Chapter 2

Multirate digital signal processing

In multirate digital signal processing the sampling rate of a signal is changed in order to increase the efficiency of various signal processing operations. Decimation, or down-sampling, reduces the sampling rate, whereas expansion, or up-sampling, followed by interpolation increases the sampling rate. Some applications of multirate signal processing are:

- Up-sampling, i.e., increasing the sampling frequency, before D/A conversion in order to relax the requirements of the analog lowpass antialiasing filter. This technique is used in audio CD, where the sampling frequency 44.1 kHz is increased fourfold to 176.4 kHz before D/A conversion.
- Various systems in digital audio signal processing often operate at different sampling rates. The connection of such systems requires a conversion of sampling rate.
- Decomposition of a signal into M components containing various frequency bands. If the original signal is sampled at the sampling frequency f_s (with a frequency band of width $f_s/2$, or half the sampling frequency), every component then contains a frequency band of width $\frac{1}{2}f_s/M$ only, and can be represented using the sampling rate f_s/M. This allows for efficient parallel signal processing with processors operating at lower sampling rates. The technique is also applied to data compression in *subband coding*, for example in speech processing, where the various frequency band components are represented with different word lengths.
- In the implementation of high-performance filtering operations, where a very narrow transition band is required. The requirement of narrow transition bands leads to very high filter orders. However, by decomposing the signal into a number of subbands containing the passband, stopband and transition bands, each component can be processed at a lower rate, and the transition band will be less narrow. Hence the required filter complexity may be reduced significantly.

The theory of multirate signal processing is quite extensive, and not entirely trivial. Here we will discuss some of the main ideas only.

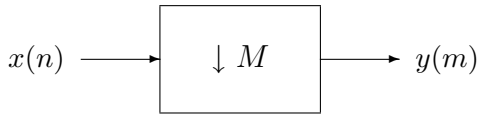

Figure 2.1: Decimation by the factor M.

2.1 Basic operations of multirate processing

2.1.1 Sampling rate conversion

Decimation.
Decimation, or down-sampling, consists of reducing the sampling rate by a factor M, cf. Figure 2.1. Here, the output is defined as

$$y(m) = x(mM) \tag{2.1}$$

i.e., it consists of every Mth element of the input signal. It is clear that the decimated signal y does not in general contain all information about the original signal x. Therefore, decimation is usually applied in filter banks and preceded by filters which extract the relevant frequency bands, cf. Figure 2.3.

In order to analyze the frequency domain characteristics of a multirate processing system with decimation, we need to study the relation between the Fourier transforms, or the z-transforms, of the signals x and y. For simplicity we consider the case $M = 2$ only. Then the decimated signal y is given by

$$y(m) = x(2m) \tag{2.2}$$

or $\{y(m)\} = \{x(0), x(2), x(4), \ldots\}$. Given the z-transform of $\{x(n)\}$,

$$\hat{x}(z) = x(0) + x(1)z^{-1} + x(2)z^{-2} + \cdots + x(n)z^{-n} + \cdots \tag{2.3}$$

we should like to have an expression for the z-transform of $\{y(m)\}$, i.e. (using (2.2)),

$$\begin{aligned} \hat{y}(z) &= y(0) + y(1)z^{-1} + y(2)z^{-2} + \cdots + y(m)z^{-m} + \cdots \\ &= x(0) + x(2)z^{-1} + x(4)z^{-2} + \cdots + x(2m)z^{-m} + \cdots \end{aligned} \tag{2.4}$$

In order to derive an expression for the z-transform of y, it is convenient to represent the decimation in two stages as follows. First, define the signal

$$\{v(n)\} = \{x(0), 0, x(2), 0, x(4), 0, \ldots\} \tag{2.5}$$

which has the z-transform

$$\hat{v}(z) = x(0) + x(2)z^{-2} + x(4)z^{-4} + \cdots + x(2m)z^{-2m} + \cdots \tag{2.6}$$

As

$$\hat{x}(-z) = x(0) - x(1)z^{-1} + x(2)z^{-2} - \cdots + x(2m)z^{-2m} + \cdots \tag{2.7}$$

it follows that

$$\hat{v}(z) = \frac{1}{2}\Big(\hat{x}(z) + \hat{x}(-z)\Big) \tag{2.8}$$

By (2.4) and (2.6), $\hat{y}(z) = \hat{v}(z^{1/2})$. Hence we have obtained the relation

$$\hat{y}(z) = \frac{1}{2}\left(\hat{x}(z^{1/2}) + \hat{x}(-z^{1/2})\right) \tag{2.9}$$

In order to determine the frequency domain characteristics of the decimated signal $\{y(m)\}$, recall that the Fourier transform is related to the z-transform according to

$$Y(\omega) = \hat{y}(z)\Big|_{z=e^{j\omega}} \tag{2.10}$$

Hence, we have from (2.9),

$$Y(\omega) = \frac{1}{2}\left(\hat{x}(e^{j\omega/2}) + \hat{x}(-e^{j\omega/2})\right) \tag{2.11}$$

Noting that $-1 = e^{j\pi}$, it follows that

$$\begin{aligned} Y(\omega) &= \frac{1}{2}\Big(\hat{x}(e^{j\omega/2}) + \hat{x}(e^{j(\omega/2+\pi)})\Big) \\ &= \frac{1}{2}\Big(X(\omega/2) + X(\omega/2+\pi)\Big) \end{aligned} \tag{2.12}$$

where X is the Fourier-transform of the sequence $\{x(n)\}$. But from the properties of the Fourier transform (periodicity and symmetry) it follows that $X(\omega/2+\pi) = X(\omega/2-\pi) = X(\pi-\omega/2)^*$. Hence

$$Y(\omega) = \frac{1}{2}\Big(X(\omega/2) + X(\pi-\omega/2)^*\Big) \tag{2.13}$$

The Fourier-transform of $\{y(m)\}$ thus cannot distinguish between the frequencies $\omega/2$ and $\pi-\omega/2$ of $\{x(n)\}$. This is equivalent to the frequency folding phenomenon occurring when sampling a continuous-time signal.

Hence, whereas the signal $\{x(n)\}$ consists of frequencies in $[0,\pi]$, the frequency contents of the decimated signal $\{y(m)\}$ are restricted to the range $[0,\pi/2]$. Moreover, after decimation of the signal $\{x(n)\}$, its frequency components in $[0,\pi/2]$ cannot be distinguished from the frequency components in the range $[\pi/2,\pi]$

Expansion.
Expansion, or up-sampling, consists of increasing the sampling rate by a factor L, cf. Figure 2.2. Here, the output is obtained by inserting $L-1$ zeros between successive values of the input $x(n)$, i.e.

$$y(m) = \begin{cases} x(m/L), & \text{for } m = 0, L, 2L, \ldots \\ 0, & \text{otherwise.} \end{cases} \tag{2.14}$$

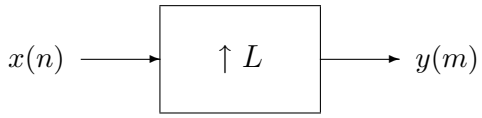

Figure 2.2: Expansion by the factor L.

The expansion operation followed by interpolation leads to a representation of the signal x at a sampling rate increased by the factor L.

By (2.14), the expanded signal $\{y(m)\}$ has the z-transform

$$\begin{aligned}\hat{y}(z) &= x(0) + x(1)z^{-L} + x(2)z^{-2L} + \cdots + x(m)z^{-mL} + \cdots \\ &= \hat{x}(z^L)\end{aligned} \tag{2.15}$$

The Fourier transform is thus given by

$$\begin{aligned}Y(\omega) &= \hat{y}(z)\Big|_{z=e^{j\omega}} = \hat{x}(z^L)\Big|_{z=e^{j\omega}} = \hat{x}(z)\Big|_{z=e^{j\omega L}} \\ &= X(\omega L)\end{aligned} \tag{2.16}$$

The transform $Y(\omega)$ at a given frequency $\omega \in [0, \pi]$ is thus equal to $X(\omega L)$, where $\omega L \in [0, L\pi]$. But as the Fourier-transform is periodic with period 2π, we have $X(\omega L) = X(\omega L + 2\pi k) = X((\omega + \frac{2\pi k}{L})L)$, and it follows that $Y(\omega) = Y(\omega + \frac{2\pi k}{L})$. Hence $Y(\omega)$ is periodic, with L repetitions of $X(\omega)$ in the frequency range $[0, 2\pi]$. For example, for $L = 2$, we have $X(2\omega) = X(2\omega - 2\pi) = X(2\pi - 2\omega)^* = X(2(\pi - \omega))^*$. Hence, for $L = 2$,

$$Y(\omega) = \begin{cases} X(2\omega), & \text{for } 0 \le \omega \le \pi/2 \\ X(2(\pi - \omega))^* = Y(\pi - \omega)^*, & \text{for } \pi/2 \le \omega \le \pi, \end{cases} \tag{2.17}$$

and $Y(\omega)$ is therefore uniquely defined by its values in the frequency band $[0, \pi/2]$.

