3

Signal Processing and Signals

Sound and voice cause vibration or wave propagation in a medium. If we register the value of a vibration of a wave field variable in a spatial position as a function of time, the result is a sound *signal*. This can be performed using a microphone or a vibration sensor, resulting in an electrical signal that can be processed and stored. Sound signals in electrical form can also be reconverted to sound by using loudspeakers.

Signal processing is the branch of engineering that provides efficient methods and techniques to analyse, synthesize, and transform signals. This chapter presents briefly signal processing fundamentals with regard to sound and voice signals.

3.1 Signals

Signals that use electrical or electronic circuits and work with signal values on a continuous scale are called *analogue signals* and methods that process them are called *analogue signal processing*. Such signals are, in most cases, considered to be continuously observable in time and thus are called *continuous-time signals*. If such a continuous-time signal is sampled properly at specific moments in time, a *discrete-time signal* is obtained. When these samples are further converted to discrete numbers, the result is called a *digital signal*. Methods and techniques to cope with such number sequences are called *digital signal processing*, or *DSP* for short.

3.1.1 Sounds as Signals

A signal, such as a wave or vibration variable as stated above, is a function of time that can be represented or approximated in different ways. A sound signal can be any of the following:

- *Mathematical function*, for example a sinusoidal signal, a *pure tone*

$$y(t) = A\sin(2\pi ft) = A\sin(\omega t), \tag{3.1}$$

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

where A is the amplitude or maximum deviation from zero, f is the frequency or the number of vibration cycles in a second, ω is the angular frequency, and t is time. Another example is a *noise* signal

$$n(t) = \text{rand}(t), \tag{3.2}$$

where rand($\cdot$) is a function that yields a randomized value for each time moment.

- *Discrete-time numeric sequence*, for example

$$x(n) = \begin{bmatrix}0.1 & 2.2 & 3.5 & 4.0 & 3.1 & -0.9 & 2.1 & 0.5 & -1.1 & -2.1 & -0.8 & 0.2\end{bmatrix}, \tag{3.3}$$

where n is a discrete-time index. In matrix notation, the sequence in Equation (3.3) is typically given as a column vector instead of a row vector.

- *Graphical presentation*, for example Figure 3.1, where the mathematical signals of Equations (3.1) and (3.2), the numerical sample sequence of Equation (3.3), a short interval

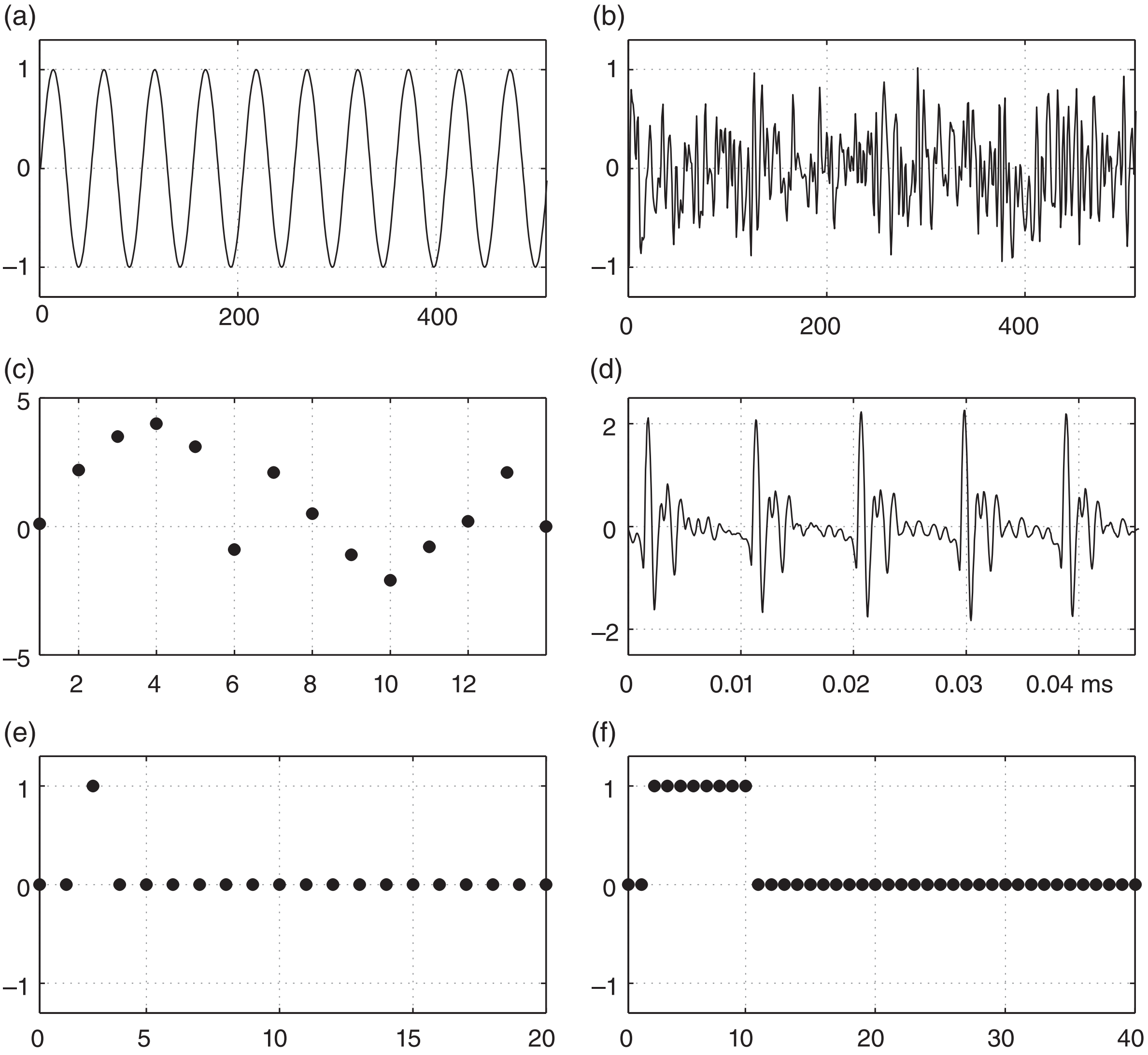

Figure 3.1 Graphical presentations of signals: (a) a sinusoidal or a pure tone signal, (b) random noise, (c) a discrete-time sample sequence, (d) a portion of recorded speech signal, (e) the unit impulse at time moment $n = 3$, and (f) a discrete-time pulse of finite duration.

of a recorded vowel speech signal, a unit impulse $\delta(n - n_0)$, $n_0 = 2$, and a pulse of finite duration are presented. In the figures, the horizontal axis is either time or the sample index and the vertical axis is the value of the signal variable.

The first two mathematically expressed signals are continuous in time while the sample sequence, the unit impulse, and the pulse are discrete-time signals. All signals in Figure 3.1 are, in fact, discrete-time sample sequences in the computer memory, although some of them are plotted as continuous curves.

3.1.2 Typical Signals

- *Pure tone*: as in Equation (3.1)
- *Amplitude-modulated tone*:

$$p(t) = A\,[1 + m\ \sin(\omega_m t)]\ \sin(\omega_0 t), \tag{3.4}$$

  where $m \in [0...1]$ is the modulation index, ω_m is the frequency of modulation, and ω_0 is the angular frequency of the tone.
- *Frequency-modulated tone*:

$$p(t) = A\ \sin[\omega_0 t + k\ \sin(\omega_m t)t], \tag{3.5}$$

  where k is the width of modulation.
- *Tone burst*: a tone which has been set to zero outside the time span $[t, t + \Delta t]$.
- *Sine wave sweep*: where a sine wave is generated with instantaneous frequency that glides linearly or logarithmically from an initial value to a final value, for example over the entire audible range.
- *Chirp signal*: a signal where a fast frequency sweep is carried out.
- *Unit impulse* or the Dirac delta function $\delta(t)$ or $\delta(n)$: a signal that has the value zero everywhere else except at the temporal position zero, where it has the value one, i.e., $\delta(n) = 1$ when $n = 0$. It holds that

$$\int_{-\infty}^{\infty} \delta(t)\mathrm{d}t = 1. \tag{3.6}$$

- *Pulses*: examples are *Gaussian waveforms* or wavelets, and pulse trains made of them.
- *White noise*: a signal where the average spectrum is flat.
- *Pink noise*: a signal that has a spectrum with a 3-dB/octave decay towards high frequencies.
- *Uniform masking noise*: a signal that has a similar spectrum to pink noise but flattens at frequencies below 500 Hz.
- *Modulated noise* (amplitude and frequency modulation) and *noise bursts*.
- *Harmonic tone complexes*:

$$p(t) = \sum_n A_n \sin(n 2\pi f_0 t + \phi_n), \tag{3.7}$$

  where A_n is the amplitude of each harmonic, f_0 is the *fundamental frequency* of the tone complex, and ϕ_n is the starting phase of each harmonic. All partials of the complex are thus integer multiples of the fundamental frequency, or have a common denominator.

- *Complex combination sounds*:

$$p(t) = \sum_i A_i \sin(2\pi f_i t + \phi_i), \tag{3.8}$$

where f_i are the arbitrarily chosen frequencies, and ϕ_i the starting phases of the partials.
- *Sawtooth wave*: a signal that, in the time domain, has linear rises and subsequent steep drops, thus having a shape reminiscent of the teeth of a saw. It can be mathematically written as

$$p(t) = t \mod T, \tag{3.9}$$

where $T = 1/f$, $\mod$ is the modulo operator, and f is the frequency of repetition of the sawtooth wave.
- *Triangle wave*: a signal that, in the time domain, has alternating linear rises and falls. It can be expressed mathematically as

$$p(t) = T/2 - |(t \mod T) - T/2|, \tag{3.10}$$

where $T = 1/f$ and f is the frequency of repetition of the triangle wave.
- *Square wave*: a signal that, in the time domain, has alternating steep rises and falls that are evenly spaced. Mathematically it can be expressed as

$$p(t) = \mathrm{sign}[(t \mod T) - T/2], \tag{3.11}$$

where $T = 1/f, f$ is the frequency of the square wave, and the sign function returns -1 for a negative argument and $+1$ for a positive argument. Sawtooth, triangle, and square waves have a harmonic spectrum, and they can be expressed using Equation (3.7).

