The Journal of Fourier Analysis and Applications
Volume 2, Number 2, 1995

Designing Multiresolution Analysis-type Wavelets and Their Fast Algorithms

Patrice Abry and Akram Aldroubi

ABSTRACT. Often, the Dyadic Wavelet Transform is performed and implemented with the Daubechies wavelets, the Battle-Lemarié wavelets, or the splines wavelets, whereas in continuous-time wavelet decomposition a much larger variety of mother wavelets is used. Maintaining the dyadic time-frequency sampling and the recursive pyramidal computational structure, we present various methods for constructing wavelets ψ_{wanted}, with some desired shape and properties and which are associated with semi-orthogonal multiresolution analyses. We explain in detail how to design any desired wavelet, starting from any given multiresolution analysis. We also explicitly derive the formulae of the filter bank structure that implements the designed wavelet. We illustrate these wavelet design techniques with examples that we have programmed with Matlab routines.

1. Introduction and Motivation

The *sampled* (or dyadic) *wavelet transform* (SWT) $(\mathbf{W}x)(j,k)$ of a signal $x(t) \in L_2$ is defined to be

$$(\mathbf{W}x)(j,k) = d(j,k) = \langle x(t), \psi_{j,k}(t) \rangle \tag{1}$$

where $\langle ., . \rangle$ is the inner product in L_2 and $\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k)$ are the translates and dilates of a basic pattern $\psi(t)$ (called the mother wavelet) [17]. In some cases, the SWT is associated with a *multiresolution analysis* (MRA) [16, 10]. Specifically, a MRA consists of a sequence of spaces V_j satisfying the following properties [10, 17, 18, 16].

i. $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$, $\bigcup_{j\in\mathbb{Z}} V_j$ is dense in $L^2(\mathbb{R})$.
ii. $V_j \subset V_{j-1}$.
iii. $s(t) \in V_j \longleftrightarrow s(2^j t) \in V_0$.
iv. There exists a function $\phi(t)$ in V_0 such that the collection $\{\phi(t-k), k \in \mathbb{Z}\}$ is an unconditional Riesz basis for V_0.

The associated wavelet spaces W_j are defined to be the orthogonal complements of the spaces V_j relative to the larger spaces V_{j-1}. Thus we have

$$W_j \oplus V_j = V_{j-1}. \tag{2}$$

Math Subject Classification. 42C15
Acknowledgements and Notes. P. Flandrin is gratefully acknowledged for useful discussions and very nice suggestions concerning the examples. The authors also want to thank H. Feichtinger, who gave them a chance to meet by inviting them to his wavelet workshop in Vienna, last spring. We also want to thank Mike Vrehl for his editorial help.

ISSN 1069-5869

It is well known that each space W_j can be generated by an orthonormal basis $\{\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t-k),\ k \in \mathbb{Z}\}$ generated by dilations and translations of a single *orthogonal wavelet* function $\psi(t)$ ($\psi(t)$ is not unique). From property i and (2), it follows that the set $\{\psi_{j,k}(t), (j,k) \in \mathbb{Z}^2\}$ is an orthonormal basis of L_2 and that any function $x(t) \in L_2$ can be recovered from its sampled wavelet transform by the reconstruction formula

$$x(t) = \sum_j \sum_k d(j,k)\psi_{j,k}(t). \tag{3}$$

Moreover, the orthogonal projection $\mathbf{P}_{V_J}x$ of the signal $x(t)$ onto the space V_J is given by

$$\mathbf{P}_{V_J}x = \sum_{j=-\infty}^{J} \sum_k d(j,k)\psi_{j,k}(t),$$

and the residual error $e_J = \mathbf{P}_{V_{J-1}}x - \mathbf{P}_{V_J}x$ that we obtain when we approximate $x(t)$ in V_J instead of the finer resolution space V_{J-1} is given by the following orthogonal projection of $x(t)$ on W_J:

$$e_J = \mathbf{P}_{W_J}x = \sum_k d(J,k)\psi_{J,k}(t).$$

For orthogonal wavelets that are associated with MRAs, the existence of a fast pyramidal algorithm for computing the coefficients $d(j,k)$ in (1) makes the SWT an efficient tool for *data analysis* [16]. Thus, the ability to choose or even design the mother wavelets associated with MRAs is of major interest. For this purpose, we can choose from a set of well-known orthogonal MRA-type wavelets such as the Haar, Daubechies [9], and Battle–Lemarié–Meyer [15] wavelets. However, since this wavelet set has a limited number of patterns, we may not be able to find the wavelet that matches our desired pattern for data processing. Alternatively, we can use one of several approaches that have been developed to approximate a desired wavelet by one that is related to a MRA [8, 27, 13, 24, 20]. However, these methods cannot simultaneously control some important properties of the designed wavelets such as regularity, symmetry, interpolation, and support. Moreover, the wavelets constructed by these techniques suffer from the drawback that the representation of the data in the spaces V_j and W_j can no longer be obtained by orthogonal projections [8, 27, 21, 12]. For instance, the residual error e_J that we obtain when we approximate a signal $x(t)$ in V_J instead of V_{J-1} is not the orthogonal projection of $x(t)$ on W_J as in the case of orthogonal wavelets.

The main purpose of this paper is to describe a general method for constructing a MRA-type wavelet ψ_{wanted} that also has some desired shape and properties (such as symmetry, regularity, and interpolation property). Unlike the other methods, we keep the orthogonality between the wavelet spaces by constructing *semi-orthogonal wavelets*, whose general framework can be found in [3]. Semi-orthogonal wavelets are also called *pre-wavelets* (see for instance [19, 6]), and a general overview of the various types of MRAs and wavelet transforms can be found in [14]. In the semi-orthogonal framework, the orthogonality between wavelet spaces at different scales is kept (i.e., W_j is orthogonal to W_k if $j \neq k$) and the set $\{\psi_{j,k}(t), (j,k) \in \mathbb{Z}^2\}$ is still a basis of L_2. However, what differentiates the semi-orthogonal case is that the wavelet functions at any given fixed scale W_j are not orthogonal to each other (i.e., $\langle \psi(2^{-j}t-k), \psi(2^{-j}t-l)\rangle \neq 0$ for $k \neq l$). By relaxing the orthonormality between a wavelet and its shifts, we are able to design MRA-type wavelets with various properties while preserving (2), which is an important property for data analysis. We then explicitly compute the coefficients of the filters used in the fast pyramidal algorithm that implements the SWT. We have developed Matlab routines to produce examples that illustrate the design of a

chosen wavelet and to perform data analysis/synthesis with such a wavelet. Although we restrict ourselves to the semi-orthogonal framework, most of our results can be transposed to the biorthogonal case.

This paper can be used to design wavelets and their fast algorithms as follows. Starting from a well-known MRA (equivalently, starting from any scaling function ϕ or its associated two-scale sequence $\{u(k)\}_{k\in\mathbb{Z}}$), one can use any of the proposed techniques of §3 to construct a MRA-type wavelet $\psi_{\text{constructed}}$ designed to match a wavelet ψ_{wanted}. One can then compute the filters' coefficients of the pyramidal algorithm that performs the SWT. Moreover, the dual function of $\psi_{\text{constructed}}$ can be approximated as closely as desired. This approximation of the dual is of interest when expanding data according to a given model. The Matlab routines that we have developed are available upon request.

2. Review, Definitions, and Notation

2.1. MRA and SWT

For a semi-orthogonal wavelet $\psi(t)$ that is associated with some given MRA $\{V_j\}_{j\in\mathbb{Z}}$, the reconstruction formula (3) becomes

$$x(t) = \sum_j \sum_k d(j,k)\mathring{\psi}_{j,k}(t) \tag{4}$$

where $\mathring{\psi}_{j,k} = 2^{-j/2}\mathring{\psi}(2^{-j}t - k)$ are the dilations and translations of a dual wavelet $\mathring{\psi}(t)$ whose explicit form can be found in [2, 3]. In fact, the set $\{\mathring{\psi}_{j,k}(t) = 2^{-j/2}\mathring{\psi}(2^{-j}t - k), k \in \mathbb{Z}\}$ is also a Riesz basis for W_j.

From ii, iii, and iv of the definition of a MRA, we immediately obtain the well-known two-scale equation (also called the refinement equation) [16, 9, 6, 19, 3]

$$\phi_1(t) = 2^{-1/2}\phi\left(\frac{t}{2}\right) = \sum_k u(k)\phi(t-k) = (u * \phi)(t) \tag{5}$$

where the sequence $\{u(k)\}_{k\in\mathbb{Z}}$ is called the generating sequence. Moreover, since $\psi(t/2) \in W_1 \subset V_0$, there exits a sequence $\{v(k)\}_{k\in\mathbb{Z}}$ such that

$$\psi_1(t) = 2^{-1/2}\psi\left(\frac{t}{2}\right) = \sum_k v(k)\phi(t-k) = (v * \phi)(t). \tag{6}$$

The two sequences $\{u(k)\}_{k\in\mathbb{Z}}$ and $\{v(k)\}_{k\in\mathbb{Z}}$ play key roles in the computational algorithm for calculating the SWT. The coefficients $d(j,k)$ in 1 can be obtained as the output of a recursive pyramidal structure as shown in Figure 1. The coefficients $h_1(n)$ and $g_1(n)$ of the low-pass and high-pass filters are derived from ϕ and ψ. The coefficients $c(j,k)$, also computed in the pyramidal algorithm (see Figure 1), result from projections onto the spaces V_j: $c(j,k) = \langle x, \phi_{j,k}\rangle$ where $\phi_{j,k}$ are the dilations and translations of the function $\phi(t)$.

As indicated by (4), $x(t)$ can be exactly recovered by weighting the elementary atoms $\mathring{\psi}_{j,k}(t)$ by the values $d(j,k)$ of the SWT in (1). This is depicted in the analysis-synthesis block diagram, sketched in Figure 2, where two new filters h_2 and g_2 are also determined given knowledge of ϕ and ψ. Starting from the two sequences $\{u(k)\}_{k\in\mathbb{Z}}$ and $\{v(k)\}_{k\in\mathbb{Z}}$, §4 will explicitly show how to calculate the four sets of coefficients $\{h_1(n), h_2(n), g_1(n), g_2(n)\}_{n\in\mathbb{Z}}$.

