Optimization of Flexible Filter Banks Based on Fast-Convolution

Juha Yli-Kaakinen and Markku Renfors
Dept. of Electronics and Communications Engineering, Tampere University of Technology, Finland
juha.yli-kaakinen@tut.fi and markku.renfors@tut.fi

***Abstract*—Multirate filter banks can be implemented very efficiently using fast-convolution (FC) processing. The main advantage of the FC filter banks compared with the conventional polyphase implementations is their increased flexibility, that is, the number of channels, their bandwidths, and the center frequencies can be independently selected. In this paper, an approach for optimizing the FC filter banks is proposed. First, a subband representation of the adjustable FC filter banks is derived. Then, the optimization problems are formulated with the aid of the subband model. Finally, these problems are conveniently solved with the aid of a nonlinear optimization algorithm. Two examples are included to illustrate the performance of the proposed overall scheme.**

***Index Terms*—Non-uniform filter banks, fast convolution, short-time Fourier transform.**

I. Introduction

This paper focuses on a special implementation scheme for multirate filter banks which is based on fast-convolution (FC) processing. The basic idea of fast convolution is that a high-order filter can be implemented effectively through multiplication in frequency domain, after taking discrete Fourier transforms (DFT) of the input sequence and the filter impulse response. Eventually, the time-domain output is obtained by inverse DFT. Commonly, efficient implementation techniques, like fast Fourier transform/inverse fast Fourier transform (FFT/IFFT), are used for the transforms, and overlap-save processing is adopted for processing long sequences.

The application of FC to multirate filters has be presented in [1], and FC implementations of channelization filters has been considered in [2]–[4]. The authors in [5] have introduced the idea of FC-implementation of nearly perfect-reconstruction filter bank systems in communication signal processing and detailed analysis and FC filter bank optimization methods are developed in [6]. These papers demonstrate the greatly increased flexibility and efficiency of FC filter bank, in comparison with the commonly used polyphase implementation structure.

This papers describes an optimization method for the FC filter banks based on the subband representation. The goal is to optimize the filter bank performance in the minimax sense for the given specifications. In [6], the focus is on communication signal processing whereas in this contribution no specific pulse shaping filtering or waveform processing is applied. It is shown that in this case, the performance of these filter banks is determined by the amount of overlap in overlap-save processing independently of transforms sizes.

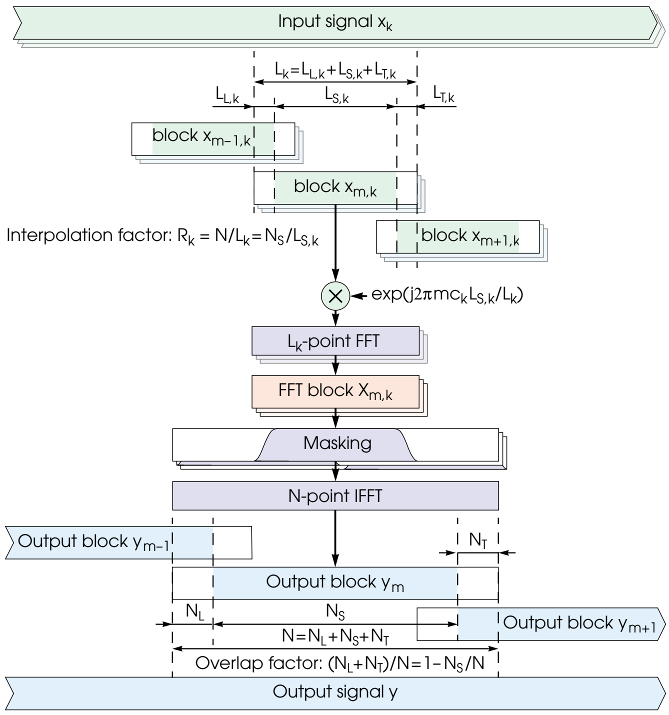

Fig. 1. Block transform representation of the fast-convolution filter bank with overlap-save processing.

This result considerably simplifies the parameterization of the optimization problem at hand.

II. Fast-Convolution Synthesis Filter Bank

In this section the overlap-save processing of fast-convolution synthesis filter bank is first described, then the subband representation of the corresponding filter bank is derived for analysis and optimization purposes.