3.2 Fundamental Concepts of Signal Processing

Signal processing includes a set of methods that are important for understanding communication by sound and voice. Among these are, for example, linear time-invariant (LTI) systems and processes and the Fourier transform and related signal analysis and synthesis, including spectrum analysis. A special topic in digital signal processing is digital filtering, an efficient method to implement LTI systems. The short presentation of these methods below is intended to refresh the memory of those who have already studied these topics and as a brief overview for those who have not. The mathematics here are kept simple, and the formulas may be skipped entirely by concentrating on the text and the graphical examples if the reader is unfamiliar with such mathematics.

In addition to the basic signal processing, this section includes an overview of some adaptive and learning computational methods (evolutionary computation), such as hidden Markov models.

3.2.1 Linear and Time-Invariant Systems

In signal processing, we are typically interested in the input–output relationship of the system under study (see the black-box formulation in Section 1 of the Introduction). If the output signal $y(t)$ of a system as a function of time depends only on the input signal $x(t)$, their relationship can be expressed generally as $y(t) = h\{x(t)\}$. A system is *linear* and *time invariant* (LTI) if the following is true:

$$h\{\mathrm{a}\,x_1(t) + \mathrm{b}\,x_2(t)\} = \mathrm{a}\,h\{x_1(t)\} + \mathrm{b}\,h\{x_2(t)\}, \tag{3.12}$$

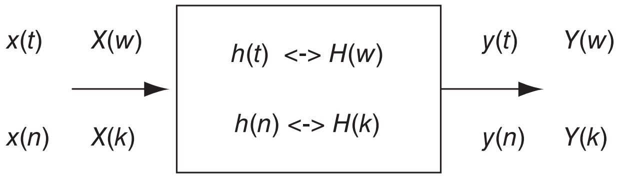

Figure 3.2 The linear and time-invariant (LTI) system as a black box with the related mathematical concepts in the time domain (lower case symbols) and in the frequency domain (upper case symbols).

where a and b are two constants and $x_1(t)$ and $x_2(t)$ are two input signals. In words, the response of an LTI system to the sum of two input signals is equal to the sum of the responses to the individual inputs separately, and the corresponding input and output signal values can be scaled linearly by a constant gain factor. Furthermore, if we apply an input to the system now or T seconds from now, the output will be identical except for a time delay of T seconds. The analysis and implementation of LTI systems is typically easier and more efficient than for systems that do not have or approximate this property. An LTI system also guarantees that it does not create any new frequency components that do not exist in the input signal, i.e., it does not generate non-linear distortion. Thus, a pure tone (a sinusoidal signal) remains a pure tone when propagating through an LTI system.

An LTI system can be represented in the *time domain* with its *impulse response* $h(t)$ or $h(n)$ and in the *frequency domain* with its *transfer function* $H(\omega)$ or $H(k)$. While the *time domain* is a more intuitive way of representing signals, the *frequency domain* has specific useful properties that are discussed throughout the book.

The system may be represented graphically as a black box, as shown in Figure 3.2. Variable t in these formulations refers to time as a continuous-valued variable and n is used as a discrete-time index variable. In the transfer functions, ω is the angular frequency used in continuous-time signals and k is a similar discrete frequency index variable corresponding to the discrete-time representations.

Examples of systems where linearity is of importance are audio recording and reproduction equipment. Amplifiers can be designed to have low non-linear distortion, but, for example, loudspeakers at relatively high power levels and at low frequencies can be highly non-linear. Linearity is a goal in signal processing, although in some tasks, such as perceptually motivated processing, non-linear processing methods are necessary.

3.2.2 Convolution

The relation between the input and output signals of an LTI system can be expressed mathematically using the convolution operation denoted by $*$ and defined as

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau)\, h(t-\tau)\, \mathrm{d}\tau \qquad \text{or} \tag{3.13a}$$

$$y(n) = x(n) * h(n) = \sum_{i=-\infty}^{+\infty} x(i)\, h(n-i). \tag{3.13b}$$

The first expression is for continuous-time signals and systems, and it is called the *convolution integral*, while the second, the *convolution sum*, is for discrete-time signals and systems. For practical signals, the limits of t and n in the integral and the sum are finite.

Although formally simple, convolution is a surprisingly complicated operation to comprehend. It may also be a computationally demanding operation if the response sequences are long. Using the LTI concept, if an arbitrary input signal is considered to be a sequence of impulses with different amplitudes, the output is the combined response of the responses to all of these input impulses.

The impulse responses $h(t)$ or $h(n)$ for a real-time system are always *causal*; that is, $h(\cdot) = 0$ when t or $n < 0$. This means that a physically realizable system does not have information on the future values of its input signal.

If a system does not meet the LTI condition of Equation (3.12), it is *time-variant* if its response properties change as a function of time, or it is *non-linear* if its transfer properties depend on the signal passing through it. In both cases the system can generate new frequency components that do not exist in the input signal. The analysis, modelling, or synthesis of such systems is typically substantially more difficult than for an LTI system. A slightly non-linear system can be approximated using a linearized model if the error is tolerable.

A good example of a highly non-linear and time-variant system is human hearing. As a result, modelling of the auditory system is a complex task, and only some of its peripheral parts may be modelled in the LTI sense.

3.2.3 Signal Transforms

A useful mathematical approach in signal processing is based on transforming signals into another form that makes processing or interpreting them easier. A particularly useful set of transforms yields a mapping between the time- and frequency-domain representations, the Fourier transform described below being the most important one.

Mathematical tools that are needed for frequency-domain representations are *complex numbers* and *complex-valued functions*. A complex number c is composed of a real part x and an imaginary part y written as

$$c = \mathrm{Re}\,\{c\} + \mathrm{j}\,\mathrm{Im}\,\{c\} = x + \mathrm{j}\,y, \tag{3.14}$$

where Re $\{\cdot\}$ means the real part of, Im $\{\cdot\}$ the imaginary part of, and j, often also denoted by i, is the imaginary unit $\mathrm{j} = \sqrt{-1}$. A fundamental equation for operating with complex numbers is the *Euler relation* for complex exponentials

$$e^{\mathrm{j}\phi} = \cos\phi + \mathrm{j}\sin\phi \tag{3.15}$$

that ties the phase angle ϕ to the real and imaginary components:

$$c = x + \mathrm{j}y = |c|e^{\mathrm{j}\phi} \tag{3.16a}$$

$$|c| = \sqrt{x^2 + y^2} \tag{3.16b}$$

$$\angle\, c = \arg\{c\} = \phi = \arctan(x/y), \tag{3.16c}$$

where $|\cdot|$ means the absolute value of or the magnitude of and $\angle$ and $\arg\{\cdot\}$ mean the phase or argument of.

3.2.4 Fourier Analysis and Synthesis

The analysis of an LTI system is mathematically simplified if it is represented in the frequency domain, and the signals are described as functions of frequency instead of time. This can be done using the *Fourier transform*

$$X(\omega) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{+\infty} x(t)\, e^{-\mathrm{j}\omega t}\, \mathrm{d}t \tag{3.17a}$$

$$X(k) = \mathcal{F}_{\mathrm{d}}\{x(n)\} = \sum_{n=0}^{N-1} x(n)\, e^{-\mathrm{j}k(2\pi/N)n}. \tag{3.17b}$$

The transform in Equation (3.17a) is valid for continuous-time signals and systems while Equation (3.17b) is for discrete-time cases. The reader is referred to a standard textbook, such as Proakis (2007), for a more thorough discussion on the mathematical details and applications of the Fourier transform.

The continuous-time and discrete-time transform operators are denoted here by $\mathcal{F}\{\cdot\}$ and $\mathcal{F}_{\mathrm{d}}\{\cdot\}$. The latter form, Equation (3.17b), is called the *discrete Fourier transform* (DFT). It is defined for a finite length sequence ($n = 0 \ldots N - 1$), and it can be computed very efficiently using the *fast Fourier transform* (FFT) (Proakis, 2007). The FFT is applicable for periodic signals, one period being the index range of the summation similar to Equation (3.17b).

An interpretation of what the Fourier transform does is that it is a correlation (a kind of similarity comparison, defined by Equation (3.26a)) of the signal $x(\cdot)$ to be transformed with a pair of sinusoids, a sine and a cosine, together expressed as a complex exponential $e^{-\mathrm{j}\omega t}$ or $e^{-\mathrm{j}k(2\pi/N)n}$. When applying this rule to each frequency (ω or n), the result represents the frequency content of $x(\cdot)$. See Figure 3.3 for an example.

The inverse transforms for Equations (3.17a) and (3.17b) are

$$x(t) = \mathcal{F}^{-1}\{X(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega)\, e^{\mathrm{j}\omega t}\, \mathrm{d}\omega \tag{3.18a}$$

$$x(n) = \mathcal{F}_{\mathrm{d}}^{-1}\{X(k)\} = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{\mathrm{j}k(2\pi/N)n}, \tag{3.18b}$$

which map the signal from the frequency domain back to the time domain. The transforms in Equations (3.17a) and (3.17b) may be interpreted as *Fourier analysis* and the transforms in Equations (3.18a) and (3.18b) as *Fourier synthesis*. Figure 3.3 illustrates how a sawtooth waveform is constructed as a linear combination of its sinusoidal components. Any signal that meets certain continuity requirements can be represented with arbitrary precision as a sum of sinusoidal components of different frequencies. These components are often called *partials*.