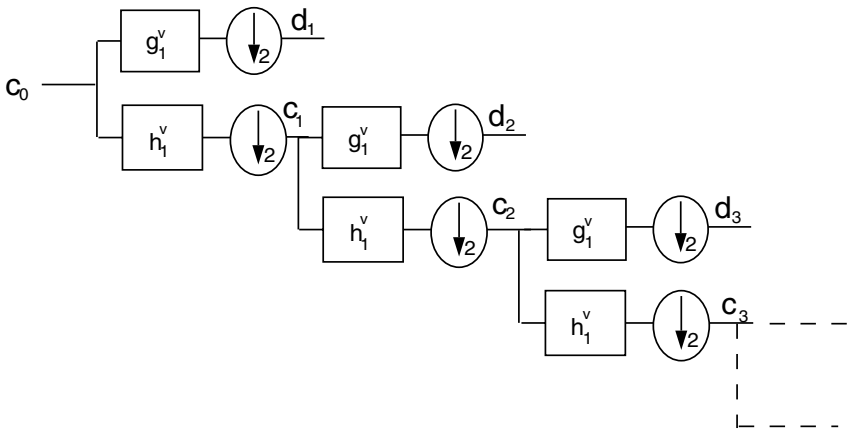

FIGURE 1. The coefficients $d(j,k)$ of a SWT can be computed from a recursive pyramidal filter-bank structure involving two analyzing filters h_1 and g_1, whose coefficients depend on ϕ and ψ. (The $\downarrow_2$ operator stands for a decimation by a factor 2, and $h_1^\vee$ (resp., $g_1^\vee$) means that the convolution is to be performed with a flipped version of the filter h_1 (resp., g_1)).

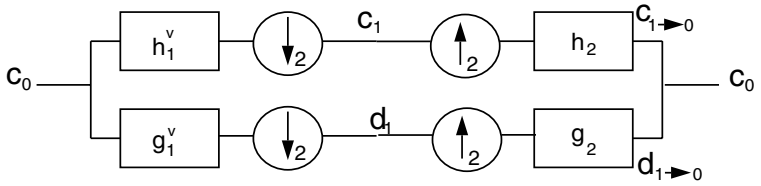

FIGURE 2. Analysis-Synthesis block diagram: the original sequence c_l can be exactly reconstructed from the approximation sequence c_{l+1} and the detail sequence d_{l+1} of the next coarsest resolution by using two synthesis filters h_2 and g_2, which are *dual* filters of the analyzing filters h_1 and g_1. (The $\uparrow_2$ operator stands for an expansion by factor 2 obtained by intertwining zeros between samples).

2.2. Notation

In this section, we list some useful notation (which are coherent with that in [3]). The signals that we consider are real valued and belong to $L^2(\mathbb{R})$. Thus, all the scaling and wavelet functions that we consider are real valued. The symbol $*$ stands for three different types of convolutions. For two signals x and y of $L^2(\mathbb{R})$, $*$ denotes the usual convolution

$$(x * y)(t) = \int x(u)y(t-u)\,du.$$

The symbol $*$ can also refer to the convolution between some signal $x(t)$ and a sequence $\{p_k\}_{k\in\mathbb{Z}}$

$$(p * x)(t) = \sum_k p_k x(t-k).$$

Finally, the discrete convolution between two sequences $\{p_k\}_{k\in\mathbb{Z}}$ and $\{q_k\}_{k\in\mathbb{Z}}$ is denoted by

$$(p * q)(k) = \sum_n p_n q_{k-n}.$$

The reflection of a signal x (resp., a sequence b) is the signal $x^\vee$ (resp., the sequence $b^\vee$) given by

$$x^\vee(t) = x(-t), \qquad t \in \mathbb{R},$$
$$b^\vee(k) = b(-k), \qquad k \in \mathbb{Z}.$$

The modulation $\tilde{b}$ of a sequence b is defined to be

$$\tilde{b}(k) = (-1)^k b(k), \qquad k \in \mathbb{Z}.$$

Whenever it exists, the inverse b^{-1} of a sequence b is defined by

$$b^{-1}(k) * b(k) = \delta_0(k), \qquad k \in \mathbb{Z},$$

$\delta_n(k)$ being the unit impulse response located at $k = n$.

The downsampling (or decimation) operator $\downarrow_2$ assigns to a sequence p, the following sequence $\downarrow_2 [p]$, which consists of the even samples of p only:

$$\downarrow_2 [p] (k) = p(2k) \quad \forall k \in \mathbb{Z}.$$

The upsampling operator $\uparrow_2$ assigns to a sequence p, the following sequence $\uparrow_2 [p]$ in which a zero has been inserted between two successive samples:

$$\left.\begin{array}{lcl} \uparrow_2 [p] (2k) & = & p(k) \\ \uparrow_2 [p] (2k+1) & = & 0 \end{array}\right\} \quad \forall k \in \mathbb{Z}.$$

The symbol $\mathbf{P}_U$ will denote the orthogonal projector onto the subspace U of $L^2(\mathbb{R})$. The orthogonal projection $y(t)$ of $x(t)$ onto the space U is denoted by

$$y(t) = (\mathbf{P}_U x)(t).$$

The Fourier transform of a signal $x(t)$ (resp., a sequence $b(k)$) is denoted by $\hat{x}(f)$ (resp., $\hat{b}(f)$ and is given by

$$\hat{x}(f) = \int_{\mathbb{R}} x(t) e^{-i2\pi f t}\, dt, \tag{7}$$

$$\hat{b}(f) = \sum_{k \in \mathbb{Z}} b(k) e^{-i2\pi f k}. \tag{8}$$

3. Designing the Analyzing Wavelet

The aim of this section is to describe a general method for constructing a chosen analyzing wavelet ψ_{wanted}. The method is based on several theorems that can be found in [3].

3.1. Principle of the Method

There are several tools that can be used to construct scaling and wavelet functions of various shapes and properties, namely:

- Admissible linear combinations (or composite convolution) of scaling and wavelet functions,
- Reflection with respect to the vertical axis,
- Projection onto multiresolution and wavelet subspaces,
- Convolution of scaling functions,
- The use of limit theorems.

3.1.1. Linear Combination. As described in [3], if ϕ is a scaling function generating V_0, then it is possible to construct another scaling function $\phi_{\approx}(t) = \sum p(k)\phi(t-k)$ as long as the real sequence $p(k)$ satisfies the conditions of the following theorem.

1. *Theorem*

The function $\phi_{\approx}(t) = \sum p(k)\phi(t-k)$ is a scaling function if and only if the Fourier transform $\hat{p}(f)$ of $p(k)$ satisfies

$$A \leq \operatorname*{ess\,inf}_{f\in[0,1/2]} |\hat{p}(f)| \leq \operatorname*{ess\,sup}_{f\in[0,1/2]} |\hat{p}(f)| \leq B \tag{9}$$

where A, B are two strictly positive constants.

The proof of Theorem 1 can be found in [3, 4]. A sequence $p(k)$ that satisfies (9) will be called *admissible*. This condition essentially means that $|\hat{p}(f)|$ (see (8)) must be bounded above and below by positive constants A and B. The generating sequence that corresponds to $\phi_{\approx} = p * \phi$ is given by

$$u_{\approx} = \uparrow_2 [p] * u * p^{-1}$$

where the sequence p^{-1} is simply the inverse Fourier transform of $1/\hat{p}(f)$. It should also be noted that any two scaling functions ϕ_1 and ϕ_2 generating the same multiresolution spaces V_j are related by an admissible sequence $p(k)$: $\phi_1 = \sum p(k)\phi_2(t-k)$ [3].

Similarly, if ψ is a wavelet generating W_0, then $\psi_{\approx}(t) = \sum q(k)\psi(t-k)$ is a wavelet for W_0 if and only if the sequence $q(k)$ is admissible. The corresponding new sequence $v_{\approx}$ (see (6)) is given by

$$v_{\approx} = \uparrow_2 [q] * v \tag{10}$$

where v is the sequence associated with the wavelet ψ. Moreover any two wavelets ψ_1, ψ_2 for W_0 must be related by an admissible sequence $q(k)$ (i.e., $\psi_1 = \sum q(k)\psi_2(t-k)$).

Using these facts, it is possible to construct infinitely many scaling and wavelet functions of various shapes by simply choosing various admissible sequences.

3.1.2. Reflection. It is not difficult to see that if $\phi(t)$ satisfies a two-scale equation (5), then its reflection $\phi^{\vee}(t) = \phi(-t)$ also satisfies (5) with $u(k)$ replaced by $u^{\vee}(k)$. Thus, $\phi^{\vee}(t)$ is a scaling function generating a multiresolution $V_j(\phi^{\vee})$. An associated wavelet is given by $\psi^{\vee}(t)$. It should be noted that the multiresolution $V_j(\phi^{\vee})$ generated by $\phi^{\vee}$ is not necessarily the same as the multiresolution $V_j(\phi)$ generated by ϕ.

3.1.3. Projection onto V_0 and W_0. Let $\lambda(t)$ be a function that is not a scaling function in the sense that it does not satisfy a two-scale equation (5). Let $\lambda_a = P_{V_0}\lambda$ be the orthogonal

projection of λ onto the level zero of the MRA $V_j(\phi)$ generated by the scaling function ϕ. Clearly, $\lambda_a(t) = \sum p_\lambda(k)\phi(t-k)$ is in V_0. If the sequence $p_\lambda(k)$ is admissible (see 3.1.1), then the function $\lambda_a(t)$ is also a scaling function generating the MRA $V_j(\phi)$. Moreover, the scaling function λ_a is a good approximation of λ since it is the least squares approximation of λ in V_0. Thus, the orthogonal projection onto V_0 allows us to obtain scaling functions that are close to a desired prescribed shape. Again, the generating sequence associated with the scaling function λ_a is given by

$$u_{\lambda_a} = \uparrow_2 [p_\lambda] * u * p_\lambda^{-1}.$$

This design technique is illustrated in Figure 3. Figures 3(a), (b) show, respectively, the desired scaling function and the starting scaling function (spline of order 1). The scaling function and its related basic wavelet (see (20) are plotted in Figures 3(c), (d).

The same arguments also hold if we use projection onto the wavelet space W_0. This allows us to construct wavelets with some desired shape.

