EE-597 Class Notes – Sub-Band Coding

Phil Schniter

June 11, 2004

1 Sub-Band Coding

1.1 Introduction and Motivation

- Of the major "signal-processing tools" used in MPEG-style audio coding schemes (see Fig. 1), sub-band coding is the last which we have yet to cover.

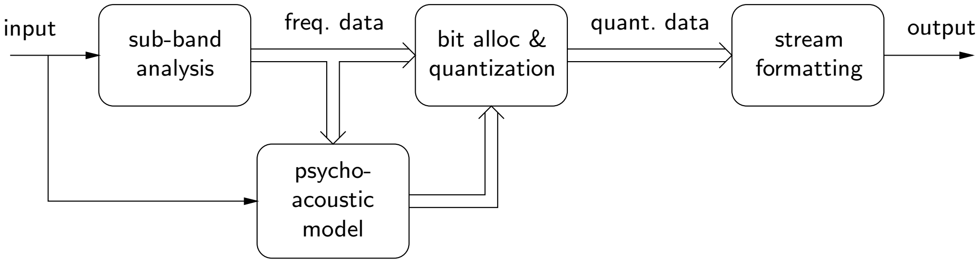


Figure 1: Simplified MPEG-style audio coding system.

- Fig. 2 illustrates a generic subband coder. In short, the input signal is passed through a parallel bank of analysis filters $\{H_i(z)\}$ and the outputs are "downsampled" by a factor of N. Downsampling-by-N is a process which passes every N^{th} sample and ignores the rest, effectively decreasing the data rate by factor N. The downsampled outputs are quantized (using a potentially different number of bits per branch—as in transform coding) for storage or transmission. Downsampling ensures that the number of data samples to store is not any larger than the number of data samples entering the coder; in Fig. 2, N sub-band outputs are generated for every N system inputs.

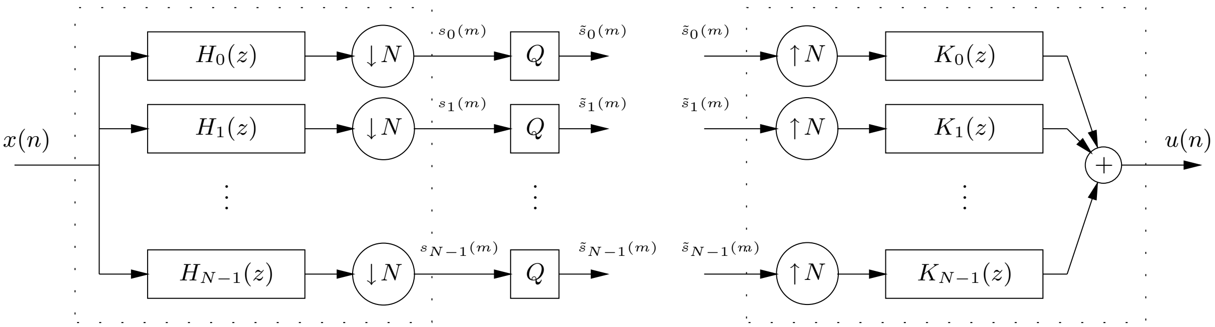


Figure 2: Sub-band coder/decoder with scalar quantization.

- *Relationship to Transform Coding:* Conceptually, sub-band coding (SC) is very similar to transform coding (TC). Like TC, SC analyzes a block of input data and produces a set of linearly transformed outputs, now called "subband outputs." Like TC, these transformed outputs are independently quantized in a way that yields coding gain over straightforward PCM. And like TC, it is possible to derive an optimal bit allocation which minimizes reconstruction error variance for a specified average bit rate.

  In fact, an N-band SC system with length-N filters is equivalent to a TC system with $N \times N$ transformation matrix $\mathbf{T}$: the decimated convolution operation which defines the i^{th} analysis branch of Fig. 2 is identical to an inner product between an N-length input block and $\mathbf{t}_i^t$, the i^{th} row of $\mathbf{T}$. (See Fig. 3.)

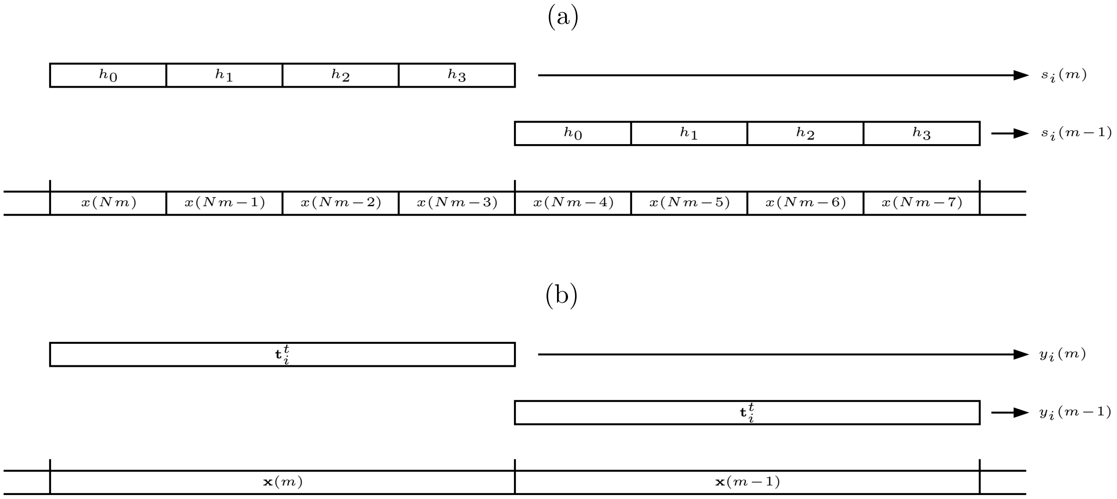


Figure 3: Equivalence between (a) N-band sub-band coding with length-N filters and (b) $N \times N$ transform coding (shown for $N = 4$). Note: impulse response coefficients $\{h_n\}$ correspond to filter $H_i(z)$.

  So what kind of frequency responses characterize the most-commonly used transformation matrices? Lets look at the DFT first. For the i^{th} row, we have

$$|H_i(\omega)| \;=\; \left|\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}in} e^{-j\omega n}\right| \;=\; \left|\sum_{n=0}^{N-1} e^{-j(\omega+\frac{2\pi}{N}i)n}\right| \;=\; \left|\frac{\sin(\frac{N}{2}(\omega+\frac{2\pi i}{N}))}{\sin(\frac{1}{2}(\omega+\frac{2\pi i}{N}))}\right|.$$

  Fig. 4 plots these magnitude responses. Note that the i^{th} DFT row acts as a bandpass filter with center frequency $2\pi i/N$ and stopband attenuation of ≈ 6 dB. Fig. 5 plots the magnitude responses of DCT filters, where we see that they have even less stopband attenuation.

- *Psycho-acoustic Motivations:* We have seen that N-band SC with length-N filters is equivalent to $N \times N$ transform coding. But is transform coding the best technique to use in high quality audio coders? It turns out that *the key to preserving sonic quality under high levels of compression is to shape the reconstruction error so that the ear will not hear it.* When we talk about psychoacoustics later in the course, we'll see that the properties of noise tolerated by the ear/brain are most easily described in the frequency domain. Hence, *bitrate allocation based on psychoacoustic models is most conveniently performed when SC outputs represent signal components in isolated frequency bands.* In other words, instead of allocating fewer bits to sub-band outputs having a smaller effect on reconstruction error variance, we will allocate fewer bits to sub-band outputs having a smaller contribution to *perceived* reconstruction error.

  We have seen that length-N DFT and DCT filters give a $2\pi/N$ bandwidth with no better than 6 dB of stopband attenuation. The SC filters required for high-quality audio coding require much

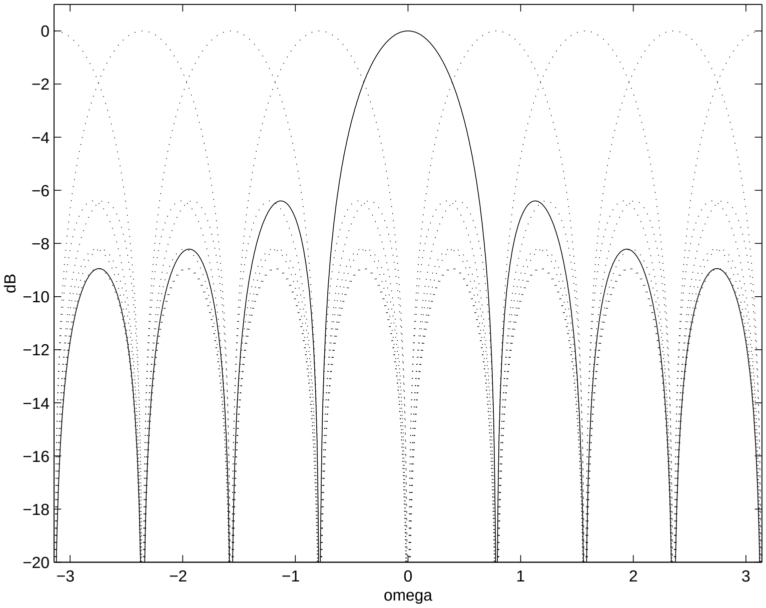


Figure 4: Magnitude responses of DFT basis vectors for $N = 8$.

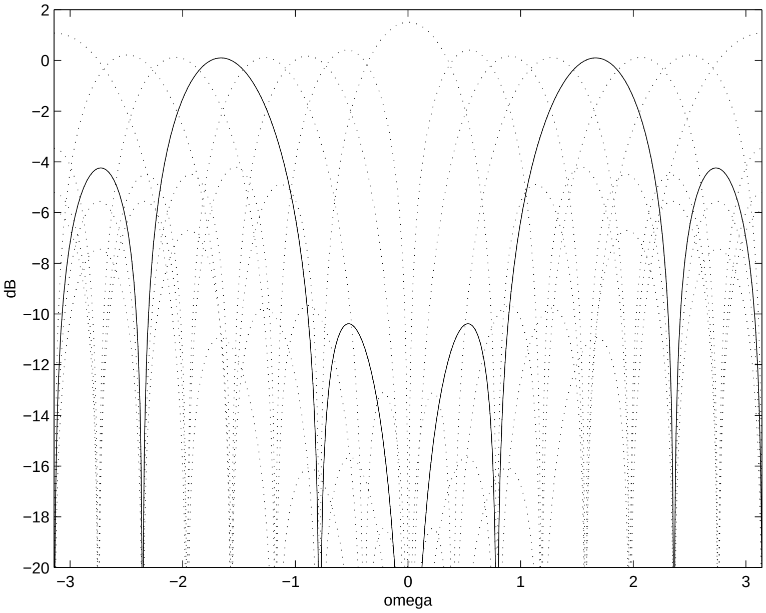


Figure 5: Magnitude responses of DCT basis vectors for $N = 8$.

better stopband performance, say > 90 dB. It turns out that filters with passband width $2\pi/N$, narrow transition bands, and descent stopband attenuation require impulse response lengths $\gg N$. In N-band SC there is no constraint on filter length, unlike N-band TC. This is the advantage of SC over TC when it comes to audio coding[1].

[1] A similar conclusion resulted from our comparison of DPCM and TC of equal dimension N; it was reasoned that the longer "effective" input length of DPCM with N-length prediction filtering gave performance improvement relative to TC.

- To summarize, the key differences between transform and sub-band coding are the following.
  1. SC outputs measure relative signal strength in different frequency bands, while TC outputs might not have a strict bandpass correspondence.
  2. The TC input window length is equal to the number of TC outputs, while the SC input window length is usually much greater than number of SC outputs (16× greater in MPEG).
- At first glance SC implementation complexity is a valid concern. Recall that in TC, fast $N \times N$ transforms such as the DCT and DFT could be performed using $\sim N \log_2 N$ multiply/adds! Must we give up this computational efficiency for better frequency resolution?

  Fortunately the answer is *no*; clever SC implementations are built around fast DFT or DCT transforms and are very efficient as a result. Fast sub-band coding, in fact, lies at the heart of MPEG audio compression [1].

