Wavelet Transform with Tunable Q-Factor

Ivan W. Selesnick, *Senior Member, IEEE*

***Abstract*—This paper describes a discrete-time wavelet transform for which the Q-factor is easily specified. Hence, the transform can be tuned according to the oscillatory behavior of the signal to which it is applied. The transform is based on a real-valued scaling factor (dilation-factor) and is implemented using a perfect reconstruction over-sampled filter bank with real-valued sampling factors. Two forms of the transform are presented. The first form is defined for discrete-time signals defined on all of $\mathbb{Z}$. The second form is defined for discrete-time signals of finite-length and can be implemented efficiently with FFTs. The transform is parameterized by its Q-factor and its oversampling rate (redundancy), with modest oversampling rates (e.g. 3-4 times overcomplete) being sufficient for the analysis/synthesis functions to be well localized.**

***Index Terms*—wavelet transform, constant-Q transform, filter bank, Q-factor.**

I. Introduction

Ideally, the Q-factor[1] of a wavelet transform should be chosen in part according to the oscillatory behavior of the signal to which it is applied. For example, when using wavelets for the analysis and processing of oscillatory signals (speech, EEG, etc), the wavelet transform should have a relatively high Q-factor. On the other hand, when processing signals with little or no oscillatory behavior (such as a scan-line from a photographic image), the wavelet transform should have a low Q-factor. However, other than the continuous wavelet transform, most wavelet transforms provide little ability to tune the Q-factor of the wavelet. The dyadic wavelet transform has a low Q-factor and is therefore suitable for non-oscillatory (i.e. piecewise-smooth) signals [14].

This paper develops a wavelet transform for discrete-time signals for which the Q-factor is easily tunable. The transform, which we denote as the tunable-Q wavelet transform (TQWT), is parameterized by its Q-factor and its oversampling rate (redundancy). The TQWT is developed using perfect reconstruction over-sampled filter banks with real-valued scaling factors. Two forms of the transform are presented. The first form is defined for discrete-time signals defined on all of $\mathbb{Z}$. The second form is defined for discrete-time signals of finite-length and can be implemented efficiently with FFTs. Modest oversampling rates (e.g. 3-4 times overcomplete) are sufficient for the analysis/synthesis functions of the TQWT to be well localized.

The author is with the Polytechnic Institute of New York University, 6 Metrotech Center, Brooklyn, NY 11201, USA. Email: selesi@poly.edu. Tel: 718-260-3416.

This work is supported in part by NSF under grant CCF-1018020.

[1]The Q-factor of an oscillatory pulse is the ratio of its center frequency to its bandwidth.

The TQWT is closely related to the rational-dilation wavelet transform (RADWT) [5]. Like the RADWT, the TQWT is fully discrete, has the perfect reconstruction property, is modestly overcomplete, is developed in terms of iterated two-channel filter banks, and implemented using the DFT. In contrast to the RADWT, the TQWT is simpler conceptually, can be more efficiently implemented using radix-2 FFTs, and its parameters are more easily related to the Q-factor of the transform. The user can directly specify the Q-factor and redundancy of the TQWT.

The filters, on which the TQWT is based, do not have rational transfer functions. They are specified directly in the frequency domain. Like the fractional spline wavelet transform [9] which is also based on filters with non-rational transfer functions, the DFT provides 1) a means for defining the transform for finite-length discrete data which preserves the perfect reconstruction property exactly, and 2) an efficient implementation using FFTs.

A. Related work

A discrete-time wavelet transform with a continuous scaling parameter is presented in [30]. The transform, based on scaling of discrete-time signals by arbitrary scaling factors using the DTFT, is similar to the continuous wavelet transform, but developed specifically for discrete-time signals. However, the transform of [30] lacks a computationally efficient implementation. Additionally, the scaling parameter needs to be discretized in practice, which requires a departure from the theory presented in [30].

A two-channel perfect reconstruction critically-sampled filter bank with arbitrary scaling factors has been described in [23]; however, the filter bank consists of ideal low-pass and high-pass filters, hence the time-domain responses (sinc functions) are not well localized. If the filter bank of [23] is used to implement a wavelet transform, the wavelets will not be well localized.

Another method to develop wavelet transforms with adjustable Q-factors constrains the scaling factor to be rational [2], [4], [7], [16]. In this case, the transform can be constructed using perfect reconstruction filter banks with fractional rate samplers [5], [6], [8], [18], [21], [22]. Other than [5], [6], previous work taking this approach concentrate on the critically-sampled case, hence do not benefit from advantages that come with some redundancy (near shift-invariance, more flexibility in design of filter bank/wavelets, etc).

Several other methods are useful and relevant to mention. For example, wavelet packets possess the computational efficiency and perfect reconstruction properties of the filter banks from which they are constructed, while providing flexibility for designing customized frequency decompositions, as in [13],

[17], [27] for example. Other filter bank structures, and their combination with FFT processing, have been proposed for constant-Q transforms [10], [11], [15], [24], [28]. Finally, discrete/approximate implementations of the continuous wavelet transform, such as described in [19], are easily tunable and can be designed to approximately satisfy the perfect reconstruction property. In comparison, the TQWT is the result of exploring how well one can develop a tunable Q-factor wavelet transform based on the structure of the discrete (dyadic) wavelet transform (DWT) [14].

B. Organization of paper

Section II defines low-pass and high-pass scaling of discrete-time signals, on which the tunable-Q wavelet transform is based. Section III introduces a two-channel filter bank with real-valued scaling factors, develops the perfect reconstruction (PR) conditions, and gives a low-pass/high-pass pair of filters satisfying the PR conditions. Section IV presents the discrete-time wavelet transform with real-valued scaling (dilation) factor, and analyzes its relevant parameters (Q-factor, etc). This form of the TQWT, developed using the DTFT, and defined for discrete-time signals defined on all of $\mathbb{Z}$, is not readily implemented. Section V presents a second form of the TQWT that is defined for discrete-time signals of finite-length. This form is readily implemented using the DFT. In Sec. V we also show how the TQWT may be implemented using only radix-2 FFTs, in order that the transform be computationally efficient.

C. Notation

This paper deals exclusively with discrete-time signals, i.e., $x(n)$ defined on $n \in \mathbb{Z}$. The discrete-time Fourier transform (DTFT) of the discrete-time signal $x(n)$ is defined as

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)\, \exp\left(-\mathrm{j}\, n\omega\right).$$

Note that the DTFT $X(\omega)$ of a discrete-time signal $x(n)$ is always a 2π-periodic function of ω; therefore, it is sufficient to specify $X(\omega)$ for $|\omega| \leq \pi$.

A finite-length discrete-time signal $x(n)$ defined for $0 \leq n \leq N-1$ will be denoted by a lower case bold letter, $\mathbf{x} = [x(0), \ldots, x(N-1)]$. The discrete Fourier transform (DFT) of the N-point sequence $\mathbf{x}$ is defined as

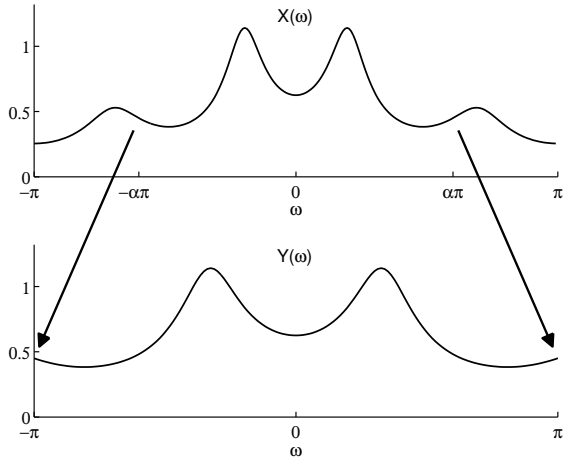
$$X(k) = \sum_{n=0}^{N-1} x(n)\, \exp\left(-\mathrm{j}\frac{2\pi}{N} nk\right), \qquad (1)$$

for $0 \leq k \leq N-1$. The DFT of a finite-length signal $\mathbf{x}$ will be denoted by an upper case letter, $\mathbf{X} = \mathrm{DFT}\{\mathbf{x}\}$, with $\mathbf{X} = [X(0), \ldots, X(N-1)]$. The unitary DFT (uDFT) is the DFT normalized so as to be unitary:

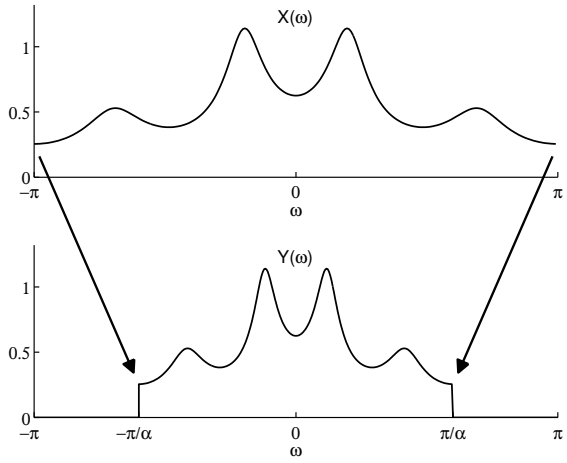
$$X(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n)\, \exp\left(-\mathrm{j}\frac{2\pi}{N} nk\right), \qquad (2)$$

for $0 \leq k \leq N-1$. We use the notation $\mathbf{X} = \mathrm{uDFT}\{\mathbf{x}\}$.

(a) Low-pass scaling block diagram



(b) Low-pass scaling with $\alpha < 1$



(c) Low-pass scaling with $\alpha > 1$.

Fig. 1. Low-pass scaling with parameter α. The output signal has a sampling rate of $\alpha\, f_s$ where f_s is the sampling rate of the input signal.

II. Scaling

A. Low-pass Scaling

By low-pass scaling, we refer to frequency-domain scaling that preserves the low-frequency content of the signal. For low-pass scaling with scaling parameter α, denoted as in Fig. 1, the output signal has a sampling rate of $\alpha\, f_s$ where f_s is the sampling rate of the input signal. Depending on the scaling parameter, low-pass scaling either increases or decreases the sampling rate of the signal.

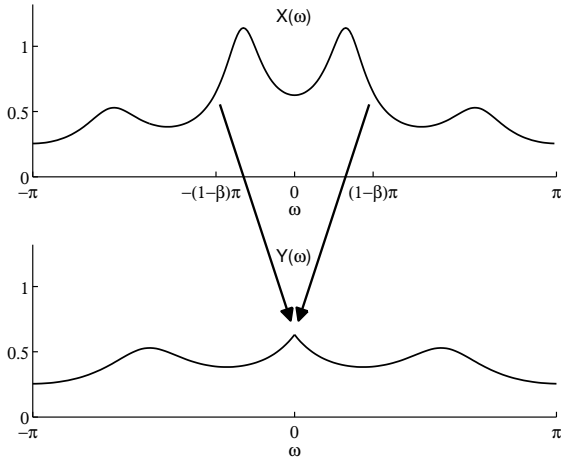
When $0 < \alpha \leq 1$, low-pass scaling with parameter α is defined as:

$$Y(\omega) = X(\alpha\, \omega), \quad |\omega| \leq \pi.$$

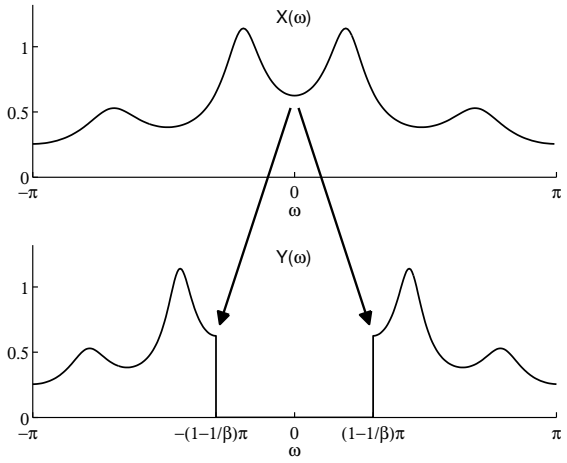
When $\alpha \geq 1$, low-pass scaling is defined as:

$$Y(\omega) = \begin{cases} X(\alpha\, \omega), & |\omega| \leq \pi/\alpha \\ 0, & \pi/\alpha < |\omega| \leq \pi. \end{cases}$$

(a) High-pass scaling block diagram



(b) High-pass scaling with $\beta < 1$



(c) High-pass scaling with $\beta > 1$.

Fig. 2. High-pass scaling with parameter β. The output signal has a sampling rate of $\beta\, f_s$ where f_s is the sampling rate of the input signal.

Note that for low-pass scaling, $Y(0) = X(0)$. Low-pass scaling preserves the signal behavior around dc ($\omega = 0$).

Low-pass scaling is essentially the conventional discrete-time rate changer where the interpolation filter is an ideal low-pass filter [20, Eqn. 13.27]. However, here we allow the rate-change α to be non-rational.

B. High-pass Scaling

By high-pass scaling, we refer to frequency-domain scaling that preserves the high-frequency content of the signal. For high-pass scaling with scaling parameter β, denoted as in Fig. 2, the output signal has a sampling rate of $\beta\, f_s$ where f_s is the sampling rate of the input signal.

