

# Chapter 1

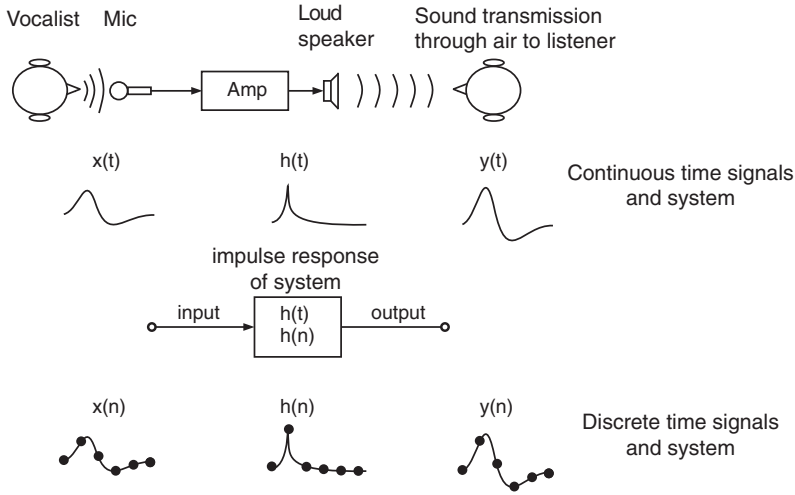
## Introduction

*U. Zölzer*

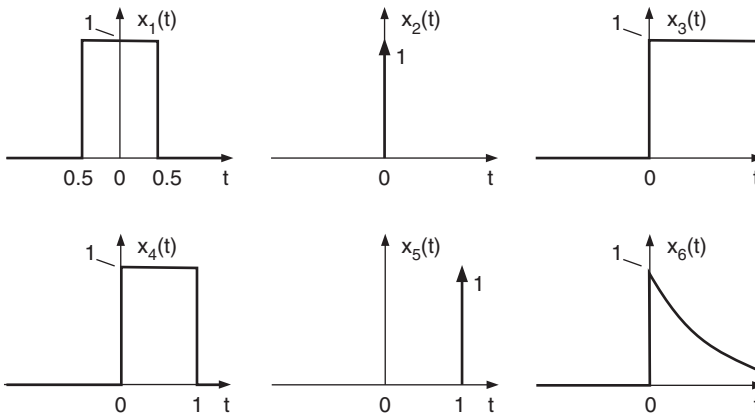
In this first chapter, we will introduce the basics of signals and systems, and describe the transmission of signals through these systems [Opp14]. These fundamental concepts and the describing algorithms lay the foundation for digital audio signal processing. We will start with analog signals and analog systems, then we will sample the analog signals and perform digital signal processing, and finally reconstruct an analog output signal from the digital output signal. Figure 1.1 shows a typical audio application of capturing a vocalist and transmission to a loudspeaker via an amplifier for reproduction in another room for a listener or listening audience. The microphone delivers an electrical input signal  $x(t)$  and the output signal  $y(t)$  is the signal that will be received by the listener's ear. Both signals are continuous-time input and output signals. The entire chain of operations from microphone, amplifier, loudspeaker, and sound transmission through the listening room to the listener can be modeled by a system with a continuous-time impulse response  $h(t)$ . Such an impulse response can be acquired by an impulse response measurement approach. The entire continuous-time approach description can also be represented by a discrete-time approach through sampling the microphone signal  $x(n)$ , using the discrete-time impulse response  $h(n)$ , and then delivering the output signal  $y(n)$ . Both continuous-time and discrete-time signal-processing techniques [Opp10, Opp14] will be introduced in the following sections.

### 1.1 Continuous-time Signals and Convolution

**Continuous-time signals**  $x(t)$ , as shown in Fig. 1.2, can be used as test signals to analyze the behavior of the response of a physical system to an excitation signal. We need a few simple test signals that will allow for the derivation of all important relations to obtain the input/output



**Figure 1.1** Audio capturing and reproduction for a listener, and representations of the operations by a signal and system model with input and output signals and by a system represented by an impulse response.



**Figure 1.2** Continuous-time signals  $x_1(t) = \text{rect}(t)$ ,  $x_2(t) = \delta(t)$ ,  $x_3(t) = \epsilon(t)$ ,  $x_4(t) = x_2(t - 0.5)$ ,  $x_5(t) = x_2(t - 1)$ , and  $x_6(t) = \exp(-at) \cdot \epsilon(t)$ .

description of an input signal transformed to an output signal (handclap → acoustical transmission through room → received by human ear). The rectangular (rect) function is defined by

$$\text{rect}\left(\frac{t}{T}\right) = \begin{cases} T & \text{for } |t| < \frac{T}{2}, \\ 0 & \text{for } |t| > \frac{T}{2}. \end{cases} \tag{1.1}$$

The Dirac impulse  $\delta(t)$  is defined by

$$\delta(t) = \lim_{T \rightarrow 0} \frac{1}{T} \text{rect}\left(\frac{t}{T}\right) \Rightarrow \int_{-\infty}^{\infty} \delta(t) dt = 1. \tag{1.2}$$

The step function is defined by

$$\epsilon(t) = \begin{cases} 0 & \text{for } t < 0, \\ 1 & \text{for } t > 0. \end{cases} \quad (1.3)$$

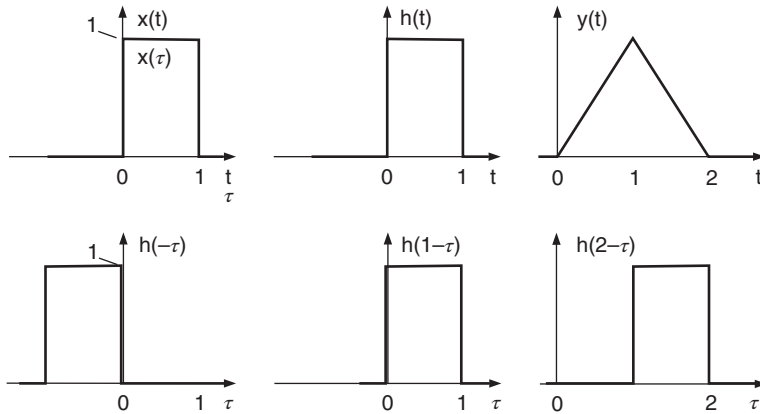
A general signal  $x(t)$  can be written using the sampling property of the Dirac impulse as

$$x(t) = \int_{-\infty}^{\infty} x(\tau)\delta(t - \tau) d\tau. \quad (1.4)$$

**Continuous-time systems** transform the input  $x(t)$  to the output  $y(t) = \mathcal{T}\{x(t)\}$ . A time-domain description can be given by the following signal flow graph:  $x(t) \rightarrow \boxed{h(t)} \rightarrow y(t)$ . The system parameter inside the box is called the *impulse response*  $h(t) = \mathcal{T}\{\delta(t)\}$  of the system. It describes the output of the system  $y(t)$  when the input is the Dirac  $x(t) = \delta(t)$ . Using Eq. (1.4), we can easily derive that the input/output relation of a system with impulse response  $h(t)$  is given by the integral (sliding the folded impulse response  $h(t - \tau)$  along the input and performing weighting and integration)

$$y(t) = \mathcal{T}\{x(t)\} = \int_{-\infty}^{\infty} x(\tau) \cdot h(t - \tau) d\tau = \int_{-\infty}^{\infty} x(t - \tau) \cdot h(\tau) d\tau, \quad (1.5)$$

which is called continuous-time convolution. The *convolution integral* describes a *filter operation* and is written as  $y(t) = x(t) * h(t)$ . *Causality* of a system implies  $h(t) = 0$  for  $t < 0$  and *stability* of a system is achieved if the integral of impulse response  $\int_{-\infty}^{\infty} |h(t)| dt < M < \infty$ . A simple example for continuous-time convolution is demonstrated in Fig. 1.3.



**Figure 1.3** Continuous-time convolution  $y(t) = \int x(\tau)h(t - \tau) d\tau$  showing the folded version of the impulse response  $h(-\tau)$  and shifted versions  $h(t - \tau)$  for  $t = 0, 1, 2$ .