1.2 Fundamentals of Multirate Signal Processing

The presence of upsamplers and downsamplers in the diagram of Fig. 2 implies that a basic knowledge of multirate signal processing is indispensible to an understanding of sub-band analysis/synthesis. This section provides the required background.

- *Modulation:* Fig. 6 illustrates modulation using a complex exponential of frequency ω_o. In the time domain,

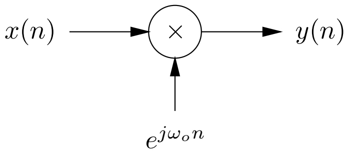


Figure 6: Modulation using $e^{j\omega_o n}$

$$\boxed{y(n) \;=\; x(n)e^{j\omega_o n}}.$$

In the z-domain,

$$Y(z) \;=\; \sum_n y(n)z^{-n} \;=\; \sum_n \big(x(n)e^{j\omega_o n}\big)z^{-n} \;=\; \sum_n x(n)\big(e^{-j\omega_o}z\big)^{-n} \;=\; \boxed{X\big(e^{-j\omega_o}z\big)}.$$

We can evaluate the result of modulation in the frequency domain by substituting $z = e^{j\omega}$. This yields

$$Y(\omega) \;=\; \sum_n y(n)e^{-j\omega n} \;=\; \boxed{X(\omega - \omega_o)}.$$

Note that $X(\omega - \omega_o)$ represents a shift of $X(\omega)$ *up* by ω_o radians, as in Fig. 7.

- *Upsampling:* Fig. 8 illustrates upsampling by factor N. In words, upsampling means the insertion of $N-1$ zeros between every sample of the input process. Formally, upsampling can be expressed in the time domain as

$$\boxed{y(n) \;=\; \begin{cases} x(n/N) & \text{when } n = mN \text{ for } m \in \mathbb{Z} \\ 0 & \text{else.} \end{cases}}$$

In the z-domain, upsampling causes

$$Y(z) \;=\; \sum_n y(n)z^{-n} \;=\; \sum_m x(m)z^{-mN} \;=\; \boxed{X\big(z^N\big)},$$

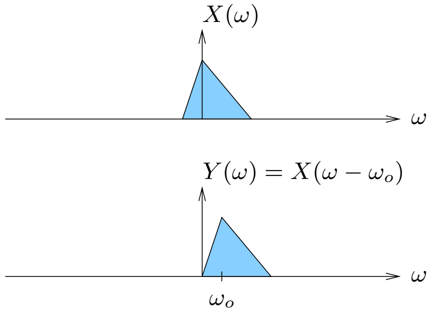


Figure 7: Frequency-domain effect of modulation by $e^{j\omega_o n}$.

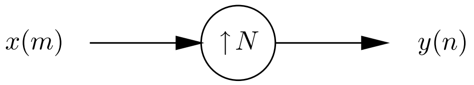


Figure 8: Upsampling by N.

and in the frequency domain,

$$Y(\omega) \;=\; \sum_n y(n) e^{-j\omega n} \;=\; \boxed{X\big(N\omega\big)}.$$

As shown in Fig. 9, upsampling shrinks $X(\omega)$ by a factor of N along the ω axis.

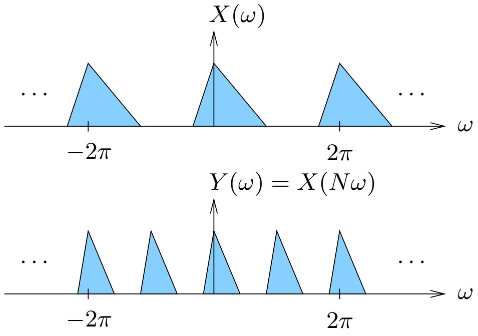


Figure 9: Frequency-domain effects of upsampling by $N\!=\!2$.

- *Downsampling:* Fig. 10 illustrates downsampling by factor N. In words, the process of downsampling keeps every N^{th} sample and discards the rest. Formally, downsampling can be written as

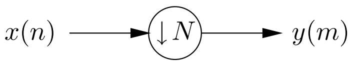


Figure 10: Downsampling by N.

$$\boxed{y(m) \;=\; x(mN).}$$

In the z domain,

$$Y(z) \;=\; \sum_m y(m) z^{-m} \;=\; \sum_m x(mN) z^{-m} \;=\; \sum_n \tilde{x}(n) z^{-n/N},$$

where

$$\tilde{x}(n) \;=\; \begin{cases} x(n) & \text{when } n = mN \text{ for } m \in \mathbb{Z} \\ 0 & \text{else.} \end{cases}$$

The neat trick

$$\frac{1}{N}\sum_{p=0}^{N-1} e^{j\frac{2\pi}{N}np} \;=\; \begin{cases} 1 & \text{when } n = mN \text{ for } m \in \mathbb{Z} \\ 0 & \text{else} \end{cases} \tag{1}$$

(which can be proven) allows us to rewrite $\tilde{x}(n)$ in terms of $x(n)$:

$$\begin{aligned} Y(z) &= \sum_n x(n)\left(\frac{1}{N}\sum_{p=0}^{N-1} e^{j\frac{2\pi}{N}np}\right) z^{-n/N} \\ &= \frac{1}{N}\sum_{p=0}^{N-1}\sum_n x(n)\big(e^{-j\frac{2\pi}{N}p} z^{1/N}\big)^{-n} \\ &= \boxed{\frac{1}{N}\sum_{p=0}^{N-1} X\big(e^{-j\frac{2\pi}{N}p} z^{1/N}\big)}. \end{aligned}$$

Translating to the frequency domain,

$$\boxed{Y(\omega) \;=\; \frac{1}{N}\sum_{p=0}^{N-1} X\left(\frac{\omega - 2\pi p}{N}\right)}.$$

As shown in Fig. 11, downsampling expands each 2π-periodic repetition of $X(\omega)$ by a factor of N along the ω axis. Note the spectral overlap due to downsampling, called "aliasing."

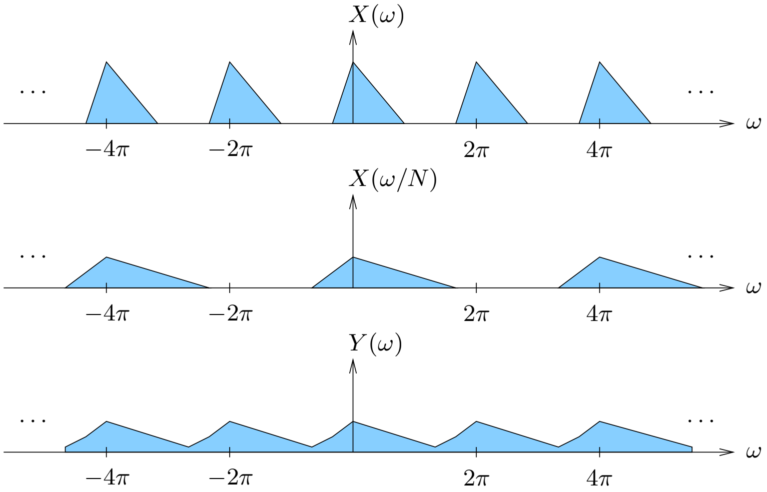


Figure 11: Frequency-domain effects of downsampling by $N\!=\!2$.

- <u>*Downsample-Upsample Cascade:*</u> Downsampling followed by upsampling (of equal factor N) is illustrated by Fig. 12. This structure is useful in understanding analysis/synthesis filterbanks that lie at the heart of sub-band coding schemes. This operation is equivalent to zeroing all but the mN^{th} samples in the input sequence, i.e.,

$$\boxed{y(n) \;=\; \begin{cases} x(n) & \text{when } n = mN \text{ for } m \in \mathbb{Z} \\ 0 & \text{else.} \end{cases}}$$

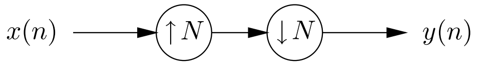


Figure 12: N-Downsampler followed by N-upsampler.

Using trick (1),

$$\begin{aligned}
Y(z) &= \sum_n y(n) z^{-n} \\
&= \sum_n x(n) \left(\frac{1}{N} \sum_{p=0}^{N-1} e^{j\frac{2\pi}{N}np} \right) z^{-n} \\
&= \frac{1}{N} \sum_{p=0}^{N-1} \sum_n x(n) \big(e^{-j\frac{2\pi}{N}p} z \big)^{-n} \\
&= \boxed{\frac{1}{N} \sum_{p=0}^{N-1} X\big(e^{-j\frac{2\pi}{N}p} z \big)},
\end{aligned} \tag{2}$$

which implies

$$\boxed{Y(\omega) \;=\; \frac{1}{N} \sum_{p=0}^{N-1} X\left(\omega - \frac{2\pi p}{N} \right)}. \tag{3}$$

The downsampler-upsampler cascade causes the appearance of $2\pi/N$-periodic copies of the baseband spectrum of $X(\omega)$. As illustrated in Fig. 13, aliasing may result.

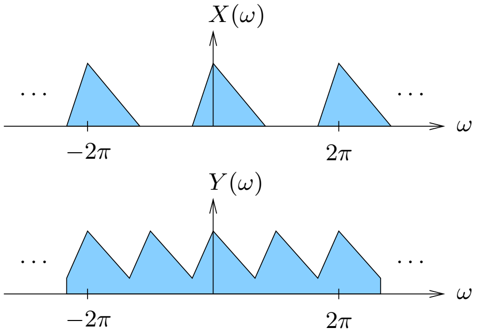


Figure 13: Frequency-domain effects of downsampler-upsampler cascade for $N = 2$.

1.3 Uniformly-Modulated Filterbanks

- *Perfect Reconstruction Filterbanks:* Recall that in our study of transform coding, we restricted our attention to orthogonal transformation matrices. Orthogonal matrices had the property that, in the absence of quantization error, the reconstruction error was zero.

  For sub-band coding, "perfect reconstruction" (PR) filterbanks (FBs) are analogous to orthogonal matrices. Specifically, a PR-FB is defined as an analysis/synthesis structure which gives zero reconstruction error when synthesis stage is fed exact (unquantized) copies of analysis outputs.

  Initially we consider the design of *ideal* sub-band analysis and synthesis FBs and later the design of practical FBs. For the purpose of FB design we ignore the effects of quantization error. Our rational is as follows: the absence of quantization error corresponds to the high bit-rate scenario, in which case we desire that the filtering operations inherent to sub-band coding introduce little or no error of their own. Removing the quantizers from Fig. 2, we obtain the analysis/synthesis FBs in Fig. 14.

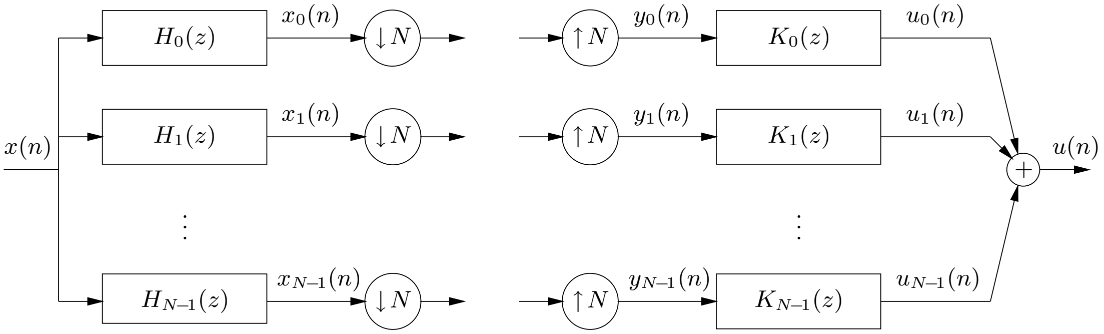


Figure 14: N-band analysis and synthesis filterbanks.

- *Uniform Modulation:* The most conceptually straightforward FB is known as the "uniformly modulated" FB. Uniform modulation means that all branches isolate signal components in non-overlapping frequency bands of equal width $2\pi/N$. We will assume that the i^{th} branch has its frequency band centered at $\omega_i = 2i\pi/N$. (See Fig. 15.)

