Chapter 1. Introduction to Signal Processing Theory

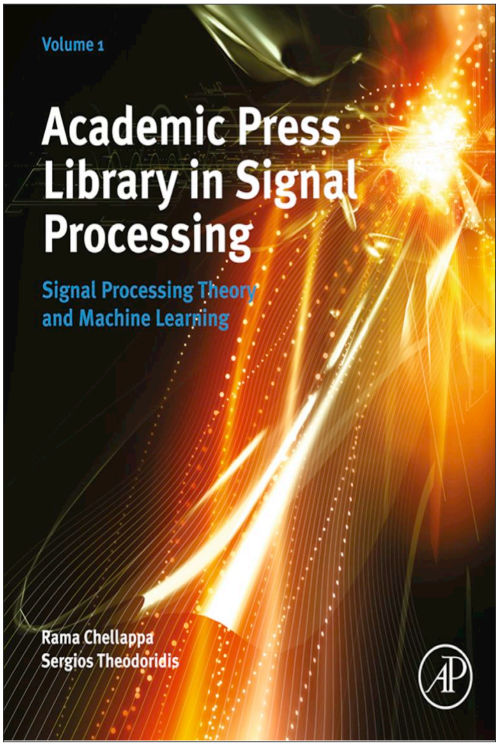
Article · December 2014

DOI: 10.1016/B978-0-12-396502-8.00001-2

2 authors:

Isabela Apolinário
Federal University of Rio de Janeiro

Paulo S. R. Diniz
Federal University of Rio de Janeiro



From Isabela F. Apolinário and Paulo S.R. Diniz, Introduction to Signal Processing Theory. In: Rama Chellappa, Sergios Theodoridis, editors, Academic Press Library in Signal Processing. Vol 1, Signal Processing Theory and Machine Learning, Chennai: Academic Press, 2014, p. 3-28.
ISBN: 978-0-12-396502-8

CHAPTER

Introduction to Signal Processing Theory

1

Isabela F. Apolinário and Paulo S.R. Diniz

Program of Electrical Engineering and the Department of Electronics and Computer Engineering, COPPE/Poli, Universidade Federal do Rio de Janeiro, Brazil

1.01.1 Introduction

Signal processing is a key area of knowledge that finds applications in virtually all aspects of modern life. Indeed the human beings are employing signal processing tools for centuries without realizing it [1]. In present days the younger generation might not be able to understand how one can live without carrying a mobile phone, traveling long distances without an almost self piloted airplane, exploring other planets without human presence, and utilizing a medical facility without a wide range of diagnostic and intervention equipments.

Signal processing consists of mapping or transforming information bearing signals into another form of signals at the output, aiming at some application benefits. This mapping defines a continuous or analog system if it involves functions representing the input and output signals. On the other hand, the system is discrete or digital if its input and output signals are represented by sequences of numbers.

Signal processing theory is very rich and as a reward it has been finding applications in uncountable areas, among which we can mention bioengineering, communications, control, surveillance, environment monitoring, oceanography, and astronomy, just to mention a few. In particular, the enormous advance in digital integrated circuit technology, responsible for the computing and information revolutions that we are so used today, enabled the widespread use of Digital Signal Processing Systems. These systems allowed advances in fields such as: speech, audio, image, video, multirate processing, besides in several applications of wireless communications and digital control.

This chapter is an overview of the classical signal processing tools whose details are encountered in numerous textbooks such as: [1–15]. The topics treated in the following chapters entail the description of many signal processing tools, usually not covered in the classical textbooks available in the market. As a result, we believe that the readers can benefit from reading many of these chapters, if not all, in order to deepen and widen their current knowledge as well as to start exploiting new ideas.

1.01.2 Continuous-time signals and systems

The processing of signals starts by accessing the type of information we want to deal with. Many signals originating from nature are continuous in time and as such, the processing involved must include analog signal acquisition systems followed by the implementation of a continuous-time system if the processing

Academic Press Library in Signal Processing. http://dx.doi.org/10.1016/B978-0-12-396502-8.00001-2

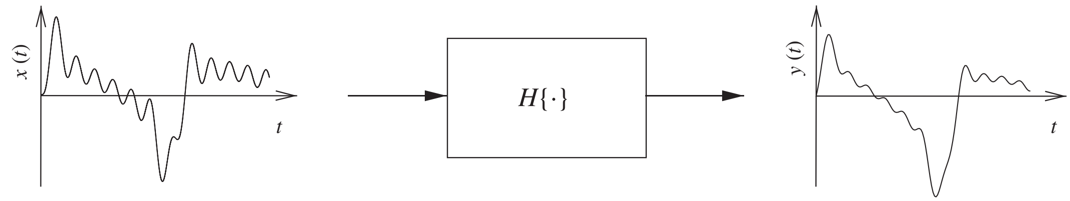


FIGURE 1.1

Continuous-time system.

remains analog all the way. Alternatively, assuming the original continuous-time signal contains limited spectral information, we can sample it in order to generate a sequence representing the continuous-time signal unambiguously[1]. In this latter form one can benefit from the advanced digital integrated circuit technology to process the signal. However, many real life applications still includes continuous-time signal processing at least at the acquisition and actuator phases of the processing.

A continuous-time system maps an analog input signal represented by $x(t)$ into an output signal represented by $y(t)$, as depicted in Figure 1.1. A general representation of this mapping is

$$y(t) = \mathcal{H}\{x(t)\},$$

where $\mathcal{H}\{\cdot\}$ denotes the operation performed by the continuous-time system. If the system is linear and time-invariant (LTI), as will be described in Chapter 2, there are several tools to describe the mapping features of the system. Typically, a general description of the systems consists of a differential equation which in turn can be solved and analyzed by employing the Laplace transform. A key feature of the LTI systems is their full representation through their impulse responses. The behavior of a nonlinear system is more complicated to analyze since it can not be fully characterized by its impulse response.

Another important tools to deal with the continuous-time signals which are periodic and non-periodic are the Fourier series and Fourier transform, respectively. As will be discussed in Chapter 2, the Fourier and Laplace transforms are closely related, being both essential tools to describe the behavior of continuous-time signals when applied to LTI systems.

The Laplace transform is suitable to represent a time-domain non-periodic function of a continuous and real (time) variable, resulting in a frequency-domain non-periodic function of a continuous and complex frequency variable. That is,

$$X(s) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-st}\,\mathrm{d}t \quad \Longleftrightarrow \quad x(t) = \frac{1}{2\pi}\mathrm{e}^{\sigma t}\int_{-\infty}^{\infty} X(\sigma + \mathrm{j}\omega)\mathrm{e}^{\mathrm{j}\omega t}\,\mathrm{d}\omega.$$

The Fourier transform is employed to represent a time-domain non-periodic function of a continuous and real (time) variable with a frequency-domain non-periodic function of a continuous and imaginary

[1]As will be seen in Chapter 5, we can completely reconstruct a continuous-time band-limited signal $x(t)$ from its sampled version $x(n)$ if the sampling frequency is chosen correctly.

frequency variable. The Fourier transform is described by

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-\mathrm{j}\Omega t}\,\mathrm{d}t \iff x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\Omega)\mathrm{e}^{\mathrm{j}\Omega t}\,\mathrm{d}\Omega.$$

The representation of a time-domain periodic function of a continuous and real (time) variable is performed by the Fourier series. In the frequency-domain, the representation consists of a non-periodic function of a discrete and integer frequency variable, as following described:

$$X(k) = \frac{1}{T}\int_{0}^{T} x(t)\mathrm{e}^{-\mathrm{j}(2\pi/T)kt}\,\mathrm{d}t \iff x(t) = \sum_{k=-\infty}^{\infty} X(k)\mathrm{e}^{\mathrm{j}(2\pi/T)kt}.$$

1.01.3 Discrete-time signals and systems

A discrete-time signal is represented by a sequence of numbers and can be denoted as $x(n)$ with $n \in Z$, where Z is the set of integer numbers. The samples $x(n)$ might represent numerically the amplitude of a continuous-time signal sample at every T seconds or can be originated from a discrete information. In many applications, the sampling period T represents the time interval between two acquired samples of the continuous-time signal. However, in other situations it might represent the distance between two sensors of two pixels of an image, or the separation between two antennas, just to mention a few examples.

Discrete-time systems map input sequences into output sequences. A general representation of this mapping is

$$y(n) = \mathcal{H}\{x(n)\},$$

where $\mathcal{H}\{\cdot\}$ denotes the operation performed by the discrete-time system. Figure 1.2 depicts the input-to-output mapping of sequences. According to the properties of $\mathcal{H}\{\cdot\}$, the discrete-time system might be LTI. This way, it benefits from a number of analysis and design tools such as frequency-domain representations. However, there are many applications employing nonlinear, time-varying, and even non-causal systems [2,5,7,8,11,12,16].

