LATTICE DECOMPOSITION OF OVERSAMPLED LINEAR-PHASE PERFECT RECONSTRUCTION FILTERBANKS

Fabrice Labeau

Center for Advanced Systems and Technologies in Communications (SYTA*Com*)
Electrical and Computer Engineering Department
McGill University,3480 University #633, Montreal, QC H3A2A7 Canada.

ABSTRACT

In this paper, we show how to compute the parameters of the lattice implementation of Linear-Phase Perfect Reconstruction Filter Banks (OLPPRFBs) [1]. This lattice implementation is based on a parametrization of the filterbank by a series of left-invertible matrices. It is generic enough to cover orthogonal (Para-unitary) and bi-orthogonal filterbanks, as well as any oversampling factor. This lattice has been used mainly for design purposes, where the parameters defining the left-invertible matrices are varied, and the corresponding filterbank response computed, until a desired frequency response is achieved. In this paper, we aim at recovering the parameter matrices from the impulse response of the analysis (and, for high oversampling ratios, synthesis) filters of the filterbank.

Index Terms— Digital Filtres, Lattice Filters

1. INTRODUCTION

Filterbanks have received a lot of attention in the last 20 years [2], with applications in the areas of signal compression and multicarrier transmission. A typical filterbank is constituted by an analysis stage followed by a synthesis stage in compression and signal analysis applications, whereas this order is inverted for transmission systems. Figure 1 shows a generic setup of an analysis filterbank, with analysis filters $h_i[n]$, $i = 0, \ldots, P-1$, followed by downsamplers by a factor M, yielding subband signals $x_i[m]$. The corresponding synthesis stage will mirror the analysis stage, with upsamplers by M followed by synthesis filters $g_i[n]$, and merging of the different subband branches. In this paper, we will mainly use the polyphase representation of filterbanks [2], where the analysis stage is represented by a $P \times M$ polyphase analysis matrix $\mathbf{E}(z)$, where the $(i,j)^{\text{th}}$ element $E_{i,j}(z)$ is the z-transform of the polyphase component $e_{i,j}[n] \stackrel{\triangle}{=} h_i[nM+j]$, $i = 0, \ldots, P-1$ and $j = 0, \ldots, M-1$ (see figure 2). Similarly [2], the synthesis stage can be represented with an $M \times P$ (type-II) polyphase matrix $\mathbf{R}(z)$, whose $(i,j)^{\text{th}}$ element $R_{i,j}(z)$ is the z-transform of the polyphase component $r_{i,j}[n] \stackrel{\triangle}{=} g_j[(n+1)M-i-1]$, $i = 0, \ldots, M-1$ and $j = 0, \ldots, P-1$. In this case, the filterbank is said to have perfect reconstruction if $\mathbf{R}(z)\mathbf{E}(z) = \mathbf{I}_M$, or at least up to a constant scaling factor and a delay.

This work was partly sponsored by the Canadian National Science and Engineering Research Council (NSERC) under grant number 249699-02.

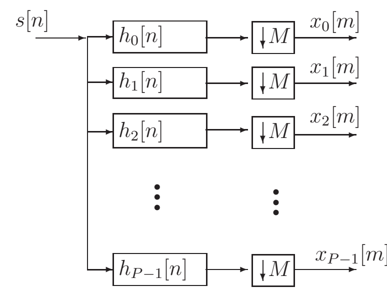

Fig. 1. Generic Oversampled Filterbank System Diagram (analysis only).

In most applications, the filterbanks used are maximally or critically decimated, i.e. the number of subbands P is equal to the decimation factor M. Recently, however, a lot of attention [3–5] has been drawn to overcomplete signal representations, that contain some redundancy. One way to obtain such a representation in a filterbank setting is to use $M \leq P$, i.e. an *oversampled* filterbank. Several reasons motivate the renewed interest surrounding oversampled filterbanks, among which (*i*) an increased design freedom and flexibility [6], since the prefect reconstructions constraints are much less severe on oversampled filterbanks than on maximally decimated ones; in particular, given a set of analysis filters described by a (full normal rank) polyphase matrix $\mathbf{E}(z)$, there are an infinite set of possible synthesis filter sets that will allow for prefect reconstruction in an oversampled case, as compared to a unique perfect reconstruction synthesis filterbank in the maximally decimated case; (*ii*) better quantization noise tolerance, and in particular the capability to carry out quantization noise shaping [7, 8]; and (*iii*) the possibility to use oversampled filterbanks as error-correcting codes acting in the source domain, in particular for mitigating at the same time impulse noise and quantization noise [4, 9].