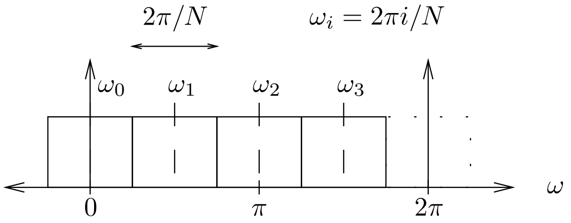


Figure 15: Frequency bands for the uniformly-modulated filterbank ($N = 4$).

- *Analysis FB:* The i^{th} frequency range may be isolated by modulating the input spectrum down by ω_i and lowpass filtering the result. (See the first two stages of the analysis bank in Fig. 17.) The ideal lowpass filter has linear phase and magnitude response that is unity for $\omega \in (-\pi/N, \pi/N)$ and zero elsewhere. (See Fig. 16.)

  With ideal lowpass filtering, the resulting signals have (double-sided) bandwidths that are N times smaller than the sampling rate. Nyquist's sampling theorem [2] says that it is possible to sample signals with this bandwidth at $1/N$ times the filter output rate without loss of information. This sample rate change can implemented via downsampling-by-N, resulting in the analysis FB of Fig. 17. Note that the downsampling operation does not induce aliasing when the analysis filter is the ideal lowpass filter described above.

- *Synthesis FB:* To reconstruct the input signal $x(n)$, the synthesis FB must restore the downsampled signals to their original sampling rate, re-modulate them to their original spectral locations, and combine them.

  Upsampling is the first stage of sampling-rate restoration. Recall from (3) (and Fig. 13) that a downsampler-upsampler cascade creates $N-1$ additional uniformly-spaced spectral copies of the original baseband spectrum. Thus, to remove the unwanted spectral images, an "anti-imaging"

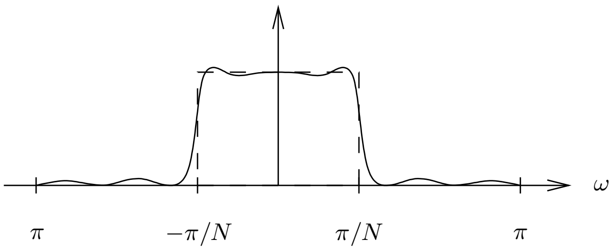


Figure 16: Ideal (dashed) and typical (solid) prototype-filter magnitude responses for the uniformly modulated filterbank.

lowpass filter is applied to each upsampler's output. Ideally, this lowpass filter is linear phase with magnitude response that is unity for $\omega \in (-\pi/N, \pi/N)$ and zero elsewhere; the same specifications given for the ideal analysis filter. (See Fig. 16.)

As shown in Fig. 17, re-modulation is accomplished by shifting the i^{th} branch *up* by ω_i.

When the analysis and synthesis filters share a common phase response, the re-modulator outputs can be combined coherently by a simple summation. Under all of these ideal conditions, the output signal $u(n)$ is a potentially delayed (but otherwise perfect) copy of the input signal $x(n)$:

$$u(n) \; = \; x(n-\delta) \;\; \text{for some delay } \delta.$$

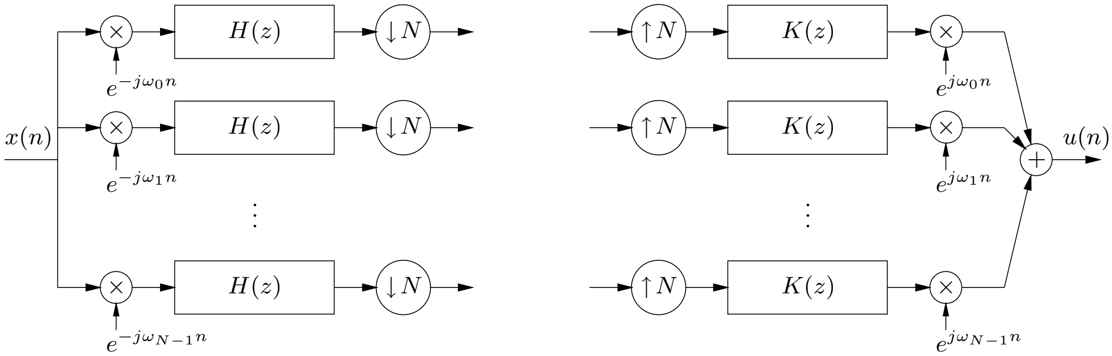


Figure 17: N-band modulated analysis/synthesis filterbanks.

- *Effect of Non-Ideal Filtering:* In practice, the analysis and synthesis filters will not have ideal lowpass responses, and thus the reconstructed output $u(n)$ will not necessarily equal a delayed version of the input $x(n)$. These shortcomings typically result from filter implementations based on a finite number of design parameters. (See Fig. 16 for a typical lowpass filter magnitude response.)

  It should be noted that, under certain conditions, it is possible to design sets of analysis filters $\{H_i(z)\}$ and synthesis filters $\{K_i(z)\}$ with *finite* parameterizations which give the "perfect reconstruction" (PR) property [3]. Though such filters guarantee PR, they do not act as ideal bandpass filters and thus do not accomplish perfect frequency analysis. (Consider the length-N DFT and DCT filter responses: by the orthogonal matrix argument, these are perfectly reconstructing, but from Figs. 4 and 5, they are far from perfect bandpass filters!)

  Due to their limited frequency-selectivity, none of the currently-known PR filterbanks are appropriate for high-quality audio applications. As a result, we focus on the design of filterbanks with

1. *near*-perfect reconstruction and
2. good frequency selectivity.

As we will see, it is possible to design practical filters with excellent frequency selectivity and responses so close to PR that the smallest quantization errors swamp out errors caused by non-PR filtering.

- *Polyphase/DFT Implementation:* When $H(z)$ and $K(z)$ are length-M FIR filters, the unique elements in Fig. 17 are the N uniform-modulation coefficients $\{e^{j2\pi n/N}; n = 0, \ldots, N-1\}$ and the $2M$ the lowpass filter coefficients $\{h_n\}$ and $\{k_n\}$. It might not be surprising that each half of the uniformly-modulated FB has an implementation that requires only one N-dimensional DFT and M multiplies to process an N-block of input samples. Fig. 18 illustrates one such implementation, where the "polyphase" filters $\{H^{(\ell)}(z)\}$ and $\{K^{(\ell)}(z)\}$ are related to the "prototype" filters $H(z)$ and $K(z)$ through the impulse response relations:

$$\begin{array}{rcl} h_m^{(\ell)} & := & h_{mN+\ell} \\ k_m^{(\ell)} & := & k_{mN+\ell} \end{array} \quad \text{for} \quad \ell = 0, \ldots, N-1.$$

The term "polyphase" comes about because the magnitude responses of well-designed $\{H^{(\ell)}(z)\}$ and $\{K^{(\ell)}(z)\}$ are nearly flat, while the slopes of the phase response of these filters differ by small amounts. The equivalence of Fig. 17 and Fig. 18 will be established in the homework.

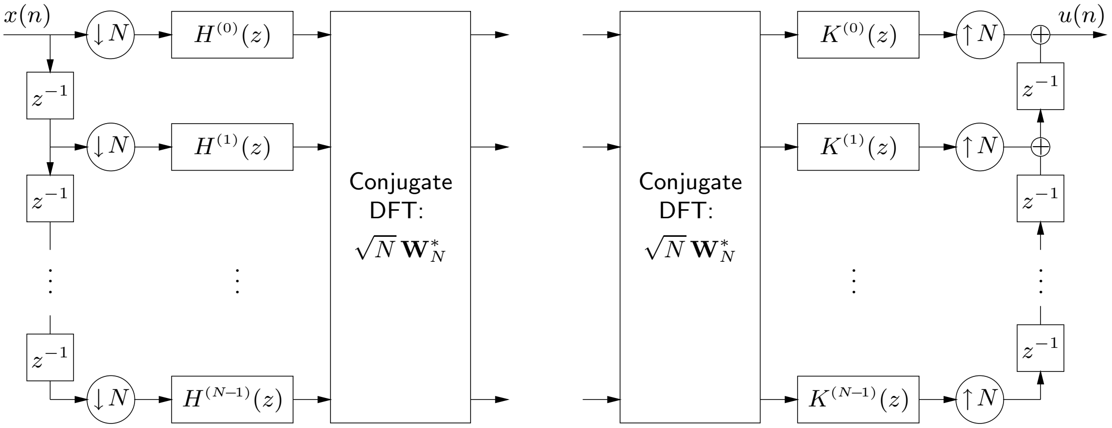


Figure 18: Polyphase/DFT implementation of N-band uniformly modulated analysis/synthesis filterbanks.

Recognize that the filter computations in Fig. 18 occur on downsampled (i.e., low-rate) data, in contrast to those in Fig. 17. To put it another way, all but one of every N filter outputs in Fig. 17 are thrown away by the downsampler, whereas none of the filter outputs in Fig. 18 are thrown away. This reduces the number of required filter computations by a factor of N.

Additional computational reduction occurs when the DFT is implemented by a fast transform. Below we give a concrete example.

Example 1.1 (Computational Savings of Polyphase/FFT Implementation):

Lets take a look at how many multiplications we save by using the polyphase/DFT analysis filterbank in Fig. 18 instead of the standard modulated filterbank in Fig. 17. Here we assume that N is a power of 2, so that the DFT can be implemented with a radix-2 FFT.

With the standard structure in Fig. 17, modulation requires $2N$ real multiplies, and filtering of the complex-valued modulator outputs requires $2 \times N \times M$ additional real multiplies, for each input point $x(n)$. This gives a total of

$$\boxed{2N(M+1)} \text{ real multiplies per input.}$$

In the polyphase/FFT structure of Fig. 18, it is more convenient to count the number of multiplies required for each block of N inputs since each new N-block produces one new sample at every filter input and one new N-vector at the DFT input. Since the polyphase filters are each length-M/N, filtering the block requires $N \times M/N = M$ real multiplies. Though the standard radix-2 N-dimensional complex-valued FFT uses $\frac{N}{2}\log_2 N$ complex multiplies, a real-valued N-dimensional FFT can be accomplished in $N\log_2 N$ real multiplies when N is a power of 2[4]. This gives a total of

$$M + N\log_2 N \text{ real multiplies per } N \text{ inputs, or } \boxed{M/N + \log_2 N} \text{ real multiplies per input!}$$

Say we have $N = 32$ frequency bands and the prototype filter is length $M = 512$ (which turn out to be the values used in the MPEG sub-band filter). Then using the formulas above, the standard implementation requires $\boxed{32832}$ multiplies per input, while the polyphase/DFT implementation requires only $\boxed{21}$!

1.4 MPEG Layers 1-3: Cosine-Modulated Filterbanks

- Though the uniformly modulated filterbank in Fig. 17 was shown to have the fast implementation in Fig. 18, the sub-band outputs are complex-valued for real-valued input, hence inconvenient (at first glance[2]) for sub-band coding of real-valued data.

  In this section we propose a closely related filterbank with the following properties.

  1. Real-valued sub-band outputs (assuming real-valued inputs),
  2. Near-perfect reconstruction,
  3. Polyphase/fast-transform implementation.

  This turns out to be the filterbank specified in the MPEG-1 and 2 (layers 1-3) audio compression standards [1].

1.4.1 Filter Design

- Real-valued Sub-band Outputs: Recall the generic filterbank structure of Fig. 14. For the sub-band outputs to be real-valued (for real-valued input), we require that the impulse responses of $\{H_i(z)\}$ and $\{K_i(z)\}$ are real-valued. We can insure this by allocating the N (symmetric) frequency band pairs shown in Fig. 19. The positive and negative halves of each band pair are centered at $\omega_i = \frac{(2i+1)\pi}{2N}$ radians.

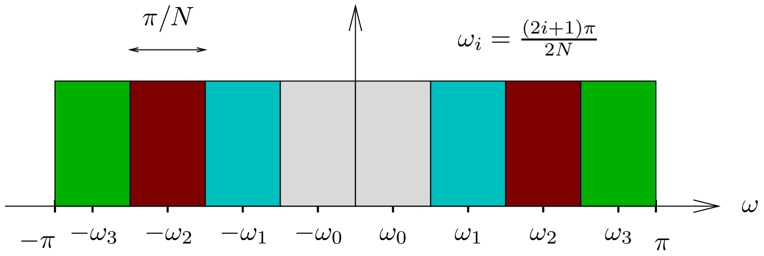


Figure 19: Frequency band pairs for the polyphase quadrature filterbank ($N = 4$).