If the system is linear and time-invariant, as will be described in Chapter 2, there are several tools to describe the mapping features of the system. Typically, a general description of discrete-time systems through a difference equation can be solved and analyzed by employing the z transform. Also a

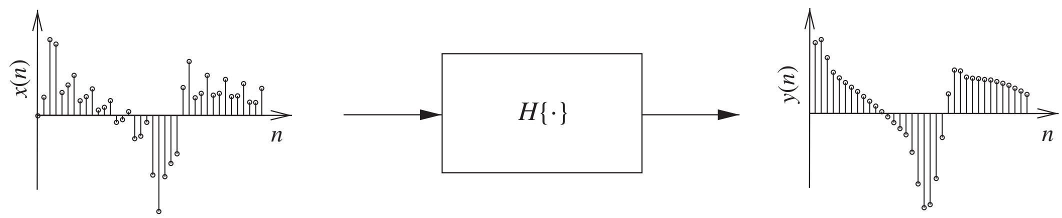


FIGURE 1.2

Discrete-time signal representation.

discrete-time LTI system is fully described by its impulse response, whereas a nonlinear system can not be fully characterized by its impulse response.

The z transform is the key tool to represent a time-domain non-periodic function of a discrete and integer variable through a frequency-domain non-periodic function of a continuous and complex frequency variable. That is,

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \quad \Longleftrightarrow \quad x(n) = \frac{1}{2\pi \mathrm{j}} \oint_C X(z) z^{n-1}\, \mathrm{d}z.$$

The discrete-time Fourier transform (DTFT) represents a time-domain non-periodic function of a discrete and integer variable by a periodic function of a continuous frequency variable in the frequency domain as follows:

$$X(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{n=-\infty}^{\infty} x(n) \mathrm{e}^{-\mathrm{j}\omega n} \quad \Longleftrightarrow \quad x(n) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\mathrm{e}^{\mathrm{j}\omega}) \mathrm{e}^{\mathrm{j}\omega n}\, \mathrm{d}\omega.$$

Finally, the discrete Fourier transform (DFT) is the right tool to represent a time-domain periodic function of a discrete and integer variable through a periodic function of a discrete and integer frequency variable in the frequency domain. The DFT is defined as

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

The DFT plays a key role in digital signal processing since it represents a finite-length sequence in the time domain through a finite-length sequence in the frequency domain. Since both domains utilize sequences, this feature makes the DFT a natural choice for time-frequency representation of information in a digital computer. A little mental exercise allows us to infer that the DFT is the perfect representation in the frequency domain of a finite-length sequence in the time domain, if we interpret the latter as a period of a periodic infinite-length sequence. In addition, by employing appropriate zero-padding in two finite-length sequences, the product of their DFTs represents the DFT of their linear convolution (see Chapter 3), turning the DFT a valuable tool for LTI system implementation. There is a plethora of applications for the DFT in signal processing and communications and some of them can be accessed in the references [2–13].

Often, whenever a DSP system is LTI, it can be described through a difference equation as follows:

$$\sum_{i=0}^{N} a_i y(n-i) + \sum_{l=0}^{M} b_l x(n-l) = 0.$$

The above description is suitable for implementation in digital computers. The difference equation represents a causal LTI system if its auxiliary conditions correspond to its initial conditions and those are zeros [2]. Assuming $a_0 = 1$, the above equation can be rewritten as

$$y(n) = -\sum_{i=1}^{N} a_i y(n-i) + \sum_{l=0}^{M} b_l x(n-l).$$

This class of system has infinite impulse response (i.e., $y(n) \neq 0$ when $n \to \infty$) and, as such, it is called IIR system or filter.

The nonrecursive system generates the output signal from past input samples, that is,

$$y(n) = \sum_{l=0}^{M} b_l x(n-l).$$

In this case, the resulting system has finite impulse response and is known as FIR systems or filters.

1.01.4 Random signals and stochastic processes

In most practical applications, we have to deal with signals that can not be described by a function of time, since their waveforms follow a random pattern [17,18]. Take, for example, the thermal noise inherent to any material. Despite the lack of exact knowledge of the signal values, it is possible to analyze and extract information the signals contain by employing the mathematical tools available to deal with random signals. Chapter 4 will present a compact and yet clarifying review of random signals and stochastic processes which are crucial to understand several concepts that will be discussed in the more advanced chapters.

The theory starts by defining a random variable as a mapping of the result of a random process experiment onto the set of numbers. A random variable X is a function that assigns a number x to every outcome of a random process experiment. An example is given in Figure 1.3(a), in which each outcome of a throw of a dice, numbers 1–6, corresponds to a determined value, x_1–x_6, respectively.

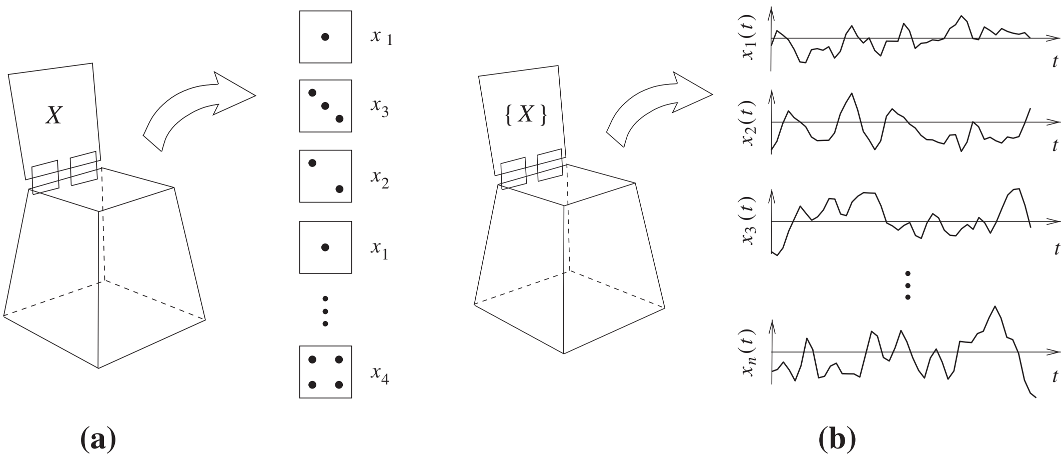


FIGURE 1.3

Examples of random variable and stochastic process. (a) Random variable. (b) Stochastic process.

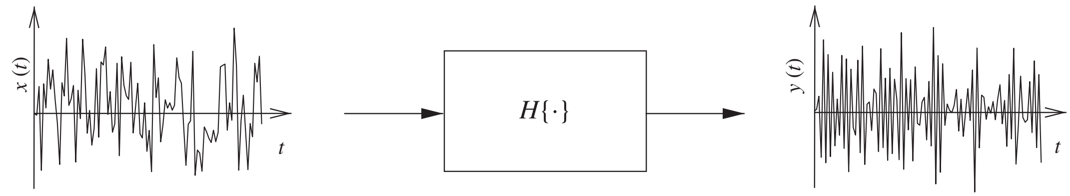


FIGURE 1.4

Filtering of a random signal.

The stochastic process is a rule to describe the time evolution of the random variable depending on the random process experiment, whereas the set of all experimental outcomes is its domain known as the ensemble. An example is given in Figure 1.3(b), in which, at any time instant t_0, the set formed by the output samples of the stochastic process $\{X\}$, $\{x_1(t_0), x_2(t_0), x_3(t_0), \ldots, x_n(t_0)\}$, represents a random variable. A single outcome $x_i(t)$ of $\{X\}$ is a random signal. Since random signals do not have a precise description of their waveforms, we have to characterize them either via measured statistics or through a probabilistic model. In general, the first- and second-order statistics (mean and variance, respectively) are sufficient for characterization of the stochastic process, particularly due to the fact that these statistics are suitable for measurements.

As an illustration, Figure 1.4 shows a random signal as an input of a highpass filter. We can observe that the output signal is still random, but clearly shows a faster changing behavior given that the low frequency contents of the input signal were attenuated by the highpass filter.

1.01.5 Sampling and quantization

A digital signal processing system whose signals originated from continuous-time sources includes several building blocks, namely: an analog-to-digital (A/D) converter; a digital signal processing (DSP) system; a digital-to-analog (D/A) converter; and a lowpass filter. Figure 1.5 illustrates a typical digital signal processing setup where:

- The A/D converter produces a set of samples in equally spaced time intervals, which may retain the information of the continuous-time signal in the case the latter is band limited. These samples are converted into a numeric representation, in order to be applied to the DSP system.
- The DSP performs the desired mapping between the input and output sequences.
- The D/A converter produces a set of equally spaced-in-time analog samples representing the DSP system output.
- The lowpass filter interpolates the analog samples to produce a smooth continuous-time signal.

Chapter 5 will discuss in detail the conditions which a continuous-time signal must satisfy so that its sampled version retains the information of the original signal, dictated by the sampling theorem. This theorem determines that a band limited continuous-time signal can be theoretically recovered from its sampled version by filtering the sequence with an analog filter with prescribed frequency response.