When $0 < \beta \leq 1$, high-pass scaling with parameter β is defined as:

$$Y(\omega) = \begin{cases} X(\beta\,\omega + (1-\beta)\pi), & 0 < \omega < \pi \\ X(\beta\,\omega - (1-\beta)\pi), & -\pi < \omega < 0 \end{cases}$$

When $\beta \geq 1$, high-pass scaling is defined as:

$$Y(\omega) = \begin{cases} 0, & |\omega| < (1-1/\beta)\,\pi \\ X(\beta\,\omega + (1-\beta)\pi), & (1-1/\beta)\,\pi < \omega < \pi \\ X(\beta\,\omega - (1-\beta)\pi), & -\pi < \omega < -(1-1/\beta)\pi. \end{cases}$$

Note that for high-pass scaling, $Y(\pi) = X(\pi)$. High-pass scaling preserves the signal behavior around the Nyquist frequency ($\omega = \pi$). High-pass scaling is illustrated in Fig. 2 for $\beta < 1$ and for $\beta > 1$.

C. Scaling Identities

Several identities will be useful in subsequent sections. First, when $\alpha \leq 1$, we have the system equivalence illustrated in Fig. 3 where the 'equivalent' frequency response is given by

$$F_{\text{eq}}(\omega) := \begin{cases} F(\omega), & |\omega| \leq \alpha\,\pi \\ 0, & \alpha\,\pi < |\omega| \leq \pi. \end{cases} \tag{3}$$

Similarly, when $\beta \leq 1$, we have the system equivalence illustrated in Fig. 4 where

$$F_{\text{eq}}(\omega) := \begin{cases} 0, & |\omega| \leq (1-\beta)\,\pi \\ F(\omega), & (1-\beta)\,\pi < |\omega| \leq \pi. \end{cases} \tag{4}$$

The wavelet transform is implemented by concatenating filter banks; therefore we need the following identity. When

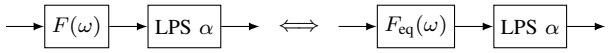


Fig. 3. The systems are equivalent when $F_{\text{eq}}(\omega)$ is defined by (3).

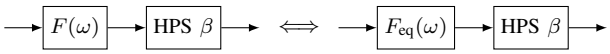


Fig. 4. The systems are equivalent when $F_{\text{eq}}(\omega)$ is defined by (4).

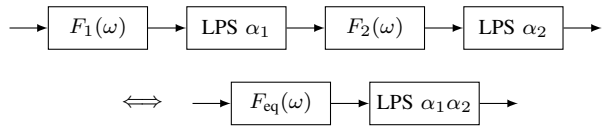


Fig. 5. The systems are equivalent when $F_{\text{eq}}(\omega)$ is defined by (5).

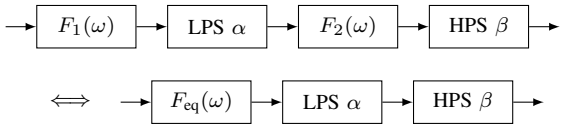


Fig. 6. The systems are equivalent when $F_{\text{eq}}(\omega)$ is defined by (6).

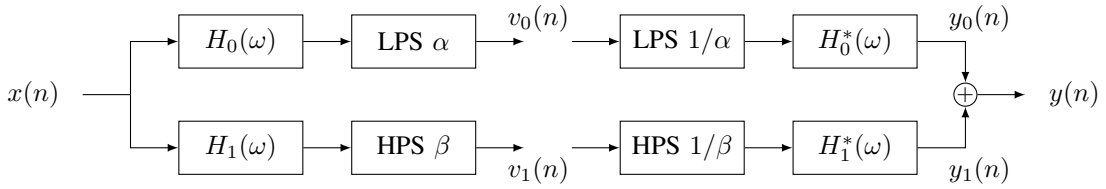


Fig. 7. Analysis and synthesis filter banks for the tunable-Q wavelet transform. The subband signal $v_0(n)$ has a sampling rate of αf_s where f_s is the sampling rate of the input signal $x(n)$. Likewise, the subband signal $v_1(n)$ has a sampling rate of βf_s. LPS and HPS represent low-pass scaling and high-pass scaling respectively.

$\alpha_1 \leq 1$, $\alpha_2 \leq 1$, we have the system equivalence illustrated in Fig. 5where

$$F_{\text{eq}}(\omega) := \begin{cases} F_1(\omega)F_2(\omega/\alpha_1), & |\omega| \leq \alpha_1\alpha_2\,\pi \\ 0, & \alpha_1\alpha_2\,\pi < |\omega| \leq \pi. \end{cases} \tag{5}$$

Similarly, when $\alpha \leq 1$, $\beta \leq 1$, we have the system equivalence illustrated in Fig. 6 where

$$F_{\text{eq}}(\omega) := \begin{cases} 0, & |\omega| < (1-\beta)\,\alpha\,\pi \\ F_1(\omega)F_2(\omega/\alpha), & (1-\beta)\,\alpha\,\pi \leq |\omega| \leq \alpha\,\pi \\ 0, & \alpha\,\pi < |\omega| \leq \pi. \end{cases} \tag{6}$$

III. Filter Bank

The tunable-Q wavelet transform will be based on the multirate filter bank illustrated in Fig. 7. The low-pass subband signal $v_0(n)$ and high-pass subband signal $v_1(n)$ have sampling rates of αf_s and βf_s respectively, where f_s is the sampling rate of the input signal $x(n)$. The scaling parameters satisfy

$$0 < \beta \leq 1, \qquad 0 < \alpha < 1$$

so as to ensure the wavelet transform will not be overly redundant. In order that perfect reconstruction be possible, it is necessary that $\alpha + \beta \geq 1$. In order that the filter responses be well localized, we ask that the filter bank be strictly oversampled; so we require that

$$\alpha + \beta > 1.$$

For perfect reconstruction, the frequency responses $H_i(\omega)$, $i = 0, 1$, must be chosen so that the reconstructed signal $y(n)$ equals the input signal $x(n)$. Using the low-pass and high-pass scaling relations defined in Sections II-A and II-B, the Fourier transforms (DTFT) of $y_0(n)$ and $y_1(n)$ in Fig. 7 are given by

$$Y_0(\omega) = \begin{cases} |H_0(\omega)|^2\,X(\omega), & |\omega| \leq \alpha\,\pi \\ 0, & \alpha\,\pi < |\omega| \leq \pi \end{cases} \tag{7}$$

and

$$Y_1(\omega) = \begin{cases} 0, & |\omega| < (1-\beta)\,\pi \\ |H_1(\omega)|^2\,X(\omega), & (1-\beta)\,\pi \leq |\omega| \leq \pi. \end{cases} \tag{8}$$

Hence, the Fourier transform of $y(n)$ is given by

$$Y(\omega) = \begin{cases} |H_0(\omega)|^2\,X(\omega), & \omega \in \mathcal{P} \\ (|H_0(\omega)|^2 + |H_1(\omega)|^2)\,X(\omega), & \omega \in \mathcal{T} \\ |H_1(\omega)|^2\,X(\omega), & \omega \in \mathcal{S} \end{cases}$$

where the intervals are defined as

$$\mathcal{P} = \{|\omega| < (1-\beta)\,\pi\}$$
$$\mathcal{T} = \{(1-\beta)\,\pi \leq |\omega| < \alpha\,\pi\}$$
$$\mathcal{S} = \{\alpha\,\pi \leq |\omega| \leq \pi\}.$$

Note that these three sets are a partition of $\{|\omega| \leq \pi\}$, as illustrated in Fig. 8.

Perfect reconstruction requires that $Y(\omega) = X(\omega)$. Therefore, for perfect reconstruction, the low-pass filter $H_0(\omega)$ should satisfy

$$|H_0(\omega)| = 1, \qquad |\omega| \leq (1-\beta)\,\pi \tag{9}$$
$$H_0(\omega) = 0, \qquad \alpha\,\pi \leq |\omega| \leq \pi \tag{10}$$

and the high-pass filter $H_1(\omega)$ should satisfy

$$H_1(\omega) = 0, \qquad |\omega| \leq (1-\beta)\,\pi \tag{11}$$
$$|H_1(\omega)| = 1, \qquad \alpha\,\pi \leq |\omega| \leq \pi \tag{12}$$

as illustrated in Fig. 8b. Additionally, the transition bands of $H_0(\omega)$ and of $H_1(\omega)$ must be chosen so that

$$|H_0(\omega)|^2 + |H_1(\omega)|^2 = 1, \quad \omega \in \mathcal{T}. \tag{13}$$

The interval $\mathcal{P}$ constitutes the pass-band of $H_0(\omega)$ and the stop-band of $H_1(\omega)$. The interval $\mathcal{S}$ constitutes the stop-band of $H_0(\omega)$ and the pass-band of $H_1(\omega)$. While the interval $\mathcal{T}$ constitutes the transition-bands of the filters.

The transition bands of $H_0(\omega)$ and $H_1(\omega)$ can be constructed using any 2π-periodic power-complementary function. If a function $\theta(\omega)$ satisfies

$$\theta^2(\omega) + \theta^2(\pi - \omega) = 1, \tag{14}$$

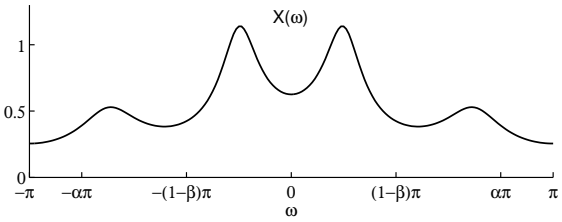
then by scaling and translating $\theta(\omega)$ and $\theta(\pi - \omega)$ from the interval $[0,\, \pi]$ to the interval $[(1-\beta)\,\pi,\, \alpha\,\pi]$ we obtain transition functions for $H_0(\omega)$ and $H_1(\omega)$ respectively so that (13) is satisfied. Specifically, the transition bands are given in terms of $\theta(\omega)$ by

$$H_0(\omega) = \theta\left(\frac{\omega + (\beta-1)\pi}{\alpha+\beta-1}\right), \tag{15}$$

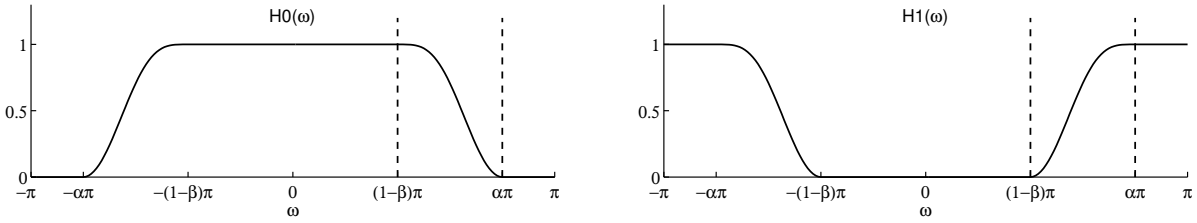
$$H_1(\omega) = \theta\left(\frac{\alpha\,\pi - \omega}{\alpha+\beta-1}\right), \tag{16}$$

for $(1-\beta)\,\pi < \omega < \alpha\,\pi$. As in [5, Eqn. 28], we choose to use the Daubechies frequency response [14] with two vanishing moments,

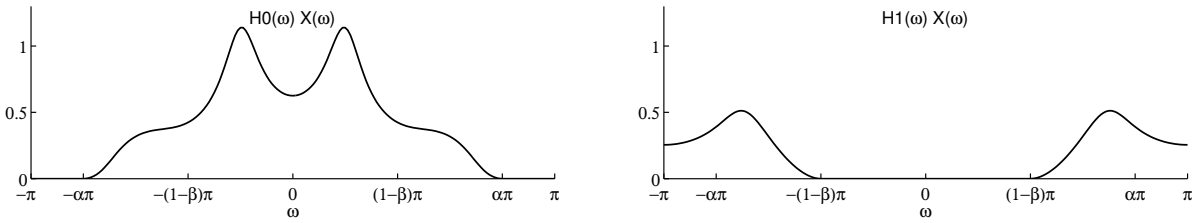
$$\theta(\omega) = 0.5\,(1+\cos\omega)\,\sqrt{2-\cos\omega}, \quad |\omega| \leq \pi, \tag{17}$$



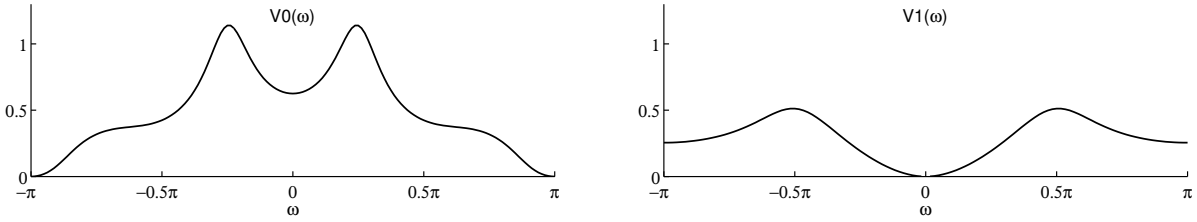
(a) Fourier transform of input signal, $X(\omega)$.



(b) Frequency responses $H_0(\omega)$ and $H_1(\omega)$.



(c) Fourier transforms of input signal after filtering.



(d) Fourier transforms after scaling.

Fig. 8. Behavior of the two-channel analysis filter bank, illustrated using $\alpha = 0.8$ and $\beta = 0.6$.

which satisfies (14).

Given α and β, the low-pass filter $H_0(\omega)$ given by (9), (10) and (15), and the high-pass filter $H_1(\omega)$ given by (11), (12) and (16), together form a perfect reconstruction pair of filters for the two-channel filter bank illustrated in Fig. 7.