In order to reconstruct the correct interpolating signal at the higher sampling rate, an interpolating filter has to be introduced after the expansion. This is equivalent to the situation in D/A conversion, where a low-pass filter is used after the hold function.

Sampling rate conversion by non-integer factors.
Sampling rate conversion by a non-integer factor F may be achieved by the use of expansion and decimation operations. If the conversion factor can be expressed as a rational number, i.e., the ratio of two integers, $F = L/M$, then the obvious way to achieve the conversion is to apply expansion by the factor L followed by lowpass filtering (to extract the low-frequency component of the expanded signal) and decimation with the factor M. A problem with this direct approach occurs when the integers L and M are large, in which case there will be very high requirements on the anti-aliasing filters. In practice this problem is avoided in multirate signal processing by performing the sampling rate conversions in several stages with small integer factors (for example, $M = L = 2$) at each stage.

An implementation of sampling rate conversion is given by the Matlab routine

```
y = resample(x,L,M)
```

which resamples the sequence in vector x at L/M times the original sampling rate. Before resampling, an anti-aliasing (lowpass) FIR filter is applied to x.

Sampling rate conversion of a finite-length sequence.
A sequence of finite length can be resampled in a very straightforward way as follows. Assume that the sequence

$$\{x(n)\} = \{x(0), x(1), \ldots, x(N-1)\}$$

of length N has been obtained by sampling a continuous-time, analog, signal $x_a(t)$ with sampling frequency f_x, so that

$$x(n) = x_a(nT_x)$$

where $T_x = 1/f_x$ is the associated sampling period.

We want to determine another sequence

$$\{y(m)\} = \{y(0), y(1), \ldots, y(M-1)\}$$

of length $M > N$ corresponding to the sampling frequency $f_y = \frac{M}{N} f_x$, so that

$$y(m) = x_a(mT_y)$$

where $T_y = 1/f_y$ is the associated sampling period. Notice that $T_y = \frac{N}{M} T_x$, so that $MT_y = NT_x$ and the sequences $\{x(n)\}$ and $\{y(m)\}$ cover the same continuous time interval.

The resampled sequence can be determined by observing that there is a simple relationship between the Fourier transforms of the given sequence $\{x(n)\}$ and the resampled sequence $\{y(m)\}$.

Let $\{x(n)\}$ have Fourier transform $\{X(k)\}$. Assuming that the underlying analog signal $x_a(t)$ is bandlimited, so that its Fourier transform $X_a(\omega)$ vanishes for $\omega \geq \pi f_x$, i.e., there is no aliasing when $x_a(t)$ is sampled using sampling frequency f_x, we have (see introductory course in signal processing)

$$X(k) = \frac{1}{T_x} X_a(2\pi f_x k/N), \;\; k = 0, 1, \ldots, N/2$$

and

$$X(N-k) = X(k)^*$$

where we have for convenience assumed that N is even. Similarly, for the Fourier transform $\{Y(l)\}$ of the resampled sequence $\{y(m)\}$ we have

$$Y(l) = \frac{1}{T_y} X_a(2\pi f_y l/M), \;\; l = 0, 1, \ldots, M/2$$

and

$$Y(M-l) = Y(l)^*$$

But $f_y = \frac{M}{N} f_x$ implies $f_y/M = f_x/N$ Hence the first $N/2$ elements of $Y(l)$ are given by

$$\begin{aligned} Y(l) &= \frac{1}{T_y} X_a(2\pi f_x l/N) = \frac{T_x}{T_y} X(l) \\ &= \frac{M}{N} X(l), \;\; l = 0, 1, \ldots, N/2 \end{aligned}$$

As $X_a(\omega)$ vanishes for larger frequencies, the corresponding elements $Y(l)$ are zero,

$$Y(l) = 0, \;\; k = N/2 + 1, \ldots, M/2$$

Finally, by symmetricity of the Fourier transform we have

$$Y(M - l) = Y(l)^*$$

Having obtained its Fourier transform, the resampled sequence $\{y(m)\}$ can be determined by taking the inverse Fourier transform of $\{Y(l)\}$.

Notice that the above procedure requires only a standard Fourier transform of the given sequence followed by scaling, zero padding and inverse Fourier transform. Observe that if the intermediate Fourier transform operations are eliminated, the procedure reduces to Shannon reconstruction, where the resampled sequence $\{y(m)\}$ is expressed explicitly in terms of the original sequence $\{x(n)\}$.

2.1.2 Analysis and synthesis filter banks

A basic operation in multirate signal processing is to decompose a signal into a number of subband components, which can be processed at a lower rate corresponding to the bandwidth of the frequency bands. Recall from equation (2.13) that down-sampling mixes frequency components in the original signal by aliasing and frequency folding. Therefore, the signal should be filtered before decimation. Figure 2.3 shows the decomposition of a signal into two subband components. The purpose of the filters H_1 and H_2 is to extract the low- and high-frequency components of the signal x before decimation. For example, the component x_{D1} may contain the low-frequency components of x, and the component x_{D2} may contain the low-frequency components of x. The set of filters shown in Figure 2.3 is called an *analysis filter bank*. If required, the signals may be decimated further into narrower subband components, cf. Figure 2.4. If H_1 is a low-pass filter and H_2 a high-pass filter, x_{D1} is the low-frequency component consisting of the subband $[0, \pi/2]$, whereas x_{D21} and x_{D22} contain the subbands $[\pi/2, 3\pi/4]$ and $[3\pi/4, \pi]$, respectively.

Notice that if possible, a convenient way to implement the decimation is to use stages with the decimation factor $M = 2$ as shown in Figure 2.4. Then only one low-pass and one high-pass filter is required. For $M > 2$, bandpass filters with different passbands are required as well.

After processing of the separate subband components, they are combined to reconstruct (a properly processed version of) the original signal at the original, higher

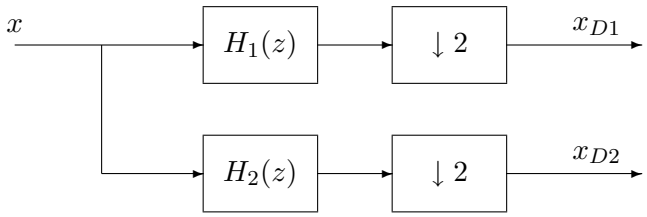

Figure 2.3: Signal decomposition into subband components.

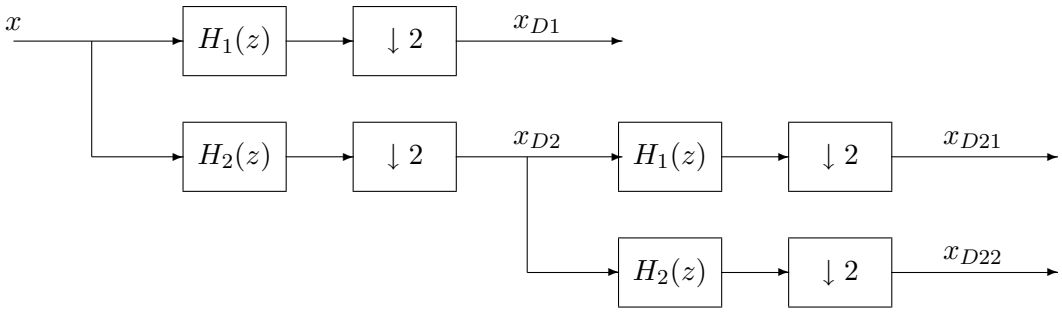

Figure 2.4: Filter bank for signal decomposition.