It is also important to mention that, while processing the signals in the digital domain, these are subject to quantization errors, such as: roundoff errors originated from internal arithmetic operations

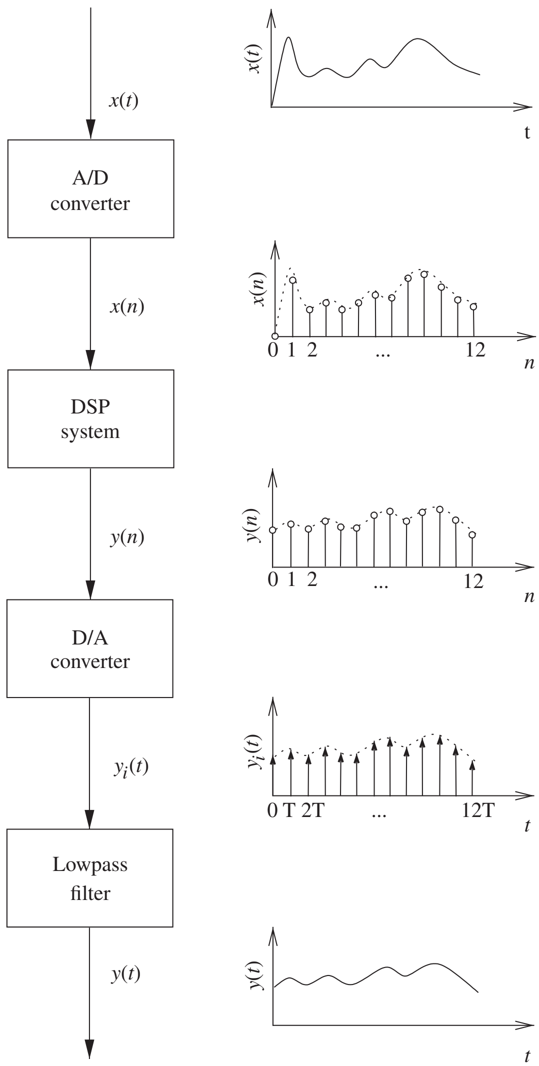


FIGURE 1.5

DSP system.

performed in the signals; deviations in the filter response due to finite wordlength representation of the multiplier coefficients inherent to the signal processing operation; and errors due to representation of the acquired continuous-time signals with a set of discrete levels.

The actual implementation of a DSP system might rely on general purpose digital machines, where the user writes a computer software to implement the DPS tasks. This strategy allows fast prototyping and testing. Other mean is to use special-purpose commercially available CPUs, known as Digital Signal

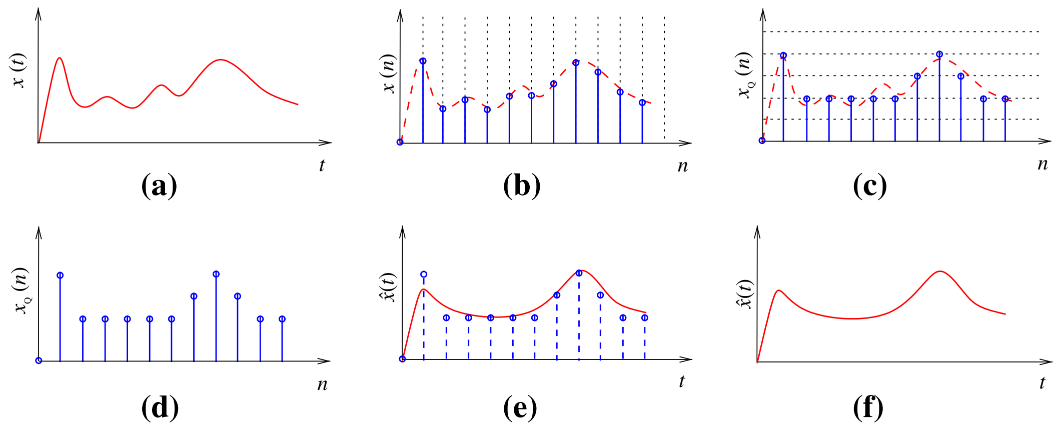


FIGURE 1.6

A digital signal generation. (a) Original continuous-time signal. (b) A/D converter: sampling. (c) A/D converter: quantization. (d) Digital signal. (e) D/A converter.(f) Recovered continuous-time signal.

Processors, which are capable of implementing sum of product operations in a very efficient manner. Yet another approach is to employ special purpose hardware tailored for the given application.

Figure 1.6 depicts a continuous-time signal, its digitized version, and its recovered continuous-time representation. We can notice, from Figures 1.6(a) to 1.6(f), the effects of a low sampling frequency and a small number of quantization steps on the A/D and D/A processes.

Figure 1.7 is a detailed example of the steps entailing an A/D and a D/A conversion, where in Figure 1.7(a) a continuous-time signal is depicted along with its equally spaced samples. These samples are then quantized as illustrates Figure 1.7(b). Assuming the quantized samples are converted into an analog signal through a zero-order hold, the output of the D/A converter becomes as illustrated in Figure 1.7(c). The sampled-and-held continuous-time signal is lowpass filtered in order to recover the original continuous-time signal. As can be observed in Figure 1.7(d), the recovered signal resembles the original where the difference originates from the quantization effects and the nonideal lowpass filter at the D/A converter output.

1.01.6 FIR and IIR filter design

The general form of an FIR transfer function is given by

$$H(z) = \sum_{l=0}^{M} b_l z^{-l} = H_0 z^{-M} \prod_{l=0}^{M} (z - z_l).$$

Since all the poles of the FIR transfer function are placed at $z = 0$, it is always stable. As a result, the transfer function above will always represent a stable filter. The main feature of the FIR filters is

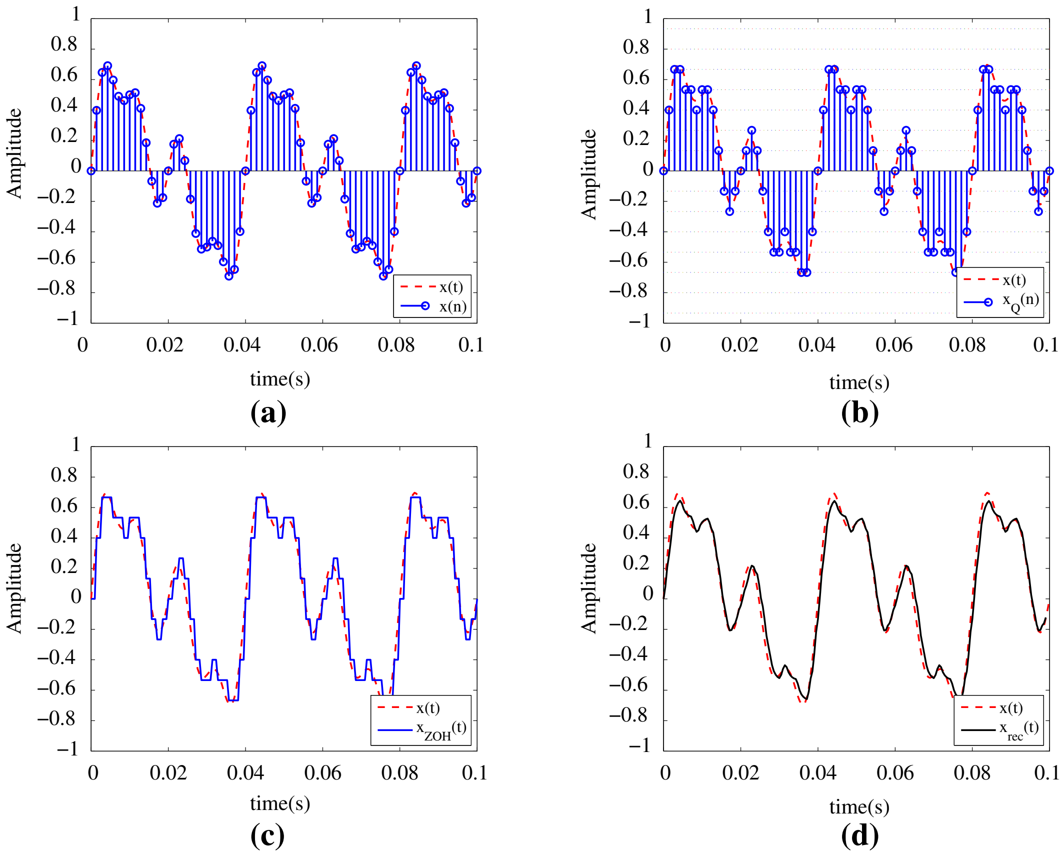


FIGURE 1.7

A real digital signal generation. (a) A/D converter: sampling. (b) A/D converter: quantization. (c) D/A converter: zero-order hold. (d) D/A converter: lowpass filter.

the possibility of designing structurally induced linear-phase filters. On the other hand, an FIR filter requires a higher number of multiplications, additions and storage elements when compared to their IIR counterparts in order to satisfy a prescribed specification for the magnitude response.