As illustrated in Fig. 8b, the two frequency responses are identically unity in their pass-bands and identically zero in their stop-bands. But they are not ideal low-pass and high-pass filters due their transition bands over the common interval $(1-\beta)\,\pi \leq |\omega| \leq \alpha\,\pi$. Note that the width of the transition band, $(\alpha + \beta - 1)\pi$, is exactly the amount by which the filter bank exceeds the critical sampling rate. If $\alpha + \beta = 1$, then the filter bank in Fig. 7 is critically-sampled, the transition-band has width zero, $H_0(\omega)$ and $H_1(\omega)$ are 'ideal' filters, and their time-domain responses (being sampled sinc functions) are poorly localized; this is not the sought behavior. It is important that $\alpha + \beta$ be strictly greater than unity so that the time-domain responses, $h_0(n)$ and $h_1(n)$, are well localized.

Figure 8 illustrates the behavior of the two-channel filter bank. Figure 8a illustrates the Fourier transform (DTFT), $X(\omega)$, of a discrete-time signal. Figure 8b illustrates the low-pass and high-pass frequency responses. Figure 8c illustrates the Fourier transforms after filtering; and Fig. 8d illustrates the Fourier transforms after subsequent low-pass and high-pass scaling. Note that after scaling, the spectra occupy the full frequency band.

IV. Wavelet Transform

The tunable-Q wavelet transform (TQWT) is implemented by iteratively applying the two-channel filter bank on its low-pass channel. For example, a three-stage wavelet transform is illustrated in Fig. 9. The wavelet transform inherits the perfect reconstruction property from the two channel filter bank. We denote the wavelet subband signals by $w^{(j)}(n)$ for $j \geq 1$, with

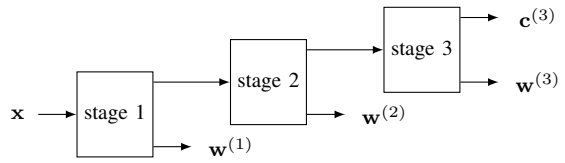


Fig. 9. Wavelet filter bank. Each stage consists of the two-channel analysis filter bank in Fig. 7.

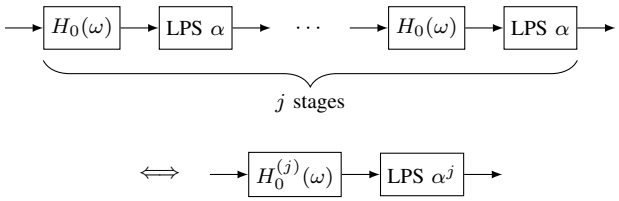


Fig. 10. The systems are equivalent where $H_0^{(j)}(\omega)$ is given by (18).

$j = 1$ being the high-pass subband produced by the first stage (as in Fig. 9). The sampling rate at subband j is given by $\beta\,\alpha^{j-1}\,f_s$ where f_s is the sampling rate of the input signal.

To derive the frequency decomposition provided by the wavelet transform, we need to analyze the iteration (cascade) of several filters and scalings. Using the basic scaling identities in Section II-C, we can obtain the following identities.

When $\alpha \leq 1$, we have the system equivalence illustrated in Fig. 10 where

$$H_0^{(j)}(\omega) := \begin{cases} \displaystyle\prod_{m=0}^{j-1} H_0(\omega/\alpha^m), & |\omega| \leq \alpha^j\,\pi \\ 0, & \alpha^j\,\pi < |\omega| \leq \pi. \end{cases} \tag{18}$$

When $\alpha \leq 1$, $\beta \leq 1$, we have the system equivalence illustrated in Fig. 11 where the equivalent frequency response is given by

$$H_1^{(j)}(\omega) := \begin{cases} H_1(\omega/\alpha^{j-1}) \displaystyle\prod_{m=0}^{j-2} H_0(\omega/\alpha^m), \\ \qquad (1-\beta)\,\alpha^{j-1}\,\pi \leq |\omega| \leq \alpha^{j-1}\,\pi \\ 0, \qquad \text{for other } \omega \in [-\pi,\,\pi] \end{cases} \tag{19}$$

This frequency response relates the input signal $x(n)$ to the subband $w^{(j)}(n)$. Figure 12 illustrates the frequency response $H_1^{(j)}(\omega)$. Figure 13 illustrates the frequency responses $H_1^{(j)}(\omega)$, for $1 \leq j \leq J$ of a J-level transform, for four different values of (α, β).

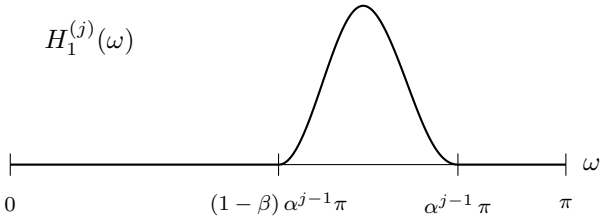


Fig. 12. The subband-j frequency response $H_1^{(j)}(\omega)$, given by (19), is non-zero only on the interval indicated.

The samples of discrete-time wavelet $\psi(t)$ are obtained as the inverse DTFT of $H_1^{(j)}(\omega)$ in the limit as j goes to infinity.

A. Parameters

Figure 13 illustrates the frequency decomposition of the wavelet transform for four choices of the filter bank parameters α and β. By varying α and β, the type of frequency decomposition can be adjusted with some flexibility. In particular, the Q-factor can be continuously tuned. The following discussion clarifies the relationship between the parameters α and β, and the characteristics of the resulting frequency decomposition.

Oversampling rate (redundancy): The two-channel filter bank illustrated in Fig. 7 is oversampled by a factor of $\alpha + \beta$. If the two-channel filter bank is iterated on its low-pass output ad infinitum so as to implement a wavelet transform, then the wavelet transform is oversampled by a factor of

$$r = \frac{\beta}{1-\alpha} \tag{20}$$

which we call the redundancy r of the wavelet transform. This expression is obtained by noting that the sampling rate at subband j (with $j \geq 1$) is given by $\beta\,\alpha^{j-1}\,f_s$ where f_s is the sampling rate of the input signal. The sum of the sampling rates over all subbands $j \geq 1$ gives $\beta/(1-\alpha)\,f_s$ and hence the oversampling rate in (20).

Center frequency: From (19), the level-j frequency response, denoted $H_1^{(j)}(\omega)$, is non-zero in the interval (ω_1, ω_2) where

$$\omega_1 = (1-\beta)\,\alpha^{j-1}\,\pi, \quad \omega_2 = \alpha^{j-1}\,\pi, \tag{21}$$

as illustrated in Fig. 12. The center frequency at level j is (approximately) the average of ω_1 and ω_2,

$$\omega_c = \frac{1}{2}(\omega_1 + \omega_2) = \alpha^j\,\frac{2-\beta}{2\alpha}\,\pi \tag{22}$$

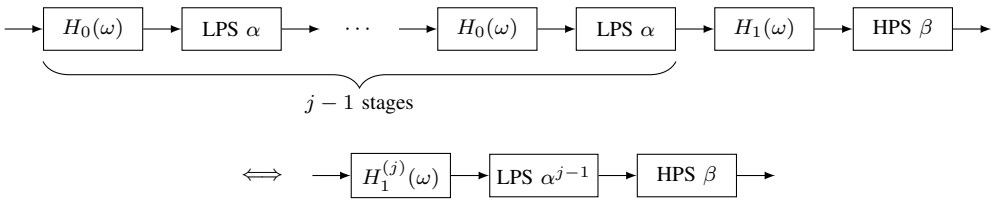
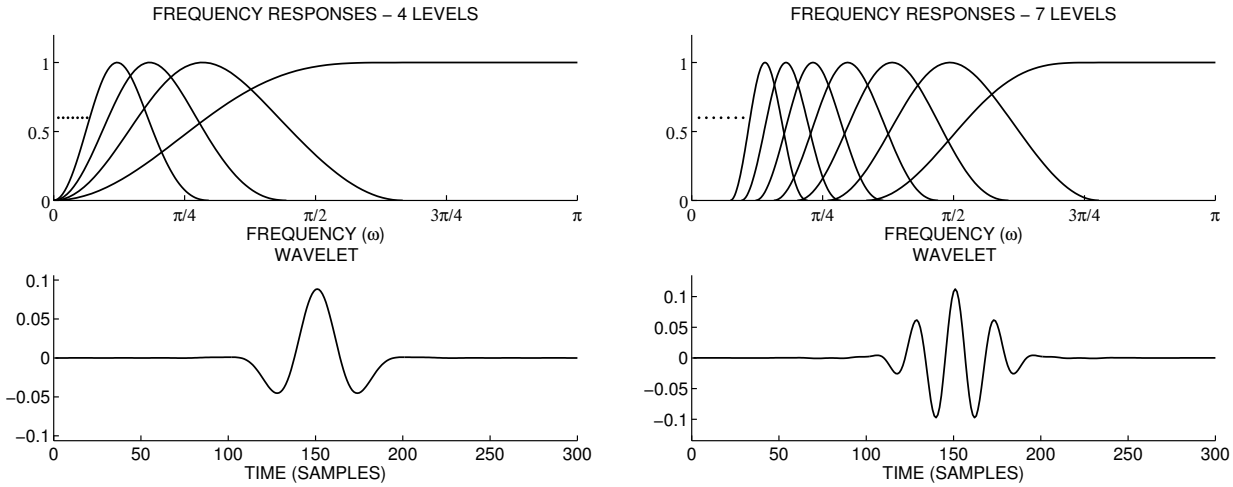
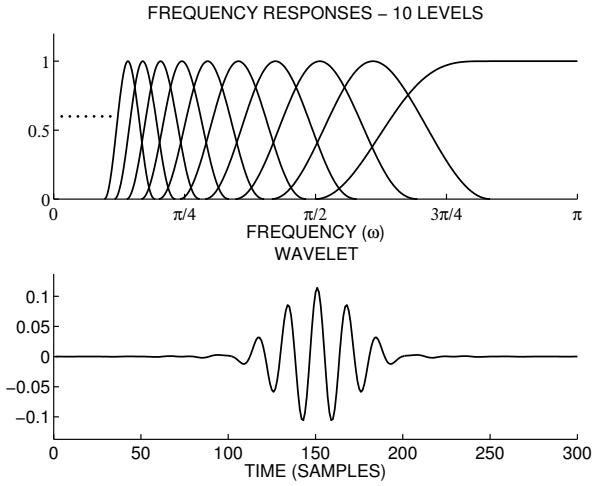


Fig. 11. The systems are equivalent where $H_1^{(j)}(\omega)$ is given by (19).

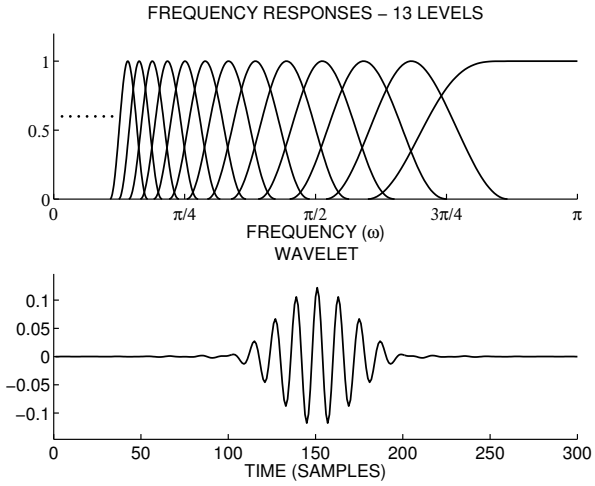


(a) $Q = 1$, $r = 3$ ($\alpha = 0.667$, $\beta = 1$)

(b) $Q = 2$, $r = 3$ ($\alpha = 0.778$, $\beta = 0.667$)



(c) $Q = 3$, $r = 3$ ($\alpha = 0.833$, $\beta = 0.5$)



(d) $Q = 4$, $r = 3$ ($\alpha = 0.867$, $\beta = 0.4$)

Fig. 13. Frequency decomposition of the wavelet transform with tunable Q-factor, implemented by iteration of the filter bank in Fig. 7. In the figure, each frequency response has been normalized to have unity peak gain.

The units of ω_c are radians per sample. In terms of the input signal sampling rate f_s, the center frequency at level j is

$$f_c = \alpha^j \, \frac{2-\beta}{4\alpha} \, f_s. \tag{23}$$

Bandwidth: From Fig. 12, it can be seen that the bandwidth of the frequency response producing subband j is approximately half the width of the interval over which the frequency response is non-zero. Using this approximation, the bandwidth is given by

$$\text{BW} = \frac{1}{2}(\omega_2 - \omega_1) = \frac{1}{2}\beta \, \alpha^{j-1} \, \pi. \tag{24}$$

Q-factor: As the tunability of the Q-factor is one motivation for the wavelet transform described here, it is useful to express the Q-factor in terms of α and β. Using (22) and (24), the Q-factor of the level-j frequency response is given by

$$Q := \frac{\omega_c}{\text{BW}} = \frac{2-\beta}{\beta}. \tag{25}$$

Note that the Q-factor does not depend on the level, j. As expected, the wavelet transform is a constant-Q transform. Moreover, it depends only on the filter bank parameter β. (Note, however, that (22)-(25) are not valid for the first level ($j = 1$) because, as illustrated in Fig. 13, the level-1 frequency response is wider than those of subsequent levels. From (19), the level-1 frequency response $H_1^{(1)}(\omega)$ is just the high-pass frequency response $H_1(\omega)$.)