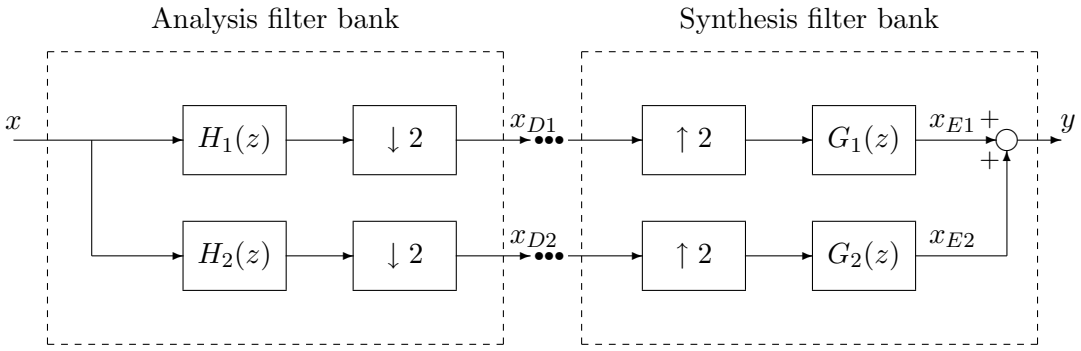

Figure 2.5: Multirate signal processing system with analysis and synthesis filter banks.

sampling rate. By (2.16), up-sampling generates aliasing frequencies. Therefore, the expanded signals should be filtered in order to extract the correct frequency components. The set of filters used to reconstruct the desired signal is called *synthesis filter bank*.

Obviously, the analysis and synthesis filter banks form important parts of multirate signal processing systems with subband processing. The incoming signal is first decomposed by an analysis filter bank, and the outgoing signal is constructed using a synthesis filter bank, cf. Figure 2.5.

2.2 Subband decomposition

In order to present the basic techniques involved in decomposing a signal into subband components, let's consider a simple case where a signal is decomposed into two components: a low-frequency component and a high-frequency component. See Figure 2.3. The purpose of the filters H_1 and H_2 is to extract the low- and high-frequency components of the signal x. For perfect signal decomposition, H_1 should be an ideal low-pass filter with the passband $[0, \pi/2]$, and H_2 should be an ideal high-pass filter with the passband $[\pi/2, \pi]$, cf. Figure 2.6. Filters having the characteristics shown in Figure 2.6 are called *brickwall filters*.

In order to gain insight into the subband decomposition problem, suppose for the moment that it were possible to design ideal brickwall filters with the responses shown in Figure 2.6. As the outputs of the brickwall filters are composed of frequencies in bands having widths $\pi/2$, they can be exactly represented using only half of the original sampling rate. More precisely, it follows from the properties of the brickwall filters H_1 and H_2 and the relation (2.13) that the down-sampled signals x_{D1} and x_{D2} in Figure 2.3 have the Fourier-transforms

$$X_{D1}(\omega) = \frac{1}{2} X(\omega/2), \;\; 0 \leq \omega < \pi \tag{2.18}$$

and

$$X_{D2}(\omega) = \frac{1}{2} X(\pi - \omega/2)^*, \;\; 0 \leq \omega < \pi \tag{2.19}$$

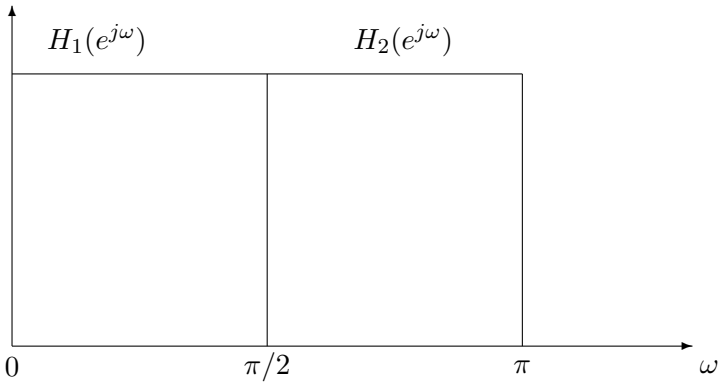

Figure 2.6: Brickwall filters.

Hence x_{D1} contains complete information of the low-frequency component of x, whereas x_{D2} contains complete information of the high-frequency component of x. Hence no information is lost in the decomposition, and it is thus possible to reconstruct the original signal x from the down-sampled components x_{D1} and x_{D2}.

Similarly, the synthesis filter bank (see Figure 2.5) should ideally consist of brickwall filters in order to reproduce the correct frequency band of the up-sampled signal, cf. equation (2.17).

In practice, however, it is not possible to construct ideal brickwall filters. Real filters characteristics resemble more the general form shown in Figure 2.7. It follows that it is not possible to separate the frequency bands exactly, but instead either some aliasing between the frequency bands is unavoidable, or, if the frequencies at the band edges are attenuated completely, some frequencies are lost altogether. The standard solution to the aliasing problem is to design the filters H_1 and H_2 in such a way that despite aliasing, it is still possible to reconstruct the original signal from the components. This can be achieved with *quadrature mirror filters*.

In order to describe the idea of quadrature mirror filters, consider the setup in Figure 2.8, showing the decomposition and reconstruction of a signal using two subband components. A desirable property of the analysis filters H_1 and H_2 and the synthesis filters G_1 and G_2 used in subband decomposition is that the output y is equal to the original signal x in Figure 2.8 when no processing of the subbands is performed.

Consider the signals in Figure 2.8. By the definitions of the down-sampling and up-sampling operators, v_1 and v_2 are given by

$$\{v_1(n)\} = \{x_1(0), 0, x_1(2), 0, x_1(4), 0, \ldots\} \tag{2.20}$$
$$\{v_2(n)\} = \{x_2(0), 0, x_2(2), 0, x_2(4), 0, \ldots\} \tag{2.21}$$

By (2.8) we then have

$$\hat{v}_1(z) = \frac{1}{2}\Big(\hat{x}_1(z) + \hat{x}_1(-z)\Big) \tag{2.22}$$

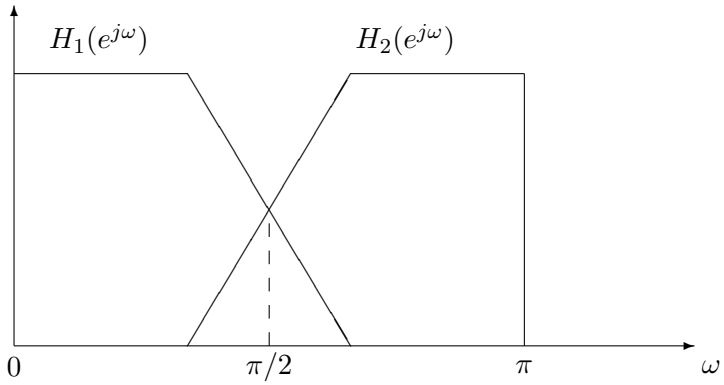

Figure 2.7: Characteristics of real filters in subband decomposition.

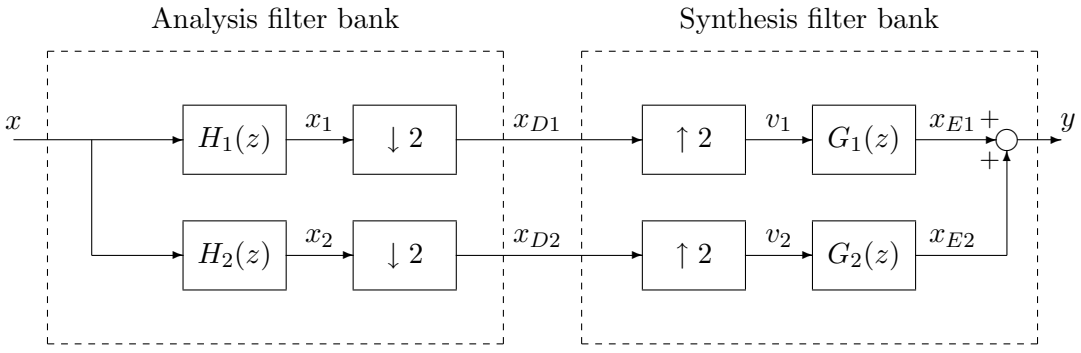

Figure 2.8: Signal decomposition and recovery with analysis and synthesis filter banks.