We can consider each filter $H_i(z)$ as some combination of symmetric positive-frequency and negative-frequency components

$$H_i(z) \;=\; a_i F_i(z) + b_i G_i(z)$$

as shown in Fig. 20. When $b_i = a_i^*$ and the pairs $\{F_i(z), G_i(z)\}$ are modulated versions of the

[2] In the structure in Fig. 17, it would be reasonable to replace the standard DFT with a real-valued DFT (defined in the notes on transform coding), requiring $\approx N\log_2 N$ real-multiplies when N is a power of 2. Though it is not clear to the author why such a structure was not adopted in the MPEG standards, the cosine modulated filterbank derived in this section has equivalent performance and, with its polyphase/DCT implementation, equivalent implementation cost.

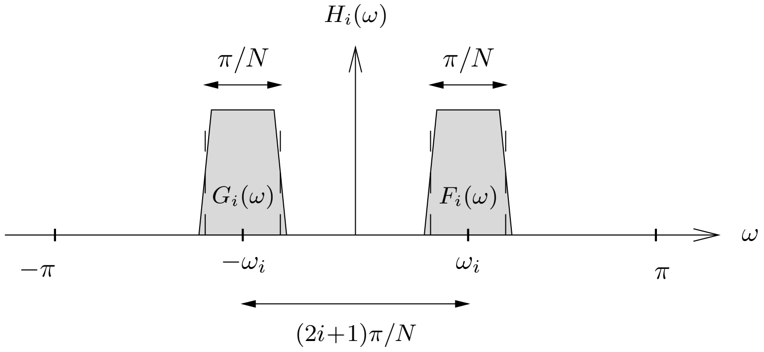


Figure 20: Positive- and negative-frequency decomposition of $H_i(\omega)$. Note $K_i(\omega)$ will have a similar, if not identical, frequency response.

same prototype filter $H(z)$, we can show that $H_i(z)$ must be real-valued:

$$\begin{aligned}
H_i(z) &= a_i \underbrace{H(e^{-j\pi\frac{2i+1}{2N}}z)}_{F_i(z)} + a_i^* \underbrace{H(e^{j\pi\frac{2i+1}{2N}}z)}_{G_i(z)} \\
&= a_i \sum_n h_n e^{j\pi\frac{2i+1}{2N}n} z^{-n} + a_i^* \sum_n h_n e^{-j\pi\frac{2i+1}{2N}n} z^{-n} \\
&= \mathrm{Re}(a_i) \sum_n h_n z^{-n} \left(e^{j\pi\frac{2i+1}{2N}n} + e^{-j\pi\frac{2i+1}{2N}n}\right) + j\mathrm{Im}(a_i) \sum_n h_n z^{-n} \left(e^{j\pi\frac{2i+1}{2N}n} - e^{-j\pi\frac{2i+1}{2N}n}\right) \\
&= \mathrm{Re}(a_i) \sum_n h_n z^{-n} \cdot 2\cos\left(\pi\frac{2i+1}{2N}n\right) + j\mathrm{Im}(a_i) \sum_n h_n z^{-n} \cdot 2j\sin\left(\pi\frac{2i+1}{2N}n\right) \\
&= 2\sum_n \left[\mathrm{Re}(a_i)\, \cos\left(\pi\frac{2i+1}{2N}n\right) - \mathrm{Im}(a_i)\, \sin\left(\pi\frac{2i+1}{2N}n\right)\right] h_n z^{-n} \qquad (4)
\end{aligned}$$

- *Aliasing Cancellation:* Recall again the generic filterbank in Fig. 14. Here we determine conditions on real-valued $\{H_i(z)\}$ and $\{K_i(z)\}$ which lead to near-perfect reconstruction.

  It will be insightful to derive an expression for the input to the i^{th} reconstruction filter, $\{y_i(n)\}$. The downsample-upsample-cascade equation (2) implies that

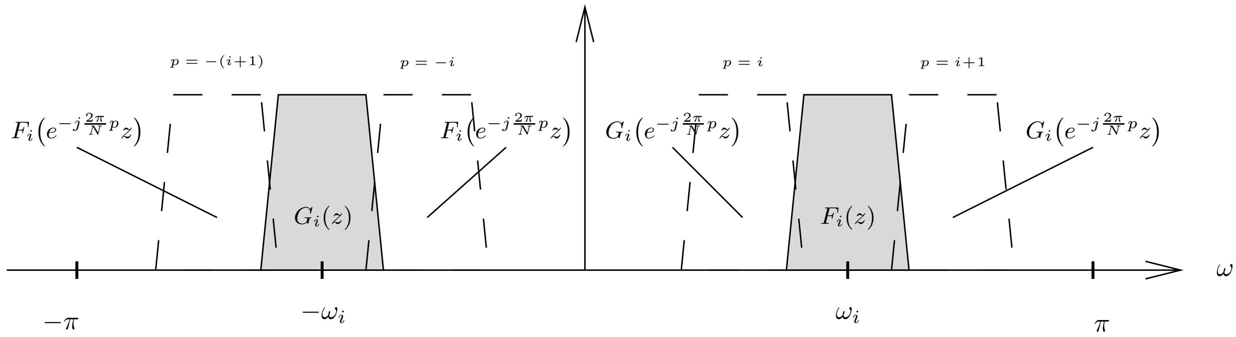
$$\begin{aligned}
Y_i(z) &= \frac{1}{N}\sum_{p=0}^{N-1} X_i\big(e^{-j\frac{2\pi}{N}p}z\big) \\
&= \frac{1}{N}\sum_{p=0}^{N-1} H_i\big(e^{-j\frac{2\pi}{N}p}z\big) X\big(e^{-j\frac{2\pi}{N}p}z\big) \\
&= \frac{1}{N}\sum_{p=0}^{N-1} \Big[a_i F_i\big(e^{-j\frac{2\pi}{N}p}z\big) + a_i^* G_i\big(e^{-j\frac{2\pi}{N}p}z\big)\Big]\, X\big(e^{-j\frac{2\pi}{N}p}z\big) \qquad (5) \\
&= \underbrace{\frac{1}{N}\left[a_i F_i(z) + a_i^* G_i(z)\right] X(z)}_{\text{desired}} + \underbrace{\frac{1}{N}\sum_{p=1}^{N-1} \Big[a_i F_i\big(e^{-j\frac{2\pi}{N}p}z\big) + a_i^* G_i\big(e^{-j\frac{2\pi}{N}p}z\big)\Big]\, X\big(e^{-j\frac{2\pi}{N}p}z\big)}_{\text{undesired images}}.
\end{aligned}$$

  Thus the input to the i^{th} reconstruction filter is corrupted by unwanted spectral images, and the reconstruction filter's job is the removal of these images.

  The reconstruction filter $K_i(z)$ will have a bandpass frequency response similar (or identical) to that of $H_i(z)$ illustrated in Fig. 20. Due to the practical design considerations, neither $K_i(z)$ nor $H_i(z)$

will be perfect bandpass filters, but we will assume that the only significant out-of-band energy passed by these filters will occur in the frequency range just outside of their passbands. (Note the limited "spillover" in Fig. 20.)

Under these assumptions, the only undesired images in $Y_i(\omega)$ that will not be completely attenuated by $K_i(\omega)$ are the images adjacent to $F_i(\omega)$ and $G_i(\omega)$. Which indices p in (5) are responsible for these *adjacent* images? Equation (5) implies that index $p = \ell$ shifts the frequency response up by $2\pi\ell/N$ radians. Since the passband centers of $F_i(z)$ and $G_i(z)$ are $(2i+1)\pi/N$ radians apart, the passband of $G_i\big(e^{-j\frac{2\pi}{N}p}z\big)$ will reside directly to the left of the passband of $F_i(z)$ when $p = i$. Similarly, the passband of $G_i\big(e^{-j\frac{2\pi}{N}p}z\big)$ will reside directly to the right of the passband of $F_i(z)$ when $p = i+1$. See Fig. 21 for an illustration. Using the same reasoning, the passband of $F_i\big(e^{-j\frac{2\pi}{N}p}z\big)$ will reside directly to the right of the passband of $G_i(z)$ when $p = -i$ and directly to the left when $p = -(i+1)$. The only exceptions to this rule occur when $i = 0$, in which case the images to the right of $G_i(z)$ and to the left of $F_i(z)$ are desired, and when $i = N-1$, in which case the images to the left of $G_i(z)$ and to the right of $F_i(z)$ are desired.



$p = -(i+1)$
$p = -i$
$p = i$
$p = i+1$
$F_i\big(e^{-j\frac{2\pi}{N}p}z\big)$
$F_i\big(e^{-j\frac{2\pi}{N}p}z\big)$
$G_i\big(e^{-j\frac{2\pi}{N}p}z\big)$
$G_i\big(e^{-j\frac{2\pi}{N}p}z\big)$
$G_i(z)$
$F_i(z)$
ω
$-\pi$
$-\omega_i$
ω_i
π

Figure 21: Spectral images of $Y_i(\omega)$ not completely attenuated by $K_i(\omega)$.

Based on the arguments above, we can write $\{u_i(n)\}$, the output of the i^{th} reconstruction filter, as follows:

$$\begin{aligned}
U_i(z) &= K_i(z)Y_i(z) \\
&= \underbrace{\frac{1}{N}K_i(z)\big[a_iF_i(z)X(z) + a_i^*G_i(z)X(z)\big]}_{\text{desired}} \\
&\quad + \underbrace{\frac{1}{N}K_i(z)\left[a_iF_i\big(e^{j\frac{2\pi}{N}i}z\big)X\big(e^{j\frac{2\pi}{N}i}z\big) + a_i^*G_i\big(e^{-j\frac{2\pi}{N}i}z\big)X\big(e^{-j\frac{2\pi}{N}i}z\big)\right]}_{\text{aliasing from inner undesired images when } 1 \le i \le N-1} \\
&\quad + \underbrace{\frac{1}{N}K_i(z)\left[a_iF_i\big(e^{j\frac{2\pi}{N}(i+1)}z\big)X\big(e^{j\frac{2\pi}{N}(i+1)}z\big) + a_i^*G_i\big(e^{-j\frac{2\pi}{N}(i+1)}z\big)X\big(e^{-j\frac{2\pi}{N}(i+1)}z\big)\right]}_{\text{aliasing from outer undesired images when } 0 \le i \le N-2}.
\end{aligned} \quad (6)$$

The previous equation shows that $U_i(z)$ is corrupted by the portions of the undesired images not completely removed by the reconstruction filter $K_i(z)$. In the filterbank context, this undesired behavior is referred to as aliasing. But notice that aliasing contributions to the signal $U(z) = \sum_i U_i(z)$ will vanish if the inner aliasing components in $U_i(z)$ cancel the outer aliasing components in $U_{i-1}(z)$. This happens when

$$\begin{aligned}
&K_i(z)\left[a_iF_i\big(e^{j\frac{2\pi}{N}i}z\big)X\big(e^{j\frac{2\pi}{N}i}z\big) + a_i^*G_i\big(e^{-j\frac{2\pi}{N}i}z\big)X\big(e^{-j\frac{2\pi}{N}i}z\big)\right] \\
&\quad = -K_{i-1}(z)\left[a_{i-1}F_{i-1}\big(e^{j\frac{2\pi}{N}i}z\big)X\big(e^{j\frac{2\pi}{N}i}z\big) + a_{i-1}^*G_{i-1}\big(e^{-j\frac{2\pi}{N}i}z\big)X\big(e^{-j\frac{2\pi}{N}i}z\big)\right].
\end{aligned}$$

which occurs under satisfaction of the two conditions below.

$$
\begin{aligned}
a_i K_i(z) F_i\big(e^{j\frac{2\pi}{N}i}z\big) &= -a_{i-1} K_{i-1}(z) F_{i-1}\big(e^{j\frac{2\pi}{N}i}z\big) && (7)\\
a_i^* K_i(z) G_i\big(e^{-j\frac{2\pi}{N}i}z\big) &= -a_{i-1}^* K_{i-1}(z) G_{i-1}\big(e^{-j\frac{2\pi}{N}i}z\big). && (8)
\end{aligned}
$$