An important advantage of using the Fourier transform is that it converts the computationally expensive convolution in the time domain into a much simpler multiplication in the frequency domain.

$$\mathcal{F}\{x(t) * y(t)\} = X(\omega) \cdot Y(\omega) \tag{3.19a}$$

$$\mathcal{F}_{\mathrm{d}}\{x(n) * y(n)\} = X(k) \cdot Y(k), \tag{3.19b}$$

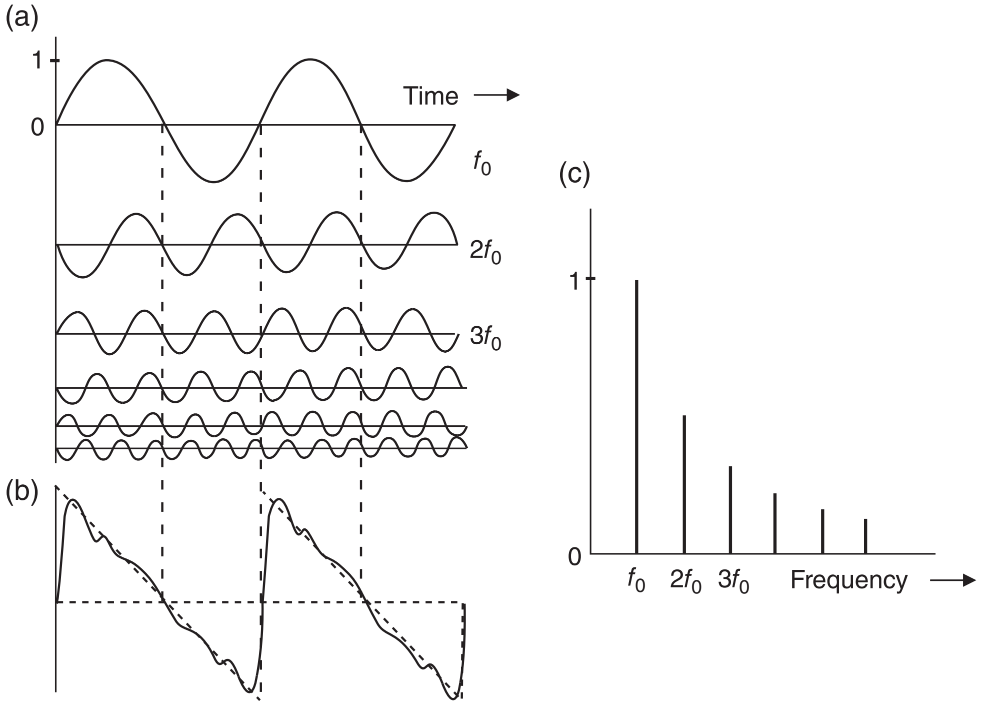

Figure 3.3 The decomposition of a sawtooth waveform into its sinusoidal frequency components. Putting together these components is called Fourier synthesis, and breaking up a waveform into its components is called Fourier analysis.

where the lower case symbols x and y denote time signals and the upper case symbols X and Y their frequency-domain transforms. Note that the definition of the Fourier transform involves the assumption of infinite periodicity of the signals $x(n)$ and $y(n)$. However, by appending the signals with sequences of zeros, the equivalent of a linear time-domain convolution is nevertheless obtained. The process of appending zeros to the start or end of the signal is often called *zero padding*. Having accounted this for, the FFT and the inverse fast Fourier transform (IFFT) can be utilized in the efficient computation of convolution in the following way:

$$x(t) * y(t) = \mathcal{F}^{-1}\{X(\omega) \cdot Y(\omega)\} = \mathcal{F}^{-1}\{\mathcal{F}\{x(t)\} \cdot \mathcal{F}\{y(t)\}\} \tag{3.20a}$$

$$x(n) * y(n) = \mathcal{F}_{\mathrm{d}}^{-1}\{X(k) \cdot Y(k)\} = \mathcal{F}_{\mathrm{d}}^{-1}\{\mathcal{F}_{\mathrm{d}}\{x(n)\} \cdot \mathcal{F}_{\mathrm{d}}\{y(n)\}\}. \tag{3.20b}$$

3.2.5 Spectrum Analysis

Representing signals in the frequency domain, as decompositions into their frequency components, is important not only as a powerful signal processing technique but also because it resembles the way the human ear analyses signals. Audio signals that we are able to hear as sounds are thus often analysed by means of frequency transforms in order to find their audible features and cues. The result of the Fourier transform (Equations (3.17a) and (3.17b)) is complex-valued. Such a complex transform can be equivalently expressed by a pair of separately plotted *spectra*, the *magnitude spectrum* on the logarithmic decibel scale

$$|X(\omega)|_{\mathrm{dB}} = 20\log_{10}|X(\omega)| \tag{3.21a}$$

$$|X(k)|_{\mathrm{dB}} = 20\log_{10}|X(k)|, \tag{3.21b}$$

and the *phase spectrum*

$$\varphi(\omega) = \angle X(\omega) = \arg\{X(\omega)\} \tag{3.22a}$$

$$\varphi(k) = \angle X(k) \ = \arg\{X(k)\}. \tag{3.22b}$$

In principle, $\varphi(k)$ can be unequivocally solved from $\text{Im}\{X(k)\}$ and $\text{Re}\{X(k)\}$ using Equation (3.16c). Unfortunately, the expression for the phase spectrum cannot be expressed explicitly. In many computer languages, such as in Matlab, φ(k) = angle(X(k)) = atan2(Im {X(k)} , Re {X(k)}).

For sound signals, the concept of *spectrum analysis* typically refers to a representation of the magnitude spectrum only, because, as is well known, the auditory system is relatively insensitive to the phase of a signal. The motivation to use the logarithmic dB scale for the magnitude spectrum comes from the fact that we perceive the level of a signal more logarithmically than linearly. The dB scale is also appropriate for the graphical representation of spectra. However, as will be shown in later chapters, the detailed behaviour of the auditory system deviates from the simple Fourier spectrum analysis in many ways.

The phase spectrum $\varphi(\cdot)$ from Equations (3.22a) and (3.22b) is cyclically limited between $-\pi$ and π, exhibiting discontinuous jumps between these boundaries. If a continuous phase spectrum is desired, a *phase unwrapping* operation is necessary.

Often a more useful representation than the phase function itself involves the *group delay* τ_{g} and the *phase delay* τ_{p}

$$\tau_{\mathrm{p}}(\omega) = -\varphi(\omega)/\omega, \tag{3.23a}$$

$$\tau_{\mathrm{g}}(\omega) = -\mathrm{d}\varphi(\omega)/\mathrm{d}\omega. \tag{3.23b}$$

The phase delay represents the delay of a frequency component when propagating through a system. Group delay, as the frequency derivative of the phase, describes the delay of the modulation, such as the amplitude envelope, of a frequency component. From the point of view of the auditory system, the group delay is the most relevant of these phase representations.

In practice, spectrum analysis must be localized in time, since the spectral properties of sound signals typically vary over time. *Windowing* is used to accomplish this by multiplying the signal with a *window function* which is then Fourier analysed:

$$X(\omega) = \int_{t_{\mathrm{b}}}^{t_{\mathrm{e}}} w(t)\, x(t)\, e^{-\mathrm{j}\omega t}\, \mathrm{d}t \tag{3.24a}$$

$$X(k) = \sum_{n=n_{\mathrm{b}}}^{n_e} w(n)\, x(n)\, e^{-\mathrm{j}k(2\pi/N)n}, \tag{3.24b}$$

where $w(t)$ is a window function (weight function) that is non-zero only in the time span denoted by the limits of integration and summation in Equations (3.24a) and (3.24b) and zero elsewhere.

Some frequently applied window functions are the Hamming, Hann (a.k.a. Hanning window), Blackman, and Kaiser windows (Mitra and Kaiser, 1993). Note that cropping a span from a signal and zeroing elsewhere corresponds to using a rectangular window. Figure 3.4 illustrates an example where a sinusoidal signal is analysed with different windows, including rectangular and Hamming windows, with and without synchrony to the periodicity of the sine wave.

