15

Time–Frequency-domain Processing and Coding of Audio

A common trend in the field of audio is to process the audio signal in the time–frequency domain. In other words, the input audio signal is divided into a number of frequency bands which are processed separately and also depending on time. The aim in such processing is, for example, data compression, audio effects, or the enhancement of audio quality. The benefit of time–frequency processing in such tasks is that the structure of human hearing mechanisms is based on similar time–frequency analysis of the ear canal signals. Already many applications, such as the perceptual coding of audio, take advantage of the human hearing resolution in the time–frequency domain. An emerging field is multi-channel and spatial applications utilizing time–frequency processing.

15.1 Basic Techniques and Concepts for Time–Frequency Processing

The use of time–frequency transforms to visualize audio signals was already touched on in Section 3.2.6 on page 53. This chapter elaborates on the techniques and introduces some phenomena, concepts, and issues related to the processing of audio in the time–frequency domain. We will describe the time–frequency processing methods first using the concepts of frame-based analysis, and second using the concepts of downsampled filter banks.

15.1.1 Frame-Based Processing

Many time–frequency-domain audio techniques are implemented such that an input signal(s) is first divided into overlapping time frames, after which the frames are processed separately. We will first review the basic concepts and techniques in such approaches, which will then make it easier to understand the methods better.

Let us first denote the original signal as $x_{\mathrm{a}}(n_a)$, where n_a is a whole number running from zero to the number of samples R in the audio data in a file. The signal is divided into *frames*, which are short portions of the signal, typically with a length between 1 ms and 100 ms, as shown in Figure 15.1.

Communication Acoustics: An Introduction to Speech, Audio, and Psychoacoustics, First Edition.
Ville Pulkki and Matti Karjalainen.

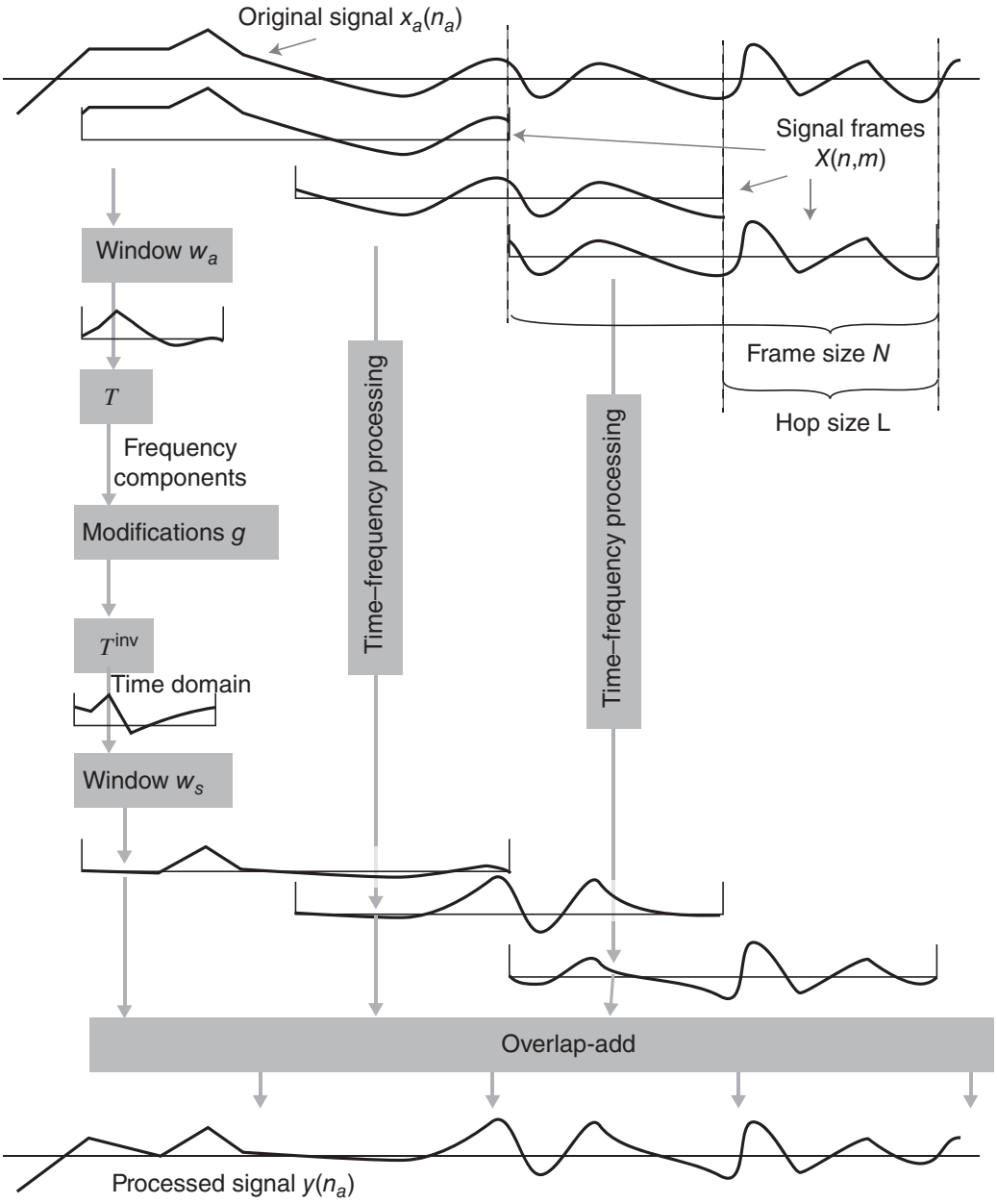

Figure 15.1 The principle of frame-based time–frequency-domain processing of audio. In this schematic example, the modifications of the spectral components implement a high-pass filter.

For each frame index m, a set of N samples is chosen from the input signal so that $x(n, m) = x_{\mathrm{a}}(mL + n - 1)$ where $n \in [0, N - 1]$ is the time index within the frame, and L is the *hop size*, indicating by how much the frame is advanced in x_{a}. The difference between x_{a} and x is that x_{a} contains the whole data to be processed and x is a short portion of x_{a} that is currently being processed.

A frame of the signal, which may be windowed, is transformed into its frequency-domain representation $X(k, m)$ as

$$X(k, m) = \mathcal{T}(w_{\mathrm{a}}(n)x(n, m)), \tag{15.1}$$

where $k = 0, \ldots, N-1$ is the frequency index, $w_{\mathrm{a}}(n)$ is the analysis time window, and $\mathcal{T}$ is the transformation operation. $X(k,m)$ is a real- or complex-valued sample that represents the original signal in the time–frequency domain. $X(k,m)$ is often also called a *frequency bin*; however, note that the term in some contexts may also mean the spacing of the samples in frequency.

The frequency-domain representation $X(k,m)$ is transformed back into the time domain by

$$\hat{x}(n,m) = w_{\mathrm{s}}(n)\mathcal{T}^{\mathrm{inv}}(X(k,m)), \tag{15.2}$$

where $w_{\mathrm{s}}(n)$ is the synthesis time window and $\mathcal{T}^{\mathrm{inv}}$ is the inverse transformation operation. Assuming $L = N/2$, the design of $w_{\mathrm{a}}(n)$ and $w_{\mathrm{s}}(n)$ is typically such that it preserves the amplitudes across the overlapping frames:

$$w_{\mathrm{a}}(n)w_{\mathrm{s}}(n) + w_{\mathrm{a}}(n+L)w_{\mathrm{s}}(n+L) = 1, \qquad \text{for } n = 0, ..., (N/2-1), \tag{15.3}$$

which is known as the constant overlap-add (COLA) (Smith, 2011). An example fulfilling Equation (15.3) is to design both $w_{\mathrm{a}}(n)$ and $w_s(n)$ as sequences which have entries that are the square roots of the corresponding entries of a Hann window sequence.

The final synthesized signal is computed as the sum of all signals

$$y\left(n_a\right) = \sum_m \hat{x}\left(n_{\mathrm{a}} - mL, m\right), \tag{15.4}$$

where $\hat{x} = 0$ when $n_a - mL < 0$ or $n_a - mL \geq N$, and $0 \leq n_a < R$ runs through all samples in the original data. The summing operation thus takes subsequent frames of audio and adds them together, overlapping in time in a process called *overlap-add* (OLA), as also shown in Figure 15.1.

Audio coding applications aim to get the approximation $y(n_a) \approx x_{\mathrm{a}}(n_a)$, striving to reconstruct the original signal exactly or to make any error due to quantization to have the least perceptual impact. In many other applications, such as those involving audio enhancement of any kind, the signal has to be modified somehow, and Equation (15.2) must thus be rewritten as

$$\hat{x}(n,m) = w_{\mathrm{s}}(n)\mathcal{T}^{\mathrm{inv}}(g(k,m)X(k,m)), \tag{15.5}$$

where $g(k,m)$ are frequency- and/or time-dependent real or complex gain values. The values of $g(k,m)$ affect the magnitudes and/or phases of corresponding frequency components, which, in principle, enables arbitrary filtering operations. However, depending on the applied transformation, time–frequency processing can be prone to aliasing artefacts, which must be avoided in the processing, as discussed below.

15.1.2 Downsampled Filter-Bank Processing

Various *downsampled filter banks* are used in time–frequency processing of audio. As briefly mentioned in Section 3.2.7, a filter bank is an array of band-pass filters which divide a broadband time-domain input signal into time-domain signals with narrowband frequency content. The process of downsampling in the context of filter banks is now discussed.

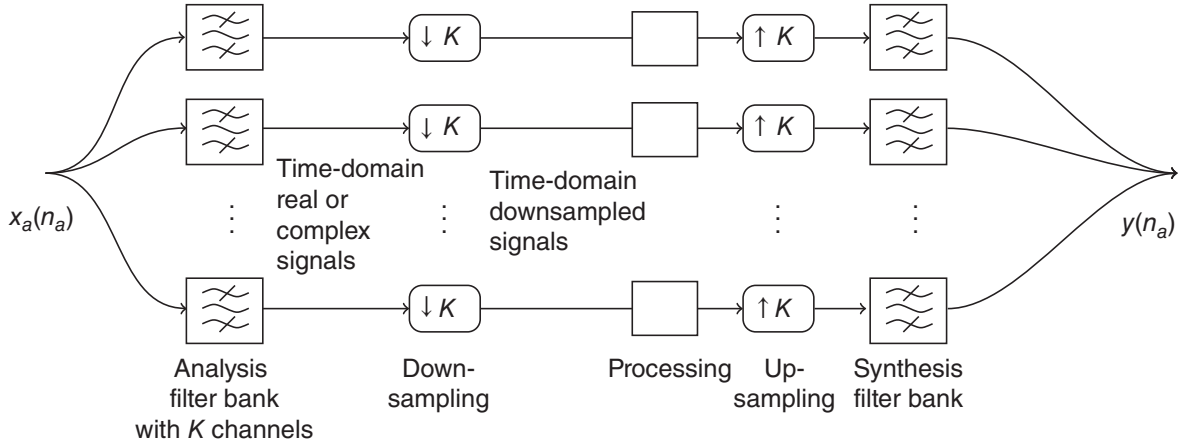

Figure 15.2 A schematic diagram of time–frequency processing of audio utilizing a downsampled filter bank.