Selecting α and β: The foregoing discussion suggests how the filter bank parameters α and β should be chosen so as to achieve a wavelet transform with the desired Q-factor and oversampling rate r. Namely, using (20) and (25), we can express α and β in terms of the Q-factor and redundancy:

$$\beta = \frac{2}{Q+1}, \quad \alpha = 1 - \frac{\beta}{r}. \tag{26}$$

The specified Q-factor should be chosen subject to $Q \geq 1$. Setting $Q = 1$ leads to a wavelet transform for which the wavelet resembles the second derivative of a Gaussian, as illustrated in Fig. 13a. Higher values of Q lead to more

oscillatory wavelets. The specified oversampling rate r must be strictly greater than 1. If r is close to unity, then the transition bands of $H_0(\omega)$ and $H_1(\omega)$ will be relatively narrow and the time-domain response (wavelet) will not be well localized. (For $r \approx 1$, the wavelet will resemble the sinc wavelet.) In order to avoid this issue, it is sufficient to select $r \geq 3$. With $r \geq 3$, the pass-band of the level-j frequency response will not have a 'flat top' (wherein the frequency response is equal to a constant over a sub-interval of its pass-band) as discussed in [5].

Note that neither Q nor r need be integers (although integer values are used Fig. 13).

Vanishing moments: When processing piecewise smooth (locally-polynomial) signals, the number of vanishing moments of a wavelet transform is of some interest. On the other hand, when processing oscillatory signals, the number of vanishing moments does not seem to be particularly relevant (as vanishing moments relate specifically to polynomial approximation properties). Therefore, we discuss the vanishing moments properties of the TQWT only briefly. If $\beta = 1$ (that is, $Q = 1$) then the TQWT has two vanishing moments due to the use of the Daubechies filter with two vanishing moments in (17). (Using a Daubechies filter with K vanishing moments in (17) would yield a TQWT with K vanishing moments). If $\beta < 1$ (that is, $Q > 1$), then the TQWT has infinitely many vanishing moments; this is because the stop-band of the filter $H_1(\omega)$ is identically zero for $|\omega| \leq (1-\beta)\pi$, an interval containing the origin.

V. Signals of finite-length

The form of the tunable Q-factor wavelet transform (TQWT) described in Sec. IV calls for unrealizable filters. (If the frequency response of a filter is constant-valued over some interval, as above, then the filter can not be implemented using a finite-order difference equation.) However, the transform can be adapted to finite-length signals in such a way that the implementation is relatively straight-forward. This section develops the TQWT for discrete-time signals of finite-length. This form of the transform is readily implemented using the DFT and maintains the perfect reconstruction property. For this DFT-based implementation, one still has the ability to finely tune the Q-factor of the transform.

In order to adapt the tunable Q-factor wavelet transform to finite-length signals, the definition of low-pass scaling and high-pass scaling in Sec. II-A and II-B is adapted to finite-length signals in Sec. V-A and V-B. Section V-C adapts the filter bank analysis of Sec. III to the finite-length case and develops the corresponding perfect reconstruction conditions. Section V-D describes the TQWT for finite-length signals. Section V-E describes how the finite-length TQWT can be implemented using only radix-2 FFTs.

A. Low-pass scaling: finite-length signals

To define low-pass scaling for finite-length signals, it is convenient to specify the length of the input and output signals explicitly. We will use the notation $N : N_0$ where N denotes the length of the input signal, and N_0 denotes the length of the output signal. Then, when $N_0 < N$, scaling constitutes a reduction of the sampling rate.

Let $x(n)$ be an N-point signal, defined for $0 \leq n \leq N-1$. If $N_0 < N$, and N_0 and N are both even, then we define low-pass scaling $N : N_0$ as

$$\begin{aligned} &Y(k) = X(k), && 0 \leq k \leq N_0/2 - 1 \\ &Y(N_0/2) = X(N/2) && \\ &Y(N_0 - k) = X(N-k), && 1 \leq k \leq N_0/2 - 1 \end{aligned}$$

where $X(k)$ and $Y(k)$ denote the DFTs of input and output signals. Similarly, if $N_0 > N$, then we define low-pass scaling $N : N_0$ as

$$\begin{aligned} &Y(k) = X(k), && 0 \leq k \leq N/2 - 1 \\ &Y(k) = 0, && N/2 \leq k \leq N_0/2 - 1 \\ &Y(N_0/2) = X(N/2) && \\ &Y(N_0 - k) = 0, && N/2 \leq k \leq N_0/2 - 1 \\ &Y(N_0 - k) = X(N-k), && 1 \leq k \leq N/2 - 1. \end{aligned}$$

This constitutes an increase of the sampling rate. It is essentially DFT-based interpolation.

Low-pass scaling is illustrated in Fig. 14. Note that if $N_0 \geq N$, then low-pass scaling $N : N_0$ is invertible, with the inverse being low-pass scaling $N_0 : N$. Low-pass scaling is defined so as to preserve $X(N_0/2)$ so that this inverse property holds. Note that in the filter bank of interest, low-pass scaling follows low-pass filtering which annihilates $X(N_0/2)$, so this DFT value is not propagated in the TQWT.

B. High-pass scaling: finite-length signals

High-pass scaling preserves the spectrum of the signal around the Nyquist frequency. For an N-point signal, that corresponds to the DFT coefficient with index $k = N/2$.

Let $x(n)$ be an N-point signal, defined for $0 \leq n \leq N-1$. If $N_1 < N$, and N_1, N are both even, then we define high-pass scaling $N : N_1$ as

$$\begin{aligned} &Y(0) = X(0) \\ &Y(N_1/2 - k) = X(N/2 - k), \quad |k| \leq N_1/2 - 1. \end{aligned}$$

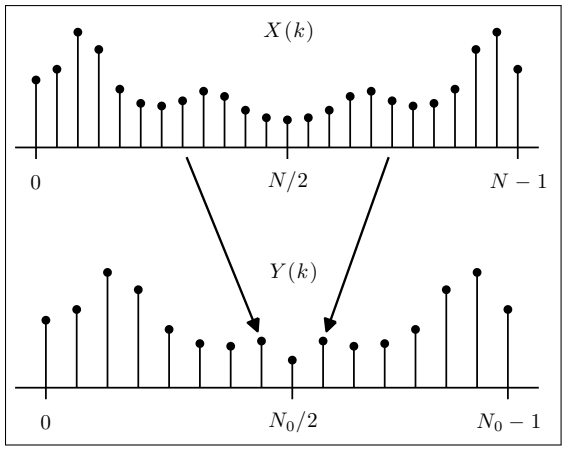
Similarly, if $N_1 > N$, then we define high-pass scaling $N : N_1$ as

$$\begin{aligned} &Y(0) = X(0) && \\ &Y(k) = 0, && 1 \leq k \leq (N_1 - N)/2 \\ &Y(N_1/2 - k) = X(N/2 - k), && |k| \leq N/2 - 1 \\ &Y(N_1 - k) = 0, && 1 \leq k \leq (N_1 - N)/2. \end{aligned}$$

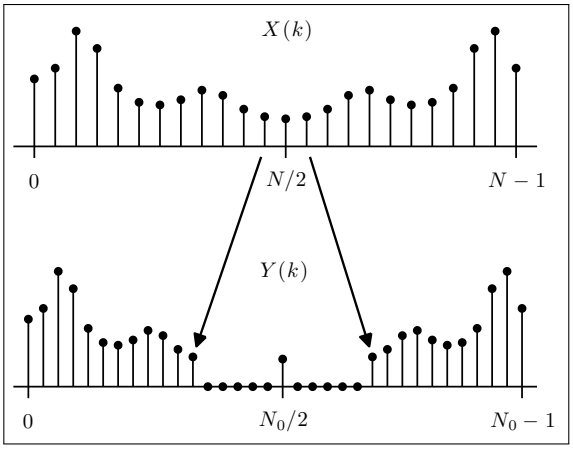
High-pass scaling is illustrated in Fig. 15. Note that if $N_1 \geq N$, then high-pass scaling $N : N_1$ is invertible, with the inverse being high-pass scaling $N_1 : N$. High-pass scaling is defined so as to preserve $X(0)$ so that this inverse property holds. In the filter bank of interest, high-pass scaling follows high-pass filtering which annihilates $X(0)$, so this DFT value is not present in the wavelet subbands.

Low-pass and high-pass scaling can also be defined for signals of odd-length, but we omit it here, assuming the input signal can be taken to be even-length.

(a) Low-pass scaling block diagram

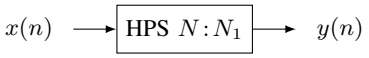


(b) Low-pass scaling with $N_0 < N$

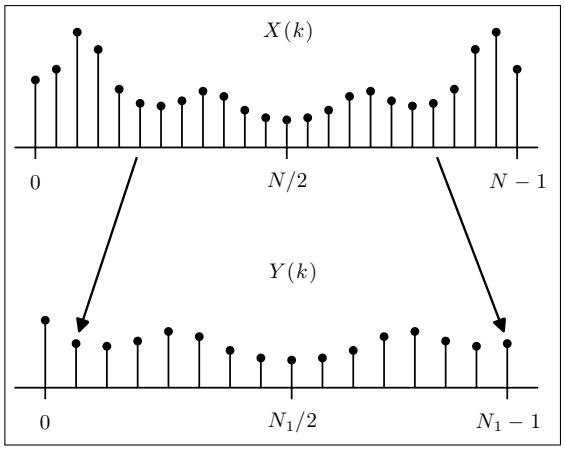


(c) Low-pass scaling with $N_0 > N$.

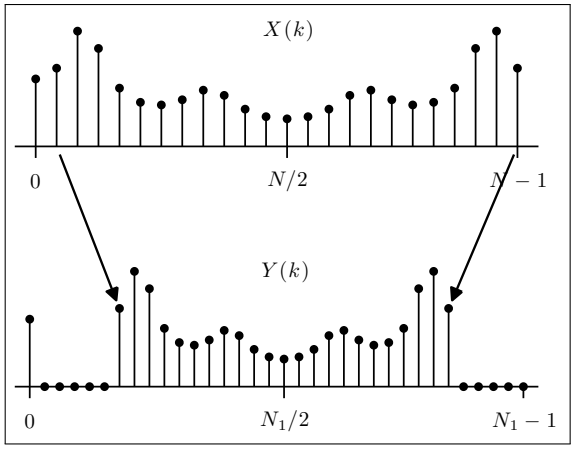
Fig. 14. Low-pass scaling $N : N_0$. The input and output signals are of lengths N and N_0 respectively.



(a) High-pass scaling block diagram



(b) High-pass scaling with $N_1 < N$



(c) High-pass scaling with $N_1 > N$.

Fig. 15. High-pass scaling $N : N_1$. The input and output signals are of lengths N and N_1 respectively.

C. Two-channel filter bank

Figure 16 illustrates the filter bank of Fig. 7 adapted for finite-length signals. Referring to Fig. 16, the integer N denotes the length of the input signal. The integers N_0 and N_1 denote the lengths of the respective subband signals, $v_0(n)$ and $v_1(n)$. The filters are implemented in the DFT-domain by point-by-point multiplication, hence $H_0(k)$ and $H_1(k)$ need to be defined for integers k, $0 \le k \le N-1$. Instead of low-pass/high-pass scalings by real α and β, the filter bank uses the low-pass/high-pass scaling operations defined in Sec. V-A and V-B.

For a given input signal length N, and given parameters α and β, we would like that the filter bank in Fig. 16 closely emulates the filter bank in Fig. 7. Therefore, we should have $N_0 \approx \alpha N$ and $N_1 \approx \beta N$. However, αN and βN will generally not be integers, so they must be rounded. For convenience[2], we round αN and βN to the nearest even integers,

$$N_0 = 2\,\mathrm{round}\Big(\frac{\alpha}{2}N\Big), \quad N_1 = 2\,\mathrm{round}\Big(\frac{\beta}{2}N\Big). \qquad (27)$$

Accordingly, the effective scalings parameters are N_0/N and N_1/N instead of α and β. That is, the scaling parameters can not be continuously varied. However, for a signal of reasonable length (N large), the approximation of α by N_0/N can be quite accurate. In contrast, filter banks based on rational-sampling factors usually require in practice that the rational factor be a ratio of two relatively small integers. In the DFT-based implementation described here, there is no need to impose such a constraint. Therefore, the scaling parameter can

[2]In practice, the radix-2 version of the TQWT in Sec. V-E will be the most efficient implementation, and in this version all signals are of even length anyway.

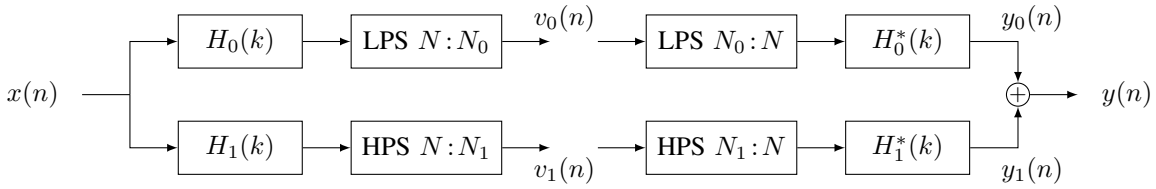


Fig. 16. Analysis and synthesis filter banks for a finite-length N-point signal $x(n)$. The subband signals $v_0(n)$ and $v_1(n)$ are of length N_0 and N_1 respectively.

be quite finely varied even though it can not be continuously varied.