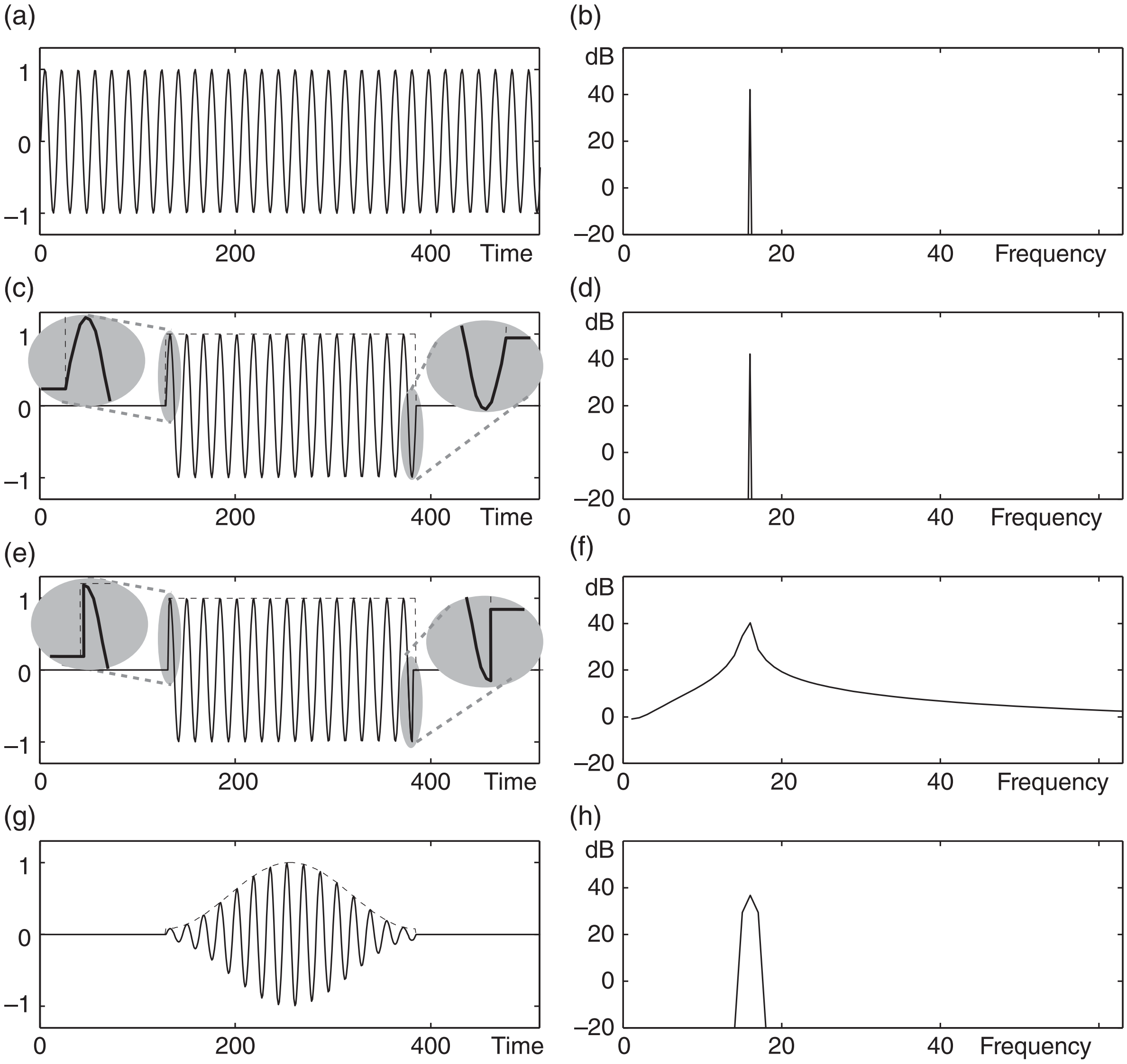

Figure 3.4 Spectrum analysis using the Fourier transform and windowing. (a) A sine wave and (b) its magnitude spectrum with a very long window. (c) A rectangular window that is synchronized with the periodicity of the sine wave. The ending of the sinusoid in the window and the starting of it join together circularly continuously in amplitude and slope. (d) The corresponding spectrum with no artefacts. (e) A rectangular window that is not in periodicity synchrony with the signal, showing as a discontinuity in amplitude between the starting and ending positions. (f) The resulting spectrum that shows the spreading of the spectrum. The Hamming window (dashed envelope line) in (g) always removes most of the spectral spreading far from the peak, but the spectrum peak itself will be broadened, as shown in (h).

The selection of a window function is always a compromise between spectral and temporal resolution. The longer the window, the better the spectral resolution and the worse the temporal resolution, and vice versa. Theoretically, the resolutions in time (Δt) and frequency (Δf) are bound by the equation

$$\Delta t \cdot \Delta f \geq 0.5. \tag{3.25}$$

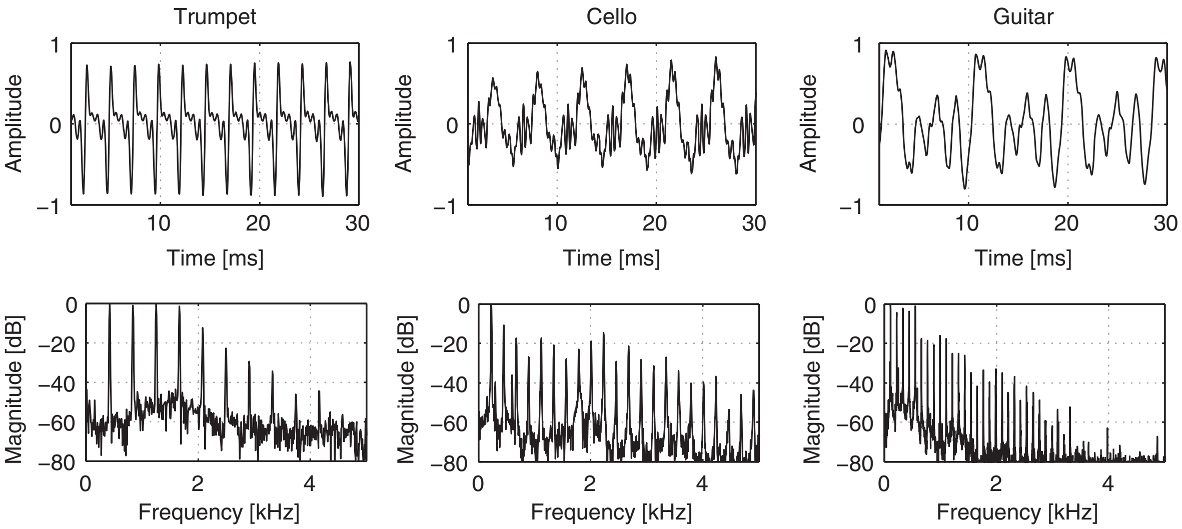

Figure 3.5 The waveforms and spectra of three instrument sounds.

For audio signals, the length of the window is often motivated by the time-vs.-frequency resolution of the auditory system. If only a single window length is used, the value is typically selected to be between 10 and 30 milliseconds, which, using Equation (3.25), limits the frequency selectivity to between 50 and 17 Hz. Noting the non-linear scale of auditory perception, the selection of the window length is especially relevant for analysis in the lowest frequencies.

Examples of real audio signals of musical instruments and their magnitude spectra are shown in Figure 3.5. See Figure 5.7 on page 87 for spectra of vowels and consonants.

3.2.6 Time–Frequency Representations

A time–frequency representation is obtained, for example, by applying a window, such as a 20-ms Hamming window, to a portion of a signal, taking the Fourier transform, and then moving to the next portion of the signal and repeating the procedure. This spectral sampling interval, called the *hop size*, is typically about 10 ms for analysing speech and other audio signals. Such a frame-based analysis of spectra is called *short-time Fourier analysis*, and its graphical representation is called a *spectrogram*.

Since a spectrogram is a three-dimensional mapping – the magnitude level as a function of time and frequency – it cannot be illustrated by a single curve. One typical graphical representation is as an intensity map where the grey shade or colour at each point stands for the magnitude (see Figure 3.6). Another representation is the ‘waterfall’ or mesh plot with a set of curves, as shown in Figure 6.5 on page 105.

Short-time Fourier analysis is a special case of a *time–frequency representation*. Other choices are *wavelet analysis* and *Wigner distributions*. In wavelet analysis (Cohen, 1995; Vetterli and Kovacevic, 1995), the frequency and time resolution are not uniform but vary so that at high frequencies the time resolution, is better with a coarser frequency resolution, and vice versa at low frequencies. A method related to wavelet processing is the constant-Q transform (Holighaus *et al.*, 2013; Schörkhuber *et al.*, 2013), where the signal is divided into time-frequency tiles of equal area, but the bandwidth increases with frequency to maintain a constant Q (Equation (3.30)). The temporal length of the tiles decreases with increasing frequency, which makes the processing a bit more complicated. However, perfect reconstruction processing has been achieved with constant-Q methods.