Based on the above motivations to use oversampled filterbanks, we study in this paper a particular problem related to the design of a particular family of oversampled filterbanks: FIR oversampled linear-phase perfect reconstruction filterbanks (OLPPRFB [1]), which enable perfect reconstruction with linear-phase analysis and synthesis filters. OLPPRFBs have been studied in detail in [1], and their orthogonal (*paraunitary*) counterparts in [10], where some parameterizations have been defined for their polyphase matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$. These parameterizations lead to a matrix version of a lattice implementation of a filter (see [1] for details). The parametrization in [1] is constructive, in the sense that authors show that OLPPRFBs can be constructed by any possible choice of the free parameters in the lattice; this parametrization can thus also be used for design, by optimizing the free parameters against a set of design criteria.

In this paper, we are interested in a reverse factorization: given an OLPPRFB filterbank (characterized by matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$, we aim to find the corresponding lattice parameters. Our main goal here is to be able to map a given set of filters to a given parametrization, with several potential applications(see also section 3): (*i*) *initialization for design*: design using the lattice parametrization in [1] requires a lot of iterations, in particular when the number of free parameters is high (e.g. for high oversampling ratio, $M \ll P$), and the number of iterations is very much dependent on the starting values used; the proposed lattice decomposition can be used to find lattice parameters from a known OLPPRFB with desirable properties, and use these lattice parameters as starting conditions for design; (*ii*) *approximation of a filterbank by the closest OLPPRFB*: when an existing filterbank is not OLPPRFB, but has good spectral properties and the linear phase and perfect reconstruction properties are desirable, one could use the proposed decomposition to try to approximate a given filterbank by the closest OLPPRFB; (*iii*) *synthesis filter design*: given a set of analysis filters, the lattice decomposition proposed here can easily yield a parametrization of all valid linear phase perfect reconstruction synthesis filterbanks, as these synthesis filterbanks are easily generated from the left inverses of the lattice parameters.

The rest of this paper is organized as follows: section 2 will review the existing lattice parametrization from [1] and derive our propose lattice decomposition. Section 3 will review possible applications of the method, and section 4 will outline some future work.

2. TYPE-I FILTERBANKS

The original lattice parametrization [1] lists two types of OLPPRFBs; we will in this paper concentrate on the lattice decomposition of Type-I filterbanks, and will leave the extension to Type-II filterbanks as future work. The case of Type-I OLPPRFBs covers cases where the number of symmetric and antisymmetric filters in the analysis bank is the same, and is

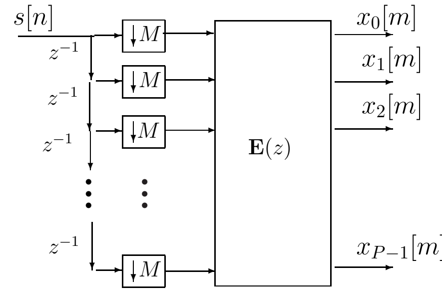

Fig. 2. Generic Oversampled Filterbank – Polyphase Version (analysis only).

thus equal to $P/2$ (so that it assumes an even number of subbands P). In this case, the lattice factorization of an order-$(K-1)$ OLPPRFB in the polyphase domain reads:

$$\mathbf{E}(z) = \prod_{i=K_0}^{1} \mathbf{G}_i(z)\mathbf{E}_0. \tag{1}$$

Each matrix $\mathbf{G}_i(z)$ is given by

$$\mathbf{G}_i(z) = \frac{1}{2}\begin{pmatrix} \mathbf{U}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & z^{-1}\mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{pmatrix}, \tag{2}$$