A. Overlap-Save Processing

The overlap-save processing of synthesis fast-convolution filter bank is illustrated in Fig. 1. We consider a case where K incoming low-rate, narrowband signals $x_k(n)$ for $k = 0, 1, \dots, K-1$ with adjustable frequency responses and with possibly different sampling rates are to be combined into single wideband signal $y(p)$.

In the block transform signal processing, the incoming discrete-time input signals $x_k(n)$'s are first transformed to frequency domain using discrete short-time Fourier transform as given by

$$X_{k,b}(m) = \sum_{n=0}^{L_k-1} x_k(n + mL_{\mathrm{S},k})\theta_k(m)\exp(-\mathrm{j}2\pi bn/L_k) \quad (1)$$

for $b = 0, 1, \ldots, L_k - 1$. Here, L_k is the transform size, b denotes the frequency bin index, the decimated sample time m denotes the frame (or block) index, whereas $L_{\mathrm{S},k}$ is the hop size, i.e., the number of samples between two consecutive frames as depicted in Fig. 1. The phase shift, $\theta_k(m)$, required to concatenate the transformed signal back to time domain is given by (4).

The resulting frequency-domain signals are then shifted in frequency-domain to their desired positions and weighted such that their spectra do not undesirably overlap. The weighted and shifted frequency-domain subband signals are combined in frequency domain and converted back to the time-domain with the aid of inverse short-time Fourier transform. The process of shifting, weighting, and summing the subband signals and then converting the combined signal back to the time domain can be expressed as

$$y_m(r) = \frac{1}{N}\sum_{k=0}^{K-1}\sum_{b=0}^{L_k-1} W_k(b)X_{k,b}(m)\exp(\mathrm{j}2\pi(b + b_k)r/N) \quad (2)$$

for $r = 0, 1, \ldots, N - 1$. Here, N is the size of the inverse transform, b_k is the index of the first bin of the kth subband whereas $W_k(b)$ is the frequency-domain weight function determining the frequency response of the corresponding subband signal.

The output signal $y(p)$ can be obtained from the output blocks $y_m(r)$'s either using the *overlap-add* or *overlap-save* processing. The latter can be expressed as

$$y(p) = \sum_m f(p - mN_\mathrm{S})y_m(p - mN_\mathrm{S}), \quad (3a)$$

where N_S is the number of non-overlapping samples in overlap-save processing (cf. Fig. 1) and

$$f(n) = \begin{cases} 1 & N_\mathrm{L} + 1 \le n \le N_\mathrm{L} + N_\mathrm{S} \\ 0 & \text{otherwise} \end{cases} \quad (3b)$$

is the function for selecting the samples to be concatenated in overlap-save processing.

In order to maintain the phase continuity between consecutive overlapping processing blocks, an additional phase shift as given by

$$\theta_k(m) = \exp(\mathrm{j}2\pi m\Phi_k) \quad \text{with} \quad \Phi_k = c_k L_{\mathrm{S},k}/L_k \quad (4)$$

is required in the synthesis processing of (1) [6]. Here, c_k is the center bin of the kth (bandpass) filter.

In the case of block transform processing, the number of non-overlapping samples is typically determined with the aid of *overlap factor* as given by

$$\lambda = 1 - L_{\mathrm{S},k}/L_k. \quad (5)$$

In order to process input signal with different sampling rates and to utilize the overlap-save processing in forming the output signal, the overlap factor has to be equal for all (forward and inverse) transforms. The number of leading and tailing overlapping samples for the forward transforms (FFT) are defined as

$$L_{\mathrm{L},k} = \lceil \lambda L_k/2 \rceil \quad \text{and} \quad L_{\mathrm{T},k} = \lfloor \lambda L_k/2 \rfloor, \quad (6)$$

respectively, that is, there are $L_{\mathrm{L},k} + L_{\mathrm{T},k} = L_k - L_{\mathrm{S},k}$ overlapping samples between frames. The corresponding notations for the inverse transform are depicted in Fig. 1.

In the synthesis filter bank case, it assumed that the inverse transform size is larger than the forward transform sizes L_k's and, therefore, the above process increses the sampling rate of the subband signal by factors

$$R_k = N/L_k. \quad (7)$$

In other words, the sampling rate conversion factor is determined by the FFT size, and can be configured for each subband individually. Naturally, the FFT size determines the maximum number of frequency bins, i.e., the bandwidth of the subband.