$$\hat{v}_2(z) = \frac{1}{2}\Big(\hat{x}_2(z) + \hat{x}_2(-z)\Big) \tag{2.23}$$

or

$$\hat{v}_1(z) = \frac{1}{2}\Big(H_1(z)\hat{x}(z) + H_1(-z)\hat{x}(-z)\Big) \tag{2.24}$$

$$\hat{v}_2(z) = \frac{1}{2}\Big(H_2(z)\hat{x}(z) + H_2(-z)\hat{x}(-z)\Big) \tag{2.25}$$

The output is given by

$$\begin{aligned}\hat{y}(z) &= \hat{x}_{E1}(z) + \hat{x}_{E2}(z) \\ &= G_1(z)\hat{v}_1(z) + G_2(z)\hat{v}_2(z)\end{aligned} \tag{2.26}$$

Combining (2.26) and (2.24), (2.25) gives

$$\begin{aligned}\hat{y}(z) \;=\; &\frac{1}{2}\Big(G_1(z)H_1(z) + G_2(z)H_2(z)\Big)\hat{x}(z) \\ &+\frac{1}{2}\Big(G_1(z)H_1(-z) + G_2(z)H_2(-z)\Big)\hat{x}(-z)\end{aligned} \tag{2.27}$$

The first term in (2.27) is the desired output signal, whereas the second term represents the spurious component caused by aliasing. Aliasing is avoided if we require that

$$G_1(z)H_1(-z) + G_2(z)H_2(-z) = 0 \tag{2.28}$$

This can be achieved in a straightforward way by selecting $G_1(z)$ and $G_2(z)$ as

$$G_1(z) = 2H_2(-z) \tag{2.29}$$

$$G_2(z) = -2H_1(-z) \tag{2.30}$$

where the scaling factors 2 are introduced for convenience.

Moreover, if the analysis filter $H_1(z)$ is a lowpass filter, then $H_2(z)$ is typically its symmetric highpass filter, i.e., cf. equation (1.11),

$$H_2(z) = H_1(-z) \tag{2.31}$$

Then (2.29) and (2.30) imply

$$G_1(z) = 2H_1(z) \tag{2.32}$$

$$G_2(z) = -2H_2(z) \tag{2.33}$$

Combining the expressions for G_1 and G_2 with the expression (2.27) for the system output gives

$$\hat{y}(z) = \Big(H_1(z)^2 - H_1(-z)^2\Big)\,\hat{x}(z) \tag{2.34}$$

Equation (2.34) implies that there is no aliasing affecting the output, which is a very desirable property indeed. The condition for exact signal recovery is

$$H_1(z)^2 - H_1(-z)^2 = Kz^{-P} \tag{2.35}$$

where K is a constant and P denotes a time delay which cannot be avoided when using causal filters.

The filters which satisfy the relations (2.31)–(2.33) are called quadrature mirror filters, because the frequency responses of the two input and output filters are mirror images about the quadrature frequency $2\pi/4 = \pi/2$.

Haar filters.
The only FIR filter which achieves perfect reconstruction according to (2.35) is the Haar filter $H_1(z) = \frac{1}{2}(1 + z^{-1})$. By the procedure described above, the other filters are then uniquely defined as follows:

$$H_1(z) = \frac{1}{2}\left(1 + z^{-1}\right), \;\; G_1(z) = 1 + z^{-1} \tag{2.36}$$

$$H_2(z) = \frac{1}{2}\left(1 - z^{-1}\right), \;\; G_2(z) = -1 + z^{-1} \tag{2.37}$$

and the relation (2.34) reduces to

$$\hat{y}(z) = z^{-1}\hat{x}(z) \tag{2.38}$$

which corresponds to a simple time delay, i.e., the output is given by $y(n) = x(n-1)$. The subband decomposition and reconstruction with the Haar filters is illustrated by the following example

Example 2.1 Consider the sequence

$$\{x(n)\} = \{2, 6, 4, 8\}$$

Referring to Figure 2.8 and the filters (2.36), (2.37), the sequence $\{x_1(n)\}$ is defined as $x_1(n) = \frac{1}{2}(x(n) + x(n-1))$, and the sequence $\{x_2(n)\}$ is defined as $x_2(n) = \frac{1}{2}(x(n) - x(n-1))$, giving

$$\{x_1(n)\} = \{1, 4, 5, 6, 4\}, \;\; \{x_2(n)\} = \{1, 2, -1, 2, -4\}$$

Down-sampling gives the subband components

$$\{x_{D1}(n)\} = \{1, 5, 4\}, \;\; \{x_{D2}(n)\} = \{1, -1, -4\}$$

For reconstruction, the signals $\{x_{D1}(n)\}$ and $\{x_{D2}(n)\}$ are up-sampled, to give

$$\{v_1(n)\} = \{1, 0, 5, 0, 4, 0\}, \;\; \{v_2(n)\} = \{1, 0, -1, 0, -4, 0\}$$

Finally, the filters $G_1(z) = 1+z^{-1}$ and $G_2(z) = -1+z^{-1}$ give $x_{E1}(n) = v_1(n)+v_1(n-1)$ and $x_{E2}(n) = -v_2(n) + v_2(n-1)$, or

$$\{x_{E1}(n)\} = \{1, 1, 5, 5, 4, 4\}, \;\; \{x_{E2}(n)\} = \{-1, 1, 1, -1, 4, -4\}$$

and the reconstructed signal $y(n) = x_{E1}(n) + x_{E2}(n)$ is thus

$$\{y(n)\} = \{0, 2, 6, 4, 8, 0\}$$

In compliance with (2.38), $\{y(n)\}$ is a delayed version of the original signal $\{x(n)\}$.

Example 2.1 illustrates one practical difficulty encountered when applying causal filters only: in order to preserve all information, the intermediate signals may have to have varying lengths. For example, while the signal length in Example 2.1 is $N = 4$, x_1 and x_2 have lengths $N + 1 = 5$, and the decimated signal have lengths $N/2 + 1 = 3$. The reason for having to deal with varying signal lengths is the fact that the causal filters introduce a time delay, which shifts the signal in time.

The subband decomposition procedure can be made in a more streamlined manner by introducing non-causal filters. This is usually not restrictive in subband decomposition. Instead of the causal filters (2.36) and (2.37), introduce the filters

$$H_1(z) = \frac{1}{2}(z+1), \;\; G_1(z) = 1 + z^{-1} \tag{2.39}$$

$$H_2(z) = \frac{1}{2}(z-1), \;\; G_2(z) = -1 + z^{-1} \tag{2.40}$$

where $H_1(z)$ and $H_2(z)$ are non-causal, while $G_1(z)$ and $G_2(z)$ are defined as before. The filters (2.39) and (2.40) satisfy the perfect reconstruction condition (2.35), and the relation (2.34) reduces to

$$\hat{y}(z) = \hat{x}(z) \tag{2.41}$$

i.e., perfect reconstruction without delay is achieved. Decomposition and reconstruction using the Haar filters (2.39) and (2.40) can be performed in a very systematic way, as shown by the following example.

Example 2.2 We consider again the sequence

$$\{x(n)\} = \{2, 6, 4, 8\}$$

By Figure 2.8 and the filters (2.39), (2.40), the sequence $\{x_1(n)\}$ is now obtained as $x_1(n) = \frac{1}{2}(x(n+1) + x(n))$, and the sequence $\{x_2(n)\}$ as $x_2(n) = \frac{1}{2}(x(n+1) - x(n))$, giving

$$\{x_1(n)\} = \{4, 5, 6, 4\}, \;\; \{x_2(n)\} = \{2, -1, 2, -4\}$$

Down-sampling gives the subband components

$$\{x_{D1}(n)\} = \{4, 6\}, \;\; \{x_{D2}(n)\} = \{2, 2\}$$

Notice in particular, that $\{x_{D1}(n)\}$ and $\{x_{D2}(n)\}$ are obtained by pairwise averaging and differencing of the elements of $\{x(n)\}$.