where all matrices have dimension $P/2 \times P/2$. K_0 is equal to $K-1$. For ease of notation, we will denote by $\mathbf{L}_{2q}$ the matrix $\begin{pmatrix} \mathbf{I}_q & \mathbf{I}_q \\ \mathbf{I}_q & -\mathbf{I}_q \end{pmatrix}$ for any integer q. Similarly $\mathbf{\Lambda}_{2q}(z) = \begin{pmatrix} \mathbf{I}_q & \mathbf{0}_q \\ \mathbf{0}_q & z^{-1}\mathbf{I}_q \end{pmatrix}$. With this notation, $\mathbf{G}_i(z)$ rewrites as $\mathbf{G}_i(z) = \frac{1}{2}\begin{pmatrix} \mathbf{U}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}\mathbf{L}_P\mathbf{\Lambda}_P(z)\mathbf{L}_P$.

Each $\mathbf{G}_i(z)$ propagates the LP and PR properties, while also increasing the degree of the polyphase filter by 1. $\mathbf{E}_0$ itself is a constant $P \times M$ matrix parametrized as

$$\mathbf{E}_0(z) = \frac{1}{\sqrt{2}}\begin{pmatrix} \mathbf{U}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_0 \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{pmatrix}, \tag{3}$$

when $M = 2m$ is even, and

$$\mathbf{E}_0(z) = \frac{1}{\sqrt{2}}\begin{pmatrix} \mathbf{U}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_0 \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \sqrt{2} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & -\mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{pmatrix}, \tag{4}$$

when $M = 2m+1$ is odd.

In order to find the lattice of a given polyphase filter $\mathbf{E}(z)$, one has to determine all the matrices $\mathbf{U}_i$ and $\mathbf{V}_0$.

In order to find the $\mathbf{U}_i$ matrices, one can work backwards, by first determining $\mathbf{U}_{K_0}$, which completely specifies $\mathbf{G}_{K_0}(z)$. From there, by pre-multiplying $\mathbf{E}(z)$ by $\mathbf{G}_i^{-1}(z)$, one gets a lower-order OLPPRFB, on which the same process can be iterated until the order is 0, in which case the iteration stops, and the matrices $\mathbf{U}_0$ and $\mathbf{V}_0$ have to be deduced from the structure of $\mathbf{E}_0$ above.

2.1. Iterations on $\mathbf{G}_i(z)$

One can express the FIR polyphase filter $\mathbf{E}(z) = \sum_{k=0}^{K-1} \mathbf{E}_k z^{-k}$ as

$$\mathbf{E}(z) = \mathbf{G}_{K_0}(z)\mathbf{F}(z), \tag{5}$$

where $\mathbf{F}(z)$ is an order-$(K-2)$ OLPPRFB, written as

$$\mathbf{F}(z) = \sum_{k=0}^{K-2} \mathbf{F}_k z^{-k} = \sum_{k=0}^{K-2} \begin{pmatrix} \mathbf{F}_k^a & \mathbf{F}_k^b \\ \mathbf{F}_k^c & \mathbf{F}_k^d \end{pmatrix} z^{-k}, \tag{6}$$

where each matrix $\mathbf{F}_k$ is partitioned into $P/2 \times P/2$ submatrices.

Writing out the expression of $\mathbf{G}_{K_0}(z)$ from (2), one gets

$$\mathbf{E}(z) = \frac{1}{2} \begin{pmatrix} \mathbf{U}_{K_0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \sum_{k=0}^{K-1} \Bigg\{ z^{-k} \\ \times \begin{pmatrix} \mathbf{F}_k^a + \mathbf{F}_k^c + \mathbf{F}_{k-1}^a - \mathbf{F}_{k-1}^c & \mathbf{F}_k^b + \mathbf{F}_k^d + \mathbf{F}_{k-1}^b - \mathbf{F}_{k-1}^d \\ \mathbf{F}_k^a + \mathbf{F}_k^c - \mathbf{F}_{k-1}^a + \mathbf{F}_{k-1}^c & \mathbf{F}_k^b + \mathbf{F}_k^d - \mathbf{F}_{k-1}^b + \mathbf{F}_{k-1}^d \end{pmatrix} \Bigg\},$$

where all $\mathbf{F}_k^\bullet$ matrices with indices $k < 0$ or $k > K-2$ are assumed to be equal to $\mathbf{0}$.