Using the complex exponential  $x(t) = e^{j\omega t}$  as input with  $\omega = 2\pi f$ , the output is given by the convolution integral as

$$y(t) = \int_{-\infty}^{\infty} e^{j\omega(t-\tau)} \cdot h(\tau) d\tau = e^{j\omega t} \cdot \underbrace{\int_{-\infty}^{\infty} h(\tau) \cdot e^{-j\omega\tau} d\tau}_{H(j\omega) \rightarrow \text{Fourier integral}} \quad (1.6)$$

$$= e^{j\omega t} \cdot H(j\omega) = e^{j\omega t} \cdot |H(j\omega)|e^{j\phi(\omega)}. \quad (1.7)$$

This shows that for an exponential input  $x(t)$ , the output  $y(t)$  is again an exponential signal where the input signal is weighted by the complex number  $H(j\omega) = |H(j\omega)|e^{j\varphi(\omega)}$ , which is the Fourier transform (integral) of the impulse response  $h(t)$ , and is also called the *frequency response* of a continuous-time system given by

$$H(j\omega) = \int_{-\infty}^{\infty} h(t) \cdot e^{-j\omega t} dt. \quad (1.8)$$

From  $H(j\omega)$ , we can compute the magnitude response

$$|H(j\omega)| = \sqrt{\operatorname{Re}^2\{H(j\omega)\} + \operatorname{Im}^2\{H(j\omega)\}} \quad (1.9)$$

and the phase response

$$\varphi(\omega) = \arctan\left(\frac{\operatorname{Im}\{H(j\omega)\}}{\operatorname{Re}\{H(j\omega)\}}\right), \operatorname{Re}\{H(j\omega)\} > 0 \quad (1.10)$$

of a continuous-time system. For a given signal  $x(t)$ , we can give its *continuous-time Fourier transform* as

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt \wedge X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt. \quad (1.11)$$

The *Fourier integral* describes a *spectral transform* from time domain  $x(t)$  to frequency domain  $X(j\omega)$ , which is called the *Fourier spectrum* or *Fourier transform* of  $x(t)$ . The *inverse continuous-time Fourier transform* is given by

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\omega) \cdot e^{j\omega t} d\omega \wedge x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{j2\pi ft} df, \quad (1.12)$$

which takes the Fourier spectrum  $X(j\omega)$  and reconstructs the input  $x(t)$ . In the following, useful Fourier transform pairs are listed in Eqs. (1.13)–(1.23). An important relation between the time domain  $x(t) \rightarrow \boxed{h(t)} \rightarrow y(t)$ , using  $x(t)$  and  $h(t)$  giving  $y(t)$ , and frequency domain  $X(j\omega) \rightarrow \boxed{H(j\omega)} \rightarrow Y(j\omega)$ , using  $X(j\omega)$  and  $H(j\omega)$  giving  $Y(j\omega)$ , shows that convolution in the time domain can be described by multiplication in the frequency domain.

## Fourier Transform Pairs<sup>1</sup>

$$x(t) \circ\bullet X(f) \quad (1.13)$$

$$x(t - t_0) \circ\bullet X(f) \cdot e^{-j2\pi ft_0} \quad (1.14)$$

$$x(t) \cdot e^{j2\pi f_c t} \circ\bullet X(f - f_c) \quad (1.15)$$

$$x(t) \cdot \cos(2\pi f_c t) \circ\bullet \frac{1}{2}[X(f - f_c) + X(f + f_c)] \quad (1.16)$$

$$c_1 \cdot x_1(t) + c_2 \cdot x_2(t) \circ\bullet c_1 \cdot X_1(f) + c_2 \cdot X_2(f) \quad (1.17)$$

$$x_1(t) * x_2(t) \circ\bullet X_1(f) \cdot X_2(f) \quad (1.18)$$

---

<sup>1</sup>  $\operatorname{sinc}(f) = \frac{\sin(\pi f)}{\pi f}$ .

$$x(t) \rightarrow \boxed{h(t)} \rightarrow y(t) \circ \bullet X(j\omega) \rightarrow \boxed{H(j\omega)} \rightarrow Y(j\omega) \tag{1.19}$$

$$\delta(t) \circ \bullet 1 \tag{1.20}$$

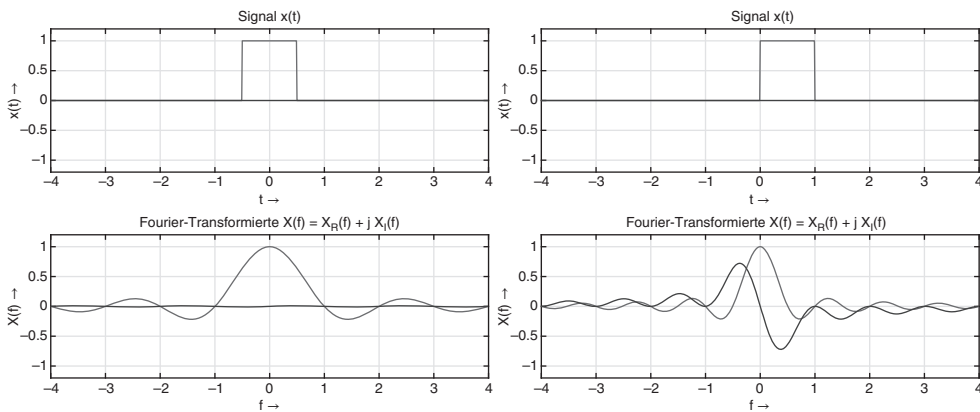
$$\delta(t - t_0) \circ \bullet e^{-j2\pi f t_0} \tag{1.21}$$

$$\text{rect}\left(\frac{t}{T}\right) \circ \bullet T \cdot \text{sinc}(Tf) \tag{1.22}$$

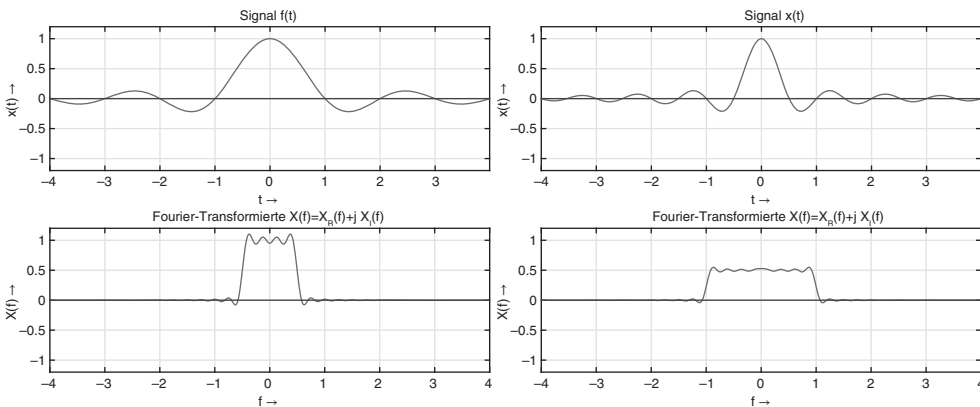
$$f_S \cdot \text{sinc}(f_S t) \circ \bullet \text{rect}\left(\frac{f}{f_S}\right) \tag{1.23}$$

### Fourier Transforms of Even and Causal Signals

Figure 1.4 shows the Fourier transforms of even and causal rect signals and Fig. 1.5 shows two even sinc signals and their Fourier transforms. The ripple in the passband is based on the truncated length of the sinc signal.



**Figure 1.4** Fourier transforms of an even and a causal rect signal. The small imaginary part of the lower left plot arises from a small asymmetry of the rect signal in the upper left plot.



**Figure 1.5** Fourier transforms of two even sinc signals.

## 1.2 Continuous-time Fourier Transform and Laplace Transform

The extension of the continuous-time Fourier transform to the Laplace transform allows for the transform of signals and impulse responses where the Fourier transform does not converge but the Laplace transform converges for a given convergence region. This extension of the continuous-time Fourier transform

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j\omega t} dt \quad (1.24)$$

is achieved by introducing a real part to the imaginary part according to a new complex variable  $s = \sigma + j\omega$ , which then gives

$$X(\sigma + j\omega) = \int_{-\infty}^{\infty} x(t) \cdot e^{-(\sigma + j\omega)t} dt \quad (1.25)$$

$$= \int_{-\infty}^{\infty} x(t) \cdot e^{-\sigma t} \cdot e^{-j\omega t} dt \quad (1.26)$$

and thus the Laplace transform

$$X(s) = \int_{-\infty}^{\infty} x(t) \cdot e^{-st} dt. \quad (1.27)$$

The Laplace transform of signals often leads to a rational function  $X(s) = \frac{N(s)}{D(s)}$  with a numerator polynomial  $N(s)$  and a denominator polynomial  $D(s)$  in the variable  $s$ . The zeros of the numerator  $N(s)$  are called the zeros of  $X(s)$  and the zeros of  $D(s)$  are called the poles of  $X(s)$ . The rational function  $X(s)$  can be given in polynomial, pole/zero, and partial expansion forms.

## 1.3 Sampling and Reconstruction

For digital signal processing, the sampling of  $x(t)$  with a sampling rate  $f_s = \frac{1}{T_s}$  and a sampling interval  $T_s$  is performed, which leads to a sequence of numbers  $x(n)$  with time index  $n$ . According to the sampling theorem, the input signal  $x(t)$  must be band limited to  $f_s/2$ . The sampling and the reconstruction of  $x(t)$  from the number sequence  $x(n)$  is achieved by the following sequence of operations:  $x(t) \rightarrow \text{ADC} \rightarrow x(n) \rightarrow \text{DAC} \rightarrow x(t)$ . Both operations are performed by an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC). The converters can be considered as mixed continuous-time and discrete-time systems.