A schematic diagram of a downsampled filter bank is shown in Figure 15.2. The input signal $x_{\mathrm{a}}(n_a)$ is divided into K narrowband frequency bands using real- or complex-valued analysis filters, and the bands are downsampled by the same factor K. Downsampling is a procedure that is necessary for efficiency. Firstly, downsampling reduces the data rate for the processing or coding in the time–frequency domain. Without downsampling, dividing a signal into K frequency bands multiplies the number of data points per second by the factor K. Downsampling by a factor of K means that only every Kth sample is preserved while the others are discarded. This changes the sampling rate and causes all frequencies beyond the new Nyquist frequencies $\pm F_{\mathrm{s}}/(2K)$ to fold back to between this interval.

Downsampling discards excessive data by an amount where the core signal information is still preserved. Secondly, the computational efficiency of the transform operation itself can be optimized when the output channels are downsampled. It is typical of symmetric downsampled structures, such as that in Figure 15.2, that they can be implemented with efficient DSP techniques. Similarly, the fast Fourier transform (FFT) is an efficient implementation of the discrete Fourier transform (DFT). Although various implementations can be realized, the working principles and properties of such filter banks can be considered in terms of this basic schematic diagram, as it is easy to comprehend.

The mechanism of how downsampling preserves the signal information with narrowband signals is discussed next. Discrete signals with the sampling rate F_{s} can only represent frequencies below the Nyquist frequency $f = F_{\mathrm{s}}/2$, which is the same as the normalized angular frequency $\omega = \pi$ radians per sample. This property is observed in Equation (3.17b), where the tone sequences with frequencies beyond π radians per sample produce spectral coefficients that are equivalent to specific tone sequences with frequencies within the interval $\pm\pi$.

Upsampling is applied to re-map the frequencies to their original intervals. When upsampling by a factor K, a sequence of $K - 1$ zeros is appended, or concatenated, after every sample, in which case the repetitive nature of the spectrum is actualized as a set of duplicate frequency components in the new sampling interval. The spectral effects of downsampling and upsampling are illustrated in Figure 15.3.

The original spectral content is thus preserved, but the aliased components are spread all over the spectrum. In the design of time–frequency transform methods, this effect must be taken into account, and the aliased components must be removed either with a synthesis filter

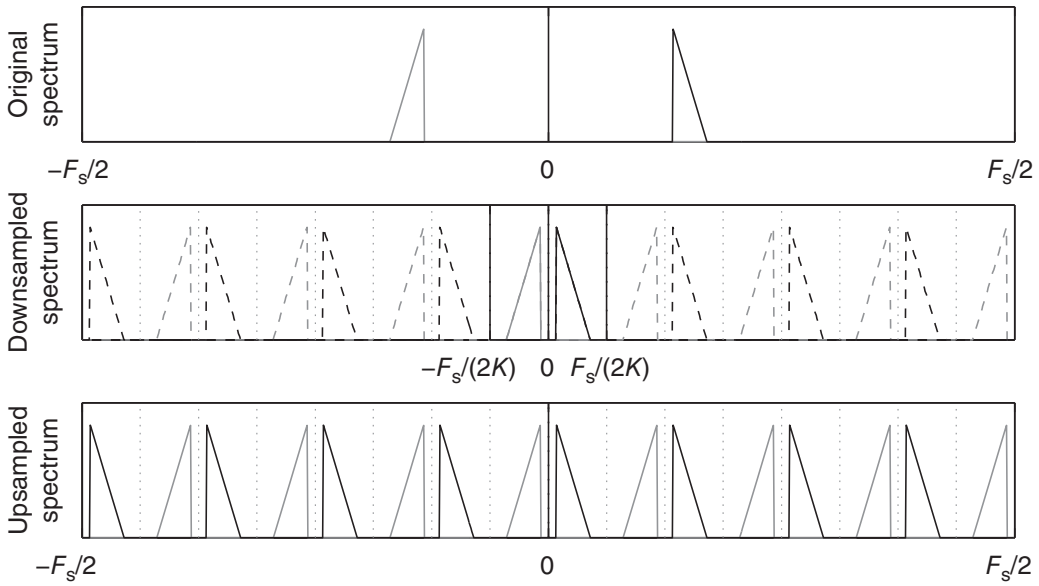

Figure 15.3 A signal with a band-limited spectrum is first downsampled (by the factor 8). The frequencies are folded to the limits set by the new sampling frequency. After upsampling by the same factor, the frequencies are replicated repetitively. They contain the original spectrum and a set of aliased spectra. Courtesy of Juha Vilkamo.

or through interference between frequency channels or time frames. In the downsampled filter bank processing shown in Figure 15.2, the aliased components are removed with a narrowband synthesis filter after upsampling, and the remaining channels are summed to obtain the original or modified signal. The modifications can again be implemented by some processing of the sub-band signals, such as by multiplication with real or complex values.

15.1.3 Modulation with Tone Sequences

The filter-bank processing discussed previously included specific filters for analysis and synthesis, but the design of the filters was not discussed at all. *Modulation* is often used as a tool in the design of the analysis and synthesis filter banks for time–frequency transforms, as will be shown in Section 15.2.4, and so the basics of modulation are reviewed in this section.

The modulation of a signal sequence $x(n)$ with a tone sequence refers to an operation in which each of its samples is multiplied by the corresponding sample from the tone sequence. In general, the signal sequence can be any signal, and in this context one may consider it to be the impulse response of an FIR filter. The real-valued modulation is

$$x_{\mathrm{modR}}(n) = x(n)\cos(\omega n) = x(n)\frac{e^{\mathrm{j}\omega n} + e^{-\mathrm{j}\omega n}}{2}, \tag{15.6}$$

and the complex-valued modulation is

$$x_{\mathrm{modC}}(n) = x(n)e^{\mathrm{j}\omega n}. \tag{15.7}$$

When modulated, the spectrum of the signal is centred with respect to the frequency of the modulator. A real modulator produces positive and negative frequencies, while a complex

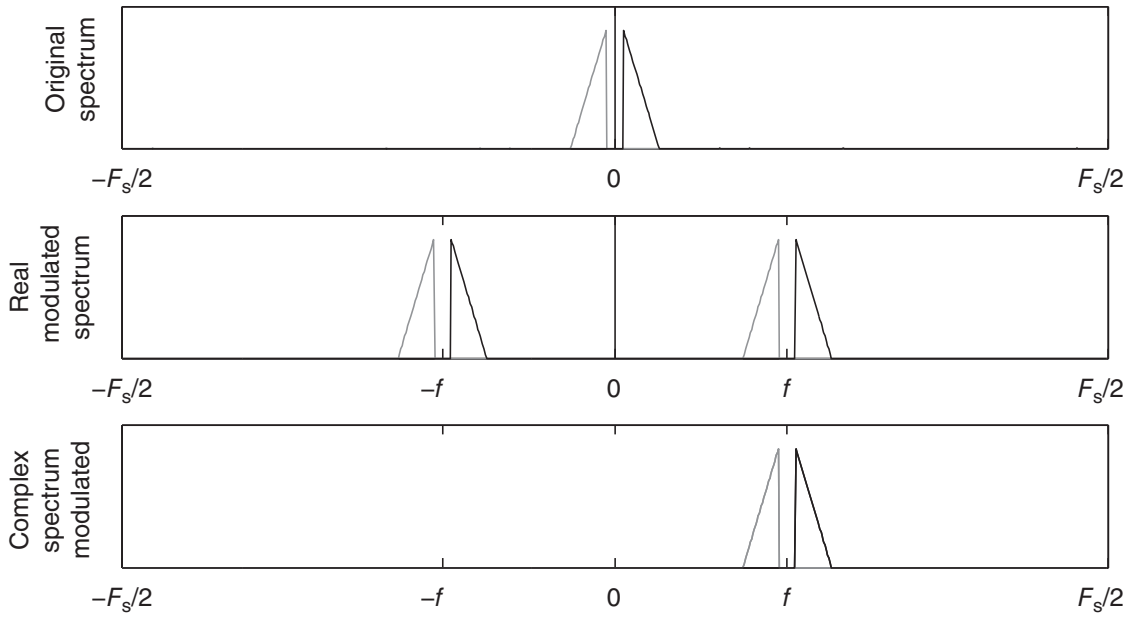

Figure 15.4 When a signal is modulated by a tone sequence with frequency $f = \omega F_S/(2\pi)$, its spectrum is centred with respect to that frequency. A real-valued modulator produces positive and negative frequencies, while a complex-valued modulator produces only positive frequencies. Courtesy of Juha Vilkamo.

modulator produces only one of the polarities, usually the positive frequencies, as shown in Figure 15.4. In time–frequency processing, modulation is applied to the impulse response of an FIR filter. The band-pass filters for time–frequency transforms can be designed by shifting the spectrum of a low-pass FIR filter using modulation to the desired centre frequencies.

Note that modulation as defined here is different from the amplitude modulation defined in Section 3.1.2. The modulation used in this chapter is real- or complex-valued sinusoid, in contrast to amplitude modulation, where the sinusoidal modulator is defined to be positive-valued – a sinusoid offset by a positive value.

15.1.4 Aliasing

Aliasing is a non-linear effect in digital signal processing by which signal components appear where none existed in the original signals. The effect is similar to non-linear distortion, but the term 'aliasing' is used to mean those typical distortions created by sampling. In time–frequency DSP, aliasing must be avoided, cancelled, or suppressed sufficiently to avoid perceived distortions in the sound.

Aliasing may occur in signals that are processed considerably in the frequency–domain. Aliasing components emerge especially if the applied time–frequency transform relies significantly on signal cancellation properties of the different samples in the time–frequency domain, which is a typical feature of non-redundant time–frequency representations. *Non-redundant transforms* are transforms that provide the lowest possible amount of data points per second without losing any signal information. A transform is *redundant* if some of the information it produces is repeated more than once, which may be necessary to avoid aliasing if the signal is processed in the frequency domain.

