BALANCING MULTIWAVELETS

Thèse n$^{\underline{o}}$ 2192 (2000)

Présentée le 31 mai 2000
au Département de Systèmes de Communication

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

pour l'obtention du grade de Docteur ès Sciences
par

Jérôme LEBRUN
Ingénieur mathématicien diplômé EPF
de nationalité française

Acceptée sur proposition du jury

Prof. J. Rosenthal, président du jury
Prof. M. Vetterli, directeur de thèse
Dr. T. Blu, rapporteur
Prof. A. Derighetti, rapporteur
Prof. I. Selesnick, rapporteur
Prof. G. Strang, rapporteur
Prof. M. Unser, rapporteur

Lausanne, EPFL, 2000

About balancing order and other discrete-time properties of multiwavelets

Jérôme Lebrun

May 4, 2000

Abstract

Multiwavelets are a recent generalization of wavelets where one allows the multiresolution analysis to be generated by a finite number of scaling functions instead of only one so as to overcome some classical limitations in the design of filter banks preventing us from constructing non trivial orthogonal, compactly supported wavelets with symmetries (for example using fractal interpolating functions). The new theory of multiwavelets have yielded new possibilities in the design but also new limitations.

Focusing on the limitations, we showed that some new conditions have to be imposed in the design of multiwavelets in order to obtain multifilter banks that are easily usable for processing scalar signals. We defined this way the concept of *balanced* multiwavelets which is now also investigated by other research groups. We worked also further in that direction and generalized what was already done about balanced multiwavelets to the preservation of polynomial signals of higher degree, calling thus the property *high order balancing*. A thorough study of the relations with other discrete-time properties of multifilters has been achieved. In particular, balancing has been shown to be equivalent to a special case of Plonka factorization of the refinement mask and some Strang-Fix conditions on a time-varying scalar subdivision operator. These equivalences turn out to be key results for the construction of balanced multifilter banks.

By giving a special attention to the iteration of multifilter banks, we derived new results making connections between balancing order and properties of the associated multiresolution analysis (approximation power, moments of the scaling functions, superfunction theory). These results strengthen the role of the balancing concept in the theory of multiwavelets. Besides, it also lead us to generalize the concept of Coiflets and introduce multiCoiflets. A family of orthogonal, compactly supported, symmetric multiCoiflets has then been constructed. Besides, we investigated the influence of ergodic properties (zeros at pre-periodic points of invariant cycles) on the smoothness of multiwavelets. This brought us to introduce the new concept of *balanced smoothness* of a multifilter bank. We also proved that the shortest length orthonormal balanced multifilter banks are in fact constructed from multiplexed Daubechies filters.

Résumé

Les méthodes *ondelettes* ont eu un impact important dans la théorie du signal et ses applications. Introduites récemment comme généralisation des ondelettes, les multiondelettes sont construites à partir d'une famille finie de fonctions d'échelle au lieu d'une seule. Cette approche permet de contourner certaines limitations liées aux ondelettes et de construire ansi des analyses multirésolution orthogonales, non triviales, à partir de fonctions d'échelle symétriques et à support compact, comme par exemple, en utilisant des fonctions fractales interpolantes. Cependant, de par l'aspect vectoriel des multiondelettes, de nouveaux problèmes et de nouvelles limitations surgissent.

Par l'étude précise de la structure vectorielle des bancs de multifiltres, nous avons ainsi montré qu'il était nécessaire d'introduire de nouvelles conditions dans la conception des multiondelettes afin d'obtenir des bancs de multifiltres utilisables pour le traitement de signaux scalaires. Par celà, nous avons défini le concept de multiondelettes balancées, concept qui a été depuis repris par d'autre groupes de chercheurs. Ensuite, en généralisant cette approche à des signaux polynomiaux de dégrés supérieurs, nous avons étendu le concept sous le nom de balancement d'ordre supérieur. Une étude précise des relations avec d'autres propriétés discrètes des bancs de multifiltres à été accomplie. En particulier, nous avons montré que la propriété de balancement était équivalente d'une part à une forme spéciale, particulièrement simple, de factorisation de Plonka du masque associé au multifiltre passe-bas de synthèse, ainsi que d'autre part à des conditions de type Strang-Fix sur un opérateur de subdivision scalaire variant dans le temps. Ces différents résultats d'équivalence se sont avérés particulièrement utiles lors de la construction de familles de multiondelettes balancées.

Des résultats liant l'ordre de balancement à certaines propriétés de l'analyse multirésolution associée (ordre d'approximation, moments des fonctions d'échelle, superfonction associée) ont été démontrés. Ils renforcent l'importance du concept de balancement dans la théorie des multiondelettes. D'autre part, ceci nous a aussi amené à généraliser le concept de Coiflettes pour introduire les multiCoiflettes. Il a été possible de construire une famille de multiCoiflettes symétriques, orthogonales avec support compact. L'analyse de l'aspect ergodiques de la régularité des multifiltres a aussi entraîné l'introduction d'un nouveau concept de régularité balancée. Enfin, nous avons prouvé la minimalité des filtres de Daubechies multiplexés comme bancs de multifiltres balancés orthogonaux.

Contents

Introduction

In recent years, similar techniques developed in different fields, namely wavelets in applied mathematics, subband coding in digital signal processing and multiresolution in computer vision have converged to form a unified theory.

In the mathematical community, the first kick that would lead much later to the general concept of wavelet bases was given by Haar [1910] with the construction of an orthogonal basis of $L^2(\mathbb{R})$ generated by the dyadic translations and dilations: $\psi_{n,k}(t) := \psi(2^n t - k)$ of a single function $\psi(t)$ defined by

$$\psi(t) := 1_{[0,\frac{1}{2}[}(t) - 1_{[\frac{1}{2},1]}(t). \tag{1}$$

Occasionally, other bases of this kind were constructed, for example by Stromberg [1982] using infinitely supported piecewise polynomials. However, it took more than seventy years for the concept of the wavelet transform to really hatch out. At the beginning of the eighties, Grossmann and Morlet [1984], motivated by the analysis of seismic signals, were looking for alternatives to the classical Fourier and Gabor transform methods. They got the idea of replacing the modulation operation in these transforms by a dilation operation. Doing so, they were able to construct adaptative time-frequency tilings allowing a precise analysis of the transient parts in their signals. Starting from a normalized function $\psi \in L^2$ with well-behaved Fourier transform

$$C_\psi := \int |\omega|^{-1} |\widehat{\psi}(\omega)|^2 d\omega < \infty, \tag{2}$$

they introduced the family of functions

$$\psi_{a,b}(t) := \sqrt{a}\, \psi(\tfrac{t-b}{a}). \tag{3}$$

They were able to construct a transform (called the wavelet transform) by

$$\mathscr{W} f(a,b) := \int f(t) \psi_{a,b}(t) dt \tag{4}$$

with the following reconstruction formula

$$f(t) = \tfrac{1}{C_\psi} \int_0^\infty \int \mathscr{W} f(a,b) \psi_{a,b}(t) \tfrac{da db}{a^2}. \tag{5}$$

This new formula soon triggered a great interest among mathematicians: Meyer [1985-1986] quickly developed the theory of wavelets.

At the same time, motivated by new problems linked to the recent digital revolution, some members of the signal processing community were investigating efficient ways of decomposing signals into lowpass and highpass components at half the rate of the input signal (so as to keep the same amount of data) in such a way that it is possible to exactly reconstruct the input signal from these components. This new subject of interest called subband coding with multirate filter banks became a hot topic when Croisier et al. [1976] showed it was possible to construct filter banks with aliasing cancellation using quadrature mirror filters and simple downsampling and upsampling operations. However, their design had some limitations: the only quadrature mirror filter (QMF) with finite impulse response is the Haar filter. Smith and Barnwell [1984] and Mintzer [1985] were finally able to overcome this limitation by introducing the conjugate quadrature filters (CQF). It was now possible to construct perfect reconstruction filter banks with orthogonal FIR filters. The subject was completed by the introduction of the biorthogonality equations by Vetterli [1986] and the general theory of paraunitary matrices by Vaidyanathan [1987].

Concurrently, under the impulse of the pyramid algorithm devised by Burt and Adelson [1983], similar ideas were investigated in the computer vision and computer graphics communities. In much the same direction, Dubuc [1986]; Deslauriers and Dubuc [1989]; Cavaretta et al. [1991] were developing a related approach with the concept of subdivision schemes for numerical analysis purposes.

It is always a difficult task to give credit to someone in particular and it is especially true in a field like wavelets. However, most people of the wavelet community agree that the success of wavelets in now so (too?) many fields is mostly due to a small bunch of *free thinkers*, always interested in learning what's going on in other fields and what it could bring to their own and finally how to convince (almost,...) everybody to rethink the old problems they were working on for years, usually without much success, in the new framework. One could say that the wavelet domain was born from this cross-fertilization: Daubechies [1988], from the mathematical physics community used tools from signal processing to design her now omnipresent wavelets; Mallat [1989] from the computer vision world formalized (with the interaction of Meyer) the concept of multiresolution analysis from which he derived his famous efficient algorithm; Strang [1989] from numerical analysis made the link with approximation theory by showing the importance of the well-known Strang and Fix [1973] conditions for wavelets; Vetterli and Herley [1990] promoted the wavelet paradigm in filter bank design, convincing most of the signal processing community to forget about the classical theory of filter and to go instead "continuous-time". This is just a short list and of course, many others had an almost as important influence. It would be unfair not to mention them: obviously, Cohen in the math community, Unser and Vaidyanathan for signal processing, well and so many others... .

Following the signal processing approach, we will give a simple introduction to wavelets. After presenting a rigorous but simple framework for multirate signal processing, we will introduce filter banks based on conjugate quadrature filters. Iteration of the filter bank on the lowpass analysis generates discrete-time wavelet bases. To the limit, we will end up with wavelet bases and the concept of multiresolution analysis. It is from this starting point that we will highlight the need of introducing multiwavelets as a way to overcome some limitation in the design of CQF. The goal of this part is to develop the intuition on what will be done rigorously in the next chapter. Readers interested in a more detailed presentation of filter bank and wavelet theory are referred to the classical books of Daubechies [1992]; Vaidyanathan [1993]; Vetterli and Kovačević [1995]; Strang and Nguyen [1996]; Mallat [1998].

Discrete-time signal processing

Because it is so easy to deal with convolutions and Fourier transforms in them, the most convenient setting for signal processing are indeed the spaces of distributions. We will not recall here the theory of distribution, but just the usual notations. Writing $\mathscr{D} := C_0^\infty(\mathbb{R}, \mathbb{C})$ for the space of test function, $\mathscr{D}'$ is the space of distributions (linear forms on $\mathscr{D}$) with its usual topology of pointwise convergence. Let $\mathscr{S}$ be the space of rapidly decreasing functions and $\mathscr{S}'$ its dual, the space of tempered distributions. As usual, δ is the Dirac distribution, τ the translation operator, (defined on functions by $\tau_h f(t) := f(t-h)$) and $\delta_n := \tau_n \delta$. We write $\check{f}(t) := f(-t)$ for the time-reversal operator and define the Fourier transform on $\mathscr{S}'$ by duality [Rudin, 1991] from its definition for $f \in \mathscr{S}$,

$$\widehat{f}(\omega) := \int f(t) e^{-j\omega t} dt. \tag{6}$$

We will call *signal* any distribution of the form

$$x = \sum_n \mathsf{x}[n] \delta_{n\alpha}, \tag{7}$$

where $\{\mathsf{x}[n]\}_{n\in\mathbb{Z}}$ is a sequence of complex numbers and $\alpha > 0$ gives the rate of the signal. Signals are then distributions with discrete shift invariant support. We introduce the canonical space of signals

$$\varsigma_0\mathscr{D}' := \{x \in \mathscr{D}' \,|\, x = \sum_n \mathsf{x}[n]\delta_n\} \tag{8}$$

endowed with the topology of pointwise convergence induced from $\mathscr{D}'$. There is a natural injection of the space $\varsigma_0\mathscr{D}'$ into the space $\mathbb{C}^\mathbb{Z}$ of complex sequences. We will often use this property. So as to discriminate easily, bona-fide signals will be however noted in script-style x and sequence in sans-serif x.

Our working spaces X will be subsets of $\varsigma_0\mathscr{D}'$ endowed with topologies at least as fine as the topology induced by $\mathscr{D}'$. They should contain δ and be invariant by translations $\tau X = X$. Typically, we will consider

- $\varsigma_0\mathscr{S}' := \varsigma_0\mathscr{D}' \cap \mathscr{S}'$, the space of tempered signals.
- $\varsigma_0\ell^p := \{x = \sum_n \mathsf{x}[n]\delta_n \mid \mathsf{x} \in \ell^p\}$, the space of ℓ^p signals. This space is the distributional version of the classical ℓ^p.
- $\varsigma_0\mathscr{S} := \{x = \sum_n \mathsf{x}[n]\delta_n \mid \forall N \geq 0,\ \sup_{k\leq N}\sup_n (1+n^2)^N |\Delta^k \mathsf{x}[n]| < \infty\}$ where Δ is the finite difference operator. This is the space of rapidly decreasing signals.
- $\varsigma_0\mathscr{E}' := \varsigma_0\mathscr{D}' \cap \mathscr{E}'$, the space of signals with finite support, usually called finite impulse response (FIR) signals. This space is clearly isomorphic to c^{oo} the space of sequences with finite support.

We clearly have

$$\varsigma_0\mathscr{E}' \subset \varsigma_0\mathscr{S} \subset \varsigma_0\ell^p \subset \varsigma_0\mathscr{S}' \subset \varsigma_0\mathscr{D}'. \tag{9}$$

In the rest of the text, unless mentioned, signals will be assumed to belong to the space $\varsigma_0\mathscr{S}'$ since the two most important operations of signal processing, convolution and Fourier transform, are well-defined in this space. Namely, given a rapidly decreasing signal $h \in \varsigma_0\mathscr{S}$, we define the *filter* T_h as the convolution operator

$$\begin{aligned} T_h : \varsigma_0\mathscr{S}' &\to \varsigma_0\mathscr{S}' \\ x &\mapsto T_h x := h * x. \end{aligned} \tag{10}$$

With $y = \sum_n \mathsf{y}[n]\delta_n = h * x$, we have $y \in \varsigma_0\mathscr{S}'$ and $\mathsf{y}[n] = \sum_k \mathsf{h}[n-k]\mathsf{x}[k]$. It is easy to see that $T_h \in \mathscr{L}(\varsigma_0\mathscr{S}')$ (space of continuous linear operators on $\varsigma_0\mathscr{S}'$) and that it commutes with τ. By abuse, we will also call h the filter.

For a signal x, we denote its Fourier transform by $\widehat{x}(\omega)$ and usually call it the spectrum of the signal. In the case h is a filter, we call $\widehat{h}(\omega)$ the frequency response of the filter. Since $h \in \varsigma_0\mathscr{S}$, $\widehat{h}(\omega)$ is a well defined 2π-periodic function. We say that h is lowpass iff $|\widehat{h}(\omega)| \approx 1$ for ω around 0 and $\widehat{h}(\omega) \approx 0$ for ω around π (highpass is the opposite). Since, $\widehat{h}(\omega)$ is a function, we can write without ambiguity

$$\widehat{h * x}(\omega) = \widehat{h}(\omega)\widehat{x}(\omega). \tag{11}$$

For $x \in \varsigma_0\mathscr{S}'$, we have that $\widehat{x}(\omega) = \sum_n \mathsf{x}[n]e^{-j\omega n}$. The spectrum of a tempered signal usually exists only as a distribution. Now, for $z \in \mathbb{T}$, we introduce the z-transform of x by

$$X(z) := \sum_n \mathsf{x}[n]z^{-n}. \tag{12}$$

We clearly have $X(e^{j\omega}) = \widehat{x}(\omega)$. This formulation emphasizes the 2π-periodicity of $\widehat{x}$. $y = h*x$ gives in the z-domain $Y(z) = H(z)X(z)$. Again, in most cases, these z-transforms only exist as distributions. The multiplication is well-defined because $H(z)$ is a function. We can also easily define convolution (and so filtering) in the following common cases:

- For $h \in \varsigma_0\ell^1$ and $x \in \varsigma_0\ell^\infty$, we get $h * x \in \varsigma_0\ell^\infty$ (this property is called bounded-input, bounded-output stability).
- For $h, x \in \varsigma_0\ell^2$, we have $h * x \in \varsigma_0\ell^2$.

A filter h is said to have linear phase iff there exist α, β such that for $|\omega| < \pi$, we can write

$$H(e^{j\omega}) = |H(e^{j\omega})| e^{-j(\alpha\omega+\beta)}. \tag{13}$$

This property implies nice symmetries on the filter that enable a clean processing of finite length signals [Strang and Nguyen, 1996; Oppenheim et al., 1999]: by symmetric extension of the signal, we are able to keep the same length for the output of the filter. This is particularly important in applications like compression.
For more details about the distributional approach of signal processing and distributions in general, the reader is referred to [Gasquet and Witomski, 1999].

Along with filters, downsampling and upsampling operators are the building blocks of multi-rate signal processing. Given a sequence $\mathsf{x} = \{\mathsf{x}[n]\}_n$, we introduce the operator $(\downarrow M)$ of downsampling by M

$$\mathsf{y}[n] = ((\downarrow M)x)[n] := \mathsf{x}[nM]. \tag{14}$$

This gives in z-domain,

$$Y(z) = \sum_n \mathsf{x}[nM] z^{-n} = \tfrac{1}{M} \sum_{k=0}^{M-1} X(e^{jk\frac{2\pi}{M}} z^{\frac{1}{M}}). \tag{15}$$

By extension, we write $Y(z) = (\downarrow M)X(z)$. The components $X(e^{jk\frac{2\pi}{M}} e^{j\frac{\omega}{M}}) = \widehat{x}(\frac{\omega}{M} + k\frac{2\pi}{M})$ introduce a frequency folding for $k \neq 0 \pmod M$. It creates aliasing in the spectrum $\widehat{y}(\omega)$ of the output signal y.

Inversely, we introduce upsampling by M as an expansion with insertion of zeros in the sequence $\{\mathsf{x}[n]\}_n$

$$\mathsf{y}[n] = ((\uparrow M)\mathsf{x})[n] := \begin{cases} \mathsf{x}[k] & \text{for} \quad n = kM \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

This gives in the z-domain

$$Y(z) = \sum_k \mathsf{x}[k] z^{-kM} = X(z^M). \tag{17}$$

By extension, we write $Y(z) = (\uparrow M)X(z)$. The operators $(\downarrow M)$ and $(\uparrow M)$ are linear and they satisfy: $(\downarrow M)(\uparrow M) = I, (\uparrow M)(\downarrow M) \neq I$. Furthermore, $(\downarrow M)\tau_M = \tau(\downarrow M)$ and

$(\uparrow M)\tau = \tau_M(\downarrow M)$. These last properties, called $M-1$ and $1-M$ shift invariance lead to the introduction of scaled spaces of signals. For $M = 2$, we define for $N \in \mathbb{Z}$

$$\varsigma_N \mathscr{D}' := \{x \in \mathscr{D}' \,|\, x = \sum_k x[k]\delta_{k2^{-N}}\}, \tag{18}$$

where N is called the scale of the signal space. It gives us a nested sequence of scaled spaces

$$\ldots \subset \varsigma_{-N}\mathscr{S}' \subset \varsigma_{-N+1}\mathscr{S}' \subset \ldots \subset \varsigma_0\mathscr{S}' \subset \varsigma_1\mathscr{S}' \subset \ldots \subset \mathscr{S}'. \tag{19}$$

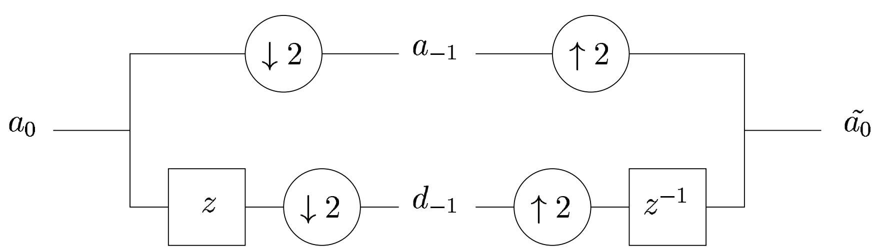
We define in the same way the scaled versions of our working spaces. In this context, $(\downarrow 2)$ is the canonical downscale projector $(\downarrow 2) : \varsigma_N\mathscr{D}' \to \varsigma_{N-1}\mathscr{D}'$ and $(\uparrow 2)$ the canonical upscale injection $(\uparrow 2) : \varsigma_N\mathscr{D}' \to \varsigma_{N+1}\mathscr{D}'$.

We also easily derive the following very useful identities [Vaidyanathan, 1993]:

$$\begin{aligned} \text{First noble identity :} \quad & (\downarrow M)H(z^M) = H(z)(\downarrow M). \\ \text{Second noble identity :} \quad & H(z^M)(\uparrow M) = (\uparrow M)H(z). \end{aligned} \tag{20}$$

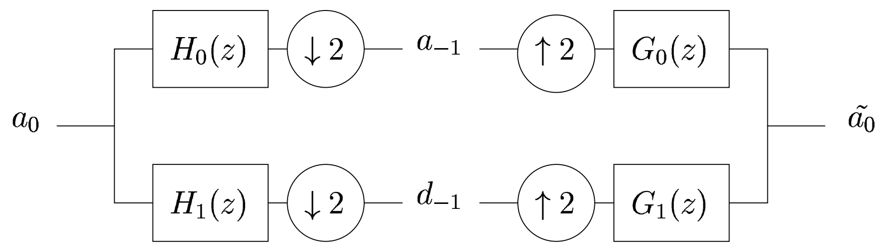
Filter banks

For simplicity, we will require all signals and filters to be real valued. Using only downsampling/upsampling operators and delays, one can construct a very simple system performing a decomposition of a signal $a_0 \in \varsigma_0\mathscr{S}'$ into two components $a_{-1}, d_{-1} \in \varsigma_{-1}\mathscr{S}'$ from which it is possible to reconstruct a_0.



This system is called the *lazy* filter bank. Apart from showing that multirate signal processing is possible, this filter bank has little interest in applications since the two signals a_{-1} and d_{-1} share up to a modulation the same spectrum $\widehat{d}_{-1}(\omega) = e^{-j\frac{\omega}{2}}\widehat{a}_{-1}(\omega)$.

Obviously, we would like the filter bank to perform a subband decomposition [Crochiere et al., 1976] of the input signal in the sense that a_{-1} and d_{-1} have more or less disjoint spectrums. By adding some filtering in the structure of what will be called from now on a filter bank, we get



Consequently, a two-channel multirate filter bank first convolves the input signal a_0 with a lowpass filter h_0 and a highpass filter h_1 to minimizing aliasing and then downsamples these two signals. This step is called the analysis

$$a_{-1} = (\downarrow 2) h_0 * a_0 \quad \text{and} \quad d_{-1} = (\downarrow 2) h_1 * a_0. \tag{21}$$

An output signal is then reconstructed by upsampling a_{-1} and d_{-1} and filtering again with a lowpass filter g_0 and a highpass filter g_1 to reject the out-of-band components in the spectrum. The synthesis is given by

$$\tilde{a_0} := g_0 * ((\uparrow 2) a_{-1}) + g_1 * ((\uparrow 2) d_{-1}). \tag{22}$$

We require that this system leaves a_0 unchanged: $\tilde{a_0} = a_0$. Vetterli [1986] gave the necessary and sufficient conditions in the z-domain for perfect reconstruction

$$\begin{cases} G_0(z)H_0(z) + G_1(z)H_1(z) & = 2 \\ G_0(z)H_0(-z) + G_1(z)H_1(-z) & = 0. \end{cases} \tag{23}$$

We get that g_0 and g_1 are uniquely determined from h_0 and h_1 by rewriting the previous equations

$$\begin{bmatrix} H_0(z) & H_1(z) \\ H_0(-z) & H_1(-z) \end{bmatrix} \begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}. \tag{24}$$

Introducing $D(z) := H_0(z)H_1(-z) - H_0(-z)H_1(z)$, supposed to be non vanishing on $\mathbb{T}$, we get

$$G_0(z) = \tfrac{2}{D(z)} H_1(-z) \quad \text{and} \quad G_1(z) = -\tfrac{2}{D(z)} H_0(-z). \tag{25}$$

Now, if we require further that all filters are FIR, then only two choices are possible for $D(z)$. Namely,

Quadrature mirror filters: $D(z) = 2$.

This gives $G_0(z) = H_1(-z)$ and $G_1(z) = -H_0(-z)$. Croisier et al. [1976] additionally imposed h_0 and h_1 to be mirror filters ($H_1(z) := H_0(-z)$), we then get

$$H_0^2(z) - H_0^2(-z) = 2. \tag{26}$$

The solutions of this equation are naturally called quadrature mirror filters (QMF). Unfortunately, the only FIR QMF is the Haar filter $H_0(z) = 1 + z^{-1}$. The interest of these filters is then limited.

Conjugated quadrature filters: $D(z) = 2z^{-1}$.

We get $G_0(z) = zH_1(-z)$ and $G_1(z) = -zH_0(-z)$. Smith and Barnwell [1984] and Mintzer [1985] were able to overcome the major limitation of QMF by imposing h_0 and h_1 to be conjugated quadrature filters: $H_1(z) := z^{-1}H_0(-z^{-1})$. We get

$$H_0^2(z) + H_0^2(-z) = 2. \tag{27}$$

With this slight change, FIR solutions are now possible. As we will see, these filters are closely linked to wavelets.

Wavelets

Assuming an FIR CQF filter bank, the decomposition of a signal in this filter bank may be interpreted as its expansion in an orthonormal basis of ℓ^2. Namely, we have

$$a_{-1}[n] = \sum_k h_0[2n-k]a_0[k] \quad \text{and} \quad d_{-1}[n] = \sum_k h_1[2n-k]a_0[k] \tag{28}$$

and

$$a_0[n] = \sum_l a_{-1}[l]g_0[n-2l] + \sum_l d_{-1}[l]g_1[n-2l]. \tag{29}$$

From the properties of CQF filter bank, $h_0[2n-k] = g_0[n-2k]$ and $h_1[2n-k] = g_1[n-2k]$. Introducing

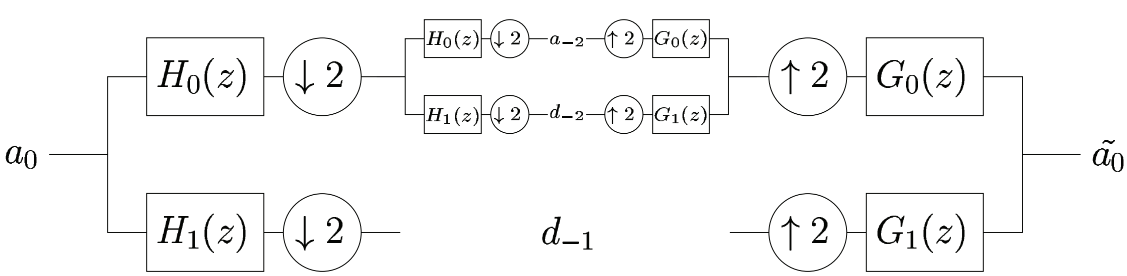
$$s_{-1,n}[k] := g_0[n-2k] \quad \text{and} \quad w_{-1,n}[k] := g_1[n-2k], \tag{30}$$

the decomposition can then be rewritten

$$a_0[n] = \sum_l \langle a_0, s_{-1,l}\rangle s_{-1,l}[n] + \sum_l \langle a_0, w_{-1,l}\rangle w_{-1,l}[n]. \tag{31}$$

The perfect reconstruction and CQF conditions ensure furthermore that $\{s_{-1,l}, w_{-1,l}\}_l$ is an orthogonal set of vectors. We thus have an orthogonal basis of ℓ^2. This result can be generalized to the case of IIR (i.e. non FIR) filter banks [Evangelista, 1989; Herley and Vetterli, 1993]. Also, if the filters are not imposed to be CQF anymore, we get biorthogonal bases of ℓ^2 [Vetterli and Herley, 1992].

Iterating on the lowpass signal a_{-1}, we get coarser approximations of the input signal.



The idea of filter bank trees is to cascade this iteration up to a certain level K. We then have $K+1$ signals: the coarse signal a_{-K} and the details signals $d_{-K}, \ldots, d_{-1}$. And we can exactly reconstruct a_0 from these signals. Now, if we omit some details signals in the reconstruction (this is the principle of compression), the "quality" of the signal reconstructed will depend a lot on the "smoothness" of the basis vectors we are reconstructing with. Hence the need of studying the "smoothness" of the filter bank [Mallat, 1989; Daubechies, 1992; Rioul, 1992, 1993a,b]. Using the noble identities (20) and following Daubechies [1988] ideas, we introduce the iterated filters:

$$G_0^{(K)}(z) := \prod_{k=0}^{K-1} G_0(z^{2^k}) \quad \text{and} \quad G_1^{(K)}(z) := G_1(z^{2^{K-1}}) \prod_{k=0}^{K-2} G_0(z^{2^k}). \tag{32}$$

Then, introducing

$$s_{-K,n}[k] := g_0^{(K)}[n - k2^K] \quad \text{and} \quad w_{-K,n}[k] := g_1^{(K)}[n - k2^K], \tag{33}$$

we get that $\{s_{-K,l}, w_{-j,l}\}_{1 \le j \le K, l \in \mathbb{Z}}$ is an orthogonal basis of ℓ^2. Furthermore, defining $V_{-k} := \text{span}\{s_{-j,l} \,|\, l \in \mathbb{Z}\}$ and $W_{-k} := \text{span}\{w_{-j,l} \,|\, l \in \mathbb{Z}\}$, we get a nested sequence of subspaces

$$V_{-K} \subset V_{-K+1} \subset V_{-1} \subset V_0 \tag{34}$$

with $V_{-k+1} = V_{-k} \overset{\perp}{\oplus} W_{-k}$. This structure is called a discrete-time multiresolution analysis (dMRA) of ℓ^2. Now, in order to study the quality of the multiresolution analysis, we introduce

$$\varphi^{(k)} := \sum_l g_0^{(k)}[l]\delta_{l2^{-k}} \quad \text{and} \quad \psi^{(k)} := \sum_l g_1^{(k)}[k]\delta_{l2^{-k}}. \tag{35}$$

Taking the Fourier transforms, we get that

$$\widehat{\varphi}^{(k)}(\omega) = \prod_{l=1}^{k} \widehat{g}_0(\tfrac{\omega}{2^l}) \quad \text{and} \quad \widehat{\psi}^{(k)}(\omega) = \widehat{g}_1(\tfrac{\omega}{2})\widehat{\varphi}^{(k)}(\tfrac{\omega}{2}). \tag{36}$$

It is then enough to study the convergence of $\varphi^{(k)}$ as $k \to \infty$. Under the condition $\widehat{g}_0(\omega) = 1$, $\varphi^{(k)}$ converge to a distribution φ that satisfies

$$\hat{\varphi}(2\omega) = \widehat{g_0}(\omega)\hat{\varphi}(\omega). \tag{37}$$

We derive the convergence of $\psi^{(k)}$ to the distribution given by

$$\hat{\psi}(2\omega) := \widehat{h_1}(\omega)\hat{\varphi}(\omega). \tag{38}$$

Now, if furthermore $\inf_{|\omega| < \frac{\pi}{3}} |\widehat{g}_0(\omega)| > 0$ [Cohen, 1992], then the convergence is in L^2 norm to bona-fide L^2 function. $\varphi(t)$ and $\psi(t)$ satisfy the following two-scale equations [Mallat, 1989; Strang, 1989]

$$\varphi(t) = 2\sum_k g_0[k]\varphi(2t - k) \quad \text{and} \quad \psi(t) = 2\sum_k g_1[k]\varphi(2t - k). \tag{39}$$

In that case, these two functions generate a multiresolution analysis of L^2 as defined by Mallat [1989]. Defining $V_k := \text{span}\{\varphi(2^k t - n) \,|\, n \in \mathbb{Z}\}$, we get by the two-scale equations, a nested sequence of subspaces of L^2 satisfying

- $V_n \subset V_{n+1}$.
- $\cap_n V_n = \{0\}$ and $\overline{\cup_n V_n} = L^2$.
- $f \in V_n \Leftrightarrow f(2t) \in V_{n+1}$.
- $f \in V_0 \Leftrightarrow f(t-k) \in V_0, \forall k \in \mathbb{Z}$.
- $\{\varphi(t-k) \,|\, k \in \mathbb{Z}\}$ is an orthonormal basis of V_0.

Introducing $W_k := \text{span}\{\psi(2^k t - n) \,|\, n \in \mathbb{Z}\}$, we get $V_{n+1} = V_n \overset{\perp}{\oplus} W_n$ and so $\oplus_n W_n = L^2$. We thus prove that $\{\psi(2^k t - n) \,|\, k, n \in \mathbb{Z}\}$ is an orthogonal basis of L^2. Starting from a CQF, we have constructed a basis of L^2 from dyadic dilations and translations of a single function. ψ is called an orthogonal wavelet, φ is called the associated scaling function. Now, as we will detail in next section, the "smoothness" of the filter bank when iterated is linked to the number of zeros at π of $\widehat{g_0}(\omega)$.

This way of constructing wavelets from iterated filter banks was pioneered by Daubechies [Daubechies, 1988]. It became since a standard way to derive orthogonal and biorthogonal wavelet bases. The underlying CQF filter banks are now well-studied, the design procedure is well-understood. By the structure of the problem, certain solutions are however ruled out: it is impossible to design FIR linear-phase conjugated quadrature filter with real coefficients other than the Haar filter. This implies that the only orthogonal wavelet with compact support and symmetry is the Haar wavelet

Multiwavelets

Generalizing the wavelet case, one can allow a multiresolution analysis $\{V_n\}_{n\in\mathbb{Z}}$ of $L^2(\mathbb{R})$ to be generated by a finite number of scaling functions $\phi_0(t), \phi_1(t), \ldots, \phi_{r-1}(t)$ and their integer translates (this finitely generated multiresolution analysis is then said to be of *multiplicity* r). In this framework, the multiscaling function $\boldsymbol{\phi}(t) := [\phi_0(t), \ldots, \phi_{r-1}(t)]^\top$ satisfies a two-scale equation

$$\boldsymbol{\phi}(t) = \sum_k \mathbf{M}[k]\boldsymbol{\phi}(2t-k) \tag{40}$$

where now $\{\mathbf{M}[k]\}_k$ is a sequence of $r \times r$ matrices of real coefficients. The multiresolution analysis structure gives $V_1 = V_0 \oplus W_0$ where W_0 is the orthogonal complement of V_0 in V_1. Again, starting from the orthonormal basis, $\phi_0(t), \phi_1(t), \ldots, \phi_{r-1}(t)$ and their integer translates, we can construct an orthonormal basis of W_0 generated by $\psi_0(t), \psi_1(t), \dots, \psi_{r-1}(t)$ and their integer translates with $\boldsymbol{\psi}(t) := [\psi_0(t), \dots, \psi_{r-1}(t)]^\top$ derived by

$$\boldsymbol{\psi}(t) := \sum_k \mathbf{N}[k]\boldsymbol{\phi}(2t-k) \tag{41}$$

where $\{\mathbf{N}[k]\}_k$ is a sequence of $r \times r$ matrices of real coefficients obtained by *completion* of $\{\mathbf{M}[k]\}_k$ (a detailed exposition of the completion scheme is given in [Lawton et al., 1996]). For obvious reasons, $\boldsymbol{\psi}(t)$ is called a multiwavelet.

With this approach, one is finally able to overcome some of the limitations of CQF filter banks. It is now possible to get finitely generated multiresolution analysis with all the scaling functions and wavelets orthogonal, compactly supported and (anti)symmetric.

The first multiwavelets were designed by Alpert [1993] and Hervé [1994]; Goodman and Lee [1994]. Their method was completely different from the afore mentioned filter bank approach for wavelets. Namely, they used techniques similar to the ones used in numerical analysis (finite elements and splines methods). In Alpert [1993], the scaling functions are r polynomials of degree $r-1$ supported on $[0,1]$. The simplest one is given by $\phi_0(t) = 1$ and $\phi_1(t) = \sqrt{3}(1-2t)$ for $0 \leq t \leq 1$.

Using similar methods, actually fractal interpolation, Geronimo et al. [1994] built a multiresolution analysis having approximation of order 2 (1 and t can be reconstructed from the translates of the scaling functions) using two symmetric, compactly supported, orthogonal scaling functions that are furthermore Lipschitz. Their outstanding achievement triggered many attempts to construct new multiwavelet bases [Vetterli and Strang, 1994; Strang and Strela, 1995; Donovan et al., 1996; Chui and Lian, 1996] and motivated a thorough study of the theory of multiwavelets [Turcajová and Kautsky, 1995; Heil et al., 1996; Cohen et al., 1997; Plonka, 1997; Plonka and Strela, 1998].

Consider a finitely generated multiresolution analysis with orthonormal multiscaling function $\boldsymbol{\phi}(t)$ and multiwavelet $\boldsymbol{\psi}(t)$. For $\mathbf{s}(t) \in V_0$, we have

$$\mathbf{s}(t) = \sum_n \boldsymbol{s}_0^\top[n]\boldsymbol{\phi}(t-n) \tag{42}$$

then from $V_0 = V_{-1} \overset{\perp}{\oplus} W_{-1}$, we get

$$\mathbf{s}(t) = \sum_n \boldsymbol{s}_{-1}^\top[n]\boldsymbol{\phi}(\frac{t}{2}-n) + \boldsymbol{d}_{-1}^\top[n]\boldsymbol{\psi}(\frac{t}{2}-n). \tag{43}$$

We derive the well-known Mallat [1989] algorithm for multiwavelets. For the analysis step

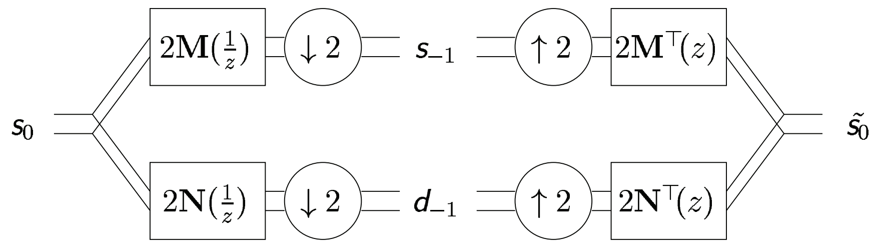
$$\boldsymbol{s}_{-1}[n] = \sum_k \mathbf{M}[k-2n]\boldsymbol{s}_0[k] \tag{44}$$

$$\boldsymbol{d}_{-1}[n] = \sum_k \mathbf{N}[k-2n]\boldsymbol{s}_0[k] \tag{45}$$

and for the synthesis, we get

$$s_0[n] = \sum_k \mathbf{M}^\top[n-2k]s_{-1}[k] + \mathbf{N}^\top[n-2k]d_{-1}[k]. \tag{46}$$

These relations enable us to construct a multi-input multi-output filter bank (multifilter bank) as shown below.



Because of their inherent vector nature, in order to process scalar signal, multifilter banks require a vectorization of the input signal to produce an new r-dimensional input signal. Introduced by Evangelista [1993]; Herley and Vetterli [1994]; Vetterli and Strang [1994], a simple way to do this vectorization is to split scalar signals into their polyphase components. Introducing

$$\begin{bmatrix} m_0(z) \\ m_1(z) \\ \vdots \\ m_{r-1}(z) \end{bmatrix} := 2\mathbf{M}(z^r) \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(r-1)} \end{bmatrix} \tag{47}$$

and in the same way $n_0(z), n_1(z), \ldots, n_{r-1}(z)$, the system can then be rewritten as a $2r$ channel time-varying filter bank (Fig. 1 for the case $r = 2$). Intuitively, this is a filter bank with relaxed requirements on the time invariance. In each filtering block, we periodically alternate between different filters. In [Lebrun and Vetterli, 1997a, 1998a] we first pointed out that, if the components $m_0(z), m_1(z), \ldots, m_{r-1}(z)$ of the lowpass branch have different spectral behavior, e.g. lowpass behavior for one, highpass for another, this will lead to unbalanced channels that will mix the coarse resolution signal and details coefficients and will create strong oscillations in the reconstructed signal.

Clearly, this problem relates to the basic principle of filter banks: one expects a reasonable class of smooth signals to be preserved by the lowpass branch and canceled by the highpass one. In the wavelet case, the two important issues of the reproduction of polynomials by the associated multiresolution analysis (approximation theory issue) and the preservation/cancellation of discrete-time polynomial signals by the associated filter bank (subband coding and compression issue) are tightly connected since they have been proved to be equivalent to the same condition: the number of zeros at π in the factorization of the lowpass filter $H_0(e^{j\omega})$ of the filter bank. In the orthogonal case, the lowpass filter $H_0(e^{j\omega})$ is said to be of *regularity* N iff any of the following equivalent conditions [Daubechies, 1992] hold:

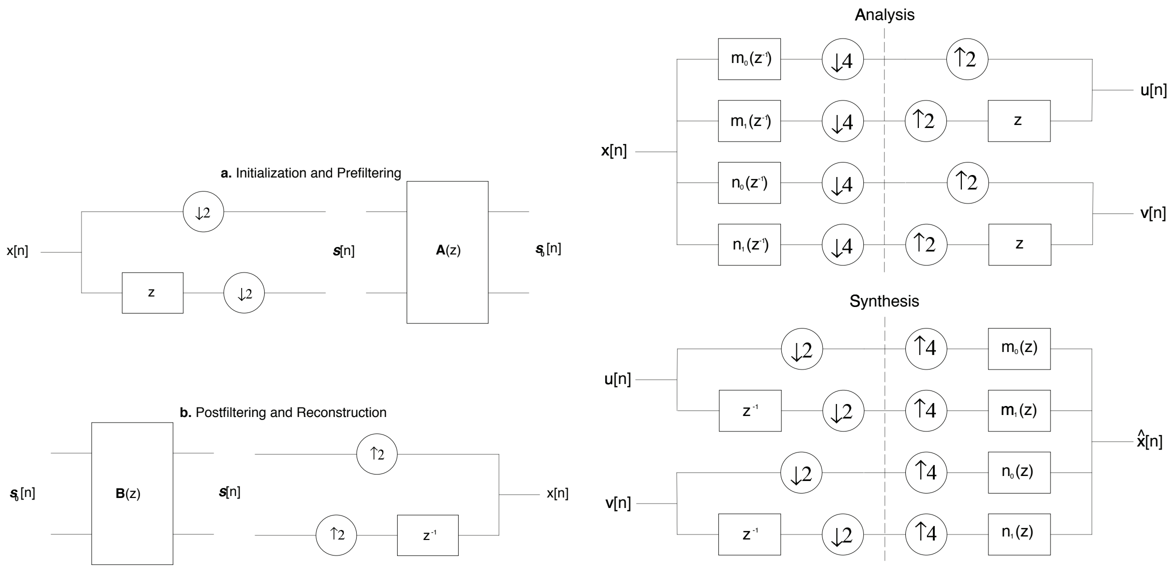


Figure 1: Left: general vectorization and folding for a multifilter bank, Right: multifilter bank seen as a time-varying filter bank.

- The lowpass filter $H_0(e^{j\omega})$ has a zero of order N at $\omega = \pi$.
- The corresponding highpass filter $H_1(e^{j\omega})$ has a zero of order N at $\omega = 0$.
- Discrete-time polynomial sequences of degree $n < N$ are perfectly represented by shifts of the scaling sequence $h_0[k]$ (i.e. the discrete-time polynomial signals are preserved by the lowpass branch of the filter bank).
- The associated wavelet function $\psi(t)$ has N vanishing moments.
- Continuous-time polynomials of degree $n < N$ are perfectly represented by shifts of the scaling function $\phi(t)$ (i.e. the associated multiresolution analysis has approximation order N).

Also, the smoothness of the scaling function $\phi(t)$ (and of the wavelet $\psi(t)$ if the filters are FIR) is governed in a certain sense [Eirola, 1992] by the regularity of the lowpass filter. Of course, similar relations also hold for the biorthogonal case.

The regularity issue is however different for multiwavelets. In [Lebrun and Vetterli, 1998a,b], interested in the subband coding issue in general and the problem of processing one dimensional signals with multiwavelets in particular, we showed that the approximation order property did not guarantee the preservation of discrete-time polynomial signals by the lowpass branch of the filter bank. Consequently, we introduced the concept of balanced multiwavelets which is the main subject of study of this dissertation. Our approach was also further investigated by other authors in several other papers [Jiang, 1998; Rieder and Nossek, 1997; Selesnick, 1998].

Outline

We will give here a short outline of the dissertation.

In Chapter 1, we generalize the usual scalar filter banks in order to overcome some of the limitations in the design that were mentioned above. To do so, we first recall and analyze the fundamental properties of the operators involved in scalar filter banks. Generalizing these properties for vector-valued signals, we develop a theory of vector filter banks. The fundamentals of multifilter banks theory are developed with a special highlight on the connection to time-varying scalar filter banks. Later on, a very special attention is given to the processing of scalar signals within this structure. This leads us to one of the major concepts developed in this dissertation: the property of balanced multifilter banks. We then link balancing to other discrete-time properties of multifilter banks. In particular, balancing is shown to be equivalent to a special case of Plonka factorization of the refinement mask and various versions of the Strang-Fix conditions. These equivalences are key results in the construction of balanced multifilter banks.

In chapter 2, we give special attention to the process of iterating the multifilter bank. We look at the cascading of the multifilter bank on the lowpass branch and show that under some natural conditions, it leads to a function of L_r^2 (called the multiscaling function) that generates a multiresolution analysis (MRA) of L^2 by the functional equation it satisfies. By looking closely at the multiscaling function and the MRA, a lot can then be said about some other properties of the multifilter bank: regularity, smoothness and interpolating properties are the most important ones. These different continuous-time properties are related to the balancing order of the multifilter bank. This forces us to reconsider some of these notions in the framework of multiwavelets. A new concept of discrete-time smoothness is thus introduced for multifilter bank. We conclude this chapter by introducing multiCoiflets as a special case of balanced multiwavelets.

