TREIZIÈME COLLOQUE GRETSI - JUAN-LES-PINS DU 16 AU 20 SEPTEMBRE 1991

Fast Realization of Pseudo-Quadrature Mirror Filter Banks

C. Costantini, M. Lazzer, G.A. Mian
University of Padova, Telettra S.p.A., Cefriel

Abstract

Pseudo Quadrature Mirror filter banks are well known in digital signal processing. As the filters have convenient characteristics (in the sense of stopband attenuation and transition band) and quasi-perfect reconstruction is reached, they are very useful in a great number of applications e.g. audio coding. So, there is a need to have an efficient realization of this kind of filter banks. Here is described a realization using the Fast Hartley Transform: this fact results in a drastical reduction of the number of operations so that it is possible to implement in many commercial DSP very large filter banks.

1 Introduction

A class filter banks that allows quasi-perfect reconstruction has been proposed in [1] and [2]. In this paper an efficient implementation for Pseudo Quadrature Mirror Filter Banks that guarantees a completely flat overall response is proposed.

The solution was found fundamental in order to reduce computational effort associated to high quality coding of music according to MASCAM systems, the application which motivated the present work. The analysis bank computations are performed each K time interval, and can be viewed as a linear mapping from a vector with N components (the shift register of the input signal) to a vector with K components (the samples of each of K sub-bands).

Dually, each K time intervals, synthesis bank computations are performed, and consist of another linear mapping from a vector with N components (the last N/K transmitted block samples) to a vector with K components (K successive samples of the reconstructed signal). So we have two different $K \times N$ fixed matrices. The computational complexity reduction is obtained with an opportune factorization of these matrices.

As in classical complex filter banks involving FFT, here the operations are splitted in a filtering part and in a modulation part. Modulation is carried applying a real symmetrical and orthogonel matrix to a K components vector. The corresponding matrix product can be executed using a fast algorithm.

2 Polyphase Representation

Consider a filterbank with K equally spaced channels and critical sampling. The impulse responses of the filters in the analysis stage and in the synthesis stage are, respectively, $h_k(n)$ and $g_k(n)$, for $k = 0, \ldots, K-1$ and $n = 0, \ldots, N-1$ (we suppose N to be an integer multiple of K). Using the poliphase representation [4], the K outputs of the analysis stage are given by:

$$\begin{aligned}\mathbf{Y}(z) &\equiv \begin{bmatrix} Y_0(z) \\ Y_1(z) \\ \vdots \\ Y_{K-1}(z) \end{bmatrix} \\ &= \sum_{q=0}^{\frac{N}{K}-1} \mathbf{H}_q z^{-qK} \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-K+1} \end{bmatrix} X(z) \qquad (1)\end{aligned}$$

where:

$$\mathbf{H}_q = \begin{bmatrix} h_0(qK) & \ldots & h_0(qK+K-1) \\ \vdots & \ldots & \vdots \\ h_{K-1}(qK) & \ldots & h_{K-1}(qK+K-1) \end{bmatrix}$$

The reconstructed signal is given by:

$$\begin{aligned}\hat{X}(z) &= [1, z^{-1}, z^{-2}, \ldots, z^{-K+1}] \\ &\times \sum_{q=0}^{\frac{N}{K}-1} \mathbf{G}_q z^{-qK} \mathbf{Y}(z) \qquad (2)\end{aligned}$$

where:

$$\mathbf{G}_q = \begin{bmatrix} g_0(qK) & \ldots & g_{K-1}(qK) \\ \vdots & \ldots & \vdots \\ g_0(qK+K-1) & \ldots & g_{K-1}(qK+K-1) \end{bmatrix}$$

Eqs. (1) and (2) show that, every K input/output time steps, both the analysis and the synthesis filter bank perform N/K products between a $K*K$ matrix and a vector. This gives a computational complexity of N multiplications and additions per time step.