In particular, equating the coefficients of $z^{-(K-1)}$ in the right and left hand sides of this equation, and partitioning $\mathbf{E}_{K-1}$ as $\begin{pmatrix} \mathbf{E}^a & \mathbf{E}^b \\ \mathbf{E}^c & \mathbf{E}^d \end{pmatrix}$, one gets that

$$\begin{pmatrix} \mathbf{E}^a & \mathbf{E}^b \\ \mathbf{E}^c & \mathbf{E}^d \end{pmatrix} \\ = \begin{pmatrix} \mathbf{U}_{K_0}(\mathbf{F}_{K_0-1}^a - \mathbf{F}_{K_0-1}^c)/2 & \mathbf{U}_{K_0}(\mathbf{F}_{K_0-1}^b - \mathbf{F}_{K_0-1}^d)/2 \\ -(\mathbf{F}_{K_0-1}^a - \mathbf{F}_{K_0-1}^c)/2 & -(\mathbf{F}_{K_0-1}^b - \mathbf{F}_{K_0-1}^d)/2 \end{pmatrix}.$$

Denoting by $\mathbf{U}_i^\dagger$ a left inverse[1] of $\mathbf{U}_i$, one gets that

$$\mathbf{U}_{K_0}^\dagger \mathbf{E}^a = -\mathbf{E}^c \tag{7}$$
$$\mathbf{U}_{K_0}^\dagger \mathbf{E}^b = -\mathbf{E}^d, \tag{8}$$

which can be rewritten as

$$\begin{pmatrix} \mathbf{U}_{K_0}^\dagger & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{E}^a & \mathbf{E}^b \\ \mathbf{E}^c & \mathbf{E}^d \end{pmatrix} = \begin{pmatrix} \mathbf{U}_{K_0}^\dagger & \mathbf{I} \end{pmatrix} \mathbf{E}_{K-1} = \mathbf{0}_{P/2\times M}. \tag{9}$$

It is clear that the rows of the $P/2 \times P$ matrix $\begin{pmatrix} \mathbf{U}_{K_0}^\dagger & \mathbf{I} \end{pmatrix}$ have to form a basis for the left null space of $\mathbf{E}_{K-1}$. In order to find $\mathbf{U}_{K_0}$, one has to find a basis for the left null space of $\mathbf{E}_{K-1}$, and stack these vectors in a matrix $\mathbf{Q}$. Upon partitioning $\mathbf{Q}$ as $\begin{pmatrix} \mathbf{Q}^a & \mathbf{Q}^b \end{pmatrix}$, one can find $\mathbf{U}_{K_0}^\dagger = \mathbf{Q}^{b\dagger}\mathbf{Q}^a$.

In the case where $M \geq P/2$, an easier way to compute $\mathbf{U}_{K_0}^\dagger$ is to simply use a right pseudo-inverse of $\begin{pmatrix} \mathbf{E}^a & \mathbf{E}^b \end{pmatrix}$, and compute

$$\mathbf{U}_{K_0}^\dagger = -\begin{pmatrix} \mathbf{E}^a & \mathbf{E}^b \end{pmatrix}^\dagger \begin{pmatrix} \mathbf{E}^c & \mathbf{E}^d \end{pmatrix}.$$

[1] known to exist as per the definition of $\mathbf{U}_i$

When $M < P/2$, one can resort to a QR decomposition of $\mathbf{E}_{K-1}$ such that

$$\mathbf{Q}\mathbf{E}_{K-1} = \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix},$$

where $\mathbf{R}$ is upper triangular of size $\rho \times M$, where $\rho = \text{rank}\,(\mathbf{E}_{K-1})$. Partitioning $\mathbf{Q}$ as

$$\mathbf{Q} = \begin{pmatrix} \times & \times \\ \mathbf{Q}^a & \mathbf{Q}^b \\ \times & \times \end{pmatrix} \begin{matrix} \updownarrow \rho \\ \updownarrow P/2 \\ \updownarrow P/2-\rho \end{matrix},$$