In chapter 3, we get *practical* and show how to construct high order balanced multifilters with different useful properties. We first emphasize the limitations of straight design (modifying existing unbalanced multifilters, adapting complex filters) and show some surprising result on the shortest length refinement masks leading to orthonormal multiwavelets of multiplicity $r = 2$ and balancing order p which happen to be multiplexed Daubechies filters. We then give a digest of the hardcore algebra we need to analyze and solve the systems of polynomial equations that we face in the design of families of high order balanced multifilters. Given these tools, we design several families of balanced orthonormal compactly supported multiwavelets with nice symmetries. In this framework, a special attention is also given on the conditions leading to higher smoothness. We also construct a family of orthonormal symmetric multiCoiflets. Finally, we detail an application of high order balanced multifilters to image coding.

Chapter 1

A Toeplitz approach to multifilter banks

In this chapter, we will generalize the usual scalar filter banks in order to overcome some of the limitations in the design that were mentioned in the introduction chapter. To do so, we will first recall and analyze the fundamental properties of the operators involved in scalar filter banks. Generalizing these properties for vector-valued signals, we will develop a theory of vector filter banks (multifilter banks) inspired by the works of Strang [1989]; Shensa [1992]; Rioul [1993a]; Aldroubi et al. [1994]; Turcajová and Kautsky [1995]. Later on, a very special attention will be given to the processing of scalar signals within this structure. This will lead us to one of the major concepts developed in this dissertation: the property of balanced multifilter banks.

The basic idea of the system called filter bank is to decompose an input signal (usually $x \in \ell^\infty$) into several signals of *smaller* size. Here, we will decompose into two signals, each of these signals being roughly half the size and amount of information of the input signal. The first will represent a coarser version of the input signal. The second signal will be the details. It will also be roughly half the size of the input signal so as to maintain the same total number of samples. This first part is called the analysis bank. The second part of the filter banks, called the synthesis bank, reconstructs the signal from the coarse version and the details.

A natural way to formalize the filter bank structure and to generalize it to vector-valued signals is then to define a multifilter bank $\mathbb{M}$ by the two operators:

The analysis bank: $\mathbb{A} := \left[\begin{smallmatrix} A_0 \\ A_1 \end{smallmatrix}\right]$, where $A_0, A_1 \in \mathscr{L}(\ell_r^\infty)$, i.e. $\mathbb{A} \in \mathscr{L}(\ell_r^\infty, \ell_r^\infty \times \ell_r^\infty)$. Furthermore, we will require that c_r^{oo} is invariant by A_0 and A_1 (i.e. finite length signals remain finite length). Finally, we also require that shifts are downscaled by $\mathbb{A}$ i.e. $\mathbb{A}\boldsymbol{\tau}_2 = \boldsymbol{\tau}\mathbb{A}$ (i.e. $A_i\boldsymbol{\tau}_2 = \boldsymbol{\tau}A_i$ for $i = 0, 1$). This last condition says that $\mathbb{A}$ is $2-1$ shift invariant.

The synthesis bank: $\mathbb{S} := [S_0\, S_1] \in \mathscr{L}(\ell_r^\infty \times \ell_r^\infty, \ell_r^\infty)$. We also require c_r^{oo} to be invariant and that $\mathbb{S}$ is $1-2$ shift invariant (i.e. it upscales shifts): $\mathbb{S}\boldsymbol{\tau} = \boldsymbol{\tau}_2\mathbb{S}$.

We then require that:

- We can reconstruct the input signal from the analysis signals: $\mathbb{S}\mathbb{A} = S_0A_0 + S_1A_1 = I$. This condition is called *perfect reconstruction* (usually abbreviated PR).

- We also impose S_0 and S_1 to be one-to-one operators and that the two branches $\mathsf{P}_i := \mathsf{S}_i\mathsf{A}_i$, for $i = 0, 1$ are bona-fide projectors of $\mathscr{L}(\ell_r^\infty)$: $\mathsf{P}_i^2 = \mathsf{P}_i$ for $i = 1, 2$. This condition will enable us to construct a multiresolution analysis of ℓ_r^2. Namely, assuming PR condition, taking $V_0 := \ell_r^2$ and defining recursively for $n \leq 0$, the spaces $V_{n-1} := \mathsf{P}_0(V_n)$ and $W_{n-1} := \mathsf{P}_1(V_n)$, we construct a multiresolution of ℓ_r^2. This condition (abbreviated dMRA for discrete multiresolution analysis) together with the PR condition imply, as we will see later, the biorthogonality of the multifilter bank: $\mathbb{AS} = \mathbb{I} = \left[\begin{smallmatrix} I & O \\ O & I \end{smallmatrix}\right]$.

Now, since c_r^{oo} is invariant by all the operators, and c_r^{oo} is dense in ℓ_r^2, the analysis and synthesis operators can also be restricted to ℓ_r^2. We then define the adjoint banks $\mathbb{A}^*$ and $\mathbb{S}^*$. It is then easily seen that $\mathbb{A}^*$ is a synthesis bank and $\mathbb{S}^*$ is an analysis bank, and thus that analysis and synthesis are adjoint operators. Consequently, it is equivalent to study either analysis or synthesis banks. Here, we will take the usual approach of studying synthesis operators since it is more intuitive. A multifilter bank with analysis bank $\mathbb{A}$ and synthesis bank $\mathbb{S}$ will be written $\mathbb{M} = (\mathbb{A}, \mathbb{S})$.

1.1 Discrete-time operators

In the following, we will show that the simple requirements introduced above completely characterize the analysis and synthesis banks of $\mathbb{M}$.

1.1.1 Toeplitz operators

An operator $\mathsf{T} \in \mathscr{L}(\ell_r^\infty)$ is said to be a Toeplitz operator iff it commutes with the shift operator $\boldsymbol{\tau}$, i.e. $\mathsf{T}\boldsymbol{\tau} = \boldsymbol{\tau}\mathsf{T}$. Such operators are naturally associated with bounded-input, bounded-output time-invariant linear systems [Oppenheim et al., 1999].

Lemma 1.1 *T is a Toeplitz operator iff $\mathsf{T}\boldsymbol{\tau}_n = \boldsymbol{\tau}_n\mathsf{T}, \forall n \in \mathbb{Z}$.*

Proof. By induction: $\mathsf{T}\boldsymbol{\tau}_0 = \boldsymbol{\tau}_0\mathsf{T}$, and assuming $\mathsf{T}\boldsymbol{\tau}_n = \boldsymbol{\tau}_n\mathsf{T}$, then $\mathsf{T}\boldsymbol{\tau}_{n+1} = \mathsf{T}\boldsymbol{\tau}\boldsymbol{\tau}_n = \boldsymbol{\tau}\mathsf{T}\boldsymbol{\tau}_n = \boldsymbol{\tau}\boldsymbol{\tau}_n\mathsf{T} = \boldsymbol{\tau}_{n+1}\mathsf{T}$, hence the result for $n \geq 0$. Now, for $n < 0$, we repeat the induction using: $\boldsymbol{\tau}_{-1}\mathsf{T} = \boldsymbol{\tau}_{-1}\mathsf{T}\boldsymbol{\tau}\boldsymbol{\tau}_{-1} = \boldsymbol{\tau}_{-1}\boldsymbol{\tau}\mathsf{T}\boldsymbol{\tau}_{-1} = \mathsf{T}\boldsymbol{\tau}_{-1}$. ∎

Proposition 1.2 *If T is a Toeplitz operator and leaves invariant c_r^{oo}, then there exists $\boldsymbol{\alpha} \in c_{r\times r}^{oo}$ such that $\forall \mathsf{x} \in \ell_r^\infty,\ \mathsf{Tx} = \boldsymbol{\alpha} * \mathsf{x}$.*

Proof. We will prove this result for $r = 1$ and then prove it induces the matrix case ($r > 1$). Let $\boldsymbol{\alpha}[n] := \langle \boldsymbol{\delta}_n, \mathsf{T}\boldsymbol{\delta}_0 \rangle$ (this is well defined since c_r^{oo} is invariant by T and a subspace of ℓ_r^2). Clearly, $\boldsymbol{\alpha} \in c_{r\times r}^{oo}$. We get that $\langle \boldsymbol{\delta}_n, \mathsf{T}\boldsymbol{\delta}_k \rangle = \boldsymbol{\alpha}[n-k]$. Namely, $\langle \boldsymbol{\delta}_n, \mathsf{T}\boldsymbol{\delta}_k \rangle = \langle \boldsymbol{\delta}_n, \mathsf{T}\boldsymbol{\tau}_k\boldsymbol{\delta}_0 \rangle = \langle \boldsymbol{\delta}_n, \boldsymbol{\tau}_k\mathsf{T}\boldsymbol{\delta}_0 \rangle = \langle \boldsymbol{\tau}_{-k}\boldsymbol{\delta}_n, \mathsf{T}\boldsymbol{\delta}_0 \rangle = \langle \boldsymbol{\delta}_{n-k}, \mathsf{T}\boldsymbol{\delta}_0 \rangle = \boldsymbol{\alpha}[n-k]$. Now, for $\mathsf{x} \in c_r^{oo}$, let $\mathsf{y} = \mathsf{Tx}$, then $\mathsf{y}[n] = \langle \boldsymbol{\delta}_n, \mathsf{Tx} \rangle = \langle \boldsymbol{\delta}_n, \mathsf{T}\sum_k \mathsf{x}[k]\boldsymbol{\delta}_k \rangle = \sum_k \langle \boldsymbol{\delta}_n, \mathsf{T}\boldsymbol{\delta}_k \rangle \mathsf{x}[k] = \sum_k \boldsymbol{\alpha}[n-k]\mathsf{x}[k] = (\boldsymbol{\alpha} * \mathsf{x})[n]$. Now, since c_r^{oo} is dense in ℓ_r^∞, the result holds for any $\mathsf{x} \in \ell_r^\infty$.

Now for $r > 1$, we get the result by introducing $\boldsymbol{\alpha}_{i,j}[n] := \langle(\boldsymbol{\delta}_n\mathbf{e}_i), \boldsymbol{T}(\boldsymbol{\delta}_0\mathbf{e}_j)\rangle_r$ and using the same approach as before. ■

Remark 1.3

1. The Proposition 1.2 is easily specialized to the case $\boldsymbol{T} \in \mathscr{L}(\ell^2_r)$ and $\boldsymbol{T\tau} = \boldsymbol{\tau T}$ (no assumption on the invariance of c^{oo}_r). We get that there exists $\boldsymbol{\alpha} \in \ell^2_{r\times r}$ such that $\forall \boldsymbol{x} \in \ell^2_r,\ \boldsymbol{Tx} = \boldsymbol{\alpha} * \boldsymbol{x}$.

2. Toeplitz operators are also called convolutional operators in the scalar case ($r = 1$). In the following text, we will write $\boldsymbol{T_\alpha}$ to emphasize the associated matrix sequence, and we will often see $\boldsymbol{T_\alpha}$ as an infinite size matrix $[\boldsymbol{T_\alpha}]$ with matrix coefficients $[\boldsymbol{T_\alpha}]_{k,l} = \boldsymbol{\alpha}[k-l]$, i.e.
$$[\boldsymbol{T_\alpha}] = \begin{bmatrix} \ddots & & & & & & & & \\ \dots & \boldsymbol{\alpha}[2] & \boldsymbol{\alpha}[1] & \boldsymbol{\alpha}[0] & \boldsymbol{\alpha}[-1] & \boldsymbol{\alpha}[-2] & \dots & & \\ & \dots & \boldsymbol{\alpha}[2] & \boldsymbol{\alpha}[1] & \boldsymbol{\alpha}[0] & \boldsymbol{\alpha}[-1] & \boldsymbol{\alpha}[-2] & \dots & \\ & & \dots & \boldsymbol{\alpha}[2] & \boldsymbol{\alpha}[1] & \boldsymbol{\alpha}[0] & \boldsymbol{\alpha}[-1] & \boldsymbol{\alpha}[-2] & \dots \\ & & & \dots & \boldsymbol{\alpha}[2] & \boldsymbol{\alpha}[1] & \boldsymbol{\alpha}[0] & \boldsymbol{\alpha}[-1] & \boldsymbol{\alpha}[-2] \ \dots \\ & & & & & & & & \ddots \end{bmatrix} \tag{1.1}$$

3. The space of Toeplitz operators $\boldsymbol{T_\alpha}$ with $\boldsymbol{\alpha} \in c^{oo}_{r\times r}$ has a natural structure of non commutative C^*-algebra (since $\boldsymbol{T}_{\boldsymbol{\alpha}_1}\boldsymbol{T}_{\boldsymbol{\alpha}_2} = \boldsymbol{T}_{\boldsymbol{\alpha}_1 * \boldsymbol{\alpha}_2}$). The involution is given by $\boldsymbol{T}^*_{\boldsymbol{\alpha}} = \boldsymbol{T}_{\check{\boldsymbol{\alpha}}^*}$. The associated norm satisfies $||\boldsymbol{T_\alpha}||_{\mathscr{L}(\ell^\infty_r)} = |\boldsymbol{\alpha}|_{\ell^1_r}$.

Inspired by the z-domain formalism of scalar signals, we introduce the *refinement mask* $\boldsymbol{\alpha}(z)$ associated to the sequence $\boldsymbol{\alpha} \in c^{oo}_r$
$$\boldsymbol{\alpha}(z) := \tfrac{1}{2}\sum_k \boldsymbol{\alpha}[k]z^{-k}. \tag{1.2}$$
By convention (and reasons that will be clear in next chapter), the z-transform of matrix sequences (excluding scalar) is normalized by $\frac{1}{2}$. Vector sequences are still normalized by 1. When $\boldsymbol{\alpha} \in c^{oo}_{r\times r}$, $\boldsymbol{\alpha}(z)$ can also be seen as a matrix polynomial ($\boldsymbol{\alpha}(z) \in \mathbb{C}^{r\times r}[z^{-1}]$) and so $\boldsymbol{\alpha}(z)$ is well defined for any $z \in \mathbb{C}^*$. We also introduce $\boldsymbol{\alpha}^\top(z) := \frac{1}{2}\sum_k \boldsymbol{\alpha}^\top[k]z^{-k}$, $\boldsymbol{\alpha}_*(z) := \frac{1}{2}\sum_k \boldsymbol{\alpha}^*[k]z^{-k}$ and $\boldsymbol{\alpha}^*(z) := \frac{1}{2}\sum_k \boldsymbol{\alpha}^*[k]z^{k} = \boldsymbol{\alpha}_*(z^{-1})$ for $z \in \mathbb{T}$ ($\boldsymbol{\alpha}^*[k]$ is the Hermitian transpose of $\boldsymbol{\alpha}[k]$). $\boldsymbol{\alpha}(z)$ induces a multiplication operator on $L^2_r(\mathbb{T})$: for $\boldsymbol{y} = \boldsymbol{T_\alpha x}$ with $\boldsymbol{x} \in \ell^2_r$, we have in z-domain
$$\mathbf{Y}(z) = 2\boldsymbol{\alpha}(z)\mathbf{X}(z). \tag{1.3}$$
Writing $z = e^{j\omega}$ with $\omega \in]-\pi,\pi]$, it gets clear that $\boldsymbol{\alpha}(z)$ is a renormalized version of the Fourier transform $\hat{\boldsymbol{\alpha}}(\omega)$ of $\boldsymbol{\alpha}$ (we sometimes use, for normalization reasons, the notation $\hat{\boldsymbol{\alpha}}(e^{j\omega})$). There is an isomorphism
$$(\boldsymbol{\alpha}, *) \leftrightarrow (\boldsymbol{T_\alpha}, \circ) \leftrightarrow (\boldsymbol{\alpha}(z), .).$$
For more details on Toeplitz operators, the reader is referred to the classical book of Böttcher and Silbermann [1990].

1.1.2 Approximation and subdivision operators

Now, we introduce the approximation and subdivision operators: $A \in \mathscr{L}(\ell_r^\infty)$ is said to be an approximation operator iff $A\boldsymbol{\tau}_2 = \boldsymbol{\tau}A$; conversely, $S \in \mathscr{L}(\ell_r^\infty)$ is said to be a subdivision operator iff $S\boldsymbol{\tau} = \boldsymbol{\tau}_2 S$. These operators will be the basic tools to construct multifilter banks: approximation operators for analyzing the input signal by decomposing it into an approximated signal (coarse version) and some details; subdivision operators for re-synthesizing the signal from the approximated signal and the details. We easily prove again that: $A\boldsymbol{\tau}_{2n} = \boldsymbol{\tau}_n A$ and $S\boldsymbol{\tau}_n = \boldsymbol{\tau}_{2n}S$, $\forall n \in \mathbb{Z}$ (the proof is just a rewriting of the Toeplitz case). Now, by naturally extending the downsampling $(\downarrow 2)$ and upsampling $(\uparrow 2)$ operators to vector signals, we have

Proposition 1.4 *If A (resp. S) is an approximation (resp. subdivision) operator that leaves c_r^{oo} invariant, then there exists $\boldsymbol{\alpha} \in c_{r\times r}^{oo}$ (resp. $\boldsymbol{\sigma} \in c_{r\times r}^{oo}$) such that $A = (\downarrow 2)T_{\check{\boldsymbol{\alpha}}}$ (resp. $S = T_{\boldsymbol{\sigma}}^*(\uparrow 2)$).*

Proof. This proof is an adaptation of the proof of Proposition 1.2. Namely,
for $r = 1$, if we define $\boldsymbol{\alpha}[k] := \langle \boldsymbol{\delta}_0, A\boldsymbol{\delta}_k\rangle$ and $\boldsymbol{\sigma}^*[n] := \langle \boldsymbol{\delta}_n, S\boldsymbol{\delta}_0\rangle$, we get that $\langle \boldsymbol{\delta}_n, A\boldsymbol{\delta}_k\rangle = \boldsymbol{\alpha}[k-2n]$ and $\langle \boldsymbol{\delta}_n, S\boldsymbol{\delta}_k\rangle = \boldsymbol{\sigma}^*[n-2k]$. Thus, for $\boldsymbol{x} \in c_r^{oo}$, we deduce that $A\boldsymbol{x}[n] = \sum_k \boldsymbol{\alpha}[k-2n]\boldsymbol{x}[k] = ((\downarrow 2)T_{\check{\boldsymbol{\alpha}}}\boldsymbol{x})[n]$ and that $S\boldsymbol{x}[n] = \sum_k \boldsymbol{\sigma}^*[n-2k]\boldsymbol{x}[k] = (T_{\boldsymbol{\sigma}^*}(\uparrow 2)\boldsymbol{x})[n]$.
Now, for $r > 1$, introducing $\boldsymbol{\alpha}_{i,j}[k] := \langle(\boldsymbol{\delta}_0\mathrm{e}_i), A(\boldsymbol{\delta}_k\mathrm{e}_j)\rangle_r$ and $\boldsymbol{\sigma}_{i,j}^*[n] := \langle(\boldsymbol{\delta}_n\mathrm{e}_j), S(\boldsymbol{\delta}_0\mathrm{e}_i)\rangle_r$, we get the result. ∎

Remark 1.5

1. $\forall \boldsymbol{x} \in \ell_r^\infty,\ A_{\boldsymbol{\alpha}}\boldsymbol{x}[n] = \sum_k \boldsymbol{\alpha}[k-2n]\boldsymbol{x}[k]$ and $S_{\boldsymbol{\sigma}}\boldsymbol{x}[n] = \sum_k \boldsymbol{\sigma}^*[n-2k]\boldsymbol{x}[k]$.
2. Approximation and subdivision operators are conveniently seen as infinite size matrices $[A_{\boldsymbol{\alpha}}]$ and $[S_{\boldsymbol{\sigma}}]$ with matrix coefficients $[A_{\boldsymbol{\alpha}}]_{k,l} = \boldsymbol{\alpha}[l-2k]$ and $[S_{\boldsymbol{\sigma}}]_{k,l} = \boldsymbol{\sigma}^*[k-2l]$, i.e.

$$[A_{\boldsymbol{\alpha}}] = \begin{bmatrix} \cdots & & & & & & & & & \\ \cdots & \boldsymbol{\alpha}[-1] & \boldsymbol{\alpha}[0] & \boldsymbol{\alpha}[1] & \boldsymbol{\alpha}[2] & \boldsymbol{\alpha}[3] & \cdots & & & \\ & \cdots & \boldsymbol{\alpha}[-2] & \boldsymbol{\alpha}[-1] & \boldsymbol{\alpha}[0] & \boldsymbol{\alpha}[1] & \boldsymbol{\alpha}[2] & \cdots & & \\ & & \cdots & \boldsymbol{\alpha}[-3] & \boldsymbol{\alpha}[-2] & \boldsymbol{\alpha}[-1] & \boldsymbol{\alpha}[0] & \boldsymbol{\alpha}[1] & \cdots & \\ & & & \cdots & \boldsymbol{\alpha}[-4] & \boldsymbol{\alpha}[-3] & \boldsymbol{\alpha}[-2] & \boldsymbol{\alpha}[-1] & \boldsymbol{\alpha}[0] & \cdots \\ & & & & & & & & & \cdots \end{bmatrix} \tag{1.4}$$

$$[S_{\boldsymbol{\sigma}}] = \begin{bmatrix} \ddots & & & \\ & \boldsymbol{\sigma}^*[0] & & \\ & \boldsymbol{\sigma}^*[1] & \boldsymbol{\sigma}^*[-1] & \\ & \boldsymbol{\sigma}^*[2] & \boldsymbol{\sigma}^*[0] & \boldsymbol{\sigma}^*[-2] \\ & & \boldsymbol{\sigma}^*[1] & \boldsymbol{\sigma}^*[-1] \\ & & & \boldsymbol{\sigma}^*[0] \\ & & & & \ddots \end{bmatrix} \tag{1.5}$$

3. Approximation and subdivision are adjoint operators:

$$A_{\boldsymbol{\alpha}}^* = ((\downarrow 2) T_{\check{\boldsymbol{\alpha}}})^* = T_{\check{\boldsymbol{\alpha}}}^* (\downarrow 2)^* = T_{\boldsymbol{\alpha}^*} (\uparrow 2) = S_{\boldsymbol{\alpha}} \tag{1.6}$$

$$S_{\boldsymbol{\alpha}}^* = (T_{\boldsymbol{\alpha}^*}(\uparrow 2))^* = (\downarrow 2) T_{\boldsymbol{\alpha}^*}^* = (\downarrow 2) T_{\check{\boldsymbol{\alpha}}} = A_{\boldsymbol{\alpha}} \tag{1.7}$$

In z-domain, $(\downarrow 2)\mathbf{X}(z) = \frac{1}{2}(\mathbf{X}(z^{\frac{1}{2}}) + \mathbf{X}(-z^{\frac{1}{2}}))$ and $(\uparrow 2)\mathbf{X}(z) = \mathbf{X}(z^2)$. Letting $y_1 = A_{\boldsymbol{\alpha}} x = (\downarrow 2) T_{\check{\boldsymbol{\alpha}}} x$ and $y_2 = S_{\boldsymbol{\alpha}} x = T_{\boldsymbol{\alpha}^*}(\uparrow 2) x$ for $x \in \ell_r^2$, we get in z-domain

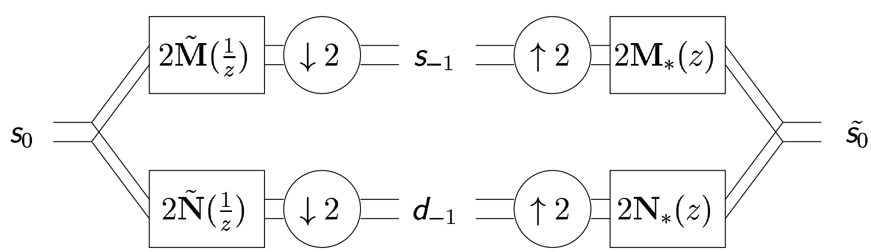
$$\begin{aligned} \mathbf{Y}_1(z) &= (\downarrow 2)(2\boldsymbol{\alpha}(z^{-1})\mathbf{X}(z)) = \boldsymbol{\alpha}(z^{-\frac{1}{2}})\mathbf{X}(z^{\frac{1}{2}}) + \boldsymbol{\alpha}(-z^{-\frac{1}{2}})\mathbf{X}(-z^{\frac{1}{2}}) && (1.8) \\ \mathbf{Y}_2(z) &= 2\boldsymbol{\alpha}_*(z)((\uparrow 2)\mathbf{X}(z)) = 2\boldsymbol{\alpha}_*(z)\mathbf{X}(z^2). && (1.9) \end{aligned}$$

The noble identities are easily extended to the matrix case:

$$\begin{aligned} \text{First noble identity}: &\quad (\downarrow 2)\boldsymbol{\alpha}(z^2) = \boldsymbol{\alpha}(z)(\downarrow 2). \\ \text{Second noble identity}: &\quad \boldsymbol{\alpha}(z^2)(\uparrow 2) = (\uparrow 2)\boldsymbol{\alpha}(z). \end{aligned} \tag{1.10}$$

1.1.3 Multifilter banks

Again, we will assume we are working with signals in ℓ_r^2. Now, let a multifilter bank: $\mathbb{M} = (\mathbb{A}, \mathbb{S})$ with $A_0 = A_{\mathbf{M}}, A_1 = A_{\tilde{\mathbf{N}}}$ and $S_0 = S_{\mathbf{M}}, S_1 = S_{\mathbf{N}}$ where $\mathbf{M}, \mathbf{N}, \tilde{\mathbf{M}}, \tilde{\mathbf{N}} \in c_r^{oo}$. Suppose $\mathbb{M}$ satisfies the PR condition: $\mathbb{S}\mathbb{A} = I$ and the dMRA condition: S_0 and S_1 to are one-to-one operators and the operators $P_i := S_i A_i$, for $i = 0, 1$ are bona-fide projectors of $\mathscr{L}(\ell_r^\infty)$



First, we notice that S_i being one-to-one implies that A_i are onto ℓ_r^2: since $\operatorname{im} A_i = (\ker A_i)^\top = \ker S_i^* = (\{O\})^\top = \ell_r^2$. So, A_0, A_1 are onto ℓ_r^2. We also easily prove that this implies that A_i and S_i cannot be O for $i = 0, 1$. Namely, assuming either $A_0 = O$ or $S_0 = O$, we necessarily get $S_1 A_1 = I$. This implies that S_1 is onto ℓ_r^2, since it is already one-to-one, it is then bijective, and so is A_1 which is clearly impossible.
Now, we will prove that under these conditions, the analysis and synthesis banks are invertible. We already know that $\mathbb{S}\mathbb{A} = I$. We will also prove that the dMRA condition implies that $\mathbb{A}\mathbb{S} = \mathbb{I}$ which is known as the biorthogonality condition.

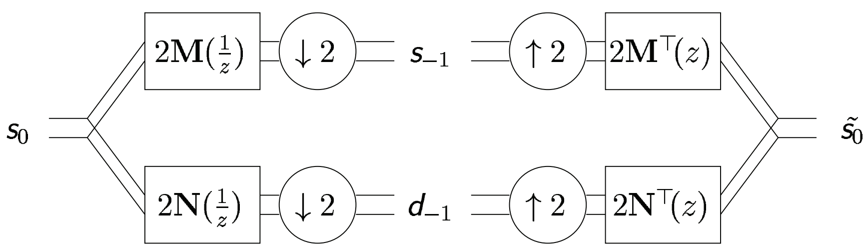
Proposition 1.6 *Under PR condition, dMRA and biorthogonality are equivalent.*

Proof.
$[\Rightarrow]$: $P_0^2 = P_0$ implies $S_0A_0(S_0A_0 - I) = O$. But, since S_0 is one-to-one, then $A_0(S_0A_0 - I) = O$, i.e. $(A_0S_0 - I)A_0 = O$ and since A_0 is onto, then $A_0S_0 = I$. We get in the same way $A_1S_1 = I$. Now, $S_0A_0 + S_1A_1 = I$, then $A_1S_0A_0 + A_1S_1A_1 = A_1$, so $A_1S_0A_0 = O$ and since A_0 is onto, then $A_1S_0 = O$. Similarly, we have $A_0S_1 = O$, and then $\mathbb{A}\mathbb{S} = \mathbb{I}$.
$[\Leftarrow]$: From $A_0S_0 = I$, $P_0^2 = S_0A_0S_0A_0 = S_0(A_0S_0)A_0 = S_0A_0 = P_0$, same thing for P_1. Hence the result. ∎

Translating the biorthogonality conditions in the z-domain, we get

$$\begin{aligned}
\tilde{\mathbf{M}}(z)\mathbf{M}^*(z) + \tilde{\mathbf{M}}(-z^{-1})\mathbf{M}^*(-z^{-1}) &= \mathbf{I} \\
\tilde{\mathbf{N}}(z)\mathbf{N}^*(z) + \tilde{\mathbf{N}}(-z^{-1})\mathbf{N}^*(-z^{-1}) &= \mathbf{I} \\
\tilde{\mathbf{M}}(z)\mathbf{N}^*(z) + \tilde{\mathbf{M}}(-z^{-1})\mathbf{N}^*(-z^{-1}) &= \mathbf{0} \\
\tilde{\mathbf{N}}(z)\mathbf{M}^*(z) + \tilde{\mathbf{N}}(-z^{-1})\mathbf{M}^*(-z^{-1}) &= \mathbf{0}
\end{aligned} \tag{1.11}$$

Besides, we already know that analysis and synthesis are adjoint. We say that a multifilter bank $\mathbb{M} = (\mathbb{A}, \mathbb{S})$ is orthogonal if $\mathbb{A}$ and $\mathbb{S}$ are not only inverse of one another but also transpose, i.e. $\mathbb{A} = \mathbb{S}^*$. We get in that case $\tilde{\mathbf{M}} = \mathbf{M}$ and $\tilde{\mathbf{N}} = \mathbf{N}$. The multifilter bank reduces to the case shown below.



1.2 Signal processing with multifilter banks

Assuming the conditions of PR and biorthogonality, let's consider the multifilter bank $\mathbb{M} = (\mathbb{A}, \mathbb{S})$ with $\mathbb{A} = \begin{bmatrix} A_{\tilde{\mathbf{M}}} \\ A_{\tilde{\mathbf{N}}} \end{bmatrix}$ and $\mathbb{S} = [S_{\mathbf{M}}\ S_{\mathbf{N}}]$, where $\mathbf{M}, \tilde{\mathbf{M}}, \mathbf{N}, \tilde{\mathbf{N}} \in c^{oo}_{r\times r}$. We will call $P_{\mathbf{M}} := S_{\mathbf{M}}A_{\tilde{\mathbf{M}}}$ the lowpass projector and $P_{\mathbf{N}}$ the highpass projector. This arbitrary denomination will be justified later in the text by the introduction of the concept of balancing in the next section (the lowpass projector preserves the constant signals) and in the process of iteration in the next chapter (we iterate on the lowpass branch so as to get coarser and coarser approximations). Our main concern here is that multifilter banks are intrinsically designed to process vector-valued signals. The input signal s_0 is decomposed into a coarse version $s_{-1} := A_{\tilde{\mathbf{M}}}s_0$ and

a detail version $d_{-1} := A_{\tilde{\mathbf{N}}} s_0$, that will form through the synthesis bank a decomposition of $s_0 = S_{\mathbf{M}} s_{-1} + S_{\mathbf{N}} d_{-1} = P_{\mathbf{M}} s_0 + P_N s_0$. By iterating on the coarse version, we get a vector version of the *Mallat algorithm* [Mallat, 1989]:
For the analysis, $n \leq 0$,

$$s_{n-1}[l] = \sum_k \tilde{\mathbf{M}}[k-2l] s_n[k] \tag{1.12}$$

$$d_{n-1}[l] = \sum_k \tilde{\mathbf{N}}[k-2l] s_n[k] \tag{1.13}$$

and for the synthesis, we get

$$s_n[l] = \sum_k \mathbf{M}^*[l-2k] s_{n-1}[k] + \mathbf{N}^*[l-2k] d_{n-1}[k]. \tag{1.14}$$

Similarly to the scalar case, these formula can be interpreted as the expansion of s_0 in a biorthogonal basis of ℓ_r^2. Vector signals are expanded in $\{v_{1,n}^{(-1)}, \ldots, v_{r,n}^{(-1)}, w_{1,n}^{(-1)}, \ldots, w_{r,n}^{(-1)}\}_n$ where $v_{i,n}^{(-1)}[k] = \mathbf{M}^*[n-2k]\mathbf{e}_i$ and $w_{i,n}^{(-1)}[k] = \mathbf{N}^*[n-2k]\mathbf{e}_i$ ($\mathbf{e}_i$ is the i^{th} canonical vector). The dual basis is given by $v_{i,n}^{(-1)}[k] = \tilde{\mathbf{M}}^*[n-2k]\mathbf{e}_i$ and $w_{i,n}^{(-1)}[k] = \tilde{\mathbf{N}}^*[n-2k]\mathbf{e}_i$. Now, by iterating the multifilter bank on the lowpass branch, we get a biorthogonal multiresolution analysis of ℓ_r^2 (cf. [Aldroubi et al., 1994] for the scalar case).

1.2.1 Processing scalar signals

Now if we have to process a scalar ("1D") signal $x \in \ell^\infty$ using a multifilter bank, we have first to *vectorize* the signal, i.e. to produce a vector version $s \in \ell_r^\infty$ of this signal. We are then able to process this signal by the multifilter bank $\mathbb{M}$. The output is again a vector valued signal that we will *fold* to get back a scalar signal.

The *canonical* way of doing vectorization and folding is by introducing the *polyphase* decomposition of a signal. Given a scalar signal x, we introduce the r signals $x_i[n] := x[rn+i]$. In the time domain, we get the polyphase vectorization operator $(\Rsh_r)$ defined by

$$((\Rsh_r)x)[n] := \begin{bmatrix} x_0[n] \\ x_1[n] \\ \vdots \\ x_{r-1}[n] \end{bmatrix}. \tag{1.15}$$

Conversely, we introduce the polyphase folding $({}_r\Lsh)$. For $y = ({}_r\Lsh)[x_0, x_1, \ldots, x_{r-1}]^\top$, we let in the z-domain,

$$Y(z) := X_0(z^r) + z^{-1}X_1(z^r) + \ldots + z^{-(r-1)}X_{r-1}(z^r). \tag{1.16}$$

It is then clear that $(\Rsh_r)$ and $({}_r\Lsh)$ are adjoint operators.

We now introduce the general definition of vectorization and folding operators: $\mathsf{V} \in \mathscr{L}(\ell^\infty, \ell_r^\infty)$ is said to be a *vectorization* operator iff $\mathsf{V}\tau_r = \boldsymbol{\tau}\mathsf{V}$ and $\mathsf{V}(c^{oo}) \subset c_r^{oo}$, similarly $\mathsf{F} \in \mathscr{L}(\ell_r^\infty, \ell^\infty)$ is said to be a *folding* operator iff $\mathsf{F}\boldsymbol{\tau} = \tau_r\mathsf{F}$ and $\mathsf{F}(c_r^{oo}) \subset c^{oo}$. These conditions impose the structure of the vectorization and folding operators.

Proposition 1.7 *If V (resp. F) is a vectorization (resp. folding) operator, then there exists $\boldsymbol{\nu} \in c_r^{oo}$ (resp. $\boldsymbol{\varsigma} \in c_r^{oo}$) such that for $\mathsf{x} \in \ell^\infty$, we have $\mathsf{V}\mathsf{x}[n] = \sum_k \boldsymbol{\nu}[k - rn]\mathsf{x}[k]$ (resp. for $\boldsymbol{\mathsf{x}} \in \ell_r^\infty$, we have $\mathsf{F}\boldsymbol{\mathsf{x}}[n] = \sum_k \boldsymbol{\varsigma}^*[n - rk]\boldsymbol{\mathsf{x}}[k]$).*

Proof. The proof is very similar again to the proofs of the propositions characterizing Toeplitz, approximation and subdivision operators. ■

Extending naturally the definition of $\boldsymbol{\mathsf{T}}_{\boldsymbol{\alpha}}$ for $\boldsymbol{\alpha} \in c_r^{oo}$ (vector), we get that $\mathsf{V} = (\downarrow r)\mathsf{T}_{\check{\boldsymbol{\nu}}}$ and $\mathsf{F} = \mathsf{T}_{\boldsymbol{\varsigma}^*}(\uparrow r)$. Introducing $\boldsymbol{\pi} := \sum_{k=0}^{r-1} \boldsymbol{\delta}_k \mathbf{e}_{k+1}$, we have that $(\Rsh_r) = (\downarrow r)\mathsf{T}_{\check{\boldsymbol{\pi}}}$ and $({}_r\Lsh) = \mathsf{T}_{\boldsymbol{\pi}^\top}(\uparrow r)$. Also, using the polyphase decomposition and imposing vectorization and folding to be adjoint, we can easily rewrite any vectorization operator (resp. folding operator) as $\mathsf{V} = \boldsymbol{\mathsf{T}}_{\check{\boldsymbol{\varrho}}}(\Rsh_r)$ (resp. $\mathsf{F} = ({}_r\Lsh)\boldsymbol{\mathsf{T}}_{\boldsymbol{\varrho}^*}$) with $\boldsymbol{\varrho} \in c_{r\times r}^{oo}$.

Now, to maintain the perfect reconstruction of the multifilter bank when dealing with scalar signals, it is natural to impose that $\mathsf{F}\mathsf{V}\mathsf{x} = \mathsf{x}$ for x belonging to some set of signals that we want to be preserved. If we require $\mathsf{F}\mathsf{V}\mathsf{x} = \mathsf{x}, \forall \mathsf{x} \in \ell^\infty$, since $(\Rsh_r)$ and $({}_r\Lsh)$ are bijective from ℓ^∞ to ℓ_r^∞, then it is equivalent to require $\boldsymbol{\mathsf{T}}_{\boldsymbol{\varrho}^*}\boldsymbol{\mathsf{T}}_{\check{\boldsymbol{\varrho}}} = \boldsymbol{\mathsf{I}}$ i.e. $\boldsymbol{\varrho}^*(z)\boldsymbol{\varrho}(z) = 1, \forall z \in \mathbb{T}$. A matrix $\boldsymbol{\varrho}(z)$ satisfying this condition is called paraunitary. Paraunitary matrices have a very nice structure since they factorize [Vaidyanathan, 1993] as

$$\boldsymbol{\varrho}(z) = \boldsymbol{\varrho}(1)\prod_{k=1}^{n}(I - \mathbf{R}_k + \mathbf{R}_k z^{\epsilon_k}) \tag{1.17}$$

where $\boldsymbol{\varrho}(1), \mathbf{R}_1, \ldots, \mathbf{R}_n$ are orthogonal matrices and $\epsilon_k = \pm 1$. Such vectorization and folding operators are usually called orthogonal pre/post filters. For more details on the design of such operators for multifilters, the reader is referred to the papers of Xia et al. [1996]; Hardin and Roach [1997]; Strela et al. [1999]; Attakitmongcol et al. [1999].

Another idea would be to restrict the set of signals that have to be preserved by the vectorization/folding operators. This enables a greater freedom in the design. A natural set of signals that one usually wants to be preserved are the polynomial signals, i.e. the signals of the form $\mathsf{x}[n] := p(n)$ where $p \in \mathbb{C}[t]$ is a polynomial. For convenience, we will denote $\mathsf{p}[n]$ such a signal. We also say that a signal x is *polynomial* if there exists $p \in \mathbb{C}[t]$ such that $\mathsf{x}[n] = p(n)$. Then, there is an obvious isomorphism π between the space $\mathbb{C}[t]$ of polynomials $p(t)$ and $\mathcal{C}$ the space of the associated polynomial signals $\mathsf{p}[n]$. We will denote $\mathcal{C}_k$ the subspace of polynomial sequences of degree less or equal to k (the continuous counterpart being $\mathbb{C}_k[t]$).

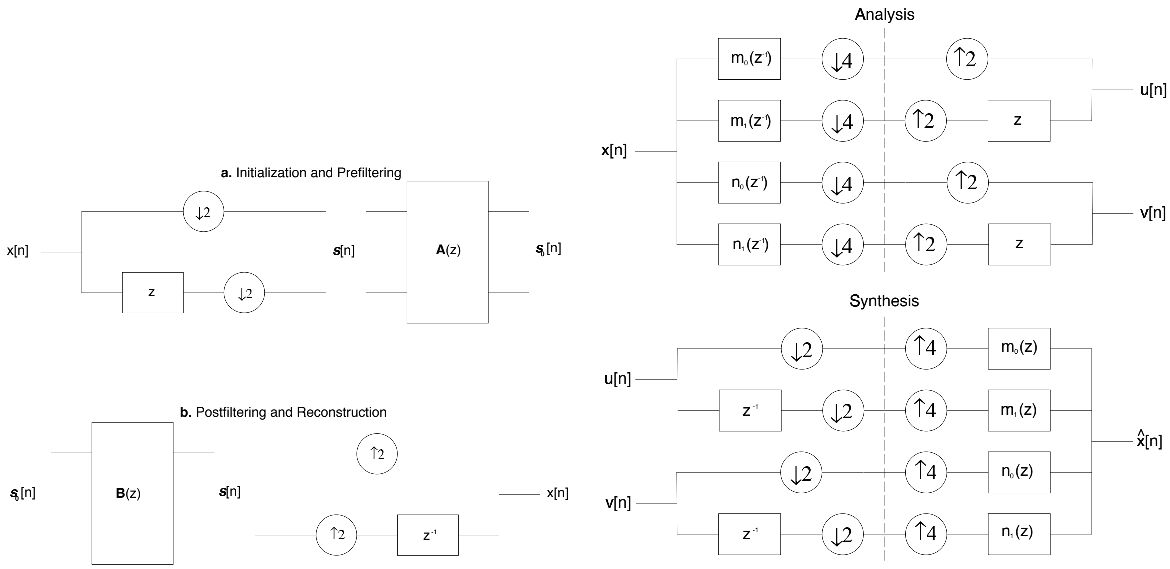


Figure 1.1: Left: general vectorization and folding for a multifilter bank, Right: multifilter bank seen as a time-varying filter bank.

We then look at vectorization/folding operators V, F such that $\forall p \in \mathcal{C}_{k-1}$, we have $FVp = p$. Since $\mathcal{C}_{k-1}$ is finite dimensional and $F = S^*$, this implies V to be bijective from $\mathcal{C}_{k-1}$ to $V\mathcal{C}_{k-1}$ and so $\forall \mathbf{x} \in V\mathcal{C}_{k-1}$, $VF\mathbf{x} = \mathbf{x}$. In this framework, we define

Definition 1.8 *A multifilter bank $\mathbb{M}$ is said to be polynomial preserving of order k iff there exists V and F, vectorization and folding operators, such that $FS_{\mathbf{M}}V$ keeps $\mathcal{C}_{k-1}$ invariant.*

Let $L := FS_{\mathbf{M}}V$ and $\tilde{L} := FA_{\tilde{\mathbf{M}}}V$ (and similarly H and $\tilde{H}$). Polynomial preservation of order k does not imply that L exactly preserves polynomial signals. However, since $\mathcal{C}_{k-1}$ has finite dimension and $S_{\mathbf{M}}$ is one-to-one, we get that $L\mathcal{C}_{k-1} = \mathcal{C}_{k-1}$ i.e. the $\mathcal{C}_{k-1}$ is globally preserved. We also get from the PR and biorthonormality conditions that

$$\tilde{H}\mathcal{C}_{p-1} = \tilde{H}L\mathcal{C}_{p-1} = FA_{\tilde{\mathbf{N}}}VFS_{\mathbf{M}}V\mathcal{C}_{p-1} = \{\mathbf{0}\}.$$

Furthermore, this also gives that $L\tilde{L}x = x$, for every $x \in \mathcal{C}_{p-1}$ i.e. the polynomial structure (up to degree $p-1$) of the input signal is canceled by the highpass branch and exactly preserved by the lowpass branch (hence the lowpass term).

1.2.2 Balancing

Here, we will get interested in the special case when vectorization and folding are simply the polyphase operators $(\Rsh_r)$ and $({}_r\Lsh)$. We introduce the polyphase description of the refinement

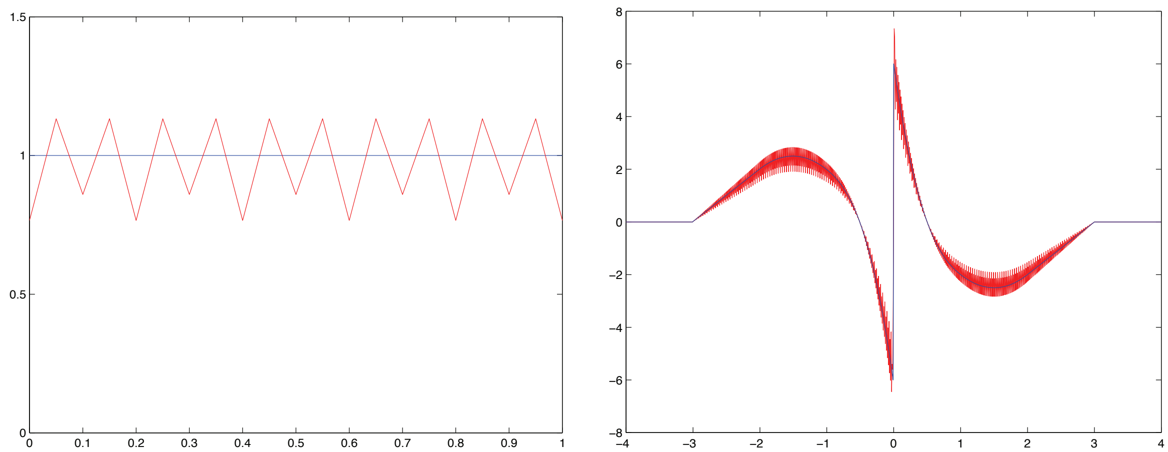


Figure 1.2: Reproduction of two input signals (Left: constant signal, Right: piecewise polynomial) by the lowpass branch of a DGHM multifilter bank; it illustrates the behavior of a unbalanced multifilter bank without prefiltering.

mask $\mathbf{M}(z)$

$$\mathbf{m}(z) = \begin{bmatrix} m_0(z) \\ m_1(z) \\ \cdots \\ m_{r-1}(z) \end{bmatrix} := 2\mathbf{M}(z^r) \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(r-1)} \end{bmatrix}. \tag{1.18}$$

This enables to rewrite $L = ({}_r\Lsh) S_{\mathbf{M}} (\Rsh_r)$ as a time-varying filter. Namely, let $x \in \ell^2$ and $y = Lx$, going to z-domain, we get that

$$Y(z) = \begin{bmatrix} 1, & z^{-1}, & \dots, & z^{-(r-1)} \end{bmatrix} 2\mathbf{M}_*(z^r) \begin{bmatrix} X_0(z^{2r}) \\ \vdots \\ X_{r-1}(z^{2r}) \end{bmatrix} = \mathbf{m}_*(z) \begin{bmatrix} X_0(z^{2r}) \\ \vdots \\ X_{r-1}(z^{2r}) \end{bmatrix}$$

where $[X_0(z), \dots, X_{r-1}(z)]$ is the polyphase decomposition of $X(z)$. Introducing in the same way, $\tilde{\mathbf{m}}(z), \mathbf{n}(z)$ and $\tilde{\mathbf{n}}(z)$, the multifilter bank is then easily transformed into a $2r$ channels time-varying filter bank (Fig. 1.1).

