, $l_0<M$ and $m_1\neq 0$

Case 3: $l_1=0$, $l_0=M$

Case 4: $l_1=0$, $l_0>M$

Case 5: $l_1=1$, $l_0<M$, and $m_1=0$

Case 6: $l_1=1$, $l_0<M$, and $m_1\neq 0$

Case 7: $l_1=1$, $l_0=M$

Case 8: $l_1=1$, $l_0>M$

Table 1 shows the numbers of the 2×2 lattices used in the implementation of $\mathbf{G}(z)$ for the eight different cases. Here we consider Case 1 as an example. In this case, the number of subchannels M must be odd , and l_0 is equal to $(M-1)/2$. $\mathbf{G}(z)$ can be expressed as:

$$\mathbf{G}(z)=\begin{bmatrix}
0 & 0 & 0 & \sqrt{2}G_{l_0} & 0 & 0 & 0\\
0 & 0 & G_{l_0-1} & 0 & G_{l_0+1} & 0 & 0\\
0 & \iddots & 0 & 0 & 0 & \ddots & 0\\
G_0 & 0 & 0 & 0 & 0 & 0 & G_{M-1}\\
z^{-1}G_M & 0 & 0 & 0 & 0 & 0 & -z^{-1}G_{2M-1}\\
0 & \ddots & 0 & 0 & 0 & \iddots & 0\\
0 & 0 & z^{-1}G_{M+l_0-1} & 0 & -z^{-1}G_{M+l_0+1} & 0 & 0
\end{bmatrix} \tag{11}$$

where G_k stands for $\sqrt{2M}G_k(-z^2)$. $\mathbf{G}(z)$ can be implemented in parallel through a set of 2×2 lossless systems:

$$\mathbf{L}_k(z)=\sqrt{2M}\begin{bmatrix}G_k(-z^2) & G_{M+m_1-1-k}(-z^2)\\ z^{-1}G_{M+k}(-z^2) & -z^{-1}G_{2M+m_1-1-k}(-z^2)\end{bmatrix} \tag{12}$$
$$=\Lambda(z)\mathbf{L}_k^{(2)}(-z^2)$$

where $0\le k\le(M-3)/2$, and a delay system $2\sqrt{M}G_{l_0}(-z^2)$.

4. IMPLEMENTATION COMPLEXITY

In the previous section, we have shown that the CMFB can be implemented through a parallel bank of 2×2 lossless lattices and some delays followed by the standard DCT. For the DCTs, fast algorithms are available [7,8]. All the lattices are related to $\mathbf{L}_k^{(i)}$, $i=1,2,3$, without additional multipliers and adders required for the implementation. From Table 1, it is obvious that the total number is less than or equal to $M/2$ for each case. The implementation cost of the CMFB is that of about $M/2$ 2×2 lattices plus one M-point DCT matrix working at M-fold decimated rate.

In the traditional implementation structure of an M-channel CMFB, the number of the 2×1 lattices is M. Ignoring the trivial lattices, which are under *mode c* and *mode d*, the number is exactly twice as that in the new implementation structure. Since only one additional multiplier and three additional adders are required to implement the corresponding 2×2 lattice of a 2×1 one, the complexity of implementing a set of 2×2 lattices is lower than that of doubled 2×1 ones. When the section numbers are large, the complexities of the two type lattices are approached, and hence the cost can be saved nearly one half to implement a set of 2×2 lattices instead of doubled 2×1 lattices.

As an example, we consider Case 5 with $M = 2^m$ to see the efficiency of the new implementation structure. In this case, m_1 is equal to zero. $M/2$ lattices $\mathbf{L}_k^{(2)}(z)$ and type IV DCT are used. By using the fast algorithm presented in [8], $(M/2)\log_2 M + M$ multiplications and $(3M/2)\log_2 M$ additions are required to compute the M-point DCT-IV. The total cost of implementing the CMFB is $M(2m_0 + 3 + \log_2 M)/2$ multiplications and $M(2m_0 + 1 + 3\log_2 M)/2$ additions per M input samples. It is the same cost required by Malvar's ELT with the same length [2]. In the traditional implementation, $M(4m_0 + 2 + \log_2 M)/2$ multiplications and $M(4m_0 + 3\log_2 M)/2$ additions are required for M input samples [1,9]. In Fig.1, we plot the average numbers of the multiplications and additions per input sample (MPIS and APIS) *vers* m_0 with $M = 16$. It can be seen that by using the new implementation structure, the save of operations becomes more significant as m_0 increases.