3 Factorizations

In [1], [2] and [3] the filters that guarantees aliasing cancellation and flat overall response are specified. Let $h(n)$ represent the prototype linear phase impulse response of a K band analysis filter bank. The passband filters are given by:

$$h_k(n) = h(n)m_k(n) \tag{3}$$

where:

$$m_k(n) = \sin\left[\frac{(2k+1)(2n+1-N)}{4K}\pi + \theta_k\right] \tag{4}$$

We choose (see [3]):

$$\theta_k = \frac{2k+1}{4}\pi \tag{5}$$

Substituting (5) into (4) we obtain:

$$m_k(n) = \sin\frac{(2k+1)(2n+1-N+K)}{4K}\pi \tag{6}$$

The synthesis bank passband filters are just time reversed versions of the analysis bank filters. Remembering that $h(n)$ has linear phase the impulse responses are:

$$\begin{aligned} g_k(n) &= h_k(N-1-n) \\ &= h(n)r_k(n) \end{aligned} \tag{7}$$

where:

$$\begin{aligned} r_k(n) &= m_k(N-1-n) \\ &= \sin\frac{(2k+1)(2n+1-N+3K)}{4K}\pi \end{aligned} \tag{8}$$

Two factorizations can be used in order to separate the prototype impulse response from matrices $\mathbf{H}_q$ and $\mathbf{G}_q$

$$\mathbf{H}_q = \mathbf{A}_q\mathbf{D}_q \tag{9}$$

$$\mathbf{G}_q = \mathbf{D}_q\mathbf{B}_q \tag{10}$$

where:

$$\mathbf{D}_q = \mathbf{diag}\left[h(qK), \ldots, h(qK+K-1)\right]$$

$$\mathbf{A}_q = \begin{bmatrix} m_0(qK) & \ldots & m_0(qK+K-1) \\ \vdots & \ldots & \vdots \\ m_{K-1}(qK) & \ldots & m_{K-1}(qK+K-1) \end{bmatrix}$$

$$\mathbf{B}_q = \begin{bmatrix} r_0(qK) & \ldots & r_{K-1}(qK) \\ \vdots & \ldots & \vdots \\ r_0(qK+K-1) & \ldots & r_{K-1}(qK+K-1) \end{bmatrix}$$

Consider the function:

$$s_k(n) \equiv \sin\frac{(2k+1)(2n+1)}{4K}\pi \tag{11}$$

Disregarding the sign, it takes on K different values. Moreover, $m_k(n)$ and $r_k(n)$ are shifted versions of $s_k(n)$ (supposing $N+K$ to be even). In formulas:

$$m_k(n) = s_k(n - \frac{N-K}{2}) \tag{12}$$

$$r_k(n) = s_k(n - \frac{N-3K}{2}) \tag{13}$$

This means that, disregarding the sign, all columns of matrices $\mathbf{A}_q$ are also columns of the matrix $\mathbf{S}$ (symmetrical and orthogonal), defined as:

$$\mathbf{S} \equiv \begin{bmatrix} s_0(0) & \ldots & s_0(K-1) \\ \vdots & \ldots & \vdots \\ s_{K-1}(0) & \ldots & s_{K-1}(K-1) \end{bmatrix} \tag{14}$$

Similarly, disregarding the sign, all rows of matrices $\mathbf{B}_q$ are also rows of the matrix $\mathbf{S}$.

A useful consequence of this facts is that all the columns of the matrix $\mathbf{S}^{-1}\mathbf{A}_q$ have all the components equal to zero except for one that can be 1 or -1. The same property is satisfied by all rows of the matrix $\mathbf{B}_q\mathbf{S}^{-1}$. It follows that $\mathbf{S}^{-1}\mathbf{H}_q$ has only a non-zero component per column, and that $\mathbf{G}_q\mathbf{S}^{-1}$ has only a non-zero component per row. These proprierties suggest a more efficient operating mode of filterbanks instead of (1) and (2):

$$\mathbf{Y}(z) = \mathbf{S}\sum_{q=0}^{\frac{N}{K}-1}\mathbf{S}^{-1}\mathbf{H}_q z^{-qK}\begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-K+1} \end{bmatrix} X(z) \tag{15}$$

$$\begin{aligned} \hat{X}(z) &= [1, z^{-1}, z^{-2}, \ldots, z^{-K+1}] \\ &\times \sum_{q=0}^{\frac{N}{K}-1}\mathbf{G}_q\mathbf{S}^{-1}z^{-qK}\mathbf{S}\mathbf{Y}(z) \end{aligned} \tag{16}$$