An intuitive way of understanding the problems that may appear in that situation is then the following: if the components $m_0(z), m_1(z), \dots, m_{r-1}(z)$ of the lowpass synthesis operator have different spectral behavior, e.g. lowpass behavior for one, highpass for the other, it then leads to *unbalanced* channels that cannot preserve even constant signals. In that case, the polyphase method of vectorization leads to a mixing of the coarse resolution and details coefficients creating strong oscillations in the signal reconstructed from $\mathbf{s}_{-1}[n]$ only (Fig. 1.2). This problem is crucial. One of the important issues of subband coding is the behavior of the lowpass branch on polynomial signals. Namely, this branch should carry all the information on the input signal when this one is sufficiently smooth. In other terms, one expects some class of smooth signals to be well reproduced using only the lowpass coefficients, i.e. one expect these signals to be *eigensignals* of the lowpass branch.

However, most of the multifilter banks constructed so far don't verify this simple requirement of as illustrated in Fig. 1.2. We will see in Chapter 3 how to add some pre/post filtering of the input/output signal to adapt it to the spectral imbalance of the filter bank. But, as we will show here, one may rather directly design orthonormal multifilters with a good balancing between the polyphase components of the synthesis lowpass operator $S_{\mathbf{M}}$. Consequently, recalling that $L = ({}_r\Lsh) S_{\mathbf{M}} (\Rsh_r)$ stands for the 1D version of the lowpass synthesis operator (and in the same way $\tilde{L}$, H and $\tilde{H}$), we will look for the preservation of constant signals by L. Let $u_0[n] := 1, \forall n$, we introduce

Definition 1.9 *A multifilter bank $\mathbb{M} = (\mathbb{A}, \mathbb{S})$ is said to be balanced (of order 1) iff the lowpass synthesis operator $L := ({}_r\Lsh) S_{\mathbf{M}} (\Rsh_r)$ preserves the constant signals.*

By linearity, and without loss of generality, it is sufficient to impose $Lu_0 = u_0$. Now, by the PR and biorthonormality relations

$$\begin{bmatrix} L & H \end{bmatrix} \begin{bmatrix} \tilde{L} \\ \tilde{H} \end{bmatrix} = I \quad \text{and} \quad \begin{bmatrix} \tilde{L} \\ \tilde{H} \end{bmatrix} \begin{bmatrix} L & H \end{bmatrix} = \mathbb{I}$$

we get $L\tilde{L} + H\tilde{H} = I, \tilde{L}L = I, \tilde{L}H = \tilde{H}L = O$ and $\tilde{H}H = I$. From $Lu_0 = u_0$ we get $L\tilde{L}u_0 = u_0$ and $\tilde{H}u_0 = u_0$ i.e. u_0 is preserved by the lowpass branch and canceled by the highpass branch.

Now, we can state the following result giving equivalent conditions for balancing, and especially linking balancing to a condition on the factorization of the refinement mask $\mathbf{M}(z)$:

Theorem 1.10 *Balancing of order 1 is equivalent to any of the following conditions:*

B0. $Lu_0 = u_0$.

B1. $[\,1 \cdots 1\,]\,\mathbf{M}(1) = [\,1 \cdots 1\,]$ *and* $[\,1 \cdots 1\,]\,\mathbf{M}(-1) = \mathbf{0}^\top$.

B3. $\mu(z) := \sum_{i=0}^{r-1} m_i(z)$ *has zeros*[1] *at* $z = e^{jk\pi/r}$ *for* $k = 1, \ldots, 2r-1$ *and* $\mu(1) = 2r$.

[1] Conditions B3 and its generalization to higher order balancing were first given by Selesnick [1998].

$\mathbf{B4}$. *One can factorize*[2] $\mathbf{M}(z) = \frac{1}{2}\boldsymbol{\Delta}(z^2)\mathbf{M}_0(z)\boldsymbol{\Delta}^{-1}(z)$ *with*

$$\boldsymbol{\Delta}(z) := \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 1 & -1 \\ -z^{-1} & 0 & \dots & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{M}_0(1)\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} = \begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}.$$

Proof.

- [B0⇒B1]: Assuming B0, we have by transposition $u_0^* L^* = u_0^*$. Writing explicitly the equations, we get $[\,1\cdots1\,]\sum_k \mathbf{M}[2k+1] = [\,1\cdots1\,]\sum_k \mathbf{M}[2k] = [\,1\cdots1\,]$. So $[\,1\cdots1\,]\sum_k \mathbf{M}[k] = 2\,[\,1\cdots1\,]$ and $[\,1\cdots1\,]\sum_k (-1)^k \mathbf{M}[k] = \mathbf{0}$. Since $\mathbf{M}(1) = \frac{1}{2}\sum_k \mathbf{M}[k]$ and $\mathbf{M}(-1) = \frac{1}{2}\sum_k (-1)^k \mathbf{M}[k]$, we have condition B1.

- [B1⇒B3]: $\mu(z) = \sum_{i=0}^{r-1} m_i(z) = [1, 1, \dots, 1]2\mathbf{M}(z^r)[1, z^{-1}, \dots, z^{-(r-1)}]^\top$. So $\mu(1) = [1, 1, \dots, 1]2\mathbf{M}(1)[1, 1, \dots, 1]^\top = 2r$. Now, if $z = e^{jk\pi/r}$ with $k = 2l+1$ and $l = 0, \dots, r$, then $z^r = -1$ and so $\mu(z) = 0$. If $z = e^{jk\pi/r}$ with $k = 2l$ and $l = 1, \dots, r-1$, then $z^r = 1$ and $\mu(z) = 2\sum_{k=0}^{r-1} z^k = 2\frac{1-z^{-r}}{1-z^{-1}} = 0$. So $\mu(z)$ has roots at $z = e^{jk\pi/r}$ for $k = 1, \dots, 2r-1$.

- [B3⇒B0]: Taking $x := u_0$, from the time-varying filter bank representation (Fig. 1.1), we get that the $2r$ possible outputs are $y[2rn+l] = \sum_k \sum_{i=0}^{r-1} m_i^*[2rk+l]$ for $l = 0, \dots, 2r-1$. Denoting $\mu^{(l)}(z)$ the l[th] polyphase component of $\mu(z)$, we get that

$$y^*[2rn+l] = \mu_*^{(l)}(1) = \frac{1}{2r}\sum_{k=0}^{2r} \mu_*(e^{jk\pi/r}) = \frac{1}{2r}\Big(\sum_{k=0}^{2r} \mu(e^{-jk\pi/r})\Big)^* = \frac{1}{2r}(\mu(1))^* = 1.$$

  Hence u_0 is an eigenvector of the operator L.

- [B1⇒B4]: This is Theorem 4.1 [(b)⇒(c)] from [Plonka, 1997] for the special case $\mathbf{y} = [1, \dots, 1]^\top$.

- [B4⇒B3]: Assuming B4, since

$$\sum_{i=0}^{r-1} m_i(z) = \begin{bmatrix}1, & \dots, & 1\end{bmatrix} 2\mathbf{M}(z^r)\begin{bmatrix}1\\ \vdots \\ z^{-(r-1)}\end{bmatrix}$$

[2]This factorization is a special case of the Strela/Plonka factorizations [Plonka, 1997; Plonka and Strela, 1998] (aka 2-scale similarity transforms).

we have

$$\sum_{i=0}^{r-1} m_i(z) = \tfrac{1}{2}\begin{bmatrix}1, & \dots, & 1\end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 1 & -1 \\ -z^{-2r} & 0 & \dots & 0 & 1 \end{bmatrix} 2\mathbf{M}_0(z^r) \dots$$

$$\dots \tfrac{1}{1-z^{-r}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ z^{-r} & 1 & 1 & \ddots & \vdots \\ \vdots & z^{-r} & \ddots & \ddots & 1 \\ \vdots & \vdots & \ddots & 1 & 1 \\ z^{-r} & z^{-r} & \dots & z^{-r} & 1 \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \\ \vdots \\ z^{-(r-1)} \end{bmatrix}$$

so

$$\sum_{i=0}^{r-1} m_i(z) = \left(\tfrac{1-z^{-2r}}{1-z^{-1}}\right) \begin{bmatrix}1, & 0, & \dots, & 0\end{bmatrix} \mathbf{M}_0(z^r) \begin{bmatrix} 1 \\ z^{-1} \\ \vdots \\ z^{-(r-1)} \end{bmatrix}$$

and this is condition B3.

■

1.2.3 High order balancing

A natural generalization of the concept of balancing is then to impose higher degree polynomial signals $[\dots, q(-2), q(-1), q(0), q(1), q(2), \dots]^\top$ (where $q(t)$ is any polynomial of degree smaller than p) to be also preserved by the lowpass branch. Thus, we define:

Definition 1.11 *A multifilter bank is said to be balanced of order p iff the lowpass synthesis operator L preserves the polynomial signals of degree less or equal to $p-1$, i.e. $\mathcal{C}_{p-1}$ is invariant by L.*

Again, this condition does not imply that L exactly preserves polynomial signals. However, since $\mathcal{C}_{p-1}$ has finite dimension, and $S_{\mathbf{M}}$ is one-to-one, we have that $L\mathcal{C}_{p-1} = \mathcal{C}_{p-1}$ i.e. the $\mathcal{C}_{p-1}$ is globally preserved. We also get from the PR and biorthonormality conditions that $\tilde{H}\mathcal{C}_{p-1} = \tilde{H}L\mathcal{C}_{p-1} = \{\mathbf{0}\}$. Furthermore, this also gives that $L\tilde{L}x = x$, for every $x \in \mathcal{C}_{p-1}$ i.e. the polynomial structure (up to degree $p-1$) of the input signal is canceled by the highpass branch and exactly preserved by the lowpass branch.

Using an approach introduced by [Herley, 1995; Selesnick, 1998] based on the interpolation of all the polyphase components of a polynomial signal from a single one, we will get that on polynomial signals of degree smaller than the order of balancing (i.e. $\mathcal{C}_{p-1}$), the lowpass synthesis operator L (with its intricate time-varying structure) is in fact equivalent to a scalar subdivision scheme (on which the classical results from the scalar wavelet theory apply). The following lemma is the core result of this approach.

Lemma 1.12 *Let $u_{0,r}^{(n)}(z), u_{1,r}^{(n)}(z), \ldots, u_{r-1,r}^{(n)}(z)$ be the z-transform of the polyphase components of the polynomial signal $u_n[k] := k^n$ (seen as a tempered distribution), i.e.*

$$u_{i,r}^{(n)}(z) := \sum_{k \in \mathbb{Z}} (kr+i)^n z^{-k}$$

then for $i = 1, \ldots, r-1$, there exists a unique polynomial $\alpha_{i,r}^{(n)}(z)$ of degree n such that we have the equality $u_{i,r}^{(n)}(z) = \alpha_{i,r}^{(n)}(z) u_{0,r}^{(n)}(z)$ and the normalization $\alpha_{i,r}^{(n)}(1) = 1$. Similarly, there exists a unique normalized polynomial $\beta_{i,r}^{(n)}(z)$ of degree n such that we have $u_{0,r}^{(n)}(z) = \beta_{i,r}^{(n)}(z) u_{i,r}^{(n)}(z)$.

Proof. Using Padé approximants [Nikishin and Sorokin, 1991; Baker, 1996], we construct the Hörner scheme of interpolation $\alpha_{i,r}^{(n)}(z)$ of the sequence $u_{i,r}^{(n)}[k] := (kr+i)^n$ from the sequence $u_{0,r}^{(n)}[k] := (kr)^n$ given by

$$\begin{aligned}\alpha_{i,r}^{(n)}(z) = 1 + \tfrac{i}{r}(1-z^{-1})\underset{1}{[}1 + \tfrac{i+r}{2r}(1-z^{-1})\underset{2}{[}1 + \tfrac{i+2r}{3r}(1-z^{-1})\underset{3}{[}1 + \ldots \\ \ldots)\underset{n-2}{[}1 + \tfrac{i+(n-2)r}{(n-1)r}(1-z^{-1})\underset{n-1}{[}1 + \tfrac{i+(n-1)r}{nr}(1-z^{-1})\underset{n-1}{]}\underset{n-2}{]}\ldots\underset{1}{]}.\end{aligned} \tag{1.19}$$

Thus, we have

$$\alpha_{i,r}^{(n)}(z) = 1 + \sum_{k=1}^{n} \frac{\Gamma(k+\frac{i}{r})}{\Gamma(k+1)\Gamma(\frac{i}{r})}(1-z^{-1})^k. \tag{1.20}$$

Furthermore, it is easily seen that one can write

$$u_{i,r}^{(n)}(z) = \frac{1}{[(1-z)(1-z^{-1})]^{n+1}} \sum_{k=-n}^{n} \upsilon_{i,r}^{(n)}[k] z^{-k} \tag{1.21}$$

consequently $u_{i,r}^{(n)}(z)$ cannot be formally canceled by multiplication with a polynomial, and so we have the uniqueness. Similarly,

$$\begin{aligned}\beta_{i,r}^{(n)}(z) = 1 + \tfrac{i-r}{r}(1-z^{-1})\underset{1}{[}1 + \tfrac{i}{2r}(1-z^{-1})\underset{2}{[}1 + \tfrac{i+r}{3r}(1-z^{-1})\underset{3}{[}1 + \ldots \\ \ldots)\underset{n-2}{[}1 + \tfrac{i+(n-3)r}{(n-1)r}(1-z^{-1})\underset{n-1}{[}1 + \tfrac{i+(n-2)r}{nr}(1-z^{-1})\underset{n-1}{]}\underset{n-2}{]}\ldots\underset{1}{]}.\end{aligned} \tag{1.22}$$

interpolates $\boldsymbol{u}_{0,r}^{(n)}$ from the sequence $\boldsymbol{u}_{i,r}^{(n)}$. As before, we have the uniqueness and we can write

$$\beta_{i,r}^{(n)}(z) = 1 + \sum_{k=1}^{n} \frac{\Gamma(k-\frac{i}{r})}{\Gamma(k+1)\Gamma(-\frac{i}{r})}(1-z^{-1})^k. \tag{1.23}$$

■

Remark 1.13 For the case $r=2$, we give $\alpha_{1,2}^{(n)}(z)$ and $\beta_{1,2}^{(n)}(z)$ for $n=0,1,2,3$.

$$\begin{aligned}
&\alpha_{1,2}^{(0)}(z) = 1 && \beta_{1,2}^{(0)}(z) = 1 \\
&\alpha_{1,2}^{(1)}(z) = \tfrac{1}{2}(3-z^{-1}) && \beta_{1,2}^{(1)}(z) = \tfrac{1}{2}(1+z^{-1}) \\
&\alpha_{1,2}^{(2)}(z) = \tfrac{1}{8}(15-10z^{-1}+3z^{-2}) && \beta_{1,2}^{(2)}(z) = \tfrac{1}{8}(3+6z^{-1}-z^{-2}) \\
&\alpha_{1,2}^{(3)}(z) = \tfrac{1}{16}(35-35z^{-1}+21z^{-2}-5z^{-3}) && \beta_{1,2}^{(3)}(z) = \tfrac{1}{16}(5+15z^{-1}-5z^{-2}+z^{-3})
\end{aligned}$$

By natural extension, we also define $\alpha_{0,r}^{(n)}(z) := \beta_{0,r}^{(n)}(z) := 1$. We then introduce the vectors

$$\boldsymbol{\alpha}_n^\top(z) := \left[\alpha_{0,r}^{(n)}(z), \quad \alpha_{1,r}^{(n)}(z), \quad \ldots, \quad \alpha_{r-1,r}^{(n)}(z)\right] \tag{1.24}$$

$$\boldsymbol{\beta}_n^\top(z) := \left[\beta_{0,r}^{(n)}(z), \quad \beta_{1,r}^{(n)}(z), \quad \ldots, \quad \beta_{r-1,r}^{(n)}(z)\right]. \tag{1.25}$$

Introducing for $k \geq 1$,

$$\mathbf{\Gamma}_k := \left[0 \quad \frac{\Gamma(k+\frac{1}{r})}{\Gamma(k+1)\Gamma(\frac{1}{r})} \quad \cdots \quad \frac{\Gamma(k+\frac{r-1}{r})}{\Gamma(k+1)\Gamma(\frac{r-1}{r})}\right], \tag{1.26}$$

and $\mathbf{\Gamma}_0 := [1, \ldots, 1]^\top$, we get

$$\boldsymbol{\alpha}_n^\top(z) = \sum_{k=0} n\mathbf{\Gamma}_k(1-z^{-1})^k. \tag{1.27}$$

We set

$$\mu_p(z) := \sum_{k=0}^{r-1} \alpha_{k,r}^{(p)}(z^{2r}) m_k(z) = \boldsymbol{\alpha}_p^\top(z)\mathbf{m}(z) = \boldsymbol{\alpha}_p^\top(z) 2\mathbf{M}(z^r)\boldsymbol{\pi}(z), \tag{1.28}$$

with $\boldsymbol{\pi}(z) = [1, z^{-1}, \ldots, z^{-(r-1)}]^\top$. We then have

Proposition 1.14 *For any $p \geq 1$, the two operators $L := ({}_r\!\!\Lsh)S_{\mathbf{M}}(\Rsh_r)$ and $S_{\mu_{p-1}}(\downarrow r)$ agree on the space $\mathcal{C}_{p-1}$ of polynomial signals of degree less or equal to $p-1$.*

Proof. By linearity, we have that $\alpha_{i,r}^{(p-1)}(z)$ interpolates the i$^{\text{th}}$ polyphase component of any polynomial signal of degree smaller than p from the 0^{th} polyphase component. That means that on $\mathcal{C}_{p-1}$, we have $(\Lsh_r) = T_{\boldsymbol{\alpha}_{p-1}}(\downarrow r)$. Letting $x \in \mathcal{C}_{p-1}, y = Lx$ and going to z-domain, we get that

$$\begin{aligned} Y(z) &= \begin{bmatrix} 1, & z^{-1}, & \ldots, & z^{-(r-1)} \end{bmatrix} 2\mathbf{M}_*(z^r) \begin{bmatrix} X_0(z^{2r}) \\ \vdots \\ X_{r-1}(z^{2r}) \end{bmatrix} \\ &= \mathbf{m}_*(z)\boldsymbol{\alpha}_{p-1}(z^{2r})X_0(z^{2r}) = (\boldsymbol{\alpha}_{p-1}^\top(z^{2r})\mathbf{m}(z))_* X_0(z^{2r}). \end{aligned} \tag{1.29}$$

Thus, the two operators $L := ({}_r\Lsh) S_{\mathbf{M}}(\Lsh_r)$ and $S_{\mu_{p-1}}(\downarrow r)$ agree on $\mathcal{C}_{p-1}$. This gives in block diagram notation that L, i.e.

$$-(\Lsh_r) = (\uparrow 2) = \boxed{2\mathbf{M}_*(z)} = ({}_r\Lsh)\ - \tag{1.30}$$

is equivalent on $\mathcal{C}_{p-1}$ to the scalar $2r$-subdivision scheme

$$-(\downarrow r) - (\uparrow 2r) - \boxed{(\mu_{p-1})_*(z)}\ - \tag{1.31}$$

Hence the result. ∎

Remark 1.15 The operator $V = T_{\boldsymbol{\alpha}_{p-1}}(\downarrow r)$ gives an example of a non trivial vectorization operator such that $FV = I$ on $\mathcal{C}_{p-1}$ (we can take $F = ({}_r\Lsh)$). However, V and F are not adjoint operators.

As a result, we will study thoroughly in next section the preservation of polynomial signals by scalar $2r$-subdivision operators.

1.3 Scalar subdivision operators

In this section, we will study the algebraic properties of scalar M-subdivision operators. The principal issue we are concerned with is the behavior of these operators on polynomial signals. Our approach owes a lot to the very comprehensive monograph of Cavaretta et al. [1991] and many of the results given here are only adaptations or easy extensions for our special needs. First, we will introduce the polyphase decomposition approach. This will enable us to give necessary and sufficient conditions on the preservation of polynomial signals by scalar M-subdivision operators. In this direction, the major result obtained is certainly a theorem where, once more, the famous Strang-Fix conditions [Strang and Fix, 1973] appear to be central. Finally, we show that the eigenvectors of such operators have a particular nice structure since they appear to be the discrete-time version of Appell sequence of polynomials.

In all this section, M will be a fixed positive integer. We introduce the scalar M-subdivision operator $S_g : \ell^\infty \to \ell^\infty$ defined by

$$S_g x[n] := \sum_k g^*[n - kM]x[k] \tag{1.32}$$

where $\{g[k]\}_k$ is a finite length sequence of complex numbers. For $a = 0, \dots, M-1$, we introduce the sequences

$$w_a[n] := \begin{cases} 1 & \text{if } n = a \pmod{M}, \\ 0 & \text{otherwise.} \end{cases} \tag{1.33}$$

One easily see that for $a = 0, \dots, M-1$,

- w_a is M-periodic.
- $\Lambda_a := a + M\mathbb{Z}$ form a sublattice partition of $\mathbb{Z}$.
- M-periodic sequences are constant on Λ_a.
- $w_a[n] = 1_{\Lambda_a}(n)$.

So every M-periodic sequence x can be written as $x[n] = \sum^a x[a]w_a[n]$, where for convenience, we write $\sum^a$ for the summation $\sum_{a=0}^{M-1}$. We can endow the space $\mathcal{P}_M$ of M-periodic sequences with an Hilbert structure with scalar product $\langle x, y\rangle_M := \sum^a x^*[a]y[a]$. $\mathcal{P}_M$ has dimension M and $\{w_a\}_{a=0}^{M-1}$ is an obvious basis. We also introduce for $k, l \in \mathbb{Z}, \Omega_{k,l} := \sum^a w_a[k]w_a[l]$. For $0 \le a_1, a_2 \le M-1$, we get that $\Omega_{a_1,a_2} = \delta_{a_1,a_2}$. Intuitively, $\Omega_{k,l}$ is just a modulo M version of the Kronecker symbol. Its enables to split a sequence as a sum of M-downsampled subsequences.

Lemma 1.16 *For $x \in \ell^\infty$, we have $S_g x[n] = \sum_k \Omega_{n,k} g^*[k]x[\frac{n-k}{M}]$.*

Proof. We can always write $n = a + kM$ for some a and k, we then have

$$\begin{aligned} S_g x[n] &= \sum_l g^*[a + kM - lM]x[l] = \sum_l g^*[a + (k-l)M]x[l] \\ &= \sum_{l'} g^*[a + l'M]x[k - l'] = \sum^b \sum_{l'} \delta_{a,b}\, g^*[b + l'M]x[k - l'] \\ &= \sum^b \sum_{l'} \Omega_{a+kM, b+l'M} g^*[b + l'M]x[k - l'] = \sum_m \Omega_{n,m} g^*[m]x[\tfrac{n-m}{M}]. \end{aligned}$$

■

We introduce the subspace $\mathcal{Z}_M$ of M-periodic sequences x with zero mean, i.e. $\sum^a x[a] = 0$.

Lemma 1.17 $\lambda_a[k] := \zeta_M^{ka} = \exp(j\frac{2\pi}{M}ka)$, *for* $a = 1, \ldots, M-1$, *form an orthogonal basis of* $\mathcal{Z}_M$ *(usually called the DFT basis).*

Proof. $\mathcal{Z}_M$ is a subspace of $\mathcal{P}_M$, then obviously $\dim \mathcal{Z}_M < M$ and since $\{\mathsf{w}_a - \mathsf{w}_0\}_{a=1}^{M-1} \subset \mathcal{Z}_M$ and are linearly independent, then $\dim \mathcal{Z}_M = M-1$.
Now, all we have to prove is that $\lambda_a[k] := \zeta_M^{ka}$ for $a = 1, \ldots, M-1$ are linearly independent. Namely, for $1 \le a_1, a_2 \le M-1$,

$$\langle \mathsf{x}_{a_1}, \mathsf{x}_{a_2} \rangle_M = \sum\nolimits^b (\zeta_M^{a_1 b})^* \zeta_M^{a_2 b} = \sum\nolimits^b \zeta_M^{(a_2-a_1)b} = M\delta_{a_1,a_2}.$$

So $\{\lambda_a\}_{a=1}^{M-1}$ spans $\mathcal{Z}_M$, hence the result. ∎

1.3.1 Polynomial signals

The previous polyphase decomposition of the output of the subdivision operator make it now simple to study the influence of subdivision operators on polynomial signals. As before, we will write p for the polynomial signal associated to a polynomial $p(t)$ (i.e. we have $sp[n] := p(n)$). We also write $\mathcal{C}$ for the space of polynomial signals and $\mathcal{C}_k$ denotes the subspace of polynomial signals associated to a polynomial of degree less or equal to k.

Lemma 1.18 *For* $\mathsf{p} \in \mathcal{C}$, $\exists!\, \mathsf{p}_0, \ldots, \mathsf{p}_{M-1} \in \mathcal{C}$ *such that* $\mathsf{S_g p} = \sum^a \mathsf{w}_a \mathsf{p}_a$.

Proof. Take $p_a(t) := \sum_k \mathsf{w}_a[k]\mathsf{g}^*[k]p(\frac{t-k}{M})$, then for $n \in \mathbb{Z}$, we have

$$\begin{aligned}
\sum\nolimits^a \mathsf{w}_a[n]\mathsf{p}_a[n] &= \sum\nolimits^a \mathsf{w}_a[n] \sum_k \mathsf{w}_a[k]\mathsf{g}^*[k]p(\tfrac{n-k}{M}) \\
&= \sum_k \Big(\sum\nolimits^a \mathsf{w}_a[n]\mathsf{w}_a[k]\Big)\mathsf{g}^*[k]p(\tfrac{n-k}{M}) \\
&= \sum_k \Omega_{n,k}\mathsf{g}^*[k]p(\tfrac{n-k}{M}) = \mathsf{S_g p}[n]
\end{aligned}$$

The last equality being just Lemma 1.16. From the definition of w_a and the isomorphism of $\mathbb{C}[t]$ and $\mathcal{C}$, we have clearly the uniqueness. ∎

Proposition 1.19 *Let* $\mathsf{p} \in \mathcal{C}$ *a polynomial signal, a necessary and sufficient condition for* $\mathsf{S_g p}$ *to be polynomial is*

$$\forall \mathsf{d} \in \mathcal{Z}_M, \sum_k \mathsf{d}[k]\mathsf{g}^*[k]p(\tfrac{t-k}{M}) = 0, \forall t \in \mathbb{R}. \tag{1.34}$$

In this case, the polynomial signal $\mathsf{S}_g\mathsf{p}$ is associated to the polynomial

$$q(t) = \frac{1}{M}\sum_k g^*[k]p(\tfrac{t-k}{M}). \tag{1.35}$$

Proof.
[$\Rightarrow$] : Assume that $\mathsf{S}_g\mathsf{p}$ is polynomial and let $\mathsf{q} := \mathsf{S}_g\mathsf{p}$. From the previous lemma, we have that $\mathsf{q}[n] = \sum^a \mathsf{w}_a[n]\mathsf{p}_a[n]$. Then, for $a = 0,\dots,M-1$, taking $n \in \Lambda_a$, we get $q(n) = p_a(n)$ by the isomorphism π between $\mathbb{C}[t]$ and cC, so $\forall a$, $q = p_a$, i.e. $q = p_0$ and $p_a = p_0$. Using the definition of p_a, this means that $\forall a, \sum_k(\mathsf{w}_a[k] - \mathsf{w}_0[k])g^*[k]p(\frac{t-k}{M}) = 0$. Now, since $\{\mathsf{w}_a - \mathsf{w}_0\}_{a=1}^{M-1}$ spans $\mathcal{Z}_M$, we get

$$\forall \mathsf{d} \in \mathcal{Z}_M, \sum_k \mathsf{d}[k]g^*[k]p(\tfrac{t-k}{M}) = 0, \forall t \in \mathbb{R}.$$

[$\Leftarrow$] : Reciprocally, taking $\mathsf{d} := \mathsf{w}_a - \mathsf{w}_0$, we get $\mathsf{p}_a = \mathsf{p}_0, \forall a$. Thus $\mathsf{S}_g\mathsf{p} = \mathsf{p}_0$, i.e. $\mathsf{S}_g\mathsf{p}$ is polynomial.

Now, using $1 = \sum^a(\mathsf{w}_a - \mathsf{w}_0) + M\mathsf{w}_0$, we get that

$$\begin{aligned}\sum_k g[k]p(\tfrac{t-k}{M}) &= M\sum_k \mathsf{w}_0[k]g[k]p_0(\tfrac{t-k}{M}) + \sum^a\sum_k(\mathsf{w}_a[k] - \mathsf{w}_0[k])g[k]p_0(\tfrac{t-k}{M})\\ &= M\sum_k \mathsf{w}_0[k]g[k]p_0(\tfrac{t-k}{M}) = Mp_0(t).\end{aligned}$$

Consequently, $q(t) = \frac{1}{M}\sum_k g[k]p(\frac{t-k}{M})$, hence the result. ∎

At the continuous-time level, we introduce the spaces $\mathscr{I}_g := \{p \in \mathbb{C}[t] \,|\, \mathsf{S}_g\mathsf{p} \text{ is polynomial}\}$ and for $p \in \mathbb{C}[t]$, $\mathscr{T}(p) := \operatorname{span}\{\tau_h p \,|\, h \in \mathbb{R}\}$. We also introduce their discrete-time counterparts $\mathcal{I}_g := \pi(\mathscr{I}_g) = \{\mathsf{p} \in \mathcal{C} \,|\, \mathsf{S}_g\mathsf{p} \text{ is polynomial}\}$ and $\mathcal{T}(\mathsf{p}) := \pi(\mathscr{T}(p))$. Furthermore, one says that $\mathcal{Q} \subset \mathcal{C}$ is *translation invariant* iff $\pi^{-1}(\mathcal{Q})$ is translation invariant i.e $\forall p \in \pi^{-1}(\mathcal{Q}), \forall h \in \mathbb{R}, \tau_h p \in \pi^{-1}(\mathcal{Q})$. In that case, we define $\boldsymbol{\tau}_h\mathsf{p} := \pi(\tau_h p)$. Clearly, $\mathscr{T}(p)$ and so $\mathcal{T}(\mathsf{p})$ are translation invariant for any $p \in \mathbb{C}[t]$. Furthermore,

Lemma 1.20 $\forall p \in \mathbb{C}[t], \mathscr{T}(p) = \operatorname{span}\{D^n p \,|\, n \geq 0\}$.

Proof. Namely, let $q \in \mathscr{T}(p)$, then there exists finite sets $K \subset \mathbb{Z}, \{\alpha_k\}_{k\in K} \subset \mathbb{C}^K$ and $\{y_k\}_{k\in K} \subset \mathbb{C}^K$, such that $q(t) = \sum_{k\in K}\alpha_k p(t - y_k)$. First, we notice that necessarily $\deg q \leq \deg p = N$. Thus, $\mathscr{T}(p) \subset \mathbb{C}_N[t]$ and so $\dim \mathscr{T}(p) < \infty$. $\mathscr{T}(p)$ is then a compact subspace of $C^\infty(\mathbb{R},\mathbb{C})$ (with its usual Fréchet topology). And since $\forall h \in \mathbb{R}, \Delta_h p(t) := \frac{p(t+y)-p(t)}{y} \in \mathscr{T}(p)$, by compactness, $p'(t) = \lim_{h\to 0}\Delta_h p(t) \in \mathscr{T}(p)$. By induction, we get that $\forall n \geq 0, D^n p \in \mathscr{T}(p)$. Finally, by the Taylor expansion $p(t+h) = \sum_{k\geq 0}\frac{1}{k!}h^k D^k p(t)$, we get $\mathscr{T}(p) = \operatorname{span}\{D^k p \,|\, k \geq 0\}$. ∎

Proposition 1.21 *We have then the following results:*

1. *$\mathcal{I}_g$ is translation invariant.*

2. *For $\mathcal{Q} \subset \mathcal{I}_g$, if $\mathcal{Q}$ is translation invariant, then $S_g(\mathcal{Q})$ is also translation invariant.*

3. *For $p \in \mathcal{I}_g$, $\mathcal{T}(S_g p) = S_g(\mathcal{T}(p))$.*

Proof. We have

1. Let $p \in \mathcal{I}_g$. For $y \in \mathbb{R}$, introduce $q(t) := \tau_y p(t) = p(t-y)$. Now, taking $t = x - My$, we have by Proposition 1.19, $\forall d \in \mathcal{Z}_M, \forall x \in \mathbb{R}$,

$$0 = \sum_k d[k]g^*[k]p(\tfrac{x-My-k}{M}) = \sum_k d[k]g^*[k]p(\tfrac{x-k}{M} - y) = \sum_k d[k]g^*[k]q(\tfrac{x-k}{M}).$$

   Hence, $q \in \mathcal{I}_g$, i.e. $\mathcal{I}_g$ is translation invariant.

2. Let $y \in \mathbb{R}$. For $q \in S_g(\mathcal{Q}), \exists\, p \in \mathcal{Q} \subset \mathcal{I}_g$ such that $\forall t \in \mathbb{R}$,

$$q(t - yM) = \tfrac{1}{M}\sum_k g^*[k]p(\tfrac{t-yM-k}{M}) = \tfrac{1}{M}\sum_k g^*[k]p(\tfrac{t-k}{M} - y).$$

   So,

$$\tau_{yM} q(t) = \tfrac{1}{M}\sum_k g^*[k]\tau_y p(\tfrac{t-k}{M}).$$

   Furthermore, since $\mathcal{Q}$ is translation invariant, $\boldsymbol{\tau}_y p \in \mathcal{Q} \subset \mathcal{I}_g$ and $S_g\boldsymbol{\tau}_y p$ is polynomial, so $\exists q_y \in \mathbb{C}[t]$ such that $q_y = S_g\boldsymbol{\tau}_y p$ and $q_y(t) = \frac{1}{M}\sum_k g^*[k]\tau_y p(\frac{t-k}{M})$. Necessarily, $q_y = \tau_{yM} q$ and $\boldsymbol{\tau}_{yM} q = S_g\boldsymbol{\tau}_y p \in S_g(\mathcal{Q})$. That means that $S_g(\mathcal{Q})$ is also translation invariant.

3. Since $\mathcal{I}_g$ is a translation invariant linear subspace of $\mathcal{C}$, assuming that $p \in \mathcal{I}_g$, we get clearly that $\mathcal{T}(p) \subset \mathcal{I}_g$ and so that $S_g(\mathcal{T}(p))$ is also translation invariant. Furthermore, from the previous part, we have that $\forall h \in \mathbb{R}, \boldsymbol{\tau}_{hM} S_g p = S_g\boldsymbol{\tau}_h p$, hence $\mathcal{T}(S_g p) = S_g(\mathcal{T}(p))$.

∎

Denoting the monomials $u_N := t^N$, where N is a positive integer, we get that $\mathscr{T}(u_N) = \mathrm{span}\{u_k \,|\, 0 \le k \le N\}$, and

Proposition 1.22 *$S_g u_N$ is polynomial iff $\mathcal{T}(u_N)$ is invariant by S_g.*

Proof.
$[\Rightarrow]$: By the previous proposition, $S_g(\mathcal{T}(u_N)) = \mathcal{T}(S_g u_N)$. Furthermore, denoting $q_N := S_g u_N$, we have

$$\begin{aligned} q_N(t) &= \tfrac{1}{M}\sum_k g[k]u_N(\tfrac{t-k}{M}) = \tfrac{1}{M}\sum_k g[k](\tfrac{t-k}{M})^N \\ &= \tfrac{1}{M^{N+1}}\sum_k g[k](t-k)^N = \sum_k \tfrac{1}{M^{N+1}} g[k]\tau_k u_N(t). \end{aligned}$$

So $q_N \in \mathcal{T}(u_N)$, then $S_g(\mathcal{T}(u_N)) \subset \mathcal{T}(u_N)$.
$[\Leftarrow]$: If $S_g(u_N) \subset \mathcal{T}(u_N)$, then there exist finite sets $K \subset \mathbb{Z}$, $\{\alpha_k\}_{k\in K} \subset \mathbb{C}^K$ and $\{y_k\}_{k\in K} \subset \mathbb{C}^K$, such that

$$S_g(u_N) = \sum_{k\in K} \alpha_k \boldsymbol{\tau}_{y_k} u_N.$$

Let $q_n(t) := \sum_{k\in K}\alpha_k(t-y_k)^N \in \mathbb{C}[t]$, then $S_g(u_N) = q_N$ and so $S_g(u_N)$ is polynomial. ∎

Remark 1.23 For $g \in c^{oo}$, we introduce the operator

$$\begin{aligned} \mathscr{S}_g : \mathbb{C}[t] &\to \mathbb{C}[t] \\ p(t) &\mapsto \mathscr{S}_g p(t) := \tfrac{1}{M}\textstyle\sum_k g^*[k]p(\tfrac{t-k}{M}) \end{aligned} \tag{1.36}$$

Then, clearly $\deg \mathscr{S}_g p \le \deg p$. If furthermore, $\sum_k g[k] \neq 0$, then $\deg \mathscr{S}_g p = \deg p$. In that case, taking $p = u_N$, we get that $S_g u_N$ is polynomial iff $S_g(\mathcal{T}(u_N)) = \mathcal{T}(u_N)$.

1.3.2 Strang-Fix

Here, we get the central result of this part: an equivalence between the property of preservation of polynomial signals and the famous Strang-Fix conditions on the z-transform $G(z) = \sum_k g[k]z^{-k}$ of the sequence $\{g[k]\}_k$. Then, denoting $D := \frac{d}{d\omega}$, we have

Proposition 1.24 *For $p \in \mathcal{C}$, $S_g p$ is polynomial iff* $\forall q \in \mathscr{T}(p), q^*(\frac{1}{M}jD)G(\zeta_M^k) = 0$ *for* $k = 1, \dots, M-1$.

Proof.
$[\Leftarrow]$: Expanding the notations, we have $D^n G(e^{j\omega}) = \sum_k g[k](-jk)^n e^{-jk\omega}$ and for $q(t) =$

$\sum_{n=0}^{\deg q} q_n t^n$, we get $q^*(\frac{1}{M}jD) = \sum_{n=0}^{\deg q} q_n^*(-\frac{1}{M}jD)^n$. Then

$$
\begin{aligned}
q^*(\tfrac{1}{M}jD)G(e^{j\omega}) &= \sum_{n=0}^{\deg q} q_n^*(-1)^n j^n \tfrac{1}{M^n} \sum_k \mathbf{g}[k](-j)^n k^n e^{-jk\omega} \\
&= \sum_k (\sum_{n=0}^{\deg q} q_n^*(-\tfrac{k}{M})^n)\mathbf{g}[k]e^{-jk\omega} \qquad (1.37)\\
&= \sum_k q^*(-\tfrac{k}{M})\mathbf{g}[k]e^{-jk\omega}.
\end{aligned}
$$

Hence, for $l = 1, \ldots, M-1$,

$$
\sum_k q(-\tfrac{k}{M})\mathbf{g}^*[k]\zeta_M^{kl} = \left(\sum_k q^*(-\tfrac{k}{M})\mathbf{g}[k]\zeta_M^{-kl}\right)^* = ((q^*(\tfrac{1}{M}jD)G)(\zeta_M^l))^* = 0.
$$

Since this is true $\forall q \in \mathcal{T}(p)$, taking $q_x(t) := p(x+t)$, we have $\forall x \in \mathbb{R}$,

$$
\sum_k (\zeta_M^l)^k \mathbf{g}^*[k]p(x - \tfrac{k}{M}) = 0.
$$

Now, since $\{\lambda_a\}_{a=1}^{M-1}$ spans $\mathcal{Z}_M$, $\forall \mathbf{d} \in \mathcal{Z}_M, \forall y \in \mathbb{R}$, $\sum_k \mathbf{d}[k]\mathbf{g}^*[k]p(\frac{y-k}{M})$, i.e. by Proposition 1.19, $\mathcal{S}_{\mathbf{g}}\mathbf{p}$ is polynomial.
$[\Rightarrow]$ If $\mathcal{S}_{\mathbf{g}}\mathbf{p}$ is polynomial, by Proposition 1.19, for $l = 1, \ldots, M-1$ and $\forall t \in \mathbb{R}$,

$$
\sum_k (\zeta_M^l)^k \mathbf{g}^*[k]p(t - \tfrac{k}{M}) = 0.
$$

Now, writing $q_t(x) := p(x+t)$, we get for $l = 1, \ldots, M-1$, $\sum_k q_t^*(-\frac{k}{M})\mathbf{g}[k](\zeta_M^{-kl}) = 0$, which generalize to all $q \in \mathscr{T}(p)$. Hence, from (1.37), $\forall q \in \mathscr{T}(p)$, $q^*(\frac{1}{M}jD)G(\zeta_M^l) = 0$, for $l = 1, \ldots, M-1$. ∎

Corollary 1.25 *$\mathcal{S}_{\mathbf{g}}\mathbf{u}_N$ is polynomial iff $D^k G(\zeta_M^l) = 0$ for $k = 0, \ldots, N$ and $l = 1, \ldots, M-1$.*

Proof. This comes from $\mathscr{T}(u_N) = \mathrm{span}\{t^k \mid k \leq N\}$. ∎

1.3.3 Appell sequences

Assuming that $\mathcal{S}_{\mathbf{g}}\mathbf{u}_N$ is polynomial, we will prove that we can construct a sequence of polynomial signals $\{\boldsymbol{\rho}_k\}_{k\geq 0}$ that are eigenvectors of the subdivision operator $\mathcal{S}_{\mathbf{g}}$. This sequence has a very nice structure. As we will see, it is associated to an Appell sequence of polynomials (special case of Sheffer sequence [Roman, 1984]).

Definition 1.26 *A sequence $\{\rho_k\}_{k\geq 0}$ of polynomials of $\mathbb{C}[t]$ is called an Appell sequence iff $\forall k, \deg \rho_k = k$ and $\rho'_{k+1}(t) = (k+1)\rho_k(t)$.*

Proposition 1.27 *Appell sequences have the following properties:*

- *They are uniquely determined by a sequence of scalar $\{r_k\}_{k\geq 0}$ such that $\forall n, \rho_n(t) = \sum_{k=0}^{n} \binom{n}{k} r_{n-k} t^k$.*
- *Introducing the formal series $r(z) := \sum_{k\geq 0} r_k z^k$, we have that*

$$r(z)e^{tz} = \sum_{k\geq 0} \frac{1}{k!}\rho_k(t) z^k. \tag{1.38}$$

  $r(z)$ is called the generating formal series associated to the Appell sequence $\{\rho_k\}_{k\geq 0}$.
- *Appell identity: $\rho_n(t+h) = \sum_{k=0}^{n} \binom{n}{k} \rho_{n-k}(t) h^k$.*

Proof. The reader is referred to this very good reference on formal series [Roman, 1984]. ∎

We can see Appell sequences as generalization of the canonic Appell sequence: $\rho_n(t) := t^n$ (in that case $r(z) = 1$ and the Appell identity turns out to be the binomial formula).