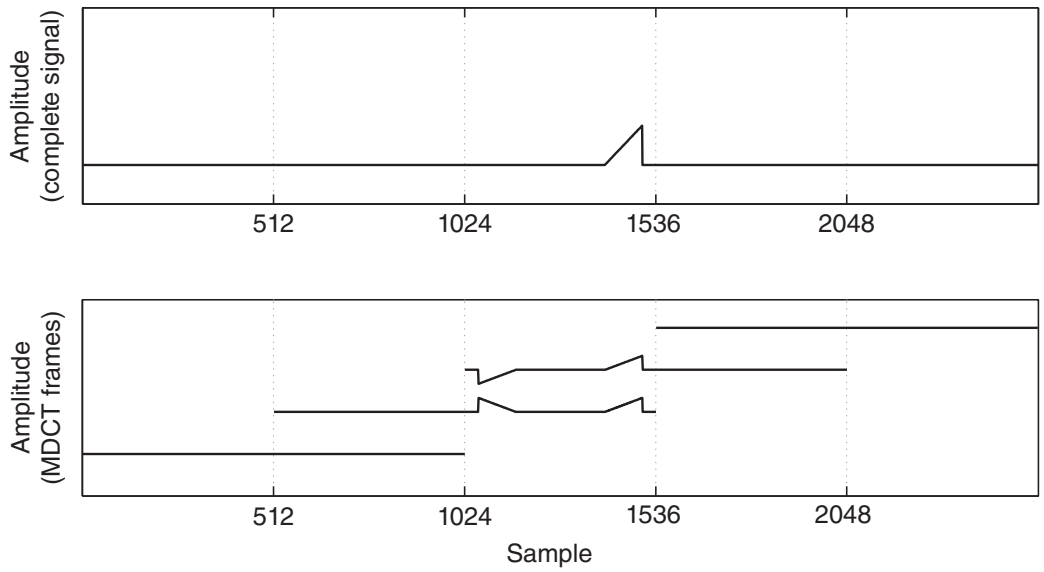

Figure 15.5 Illustration of a condition where time aliasing can occur during adaptive processing with frame-based time–frequency processing of audio. The upper graph is the original signal, and the lower one shows the time-domain counterparts of the modified discrete cosine transforms (MDCTs), described in Section 15.2.3, of frames of length 1024 samples each. The time-domain aliasing component is cancelled if the signals are unprocessed. In this illustration, the analysis and the synthesis windows of the MDCT are rectangular. Courtesy of Juha Vilkamo.

Consider Figure 15.5, where the upper graph shows the original signal and the lower one shows four subsequent time frames reproduced. Note that the second and third frames have a triangular component near sample 1024 that does not correspond to the original signal and is cancelled out when the frames are overlap-added. If the level of the second or third frame is modified in the processing of frequency-domain signals, the triangular component of the second frame produces a clearly audible aliasing component.

Similarly, a potential case of frequency aliasing is shown in Figure 15.6. A single frame of a static sinusoid is transformed into time-domain sub-bands using a filter bank. An added sinusoid of slightly higher frequency occurs in the two sub-bands with the highest energy, but the added sinusoids are exactly out of phase. If and when the sub-bands are added together during synthesis, the added sinusoids will cancel each other, removing the potential aliasing component. If, however, the gains g for the sub-bands differ, the cancellation is not perfect, and an aliasing component will remain. Note that these were examples of non-redundant transforms. Further on in this chapter, more robust transforms, with some redundancy, will also be discussed.

15.2 Time–Frequency Transforms

This section describes a few *time–frequency transforms* that are commonly used in the audio industry. The following concepts are applied to discuss the properties of filter banks:

- *Critical sampling* is a property whereby the combined sampling rate of the transformed frequency bands is the same as that of the original band. Critical sampling is exploited to minimize redundancy, which is a relevant and desirable property for audio coding. An

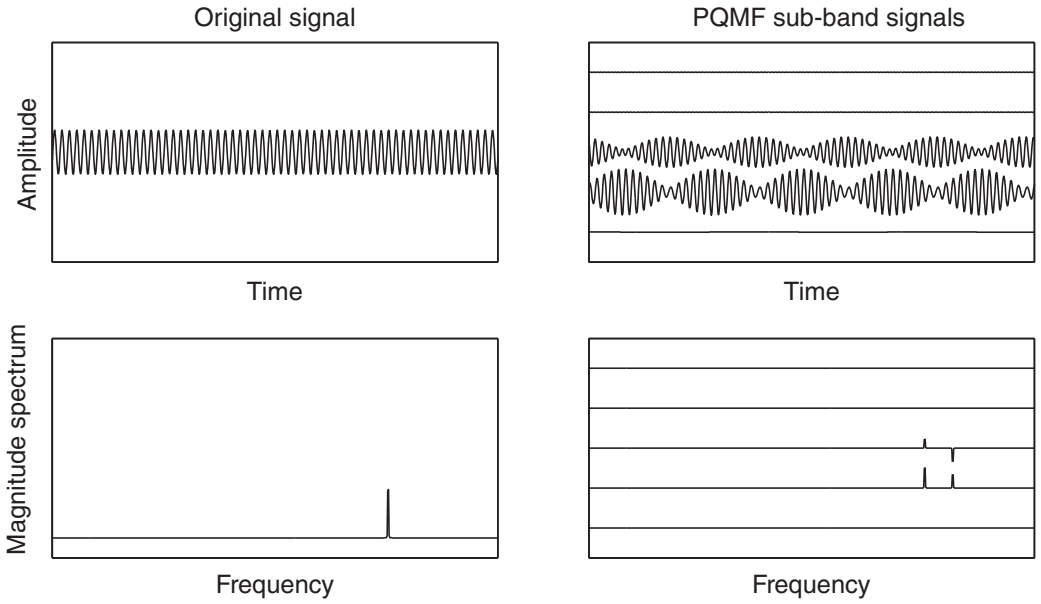

Figure 15.6 Illustration of a condition where frequency aliasing can occur during adaptive processing with filter-bank processing. The original tone and its spectrum are shown in the left column, and the time-domain counterparts of the pseudo-quadrature mirror filter (PQMF) bank sub-band signals, described in Section 15.2.4, and their spectra are shown in the right column. The right-hand peaks in the PQMF band spectra show the aliased frequency components, which are antiphasic with respect to each other in the sub-bands, visualized as polarity-up or polarity-down. The antiphasic peaks are cancelled if the sub-band signals are not processed. The complementary nature of the band signals is also visible in their waveforms. Courtesy of Juha Vilkamo.

example of a critically sampled transform is one that decomposes a real-valued signal using the sampling rate F_s into K real-valued bands, each with a sampling rate of F_s/K.

- *Oversampling* is a property whereby the combined sampling rate of the transformed bands is higher than that of the original signal. This property is typical of transforms intended for robust signal adjustments. Consider the previous example with the sampling rate of F_s and the decomposition into K bands, each with a sampling rate of F_s/K. However, if the produced frequency-band signals are complex-valued, the transformed signal is oversampled by a factor of two. An example of such a transform is the complex-modulated QMF described in Section 15.2.5.
- *Perfect reconstruction* means that the original signal waveform is retrieved exactly after the inverse transform, if the signals are not otherwise altered during the process.
- *Near-perfect reconstruction* implies that the original signal can be retrieved with a high degree of accuracy, but not exactly. The distortion produced by the transform itself is present but negligible in practice. The difference in properties of perfect and near-perfect reconstruction is mostly descriptive when it comes to audio.

15.2.1 Short-Time Fourier Transform (STFT)

The *short-time Fourier transform* (STFT) is an approach often applied in the field of audio. In STFT, an analysis window sequence $w_a(n)$ is applied to a signal sequence $x(n)$, after which

the result is transformed into frequency bands using the DFT in Equation (3.17b). For a single frame of OLA processing, STFT analysis can be expressed as

$$X(k) = \sum_{n=0}^{N-1} w_{\mathrm{a}}(n)x(n)e^{-\mathrm{j}2\pi kn/N}, \tag{15.8}$$

and the inverse STFT, with the synthesis window sequence $w_{\mathrm{s}}(n)$, as

$$y(n) = \frac{1}{N} w_{\mathrm{s}}(n) \sum_{k=0}^{N-1} X(k)e^{\mathrm{j}2\pi kn/N}. \tag{15.9}$$

In practice, the STFT is computed using the fast Fourier transform (FFT). Due to the windowing and the overlapping frames, the representation is oversampled. STFT can be configured to be a perfect reconstruction transform by designing the windowing functions as in Equation (15.3).

Often, the STFT processes only the positive, DC, and Nyquist frequencies. Prior to the inverse FFT, the positive frequencies are complex-conjugate mirrored to the corresponding negative frequencies so as to obtain real-valued output signals, or, alternatively, an inverse FFT that assumes a real-valued output is applied.

An illustrative feature of the STFT representation for time–frequency adaptive processing is that each STFT sample, due to the definition of the DFT, corresponds to a circularly continuous tone (see Figure 15.7). If the forward transform involves windowing or zero padding prior to the FFT, the STFT components are simply phase-organized so that the frequency band signals cancel or amplify each other to form the envelope of the window and zero padding.

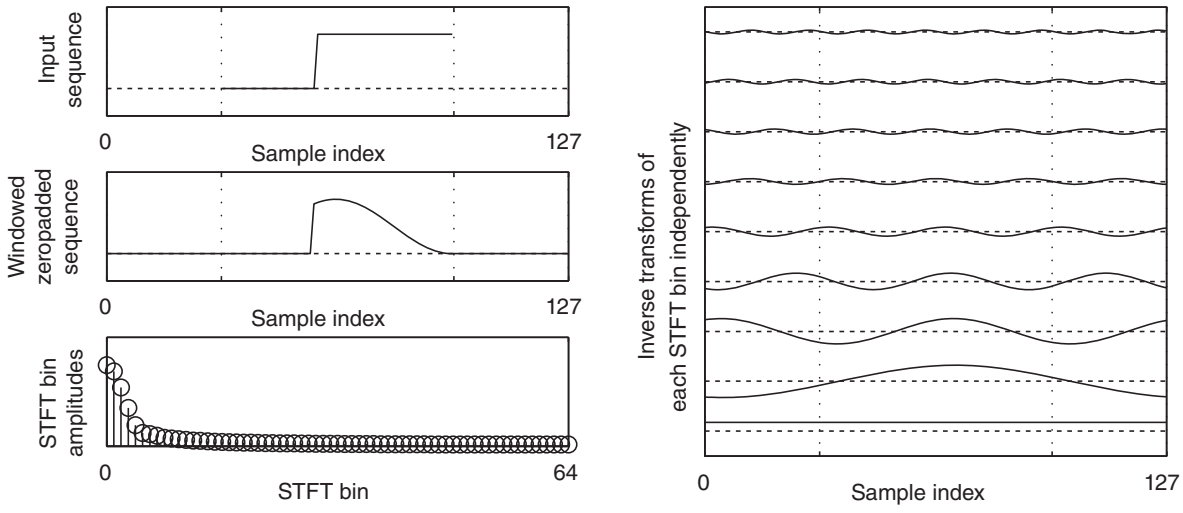

Figure 15.7 STFT analysis of a Hann-windowed and zero-padded step function. The STFT bins $X(k)$, $k \in [0, 1, \ldots, 63]$ correspond to circular tones through the entire window, but organized in phase and amplitude such that when summed, the original signal shape is obtained. Significant magnitude or phase processing removes the phase organization, which potentially spreads the waveform across the whole frame. In a typical implementation, such effects are suppressed by a synthesis window. Courtesy of Juha Vilkamo.