There are many methods to design an FIR filter [19], ranging from the simple ones, such as the window and frequency sampling methods, to more efficient and sophisticated, that rely on some kind of optimization method [20–22]. The simple methods lead to solutions which are far from optimum in most practical applications, in the sense that they either require higher order or cannot satisfy prescribed specifications. A popular optimization solution is the minimax method based on the Remez exchange algorithm, which leads to transfer functions with minimum order to satisfy prescribed specifications.

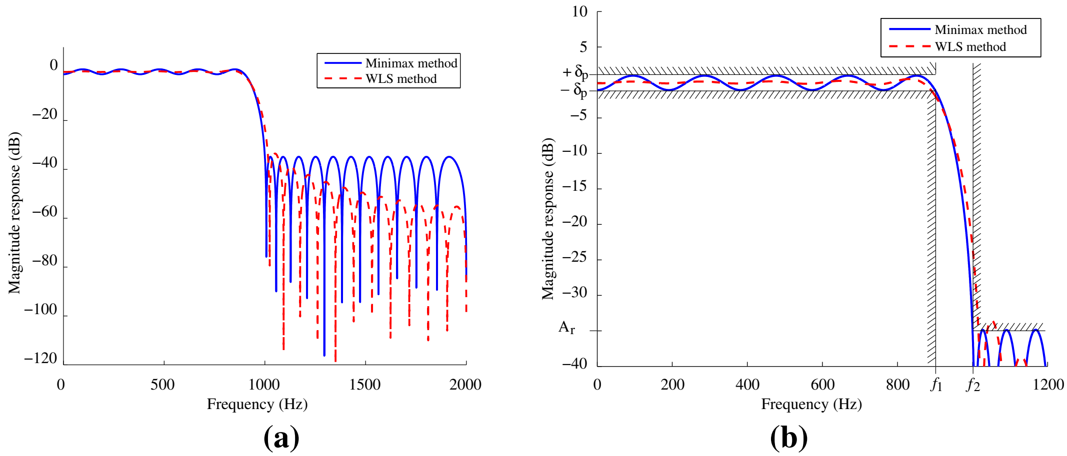


FIGURE 1.8

Typical magnitude responses of FIR filters: Minimax and WLS. (a) Magnitude response. (b) Specifications for a lowpass filter using minimax design.

Nowadays, with the growing computational power, it is possible to design quite flexible FIR filters by utilizing weighted least squares (WLS) solutions. These filters are particularly suitable for multirate systems, since the resulting transfer functions can have decreasing energy in their stopband [2]. The WLS method enables the design of filters satisfying prescribed specifications, such as the maximum deviation in the passband and in part of the stopband, while minimizing the energy in a range of frequencies in the stopband. As can be observed in Figure 1.8, for a given filter order, the minimax design leads to lower maximum stopband attenuation, whereas the WLS solution leads to lower stopband energy. Note there are solutions in between where some maximum values of stopband ripples can be minimized while the energy of the remaining ones is also minimized. It is possible to observe that the WLS solution does not satisfy the prescribed specifications given that it has the same filter order as the minimax solution.

The typical transfer function of an IIR filter is given by

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{l=0}^{M} b_l z^{-l}}{1 + \sum_{i=1}^{N} a_i z^{-i}},$$

which can be rewritten in a form explicitly showing the poles positions as

$$H(z) = H_0 \frac{\prod_{l=0}^{M} (1 - z^{-1} z_l)}{\prod_{i=0}^{N} (1 - z^{-1} p_i)} = H_0 z^{N-M} \frac{\prod_{l=0}^{M} (z - z_l)}{\prod_{i=0}^{N} (z - p_i)}.$$

IIR filters are usually designed by using well-established analog filter approximations, such as Butterworth, Chebyshev, and elliptic methods (Figure 1.9) The prototype analog transfer function is then

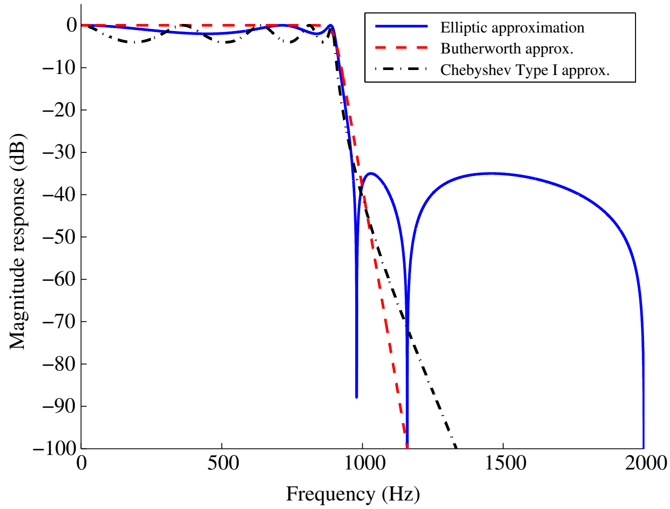


FIGURE 1.9

Typical magnitude response of an IIR filter.

transformed into a digital transfer function by using an adequate transformation method. The most widely used transformation methods are bilinear and the impulse invariance methods [2].

1.01.7 Digital filter structures and implementations

From a closer examination of the FIR and IIR transfer functions, one can infer that they can be realized with three basic operators: adder, multiplier and delay (represented by z^{-1}).

For example, Figure 1.10 shows a linear-phase FIR filter realization of odd order. Linear-phase filters have symmetric or anti-symmetric impulse response and the symmetry should be exploited in order to minimize the number of multipliers in the filter implementation, as illustrates Figure 1.10.

Figure 1.11 depicts a possible direct-form realization of an IIR transfer function. This realization requires the minimum number of multiplications, adders and delays to implement the desired IIR transfer function.

There are many alternative structures to implement the FIR and IIR filters. For example, IIR transfer functions can be implemented as a product or as a summation of lower order IIR structures by starting with their description as:

$$H(z) = \prod_{k=1}^{m} \frac{\gamma_{0k} + \gamma_{1k} z^{-1} + \gamma_{2k} z^{-2}}{1 + m_{1k} z^{-1} + m_{2k} z^{-2}}$$
$$= h_0^p + \sum_{k=1}^{m} \frac{\gamma_{1k}^p z + \gamma_{2k}^p}{z^2 + m_{1k} z + m_{2k}},$$

respectively. The right choice for a filter realization must take into consideration some issues, such as: modularity, quantization effects, design simplicity, power consumption, etc. [2]. Chapter 6 will address

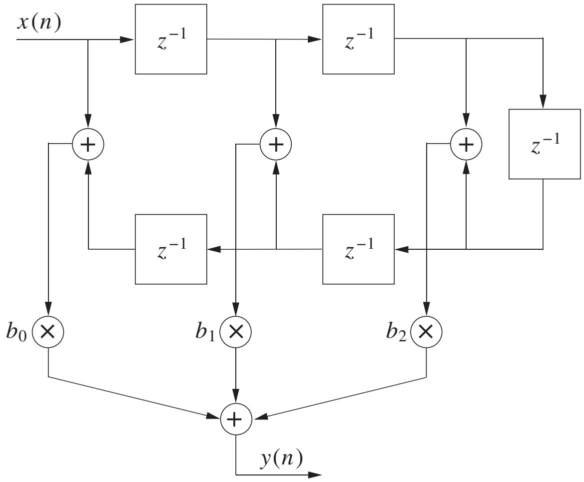


FIGURE 1.10

Odd-order linear-phase FIR filter structure with $M = 5$.

advanced methods for FIR and IIR filter realization and implementation, where elegant and creative solutions are introduced in a clear manner. This chapter will also discuss strategies to implement digital filters efficiently.

1.01.8 Multirate signal processing

In digital signal processing it is easy to change the sampling rate of the underlying sequences by a ratio of integer values. This feature is highly desirable whenever we want to merge information of systems employing distinct sampling rates. In these cases, those rates should be made compatible [6,9,10].

Systems that internally employ multiple sampling rates are collectively referred to as multirate systems. In most cases, these systems are time-varying or, at most, periodically time-invariant.

The basic operations of the multirate systems are the decimation and interpolation. The right composition of these operations allows arbitrary rational sampling rate changes. If this sampling rate change is arbitrary and non-rational, the only solution is to recover the bandlimited continuous-time signal $x(t)$ from its samples $x(m)$, and then re-sample it with a different sampling rate, thus generating a distinct discrete-time signal $\overline{x}(n)$.