Unlike the filter bank of Fig. 7, this filter bank is readily and exactly realized. To implement the analysis filter bank, one proceeds as follows. Let the input signal $x(n)$ be defined for $0 \le n \le N-1$. First, one computes the N-point DFT of the input signal $x(n)$ to obtain DFT coefficients $X(k)$ for $0 \le k \le N-1$. Second, one performs point-by-point multiplication to form $R_i(k) = X(k)H_i(k)$, for $i = 0, 1$. Third, one performs low-pass/high-pass scaling on $R_i(k)$ to obtain the DFT coefficients $V_i(k)$, $0 \le k \le N_i$, for $i = 0, 1$. Finally one computes the N_0-point inverse DFT of $V_0(k)$ to obtain $v_0(n)$ for $0 \le n \le N_0 - 1$; and one computes the N_1-point inverse DFT of $V_1(k)$ to obtain $v_1(n)$ for $0 \le n \le N_1 - 1$.

Perfect reconstruction: Let N, N_0, and N_1 be even integers with $N_0 + N_1 > N$. Using the definitions of low-pass and high-pass scaling in Sections V-A and V-B, the DFT of the N-point signals $y_0(n)$ and $y_1(n)$ in Fig. 16 can be expressed as follows. First we define the integers

$$P = (N - N_1)/2 \tag{28}$$
$$S = (N - N_0)/2 \tag{29}$$
$$T = (N_0 + N_1 - N)/2 - 1 \tag{30}$$

and the subsets of $\mathbb{Z}$,

$$\begin{aligned}\mathcal{P} &= \{0 \le k \le P\} \cup \{N - P \le k \le N - 1\}\\ \mathcal{T} &= \{P + 1 \le k \le P + T\}\\ &\qquad \cup \{N - P - T \le k \le N - P - 1\}\\ \mathcal{S} &= \{N/2 - S \le k \le N/2 + S\}.\end{aligned}$$

Note that these three sets form a partition of $\{0 \le k \le N-1\}$. Then, after some simplification,

$$Y_0(k) = \begin{cases} |H_0(k)|^2\, X(k), & k \in \mathcal{P} \cup \mathcal{T} \\ 0, & k \in \mathcal{S} \end{cases} \tag{31}$$

and

$$Y_1(k) = \begin{cases} 0, & k \in \mathcal{P} \\ |H_1(k)|^2\, X(k), & k \in \mathcal{T} \cup \mathcal{S}. \end{cases} \tag{32}$$

Equations (31) and (32) are analogous to (7) and (8). In deriving (31) we have used $H_0(N/2) = 0$ which is anyway desired of $H_0(k)$, it being a low-pass filter. Likewise, in deriving (32) we have used $H_1(0) = 0$, which is desired of $H_1(k)$, it being a high-pass filter.

Using (31) and (32), the DFT of the N-point output signal $y(n)$ is given by

$$Y(k) = \begin{cases} |H_0(k)|^2\, X(k), & k \in \mathcal{P} \\ (|H_0(k)|^2 + |H_1(k)|^2)\, X(k), & k \in \mathcal{T} \\ |H_1(k)|^2\, X(k), & k \in \mathcal{S}. \end{cases}$$

Perfect reconstruction requires that $Y(k) = X(k)$ for $0 \le k \le N-1$. Therefore, for perfect reconstruction, the filters should satisfy

$$H_0(k) = 1, \quad H_1(k) = 0, \qquad k \in \mathcal{P} \tag{33}$$
$$|H_0(k)|^2 + |H_1(k)|^2 = 1, \qquad k \in \mathcal{T} \tag{34}$$
$$H_0(k) = 0, \quad H_1(k) = 1, \qquad k \in \mathcal{S}. \tag{35}$$

Hence, from (33), the set of indices $\mathcal{P}$ constitutes the pass-band of $H_0(k)$ and stop-band of $H_1(k)$. This is analogous to the interval $\{|\omega| \le (1-\beta)\pi\}$ in (9) being the pass-band of $H_0(\omega)$ and stop-band of $H_1(\omega)$ in Sec. III. Likewise, the set of indices $\mathcal{S}$ constitutes the stop-band of $H_0(k)$ and pass-band of $H_1(k)$. And the set of indices $\mathcal{T}$ constitutes the transition-bands.

Therefore, the pass-band of $H_0(k)$ consists of $|\mathcal{P}| = 2P+1$ DFT bins, the stop-band of $H_0(k)$ consists of $|\mathcal{S}| = 2S+1$ DFT bins, and each transition-band consists of T DFT bins. Note that $2P+2S+2T+2 = N$, the length of the input signal $x(n)$. For example, as illustrated in Fig. 17b, when $N = 24$, $N_0 = 20$, and $N_1 = 16$, then $P = 4$, $S = 2$, $T + 5$, so the pass-band of $H_0(k)$ consists of 9 DFT bins, the stop-band of $H_0(k)$ consists of 5 bins, and each transition band consists of 5 bins.

The transition bands of $H_0(k)$ and $H_1(k)$ can be constructed as in Sec. 7; in this case using any T-point sequence, $\theta(k)$, satisfying

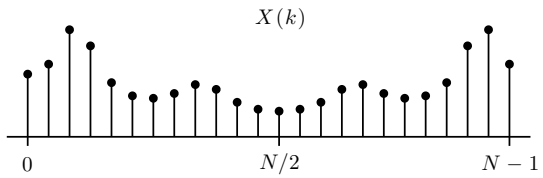
$$\theta^2(k) + \theta^2(T + 1 - k) = 1, \quad 1 \le k \le T.$$

With such a sequence, the transition bands of the filters can be written as

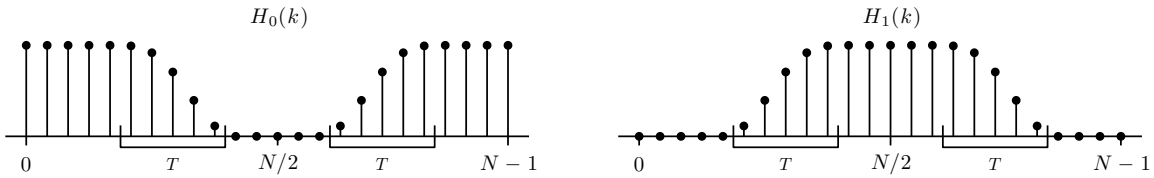
$$\begin{aligned} H_0(P + k) &= \theta(k),\\ H_0(N - P - k) &= \theta(k),\\ H_1(P + k) &= \theta(T + 1 - k),\\ H_1(N - P - k) &= \theta(T + 1 - k), \end{aligned} \tag{36}$$

for $1 \le k \le T$. Using the Daubechies filter with two-vanishing moments, as in (17), we obtain,

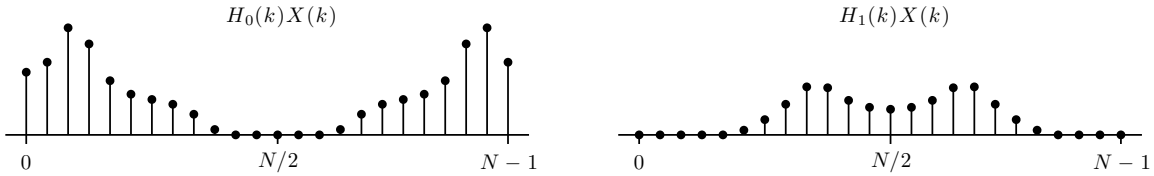
$$\theta(k) = \frac{1}{2}\Big(1 + \cos\Big(\frac{k\,\pi}{T+1}\Big)\Big)\sqrt{2 - \cos\Big(\frac{k\,\pi}{T+1}\Big)},$$



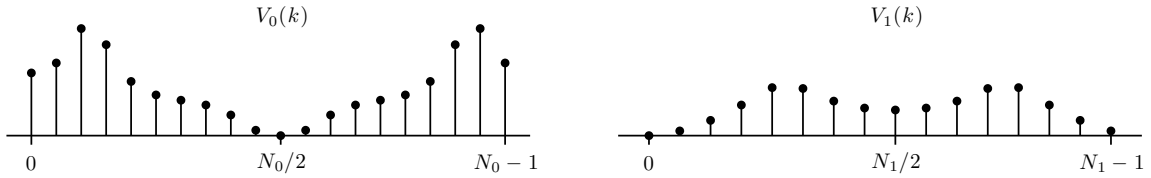
(a) DFT of input signal, $X(k)$.



(b) Filters $H_0(k)$ and $H_1(k)$. The transitions-bands are indicated by 'T'.



(c) DFT of input signal after filtering.



(d) DFT after scaling.

Fig. 17. Behavior of the two-channel analysis filter bank for finite-length signals, illustrated using $N = 24$, $N_0 = 20$, $N_1 = 16$.

for $1 \leq k \leq T$. Given N, N_0, and N_1 (all even), the filters $H_0(k)$ and $H_1(k)$ given by (33), (35), and (36), are a perfect reconstruction pair of filters for the two-channel filter bank illustrated in Fig. 16.

Figure 17 illustrates the behavior of the filter bank. Figure 17a illustrates the N-point DFT, $X(k)$, of an N-point signal. Figure 17b illustrates the low-pass and high-pass frequency responses. Figure 17c illustrates the DFT sequences obtained by filtering; and Fig. 17d illustrates the DFT sequences obtained by subsequent low-pass and high-pass scaling. Note that $V_0(k)$ and $V_1(k)$ occupy the full (discrete) frequency band, for the exception of $V_0(N_0/2) = 0$ and $V_1(0) = 0$.

Analysis filter bank: The process of obtaining the DFT sequences $V_0(k)$ and $V_1(k)$ from the DFT sequence $X(k)$, illustrated in Fig. 17, can be simplified. Specifically, sequences $V_0(k)$ and $V_1(k)$ can be obtained directly from $X(k)$ using the following equations. It is assumed, as above, that N, N_0 and N_1 are even integers, and that the filters $H_0(k)$ and $H_1(k)$ are as specified above. The integers S, P, and T are defined in (29), (28) and (30). The N_0-point sequence $V_0(k)$, $0 \leq k \leq N_0 - 1$, is given by

$$\begin{aligned}
&V_0(0) = X(0),\\
&V_0(p) = X(p),\\
&V_0(P+t) = \theta(t)\, X(P+t),\\
&V_0(N_0/2) = 0,\\
&V_0(N_0 - P - t) = \theta(t)\, X(N-P-t),\\
&V_0(N_0 - p) = X(N-p),
\end{aligned}$$

for $p = 1, \ldots, P$ and $t = 1, \ldots, T$.

The N_1-point sequence $V_1(k)$, $0 \leq k \leq N_1 - 1$, is given by

$$\begin{aligned}
&V_1(0) = 0\\
&V_1(t) = \theta(T+1-t)\, X(P+t),\\
&V_1(T+s) = X(P+T+s),\\
&V_1(N_1/2) = X(N/2)\\
&V_1(N_1 - T - s) = X(N-P-T-s),\\
&V_1(N_1 - t) = \theta(T+1-t)\, X(N-P-t),
\end{aligned}$$

for $t = 1, \ldots, T$ and $s = 1, \ldots, S$.

The N_0-point signal $v_0(n)$ is then obtained as the N_0-point inverse DFT of $V_0(k)$. The N_1-point signal $v_1(n)$ is obtained likewise.

function AFB($\mathbf{x}, N_0, N_1$)
require: length($\mathbf{x}$) even
require: N_0, N_1 even, $N_0 + N_1 > \text{length}(\mathbf{x})$
output: $\mathbf{V}_0, \mathbf{V}_1$ (lengths N_0, N_1)
$N = \text{length}(\mathbf{x})$
$P = (N - N_1)/2$
$T = (N_0 + N_1 - N)/2 - 1$
$S = (N - N_0)/2$
for $1 \le k \le T$ **do**
$\theta(k) = 0.5\,(1 + \cos(k\pi/(T+1)))\,\text{sqrt}(2 - \cos(k\pi/(T+1)))$
▷ low-pass subband:
$V_0(0) = X(0)$
for $1 \le k \le P$ **do**
$V_0(k) = X(k)$
$V_0(N_0 - k) = X(N - k)$
for $1 \le k \le T$ **do**
$V_0(P + k) = X(P + k)\,\theta(k)$
$V_0(N_0 - P - k) = X(N - P - k)\,\theta(k)$
$V_0(N_0/2) = 0$
▷ high-pass subband:
$V_1(0) = 0$
for $1 \le k \le T$ **do**
$V_1(k) = X(P + k)\,\theta(T + 1 - k)$
$V_1(N_1 - k) = X(N - P - k)\,\theta(T + 1 - k)$
for $1 \le k \le S$ **do**
$V_1(T + k) = X(P + T + k)$
$V_1(N_1 - T - k) = X(N - P - T - k)$
$V_1(N_1/2) = X(N/2)$

Fig. 18. Pseudocode for analysis filter bank.