For reconstruction, the signals $\{x_{D1}(n)\}$ and $\{x_{D2}(n)\}$ are up-sampled, to give

$$\{v_1(n)\} = \{4, 0, 6, 0\}, \;\; \{v_2(n)\} = \{2, 0, 2, 0\}$$

The filters $G_1(z) = 1 + z^{-1}$ and $G_2(z) = -1 + z^{-1}$ give finally $x_{E1}(n) = x(n) + x(n-1)$ and $x_{E2}(n) = -x(n) + x(n-1)$, or

$$\{x_{E1}(n)\} = \{4, 4, 6, 6\}, \;\; \{x_{E2}(n)\} = \{-2, 2, -2, 2\}$$

Hence $y(n) = x_{E1}(n) + x_{E2}(n)$ is

$$\{y(n)\} = \{2, 6, 4, 8\}$$

and perfect reconstruction of the original signal $\{x(n)\}$ is achieved. Notice that $\{y(n)\}$ is obtained by alternately forming differences and sums of the elements of $\{x_{D1}(n)\}$ and $\{x_{D2}(n)\}$.

Generalizing the calculations of Example 2.2, the subband decomposition procedure using Haar filters may be described as follows. Referring to Figure 2.8 and the filters (2.39), (2.40), the sequence $\{x_1(n)\}$ is defined as $x_1(n) = \frac{1}{2}(x(n+1) + x(n))$, and the decimated signal $\{x_{D1}(n)\}$ is thus given by

$$\begin{aligned} x_{D1}(0) &= x_1(0) = \frac{1}{2}(x(1) + x(0)) \\ x_{D1}(1) &= x_1(2) = \frac{1}{2}(x(3) + x(2)) \\ &\vdots \qquad (2.42) \\ x_{D1}(m) &= x_1(2m) = \frac{1}{2}(x(2m+1) + x(2m)) \\ &\vdots \qquad (2.43) \end{aligned}$$

Hence the decimated signal is obtained by averaging of successive, non-overlapping, pairs of $\{x(n)\}$. Similarly, the sequence $\{x_2(n)\}$ is defined as $x_1(n) = \frac{1}{2}(x(n+1) - x(n))$, and the decimated signal $\{x_{D2}(n)\}$ is thus given by

$$\begin{aligned} x_{D2}(0) &= x_2(0) = \frac{1}{2}(x(1) - x(0)) \\ x_{D2}(1) &= x_2(2) = \frac{1}{2}(x(3) - x(2)) \\ &\vdots \qquad (2.44) \\ x_{D2}(m) &= x_2(2m) = \frac{1}{2}(x(2m+1) - x(2m)) \\ &\vdots \qquad (2.45) \end{aligned}$$

i.e., the decimated signal is obtained by forming the differences between successive (non-overlapping) pairs of $\{x(n)\}$.

On the reconstruction side, up-sampling of $\{x_{D1}(n)\}$ gives

$$\{v_1(n)\} = \{x_{D1}(0), 0, x_{D1}(1), 0, x_{D1}(2), 0, \ldots\}$$

and the filter $G_1(z)$ implies $x_{E1}(n) = v_1(n) + v_1(n-1)$, i.e.,

$$\{x_{E1}(n)\} = \{x_{D1}(0), x_{D1}(0), x_{D1}(1), x_{D1}(1), x_{D1}(2), x_{D1}(2), \ldots\} \qquad (2.46)$$

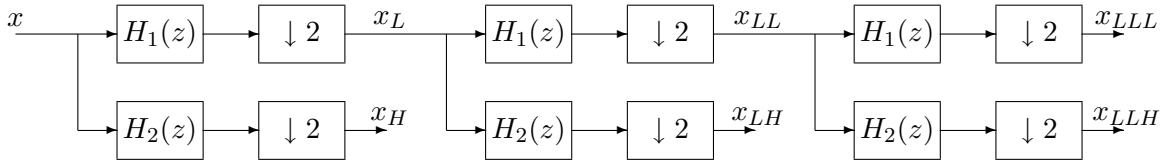

Figure 2.9: Filter bank for subband coding.

Similarly, we have

$$\{x_{E2}(n)\} = \{-x_{D2}(0), x_{D2}(0), -x_{D2}(1), x_{D2}(1), -x_{D2}(2), x_{D2}(2), \ldots\} \tag{2.47}$$

The relations (2.46) and (2.47) imply that the reconstructed signal is obtained as

$$\begin{aligned}\{y(n)\} \;=\; \{&x_{D1}(0) - x_{D2}(0), x_{D1}(0) + x_{D2}(0), x_{D1}(1) - x_{D2}(1),\\ &x_{D1}(1) + x_{D2}(1), x_{D1}(2) - x_{D2}(2), \ldots\}\end{aligned} \tag{2.48}$$

or

$$y(2m) = x_{D1}(m) - x_{D2}(m) \tag{2.49}$$

$$y(2m+1) = x_{D1}(m) + x_{D2}(m), \;\; m = 0, 1, 2, \ldots N/2 - 1 \tag{2.50}$$

i.e., the elements are obtained by forming the differences and sums of the elements of $\{x_{D1}(n)\}$ and $\{x_{D2}(n)\}$.

The Haar filter is a standard tool in subband coding and multiresolution analysis. This topic will be developed in more detail in Section 2.3.

The Haar filter is the only FIR quadrature mirror filter which achieves perfect reconstruction according to (2.35). Unfortunately, the performance of the filter in subband coding is limited by the fact that it does not provide a very good separation of the frequency bands. There are, however, methods to design more efficient filters which also achieve perfect reconstruction. This will be discussed later.

2.3 Subband coding and multiresolution analysis

A number of subband coding and multiresolution techniques are based on the subband decomposition and reconstruction procedure described in Section 2.2. A common filter structure for subband decomposition is the filter bank in Figure 2.9, which shows a three-level decomposition scheme. The filter bank performs a number of successive lowpass filtering and down-sampling operations. The signal from each stage is high-pass filtered and down-sampled. This process generates the transformed signals $x_H, x_{LH}, x_{LLH}, x_{LLL}$ (for $K = 3$ levels), which have a total length equal to that of the original signal. The process shown in Figure 2.9 is called the *pyramid algorithm*, due to the structure of the filter bank.

The filters H_1 and H_2 are often taken as the Haar filters (2.39) and (2.40), i.e.,

$$H_1(z) = \frac{1}{2}\Big(z+1\Big) \tag{2.51}$$

$$H_2(z) = \frac{1}{2}\Big(z-1\Big) \tag{2.52}$$

Referring to Figures 2.8 and 2.9, the original signal in Figure 2.9 is obtained recursively as

$$\begin{aligned} \hat{x}_{LL}(z) &= G_1(z)\hat{v}_{LLL} + G_2(z)\hat{v}_{LLH} \\ \hat{x}_{L}(z) &= G_1(z)\hat{v}_{LL} + G_2(z)\hat{v}_{LH} \\ \hat{x}(z) &= G_1(z)\hat{v}_{L} + G_2(z)\hat{v}_{H} \end{aligned} \tag{2.53}$$

where $G_1 = 1+z^{-1}$ and $G_2 = -1+z^{-1}$ (cf. (2.39) and (2.40)), and $\hat{v}_{LLL}, \hat{v}_{LLH}, \ldots$ denote the up-sampled versions of $\hat{x}_{LLL}, \hat{x}_{LLH}, \ldots$. By construction of the filters H_1, H_2, G_1 and G_2, perfect reconstruction of the original signal x is achieved.

Recall from Section 2.2 that the filter $H_1(z)$ forms the average of two successive signal values, so that the function of $H_1(z)$ followed by down-sampling is equivalent to forming pairwise averages of the signal. It follows that the low-pass filtered signals in Figure 2.9 are given by

$$\begin{aligned} x_L(m) &= \frac{1}{2}\Big(x(2m+1) + x(2m)\Big) \\ x_{LL}(m) &= \frac{1}{2}\Big(x_L(2m+1) + x_L(2m)\Big) \\ &= \frac{1}{4}\Big(x(4m+3) + x(4m+2) + x(4m+1) + x(4m)\Big) \\ x_{LL}(m) &= \frac{1}{8}\Big(x(8m+7) + x(8m+6) + x(8m+5) + x(8m+4) \\ &\quad + x(8m+3) + x(8m+2) + x(8m+1) + x(8m)\Big) \end{aligned}$$

and, thus, they define signal averages at various resolutions. Similarly, the filter $H_2(z)$ followed by down-sampling is equivalent to forming pairwise differences of the signal values. Hence

$$\begin{aligned} x_H(m) &= \frac{1}{2}\Big(x(2m+1) - x(2m)\Big) \\ x_{LH}(m) &= \frac{1}{2}\Big(x_L(2m+1) - x_L(2m)\Big) \\ x_{LLH}(m) &= \frac{1}{2}\Big(x_{LL}(2m+1) - x_{LL}(2m)\Big) \end{aligned} \tag{2.54}$$

and the components x_H, x_{LH} and x_{LLH} therefore describe the variation of the signal x at various resolutions. The decomposition described here is called Haar multiresolution.