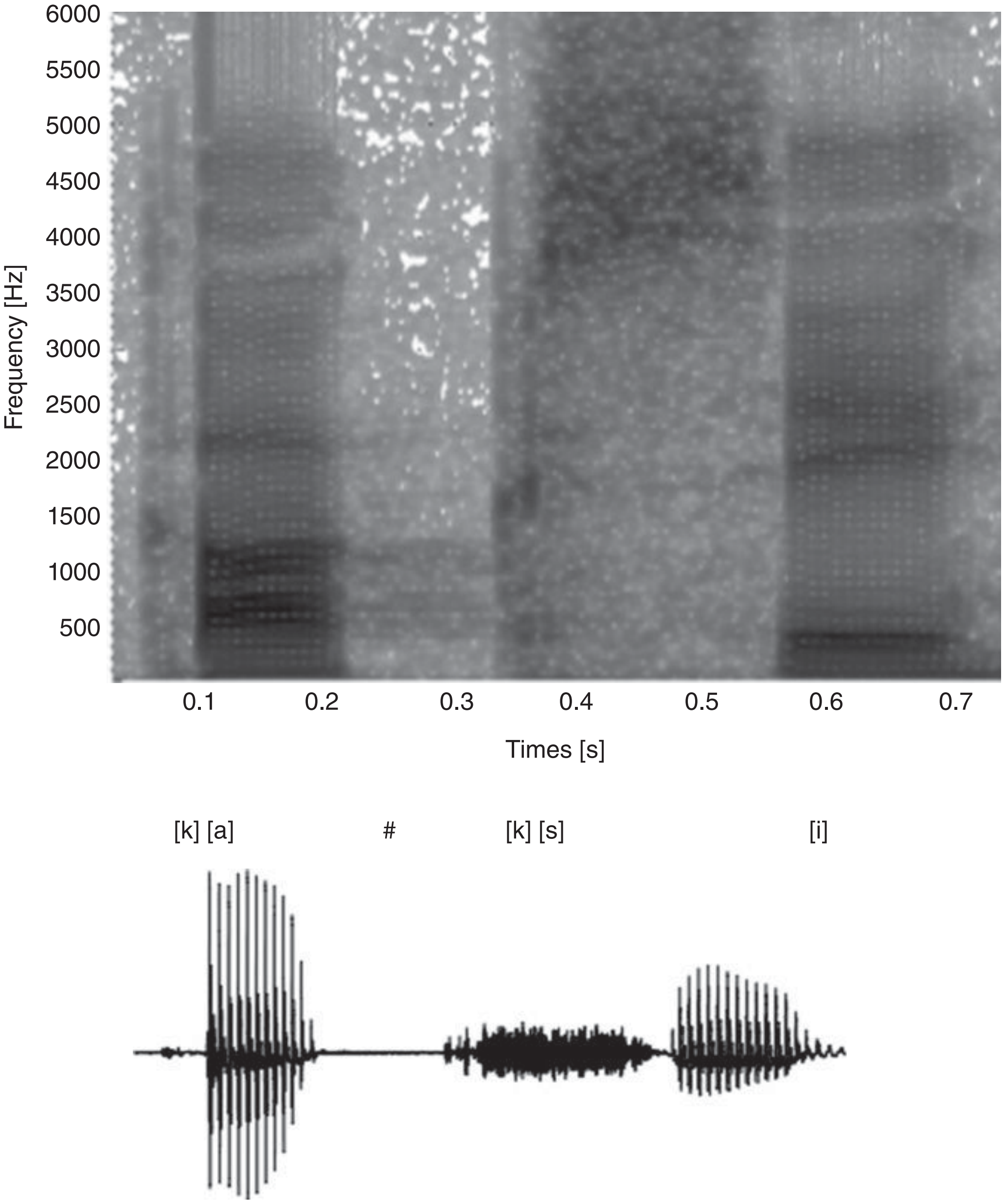

Figure 3.6 The spectrogram of a spoken word (top) aligned in time with the original waveform (bottom). Higher magnitude levels are shown with a darker shade of gray in the spectrogram. The voiced parts the formant resonances are darker horizontal stripes, while in the noise-like fricative part only relatively high frequencies have noticeable energy. Matti Karjalainen utters the word 'kaksi', meaning 'two' in Finnish, which he often used to test audio and speech techniques. The uttered phones are shown, where '#' denotes silence.

3.2.7 Filter Banks

A *filter bank* is a set of band-pass filters that is fed the same input, where the centre frequencies of the filters vary over desired ranges of frequencies. It thus separates a broadband input signal into multiple time-domain narrowband signals, which are called sub-bands. This is similar to the functioning of hearing, as will be discussed later in Chapter 7.

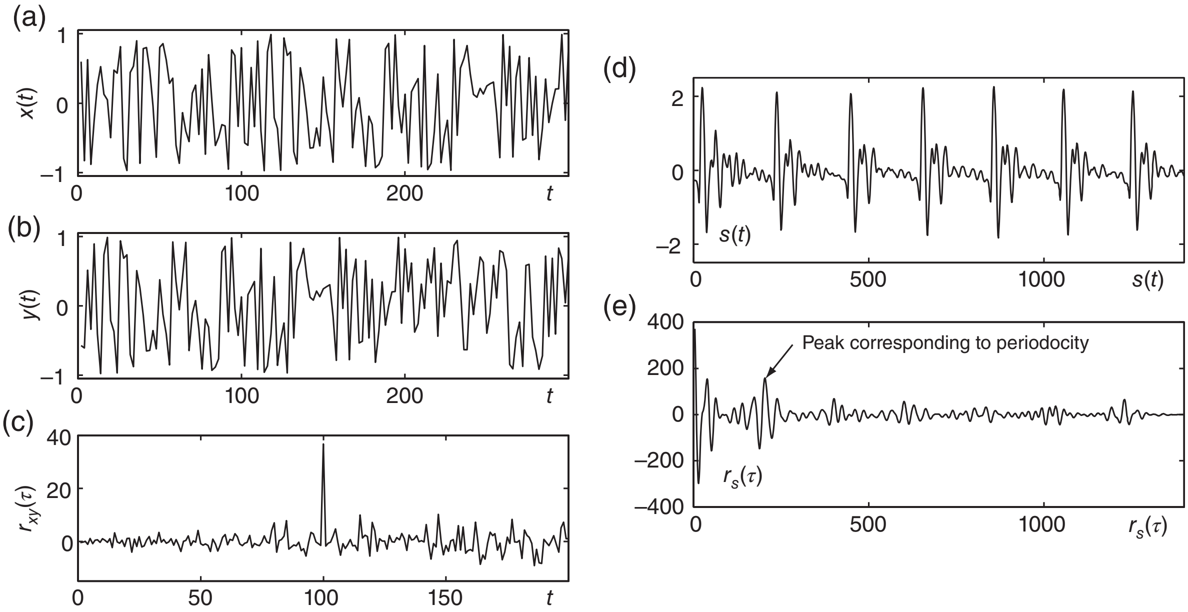

Figure 3.7 Examples of cross- and autocorrelations: (a) A random signal $x(t)$, (b) the same delayed signal $y(t) = x(t - 100)$, and (c) the cross-correlation $r_{xy}(\tau)$ showing a peak that indicates the time delay. (d) A voiced speech signal $s(t)$, and (e) its autocorrelation $r_{\mathrm{S}}(\tau)$ indicating the periodicity (the inverse of the fundamental frequency).

Filter banks have a number of uses. For example, when used as an equalizer, the signals in sub-bands are amplified according to the desired equalization curve, and they are then combined to form the equalized signal. Since sometimes *perfect reconstruction* is desired, the filters must be designed so that the original signal can be recovered when the sub-band signals are combined. Filter banks have various applications in audio effects, audio coding, and spatial audio reproduction, as will be discussed later in Chapter 15.

3.2.8 Auto- and Cross-Correlation

The similarity of two waveforms can be analysed by computing their *cross-correlation*

$$r_{xy}(\tau) = \int_{-\infty}^{+\infty} x(t)\, y(t+\tau)\, \mathrm{d}\tau \tag{3.26a}$$

$$r_{xy}(k) = \sum_{i=0}^{N-1} x(i)\, y(i+k). \tag{3.26b}$$

Figures 3.7a–c depict a case where the cross-correlation of a signal and its time-delayed counterpart indicate the amount of delay. A special case of correlation is *autocorrelation*, where a signal is compared to itself in order to find the periodicity, or repeatability, of the waveform. In this case $x(\cdot) = y(\cdot)$ in Equations (3.26a) and (3.26b). In the example of Figures 3.7d–e, the autocorrelation of a voiced speech signal has a peak corresponding to its periodicity. The autocorrelation function is periodic, showing maxima also for integer multiples of the fundamental period.

3.2.9 Cepstrum

The *cepstrum* (Oppenheim and Schafer, 1975) is a transform that shows some resemblance to autocorrelation. It is computed as the inverse Fourier transform of the logarithmic magnitude spectrum,

$$c_x(t) = \mathcal{F}^{-1}\{\log |\mathcal{F}\{x(t)\}|\}. \tag{3.27}$$

This resemblance can be understood from the fact that the logarithmic magnitude spectrum is a real-valued function and that the inverse Fourier transform is an operation very similar to the Fourier transform, as is evident from Equations (3.17a) and (3.18a). The computation of the cepstrum thus treats the magnitude spectrum similarly to a time-domain signal. The result can be thought to be 'the spectrum of a magnitude spectrum curve'. The logarithm of the magnitude spectrum provides a particular representation for differentiating specific processes affecting a speech spectrum: the harmonic response of the glottis is represented by a fast-changing, periodic log-spectrum, while the formants produced by the vocal tract are represented by a longer and smoother envelope and convey the information of which phoneme is uttered. The inverse-Fourier-transform operator projects these two envelopes to different points in the cepstrum $c_x(t)$, providing a representation that is practical for processes such as speech recognition.

3.3 Digital Signal Processing (DSP)

Digital signal processing (DSP) means discrete-time numerical processing of signals (Mitra and Kaiser, 1993; Oppenheim *et al.*, 1983; Strawn, 1985). If a signal to be processed is originally in analogue form, i.e., continuous in time and amplitude, it must first be converted to a number sequence by *analogue-to-digital conversion* (A/D-conversion). Conversely, a digital signal (a number sequence) can be converted back to continuous-time form by *digital-to-analogue conversion* (D/A-conversion). Signal processing itself is carried out by a *digital signal processor*, which can be a special digital circuit, a programmable digital signal processor, or a general purpose processor or computer. A generic structure for such a system is shown in Figure 3.8.

3.3.1 Sampling and Signal Conversion

When converting between analogue and digital signals, the *sampling theorem* (or the *Nyquist theorem*) requires that the sampling rate must be at least twice as high as the highest signal component to be converted. Otherwise *aliasing* will occur, whereby signal components of frequency higher than the *Nyquist frequency* (half of the sampling rate) will be mirrored to below the Nyquist frequency, thus distorting the signal. To avoid aliasing, an A/D-converter normally includes a low-pass filter that yields enough attenuation above the Nyquist frequency.

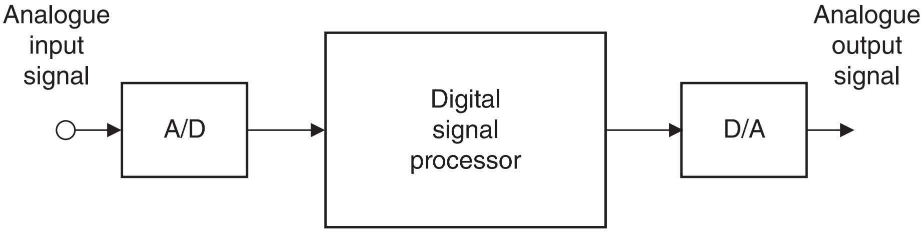

Figure 3.8 A block diagram for the digital signal processing of analogue signals, including an A/D-converter, a digital signal processor, and a D/A-converter.

D/A-conversion also typically includes a low-pass filter (reconstruction filter) to make the output continuous in time and free from frequencies above the Nyquist frequency. This means that digital signal processing deals with band-limited signals.