If we assume that a discrete-time signal signal $x(m)$ was generated from an continuous-time signal $y(t)$ with sampling period T_1, so that $x(m) = y(mT_1)$, with $m = \ldots, 0, 1, 2 \ldots$, the sampling theorem requires that $y(t)$ should be bandlimited to the range $[-\frac{\pi}{T_1}, \frac{\pi}{T_1}]$. The sampled continuous-time signal is given by:

$$y'(t) = \sum_{m=-\infty}^{\infty} x(m)\delta(t - mT_1).$$

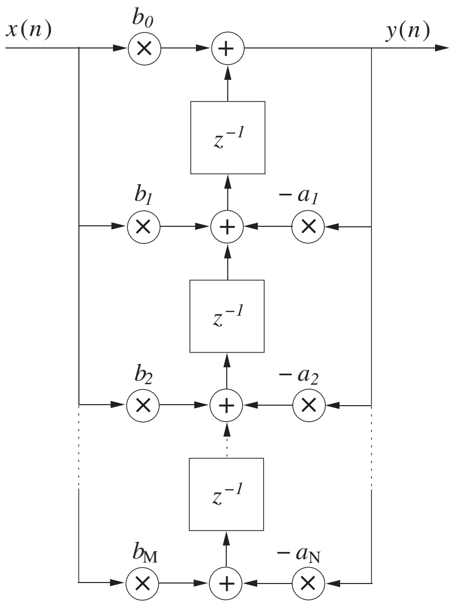


FIGURE 1.11

Direct-Form IIR filter structure, for M = N.

The spectrum of the sampled signal is periodic with period $\frac{2\pi}{T_1}$. The original continuous-time signal $y(t)$ can be recovered from $y'(t)$ through an ideal lowpass filter. This interpolation filter, whose impulse response is denoted as $h(t)$, has ideal frequency response $H(\mathrm{e}^{\mathrm{j}\omega})$ as follows:

$$H(\mathrm{e}^{\mathrm{j}\omega}) = \begin{cases} 1, & \omega \in [-\frac{\pi}{T_1}, \frac{\pi}{T_1}] \\ 0, & \text{otherwise,} \end{cases}$$

so that

$$y(t) = y'(t) * h(t) = \frac{1}{T_1} \sum_{m=-\infty}^{\infty} x(m)\mathrm{sinc}\frac{\pi}{T_1}(t - mT_1).$$

By re-sampling $y(t)$ with period T_2, we obtain the discrete-time signal in the new desired sampling rate as follows:

$$\overline{x}(n) = \frac{1}{T_1} \sum_{m=-\infty}^{\infty} x(m)\mathrm{sinc}\frac{\pi}{T_1}(nT_2 - mT_1),$$

where $\overline{x}(n) = y(nT_2)$, with $n = \ldots, 0, 1, 2 \ldots$.

In this general equation governing sampling rate changes, there is no restriction on the values of T_1 and T_2. However, if $T_2 > T_1$ and aliasing is to be avoided, the interpolation filter must have a zero gain for $\omega \notin [-\frac{\pi}{T_2}, \frac{\pi}{T_2}]$.

In the case the sampling rate change corresponds to a ratio of integer numbers, all we need is simple decimation and interpolation operations. The decimation operation, or down-sampling, consists of retaining every Mth sample of a given sequence $x(m)$. Since we are disposing samples from the original sequence, either the signal is sufficiently limited in band or the signal has to be filtered by a lowpass filter so that the decimated signal retains the useful information of the original signal. Decimation is a time-varying operation, since, if $x(m)$ is shifted by m_0, the output signal will, in general, differ from the unshifted output shifted by m_0. Indeed, decimation is a periodically time-invariant operation, which consist of an extra degree of freedom with respect to the traditional time-invariant systems.

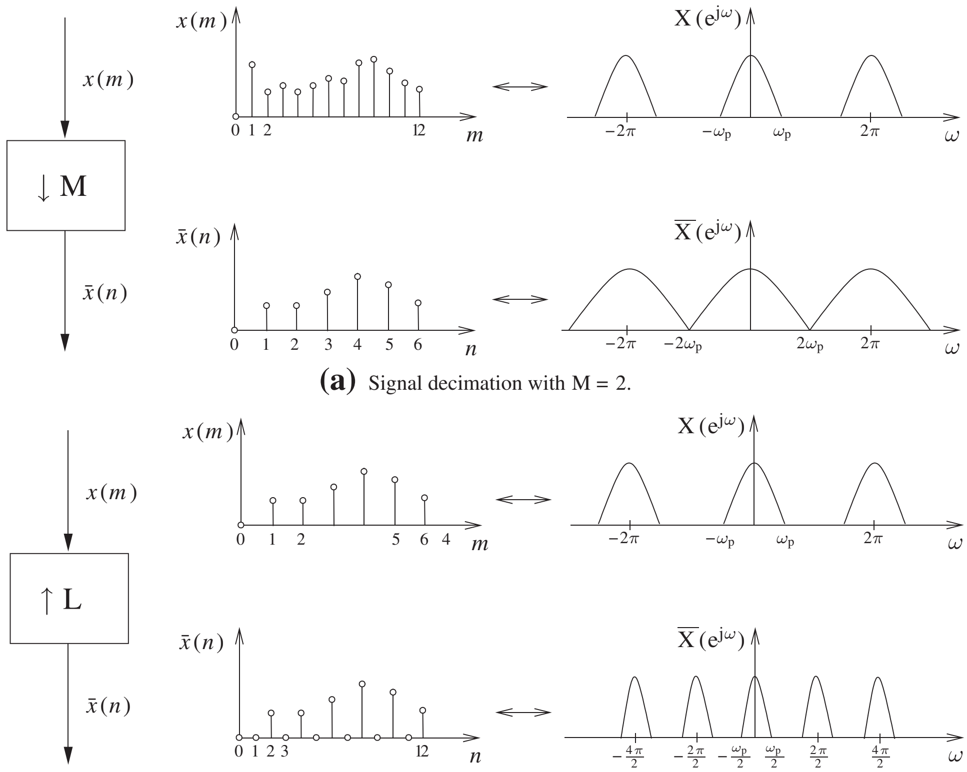
The interpolation, or up-sampling, of a discrete-time signal $x(m)$ by a factor of L consists of inserting $L - 1$ zeros between each of its samples. In the frequency domain, the spectrum of the up-sampled signal is periodically repeated. Given that the spectrum of the original discrete-time signal is periodic with period 2π, the interpolated signal will have period $\frac{2\pi}{L}$. Therefore, in order to obtain a smooth interpolated version of $x(m)$, the spectrum of the interpolated signal must have the same shape of the spectrum of $x(m)$. This can be obtained by filtering out the repetitions of the spectra beyond $[-\frac{\pi}{L}, \frac{\pi}{L}]$. Thus, the up-sampling operation is generally followed by a lowpass filter. The interpolation is only periodically time-invariant and does not entail any loss of information.

Figure 1.12 illustrates discrete-time signal decimation and interpolation operations. As can be observed in Figure 1.12(a), the decimation factor of two widens the spectrum of the sampled signal in comparison to the new sampling rate. Figure 1.12(b) depicts the effect of increasing the sampling rate of a given sequence by two, where it can be seen that the frequency contents of the original signal repeats twice as as often and appears narrower in the new sampling rate.

Chapter 7 will further discuss the theory and practice of multirate signal processing. This chapter will show how sophisticated real-life signal processing problems can be addressed starting from the basic theory. In particular, the authors address the design of flexible communications systems incorporating several advanced signal processing tools [4,23–25].

1.01.9 Filter banks and wavelets

In many applications it is desirable to decompose a wideband discrete-time signal into several non-overlapping narrowband subsignals in order to be transmitted, quantized, or stored. In this case, each narrow-band channel can have its sampling rate reduced, since the subband signals have lower bandwidth than their originator signal. The signal processing tool that performs these tasks is the analysis filter bank. The analysis filters, represented by the transfer functions $H_i(z)$, for $i = 0, 1, \ldots, M - 1$, consist of a lowpass filter $H_0(z)$, bandpass filters $H_i(z)$, for $i = 1, 2, \ldots, M-2$, and a highpass filter $H_{M-1}(z)$. Ideally, these filters have non-overlapping passbands, whereas their spectrum combination should cover the overall spectrum of the input signal. The outputs of the analysis filters, denoted as $x_i(n)$, for $i = 0, 1, \ldots, M-1$, have together M times the number of samples of the original signal $x(n)$. However, since each subband signal occupies a spectrum band M times narrower than the input signal, they can be decimated so the number of samples in the subbands does not increase. Indeed, if the input signal is uniformly split into subbands, we can decimate each $x_i(n)$ by a factor of L smaller or equal to M without generating unrecoverable aliasing effects. Figure 1.13 shows the general configuration of a maximally (or critically) decimated analysis filter, where the decimation factor is equal to the number of subbands, i.e., $L = M$.



(a) Signal decimation with M = 2.

(b) Signal interpolation with L = 2.

FIGURE 1.12

Signal decimation and interpolation. (a) Signal decimation with $M = 2$. (b) Signal interpolation with $L = 2$.

Whenever the input signal is split in subbands and decimated, if $L \leq M$, it is always possible to recover the input signal by properly designing the analysis filters in conjunction with the synthesis filters $G_i(z)$, for $i = 0, 1, \ldots, M-1$. The synthesis filters are placed after interpolators and they have the task of smoothing the up-sampled signals. Figure 1.14 illustrates the general configuration of the synthesis filter bank. A key feature of the synthesis filter bank is to cancel out or reduce the aliasing effects.