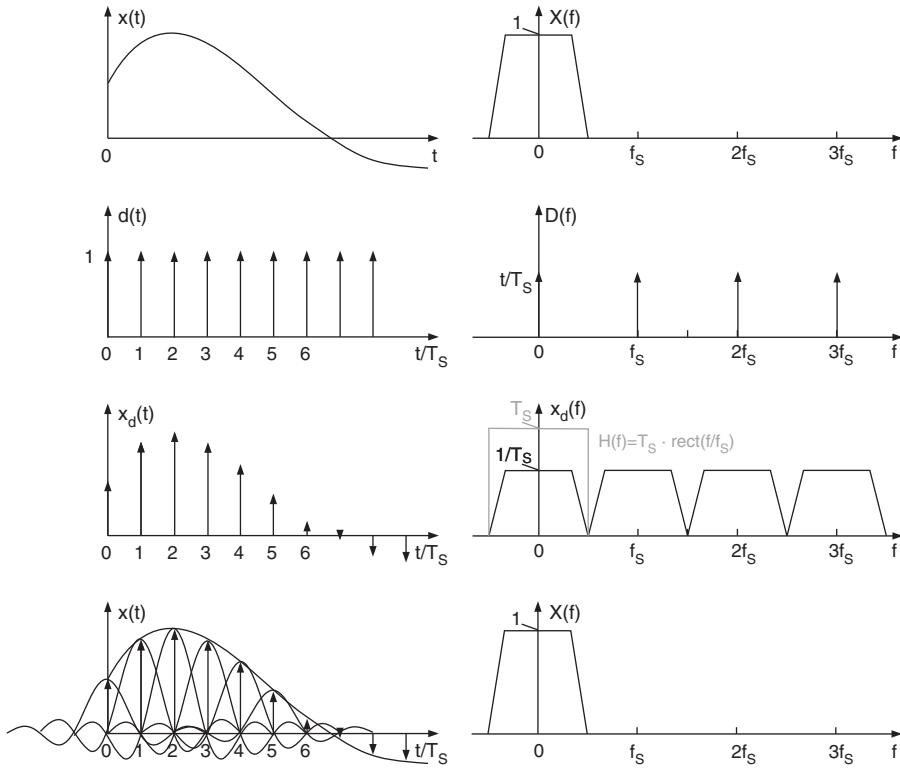
**Sampling and quantization** (analog-to-digital conversion) can be described by

$$x_d(t) = x(t) \cdot d(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad (1.28)$$

$$= \sum_{n=-\infty}^{\infty} x_d(nT_s) \cdot \delta(t - nT_s), \quad (1.29)$$

$$x(n) = [x_d(nT_s)]_Q, \quad (1.30)$$

where the input  $x(t)$  is sampled by multiplying it with a series of Dirac impulses  $d(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$  giving the ideal sampled  $x_d(t)$  and then quantization of the samples  $x_d(nT_s)$  to the sequence of numbers  $x(n)$  with a finite number representation. Figure 1.6 shows in the left column the time-domain signals involved.



**Figure 1.6** Sampling and reconstruction – Time-domain signals (left column) and corresponding Fourier spectra (right column).

**Reconstruction** (digital-to-analog conversion) of the continuous-time  $x(t)$  from the sampled sequence  $x(n)$  can be written as a convolution operation given by

$$x(t) = \sum_{n=-\infty}^{\infty} x(n) \cdot \text{sinc}(f_s(t - nT_s)), \tag{1.31}$$

which is shown in the bottom left plot of Fig. 1.6. The **Fourier transforms** of the individual signals for sampling are given by

$$x(t) \circ \bullet X(f) \tag{1.32}$$

$$d(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \circ \bullet D(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} \delta(f - nf_s) \tag{1.33}$$

$$x_d(t) = x(t) \cdot d(t) \circ \bullet X_d(f) = X(f) * D(f) \tag{1.34}$$

$$= \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - nf_s) \tag{1.35}$$

and are shown in Fig. 1.6 (right column). Sampling leads to the periodic extension of the baseband spectrum at multiples of the sampling rate  $f_S$  and the scaling by  $1/T_S$ . The reconstruction of

$$x(t) = x_d(t) * h(t) \quad \bullet \rightarrow X(f) = X_d(f) \cdot H(f) \quad (1.36)$$

$$= X_d(f) \cdot T_S \cdot \text{rect}\left(\frac{f}{f_S}\right) \quad (1.37)$$

is achieved by convolution with a system impulse response  $h(t)$ , which acts as an ideal lowpass filter with cutoff frequency  $f_S/2$  and passband gain of  $T_S$  to compensate for the scaling of the sampling operation, as shown in the bottom plots of Fig. 1.6.

## 1.4 Discrete-time Signals and Convolution

With the discrete-time input signal  $x(n)$ , we can describe an input/output relation in the discrete-time domain, in signal flow graph form, as  $x(n) \rightarrow \overline{h(n)} \rightarrow y(n) = \mathcal{T}\{x(n)\}$ , where the input is transformed into an output signal  $y(n)$ . Again we are using discrete-time test signals represented in Fig. 1.7, such as the unit pulse

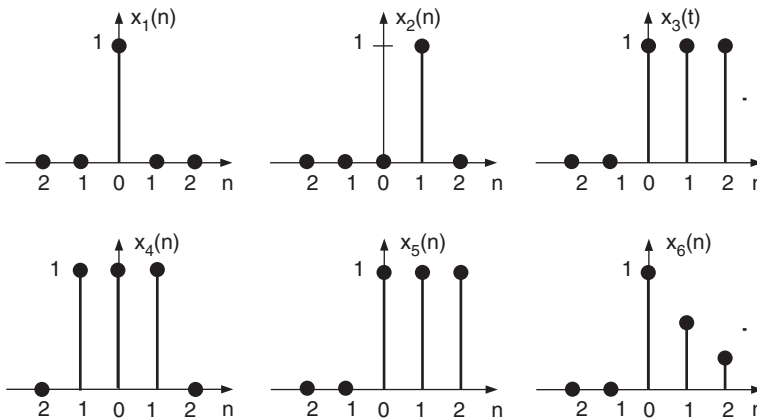
$$\delta(n) = \begin{cases} 1 & \text{for } n = 0, \\ 0 & \text{for } n \neq 0, \end{cases} \quad (1.38)$$

and the step sequence

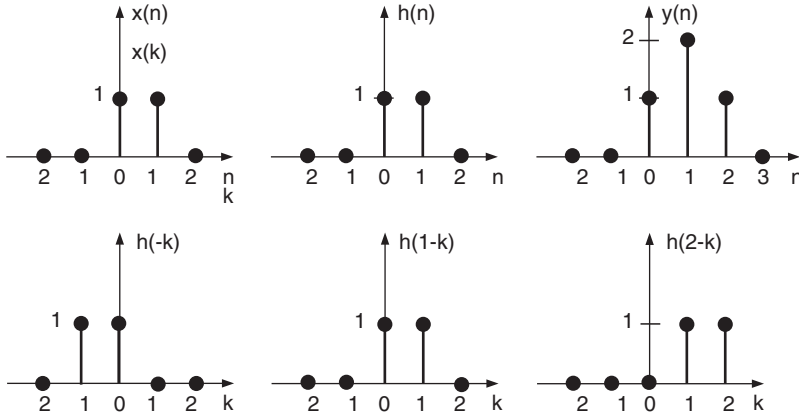
$$\epsilon(n) = \begin{cases} 1 & \text{for } n \geq 0, \\ 0 & \text{for } n < 0. \end{cases} \quad (1.39)$$

The lower row in Fig. 1.7 shows further discrete-time signals which can be formulated using delayed unit pulses  $\delta(n - k)$  according to

$$x(n) = \sum_k x(k) \cdot \delta(n - k) \quad (1.40)$$



**Figure 1.7** Discrete-time signals  $x_1(n) = \delta(n)$ ,  $x_2(n) = \delta(n - 1)$ ,  $x_3(n) = \epsilon(n)$ ,  $x_4(n) = \delta(n + 1) + \delta(n) + \delta(n - 1)$ ,  $x_5(n) = x_4(n - 1)$ , and  $x_6(n) = \exp(-an) \cdot \epsilon(n)$ .



**Figure 1.8** Discrete-time convolution  $y(n) = \sum_k x(k)h(n - k)$  showing the folded version of the impulse response  $h(-k)$  and shifted versions  $h(n - k)$  for  $n = 0, 1, 2$ .

and explicitly as, for example,  $x_6(n) = \exp(-an) \cdot \epsilon(n)$ . The discrete-time system described by  $y(n) = \mathcal{T}\{x(n)\}$  leads to the definition of the impulse response  $h(n) = \mathcal{T}\{\delta(n)\}$  of the discrete-time system when the unit pulse  $\delta(n)$  is used as an input signal  $x(n)$ . Using Eq. (1.40), we can derive the discrete-time convolution as

$$y(n) = \mathcal{T}\{x(n)\} = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n - k) = \sum_{k=-\infty}^{\infty} x(n - k) \cdot h(k). \quad (1.41)$$

The *convolution sum* describes a *digital filter operation* and is written as  $y(n) = x(n) * h(n)$ . The discrete-time convolution is illustrated in Fig. 1.8. *Causality* of a discrete-time system implies  $h(n) = 0$  for  $n < 0$  and *stability* of a discrete-time system is achieved if  $\sum_n |h(n)| < M < \infty$ .