In (1), when the matrix $\mathbf{H}_q$ operates on a vector, it takes K^2 multiplications. On the contrary, in (15) we have the matrix $\mathbf{S}^{-1}\mathbf{H}_q$ that needs only K multiplications to operate on a vector.

The same consideration can be repeated for $\mathbf{G}_q$ and $\mathbf{G}_q\mathbf{S}^{-1}$.

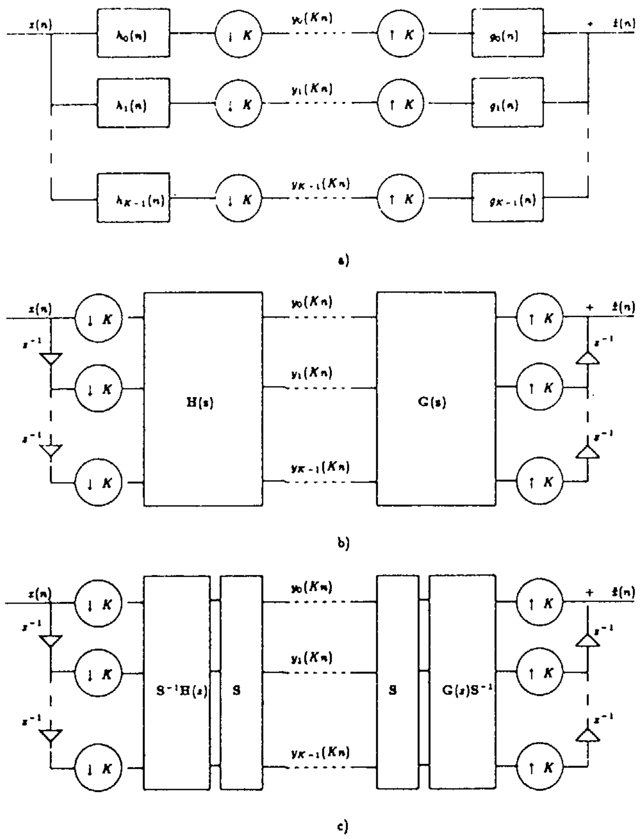

Figure 1: a) direct implementation, b) polyphase implementation, c) fast polyphase implementation

The computational complexity of both stages slows down from N multiplications per input samples to $N/K+K$. This number takes into account that every K time steps a product is performed between the matrix $\mathbf{S}$ and a vector (K^2 multiplications). The computational complexity can be further reduced resorting to a fast algorithm to perform this product.

In the derivation we were supposing $K+N$ to be even, otherwise $r_k(n)$ and $m_k(n)$ are no more shifted versions of $s_k(n)$, as defined in (11). If however $K+N$ is odd, a similar factorization is possible. In this case, the function (11) could be substituted by:

$$s_k(n) \equiv \sin \frac{(2k+1)(n+1)}{2K}\pi$$

4 Fast S-Transform

The product $\mathbf{u} = \mathbf{S}\mathbf{v}$ can be convenientely simplified using a proper factorization of the matrix $\mathbf{S}$. We have:

$$u_k = \sum_{n=0}^{K-1} v_n \sin \frac{(2k+1)(2n+1)}{4K}\pi \qquad (17)$$

Suppose K to be even and define the vector $\mathbf{w}$ as

$$\begin{aligned} w_i &= v_{2i} \\ w_{K-1-i} &= v_{2i+1} \qquad i = 0 \ldots \frac{K}{2} - 1 \end{aligned} \qquad (18)$$