Similar to [20], we can estimate the error

$$e_{j,k} = \langle g, 2^{-j/2}\psi_a(2^{-j}x - k)\rangle - \langle g, 2^{-j/2}\psi(2^{-j}x - k)\rangle$$

obtained when computing the SWT of the function g using $\psi_a = P_{W_0}\psi$ instead of ψ (note that the

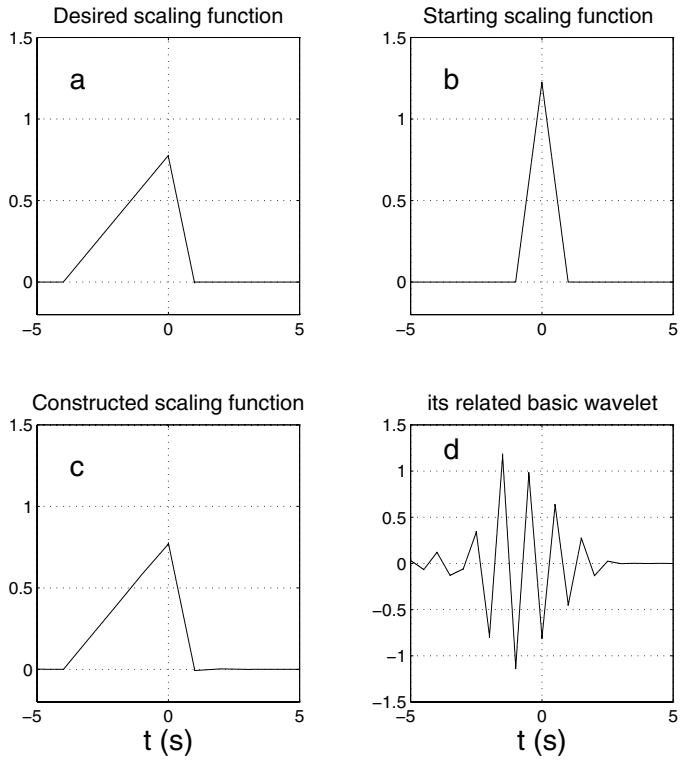

FIGURE 3. Example showing a (n asymmetric) scaling function obtained by the method of projection. (a) The desired scaling function. (b) The B-spline scaling function of order 1. (c) The approximation of the desired scaling function by the projection method in the spline space of order 1. (d) The basic wavelet related to this new scaling function.

factor $2^{-j/2}$ is the usual normalization factor). Without loss of generality, we will assume that the desired wavelet ψ has norm one. As expected, the error does not grow as the scale becomes coarser.

2. ***Proposition***
For $j \geq 0$ we have $||e_{j,k}||_\infty \leq ||g||_{L_2}$.

Proof of Proposition 2. We have that

$$|e_{j,k}| = \left| \int_{\mathbb{R}} g(x)\Delta_{j,k}(x)\,dx \right| \tag{11}$$

where $\Delta_{j,k}(x) = 2^{-j/2}\left(\psi_a(2^{-j}x - k) - \psi(2^{-j}x - k)\right)$. Using the Schwarz inequality and a change of variable we get

$$|e_{j,k}| \leq \|g\|_{L_2} \left\|\Delta_{0,k}\right\|_{L_2}. \tag{12}$$

The proof of the proposition then follows from this last inequality and the facts that $||\psi_a - \psi|| \leq ||\psi||$ and that $||\psi|| = 1$. □

3.1.4. Convolution and Limit Theorems. Often, it is important to have scaling and wavelet functions with a certain degree of regularity (or smoothness) [9]. A simple way of obtaining scaling functions that are as regular as we wish is to start from any appropriate scaling function $\phi(t)$ and use the n-fold convolution to obtain a scaling function

$$\phi^n(t) = \phi * \phi * \cdots * \phi \qquad ((n-1) \text{ convolutions}) \tag{13}$$

that is n times more regular. The appropriate conditions on the starting function ϕ are given in [3].

3. ***Theorem***
Let $u(k)$ be a real sequence such that $u(k) = O(k^{-2})$, and let $\hat{u}(f) = \sum_{k\in\mathbb{Z}} u(k)e^{-i2\pi kf}$ denote its Fourier transform. If $\hat{u}(f)$ satisfies the conditions

$$2^{-1/2}\left|\hat{u}(0)\right| = 1, \tag{14}$$

$$\hat{u}(f) \neq 0 \quad \forall f \in \left[-\frac{1}{4}, \frac{1}{4}\right], \tag{15}$$

and if

$$\hat{\phi}(f) := \prod_{i=1}^{\infty} 2^{-1/2}\hat{u}\left(\frac{f}{2^i}\right) = O(|f|^{-r}), \qquad r > \frac{1}{2}, \tag{16}$$

is such that $\hat{\phi}(f) \in C^2(\mathbb{R})$, $D^{(i)}\hat{\phi}(f) = O(|f|^{-s})$, $i = 0, 1, 2$, with $s > 2$ ($D^{(i)}$ is the derivative of order i), then ϕ is a scaling function generating the multiresolution $V_{(j)}(\phi)$.

These conditions are almost always satisfied. For example, the conditions hold for any Daubechies and for the Battle/Lemarié wavelet of any order. The new generating sequence as-

sociated with $\phi^n(t)$ is given by [3]

$$u^n = 2^{-(n-1)/2}\, u * \cdots * u \qquad ((n-1) \text{ convolutions}).$$

Remark. Since all the functions and discrete sequences we consider are real valued, the modulus of their Fourier transforms are always symmetrical with respect to the vertical axis. Thus, condition (15) can be replaced by $\hat{u}(f) \neq 0\ \forall f \in [0, 1/4]$. For the same reason, the Fourier transform $\hat{\phi}(f)$ of any scaling function we consider must satisfy $|\hat{\phi}(-f)| = |\hat{\phi}(f)|$.

It is not true in general that the n-fold convolution of a wavelet associated with a MRA gives a wavelet that is associated with a MRA. Nevertheless, any wavelet ψ^n associated with ϕ^n (generated by (20), for example) is n times more regular than the wavelet ψ associated with ϕ. Thus, regularity on the wavelet can be achieved by increasing the regularity of the MRA by using the n-fold convolution on the scaling function. The n-fold convolution together with the limit theorems in [3] are used together to obtain scaling and wavelet functions with some specifiable properties as will be further developed in §3.3. For completeness, we restate some of the limit theorems in [3] that are useful for our constructions. From the constructed scaling functions ϕ^n in (13), we construct the orthogonal scaling functions ϕ_o^n [3] (i.e., $\langle \phi_o^n(x), \phi_o^n(x-k)\rangle = 0$ for $k \neq 0$). The sequence of functions $\hat{\phi}_o^n(f)$ tends to a characteristic function as n goes to infinity.

4. *Theorem*

If ϕ is a scaling function satisfying the conditions of Theorem 3*, and if its Fourier transform $\hat{\phi}(f)$ satisfies*

$$\min_{f \in I} \left|\hat{\phi}(f)\right| > \left|\hat{\phi}(f)\right| \quad \forall\ f \notin I = \left[-\frac{1}{2}, \frac{1}{2}\right], \tag{17}$$

then the modulus $|\hat{\phi}_o^n(f)|$ of the Fourier transforms of the orthogonal functions and their duals $\mathring{\phi}_o^n = (\phi_o^n)^\vee$ converge pointwise a.e. and in L_p-norms, $p \in [1, \infty)$, to the ideal low-pass filter as the order n tends to infinity:

$$L_p - \lim_{n\to\infty} \left|\hat{\phi}_o^n(f)\right| = \text{rect}(f). \tag{18}$$

Here, $\text{rect}(f)$ *is the characteristic function on the interval* $[-\frac{1}{2}, \frac{1}{2}]$*, i.e.,* $\text{rect}(f) = 1\ \forall f \in [-\frac{1}{2}, \frac{1}{2}]$ *and* $\text{rect}(f) = 0$ *elsewhere.*

The associated orthogonal wavelets ψ_o^n satisfying $\langle \psi_o^n(x), \psi_o^n(x-k)\rangle = 0$ for $k \neq 0$ tend to an ideal bandpass filter:

5. *Theorem*

The orthogonal wavelet sequence $\left|\hat{\psi}_o^n(f)\right|$ associated with a scaling function satisfying condition (17) *of Theorem* 3 *converges pointwise and in L_p, $1 \leq p < \infty$, to the ideal bandpass filter* $\text{BP}(f) = \text{rect}(2f - 1.5) + \text{rect}(2f + 1.5)$ *as the order n tends to infinity:*

$$L_p - \lim_{n\to\infty} \left|\hat{\psi}_o^n(f)\right| = BP(f). \tag{19}$$

3.2. Designing ψ by Making Changes on ϕ

Any wavelet $\psi(t)$ associated with the multiresolution $V_j(\phi)$ is related to ϕ by (6). If we choose the sequence v to be

$$v = \delta_1 * \tilde{u}^{\vee} * \tilde{a} \tag{20}$$

where a stands for the sampled autocorrelation function of the scaling function

$$a(k) \equiv (\phi * \phi^{\vee})(k),$$

we obtain the basic semi-orthogonal wavelet $\psi_b(t)$ [2, 3], which is the generalization of the B-spline wavelet [25, 26, 7] (the operators "$\sim$" and "$\vee$" are defined in §2.2). This choice allows us to control some of the properties of ψ_b by choosing ϕ appropriately. For instance, the three following properties are easily controlled by this method:

1. The smoothness (regularity) of ψ_b,
2. Axial symmetry,
3. The support of the wavelet ψ_b.

3.2.1. Regularity. In [3], it is shown that it is possible to obtain a new scaling function ϕ_{new} by convolving existing scaling functions: $\phi_{\text{new}} \equiv \phi_1 * \phi_2$, generating a new MRA. Specifically, we have the following result.

6. *Proposition*

If ϕ_1 and ϕ_2 are two scaling functions satisfying the conditions of Theorem 3, then $\phi_{\text{new}} = \phi_1 * \phi_2$ is also a scaling function.

The new generating sequence is then given by $u_{\text{new}} \equiv 1/\sqrt{2} u_1 * u_2$. As we have seen in the previous section, we can start from an arbitrary scaling function ϕ and construct scaling functions ϕ^n that are as regular as we wish. The corresponding basic wavelets ψ_b^n obtained from (6) and (20) have more and more regularity as the degree n increases. This regularity property is also true for all the equivalent wavelets $\psi_{\approx}^n = q_{\approx} * \psi_b^n$ that are obtained using admissible sequences $q_{\approx}$ as described in 3.1.1.