Sampling rates common in audio technology are 44.1 kHz (compact disc), 48 kHz (professional audio), 32 kHz (less demanding audio), and for very demanding audio 96 kHz or even 192 kHz. Speech technology uses the sampling rate 8–16 kHz. The telephone bandwidth of 300–3400 Hz requires a sampling rate of about 8 kHz.

Numerical samples from A/D-conversion may be coded in various ways. The most straightforward representation is to use *PCM coding* (pulse-code modulation). Each sample is *quantized* into a binary number where the number of bits implies the precision of the result. Figure 3.9 illustrates the principle of such a conversion using four bits, which corresponds to 16 levels. Sample values of an analogue signal are mapped onto binary numbers so that there are 2^n discrete levels when the number of bits is n.

Quantization with finite precision generates an error called *quantization noise*. The *signal-to-noise ratio* (SNR, see Section 4.2.6) describes the level of the signal compared to the level of noise, and for quantization noise it improves by 6 dB for each added bit, so that 16 bits, often used in audio, yield a maximum SNR of about 96 dB. Because the dynamic range of the auditory system is about 130 dB, even more than 22 bits may be needed.

Digital signal processing has many advantages compared to analogue techniques. It is predictable, and a single DSP processor may be programmed to compute any DSP program that does not exceed its processing capacity. For example, real-time spectrum analysis can be implemented using the FFT.

3.3.2 Z Transform

The *z transform* is a fundamental mathematical tool for describing LTI systems in digital signal processing. The z transform of a digital signal (sample sequence) $x(n)$ is

$$X(z) = \mathcal{Z}\{x(n)\} = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n} \tag{3.28}$$

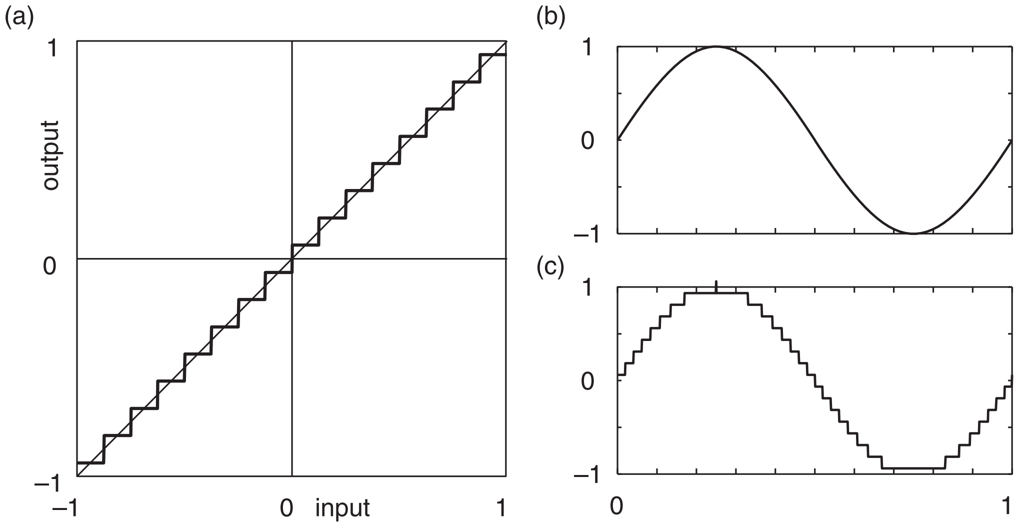

Figure 3.9 Quantization and PCM representation of a sinusoidal signal with 4 bits (16 levels): (a) characteristics of the quantization curve, (b) sinusoidal analogue input, and (c) quantized signal waveform.

The complex variable z used in the transform is related to the *unit delay* between two adjacent samples as

$$\mathcal{Z}\{x(n-1)\} = z^{-1}X(z). \tag{3.29}$$

3.3.3 Filters as LTI Systems

A filter modifies the magnitude or phase spectrum of an input signal. Often, filters are applied to attenuate some frequencies and leave others untouched. In most cases, the processing also changes the phase spectrum. In some cases, the filters do not change the magnitude spectrum, but only the phase spectrum.

There are several types of filters:

- A *low-pass filter* is one that leaves low-frequency signals unmodified and attenuates signals with frequencies higher than the *cutoff frequency*. In the design of such filters, the response is often set to be 0 dB for frequencies in the *passband* (frequencies below the cutoff frequency), −3 dB at the cutoff frequency, and to lower values in the *stopband* (frequencies higher than the cutoff). The response in the stopband depends on the filter type and design.
- A *high-pass filter* is the opposite of a low-pass filter, meaning that the passband is located at frequencies higher than the cutoff and the stopband correspondingly at lower frequencies.
- A *band-pass filter* leaves a band of frequencies unmodified and attenuates all other frequencies. It can be implemented as a combination of a low-pass and a high-pass filter in a cascade. The *bandwidth* Δf is defined as the difference between the upper and lower cutoff frequencies $\Delta f = f_\mathrm{u} - f_\mathrm{l}$. The Q value of a band-pass filter is then defined as

$$Q = f_\mathrm{c}/\Delta f. \tag{3.30}$$

A high Q value thus implies a narrow band-pass filter, and vice versa.
- A *band-reject filter* is the opposite of a band-pass filter: it removes a certain band of frequencies and leaves the rest unmodified. It can thus also be implemented as a combination of low-pass and high-pass filters, but this time in parallel.
- An *all-pass filter* leaves the amplitude of all frequencies unchanged, but changes their phase relationships.
- *Arbitrary-response filters* are designed to have an arbitrary response both in magnitude and in phase.

Most commonly, filters are implemented as digital filters using DSP structures, or as analogue filters using electronic circuits. Many acoustic phenomena can be interpreted as filters. For example, the effect of atmospheric absorption of sound is effectively a low-pass filter, and the acoustic effect of a room can also be considered as a filter with an arbitrary response.

3.3.4 Digital Filtering

Digital filtering (Haykin, 1989; Jackson, 1989; Parks and Burrus, 1987) is a fundamental technique in digital signal processing. The input–output relationship of any band-limited LTI system (Figure 3.2) can be represented and implemented using a digital filter. Since algorithms for computing digital filters can be well optimized, this approach is useful when simulating or solving LTI engineering problems.

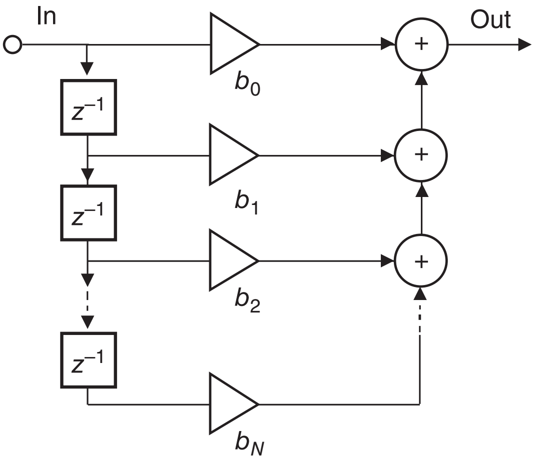

Figure 3.10 FIR filtering as a signal-flow graph. Elements z^{-1} represent unit delays, and multipliers b_n correspond to the coefficient values (tap coefficients) of the impulse response.

The two main types of digital filters are the finite impulse response *(FIR) filter* and the infinite impulse response *(IIR) filter.*

An FIR filter is computed using the principle in Figure 3.10 as a linear combination of delayed versions of the input signal; that is, by a convolution of the input and the impulse response of the filter. Each block z^{-1} is a unit delay, and the tap coefficients b_n before summation are directly the coefficient values of the impulse response $h(n)$. The transfer function of an FIR filter is simply

$$H_{\mathrm{FIR}}(z) = \sum_{n=0}^{N-1} b_n z^{-n} = b_0 + b_1 z^{-1} + \cdots + b_{N-1} z^{-(N-1)} \tag{3.31}$$

FIR filter design and signal processing with them is relatively straightforward, but FIR filters may be computationally expensive.

The transfer function of an IIR filter is

$$H_{\mathrm{IIR}}(z) = \frac{\sum_{n=0}^{N-1} b_n z^{-n}}{1 + \sum_{p=1}^{P-1} a_p z^{-p}} = \frac{b_0 + b_1 z^{-1} + \cdots + b_{N-1} z^{-(N-1)}}{1 + a_1 z^{-1} + \cdots + a_{P-1} z^{-(P-1)}} \tag{3.32}$$

One typical signal-flow graph formulation of an IIR filter, the direct form II, is depicted in Figure 3.11. It is different from an FIR filter in that there are feedback paths through multipliers a_n. An IIR filter may be unstable if it is not designed properly, which means that it can produce arbitrarily large output values for a finite input signal if a frequency is exponentially amplified in the feedback structure. The stability criterion for an IIR filter is that the poles of the transfer function, that is, the roots of the denominator polynomial of Equation (3.32), must be inside the unit circle on the complex plane $|z| < 1$. The zeros (roots of the numerator) don't have such a limitation.