From now on, we will assume that $G(1) = \sum_k g[k] = 2$. We then get

Theorem 1.28 *$\mathsf{S}_g \mathsf{u}_N$ is polynomial iff there exists an Appell sequence $\{\rho_k\}_k$ such that $\mathsf{S}_g \boldsymbol{\rho}_k = M^{-k}\boldsymbol{\rho}_k$ for $k = 0, \ldots, N$.*

Proof. $[\Rightarrow]$: $\mathsf{S}_g \mathsf{u}_N$ is polynomial and $\sum_k g[k] \neq 0$, so $\mathsf{S}_g \mathcal{T}(\mathsf{u}_N) = \mathcal{T}(\mathsf{u}_N)$. Thus, $\mathscr{S}_g$ is one-to-one onto $\mathscr{T}(u_N)$ and we have the commutative diagram:

$$\begin{tikzcd} \mathscr{T}(u_N) \arrow[r, "\mathscr{S}_g"] \arrow[d, "\pi"'] & \mathscr{T}(u_N) \arrow[d, "\pi"] \\ \mathcal{T}(\mathsf{u}_N) \arrow[r, "\mathsf{S}_g"'] & \mathcal{T}(\mathsf{u}_N) \end{tikzcd} \tag{1.39}$$

Now, consider the functional equation:

$$\begin{cases} f(Mz) = \frac{1}{M} G_*(e^z) f(z) \\ f(0) = 1. \end{cases}$$

Since $G_*(e^z)$ is an entire function and $G_*(e^z)_{|z=0} = M$, then there exists an unique solution such that $f(0) = 1$ and $f(z)$ is analytic at $z = 0$. Let $r(z) := \frac{1}{f(z)} = \sum_{k\geq 0} r_k z^k$, it satisfies

$$\begin{cases} r(z) = \frac{1}{M} G_*(e^z) r(Mz) \\ r(0) = 1. \end{cases} \tag{1.40}$$

We introduce the Appell sequence $\{\rho_k\}_k$ associated to $r(z)$.
Now, we will verify that $\mathscr{S}_g \rho_k(t) = M^{-k}\rho_k(t)$. Namely, from (1.38),

$$\begin{aligned}
\sum_{k\geq 0} M^{-k}\frac{1}{k!}\rho_k(t)z^k &= e^{t\frac{z}{M}} r(\tfrac{z}{M}) = \tfrac{1}{M} e^{t\frac{z}{M}} G_*(e^{\frac{z}{M}}) r(z) \\
&= \tfrac{1}{M} e^{t\frac{z}{M}} r(z) \sum_n g^*[n] e^{-n\frac{z}{M}} = \tfrac{1}{M} r(z) \sum_n g^*[n] e^{z\frac{t-n}{M}} \\
&= \tfrac{1}{M}\sum_n g^*[n](r(z)e^{z\frac{t-n}{M}}) = \tfrac{1}{M}\sum_n g^*[n]\sum_{l\geq 0}\frac{1}{l!}\rho_l(\tfrac{t-n}{M})z^l \\
&= \sum_{l\geq 0}(\frac{1}{l!}\sum_n \tfrac{1}{M} g^*[n]\rho_l(\tfrac{t-n}{M}))z^l = \sum_{l\geq 0}\frac{1}{l!}\mathscr{S}_g\rho_l(t)z^l.
\end{aligned}$$

Then, for $k = 0,\dots,N$, $\mathscr{S}_g\rho_k(t) = M^{-k}\rho_k(t)$. Now, by the isomorphisms of the commutative diagram, we get for $k = 0,\dots,N$, that $S_g \boldsymbol{\rho}_k = M^{-k}\boldsymbol{\rho}_k$.
$[\Leftarrow]$: By induction: $S_g\boldsymbol{\rho}_0 = \boldsymbol{\rho}_0$ so $S_g u_0$ is polynomial. Now, assuming $q_k := S_g u_k$ is polynomial for $k = 0,\dots,n-1$, since

$$\begin{aligned}
S_g u_n &= S_g(\boldsymbol{\rho}_n - \sum_{k=0}^{n-1}\binom{n}{k} r_{n-k} u_k) \\
&= M^{-n}\boldsymbol{\rho}_n - \sum_{k=0}^{n-1}\binom{n}{k} r_{n-k} S_g u_k.
\end{aligned}$$

Then, taking $q_n(t) := M^{-n}\rho_n(t) - \sum_{k=0}^{n-1}\binom{n}{k} r_{n-k} q_{n-k}(t)$, we get that $S_g\boldsymbol{\rho}_n = q_n$, so $S_g u_n$ is also polynomial, hence the result by induction up to N. ∎

Remark 1.29

1. The generating function $r(z)$ is the unique entire solution of

$$\begin{cases} r(z) = \tfrac{1}{M} G_*(e^z) r(Mz), \quad \forall z \in \mathbb{C} \\ r(0) = 1. \end{cases} \tag{1.41}$$

2. Introducing for $a = 0,\dots,M-1$, the polyphase operators $\mathscr{S}_g^a$ in continuous-time,

$$\begin{aligned}
\mathscr{S}_g^a : \mathbb{C}[t] &\to \mathbb{C}[t] \\
p(t) &\mapsto \mathscr{S}_g^a p(t) := \textstyle\sum_k w_a[k] g^*[k] p(\tfrac{t-k}{M})
\end{aligned}$$

we get that $S_g u_N$ is polynomial iff there exists an Appell sequence $\{\rho_k\}_k$ of common eigenvectors of $\mathscr{S}_g^a$ for the eigenvalues M^{-k} for $k = 0,\dots,N$.

1.4 On balancing conditions

In this section, we will connect balancing order to other discrete-time properties of multifilter banks. In particular, balancing will be shown to be equivalent to a special, much simpler, case of Plonka and Strela [1998] factorization of the refinement mask and various versions of the Strang-Fix conditions. These equivalences will be key results for the design of balanced multifilter banks in Chapter 3.

1.4.1 The scalar approach

It was proven in Proposition 1.14, that on polynomial signals of degree less or equal to $p-1$, the two operators $L := ({}_r\Lsh) S_{\mathbf{M}} (\Rsh_r)$ and $S_{\mu_{p-1}}(\downarrow r)$ agree, .i.e.

$$-(\Rsh_r) = (\uparrow 2) = \boxed{2\mathbf{M}_*(z)} = ({}_r\Lsh)\ - \tag{1.42}$$

is equivalent on $\mathcal{C}_{p-1}$ to the scalar $2r$-subdivision scheme

$$-(\downarrow r) - (\uparrow 2r) - \boxed{\textstyle\sum_{k=0}^{r-1} \alpha_{k,r}^{(p-1)}(z^{2r}) m_k(z)}\ - \tag{1.43}$$

Now, using the results from previous section, we get

Proposition 1.30 *Balancing of order p is equivalent to $\mu_{p-1}(z) = \sum_{k=0}^{r-1} \alpha_{k,r}^{(p-1)}(z^{2r}) m_k(z)$ having zeros of order p at $z = e^{jk\pi/r}$ for $k = 1, \dots, 2r-1$ and $\mu_{p-1}(1) = 2r$.*

Proof.
$[\Rightarrow]$: Since $(\downarrow r)\mathsf{p}[n] = \mathsf{p}[rn] = p(rn)$, $(\downarrow r)\mathcal{C}_{p-1} = \mathcal{C}_{p-1}$ and so balancing of order p implies that $\mathcal{C}_{p-1}$ is preserved by the scalar $2r$-subdivision operator $S_{\mu_{p-1}}$. In particular, $S_{\mu_{p-1}} u_{p-1}$ is polynomial, hence the result by Proposition 1.24 and its Corollary 1.25.

$[\Leftarrow]$: From Proposition 1.24 and Proposition 1.22, we have that $\mathcal{C}_{p-1} = \mathcal{T}(u_{p-1})$ is invariant by $S_{\mu_{p-1}}$. Since furthermore $(\downarrow r)\mathcal{C}_{p-1} = \mathcal{C}_{p-1}$, we get that $\mathcal{C}_{p-1}$ is invariant by $S_{\mu_{p-1}}(\downarrow r)$ and so by L, hence the result. ∎

We will now prove that balancing implies that the eigenvectors of L are also derived from an Appell sequence of polynomials.

Proposition 1.31 *Balancing of order p is equivalent to the existence of an Appell sequence $\{\rho_n(t)\}_{n\geq 0}$ such that the polynomial signals $\mathsf{v}_n[l] := \rho_n(\frac{l}{r})$ are eigenvectors of L for the eigenvalues 2^{-n}, i.e. they satisfy $L\mathsf{v}_n = 2^{-n}\mathsf{v}_n$ for $n = 0, \dots, p-1$.*

Proof.
$[\Rightarrow]$: This is an adaptation of the proof of Theorem 1.28. All we have to prove is that there

exists an Appell sequence of polynomials $\{\rho_k\}_k$ verifying $\mathscr{S}_{\mu_{p-1}}\rho_k(rt) = 2^{-k}\rho_k(t)$, since this will imply by the isomorphisms of the commutative diagram (1.39) that $S_{\mu_{p-1}}(\downarrow r)v_k = 2^{-k}v_k$.

Consider the functional equation:

$$\begin{cases} f(2z) = \frac{1}{2r}(\mu_{p-1})_*(e^{\frac{z}{r}})f(z) \\ f(0) = 1. \end{cases} \tag{1.44}$$

Since $(\mu_{p-1})_*(e^{\frac{z}{r}})$ is an entire function and $(\mu_{p-1})_*(e^{\frac{z}{r}})_{|z=0} = 2r$, then there exists an unique solution such that $f(0) = 1$ and $f(z)$ is analytic at $z = 0$. Let $h(z) := \frac{1}{f(z)} = \sum_{k\geq 0} h_k z^k$, it satisfies

$$\begin{cases} h(z) = \frac{1}{2r}(\mu_{p-1})_*(e^{\frac{z}{r}})h(2z) \\ h(0) = 1. \end{cases} \tag{1.45}$$

We introduce the Appell sequence $\{\rho_k\}_k$ associated to $h(z)$.
Now, we will verify that $\mathscr{S}_{\mu_{p-1}}\rho_k(rt) = 2^{-k}\rho_k(t)$. Namely, from (1.38),

$$\begin{aligned}
\sum_{k\geq 0} 2^{-k}\frac{1}{k!}\rho_k(t)z^k &= e^{t\frac{z}{2}}h(\tfrac{z}{2}) = \tfrac{1}{2r}e^{t\frac{z}{2}}(\mu_{p-1})_*(e^{\frac{z}{2r}})h(z) \\
&= \tfrac{1}{2r}e^{t\frac{z}{2}}h(z)\sum_n \mu^*_{p-1}[n]e^{-n\frac{z}{2r}} = \tfrac{1}{2r}h(z)\sum_n \mu^*_{p-1}[n]e^{z\frac{rt-n}{2r}} \\
&= \tfrac{1}{2r}\sum_n \mu^*_{p-1}[n](h(z)e^{z\frac{rt-n}{2r}}) = \tfrac{1}{2r}\sum_n \mu^*_{p-1}[n]\sum_{l\geq 0}\frac{1}{l!}\rho_l(\tfrac{rt-n}{2r})z^l \\
&= \sum_{l\geq 0}\frac{1}{l!}(\tfrac{1}{2r}\sum_n \mu_{p-1}[n]\rho_l(\tfrac{rt-n}{2r}))z^l = \sum_{l\geq 0}\frac{1}{l!}\mathscr{S}_{\mu_{p-1}}\rho_l(rt)z^l.
\end{aligned}$$

Then, for $k = 0, \ldots, N$, $\mathscr{S}_{\mu_{p-1}}\rho_k(t) = 2^{-k}\rho_k(\frac{t}{r})$. Now, by the isomorphisms of the commutative diagram, taking $v_k[l] := \rho_k(\frac{l}{r})$, we get for $k = 0, \ldots, N$, that $S_{\mu_{p-1}}(\downarrow r)v_k = 2^{-k}v_k$ and so $Lv_k = 2^{-k}v_k$.

[$\Leftarrow$]: Since $\operatorname{span}\{v_n \mid 0 \leq n \leq p-1\} = \mathcal{C}_{p-1}$, we get that $\mathcal{C}_{p-1}$ is invariant by L. ■

1.4.2 Balanced vanishing moments

From the definition of vanishing moments in [Plonka, 1997; Plonka and Strela, 1998], we introduce

Definition 1.32 *A multifilter bank has balanced vanishing moments of order p iff there exist an Appell sequence $\{\rho_n(t)\}_{n\geq 0}$ such that for $n = 0, \ldots, p-1$, the lowpass synthesis refinement* *mask* $\mathbf{M}(z)$ *has the following vanishing moments*

$$\begin{aligned} \sum_{k=0}^{n} \binom{n}{k} \mathbf{y}_k^*[0](2j)^{k-n} \tfrac{d^{n-k}}{d\omega^{n-k}}[\mathbf{M}(e^{j\omega})]\Big|_{\omega=0} &= 2^{-n}\mathbf{y}_n^*[0] \\ \sum_{k=0}^{n} \binom{n}{k} \mathbf{y}_k^*[0](2j)^{k-n} \tfrac{d^{n-k}}{d\omega^{n-k}}[\mathbf{M}(e^{j\omega})]\Big|_{\omega=\pi} &= \mathbf{0}^\top \end{aligned} \tag{1.46}$$

with $\mathbf{y}_n[0] := \left[\rho_n(\frac{0}{r}), \quad \rho_n(\frac{1}{r}), \ldots, \rho_n(\frac{r-1}{r})\right]^\top$.

Now, by a straightforward adaptation of the proof of Theorem 3.2 in [Plonka and Strela, 1998], we get

Proposition 1.33 *If there exist an Appell sequence* $\{\rho_n(t)\}_{n\geq 0}$ *such that the polynomial signals* $\mathsf{v}_n[l] := \rho_n(\frac{l}{r})$ *satisfy* $\mathsf{L}\mathsf{v}_n = 2^{-n}\mathsf{v}_n$ *for* $n = 0, \ldots, p-1$*, then* $\mathbf{M}(z)$ *has balanced vanishing moments of order* p*.*

1.4.3 Equivalences on balancing

Let

$$\mathbf{E}(z) := (1-z^{-1})\mathbf{\Delta}^{-1}(z) = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ z^{-1} & 1 & 1 & \ddots & \vdots \\ \vdots & z^{-1} & \ddots & \ddots & 1 \\ \vdots & \vdots & \ddots & 1 & 1 \\ z^{-1} & z^{-1} & \ldots & z^{-1} & 1 \end{bmatrix} \tag{1.47}$$

where $\mathbf{\Delta}(z)$ is the finite differences matrix

$$\mathbf{\Delta}(z) = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 1 & -1 \\ -z^{-1} & 0 & \ldots & 0 & 1 \end{bmatrix}, \tag{1.48}$$

it is easily seen that $\mathbf{E}(z)\boldsymbol{\pi}(z) = \frac{1-z^{-r}}{1-z^{-1}}\boldsymbol{\pi}(z)$. We then have the following useful lemma

Lemma 1.34 $\forall n \geq 0$*, we can factorize*

$$\boldsymbol{\alpha}_n^\top(z)\mathbf{\Delta}^{n+1}(z) = (1-z^{-1})^{n+1}\boldsymbol{\gamma}_n^\top(z) \tag{1.49}$$

where $\boldsymbol{\gamma}_n(z) \in \mathbb{Q}^r[z^{-1}]$ *i.e. vector polynomial in* z^{-1}*.*

Proof. By induction on n,
$n = 0$:

$$\boldsymbol{\alpha}_0^\top(z)\boldsymbol{\Delta}^1(z) = \begin{bmatrix}1, & \dots, & 1\end{bmatrix}\begin{bmatrix} 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 1 & -1 \\ -z^{-1} & 0 & \dots & 0 & 1 \end{bmatrix} = (1 - z^{-1})\begin{bmatrix}1, & 0, & \dots, & 0\end{bmatrix}.$$

Now, assume for $k = 0, \dots, n-1$, that

$$\boldsymbol{\alpha}_k^\top(z)\boldsymbol{\Delta}^{k+1}(z) = (1 - z^{-1})^{k+1}\boldsymbol{\gamma}_k^\top(z)$$

with $\boldsymbol{\gamma}_k(z)$ polynomial. Then since $\boldsymbol{\alpha}_k(z) = \boldsymbol{\alpha}_{k-1}(z) + (1 - z^{-1})^k\boldsymbol{\Gamma}_k$, we get that

$$\begin{aligned}\boldsymbol{\alpha}_n^\top(z)\boldsymbol{\Delta}^{n+1}(z) &= \boldsymbol{\alpha}_{n-1}^\top(z)\boldsymbol{\Delta}^{n+1}(z) + (1 - z^{-1})^n\boldsymbol{\Gamma}_n^\top\boldsymbol{\Delta}^{n+1}(z) \\ &= (1 - z^{-1})^n(\boldsymbol{\gamma}_{n-1}^\top(z) + \boldsymbol{\Gamma}_n^\top\boldsymbol{\Delta}^n(z))\boldsymbol{\Delta}(z).\end{aligned}$$

Now, it is easily computed that $\boldsymbol{\gamma}_{n-1}^\top(1) + \boldsymbol{\Gamma}_n\boldsymbol{\Delta}^n(1) = [1, \dots, 1]^\top$. So, there exists $\boldsymbol{\gamma}_n(z) \in \mathbb{Q}^r[z^{-1}]$, such that

$$(\boldsymbol{\gamma}_{n-1}^\top(z) + \boldsymbol{\Gamma}_n^\top\boldsymbol{\Delta}^n(z))\boldsymbol{\Delta}(z) = (1 - z^{-1})\boldsymbol{\gamma}_n^\top(z). \tag{1.50}$$

Thus, $\boldsymbol{\alpha}_n^\top(z)\boldsymbol{\Delta}^{n+1}(z) = (1 - z^{-1})^{n+1}\boldsymbol{\gamma}_n^\top(z)$ with $\boldsymbol{\gamma}_n(z)$ polynomial, hence the result. ■

We have

Theorem 1.35 *Balancing of order p is equivalent to any of the following conditions:*

- $\mathbf{B0_p}$. *There exists an Appell sequence $\{\rho_n(t)\}_{n\geq 0}$ such that the discrete-time polynomial signals $\mathsf{v}_n[l] := \rho_n(\frac{l}{r})$ are eigenvectors of L for the eigenvalues 2^{-n}, i.e. they satisfy $\mathsf{L}\mathsf{v}_n = 2^{-n}\mathsf{v}_n$ for $n = 0, \dots, p-1$.*
- $\mathbf{B1_p}$. *$\mathbf{M}(z)$ has balanced vanishing moments of order p.*
- $\mathbf{B3_p}$. *$\mu_{p-1}(z) = \boldsymbol{\alpha}_{p-1}^\top(z^{2r})\mathbf{m}(z)$ has zeros of order p at $z = e^{jk\pi/r}$ for $k = 1, \dots, 2r-1$ and $\mu_{p-1}(1) = 2r$.*
- $\mathbf{B4_p}$. *For $n = 1, \dots, p$, $\mathbf{M}(z)$ can be factored as*

$$\mathbf{M}(z) = \tfrac{1}{2^n}\boldsymbol{\Delta}^n(z^2)\mathbf{M}_{n-1}(z)\boldsymbol{\Delta}^{-n}(z) \tag{1.51}$$

with $\mathbf{M}_{n-1}(1)\begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix} = \begin{bmatrix}1\\ \vdots\\ 1\end{bmatrix}$ *and $\boldsymbol{\Delta}(z)$ defined as before.*

Proof.

- From Proposition 1.30 and Proposition 1.31, we have the equivalence between balancing of order p, $\mathrm{B0_p}$ and $\mathrm{B3_p}$.

- [$\mathrm{B0_p}\Rightarrow\mathrm{B1_p}$]: This is Proposition $\mathrm{B0_p}$.

- [$\mathrm{B1_p}\Rightarrow\mathrm{B4_p}$]: Applying Corollary 4.3. from [Plonka, 1997], we get the factorization

$$\mathbf{M}(z) = \tfrac{1}{2^p}\mathbf{C}_0(z^2)\ldots\mathbf{C}_{p-1}(z^2)\mathbf{M}_{p-1}(z)\mathbf{C}_{p-1}^{-1}(z)\ldots\mathbf{C}_0^{-1}(z) \tag{1.52}$$

  with

$$\mathbf{C}_n(z) := \begin{bmatrix} c_{0,n}^{-1} & -c_{0,n}^{-1} & 0 & \ldots & 0 \\ 0 & c_{1,n}^{-1} & -c_{1,n}^{-1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & c_{r-2,n}^{-1} & -c_{r-2,n}^{-1} \\ -z^{-1}c_{r-1,n}^{-1} & 0 & \ldots & 0 & c_{r-1,n}^{-1} \end{bmatrix} \tag{1.53}$$

  and the polynomial matrix $\mathbf{M}_{p-1}(z)$ verifying $\mathbf{M}_{p-1}(1)\mathbf{c}_{p-1} = \mathbf{c}_{p-1}$ where

$$\mathbf{c}_n^\top := \begin{bmatrix} c_{0,n}, & \ldots, & c_{r-1,n} \end{bmatrix} = 2^{-n}\begin{bmatrix} 1, & \ldots, & 1 \end{bmatrix}$$

  obtained recursively from $\mathbf{y}_0[0], \ldots, \mathbf{y}_{p-1}[0]$. So, for $n = 0, \ldots, p-1$, we get $\mathbf{C}_n(z) = 2^n\mathbf{\Delta}(z)$ and $\mathbf{M}_{n-1}(1)[1, \ldots, 1]^\top = [1, \ldots, 1]^\top$.

- [$\mathrm{B4_p}\Rightarrow\mathrm{B3_p}$]: First, we give a digest of the proof in the case $r = 2$, for $p = 2, 3$ (case $p = 1$ is a consequence of Theorem 1.10).

  - For $p = 2$, we have

$$\begin{aligned} 2(m_0(z) + \alpha_{1,2}^{(1)}(z^4)m_1(z)) &= 2m_0(z) + (3 - z^{-4})m_1(z) \\ &= \begin{bmatrix} 2, & 3 - z^{-4} \end{bmatrix}\mathbf{M}(z^2)\begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \\ &= \tfrac{1}{4}\left(\tfrac{1-z^{-4}}{1-z^{-1}}\right)^2\begin{bmatrix} 2, & -1 \end{bmatrix}\mathbf{M}_1(z^2)\begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \end{aligned} \tag{1.54}$$

  - For $p = 3$,

$$\begin{aligned} &8(m_0(z) + \alpha_{1,2}^{(2)}(z^4)m_1(z)) \\ &\qquad = 8m_0(z) + (15 - 10z^{-4} + 3z^{-8})m_1(z) \\ &\qquad = \tfrac{1}{8}\left(\tfrac{1-z^{-4}}{1-z^{-1}}\right)^3\begin{bmatrix} 8 + 3z^{-4}, & -9 \end{bmatrix}\mathbf{M}_2(z^2)\begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \end{aligned} \tag{1.55}$$

For the general case: using Lemma 1.34 and the hypothesis,

$$\begin{aligned}
\mu_{p-1}(z) &= \boldsymbol{\alpha}_{p-1}^{\top}(z^{2r})2\mathbf{M}(z^r)\boldsymbol{\pi}(z) \\
&= \tfrac{1}{2^{p-1}}\boldsymbol{\alpha}_{p-1}^{\top}(z^{2r})\boldsymbol{\Delta}^p(z^{2r})\mathbf{M}_{p-1}(z^r)\boldsymbol{\Delta}^{-p}(z^r)\boldsymbol{\pi}(z) \\
&= \tfrac{1}{2^{p-1}}(1-z^{-2r})^p\boldsymbol{\gamma}_{p-1}^{\top}(z^{2r})\mathbf{M}_{p-1}(z^r)\tfrac{1}{(1-z^{-r})^p}\mathbf{E}(z^r)\boldsymbol{\pi}(z) \\
&= \tfrac{1}{2^{p-1}}(1-z^{-2r})^p\boldsymbol{\gamma}_{p-1}^{\top}(z^{2r})\mathbf{M}_{p-1}(z^r)\tfrac{1}{(1-z^{-1})^p}\boldsymbol{\pi}(z) \\
&= \tfrac{1}{2^{p-1}}\left(\tfrac{1-z^{-2r}}{1-z^{-1}}\right)^p\boldsymbol{\gamma}_{p-1}^{\top}(z^{2r})\mathbf{M}_{p-1}(z^r)\boldsymbol{\pi}(z).
\end{aligned} \tag{1.56}$$

Now, we get the result from the fact $\boldsymbol{\gamma}_{p-1}^{\top}(z^{2r})\mathbf{M}_{p-1}(z^r)\boldsymbol{\pi}(z)$ is clearly polynomial. ■

Remark 1.36 Using the equivalence between conditions $\mathrm{B1}_p$ and $\mathrm{B3}_p$, condition $\mathrm{B1}_p$ (balanced vanishing moments of order p) can be weakened in a more elegant form:

$\mathbf{B1}^*_{\mathbf{p}}$. $\boldsymbol{\alpha}_{p-1}^{\top}(1)\mathbf{M}(1) = \boldsymbol{\alpha}_{p-1}^{\top}(1)$ and $\frac{d^n}{d\omega^n}[\boldsymbol{\alpha}_{p-1}^{\top}(e^{j2\omega})\mathbf{M}(e^{j\omega})]\Big|_{\omega=\pi} = \mathbf{0}^{\top}$ for $n = 0, \dots, p-1$.

Again, these conditions can be seen as a *matrix* version of Strang-Fix conditions (Proposition 1.24).

Proof.
[$\mathrm{B4_p}\Rightarrow\mathrm{B1^*_p}$]: $\boldsymbol{\alpha}_{p-1}^{\top}(1)\mathbf{M}(1) = [\,1\cdots1\,]\,\mathbf{M}(1) = [\,1\cdots1\,] = \boldsymbol{\alpha}_{p-1}^{\top}(1)$. From the factorization, we have

$$\boldsymbol{\alpha}_{p-1}^{\top}(z^2)\mathbf{M}(z) = \tfrac{1}{2}\boldsymbol{\alpha}_{p-1}^{\top}(z^2)\boldsymbol{\Delta}^p(z^2)\mathbf{M}_{p-1}(z)\boldsymbol{\Delta}^{-p}(z).$$

By Lemma 1.34, $\boldsymbol{\alpha}_{p-1}^{\top}(z)\boldsymbol{\Delta}^p(z) = (1-z^{-1})^p\boldsymbol{\gamma}_{p-1}^{\top}(z)$ with $\boldsymbol{\gamma}_{p-1}(z)$ polynomial in z^{-1}. Thus,

$$\begin{aligned}
\boldsymbol{\alpha}_{p-1}^{\top}(z^2)\mathbf{M}(z) &= (1-z^{-2})^p\boldsymbol{\gamma}_{p-1}^{\top}(z^2)\mathbf{M}_{p-1}(z)\boldsymbol{\Delta}^p(z) \\
&= (\tfrac{1-z^{-2}}{1-z^{-1}})^p\boldsymbol{\gamma}_{p-1}^{\top}(z^2)\mathbf{M}_{p-1}(z)\mathbf{E}^p(z) \\
&= (1+z^{-1})^p\boldsymbol{\gamma}_{p-1}^{\top}(z^2)\mathbf{M}_{p-1}(z)\mathbf{E}^p(z).
\end{aligned}$$

Since all the terms of $\boldsymbol{\gamma}_{p-1}^{\top}(z^2)\mathbf{M}_{p-1}(z)\mathbf{E}^p(z)$ are polynomials in z^{-1}, then $\boldsymbol{\alpha}_{p-1}^{\top}(z^2)\mathbf{M}(z)$ have a zero of order p in $z=-1$, hence the result.

[$\mathrm{B1^*_p}\Rightarrow\mathrm{B3_p}$]: $\boldsymbol{\alpha}_{p-1}^{\top}(z^2)\mathbf{M}(z)$ has a zero of order p at $z=-1$. Taking for $l = 1, \dots, 2r-1$, $\zeta_{2r}^l = \exp(j\frac{2\pi}{2r}l)$, we get that $z = \zeta_{2r}^l$ is a zero of order p of $\boldsymbol{\alpha}_{p-1}^{\top}(z^{2r})\mathbf{M}(z^r)\boldsymbol{\pi}(z) = \mu_{p-1}(z)$, hence the result. ■

Chapter 2

Going continuous-time

In the previous chapter, we gave a detailed study of multifilter banks and their major properties. However, we gave little consideration to the process of iterations of the multifilter bank. Multifilter banks are rarely applied only once on an input signal. We usually expect that by iterating the analysis on the coarse signal, we may get some better extraction of the fundamental information conveyed by the signal. This is of great importance for applications like compression or denoising. Here, we will look at the iterations of the multifilter bank on the lowpass branch and show that under some natural conditions, it leads to a function of L_r^2 (called the scaling function) that generates a multiresolution analysis (MRA) of L^2 by the functional equation it satisfies. By looking closely at the scaling function and the MRA, a lot can then be said about some other properties of the multifilter bank: regularity, smoothness and interpolating properties are the most important ones.

2.1 From multifilters to multiwavelets

Here, we will first construct the scaling function by the cascade algorithm. It will help us analyzing the convergence of the subdivision scheme associated to the lowpass synthesis operator. Finally, we will describe the MRA generated by the scaling function and introduce the associated multiwavelets.

2.1.1 Transition and bracket operators

For $\boldsymbol{\alpha} \in c_{r\times r}^{oo}$, we introduce the transition operator,

$$\begin{aligned} \daleth_{\boldsymbol{\alpha}} : \mathscr{T}(\ell_r^\infty) &\to \mathscr{T}(\ell_r^\infty) \\ \mathsf{T}_\gamma &\mapsto \daleth_{\boldsymbol{\alpha}} \mathsf{T}_\gamma := \tfrac{1}{2} \mathsf{A}_{\boldsymbol{\alpha}} \mathsf{T}_\gamma \mathsf{S}_{\boldsymbol{\alpha}} \end{aligned} \tag{2.1}$$

We first have to prove that this operator maps Toeplitz operators into Toeplitz operators. Indeed, for $\mathsf{T}_\gamma \in \mathscr{T}(\ell_r^\infty)$, clearly $\daleth_{\boldsymbol{\alpha}} \mathsf{T}_\gamma \in \mathscr{L}(\ell_r^\infty)$. Furthermore, $(\daleth_{\boldsymbol{\alpha}} \mathsf{T}_\gamma)\boldsymbol{\tau} = \frac{1}{2}\mathsf{A}_{\boldsymbol{\alpha}} \mathsf{T}_\gamma \mathsf{S}_{\boldsymbol{\alpha}} \boldsymbol{\tau} = \frac{1}{2}\mathsf{A}_{\boldsymbol{\alpha}} \mathsf{T}_\gamma \boldsymbol{\tau}_2 \mathsf{S}_{\boldsymbol{\alpha}} = \frac{1}{2}\mathsf{A}_{\boldsymbol{\alpha}} \boldsymbol{\tau}_2 \mathsf{T}_\gamma \mathsf{S}_{\boldsymbol{\alpha}} = \frac{1}{2}\boldsymbol{\tau} \mathsf{A}_{\boldsymbol{\alpha}} \mathsf{T}_\gamma \mathsf{S}_{\boldsymbol{\alpha}} = \boldsymbol{\tau}(\daleth_{\boldsymbol{\alpha}} \mathsf{T}_\gamma)$, so $\daleth_{\boldsymbol{\alpha}} \mathsf{T}_\gamma \in \mathscr{T}(\ell_r^\infty)$.

Proposition 2.1 *For $\boldsymbol{\alpha}, \boldsymbol{\gamma} \in c^{oo}_{r\times r}$, we have that $\daleth_{\boldsymbol{\alpha}} T_{\boldsymbol{\gamma}} = T_{\frac{1}{2}(\downarrow 2)(\check{\boldsymbol{\alpha}}*\boldsymbol{\gamma}*\boldsymbol{\alpha}^*)}$*

Proof. Taking $T_{\boldsymbol{\gamma}} \in \mathscr{T}(\ell^\infty_r)$, $A_{\boldsymbol{\alpha}} T_{\boldsymbol{\gamma}} S_{\boldsymbol{\alpha}} = (\downarrow 2) T_{\check{\boldsymbol{\alpha}}} T_{\boldsymbol{\gamma}} T_{\boldsymbol{\alpha}^*} (\uparrow 2) = (\downarrow 2) T_{\check{\boldsymbol{\alpha}}*\boldsymbol{\gamma}*\boldsymbol{\alpha}^*}(\uparrow 2)$. Then, for $\boldsymbol{x} \in \ell^\infty_r$, let $\boldsymbol{y} := T_{\check{\boldsymbol{\alpha}}*\boldsymbol{\gamma}*\boldsymbol{\alpha}}(\uparrow 2)\boldsymbol{x}$. Letting $\boldsymbol{\beta} := \check{\boldsymbol{\alpha}} * \boldsymbol{\gamma} * \boldsymbol{\alpha}^*$, we get that $\boldsymbol{y}[n] = \sum_k \boldsymbol{\beta}[n-2k]\boldsymbol{x}[k]$. Now, $\daleth_{\boldsymbol{\alpha}} T_{\boldsymbol{\gamma}} \boldsymbol{x}[n] = \frac{1}{2}((\downarrow 2)\boldsymbol{y})[n] = \frac{1}{2} y[2n]$, then $\daleth_{\boldsymbol{\alpha}} T_{\boldsymbol{\gamma}} \boldsymbol{x}[n] = \frac{1}{2}\sum_k \boldsymbol{\beta}[2n-2k]\boldsymbol{x}[k] = \sum_k (\frac{1}{2}(\downarrow 2)\boldsymbol{\beta})[n-k]\boldsymbol{x}[k]$, hence the result. ■

Remark 2.2 Let $T_{\boldsymbol{\beta}} := \daleth_{\boldsymbol{\alpha}} T_{\boldsymbol{\gamma}}$. From above, we have $\boldsymbol{\beta} = \frac{1}{2}(\downarrow 2)(\check{\boldsymbol{\alpha}} * \boldsymbol{\gamma} * \boldsymbol{\alpha}^*)$ and so, we get in the z-domain, for $z = e^{j\omega}$,

$$\boldsymbol{\beta}(e^{-j2\omega}) = \boldsymbol{\alpha}(e^{j\omega})\boldsymbol{\gamma}(e^{-j\omega})\boldsymbol{\alpha}^*(e^{j\omega}) + \boldsymbol{\alpha}(e^{j(\omega+\pi)})\boldsymbol{\gamma}(e^{-j(\omega+\pi)})\boldsymbol{\alpha}^*(e^{j(\omega+\pi)})). \tag{2.2}$$

Proposition 2.3 *Let $\boldsymbol{\alpha} \in c^{oo}_{r\times r}$ such that* $\operatorname{supp} \boldsymbol{\alpha} \subset [0, K]$ *then*

$$\mathscr{E}_{\boldsymbol{\alpha}} := \{T_{\boldsymbol{\gamma}} \in \mathscr{T}(\ell^\infty_r) \mid \operatorname{supp} \boldsymbol{\gamma} \subset [-K, K]\} \tag{2.3}$$

is a finite dimensional subspace of $\mathscr{T}(\ell^\infty_r)$ invariant by $\daleth_{\boldsymbol{\alpha}}$.

Proof. Clearly, $\mathscr{E}_{\boldsymbol{\alpha}}$ is a subspace of $\mathscr{T}(\ell^\infty_r)$ having finite dimension $\dim \mathscr{E}_{\boldsymbol{\alpha}} = r(2K+1)$. Now, for $T_{\boldsymbol{\gamma}} \in \mathscr{E}_{\boldsymbol{\alpha}}$, let $T_{\boldsymbol{\beta}} := \daleth_{\boldsymbol{\alpha}} T_{\boldsymbol{\gamma}}$. From the assumptions, we get that $\operatorname{supp} \boldsymbol{\gamma} * \boldsymbol{\alpha}^* \subset [-K, 2K]$ and so that $\operatorname{supp} \check{\boldsymbol{\alpha}} * \boldsymbol{\gamma} * \boldsymbol{\alpha}^* \subset [-2K, 2K]$. Since $\boldsymbol{\beta} = \frac{1}{2}(\downarrow 2)(\check{\boldsymbol{\alpha}} * \boldsymbol{\gamma} * \boldsymbol{\alpha}^*)$, we get that $\operatorname{supp} \boldsymbol{\beta} \subset [-K, K]$, hence the result. ■

From this last result, we introduce $\tilde{\daleth}_{\boldsymbol{\alpha}}$ as the restriction $\daleth_{\boldsymbol{\alpha}|\mathscr{E}_{\boldsymbol{\alpha}}}$ of $\daleth_{\boldsymbol{\alpha}}$ to $\mathscr{E}_{\boldsymbol{\alpha}}$. Furthermore, since $\mathscr{E}_{\boldsymbol{\alpha}}$ has finite dimension, by taking an orthonormal basis of $\mathscr{E}_{\boldsymbol{\alpha}}$, we can represent every $X \in \mathscr{E}_{\boldsymbol{\alpha}}$ by a vector $\mathbf{x}$ in this basis and $\tilde{\daleth}_{\boldsymbol{\alpha}}$ by its associated matrix $\mathbf{G}_{\boldsymbol{\alpha}}$ in this basis. Then, $\daleth_{\boldsymbol{\alpha}} X$ is naturally represented by $\mathbf{G}_{\boldsymbol{\alpha}}\mathbf{x}$. Assuming that the transition operator $\daleth_{\boldsymbol{\alpha}}$ satisfies

Condition E: its associated matrix $\mathbf{G}_{\boldsymbol{\alpha}}$ has all its eigenvalues $|\lambda| < 1$ except for a simple eigenvalue $\lambda = 1$,

we get that $\mathbf{G}_{\boldsymbol{\alpha}}$ has a block-Jordan factorization of the form

$$\mathbf{G}_{\boldsymbol{\alpha}} = \mathbf{R} \begin{bmatrix} 1 & \\ & \mathbf{J} \end{bmatrix} \mathbf{R}^{-1} = \begin{bmatrix} \mathbf{x} & \vdots \end{bmatrix} \begin{bmatrix} 1 & \\ & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{y}^* \\ \dots \end{bmatrix} \tag{2.4}$$

where $\mathbf{J}$ is a matrix composed of Jordan blocks and $\mathbf{x}$ and $\mathbf{y}$ are resp. the right and left normalized ($\mathbf{y}^*\mathbf{x} = 1$) eigenvectors of $\mathbf{G}_{\boldsymbol{\alpha}}$ for the simple eigenvalue $\lambda = 1$. Furthermore, since all other eigenvalues $|\lambda| < 1$, then $\lim_{n\to\infty} \mathbf{J}^n = \mathbf{0}$ and so $\lim_{n\to\infty} \mathbf{G}^n_{\boldsymbol{\alpha}} = \mathbf{x}\mathbf{y}^*$. We write $\mathbf{G}^\infty_{\boldsymbol{\alpha}} := \mathbf{x}\mathbf{y}^*$, this defines an operator $\tilde{\daleth}^\infty_{\boldsymbol{\alpha}}$ on $\mathscr{E}_{\boldsymbol{\alpha}}$. We have then the following proposition,

Proposition 2.4 *For any $X_0 \in \mathscr{E}_{\boldsymbol{\alpha}}$, the sequence $X_n := \tilde{\daleth}^n_{\boldsymbol{\alpha}} X_0$ converges (in the sense of any norm on the finite space $\mathscr{E}_{\boldsymbol{\alpha}}$) to $\tilde{\daleth}^\infty_{\boldsymbol{\alpha}} X_0 \in \mathscr{E}_{\boldsymbol{\alpha}}$.*

Now, for $\boldsymbol{\phi}, \boldsymbol{\psi} \in L_r^2$ with compact support, we define $\Phi_t := [\dots, \boldsymbol{\phi}(t+1), \boldsymbol{\phi}(t), \boldsymbol{\phi}(t-1), \dots]^\top$ and Ψ_t in the same way. We introduce the bracket operator,

$$[\Phi, \Psi] := \int \Phi_t \Psi_t^* dt. \tag{2.5}$$

$[\Phi, \Psi]$ is an infinite size block matrix with coefficients $[\Phi, \Psi]_{k,l} = \int \boldsymbol{\phi}(t-k)\boldsymbol{\psi}^*(t-l)dt = \int \boldsymbol{\phi}(t)\boldsymbol{\psi}^*(t+k-l)dt \in \mathbb{C}^{r\times r}$. Thus, taking $\boldsymbol{\alpha}[n] := \int \boldsymbol{\phi}(t)\boldsymbol{\psi}^*(t+n)dt$, we have $\boldsymbol{\alpha} \in c_{r\times r}^{oo}$ (because of the finite supports of $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$) and we can identify $[\Phi, \Psi]$ with the Toeplitz operator $T_{\boldsymbol{\alpha}} \in \mathscr{T}(\ell_r^\infty)$. Besides, for $L \in \mathscr{L}(\ell_r^\infty)$, we have that $[L\Phi, \Psi] = \int L\Phi_t\Psi_t^* dt = L\int \Phi_t\Psi_t^* dt$ (by linearity), so, $[L\Phi, \Psi] = L[\Phi, \Psi]$. Similarly, we get $[\Phi, L\Psi] = [\Phi, \Psi]L^*$. Intuitively, the bracket operator can be seen as a measurement of the correlation between the two finitely generated shift invariant spaces $\operatorname{span}\{\phi_i(t-k) \mid 0 \le i \le r-1, k \in \mathbb{Z}\}$ and $\operatorname{span}\{\psi_i(t-k) \mid 0 \le i \le r-1, k \in \mathbb{Z}\}$. When $\boldsymbol{\psi} = \boldsymbol{\phi}$, it then characterizes the orthogonality of the basis $\{\phi_i(t-k) \mid 0 \le i \le r-1, k \in \mathbb{Z}\}$.

2.1.2 Iterations and convergence

In this paragraph, we will show that the previous result on the convergence of the transition operator implies under some natural conditions the existence of a limit function $\boldsymbol{\phi}(t)$ that characterize the multifilter $\mathbb{M}$. The approach taken here is a continuation of the work of Strang [1996] and Turcajová and Kautsky [1995]. Assuming a PR biorthogonal multifilter bank $\mathbb{M}$ with multifilters $\tilde{\mathbf{M}}, \mathbf{M}$ and $\tilde{\mathbf{N}}, \mathbf{N}$, we first have

Lemma 2.5 *Assuming that $\mathbb{M}$ is polynomial preserving of order 1, then there exists $\mathbf{u} \in \mathbb{C}^r$ with $\mathbf{u}^*\mathbf{u} = 1$, such that $\mathbf{u}^*\mathbf{M}(1) = \mathbf{u}^*$ and $\mathbf{u}^*\mathbf{M}(-1) = \mathbf{0}^\top$.*

Proof. Let V and F be the vectorization and folding operators associated to $\mathbb{M}$, and let $L := FS_{\mathbf{M}}V$, then polynomial preservation of order 1 means that constant signals are preserved by L, i.e. $Lu_0 = u_0$, i.e. $FS_{\mathbf{M}}Vu_0 = u_0$. Now, since u_0 is polynomial, we get $S_{\mathbf{M}}Vu_0 = Vu_0$. Clearly $Vu_0 \neq 0$ and from $\tau u_0 = u_0$, we derive $Vu_0 = V\tau_r u_0 = \boldsymbol{\tau} Vu_0$. Then, $\exists \mathbf{u} \neq \mathbf{0}$ such that $Vu_0 = \left[\mathbf{u}\right] := [\dots, \mathbf{u}, \mathbf{u}, \mathbf{u}, \dots]^\top$. So, $S_{\mathbf{M}}\left[\mathbf{u}\right] = \left[\mathbf{u}\right]$, i.e. $\forall n, \sum_k \mathbf{M}^*[n-2k]\mathbf{u} = \mathbf{u}$. This gives $\mathbf{u}^* \sum_k \mathbf{M}[2k+1] = \mathbf{u}^* \sum_k \mathbf{M}[2k] = \mathbf{u}^*$, so $\mathbf{u}^*\mathbf{M}(1) = \mathbf{u}^*$ and $\mathbf{u}^*\mathbf{M}(-1) = \mathbf{0}^\top$. Besides, we can always normalize $\mathbf{u}$ such that $\mathbf{u}^*\mathbf{u} = 1$. ∎

Remark 2.6 This also implies that $\exists \mathbf{v} \in \mathbb{C}^r$ such that $\mathbf{M}(1)\mathbf{v} = \mathbf{v}$ and $\mathbf{u}^*\mathbf{v} = 1$. Thus, $A_{\mathbf{M}}\left[\mathbf{v}\right] = 2\left[\mathbf{v}\right]$. Furthermore, from $S_{\mathbf{M}}^* = A_{\mathbf{M}}$, we get $\left[\mathbf{u}\right]^* A_{\mathbf{M}} = \left[\mathbf{u}\right]^*$ and $\left[\mathbf{v}\right]^* S_{\mathbf{M}} = 2\left[\mathbf{v}\right]^*$.

We will, from now on, assume that the multifilter bank is polynomial preserving of order 1 and that $\lambda = 1$ is a simple eigenvalue for $\mathbf{M}(1)$. This is often abbreviated as *Condition A_1*. We also impose that the transition operator associated to $\mathbf{M}$ satisfies *Condition E*. Under these conditions, we will prove that the cascade algorithm, as defined below, converges.