3.2.2. Wavelets with Compact Support. Clearly if u is a finite sequence, then ϕ has compact support [9]. This implies that $a(k)$ is a finite sequence. Thus, from (20) and (6), ψ_b will also have compact support. Hence, we can always obtain wavelets with compact support by starting from a finite generating sequence u (equivalently from a compactly supported scaling function ϕ) and obtain a compactly supported wavelet ψ_b from (20). Clearly all other equivalent compactly supported wavelets (i.e., generating the same space W_0) can be generated by linear combination with admissible sequences $q(k)$ that are finite. Although these equivalent wavelets generate the same space W_0, they will have different shapes that depend on the choice of the sequence q.

3.2.3. Symmetrical Wavelets. It should be noted that if the sequence u in (20) is symmetric, then so is ϕ. Moreover, the sampled autocorrelation function a is always symmetric. From the form of (20), we then immediately conclude that ψ_b has axial symmetry. Thus, it is always possible to construct wavelets that have axial symmetry by starting from a symmetrical generating

sequence (equivalently by starting from a symmetrical scaling function ϕ) and by constructing the wavelet ψ_b. Clearly, other symmetrical wavelets can be constructed by using admissible symmetrical sequences q and constructing the equivalent symmetrical wavelets $\psi_{\approx} = q * \psi_b$. The associated sequence $v_{\approx}$ for $\psi_{\approx}$ is related to the sequence v of ψ_b through $v_{\approx} = \uparrow_2 [q] * v$. Although the two symmetrical wavelets generate the same wavelet spaces W_j, they do not necessarily have the same shape or properties.

Another simple way of obtaining symmetrical scaling functions (and therefore symmetrical wavelets by the method described above) is to start from any asymmetric scaling function and then convolve it with its time-reversed version:

$$\phi_s \equiv \phi * \phi^{\vee}.$$

From our discussion on reflection in §3.1 and Proposition 6, we know that ϕ_s is a scaling function. The corresponding generating sequence u_s is obtained from the generating sequence u of ϕ by simple convolution:

$$u_s = 2^{-1/2}\, u * u^{\vee}.$$

Figure 4 illustrates the method of producing symmetrical wavelets. We chose a *Daubechies* 6 *scaling function*, plotted in Figure 4(a), as a starting function (Figure 4(b) shows the corresponding wavelet). Figures 4(c, d) show the scaling and wavelet functions obtained using the two equations above and from (6) and (20). Moreover, this basic wavelet has compact support as we showed in §3.2.2. In this particular case, one can check that the designed wavelet is exactly equal to the autocorrelation of the original wavelet. Such a choice was made by Saito and Beylkin [23] in which the mother wavelet is chosen to be the autocorrelation of a compactly supported wavelet. However, it is proved [3] that the convolution of two wavelets associated with MRAs does not, in general, result into a wavelet associated with a MRA even though this statement is valid for scaling functions (see 3.2.1). Thus, the production of symmetrical MRA-type wavelets based on the autocorrelation of the scaling functions rather than the autocorrelation of the wavelet itself is the procedure needed to construct symmetrical MRA-type wavelets.

3.3. Modeling the Analyzing Wavelet

3.3.1. Projections onto W_0. Once a scaling function ϕ is chosen, a basic wavelet ψ_b can be selected by (20). As stated in detail in §3.1.1, one can modify ψ_b by linear combination with an admissible sequence q: $\psi_{\approx} \equiv q * \psi_b$. This technique was used in [3] to get orthonormal or interpolating wavelets. But, it can also be used to obtain a wavelet that is closer to some desired function ψ_{wanted}. A first approximation $\psi_{\text{constructed}}$ of ψ_{wanted} can be obtained by projection of ψ_{wanted} onto the wavelet space W_0 as described in §3.1.3:

$$\psi_{\text{constructed}} = P_{W_0} \psi_{\text{wanted}} = q * \psi \tag{21}$$

where

$$q(k) = \left\langle \psi_{\text{wanted}}(.),\, \mathring{\psi}(. - k) \right\rangle.$$

The sequence $q(k)$ is obtained by direct computation of the inner products between ψ_{wanted} and

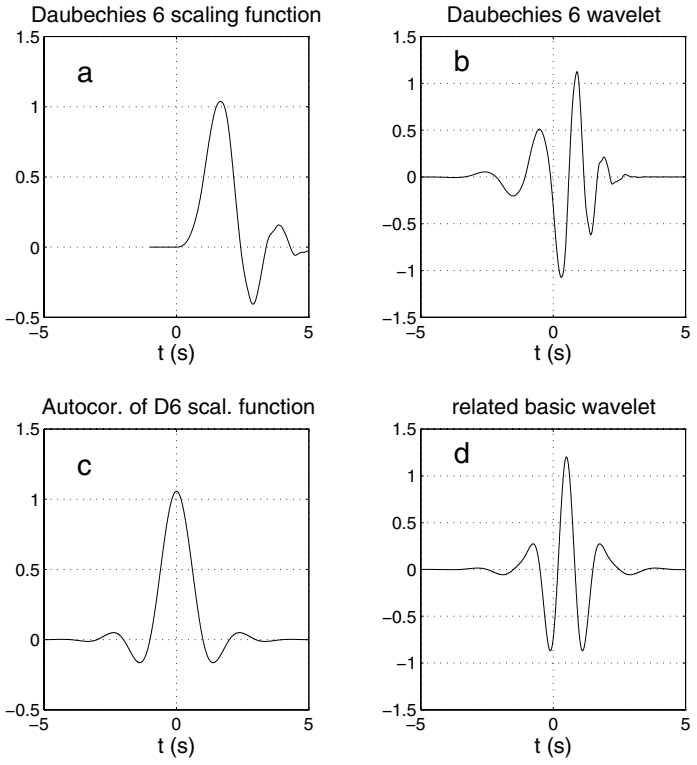

FIGURE 4. A sample construction of a symmetrical scaling and wavelet function. (a) The *Daubechies 6* scaling function. (b) The corresponding *Daubechies 6* wavelet. (c) The scaling function obtained as the autocorrelation of a *Daubechies 6 scaling function*. (d) The corresponding symmetrical basic wavelet.

shifted versions of $\mathring{\psi}$, which is the dual wavelet of ψ. These dual functions in turn can be obtained as proposed in §4.5. The sequence $v_{\text{constructed}}$ related to this new wavelet is given by

$$v_{\text{constructed}} = \uparrow_2 [q] * v$$

and, therefore,

$$v_{\text{constructed}} = \uparrow_2 [q] * \delta_1 * \tilde{u}^{\vee} * \tilde{a}.$$

Instead of the sequences u and v, one has to start with the sequences u and $v_{\text{constructed}}$ in order to derive the filters involved in performing the SWT associated to this new wavelet.

In what follows, we illustrate this wavelet design technique on two examples. By definition of the SWT, the mother wavelet and its shifted versions generate the wavelet space labeled W_0. In fact, it is the data sampling frequency that gives the physical meaning of the space W_0 (see [1]). In particular, the Fourier transform of the mother wavelet has to spread mostly within the normalized frequency band $\left[-1, -\frac{1}{2}\right] \cup \left[\frac{1}{2}, 1\right]$. Therefore, in all the examples proposed below, we will try to obtain wavelets ψ_{wanted} whose central frequency are close to $\frac{3}{4}$ and whose *quality factor* is not too small compared to $(\frac{3}{4})/(\frac{1}{2}) = \frac{3}{2}$. For our first example, we approximate the *Mexican Hat function* (MHF). This function

is often used in conjunction with a continuous-time wavelet and is given by the second derivative of the Gaussian function: $mh(t) = (2\pi\sigma^2)^{-1/2}(1 - (\frac{t}{\sigma})^2)\exp(-(\frac{1}{2})(\frac{t}{\sigma})^2)$. The standard deviation of the Gaussian function has been set to $\sigma = (\frac{3}{4})/(\sqrt{2}\pi)$ to ensure that the maximum of the Fourier transform of the MHF occurs at $f_0 = \frac{3}{4}$. Since the MHF only presents one free parameter, one cannot simultaneously control its quality factor, which in this case is significantly smaller than $\frac{3}{2}$. Indeed, the Fourier transform of the MHF spreads widely apart from the band $\left[-1, -\frac{1}{2}\right] \cup \left[\frac{1}{2}, 1\right]$. Thus, if we seek a wavelet $\psi_{\text{constructed}}$, which has good time-shape approximation of the MHF, we should start with wavelets that are poorly localized in frequency. Figure 5 displays the wavelets $\psi_{\text{constructed}}$ that we obtain by projecting the MHF on spline wavelets of orders 1 and 3.