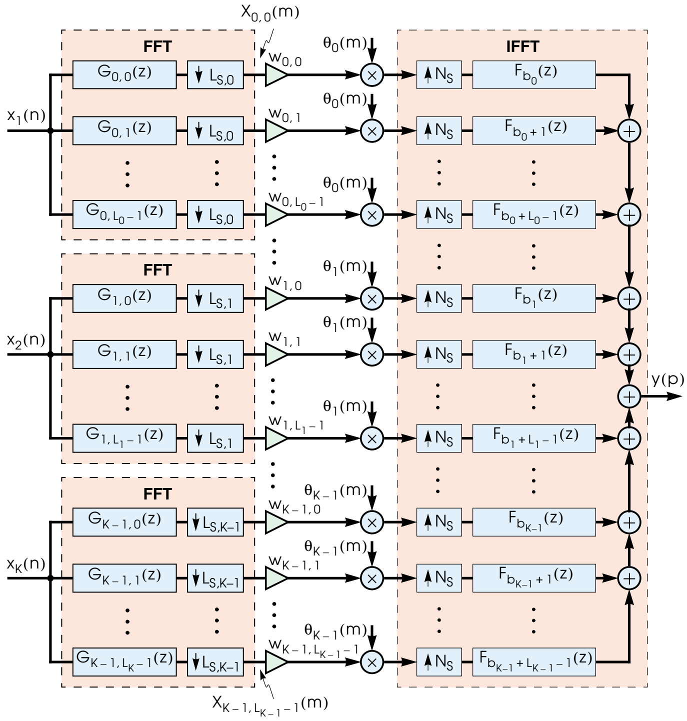

Fig. 2. Subband representation of the FC synthesis filter bank.

B. Subband Representation

Following the well-known duality between the block transform representation and subband representation [7]–[9], the fast-convolution synthesis filter bank can be identically modeled as a subband filter bank as depicted in Fig. 2.

The analysis transfer functions $G_{k,b}(z)$'s can be derived straightforwardly from the analysis equation (1) by first assuming that the hop size is $L_{\mathrm{S},k} = 1$. After substituting $s = n + m$

and $\hat{n} = m$, (1) can be rewritten as

$$X_{k,b}(\hat{n}) = \sum_{s=\hat{n}}^{\hat{n}+L_k-1} x_k(s)\theta_k(\hat{n})\exp(\mathrm{j}2\pi b(\hat{n}-s)/L_k) = \left[\sum_{s=\hat{n}}^{\hat{n}+L_k-1} x_k(s) g_{k,b}(\hat{n}-s)\right]\theta_k(\hat{n}), \quad (8)$$

that is, the analysis can be realized by filtering the sequence $x_k(n)$ by analysis filters

$$g_{k,b}(n) = \exp(\mathrm{j}2\pi bn/L_k) \quad (9)$$

followed by subsequent modulation by $\theta_k(\hat{n})$. Correspondingly, the synthesis filters can written as

$$f_b(n) = \exp(\mathrm{j}2\pi bn/N). \quad (10)$$

Subsampling $X_{k,b}(\hat{n})$ by $L_{\mathrm{S},k} > 1$ results

$$X_{k,b}(L_{\mathrm{S},k}\hat{n}) = \sum_{n=0}^{L_k-1} x_k(n+\hat{n}L_{\mathrm{S},k})\theta_k(L_{\mathrm{S},k}\hat{n})\exp(-\mathrm{j}2\pi bn/L_k) = X_{k,b}(m). \quad (11)$$

Due to the subsampling, there are $L_{\mathrm{S},k}$ different transfer functions for the analysis filters depending on the sampling phase. Consequently, the decimated subband signals can be written as

$$X_{k,b}(z) = \frac{1}{D_k}\sum_{q=0}^{D_k-1} G_{k,b}(z^{1/D_k}\mathrm{e}^{-\mathrm{j}2\pi q/D_k},\Delta)X_k(z^{1/D_k}\mathrm{e}^{-\mathrm{j}2\pi q/D_k}), \quad (12)$$

where $G_{k,b}(z,\Delta)$'s are the transfer functions of the analysis filters for the sampling shift $\Delta = 0,1,\ldots,L_{\mathrm{S},k}-1$ and $D_k = L_{\mathrm{S},k}$ is the decimation factor. The resulting output signal can then be expressed as

$$Y(z) = \sum_{k=0}^{L_k-1}\sum_{b=0}^{L_k-1} W_k(b)X_{k,b}(z^{I_k}\mathrm{e}^{-\mathrm{j}2\pi\Phi_k})F_{b+b_{0,k}}(z), \quad (13)$$

where $F_b(z)$'s are the transfer functions of the synthesis filters, $I_k = N_\mathrm{S}$ is the interpolation factor, and Φ_k is given by (4).