The *cascade* algorithm:

- Start from $\boldsymbol{\phi}^{(0)} \in L^1_r \cap L^2_r$ such that $\int \boldsymbol{\phi}^{(0)}(t)dt = \mathbf{v}$ and $\operatorname{supp} \boldsymbol{\phi}^{(0)} \subset [0, K]$.
- Iterate on n,

$$\boldsymbol{\phi}^{(n+1)}(t) := \sum_k \mathbf{M}[k]\boldsymbol{\phi}^{(n)}(2t-k). \tag{2.6}$$

Thus, we construct the sequence of functions $\{\boldsymbol{\phi}^{(n)}\}_{n\geq 0}$. Clearly, $\boldsymbol{\phi}^{(n)} \in L^1_r \cap L^2_r, \forall n$. Now assuming, $\int \boldsymbol{\phi}^{(n)}(t)dt = \mathbf{v}$ and $\operatorname{supp} \boldsymbol{\phi}^{(n)} \subset [0, K]$, we get

$$\int \boldsymbol{\phi}^{(n+1)}(t)dt = \sum_k \mathbf{M}[k] \int \boldsymbol{\phi}^{(n)}(2t-k)dt = \tfrac{1}{2}\sum_k \mathbf{M}[k] \int \boldsymbol{\phi}^{(n)}(t)dt = \mathbf{M}(1)\mathbf{v} = \mathbf{v}.$$

Also, since $\operatorname{supp} \boldsymbol{\phi}^{(n)}(2t-k) \subset [\frac{k}{2}, \frac{k}{2} + \frac{K}{2}]$ and $\operatorname{supp} \mathbf{M} \subset [0, K]$, then $\operatorname{supp} \boldsymbol{\phi}^{(n+1)} \subset [0, K]$. So, by induction on n, we get $\forall n, \boldsymbol{\phi}^{(n)} \in L^1_r \cap L^2_r, \int \boldsymbol{\phi}^{(n)}(t)dt = \mathbf{v}$ and $\operatorname{supp} \boldsymbol{\phi}^{(n)} \subset [0, K]$. From this last property, we also get that $\forall n, p \geq 0, [\Phi^{(n)}, \Phi^{(p)}] \in \mathscr{E}_{\mathbf{M}}$. Also, since

$$\boldsymbol{\phi}^{(n+1)}(t-l) = \sum_k \mathbf{M}[k]\boldsymbol{\phi}^{(n)}(2t-2l-k) = \sum_k \mathbf{M}[k-2l]\boldsymbol{\phi}^{(n)}(2t-k),$$

by rewriting this in vector form, we have $\Phi^{(n+1)}_t = \mathsf{A}_{\mathbf{M}}\Phi^{(n)}_{2t}$. Thus, for $n \geq p \geq 0$, we get

$$\begin{aligned}
[\Phi^{(n)}, \Phi^{(p)}] &= [\mathsf{A}_{\mathbf{M}}\Phi^{(n-1)}_{2t}, \mathsf{A}_{\mathbf{M}}\Phi^{(p-1)}_{2t}] = \mathsf{A}_{\mathbf{M}}[\Phi^{(n-1)}_{2t}, \Phi^{(p-1)}_{2t}]\mathsf{A}^*_{\mathbf{M}} \\
&= \mathsf{A}_{\mathbf{M}}[\Phi^{(n-1)}_{2t}, \Phi^{(p-1)}_{2t}]\mathsf{S}_{\mathbf{M}} = \tfrac{1}{2}\mathsf{A}_{\mathbf{M}}[\Phi^{(n-1)}_{t}, \Phi^{(p-1)}_{t}]\mathsf{S}_{\mathbf{M}} \\
&= \tilde{\daleth}_{\mathbf{M}}[\Phi^{(n-1)}, \Phi^{(p-1)}] = \ldots = \tilde{\daleth}^p_{\mathbf{M}}[\Phi^{(n-p)}, \Phi^{(0)}].
\end{aligned} \tag{2.7}$$

In addition,

$$\begin{aligned}
\mathbf{u}^* \sum_l [\Phi^{(i)}, \Phi^{(0)}]_{k,l} &= \mathbf{u}^* \sum_l \int \boldsymbol{\phi}^{(i)}(t-k)(\boldsymbol{\phi}^{(0)}(t-l))^* dt \\
&= \mathbf{u}^* \sum_l \int_l^{l+1} \boldsymbol{\phi}^{(i)}(t-k)\mathbf{v}^* dt \\
&= \mathbf{u}^* \int \boldsymbol{\phi}^{(i)}(t)dt\, \mathbf{v}^* = \mathbf{u}^*\mathbf{v}\mathbf{v}^* = \mathbf{v}^*,
\end{aligned}$$

and similarly,

$$\begin{aligned}
\mathbf{u}^* \sum_l [\Phi^{(0)}, \Phi^{(i)}]_{k,l} &= \mathbf{u}^* \sum_l \int \boldsymbol{\phi}^{(0)}(t-k)(\boldsymbol{\phi}^{(i)}(t-l))^* dt \\
&= \mathbf{u}^* \sum_l \int_k^{k+1} \mathbf{v}(\boldsymbol{\phi}^{(i)}(t-l))^* dt \\
&= \mathbf{u}^*\mathbf{v} \int (\boldsymbol{\phi}^{(i)}(t))^* dt = \mathbf{u}^*\mathbf{v}\mathbf{v}^* = \mathbf{v}^*.
\end{aligned}$$

Thus, we have $\forall i \geq 0$,

$$\left[\mathbf{u}\right]^* [\Phi^{(i)}, \Phi^{(0)}] = \left[\mathbf{u}\right]^* [\Phi^{(0)}, \Phi^{(i)}] = \left[\mathbf{v}\right]^*. \tag{2.8}$$

We then introduce the subset of $\mathscr{E}_{\mathbf{M}}$,

$$\mathscr{I}_{\mathbf{M}} := \{X \in \mathscr{E}_{\mathbf{M}} \mid \left[\mathbf{u}\right]^* X = \left[\mathbf{v}\right]^*\}. \tag{2.9}$$

This is the kernel of an affine map on the finite-dimensional space $\mathscr{E}_{\mathbf{M}}$. Consequently, $\mathscr{I}_{\mathbf{M}}$ is a closed convex subset of $\mathscr{E}_{\mathbf{M}}$. In addition, $\mathscr{I}_{\mathbf{M}}$ is invariant by $\tilde{\daleth}_{\mathbf{M}}$. Namely, for $X \in \mathscr{I}_{\mathbf{M}}$,

$$\left[\mathbf{u}\right]^* \tilde{\daleth}_{\mathbf{M}} X = \tfrac{1}{2} \left[\mathbf{u}\right]^* A_{\mathbf{M}} X S_{\mathbf{M}} = \tfrac{1}{2} \left[\mathbf{u}\right]^* X S_{\mathbf{M}} = \tfrac{1}{2} \left[\mathbf{v}\right]^* S_{\mathbf{M}} = \tfrac{1}{2} 2 \left[\mathbf{v}\right]^* = \left[\mathbf{v}\right]^*.$$

Furthermore, this set has the interesting property that the equation $\daleth_{\mathbf{M}} X = X$ has a unique solution in it. Namely, by condition E, the solutions of $\daleth_{\mathbf{M}} X = X$ in $\mathscr{E}_{\mathbf{M}}$ form a linear space of dimension 1. Taking any solution $X \neq O$, we have

$$\left[\mathbf{u}\right]^* X = \left[\mathbf{u}\right]^* \daleth_{\mathbf{M}} X = \tfrac{1}{2} \left[\mathbf{u}\right]^* A_{\mathbf{M}} X S_{\mathbf{M}} = \tfrac{1}{2} \left[\mathbf{u}\right]^* X S_{\mathbf{M}}.$$

But, since $\left[\mathbf{v}\right]^* S_{\mathbf{M}} = 2 \left[\mathbf{v}\right]^*$ and $\lambda = 1$ is a simple eigenvalue of $\mathbf{M}(1)$, we then get that $\left[\mathbf{u}\right]^* X = c \left[\mathbf{v}\right]^*$ for some $c \in \mathbb{C}$. Now, let's write X_∞ for the unique solution such that $c = 1$, i.e. $\left[\mathbf{u}\right]^* X_\infty = \left[\mathbf{v}\right]^*$. Thus, X_∞ is the one and only solution of $\daleth_{\mathbf{M}} X = X$ in $\mathscr{I}_{\mathbf{M}}$. This gives

Lemma 2.7 *For every $X_0 \in \mathscr{I}_{\mathbf{M}}$, the sequence $X_n := \tilde{\daleth}^n_{\boldsymbol{\alpha}} X_0$ converges to X_∞.*

Proof. By Proposition 2.4, $X_n := \tilde{\daleth}^n_{\boldsymbol{\alpha}} X_0$ converges to some $X \in \mathscr{E}_{\mathbf{M}}$. Since $\mathscr{I}_{\mathbf{M}}$ is closed and invariant by $\tilde{\daleth}_{\boldsymbol{\alpha}}$, we get that $X \in \mathscr{I}_{\mathbf{M}}$. Furthermore, by continuity, we have $X = \tilde{\daleth}^n_{\boldsymbol{\alpha}} X$, so necessarily $X = X_\infty$. ■

We then have,

Theorem 2.8 *The cascade algorithm converges in norm L^2_r to the unique solution $\boldsymbol{\phi} \in L^1_r \cap L^2_r$ of the two-scale equation,*

$$\boldsymbol{\phi}(t) = \sum_k \mathbf{M}[k] \boldsymbol{\phi}(2t - k) \tag{2.10}$$

such that $\int \boldsymbol{\phi}(t) dt = \mathbf{v}$. Furthermore, $\operatorname{supp} \boldsymbol{\phi} \subset [0, K]$.

Proof. Taking $\boldsymbol{\phi}^{(0)} := 1_{[0,1]} \mathbf{v}$, we will prove that $\{\boldsymbol{\phi}^{(n)}\}_n$ is a Cauchy sequence in L^2_r.

$$\begin{aligned}
[\Phi^{(n)} - \Phi^{(p)}, \Phi^{(n)} - \Phi^{(p)}] &= [\Phi^{(n)}, \Phi^{(n)}] - [\Phi^{(n)}, \Phi^{(p)}] - [\Phi^{(p)}, \Phi^{(n)}] + [\Phi^{(p)}, \Phi^{(p)}] \\
&= \daleth^n_{\mathbf{M}} [\Phi^{(0)}, \Phi^{(0)}] - \daleth^p_{\mathbf{M}} [\Phi^{(n-p)}, \Phi^{(0)}] - \daleth^p_{\mathbf{M}} [\Phi^{(0)}, \Phi^{(n-p)}] + \daleth^p_{\mathbf{M}} [\Phi^{(0)}, \Phi^{(0)}].
\end{aligned}$$

Furthermore, from the previous lemma, we have $\forall i \geq 0$,

$$\lim_{n\to\infty} \urcorner_{\mathbf{M}}^{n}[\Phi^{(i)}, \Phi^{(0)}] = \lim_{n\to\infty} \urcorner_{\mathbf{M}}^{n}[\Phi^{(0)}, \Phi^{(i)}] = \mathsf{X}_\infty.$$

Now, using a diagonal sequence argument, we also get for $n > p \to \infty$ that

$$\begin{aligned} \urcorner_{\mathbf{M}}^{p}[\Phi^{(n-p)}, \Phi^{(0)}] &\xrightarrow[n>p\to\infty]{} \mathsf{X}_\infty \\ \urcorner_{\mathbf{M}}^{p}[\Phi^{(0)}, \Phi^{(n-p)}] &\xrightarrow[n>p\to\infty]{} \mathsf{X}_\infty. \end{aligned}$$

Thus,

$$[\Phi^{(n)} - \Phi^{(p)}, \Phi^{(n)} - \Phi^{(p)}] \xrightarrow[n,p\to\infty]{} \mathsf{O}.$$

Taking $[\Phi^{(n)} - \Phi^{(p)}, \Phi^{(n)} - \Phi^{(p)}]_{0,0}$, we then get

$$\|\boldsymbol{\phi}^{(n)} - \boldsymbol{\phi}^{(p)}\|_2 \xrightarrow[n,p\to\infty]{} 0.$$

This mean that $\{\boldsymbol{\phi}^{(n)}\}_n$ is a Cauchy sequence in L_r^2. By completeness, there exists a unique $\boldsymbol{\phi} \in L_r^2$ such that

$$\boldsymbol{\phi}^{(n)} \xrightarrow[n\to\infty]{L_r^2} \boldsymbol{\phi}.$$

Furthermore, from (2.7), we get that $\mathsf{X}_\infty = [\Phi, \Phi]$ and so $\boldsymbol{\phi} \neq \mathbf{0}$. In addition, since $\{\boldsymbol{\phi} \in L_r^2 \mid \operatorname{supp} \boldsymbol{\phi} \subset [0, K]\}$ is a closed subspace of L_r^2, we get that $\operatorname{supp} \boldsymbol{\phi} \subset [0, K]$. So, using the Cauchy-Schwarz inequality, we get that $\boldsymbol{\phi} \in L_r^1 \cap L_r^2$ and L_r^1 convergence. Thus, we also have $\int \boldsymbol{\phi}(t)dt = \hat{\boldsymbol{\phi}}(0) = \mathbf{v}$.

Now, for the uniqueness: by continuity of the cascade operator, we get that $\boldsymbol{\phi}$ satisfies the two-scale equation,

$$\boldsymbol{\phi}(t) = \sum_k \mathbf{M}[k]\boldsymbol{\phi}(2t - k). \tag{2.11}$$

Going to Fourier domain, we have the refinement equation,

$$\hat{\boldsymbol{\phi}}(2\omega) = \mathbf{M}(e^{j\omega})\hat{\boldsymbol{\phi}}(\omega). \tag{2.12}$$

By iterating this product, we get

$$\hat{\boldsymbol{\phi}}(\omega) = \mathbf{M}^{(n)}(e^{j\omega})\hat{\boldsymbol{\phi}}(\tfrac{\omega}{2^n}) = \prod_{i=1}^{n} \mathbf{M}(e^{j\frac{\omega}{2^i}})\hat{\boldsymbol{\phi}}(\tfrac{\omega}{2^n}). \tag{2.13}$$

Now, conditions A_1 and E ensure that the product $\prod_{i=1}^{n} \mathbf{M}(e^{j\frac{\omega}{2^i}})$ converges uniformly on compact sets [Cohen et al., 1997] to $\mathbf{M}^{(\infty)}(e^{j\omega})$. Also, $\boldsymbol{\phi} \in L_r^1$ implies that $\hat{\boldsymbol{\phi}}$ is continuous, and so we get that

$$\hat{\boldsymbol{\phi}}(\omega) = \mathbf{M}^{(\infty)}(e^{j\omega})\hat{\boldsymbol{\phi}}(0) = \prod_{i=1}^{\infty} \mathbf{M}(e^{j\frac{\omega}{2^i}})\hat{\boldsymbol{\phi}}(0). \tag{2.14}$$

We get from this the uniqueness of the solution of the two-scale equation.

Now, one can prove that we have convergence of the cascade algorithm for any $\boldsymbol{\phi}^{(0)}$ satisfying the condition $\left[\mathbf{u}\right]^* \Phi_t^{(0)} = 1$ [Durand, 1996]. ■

It is easily seen that condition A_1 on $\mathbf{M}(z)$ imposes by biorthonormality that $\tilde{\mathbf{M}}(z)$ satisfies also condition A_1 with $\mathbf{v}$: $\mathbf{v}^*\tilde{\mathbf{M}}(1) = \mathbf{v}^*$ and $\mathbf{v}^*\tilde{\mathbf{M}}(-1) = \mathbf{0}$. If furthermore, $\urcorner_{\tilde{\mathbf{M}}}$ satisfies condition E, then the cascade algorithm associated to $\tilde{\mathbf{M}}(z)$ converges in norm L_r^2 to the unique solution $\tilde{\boldsymbol{\phi}} \in L_r^1 \cap L_r^2$ of the two-scale equation,

$$\tilde{\boldsymbol{\phi}}(t) = \sum_k \tilde{\mathbf{M}}[k]\tilde{\boldsymbol{\phi}}(2t-k) \tag{2.15}$$

such that $\int \tilde{\boldsymbol{\phi}}(t)dt = \mathbf{u}$. Furthermore, $\tilde{\boldsymbol{\phi}}$ is compactly supported and is biorthogonal to $\boldsymbol{\phi}$,

$$\int \tilde{\boldsymbol{\phi}}^*(t-k)\boldsymbol{\phi}(t-l)dt = \int \boldsymbol{\phi}^*(t-k)\tilde{\boldsymbol{\phi}}(t-l)dt = \delta_{k,l}. \tag{2.16}$$

Now, considering the subdivision operator associated to $\mathbf{M}(z)$, we construct the *subdivision* scheme:

- Start from $\mathbf{s}_0 \in \ell_r^\infty$.
- Iterate on n,

$$\mathbf{s}_{n+1}[k] := \sum_l \mathbf{M}^*[k-2l]\mathbf{s}_n[l]. \tag{2.17}$$

Now, for each n, we associate a function $\mathbf{f}^{(n)}(t)$, defined by

$$\mathbf{f}^{(n)}(t) = 2^{-\frac{n}{2}} \sum_k \mathbf{s}_n[k]\mathrm{sinc}(2^n t - k). \tag{2.18}$$

Now, we look at the convergence of the sequence of functions$\{\mathbf{f}^{(n)}\}_n$. More precisely, a subdivision scheme is said to be convergent iff for any starting sequence $\mathbf{s}_0 \in \ell_r^\infty$, there exist a continuous function $\mathbf{f}$ such that the sequence $\mathbf{f}_n[k] := \mathbf{f}(2^{-n}k)$ satisfies

$$\|\mathbf{f}_n - \mathbf{s}_n\|_\infty \xrightarrow[n\to\infty]{} \mathbf{o}. \tag{2.19}$$

It is in fact sufficient to study the convergence for sequences of the form $\mathbf{s}_0[k] = \delta[k]\mathbf{e}_i$ where $i = 1, \ldots, r$. Under the conditions of the previous section, we get that the scheme converges in the Fourier domain to

$$\hat{\mathbf{f}}(\omega) = \mathbf{e}_i^\top \mathbf{M}^{(\infty)}(e^{j\omega}). \tag{2.20}$$

For more details on vector subdivision schemes, the reader is referred to Michelli and Sauer [1997].

2.1.3 Multiwavelets and MRA

If furthermore $\mathbf{G}_{\mathbf{M}}$ and $\mathbf{G}_{\tilde{\mathbf{M}}}$ are invertible [Shen, 1998], $\{\phi_0(t-k), \ldots, \phi_{r-1}(t-k) \,|\, k \in \mathbb{Z}\}$ and $\{\tilde{\phi}_0(t-k), \ldots, \tilde{\phi}_{r-1}(t-k) \,|\, k \in \mathbb{Z}\}$ are dual Riesz bases of L^2. We then construct a biorthogonal multiresolution analysis of L^2 from $V_0 := \text{span}\{\phi_0(t-k), \ldots, \phi_{r-1}(t-k) \,|\, k \in \mathbb{Z}\}$. For more details, the reader is referred to Strang [1996].

Now, assuming that $\tilde{\boldsymbol{\phi}}, \boldsymbol{\phi}, \tilde{\boldsymbol{\psi}}, \boldsymbol{\psi}$ generate a biorthonormal MRA, then for $s(t) \in V_0$, we get

$$s(t) = \sum_n \mathbf{s}_0^*[n]\boldsymbol{\phi}(t-n) \quad \text{where} \quad \mathbf{s}_0[n] = \int s^*(t)\tilde{\boldsymbol{\phi}}(t)dt. \tag{2.21}$$

Then from $V_0 = V_{-1} + W_{-1}$, we get

$$s(t) = \sum_n \mathbf{s}_{-1}^*[n]\boldsymbol{\phi}(\tfrac{t}{2} - n) + \mathbf{d}_{-1}^*[n]\boldsymbol{\psi}(\tfrac{t}{2} - n) \tag{2.22}$$

hence the well known *Mallat algorithm* giving the relations between the coefficients at the analysis step

$$\mathbf{s}_{-1}[n] = \sum_k \tilde{\mathbf{M}}[k-2n]\mathbf{s}_0[k] \tag{2.23}$$

$$\mathbf{d}_{-1}[n] = \sum_k \tilde{\mathbf{N}}[k-2n]\mathbf{s}_0[k] \tag{2.24}$$

and for the synthesis, we get

$$\mathbf{s}_0[n] = \sum_k \mathbf{M}^*[n-2k]\mathbf{s}_{-1}[k] + \mathbf{N}^*[n-2k]\mathbf{d}_{-1}[k]. \tag{2.25}$$

These relations make the link between the MRA and the multifilter bank.

2.2 The regularity issue

Here, we will clarify how the property of balancing on the multifilter bank relates to the more classical notions of *regularity* on the multiresolution analysis: approximation order and polynomial reproduction.

2.2.1 Approximation order

A multiresolution analysis $\{V_n\}_n$ of L^2 is said to have *approximation order p* iff for any function $f \in H^p$, there exists a constant C_f such that $\forall n$,

$$d(f, V_n) := \inf_{g \in V_n} \|f - g\|_2 \leq C_f (2^{-n})^p. \tag{2.26}$$

For a MRA generated by the function $\boldsymbol{\phi} \in L_r^2$, we also say that $\boldsymbol{\phi}$ has approximation order p. Furthermore, in the case $\boldsymbol{\phi}$ has compact support, we can prove that approximation order p is equivalent to the property of polynomial reproduction of order p: one can reconstruct the polynomials $1, t, t^2, \ldots, t^{p-1}$ using only $\phi_0(t), \phi_1(t), \ldots, \phi_{r-1}(t)$ and their integer translates, i.e. for $n = 0, \ldots, p-1$, there exists a sequence $\mathbf{y}_n$ such that

$$t^n = \sum_k \mathbf{y}_n^*[k] \boldsymbol{\phi}(t - k). \tag{2.27}$$

For more details on the subjects of approximation order and shift-invariant spaces, the reader is referred to the classic paper by Jia and Lei [1993].

Now, assuming that the multifilter bank $\mathbb{M}$ is balanced of order p, we get that the lowpass synthesis refinement mask $\mathbf{M}(z)$ factorizes for $n = 1, \ldots, p$ as

$$\mathbf{M}(z) = \tfrac{1}{2^n} \boldsymbol{\Delta}^n(z^2) \mathbf{M}_{n-1}(z) \boldsymbol{\Delta}^{-n}(z) \tag{2.28}$$

with $\mathbf{M}_{n-1}(1)[1, \ldots, 1]^\top = [1, \ldots, 1]^\top$. So applying p times Theorem 2.6 [Plonka and Strela, 1998], we get that $\boldsymbol{\phi}(t)$ has at least approximation order p. Hence,

Proposition 2.9 *Whenever the multifilter bank $\mathbb{M}$ produces a bona-fide MRA, balancing of order p implies that the synthesis multiresolution analysis $\{V_n\}_n$ has at least approximation order p.*

We can notice that the converse is false: the DGHM [Donovan et al., 1996] multiwavelet has an approximation order of 2 but the associated multifilter bank is not even balanced, as detailed in next chapter. Nevertheless, by adding some conditions on the moments of the scaling functions, we get the following equivalence result,

Theorem 2.10 *Whenever the multifilter bank $\mathbb{M}$ produces a bona-fide MRA, balancing of order p is equivalent to the following condition:*

$\mathbf{B2_p}$. *$\boldsymbol{\phi}(t)$ has approximation order p and, for $i = 0, \ldots, r-1$, the shifted analysis scaling functions $\tilde{\phi}_i(t + \frac{i}{r})$ have identical p first moments: $\int t^n \tilde{\phi}_i(t + \frac{i}{r}) dt = \int t^n \tilde{\phi}_0(t) dt$ for $i = 0, \ldots, r-1$ and $n = 0, \ldots, p-1$.*

Proof. We recall that a refinement mask $\mathbf{M}(z)$ is said to have balanced vanishing moments of order p iff there exist an Appell sequence of polynomials $[\rho_n(t)]_n$ such that $\mathbf{M}(z)$ has, for $n = 0, \ldots, p-1$, the following vanishing moments:

$$\sum_{k=0}^{n} \binom{n}{k} \mathbf{y}_k^*[0](2j)^{k-n} \tfrac{d^{n-k}}{d\omega^{n-k}}[\mathbf{M}(e^{j\omega})]\Big|_{\omega=0} = 2^{-n}\mathbf{y}_n^*[0] \tag{2.29}$$

$$\sum_{k=0}^{n} \binom{n}{k} \mathbf{y}_k^*[0](2j)^{k-n} \tfrac{d^{n-k}}{d\omega^{n-k}}[\mathbf{M}(e^{j\omega})]\Big|_{\omega=\pi} = \mathbf{0}^\top \tag{2.30}$$

where

$$\mathbf{y}_n[0] := \begin{bmatrix} \rho_n(\frac{0}{r}) & \rho_n(\frac{1}{r}) & \ldots & \rho_n(\frac{r-1}{r}) \end{bmatrix}^\top. \tag{2.31}$$

It was proven in Theorem 1.35 that this condition on the lowpass synthesis refinement mask $\mathbf{M}(z)$ is equivalent to balancing of order p. We will prove that $\mathrm{B2_p}$ is equivalent to balanced vanishing moments of order p.

[$\mathrm{B1_p} \Rightarrow \mathrm{B2_p}$]: From the previous lemma, balancing of order p implies that $\boldsymbol{\phi}$ reproduces the polynomials up to degree $p-1$ and so has approximation order p. More precisely, for $n = 0, \ldots, p-1$, we have

$$t^n = \sum_k \mathbf{y}_n^*[k]\boldsymbol{\phi}(t-k) \quad \text{with} \quad \mathbf{y}_n[k] = \sum_{l=0}^{n} \binom{n}{l} k^{n-l}\mathbf{y}_l[0] \tag{2.32}$$

where by $\mathrm{B1_p}$, $\mathbf{y}_n[0] = [\rho_n(\frac{0}{r}), \rho_n(\frac{1}{r}), \ldots, \rho_n(\frac{r-1}{r})]^\top$. Now, by biorthonormality of $\tilde{\boldsymbol{\phi}}$ and $\boldsymbol{\phi}$, we have

$$\int t^n \tilde{\boldsymbol{\phi}}(t)dt = \mathbf{y}_n[0] = \begin{bmatrix} \rho_n(\frac{0}{r}) & \rho_n(\frac{1}{r}) & \ldots & \rho_n(\frac{r-1}{r}) \end{bmatrix}^\top. \tag{2.33}$$

Then, for $n = 0$, we get $\int \tilde{\boldsymbol{\phi}}(t)dt = \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}^\top$, i.e. for $i = 0, \ldots, r-1$, $\int \tilde{\phi}_i(t + \frac{i}{r})dt = \int \tilde{\phi}_0(t)dt = 1 = r_0$.

Assume that for $m \leq n$ and $i = 0, \ldots, r-1$, we have $\int t^n \tilde{\phi}_i(t + \frac{i}{r})dt = \int t^n \tilde{\phi}_0(t)dt = r_n$. We get then for $n+1$,

$$\begin{aligned}
\int (t + \tfrac{i}{r})^{n+1} \tilde{\phi}_i(t + \tfrac{i}{r})dt &= \int \sum_{k=0}^{n+1} \binom{n+1}{k} (\tfrac{i}{r})^{n+1-k} t^k \tilde{\phi}_i(t + \tfrac{i}{r})dt \\
&= \sum_{k=0}^{n+1} \binom{n+1}{k} (\tfrac{i}{r})^{n+1-k} \int t^k \tilde{\phi}_i(t + \tfrac{i}{r})dt \\
&= \sum_{k=0}^{n} \binom{n+1}{k} (\tfrac{i}{r})^{n+1-k} r_k + \int t^{n+1} \tilde{\phi}_i(t + \tfrac{i}{r})dt
\end{aligned}$$

and from (2.33),

$$\int (t+\tfrac{i}{r})^{n+1}\tilde{\phi}_i(t+\tfrac{i}{r})dt = \int t^{n+1}\tilde{\phi}_i(t)dx = \rho_{n+1}(\tfrac{i}{r}) = \sum_{k=0}^{n+1}\binom{n+1}{k}(\tfrac{i}{r})^{n+1-k}r_k$$

we then get $\int t^{n+1}\tilde{\phi}_i(t+\frac{i}{r})dt = r_{n+1}$. Hence, the result by induction on n.

[$\mathrm{B2_p}\Rightarrow\mathrm{B1_p}$]: Approximation order p implies that $\boldsymbol{\phi}$ reproduces the polynomials up to degree $p-1$, i.e. for $n=0,\ldots,p-1$, there exists sequences $\mathbf{y}_n$, such that

$$t^n = \sum_k \mathbf{y}_n^*[k]\phi(t-k).$$

Furthermore, we have $\mathbf{y}_n[k] = \sum_{l=0}^{n}\binom{n}{l}k^{n-l}\mathbf{y}_l[0]$, so by biorthonormality, $\mathbf{y}_n[0] = \int t^n\tilde{\boldsymbol{\phi}}(t)dt$. Now, let's define $r_n := \int t^n\tilde{\phi}_0(t)dt$, we then get

$$\begin{aligned}\int t^n\tilde{\phi}_i(t)dt &= \int (t+\tfrac{i}{r})^n\tilde{\phi}_i(t+\tfrac{i}{r})dt\\ &= \sum_{k=0}^{n}\binom{n}{k}(\tfrac{i}{r})^k\int t^{n-k}\tilde{\phi}_i(t+\tfrac{i}{r})dt\\ &= \sum_{k=0}^{n}\binom{n}{k}(\tfrac{i}{r})^k r_{n-k}.\end{aligned}$$

So, defining $\rho_n(t) := \sum_{k=0}^{n}\binom{n}{k}r_{n-k}t^k$, we get an Appell sequence $[\rho_n(t)]_n$ such that $\mathbf{y}_n[0] = [\rho_n(\frac{0}{r}), \rho_n(\frac{1}{r}), \ldots, \rho_n(\frac{r-1}{r})]^\top$ for $n=0,\ldots,p-1$. Also by Lemma 1.33, approximation order p implies that $\mathbf{M}(z)$ has vanishing moments for $n=0,\ldots,p-1$,

$$\sum_{k=0}^{n}\binom{n}{k}\mathbf{y}_k^*[0](2j)^{k-n}\tfrac{d^{n-k}}{d\omega^{n-k}}[\mathbf{M}(e^{j\omega})]\Big|_{\omega=0} = 2^{-n}\mathbf{y}_n^*[0]$$
$$\sum_{k=0}^{n}\binom{n}{k}\mathbf{y}_k^*[0](2j)^{k-n}\tfrac{d^{n-k}}{d\omega^{n-k}}[\mathbf{M}(e^{j\omega})]\Big|_{\omega=\pi} = \mathbf{0}^\top$$

i.e. $\mathbf{M}(z)$ has balanced vanishing moments of order p. ■

2.2.2 Superfunction theory

First, recall the Strang-Fix theorem for multiple generator and superfunction theory. Here, we show that in the case of multiwavelets balanced of order p, the superfunction has a simple expression.

Theorem 2.11 *Whenever the multifilter bank* $\mathbb{M}$ *produces a bona-fide MRA, balancing of order* p *is equivalent to the following condition:*

$\mathbf{B5_p}$. *Let* $\varphi_0 \in L^2(\mathbb{R})$ *defined by* $\hat{\varphi}_0(\omega) := \frac{1}{\sqrt{r}}\boldsymbol{\alpha}_{p-1}^\top(e^{j\omega})\hat{\boldsymbol{\phi}}(\omega)$, *then* φ_0 *verifies the Strang-Fix conditions of order* p: $\hat{\varphi}_0(0) = 1$ *and* $\frac{d^n}{d\omega^n}\hat{\varphi}_0(k2\pi) = 0$, *for* $n = 0, \dots, p-1$ *and* $k \neq 0$.

Proof.

[$\mathrm{B1_p^*}\Rightarrow\mathrm{B5_p}$]: First, $\hat{\varphi}_0(0) = \frac{1}{\sqrt{r}}\boldsymbol{\alpha}_{p-1}^\top(1)\hat{\boldsymbol{\phi}}(0) = \frac{1}{\sqrt{r}}[1, \dots, 1]^\top\mathbf{v} = 1$.
Now, we have by the refinement equation

$$\hat{\boldsymbol{\phi}}(2\omega) = \mathbf{M}(e^{j\omega})\hat{\boldsymbol{\phi}}(\omega)$$

that

$$\hat{\varphi}_0(\omega) := \frac{1}{\sqrt{r}}\boldsymbol{\alpha}_{p-1}^\top(e^{j\omega})\hat{\boldsymbol{\phi}}(\omega) = \frac{1}{\sqrt{r}}\boldsymbol{\alpha}_{p-1}^\top(e^{j\omega})\mathbf{M}(e^{j\frac{\omega}{2}})\hat{\boldsymbol{\phi}}(\tfrac{\omega}{2}).$$

So, from condition $\mathrm{B1_p^*}$, we get that $\frac{d^n}{d\omega^n}\hat{\varphi}_0(k2\pi) = 0$, for $n = 0, \dots, p-1$. We get the result for all $k2\pi$ by iterating the refinement since we can always write $k2\pi$ as $2^j l\pi$ with l odd.

[$\mathrm{B1_p^*}\Rightarrow\mathrm{B3_p}$]: By the Strang-Fix theorem, we get that $V(\varphi_0)$ has approximation order p. Since, $V(\varphi_0) \subset V_0$, $\boldsymbol{\phi}$ has approximation order p and so $\mathbf{M}(z)$ has vanishing moments with vectors $\mathbf{y}_n[0]$. Furthermore, we show that $\boldsymbol{\alpha}_{p-1}(z)$ imposes a structure on the $\mathbf{y}_n[0]$, so that we get in fact balanced vanishing moments.

■

Remark 2.12 $\varphi_0(t)$ is called the *superfunction* [Plonka and Amos, 1998] corresponding to $\phi_0(t)$. $\{\varphi_0(t-k)\}_{k\in\mathbb{Z}}$ generates a closed linear subspace $V(\varphi_0) \subset V_0$ having the same approximation order as $\boldsymbol{\phi}(t)$.

Recalling, for completeness, the definition of balancing of order p:

> A multifilter bank is balanced of order p iff its associated lowpass synthesis operator $\mathsf{L} = ({}_r\!\!\uparrow)\mathsf{S}_{\mathbf{M}}(\downarrow_r)$ preserves discrete-time polynomial signals of degree less than p (i.e. $\mathcal{C}_{p-1}$ is invariant by L).

We then have

Theorem 2.13 *Assuming the multifilter bank* $\mathbb{M}$ *produces a bona-fide MRA, we get that balancing of order* p *is equivalent to any of the following conditions:*

$\mathbf{B0_p}$. *There exists an Appell sequence* $\{\rho_n(t)\}_{0\le n\le p-1}$ *such that the discrete-time polynomial signals* $\mathsf{v}_n[l] := \rho_n(\frac{l}{r})$ *are eigenvectors of* L *for the eigenvalues* 2^{-n}, *i.e.* $\mathsf{L}\mathsf{v}_n = 2^{-n}\mathsf{v}_n$ *for* $n = 0, \dots, p-1$.

B1$_\mathbf{p}$. *$\mathbf{M}(z)$ has balanced vanishing moments of order p.*

B1$^*_\mathbf{p}$. *$\boldsymbol{\alpha}^\top_{p-1}(1)\mathbf{M}(1) = \boldsymbol{\alpha}^\top_{p-1}(1)$ and $\frac{d^n}{d\omega^n}[\boldsymbol{\alpha}^\top_{p-1}(e^{j2\omega})\mathbf{M}(e^{j\omega})]\Big|_{\omega=\pi} = \mathbf{0}^\top$ for $n = 0, \ldots, p-1$.*

B2$_\mathbf{p}$. *$\boldsymbol{\phi}(t)$ has an approximation order of p and for $i = 0, \ldots, r-1$, the shifted analysis scaling functions $\tilde{\phi}_i(t + \frac{i}{r})$ have identical p first moments i.e. $\int \tilde{\phi}_i(t + \frac{i}{r})t^n dt = \int \tilde{\phi}_0(t)t^n dt$ for $i = 0, \ldots, r-1$ and $n = 0, \ldots, p-1$.*

B3$_\mathbf{p}$. *The filter $\mu_{p-1}(z) = \sum_{k=0}^{r-1} \alpha_{k,r}^{(p-1)}(z^{2r})m_k(z)$ has zeros of order p at $z = e^{jk\pi/r}$ for $k = 1, \ldots, 2r-1$ and $\mu_{p-1}(1) = 2r$.*

B4$_\mathbf{p}$. *For $n = 1, \ldots, p$, $\mathbf{M}(z)$ can be factored as*

$$\mathbf{M}(z) = \tfrac{1}{2^n}\boldsymbol{\Delta}^n(z^2)\mathbf{M}_{n-1}(z)\boldsymbol{\Delta}^{-n}(z)$$

with $\mathbf{M}_{n-1}(1)[1, \ldots, 1]^\top = [1, \ldots, 1]^\top$ and $\boldsymbol{\Delta}(z)$ defined as before.

B5$_\mathbf{p}$. *Let $\varphi_0 \in L^2(\mathbb{R})$ defined by $\hat{\varphi}_0(\omega) := \frac{1}{\sqrt{r}}\boldsymbol{\alpha}^\top_{p-1}(e^{j\omega})\boldsymbol{\phi}(\omega)$, then $\varphi_0(t)$ verifies the Strang-Fix conditions of order p: $\hat{\varphi}_0(0) = 1$ and $\frac{d^n}{d\omega^n}\hat{\varphi}_0(k2\pi) = 0$, for $n = 0, \ldots, p-1$ and $k \neq 0$.*

2.2.3 MultiCoiflets

Assume the multifilter bank $\mathbb{M}$ is balanced of order p, then $\boldsymbol{\phi}(t)$ has an approximation order of p and for $i = 0, \ldots, r-1$, the shifted analysis scaling functions $\tilde{\phi}_i(t + \frac{i}{r})$ have identical p first moments i.e. $\int \tilde{\phi}_i(t + \frac{i}{r})t^n dt = \int \tilde{\phi}_0(t)t^n dt$ for $i = 0, \ldots, r-1$ and $n = 0, \ldots, p-1$. Now, if the scaling function $\tilde{\phi}_0(t)$ has furthermore $p-1$ vanishing moments (i.e. $\mu_n = \delta_n$ for $n = 0, \ldots, p-1$), we get a multiwavelet generalization of Coiflets [Daubechies, 1992]. We have then the following properties:

- $\rho_n(t) := t^n$ and $\mathbf{y}^\top_{0,n} := \left[(\frac{0}{r})^n, \ (\frac{1}{r})^n, \ldots, (\frac{r-1}{r})^n\right]$.
- $\int t^n \tilde{\phi}_i(t)dt = (\frac{i}{r})^n$ for $n = 0, \ldots, p-1$.
- $\varphi_0 \in L^2$ defined by $\hat{\varphi}_0 = \boldsymbol{\alpha}^\top_{p-1}(e^{j\omega})\boldsymbol{\phi}(\omega)$ satisfies now *extended* Strang-Fix conditions $\frac{d^n}{d\omega^n}\hat{\varphi}_0(k2\pi) = \delta_n\delta_k$ for $n = 0, \ldots, p-1$. This function is called the *canonical* [Plonka and Amos, 1998] superfunction associated to the MRA: among all functions in V_0 satisfying the extended Strang-Fix conditions, it has the smallest support.

MultiCoiflets are then constructed as balanced multiwavelets with more stringent conditions on the moments of $\phi_0(t)$. For practical design, we will use the following extension of the B1$^*_\mathrm{p}$. condition. For $n = 0, \ldots, p-1$, we have

$$\frac{d^n}{d\omega^n}[\boldsymbol{\alpha}^\top_{p-1}(e^{j2\omega})\mathbf{M}(e^{j\omega})]\Big|_{\omega=0} = \delta_n\boldsymbol{\alpha}^\top_{p-1}(1) \tag{2.34}$$

$$\frac{d^n}{d\omega^n}[\boldsymbol{\alpha}^\top_{p-1}(e^{j2\omega})\mathbf{M}(e^{j\omega})]\Big|_{\omega=\pi} = \mathbf{0}^\top. \tag{2.35}$$

2.3 Smoothness

Smoothness is a very important and well-understood property at the continuous-time level. As shown by [Blu and Unser, 1999], the smoothness is of first importance in the computation of the constant C_f (2.26) involved in the approximation order of the multiresolution . The smoothness of the scaling functions gives a good idea of the “quality” of the associated multiresolution analysis. This explains for example why splines yield better subdivision schemes. Besides, it was shown in the framework of scalar wavelets that the smoothness of the scaling function was inducing some good behavior of the filter bank. Namely, Cavaretta et al. [1991] proved that having p continuous derivatives on the scaling function implies that the associated subdivision operator preserves the polynomial signals up to degree p. Rioul [1993b] motivated his intensive analysis of smoothness by the fact that smooth wavelets produce filter banks that are less sensitive to quantization error, round-off errors or missing samples. This is of the highest importance in applications like coding or denoising. However, it is not clear that these relations still hold in the case of multiwavelets.

In this section, we will give two very different approaches to the estimation of the smoothness of a multifilter bank. In the first part, we will quickly recall the classical results extended from the scalar case. They are mostly based on iterated matrix methods. These results will be applied to our special case of balanced multiwavelets. For the general results and most of the details, the reader is referred to the works of Cohen, Daubechies and Plonka [Cohen et al., 1997]. In the second part, we will give an alternative strategy to deal with the notion of smoothness by focusing more on the multifilter bank and its equivalent time-varying subdivision scheme.

2.3.1 Iterated matrix approach

We first recall that a function is said to be Hölder continuous $\boldsymbol{\phi} \in C_r^\gamma$ where $n \leq \gamma < n+1$, if $\boldsymbol{\phi} \in C_r^n$ and there exists a constant C, such that $\forall t$ and $|h| \leq 1$, we have

$$|D^n \boldsymbol{\phi}(t+h) - D^n \boldsymbol{\phi}(t)| \leq C|h|^{\gamma - n}. \tag{2.36}$$

We also introduce the classical Sobolev smoothness,

$$s(\boldsymbol{\phi}) := \sup\{s \mid \int \|\hat{\boldsymbol{\phi}}(\omega)\|^2 (1+|\omega|^2)^s d\omega < \infty\}. \tag{2.37}$$

We have by the Sobolev inclusion property that $\gamma \geq s(\boldsymbol{\phi}) - \frac{1}{2}$.

Characterizations of the Sobolev smoothness can be done by analyzing the decay of $\hat{\boldsymbol{\phi}}(\omega)$ as $|\omega| \to \infty$. For example, we get Sobolev smoothness s by proving that for $\epsilon > 0$ arbitrarily small, we have

$$|\hat{\boldsymbol{\phi}}(\omega)| \leq C(1+|\omega|)^{-s+\epsilon}.$$

Now, in the special case the multifilter banks has balancing order p, we have the factorization for $n = 1, \dots, p$,

$$\mathbf{M}(z) = \tfrac{1}{2^n}\boldsymbol{\Delta}^n(z^2)\mathbf{M}_{n-1}(z)\boldsymbol{\Delta}^{-n}(z)$$

with $\mathbf{M}_{n-1}(1)[1,\dots,1]^\top = [1,\dots,1]^\top$. Assuming furthermore that $\rho(\mathbf{M}_{p-1}(1)) < 2$ and introducing

$$\gamma_k := \tfrac{1}{k}\log_2 \sup_{\omega\in]-\pi,\pi]} \|\mathbf{M}_{p-1}(e^{j\frac{\omega}{2}})\dots\mathbf{M}_{p-1}(e^{j\frac{\omega}{2^k}})\|, \tag{2.38}$$

we get by Theorem 4.1 [Cohen et al., 1997] that there exists a constant $C > 0$, such that $\forall\omega\in\mathbb{R}$,

$$|\hat{\boldsymbol{\phi}}(\omega)| \leq C(1+|\omega|)^{-p+\gamma_k}. \tag{2.39}$$

However, the computation of this supremum is highly impractical. Here, we introduce the heuristic of the invariant cycles that have been proved to be optimal in many cases [Cavaretta et al., 1991; Cohen and Daubechies, 1996]. Intuitively, to characterize the smoothness, we are interested in the decay as $n\to\infty$ of $\hat{\boldsymbol{\phi}}(2^{kn}\omega_0)$ for $\omega_0 \in]-\pi,\pi]$. From the convergence (2.14), we form the truncated products $\mathbf{M}_{p-1}^{(n)}(\omega) := \prod_{i=1}^n \mathbf{M}(e^{-j\frac{\omega}{2^i}})$. Evaluating these on the invariant cycle $\{\omega_0,\dots,\omega_{k-1}\}$ of $\omega\mapsto 2\omega \pmod{2\pi}$, we get

$$\begin{aligned}\mathbf{M}_{p-1}^{(kn)}(2^{kn}\omega_0) &= \prod_{i=1}^{kn}\mathbf{M}(e^{-j2^{-i}2^{kN}\omega_0})\\ &= \big(\mathbf{M}(e^{-j\omega_{k-1}})\dots\mathbf{M}(e^{-j\omega_0})\big)^n\end{aligned} \tag{2.40}$$

then we study the asymptotic behavior of this product by looking at the eigenvalues of

$$\mathbf{M}(e^{-j\omega_{k-1}})\dots\mathbf{M}(e^{-j\omega_0}) = \mathbf{U}_k\Lambda_k\mathbf{U}_k^\top \tag{2.41}$$

where $\Lambda_k = \operatorname{diag}(\lambda_0^{(k)},\lambda_1^{(k)},\dots,\lambda_{r-1}^{(k)})$. If $\rho(\Lambda_k) = \max\{|\lambda_0^{(k)}|,|\lambda_1^{(k)}|,\dots,|\lambda_{r-1}^{(k)}|\} \geq 2^{-ki}$ then the scaling functions cannot have Sobolev exponent of more than i and so cannot be more than $\lfloor i-1/2\rfloor$ times continuously differentiable [Eirola, 1992; Heller and Wells, 1996]. Thus, we get an upper-bound on the smoothness:

Proposition 2.14 *If an orthonormal multiwavelet system has balancing order p and the spectral radius of* $\mathbf{M}_{p-1}(z)$ *in the factorization* (1.51) *verifies* $\rho(\mathbf{M}_{p-1}(1)) < 2$*, then defining*

$$\gamma_k := \frac{1}{k}\log_2\rho(\mathbf{M}_{p-1}(e^{-j\omega_{k-1}})\dots\mathbf{M}_{p-1}(e^{-j\omega_0})) \tag{2.42}$$

with $\{\omega_0,\dots,\omega_{k-1}\}$ *invariant cycles of* $\omega\mapsto 2\omega \pmod{2\pi}$*, and* $\gamma := \inf_k \gamma_k$*, we get that* $\boldsymbol{\phi}(t)$ *is at most* $p-\gamma-\frac{1}{2}\rfloor$ *Hölder continuous (and has at most Sobolev exponent* $s = p-\gamma$*).*

As proved in some simple cases [Cavaretta et al., 1991; Cohen and Daubechies, 1996; Heller and Wells, 1996], the supremum

$$\sup_{\omega \in]-\pi,\pi]} \|\mathbf{M}_{p-1}(e^{j\frac{\omega}{2}}) \ldots \mathbf{M}_{p-1}(e^{j\frac{\omega}{2^k}})\|$$

is usually attained on invariant cycles. Furthemore, it is often achieved on the smallest length invariant cycle. One can then take $s = p - \gamma$ for the smallest invariant cycle as a good estimate of the Sobolev exponents of $\boldsymbol{\phi}(t)$ and so $\boldsymbol{\psi}(t)$.

For example, in the case of the Haar multiwavelet (multiplexed scalar Haar filter [Vetterli and Strang, 1994]), with $\omega_0 = 2\pi/3$, $\lambda_0 = 0, \lambda_1 = \frac{1}{4}$, it then proves that the scaling functions cannot be continuous. In the case of the DGHM multiwavelet, $\lambda_0 = \frac{1}{100}, \lambda_1 = \frac{1}{4^2}$, it proves that the scaling functions can be at most C^1. DGHM scaling functions and wavelets are in fact Lipschitz.