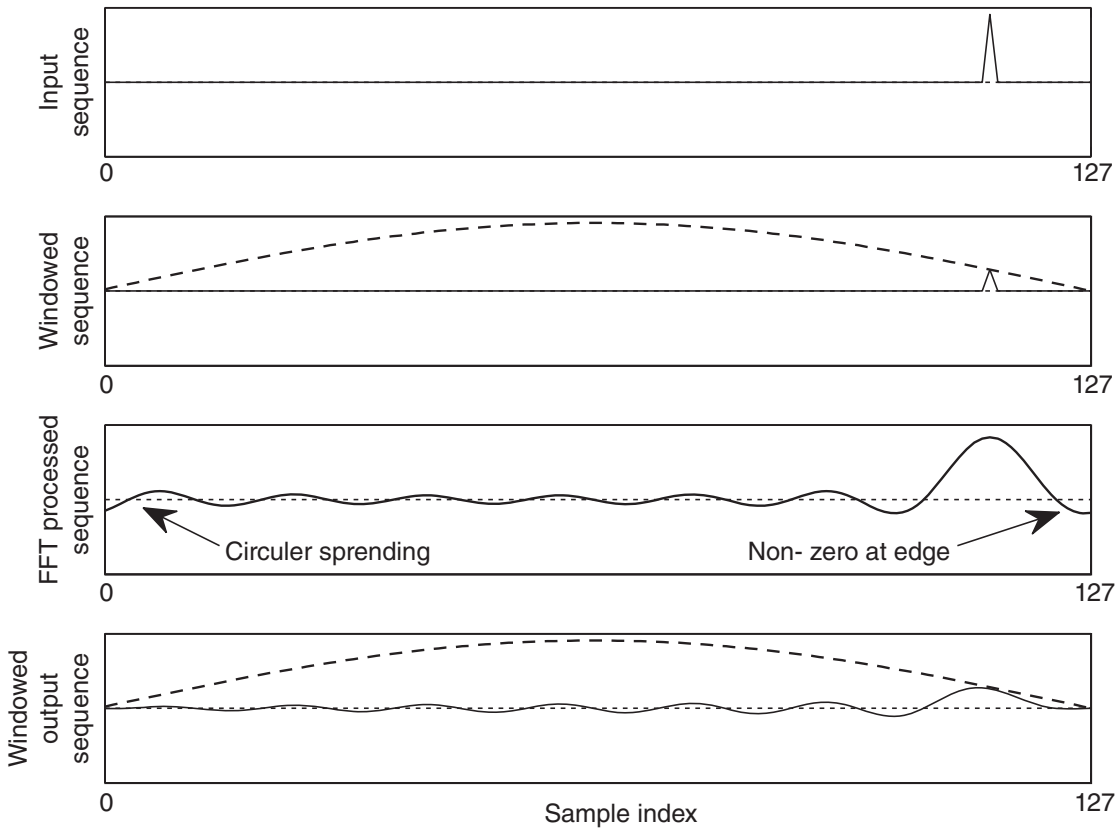

Figure 15.8 An example of circular effects that can occur during STFT processing. The input is a unity impulse. The signal is windowed, in this case using a square-root Hann window, and then processed using a zero-phase low-pass filter. The resulting signal is aliased in time. The synthesis filter suppresses the effects of the edge discontinuity. Courtesy of Juha Vilkamo.

Zero padding refers to the adding of a certain number of zeros before and/or after the windowed frame prior to computing the frequency transform.

When the STFT data are arbitrarily modified, the time-domain representation of the same data no longer has the original windowed shape. Furthermore, due to the circular representation, the values at the edges of the frame may deviate from zero, as shown in Figure 15.8. The non-zero edges then produce wideband transient artefacts in the output signal, and the conventional method for suppressing such effects is to apply a synthesis window, as shown in Figure 15.8. The synthesis window forces the time-domain signal values to zero at the start and at the end of the frame. The suppressed components of the signal are basically lost, which again may cause quality degradation.

15.2.2 Alias-Free STFT

A variant of STFT processing has been proposed, in which potential aliasing components can be avoided (Vickers, 2012). With a compromise in the form of an increased CPU load, it is possible to process the STFT bins so that the processing is equivalent to a time-domain convolution with an adaptive FIR filter. Such a linearized approach avoids the circular effects of the

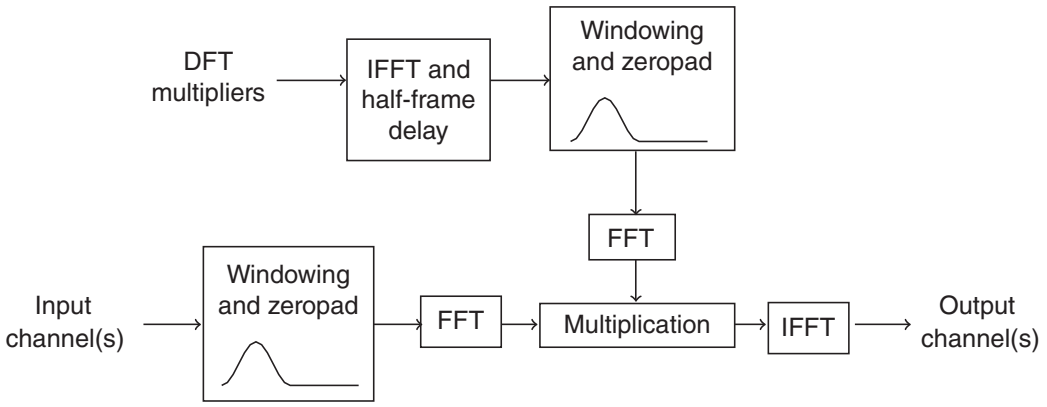

Figure 15.9 Alias-free STFT processing. The multipliers at the individual STFT lines are pre-processed: first, they are transformed into the time domain, and the response is shifted circularly by a half-frame delay, which centres the response to the middle of the window. The coefficients are then time-windowed so that abrupt window edges do not affect the spectrum. The response is appended with zero padding, and the result is transformed using FFT to obtain the processed STFT-domain multipliers. The combined length of the zero padding must be at least the combined length of the non-zero parts to obtain alias-free STFT processing. No synthesis window is necessary in this approach.

traditional STFT. The method involves extending the signal frame with zero padding and also pre-processing the complex processing multipliers (the STFT-domain magnitude and phase operators) in such a way that the non-zero part of their time-domain counterpart is limited in length. The steps for processing the multipliers involve applying the inverse FFT, windowing, zero padding, and FFT prior to applying them to the STFT signal frame, as shown in Figure 15.9. The combined length of the non-zero parts of the time-domain counterparts needs to be at most the same as the combined length of the zero-padding parts to avoid the circular convolution effects completely.

15.2.3 Modified Discrete Cosine Transform (MDCT)

The *modified discrete cosine transform* (MDCT) is a perfect reconstruction, real-valued, and critically sampled transform. The MDCT is widely applied in audio coding due to its non-redundancy and its property of representing narrowband signals with a relatively small number of prominent spectral coefficients (Geiger *et al.*, 2001; Neuendorf *et al.*, 2013). However, the MDCT is not designed to be robust for considerable signal adjustments in the time–frequency domain.

The MDCT analyses the signal in half-overlapping frames; that is, for windows of N samples the hop size is $N/2$ samples. The frequency partials of the MDCT represent sinusoids that are odd symmetric in the first half and even symmetric in the second half of the frame, as seen in Figure 15.10. For this reason, the individual MDCT frames generate temporal aliasing, which is countered by the opposite effect of the next frame. The real-valued amplitudes of the MDCT representation determine the amplitudes of the corresponding frequency components within the frame. Figure 15.5 is also an illustration of the functioning of the transform.

Like the STFT, the MDCT formulates a correlation of a windowed signal sequence with a set of tone sequences. With the MDCT, the tone sequences are real-valued. The MDCT can be

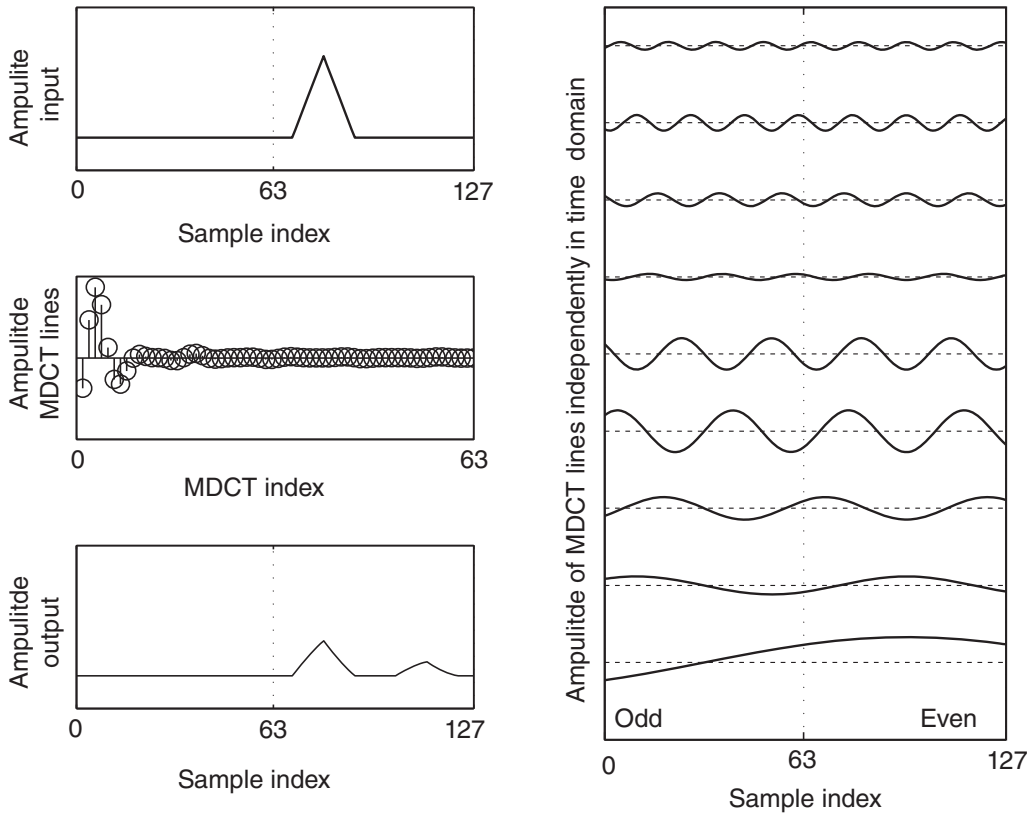

Figure 15.10 The MDCT decomposition of a triangular impulse using the window function in Equation (15.12). Before synthesis windowing, the MDCT frequency partials (right column) correspond to sinusoids that are odd symmetric in the first half of the frame and even symmetric in the second. The property results in the time-aliasing components, which are cancelled out by the opposite effect of the next frame. The output frame (bottom left) is synthesis windowed. Courtesy of Juha Vilkamo.