Synthesis filter bank: With reference to Fig. 16, the process of obtaining the DFT sequences $Y_0(k)$ and $Y_1(k)$ from the DFT sequences $V_0(k)$ and $V_1(k)$ can also be simplified. The integers N, N_0, N_1, S, P, and T are defined as above. The N-point sequences $Y_0(k)$ and $Y_1(k)$ can be obtained directly from $V_0(k)$ and $V_1(k)$ by

$$
\begin{aligned}
&Y_0(0) = V_0(0)\\
&Y_0(p) = V_0(p),\\
&Y_0(P + t) = \theta(t)\,V_0(P + t),\\
&Y_0(P + T + s) = 0,\\
&Y_0(N/2) = 0\\
&Y_0(N - P - T - s) = 0,\\
&Y_0(N - P - t) = \theta(t)\,V_0(N_0 - P - t),\\
&Y_0(N - t) = V_0(N_0 - t),
\end{aligned}
$$

and

$$
\begin{aligned}
&Y_1(0) = 0\\
&Y_1(p) = 0,\\
&Y_1(P + t) = \theta(T + 1 - t)\,V_1(t),\\
&Y_1(P + T + s) = V_1(T + s),\\
&Y_1(N/2) = V_1(N_1/2)\\
&Y_1(N - P - T - s) = V_1(N_1 - T - s),\\
&Y_1(N - P - t) = \theta(T + 1 - t)\,V_1(N_1 - t),\\
&Y_1(N - p) = 0,
\end{aligned}
$$

for $p = 1, \ldots, P$, $t = 1, \ldots, T$ and $s = 1, \ldots, S$.

function SFB($\mathbf{V}_0, \mathbf{V}_1, N$)
require: $\mathbf{V}_0, \mathbf{V}_1$ as in AFB, N even
output: $\mathbf{Y}$ (length N)
$N_0 = \text{length}(\mathbf{V}_0)$
$N_1 = \text{length}(\mathbf{V}_1)$
$P = (N - N_1)/2$
$T = (N_0 + N_1 - N)/2 - 1$
$S = (N - N_0)/2$
for $1 \le k \le T$ **do**
$\theta(k) = 0.5\,(1 + \cos(k\pi/(T+1)))\,\text{sqrt}(2 - \cos(k\pi/(T+1)))$
▷ low-pass subband:
$Y_0(0) = V_0(0)$
for $1 \le k \le P$ **do**
$Y_0(k) = V_0(k)$
$Y_0(N - k) = V_0(N_0 - k)$
for $1 \le k \le T$ **do**
$Y_0(P + k) = V_0(P + k)\,\theta(k)$
$Y_0(N - P - k) = V_0(N_0 - P - k)\,\theta(k)$
for $1 \le k \le S$ **do**
$Y_0(P + T + k) = 0$
$Y_0(N - P - T - k) = 0$
$Y_0(N/2) = 0$
▷ high-pass subband:
$Y_1(0) = 0$
for $1 \le k \le P$ **do**
$Y_1(k) = 0$
$Y_1(N - k) = 0$
for $1 \le k \le T$ **do**
$Y_1(P + k) = V_1(k)\,\theta(T + 1 - k)$
$Y_1(N - P - k) = V_1(N_1 - k)\,\theta(T + 1 - k)$
for $1 \le k \le S$ **do**
$Y_1(P + T + k) = V_1(T + k)$
$Y_1(N - P - T - k) = V_1(N_1 - T - k)$
$Y_1(N/2) = V_1(N_1/2)$
for $0 \le k \le N - 1$ **do**
$Y(k) = Y_0(k) + Y_1(k)$ ▷ sum subbands

Fig. 19. Pseudocode for synthesis filter bank.

Setting $Y(k) = Y_0(k) + Y_1(k)$ for $0 \le k \le N - 1$, the reconstructed N-point signal $y(n)$ is obtained as the N-point inverse DFT of $Y(k)$.

Pseudocode is given in Figs. 18 and 19. The function AFB (for 'analysis filter bank') takes as input the N-point DFT sequence $X(k)$ and the parameters N_0 and N_1. This function returns the DFT sequences $V_0(k)$ and $V_1(k)$ of lengths N_0 and N_1 respectively. The function SFB (for 'synthesis filter bank') reverses the procedure. Note that neither AFB nor SFB themselves involve computing the DFT. The functions AFB and SFB will be called by the pseudocode for the tunable-Q wavelet transform below. The computational cost of AFB and SFB is quite low. Excluding the computation of $\theta(k)$ which could be precomputed, AFB requires $4T \approx 2(N_0 + N_1 - N) \approx 2(\alpha + \beta - 1)N = 2\beta(1 - 1/r)N < 2\beta N$ multiplications, where we have used the definition of α in (26). The function SFB requires the same number of multiplications but also an additional N additions (or, excluding additions involving zero, $2T$ additions).

D. Wavelet transform

The tunable-Q wavelet transform (TQWT) of a finite-length signal is implemented by repeatedly applying the filter bank of Fig. 16 to its low-pass channel, as illustrated in Fig. 9. Note that in this case, the parameters N, N_0 and N_1 must be specified at each level. This is in contrast with the wavelet transform of Sec. IV wherein α and β can be the same for all levels. We use the notation $N^{(j)}$, $N_0^{(j)}$ and $N_1^{(j)}$ to designate the level-dependent parameters, where $1 \le j \le J$ with J being the number of levels. The parameter $N^{(j)}$ denotes the length of the input signal to the level j filter bank, and $N_0^{(j)}$ and $N_1^{(j)}$ denote the lengths of the subband signals produced by the level j filter bank. At the first level, $N^{(1)} = N$, the length of the input signal $x(n)$. As the low-pass subband produced by level j, denoted $\mathbf{c}^{(j)}$, serves as the input signal to level $j+1$, we have $N^{(j)} = N_0^{(j-1)}$ for $2 \le j \le J$. As the high-pass subbands constitute the wavelet coefficients, the wavelet subband $\mathbf{w}^{(j)}$ is of length $N_1^{(j)}$.

In order that this DFT-based tunable-Q wavelet transform emulate the behavior (Q-factor, etc) of the wavelet transform described in Sec. IV, we should set $N_0^{(j)} \approx \alpha^j N$ and $N_1^{(j)} \approx \alpha^{j-1}\beta N$. As in Sec. V-C, we round these values to the nearest even integers, setting,

$$N_0^{(j)} = 2\,\mathrm{round}\Big(\frac{\alpha^j}{2}N\Big), \tag{37}$$

$$N_1^{(j)} = 2\,\mathrm{round}\Big(\frac{\beta\alpha^{j-1}}{2}N\Big), \tag{38}$$

for $1 \le j \le J$. And, as noted above

$$N^{(1)} = N, \qquad N^{(j)} = N_0^{(j-1)}, \quad 2 \le j \le J. \tag{39}$$

Using (37), (38) and (39), the wavelet transform is fully determined by the length N of the input signal, parameters α and β, and the number of levels J.

The tunable-Q wavelet transform for finite-length N-point input signal $\mathbf{x}$ is implemented as:

$$\mathbf{C}^{(0)} \leftarrow \mathrm{DFT}\{\mathbf{x}\} \tag{40}$$

$$\{\mathbf{C}^{(j)}, \mathbf{W}^{(j)}\} \leftarrow \textsc{afb}(\mathbf{C}^{(j-1)}, N_0^{(j)}, N_1^{(j)}), \tag{41}$$

$$\mathbf{w}^{(j)} \leftarrow \mathrm{DFT}^{-1}\{\mathbf{W}^{(j)}\}, \tag{42}$$

$$\mathbf{c}^{(J)} \leftarrow \mathrm{DFT}^{-1}\{\mathbf{C}^{(J)}\} \tag{43}$$

for $1 \le j \le J$, where AFB denotes the analysis filter bank. Here $\mathbf{c}^{(j)}$ and $\mathbf{w}^{(j)}$ are the low-pass and high-pass subband signals produced by the level j filter bank. The inverse transform is implemented as:

$$\mathbf{C}^{(J)} \leftarrow \mathrm{DFT}\{\mathbf{c}^{(J)}\} \tag{44}$$

$$\mathbf{W}^{(j)} \leftarrow \mathrm{DFT}\{\mathbf{w}^{(j)}\}, \tag{45}$$

$$\mathbf{C}^{(j-1)} \leftarrow \textsc{sfb}(\mathbf{C}^{(j)}, \mathbf{W}^{(j)}, N^{(j)}), \tag{46}$$

$$\mathbf{y} \leftarrow \mathrm{DFT}^{-1}\{\mathbf{C}^{(0)}\} \tag{47}$$

for $1 \le j \le J$, where SFB denotes the synthesis filter bank.

For some signal processing algorithms, it is useful that the transform have the energy preservation property (Parseval's theorem). This is easily achieved by using the unitary DFT (2) in place of the standard DFT; the only difference being the normalization of the DFT. Hence, using the unitary DFT, the wavelet coefficients satisfy,

$$\sum_{n=0}^{N-1}|x(n)|^2 = \sum_{j=1}^{J}\sum_{n=0}^{N_1^{(j)}-1}|w^{(j)}(n)|^2 + \sum_{n=0}^{N_0^{(J)}-1}|c^{(J)}(n)|^2. \tag{48}$$

Pseudocode for the N-point tunable-Q wavelet transform and its inverse is given in Figs. 20 and 21. In the pseudocode, the unitary DFT is used.

function TQWT($\mathbf{x}, Q, r, J$)
  require: length($\mathbf{x}$) even, $Q \ge 1$, $r > 1$, $J \in \mathbb{N}$
  output: $\mathbf{c}$ and $\mathbf{w}^{(j)}$, $1 \le j \le J$
  $\beta = 2/(Q+1)$
  $\alpha = 1 - \beta/r$
  $\mathbf{X} = \mathrm{uDFT}(\mathbf{x})$
  $N = \mathrm{length}(\mathbf{x})$
  for $j = 1$ **to** J **do**
    $N_0 = 2\,\mathrm{round}(\alpha^j N/2)$
    $N_1 = 2\,\mathrm{round}(\beta\,\alpha^{j-1} N/2)$
    $(\mathbf{X}, \mathbf{W}) = \textsc{afb}(\mathbf{X}, N_0, N_1)$
    $\mathbf{w}^{(j)} = \mathrm{uDFT}^{-1}(\mathbf{W})$
  $\mathbf{c} = \mathrm{uDFT}^{-1}(\mathbf{X})$

Fig. 20. Pseudocode for tunable Q-factor wavelet transform.

function ITQWT($\mathbf{c}, \mathbf{w}^{(j)}, Q, r, J$)
  require: Q, r, J as in TQWT
  output: $\mathbf{y}$
  $\beta = 2/(Q+1)$
  $\alpha = 1 - \beta/r$
  $\mathbf{Y} = \mathrm{uDFT}(\mathbf{c})$
  for $j = J$ **down to** 1 **do**
    $\mathbf{W} = \mathrm{uDFT}(\mathbf{w}^{(j)})$
    $M = 2\,\mathrm{round}(\alpha^{j-1} N/2)$
    $\mathbf{Y} = \textsc{sfb}(\mathbf{Y}, \mathbf{W}, M)$
  $\mathbf{y} = \mathrm{uDFT}^{-1}(\mathbf{Y})$

Fig. 21. Pseudocode for inverse TQWT.

Recall that for the TQWT described in Sec. IV, subband j has a sampling rate of $\beta\alpha^{j-1} f_s$ where f_s is the input signal sampling rate. For the finite-length TQWT described here, $\beta\alpha^{j-1}$ is approximated by $N_1^{(j)}/N$. The approximation is unavoidable due to its being a transform for finite-length signals; but for input signals of reasonable length, the approximation should be more than sufficiently accurate. In comparison, wavelet transforms based on filter banks with time-domain fractional rate changers (using up-sampling and down-sampling) generally have less flexibility in the scaling parameters. Therefore, although the Q-factor can not be continuously varied, the finite-length TQWT provides a good approximation for a wide range of Q-factors, as described following (27).

Maximum number of levels: The number of levels is limited by the length of the signal. After a certain number of levels (dependent on α and β) the signal will be too short to further

decompose into low-pass and high-pass subbands. Specifically, the maximum number of levels J is the maximum integer J such that $N_0^{(J)} \geq 2$ and $N_1^{(J)} \geq 2$ where $N_0^{(j)}$ and $N_1^{(j)}$ are give by (37) and (38). (Simply, the subband signals produced by the level-J filter bank must be of positive length.)

However, with this many levels, the resulting coefficients may be difficult to interpret. To avoid that, the number of levels J should generally be limited so that the level-J wavelet is not longer than the signal under analysis. Based on this criterion, the maximum number of levels can be found as follows.

The approximate duration of the wavelets in samples can be expressed in terms of the bandwidth BW of the equivalent filter $H_1^{(j)}(\omega)$. From (24), the bandwidth at level j is $0.5\beta\,\alpha^{j-1}\,\pi$ radians/sample. Equivalently, the bandwidth is $0.25\,\beta\,\alpha^{j-1}$ cycles/sample. Approximating the duration of the level j wavelet as $2/\mathrm{BW}$ gives a duration of about $8/(\alpha^{j-1}\beta)$ samples. Hence, asking that the duration of the level j wavelet be less than the length of the input signal gives the condition

$$\frac{8}{\alpha^{j-1}\beta} \leq N.$$

Therefore, it is reasonable to set the maximum number of levels to

$$J_{\max} = \left\lfloor \frac{\log(\beta N/8)}{\log(1/\alpha)} \right\rfloor.$$

Computational cost: Most of the computational cost of the wavelet transform is due to the DFT computations[3]. The transform requires a total of $J+2$ DFT computations: one DFT for the input signal, one DFT for each of the J wavelet (high-pass) subbands $\mathbf{w}^{(j)}$, and one DFT for the final low-pass subband $\mathbf{c}^{(J)}$. The inverse transform requires the same DFT computations. Hence, the required DFTs are of lengths: N, $N_1^{(j)}$ for $1 \leq j \leq J$, and $N_0^{(J)}$.