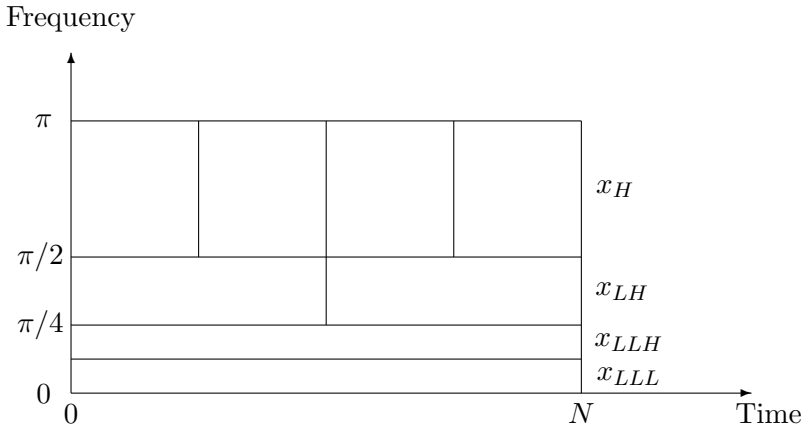

Figure 2.10: Time-frequency resolutions of a signal of length $N = 8$ with the three-stage filter bank in Figure 2.9.

The algorithm can be regarded as a transform of the input signal x defined by merging the outputs of the pyramid filter in Figure 2.9,

$$x_{Haar} = [x_{LLL}, x_{LLH}, x_{LH}, x_H] \tag{2.55}$$

In subband coding based on Haar multiresolution, the signal is compressed by discarding sufficiently small elements of the various components of x_{Haar}, setting them equal to zero, i.e., defining the truncated values

$$\tilde{x}_{Haar}(m) = \begin{cases} x_{Haar}(m), & \text{if } |x_{Haar}(m)| > d \\ 0, & \text{if } |x_{Haar}(m)| \leq d \end{cases}, \tag{2.56}$$

where $d > 0$ is a specified threshold which determines the allowed approximation error. The procedure exploits the fact that the signal variations can locally often be described in a very simple way: in regions where the variations are slow, an accurate representation is achieved with components having the coursest resolution (such as x_{LLL} and x_{LLH}), whereas components with higher resolution (such as x_H) describe the signal best in regions with stronger variations. The approach thus provides a resolution of the signal in both the time and frequency domains. In contrast, the standard Fourier-transform provides only a frequency-domain resolution. The time-frequency resolution is illustrated in Figure 2.10 for a signal of length $N = 8$ for the three-stage filter bank in Figure 2.9. It is seen that as the frequency decreases, the time resolution decreases and the frequency resolution increases in such a way that their product remains constant. This is in compliance with the fact that a longer time period is required to represent a lower frequency.

The following example is often used in the literature to demonstrate the multiresolution technique.

Problem 2.1
Use the Haar multiresolution technique with three levels ($K = 3$) to decompose the signal

$$\{x(n)\} = \{37, 35, 28, 28, 58, 18, 21, 15\} \tag{2.57}$$

into subband components. Reconstruct the original signal from the components using

- exact representation of the subband components, and

- data compression according to (2.56) using the thresholds $d = 2, 3$ and 4. How much compression is achieved in each case?

The signal $\{x(n)\}$ and the reconstructed signals with various thresholds are shown in Figure 2.11. In contrast, in the Fourier transform of $\{x(n)\}$ all frequency components are required to represent the signal accurately. This is due to the fact that the Fourier transform provides no time resolution.

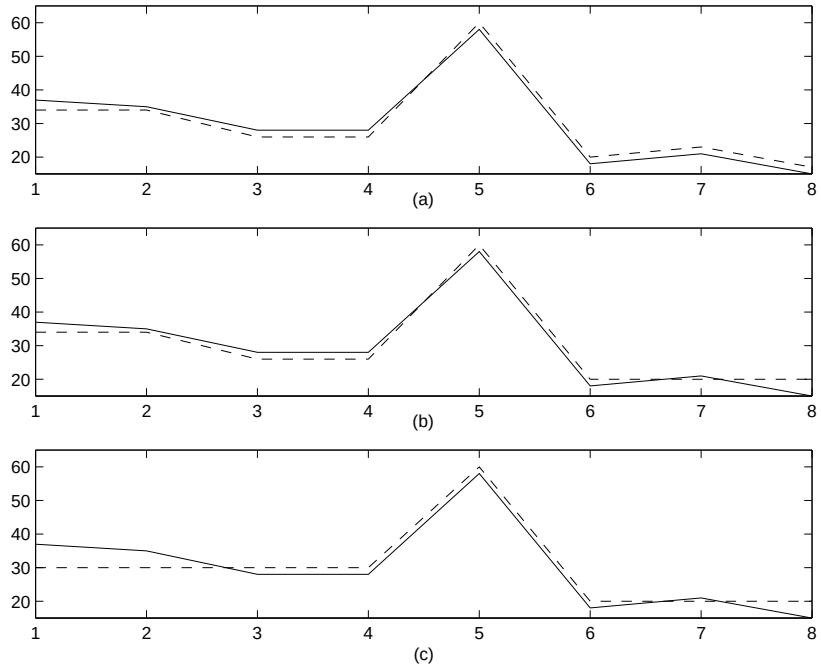

Figure 2.11: Original (full lines) and reconstructed (dashed lines) signals in Problem 2.1 for various thresholds d. (a) $d = 2$, (b) $d = 3$ and (c) $d = 4$.

2.4 The discrete wavelet transform

The multiresolution decomposition described in Section 2.3 is closely related to the discrete wavelet transform. In this section we will make use of this connection to

characterize the discrete wavelet transform in terms of the pyramid filter bank in Figure 2.9.

A discrete wavelet expansion of a signal $\{x(n)\}$ of length N is defined as

$$x(n) = \sum_{m=0}^{N/2^J-1} X_{DWT}(0,m)\phi(2^{-J}n-m) + \sum_{i=1}^{J}\sum_{m=0}^{N/2^i-1} X_{DWT}(i,m)\psi(2^{-i}n-m) \quad (2.58)$$

where $\phi(t)$ and $\psi(t)$ are given functions (wavelets), and the set of coefficients $X_{DWT}(m,i)$ is the discrete wavelet transform (DWT) of the signal $\{x(n)\}$. A characteristic feature of the expansion (2.58) is that the signal is expanded in terms of the functions $\phi(t)$ and $\psi(t)$ as well as *dilated* (or stretched) and translated copies of the function $\psi(t)$. Notice that if $\psi(t)$ is defined for $0 \le t < 1$, and is zero elsewhere, then:

- $\psi(2^{-i}n)$ is defined for $0 \le n < 2^i$, and is thus a dilated (stretched) version of $\psi(t)$,
- $\psi(2^{-i}n-m)$ is defined for $m2^i \le n < (m+1)2^i$, and is thus a dilated and translated version of $\psi(t)$.

The situation is illustrated in Figure 2.12, which shows a function together with a dilated and a dilated and translated copies of it. In standard wavelet jargon, $\phi(t)$ is the *father wavelet*, $\psi(t)$ is the *mother wavelet*, and the dilated and translated copies $\psi(2^{-i}n-m)$ are called *daughter wavelets*.

The number J in (2.58) defines the number of resolution levels. For a signal of length $N = 2^K$, $J \le K$.

By its construction, and in analogy with the multiresolution decomposition in Section 2.3, the wavelet transform provides a resolution of the signal in both the time and frequency domains. It turns out that the wavelet transform in (2.58) can be characterized in terms of a pyramid filter bank of the form shown in Figure 2.9 (with $J = 3$). Here, the filters $H_1(z)$ and $H_2(z)$ are associated with the father and mother wavelets $\phi(t)$ and $\psi(t)$, respectively, the down-samplers describe dilation by the factor two, and the discrete wavelet transform $X_{DWT}(i,m)$ consists of the filter outputs $x_{LLL}, x_{LLH}, x_{LH}, x_H$. The perfect reconstruction condition ensures that the original signal x can be expressed in terms of the transform according to (2.58).