3.3.5 Linear Prediction

A signal, or the system that has generated it, can be modelled in the LTI sense using *linear prediction* (LP) (Markel and Gray, 1976). In the literature the term *linear predictive*

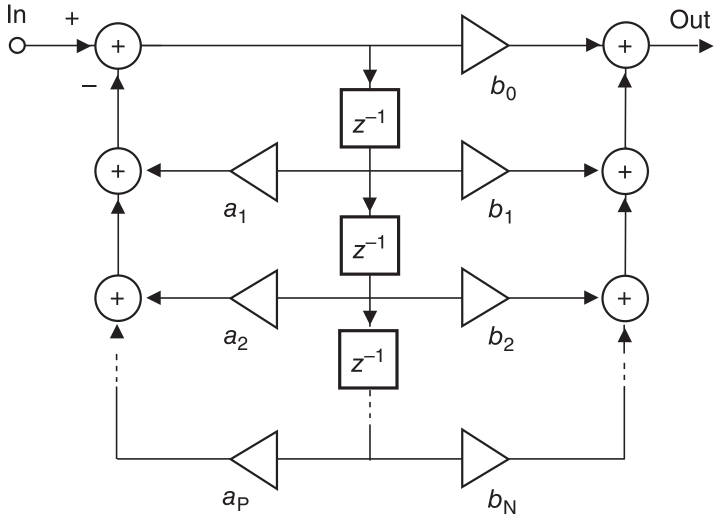

Figure 3.11 IIR filtering as a signal-flow graph. Coefficients b_n correspond to non-recursive FIR-like substructures (Figure 3.10), and a_n are the coefficients for recursive feedback.

coding and especially its abbreviation LPC is used as a general concept, but a more systematic approach is to use the term *linear prediction* for modelling and LPC only for coding purposes where LP analysis is quantized or otherwise coded. LP modelling also has other names, such as *autoregressive (AR) modelling* in estimation theory.

In LP modelling, we can imagine that the signal to be analysed is generated by an IIR system (Figure 3.11), where in the numerator of Equation (3.32) the coefficient $b_0 = 1$ and all the other coefficients $b_n = 0$ for $n \geq 1$. This is called an *all-pole* type IIR system. *LP analysis* yields optimal values in the least mean square sense for the denominator polynomial coefficients a_p, so that the resulting all-pole IIR filter system is the best one for predicting a new signal sample as a linear combination from the previous samples.

The most frequently used form of LP analysis is the *autocorrelation method*, where the coefficients a_p are solved from a linear matrix equation (normal equations)

$$\begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_{P-1} \\ r_1 & r_0 & r_1 & \cdots & r_{P-2} \\ r_2 & r_1 & r_0 & \cdots & r_{P-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{P-1} & r_{P-2} & r_{P-3} & \cdots & r_0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_P \end{bmatrix}, \tag{3.33}$$

where P is the order of LP analysis, that is, the order of the all-pole filter, and r_k are autocorrelation coefficients $r_x(k)$ from

$$r_x(k) = \sum_{i=0}^{N-1-k} x(i)\, x(i+k) \tag{3.34}$$

for the signal frame under study consisting of N samples.

The IIR filter $1/A(z)$ is called the *synthesis filter*, where $A(z) = 1 - \sum a_p z^{-p}$ and a_p are the coefficients from Equation (3.33). The FIR filter $A(z)$ itself is called the *inverse filter*. If it acts on the original signal, a *residual signal* that is spectrally flattened (whitened) is obtained.

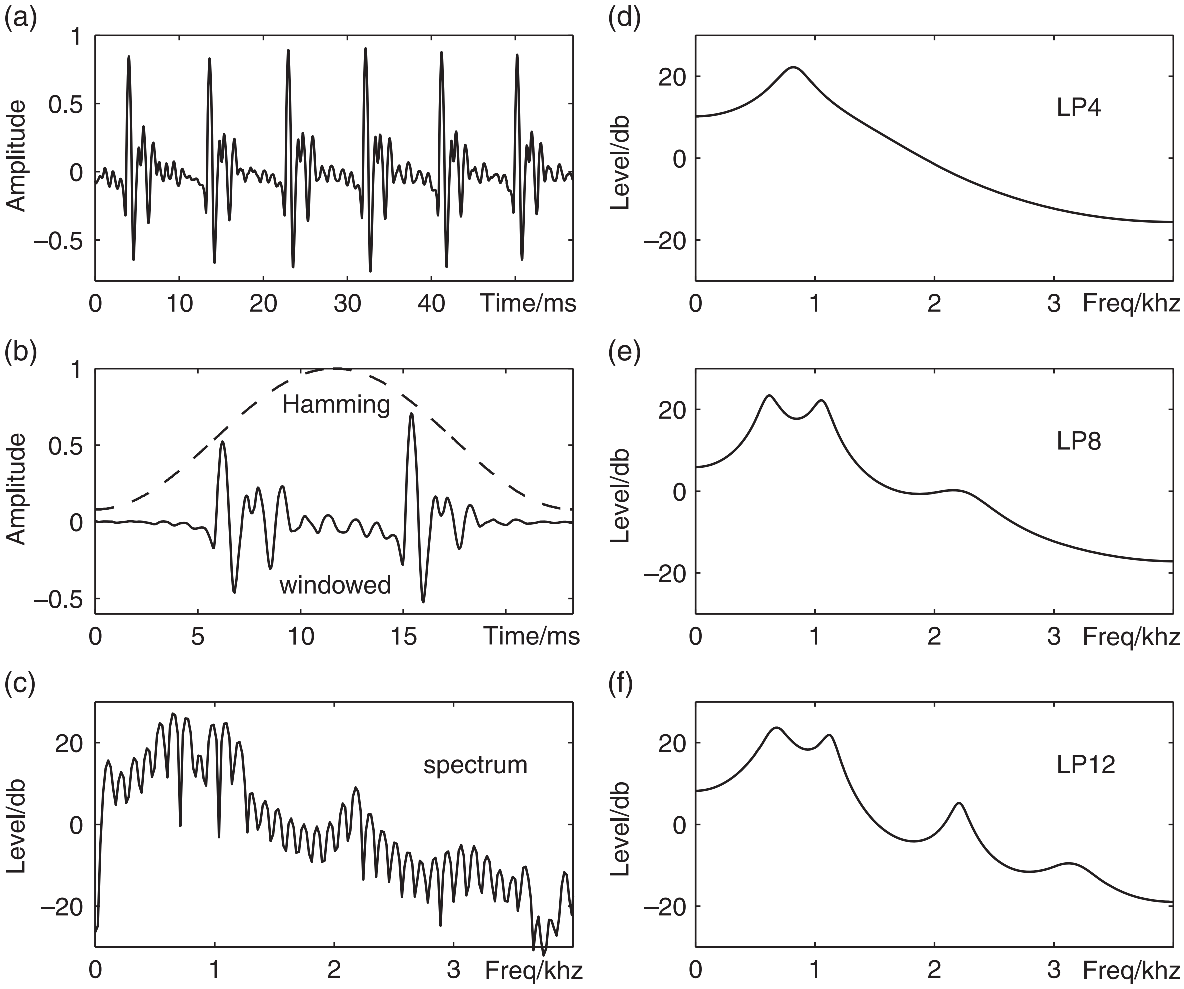

Figure 3.12 (a) A voiced speech signal (vowel) as a function of time. (b) The Hamming window (dashed line) and a Hamming-windowed frame of the signal (solid line), (c) its Fourier spectrum, and linear prediction spectra for LP orders of (d) 4 , (e) 8, and (f) 12. A sampling rate of 8 kHz has been used.

Naturally, if this residual is used as the excitation signal for filter $1/A(z)$, the original signal is synthesized, hence the name synthesis filter. In Section 5.3 we discuss the modelling of speech using source–filter models, whereby linear prediction is a natural choice and an effective technique.

LP analysis makes it possible to easily compute *spectral envelopes* or the *LP spectrum* that is, to remove the spectral fine structure of a speech or audio spectrum. Especially for speech signals, LP analysis is an effective way to separate the source (excitation) and filter (vocal tract transfer properties) in a source–filter model (see Figure 5.11 on page 91). Figures 3.12d–f illustrate LP spectra computed from the speech signal in Figure 3.12a, first windowed to obtain the signal in Figure 3.12b. As a reference, a Fourier spectrum of the speech frame is shown in Figure 3.12c. When the LP order is increased, the LP spectrum resolution improves. The LP order 8 (Figure 3.12e) already yields a fairly good approximation, and orders 10–12 are considered high enough for speech with a sample rate of 8 kHz. More generally, an order equal to the sample rate in kHz plus two is recommended for speech, since it is enough to represent speech formants and the general shape of the spectrum.

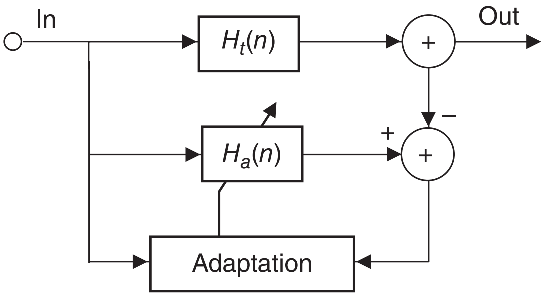

Figure 3.13 An example of adaptive filtering, where filter $H_a(n)$ adapts to the input–output relationship of target system $H_t(n)$, thus modelling the target system.

3.3.6 Adaptive Filtering

Digital filters with constant coefficients are an efficient and flexible approach for modelling LTI systems and signals. However, real world signals and systems can change their parameters as a function of time. There is a need for signal processing systems and algorithms that are able to adapt to changing conditions, or that are able to learn or can be trained to behave in a desired manner. Often there exists a rule that can be applied to change the parameters of a DSP system to adapt to the environment, or there are examples of desired behaviour that can be taught to the system. In such cases it would be preferable if the DSP system could automatically adapt to the external conditions. Sometimes it is enough to do the adaptation or learning only once or every now and then in a controlled manner.

In *adaptive filtering* (Haykin, 1989, 2005; Widrow and Stearns, 1985), the goal is that a signal processing function using digital filters can adapt to properties of the input signal in a meaningful way. Technically, adaptation means that the transfer function of the filter is controlled by changing the filter coefficients. Thus, the filter is no longer time-invariant. This time-variance problem can be formulated in different ways.