We assume from this point on that the real-valued filters $\{H_i(z)\}$ and $\{K_i(z)\}$ are constructed using modulated versions of a lowpass prototype filter $H(z)$. (This assumption is required for the existence of a polyphase filterbank implementation.)

$$
\begin{aligned}
H_i(z) &= a_i F_i(z) + a_i^* G_i(z)\\
K_i(z) &= c_i F_i(z) + c_i^* G_i(z)
\end{aligned}
\quad \text{where} \quad
\left\{
\begin{aligned}
F_i(z) &= H\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big)\\
G_i(z) &= H\big(e^{j\frac{\pi}{2N}(2i+1)}z\big)
\end{aligned}
\right.
$$

Then condition (7) becomes

$$
\begin{aligned}
&a_i c_i H\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big) H\big(e^{j\frac{\pi}{2N}(2i-1)}z\big) + a_i c_i^* H\big(e^{j\frac{\pi}{2N}(2i+1)}z\big) H\big(e^{j\frac{\pi}{2N}(2i-1)}z\big)\\
&\quad = \; -a_{i-1} c_{i-1} H\big(e^{-j\frac{\pi}{2N}(2i-1)}z\big) H\big(e^{j\frac{\pi}{2N}(2i+1)}z\big) - a_{i-1} c_{i-1}^* H\big(e^{j\frac{\pi}{2N}(2i-1)}z\big) H\big(e^{j\frac{\pi}{2N}(2i+1)}z\big). \quad (9)
\end{aligned}
$$

Lets take a closer look at the products $H\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big)H\big(e^{j\frac{\pi}{2N}(2i-1)}z\big)$ in the previous equation. As illustrated in Fig. 22, these products equal zero when $1 \le i \le N/2$ since their passbands do not overlap. Setting these products to zero in (9) yields the condition

$$
\boxed{a_i c_i^* \;=\; -\,a_{i-1} c_{i-1}^* \quad \text{for} \quad 1 \le i \le N-1}, \qquad (10)
$$

which can also be shown to satisfy (8).

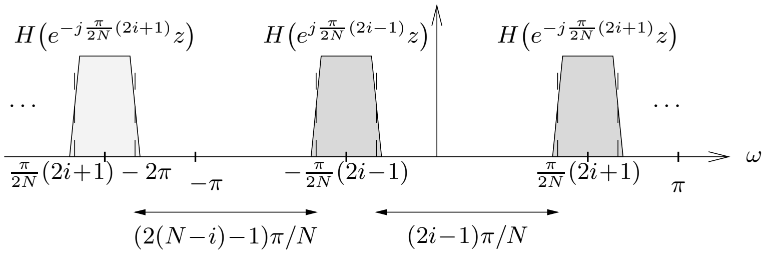

$H\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big)$
$H\big(e^{j\frac{\pi}{2N}(2i-1)}z\big)$
$H\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big)$
$\cdots$
$\cdots$
$\frac{\pi}{2N}(2i+1) - 2\pi$
$-\pi$
$-\frac{\pi}{2N}(2i-1)$
$\frac{\pi}{2N}(2i+1)$
π
ω
$(2(N-i)-1)\pi/N$
$(2i-1)\pi/N$

Figure 22: Illustration of vanishing terms in (9).

Next we concern ourselves with the requirements on a_0 and c_0. Assuming (10) is satisfied, we know that inner aliasing in $U_i(z)$ cancels outer aliasing in $U_{i-1}(z)$ for $1 \le i \le N-1$. Hence, from (6) and (9),

$$
\begin{aligned}
U(z) &= \sum_{i=0}^{N-1} U_i(z)\\
&= \frac{1}{N} \sum_{i=0}^{N-1} K_i(z) H_i(z) X(z)\\
&= \frac{1}{N} \sum_{i=0}^{N-1} \Big[c_i H\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big) + c_i^* H\big(e^{j\frac{\pi}{2N}(2i+1)}z\big)\Big]\\
&\qquad \cdot \Big[a_i H\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big) + a_i^* H\big(e^{j\frac{\pi}{2N}(2i+1)}z\big)\Big] X(z)
\end{aligned}
$$

Noting that the passbands of $H\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big)$ and $H\big(e^{j\frac{\pi}{2N}(2i+1)}z\big)$ do not overlap for $1 \le i \le N-2$, we have

$$\begin{aligned} U(z) &= \frac{1}{N}\Bigg[(a_0 c_0^* + a_0^* c_0) H\big(e^{-j\frac{\pi}{2N}}z\big) H\big(e^{j\frac{\pi}{2N}}z\big) \\ &\quad + (a_{N-1} c_{N-1}^* + a_{N-1}^* c_{N-1}) H\big(e^{-j\frac{\pi}{2N}(2N-1)}z\big) H\big(e^{j\frac{\pi}{2N}(2N-1)}z\big) \\ &\quad + \sum_{i=0}^{N-1}\Big(a_i c_i H^2\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big) + a_i^* c_i^* H^2\big(e^{j\frac{\pi}{2N}(2i+1)}z\big)\Big)\Bigg] X(z). \end{aligned} \tag{11}$$

The first two terms in (11) represent aliasing components that prevent flat overall response at $\omega = 0$ and $\omega = \pi$, respectively. These aliasing terms vanish when

$$\boxed{\begin{aligned} a_0 c_0^* &= -a_0^* c_0 \\ a_{N-1} c_{N-1}^* &= -a_{N-1}^* c_{N-1} \end{aligned}} \tag{12}$$

What remains is

$$U(z) = \frac{1}{N}\sum_{i=0}^{N-1}\Big(a_i c_i H^2\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big) + a_i^* c_i^* H^2\big(e^{j\frac{\pi}{2N}(2i+1)}z\big)\Big) X(z).$$

- Phase Distortion: Perfect reconstruction requires that the analysis/synthesis system has no phase distortion. To guarantee the absence of phase distortion, we require that the composite system

$$Q(z) := \frac{U(z)}{X(z)} = \frac{1}{N}\sum_{i=0}^{N-1} a_i c_i H^2\big(e^{-j\frac{\pi}{2N}(2i+1)}z\big) + a_i^* c_i^* H^2\big(e^{j\frac{\pi}{2N}(2i+1)}z\big)$$

has a linear phase response. (Recall that a linear phase response is equivalent to a pure delay in the time domain.) This linear-phase constraint will provide the final condition used to specify the constants $\{a_i\}$ and $\{c_i\}$.

We start by examining the impulse response of $Q(z)$. Using a technique analogous to (4), we can write

$$Q(z) = \frac{2}{N}\sum_{n=0}^{2M-2}\left(\sum_{i=0}^{N-1}\mathrm{Re}(a_i c_i)\cos\left(\pi\frac{2i+1}{2N}n\right) - \mathrm{Im}(a_i c_i)\sin\left(\pi\frac{2i+1}{2N}n\right)\right)\left(\sum_k h_k h_{n-k}\right) z^{-n}$$

Above, we have used the property that multiplication in the z-domain implies convolution in the time domain.

For $Q(z)$ to be linear phase, it's impulse response must be symmetric. Let us assume that the prototype filter $H(z)$ is linear phase, so that $\{h_n\}$ is symmetric. Thus $\sum_k h_m h_{n-k}$ is symmetric about $n = M-1$, and thus for linear phase $Q(z)$, we require that the quantity

$$\sum_{i=0}^{N-1}\mathrm{Re}(a_i c_i)\cos\left(\pi\frac{2i+1}{2N}n\right) - \mathrm{Im}(a_i c_i)\sin\left(\pi\frac{2i+1}{2N}n\right)$$

is symmetric about $n = M-1$, i.e.,

$$\begin{aligned} &\sum_{i=0}^{N-1}\mathrm{Re}(a_i c_i)\cos\left(\pi\frac{2i+1}{2N}(M-1+n)\right) - \mathrm{Im}(a_i c_i)\sin\left(\pi\frac{2i+1}{2N}(M-1+n)\right) \\ &= \sum_{i=0}^{N-1}\mathrm{Re}(a_i c_i)\cos\left(\pi\frac{2i+1}{2N}(M-1-n)\right) - \mathrm{Im}(a_i c_i)\sin\left(\pi\frac{2i+1}{2N}(M-1-n)\right) \end{aligned}$$

for $n = 0, \dots, M-1$. Using trigonometric identities, it can be shown that the condition above is equivalent to

$$0 \;=\; \sum_{i=0}^{N-1} \sin\left(\pi \frac{2i+1}{2N} n\right) \left[\mathrm{Re}(a_i c_i) \sin\left(\pi \frac{2i+1}{2N}(M-1)\right) + \mathrm{Im}(a_i c_i) \cos\left(\pi \frac{2i+1}{2N}(M-1)\right)\right],$$

which is satisfied when

$$\frac{\mathrm{Im}(a_i c_i)}{\mathrm{Re}(a_i c_i)} \;=\; -\frac{\sin\left(\pi \frac{2i+1}{2N}(M-1)\right)}{\cos\left(\pi \frac{2i+1}{2N}(M-1)\right)} \;=\; \tan\left(-\pi \frac{2i+1}{2N}(M-1)\right).$$

Restricting $|a_i| = |c_i| = 1$, the previous equation requires that

$$\boxed{a_i c_i \;=\; e^{-j\pi \frac{2i+1}{2N}(M-1)}}. \tag{13}$$

It can be easily verified that the following $\{a_i\}$ and $\{c_i\}$ satisfy conditions (10), (12), and (13):

$$\boxed{\begin{aligned} a_i &= e^{-j\pi \frac{M+N-1}{4N}(2i+1)} \\ c_i &= e^{-j\pi \frac{M-N-1}{4N}(2i+1)}. \end{aligned}} \tag{14}$$

Plugging these into the expression for $H_i(z)$ we find that

$$\begin{aligned} H_i(z) &= a_i H\big(e^{-j\pi \frac{2i+1}{2N}} z\big) + a_i^* H\big(e^{j\pi \frac{2i+1}{2N}} z\big) \\ &= \sum_{n=0}^{M-1} \left(a_i e^{j\pi \frac{2i+1}{2N} n} + a_i^* e^{-j\pi \frac{2i+1}{2N} n} \right) h_n z^{-n} \\ &= \sum_{n=0}^{M-1} \left(e^{j\pi \frac{2i+1}{2N}(n - \frac{M+N-1}{2})} + e^{-j\pi \frac{2i+1}{2N}(n - \frac{M+N-1}{2})} \right) h_n z^{-n} \\ &= \sum_{n=0}^{M-1} \underbrace{\boxed{2\cos\left(\pi \frac{2i+1}{2N}\left(n - \frac{M+N-1}{2}\right)\right) h_n}}_{\text{impulse response of } H_i(z)} z^{-n}. \end{aligned} \tag{15}$$

Repeating this procedure for $K_i(z)$ yields

$$K_i(z) \;=\; \sum_{n=0}^{M-1} \underbrace{\boxed{2\cos\left(\pi \frac{2i+1}{2N}\left(n - \frac{M-N-1}{2}\right)\right) h_n}}_{\text{impulse response of } K_i(z)} z^{-n}. \tag{16}$$

At this point we make a few comments on the design of the lowpass prototype $H(z)$. The perfect $H(z)$ would be an ideal linear-phase lowpass filter with cutoff at $\omega = \pi/2N$, as illustrated in Fig. 23. Such a filter would perfectly separate the subbands as well as yield flat composite magnitude response, as per (13). Unfortunately, however, this perfect filter is not realizable with a finite number of filter coefficients. So, what we really want is a finite-length FIR filter having good frequency selectivity, nearly-flat composite response, and linear phase. The length-512 prototype filter specified in the MPEG standards is such a filter, as evidenced by the responses in Fig. 24. Unfortunately, the standards do not describe how this filter was designed, and a thorough discussion of multirate filter design is outside the scope of this course. For more on prototype filter design, we point the interested reader to [3, p. 358] and [5].

To conclude, (15) and (16) give impulse response expressions for a set of real-valued filters that comprise a near-perfectly reconstructing filterbank (under suitable selection of $\{h_i\}$). This is commonly referred to[3] as a "cosine-modulated filterbank" because all filters are based on cosine modulations

[3] The MPEG standards refer to this filterbank as a "polyphase quadrature" filterbank (PQF), the name given to the technique by an early technical paper [6].