The connection between multiresolution filter banks and wavelets will be explored in more detail below. In Section 2.4.1 it is first shown that the Haar multiresolution described in Section 2.3 can be characterized as a wavelet transform. The result paves the way to the more general Daubechies wavelets which will be described in Section 2.4.2.

2.4.1 Haar wavelets

In order to introduce the Haar wavelet, consider how the original signal x in Figure 2.9 is reconstructed from the components x_H, x_{LH}, x_{LLH} and x_{LLL}. Referring to Section 2.2 and Figure 2.8, we have (cf. equation (2.49))

$$\begin{aligned} x_{LL}(2m) &= x_{LLL}(m) - x_{LLH}(m) \\ x_{LL}(2m+1) &= x_{LLL}(m) + x_{LLH}(m), \end{aligned} \quad (2.59)$$

$m = 0, 1, \ldots, N/2 - 1$. Similarly,

$$\begin{aligned} x_L(4m) &= x_{LL}(2m) - x_{LH}(2m) \\ &= x_{LLL}(m) - x_{LLH}(m) - x_{LH}(2m) \\ x_L(4m+1) &= x_{LL}(2m) + x_{LH}(2m) \\ &= x_{LLL}(m) - x_{LLH}(m) + x_{LH}(2m) \\ x_L(4m+2) &= x_{LL}(2m+1) - x_{LH}(2m+1) \\ &= x_{LLL}(m) + x_{LLH}(m) - x_{LH}(2m+1) \\ x_L(4m+3) &= x_{LL}(2m+1) + x_{LH}(2m+1) \\ &= x_{LLL}(m) + x_{LLH}(m) + x_{LH}(2m+1), \end{aligned} \tag{2.60}$$

$m = 0, 1, \ldots, N/4 - 1$. Finally, the signal x is reconstructed according to

$$\begin{aligned} x(8m) &= x_L(4m) - x_H(4m) \\ x(8m+1) &= x_L(4m) + x_H(4m) \\ x(8m+2) &= x_L(4m+1) - x_H(4m+1) \\ x(8m+3) &= x_L(4m+1) + x_H(4m+1) \\ x(8m+4) &= x_L(4m+2) - x_H(4m+2) \\ x(8m+5) &= x_L(4m+2) + x_H(4m+2) \\ x(8m+6) &= x_L(4m+3) - x_H(4m+3) \\ x(8m+7) &= x_L(4m+3) + x_H(4m+3) \end{aligned}$$

Introducing the expressions from (2.60) gives

$$\begin{aligned} x(8m) &= x_{LLL}(m) - x_{LLH}(m) - x_{LH}(2m) - x_H(4m) \\ x(8m+1) &= x_{LLL}(m) - x_{LLH}(m) - x_{LH}(2m) + x_H(4m) \\ x(8m+2) &= x_{LLL}(m) - x_{LLH}(m) + x_{LH}(2m) - x_H(4m+1) \\ x(8m+3) &= x_{LLL}(m) - x_{LLH}(m) + x_{LH}(2m) + x_H(4m+1) \\ x(8m+4) &= x_{LLL}(m) + x_{LLH}(m) - x_{LH}(2m+1) - x_H(4m+2) \\ x(8m+5) &= x_{LLL}(m) + x_{LLH}(m) - x_{LH}(2m+1) + x_H(4m+2) \\ x(8m+6) &= x_{LLL}(m) + x_{LLH}(m) + x_{LH}(2m+1) - x_H(4m+3) \\ x(8m+7) &= x_{LLL}(m) + x_{LLH}(m) + x_{LH}(2m+1) + x_H(4m+3), \end{aligned}$$

$m = 0, 1, \ldots, N/8 - 1$. We see that the coefficients for x_{LLL}, x_{LLH}, x_{LH} and x_H have a very characteristic pattern. Indeed, introducing the *Haar function*

$$\psi_H(t) = \begin{cases} 1, & 0 \le t < 1/2 \\ -1, & 1/2 \le t < 1 \\ 0, & \text{otherwise,} \end{cases} \tag{2.61}$$

and the function

$$\phi_H(t) = \begin{cases} 1, & 0 \le t < 1 \\ 0, & \text{otherwise,} \end{cases} \tag{2.62}$$

we see that x can be expressed as

$$x(n) = \sum_{i=0}^{N/2^3-1} x_{LLL}(i)\phi_H(2^{-3}n-i) - \sum_{i=0}^{N/2^3-1} x_{LLH}(i)\psi_H(2^{-3}n-i) - \sum_{i=0}^{N/2^2-1} x_{LH}(i)\psi_H(2^{-2}n-i) - \sum_{i=0}^{N/2-1} x_H(i)\psi_H(2^{-1}n-i) \quad (2.63)$$

This expression is a wavelet decomposition of the signal x, where the wavelet transform $X_{DWT}(i,m)$ is given by the sequences $x_{LLL}, -x_{LLH}, -x_{LH}$ and $-x_H$, and the associated wavelet is defined by (2.62) and (2.61), cf. equation (2.58). The Haar wavelet (2.61) and two dilated and translated copies (of it are shown in Figure 2.12.

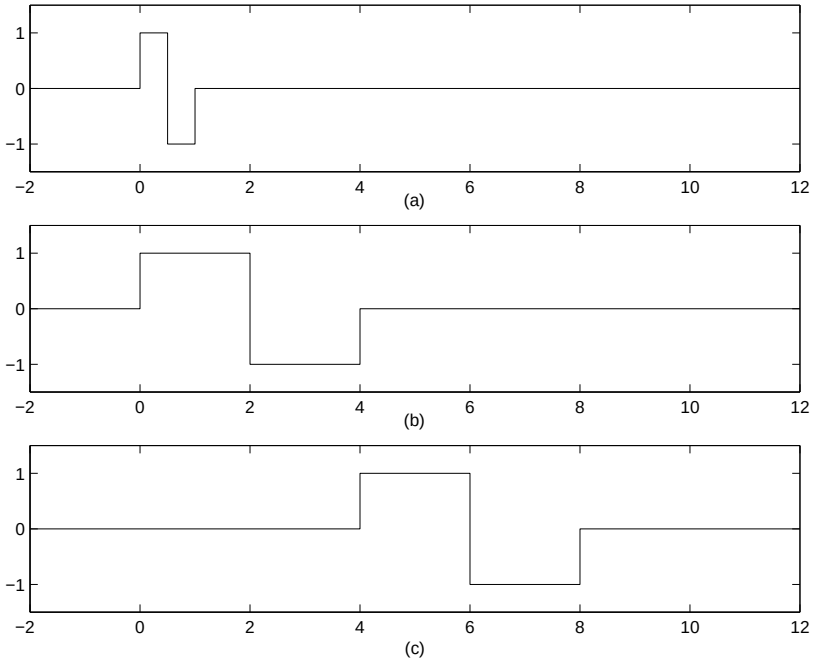

Figure 2.12: Haar wavelets. (a) $\psi_H(t)$, (b) dilated copy $\psi_H(2^{-2}t)$ and (c) dilated and translated copy $\psi_H(2^{-2}t-1)$.

2.4.2 Daubechies wavelets

In Section 2.4.1 we have seen that the pyramid algorithm based on the Haar filters is equivalent to the Haar wavelet transform. The Haar filter does not provide a very good separation of the frequency bands, however, and it is therefore well motivated to study more complex filters and transforms. The equivalence between pyramid filter banks and wavelet transforms can be generalized to other filters and wavelet transforms. In

order to obtain a satisfactory wavelet transform, the associated filter bank should have the following properties:

- The filters $H_1(z)$ and $H_2(z)$ should be FIR filters, and they should provide a good frequency separation,
- the filters $H_1(z)$ and $H_2(z)$ should be related in such a way that the filter bank generates a wavelet expansion of the form (2.58) for some wavelet function $\phi(t)$ and $\psi(t)$.
- the filter bank should have the perfect reconstruction property.

It is not at all evident that the above conditions can be satisfied. Recall for example from Section 2.2 that the Haar filter is the only FIR quadrature mirror filter which achieves the perfect reconstruction property. The fact that there does indeed exist a class of filters which satisfy the conditions has been shown in a remarkable result by Ingrid Daubechies (1988).