Using the complex exponential  $x(n) = e^{j\Omega n}$  as input with  $\Omega = 2\pi \frac{f}{f_s}$ , the output is given by the convolution sum as

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} e^{j\Omega(n-k)} \cdot h(k) = e^{j\Omega n} \cdot \underbrace{\sum_k h(k) \cdot e^{-j\Omega k}}_{H(e^{j\Omega})} \\ &= e^{j\Omega n} \cdot H(e^{j\Omega}) = e^{j\Omega n} \cdot |H(e^{j\Omega})| e^{-j\varphi(\Omega)}. \end{aligned}$$

This shows that for an exponential input  $x(n)$ , the output  $y(n)$  is again an exponential signal where the input signal is weighted by the complex number  $H(e^{j\Omega}) = |H(e^{j\Omega})| e^{-j\varphi(\Omega)}$ , which is the discrete-time Fourier transform of the impulse response  $h(n)$ , and is also called the *frequency response*

$$H(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} h(n) \cdot e^{-j\Omega n} \quad (1.42)$$

of a discrete-time system. From  $H(e^{j\Omega})$ , we can compute the magnitude response

$$|H(e^{j\Omega})| = \sqrt{\text{Re}^2\{H(e^{j\Omega})\} + \text{Im}^2\{H(e^{j\Omega})\}} \quad (1.43)$$

and the phase response

$$\varphi(\Omega) = \arctan \left( \frac{\text{Im}\{H(e^{j\Omega})\}}{\text{Re}\{H(e^{j\Omega})\}} \right), \text{Re}\{H(e^{j\Omega})\} > 0 \quad (1.44)$$

of a discrete-time system. For a given signal  $x(n)$ , we can give its *discrete-time Fourier transform* as

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n) \cdot e^{-j\Omega n}. \quad (1.45)$$

The *discrete-time Fourier Transform* describes a *spectral transform* from the discrete-time domain  $x(n)$  to the frequency domain  $X(e^{j\Omega})$ , which is called the *Fourier spectrum* or *Fourier transform* of  $x(n)$ . The *inverse discrete-time Fourier transform* is given by

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\Omega}) \cdot e^{j\Omega n} d\Omega, \quad (1.46)$$

which takes the discrete-time Fourier spectrum  $X(e^{j\Omega})$  and reconstructs the input  $x(n)$ . In the following, useful discrete-time Fourier transform pairs are listed in Eqs. (1.47)–(1.56). An important relation between the time domain  $x(n) \rightarrow \boxed{h(n)} \rightarrow y(n)$ , using  $x(n)$  and  $h(n)$  giving  $y(n)$ , and the frequency domain  $X(j\omega) \rightarrow \boxed{H(j\omega)} \rightarrow Y(e^{j\Omega})$ , using  $X(e^{j\Omega})$  and  $H(e^{j\Omega})$  giving  $Y(e^{j\Omega})$ , shows that convolution in the time domain can be described by multiplication in the frequency domain.

## Discrete-time Fourier Transform Pairs

$$x(n) \circ\!\!\!\rightarrow X(e^{j\Omega}) \quad (1.47)$$

$$x(n-m) \circ\!\!\!\rightarrow X(e^{j\Omega})e^{-jm\Omega} \quad (1.48)$$

$$x(n)e^{j\Omega_0 n} \circ\!\!\!\rightarrow X(e^{j(\Omega-\Omega_0)}) \quad (1.49)$$

$$x_1(n) * x_2(n) \circ\!\!\!\rightarrow X_1(e^{j\Omega}) \cdot X_2(e^{j\Omega}) \quad (1.50)$$

$$x(n) \rightarrow \boxed{h(n)} \rightarrow y(n) \circ\!\!\!\rightarrow X(e^{j\Omega}) \rightarrow \boxed{H(e^{j\Omega})} \rightarrow Y(e^{j\Omega}) \quad (1.51)$$

$$\delta(n) \circ\!\!\!\rightarrow 1 \quad (1.52)$$

$$\delta(n-m) \circ\!\!\!\rightarrow e^{-jm\Omega} \quad (1.53)$$

$$\delta(n) + \delta(n-1) \circ\!\!\!\rightarrow 1 + e^{-j\Omega} = 1 + \cos \Omega - j \sin \Omega \quad (1.54)$$

$$\sum_{k=0}^{N-1} \delta(n-k) \circ\!\!\!\rightarrow \sum_{k=0}^{N-1} e^{-jk\Omega} \quad (1.55)$$

$$\sum_{k=0}^{N-1} h(k)\delta(n-k) \circ\!\!\!\rightarrow \sum_{k=0}^{N-1} h(k)e^{-jk\Omega} \quad (1.56)$$

## 1.5 Discrete-time Fourier Transform and Z-Transform

The extension of the discrete-time Fourier transform to the Z-transform allows for the transform of signals and impulse responses where the discrete-time Fourier transform does not converge but the Z-transform converges for a given convergence region. This extension of the discrete-time Fourier transform

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n) \cdot e^{-j\Omega n} \quad (1.57)$$

is achieved by introducing a radius  $r$  to the complex exponential  $e^{j\Omega}$  according to a new complex variable  $z = re^{j\Omega}$ , which then gives

$$X(re^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n) \cdot (re^{j\Omega})^{-n} \quad (1.58)$$

$$= \sum_{n=-\infty}^{\infty} x(n) \cdot r^{-n} \cdot e^{-j\Omega n} \quad (1.59)$$

and thus the Z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) \cdot z^{-n}. \quad (1.60)$$

The Z-transform of many signals leads to a rational function  $X(z) = \frac{N(z)}{D(z)}$ , with a numerator polynomial  $N(z)$  and a denominator polynomial  $D(z)$  in the variable  $z$ . The zeros of the numerator  $N(z)$  are called the zeros of  $X(z)$  and the zeros of  $D(z)$  are called the poles of  $X(z)$ . The rational function  $X(z)$  can be given in polynomial, pole/zero, and partial expansion forms.

## 1.6 Discrete Fourier Transform

To reduce the the length of the input signal  $x(n)$  to  $N$  samples and to compute the Fourier spectrum only at a finite number of  $N$  frequency samples, we can discretize the unit circle  $\Omega_k = \frac{2\pi}{N}k$  at  $N$  frequency samples, and using  $N$  samples,  $x(n)$  gives

$$X(e^{j\Omega_k}) = \sum_{n=0}^{N-1} x(n) e^{-j\Omega_k n}, \quad \Omega_k = \frac{2\pi}{N}k, \quad k = 0, 1, \dots, N-1 \quad (1.61)$$

and the discrete Fourier transform (DFT)

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \quad (1.62)$$

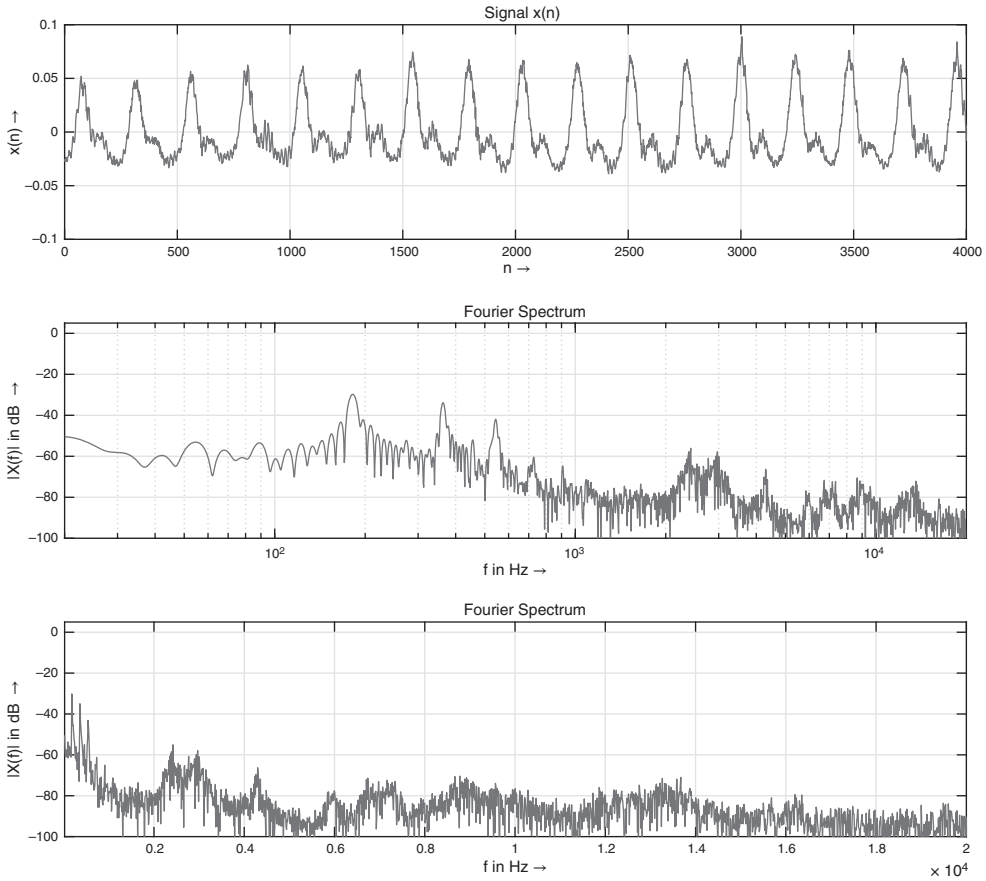
$$k = 0, 1, \dots, N-1$$

and the inverse discrete Fourier transform (IDFT)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}kn} \quad (1.63)$$

$$n = 0, 1, \dots, N-1.$$

The DFT of a causal audio signal is shown in Fig. 1.9.