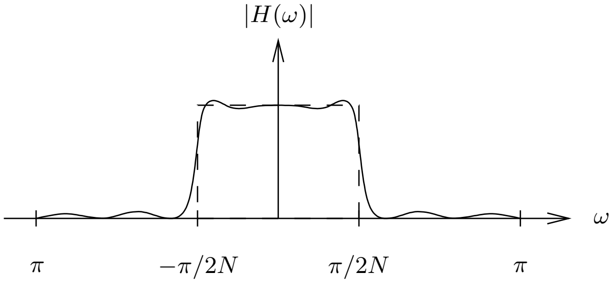


Figure 23: Ideal (dashed) and typical (solid) prototype-filter magnitude responses for the cosine-modulated filterbank. Note bandwidth relative to Fig. 16.

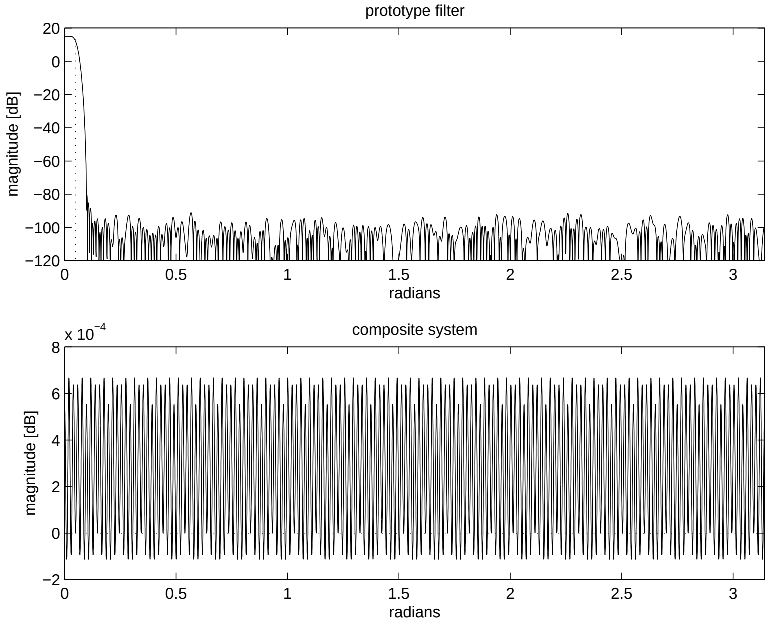


Figure 24: Magnitude response of $|H(\omega)|$ of MPEG prototype filter and the resulting composite response $|Q(\omega)|$, where $N = 32$ and $M = 16N = 512$.

of a real-valued linear-phase lowpass prototype $H(z)$. The near-perfect reconstruction property follows from the frequency-domain cancellation of adjacent-spectrum aliasing and the lack of phase distortion.

It should be noted that our derivation of the cosine modulated filterbank is similar to that in [6] except for the treatments of phase distortion. See Chapter 8 of [3] for a more comprehensive view of cosine-modulated filterbanks.

- *Polyphase Implementations:* Recall the uniformly modulated filterbank in Fig. 17, whose combined modulator-filter coefficients can be constructed using products of the terms h_n and $e^{j\frac{\pi}{N}in}$. Fig. 18 shows a computationally-efficient polyphase/DFT implementation of the analysis filter which requires only M multiplies and one N-dimensional DFT computation for calculation of N subband outputs. We might wonder: Is there a similar polyphase/fast-transform implementation of the cosine-modulated filterbank derived in this section?

  From (15), we see that the impulse responses of $\{H_i(z)\}$ are products of the terms h_n and $\cos\left(\pi\frac{2i+1}{2N}\left(n - \frac{M+N-1}{2}\right)\right)$ for $n = 0, \ldots, M-1$. Note that the inverse-DCT matrix $\mathbf{C}_n^t$ can be specified via components with form similar to the cosine term in (15):

$$\left[\mathbf{C}_N^t\right]_{i,n} \;=\; \sqrt{\frac{2}{N}}\,\alpha_n \cos\left(\pi\frac{(2i+1)}{2N}n\right); \quad i,n = 0\ldots N-1.$$

$$\text{for} \;\; \alpha_0 = 1/\sqrt{2}, \;\; \alpha_{n\neq 0} = 1.$$

Thus it may not be surprising that there exist polyphase/DCT implementations of the cosine-modulated filterbank. Indeed, one such implementation is specified in the MPEG-2 audio compression standard [1]. This particular implementation is the focus of the next section.

1.4.2 MPEG Filterbank Implementation

- Since MPEG audio compression standards are so well-known and widespread, a detailed look at the MPEG filterbank implementation is warranted. The cosine-modulated, or polyphase-quadrature filterbank described in the previous section is used in MPEG Layers 1-3. (The MPEG hierarchy will be described in a later chapter.) This section discusses the specific implementation suggested by the MPEG-2 standard [1].

- The MPEG standard specifies 512 prototype filter coefficients, the first of which is zero. To adapt the MPEG filter to our cosine-modulated-filterbank framework, we append a zero-valued 513^{th} coefficient so that the resulting MPEG prototype filter becomes symmetric and hence linear phase. Since the standard specifies $N = 32$ frequency bands, we have

  $$M \;=\; 513 \;=\; 16N+1.$$

  Plugging this value of M into the filter expressions (15) and (16), the 2π-periodicity of the cosine implies that they may be rewritten as follows.

  $$H_i(z) \;=\; \sum_{n=0}^{16N-1} \underbrace{2\cos\left(\pi\tfrac{2i+1}{2N}\left(n-\tfrac{N}{2}\right)\right)h_n}_{\text{impulse response of } H_i(z)} z^{-n}$$

  $$K_i(z) \;=\; \sum_{n=0}^{16N-1} \underbrace{2\cos\left(\pi\tfrac{2i+1}{2N}\left(n+\tfrac{N}{2}\right)\right)h_n}_{\text{impulse response of } K_i(z)} z^{-n}.$$

- *Encoding:* Here we derive the encoder filterbank implementation suggested in the MPEG-2 standard [1]. Using $x_i(n)$ to denote the output of the i^{th} analysis filter, we have

  $$x_i(n) \;=\; \sum_{k=0}^{16N-1}\left[2\cos\left(\pi\tfrac{2i+1}{2N}\left(k-\tfrac{N}{2}\right)\right)h_k\right]x(n-k).$$

  The relationship between $x_i(n)$ and its downsampled version $s_i(m)$ is given by

  $$s_i(m) = x_i(mN),$$

  so that the downsampled analysis output $s_i(m)$ can be written as

  $$s_i(m) \;=\; \sum_{n=0}^{16N-1}\left[2\cos\left(\pi\tfrac{2i+1}{2N}\left(n-\tfrac{N}{2}\right)\right)h_n\right]x(mN-n).$$

  Using the substitution $n = kN+\ell$ for $0 \leq \ell \leq N-1$,

  $$\begin{aligned} s_i(m) \;&=\; 2\sum_{k=0}^{15}\sum_{\ell=0}^{N-1} \underbrace{\cos\left(\pi\tfrac{2i+1}{2N}\left(kN+\ell-\tfrac{N}{2}\right)\right)}_{\substack{\text{repeats every 4 increments of } k \\ \text{sign changes every 2 increments of } k}} \quad h_{kN+\ell}\, x\big((m-k)N-\ell\big) \\ &=\; \sum_{k=0}^{15}\sum_{\ell=0}^{N-1} \underbrace{\cos\left(\pi\tfrac{2i+1}{2N}\left(\langle k\rangle_2 N+\ell-\tfrac{N}{2}\right)\right)}_{\text{repeats every 2 increments of } k}\; \underbrace{2\,(-1)^{\lfloor k/2\rfloor}h_{kN+\ell}}_{\text{analysis window}}\; x\big((m-k)N-\ell\big) \end{aligned}$$

Fig. 25 illustrates this process.

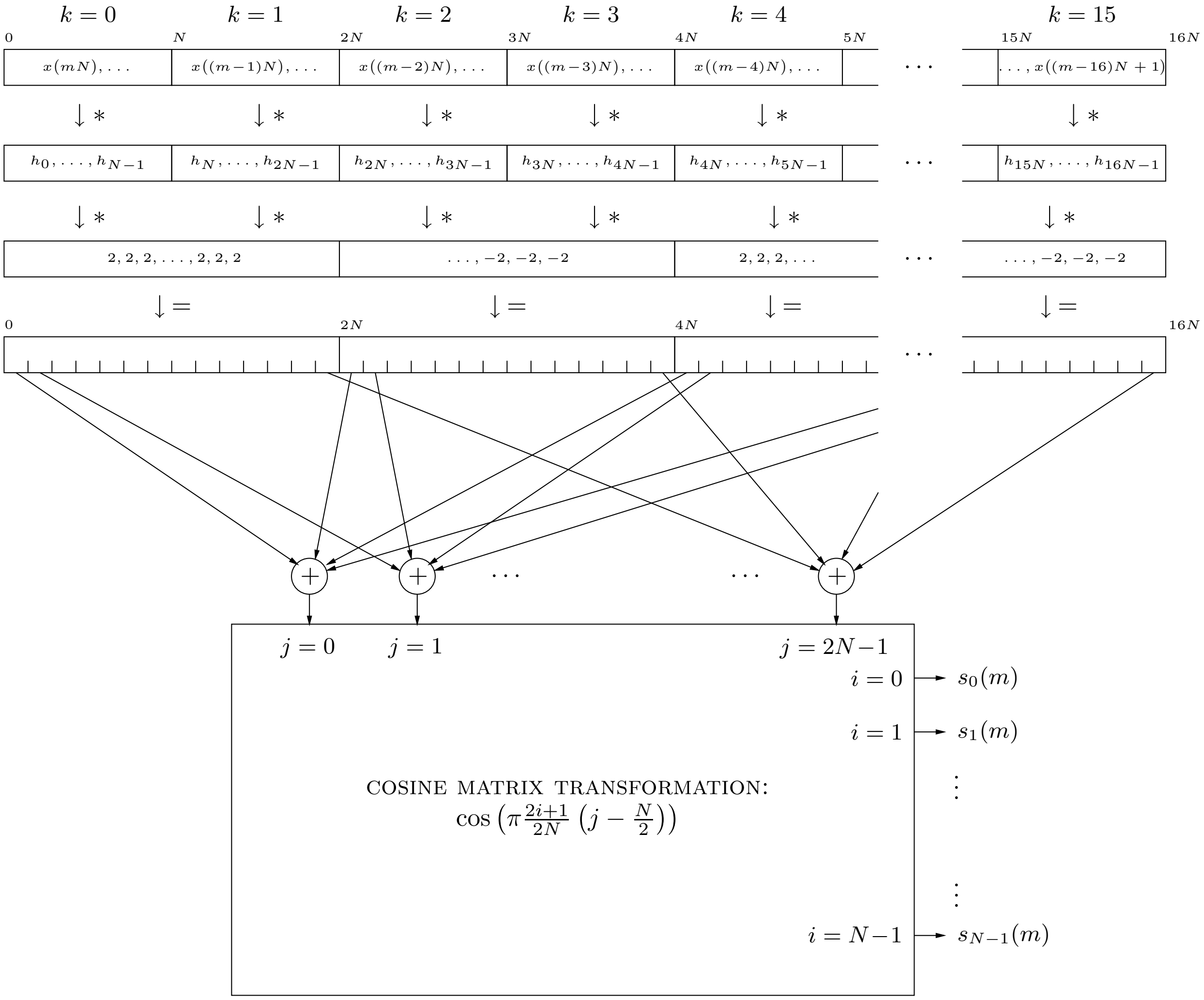


Figure 25: MPEG encoder filterbank implementation suggested in [1].