Cohen et al. [1997] developed another method using the transition operator. This method gives the exact Sobolev smoothness of $\boldsymbol{\phi}(t)$ and $\boldsymbol{\psi}(t)$. Another approach giving a good lower bound of the Sobolev smoothness for each scaling function $\phi_i(t)$ is detailed in [Plonka and Amos, 1998].

2.3.2 Discrete time approach

As insinuated above, it seems that in the case of multiwavelets, the concept of smoothness in continuous-time hardly transfer to discrete-time. First, the result of Cavaretta et al. [1991] cannot be generalized. In the multiwavelet case, the preservation of polynomial signals is e-quivalent to the property of balancing and it is easy to construct very smooth multiwavelets that don't give balanced multifilter banks. In addition, Selesnick [2000] pointed out that the continuous-time smoothness associated to a refinement mask was not even invariant by shifts on the coefficients of the refinement mask. This later subject has also been investigated by Blu [1993] in the framework of rational filter banks with the concept of *amnesia*.

Consequently, we need a new notion of smoothness that is more related to the discrete-time properties of the multifilter bank on scalar signals. A good candidate for this notion is to consider the behavior of the subdivision operator on smooth signals and particularly polynomial signals. This behavior is furthermore easy to analyze since on discrete-time polynomial signals, the lowpass synthesis operator L (with its intricate time-varying structure) is in fact equivalent to a scalar subdivision scheme (on which the classical results from the scalar wavelet theory apply). From Proposition 1.14, we have in block diagram notation that L, i.e.

$$- (\Rsh_r) = (\uparrow 2) = \boxed{2\mathbf{M}_*(z)} = ({}_r\Lsh) \; - \tag{2.43}$$

is equivalent on $\mathcal{C}_{p-1}$ to the scalar $2r$-subdivision scheme

$$- (\downarrow r) - (\uparrow 2r) - \boxed{(\mu_{p-1})_*(z)} \; - \tag{2.44}$$

Thus, on polynomial signals, iterating L is equivalent to iterating $S_{\mu_{p-1}}(\downarrow r)$. Then, using the noble identities, L^n will be equal on $\mathcal{C}_{p-1}$ to

$$- [(\downarrow r) - (\uparrow 2r)]^n - \boxed{\prod_{k=0}^{n-1}(\mu_{p-1})_*(z^{2^k})} - \tag{2.45}$$

Thus, the smoothness associated to the time-varying subdivision operator $S_{\mu_{p-1}}(\downarrow r)$ can be characterized by the smoothness of the scaling function $f_{p-1}(t)$ satisfying the two-scale equation

$$f_{p-1}(t) = \tfrac{1}{r}\sum_k \mu^*_{p-1}[k] f_{p-1}(2t-k). \tag{2.46}$$

We will call total balanced smoothness of a multifilter bank the vector $[s_\phi; s_{f_0}, \ldots, s_{f_{p-1}}]$ where s_ϕ is the smoothness of ϕ.

Rioul's method

Rioul [1993b] gave an extremely efficient method to compute the smoothness of the limit function associated to a scalar subdivision operator by looking at how the shape of the function remains stable in the cascade algorithm. We will here adapt this method to our case.

By the property of balancing of order p, the interpolation filter $\mu_{p-1}(z)$ has p zeros at $z = -1$. So we can factorize

$$\mu_{p-1}(z) = (\tfrac{1+z^{-1}}{2})^p \nu_{p-1}(z). \tag{2.47}$$

We then introduce the iterated products

$$\nu^{(i)}_{p-1}(z) := \prod_{k=0}^{i-1} \nu_{p-1}(z^{2^k}) \tag{2.48}$$

with the associated sequences $\nu^{(i)}_{p-1}[n]$.

Introducing the matrix $\mathbf{V}$,

$$\mathbf{V}_{k,l} := \nu_{p-1}[2k-l+1], \tag{2.49}$$

for $0 \leq k, l \leq N_{p-1} - 3$ with $N_{p-1} := \text{length}\, \nu_{\text{p}-1}(\text{z})$. The Hölder exponent $s_{f_{p-1}}$ associated to $\mu_{p-1}(z)$ satisfies the following inequality:

$$p + \tfrac{1}{i}\log_2 \max_{0\leq n \leq 2^i} \sum_k |\nu^{(i)}_{p-1}[n+2^i k]| \leq s_{f_{p-1}} \leq p - \log_2 \max\{|\nu_{p-1}[0]|, |\nu_{p-1}[N_{p-1}]|, \rho(\mathbf{V})\} \tag{2.50}$$

with the lower bound converging quickly as $i \to \infty$ to the exact Hölder exponent associated to $\mu_{p-1}(z)$. For more details, the reader is referred to [Rioul, 1992, 1993b].

We mention finally, that one could also put other notions of smoothness on $S_{\mu_{p-1}}(\downarrow r)$. Namely, the existence of smooth derivatives for the scaling function associated to $\mu_{p-1}(z)$ is not a necessity. It has even been noticed in applications like compression that having rather a slowly varying limit function may be more important. This leads to concepts like bounded variations. Odegard and Burrus [1996] gave an interesting highlight on this issue.

Chapter 3

Design and applications

In this chapter, we will at last get *practical* and show how to construct high order balanced multifilters with different nice properties. After having first highlighted the limitations of direct design (modifying existing unbalanced multifilters, complex filters design), we will introduce the hardcore algebra we need to analyze and solve the systems of polynomial equations that we face in the design of families of high order balanced multifilters. Finally, we will detail an application of high order balanced multifilters to image coding.

3.1 Straight design

Here we give some simple schemes for the design of balanced multifilters. We also show the limits of these approaches and some surprising minimality result on the design of high order balanced orthonormal multifilters.

3.1.1 Balancing the unbalanced

Many multifilters leading to multiwavelets have been constructed these last years. Nevertheless, none has acquired the celebrity of the one designed by Geronimo et al. [1994]; Strang and Strela [1995]; Donovan et al. [1996] that were the first to exhibit that it was possible to design an orthonormal, finite length multifilter with symmetric filters that lead to smooth, compactly supported, orthonormal, (anti)symmetric scaling functions and wavelets generating a multiresolution analysis.

$$\mathbf{M}(z) := \frac{1}{2}\left(\begin{bmatrix} \frac{3}{5} & \frac{4\sqrt{2}}{5} \\ -\frac{\sqrt{2}}{20} & -\frac{3}{10} \end{bmatrix} + \begin{bmatrix} \frac{3}{5} & 0 \\ \frac{9\sqrt{2}}{20} & 1 \end{bmatrix} z^{-1} + \begin{bmatrix} 0 & 0 \\ \frac{9\sqrt{2}}{20} & -\frac{3}{10} \end{bmatrix} z^{-2} + \begin{bmatrix} 0 & 0 \\ -\frac{\sqrt{2}}{20} & 0 \end{bmatrix} z^{-3}\right) \tag{3.1}$$

However, at the same time, this multifilter turned out to be rather disappointing for applications: it required tedious pre/postfiltering of the signals. As we know from the previous chapters, this is mainly due to the fact that this multifilter is not balanced. We will show here that with some

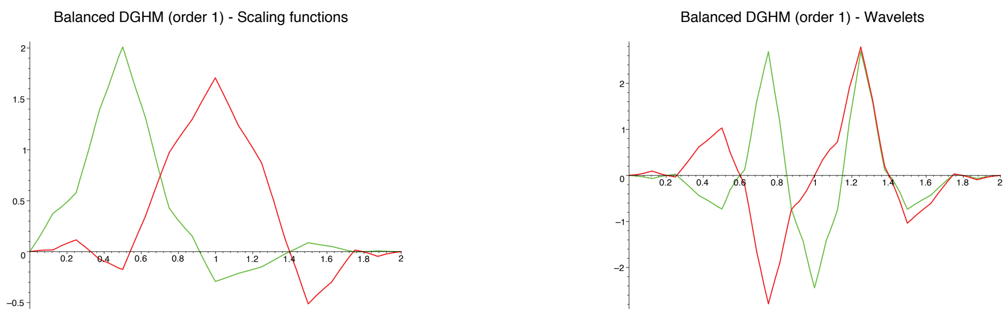


Figure 3.1: Balancing of order 1 of the DGHM multifilter: orthogonality and compact support are maintained, the symmetries are preserved on the wavelets but are lost on the scaling functions.

relaxation in the properties, we can derive an orthonormal balanced multifilter from $\mathbf{M}(z)$. Namely, we have that $[\sqrt{2}, 1]\mathbf{M}(1) = [\sqrt{2}, 1]$ and we would like in fact $[1, 1]$ to be a left eigenvector associated with eigenvalue 1 of $\mathbf{M}(1)$. The way to achieve this is then to introduce a unitary transform of the refinement mask. So let introduce a unitary matrix $\mathbf{R}$ such that $\frac{1}{\sqrt{2}}[1, 1]\mathbf{R} = \frac{1}{\sqrt{3}}[\sqrt{2}, 1]$. This gives

$$\mathbf{R} = \frac{1}{\sqrt{6}} \begin{bmatrix} 1+\sqrt{2} & 1-\sqrt{2} \\ -1+\sqrt{2} & 1+\sqrt{2} \end{bmatrix} \quad \text{or} \quad \mathbf{R} = \frac{1}{\sqrt{6}} \begin{bmatrix} -1+\sqrt{2} & 1+\sqrt{2} \\ 1+\sqrt{2} & 1-\sqrt{2} \end{bmatrix},$$

we then get

$$[1, 1]\mathbf{R}\mathbf{M}(1)\mathbf{R}^\top = [1, 1]. \tag{3.2}$$

Then defining the new refinement mask

$$\mathbf{P}(z) := \mathbf{R}\mathbf{M}(z)\mathbf{R}^\top \tag{3.3}$$

and the new two-scale equation

$$\hat{\phi}_P(2\omega) = \mathbf{P}(e^{j\omega})\hat{\phi}_P(\omega) \tag{3.4}$$

we get that $[1, 1]$ is a left eigenvector of $\mathbf{P}(1)$ for $\lambda_0(1) = 1$ and since the transformation is unitary, $\mathbf{P}(z)$ is orthonormal. Furthermore, $\hat{\phi}_P(0) = [1, 1]^\top$. We notice that in the iteration, $\mathbf{R}$ and $\mathbf{R}^\top$ cancel, except for the first and last term. The convergence of the iterated matrix product (2.12) for $\mathbf{M}(z)$ imply then the convergence for $\mathbf{P}(z)$ and the smoothness and approximation order are also unchanged. Moreover, the whole orthogonality of the filter bank is maintained

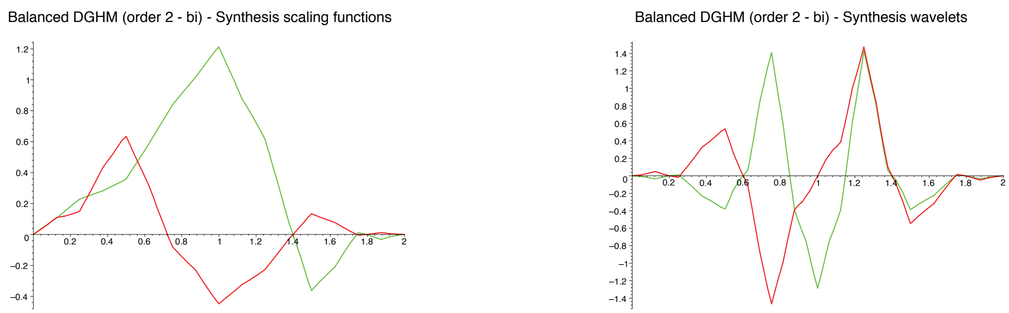


Figure 3.2: Balancing of order 2 of the DGHM multifilter: orthogonality is lost for the scaling functions, the symmetries and orthogonality are again preserved on the wavelets.

and although the symmetry of the scaling functions is usually lost, the symmetry / antisymmetry of the multiwavelets can be maintained, by taking for the highpass refinement mask $\mathbf{Q}(z) := \mathbf{N}(z)\mathbf{R}^\top$. Namely,

$$\begin{aligned}\hat{\psi}_P(\omega) &= \mathbf{N}(e^{j\frac{\omega}{2}})\mathbf{R}^\top \left[\prod_{i=2}^{\infty} \mathbf{R}\mathbf{M}(e^{j\frac{\omega}{2^i}})\mathbf{R}^\top\right] \begin{bmatrix}1\\1\end{bmatrix} \\ &= \mathbf{N}(e^{j\omega})\mathbf{M}_\infty(\omega)\hat{\phi}_M(0) = \hat{\psi}_M(\omega)\end{aligned} \tag{3.5}$$

We then obtained orthogonal, compactly supported scaling functions and symmetric multiwavelets as seen in Fig. 3.1.

One can generalize what was done above for balancing non-balanced multifilters to higher order balancing. Namely, in the case of DGHM, since we have approximation order of 2, we should be able to balance the multifilter up to order 2. Introducing the new refinement mask

$$\mathbf{P}(z) := \mathbf{A}\mathbf{M}(z)\mathbf{A}^{-1} \tag{3.6}$$

we want $\mathbf{P}(z)$ to be balanced of order 2. Setting up the system, by imposing condition $\mathrm{B3_p}$, we get a finite number of solutions for $\mathbf{A}$, the one leading to the smoothest solution being

$$\mathbf{A} := \sqrt{-4+3\sqrt{2}} \begin{bmatrix} 1 & -1 \\ \sqrt{2} & 2-\frac{1}{\sqrt{2}} \end{bmatrix}.$$

We get the new two-scale equation

$$\hat{\phi}_P(2\omega) = \mathbf{P}(e^{j\omega})\hat{\phi}_P(\omega). \tag{3.7}$$

Then, the time-varying filter bank based on this refinement mask keeps constant and linear input signals unchanged. Again, the convergence of the matrix product for $\mathbf{M}$ implies the convergence for $\mathbf{P}$ and the smoothness and approximation order are therefore unchanged. However, this time, the symmetry and orthogonality by shifts of the scaling functions is lost. Nevertheless, the system remains orthogonal in the sense that the scaling functions are orthogonal to the wavelets and so it still decorrelates coarse resolution and details. Moreover, as seen in Fig. 3.2, the symmetry/antisymmetry and orthogonality by shifts of the multiwavelets can be maintained by taking for the highpass refinement mask $\mathbf{Q}(z) := \mathbf{N}(z)\mathbf{A}^{-1}$. We notice that in this case $\widehat{\phi}_0(0) \neq \widehat{\phi}_1(0)$.

The approach developed is in fact closely linked to the design of orthogonal FIR prefiltering/postfiltering that preserve the approximation order. For a detailed exposition, we refer the reader to the papers of Hardin and Roach [1997]; Attakitmongcol et al. [1999]. This method has however some limitations in practical signal processing. Namely, the FIR, orthogonal, approximation order preserving vectorization/folding operators don't have symmetries that enables a *clean* processing of finite length signals.

In a another direction, Selesnick [2000] achieved the construction of balanced (up to order 4) DGHM *like* multifilters leading to scaling functions and wavelets that are orthgonal, compactly supported and (anti-)symmetric.

3.1.2 Daubechies complex filters

Another simple way to construct balanced multiwavelets of arbitrary order is to derive them from the complex Daubechies filters. Daubechies filters are constructed using the halfband filter

$$P(z) := c(1+z^{-1})^N(1+z)^N R(z) \tag{3.8}$$

such that $P(z) + P(-z) = 1$ with $R(e^{j\omega}) \geq 0$ and $R(e^{j\omega}) = R(e^{-j\omega})$. One gets the usual Daubechies lowpass filters: $D_N(z) := (1+z^{-1})^N B(z)$ with $B(z)$ a spectral factor of $R(z)$ with real coefficients. We can't achieve orthogonality and symmetry with real coefficients [Daubechies, 1992], however by allowing complex coefficients in the spectral factorization, one can construct symmetric, orthogonal FIR filters [Lawton, 1993; Lina and Mayrand, 1995]. Writing $[a[0], \dots, a[N], a[N], \dots, a[0]]$ for the lowpass filter, we construct the matrix coefficients

$$\mathbf{A}[i] := \begin{bmatrix} -\mathrm{Im}\, a[i] & \mathrm{Re}\, a[i] \\ \mathrm{Re}\, a[i] & \mathrm{IM}\, a[i] \end{bmatrix} \tag{3.9}$$

and the refinement mask is then

$$\mathbf{M}(z) := \frac{1}{2}\left(\sum_{k=0}^{N} \mathbf{A}[k] z^{-k} + z^{-(N+1)} \sum_{k=0}^{N} \mathbf{A}[N-k] z^{-k}\right) \tag{3.10}$$

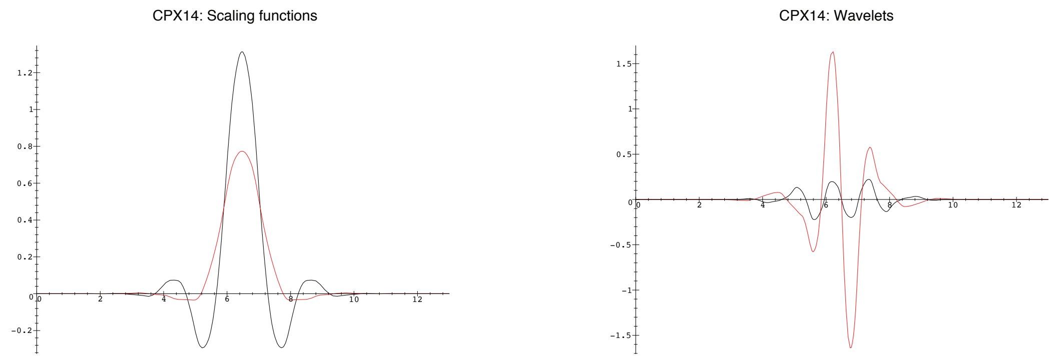


Figure 3.3: Balanced (order 1) multiwavelet derived from the complex Daubechies filters (same approximation order and smoothness as D14). Left: scaling functions, Right: wavelets.

The multifilter bank is semi-orthogonal: $V_0 \perp W_0$, but the scaling functions and wavelets are not orthogonal to their own shifted versions. For more details on semi-orthogonal multiresolution, the reader is referred to the papers of Abry and Aldroubi [1995]; Aldroubi [1997]. It is easily seen that the smoothness and approximation order of the Daubechies complex scaling function and wavelet transfer to the scaling functions and multiwavelets. Namely, by defining $\varphi(t) := \phi_1(t) + j\phi_0(t)$ where $[\phi_0(t), \phi_1(t)]$ is the multiscaling function associated to $\mathbf{M}(z)$, we get that $\varphi(t)$ verifies the two-scale equation

$$j\varphi^*(t) = \sum_{k=0}^{N} a[k]\varphi(2t-k) + \sum_{k=N+1}^{2N+1} a[2N+1-k]\varphi(2t-k) \tag{3.11}$$

so $\varphi(t)$ is the scaling function associated to the complex Daubechies filters, hence we get the same smoothness and approximation power for the scaling functions and the multiwavelets. We also easily derive that the multiscaling functions and multiwavelets are symmetric / antisymmetric as seen in Fig. 6. However, this refinement mask when iterated doesn't converge properly because $\mathbf{M}(1)$ has eigenvalues $1, -1$ with left eigenvectors $[1, 1], [1, -1]$, i.e. $\mathbf{M}(z)$ doesn't satisfy condition A_1. We get only a *constrained* [Heil and Colella, 1996] convergence, hence the poor behavior of these multiwavelets as more detailed in [Lebrun and Vetterli, 1998a].

3.1.3 Minimality

Trying to design the *shortest length* orthogonal multifilter banks for a given balancing order (no other condition imposed), we were always ending up constructing degenerated multifilters in the sense that $\mathbf{M}(z)$ was in fact a multiplexed filter:

$$\mathbf{M}(z) = \begin{bmatrix} m_{00}(z) & m_{01}(z) \\ z^{-1}m_{00}(z) & z^{-1}m_{01}(z) \end{bmatrix},$$

thus leading to wavelets and not really bona-fide multiwavelets. Looking more closely at the problem, we proved the following surprising result (for the case $r = 2$):

Theorem 3.1 *The orthonormal multifilter bank of multiplicity $r = 2$ and balancing order p with the shortest length refinement masks is a multiplexed Daubechies filter of length $2p$.*

Remark 3.2 We define the length of a matrix Laurent polynomial $\mathbf{M}(z) = \sum_{k=-N_1}^{N_2} \mathbf{M}[k]z^{-k}$ with $\mathbf{M}[-N_1] \neq \mathbf{0}$ and $\mathbf{M}[N_2] \neq \mathbf{0}$, to be $\mathrm{length}(\mathbf{M}(z)) = N_2 - N_1 + 1 = \deg(z^{N_2}\mathbf{M}(z)) + 1$. One verifies easily that $\mathrm{length}(\mathbf{M}(z)\mathbf{N}(z)) \leq \mathrm{length}(\mathbf{M}(z)) + \mathrm{length}(\mathbf{N}(z)) - 1$.

To prove the theorem, we will first prove that the minimal length condition with balancing and orthogonality implies that the refinement mask has a multiplexed filter structure. That means that the time-varying filter bank can be simplified into a scalar wavelet filter bank, the result is then easily derived.

Lemma 3.3 *Let $\mathbf{M}(z)$ be the lowpass refinement mask associated to an orthonormal multifilter bank of multiplicity $r = 2$ and balancing order p. If $\mathbf{M}(z)$ is of minimal length, then $m_1(z) = z^{-2}m_0(z)$.*

Proof. Assuming $\mathbf{M}(z)$ is the refinement mask associated with an orthonormal multiscaling function of balancing order p, we have by orthonormality condition:

$$\mathbf{M}(z)\mathbf{M}^\top(z^{-1}) + \mathbf{M}(-z)\mathbf{M}^\top(-z^{-1}) = \mathbf{I} \tag{3.12}$$

Besides, balancing of order p gives us

$$\mathbf{M}(z) = \frac{1}{2^p}\mathbf{\Delta}^p(z^2)\mathbf{M}_{p-1}(z)\mathbf{\Delta}^{-p}(z) \tag{3.13}$$

with $\mathbf{M}_{p-1}(1)\begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix}$ and $\mathbf{\Delta}(z) = \begin{bmatrix}1 & -1\\ -z^{-1} & 1\end{bmatrix}$.

Introducing $\mathbf{V}_{p-1}(z) := 2^{-p}(1-z^{-2})^p\mathbf{M}_{p-1}(z)\mathbf{\Delta}^{-p}(z)$, one gets

$$\mathbf{V}_{p-1}(z)\mathbf{V}_{p-1}^\top(z^{-1}) + \mathbf{V}_{p-1}(-z)\mathbf{V}_{p-1}^\top(-z^{-1}) = \begin{bmatrix}2 & 1+z^2\\ 1+z^{-2} & 2\end{bmatrix}^p \tag{3.14}$$

Furthermore, one can write

$$\mathbf{U}_{p-1}(z) := \begin{bmatrix}2 & 1+z^2\\ 1+z^{-2} & 2\end{bmatrix}^p = \sum_{k=-\lceil p/2\rceil}^{k=\lceil p/2\rceil} \mathbf{U}_{p-1}[k]z^{-2k} \tag{3.15}$$

with $\mathbf{U}_{p-1}[-k] = \mathbf{U}_{p-1}^\top[k]$.

Thus for $\mathbf{W}(z) = \sum_{k=-N_1}^{k=N_2} \mathbf{W}[k] z^{-k}$ to verify

$$\mathbf{W}(z) + \mathbf{W}(-z) = \mathbf{U}_{p-1}(z) \tag{3.16}$$

one needs $N_1, N_2 \geq 2\lceil p/2 \rceil$. Introducing $\mathbf{u}(z) := \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix}$, then

$$\mathbf{W}_{p-1}(z) := (1+z^{-1})^p (1+z)^p \mathbf{u}(z) \mathbf{u}^\top(z) = (1+z^{-1})^p (1+z)^p \begin{bmatrix} 1 & z \\ z^{-1} & 1 \end{bmatrix} \tag{3.17}$$

is an obvious minimal length solution of (3.16). So one has to prove now that there is no other minimal length solution. Since all even degrees of $\mathbf{W}(z)$ are uniquely determined by (3.16), all the other minimal length solutions will be of the form $\mathbf{W}(z) = \mathbf{W}_{p-1}(z) + z^{-1}\mathbf{R}(z^2)$ and should factorize as $\mathbf{W}(z) = \mathbf{V}(z)\mathbf{V}^\top(z^{-1})$. Thus,

$$z^{-1}\mathbf{R}(z^2) = \mathbf{V}(z)\mathbf{V}^\top(z^{-1}) - (1+z^{-1})^p (1+z)^p \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \begin{bmatrix} 1 & z \end{bmatrix} \tag{3.18}$$

Since $\mathbf{u}^\top(z)\mathbf{u}(-z^{-1}) = \mathbf{u}^\top(-z^{-1})\mathbf{u}(z) = 0$, multiplying (3.18) by $\mathbf{u}^\top(-z^{-1})$ on the left and by $\mathbf{u}(-z)$ on the right, we have

$$z^{-1}\mathbf{u}^\top(-z^{-1})\mathbf{R}(z^2)\mathbf{u}(-z) = \mathbf{u}^\top(-z^{-1})\mathbf{V}(z)\mathbf{V}^\top(z^{-1})\mathbf{u}(-z) \tag{3.19}$$

For $|z| = 1$, we obtain

$$z^{-1}\mathbf{u}^\top(-z^{-1})\mathbf{R}(z^2)\mathbf{u}(-z) = \left|\mathbf{V}^\top(z^{-1})\mathbf{u}(-z)\right|^2 \tag{3.20}$$

Changing $z \to -z$, we also get

$$-z^{-1}\mathbf{u}^\top(z^{-1})\mathbf{R}(z^2)\mathbf{u}(z) = \left|\mathbf{V}^\top(-z^{-1})\mathbf{u}(z)\right|^2 \tag{3.21}$$

Now again, multiplying (3.18) by $\mathbf{u}^\top(z^{-1})$ on the left and by $\mathbf{u}(z)$ on the right, we get

$$\begin{aligned} z^{-1}\mathbf{u}^\top(z^{-1})\mathbf{R}(z^2)\mathbf{u}(z) &= \mathbf{u}^\top(z^{-1})\mathbf{V}(z)\mathbf{V}^\top(z^{-1})\mathbf{u}(z) \\ &\quad - (1+z^{-1})^p(1+z)^p \begin{bmatrix} 1 & z \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \begin{bmatrix} 1 & z \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix} \end{aligned} \tag{3.22}$$

i.e. for $|z| = 1$,

$$z^{-1}\mathbf{u}^\top(z^{-1})\mathbf{R}(z^2)\mathbf{u}(z) = |\mathbf{V}^\top(z^{-1})\mathbf{u}(z)|^2 - 4(1+z^{-1})^p(1+z)^p \tag{3.23}$$

So adding equations (3.21) and (3.23), one gets

$$|\mathbf{V}^\top(z^{-1})\mathbf{u}(z)|^2 + |\mathbf{V}^\top(-z^{-1})\mathbf{u}(z)|^2 = 4(1+z^{-1})^p(1+z)^p \tag{3.24}$$

hence,

$$|\mathbf{V}^\top(z^{-1})\mathbf{u}(z)| = O((1+z^{-1})^p \tag{3.25}$$

$$|\mathbf{V}^\top(-z^{-1})\mathbf{u}(z)| = O((1+z^{-1})^p) \tag{3.26}$$

thus $|\mathbf{V}^\top(z^{-1})\mathbf{u}(-z)| = O((1-z^{-1})^p)$

By multiplying (3.18) by $\mathbf{u}^\top(z^{-1})$ on the left and by $\mathbf{u}(-z)$ on the right, we get

$$z^{-1}\mathbf{u}^\top(z^{-1})\mathbf{R}(z^2)\mathbf{u}(-z) = \mathbf{u}^\top(z^{-1})\mathbf{V}(z)\mathbf{V}^\top(z^{-1})\mathbf{u}(-z) \tag{3.27}$$

and so

$$z^{-1}\mathbf{u}^\top(z^{-1})\mathbf{R}(z^2)\mathbf{u}(-z) = O((1-z^2)^p) \tag{3.28}$$

i.e. there exits a unique Laurent polynomial $q(z)$ such that

$$k(z) := z^{-1}\mathbf{u}^\top(z^{-1})\mathbf{R}(z^2)\mathbf{u}(-z) = (1-z^2)^p q(z) \tag{3.29}$$

Furthermore, from (3.18), $z\mathbf{R}^\top(z^{-2}) = z^{-1}\mathbf{R}(z^2)$, then $k(-z^{-1}) = -k(z)$, and so $q(-z^{-1}) = -q(z)$. Hence, $q(z)$ is at least of length 3. Thus $z^{-1}\mathbf{R}(z^2)$ is at least of length $2p$ and because it is of odd length by structure, so it is at least of length $2p+1$. Hence, we have that $\mathbf{W}_{p-1}(z)$ is the unique minimal length solution.

Since $\mathbf{W}_{p-1}(z) = \mathbf{V}_{p-1}(z)\mathbf{V}_{p-1}^\top(z^{-1})$ for $\mathbf{V}_{p-1}(z) = 2^{-p}(1-z^{-2})^p\mathbf{M}_{p-1}(z)\mathbf{\Delta}^{-p}(z)$, then

$$\mathbf{M}(z)\mathbf{M}^\top(z^{-1}) = 2^{-2p}(1-z^{-2})^{-p}(1-z^2)^{-p}\mathbf{\Delta}^p(z^2)\mathbf{W}_{p-1}(z)\mathbf{\Delta}^{p\top}(z^{-2}) \tag{3.30}$$

Now, since $\mathbf{\Delta}(z^2)\mathbf{u}(z) = (1-z^{-1})\mathbf{u}(z)$, we have

$$\mathbf{M}(z)\mathbf{M}^\top(z^{-1}) = \frac{1}{2}\begin{bmatrix} 1 & z \\ z^{-1} & 1 \end{bmatrix} \tag{3.31}$$

Also, since $\det(\mathbf{W}_{p-1}(z)) = 0$, one can write

$$\mathbf{M}(z) = \begin{bmatrix} 1 \\ \lambda(z) \end{bmatrix} \begin{bmatrix} m_{00}(z) & m_{01}(z) \end{bmatrix} \tag{3.32}$$

so

$$\mathbf{M}(z)\mathbf{M}^\top(z^{-1}) = (m_{00}(z)m_{00}(z^{-1}) + m_{01}(z)m_{01}(z^{-1})) \begin{bmatrix} 1 & \lambda(z^{-1}) \\ \lambda(z) & \lambda(z)\lambda(z^{-1}) \end{bmatrix} \tag{3.33}$$

Hence $\lambda(z) = z^{-1}$ and so $m_1(z) = z^{-2}m_0(z)$. ■

We are now able to prove Theorem 3.1

Proof. Using Lemma 3.3 and the balancing order condition $\mathrm{B3_p}$, we get that

$$m_0(z) + \alpha_{1,2}^{(p-1)}(z^4)m_1(z) = (1 + z^{-2}\alpha_{1,2}^{(p-1)}(z^4))m_0(z)$$

must have zeros of order p at $j, -1, -j$.
Moreover for $z = -1$, we have

$$1 + z^{-2}\alpha_{1,2}^{(p-1)}(z^4) = 1 + (-1)^{-2}\alpha_{1,2}^{(p-1)}((-1)^4) = 1 + \alpha_{1,2}^{(p-1)}(1) = 2$$

which implies that $m_0(z)$ must have p zeros at $z = -1$. Since

$$\mathbf{M}(z) = \begin{bmatrix} m_{00}(z) & m_{01}(z) \\ z^{-1}m_{00}(z) & z^{-1}m_{01}(z) \end{bmatrix}$$

with $m_{00}(z)$ and $m_{01}(z)$ the polyphase components of $m_0(z)$, then the orthonormality condition gives that $m_0(z)$ is a real conjugate mirror filter. Then, from the well-known theorem of Daubechies [1992], this implies that $m_0(z)$ has at least $2p$ non-zero coefficients, and that the minimal length filters are Daubechies filters (classical D_{2p} or Symlets of order p). ■

This also implies that:

Corollary 3.4 *An orthonormal multifilter bank of multiplicity $r = 2$ and balancing order p has a lowpass refinement mask $\mathbf{M}(z)$ with at least $p + 1$ non-zero (2×2) taps.*

3.2 A computational algebra digest

Recently, major advances have been achieved in the field of computational algebraic geometry that lead to new efficient ways to deal with one of the central application of computer algebra: solving systems of multivariate polynomial equations. Using the new algorithms that have been developed, practical problems can now be solved exactly in a way that is very competitive with numerical methods. One of the most promising approach to solve systems of polynomial equations has been by computing Gröbner bases. We will describe here this approach (that will be applied in the next section to the design of balanced multiwavelets). At the same time, even though the computation of a Gröbner basis is the crucial point in our approach, one should not forget that it is only the first step in the solving process. Methods to implement change of ordering of the Gröbner basis, and alternative approaches like triangular systems and rational univariate representation of the system are also key tools. We will also discuss some of these methods in the following.

3.2.1 Introducing Gröbner bases

In this paragraph, we will review the major algorithms involved in the computation of Gröbner bases. We will not go much into the details, since many good books ranging from introductory [Fröberg, 1997] to advanced level [Cox et al., 1992, 1998] have been written on this now popular subject. We will rather develop an intuitive understanding of what a Gröbner basis is and describe some ways to compute them by using analogies to linear algebra.

We define a multivariate polynomial p to be a finite sum of terms $\sum_\alpha c_\alpha x^\alpha$, where a term $c_\alpha x^\alpha$ is the product of a coefficient c_α and a monomial x^α. One can draw an analogy between solving linear systems that can be seen as the study of the associated vector subspace and solving a polynomial system that can be seen as the study of the associated ideal. Namely, a polynomial system of equations is defined by a list $\{p_1, \dots, p_N\}$ of multivariate polynomials with rational coefficients ($p_1, \dots, p_N \in \mathbb{Q}[x_1, \dots, x_n]$). We associate to this system the generated ideal $I = \langle p_1, \dots, p_N \rangle$ i.e. the smallest ideal containing $p_1, \dots, p_N$ as well as $\sum_{k=1}^N h_k p_k$, for any $h_1, \dots, h_N \in \mathbb{Q}[x_1, \dots, x_n]$. Intuitively, the idea is that the polynomials $p_1, \dots, p_N$ have a common zero iff any polynomial of the ideal I vanishes also at that location. It is then equivalent to study a system of polynomial equations or the ideal generated by the polynomials.

For a set of linear equation, the Gauss elimination algorithm enables us to compute an equivalent triangular system by canceling the leading term of each equation. Here we will see that a similar algorithm can be developed for the case of multivariate polynomials. An important aspect of the Gauss elimination algorithm is in the choice of the pivots that are used during the triangularization of the system. For the same reasons, the first thing we will have to define is an ordering on the monomials (that need to be compatible with polynomial multiplication). We introduce here two monomial orderings that will be used intensively in the following:

- The *lexicographic* ordering, abbreviated lex. This is the ordering used in dictionaries:

$$x^{(\alpha_1, \cdots, \alpha_n)} <_{\text{lex}} x^{(\beta_1, \cdots, \beta_n)} \Leftrightarrow \exists n_0, \forall n \leq n_0, \alpha_n = \beta_n \text{ and } \alpha_{n_0} < \beta_{n_0}. \tag{3.34}$$

- The *degree reverse lexicographic* order, abbreviated drl. This is a modified reversed lexicographic ordering taking first into account the total degree of the polynomials:

$$x^{(\alpha_1, \cdots, \alpha_n)} <_{\text{drl}} x^{(\beta_1, \cdots, \beta_n)} \Leftrightarrow \begin{cases} \sum_k \alpha_k < \sum_k \beta_k \\ \text{or} \\ \sum_k \alpha_k = \sum_k \beta_k \text{ and } x^{(\beta_n, \cdots, \beta_1)} <_{\text{lex}} x^{(\alpha_n, \cdots, \alpha_1)}. \end{cases} \tag{3.35}$$

We then introduce the leading term $\mathrm{lt}(p, <)$ of a polynomial p as its term with the highest order according to the ordering $<$, we also introduce the leading monomial $\mathrm{lm}(p, <)$ as the leading term with a coefficient normalized to 1 and $\mathrm{lc}(p, <)$ as the leading coefficient. Notice that when no doubt remains, we will omit to mention the ordering. In a very similar way to what is done in the Gauss elimination algorithm, we introduce the *Spolynomial* as a monomial combination of two polynomials so as to cancel their leading terms.

$$\mathrm{Spol}(p_1, p_2) := \frac{\mathrm{lcm}(\mathrm{lt}(p_1), \mathrm{lt}(p_2))}{\mathrm{lm}(p_1)} p_1 - \frac{\mathrm{lcm}(\mathrm{lt}(p_1), \mathrm{lt}(p_2))}{\mathrm{lm}(p_2)} p_2 \tag{3.36}$$

where lcm stands for the *least common multiple* of a set of polynomials. For example, with the $<_{\text{lex}}$ ordering $(x > y > z)$, $p_1 = 2x^3y + \dots$ and $p_2 = x^2y^2 + \dots$, we get $\mathrm{Spol}(p_1, p_2) =$

$yp_1 - 2xp_2$. We have canceled the leading terms of p_1 and p_2. Of particular interest is when $\text{Spol}(p_1, p_2) = p_1 - qp_2$ for some polynomial q (e.g., $p_1 = 3x^3y + \dots$ and $p_2 = xy + \dots$). In that case, we say that p_1 is *reducible* by p_2 and that q is the *reduction* of p_1 by p_2. Formalizing the reduction algorithms, we get

DEFINE: $\text{Reducible}(p_1, p_2, <)$
Input: $p_1, p_2, <$
Output: Boolean
  if $\text{lm}(p_2)$ divides $\text{lm}(p_1)$ **then**
    return true
  else
    return false
  end if

DEFINE: $\text{Reduce}(p_1, p_2, <)$
Input: $p_1, p_2, <$
Output: Polynomial
  if $\text{Reducible}(p_1, p_2)$ **then**
    return $p_1 - \frac{\text{lt}(p_1)}{\text{lt}(p_2)} p_2$
  else
    return p_1
  end if

This last algorithm can easily be extended to the reduction of a polynomial by an ordered list of polynomials, $L = [q_1, \dots, q_N]$.

DEFINE: $\text{Reduce}(p, L, <)$
Input: $p, [q_1, .., q_N], <$
Output: Polynomial
  for $k = 1$ to N **do**
    if $\text{Reducible}(p, q_k)$ **then**
      $p \leftarrow \text{Reduce}(p, q_k)$
      $\text{Reduce}(p, L)$
    end if
  end for
  return p

We shall emphasize the importance of the order in which the reductions are done: the same set of polynomials reordered in a different list will usually give rise to a different output of the reduction process. We will see in the following, that for any list of polynomials there exists an equivalent list such that the order has no influence anymore.
We now introduce the famous Buchberger algorithm that transforms a general ordered list of polynomials generating the ideal I into an equivalent one that makes it much easier to deal with

the ideal generated. The list of polynomials obtained by the Buchberger algorithm is called a Gröbner basis. One of the major properties of Gröbner bases is that it makes it algorithmically easy to verify if a given polynomial belongs or not to the ideal generated.

DEFINE: Buchberger$(L, <)$
Input: $[p_1, .., p_N], <$
Output: An ordered set of polynomials
  # define the set of pairs
  $\mathscr{P} \leftarrow \{[p_i, p_j] \mid i < j\}$
  while $\mathscr{P} \neq \emptyset$ **do**
    # selection step
    choose $[p_i, p_j]$ in $\mathscr{P}$
    $\mathscr{P} \leftarrow \mathscr{P} \setminus \{[p_i, p_j]\}$
    # reduction step
    $p_{N+1} \leftarrow \mathrm{Reduce}(\mathrm{Spol}(p_i, p_j), [p_1, .., p_N])$
    # test the termination of the reduction process
    if $p_{N+1} \neq 0$ **then**
      $\mathscr{P} \leftarrow \mathscr{P} \cup \{[p_i, p_{N+1}] \mid i < N+1\}$
      $N \leftarrow N+1$
    end if
  end while
  return $[p_1, .., p_N]$

The major features of the Buchberger algorithm is that the list obtained $G := [g_1, \ldots, g_N]$ still generates I and satisfies the following property: $\mathrm{Spol}(g_i, g_j)$ reduces to 0 modulo G, for every $g_i, g_j \in G$. Such a list is called a Gröbner basis of I. It is easily seen that Gröbner bases have the following equivalent characterizations:

- $f \in I$ iff f reduces to 0 modulo G ($\mathrm{Reduce}(f, G) = 0$).
- The leading term of any element of I is divisible by at least one leading term $\mathrm{lt}(g_i)$of G.

For an ideal I, let $\langle \mathrm{LT}(I) \rangle$ denote the ideal of leading terms of I, i.e. the ideal generated by the set of leading terms $\mathrm{LT}(I) := \{cx^\alpha | \exists f \in I, \mathrm{lt}(f) = cx^\alpha\}$. We then get that $G := [g_1, \ldots, g_N]$ is a Gröbner basis of I iff the ideal of leading terms of I is generated by the leading terms of G i.e. $\langle \mathrm{LT}(I) \rangle = \langle \mathrm{lt}(g_1), \ldots, \mathrm{lt}(g_N) \rangle$.

Usually, one can compute infinitely many Gröbner bases. However, among all theses, one satisfies some nicer properties: every element g_i of the basis G has its leading term normalized (coefficient equal to 1) and $\forall g_i \in G$, no term of g_i is divisible by a leading monomial $\mathrm{lm}(g_j)$ $(j \neq i)$. This particular basis is called the *reduced* Gröbner basis: one verifies that for a given monomial ordering monomial $<$, a non empty polynomial ideal has always a unique reduced Gröbner basis.

We give here how to derive a reduced Gröbner basis from the output of the Buchberger algorithm:

DEFINE: $\mathrm{Gbasis}(L, <)$
Input: $L, <$
Output: A reduced Gröbner basis

```
# compute a Gröbner basis
[p_1, ..., p_N] ← Buchberger(L)
# eliminate the reducible elements
G ← []
for all i such that ∀j ≠ i, Reducible(p_i, p_j) is false do
    G ← G ∪ [p_i]
end for
# reduce completely and normalize G = [g_1, ..., g_N']
for all i do
    g_i ← Reduce(g_i, G \ [g_i])
    g_i ← (1/lc(g_i)) g_i
end for
return G
```

With the reduced Gröbner basis, we get the very nice feature that the output of $\mathrm{Reduce}(p, G)$ does not depend anymore on the order of the polynomials in the list: $\mathrm{Reduce}(\cdot, G)$ is the *canonical reduction* modulo I.

In the case $<$ is lexicographic, the reduced Gröbner basis has a very nice structure. Namely, the reduced Buchberger algorithm gives a union of triangular arrays of polynomials of the following form:

$$\begin{cases} h_0^{(d+1)}(x_1, \dots, x_d)(x_{d+1}^{k_{d+1}} + \frac{h_1^{(d+1)}(x_1,\dots,x_d)}{h_0^{(d+1)}(x_1,\dots,x_d)} x_{d+1}^{k_{d+1}-1} + \frac{h_2^{(d+1)}(x_1,\dots,x_d)}{h_0^{(d+1)}(x_1,\dots,x_d)} x_{d+1}^{k_{d+1}-2} + \dots) \\ h_0^{(d+2)}(x_1, \dots, x_{d+1})(x_{d+2}^{k_{d+2}} + \frac{h_1^{(d+2)}(x_1,\dots,x_{d+1})}{h_0^{(d+2)}(x_1,\dots,x_{d+1})} x_{d+2}^{k_{d+2}-1} + \frac{h_2^{(d+2)}(x_1,\dots,x_{d+1})}{h_0^{(d+2)}(x_1,\dots,x_{d+1})} x_{d+2}^{k_{d+2}-2} + \dots) \\ \dots \\ h_0^{(n)}(x_1, \dots, x_{n-1})(x_n^{k_n} + \frac{h_1^{(n)}(x_1,\dots,x_{n-1})}{h_0^{(n)}(x_1,\dots,x_{n-1})} x_n^{k_n-1} + \frac{h_2^{(n)}(x_1,\dots,x_{n-1})}{h_0^{(n)}(x_1,\dots,x_{n-1})} x_n^{k_n-2} + \dots) \end{cases} \tag{3.37}$$

where d gives the number of remaining degrees of freedom of the system when all of the equations are satisfied ($x_1, \dots, x_d$ are now parameters). d is called the algebraic dimension of the ideal: the solutions of system of polynomial equations can be seen as a geometric variety that can be classified by its algebraic dimension: $d = 0$: finite number of isolated points, $d = 1$: curves, $d = 2$: surfaces and so on. In the case the system has different kind of solutions (e.g. isolated points and curves), the global dimension is just the maximum dimension of each component.

When $d = 0$, i.e. when the system has a finite number of solutions, we get that the first equation becomes a univariate polynomial equation and we can then rewrite the reduced Gröbner basis as:

$$\begin{cases} x_1^{k_1} + g_1(x_1) & \deg_{x_1}(g_1) < k_1 \\ x_2^{k_2} + g_2(x_1, x_2) & \deg_{x_2}(g_2) < k_2 \\ \dots & \dots \\ x_n^{k_n} + g_n(x_1, x_2, \dots, x_n) & \deg_{x_n}(g_n) < k_n. \end{cases} \tag{3.38}$$

On such a system, it is now easy to carry out any of the following operations (as detailed in [Gonzalez-Vega et al., 1999]):

- Count exactly all complex/real solutions including the multiplicity.
- Isolate the real roots with the desired precision (no rounding error).
- Compute approximations of the complex roots (rounding errors).