5. CONCLUSION

A more efficient implementation structure for a class of paraunitary CMFBs with arbitrary length has been developed in this paper. The linear phase property of the prototype filter is exploited to reduce the implementation cost. The new implementation structure uses 2×2 lossless matrices instead of 2×1 ones with the total number of matrices reduced by half. The implementation costs are significantly saved, especially for the CMFB with large ratio of the length to the number of channels. We have also extended this result to a class of arbitrary-length linear-phase CMFBs in [10].

REFERENCES

[1] R. D. Koilpillai, and P. P. Vaidyanathan, "Cosine-modulated FIR filter banks satisfying perfect reconstruction," IEEE Trans. SP, Vol.40, pp. 770-783, Apr. 1992.

[2] H. S. Malvar, "Extended lapped transforms: properties, applications and fast algorithms," IEEE Trans. SP, Vol.40, pp.2703-2714, Nov. 1992.

[3] T. Q. Nguyen, and R. D. Koilpillai, "Theory and design of arbitrary-length cosine-modulated filter banks and wavelets, satisfying perfect reconstruction," IEEE Trans. SP, Vol. 44, No. 3, pp. 473-486, Mar. 1996.

[4] R. A. Gopinath, and C. S. Burrus, "Theory of modulated filter banks and modulated wavelet tight frames," Proc. of IEEE ICASSP'93, Minneapolis, MN, Vol. III, pp.169-172.

[5] G. D. T. Schuller and M. J. Smith, "New framework for modulated perfect reconstruction filter banks," IEEE Trans. SP, Vol. 44, No. 8, Aug. 1996.

[6] H. Xu, W. S. Lu, A. Antoniou, "Efficient iterative design method for cosine-modulated QMF banks," IEEE Trans. SP, Vol. 44, No. 7, pp.1657-1667, June 1996.

[7] Z. D. Wang, "Fast Algorithms for the discrete W transform and for the discrete Fourier transform," IEEE Trans. ASSP, vol.32, pp.803-817, Aug. 1984.

[8] Z. D. Wang, "On computing the discrete Fourier and cosine transforms," IEEE Trans. ASSP, vol.33, pp. 1341-1344, Oct. 1985.

[9] P. P. Vaidyanathn, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice Hall, 1993.

[10] X.-Q Gao, Z.-Y. He and X.-G. Xia, "Efficient implementation of arbitrary-length cosine-modulated filter banks," Submitted to IEEE Trans. SP, 1997.

Table 1. The numbers of lattices in the new implementation structure.

Case	Number of lattices		
	$\mathbf{L}_k^{(1)}$	$\mathbf{L}_k^{(2)}$	$\mathbf{L}_k^{(3)}$
1	0	$(M-1)/2$	0
2	$\lfloor m_1/2 \rfloor$	$(M-m_1-1)/2$	0
3	$\lfloor (M-1)/2 \rfloor$	0	0
4	$M-\lfloor m_1/2 \rfloor$	0	$(m_1-M-1)/2$
5	0	$M/2$	0
6	$\lfloor m_1/2 \rfloor$	$(M-m_1)$	0
7	$\lfloor M/2 \rfloor$	0	0
8	$M-\lfloor m_1/2 \rfloor$	0	$(m_1-M-1)/2$

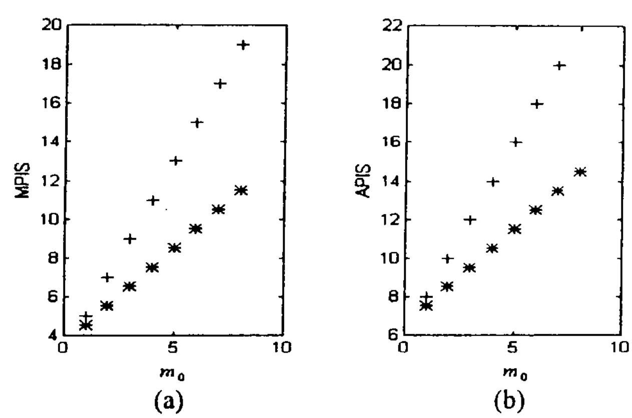

Fig.1. Computational complexity in case 5 with $M=16$: (a). the number of multiplications per input sample; (b). the number of additions per input sample. Here + and ✳ represent the complexities of the traditional and the new implementation structures respectively.