defined as follows. Let us denote $X(k)$, where $k = 0, \ldots, (N-1)$, as the MDCT components, and $x(n)$, where $n = 0, \ldots, (2N-1)$, as the time-domain sequence. The forward MDCT is

$$X(k) = \sum_{n=0}^{2N-1} w_{\mathrm{a}}(n) x(n) \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2} + \frac{N}{2}\right)\left(k + \frac{1}{2}\right)\right], \tag{15.10}$$

whereas the inverse MDCT used to obtain the time-domain output signal sequence $y(n)$ is

$$y(n) = \frac{2}{N} w_s(n) \sum_{k=0}^{N-1} X(k) \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2} + \frac{N}{2}\right)\left(k + \frac{1}{2}\right)\right]. \tag{15.11}$$

The window sequences must satisfy the amplitude preservation condition in Equation (15.3). Furthermore, they must be designed such that the aliasing cancellation property is preserved. These criteria are fulfilled, for example, by (Geiger *et al.*, 2001):

$$w_{\mathrm{a}}(n) = w_s(n) = \sin\left(\frac{\pi}{4N}(2n+1)\right). \tag{15.12}$$

15.2.4 Pseudo-Quadrature Mirror Filter (PQMF) Bank

The *pseudo-quadrature mirror filter* (PQMF) bank is a real-valued, critically sampled filter bank that has also been applied in audio coding. The working principle of the PQMF is illustrated in the schematic diagram in Figure 15.2. For a PQMF, a low-pass prototype FIR filter response $h_{\mathrm{p}}(n)$ is designed and modulated with tone sequences to obtain the analysis band-pass filter responses $h_k^{\mathrm{a}}(n)$ and the synthesis band-pass filter responses $h_k^{\mathrm{s}}(n)$, where $k = 0, \ldots, K-1$ is the band index. The effect of modulation was explained in Section 15.1.3. The resulting filter bank thus has K filters of equal bandwidth measured in Hz. As illustrated in Figure 15.2, the band-pass signals, which are obtained by applying $h_k^{\mathrm{a}}(n)$ to the signal, are downsampled to obtain the non-redundant form. At the synthesis stage, the signals are upsampled so that the frequency content is duplicated to the original frequencies, followed by the synthesis band-pass filters. Finally, the bands are combined to form the output signal. The processing might be relatively complex computationally, if implemented as shown in the figure. Mathematically equivalent but more efficient implementations also apply for this transform, such as the one outlined by Rothweiler (1983), but are beyond the scope of this book.

The necessary criteria for the prototype low-pass filter response $h_{\mathrm{p}}(n)$ are that energy is preserved between the adjacent bands and that sufficient attenuation in non-adjacent bands is obtained to suppress the aliasing components (Creusere and Mitra, 1995; Cruz-Roldán *et al.*, 2002). For example, Cruz-Roldán *et al.* designed $h_{\mathrm{p}}(n)$ by adjusting the frequency of a windowed sinc-type low-pass filter so that its frequency response approximates the energy-preservation requirement. The analysis and synthesis filters have the same magnitude spectrum, and thus the amplitudes are preserved in total between the adjacent bands. The downsampling–upsampling process is necessary because it reduces redundancy, but it also generates aliasing frequency components that remain after the synthesis filter. These aliasing components between the adjacent bands have opposite phases with respect to each other and cancel each other out if the frequency-band signals are not modified. The aliasing components between the non-adjacent bands are suppressed by the synthesis filter to the point of being negligible. Due to this feature in the design, the reconstruction by a PQMF bank is near-perfect. If frequency decomposition of very narrow bands is desired, the order of the prototype filter must be high.

The PQMF bank band filters are specified as follows. Let N be the prototype filter order, $n = 0, \ldots, N-1$ be the time index, K the number of bands, and $k = 0, \ldots, K-1$ the frequency-band index. The analysis filter responses are

$$h_k^{\mathrm{a}}(n) = h_{\mathrm{p}}(n) \cos\left[\frac{\pi}{2K}(2k+1)\left(n - \frac{N}{2} - \frac{K}{2}\right)\right], \tag{15.13}$$

and the synthesis filter responses are

$$h_k^{\mathrm{s}}(n) = h_{\mathrm{p}}(n) \cos\left[\frac{\pi}{2K}(2k+1)\left(n - \frac{N}{2} + \frac{K}{2}\right)\right]. \tag{15.14}$$

15.2.5 Complex QMF

As shown in Figure 15.6, PQMF is prone to aliasing artefacts if the values of adjacent frequency bands are modified. The ability to process the bands independently without such aliasing can be

achieved by applying complex modulators instead of the real modulators in Equations (15.13) and (15.14):

$$h_k^{\mathrm{a}}(n) = h_{\mathrm{p}}(n) \exp\left[\mathrm{j}\frac{\pi}{2K}(2k+1)\left(n - \frac{N}{2} - \frac{K}{2}\right)\right] \tag{15.15}$$

and

$$h_k^{\mathrm{s}}(n) = h_{\mathrm{p}}(n) \exp\left[\mathrm{j}\frac{\pi}{2K}(2k+1)\left(n - \frac{N}{2} + \frac{K}{2}\right)\right], \tag{15.16}$$

where $\exp[a]$ means e^a. The complex-modulation process entails a data representation that is oversampled by a factor of two. The difference between the real and complex modulators can be illustrated by their spectral representation, as in Figure 15.11. The real modulators map the low-pass prototype filter spectrum to both positive and negative frequencies, while only

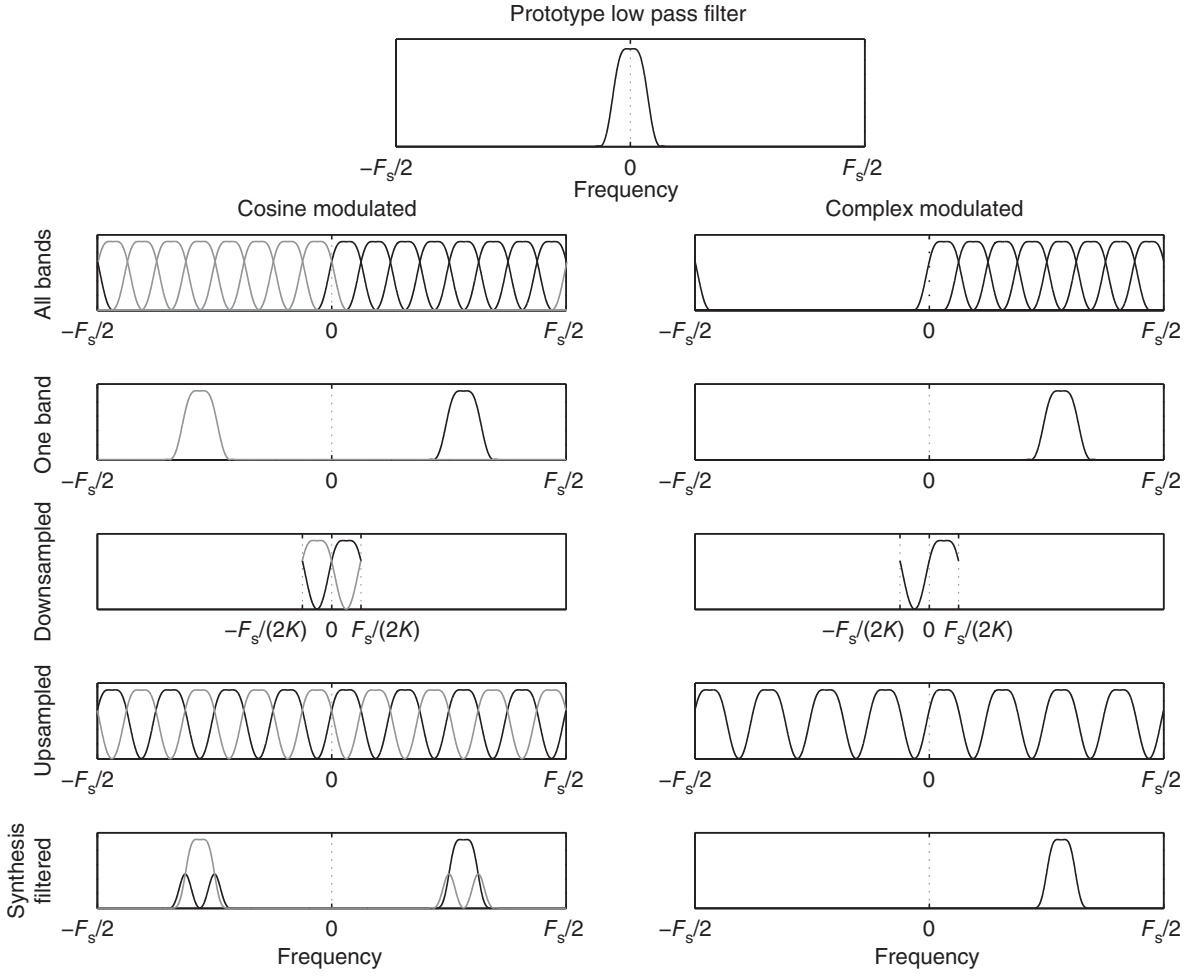

Figure 15.11 Real-modulated PQMF (left) and complex-modulated QMF (right) sub-band filtering. In real-valued processing, the positive and negative frequencies alias on top of each other during the downsampling–upsampling process. The aliased frequencies remain prominent after the synthesis filter. The neighbouring frequency bands have the same aliasing components but with opposite sign, thus cancelling the effect of aliasing. With complex-modulated processing, such aliasing components are absent in the first place, and thus the bands can be processed independently. Courtesy of Juha Vilkamo.

positive frequencies are produced when using complex modulation. In the downsampling process, the positive and negative frequencies of the real modulation wrap on top of each other to become the aliasing components of the neighbouring bands. With both modulators, the aliasing components of the non-neighboring bands are present and suppressed by the synthesis filter. The *complex-modulated QMF* is a typical transform in perceptually motivated time-frequency spatial audio processing techniques, discussed, for example by Breebaart *et al*. (2005, 2007) and Herre *et al*. (2012).