The computational cost of computing the DFT of an N-point sequence is on the order of $N \log N$. Using the chirp z-transform in conjunction with a radix-2 FFT (Bluestein's algorithm) gives a practical method to achieve $N \log N$ cost, even for signals that are not a power of two in length [26]. However, the DFT of an N-point sequence can be computed most efficiently when N is in fact a power of two, $N = 2^k$, using a radix-2 FFT. Unfortunately, the TQWT described in the preceding paragraphs require DFTs of lengths $N_1^{(j)}$ which are not powers of two, even when the length of the input signal $\mathbf{x}$ is a power of two. Therefore, in order to develop a more computationally efficient implementation, the next section describes a version of the transform for which all DFTs are powers of two in length.

E. Radix-2 Tunable-Q Wavelet Transform

In order to minimize the computational cost of the tunable-Q wavelet transform, this section describes how the transform can be modified so that all the DFTs are powers of two in length. Then only radix-2 FFTs are required for the implementation of this version of the transform, which we call the radix-2 TQWT. We will denote the next power of two by $\mathrm{next}()$, defined as

$$\mathrm{next}(k) := 2^{\mathrm{ceil}(\log_2(k))}, \quad k \in \mathbb{Z}$$

where $\mathrm{ceil}(a)$ rounds a to the nearest integer greater than or equal to a.

It can be assumed that the length of the input signal $\mathbf{x}$ is a power of two. (If it is not, then it can be zero-padded.) Hence, the DFT in (40) is a power of two in length.

In order that the DFTs in (42) be a power of two in length, prior to computing the (inverse) DFT we apply low-pass scaling as defined in Sec. V-A to the sequence $\mathbf{W}^{(j)}$ to increase its length to the next power of two. Namely, we increase its length to $\mathrm{next}(N_1^{(j)})$. As a result, the number of samples in the wavelet subband $\mathbf{w}^{(j)}$ is increased also to that length. Note that applying low-pass scaling essentially performs a rate-increase in the time-domain. It does not effect the shape of the equivalent filter $H_1^{(j)}(\omega)$ or its center frequency.

Similarly, in order that the DFT in (43) be a power of two in length, we also apply low-pass scaling prior to the (inverse) DFT, to increase the length of $\mathbf{C}^{(J)}$ to the next power of two, namely to $\mathrm{next}(N_0^{(J)})$.

In summary, let the input signal $\mathbf{x}$ be of length N, a power of two. Then the radix-2 tunable-Q wavelet transform is implemented by the algorithm:

$$\mathbf{C}^{(0)} \leftarrow \mathrm{DFT}\{\mathbf{x}\} \tag{49}$$

$$\{\mathbf{C}^{(j)}, \mathbf{W}^{(j)}\} \leftarrow \textsc{afb}(\mathbf{C}^{(j-1)}, N_0^{(j)}, N_1^{(j)}), \tag{50}$$

$$\mathbf{W}^{(j)} \leftarrow \mathrm{lps}(\mathbf{W}^{(j)}, \mathrm{next}(N_1^{(j)})), \tag{51}$$

$$\mathbf{w}^{(j)} \leftarrow \mathrm{DFT}^{-1}\{\mathbf{W}^{(j)}\}, \tag{52}$$

$$\mathbf{C}^{(J)} \leftarrow \mathrm{lps}(\mathbf{C}^{(J)}, \mathrm{next}(N_0^{(J)})) \tag{53}$$

$$\mathbf{c}^{(J)} \leftarrow \mathrm{DFT}^{-1}\{\mathbf{C}^{(J)}\} \tag{54}$$

for $1 \leq j \leq J$, where lps denotes low-pass scaling, and where $N_0^{(j)}$ and $N_1^{(J)}$ are as defined in (37) and (38). Note that converting a finite-length signal to a longer signal by low-pass scaling is perfectly invertible. Therefore, the inverse transform is implemented as:

$$\mathbf{C}^{(J)} \leftarrow \mathrm{DFT}\{\mathbf{c}^{(J)}\} \tag{55}$$

$$\mathbf{C}^{(J)} \leftarrow \mathrm{lps}(\mathbf{C}^{(J)}, N_0^{(J)}) \tag{56}$$

$$\mathbf{W}^{(j)} \leftarrow \mathrm{DFT}\{\mathbf{w}^{(j)}\}, \tag{57}$$

$$\mathbf{W}^{(j)} \leftarrow \mathrm{lps}(\mathbf{W}^{(j)}, N_1^{(j)}), \tag{58}$$

$$\mathbf{C}^{(j-1)} \leftarrow \textsc{sfb}(\mathbf{C}^{(j)}, \mathbf{W}^{(j)}, N^{(j)}), \tag{59}$$

$$\mathbf{y} \leftarrow \mathrm{DFT}^{-1}\{\mathbf{C}^{(0)}\} \tag{60}$$

for $1 \leq j \leq J$. In this implementation all DFTs are power of two in length, so that only radix-2 FFT algorithms are needed.

As in Sec. V-D, when the unitary DFT is used in place of the DFT, the transform conserves the signal energy (Parseval's theorem). Hence (48) is satisfied, except that the upper limits will be $\mathrm{next}(N_i^{(j)})$ to account for the subbands being longer.

Note that, due to the lengthening of the subbands to the next power of two, the formula for the redundancy factor r

[3]With regard to the computational cost, we do not differentiate the DFT and inverse DFT because they have essentially the same computational cost.

defined in (20) underestimates the actual redundancy factor for the radix-2 TQWT. The actual redundancy factor, given by

$$\frac{1}{N} \cdot \Big(N_0^{(J)} + \sum_{j=1}^{J} N_1^{(j)} \Big),$$

will not be more than twice r.

Computational cost: The computational cost of the radix-2 TQWT is due to the DFTs and to the operations within AFB and SFB .

The radix-2 TQWT uses radix-2 FFTs of lengths: N, $\mathrm{next}(N_1^{(j)})$ for $1 \le j \le J$, and $\mathrm{next}(N_0^{(J)})$. Consider the computational cost of the FFTs of length $\mathrm{next}(N_1^{(j)})$ in (52). An M-point radix-2 FFT has a computational cost of about $0.5M\log_2 M$ complex multiplications and $M\log_2 M$ complex additions. Therefore, denoting $M^{(j)} := \mathrm{next}(N_1^{(j)})$, the computational costs of these FFTs sum to

$$C = \sum_{j=1}^{J} 1.5\, M^{(j)} \log_2(M^{(j)})$$

complex multiplications and additions. Noting that

$$M^{(j)} = \mathrm{next}(N_1^{(j)}) \le 2N_1^{(j)} \approx 2\beta\alpha^{j-1}N,$$

where we have used (38), the computational cost C can be upper bounded as

$$C \le 1.5 \sum_{j=1}^{J} 2\beta\alpha^{j-1} N \log_2(2\beta\alpha^{j-1}N) \tag{61}$$

$$\le 3\beta N \sum_{j=1}^{J} \alpha^{j-1} \log_2(2\beta N) \quad [\text{using } \alpha < 1] \tag{62}$$

$$\le 3\beta N \log_2(2\beta N) \sum_{j=1}^{\infty} \alpha^{j-1} \quad [\text{using } \alpha > 0] \tag{63}$$

$$= 3\beta N \log_2(2\beta N)\, \frac{1}{1-\alpha} \tag{64}$$

$$= 3rN \log_2(2\beta N) \quad [\text{using (26)}] \tag{65}$$

$$\le 3rN \log_2(2N) \quad [\text{using } \beta \le 1] \tag{66}$$

Hence, the computational cost of these FFTs is $O(rN\log_2 N)$.

Assuming the input signal $\mathbf{x}$ is a power of two in length, the radix-2 FFT in (49) incurs an additional computational cost of $1.5N\log_2 N$. The radix-2 FFT in (54) to obtain the final low-pass subband costs less than $1.5N\log_2 N$ because it is a shorter signal (usually $\mathbf{c}^{(J)}$ is much shorter than input signal $\mathbf{x}$, so the FFT in (54) will be negligible).

The computational cost of AFB in (50) is $O(N^{(j)})$, as noted in the last paragraph of Sec. V-C. Following the same procedure as above, the total computational cost of AFB over all levels can be upper bounded by $2rN$. This is less than the computational cost of the FFTs. The same is true for SFB .

Therefore, the total computational cost of the radix-2 TQWT is $O(rN\log_2 N)$ where N is the length of the input signal, and r is the redundancy factor. Note that the cost is linear in r, which is as low as can be expected. The cost as a function of N is also as low as can be expected, given that the implementation is based on the DFT. Additionally, we note that run-time in practice also depends on implementation details such as indexing, data shuffling, and memory access issues.

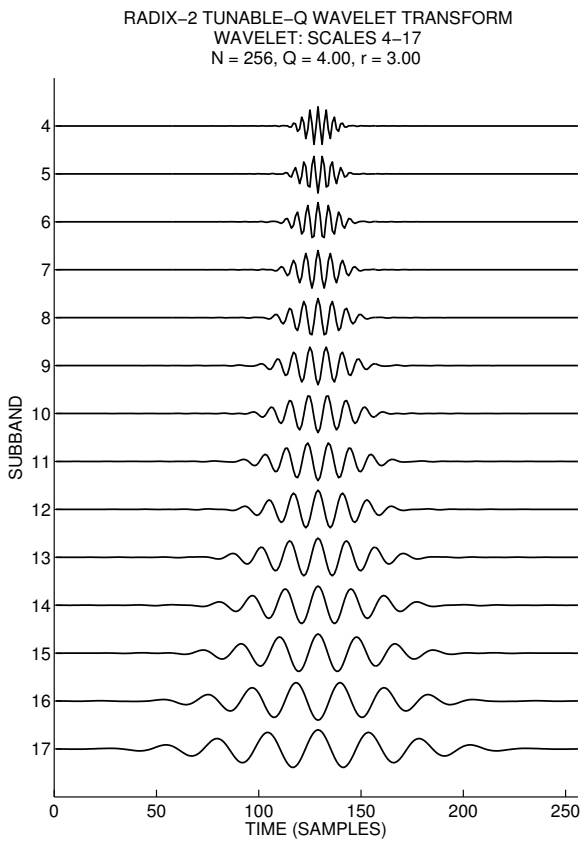


Fig. 22. The radix-2 tunable-Q discrete wavelet transform. The numerically computed wavelet is illustrated for scales 4-17.

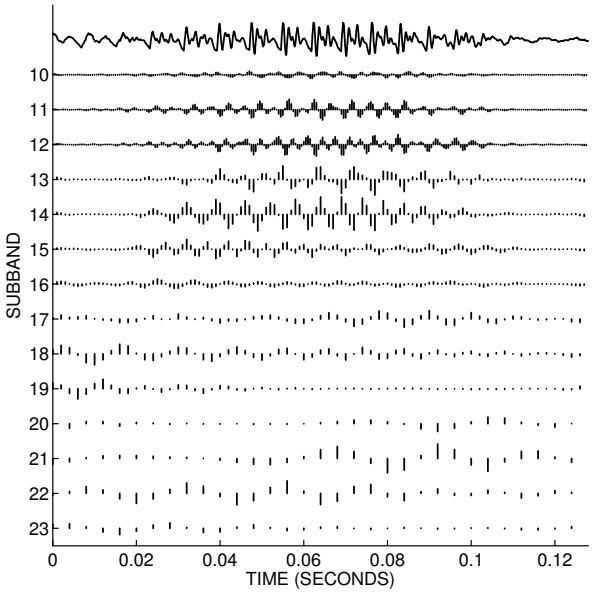
F. Example 1

To illustrate the radix-2 tunable-Q wavelet transform, suppose that the Q-factor is specified to be 4 and that the oversampling-rate is specified to be 3. Then using $Q = 4$ and $r = 3$ in (26), we have $\alpha = 0.867$ and $\beta = 0.4$. For an input signal of length $N = 256$, if the transform is computed for $J = 17$ levels, then the wavelet subbands $\mathbf{w}^{(j)}$ are of lengths

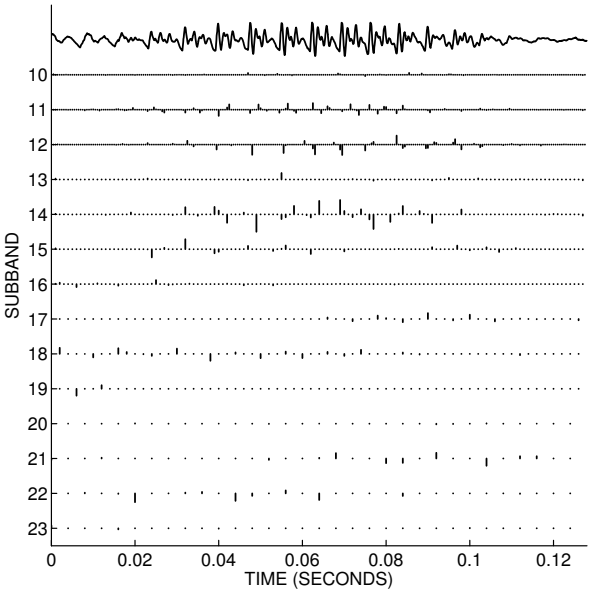
$$\mathrm{length}(\mathbf{w}^{(j)}) = \begin{cases} 128 & j = 1:4 \\ 64 & j = 5:8 \\ 32 & j = 9:13 \\ 16 & j = 14:17 \end{cases}$$

and the length of the final low-pass subband is $\mathrm{length}(\mathbf{c}^{(J)}) = 32$. The total number of wavelet coefficients among the 18 subbands including $\mathbf{c}^{(J)}$ is 1024, so the actual redundancy factor is $4 = 1024/N$ instead of 3 as specified. This is due to the lengthening of the subbands so as to use radix-2 FFTs. Note that the actual redundancy will not be more than twice r, because the subbands will not be lengthened by more than a factor of two.