**Figure 1.9** Fourier transform of an audio signal  $x(n)$  (top plot) and its magnitude spectrum  $|X(f)|$  in dB versus logarithmic (middle plot) and linear frequency axis (bottom plot).

## 1.7 FIR and IIR Filters

For causal signals  $x(n)$  and causal impulse responses  $h(n)$ , the convolution sum is given by  $y(n) = \sum_{k=0}^{\infty} h(k) \cdot x(n-k) = \sum_{k=0}^{\infty} x(k) \cdot h(n-k)$ . A further simplification is possible if the impulse response is decaying very rapidly and can be truncated in length after  $N$  samples, which then leads to a **finite impulse response (FIR)** filter with  $N$  taps (Matlab file `y=filter(b,1,x)`) and the discrete-time convolution

$$y(n) = \sum_{k=0}^{N-1} h(k) \cdot x(n-k) = \sum_{k=0}^{N-1} b(k) \cdot x(n-k), \quad (1.64)$$

which is also called a difference equation. The difference equation can be represented by an FIR signal flow graph (block diagram), as shown in Fig. 1.10. The coefficients  $b(k)$  correspond to the impulse response  $h(k)$ . Deriving the Z-transform of Eq. (1.64) gives

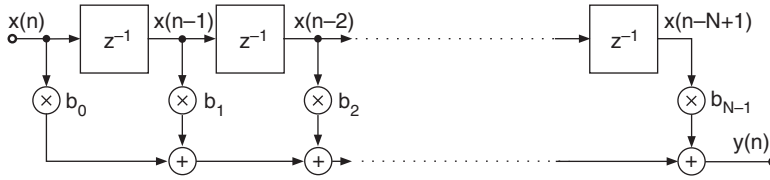


Figure 1.10 Block diagram of FIR filter.

$$Y(z) = \sum_{k=0}^{N-1} b(k) \cdot z^{-k} \cdot X(z), \tag{1.65}$$

from which we can derive the system transfer function  $H(z) = \frac{Y(z)}{X(z)}$ , which gives

$$H(z) = \sum_{k=0}^{N-1} b(k) \cdot z^{-k} = \frac{\sum_{k=0}^{N-1} b(k) \cdot z^{N-1-k}}{z^{N-1}}. \tag{1.66}$$

The frequency response of the FIR filter (Matlab file freqz(b,1)) can be derived by taking the discrete-time Fourier transform of the impulse response or evaluating the system transfer function on the unit circle by replacing  $z \rightarrow e^{j\Omega}$ , which gives

$$H_{\text{FIR}}(e^{j\Omega}) = \sum_{k=0}^{N-1} b(k) \cdot e^{-j\Omega k}, \quad \Omega = 2\pi \frac{f}{f_s}. \tag{1.67}$$

Figure 1.11 shows a symmetric FIR impulse response with five taps, the magnitude response, phase response, and the pole/zero plot.

If the impulse response is decaying infinitely long, we have an **infinite impulse response (IIR)** filter of order  $N$  (Matlab file `y=filter(b,[1 a],x)`) which is dependent on its input  $x(n)$  and  $N$  delayed versions  $x(n - k)$ , weighted by the coefficients  $b(k)$ , and  $N$  delayed versions  $y(n - k)$  of the output  $y(n)$ , weighted by the coefficients  $a(k)$  according to

$$\boxed{y(n) = \sum_{k=0}^N b(k) \cdot x(n - k) - \sum_{k=1}^N a(k) \cdot y(n - k)}, \tag{1.68}$$

which is also called a difference equation. The difference equation can be represented by an IIR signal flow graph (block diagram), as shown in Fig. 1.12. Performing the Z-transform of the recursive difference equation in Eq. (1.68) gives

$$Y(z) = \sum_{k=0}^N b(k) \cdot z^{-k} \cdot X(z) - \sum_{k=1}^N a(k) \cdot z^{-k} \cdot Y(z), \tag{1.69}$$

from which we can derive the system transfer function  $H(z) = \frac{Y(z)}{X(z)}$  by deriving

$$H(z) = \frac{\sum_{k=0}^N b(k) \cdot z^{-k}}{1 + \sum_{k=1}^N a(k) \cdot z^{-k}}. \tag{1.70}$$

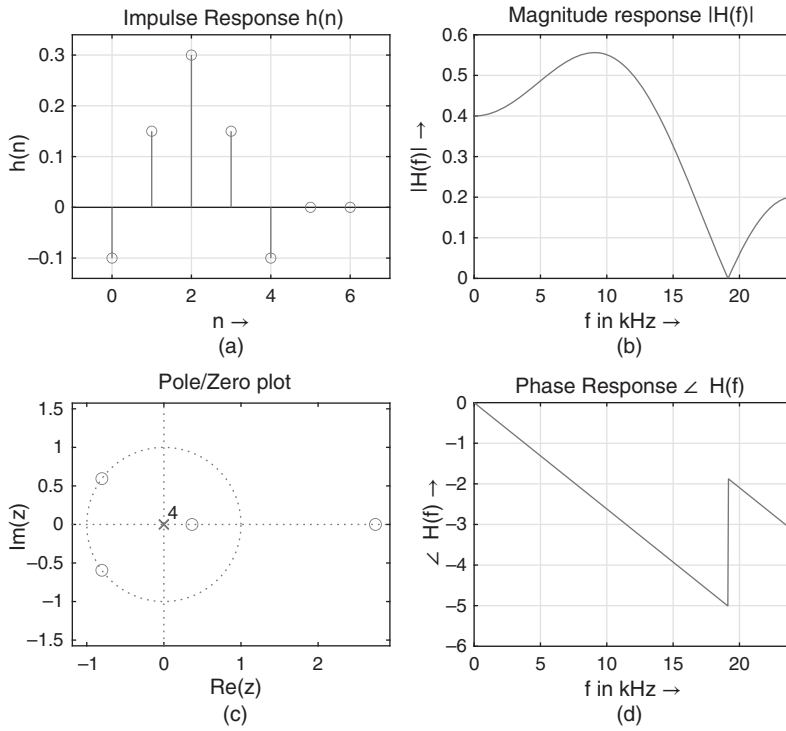


Figure 1.11 FIR filter impulse response, magnitude response, phase response, and pole/zero plot.

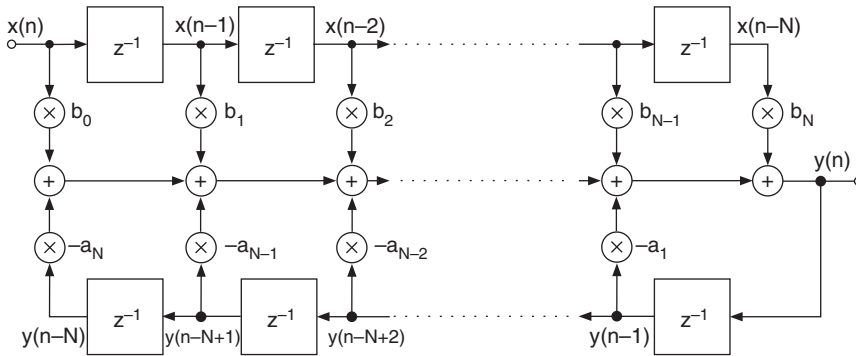


Figure 1.12 Block diagram of an IIR filter.