15.2.6 Sub-Sub-Band Filtering of the Complex QMF Bands

A typical resolution in complex QMF processing is $K = 64$ frequency bands, which, with a sampling rate of 44.1 kHz, results in bandwidths of approximately 345 Hz. As discussed earlier, the frequency resolution of hearing functions follows ERB or Bark bands (see Section 9.4.3 on page 167). Thus, this resolution is insufficient for the lowest frequencies. Therefore, the implementations discussed in Breebaart *et al*. (2005, 2007) and Herre *et al*. (2012) have applied a cascaded filter bank at the lowest frequency bands to obtain a higher frequency selectivity, which shown in Figure 15.12. Furthermore, some of the bands are summed together to reduce the complexity.

Different configurations exist for such cascading and summing, and the implementation presented in the figure involves feeding the lowest three bands to filter banks having 8, 4, and 4 bands, respectively, without further downsampling. The resulting bandwidths are then narrower at low frequencies and wider at high frequencies, although the transition is not smooth. In some perceptually motivated applications, the signal analysis is performed with frequency-band signals in which the higher QMF bands are also combined to form a perceptually motivated frequency resolution. The resulting bandwidths of one configuration in such processing are shown in Figure 15.13. The bandwidths follow somewhat the Bark frequency bands. Note that although the perceptual analysis is performed in such combined bands at the higher frequencies, the processing and inverse QMF transforms are applied at the original QMF bands because they contain the full information of the signal content.

15.2.7 Stochastic Measures of Time–Frequency Signals

In several time–frequency processing techniques for spatial audio (for example, Breebaart *et al*., 2005, 2007; Faller, 2006; and Herre *et al*., 2012) the spatial sound is synthesized by controlling the energies and interdependencies of the loudspeaker signals in the frequency bands. Let $X_1(k, m)$ and $X_2(k, m)$ be the signals of the two channels of a stereophonic set-up in the frequency domain with frequency band k and frame index m. The following parameters are often used to describe the channel or inter-channel energetic properties within a time–frequency area. Of the computations, the expectation operation $\mathrm{E}[\cdot]$ is typically implemented using a mean or a sum of the samples over a time–frequency area. First, let us define the channel energies and their cross-term:

$$\begin{aligned} E_1 &= \mathrm{E}\left[|X_1|^2\right] \\ E_2 &= \mathrm{E}\left[|X_2|^2\right] \\ \gamma_{12} &= \mathrm{E}\left[X_1 X_2^*\right], \end{aligned} \tag{15.17}$$

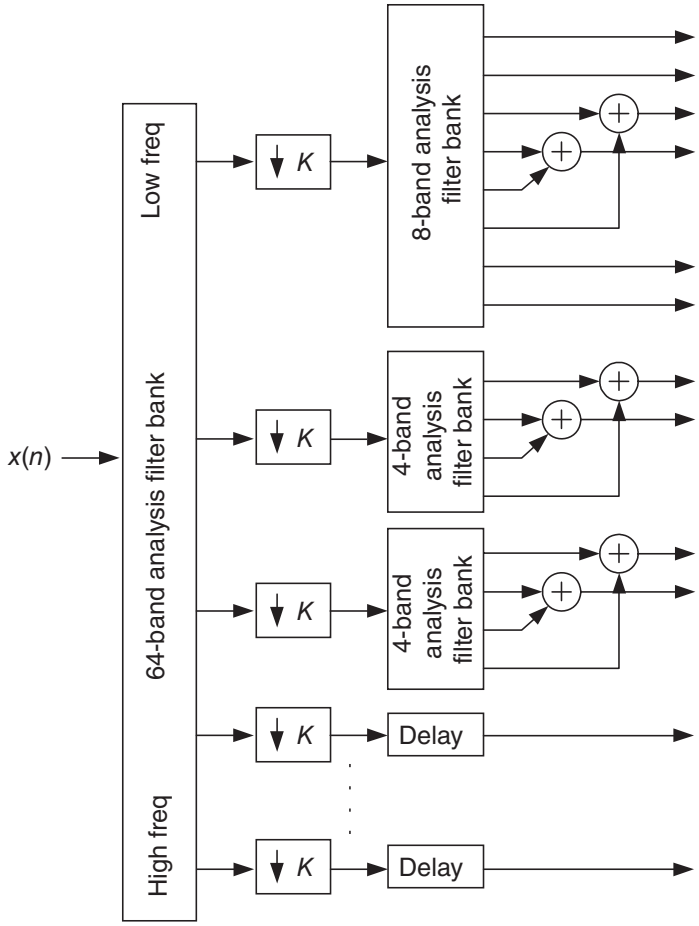

Figure 15.12 Hybrid QMF; that is, the sub-sub-band filtering of the lowest frequencies in a uniform, complex-modulated QMF bank. Some bands are combined to reduce complexity. The order of combination is specific, since the outputs of the secondary filter banks are not in the order of the absolute frequency. Adapted from Herre *et al*. (2005).

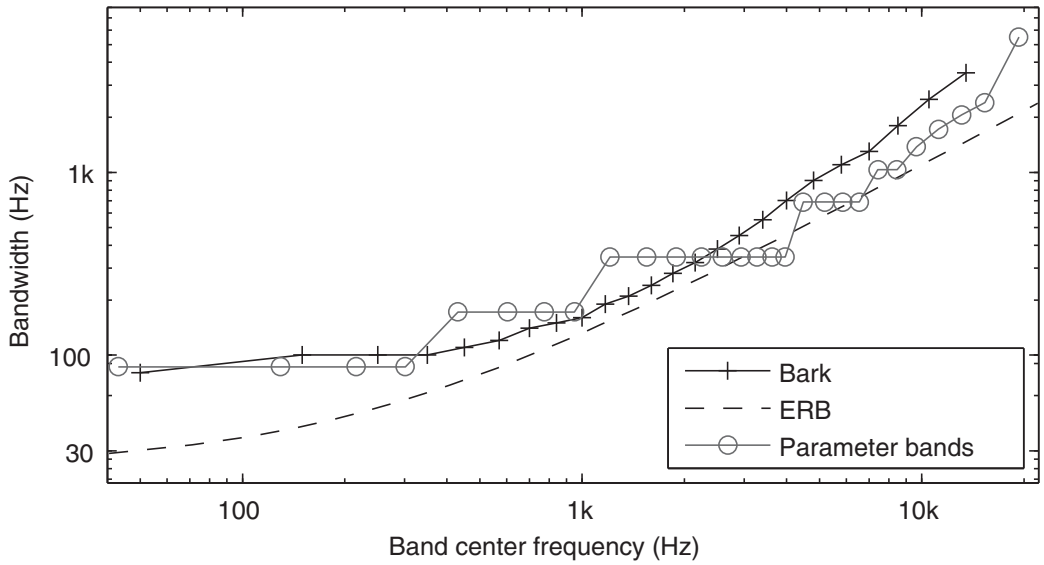

Figure 15.13 The bandwidths of an implementation using a hybrid QMF sub-sub-band filter bank that combines the bands in a perceptual fashion compared with Bark and ERB bandwidths.

where X_2^* is the complex conjugate of X_2. The inter-channel level difference in dB is

$$\text{ICLD} = 10 \log_{10} \left(\frac{E_1}{E_2} \right), \tag{15.18}$$

and the inter-channel phase difference is

$$\text{ICPD} = \arg \left(\gamma_{12} \right), \tag{15.19}$$

where the arg() operator determines the angle of a complex value in the complex plane. The inter-channel coherence is

$$\text{ICC} = \frac{|\gamma_{12}|}{\sqrt{E_1 \cdot E_2}}. \tag{15.20}$$

The value ICC is a normalized similarity index between 0 and 1, where ICC = 1 means that the signals are coherent, although potentially with level differences, and ICC = 0 means that the signals are incoherent. A similar measure without the absolute-value operator is

$$\text{ICC}' = \frac{\gamma_{12}}{\sqrt{E_1 \cdot E_2}}, \tag{15.21}$$

which also includes the angle of the complex-valued cross-term. If the imaginary part is ignored, the value ICC ranges from -1 to 1 and determines the inverse or in-phase coherence between the channel pair. Ignoring the imaginary part means that signals that are out of phase by $\pi/2$ are treated as incoherent signals, which may be suitable, for example, for stereo upmixing using direct-ambience decomposition (Avendano and Jot, 2002; Faller, 2006). The task of the processing algorithm dictates how to account for the phase offset between the signals.

15.2.8 Decorrelation

Decorrelation is a method for processing a signal so that its ICC $\approx$ 0 with respect to the original signal as well as with respect to the processed signals from other decorrelators. Ideally, a decorrelator is designed such that the perceptual characteristics of the sound are least affected. Decorrelation is necessary for applications that increase the number of independent channels, such as upmixing or surround sound rendered from a few microphone signals.

There are various implementations of decorrelators. A typical approach is to alter the time or phase structure of the signal over a short time interval. Examples of such processes are different delays in frequency bands, all-pass filters, and convolutions with short noise sequences. It is widely known in the field that decorrelators can cause degradation of the perceived sound quality with certain signal content such as applause (Kuntz *et al.*, 2011; Laitinen *et al.*, 2011), because, as a consequence of the decorrelation, the sharp temporal structure of the signal is altered. Applications employing decorrelators thus often apply specific processes to avoid or reduce such effects, for example, with onset detectors that bypass transients from the decorrelators.

15.3 Time–Frequency-Domain Audio-Processing Techniques

In this section, a set of key applications in the field of perceptually motivated time–frequency audio processing is reviewed.

15.3.1 Masking-Based Audio Coding

With audio coding methods such as MPEG-1 Layer-3 (MP3) (ISO/IEC, 1993) and MPEG-2 Advanced Audio Coding (AAC) (Bosi *et al.*, 1997; ISO/IEC, 1997), the main means of reducing the bit-rate is to transform the signals into the time–frequency domain and to optimize the quantization of the time–frequency samples using a perceptual masking model, as shown in Figure 15.14.

If quantization noise was equally spread over the entire frequency region, it would be easily audible in those frequency regions where the signal level was low. The spectrum of quantization noise can be shaped so that it follows the masking curve created by the signal, but shifted slightly lower in level. See the figures in Section 9.2 for examples of masking curves. In general, if the level is set to about 13 dB lower than that of the signal, quantization noise is no longer audible, although corresponding quantization noise with a flat spectrum is clearly annoying. This effect is known as the '13-dB miracle' from an audio demonstration given by J. D. Johnston and K. Brandenburg at AT&T Bell Labs in 1990. The audio signals used in the demonstration are described by Brandenburg and Sporer (1992).