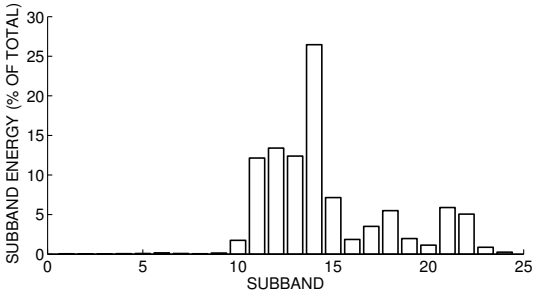
The discrete wavelets (analysis/synthesis functions) can be computed by setting all wavelet coefficients equal to zero except for one coefficient in $\mathbf{w}^{(j)}$ which is set to unity.



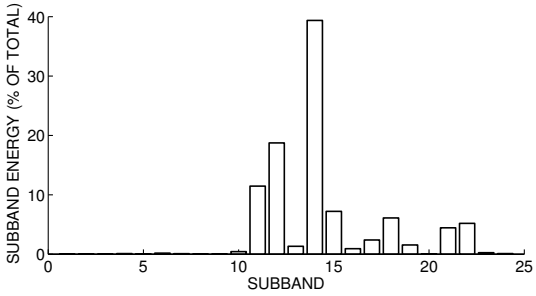
(a) Subbands of speech waveform.



(b) Subbands with sparsification.



(c) Distribution of signal energy.



(d) Distribution of energy with sparsification.

Fig. 23. Wavelet transform of a speech waveform with and without sparsification. (a) Subbands 10-23. (b) Subbands of sparse representation. (c,d) Distribution of signal energy across subbands corresponding to (a,b) respectively. The sparse representation is obtained via ℓ_1 minimization. The wavelet transform parameters are $Q = 3$ and $r = 3$.

Applying the inverse wavelet transform yields the level j wavelet. The wavelets computed in this way are illustrated in Fig. 22 for levels $4 \leq j \leq 17$. The discrete wavelets in Fig. 22 are indistinguishable from the wavelets computed likewise using the non-radix-2 TQWT of Sec. V-D. (For this reason, we have not separately illustrated the wavelets of the non-radix-2 version.)

Hence, the difference between the non-radix-2 version described in Sec. V-D and the radix-2 version is essentially the increase in redundancy, not the temporal or spectral characteristics of the transform.

G. Example 2

To illustrate sparse signal representation with the tunable-Q wavelet transform, consider the speech waveform illustrated in Fig. 23. This waveform consists of 2048 samples at 16,000 samples/second. The speech is 'o' in 'often' spoken by an adult male. We use 23 levels of the radix-2 TQWT with parameters $Q = 3$ and $r = 3$. (Hence, $\alpha = 0.833$ and $\beta = 0.5$.) The frequency decomposition is illustrated in Fig. 13c. Applying the transform to the speech waveform gives the wavelet coefficients illustrated in Fig. 23a. Only levels (subbands) 10-23 are shown because subbands 1-9 are negligible. The distribution of the signal's energy among the subbands is illustrated in Fig. 23c. Note that subband 1 corresponds to high frequencies while subband 23 corresponds to low frequencies. Specifically, using (23), the frequency of subband k is $(0.833)^k$ 7200 Hz (but this is not valid for $k = 1$, subband 1 is centered at the Nyquist frequency, 8000 Hz).

To obtain a sparse representation $\mathbf{a}$ of the signal $\mathbf{x}$ using the transform, we solve the basis pursuit problem [12]:

$$\min_{\mathbf{a}} \|\mathbf{a}\|_1 \text{ such that } \Phi\,\mathbf{a} = \mathbf{x} \tag{67}$$

where Φ is the matrix whose columns are the synthesis functions of the transform and $\mathbf{a}$ is the vector of the coefficients, $\mathbf{a} = [\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(J)}, \mathbf{c}^{(J)}]$. For the TQWT, $\Phi\Phi^t = \mathbf{I}$, which can be utilized when solving (67). Matrix multiplication by Φ^t and Φ are implemented using the TQWT and its inverse. Solving (67) using 100 iterations of a variant of SALSA [1] gives the sparse representation illustrated in Fig. 23b.[4]

[4]To account for the energy of the synthesis functions not being constant across subbands, we actually minimize the weighted ℓ_1-norm $\left(\sum_{j=1}^{J} \lambda_j \|\mathbf{w}^{(j)}\|_1\right) + \gamma\|\mathbf{c}^{(J)}\|_1$ where $\lambda_j = \|\mathbf{h}_1^{(j)}\|_2$ and $\gamma = \|\mathbf{h}_0^{(J)}\|_2$ are the 2-norms of the synthesis functions.

Note that for the sparse representation, the distribution of the signal energy, illustrated in Fig. 23d, is somewhat different than that shown in Fig. 23b. In particular, the relative energy in subband 13 is diminished. The sparse representation has less frequency-leakage than the coefficients computed directly by $\Phi^t \mathbf{x}$ (shown in Fig. 23a). While we have used ℓ_1-norm minimization to obtain a sparse representation here, we note there is ample evidence that ℓ_0 minimization is often more effective [25].

Sparse representations are used in numerous recent signal processing algorithms [3]. Sparse signal representations with constant-Q (wavelet) transforms have been used for separating oscillatory and non-oscillatory components of a signal [29]. While the examples in [29] were based on the RADWT [5], the TQWT can be used equally well. An advantage of the TQWT in comparison with the RADWT is its faster implementation using radix-2 FFTs and the ease with which its Q-factor and redundancy can be tuned.

VI. Conclusion

This paper has described a wavelet (constant-Q) transform for which the Q-factor and redundancy (oversampling rate) are easily specified. We denote the transform TQWT (for 'tunable-Q wavelet transform'). The transform is specified by two parameters: Q and r, the transform's Q-factor and redundancy. The transform is inverted by its conjugate transpose, so it satisfies Parseval's theorem (it is a tight frame). The transform is developed specifically for discrete-time signals. One form of the transform (Sec. IV) is applicable to discrete-time signals defined on all of $\mathbb{Z}$. The second form (Sec. V-D) is applicable to discrete-time signals of finite length. The first and second forms are developed using the DTFT and DFT respectively. In addition, we have described a version of the TQWT that allows fast implementation using radix-2 FFTs.[5]

Note that the frequency domain approach taken here (for filter design and implementation) is not an advantage in and of itself, but it is a means to achieve the goals of the TQWT (tunable Q-factor, tunable redundancy, PR, etc). Filters with rational transfer functions are more computationally efficient, but it appears that they may not satisfy the perfect reconstruction conditions of a filter bank with real-valued scaling factors on which the TQWT is based.

Acknowledgment

The author would like to thank Vittoria Bruni and Domenico Vitulano of the Institute for the Application of Calculus, National Council of Research, Rome, Italy, and Julian Stewart of the New York Medical College, Hawthorne, NY, for many useful questions, observations, and comments. The author also thanks Faruk Uysal for implementing the transform in C.

References

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Process.*, 19(9):2345 –2356, September 2010.

[2] P. Auscher. Wavelet bases for $L^2(R)$ with rational dilation factor. In M. B. Ruskai et al., editor, *Wavelets and Their Applications*. Jones and Barlett, Boston, 1992.

[3] R. G. Baraniuk, E. Candes, M. Elad, and Y. Ma, editors. *Proc. IEEE*, 98(6), June 2010. Special issue on Applications of Sparse Representation and Compressive Sensing.

[4] A. Baussard, F. Nicolier, and F. Truchetet. Rational multiresolution analysis and fast wavelet transform: application to wavelet shrinkage denoising. *Signal Processing*, 84(10):1735–1747, October 2004.

[5] I. Bayram and I. W. Selesnick. Frequency-domain design of overcomplete rational-dilation wavelet transforms. *IEEE Trans. Signal Process.*, 57(8):2957–2972, August 2009.

[6] İ. Bayram and I. W. Selesnick. Overcomplete discrete wavelet transforms with rational dilation factors. *Signal Processing*, 57(1):131–145, January 2009.

[7] T. Blu. Iterated filter banks with rational rate changes — connection with discrete wavelet transforms. *IEEE Trans. Signal Process.*, 41(12):3232–3244, December 1993.

[8] T. Blu. A new design algorithm for two-band orthonormal rational filter banks and orthonormal rational wavelets. *IEEE Trans. Signal Process.*, 46(6):1494–1504, June 1998.

[9] T. Blu and M. Unser. The fractional spline wavelet transform: definition and implementation. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, 2000.

[10] J. C. Brown. Calculation of a constant Q spectral transform. *J. Acoust. Soc. Am.*, 89(1):425–434, January 1991.

[11] J. C. Brown and M. S. Puckette. An efficient algorithm for the calculation of a constant Q transform. *J. Acoust. Soc. Am.*, 92(5):2698–2701, November 1992.

[12] S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.

[13] G. F. Choueiter and J. R. Glass. An implementation of rational wavelets and filter design for phonetic classification. *IEEE Trans. on Audio, Speech, and Lang. Proc.*, 15(3):939–948, March 2007.

[14] I. Daubechies. *Ten Lectures On Wavelets*. SIAM, 1992.

[15] T. Gulzow, T. Ludwig, and U. Heute. Spectral-subtraction speech enhancement in multirate systems with and without non-uniform and adaptive bandwidths. *Signal Processing*, 83(8):1613 – 1631, 2003.

[16] B. D. Johnson. Stable filtering schemes with rational dilations. *J. of Fourier Analysis and App.*, 13(5):607–621, 2007.

[17] A. Karmakar, A. Kumar, and R. K. Patney. Design of optimal wavelet packet trees based on auditory perception criterion. *IEEE Signal Processing Letters*, 14(4):240–243, April 2007.

[18] J. Kovačević and M. Vetterli. Perfect reconstruction filter banks with rational sampling factors. *IEEE Trans. Signal Process.*, 41(6):2047–2066, June 1993.

[19] Q. Li. An auditory-based transform for audio signal processing. In *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2009.

[20] S. K. Mitra. *Digital Signal Processing*. McGraw-Hill, 1998.

[21] K. Nayebi, T. P. Barnwell III, and M. J. T. Smith. The design of perfect reconstruction filter banks with rational sampling factors. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, 1991.

[22] G. Pau, B. Pesquet-Popescu, and G. Piella. Modified M-band synthesis filter bank for fractional scalability of images. *IEEE Signal Processing Letters*, 13(6):345–348, June 2006.

[23] S.-C. Pei and M.-P. Kao. A two-channel nonuniform perfect reconstruction filter bank with irrational down-sampling factors. *IEEE Signal Processing Letters*, 12(2):116–119, February 2005.

[24] T. Petersen and S. Boll. Critical band analysis-synthesis. *IEEE Trans. on Acoust., Speech, Signal Proc.*, 31(3):656–663, June 1983.

[25] J. Portilla and L. Mancera. L0-based sparse approximation: two alternative methods and some applications. In *Proceedings of SPIE*, volume 6701 (Wavelets XII), 2007.

[26] R. Rabiner, R. Schafer, and C. Rader. The chirp z-transform algorithm. *IEEE Trans. Audio Electroacoust.*, 17(2):86–92, June 1969.

[27] Y. Ren, M. T. Johnson, and J. Tao. Perceptually motivated wavelet packet transform for bioacoustic signal enhancement. *J. Acoust. Soc. Am.*, 124:316–327, 2008.

[28] C. Schörkhuber and A. Klapuri. Constant-Q transform toolbox for music processing. In *7th Sound and Music Conf.*, July 2010.

[29] I. W. Selesnick. Resonance-based signal decomposition: A new sparsity-enabled signal analysis method. *Signal Processing*, In press, corrected proof available online, 2010.

[30] W. Zhao and R. M. Rao. A discrete-time wavelet transform based on a continuous dilation framework. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 3, March 1999.

[5] Software for the implementation of the TQWT is available on the web at http://taco.poly.edu/selesi/TQWT/

Ivan Selesnick received the BS, MEE, and PhD degrees in Electrical Engineering in 1990, 1991, and 1996 from Rice University, Houston, TX. In 1997, he was a visiting professor at the University of Erlangen-Nurnberg, Germany. He then joined the Department of Electrical and Computer Engineering, Polytechnic Institute of New York University (then Polytechnic University), where he is Associate Professor. His current research interests are in the area of digital signal and image processing, wavelet-based signal processing, and biomedical signal processing.

His dissertation received the Budd Award for Best Engineering Thesis at Rice University and an award from the Rice-TMC chapter of Sigma Xi. He has received an Alexander von Humboldt Award (1997) and a National Science Foundation Career award (1999). He has been an associate editor of the IEEE Trans. on Image Processing, of IEEE Signal Processing Letters, and is currently an associate and area editor of IEEE Trans. on Signal Processing.