After some manipulations, the frequency response from the kth input with sampling shift Δ to the filter bank output is expressible as

$$H_k(\mathrm{e}^{\mathrm{j}\omega},\Delta) = \frac{1}{ND_k}\sum_{b=0}^{L_k-1} W_k(b)G_{k,b}(\mathrm{e}^{\mathrm{j}\omega},\Delta)F_{b+b_{0,k}}(\mathrm{e}^{\mathrm{j}\omega}), \quad (14a)$$

where

$$G_{k,b}(\mathrm{e}^{\mathrm{j}\omega},\Delta) = \sum_{q=0}^{D_k-1}\sum_{\ell=0}^{L_k-1}\exp(\mathrm{j}2\pi\ell(b+\lceil L_k/2\rceil)]/L_k) \times \exp(\ell+\Delta+L_{\mathrm{L},k})^{\mathrm{j}2\pi(q+\lambda k_0)/D_k} \times \exp(-\mathrm{j}I/D_k[\ell+\Delta+L_{\mathrm{L},k}]\omega) \quad (14b)$$

for $k = 1,2,\ldots,L$ and

$$F_b(\mathrm{e}^{j\omega}) = \sum_{\ell=0}^{N_S-1}\exp(\mathrm{j}2\pi b[N_L+\ell]/N)\exp(-\mathrm{j}\omega\ell) \quad (14c)$$

are the frequency responses of the analysis and synthesis filters, respectively.

III. Prototype Filter Optimization

The goal is to optimize the frequency-domain weights in such a manner that both the passband and stopband ripples of the resulting subchannels are minimized. This can be achieved by minimizing for all the lowpass prototype filters (i.e., filters with different L_k's) independently the following maximum of the normalized error function:

$$\epsilon = \max_{\substack{\omega\in[0,\omega_p]\cup\omega\in[\omega_s,\pi] \\ 0\le\Delta\le L_{\mathrm{L},k}-1}} |G(\omega)[|H_k(\mathrm{e}^{\mathrm{j}\omega},\Delta)| - D(\omega)]|, \quad (15a)$$

where

$$D(\omega) = \begin{cases} 1, & \omega\in[0,\omega_{\mathrm{p},k}] \\ 0, & \omega\in[\omega_{\mathrm{s},k},\pi], \end{cases} \quad (15b)$$

$$G(\omega) = \begin{cases} \delta_s/\delta_p, & \omega\in[0,\omega_{\mathrm{p},k}] \\ 1, & \omega\in[\omega_{\mathrm{s},k},\pi], \end{cases} \quad (15c)$$

and $H_k(\mathrm{e}^{\mathrm{j}\omega},\Delta)$ is given by (14). Here, δ_p and δ_s are the desired passband and stopband ripples, respectively, whereas $\omega_{\mathrm{p},k}$ and $\omega_{\mathrm{s},k}$ are the passband and stopband edge frequencies of kth lowpass prototype filter as given by

$$\omega_{\mathrm{p},k} = \frac{(1-\rho_k)\pi}{2R_k} \quad \text{and} \quad \omega_{\mathrm{s},k} = \frac{\pi}{2R_k}, \quad (16)$$

respectively.

The resulting optimization problems are the following: Given R_k, δ_p, δ_s, $\omega_{\mathrm{p},k}$, and $\omega_{\mathrm{s},k}$ find N, L_k, and λ to minimize the normalized error function as given by (15). These problems can be conveniently solved with the aid of any efficient nonlinear optimization algorithm, e.g., `fminimax` from the optimization toolbox provided by MathWorks, Inc [10].