Our second example consists of wavelets constructed from the famous Altes signal [5]. Such signals consist of hyperbolic frequency modulations (Hyperbolic *chirp*). These signals play key roles in some biological applications as well as in active sonar applications. To approximate such signals, the basic idea is to make use of time-reversed versions (reflections) of *Daubechies wavelet*, with a high degree of regularity [10], from 6 to 10. Indeed, the *Daubechies wavelets* already present this *chirp* behavior [11, p. 84]. For some well-chosen set of parameters characterizing the Altes signal, the chirp signal and the Daubechies wavelet could even have a very similar time-shape (up to a time reversal). Figure 6 shows some approximations of Altes signals (with parameters: duration $\Delta T = 15$s, starting frequency $\nu_1 = 0.18$Hz, finishing frequency: $\nu_2 = 2$Hz) that uses the

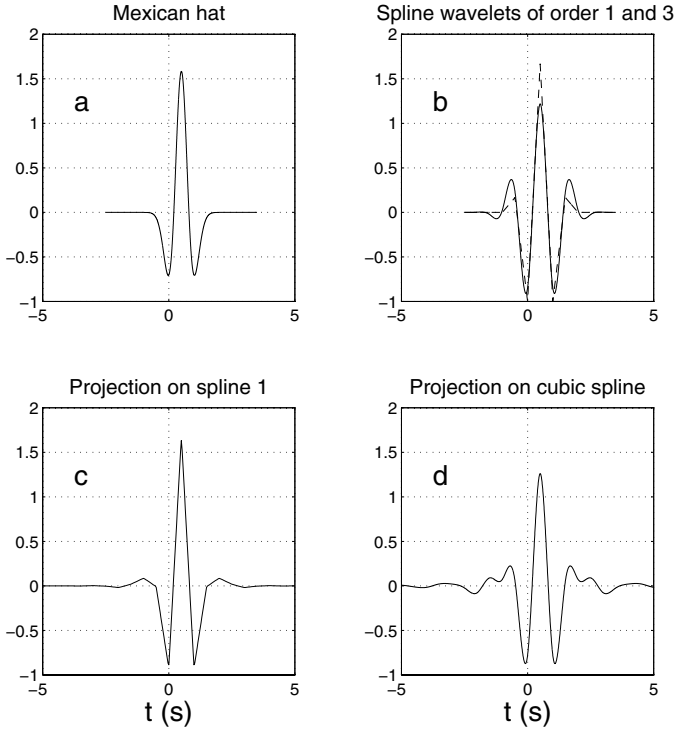

FIGURE 5. Example of a wavelet obtained by the methods of approximation. (a) The desired wavelet ψ_{wanted} is the Mexican Hat function. (b) The B-spline wavelets of order 1 (dashed line) and 3 (solid line). (c) The approximation $\psi_{\text{constructed}}$ of the Mexican Hat function by the projection method in the spline space of order 1. (d) The approximation $\psi_{\text{constructed}}$ of the Mexican Hat function on the cubic spline.

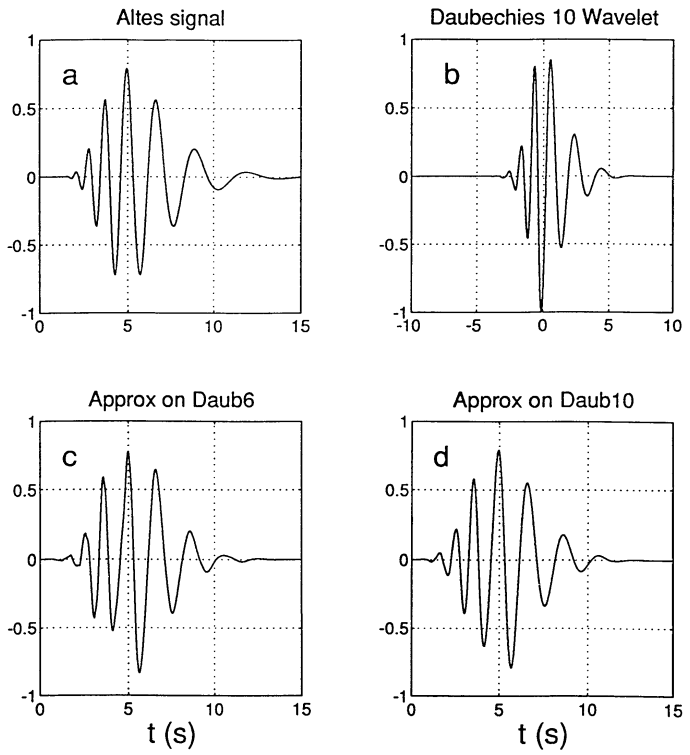

FIGURE 6. Construction of Altes-type wavelets. (a) ψ_{wanted} is a hyperbolic chirp Altes signal (hyperbolic frequency modulation). (b) (Time-reversed) Daubechies 10 wavelets. (c) $\psi_{\text{constructed}}$ of the Altes signal by the projection method using a Daubechies 6 wavelet. (d) $\psi_{\text{constructed}}$ of the Altes signal using Daubechies 10 wavelet.

Daubechies 6 and *Daubechies* 10 wavelets as bases for the construction. The wavelet thus obtained can be seen to be very close to the original pattern.

3.3.2. The Use of Limit Theorems. A more sophisticated approach for constructing the desired wavelets consists of making use of the limit theorems 4 and 5. Indeed, starting from any scaling function and convolving it a large number of times with itself, one can derive wavelets whose Fourier transform converges to the ideal bandpass filter. Then, by projecting ψ_{wanted} onto the space generated by these almost ideal bandpass wavelets, we obtain a wavelet $\psi_{\text{constructed}}$ whose Fourier transform is as close as desired to that of ψ_{wanted} in the limited frequency domain $\left[-1, -\frac{1}{2}\right] \cup \left[\frac{1}{2}, 1\right]$. The projection techniques to be used are the ones discussed in §3.3.1. These operations are summarized as follows.

- Choose a starting MRA; that is, choose the two-scale sequence u (or compute it from the chosen scaling function ϕ; see §4.5).
- Compute the two-scale sequence u_o associated with ϕ_o, the orthonormalized version of ϕ:

$$u_o = \uparrow_2 \left[a^{-1/2}\right] * u * a^{1/2}$$

where a is the sampled autocorrelation sequence of ϕ. Its derivation from the sequence u is discussed in §4.5. This step is not strictly necessary and can be skipped, but performing it often ensures a better computational conditioning.

- Compute the sequence u^n associated with $(\phi_o)^n$ (the nth convolution of ϕ_o) (see [3]):

$$u^n \;=\; 2^{-(n-1)/2}\, u_o \;*\cdots*\; u_o.$$

- Derive the generating sequence v_b^n associated with the corresponding wavelet $(\psi^n)_b$ (as in (20)):

$$v_b^n \;= \delta_1 * \tilde{u}^{n\vee} * \tilde{a}^n$$

where a^n is the autocorrelation sequence of $(\phi_o)^n$. To derive this sequence from u^n, see §4.5.

- Compute the sequence $q(k) = \langle \psi_{\text{wanted}}(.), \mathring{\psi}^n(. - k)\rangle$ defined in §3.3.1. This step requires the knowledge of the time-shape of $\mathring{\psi}^n(t)$, which is detailed in §4.5. Let us note that to produce the time-shape of $\mathring{\psi}^n$ we need to compute the coefficients of the filter g_2 (see (29)). Let us also emphasize that the filters computed at this stage are the set of filters corresponding to a SWT whose mother wavelet is $(\psi^n)_b$. They are not yet the filters corresponding to the SWT associated with ψ_{wanted}. Once the sequence $q(k)$ is computed, the sequence $v_{\text{constructed}}$ associated with $\psi_{\text{constructed}}$, which is the best approximation of ψ_{wanted}, is given by

$$v_{\text{constructed}} \;= \uparrow_2 [q] * v_b^n.$$

- Compute the coefficients of the four filters involved in the filter bank structure from the two generating sequences u^n and $v_{\text{constructed}}$ (see (29)).

To illustrate the above technique, we approximated the MHF. In Figure 7, one sees that the time-shape approximation of the MHF constructed by the above technique is not better than the one obtained by the previous approximations of Figure 5. The approximation shown in Figure 7 has more *oscillations* or *ringing*, which is a Gibbs phenomenon. This is due to the fact that we tried to approximate a function that is widely spread in frequency (the MHF) with one that is an almost ideal bandpass function. Yet, as we have predicted, Figure 7 shows that the Fourier transform of this new function $\psi_{\text{constructed}}$ is much closer to that of the MHF, within the limited-band $\left[-1, -\frac{1}{2}\right] \cup \left[\frac{1}{2}, 1\right]$, than the two previous approximations proposed in §3.3.1. This confirms the fact that increasing the number of convolutions allow us to approximate $\hat{\psi}_{\text{wanted}}(f)$ (the Fourier transform of ψ_{wanted}) as close as we wish in the frequency band $\left[-1, -\frac{1}{2}\right] \cup \left[\frac{1}{2}, 1\right]$.

The next example is displayed in Figure 8 and consists of the celebrated time-continuous *Morlet wavelet*. The frequency of the Gaussian modulated cosine waveform has been set to $\frac{3}{4}$ and the quality factor is close to $\frac{3}{2}$, ensuring a good localization in the frequency domain. With such parameters, the approximation is *so good* (see Figure 8) that one could hardly recognize the original pattern from its approximation if superimposed.

The last example is borrowed from speech processing. To synthesize and analyze spoken language, X. Rodet proposed [22] the use of the *formant-wave-function* (*FWF*), which consists of a single frequency modulated by a sharp exponential attack of short duration. This is then followed by an exponentially decaying tail; see top plot in Figure 9. The approximation of a FWF (of parameters: central frequency $f_0 = \frac{3}{4}$, attack duration: 5s, growth rate: 0.2π) obtained as a projection onto the 6th convolution of the cubic spline is plotted beneath.

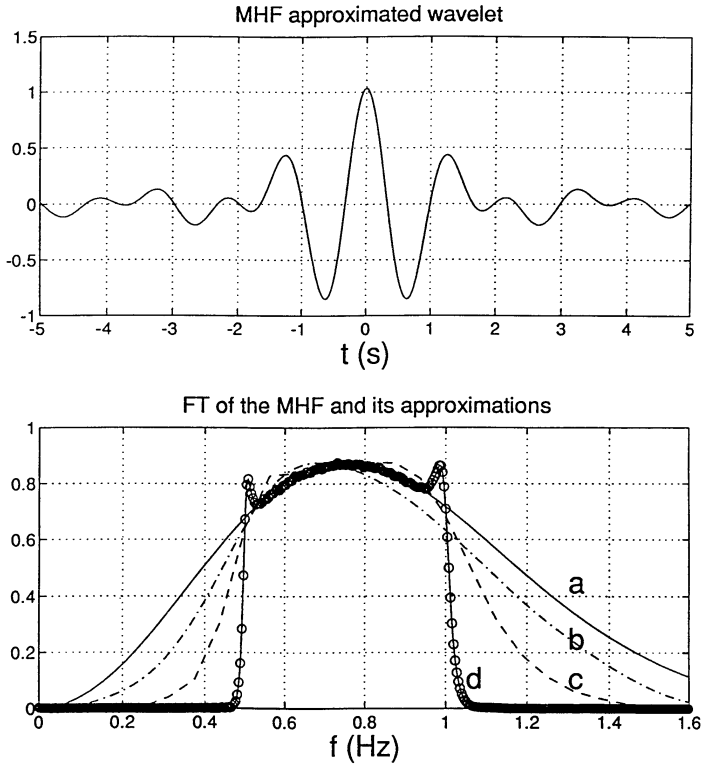

FIGURE 7. Top: Approximation of a Mexican Hat function by projection onto the wavelet space resulting from the 6th convolution of the cubic spline. Bottom: (a) The Fourier Transforms of the Mexican Hat function; (b) FT of the projection on the spline 1 wavelet; (c) FT of the projection on the spline 3 wavelet; (d) FT of the above pattern (projection on the 6th convolution of the cubic spline). One sees that this last approximation is a much better approximation to the Fourier transform of the Mexican hat in the limited band $\left[-1, -\frac{1}{2}\right] \cup \left[\frac{1}{2}, 1\right]$.