we have that $\begin{pmatrix} \mathbf{Q}^a & \mathbf{Q}^b \end{pmatrix} \mathbf{E}_{K-1} = \mathbf{0}$, and thus that $\mathbf{U}_{K_0}^\dagger = \mathbf{Q}^{b\dagger}\mathbf{Q}^a$. Note that this only defines $\mathbf{U}_{K_0}^\dagger$ up to a perturbation lying within the left null space of $\begin{pmatrix} \mathbf{E}^{(a)} \mathbf{E}^{(b)} \end{pmatrix}$: denoting by θ the rank of the matrix $\begin{pmatrix} \mathbf{E}^{(a)} \mathbf{E}^{(b)} \end{pmatrix}$, with $\theta \leq M$, then there exists a $(P/2-\theta) \times P/2$ matrix $\mathbf{N}$ whose rows span the left null space of $\begin{pmatrix} \mathbf{E}^{(a)} \mathbf{E}^{(b)} \end{pmatrix}$, i.e. $\mathbf{N} \begin{pmatrix} \mathbf{E}^{(a)} \mathbf{E}^{(b)} \end{pmatrix} = \mathbf{0}_{(P/2-\theta)\times M}$. in this case,it is easy to see that the constraint in (9) is still fulfilled when $\mathbf{U}_{K-0}^\dagger$ is replaced by $\mathbf{U}_{K-0}^\dagger + \mathbf{\Delta N}$, where $\mathbf{\Delta}$ is any $P/2 \times (P/2-\theta)$ matrix without restriction on its rank. This can also be interpreted as the fact that when the downsampling factor M is very low (below $P/2$), then not all the apparent degrees of freedom in $\mathbf{U}_{K-0}$ ($(P/2)^2$) are actually necessary to define the analysis filterbank, as at least $(P/2)(P/2-M)$ degrees of freedom are in $\mathbf{\Delta}$ and can be arbitrarily chosen. It must however be noted that the corresponding values in $\mathbf{\Delta}$ will be fixed once the choice of a particular left inverse $\mathbf{R}(z)$ for $\mathbf{E}(z)$ is made.

Remark 1: One of the advantages of the lattice factorization in [1] with respect to previously known factorizations [10, 11] is that it reduces the number of free parameters by almost fifty percent: previous methods used two matrices for each $\mathbf{G}_i(z)$, namely $\mathbf{U}_i$ and $\mathbf{V}_i$. This would amount in the previous discussion to choosing any possible basis for the left null space of $\mathbf{E}_{K-1}$. This basis is defined up to a $P/2 \times P/2$ invertible matrix, which would amount to choosing $\mathbf{U}_{K_0}^\dagger = \mathbf{Q}^a$ and $\mathbf{V}_{K_0}^\dagger = \mathbf{Q}^b$. The lattice factorization in [1] only accepts one of the infinitely many possible basis for the left null space of $\mathbf{E}_{K-1}$, hence the choice of $\mathbf{U}_{K_0}^\dagger = \mathbf{Q}^{b\dagger}\mathbf{Q}^a$ above.

Remark 2: Once $\mathbf{G}_{K_0}(z)$ is computed as above, the process can be iterated to find $\mathbf{G}_{K_0-1}(z) \cdots \mathbf{G}_1(z)$.

Remark 3: The computation of the left null space of a matrix is easily done through a QR decomposition. This decomposition yields an orthogonal basis, which can be directly used for PU lattices, where the $\mathbf{U}_i$ are orthogonal.

2.2. Parametrization of $\mathbf{E}_0$

For M even, and referring to (3), one can easily find $\mathbf{U}_0$ and $\mathbf{V}_0$ by partitioning $\mathbf{E}_0$ into $\begin{pmatrix} \mathbf{E}^a & \mathbf{E}^b \\ \mathbf{E}^c & \mathbf{E}^d \end{pmatrix}$, where each matrix has size $P/2 \times M/2$. Post-multiplying (3) by $\frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{pmatrix}$, one gets that

$$\mathbf{U}_0 = (\mathbf{E}^a + \mathbf{E}^b\mathbf{J})/\sqrt{2} \quad (10)$$

$$\mathbf{V}_0 = (\mathbf{E}^c - \mathbf{E}^d\mathbf{J})/\sqrt{2}. \quad (11)$$

When M is odd, a slightly modified version of the above holds. Partitioning this time $\mathbf{E}_0$ as $\begin{pmatrix} \mathbf{E}^a & \mathbf{E}^b & \mathbf{E}^c \\ \mathbf{E}^d & \mathbf{E}^e & \mathbf{E}^f \\ \mathbf{E}^g & \mathbf{E}^h & \mathbf{E}^i \end{pmatrix}$, one gets that