- Decoding: Here we derive the dencoder filterbank implementation suggested in the MPEG-2 standard [1]. Using $y_i(n)$ to denote the output of the i^{th} upsampler,

$$u_i(n) \quad = \quad \sum_{k=0}^{16N-1} \left[2\cos\left(\pi\tfrac{2i+1}{2N}\left(k+\tfrac{N}{2}\right)\right)h_k\right] y_i(n-k).$$

The input to the upsampler $s_i(m)$ is related to the output $y_i(n)$ by

$$y_i(n) \quad = \quad \begin{cases} s_i(n/N) & \text{when } n/N \in \mathbb{Z} \\ 0 & \text{else,} \end{cases}$$

so that

$$u_i(n) \quad = \quad \sum_{\{k:\,\frac{n-k}{N}\in\mathbb{Z}\}} \left[2\cos\left(\pi\tfrac{2i+1}{2N}\left(k+\tfrac{N}{2}\right)\right)h_k\right] s_i\left(\tfrac{n-k}{N}\right).$$

Lets write $n = mN + \ell$ for $0 \le \ell \le N-1$ and $k = pN + q$ for $0 \le q \le N-1$. Then due to the restricted ranges of ℓ and q,

$$\frac{n-k}{N} \;=\; m - p + \frac{\ell - q}{N} \;\in\; \mathbb{Z} \quad\Rightarrow\quad \ell \;=\; q.$$

Using these substitutions in the previous equation for $u_i(n)$,

$$u_i(mN+\ell) \;=\; 2\sum_{p=0}^{15} \cos\left(\pi \tfrac{2i+1}{2N}\left(pN + \ell + \tfrac{N}{2}\right)\right) h_{pN+\ell}\, s_i(m-p).$$

Summing $u_i(mN+\ell)$ over i to create $u(mN+\ell)$,

$$\begin{aligned} u(mN+\ell) \;&=\; 2\sum_{i=0}^{N-1}\sum_{p=0}^{15} \underbrace{\cos\left(\pi \tfrac{2i+1}{2N}\left(pN + \ell + \tfrac{N}{2}\right)\right)}_{\substack{\text{repeats every 4 increments of } p \\ \text{sign changes every 2 increments of } p}} \; h_{pN+\ell}\, s_i(m-p) \\ &=\; \sum_{p=0}^{15} \underbrace{2\,(-1)^{\lfloor p/2 \rfloor} h_{pN+\ell}}_{\text{synthesis window}} \sum_{i=0}^{N-1} \underbrace{\cos\left(\pi \tfrac{2i+1}{2N}\left(\langle p \rangle_2 N + \ell + \tfrac{N}{2}\right)\right)}_{=\begin{cases} \cos\left(\pi \frac{2i+1}{2N}\left(\ell + \frac{N}{2}\right)\right) & p \text{ even} \\ \cos\left(\pi \frac{2i+1}{2N}\left(\ell + N + \frac{N}{2}\right)\right) & p \text{ odd} \end{cases}} \; s_i(m-p) \end{aligned}$$

If we define

$$v_j(m) \;=\; \sum_{i=0}^{N-1} \cos\left(\pi \tfrac{2i+1}{2N}\left(j + \tfrac{N}{2}\right)\right) s_i(m) \quad \text{for} \quad 0 \le j \le 2N-1,$$

(note the range of j !) then we can rewrite

$$u(mN+\ell) \;=\; \sum_{p=0,2,\dots,14} (-1)^{\lfloor p/2 \rfloor} h_{pN+\ell}\, v_\ell(m-p) \;\;+\; \sum_{p=1,3,\dots,15} (-1)^{\lfloor p/2 \rfloor} h_{pN+\ell}\, v_{\ell+N}(m-p).$$

Fig. 26 illustrates the construction of $u(mN+\ell)$ using the notation

$$\mathbf{v}(m) \;=\; \begin{pmatrix} v_0(m) & \cdots & v_{2N-1}(m) \end{pmatrix}.$$

- *DCT Implementation of Cosine Matrixing:* As seen in Fig. 25 and Fig. 26, the filterbank implementations suggested by the MPEG standard require a cosine matrix operation that, if implemented using straightforward arithmetic, requires $32 \times 64 = 2048$ multiply/adds at both the encoder and decoder. Note, however, that the cosine transformations in Fig. 25 and Fig. 26 do bear a great deal of similarity to the DCT:

$$\begin{aligned} y_k \;&=\; \sqrt{\frac{2}{N}}\,\alpha_k \sum_{n=0}^{N-1} x_n \cos\left(\pi \tfrac{2n+1}{2N} k\right); \quad k = 0 \dots N-1, && (17) \\ &\text{for} \;\; \alpha_0 = 1/\sqrt{2}, \;\; \alpha_{k \ne 0} = 1, \\ x_n \;&=\; \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha_k\, y_k \cos\left(\pi \tfrac{2n+1}{2N} k\right); \quad n = 0 \dots N-1, && (18) \end{aligned}$$

which we know has a fast algorithm: Lee's 32×32 fast-DCT, for example, requires only 80 multiplications and 209 additions [7]. So how do we implement the matrix operation using the fast-DCT? A technique has been described clearly in [8], the results of which are summarized below.

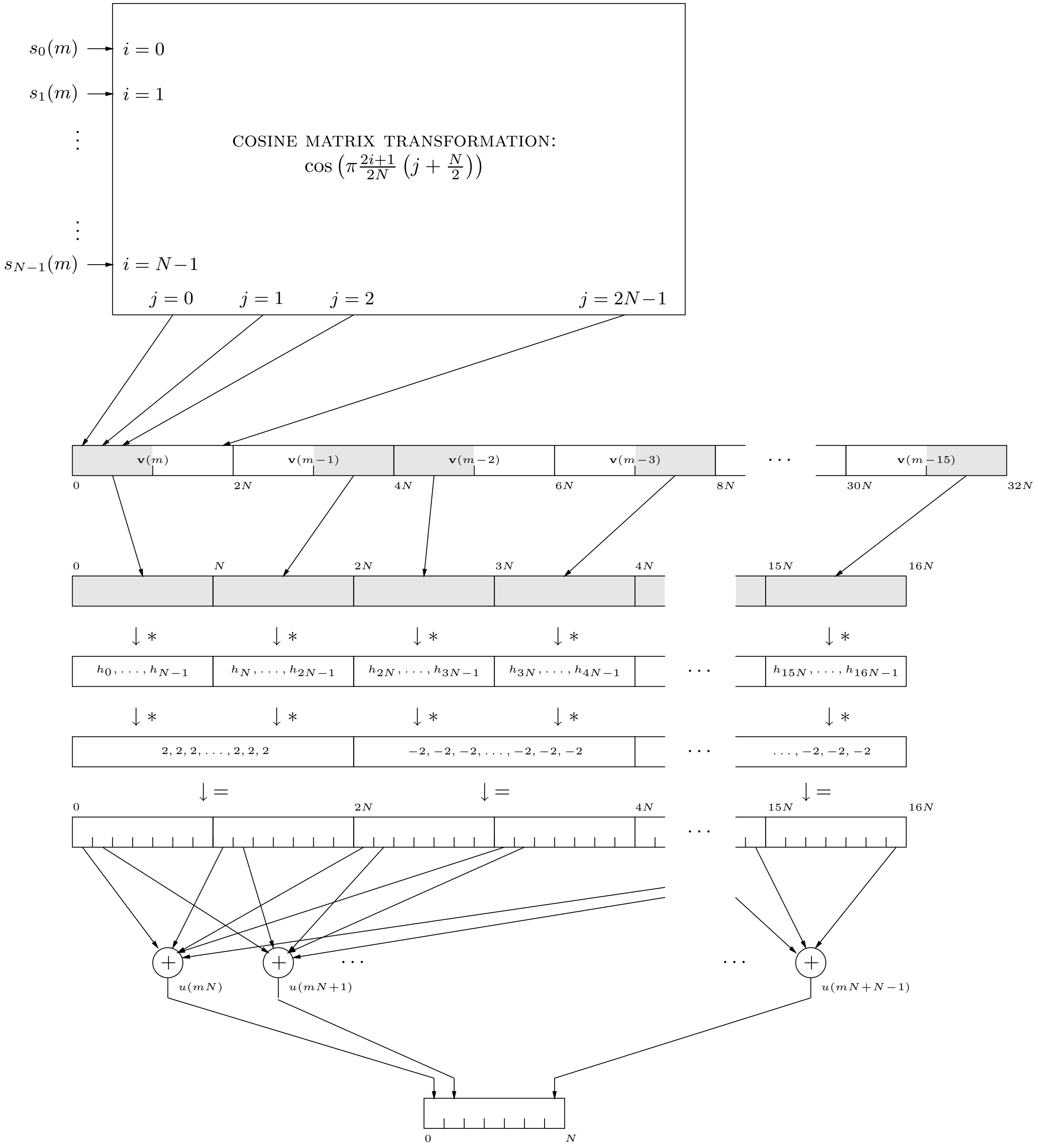


Figure 26: MPEG decoder filterbank implementation suggested in [1].

At the encoder, the matrix operation can be written

$$s_i(m) \quad = \quad \sum_{j=0}^{2N-1} \cos\left(\pi\tfrac{2i+1}{2N}\left(j-\tfrac{N}{2}\right)\right) w_j(m) \quad \text{for} \quad i=0,\ldots,N-1,$$

where $\{w_0(m), \ldots, w_{2N-1}(m)$ is created from $\{x(m), \ldots, x(m-16N+1)\}$ by windowing, shifting, and adding. (See Fig. 25.) It is shown in [8] that we can write

$$s_i(m) \;=\; \sum_{j=0}^{N-1} \cos\left(\pi \tfrac{2i+1}{2N} j\right) \bar{w}_j(m); \quad i = 0, \ldots, N-1, \tag{19}$$

where, for $N = 32$, $\{\bar{w}_j(m)\}$ is the following manipulation of $\{w_j(m)\}$:

$$\bar{w}_j(m) \;:=\; \begin{cases} w_{16}(m) & j = 0 \\ w_{16+j}(m) + w_{16-j}(m) & j = 1, 2, \ldots, 16 \\ w_{16+j}(m) - w_{80-j}(m) & j = 17, 18, \ldots, 31. \end{cases}$$

Compare (19) to the inverse DCT in (18).

At the decoder, the matrix operation can be written

$$v_j(m) \;=\; \sum_{i=0}^{N-1} \cos\left(\pi \tfrac{2i+1}{2N} \left(j + \tfrac{N}{2}\right)\right) s_i(m) \quad \text{for} \quad j = 0, \ldots, 2N-1,$$

where $\{v_0(m), \ldots, v_{2N-1}(m)\}$ are windowed, shifted, and added to compute $\{u(m)\}$. (See Fig. 26.) It is shown in [8] that, for $N = 32$, $\{v_j(m)\}$ can be calculated by first computing $\{\bar{v}_j(m)\}$:

$$\bar{v}_j(m) \;=\; \sum_{i=0}^{N-1} \cos\left(\pi \tfrac{2i+1}{2N} j\right) s_i(m); \quad j = 0, \ldots, N-1 \tag{20}$$

and rearranging the outputs according to

$$v_j(m) \;:=\; \begin{cases} \bar{v}_{j+16}(m) & j = 0, 1, \ldots, 15, \\ 0 & j = 16, \\ -\bar{v}_{48-j}(m) & j = 17, 18, \ldots, 47, \\ -\bar{v}_{j-48}(m) & j = 48, 49, \ldots, 63. \end{cases}$$

Compare (20) to the DCT in (17).

2 MDCT Filterbanks

- *Hybrid Filter Banks:* In more advanced audio coders such as MPEG "Layer-3" or MPEG "Advanced Audio Coding" (the details of which will be discussed later), the 32-band polyphase quadrature filterbank (PQF) is thought to not give adequate frequency resolution, and so an additional stage of frequency division is cascaded onto the output of the PQF. This additional frequency division is accomplished using the so-called "Modified DCT" (MDCT) filterbank. (See Fig. 27.)

- *Lapped Transforms:* The MDCT is a so-called "lapped transform." At the encoder, blocks of length $2Q$ which overlap by Q samples are windowed and transformed, generating Q subband samples each. At the decoder, the Q subband samples are inverse-transformed and windowed. The windowed output samples are overlapped with and added to the previous Q windowed outputs to form the output stream. Fig. 28 gives an intuitive view of the coding/decoding operation, while Fig. 29 and Fig. 30 specify the specific coder/decoder implementations used in the MPEG schemes.