3.4. Switching ϕ and ψ with Their Dual Basis, Unit Power Wavelets

Once the sequence of approximation spaces and the associated set of basis functions are chosen, the expansion of a signal $x(t)$ in terms of the wavelets can be performed in two theoretically equivalent ways:

$$x(t) = \sum_j \sum_k \left\langle x(t), \mathring{\psi}_{j,k}(t) \right\rangle \psi_{j,k}(t) \tag{22}$$

or

$$x(t) = \sum_j \sum_k \left\langle x(t), \psi_{j,k}(t) \right\rangle \mathring{\psi}_{j,k}(t). \tag{23}$$

Expansion (22) is relevant in cases where it is important to expand the data in terms of time-shifted and dilated versions of a single particular function (e.g., Doppler signals and Submarine acoustics). In these situations, the ability of approximating the dual function of the basic pattern

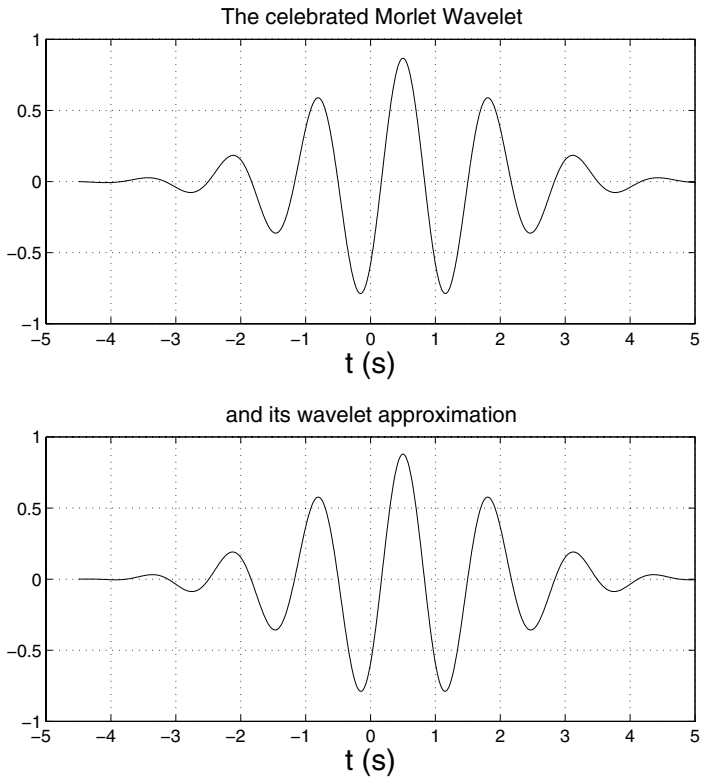

FIGURE 8. Construction of Morlet-type wavelet. Top: ψ_{wanted} is the continuous Morlet wavelet, the standard deviation of the modulated Gaussian function is such that the Fourier Transform is mostly within the $\left[-1, -\frac{1}{2}\right] \cup \left[\frac{1}{2}, 1\right]$ frequency band (the quality factor is close to $\frac{3}{2}$). Bottom: $\psi_{\text{constructed}}$ belongs to the wavelet space associated with the 6th convolution of the cubic spline. Superimposing both plots would show that it is hardly possible to recognize one from another.

is of importance; see §4.5. In contrast, when performing a SWT to analyze data, one is mostly interested in the correlation values (matching) between the signal $x(t)$ and the templates $\psi_{j,k}(t)$. These correlation coefficients are given by $d(j,k) = \langle x(t), \psi_{j,k}(t) \rangle$. Thus, to analyze a signal with a desired wavelet ψ, expansion (23) is needed for signal representation. In §4, the calculations leading to the filters in the filter-bank structure will be made assuming this last choice.

Furthermore, physicists often like to impose the normalizing condition

$$||\phi_{j,k}|| = ||\phi|| = ||\psi_{j,k}|| = ||\psi|| = 1 \ , \forall (j,k) \in \mathbb{Z}^2.$$

Obviously, (5) shows that the sequence u verifies $\sum_k u(k) = \sqrt{2}$. This automatically ensures that $||\phi_{j,k}|| = ||\phi||, \forall (j,k) \in \mathbb{Z}^2$. To obtain unit power scaling functions, we can rescale the sequence a in such a way that $a(0) = 1$. We then find that $(u^\vee * a * u)(k = 0) = a(0) = 1$. This relation replaces the well-known relation $\sum_k u^2(k) = (u^\vee * u)(0) = 1$ for the orthonormal case. Furthermore, one has $||\mathring{\phi}|| = a^{-1}(0)$. In general, $a^{-1}(0) \neq 1$; therefore, it is generally not possible to ensure simultaneously that $||\phi|| = 1$ and $||\mathring{\phi}|| = 1$. In the same way, (5) and (6) ensure that $||\psi_{j,k}|| = ||\psi|| \ \forall (j,k) \in \mathbb{Z}^2$. We want to impose the equality $||\psi|| = ||\phi|| = 1$. To do this, we

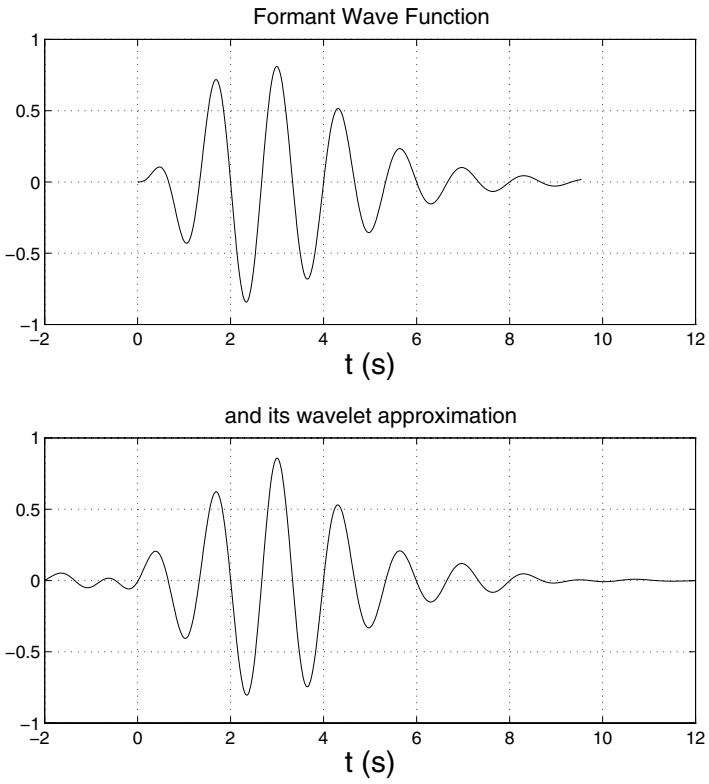

FIGURE 9. Top: ψ_{wanted} is a formant-wave-function. Bottom: $\psi_{\text{constructed}}$ belongs to the wavelet space associated with the 6th convolution of the cubic spline.

prove that this is equivalent to the identity

$$(v^{\vee} * a * v)(k = 0) = 1.$$

This last constraint can easily be satisfied by multiplying v with an appropriate constant (which is a particular case of linear combination, see §3.1.1). We get

$$v_{\text{new}} = ((v^{\vee} * a * v)(0))^{-1/2} v.$$

Again, the above relation will automatically give $||\mathring{\psi}|| = (\downarrow_2 \left[v_{\text{new}}^{\vee} * a * v_{\text{new}}\right])^{-1}(0)$, which implies that, in general, it is not possible to impose simultaneously $||\psi|| = 1$ and $||\mathring{\psi}|| = 1$.

4. Computing the Filter Bank

Once a suitable MRA-type wavelet $\psi_{\text{constructed}}$ has been obtained using the methods of the previous sections, fast analysis/synthesis algorithms are needed. For orthogonal MRA-type wavelets the algorithm is described by the well-known QMF filter bank [16]. Conversely there is a large class of perfect reconstruction filter banks that are fast algorithms for some biorthogonal wavelets that are

associated with MRAs [12, 8, 27, 21]. In contrast, our goal is to find explicitly the filter bank starting from the semi-orthogonal wavelet $\psi_{\text{constructed}}$. Therefore, in this section, we will derive explicit formulae for the fast implementation of the wavelet transform associated with the semi-orthogonal wavelet $\psi_{\text{constructed}}$. In our derivations, we use the sequences $u_{\text{constructed}}$ and $v_{\text{constructed}}$ associated with $\psi_{\text{constructed}}$, which, from now on, we relabel as u and v. As with all the wavelets that are associated with multiresolutions, the fast algorithm for the implementation of the semi-orthogonal wavelet transform consists of the repetitive application of a single procedure. The same is true for the inverse wavelet transform. The analysis and synthesis algorithms form a filter-bank structure as in Figure 2. The filters h_1, h_2, g_1, and g_2 are derived in the next sections. The reader who is only interested in the expressions of h_1, h_2, g_1, and g_2 as functions of u and v but is not interested in the derivation may skip to §4.3.