$$\mathbf{U}_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathbf{E}^a + \mathbf{E}^c\mathbf{J} & \sqrt{2}\mathbf{E}^b \\ \mathbf{E}^d + \mathbf{E}^f\mathbf{J} & \sqrt{2}\mathbf{E}^e \end{pmatrix} \quad (12)$$

$$\mathbf{V}_0 = (\mathbf{E}^g - \mathbf{E}^i)/\sqrt{2} \quad (13)$$

3. APPLICATIONS

With the above derivations, it is possible, starting from an OLPPRFB, to find a lattice implementation for it, through the knowledge of matrices $\mathbf{U}_i$ and $\mathbf{V}_0$.

Based on the above matrices, it is then possible to easily parametrize all linear-phase inverses for the given analysis bank, as they are all constructed from the many possible left inverses of $\mathbf{U}_i$ and $\mathbf{V}_0$.

It is also interesting to note that when designing an OLPPRFB, such a decomposition can be used as a means to get initial values for the parameters in order to shorten the optimization time, and to avoid local minima during optimization.

4. CONCLUSIONS AND FURTHER WORK

Some further work involves finding a lattice decomposition for filter banks that do not have the linear-phase of perfect reconstruction property. In this case, the above technique would be used to find an OLPPRFB which is close to the given filter bank, and again this would be a very useful step for initialization of design algorithms for OLPPRFBs.

Moreover, work is currently carried out to generalize the above factorization to the case of type-II OLPPRFB as defined in [1], where the building blocks $\mathbf{G}_i(z)$ now have order 2.

5. REFERENCES

[1] L. Gan and K.-K. Ma, "Oversampled linear-phase perfect reconstruction filterbanks: Theory, lattice structure and parametrization," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 744–759, Mar. 2003.

[2] P. P. Vaidyanathan, *Multirate Systems and filter banks*. Englewood Cliffs, N.J.: Prentice-Hall, 1993.

[3] H. Bölcskei and F. Hlawatsch, "Oversampled cosine modulated filter banks with perfect reconstruction," *IEEE Transactions on Circuits and systems–II : Analog and Digital Signal Processing*, vol. 45, no. 8, pp. 1057–1070, Aug. 1998.

[4] F. Labeau, R. Chiang, M. Kieffer, P. Duhamel, and L. Vandendorpe, "Oversampled filter banks as error correcting codes: Theory and impulse noise correction," *IEEE Transactions on Signal Processing*, vol. 53, no. 12, pp. 4619–4630, Dec. 2005.

[5] Z. Cvetković and M. Vetterli, "Oversampled filter banks," *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1245–1255, May 1998.

[6] F. Labeau, "Synthesis filters design for coding gain in oversampled filter banks," *IEEE Signal Processing Lett.*, vol. 12, no. 10, pp. 697–700, Oct. 2005.

[7] H. Bölcskei and F. Hlawatsch, "Oversampled filter banks: Optimal noise shaping, design freedom, and noise analysis," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, Munich, Germany, April 1997, pp. 2453–2456.

[8] T. Leppert, F. Labeau, and P. Kabal, "Performance study of noise-shaping in oversampled filter banks," in *EURASIP European Signal Processing COnference*, Antalya, Turkey, Sept. 2005, p. 4 pp.

[9] G. Rath and C. Guillemot, "Subspace algorithms for error localization with quantized dft codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2115–2124, Dec. 2004.

[10] F. Labeau, L. Vandendorpe, and B. Macq, "Structures, factorizations and design criteria for oversampled paraunitary filter banks yielding linear phase filters," *IEEE Transactions on Signal Processing*, vol. 48, no. 11, pp. 3062–3071, Nov. 2000.

[11] T. Tanaka and Y. Yamashita, "The generalized lapped pseudo-biorthogonal transform," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. II, Orlando, FL, 2002, pp. 1273–1276.