- *Perfect Reconstruction:* Based on the cancellation of time-domain aliasing components, [9] and [10] show that the MDCT acheives perfect-reconstruction when window $\{w_n\}$ is chosen so that overlapped squared copies sum to one, i.e.,

$$1 \;=\; w_{n+Q}^2 + w_n^2 \quad \text{for} \quad 0 \leq n \leq Q-1.$$

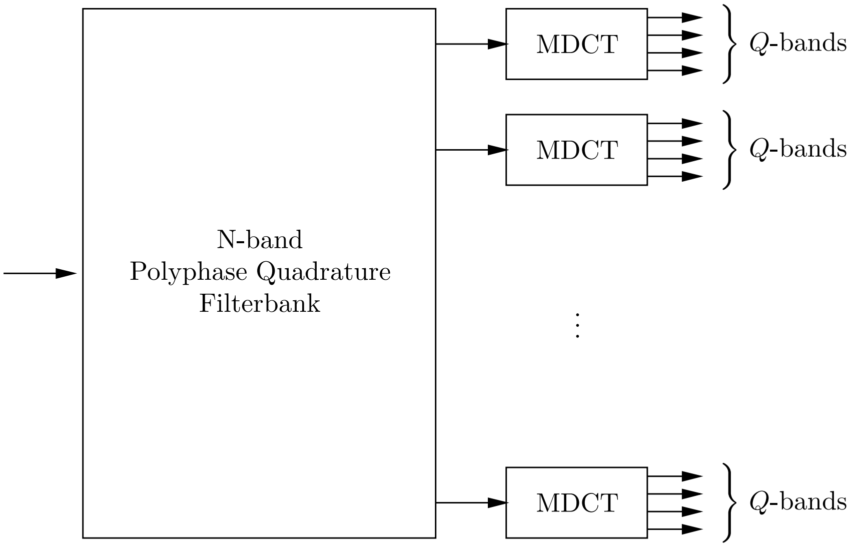


Figure 27: Hybrid filterbank scheme used in MPEG Layer-3 (where $N = 32$ and Q switches bewteen 6 and 18) and MPEG AAC (where $N = 4$ and Q switches between 128 and 1024).

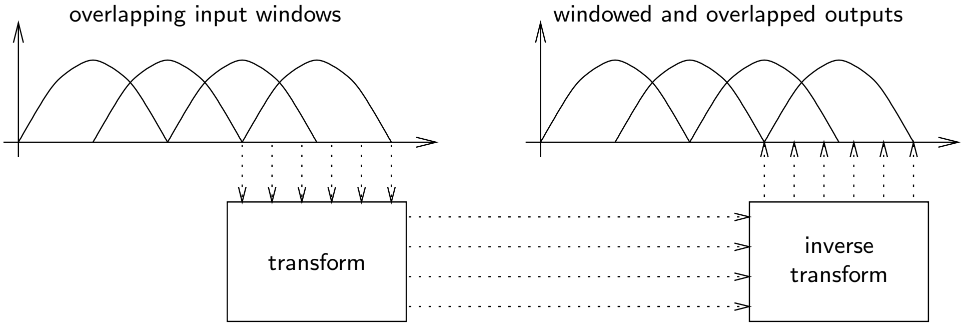


Figure 28: A lapped transform.

The "sine" window

$$w_n \;=\; \sin\left(\frac{\pi}{2Q}n\right) \quad \text{for} \quad 0 \le n \le 2Q-1$$

is one example of a window satisfying this requirement, and it turns out to be the one used in MPEG Layer-3.

- *Frequency Resolution:* With a window length that is only twice the number of transform outputs, we cannot expect very good frequency selectivity. But, it turns out that this is not a problem. In MPEG Layer-3, sine-window MDCTs appear at the outputs of a 32-band PQF where frequency selectivity is not a critical issue due to the limited frequency resolution of the human ear. In MPEG AAC, a 4-band PQF in conjunction with an optimized MDCT window function gives frequency selectivity just above that which current psychoacoustic models deem necessary [11].

- *Window Switching:* Larger values of Q lead to increased frequency resolution but decreased time resolution. Time resolution is linked to the following: error due to the quantization of one MDCT output is spread out over $\approx 2QN$ time-domain output samples. For signals of a transient nature, choosing QN too high leads to audible "pre-echoes." For less transient signals, on the other hand, the same value of QN might not be perceptible (and the increased frequency resolution might be

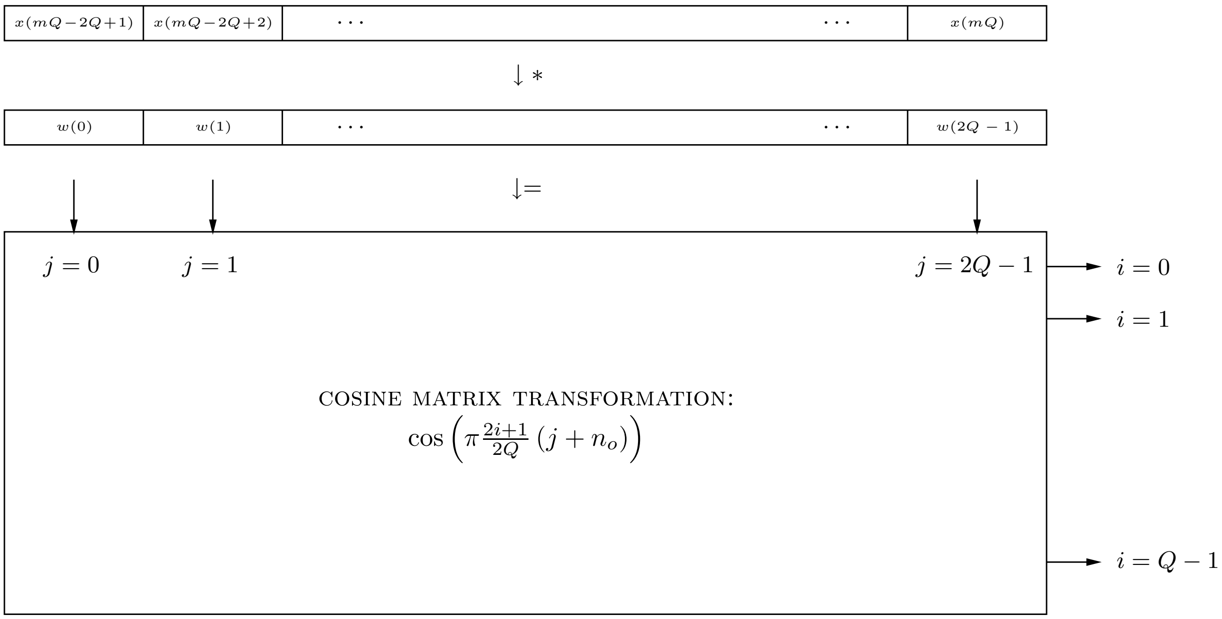


Figure 29: MDCT filterbank: encoder implementation.

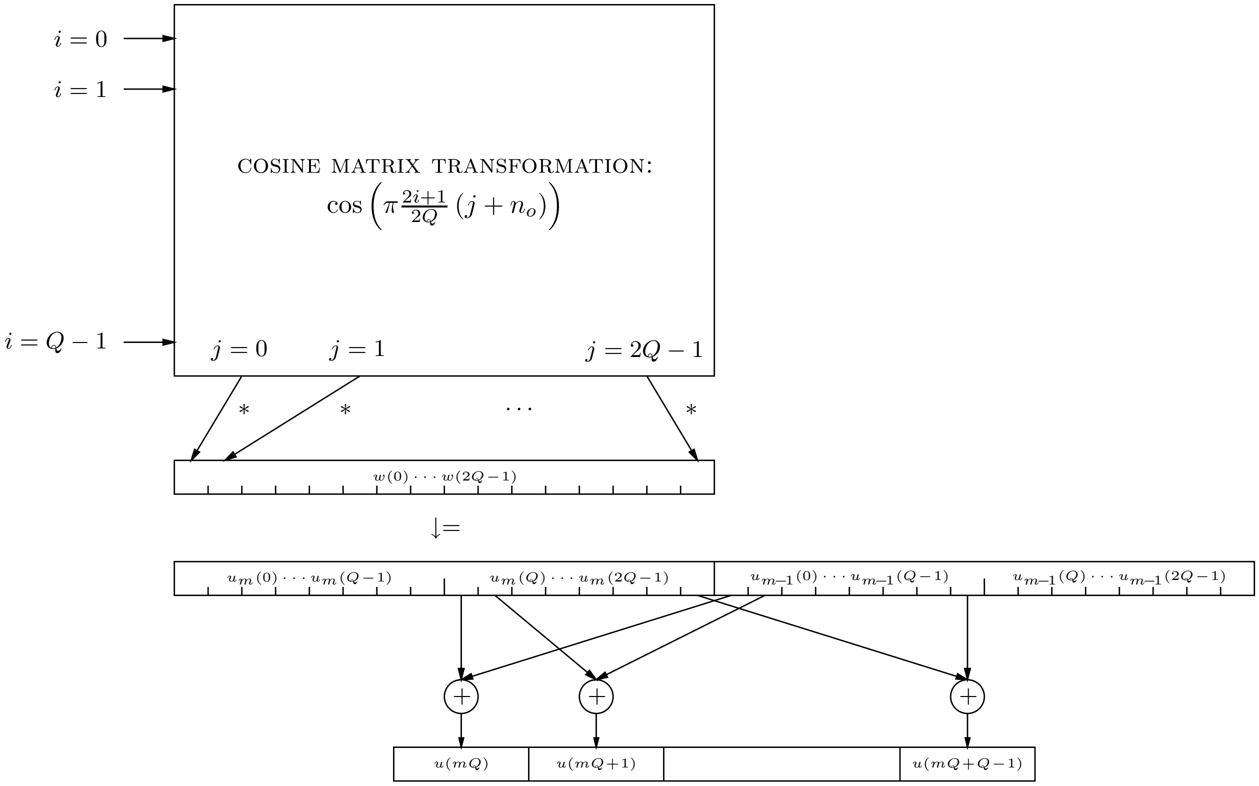


Figure 30: MDCT filterbank: decoder implementation.

very beneficial). Hence, most advanced coding schemes have a provision to switch between different time/frequency resolutions depending on local signal behavior.

In MPEG Layer-3, for example, Q switches between 6 and 18. This is accomplished using a sine window of length 36, a sine window of length 12, and intermediate windows which are used to switch between the long and short windows while retaining the perfect reconstruction property. Fig. 31 shows an example window sequence.

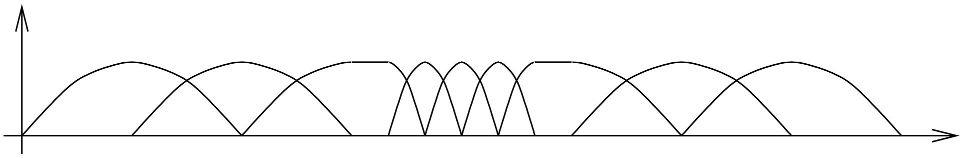


Figure 31: Example MDCT window sequence for MPEG Layer-3.

References

[1] ISO/IEC 13818-3, "Information Technology–Generic Coding of Moving Pictures and Associated Audio Information, Part 3: Audio," 1998.

[2] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[3] P.P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[4] H.V. Sorensen, D.L. Jones, M.T. Heideman, and C.S. Burrus, "Real-valued fast Fourier transform algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, pp. ??, June, 1987.

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

[6] J.H. Rothweiler, "Polyphase quadrature filters—A new subband coding technique," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (Boston, MA), pp. 1280-3, 1983.

[7] B.G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1243-5, Dec. 1984.

[8] K. Konstantinides, "Fast subband filtering in MPEG audio coding," *IEEE Signal Processing Letters*, vol. 1, no. 2, pp. 26-8, Feb. 1994.

[9] J.P. Princen, A.W. Johnson, and A.B. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (??), pp. 2161-4, 1987.

[10] J.P. Princen and A.B. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1153-61, Oct. 1986.

[11] M. Bosi, et al., "ISO/IEC MPEG-2 Advanced Audio Coding," *Journal of the Audio Engineering Society*, vol. 45, no. 10, pp. 789-812, Oct. 1997.