Typically,  $N = 1$  or  $N = 2$ , and for higher order, we can use a cascade of first- and second-order sub-filters. Figure 1.13 shows a signal flow graph for an IIR filter cascade. The frequency response of the IIR filter ( $\text{freqz}(b, [1 \ a])$ ) can be derived by evaluating the system transfer function on the unit circle by replacing  $z \rightarrow e^{j\Omega}$ , which gives the frequency response of an IIR filter of order  $N$

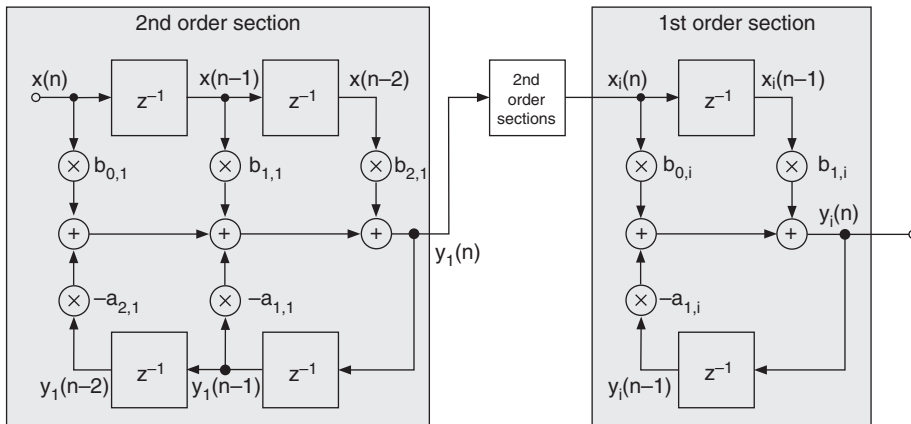


Figure 1.13 IIR filter cascade.

(freqz(b,[1 a])) according to

$$H_{IIR}(e^{j\Omega}) = \frac{\sum_{k=0}^N b(k) \cdot e^{-j\Omega k}}{1 + \sum_{k=1}^N a(k) \cdot e^{-j\Omega k}}, \quad \Omega = 2\pi \frac{f}{f_s}. \quad (1.71)$$

Figure 1.14 shows an IIR impulse response with infinite decay, the magnitude response, phase response, and the pole/zero plot.

Some specific FIR filters are the *N-tap moving average filter*,

$$h_{MA}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \delta(n - k), \quad (1.72)$$

the *triangular filter*,

$$h_{TRI}(n) = h_{MA}(n) * h_{MA}(n), \quad (1.73)$$

and the equi-ripple *Parks–McClellan filter*,

$$h_{PM}(n) = \sum_{k=0}^{N-1} h(k) \delta(n - k), \quad (1.74)$$

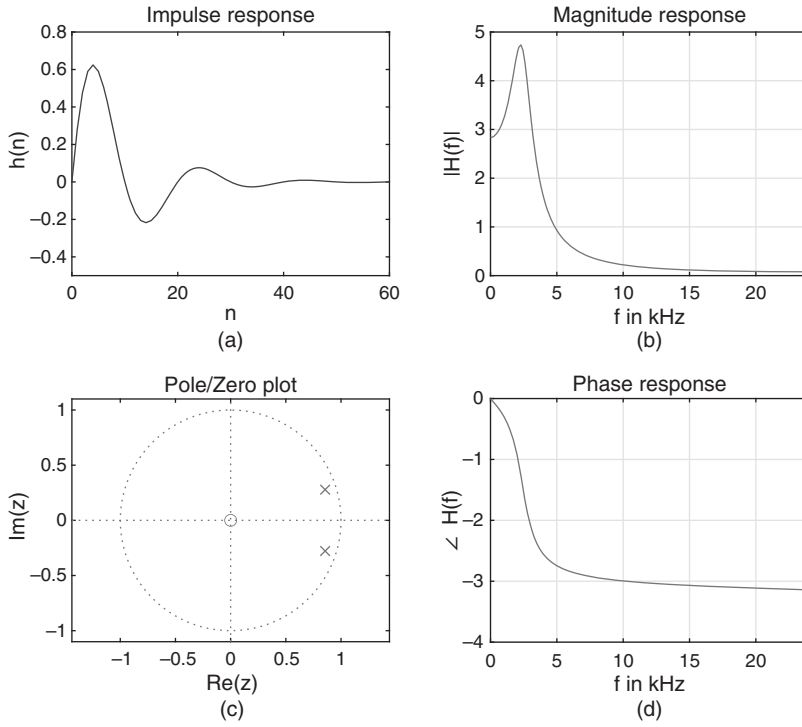
where the impulse response is computed based on the Parks–McClellan [Opp14] design approach. *Complementary filters*, according to

$$h_C(n) = \delta\left(n - \frac{N-1}{2}\right) - \sum_{k=0}^{N-1} h(k) \delta(n - k), \quad (1.75)$$

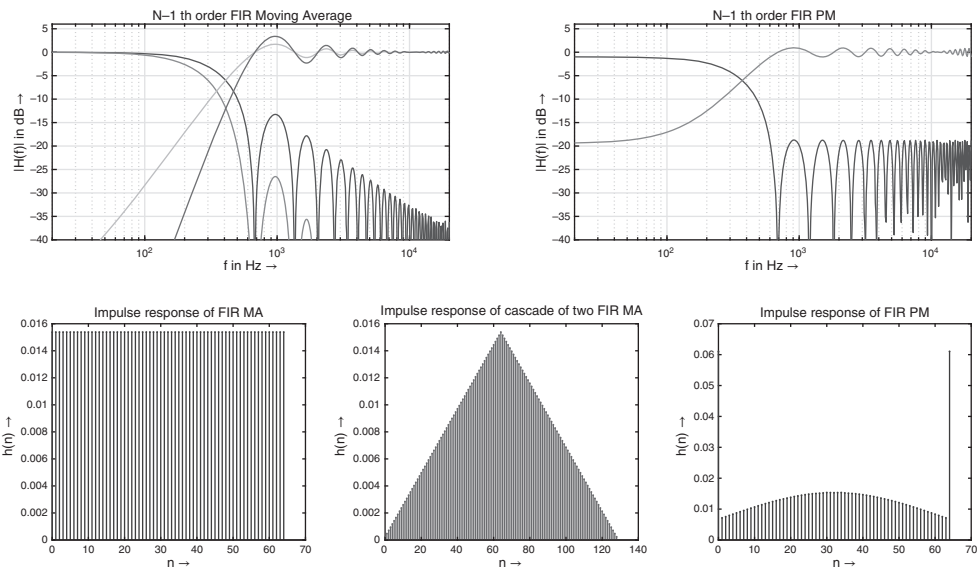
allow for the use of a lowpass FIR filter of odd length *N* to perform lowpass-to-highpass and bandpass-to-bandstop transformations. Some frequency responses of these FIR filters are shown in Fig. 1.15.

Some simple but effective **first- and second-order IIR filters** for lowpass (LP), highpass (HP), and bandpass (BP) filtering are given by

$$H_{LP}(z) = \frac{1-a}{2} \frac{1+z^{-1}}{1-az^{-1}}, \quad (1.76)$$



**Figure 1.14** IIR filter and impulse response, magnitude response, phase response, and pole/zero plot.



**Figure 1.15** Frequency responses and impulse responses of a moving average FIR and a Parks–McClellan FIR filter. In the two frequency response plots, a complementary frequency response is also shown.

$$a = \frac{1 - \sin(2\pi f_c/f_s)}{\cos(2\pi f_c/f_s)}. \tag{1.77}$$

$$H_{HP}(z) = \frac{1+a}{2} \frac{1-z^{-1}}{1-az^{-1}}, \tag{1.78}$$

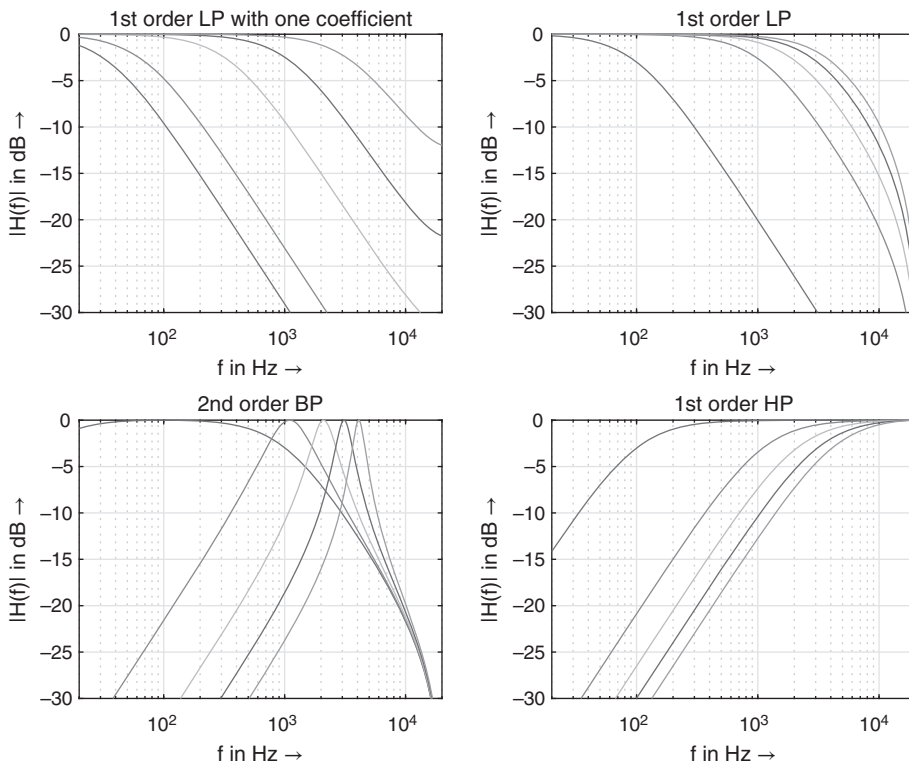
$$a = \frac{1 - \sin(2\pi f_c/f_s)}{\cos(2\pi f_c/f_s)}. \tag{1.79}$$