For example, to numerically solve the system: first solve the univariate equation $x_1^{k_1} + g_1(x_1) = 0$, then recursively substitute and solve the next equations. Moreover, in the case the variable x_1 is separating (intuitively two solutions can't have the same first component; a rigorous definition is given next section), we get that $k_2 = k_3 = \dots = k_n = 1$. The system is then of the form

$$\begin{cases} \tilde{g}_1(x_1) \\ x_2 + \tilde{g}_2(x_1) \\ \dots & \dots \\ x_n + \tilde{g}_n(x_1, \dots, x_n). \end{cases} \tag{3.39}$$

and all we have to do is to solve $\tilde{g}_1(x_1) = 0$ and substitute in the other equations. This is called the *Shape lemma* case [Rouillier, 1999].

3.2.2 Linear algebra methods

The necessary time to compute a reduced Gröbner basis by the Buchberger algorithm depends strongly on the monomial ordering that is used. In general, computing a reduced Gröbner bases for the lexicographic ordering is much more time and memory consuming than computing the corresponding drl Gröbner basis. However, this additional computational cost is worth it because, as seen before, the lexicographic ordering provides a triangular like structure (similar to the one obtained by Gauss elimination) that is really suitable for further processing. Fortunately, recent algorithms enable to compute efficiently lexicographic Gröbner bases by using an alternative approach:

- First, we compute a Gröbner basis for the drl ordering, using, for example, the standard Buchberger algorithm (note that the algorithm can be highly improved by using heuristics for the choice of the critical pairs and the reductors in the reducing process). An even better approach is to completely suppress the influence of these choices, by in fact *not choosing* anymore as introduced by Faugère [1999] in his F_4 algorithm: instead of choosing one critical pair, we take a subset of critical pairs and reduce this set. By using a linear algebra approach to deal with the pairs, the algorithm can be made extremely efficient for the computation of drl Gröbner bases. An implementation named FGb of this algorithm can be tested on the Web at `https://calfor.lip6.fr`

- Finally, we compute the lexicographic Gröbner basis from the drl one by a change of ordering. For the case when the ideal is zero-dimensional, a very efficient algorithm called FGLM [Faugère et al., 1994] has been developed using again a linear algebra approach. Implementations of this algorithm are now available in most of the computer algebra programs.

We will now give some details on how the linear algebra approach works. Again, for more details, the reader can read the survey on the subject by Mourrain [1999]. Starting from a list P of polynomials such that the generated ideal $I = \langle P \rangle$ is zero-dimensional, we show that the quotient space $\mathcal{A} := \mathbb{Q}[x_1, \ldots, x_n]/I$ inherits a structure of finite-dimensional algebra. Namely, assuming a reduced Gröbner basis $G := [g_1, \ldots, g_N]$ for some ordering $<$ (typically drl), any element of $\mathcal{A}$ has the form $\bar{p} = \text{Reduce}(p, G)$ for some $p \in \mathbb{Q}[x_1, \ldots, x_n]$. Since $\langle \text{Lt}(I) \rangle = \langle \text{lt}(g_1), \ldots, \text{lt}(g_N) \rangle$, we easily construct a linear basis of $\mathcal{A}$ from the set of monomials $\{x^\alpha \mid x^\alpha \notin \langle \text{Lt}(I) \rangle\}$, by taking in increasing order the monomials under the staircase, i.e. the x^α that are not a multiple of $\text{lt}(g_i)$ (since this implies that $\bar{x^\alpha} = \text{Reduce}(x^\alpha, G) = x^\alpha$). The linear basis $B := \{\omega_1, \ldots \omega_d\}$ obtained this way is called the *monomial basis* of $\mathcal{A}$. Finally, constructing the multiplication table $[\omega_i \omega_j]_{i,j}$ of $\mathcal{A}$, we get a full description of the linear algebraic framework in which we will deal with the polynomials.

Now, any element $\bar{p} \in \mathcal{A}$ can be expressed as a vector $[p]$ since $\bar{p} = \sum_{k=1}^{d} [p]_i \omega_i$. The FGLM algorithm can then be described using linear algebra in $\mathcal{A}$. The lexicographic Gröbner basis is obtained by detecting linear combinations of monomials in $\mathcal{A}$. The idea is to construct in parallel the lex Gröbner basis $G_{<_{\text{lex}}}$ and a full rank $d \times d$ matrix $\mathbf{G}$, by scanning the monomials x^α in increasing lex ordering (starting from 1). There are two possibilities:

1. $[x^\alpha]$ is linearly dependent of the previous vectors put in $\mathbf{G}$, i.e. $[x^\alpha] = \sum_k c_k [x^{\beta_k}]$, then we add $g_\alpha := x^\alpha - \sum_k c_k x^{\beta_k}$ to $G_{<_{\text{lex}}}$ (namely, $g_\alpha \in I$ and $\text{lt}(g_\alpha) = x^\alpha$).

2. $[x^\alpha]$ is linearly independent of the previous vectors put in $\mathbf{G}$, then add $[x^\alpha]$ to $\mathbf{G}$.

Repeat the scan until $\text{rank}(\mathbf{G}) = d$; $G_{<_{\text{lex}}}$ is then a lex Gröbner basis of I.

3.2.3 The rational univariate representation

In many situation, the computation of a lex Gröbner basis of the ideal I is a bit of an overkill in the sense that in fact, all we are really interested in, is a good description of the set of solutions of the system, $\mathcal{Z}_{\mathbb{C}}(I) := \{\boldsymbol{\alpha} \in \mathbb{C}^n \,|\, \forall p \in P, p(\boldsymbol{\alpha}) = 0\}$ (we will denote by $\mu(\boldsymbol{\alpha})$ the multiplicity of a solution $\boldsymbol{\alpha}$). Here, we will describe an alternative method to the change of ordering algorithm (FGLM). In the approach developed by Gonzalez-Vega et al. [1999]; Rouillier [1999], one constructs a list $\{\chi_u(t), g_u(1,t), g_u(x_1,t), \ldots, g_u(x_1,t)\}$ of polynomials of $\mathbb{Q}[x_1, \ldots, x_n]$ such that: if $\boldsymbol{\alpha}$ is a solution of the system, then $u(\boldsymbol{\alpha})$ is a root of $\chi_u(t)$ with the same multiplicity and conversely, if ζ is a root of $\chi_u(t)$, then

$$\left[\frac{g_u(x_1,\zeta)}{g_u(1,\zeta)}, \quad \frac{g_u(x_2,\zeta)}{g_u(1,\zeta)}, \quad \ldots, \quad \frac{g_u(x_n,\zeta)}{g_u(1,\zeta)}\right] \tag{3.40}$$

is a solution of the system with the same multiplicity. Hence, $\mathcal{Z}_{\mathbb{C}}(I)$ is fully characterized.

For any polynomial u, we introduce the linear operator $\mathbf{M}_u$ on $\mathcal{A}$,

$$\begin{aligned} \mathbf{M}_u : \mathcal{A} &\to \mathcal{A} \\ \bar{f} &\mapsto \mathbf{M}_u \bar{f} := \overline{uf}. \end{aligned} \tag{3.41}$$

We will also identify $\mathbf{M}_u$ with its $\mathbb{C}^{d\times d}$ matrix representation in the monomial basis of $\mathcal{A}$. This matrix is easily computed by expressing $\bar{u}\omega_i$ in the monomial basis, this gives the i$^{\text{th}}$ column of $\mathbf{M}_u$.

From the computation of $\mathbf{M}_u$, we derive some important information on $\mathcal{Z}_{\mathbb{C}}(I)$ and the system in general. Namely, by the Stickelberger theorem [Rouillier, 1999], we get that $\mathbf{M}_u$ has eigenvalues $u(\boldsymbol{\alpha})$ with multiplicity $\sum_{\boldsymbol{\beta}\in\mathcal{Z}(I), u(\boldsymbol{\beta})=u(\boldsymbol{\alpha})} \mu(\boldsymbol{\beta})$, where $\boldsymbol{\alpha} \in \mathcal{Z}_{\mathbb{C}}(I)$. This gives that

- $\det(\mathbf{M}_u) = \prod_{\boldsymbol{\alpha}\in\mathcal{Z}_{\mathbb{C}}(I)} u(\boldsymbol{\alpha})^{\mu(\boldsymbol{\alpha})}$.
- $\mathrm{trace}(\mathbf{M}_u) = \sum_{\boldsymbol{\alpha}\in\mathcal{Z}_{\mathbb{C}}(I)} \mu(\boldsymbol{\alpha}) u(\boldsymbol{\alpha})$.
- $\chi_u(t) := \det(t\mathbf{I} - \mathbf{M}_u) = \prod_{\boldsymbol{\alpha}\in\mathcal{Z}_{\mathbb{C}}(I)} (t - u(\boldsymbol{\alpha}))^{\mu(\boldsymbol{\alpha})}$ (characteristic polynomial of $\mathbf{M}_u$).

Computing directly the characteristic polynomial (incl. the determinant) of a matrix like $\mathbf{M}_u$ can be time and memory consuming. We detail here an alternative method based on traces of matrices and taking advantage of the special structure of the matrices $\mathbf{M}_u$. Let $\chi_u(t) = \sum_{k=0}^{d} b_k t^{d-k}$ and $\chi_u'(t)$ be its derivative, then

$$\begin{aligned} \frac{\chi_u'(t)}{\chi_u(t)} &= \sum_{\boldsymbol{\alpha}\in\mathcal{Z}_{\mathbb{C}}(I)} \frac{\mu(\boldsymbol{\alpha})}{t - u(\boldsymbol{\alpha})} = \sum_{\boldsymbol{\alpha}\in\mathcal{Z}_{\mathbb{C}}(I)} \frac{1}{t} \frac{\mu(\boldsymbol{\alpha})}{1 - \frac{u(\boldsymbol{\alpha})}{t}} \\ &= \sum_{\boldsymbol{\alpha}\in\mathcal{Z}_{\mathbb{C}}(I)} \sum_{k\geq 0} \frac{1}{t} \mu(\boldsymbol{\alpha}) u^k(\boldsymbol{\alpha}) t^{-k} = \sum_{k\geq 0} \mathrm{trace}(\mathbf{M}_{u^k}) t^{-(k+1)}. \end{aligned} \tag{3.42}$$

So, we have

$$\chi'_u(t) = \chi_u(t) \sum_{k \geq 0} \mathrm{trace}(\mathbf{M}_{u^k}) t^{-(k+1)},$$

and also

$$\chi'_u(t) = \sum_{k=0}^{d-1} (d-k) b_k t^{d-1-k},$$

we then get

$$(d-k) b_k = \sum_{l=0}^{k} \mathrm{trace}(\mathbf{M}_{u^l}) b_{k-l}.$$

Thus, we can compute $\chi_u(t)$ from the scalars: $\mathrm{trace}(\mathbf{M}_{u^k})$ for $k = 0, \ldots, d$.
We also introduce the square-free part of $\chi_u(t)$:

$$\tilde{\chi}_u(t) := \prod_{\boldsymbol{\alpha} \in \mathcal{Z}_{\mathbb{C}}(I)} (t - u(\boldsymbol{\alpha})) = \frac{\chi_u(t)}{\gcd(\chi_u(t), \chi'_u(t))}. \tag{3.43}$$

Now, assume u is separating $\mathcal{Z}_{\mathbb{C}}(P)$, i.e. on $\mathcal{Z}_{\mathbb{C}}(I)$, $\boldsymbol{\alpha} \neq \boldsymbol{\beta} \Rightarrow u(\boldsymbol{\alpha}) \neq u(\boldsymbol{\beta})$ (that implies that $\mathbf{M}_u$ has eigenvalues $u(\boldsymbol{\alpha})$ with multiplicity $\mu(\boldsymbol{\alpha})$), we finally introduce

$$\begin{array}{rl} g_u : \mathcal{A} & \to \mathbb{Q}[x_1, \ldots, x_n] \\ \bar{v} & \mapsto g_u(v,t) := \sum_{\boldsymbol{\alpha} \in \mathcal{Z}_{\mathbb{C}}(I)} \mu(\boldsymbol{\alpha}) v(\boldsymbol{\alpha}) \frac{\tilde{\chi}_u(t)}{t - u(\boldsymbol{\alpha})}. \end{array} \tag{3.44}$$

This can be rewritten as

$$g_u(v,t) = \sum_{\boldsymbol{\alpha} \in \mathcal{Z}_{\mathbb{C}}(I)} \mu(\boldsymbol{\alpha}) v(\boldsymbol{\alpha}) \prod_{\boldsymbol{\beta} \in \mathcal{Z}_{\mathbb{C}}(I) \setminus \{\boldsymbol{\alpha}\}} (t - u(\boldsymbol{\beta})). \tag{3.45}$$

Taking $\boldsymbol{\alpha} \in \mathcal{Z}_{\mathbb{C}}(I)$ and $t = u(\boldsymbol{\alpha})$, we get

$$g_u(v, u(\boldsymbol{\alpha})) = \mu(\boldsymbol{\alpha}) v(\boldsymbol{\alpha}) \prod_{\boldsymbol{\beta} \in \mathcal{Z}_{\mathbb{C}}(I) \setminus \{\boldsymbol{\alpha}\}} (u(\boldsymbol{\alpha}) - u(\boldsymbol{\beta})). \tag{3.46}$$

We then get the central result of the rational univariate representation

$$\frac{g_u(v, u(\boldsymbol{\alpha}))}{g_u(1, u(\boldsymbol{\alpha}))} = v(\boldsymbol{\alpha}). \tag{3.47}$$

Hence, for $v = x_1, \ldots, x_n$, we get

$$\boldsymbol{\alpha} = \left[\frac{g_u(x_1, u(\boldsymbol{\alpha}))}{g_u(1, u(\boldsymbol{\alpha}))}, \quad \frac{g_u(x_2, u(\boldsymbol{\alpha}))}{g_u(1, u(\boldsymbol{\alpha}))}, \quad \ldots, \quad \frac{g_u(x_n, u(\boldsymbol{\alpha}))}{g_u(1, u(\boldsymbol{\alpha}))} \right]. \tag{3.48}$$

Theorem 3.5 *If $\boldsymbol{\alpha}$ is a solution of the system, then $u(\boldsymbol{\alpha})$ is a root of $\chi_u(t)$ with the same multiplicity and conversely, if ζ is a root of $\chi_u(t)$, then*

$$\left[\frac{g_u(x_1, u(\boldsymbol{\alpha}))}{g_u(1, u(\boldsymbol{\alpha}))}, \quad \frac{g_u(x_2, u(\boldsymbol{\alpha}))}{g_u(1, u(\boldsymbol{\alpha}))}, \quad \dots, \quad \frac{g_u(x_n, u(\boldsymbol{\alpha}))}{g_u(1, u(\boldsymbol{\alpha}))}\right]$$

is a solution of the system with the same multiplicity.

Now, all we have still to detail is a practical way to compute $g_u(v, t)$. In a similar way to what is done to compute $\chi_u(t)$, we get

$$\frac{g_u(v,t)}{\tilde{\chi}_u(t)} = \sum_{k \geq 0} \operatorname{trace}(\mathbf{M}_{u^k v}) t^{-(k+1)}. \tag{3.49}$$

Writing $\tilde{\chi}_u(v,t) = \sum_{k=0}^{r} a_k t^{r-k}$ and let $H_k(\tilde{\chi}_u)(t) = \sum_{l=0}^{k} a_l t^{k-l}$ be its Hörner sequence of polynomials, we then get

$$g_u(v,t) = \sum_{k=0}^{r-1} \operatorname{trace}(\mathbf{M}_{u^k v}) H_{r-1-k}(\tilde{\chi}_u)(t). \tag{3.50}$$

So, the $g_u(v,t)$ are easily computed from $(\tilde{\chi}_u(t))$ and the $\operatorname{trace}(\mathbf{M}_{u^k v})$, for $k = 0, \dots, r$. There is furthermore an easy way to compute these traces by noticing that $\operatorname{trace}(\mathbf{M}_{fg}) = \mathbb{T}\mathrm{R}(f)[g]$ where

$$\mathbb{T}\mathrm{R}(f) := \left[\operatorname{trace}(\mathbf{M}_{f\omega_1}), \quad \dots, \quad \operatorname{trace}(\mathbf{M}_{f\omega_d})\right]. \tag{3.51}$$

Now, since $\mathbb{T}\mathrm{R}(u^{k+1}) = \mathbb{T}\mathrm{R}(u^k)\mathbf{M}_u$, we get by induction $\operatorname{trace}(\mathbf{M}_{u^{k+1}}) = \mathbb{T}\mathrm{R}(u^k)[u]$ and $\operatorname{trace}(\mathbf{M}_{u^k v}) = \mathbb{T}\mathrm{R}(u^k)[v]$.

To really complete the algorithm, we introduce the matrix defined by $[\mathbb{T}\mathrm{R}\mathbf{M}]_{i,j} := \operatorname{trace}(\mathbf{M}_{\omega_i \omega_j})$, i.e.

$$\mathbb{T}\mathrm{R}\mathbf{M} := \begin{bmatrix} \operatorname{trace}(\mathbf{M}_{\omega_1\omega_1}) & \dots & \operatorname{trace}(\mathbf{M}_{\omega_1\omega_d}) \\ \vdots & & \vdots \\ \operatorname{trace}(\mathbf{M}_{\omega_d\omega_1}) & \dots & \operatorname{trace}(\mathbf{M}_{\omega_d\omega_d}) \end{bmatrix}. \tag{3.52}$$

We easily prove that $r := \#\mathcal{Z}_{\mathbb{C}}(I) = \operatorname{rank}(\mathbb{T}\mathrm{R}\mathbf{M})$. This gives us an easy way of testing whether a polynomial u is separating: $\deg(\tilde{\chi}_u)$ should be equal to r. Furthermore, we prove that the set of polynomials $\mathcal{S}(I) := \{x_1 + kx_2 + \dots + k^{n-1}x_n \mid 0 \leq k \leq (n-1)\binom{r}{2}\}$ contains at least one separating polynomial.

We finally have all the tools to efficiently compute the rational univariate representation of a zero-dimensional ideal from a reduced Gröbner basis G:

DEFINE: $\mathrm{RUR}(G)$
Input: A zero-dimensional reduced Gröbner basis G for any ordering
Output: A rational univariate representation R

```
# setup the linear framework
compute {ω_1, ..., ω_d} the monomial basis of 𝒜
compute the matrix TrM
r ← rank(TrM)
# find a separating polynomial
repeat
   choose u ∈ 𝒮(I) # by increasing lex order
   compute M_u
   for k = 1 to d do
      compute trace(M_{u^k})
   end for
   compute χ_u
   deduce χ̃_u
until deg χ̃_u = r
# compute the traces
for k = 0 to r and l = 1 to n do
   compute trace(M_{u^k x_l})
end for
# deduce the RUR
compute g_u(1,t), g_u(x_1,t), ..., g_u(x_n,t)
return R := [χ_u(t), g_u(1,t), g_u(x_1,t), ..., g_u(x_n,t)]
```

This algorithm gives us a bijection between $\mathcal{Z}_{\mathbb{C}}(P)$, the set of solutions of the system, and the roots of the univariate polynomial $\chi_u(t)$. All we have to do now to get $\mathcal{Z}_{\mathbb{C}}(P)$ is to isolate the roots of $\chi_u(t)$. We then derive the solutions of the system using the RUR. The isolation of roots is usually a difficult problem. However, in the case we are only interested in the real solutions of the system, we can locate them very efficiently by computing the signature of trace matrices [Pedersen et al., 1993].

Alternatively, we can also factorize $\chi_u(t)$. We construct this way local algebras [Gonzalez-Vega et al., 1999] that enable us to simplify the problem of isolating the roots by lowering the degrees of characteristic polynomials in the RURs.

In short, the RUR approach appears to be a very efficient alternative to the computation of a lexicographic Gröbner basis: as detailed in [Rouillier, 1999], it is computationally easier to compute a RUR than to apply the FGLM algorithm and the characteristic polynomial $\chi_u(t)$ is usually easier to deal with than $\tilde{g}_1(x_1)$, the leading polynomial of the lexicographic Gröbner basis.

3.3 Algebraic design of multiwavelets

From the previous section, we now have all the tools and algorithms to deal with and solve the systems of polynomials equations that appear when designing high order balanced multifilters. Using the results obtained in the previous chapters (especially the factorization of the refinement mask) and inspired by the techniques used by Park et al. [1996]; Faugère et al. [1998] on similar problems of design, we are now ready to investigate the construction of orthonormal multifilters of arbitrary balancing order in a similar way to what Daubechies [1992] did for her well-known filters.

3.3.1 Symmetry oriented design: the Bat family

Given a balancing order p, we are looking for the shortest length orthonormal multifilters with real coefficients and symmetries. As seen in next section, the symmetries on the filters allow easy and practical implementations on finite length signals. The scheme of construction is then the following.

First, we construct the refinement mask $\mathbf{M}(z)$, by putting degrees of freedom on a matrix $\mathbf{M}_{p-1}(z)$.

1. Impose the order of balancing to be p, i.e. for $n = 1, \ldots, p$,

$$\mathbf{M}(z) = \frac{1}{2^n}\mathbf{\Delta}^n(z^2)\mathbf{M}_{n-1}(z)\mathbf{\Delta}^{-n}(z)$$

   with $\mathbf{M}_{n-1}(1)[1, \ldots, 1]^\top = [1, \ldots, 1]^\top$. This way we reduce the number of degrees of freedom in the design.

2. Impose the condition **O** (orthonormality) on $\mathbf{M}(z)$,

$$\mathbf{M}(z)\mathbf{M}^\top(z^{-1}) + \mathbf{M}(-z)\mathbf{M}^\top(-z^{-1}) = \mathbf{I}.$$

   This gives quadratic equations on the free variables of $\mathbf{M}_{p-1}(z)$ (the idea is to introduce the Laurent polynomial matrix $\mathbf{V}_{p-1}(z) := 2^{-p}(1 - z^{-r})^p\mathbf{M}_{p-1}(z)\mathbf{\Delta}^{-p}(z)$ and translate the orthonormality condition on this matrix; for more details, the reader may look at the proof of Lemma 3.3).

3. Impose symmetries condition: here we look for flipping property on $m_0(z), m_1(z)$ (i.e. $m_1(z) = z^{-2L+1}m_0(z^{-1})$). The flipping property enables an easy lossless symmetrization (detailed in the section about image coding) of finite length input signals.

4. We now have a system of polynomial equations. We compute the algebraic dimension of the system using a drl Gröbner basis approach and increase the degree of freedom until we get solutions (and a drl Gröbner basis of dimension 0). We used here the programs

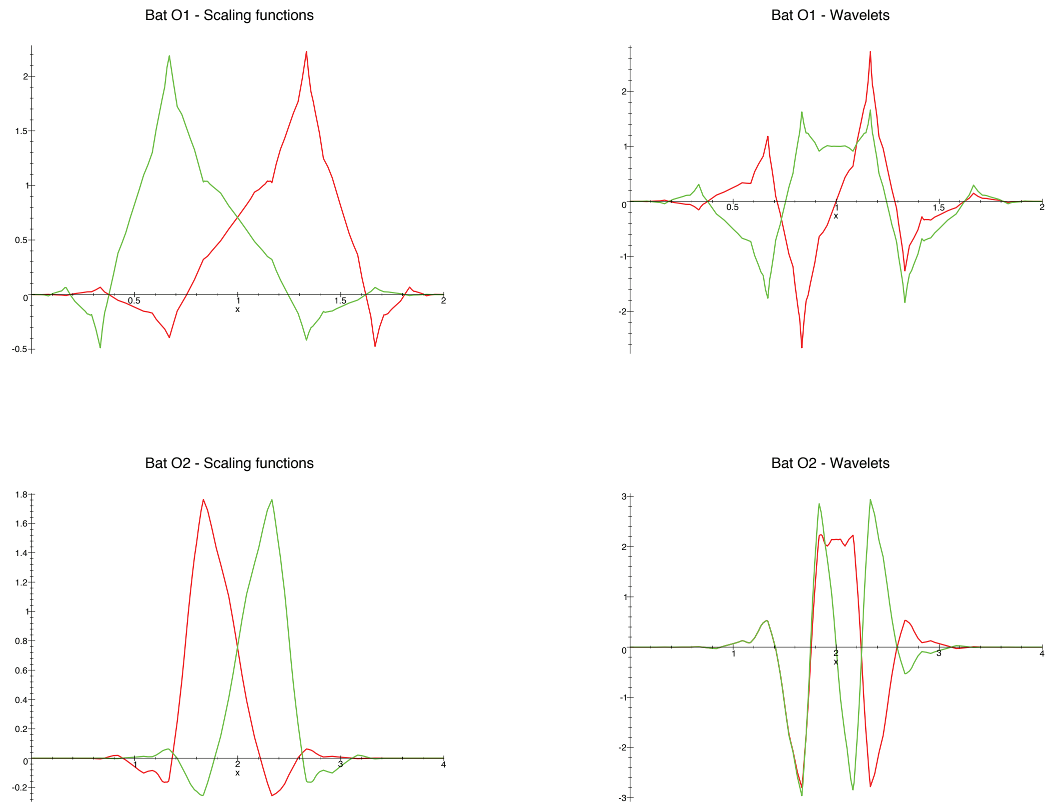


Figure 3.4: Order 1(resp. 2) of balanced orthogonal multiwavelet: the scaling functions are flipped around 1 (resp. 2), the wavelets are symmetric/antisymmetric, the length is 3(resp. 5) taps (2x2) and the estimate of the smoothness by invariant cycles gives the Sobolev exponent $s = 0.64$ (resp. 1.15).

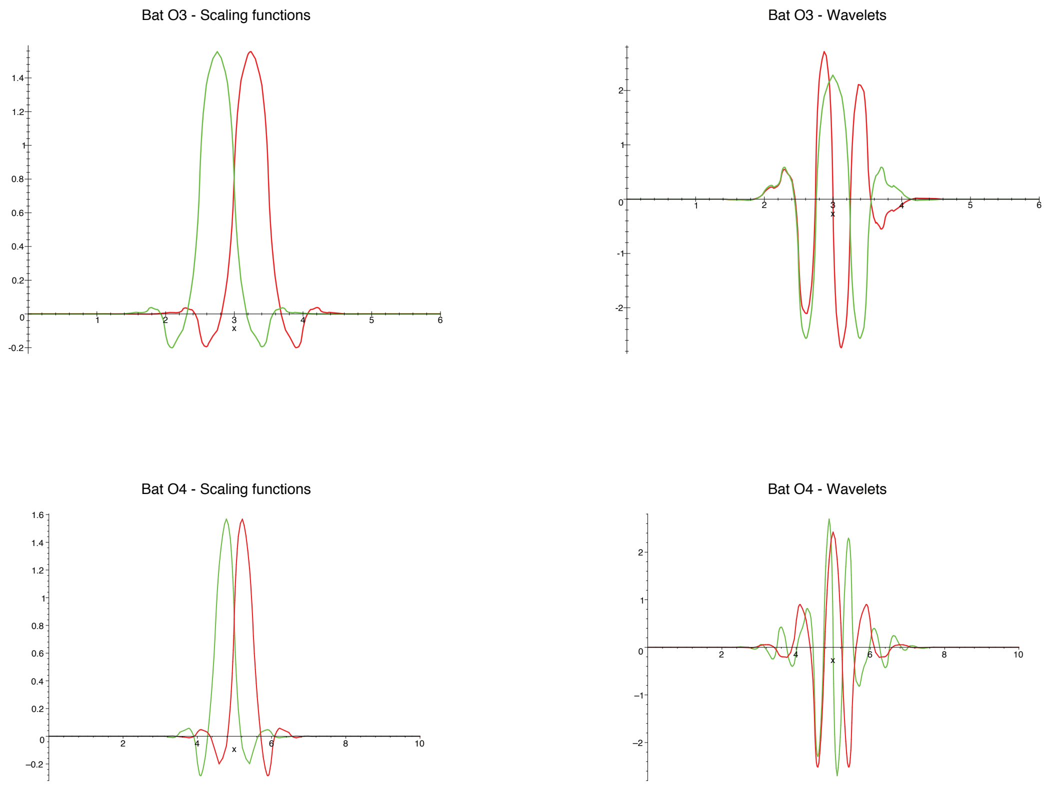


Figure 3.5: Order 3 (resp. 4) balanced orthogonal multiwavelet: the scaling functions are flipped around 3 (resp. 5), the wavelets are symmetric/antisymmetric, the length is 7(resp. 11) taps (2x2) and the estimate of the smoothness by invariant cycles gives the Sobolev exponent s = 1.71 (resp. 2.07).

`Singular` [Greuel et al., 2000] for order $p = 1, 2, 3$ and FGb [Faugère, 1999] for the order $p = 4$. We now have a zero-dimensional drl Gröbner basis $G_{<_{\mathrm{drl}}}$ that we can either transform into a lex Gröbner basis $G_{<_{\mathrm{lex}}}$ using FGLM in the case $p = 1, 2, 3$ or in the case $p = 4$, where FGLM showed its limits, we compute a rational univariate representation of $G_{<_{\mathrm{drl}}}$ by a modified version of the program RealSolving [Rouillier, 1999]. We can then factorize the leading polynomial of the lex Gröbner basis $G_{<_{\mathrm{lex}}}$ in Maple and thus get rid of the multiplicities of the solutions. This means we factorize the Gröbner basis in local algebras that are much easier to solve exactly. In the case $p = 4$, we deal with a RUR and a similar idea is applied to the characteristic polynomial $\chi_u(t)$. We then have the set of solutions for the system.

5. Among this finite number of solutions, we can look for the one leading to the smoothest scaling functions using the estimate by invariant cycles.

Then, we easily derive the highpass filters $n_0(z), n_1(z)$ from the lowpass filters $m_0(z), m_1(z)$ by imposing $n_0(z)$ to be symmetric and $n_1(z)$ to be antisymmetric. The orthonormality conditions give a unique solution up to a change of sign.

Using this approach, we have been able to construct all the orthonormal multiwavelets with compact support $\subset [0, 10]$ and flipped scaling functions, symmetric/antisymmetric wavelets for order 1,2,3 and 4 of balancing. Figures 3.4, and 3.5 show the smoothest high order balanced multiwavelets with these properties. Tables 3.1[1], 3.2 and 3.3 in appendix give the closed form expressions of the coefficients obtained. For order 4 of balancing, because of the degree of the characteristic polynomial in the RUR, a real roots localization program (included in RealSolving) has been used and only numerical solutions (in fact exact intervals containing the solutions) have been obtained.

3.3.2 Smoothness oriented design: the PPZ family

The purpose of this family of balanced multiwavelets is to test our new notion of total balanced smoothness. The design procedure is very similar to the one for the Bat family above. Thus, we will only detail the differences and the computational issues.

Following the ideas of Heller and Wells [1996], we have added the condition of imposing a zero on $\mu_{p-1}(z)$ at either $z = e^{j\frac{5\pi}{6}}$ (preperiodic point of the z^2-invariant cycle $\{e^{j\frac{2\pi}{3}}, e^{j\frac{4\pi}{3}}\}$) or at $z = e^{j\frac{4\pi}{5}}$ (preperiodic point of the z^2-invariant cycle $\{e^{j\frac{2\pi}{5}}, e^{j\frac{4\pi}{5}}, e^{j\frac{-2\pi}{5}}, e^{j\frac{-2\pi}{5}}\}$). This gives equations with polynomials in $\mathbb{Q}[\sqrt{3}][x_1, \ldots, x_n]$ or $\mathbb{Q}[\sqrt{5}][x_1, \ldots, x_n]$. The computation of the drl Gröbner basis is then achieved by introducing the dummy variable x_0 and the polynomial $x_0^2 - 3$ $x_0^2 - 5$ in the system, doing all the computations with this dumb variable and retransforming the solutions with $x_0 = \sqrt{3}$ or $x_0 = \sqrt{5}$.

[1]The coefficients of BAT O1 also appeared in [Chui and Lian, 1996; Strela et al., 1999].

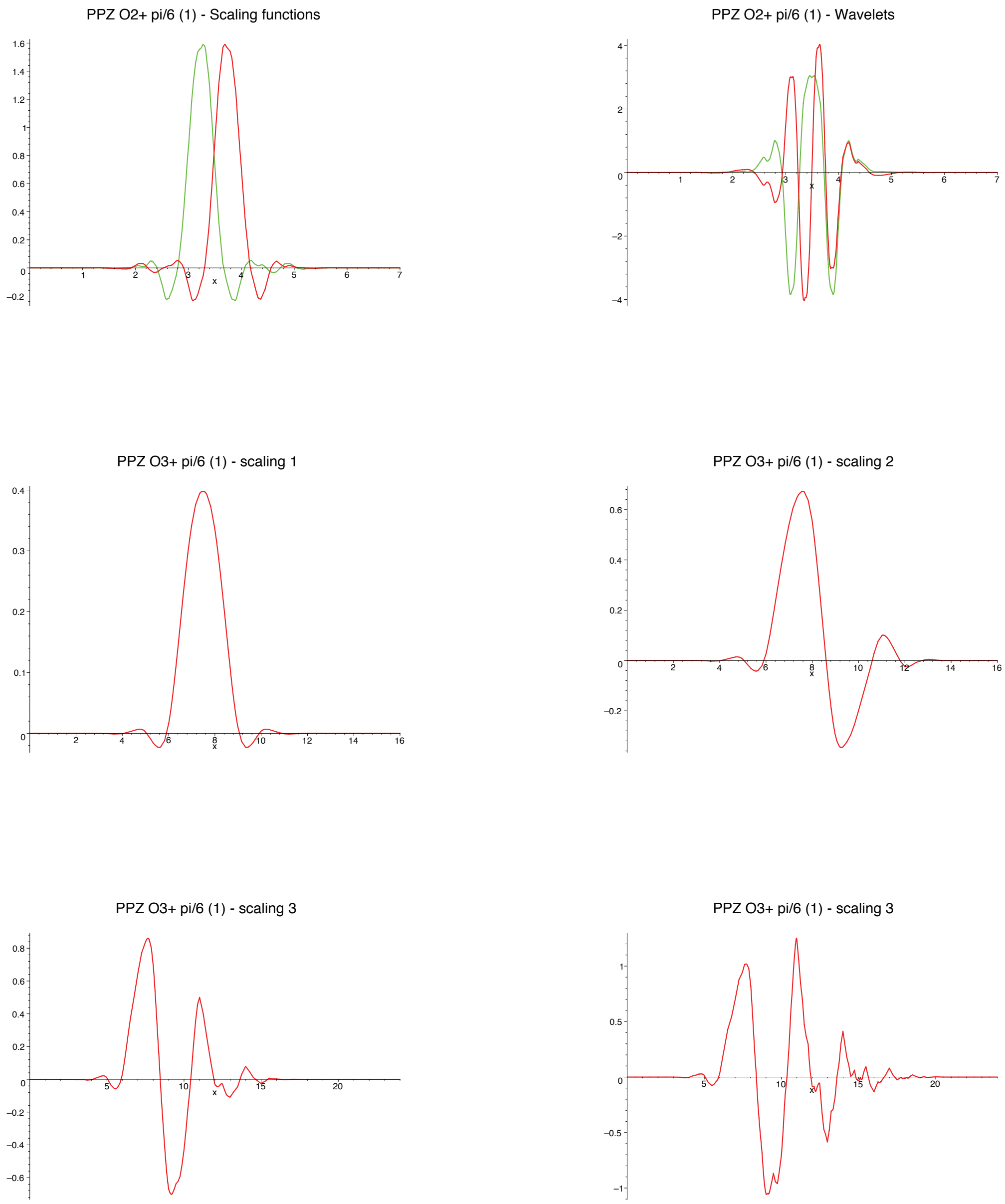


Figure 3.6: Order 3 PPZ (5Pi/6) balanced orthogonal multiwavelet: the scaling functions are flipped around 7/2, the wavelets are symmetric/antisymmetric, the length is 7 taps (2x2) and the total smoothness estimate is [2.32; 0.97, 1.67, 1.20, 0.31].

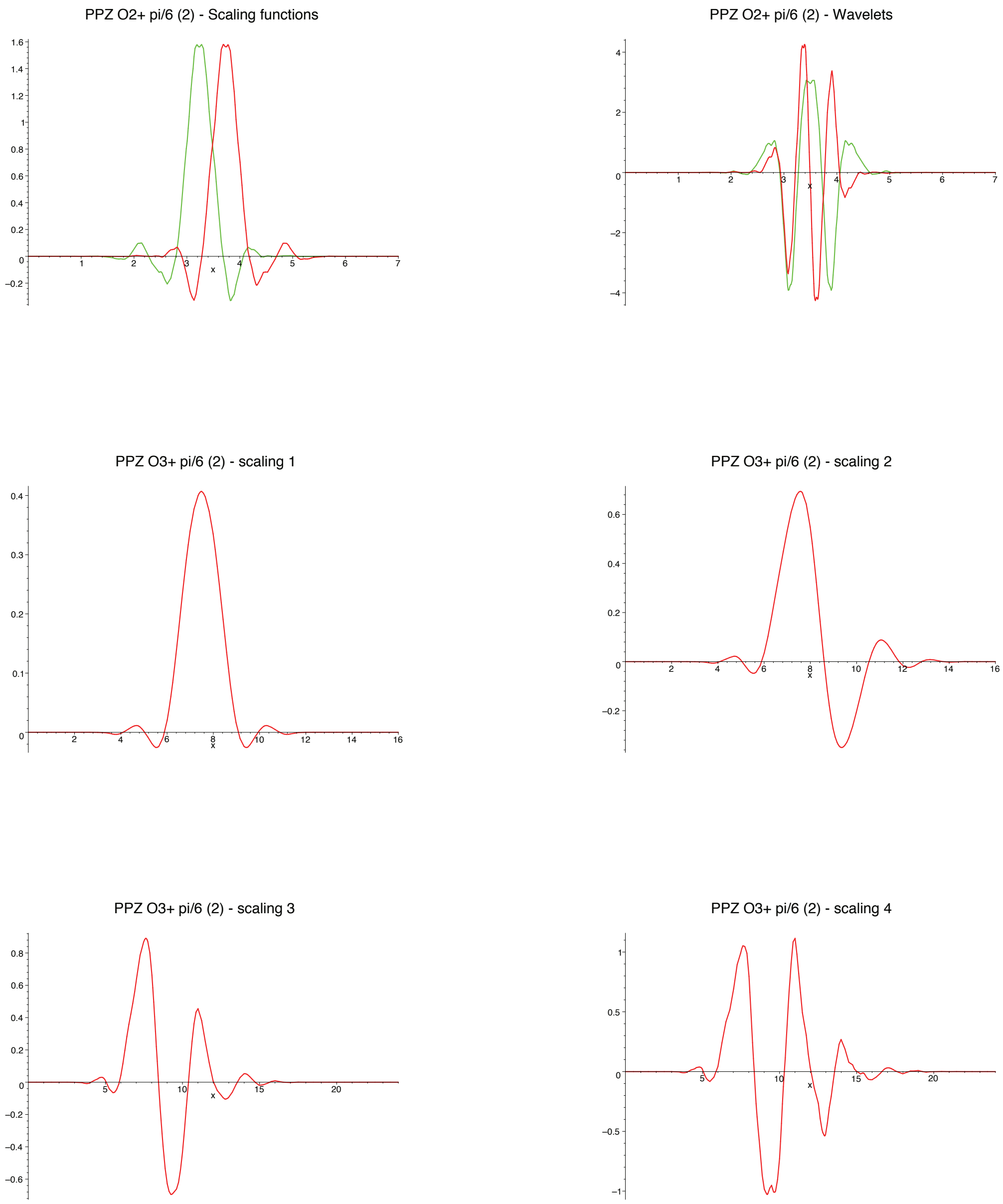


Figure 3.7: Order 3 PPZ (5Pi/6) balanced orthogonal multiwavelet: the scaling functions are flipped around 7/2, the wavelets are symmetric/antisymmetric, the length is 8 taps (2x2) and the total smoothness estimate is [1.99; 0.94, 1.69, 1.99, 1.14].

Using this approach, we constructed all the PPZ ($\frac{5\pi}{6}$) orthonormal multiwavelets with compact support, flipped scaling functions and symmetric/antisymmetric wavelets for balancing order 3. Figures 3.6, 3.7 show the two smoothest ones. As we can notice, the smoothest one in the continous sense is not the smoothest one in the discrete-time sense. In Fig. 3.8, we give also a family of PPZ ($\frac{5\pi}{6}$) orthonormal multiwavelets with compact support, flipped scaling functions and symmetric/antisymmetric wavelets. Suprisingly, the three first members of this family have the same total balanced smoothness.

3.3.3 Interpolation oriented design: M-Coiflets

Again, the design procedure is very similar to the one for the Bat family above. Two new conditions are added:

1. The filter $m_0(z)$ and $m_1(z)$ are supposed to be odd length and symmetric.
2. $\mathbf{M}(z)$ satisfies the multiCoiflet conditions for $n = 0, \ldots, p-1$,

$$\frac{d^n}{d\omega^n}[\boldsymbol{\alpha}_{p-1}^\top(e^{j2\omega})\mathbf{M}(e^{j\omega})]\Big|_{\omega=0} = \delta_n \boldsymbol{\alpha}_{p-1}^\top(1) \tag{3.53}$$

$$\frac{d^n}{d\omega^n}[\boldsymbol{\alpha}_{p-1}^\top(e^{j2\omega})\mathbf{M}(e^{j\omega})]\Big|_{\omega=\pi} = \mathbf{0}^\top. \tag{3.54}$$

Using this approach, we have been able to construct all the orthonormal multiCoiflets with compact support $\subset [-2, 3]$, symmetric scaling functions, symmetric/antisymmetric wavelets for order 1,2 and 3 of balancing. The Figure 3.8 show the smoothest multiCoiflets with these properties.

Finally let's mention that Selesnick [1999] constructed a family of cardinal multiwavelets that appear to be generalized multiCoiflets (the center of mass of the scaling functions is not on an integer, however the filters are interpolating).

3.4 Image coding with balanced multiwavelets

Most lossy transform coders can be split into three distinct stages: transform, quantization and entropy coding. While some of these might be combined, their separation not only helps the implementation, but it also enables a clean analysis of the performance impact of different design choices for each stage. Wavelet based methods, as pioneered in [Antonini et al., 1992], have become a standard and successful way of implementing transform coding.

The underlying filter banks are now well studied, and thus the design procedure is well understood. By the structure of the problem, certain issues are ruled out: e.g. impossibility of constructing orthogonal FIR linear phase filter banks. This is a serious drawback since in many applications, and especially image coding, the following three properties are important: (1) FIR

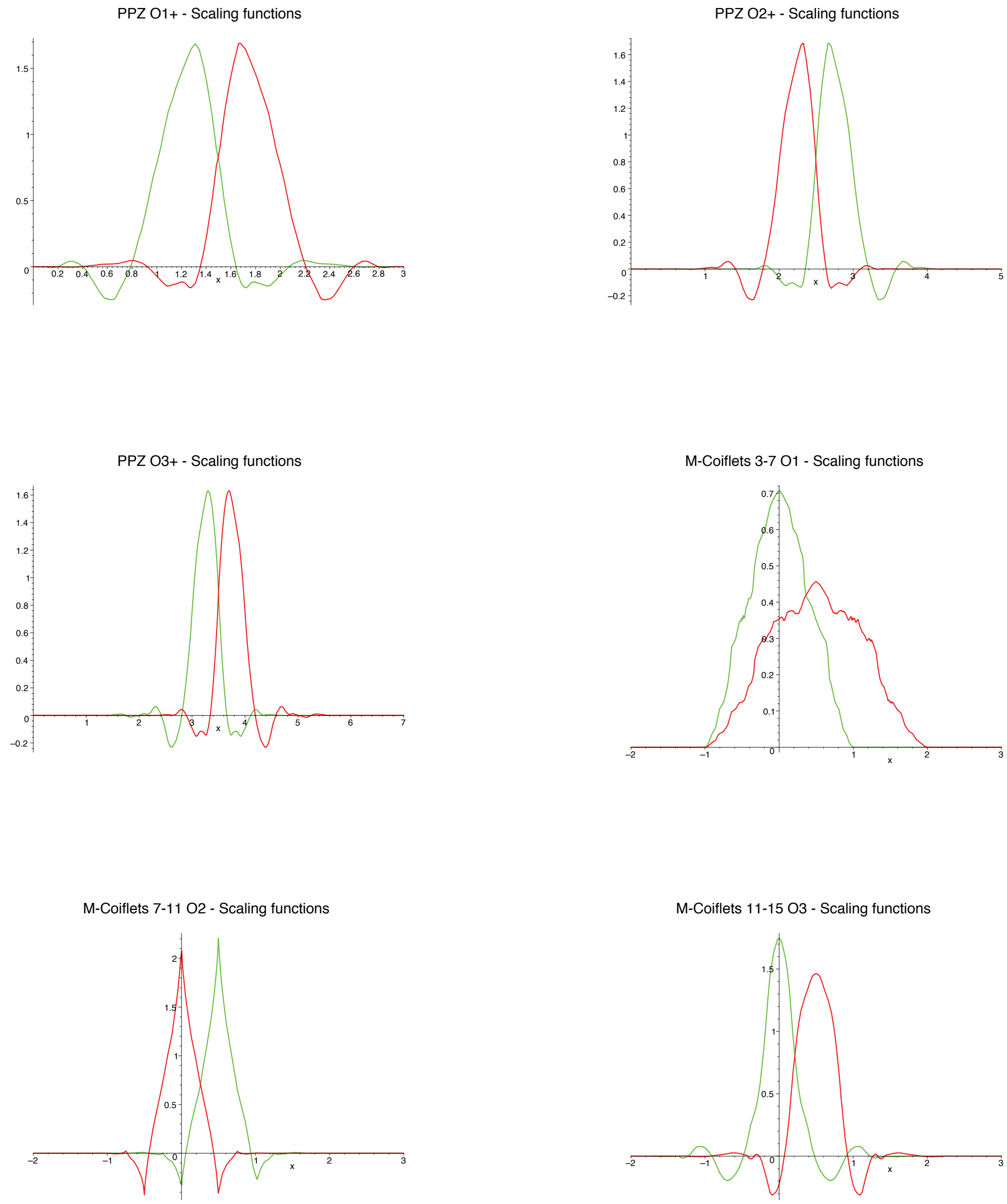


Figure 3.8: PPZ balanced orthogonal multiwavelet: the scaling functions are flipped around 1,2 and 3, the wavelets are symmetric/antisymmetric, the length is 4,6 and 8 taps (2x2) and the total smoothness estimate is [2.33; 0.94, 1.52, 1.36, 0.47]; symmetric orthogonal balanced multiCoiflets of order 1,2and 3. The respective smothness are 0.9, 1.42, 2.86.

for obvious computational reasons, (2) linear phase to work on finite length signals without redundancy and artifacts, and (3) orthogonality as a necessary condition for the decorrelation of subband coefficients. However, by relaxing the time-invariance constraint, it has been proved in the previous sections that new solutions are possible: with multiwavelets, one is finally able to construct orthogonal linear phase FIR transform systems.