Critically sampled filter banks are preferred for audio coding, since signal modifications, except those for the quantization, are not intended, and the property of having the least number of data points for the transmission is desired.

15.3.2 Audio Coding with Spectral Band Replication

By reducing the bit rate, a limit is eventually reached when the quantization noise of a traditional audio encoder significantly exceeds the masking threshold, equivalent to exceeding the −13-dB noise level in the previous example. To optimize the quality in scenarios when bit rates of, for example, 24 kilobits per second per channel are applied, the method of spectral band replication (SBR) (Ekstrand, 2002) can be applied to utilize the typical temporal similarities

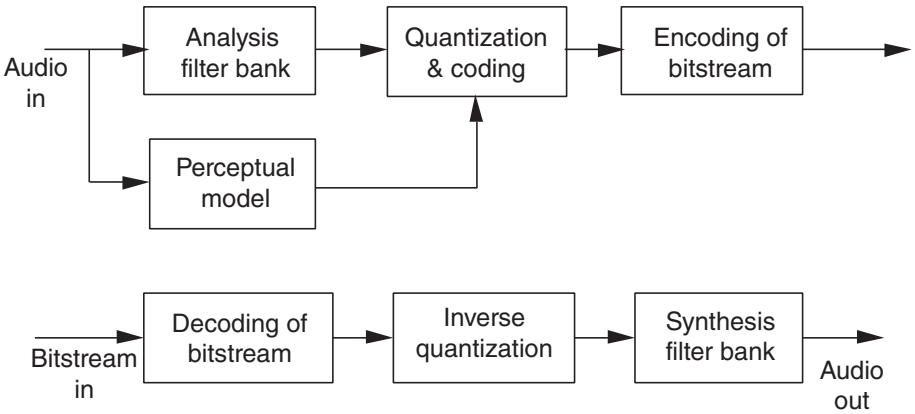

Figure 15.14 Block diagram of an audio encoder and decoder based on perceptual masking. Adapted from Brandenburg (1999).

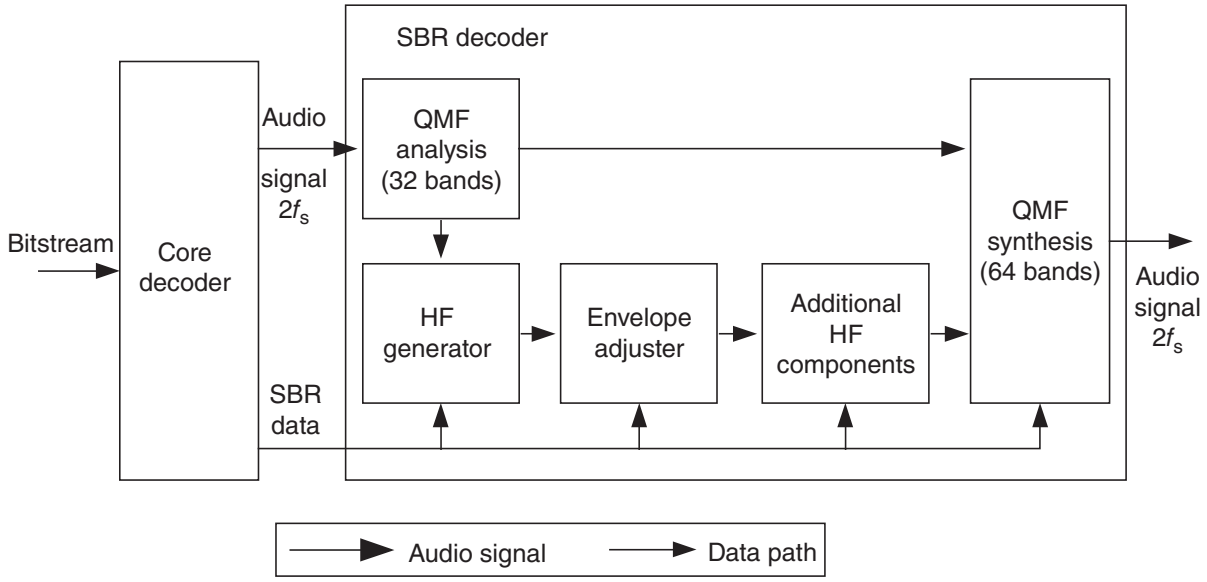

Figure 15.15 Spectral band replication, in which only the lower frequencies are transmitted by the core encoder, for example AAC, and where the higher frequencies are predicted based on the lower frequencies, and adjusted to better match the original higher frequencies based on low-bit-rate side information. In this case, the sampling frequency of the output signal is doubled from the audio signal originating from the core decoder. Other ratios of sampling frequencies are also possible. Adapted from Ekstrand (2002).

between the low and high frequencies. With SBR, the higher frequencies are not transmitted, and the bits are instead allocated to convey the lower frequencies more accurately, from which the higher frequencies are predicted (see Figure 15.15). Side information at a low bit rate is transmitted to adjust the spectral envelope of the higher frequencies to match better that of the original higher frequencies.

15.3.3 Parametric Stereo, MPEG Surround, and Spatial Audio Object Coding

In stereo and multi-channel audio signal transmission with low bit rates, the bit allocation can be optimized by transmitting the spatial aspect as low-bit-rate side information and transmitting only a reduced number of downmixed audio channels (Baumgarte and Faller, 2003; Faller and Baumgarte, 2003; Schuijers *et al.*, 2003). For two-channel stereo signals, Parametric Stereo (PS) (Breebaart *et al.*, 2005; Purnhagen, 2004), the signal-flow diagram of which is shown in Figure 15.16, can be used to convey the channel data using a mono downmix and the parametric side information containing the ICLD, ICPD, and ICC parameters in the frequency bands. MPEG Surround (Breebaart *et al.*, 2007; Herre *et al.*, 2005) and MPEG Spatial Audio Object Coding (SAOC) (Herre *et al.*, 2012) are similar parametric multi-channel techniques. MPEG Surround can be used for efficient transmission of 5.1 surround audio content in stereo or mono downmix channels with spatial metadata. SAOC provides a mixture of audio objects, that is, single-channel audio signals, in the downmix channels, the spatial rendering of which can be manually adjusted at the receiver end based on the parametric side information. An example of an SAOC application is found in the context of virtual reality, where the different talker

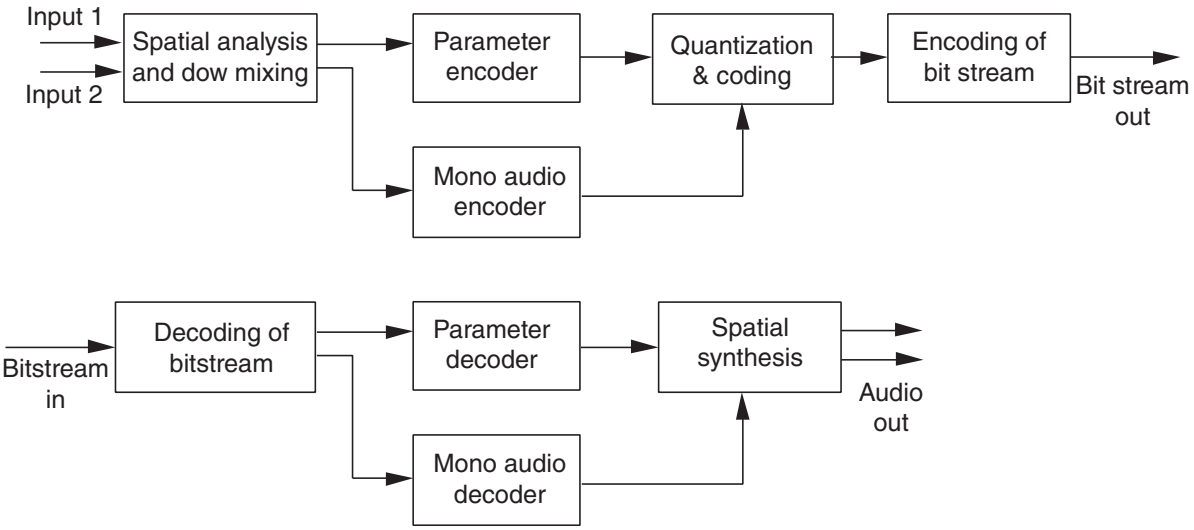

Figure 15.16 Parametric Stereo encoding and decoding. The inter-channel level differences, phase differences, and coherences are measured in the frequency bands for transmission as low-bit-rate side information. The signals are downmixed and encoded with a core encoder, such as AAC. At the receiver end, the AAC bit stream is decoded. The spatial properties of the stereo sound are re-synthesized using amplitude and phase adjustments and decorrelation. Adapted from Breebaart *et al*. (2005).

signals can be combined to save the bit rate, but spatially rendered independently based on the positioning of the talkers in the virtual environment.

The parametric side information can be embedded in the bit stream of the core coder in such a way that a receiver without a parametric decoder can decode the downmix channels. Enhanced decoders can take advantage of the parametric side information using the same bit stream. The downmix channels can be encoded using, for example, AAC or adaptively with a speech codec, which is a technique applied in the recent Unified Speech and Audio Coding (USAC) scheme (Neuendorf *et al*., 2013).

15.3.4 Stereo Upmixing and Enhancement for Loudspeakers and Headphones

One of the tasks in audio is to present a two-channel stereophonic audio track using more than two loudspeakers. In principle, it is not possible to derive more independent channels than there already are. However, when the low spatial resolution of humans is taken into account, stereophonic signals can be rendered to a higher number of loudspeakers with plausible results.