IV. Implementation Complexity

The number of real multiplications per output sample required for the overall filter bank can be expressed as follows:

$$C_\mathrm{M} = \frac{1}{N(1-\lambda)}\left(C_\mathrm{IFFT} + \sum_{k=0}^{K-1}\left[C_{\mathrm{FFT},k} + 2L_{\mathrm{T},k} + 2L_{\mathrm{S},k}\right]\right). \quad (17)$$

Here, $C_{\mathrm{FFT},k}$ and C_IFFT are the number of real multiplications required to implement the forward and inverse transforms, respectively, whereas $2L_{\mathrm{T},k}$ real multiplications are needed for weighting the complex frequency domain samples by real weight values (cf. Fig. 1) and $2L_{\mathrm{S},k}$ multiplications are required for multiplying the real overlapped input signal blocks by complex phase rotations.[1] In general $L_{\mathrm{T},k} = L_k$, however, the number of multiplications can be reduced by constraining some of the weights to be equal to one as will be described in Section V.

V. Numerical Examples

This section illustrates the overall design scheme and the complexity of the optimized filter bank in terms of two examples.

[1]It should be noted that, in the case of conventional single-band interpolators, the frequency shift is typically equal to zero [$\Theta_k = c_k = 0$ in (4)] and the rightmost-hand term can be excluded from (17).

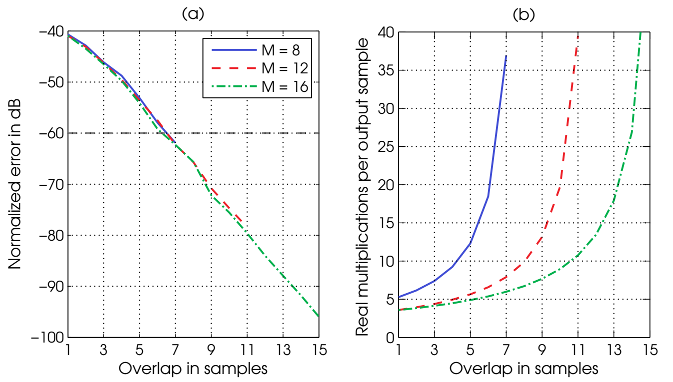

Fig. 3. (a) Normalized error function as a function of overlap in Example 1. (b) The corresponding implementation complexity.

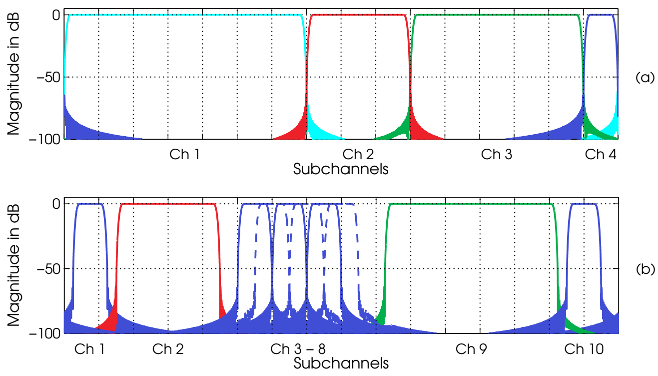

Fig. 4. (a) Magnitude responses of the optimized FC filter bank in Example 2. (b) One of the many other alternative configurations.

A. Example 1

It is desired to design a single channel ($K = 1$) filter bank, i.e., an interpolator with the sampling rate conversion factor of 28/3. The desired passband and stopband ripples are $\delta_s = 100\delta_p = 0.001$ (60-dB stopband attenuation). The roll-off factor is $\rho_1 = 0.1$, i.e., the passband edge frequency is $\omega_p = 0.0856\pi$. The lengths of the forward and inverse transforms can be chosen as $L_0 = 3M$ and $N = 28M$, respectively, where M is an integer. For a given M, the overlap factor can be selected from $\lambda = 1/M, 2/M, \ldots, (M-1)/M$.

Figure 3(a) shows the normalized error, as given by (15), as a function of overlap for the optimized filters for $M = \{8, 12, 16\}$. As can be seen from this figure, the given specification can be met for $M = 8$ with $\lambda = 7/8$, for $M = 12$ with $\lambda = 7/12$, and for $M = 16$ with $\lambda = 7/16$, that is, the frequency-domain performance for the given specifications is characterized by the overlap in samples independently of transforms sizes. However, the number of real multiplications per output sample, as depicted in Fig. 3(b), clearly reduces as the transforms sizes increase and, consequently, the overlap factor decreases. It has been also observed that some of the passband weights can be constrained to be equal to one, that is, only the transition band weights has to be implemented. This further reduces the implementation cost.