Using (18) in (17) one obtains:

$$\begin{aligned} u_k &= \sum_{n=0}^{K-1} w_n \sin \frac{(2k+1)(4n+1)}{4K}\pi \\ &= \cos \frac{2k+1}{4K}\pi \sum_{i=0}^{K-1} w_i \sin \frac{n(2k+1)}{K}\pi \\ &+ \sin \frac{2k+1}{4K}\pi \sum_{i=0}^{K-1} w_i \cos \frac{n(2k+1)}{K}\pi \end{aligned} \qquad (19)$$

Now consider the following sequence, for $i = 0 \ldots K-1$

$$z_i = w_i \cos \frac{i\pi}{K} + w_{K-i} \sin \frac{i\pi}{K} \qquad (20)$$

The summations in (19) are respectively the imaginary (with the sign changed) and the real part of the DFT of the sequence (20).

Alternatively, the Discrete Hartley Transform [1] of the sequence z_i, named Z_k, can be employed to compute the summations in (19). After some algebra we obtain:

$$\begin{aligned} u_k &= \left(\sin \frac{2k+1}{4K}\pi + \cos \frac{2k+1}{4K}\pi\right) \frac{Z_k}{2} \\ &+ \left(\sin \frac{2k+1}{4K}\pi - \cos \frac{2k+1}{4K}\pi\right) \frac{Z_{K-1-k}}{2} \end{aligned} \qquad (21)$$

In order to resume, the computation of the product $\mathbf{u} = \mathbf{S}\mathbf{v}$ can be efficiently decomposed into four steps:

- Perform the permutation defined in (18).
- Apply the rotation given by (20), ($2K$ multiplications).
- Perform Fast Hartley Transform of the sequence (20) ($K\log_2 K$ multiplications if K is a power of two).
- Apply the rotation described in (21) ($2K$ multiplications).

As an example of applications we consider a filter bank that can be used in MASCAM coding of sound signal. A reasonable choice for the number of sub-bands and the prototype filter length can be $K = 64$ and $N = 512$. In this case both analysis and synthesis filter bank need to execute only 18 multiplications per input/output time step.

[1] The Discrete Hartley Transform of a sequence x_i is defined as

$$X_k \equiv \sum_{i=0}^{K-1} x_i \left(\cos \frac{2\pi i k}{K} + \sin \frac{2\pi i k}{K}\right) \qquad k = 0 \ldots K-1$$

5 Conclusion

In the poliphase realization proposed both analysis and synthesis filter banks computations are decomposed into K inner products of length N/K and a product of a vector by an orthogonal symmetrical matrix. If K is a power of two the total number of multiplications involved per input sample is $N/K + 4 + \log_2 K$. Otherwise this number increase sligthly and a prime factor algorithm implementing FHT is needed [6].

The effective additions are only those used to perform FHT (about $\frac{3}{2}K\log_2 K$), because the inner products can be efficiently computed with MAC instruction (Multiply and Accumulate), available in many commercial DSP.

Addressing is very simple and allows very compact source code. All computation can be done in place, with no additional memory storage.

References

[1] J. H. Rothweiler, "Polyphase quadrature filters - A new sub-band coding technique," *in Proc. ICASSP 83*, pp. 1280-1283

[2] P. L. Chu, "Quadrature mirror filter design for an arbitrary number of equal bandwidth channels," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 203-218, Feb. 1985.

[3] R. V. Cox, "The design of uniformly and nonuniformly spaced pseudoquadrature mirror filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1090-1096, Oct. 1986.

[4] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks and applications: a tutorial," *Proc. IEEE, vol.78*, N. 1, Jan. 1990.

[5] C. P. Kwong and K. P. Shiu, "Structured fast Hartley transform algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1000-1002, Aug. 1986.

[6] H. V. Sorensen, D. L. Jones, C. S. Burrus, and M. T. Heideman, "On computing the discrete Hartley transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1231-1238, Oct. 1985.