In order to present the Daubechies wavelets and the associated filter bank, recall the signal decomposition according to Figure 2.8. For alias elimination, we require that (2.28) holds, and select the synthesis filters as

$$G_1(z) = z^{-M} H_2(-z) \tag{2.64}$$
$$G_2(z) = -z^{-M} H_1(-z) \tag{2.65}$$

where the time delay z^{-M} is introduced in order to avoid a time advance in the reconstruction process. In contrast to Section 2.2, the analysis filters $H_1(z)$ and $H_2(z)$ will not be selected to satisfy the quadrature mirror symmetry property (2.31). Instead, it can be shown that the filter bank corresponds to a wavelet transform if the filters are chosen to satisfy

$$H_2(z) = -z^M H_1(-z^{-1}) \tag{2.66}$$

where M is the order of $H_1(z)$ (so that the number of coefficients of the FIR filter $H_1(z)$ is $M+1$). Daubechies filters are only defined for odd values of M. Combining the relations (2.64)–(2.66) gives

$$\begin{aligned} H_1(z) &= H(z) \\ H_2(z) &= -z^M H(-z^{-1}) \\ G_1(z) &= H(z^{-1}) \quad (M \text{ odd}) \\ G_2(z) &= -z^{-M} H(-z) \end{aligned} \tag{2.67}$$

More explicitly, if

$$H(z) = h_0 + h_1 z + h_2 z^2 + \cdots + h_M z^M, \tag{2.68}$$

(2.67) implies

$$\begin{aligned} H_1(z) &= h_0 + h_1 z + h_2 z^2 + \cdots + h_M z^M \\ H_2(z) &= -h_0 z^M + h_1 z^{M-1} - h_2 z^{M-2} + \cdots + h_M \\ G_1(z) &= h_0 + h_1 z^{-1} + h_2 z^{-2} + \cdots + h_M z^{-M} \\ G_2(z) &= -h_0 z^{-M} + h_1 z^{-M+1} - h_2 z^{-M+2} + \cdots + h_M \end{aligned} \tag{2.69}$$

Notice the symmetries between $H_1(z)$ and $G_1(z)$, and $H_2(z)$ and $G_2(z)$. From (2.27) it follows that in order to achieve perfect reconstruction, we should have

$$\frac{1}{2}\Big(G_1(z)H_1(z) + G_2(z)H_2(z)\Big) = 1 \tag{2.70}$$

Introducing (2.67) it follows that (2.70) implies the following condition on $H_1(z) = H(z)$:

$$\frac{1}{2}\Big(H(z)H(z^{-1}) + H(-z)H(-z^{-1})\Big) = 1 \tag{2.71}$$

D The problem has thus been reduced to finding a FIR filter $H(z)$ which satisfies (2.71). It turns out that the solutions of (2.71) have the form

$$H(z) = \sqrt{2}\left(\frac{1+z}{2}\right)^m S(z) \tag{2.72}$$

where $S(z)$ is a FIR filter, which is normalized so that $S(1) = 1$, i.e., the filter $S(z)$ has a unit stationary gain. It can be shown (after some algebraic manipulations) that the perfect reconstruction condition (2.71) implies that $S(z)$ should satisfy

$$|S(e^{j\omega})|^2 = \sum_{k=0}^{m-1} \binom{m+k-1}{k} \sin^{2k}\left(\frac{\omega}{2}\right) \tag{2.73}$$

It turns out that for $m = 1, 2, 3, \ldots$, equation (2.73) has a polynomial solution $S(z)$ of order $m-1$. It follows that the FIR filters $H(z)$ in (2.72) have orders $M = m+m-1 = 2m-1$, i.e., $M = 1, 3, 5, \ldots$. Notice that for the case $m = 1$, $H(z)$ reduces to the Haar filter (scaled by the factor $\sqrt{2}$).

In analogy with the Haar filter, the filters (2.72) generate a wavelet transform when applied in a pyramid filter bank. The associated wavelet is called the Daubechies wavelet of order $m = 1, 2, 3, \ldots$. The Haar wavelet discussed above is thus the simplest Daubechies wavelet, corresponding to $m = 1$.

Example 2.3 The Daubechies wavelet of order 2.
As an example, consider the Daubechies wavelet of order $m = 2$. Some algebra shows that equation (2.73) has the solution

$$S(z) = \frac{1-\sqrt{3}}{2} + \frac{\sqrt{3}+1}{2}z \tag{2.74}$$

By (2.72), $H(z)$ is then given by

$$\begin{aligned} H(z) &= \sqrt{2}\left\{\frac{1-\sqrt{3}}{8} + \frac{3-\sqrt{3}}{8}z + \frac{3+\sqrt{3}}{8}z^2 + \frac{1+\sqrt{3}}{8}z^3\right\} \\ &= -0.129409522551 + 0.224143868042z + 0.836516303738z^2 + 0.482962913145z^3 \end{aligned}$$

Similarly, one can show that the filter $H(z)$ associated with the Daubechies wavelet of order $m = 3$ has the coefficients

$$h_0 = \frac{\sqrt{2}}{32}\left(1 + \sqrt{10} - \sqrt{5 + 2\sqrt{10}}\right) = 0.0352262918857095$$
$$h_1 = \frac{\sqrt{2}}{32}\left(5 + \sqrt{10} - 3\sqrt{5 + 2\sqrt{10}}\right) = -0.0854412738820267$$
$$h_2 = \frac{\sqrt{2}}{32}\left(10 - 2\sqrt{10} - 2\sqrt{5 + 2\sqrt{10}}\right) = -0.1350110200102546$$
$$h_3 = \frac{\sqrt{2}}{32}\left(10 - 2\sqrt{10} + 2\sqrt{5 + 2\sqrt{10}}\right) = 0.4598775021184914$$
$$h_4 = \frac{\sqrt{2}}{32}\left(5 + \sqrt{10} + 3\sqrt{5 + 2\sqrt{10}}\right) = 0.8068915093110924$$
$$h_5 = \frac{\sqrt{2}}{32}\left(1 + \sqrt{10} + \sqrt{5 + 2\sqrt{10}}\right) = 0.3326705529500825$$

There are efficient algorithms for calculating the filter coefficients of higher-order Daubechies wavelets. The coefficients have also been tabulated in the literature.

2.4.3 Applications of wavelets

The time-frequency properties of wavelets make them a powerful tool for a number of signal processing problems, where more traditional techniques may perform poorly. These include:

- *Data compression.* Wavelets are used extensively for lossy data compression using thresholding according to equation (2.56). The JPEG 2000 image compression standard is based on Daubechies wavelet transform and thresholding.

- *De-noising.* The wavelet transform and thresholding is also useful for removing noise in a signal. This approach is justified by the assumption that small elements of the wavelet transform are caused by noise, whereas the true signal can be represented by the larger elements. Wavelet-based noise removal is very efficient for example in cases where the signal has spikes, which are part of the true signal and should not be removed (cf. the signal in Problem 2.1, which has a spike at $n = 5$). As the frequency contents of the spikes have considerable high-frequency components, standard noise filtering techniques tend to result in unwanted distortion of the spikes. In contrast, the spikes will contribute with large elements to the wavelet transform of the signal, and will therefore be unaffected by thresholding.

In image processing applications the signal $x(n, m)$ representing an image is two-dimensional, consisting of an N by M array $\{x(n, m)\}, n = 0, 1, \ldots, N - 1, m = 0, 1, \ldots, M - 1$. The wavelet transform of a two-dimensional signal $\{x(n, m)\}$ is obtained as a straightforward generalization of the one-dimensional case, by first taking

the transform of each row, followed by the transform of each column. More precisely, first each row of $\{x(n,m)\}$ is transformed to give the array $X_{DWT,row}(n,k)$, where the nth row $X_{DWT,row}(n,k), k = 0,1,\ldots,M-1$, is the transform of the one-dimensional signal consisting of the nth row $x(n,m), m = 0,1,\ldots,M-1$. Then each column of $X_{DWT,row}(n,k)$ is transformed to give $X_{DWT}(l,k)$, where the kth column $X_{DWT}(l,k), l = 0,1,\ldots,N-1$, is the transform of the one-dimensional signal consisting of the kth column $X_{DWT,row}(n,k), n = 0,1,\ldots,N-1$.

Efficient software exists for wavelet analysis. A library of MATLAB routines for wavelet calculations is available at `http://www-stat.stanford.edu/~wavelab/`.