If the signals in the subbands are not modified, the filter bank output $y(n)$ can be a delayed copy signal of $x(n)$. The cascade connection of the analysis and synthesis filter banks satisfying this condition is called a perfect reconstruction filter bank.

The cascade of an M-channel synthesis filter bank with an M-channel analysis filter bank gives rise to a transmultiplex system. The transmultiplex model is widely used in communications to model multiuser, multicarrier, and multiple access systems [4,23,24]. Chapter 7 will utilize multirate signal processing and transmultiplexers to discuss a design procedure to be applied in cognitive radio.

Chapter 8 will present several methods for designing the analysis filters $H_i(z)$ and the synthesis filters $G_i(z)$, such that perfect reconstruction, as well as other features, are met. This chapter will also discuss

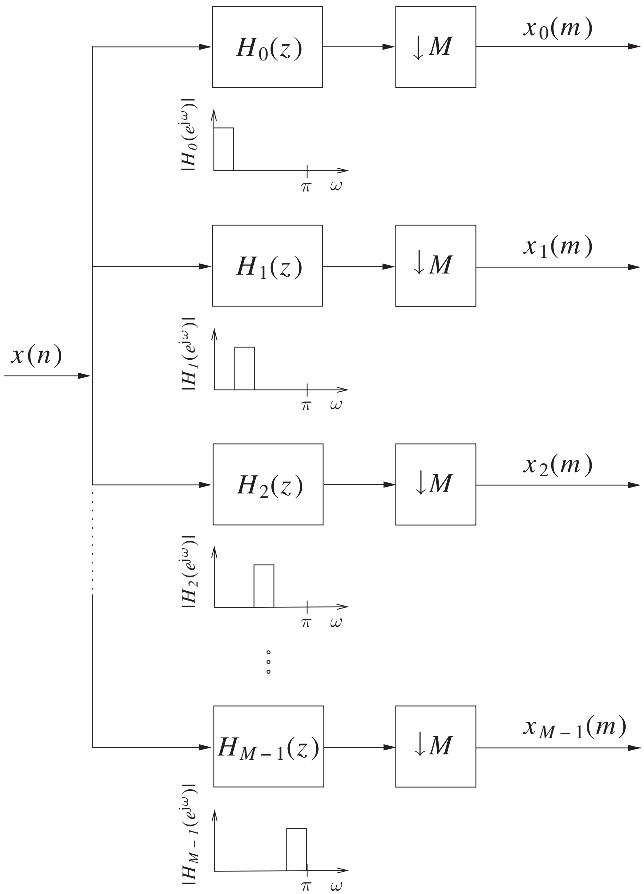


FIGURE 1.13

Analysis Filter Bank.

the design of filter banks with the subbands divided in octave bands, giving rise to the discrete-time wavelet series.

Figure 1.15 shows a four-band filter bank design and its effect in a composed signal.

Figure 1.16 illustrates how a four-band synthesis filter bank performs the recomposition of the input signal starting from the decimated subband signals.

1.01.10 Discrete multiscale and transforms

Discrete-time signals and systems can be characterized in the frequency domain by their Fourier transforms. One of the main advantages of discrete-time signals is that they can be processed and represented

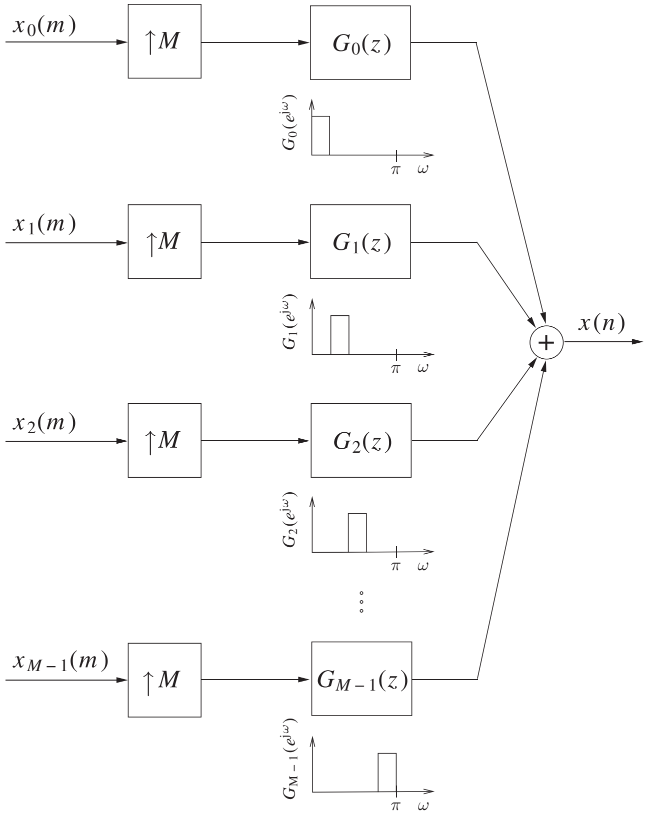


FIGURE 1.14

Synthesis Filter Bank.

in digital computers. However, when we examine the definition of the discrete-time Fourier transform,

$$X(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{n=-\infty}^{\infty} x(n)\mathrm{e}^{-\mathrm{j}\omega n},$$

we notice that such a characterization in the frequency domain depends on the continuous variable ω. This implies that the Fourier transform, as it is, is not suitable for the processing of discrete-time signals in digital computers. We need a transform depending on a discrete-frequency variable that, if possible, preserves the handy interpretations of the Fourier-based tools, which retains important information. This can be obtained from the Fourier transform itself in a very simple way, by sampling uniformly the continuous-frequency variable ω as long as the input sequence has finite length, otherwise the time-domain signal will be affected by aliasing. Using this strategy it is possible to obtain a mapping of a signal depending on a discrete-time variable n to a transform depending on a discrete-frequency

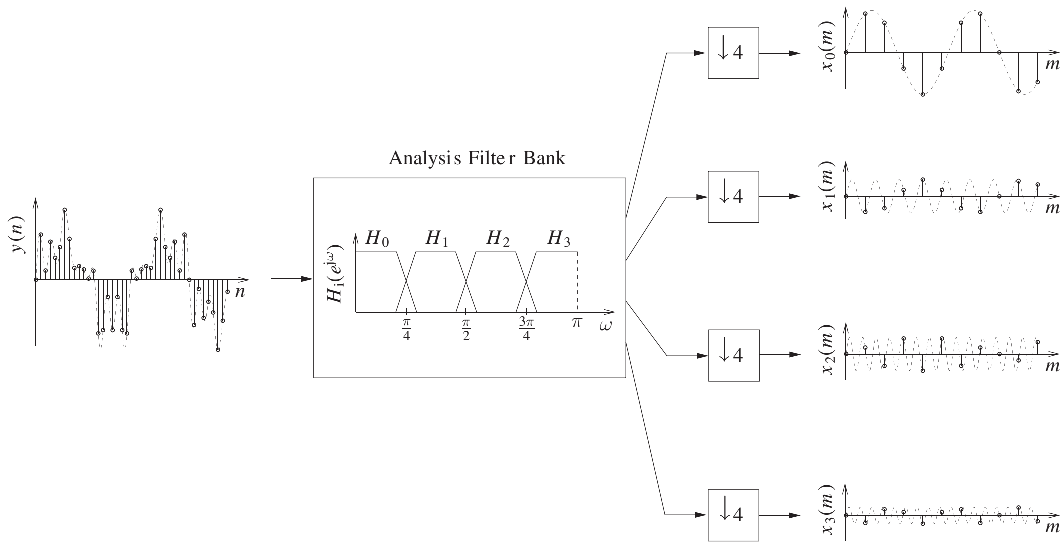


FIGURE 1.15

Analysis filter bank design.

variable k, leading to another interpretation for the DFT. Unfortunately, the DFT has its limitations in representing more general class of signals as will be following discussed.

The wavelet transform of a function belonging to $\mathcal{L}^2\{\mathbb{R}\}$, the space of the square integrable functions, is the function decomposition in a basis composed by expansions, compressions, and translations of a single mother function $\psi(t)$, called wavelet.