For $M = 16$ case, the number of real multiplications required to implement FFT and IFFT are $C_{\mathrm{FFT}} = 96$ and $C_{\mathrm{FFT}} = 1320$, respectively [11]. Using (17), the number or real multiplications per output sample is 5.98 with $L_{\mathrm{T},k} = L_k = 48$ and 5.75 with $L_{\mathrm{T},k} = 18$. The corresponding value for the polyphase realization meeting the same specifications is 9.46, that is, the reduction in computational complexity is at least 36 percent.[2]

B. Example 2

It is desired to design a four channel non-uniform filter bank with $\delta_p = \delta_s = 0.001$, $R_k = 16/\{7, 3, 5, 1\}$, and $\rho_k = 0.2/(R_k/16)$ for $k = 0, 1, 2, 3$ [12]. The specification are met by overlap of 18 samples, that is, the shortest FFT satisfying the specifications is 19. However, in order to reduce the implementation complexity, the FFT length is chosen to be multiple of 32. In this case, the FFT lengths become $L_k = \{224, 96, 160, 32\}$ for $k = 0, 1, 2, 3$ and IFFT length is 512.

Figure 4(a) shows the magnitude responses for the optimized filter bank. The number of real multiplications per output sample for the proposed design is 25.09 with $L_{\mathrm{T},k} = 14$ for $k = 0, 1, 2, 3$. For the filter bank in [12], roughly 35 real multiplications are needed per output sample. For comprehension, Fig. 4(b) shows one of the many alternative configurations being realizable with the same optimized prototype filters.

[2]It should be pointed out that, if it is desired to simultaneously shift the resulting interpolated signal in frequency domain, then the complexity of the polyphase implementation doubles due to the required complex filtering whereas for the fast-convolution based filtering the increase in complexity is from 5.98 to 6.36.

References

[1] M. Borgerding, "Turning overlap-save into a multiband mixing, downsampling filter bank," *IEEE Signal Proc. Mag.*, pp. 158–162, Mar 2006.

[2] M.-L. Boucheret, I. Mortensen, and H. Favaro, "Fast convolution filter banks for satellite payloads with on-board processing," *IEEE J. Select. Areas Commun.*, vol. 17, no. 2, pp. 238–248, Feb. 1999.

[3] C. Zhang and Z. Wang, "A fast frequency domain filter bank realization algorithm," in *Proc. Int. Conf. Signal Processing*, vol. 1, Beijing, China, Aug. 21–25 2000, pp. 130–132.

[4] L. Pucker, "Channelization techniques for software defined radio," in *Proc. Software Defined Radio Technical Conference (SDR'03)*, Orlando, FL, USA, Nov. 18–19 2003.

[5] M. Renfors and f. harris, "Highly adjustable multirate digital filters based on fast convolution," in *Proc. Euro. Conf. Circuit Theory Design*, Linköping, Sweden, Aug. 2011, pp. 9–12.

[6] M. Renfors, J. Yli-Kaakinen, and f. harris, "Analysis and design of efficient and flexible fast-convolution based multirate filter banks," submitted to *IEEE Trans. Signal Processing*, 2013.

[7] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs: NJ: Prentice-Hall, 1983.

[8] D. M. Baylon and J. S. Lim, "Transform/subband analysis and synthesis of signals," Research Laboratory of Electronics, Massachusetts Institute of Technology, Tech. Rep. 559, June 1990.

[9] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*. Prentice Hall, 2001.

[10] T. Coleman, M. A. Branch, and A. Grace, *Optimization Toolbox User's Guide*, The MathWorks, Inc., Jan. 1999, Version 2.

[11] H. V. Sorensen and C. S. Burrus, "Fast DFT and convolution algorithms," in *Handbook for Digital Signal Processing*, S. K. Mitra and J. F. Kaiser, Eds. New York: John Wiley and Sons, Inc, 1993, ch. 8, pp. 491–610.

[12] A. Eghbali, H. Johansson, and P. Löwenborg, "Reconfigurable nonuniform transmultiplexers based on uniform filter banks," in *Proc. IEEE Int. Symp. Circuits Syst.*, Paris, France, May 30–June 20 2010, pp. 2123–2126.