$$H_{BP}(z) = \frac{1-a}{2} \frac{1-z^{-2}}{1-b(1+a)z^{-1}+az^{-2}}, \tag{1.80}$$

$$a = \frac{1 - \sin(2\pi f_b/f_s)}{\cos(2\pi f_b/f_s)}, \tag{1.81}$$

$$b = \cos(2\pi f_c/f_s). \tag{1.82}$$

They all have one (LP, HP) or two coefficients (BP) for controlling their cutoff frequency  $f_c$  in Hz ( $-3$ -dB magnitude of LP or HP frequency response) and the BP bandwidth  $f_b$  in Hz (lower and higher than  $-3$ -dB magnitude of BP frequency response). Their frequency responses are shown in Fig. 1.16. These simple first- and second-order IIR filters (LP, HP, and BP) can be further implemented using first- and second-order allpass filters  $H_{AP_1}(z)$  and  $H_{AP_2}(z)$ , as shown in the



**Figure 1.16** Frequency response of IIR filter.

sequence of system transfer functions given by

$$H_{AP_1}(z) = \frac{-a + z^{-1}}{1 - az^{-1}}, \quad (1.83)$$

$$H_{AP_2}(z) = \frac{a - b(1+a)z^{-1} + z^{-2}}{1 - b(1+a)z^{-1} + az^{-2}}, \quad (1.84)$$

$$H_{LP}(z) = \frac{1}{2}(1 + H_{AP_1}(z)), \quad (1.85)$$

$$H_{HP}(z) = \frac{1}{2}(1 - H_{AP_1}(z)), \quad (1.86)$$

$$H_{BP}(z) = \frac{1}{2}(1 - H_{AP_2}(z)), \quad (1.87)$$

where the coefficients  $a$  and  $b$  are given by Eq. (1.81) and Eq. (1.82), respectively. The lowpass transfer function in Eq. (1.85) is an additive parallel connection of direct path and a first-order allpass of Eq. (1.83). The highpass transfer function in Eq. (1.86) is a subtractive parallel connection of direct path and a first-order allpass of Eq. (1.83). Finally, the bandpass transfer function in Eq. (1.87) is a subtractive parallel connection of direct path and a first-order allpass of Eq. (1.84).

## 1.8 Adaptive Filters

Adaptive filters change their filter characteristic according to a predefined error criterion. A generic structure is shown in Fig. 1.17 where a desired signal  $d(n)$  is approximated by a reference signal  $x(n)$  passing through an adaptive filter giving an output signal  $y(n)$ , which should be made close to  $d(n)$  such that the error signal  $e(n)$  between the desired signal  $d(n)$  and the output signal  $y(n)$  gets smaller with time. The coefficients of the adaptive filter are derived by minimizing the error energy. Based on the generic structure shown in Fig. 1.17, special adaptive filter approaches for system identification, inverse filtering, echo cancellation, and linear prediction can be achieved. We will show how to perform linear prediction as an adaptive FIR filter technique and derive the adaption of the filter coefficients by the autocorrelation method and the least-mean-square (LMS) method.

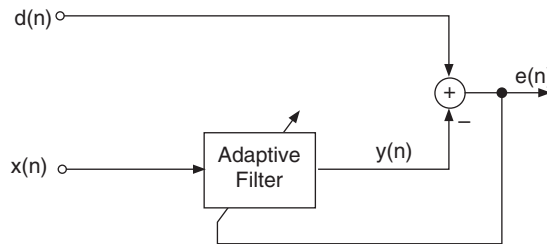


Figure 1.17 Adaptive signal approximation.

### Adaptive Linear Prediction using the Autocorrelation Method

For the task of adaptive linear prediction, the generic structure is simplified because the desired signal  $d(n)$  is equal to the input  $x(n)$ , which leads to the signal flow graph of linear prediction

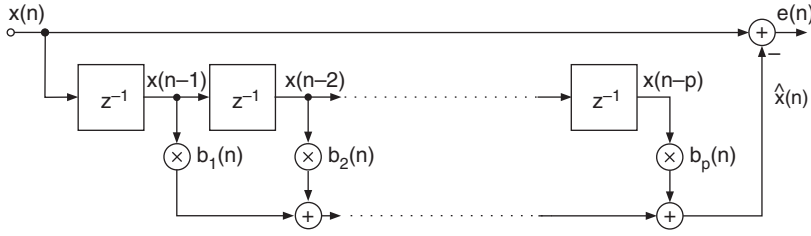


Figure 1.18 Linear prediction.

shown in Fig. 1.18. The adaptive filter is, in that case, an FIR filter with  $p$  coefficients. Linear prediction computes an estimate

$$\hat{x}(n) = \sum_{k=1}^p b(k)x(n-k) = \sum_{k=1}^p b_k x(n-k) \quad (1.88)$$

of the actual sample  $x(n)$  by a linear combination of past weighted input samples. Here  $p$  is the prediction order and  $b_k$  are the prediction coefficients. The difference between  $x(n)$  and its prediction  $\hat{x}(n)$  is called the *prediction error* and is given by

$$e(n) = x(n) - \hat{x}(n) = x(n) - \sum_{k=1}^p b_k x(n-k). \quad (1.89)$$

The filter coefficients  $b_k$  are computed by minimizing a cost function

$$J = E\{e^2(n)\} \quad (1.90)$$

and thus minimizing the quadratic error<sup>2</sup>. The partial derivatives of the cost function with respect to the filter coefficients  $b_k$  (for  $k = 1, \dots, p$ ) lead to

$$\frac{\partial J}{\partial b_k} = 2E \left\{ e(n) \cdot \frac{\partial e(n)}{\partial b_k} \right\} = -2E\{e(n)x(n-k)\} \quad (1.91)$$

$$= -2E \left\{ \left[ x(n) - \sum_{i=1}^p b_i x(n-i) \right] x(n-k) \right\} \quad (1.92)$$

$$= -2E\{x(n)x(n-k)\} + 2 \sum_{i=1}^p b_i E\{x(n-i)x(n-k)\}. \quad (1.93)$$

Computing the filter coefficients  $b_k$  by setting the partial derivative to zero gives, for  $k = 1, \dots, p$ ,

$$\frac{\partial J}{\partial b_k} = 0 \quad (1.94)$$

$$\Leftrightarrow \sum_{i=1}^p b_i E\{x(n-i)x(n-k)\} = E\{x(n)x(n-k)\}. \quad (1.95)$$

By using the autocorrelation sequence

$$r_{xx}(k) = E\{x(n)x(n-k)\} = \sum_{n=-\infty}^{\infty} x(n)x(n-k) \quad (1.96)$$

<sup>2</sup> Expectation  $E\{e^2(n)\} = \sum_n e^2(n)$ .

and the properties

$$\begin{aligned} r_{xx}(k-i) &= E \{x(n)x(n-(k-i))\} \\ &= E \{x(n-i)x(n-k)\} \end{aligned} \quad (1.97)$$

$$\text{and } r_{xx}(k) = r_{xx}(-k) \quad (\text{even sequence}) \quad (1.98)$$

leads to the set of equations (1.95) in matrix form as

$$\begin{pmatrix} r_{xx}(0) & r_{xx}(1) & r_{xx}(2) & \cdots & r_{xx}(p-1) \\ r_{xx}(1) & r_{xx}(0) & r_{xx}(1) & \cdots & r_{xx}(p-2) \\ r_{xx}(2) & r_{xx}(1) & r_{xx}(0) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & r_{xx}(1) \\ r_{xx}(p-1) & r_{xx}(p-2) & \cdots & r_{xx}(1) & r_{xx}(0) \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{pmatrix} = \begin{pmatrix} r_{xx}(1) \\ r_{xx}(2) \\ \vdots \\ r_{xx}(p) \end{pmatrix}. \quad (1.99)$$

These equations are called *Wiener–Hopf equations* or *normal equations*. The so-called autocorrelation matrix  $\mathbf{R}$  is symmetric and has identical values on each diagonal. It is a *Toeplitz matrix* and the set of equations can be solved efficiently by the *Levinson–Durbin recursion*. The described procedure is called the autocorrelation method for updating the prediction coefficients.