The wavelet transform can be defined as

$$x(t) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} c_{m,n} \overline{\psi}_{m,n}(t)$$
$$c_{m,n} = \int_{-\infty}^{\infty} x(t) \psi^*_{m,n}(t)\, \mathrm{d}t,$$

where

$$\psi_{m,n}(t) = 2^{-m/2} \psi(2^{-m}t - n)$$
$$\overline{\psi}_{m,n}(t) = 2^{-m/2} \overline{\psi}(2^{-m}t - n).$$

This pair of equations defines a biorthogonal wavelet transform which are characterized by two wavelets: the analysis wavelet, $\psi(t)$, and the synthesis wavelet, $\overline{\psi}(t)$. Any function $x(t) \in \mathcal{L}^2\{\mathbb{R}\}$ can be

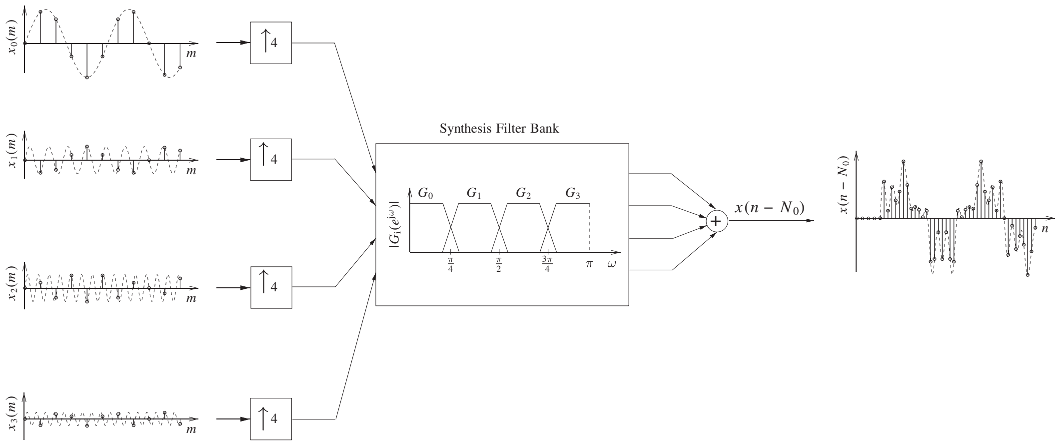


FIGURE 1.16

Synthesis filter bank design.

decomposed as a linear combination of contractions, expansions, and translations of the synthesis wavelet $\overline{\psi}(t)$. The weights of the expansion can be computed via the inner product of $x(t)$ with expansions, contractions, and translations of the analysis wavelet $\psi(t)$.

Functions $\psi_{m,n}(t)$ do not comprise an orthogonal set, so neither do the functions $\overline{\psi}_{m,n}(t)$. However, functions $\psi_{m,n}(t)$ are orthogonal to $\overline{\psi}_{m,n}(t)$ so that

$$\langle \psi_{m,n}(t), \overline{\psi}_{k,l}(t) \rangle = \delta(m-k)\delta(n-l).$$

A byproduct of the development of the wavelet transforms is the so called wavelet series, allowing the representation of signals in the time and frequency domains through sequences. The cascade of two-band filter banks can produce many alternative maximally decimated decompositions. Of the particular interest is the hierarchical decomposition achieving an octave-band decomposition, in which only the lowpass band is further decomposed. This configuration gives rise to the widely used wavelets series.

As will be discussed in Chapter 10, there are many transform-based tools in signal processing theory available to meet the requirement of different applications. The classical Fourier based transforms have been used for centuries, but their efficiency in dealing with nonstationary signals is limited. With wavelet series, for example, it is possible to obtain good resolution in time and frequency domains, unlike the Fourier-based analysis. Chapter 9 will include a myriad of transform solutions tailored for distinct practical situations.

Figure 1.17 illustrates a wavelet series representation of a composed signal, where the input signal is decomposed by an octave-band filter bank representing a wavelet series.

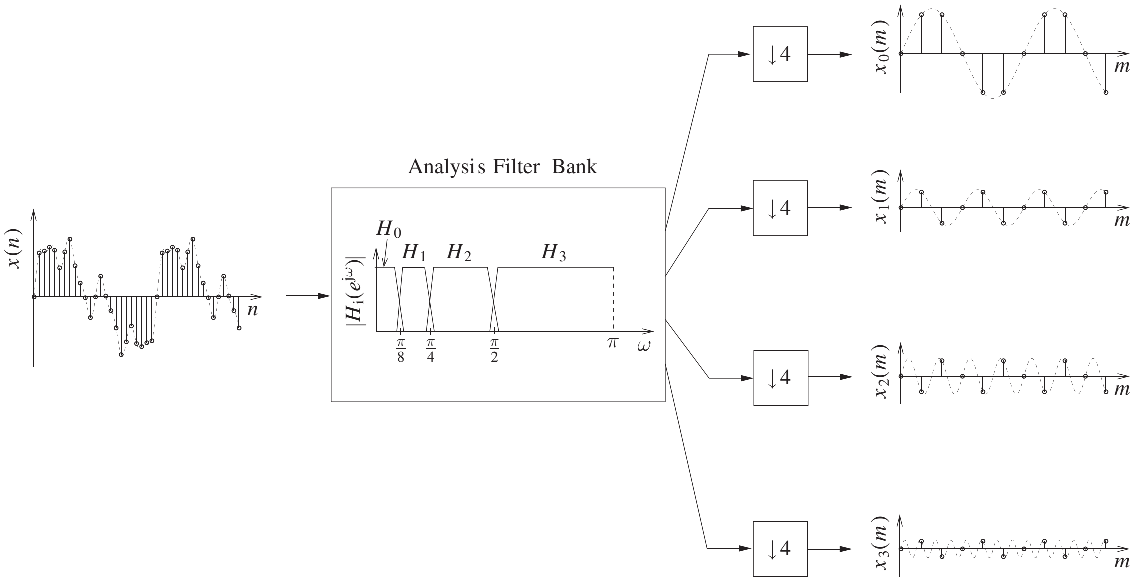


FIGURE 1.17

Wavelet series of a composed sinusoidal signal.

1.01.11 Frames

Frames consist of a set of vectors that enables a stable representation of a signal starting from an analysis operation whereby a set of coefficients is generated from the signal projection with the frame vectors, and a synthesis operation where the synthesis vectors are linearly combined to approximate the original signal. Chapter 10 will introduce the idea of frames by discussing the concepts of overcomplete representations, frames and their duals, frame operators, inverse frames and frame bounds. In particular, it will show how to implement signal analysis and synthesis employing frames generated from a fixed prototype signal by using translations, modulations and dilations; and analyzes frames of translates, Gabor frames and wavelet frames. The frames representation is usually redundant by requiring more coefficients than the minimum, where redundancy is measured by frame bounds [13]. The end result of employing the framework of frames is a set of signal processing tools such as: the Gaborgram; time-frequency analysis; the matching pursuit algorithm, etc.

Figure 1.18 shows the representation of a composed signal into frames. As can be observed the representation of $x(n)$ by $x_1(n)$ and $x_0(n)$ is redundant, since $x_1(n)$ has the same rate as the input signal $x(n)$, whereas $x_0(n)$ has half rate.

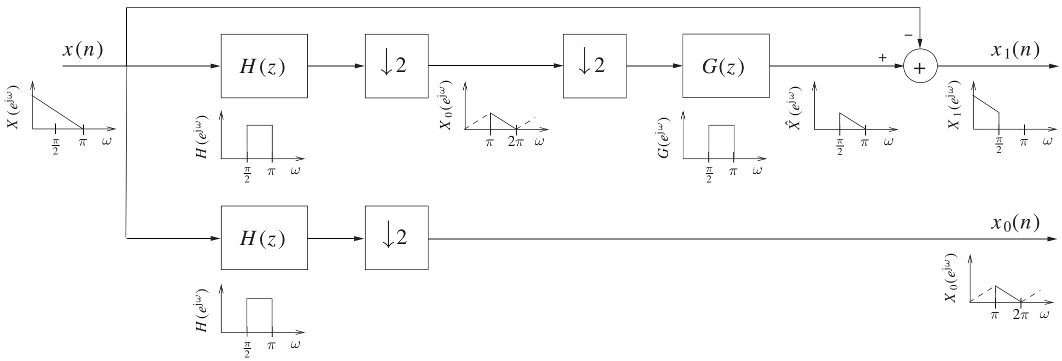


FIGURE 1.18

Frame representation.

1.01.12 Parameter estimation

In many signal processing applications it is crucial to estimate the power spectral density (PSD) of a given discrete-time signal. Examples of the use of PSD are vocal-track modeling, radar systems, antenna array, sonar systems, synthesis of speech and music, just to name a few. Usually, the first step in PSD estimation consists of estimating the autocorrelation function associated with the given data, followed by a Fourier transform to obtain the desired spectral description of the process.

In the literature we can find a large number of sources dealing with algorithms for performing spectral estimation. The alternative solutions differ in their computational complexity, precision, frequency resolution, or other statistical aspects. In general, the spectral estimation methods can be classified as nonparametric or parametric methods. The nonparametric methods do not prescribe any particular structure for the given data, whereas parametric schemes assume that the provided process follows the pattern of a prescribed model characterized by a specific set of parameters [14].

Since the parametric approaches are simpler and more accurate, they will be emphasized in Chapter 11. However, it should be mentioned that the parametric methods rely on some (a priori) information regarding the problem at hand.

Figure 1.19 shows a noisy signal record and the estimated PSDs utilizing distinct methods. As can be observed, some of the resulting PSDs expose the presence of a sinusoid, not clearly observed in the time-domain signal.