4.1. Analysis Filters

4.1.1. The Low-Pass Analysis Filter h_1. The key idea of the fast pyramidal algorithm is that the projections of the data $x(t)$ onto V_j and W_j can be obtained directly from its projection in the finer spaces V_{j-1}. Let us assume that the $c(0,k) = \langle x, \phi_{0,k}\rangle$ are known (i.e., $x_0(t) = P_{V_0}x$ is given by $x_0(t) = \sum c(0,k)\mathring{\phi}(t-k)$, where $\mathring{\phi}$ is the dual of ϕ as in [3, 4]). Let us insist on the importance of performing a correct and efficient initialization of the $c(0,k)$ when applying a SWT to analyze real sampled data $\{x_n\}_{n\in\mathbb{Z}}$ (see for instance [1], [26]). Now, from the coefficients $c(0,k)$ at level zero (in V_0), we want to get the coefficients $c(1,k)$ and $d(1,k)$ at the next coarser resolution. By projecting $x_0(t) \in V_0$ onto V_1 and using the facts that the projection operator is selfadjoint and that $P_{V_0}\phi_1 = \phi_1$, we obtain

$$\begin{aligned}(P_{V_1}x)(t) &= \sum_k c(1,k)\mathring{\phi}_1(t-2k)\\ &= \sum_k \langle x_0(.), \phi_1(t-2k)\rangle\, \mathring{\phi}_1(t-2k)\\ &= \sum_k (\downarrow_2 \left[x_0 * \phi_1^\vee\right](k))\mathring{\phi}_1(t-2k).\end{aligned}$$

Using the sampled autocorrelation function $a(k) \equiv (\phi * \phi^\vee)(k)$, we can write the dual scaling function $\mathring{\phi}$ in terms of ϕ as [3]

$$\mathring{\phi} = a^{-1} * \phi. \tag{24}$$

By substituting $\sum c(0,k)\mathring{\phi}(x-k)$ for $x_0(t)$ and by combining (5) with (24), we get

$$\begin{aligned}c(1,k) &= \downarrow_2 \left[x_0 * \phi_1^\vee\right](k)\\ &= \downarrow_2 \left[c(0,.) * (a)^{-1} * \phi * u^\vee * \phi^\vee\right](k)\\ &= \downarrow_2 \left[c(0,.) * u^\vee\right](k).\end{aligned}$$

Hence the expression for the filter h_1 is

$$h_1^\vee = u^\vee. \tag{25}$$

4.1.2. The High-Pass Analysis Filter g_1. The projection of $x(t)$ onto W_1 is given by

$$\begin{aligned}(P_{W_1}x)(t) &= \sum_k d(1,k)\mathring{\psi}_1(t-2k)\\ &= \sum_k \langle x_0, \psi_1(.-2k)\rangle \mathring{\psi}_1(t-2k)\\ &= \sum_k (\downarrow_2 \left[x_0 * \psi_1^\vee\right](k))\mathring{\psi}_1(t-2k).\end{aligned}$$

The very same calculations as that of the previous section yield

$$d(1,k) = \downarrow_2 \left[(c(0,.) * v^\vee)\right](k).$$

It immediately follows that

$$g_1^\vee = v^\vee. \tag{26}$$

4.2. Synthesis Filters

4.2.1. The Low-Pass Synthesis Filter h_2. Since $V_1 \subset V_0$, we can write the function $P_{V_1}x \in V_1$ in terms of the dual basis $\left\{\mathring{\phi}(t-k)\right\}_{k\in\mathbb{Z}}$ of V_0. This yields

$$P_{V_1}x = (\uparrow_2 [c(1,.)]) * \mathring{\phi}_1 = c_{1\to 0}(0,.) * \mathring{\phi}$$

where $c_{1\to 0}(0,.)$ are the coordinates of $P_{V_1}x$ in the basis $\left\{\mathring{\phi}(t-k)\right\}_{k\in\mathbb{Z}}$ of V_0. Furthermore, from (24), we find that

$$\mathring{\phi}_1(t) = 2^{-1/2}\mathring{\phi}\left(\frac{t}{2}\right) = \sum_k (a)^{-1}(k)\phi_1(t-2k) = \left(\uparrow_2 \left[a^{-1}\right] * \phi_1\right)(t).$$

Thus

$$\begin{aligned}(P_{V_1}x) &= \uparrow_2 [c(1,.)] * \mathring{\phi}_1\\ &= \uparrow_2 [c(1,.)] * \uparrow_2 \left[a^{-1}\right] * u * \phi\\ &= c_{1\to 0}(0,.) * \mathring{\phi}.\end{aligned}$$

By inspection of the calculation above we immediately obtain that

$$c_{1\to 0}(0,k) = \left(\uparrow_2 [c(1,.)] * \uparrow_2 \left[a^{-1}\right] * a * u\right)(k).$$

From the above equation we conclude that the filter h_2 is given by

$$h_2 = \uparrow_2 \left[a^{-1}\right] * a * u. \tag{27}$$

4.2.2. The Synthesis High-Pass Filter g_2. Similar to the case of the scaling function, the dual wavelet $\mathring{\psi}$ is given by $\mathring{\psi} = (a_\psi)^{-1} * \psi$ where $a_\psi(k) = (\psi * \psi^\vee)(k)$ is the sampled autocorrelation function of ψ. A direct calculation using (20) shows that

$$\mathring{\psi} = (\downarrow_2 \left[v^\vee * a * v\right])^{-1} * \psi.$$

Therefore, we have that

$$\mathring{\psi}_1 = \uparrow_2 \left[(\downarrow_2 \left[v^\vee * a * v\right])^{-1}\right] * \psi_1.$$

By writing $P_{W_1} x$ in the basis $\left\{\mathring{\phi}(t-k)\right\}_{k\in\mathbb{Z}}$ of V_0 we obtain

$$\begin{aligned}(P_{W_1}x) &= \uparrow_2 [d(1,.)] * \mathring{\psi}_1 \\ &= \uparrow_2 [d(1,.)] * (\uparrow_2 \left[(\downarrow_2 \left[v^\vee * a * v\right])^{-1}\right]) * v * \phi \\ &= d_{1\to 0}(0,.) * \mathring{\phi}.\end{aligned}$$

Hence,

$$d_{1\to 0}(0,k) = \uparrow_2 [d(1,.)] * \uparrow_2 \left[(\downarrow_2 \left[v^\vee * a * v\right])^{-1}\right] * v * a,$$

from which we get that the filter g_2 is given by

$$g_2 = a * \left(\uparrow_2 \left[(\downarrow_2 \left[v^\vee * a * v\right])^{-1}\right]\right) * v. \tag{28}$$

4.3. Filter-Bank Pyramidal Structure

The relation between the filters and the two sequences u and v that we calculated in the previous sections are summarized in the following set of equations (29). From the two generating sequences u and v associated with some chosen or constructed scaling function and wavelet (as in §3), one can implement a semi-orthogonal SWT with the filter-bank pyramidal structure as in Figure 1. The four filters involved are

$$\begin{aligned}h_1^\vee &= u^\vee, \\ g_1^\vee &= v^\vee, \\ h_2 &= a * \uparrow_2 \left[a^{-1}\right] * u, \\ g_2 &= a * \uparrow_2 \left[(\downarrow_2 \left[v^\vee * a * v\right])^{-1}\right] * v.\end{aligned} \tag{29}$$

The derivation of the autocorrelation sequence a from the generating sequence u will be discussed in §4.5.

If (22) is chosen instead of expansion (23), the very same kind of calculations yields the following set of filters:

$$\begin{aligned}h_1^\vee &= a * \uparrow_2 \left[a^{-1}\right] * u^\vee, \\ g_1^\vee &= a * \uparrow_2 \left[(\downarrow_2 \left[v^\vee * a * v\right])^{-1}\right]) * v^\vee, \\ h_2 &= u, \\ g_2 &= v.\end{aligned} \tag{30}$$

4.4. The Orthonormal Case

As a test, we apply our results to the well-known orthonormal SWT starting from Quadrature Mirror Filters sequences u and v. The QMF sequences must satisfy (see [9, 3] and (20)).

$$\begin{aligned} \sum_k u(k) &= \sqrt{2}, \\ \sum_k u(k)u(k+2p) &= \delta_0(p), \\ v &= \delta_1 * \tilde{u}^\vee. \end{aligned} \tag{31}$$

Since QMF filters always generate orthogonal scaling and wavelet functions, we have that $a \equiv \delta_0$. This fact, together with the fact that

$$\downarrow_2 \left[v * v^\vee\right](n) = \downarrow_2 \left[\tilde{u}^\vee * \tilde{u}\right](n) = (-1)^{2n} \sum_k u(k)u(k+2n) = \delta_0(n),$$

allow us to obtain the well-known Mallat algorithm for the orthonormal wavelet transform which is given by

$$\begin{aligned} h_1^\vee &= u^\vee, \\ g_1^\vee &= v^\vee, \\ h_2 &= u, \\ g_2 &= v. \end{aligned} \tag{32}$$

4.5. Computation of a, ϕ, ψ Starting From a Generating Sequence u

The aim of this section is to show how to obtain the scaling function ϕ from the generating sequence u and vice versa. Let us first assume that the successive nested approximation spaces V_j (equivalently, the scaling function ϕ) are chosen; one then needs the sequence $\{u(k)\}_{k\in\mathbb{Z}}$. The sequence $u(k)$ can be obtained by the inner products $u(k) = \langle \phi_1(.), \mathring{\phi}(.-k)\rangle$, but one can avoid the search for the dual basis by computing: $u'_k = \langle \phi_1(.), \phi(.-k)\rangle$. Then one can easily show that

$$u(k) = (a^{-1} * u')(k) \tag{33}$$

where $a(k)$ results from a direct computation that uses the shape of ϕ: $\langle \phi(.), \phi^\vee(.-k)\rangle$.