## Adaptive Linear Prediction using the LMS Method

In this method, a recursive algorithm for minimizing a simplified cost function  $\hat{J}$  is applied by minimizing the LMS error. All filter coefficients will be updated every sampling cycle based on previous values. This recursive approach for the filter coefficients uses the gradient descent formula

$$b_k(n+1) = b_k(n) - \frac{1}{2}\mu(n)\frac{\partial J}{\partial b_k(n)} \quad (1.100)$$

with gradient weight  $\mu(n)$ . Using the simplified cost function (instantaneous value of the quadratic error)

$$\hat{J} = e^2(n) \quad \text{and} \quad e(n) = x(n) - \sum_{k=1}^p b_k(n)x(n-k) \quad (1.101)$$

leads to

$$\frac{\partial \hat{J}}{\partial b_k(n)} = \frac{\partial e^2(n)}{\partial b_k(n)} = 2e(n) \cdot \frac{\partial e(n)}{\partial b_k} = -2e(n)x(n-k) \quad (1.102)$$

and thus, for  $k = 1, \dots, p$ , the recursive LMS algorithm

$$\boxed{b_k(n+1) = b_k(n) + \mu(n)e(n)x(n-k)}. \quad (1.103)$$

The gradient weight  $\mu$  can be adjusted using the different LMS approaches listed in Table 1.1.

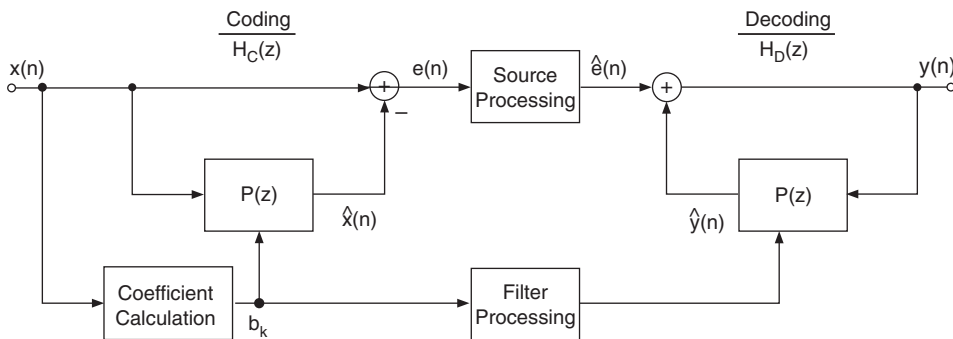
## Linear Prediction for Coding and Source-filter Processing

Linear prediction can be found in coding and source-filter processing which have the same signal flow graph of operation, as shown in Fig. 1.19. The first part is the coding operation and consists of a forward prediction and coefficient calculation using the autocorrelation or LMS method. The coder transfer function is given by

$$H_C(z) = 1 - P(z). \quad (1.104)$$

**Table 1.1** Computing the gradient weight  $\mu(n)$ .

Standard LMS	$\mu(n) = \text{const.}$
Normalized LMS (NLMS)	$\mu(n) = \frac{\alpha}{\beta + \sum_{k=0}^{N-1} x^2(n-k)}$ $\beta$ takes care of large $\mu(n)$
Power Normalized (PNLMS)	$\mu(n) = \frac{\alpha}{N \cdot \sigma_x^2(n)}$
- Limitation:	$\mu(n) = \min \{ \mu(n), \mu_{max} \}$
- Recursive computation:	$\sigma_x^2(n) = \lambda \sigma_x^2(n-1) + (1-\lambda)x^2(n),$ $0 < \lambda < 1$



**Figure 1.19** Linear prediction for coding and source-filter processing.

Then the error signal  $e(n)$  is quantized for the coding application and further processed for source-filter modifications. The receiver performs a decoding operation by inverse filtering using the prediction filter and the corresponding coefficients from the coder or processed and modified coefficients. The decoder transfer function is given by

$$H_D(z) = \frac{1}{1 - P(z)}. \tag{1.105}$$

Source-filter processing extracts, in the time domain, the source signal  $e(n)$  through linear prediction and simultaneously the filter coefficients  $b_k$  of the prediction filter  $P(z)$ . The source signal is a whitened version of the input and the filter transfer function  $H_D(z)$  represents the spectral envelope of the input  $x(n)$ . Both the source signal  $e(n)$  and the prediction coefficients  $b_k$  can be modified according to specific tasks like sound morphing (keeping the source signal and modifying the spectral envelope).

## 1.9 Exercises

### 1. Signals and Systems

1. Sketch the impulse responses:

- (a)  $h(n) = a \cdot \delta(n)$  (amplifier);

- (b)  $h(n) = \delta(n - M)$  (delay by  $M$  samples);
- (c)  $h(n) = 0,5 [\delta(n) + \delta(n - 1)]$  (2-tap moving average, simple lowpass filter);
- (d)  $h(n) = \frac{1}{M} \sum_{k=0}^M \delta(n - k)$  ( $M$ -tap moving average, simple lowpass filter);
- (e)  $h(n) = a^n \cdot \epsilon(n)$ ,  $|a| < 1$  (lowpass filter);
- (f)  $h(n) = a^n \cdot \sin\left(2\pi \frac{1}{20}n\right) \cdot \epsilon(n)$ ,  $|a| < 1$  (lowpass filter);

and compute  $y(n)$  for  $x(n) = \delta(n) + \delta(n - 1)$  using convolution (folding the impulse response and folding the input signal). Write a Matlab script for the computation of  $y(n)$ .

2. Compute the output signals:

- (a)  $y(n) = A \cos(\Omega n) \star h(n)$  (eigen signal passing through system);
- (b)  $y(n) = x(n) + ay(n - 1)$ ,  $|a| < 1$  (first-order lowpass filter);
- (c)  $y(n) = x(n) + y(n - 1)$  (integrator);
- (d)  $y(n) = x(n) - x(n - 1)$  (differentiator);

for  $x(n) = \delta(n)$  and plot the output signals for  $x(n) = \sum_{k=0}^7 \delta(n - k)$  using Matlab.

## 2. Discrete-time Fourier Transform

1. Compute the discrete-time Fourier transform of:

- (a)  $x(n) = a \cdot \delta(n)$ ;
- (b)  $x(n) = \delta(n - M)$ ;
- (c)  $x(n) = 0,5 [\delta(n) + \delta(n - 1)]$ ;
- (d)  $x(n) = a^n \cdot \epsilon(n)$ ,  $|a| < 1$ ;

and plot the results using Matlab.

2. Using the difference equation

$$y(n) = a \sin(\Omega_0 n) \cdot x(n - 1) + 2a \cos(\Omega_0 n) \cdot y(n - 1) - a^2 \cdot y(n - 2) :$$

- (a) sketch the signal flow graph;
- (b) compute the frequency response  $H(e^{j\Omega}) = \frac{Y(e^{j\Omega})}{X(e^{j\Omega})}$ ;
- (c) plot the magnitude response and phase response using Matlab;
- (d) program the difference equation and plot the impulse response.

### 3. FIR and IIR Filters

1. FIR filters with linear phase (even and odd symmetry, linear phase, constant group delay):

(a)  $h(n) = 0.5 \cdot \delta(n) + 0.5 \cdot \delta(n - 1)$ ;

(b)  $h(n) = 0.25 \cdot \delta(n) + 0.5 \cdot \delta(n - 1) + 0.25 \cdot \delta(n)$ ;

(c)  $h(n) = 0.5 \cdot \delta(n) - 0.5 \cdot \delta(n - 1)$ ;

(d)  $h(n) = 0.5 \cdot \delta(n) - 0.5 \cdot \delta(n - 2)$ .

Plot the impulse responses, magnitude and phase responses (freqz.m), pole/zero plots (zplane.m), and group delay (grpdelay.m). Discuss the results.

2. Use FIR filters to approximate a given impulse response by truncating the length of the impulse response to  $N$  taps.
3. Do experimentation with the Parks–McClellan design (firpm.m) and design several lowpass filters with linear phase. Use the Matlab files freqz.m and zplane.m for your evaluation.
4. Show the IIR signal flow graphs of first-order LP and HP filters, and second-order BP filters. Verify the frequency responses.

### 4. Adaptive Filters

1. Write a small script for linear prediction using the LMS algorithm. Use an audio signal and experiment with the prediction order and monitor the error.
2. Modify the linear prediction approach by replacing  $d(n) = x(n) \star h(n)$ , where  $h(n)$  is the unknown system impulse response. Modify the code to perform system identification using the LMS algorithm.

### References

- [Opp14] A.V. Oppenheim, A.S. Willsky, with H. Nawab: *Signals and Systems*, Pearson New International Edition, 2nd Edition, 2014.
- [Opp10] A.V. Oppenheim, R.W. Schaffer, with J.R. Buck: *Discrete-time Signal Processing*, Prentice Hall Signal Processing Series, 3rd Edition, 2010.