- A digital filter can be adapted so that the output signal is as close as possible to a target signal, so that their difference, for example in the least-squares sense, is minimized. After successfully adapting a filter to transform the input of a target system to the output of the target system, the filter can be used as a model of the target system (see Figure 3.13).
- When the input and output of the adapted system are interchanged, the adaptive filter attempts to make an inverse filter of the system to be modelled. The inverse filter can be used in series with the system so that modelled to equalize the total response so that it is close to the ideal one.
- An adaptive filter can be formulated to predict future values of the input signal by minimizing the prediction error. This is close to the idea of linear prediction, as described above.
- Adaptive filtering can also be used to cancel noise in a signal, thus enhancing signal quality. An important subproblem is *echo cancellation* (Sondhi *et al.*, 1995), where an echo in a system, acoustic or electronic, is attenuated.

3.4 Hidden Markov Models

Statistical modelling is used extensively in speech processing in particular, where a speech signal is interpreted as a sequence of units, each unit consisting of one or more states and having transition probabilities between states. Typically, a separable unit of speech is a speech

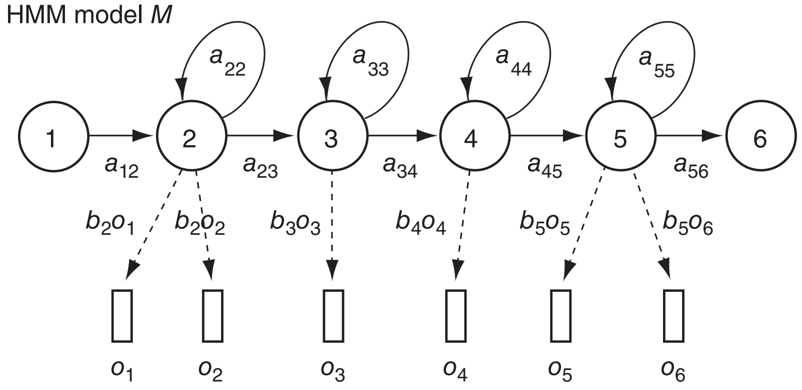

Figure 3.14 An example of a hidden Markov model having six states.

sound, for instance a phone, and a state in a phone represents a statistically stable segment in it, say the beginning, middle, or the end part. In speech recognition (Section 16.3) the problem is how to find such a sequence of units, given a speech signal or feature vectors describing it. A *hidden Markov model* (HMM) (Lee, 1989) is a popular learning method to do this.

An example of an HMM is illustrated in Figure 3.14 as a state transition diagram consisting of six states, $X = 1, 2, \ldots 6$, and transition probabilities a_{ij}. Some of the transitions return to the same state but others proceed to the next. If the model is used as an event generator, each state transition emits an observation vector $O = o_i$. In our case, traversing through states $X = 1 \ldots 6$, the model emits an observation vector sequence $o_1 \ldots o_6$. Here, the initial and final states do not emit observation vectors and states $X = 2$ and $X = 5$ emit two observations due to one transition returning to the same state. Such a model may be applied to represent any size of speech or sound event: a phone (Section 5.2), a syllable, a word, a longer message, or a sound or a passage in music.

If an HMM is used for recognition rather than generation of observations, the problem is that only observations, typically the feature vectors analysed from a speech signal, are known. The joint probability that the known observation $O = o_1, o_2, \ldots o_N$ is emitted by model M and state sequence $X = 1 \ldots 6$ is the product $P(O, X|M) = a_{12}b_2(o_1)\, a_{22}b_2(o_2) \ldots$. A simple way to use HMMs in pattern recognition is to select the recognition result for the model M that gives the highest probability $P(O, X|M)$ for the feature vector sequence $O = o_i$. The name ‘hidden’ Markov model comes from the fact that the state sequence X_i is hidden, not known.

3.5 Concepts of Intelligent and Learning Systems

There are several useful formulations for learning principles that may be combined to develop complex signal and information processing. *Artificial intelligence* is a commonly used term for the development of information and knowledge processing systems that try to mimic what is considered human intelligence. Some specific topics in intelligent and learning systems are the following:

- *Artificial neural networks* (ANNs) (Haykin, 1994; Kohonen, 1990; Lippmann, 1987; Luo and Unbehauen, 1997) are signal and information processing techniques that learn or are trained to acquire a targeted behaviour, typically an input–output relationship of interest. Some principles of ANNs are found by trying to simulate the behaviour of biological neural nets, but the similarity is not very strong, and true neural networks, for example in the brain, may work in quite different ways. Learning ability and simple self-organization are, however, important properties that ANNs can provide.

- *Pattern recognition* is the general idea of reducing complexity of signals in order to find the essential information carried by them (Duda *et al.*, 2012). Pattern recognition is typically a classification (categorization) process whereby irrelevant and redundant information is discarded and a signal is represented by features or classifications. Neural networks and hidden Markov models are among the most popular methods of pattern recognition, especially in speech recognition.
- *Fuzzy systems* are used to form logically operable representations for recognition and control but without the true/false dichotomy of classical logic (Wang, 1994). Using words and concepts that have a flexible yet quantitative interpretation allows for making fuzzy inferences and control.
- *Knowledge-based systems* utilize different paradigms of artificial intelligence, the idea being to process information in a way that resembles human conceptual processing (Hayes-Roth *et al.*, 1983). *Rule-based systems* are often combinations of logic processing and object methodology (Gupta *et al.*, 1986). *Expert systems* are knowledge engineering systems that try to simulate human expert capabilities.
- *Genetic algorithms* and *evolutionary systems* tend to mimic the principles of biological evolution (Goldberg and Holland, 1988). Genetic algorithms simulate evolution through mutation and selection of the best candidates for further evolution towards an optimal solution of a given problem.

Summary

The purpose of this chapter has been to gather concepts that are fundamental in signal processing. This topic has become a major cornerstone both for understanding human communications and for developing engineering solutions to improve communications. Signal processing and its applications will be present throughout the later chapters of this book in one form or another.

Further Reading

Since this book is not particularly about signal processing, this chapter serves only as a brief introduction to the subject, and also familiarizes the reader with the notation used in the field. There are numerous more detailed introductions and textbooks on signal processing, for example (Mitra and Kaiser, 1993; Strawn, 1985; Oppenheim and Schafer, 1975; Proakis, 2007; Steiglitz, 1996; Templaars, 1996).

There are many important topics and applications on signal processing that the interested reader can study elsewhere. The reader is encouraged to read more on signal processing as applied to *audio techniques* in general in Zölzer (2008). There is also a rich literature on adaptive and learning systems, as well as in information processing by artificial intelligence, as has been referred to above.

References

Cohen, L. (1995) *Time–Frequency Analysis*. Prentice Hall.

Duda, R.O., Hart, P.E., and Stork, D.G. (2000) *Pattern Classification*. John Wiley & Sons.

Goldberg, D.E. and Holland, J.H. (1988) Genetic algorithms and machine learning. *Machine Learning*, **3**(2), 95–99.

Gupta, A., Forgy, C., Newell, A., and Wedig, R. (1986) Parallel algorithms and architectures for rule-based systems. *ACM SIGARCH Computer Architecture News*, **14**(2), 28–37.

Hayes-Roth, F., Waterman, D.A., and Lenat, D.B. (1983) Building expert systems. *Teknowledge Series in Knowledge Engineering*.

Haykin, S. (1989) *Modern Filters*. Macmillan.

Haykin, S. (1994) *Neural Networks, A Comprehensive Foundation*. Macmillan College Publishing.

Haykin, S. (2005) *Adaptive Filter Theory*. Pearson Education.

Holighaus, N., Dorfler, M., Velasco, G.A., and Grill, T. (2013) A framework for invertible, real-time constant-Q transforms. *IEEE Trans. Audio, Speech, and Language Proc.*, **21**(4), 775–785.

Jackson, L.B. (1989) *Digital Filters and Signal Processing*. Kluwer Academic.

Kohonen, T. (1990) The self-organizing map. *Proc. of IEEE*, **78**(9), 1464–1480.

Lee, K.F. (1989) *Automatic Speech Recognition, the Development of the SPHINX System*. Kluwer Academic.

Lippmann, R.P. (1987) An introduction to computing with neural nets. *IEEE ASSP Mag.*, **4**, 4–22.

Luo, F.L. and Unbehauen, R. (1997) *Applied Neural Networks for Signal Processing*. Cambridge University Press.

Markel, J.D. and Gray, A.H. (1976) *Liner Prediction of Speech Signals*. Springer.

Mitra, S. and Kaiser, J. (eds) (1993) *Handbook of Digital Signal Processing*. John Wiley & Sons.

Oppenheim, A.V. and Schafer, R.W. (1975) *Digital Signal Processing*. Prentice-Hall.

Oppenheim, A.V., Willsky, A., and Young, I. (1983) *Signals and Systems*. Prentice-Hall.

Parks, T.W. and Burrus, C.S. (1987) *Digital Filter Design*. Wiley.

Proakis, J.G. (2007) *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th edn. Pearson Education.

Schörkhuber, C., Klapuri, A., and Sontacchi, A. (2013) Audio pitch shifting using the constant-Q transform. *J. Audio Eng. Soc.*, **61**(7/8), 562–572.

Sondhi, M.M., Morgan, D.R., and Hall, J.L. (1995) Stereophonic acoustic echo cancellation–an overview of the fundamental problem. *IEEE Signal Proc. Letters*, **2**(8), 148–151.

Steiglitz, K. (1996) *A Digital Signal Processing Primer*. Addison-Wesley.

Strawn, J. (ed.) (1985) *Digital Audio Signal Processing: An Anthology*. William Kaufmann.

Templaars, S. (1996) *Signal Processing, Speech and Music*. Swets & Zeitlinger.

Vetterli, M. and Kovacevic, J. (1995) *Wavelets and Subband Coding*. Prentice-Hall.

Wang, L.X. (1994) *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Prentice-Hall.

Widrow, B. and Stearns, S.D. (1985) *Adaptive Signal Processing*. Prentice-Hall.

Zölzer, U. (2008) *Digital Audio Signal Processing*. John Wiley & Sons.