Instead of starting from a scaling function ϕ, we can start from any appropriate generating sequence u as in Theorem 3 (see also [3]). From u, we can find the scaling function ϕ and its shape and then compute its sampled autocorrelation function. Specifically, to generate an initial scaling function $\phi(t)$ we can use any sequence $u(k)$ satisfying the conditions of Theorem 3 (u need not satisfy a QMF condition, see (31)). A function generated by such a sequence satisfying Theorem 3 always generates a MRA. However, $\{\phi(t-k)\}_{k\in\mathbb{Z}}$ are not necessarily orthogonal (clearly, it is possible to orthogonalize these basis functions [3]). The Fourier transform $\hat{\phi}$ of the scaling function ϕ is given by the infinite product

$$\hat{\phi}(f) = \prod_{i=1}^{\infty} 2^{-1/2}\hat{u}(2^{-i}f). \tag{34}$$

An approximation to $\hat{\phi}$ can be obtained by taking a finite product and then multiplying by a window function:

$$\hat{\phi}_{\text{approx}}(f) = \chi_{2^N}(f) \prod_{i=1}^{N} 2^{-1/2}\hat{u}(2^{-i}f) \tag{35}$$

where χ_{2^N} is the characteristic function on $[2^{-(N-1)}, 2^{(N-1)}]$ (i.e., $\chi_{2^N}(f) = 1$ in the interval $[2^{-(N-1)}, 2^{(N-1)}]$ and zero elsewhere). Using Shannon's sampling theorem, the inverse Fourier transform of $\hat{\phi}_{\text{approx}}$ is given by

$$\phi_{\text{approx}}(t) = \sum\nolimits_k \phi_{\text{approx}}(2^{-N}k)\text{sinc}(2^N t - k). \tag{36}$$

From this relation and (35), we conclude that the samples $\phi_{\text{approx}}(k/2^N)$ are simply obtained by the convolution equation

$$\phi_{\text{approx}}(k/2^N) = 2^{-N/2}u* \uparrow_2 [u] * \cdot* \uparrow_{2^{N-1}} [u]. \tag{37}$$

Clearly, as N becomes larger and larger, the samples thus obtained become those of the scaling function ϕ, which are taken at a finer and finer mesh. From a practical point of view, this approximation simply results from applying n-time the pyramidal synthesis algorithm, starting with a Dirac impulse sequence for the coarsest approximation and setting all the *detail sequences* to zero. For this case, the set of (30) ($h_2 = u$) is used. The computation of the sampled autocorrelation function a of ϕ from the above approximation is used to estimate numerically the inner products defining a. Clearly, this numerical approximation will tend to the exact value as N tends to infinity. Numerical experimentations showed that this method to compute a converges quickly.

The same argument holds for the calculation of the inner products defining the sequence $q(k) = \langle \psi_{\text{wanted}}(.), \mathring{\psi}(. - k)\rangle$. The samples of $\mathring{\psi}$ can be obtained from the discrete convolution

$$\mathring{\psi}_{\text{approx}}(2^{-N}k) = 2^{-N/2}h_2* \uparrow_2 [h_2] * \cdots * \uparrow_{2^{N-2}} [h_2]* \uparrow_{2^{N-1}} [g_2].$$

Practically, this can also be computed by iterating the pyramidal synthesis algorithm starting with a dirac impulse for the coarsest detail sequence and sequences of zeros for the coarsest approximation and all the other details. Figure 10 shows the dual functions of the wavelets resulting from the approximation of the Altes signal in Figure 6 and of the FWF in Figure 9.

In a similar way, approximations of ψ and $\mathring{\phi}$ are given by

$$\psi_{\text{approx}}(k/2^N) = 2^{-N/2}u* \uparrow_2 [u] * \cdots * \uparrow_{2^{N-2}} [u]* \uparrow_{2^{N-1}} [v]$$

and

$$\mathring{\phi}_{\text{approx}}(k/2^N) = 2^{-N/2}h_2* \uparrow_2 [h_2] * \cdots * \uparrow_{2^{N-1}} [h_2].$$

We have used the above techniques to plot the time-shapes of the wavelets/scaling functions in the different figures illustrating this paper.

5. Conclusion

The semi-orthogonal approach studied in this paper offers an intermediate solution between the orthogonal and the biorthogonal methods. Relaxing the constraint of an orthonormal basis enables us to use a large variety of mother wavelets, whereas preserving orthogonal projections ensures no correlation between elementary cells or atoms located at different scales.

The wavelet/scaling function design methods that we have described allow us to create an almost infinite number of wavelets and also to compute the corresponding analysis-synthesis filters for the fast filter-bank algorithm. The wavelets/scaling functions can be constructed to have various shapes and properties while keeping the least squares approximation property and the fast algorithms ($O(N)$ for signals of length N). As far as choosing the wavelet shape and properties are concerned,

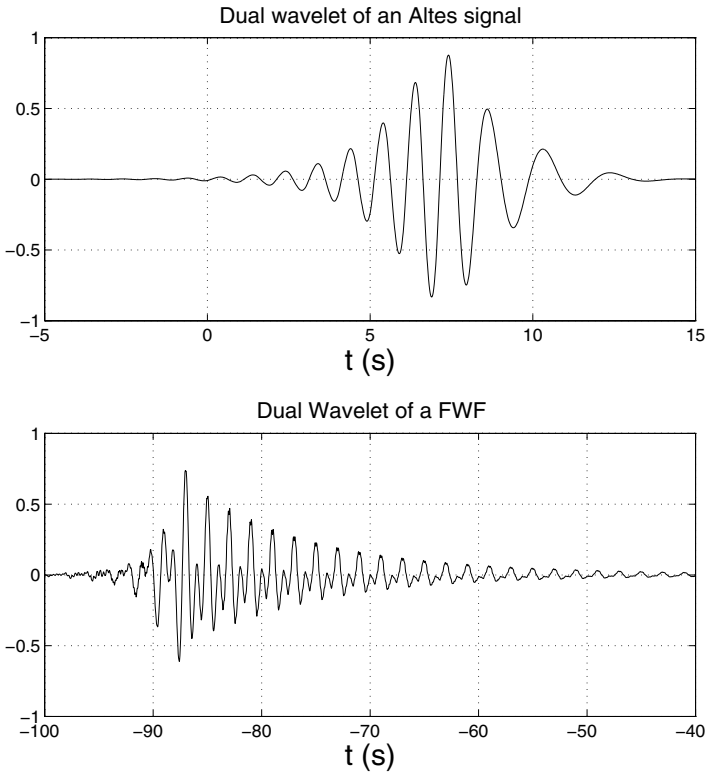

FIGURE 10. Top: The dual function of the approximation of the Altes signal. Bottom: The dual function of the formant-wave-function.

the design techniques make of the SWT a tool that is as flexible as the continuous-time wavelet transform. Moreover, this freedom in choosing the mother wavelet allows us to concentrate our research efforts on how to perform a relevant wavelet choice, given a set of data and a processing task.

References

[1] Abry, P., and Flandrin, P. (1994). On the initialization of the discrete wavelet transform. *IEEE Signal Process. Lett.*, **1**, 32–34.

[2] Aldroubi, A., and Unser, M. (1992). Families of wavelet transforms in connection with Shannon's sampling theory and the Gabor transform. *Wavelets: A Tutorial in Theory and Applications* (C. K. Chui, ed.). Academic Press, New York, 509–528.

[3] Aldroubi, A., and Unser, M. (1993). Families of multiresolution and wavelet spaces with optimal properties. *Numer. Funct. Anal. Optim.*, **14**, 417–446.

[4] Aldroubi, A., and Unser, M. (1994). Sampling procedure in function spaces and asymptotic equivalence with Shannon's sampling theory. *Numer. Funct. Anal. and Optim.*, **15**, 1–21.

[5] Altes, R. A. (1976). Sonar for generalized target description and its similarity to animal echo location systems. *J. Acoust. Soc. Amer.*, **59**, 97–105.

[6] Chui, C. K. (1991). An overview of wavelets. *Approximation theory and functional analysis* (C. K. Chui, ed.). Academic Press, New York, 47–71.

[7] Chui, C. K., and Wang, J. Z. (1991). A cardinal spline approach to wavelets. *Proc. Amer. Math. Soc.*, **113**, 785–793.

[8] Cohen, A., Daubechies, I., and Feauveau, J. C. (1992). Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, **45**, 485–560.

[9] Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, **41**, 909–996.

[10] Daubechies, I. (1992). *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

[11] Flandrin, P. (1987). Some aspects of non-stationary signal processing with emphasis on time-frequency and time-scale methods. *Wavelets, Time-Frequency Methods and Phase Space*, Marseille.

[12] Frazier M., and Jawerth, B. (1990). A discrete transform and decompositions of distribution spaces. *J. Funct. Anal.*, **93**, 34–170.

[13] Holschnider, M., Kronland-Martinet, R., Morlet, J., and Tchamitchian, Ph. (1989). A real-time algorithm for signal analysis with the help of the wavelet transform. *Wavelets, Time-Frequency Methods and Phase Space* (J. M. Combes et al., eds.). Springer, New York, 286–297.

[14] Jawerth B., and Sweldens, W. (1994). An overview of wavelet based multiresolution analyses. *SIAM Rev.*, **36**, 377–412.

[15] Lemarié, P. G., and Meyer, Y. (1986). Ondelettes et bases hilbertiennes. *Rev. Math. Iberoamericana*, **2**.

[16] Mallat, S. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Signal Process.*, **II**, 674–693.

[17] Meyer, Y. (1990). *Ondelettes et Opérateurs*. Hermann, Paris.

[18] Meyer, Y. (1992). *Ondelettes et Algorithmes Concurrents*. Hermann, Paris.

[19] Michelli, C. A. (1991). Using the refinement equation for the construction of prewavelet. *Numer. Algorithms*, **1**, 75–116.

[20] Muschietti, M. A., and Torresani, B. (1995). Pyramidal algorithms for Littlewood-Paley decompositions. *SIAM Math. Anal.*, preprint, 26.

[21] Rioul, O. (1993). A discrete-time multiresolution theory. *IEEE Trans. Signal Process.*, **41**, 2591–2606.

[22] Rodet, X. (1985). Time-domain formant-wave-function synthesis. *Computer Music J.*, **8**.

[23] Saito, N., and Belkyn, G. (1992). Multiresolution representations using the auto-correlation functions of wavelets. *International Conference on Wavelets and Applications*, Toulouse.

[24] Shensa, M. J. (1992). The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Trans. Signal Process.*, **40**, 2464–2482.

[25] Unser, M., Aldroubi, A., and Eden, M. (1992). On the asymptotic convergence of b-spline wavelets to Gabor functions. *IEEE Trans. Inform. Theory*, **38**, 864–872.

[26] Unser, M., Aldroubi, A., and Eden, M. (1993). A family of polynomial spline wavelet transforms. *Signal Process.*, **30**, 141–162.

[27] Vetterli, M., and Herley, C. (1993). Wavelets and filter banks. *IEEE Trans. Signal Process.*, **40**, 2207–2231.

Received January 3, 1994

École Normale Supérieure de Lyon (URA 1325 CNRS), 46, allée d'Italie, 69364 Lyon, Cedex 07 France

Biomedical Engineering and Instrumentation Programs, National Institutes of Health, Building 13/3N17
13 South DR MSC 5766, Bethesda, MD 20814, USA