In the light of this, we thus decided to modify an existing transform coder (the well-known SPIHT codec [Said and Pearlman, 1996] based on the significance tree quantization (STQ) principle) — based on the classic 9/7 biorthogonal wavelet — by replacing the transform stage with one based on *balanced* multiwavelets that are specially designed for signal compression. Therefore we get a quite fair comparison of the compression performance of the two *wavelets*, since the complexity of the other stages remains the same.

The organization of this section is as follows: first we gives a very short overview of standard wavelet image coders where we briefly describe the SPIHT coder, which we used as our comparison platform. We then details some implementation aspects of a significance tree image coder based on balanced multifilters. Finally, we present and discuss the achieved results. We conclude with an outlook on future research in this domain.

3.4.1 Zero trees and the SPIHT algorithm

The SPIHT algorithm designed by Said and Pearlman [1996] belongs to a class of embedded transform coders which originated with Shapiro's embedded zerotree wavelet scheme (EZW [Shapiro, 1993]). Here, we describe only its main operating principles. First, the image is transformed into its dyadic (pyramid) wavelet decomposition. The coefficients are then assigned to the nodes of a hierarchical tree, such that the coefficients in the parent nodes are *representative* (in terms of energy) for their offspring nodes. The coder output is generated by successively refining the (quantized) coefficients in order of decreasing magnitude (concept of *significance* [Davis and Chawla, 1997]). This can be done very efficiently, since the tree structure exploits the self-similarity across scales of the wavelet transform. Finally, an arithmetic coding stage removes the remaining redundancy.

3.4.2 The 2D multiwavelet transform

Since multiwavelets are defined for vector-valued signals, one might be tempted to *vectorize* an image signal by grouping pairs of rows or columns together. But besides introducing a fundamental asymmetry, this approach also doesn't fit the notion that the LL-subband represents a coarse approximation of the original image.[2] The latter is a key to the performance of SPIHT, and consequently we decided that no vectorization should be used. This is in fact possible by viewing the MW transform as a time-varying filter bank (see Fig. 1.1). The coefficients of the

[2]The main hurdle is that SPIHT cannot handle vector-valued coefficients without undergoing a major revision of the partitioning and quantization mechanisms.

two lowpass (highpass) filters are simply interleaved at the output, e.g. in the one-dimensional case we get the following lowpass signal: $[\ldots, L_0[0], L_1[0], L_0[4], L_1[4], \ldots]$. A separable 2-D transform can now be defined in the usual way as the tensor product of two 1-D transforms. Obviously this approach is symmetric in the coordinates, i.e. every combination of horizontal and vertical filters appears at the output. But now we get 16 subbands, instead of the usual 4 with scalar wavelet transforms. The question is how this fits into the subband decomposition scheme required by SPIHT.

Let us first observe that the lowpass image, corresponding to the classic LL subband, contains the four lowpass-only bands and will be composed of 2x2 blocks as follows:

$$\begin{array}{cc} L_0L_0[x,y] & L_0L_1[x,y] \\ L_1L_0[x,y] & L_1L_1[x,y] \end{array}$$

Since this lowpass image undergoes further decomposition, we have to verify that it is a coarse approximation of the original. And it is indeed: as we can see in Fig. 3.5, the two lowpass filters (scaling functions) are flipped versions of each other (mutual symmetry) with a group delay of 2. Thus our lowpass image closely resembles the output of a classic biorthogonal wavelet transform. Since the compression performance depends critically on this approximation behavior, we can say that our experimental results clearly confirm these arguments. This is no surprise since these arguments were the starting point of the balancing concept (i.e. the preservation/cancellation of discrete-time polynomial signals by the lowpass/highpass branches of the time-varying filter bank).

Another implementation aspect which differentiates multiwavelets from biorthogonal wavelets is the handling of boundary conditions. That is, the extension of a finite support input signal such that the transform coefficients have specific symmetry properties. The goal is to get a non-expansive transform, i.e. input and output dimensionality should be the same. For multirate linear phase FIR filter banks this problem has been treated extensively in [Brislawn, 1995]. But our lowpass filters are only mutually symmetric and thus we had to develop specific methods of signal extension. In the following we limit ourselves to 1-D transforms, as the extension to 2-D is implicit for separable transforms. Further we assume that the signal has length N, a multiple of 4, and that N is larger than the filter size K (to avoid wrap-around conditions).

With an N-periodic signal extension the coefficients obey the trivial symmetry condition $C[i] = C[i \mod N]$. But this simple periodic approach may introduce discontinuities and hence add energy to the high frequency bands. Therefore it should be avoided in coding applications, as confirmed by our compression tests. The method of choice is a *symmetric* extension (with period $2N$), whereby the signal $[x_0, \ldots, x_{N-1}]$ is extended to $[\ldots, x_1, x_0, x_0, x_1, \ldots x_{N-2}, x_{N-1}, x_{N-1}, x_{N-2}, \ldots]$ The boundary points have to be doubled due to the even filter support K. To make sure that only N coefficients are needed to reconstruct $[x_0, \ldots, x_{N-1}]$, we have to exploit the following filter symmetries:

$$\begin{aligned} m_0[i] &= m_1[K-1-i] \\ n_0[i] &= n_0[K-1-i] \\ n_1[i] &= -n_1[K-1-i] \end{aligned}$$

where $i = 0, \ldots, K-1$. These equations yield different symmetries for the coefficients $C[n] = \sum_{i=0}^{K-1} h[i]x[n-i-\nu]$, depending on the *filter shift* ν (h stands for any of the four filters). The natural choice $\nu = K/2$ does not work, since then there are no symmetries to compute the coefficients $C[N]$. We have determined the values $\nu = 0$ for order 1 balancing (with $K = 4$), $\nu = 2$ for order 2 (with $K = 8$), and $\nu = 4$ for order 3 ($K = 12$). Then one gets the following coefficient symmetries:

$$\begin{aligned} L_0[-i] &= L_1[i-4] & \qquad L_0[N+i] &= L_1[N-i-4] \\ L_1[-i] &= L_0[i-4] & \qquad L_1[N+i] &= L_0[N-i-4] \\ H_0[-i] &= H_0[i-4] & \qquad H_0[N+i] &= H_0[N-i-4] \\ H_1[-i] &= -H_1[i-4] & \qquad H_1[N+i] &= -H_1[N-i-4] \end{aligned}$$

for $i = 0, 4, 8, \ldots, 4\lfloor N/8 \rfloor$. Hence N coefficients suffice to represent the input signal.

3.4.3 Results, discussion and further extensions

Fig. 3.9 shows that order 2 balanced multiwavelets achieve fairly good results with PSNR values within 0.5 dB of the original SPIHT with biorthogonal 9/7-tap filters. This is pretty good, since the significance tree and arithmetic compression stages have not been fine-tuned to match the time-varying nature of the MW transform. It also means an average of 5 dB improvement in PSNR for "Lena" compared to previous multiwavelet based image coders (DGHM multiwavelet with pre/post filtering [Strela et al., 1999]). This proves the superiority of the balanced multiwavelet approach over MW systems requiring pre/post filtering of the input data because of unbalanced lowpass filters $m_0(z), m_1(z)$. Namely, with balancing one is able to take full advantage of the interesting properties of multiwavelet systems. The balancing order 3 multiwavelets are slightly better than order 2 at rates below 0.2 bpp, above that, they are slightly worse. We also implemented a coder with the new balanced DGHM multiwavelets first introduced in [Selesnick, 1998]. The bad performance can partly be explained by the fact that there is no easy way to make the transform non-expansive. We had to interpolate a coefficient to get that property. While the formula is exact for real coefficient values, the distortion introduced by the quantization could have been amplified.

On a more subjective level, a visual comparison (Fig. 3.10 and 3.11) reveals a disturbing tiling effect for balancing order 2 multiwavelets. If the order 3 MW is used instead, the more familiar ringing artifacts almost "cover up" these tiling effects. We suspect two main reasons for these

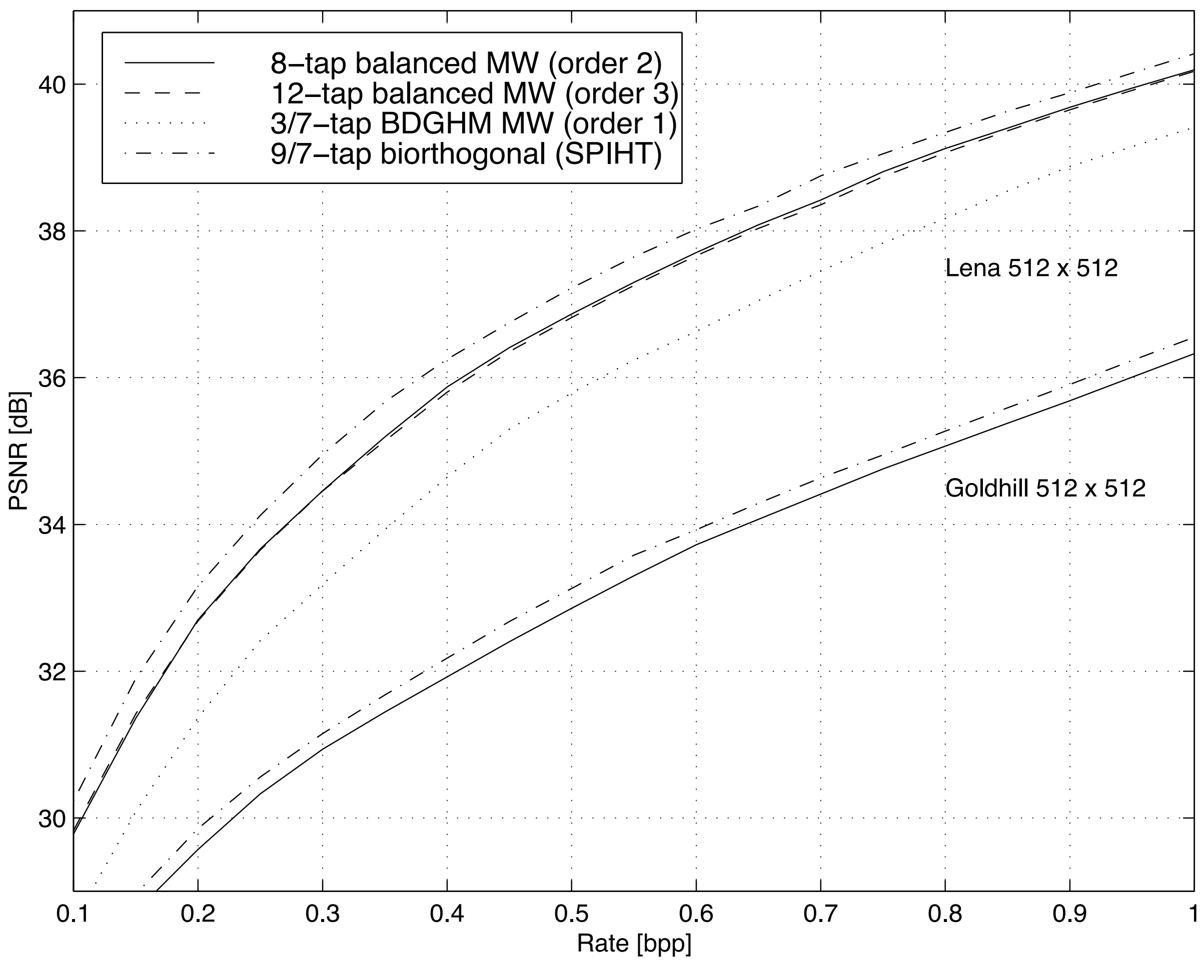


Figure 3.9: Multiwavelet performance comparison

Figure 3.10: Lena at 0.1 bpp, from top to bottom: 9/7-tap biorthogonal (30.23 dB PSNR), 8-tap multiwavelet (balancing order 2, 29.78 dB), 12-tap multiwavelet (balancing order 3, 29.83 dB)

Figure 3.11: Detail of Lena at 0.25 bpp: 8-tap multiwavelet transform (left), original SPIHT with 9/7-tap filters (right)

tiling effects to be: first, the two lowpass filters introduce a 0.5 pixel phase shift at each iteration, due to their structure and the implementation constraints (border symmetrization). Thus high energy coefficients at an image discontinuity will be less well aligned across scales, hampering the efficiency of the tree prediction as illustrated in Fig. 3.12. Secondly, the two highpass filters have very different spectral characteristics, and therefore their outputs should be treated separately in the tree/entropy coding stages. Possibly some smoothing filter could be used to lessen the disturbing tiling effect.

Given the relatively small performance gap, the question is whether balanced multiwavelets could outperform biorthogonal wavelets. Fig. 3.13 shows that the MW transform produces more high amplitude coefficients then the 9/7-tap wavelets (at threshold 8, about 18(20) % of all coefficients are significant). This is a strong indication that the MW transform is not suited for low bitrate compression. Every improvement in successive stages (tree, entropy coding) could as well be applied to biorthogonal wavelets. It is therefore the transform itself, or rather its implementation, which would have to be improved. But this proves difficult, since e.g. the constraints imposed by the downsampling factor and the border symmetry conditions imply that the mentioned lowpass phase shift cannot be avoided. On the other hand, it is possible to construct a highpass filter pair with flip-symmetry as in the lowpass branch. However the coding results are worse than for the symmetric/antisymmetric highpass construction presented here.

Here, we implemented a *balanced* multiwavelet transform for a significance tree quantization image coder (namely SPIHT). With the introduction of the *balancing* concept, it is possible to design general families of high order balanced multiwavelets with the properties required for practical signal processing (preservation/cancellation of discrete-time polynomial signals in the lowpass/highpass subbands, FIR, linear phase and orthogonality). The results obtained so far

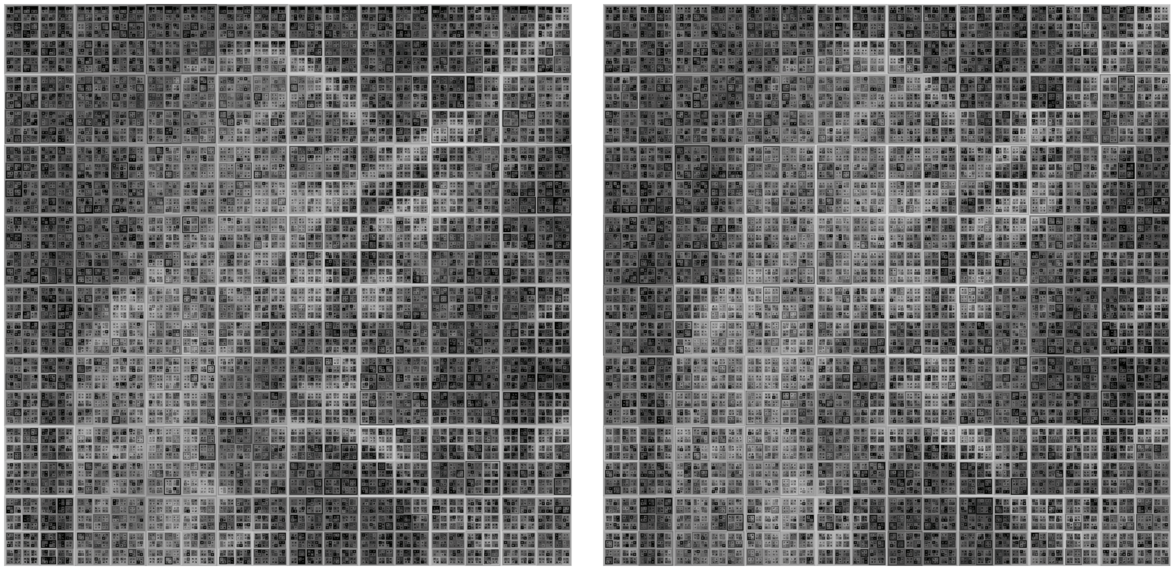


Figure 3.12: Tree decomposition of the horizontal subband magnitudes: 9/7-tap biorthogonal wavelet (left), 8-tap multiwavelet (right). The latter has less well-localized high-energy coefficients.

show substantial improvement over previous MW-based image coders [Strela et al., 1999].

On the other hand, our results are also experimental evidence for the well-known fact that strict orthogonality plays a minor role in image transform coding. Design parameters such as filter length, smoothness and regularity have heavier impact on performance. The design of non-expansive transforms, which are essential for image coding, is harder in the multiwavelet case. In conclusion, the multiwavelets known so far are still no plug-in replacements for the more traditional scalar wavelets.

Nevertheless, we strongly believe that significant improvements could be achieved by some modifications, which includes modifying the significance tree in order to account for the group delay of the two lowpass filters, and a new quantization/thresholding stage, which works on 2x2 coefficient blocks.

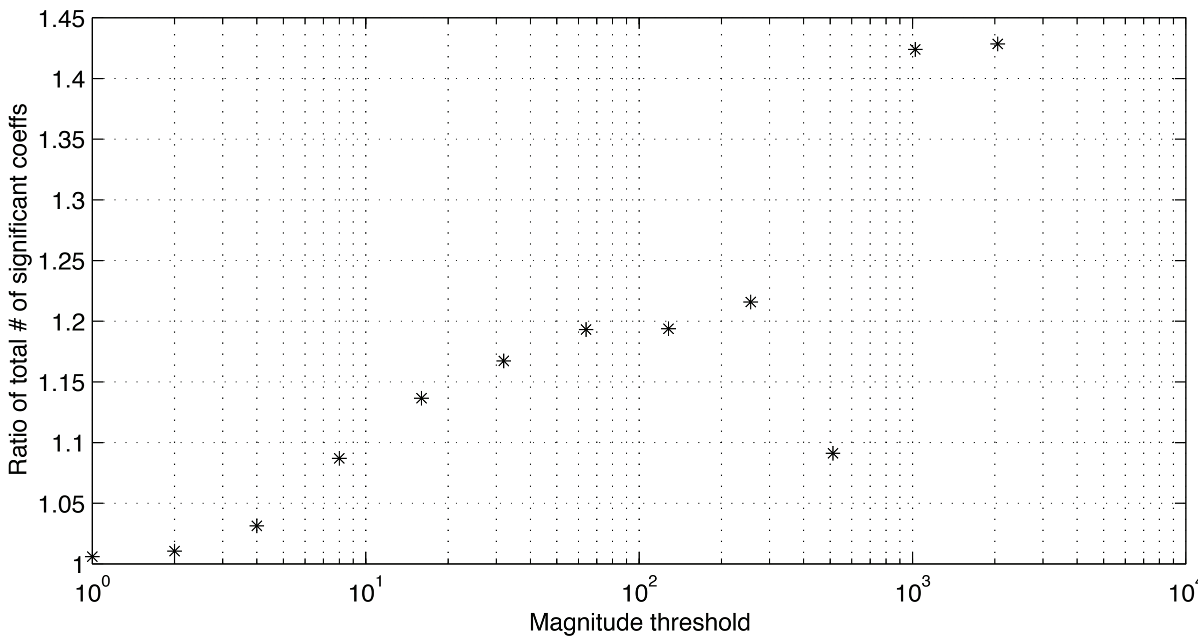


Figure 3.13: Ratio of the numbers of significant coefficients for 8-tap multiwavelets and 9/7-tap biorthogonal wavelets (Lena image). The multiwavelet transform has over 10% more significant coefficients down to threshold 8, corresponding to about 0.7 bpp.

Conclusion

By introducing the concept of *high order balancing*, we have clarified an important issue in the design of multiwavelets. We have proved that this concept is the natural counterpart of the *zeros at* π condition in the standard wavelet theory. With these results, we made it possible to design general families of high order balanced multiwavelets with the required properties for practical signal processing (preservation/cancellation of polynomial signals in the lowpass/highpass subbands and FIR, linear phase, orthogonal filters). The proposed design methods are making extensive use of computationally heavy methods (Gröbner basis decomposition, rational univariate representations) and it is not clear how far we can go with these tools. Matrix spectral factorization could be a way to overcome this limitation. Besides, we investigated the influence of ergodic properties (zeros at pre-periodic points of invariant cycles) on the smoothness of multiwavelets. This brought us to introduce the useful concept of *balanced smoothness* of a multifilter bank. Many of the results there have been obtained using quite ad hoc approaches. A more systematic study would be welcome. *MultiCoiflets* have been introduced as a special case of balanced multiwavelets. Examples of orthogonal, compactly supported, symmetric multi-Coiflets are given. We detailed also an application of high order balanced multifilters to image coding. Again, a lot could be done on that "sisyphic" subject. Finally, by its nature, the Toeplitz approach to multifilter banks could be easily extended to the case of signals living in Hilbert spaces and filters with coefficients being Hilbert-Schmidt operators. This would give a nice framework to do multirate signal processing on second order random signals.

Acknowledgments

We would like to thank I. Daubechies and G. Strang for suggesting useful remarks and ideas. A very special thank goes also to T. Blu and I. Selesnick for the very interesting discussions and fruitful help.

$m_0[k]$	0	$\frac{1}{2}+\frac{1}{4}\sqrt{7}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{2}-\frac{1}{4}\sqrt{7}$	0
$m_1[k]$	0	$\frac{1}{2}-\frac{1}{4}\sqrt{7}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{2}+\frac{1}{4}\sqrt{7}$	0
$n_0[k]$	0	$\frac{-1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{-1}{2}$	0
$n_1[k]$	0	$\frac{1}{4}$	$-\frac{1}{4}\sqrt{7}$	$\frac{1}{4}\sqrt{7}$	$\frac{-1}{4}$	0

Table 3.1: Coefficients of BAT O1: first order balanced orthogonal multiwavelet

$m_0[k]$	0	$-\frac{31}{640}+\frac{1}{640}\sqrt{31}$	$\frac{93}{640}-\frac{13}{640}\sqrt{31}$	$\frac{217}{640}+\frac{23}{640}\sqrt{31}$	$\frac{341}{640}-\frac{11}{640}\sqrt{31}$	$\frac{23}{640}+\frac{7}{640}\sqrt{31}$	$\frac{11}{640}-\frac{11}{640}\sqrt{31}$	$-\frac{1}{640}+\frac{1}{640}\sqrt{31}$	$-\frac{13}{640}+\frac{3}{640}\sqrt{31}$	0
$m_1[k]$	0	$-\frac{13}{640}+\frac{3}{640}\sqrt{31}$	$-\frac{1}{640}+\frac{1}{640}\sqrt{31}$	$\frac{11}{640}-\frac{11}{640}\sqrt{31}$	$\frac{23}{640}+\frac{7}{640}\sqrt{31}$	$\frac{341}{640}-\frac{11}{640}\sqrt{31}$	$\frac{217}{640}+\frac{23}{640}\sqrt{31}$	$\frac{93}{640}-\frac{13}{640}\sqrt{31}$	$-\frac{31}{640}+\frac{1}{640}\sqrt{31}$	0
$n_0[k]$	0	$\frac{23}{160}-\frac{3}{160}\sqrt{31}$	$\frac{11}{160}-\frac{1}{160}\sqrt{31}$	$-\frac{91}{160}+\frac{1}{160}\sqrt{31}$	$\frac{57}{160}+\frac{3}{160}\sqrt{31}$	$\frac{57}{160}+\frac{3}{160}\sqrt{31}$	$-\frac{91}{160}+\frac{1}{160}\sqrt{31}$	$\frac{11}{160}-\frac{1}{160}\sqrt{31}$	$\frac{23}{160}-\frac{3}{160}\sqrt{31}$	0
$n_1[k]$	0	$\frac{47}{320}-\frac{7}{320}\sqrt{31}$	$\frac{9}{320}+\frac{1}{320}\sqrt{31}$	$-\frac{159}{320}+\frac{9}{320}\sqrt{31}$	$\frac{103}{320}+\frac{17}{320}\sqrt{31}$	$-\frac{103}{320}-\frac{17}{320}\sqrt{31}$	$\frac{159}{320}-\frac{9}{320}\sqrt{31}$	$-\frac{9}{320}-\frac{1}{320}\sqrt{31}$	$-\frac{47}{320}+\frac{7}{320}\sqrt{31}$	0

Table 3.2: Coefficients of BAT O2: order 2 balanced orthogonal multiwavelet

$m_0[k]$	0	$-\frac{2989}{2232320}+\frac{97}{2232320}\sqrt{15199}$	$\frac{537}{2232320}+\frac{69}{2232320}\sqrt{15199}$	$-\frac{75969}{2232320}-\frac{105}{446464}\sqrt{15199}$
$m_1[k]$	0	$\frac{2481}{446464}-\frac{21}{446464}\sqrt{15199}$	$-\frac{4701}{446464}+\frac{39}{446464}\sqrt{15199}$	$\frac{13769}{2232320}-\frac{43}{2232320}\sqrt{15199}$
$n_0[k]$	0	$-\frac{2871}{279040}+\frac{33}{279040}\sqrt{15199}$	$-\frac{1177}{279040}+\frac{1}{279040}\sqrt{15199}$	$-\frac{12001}{279040}-\frac{11}{55808}\sqrt{15199}$
$n_1[k]$	0	$\frac{12021}{1116160}-\frac{63}{1116160}\sqrt{15199}$	$\frac{7697}{1116160}-\frac{101}{1116160}\sqrt{15199}$	$-\frac{25921}{1116160}-\frac{331}{1116160}\sqrt{15199}$

$m_0[k]$	$\frac{14785}{446464}+\frac{551}{2232320}\sqrt{15199}$	$\frac{601461}{1116160}-\frac{441}{1116160}\sqrt{15199}$	$\frac{450639}{1116160}+\frac{819}{1116160}\sqrt{15199}$
$m_1[k]$	$\frac{22083}{2232320}-\frac{111}{2232320}\sqrt{15199}$	$-\frac{96093}{1116160}+\frac{45}{223232}\sqrt{15199}$	$\frac{30005}{223232}-\frac{667}{1116160}\sqrt{15199}$
$n_0[k]$	$-\frac{1555}{55808}-\frac{71}{279040}\sqrt{15199}$	$\frac{75083}{139520}+\frac{19}{139520}\sqrt{15199}$	$-\frac{63171}{139520}+\frac{27}{139520}\sqrt{15199}$
$n_1[k]$	$-\frac{44869}{1116160}-\frac{241}{1116160}\sqrt{15199}$	$\frac{150307}{558080}+\frac{743}{558080}\sqrt{15199}$	$-\frac{348777}{558080}+\frac{333}{558080}\sqrt{15199}$

$m_0[k]$	$\frac{30005}{223232}-\frac{667}{1116160}\sqrt{15199}$	$-\frac{96093}{1116160}+\frac{45}{223232}\sqrt{15199}$	$\frac{22083}{2232320}-\frac{111}{2232320}\sqrt{15199}$
$m_1[k]$	$\frac{450639}{1116160}+\frac{819}{1116160}\sqrt{15199}$	$\frac{601461}{1116160}-\frac{441}{1116160}\sqrt{15199}$	$\frac{14785}{446464}+\frac{551}{2232320}\sqrt{15199}$
$n_0[k]$	$-\frac{63171}{139520}+\frac{27}{139520}\sqrt{15199}$	$\frac{75083}{139520}+\frac{19}{139520}\sqrt{15199}$	$-\frac{1555}{55808}-\frac{71}{279040}\sqrt{15199}$
$n_1[k]$	$\frac{348777}{558080}-\frac{333}{558080}\sqrt{15199}$	$-\frac{150307}{558080}-\frac{743}{558080}\sqrt{15199}$	$\frac{44869}{1116160}+\frac{241}{1116160}\sqrt{15199}$

$m_0[k]$	$\frac{13769}{2232320}-\frac{43}{2232320}\sqrt{15199}$	$-\frac{4701}{446464}+\frac{39}{446464}\sqrt{15199}$	$\frac{2481}{446464}-\frac{21}{446464}\sqrt{15199}$	0
$m_1[k]$	$-\frac{75969}{2232320}-\frac{105}{446464}\sqrt{15199}$	$\frac{537}{2232320}+\frac{69}{2232320}\sqrt{15199}$	$-\frac{2989}{2232320}+\frac{97}{2232320}\sqrt{15199}$	0
$n_0[k]$	$-\frac{12001}{279040}-\frac{11}{55808}\sqrt{15199}$	$-\frac{1177}{279040}+\frac{1}{279040}\sqrt{15199}$	$-\frac{2871}{279040}+\frac{33}{279040}\sqrt{15199}$	0
$n_1[k]$	$\frac{25921}{1116160}+\frac{331}{1116160}\sqrt{15199}$	$-\frac{7697}{1116160}+\frac{101}{1116160}\sqrt{15199}$	$-\frac{12021}{1116160}+\frac{63}{1116160}\sqrt{15199}$	0

Table 3.3: Coefficients of BAT O3: order 3 balanced orthogonal multiwavelet

Bibliography

P. Abry and A. Aldroubi. Designing multiresolution analysis-type wavelets and their fast algorithms. *J. Fourier Anal. Appl.*, 2(2):136–159, 1995.

A. Aldroubi. Oblique and hierarchical multiwavelet bases. *App. Comp. Harmonic Anal.*, 4(3): 231–263, 1997.

A. Aldroubi, M. Eden, and M. Unser. Discrete spline filters for multiresolutions and wavelets of ℓ_2. *SIAM J. Math. Anal.*, 25(5):1412–1432, September 1994.

B. Alpert. A class of bases in l^2 for the sparse representation of integral operators. *SIAM J. Math. Anal.*, 24, 1993.

M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. Image Proc.*, 1:205–220, 1992.

K Attakitmongcol, D.P. Hardin, and D.M. Wilkes. Multiwavelet prefilters ii: Optimal orthogonal prefilters. *IEEE Trans. Image Proc.*, 1999. to appear.

G.A. Baker. *Padé Approximants*. Cambridge Univ. Press, 1996.

T. Blu. Iterated filter banks with rational rates changes - connection with discrete wavelet transforms. *IEEE Trans. Signal Proc.*, 41(12):3232–3244, 1993.

Th. Blu and M. Unser. Approximation error for quasi-interpolators and (multi-)wavelet expansions. *App. Comp. Harm. Anal.*, 6:219–251, 1999.

N.K. Bose. *Digital Filters: Theory and Applications*. North-Holland, 1985.

A. Böttcher and B. Silbermann. *Analysis of Toeplitz Operators*. Springer-Verlag, Berlin, 1990.

C.M. Brislawn. Preservation of subband symmetry in multirate signal coding. *IEEE Trans. Signal Proc.*, 43:3046–3050, 1995.

P. J. Burt and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31(4):532–540, April 1983.

A.S. Cavaretta, W. Dahmen, and C.A. Micchelli. *Stationary Subdivision*, volume 93(453) of *Memoirs*. AMS, 1991.

C.K. Chui and J.-A. Lian. A study of orthonormal multi-wavelets. *Applied Numerical Mathematics*, 20(3):273–298, 1996.

A. Cohen. Biorthogonal wavelets. In C. K. Chui, editor, *Wavelets: A Tutorial in Theory and Applications*. Academic Press, New York, 1992.

A. Cohen and I. Daubechies. A new technique to estimate the regularity of refinable functions. *Rev. Mat. Iberoamericana*, 12(2):527–591, 1996.

A. Cohen, I. Daubechies, and G. Plonka. Regularity of refinable function vectors. *J. Fourier Anal. & Appl.*, 3(295–324), 1997.

D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. UTM. Springer-Verlag, 1992.

D. Cox, J. Little, and D. O'Shea. *Using Algebraic Geometry*, volume 185 of *GTM*. Springer-Verlag, 1998.

R. E. Crochiere, S. A. Webber, and J. L. Flanagan. Digital coding of speech in sub-bands. *Bell System Technical Journal*, 55(8):1069–1085, October 1976.

A. Croisier, D. Esteban, and C. Galand. Perfect channel splitting by use of interpolation/-decimation/tree decomposition techniques. In *Int. Conf. on Inform. Sciences and Systems*, pages 443–446, Patras, Greece, August 1976.

I. Daubechies. Orthonormal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, 41:909–996, November 1988.

I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, PA, 1992.

G. Davis and S. Chawla. Image coding using optimized significance tree quantization. In *Proc. IEEE DCC*, 1997.

G. Deslauriers and S. Dubuc. Symmetric iterative interpolation. *Constr. Approx.*, 5:49–68, 1989.

G. Donovan, J. Geronimo, D. Hardin, and P.Massopust. Construction of orthogonal wavelets using fractal interpolation functions. *SIAM J. Math. Anal.*, 1996.

S. Dubuc. Interpolation through an iterative scheme. *J. Math. Anal. Appl.*, 114:185–204, 1986.

S. Durand. Etude de la vitesse de convergence de l'algorithme en cascade dans la construction des ondelettes d'ingrid daubechies. *Rev. Mat. Iber.*, 12(2):277–297, 1996.

T. Eirola. Sobolev characterization of solutions of dilation equations. *SIAM J. Math. Anal.*, 23: 1015–1030, 1992.

G. Evangelista. Orthogonal wavelet transforms and filter banks. In *Proc. 23rd Asilomar Conf., IEEE*, November 1989.

G. Evangelista. Pitch synchronous wavelet representations of speech and music signals. *IEEE Transactions on Signal Processing, special issue on Wavelets and Signal Processing*, 41(12): 3313–3330, 1993.

J.-C. Faugère. A new efficient algorithm for computing gröbner bases (F_4). *J. Pure & Appl. Algebra*, 139:61–88, 1999. `https://calfor.lip6.fr/`

J.-C. Faugère, F. Moreau de Saint-Martin, and F. Rouillier. Design of regular nonseparable bidimensional wavelets using gröbner basis techniques. *IEEE Trans. Signal Proc.*, 46(4): 845–857, 1998.

J.-C. Faugère, P. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional gröbner bases by change of ordering. *J. Symb. Comp.*, 16(4):329–344, 1994.

R. Fröberg. *An Introduction to Gröbner Bases*. John Wiley & Sons, 1997.

C. Gasquet and P. Witomski. *Fourier analysis and applications*, volume 30 of *TAM*. Springer-Verlag, 1999.

J.S. Geronimo, D.P. Hardin, and P.R.Massopust. Fractal functions and wavelet expansions based on several scaling functions. *J. Approx. Theory*, 78, 1994.

L. Gonzalez-Vega, F. Rouillier, and M.-F. Roy. *Some Tapas of Computer Algebra*, chapter Symbolic recipes for polynomial system solving. Springer-Verlag, 1999.

.N.T. Goodman and S.L. Lee. Wavelets of multiplicity r. *Trans. AMS*, 342:307–324, 1994.

G.-M. Greuel, G. Pfister, and H. Schönemann. *Singular Reference Manual*. University of Kaiserslautern, 2000. `http://www.mathematik.uni-kl.de/~zca/Singular`

A. Grossmann and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM Journ. of Math. Anal.*, 15(4):723–736, July 1984.

A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Math. Annal.*, 69:331–371, 1910.

D.P. Hardin and D.W. Roach. Multiwavelets prefilters i: Orthogonal prefilters preserving approximation order $p \leq 2$. *IEEE Trans. on CAS II*, 45, 1997.

C. Heil and D. Colella. Matrix refinement equations: existence and uniqueness. *J. Fourier Anal. App.*, 2(4), 1996.

C. Heil, G. Strang, and V. Strela. Approximations by translates of refinable functions. *Numerische Mathematik*, 73:75–94, 1996.

P.N. Heller. Rank m wavelets with n vanishing moments. *SIAM J. Math. Anal.*, 16(2):502–519, 1995.

P.N. Heller and R.O. Wells. Sobolev regularity for rank m wavelets. preprint, 1996.

C. Herley. Exact interpolation and iterative subdivision schemes. *IEEE Trans. Signal Proc.*, 43 (6):1348–1359, 1995.

C. Herley and M. Vetterli. Wavelets and recursive filter banks. *IEEE Trans. Signal Proc.*, 41 (8):2536–2556, August 1993.

C. Herley and M. Vetterli. Orthogonal time-varying filter banks and wavelet packets. *IEEE Trans. Signal Proc.*, 42(10):2650–2663, 1994.

L. Hervé. Multi-resolution analysis of multiplicity d: Applications to dyadic interpolation. *App. Comp. Harmonic Anal.*, 1:299–315, 1994.

R.Q. Jia and J.J. Lei. Approximation by multi-integer translates of functions having global support. *J. Approx. Theory*, 72:2–23, 1993.

Q. Jiang. On the design of multifilter banks and orthogonal multiwavelet bases. *IEEE Trans. Signal Proc.*, 46(12):3292–3303, 1998.

W. Lawton. Applications of complex valued wavelet transforms to subband decomposition. *IEEE Trans. Signal Proc.*, 41(12):3566–3568, 1993.

W. Lawton, S.L. Lee, and Z. Shen. An algorithm for matrix extension and wavelet construction. *Math. Comp.*, 65:723–737, 1996.

J. Lebrun and M. Vetterli. Balanced multiwavelets. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, April 1997a.

J. Lebrun and M. Vetterli. Balanced multiwavelets - theory and design. Technical report, Dept. of Communication Systems, Swiss Federal Institute of Technology, Lausanne, February 1997b.

J. Lebrun and M. Vetterli. Balanced multiwavelets - theory and design. *IEEE Trans. Signal Proc.*, 46(4):119–125, April 1998a.

J. Lebrun and M. Vetterli. High order balanced multiwavelets. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, May 1998b.

J. Lebrun and M. Vetterli. High order balanced multiwavelets: recent results. In *Wavelets and applications workshop*, October 1998c.

J. Lebrun and M. Vetterli. About balancing order and some other discrete-time properties of multiwavelets. In *Proc. SPIE*, July 1999a.

J. Lebrun and M. Vetterli. High order balanced multiwavelets: Theory, factorization and design. Technical report, Dept. of Communication Systems, Swiss Federal Institute of Technology, Lausanne, February 1999b.

J. Lebrun and M. Vetterli. High order balanced multiwavelets: Theory, factorization and design. *IEEE Trans. Signal Proc.*, 2000a. to appear.

J. Lebrun and M. Vetterli. Theory and design of multicoiflets. In preparation, 2000b.

J.-M. Lina and M. Mayrand. Complex daubechies wavelets. *App. Comp. Harmonic Anal.*, 2: 219–229, 1995.

S. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. and Mach. Int.*, 11(7):674–693, July 1989.

S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998.

P.R. Massopust, D.K. Ruch, and P. J. Van Fleet. On the support properties of scaling vectors. *App. Comp. Harmonic Anal.*, 3:229–238, 1996.

Y. Meyer. Principe d'incertitude, bases hilbertiennes et algèbres d'opérateurs. In *Séminaire Bourbaki*, volume 662. Gauthier-Villars, Paris, 1985-1986.

C.A. Michelli and T. Sauer. On vector subdivision. *Math. Zeit.*, 1997. to appear.

F. Mintzer. Filters for distortion-free two-band multirate filter banks. *IEEE Trans. Acoust. Speech Signal Proc.*, 33(3):626–630, June 1985.

B. Mourrain. An introduction to linear algebra methods for solving polynomial equations. In E.A. Lipitakis, editor, *Proc. HERCMA'9*, pages 179–200, 1999.

E.M. Nikishin and V.N. Sorokin. *Rational Approximations and Orthogonality*, volume 92 of *Translations of Mathematical Monographs*. AMS, 1991.

J.E. Odegard and C.S. Burrus. Discrete finite variation: A new measure of smoothness for the design of wavelet basis. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, volume 3, pages 1467–1470, Munich, 1996.

A.V. Oppenheim, R.W. Schafer, and J.R. Buck. *Discrete-Time Signal Processing, Second Edition*. Prentice-Hall, 1999.

H. Park, T. Kalker, and M. Vetterli. Groebner bases and multidimensional fir multirate systems. *J. Multidimensional Sys. Signal*, 8:11–30, 1996.

P. Pedersen, M.-F. Roy, and A. Szpirglas. *Computational Algebraic Geometry*, pages 203–223. Birkhauser, 1993.

G. Plonka. Approximation order provided by refinable function vectors. *Const. Approx.*, 13: 221–244, 1997.

G. Plonka and R. Amos. A new factorization technique of the matrix mask of univariate refinable functions. *preprint*, 1998.

G. Plonka and V. Strela. Construction of multi-scaling functions with approximation and symmetry. *SIAM J. Math. Anal.*, 29:481–510, 1998.

S.D. Riemenschneider R.-Q. Jia and D.-X. Zhou. Vector subdivision schemes and multiple wavelets. *Math. Comp.*, 67:1533–1563, 1998.

P. Rieder and J.A. Nossek. Multiwavelets transforms based on several scaling functions: Design and implementation. preprint, 1997.

O. Rioul. Simple regularity criteria for subdivision schemes. *SIAM J. Math. Anal.*, 23:1544–1576, November 1992.

O. Rioul. A discrete-time multiresolution theory. *IEEE Trans. Signal Proc.*, 41(8):2591–2606, August 1993a.

O. Rioul. Regular wavelets: A discrete-time approach. *IEEE Trans. Signal Proc.*, 41(12): 3572–3578, December 1993b.

O. Rioul and M. Vetterli. Wavelets and signal processing. *IEEE SP Mag.*, 8(4):14–38, October 1991.

S. Roman. *The Umbral Calculus*. Academic Press, 1984.

F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *J. App. Alg. Eng., Comm. and Comp.*, 9:433–461, 1999.

W. Rudin. *Functional Analysis, second edition*. McGraw-Hill, 1991.

A. Said and W.A. Pearlman. A new fast and efficient image codec on set partitioning in hierarchical trees. *IEEE Trans. Circuits Sys. for Video Tech.*, 6:243–250, 1996.

I. Selesnick. Balanced multiwavelets bases based on symmetric fir filters. *IEEE Trans. Signal Proc.*, 2000. to appear.

I.W. Selesnick. Multiwavelet bases with extra approximation properties. *IEEE Trans. Signal Proc.*, 46:2898–2908, 1998.

I.W. Selesnick. Interpolating multiwavelet bases and the sampling theorem. *IEEE Trans. Signal Proc.*, 47(6):1615–1621, 1999.

J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Proc.*, 41(12):3445–3462, December 1993.

Z. Shen. Refinable function vectors. *SIAM J. Math. Anal.*, 29(1):235–250, 1998.

M. J. Shensa. The discrete wavelet transform: Wedding the à trous and Mallat algorithms. *IEEE Trans. Signal Proc.*, 40(10):2464–2482, October 1992.

M. J. T. Smith and T. P. Barnwell. A procedure for designing exact reconstruction filter banks for tree structured sub-band coders. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, San Diego, CA, March 1984.

G. Strang. Wavelets and dilation equations: a brief introduction. *SIAM Review*, 31(4):614–627, December 1989.

G. Strang. Eigenvalues of $(\downarrow 2)h$ and convergence of the cascade algorithm. *IEEE Trans. Signal Proc.*, 44, 1996.

G. Strang and G. Fix. *An Analysis of the Finite Elements Method*. Wellesley-Cambridge Press, 1973.

G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley Cambridge Press, Boston, 1996.

G. Strang and V. Strela. Short wavelets and matrix dilation equations. *IEEE Trans. Signal Proc.*, 43:108–115, 1995.

V. Strela, P.N. Heller, G. Strang, P. Topiwala, and C. Heil. The application of multiwavelet filter banks to signal and image processing. *IEEE Trans. Image Proc.*, 8(4):548–563, April 1999.

J.-O. Stromberg. A modified Franklin system and higher order spline systems on R^N as unconditional bases for Hardy spaces. In W. Beckner et al, editor, *Proc. of Conf. in honour of A. Zygmund*, pages 475–493. Wadsworth Mathematics series, 1982.

R. Turcajová and J. Kautsky. Block toeplitz-like operators and multiwavelets. In *Proc. SPIE*, volume 2491, pages 957–967, April 1995.

P. P. Vaidyanathan. Quadrature mirror filter banks, M-band extensions and perfect reconstruction techniques. *IEEE ASSP Mag.*, 4(3):4–20, July 1987.

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

M. Vetterli. Multidimensional subband coding: Some theory and algorithms. *Signal Proc.*, 6 (2):97–112, February 1984.

M. Vetterli. Filter banks allowing perfect reconstruction. *Signal Proc.*, 10(3):219–244, April 1986.

M. Vetterli. A theory of multirate filter banks. *IEEE Trans. Acoust. Speech Signal Proc.*, 35(3): 356–372, March 1987.

M. Vetterli and C. Herley. Wavelets and filter banks: Relationships and new results. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, April 1990.

M. Vetterli and C. Herley. Wavelets and filter banks: Theory and design. *IEEE Trans. Signal Proc.*, 40(9):2207–2232, September 1992.

M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

M. Vetterli and G. Strang. Time-varying filter banks and multiwavelets. In *Proc. of the Sixth IEEE DSP Workshop*, Yosemite, 1994.

C. Weidmann, J. Lebrun, and M. Vetterli. Significance tree image coding using balanced multiwavelets. In *Proc. IEEE Int. Conf. Image Proc.*, 1998.

X.-G. Xia, J.S. Geronimo, D.P. Hardin, and B.W. Suter. Design of prefilters for discrete multiwavelet transforms. *IEEE Trans. Signal Proc.*, 44:25–35, 1996.