In adaptive stereo upmixing (Avendano and Jot, 2004; Faller, 2006), the stereo sound is modelled in frequency bands in terms of the direct and ambient signal components that are redistributed to the extended loudspeaker set-up, as shown in Figure 15.17. An example of a direct-ambience model is to assume an amplitude-panned direct source and incoherent equal-energy ambience in the frequency bands, expressed as

$$\begin{bmatrix} X_1(k,m) \\ X_2(k,m) \end{bmatrix} = \begin{bmatrix} \sqrt{g(k,m)} \\ \sqrt{1-g(k,m)} \end{bmatrix} D(k,m) + \begin{bmatrix} A_1(k,m) \\ A_2(k,m) \end{bmatrix}, \tag{15.22}$$

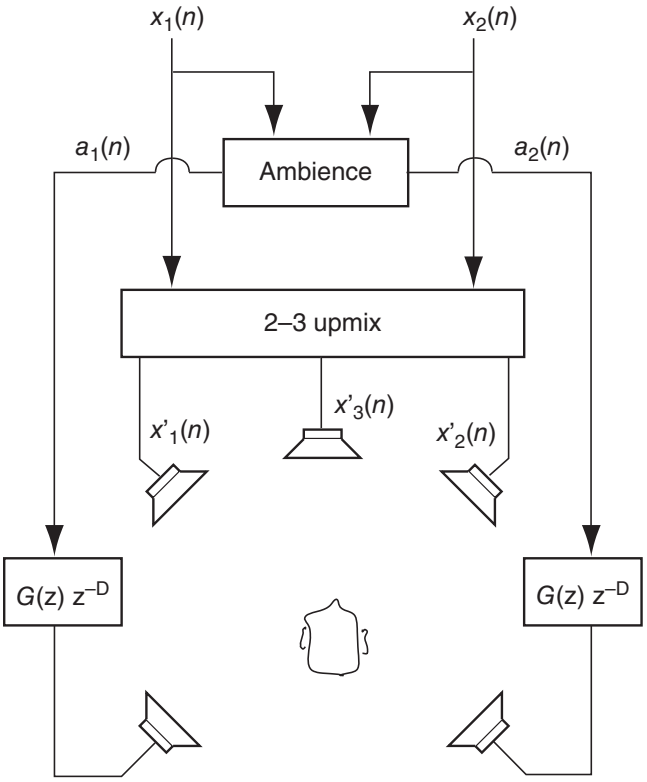

Figure 15.17 In time–frequency-domain upmixing from stereophonic sound to 5.1 surround, a direct-ambience decomposition is applied to obtain the rear ambience channels and a 2–3 upmixer is applied to obtain the centre channel. The operators $G(z)$ are decorrelating all-pass filters. Adapted from Avendano and Jot (2004).

where $0 \leq g(k,m) \leq 1$ is an energy-panning coefficient, $D(k,m)$ is the amplitude-panned signal, and $A_1(k,m)$ and $A_2(k,m)$ are the left and right channel ambience signals, respectively. According to the model, the two ambience components and the amplitude-panned components are incoherent with respect to each other, and the ambience energy is the same in both channels. The model parameters, g, $E_D = \mathrm{E}\left[|D|^2\right]$, and $E_A = \mathrm{E}\left[|A_1|^2\right] = \mathrm{E}\left[|A_2|^2\right]$, can be solved uniquely based on the inter-channel measures described in Section 15.2.7. Different direct-ambience models have been discussed by Merimaa *et al.* (2007). The benefits sought by upmixing stereo to a surround set-up with a bigger number of loudspeakers include improved directional quality of amplitude-panned virtual sources in a larger listening area, and a more evenly surrounding reproduction of the ambience, such as reverberation, also in a larger listening area (Faller, 2006).

A concept similar to upmixing, but for headphone playback, involves adaptively processing the frequency bands of a stereo signal to obtain the natural binaural characteristics. Menzer and Faller (2010) and Faller and Breebaart (2011) applied direct-ambience signal analysis to process the ambience signal to match the frequency-band coherence occurring in a diffuse sound field. The direct signal was processed using head-related transfer functions (HRTFs), which are free-field transfer functions from a source to both ears, or with binaural room impulse responses (BRIRs) (Faller and Breebaart, 2011). The procedure was reported in informal listening to improve the perceived naturalness of the sound over the original stereo (Menzer and

Faller, 2010) and to improve the width and accuracy of the sound stage over the conventional methods of binaural processing of stereo sound (Faller and Breebaart, 2011).

Summary

Various methods exist to map a time-domain audio signal to the frequency domain, and different methods have been developed for different applications. If the signal is to be processed in the frequency domain, the time–frequency transform methods used should be robust against aliasing artefacts. Processing in the time–frequency domain enables the use of efficient coding strategies for monophonic and multi-channel signals.

Further Reading

The reader might find Smith (2011) useful for learning more about time-frequency signal processing techniques. The reader is encouraged to read more on coding of audio signals in Kahrs and Brandenburg (1998). The well-known audio codecs, such as MPEG-1 Layer-3 (mp3), and Advanced Audio Coding (AAC) are explained by Brandenburg (1999). A deeper discussion on multi-channel audio reproduction is given by Breebaart and Faller (2008), and an extensive review of transform-based parametric audio coding can be found in Herre and Disch (2014).

References

Avendano, C. and Jot, J.M. (2002) Ambience extraction and synthesis from stereo signals for multi-channel audio up-mix. *IEEE Int. Conf. Acoustics, Speech, and Signal Proc.*, volume 2, pp. 1957–1960.

Avendano, C. and Jot, J.M. (2004) A frequency-domain approach to multichannel upmix. *J. Audio Eng. Soc.*, **52**(7/8), 740–749.

Baumgarte, F. and Faller, C. (2003) Binaural cue coding – Part I: Psychoacoustic fundamentals and design principles. *IEEE Trans. Speech and Audio Proc.*, **11**(6), 509–519.

Bosi, M,, Brandenburg, K., Quackenbush, S., Fielder, L., Akagiri, K., Fuchs, H., and Dietz, M. (1997) ISO/IEC MPEG-2 advanced audio coding. *J. Audio Eng. Soc.*, **45**(10), 789–814.

Brandenburg, K. (1999) MP3 and AAC explained, *17th Int. Conf. Audio Eng. Soc.* AES.

Brandenburg, K. and Sporer, T. (1992) "NMR" and "Masking Flag": Evaluation of Quality Using Perceptual Criteria. *11th Int. Audio Eng. Soc. Conf.: Test & Measurement* AES.

Breebaart, J. and Faller, C. (2008) *Spatial Audio Processing: MPEG Surround and Other Applications*. John Wiley & Sons.

Breebaart, J., Chong, K.S., Disch, S., Faller, C., Herre, J., Hilpert, J., Kjörling, K., Koppens, J., Linzmeier, K., Oomen, W., Purnhagen, H., and Rödén, J. (2007) MPEG Surround – the ISO/MPEG standard for efficient and compatible multi-channel audio coding. *Audio Eng. Soc. Convention 122*. AES.

Breebaart, J., van de Par, S., Kohlrausch, A., and Schuijers, E. (2005) Parametric coding of stereo audio. *EURASIP J. Appl. Signal Proc.*, **2005**, 1305–1322.

Creusere, C.D. and Mitra, S.K. (1995) A simple method for designing high-quality prototype filters for M-band pseudo QMF banks. *IEEE Trans. Signal Proc.*, **43**(4), 1005–1007.

Cruz-Roldán, F., Amo-López, P., Maldonado-Bascón, S., and Lawson, S.S. (2002) An efficient and simple method for designing prototype filters for cosine-modulated pseudo-QMF banks. *IEEE Signal Proc. Letters*, **9**(1), 29–31.

Ekstrand, P. (2002) Bandwidth extension of audio signals by spectral band replication. *1st IEEE Benelux Workshop on Model based Processing and Coding of Audio*.

Faller, C. (2006) Multiple-loudspeaker playback of stereo signals. *J. Audio Eng. Soc.*, **54**(11), 1051–1064.

Faller, C. and Baumgarte, F. (2003) Binaural cue coding – Part II: Schemes and applications. *IEEE Trans. Speech and Audio Proc.*, **11**(6), 520–531.

Faller, C. and Breebaart, J. (2011) Binaural reproduction of stereo signals using upmixing and diffuse rendering. *Audio Eng. Soc. Convention 131*. AES.

Geiger, R., Sporer, T., Koller, J., and Brandenburg, K. (2001) Audio coding based on integer transforms. *Audio Eng. Soc. Convention 111*. AES.

Herre, J. and Disch, S. (2014) Perceptual audio coding. In Chellappa, R. and Theodoridis, S. (eds) *Image, Video Processing and Analysis, Hardware, Audio, Acoustic and Speech Processing*. Academic Press, pp. 757–800.

Herre, J., Purnhagen, H., Breebaart, J., Faller, C., Disch, S., Kjörling, K., Schuijers, E., Hilpert, J., and Myburg, F. (2005) The reference model architecture for MPEG spatial audio coding. *Audio Eng. Soc. Convention 118*.

Herre, J., Purnhagen, H., Koppens, J., Hellmuth, O., Engdegaard, J., Hilpert, J., Villemoes, L., Terentiv, L., Falch, C., Hölzer, A., Valero, M.L., Resch, B., Mundt, H., and Oh, H.O. (2012) MPEG spatial audio object coding – the ISO/MPEG standard for efficient coding of interactive audio scenes. *J. Audio Eng. Soc.*, **60**(9), 655–673.

ISO/IEC (1993) Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s – Part 3: Audio. Standard 11172-3.

ISO/IEC (1997) MPEG-2 advanced audio coding, AAC. Standard JTC1/SC29/WG11 (MPEG).

Kahrs, M. and Brandenburg, K. (1998) *Applications of Digital Signal Processing To Audio and Acoustics*, volume 437. Springer.

Kuntz, A., Disch, S., Bäckström, T., and Robilliard, J. (2011) The transient steering decorrelator tool in the upcoming MPEG unified speech and audio coding standard. *Audio Eng. Soc. Convention 131*.

Laitinen, M-V., Küch, F., Disch, S., and Pulkki, V. (2011) Reproducing applause-type signals with directional audio coding. *J. Audio Eng. Soc.*, **59**(1/2), 29–43.

Menzer, F. and Faller, C. (2010) Stereo-to-binaural conversion using interaural coherence matching. *Audio Eng. Soc. Convention 128*.

Merimaa, J., Goodwin, M.M. and Jot, J.M. (2007) Correlation-based ambience extraction from stereo recordings. *Audio Eng. Soc. Convention 123*.

Neuendorf, M., Multrus, M., Rettelbach, N., Fuchs, G., Robilliard, J., Lecomte, J., Wilde, S., Bayer, S., Disch, S., Helmrich, C., Lefebvre, R., Gournay, P., Bessette, B., Lapierre, J., Kjörling, K., Purnhagen, H., Villemoes, L., Oomen, W., Schuijers, E., Kikuiri, K., Chinen, T., Norimatsu, T., Chong, K.S., Oh, E., Kim, M., Quackenbush, S., and Grill, B. (2013) The ISO/MPEG unified speech and audio coding standard – consistent high quality for all content types and at all bit rates. *J. Audio Eng. Soc.*, **61**(12), 956–977.

Purnhagen, H. (2004) Low complexity parametric stereo coding in MPEG-4. *7th Int. Conf. on Digital Audio Effects DAFx04*.

Rothweiler, J. (1983) Polyphase quadrature filters – a new subband coding technique. *IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, volume 8, pp. 1280–1283.

Schuijers, E., Oomen, W., and Breebaart, J. (2003) Advances in parametric coding for high-quality audio. *Audio Eng. Soc. Convention 114*.

Smith, J.O. (2011) *Spectral Audio Signal Processing*. W3K.

Vickers, E. (2012) Frequency-domain implementation of time-varying FIR filters. *Audio Eng. Soc. Convention 133*. AES.