1.01.13 Adaptive filtering

In many practical situations we can extract information from a signal by comparing it to another signal or a reference. For the cases where the specifications are neither known nor time-invariant, an adaptive filter is the natural solution. The coefficients of these filters are data driven and the resulting processing

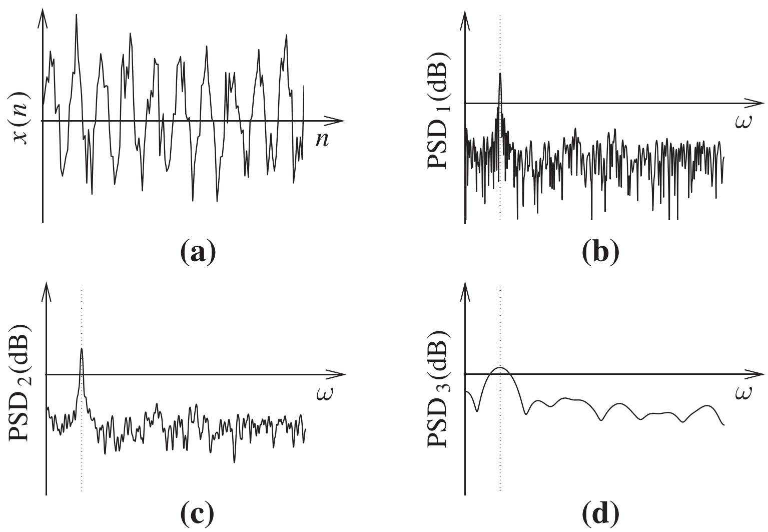


FIGURE 1.19

Spectral estimation. (a) Time-domain signal. (b) Estimated PSD1.(C) Estimated PSD2. (d) Estimated PSD3.

task does not meet the homogeneity and additive conditions of linear filtering. An adaptive filter is also time-varying since its parameters are continually changing in order to minimize a cost function.

Adaptive filters are self-designing and perform the approximation step on-line. Starting from an initial condition, the adaptive filter searches an optimal solution in an iterative format. Since adaptive filters are nonlinear filters, the analysis of their performance behavior requires the solution of time-domain equations usually involving the second-order statistics of their internal and external quantities. Although the analysis of an adaptive filter is more complicated than for fixed filters, it is a simple solution to deal with unknown stationary and nonstationary environments. On the other hand, since the adaptive filters are self-designing filters, their design can be considered less involved than those of standard fixed digital filters [3]. An important topic related to adaptive filtering is neural networks and machine learning [26], not directly covered in this book.

Figure 1.20 depicts the general setup of an adaptive-filtering environment, where, for a given iteration n, an input signal $x(n)$ excites the adaptive filter which will produce an output signal $y(n)$ which is then compared with a reference signal $d(n)$. From this comparison, an error signal $e(n)$ is formed according to $d(n) - y(n)$. The error signal is the argument of a cost (or objective) function whose minimization is achieved by properly adapting the adaptive filter coefficients. In Figure 1.21, it is shown a signal enhancement where it is recognized in the error signal that the input signal contained a sinusoidal component. In Figure 1.21 (a) it is difficult to observe the presence of a sinusoidal signal whereas the error signal clearly shows a nearly sinusoidal behavior.

Chapter 12 will present a comprehensive overview related to adaptive signal processing that unifies several concepts that go beyond the concepts covered in textbooks such as [3,27,28].

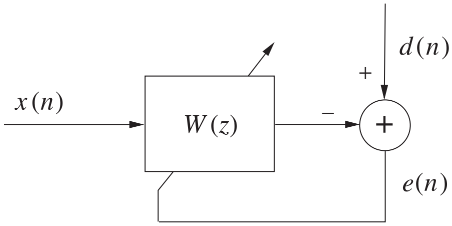


FIGURE 1.20

General adaptive filtering configuration.

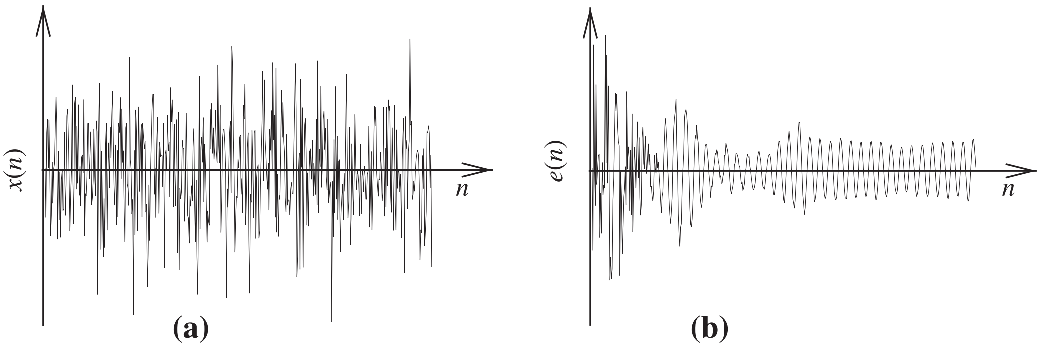


FIGURE 1.21

An adaptive filtering application. (a) Input signal. (b) Error signal.

1.01.14 Closing comments

The theory of signal processing has been developing for over four decades at a fast pace. The following chapters will cover many of the classical tools widely employed in applications that we perceive in our daily life through the use of mobile phones, media players, medical equipments, transportation and so on. These chapters will also cover recent advances and describe many new tools which can potentially be utilized in new applications, as well as can be further investigated. The authors of the chapters are frequent and important contributors to the field of signal processing theory. The goal of each chapter is to provide an overview and a brief tutorial of important topic pertaining to the signal processing theory, including key references for further studies. The present chapter attempted to describe the context in which each family of tools is employed.

The rest of this book is organized as follows:

- Chapter 2 covers the basic concepts of Continuous-Time Signals and Systems highlighting the main tools that can be employed to analyze and design such systems. Several examples are included to illustrate the use of the tools in a full continuous-time environment. The content of this chapter is also essential to make the connection between the continuous-time and the discrete-time systems.

- Chapter 3 addresses Discrete-Time Signals and Systems emphasizing the basic analysis tools that will be crucial to understand some more advanced techniques presented in the forthcoming chapters. This chapter shows how the powerful state-space representation of discrete-time systems can be employed as an analysis and design tool.
- Chapter 4 on Random Signals and Stochastic Processes describes in a comprehensive way the basic concepts required to deal with random signals. It starts with the concept of probability useful to model chance experiments, whose outcomes give rise to the random variable. The time-domain signals representing non-static outcomes are known as stochastic processes where their full definition is provided. The chapter describes the tools to model the interactions between random signals and linear-time invariant systems.
- Chapter 5 covers Sampling and Quantization where the basic concepts of properly sampling a continuous-time signal and its representation as a sequence of discrete-valued samples are addressed in detail. The chapter discusses a wide range of topics rarely found in a single textbook such as: uniform sampling and reconstruction of deterministic signals; the extension to sampling and reconstruction of stochastic processes; time-interleaved ADCs for high-speed A/D conversion. In addition, topics related to the correction to analog channel mismatches, principles of quantization, and oversampled ADCs and DACs are briey discussed. The chapter also includes the more advanced method for discrete-time modeling of mixed-signal systems employed in $\Sigma\Delta$-modulator-based ADCs.
- Chapter 6 takes us to a tour on design of FIR and IIR transfer functions of fixed filters satisfying prescribed specifications, explaining their basic realizations and exploring more sophisticate realizations. For IIR filters, the chapter introduces the concept of wave digital filter in a concise and clear manner, providing the motivation for its conception from the analog filter structures. The resulting IIR filter realizations keep the low sensitivity properties of their analog originators. For FIR filter structures, the chapter covers the frequency masking approach that exploits redundancy in the impulse response of typical FIR filters with high selectivity, aiming at reducing the computational complexity. The chapter also explains in detail how to implement digital filters efficiently in specific hardware by properly scheduling the arithmetic operations.
- Chapter 7 on Multirate Signal Processing for Software Radio Architecture applies several concepts presented in the previous chapters as tools to develop and conceive the implementation of radio defined by software, a subject of great interest in modern communications systems. Software defined radio entails implementing radio functionality into software, resulting in flexible radio systems whose main objective is to provide multi-service, multi-standard, multi-band features, all reconfigurable by software. From the signal processing perspective, several basic concepts of multirate systems such as, interpolation, decimation, polyphase decomposition, and transmultiplex play a central role. The chapter reviews the required concepts of signal processing theory and propose a software based radio architecture.
- Chapter 8 on Modern Transform Design for Practical Audio/Image/Video Coding Applications addresses the design of several applications-oriented transform methods. Most signal processing textbooks present the classical transforms requiring large amount of multiplications, however the demand for low-power multimedia platforms requires the development of computationally efficient

transforms for coding applications. This chapter has the unique feature of presenting systematic procedures to design these transforms while illustrating their practical use in standard codecs.

- Chapter 9 entitled Discrete Multi-Scale Transforms in Signal Processing presents a deeper high level exposition of the theoretical frameworks of the classical wavelets and the anisotropic wavelets. The aim is to enable the reader with enough knowledge of the wavelet-based tools in order to exploit their potential for applications in information sciences even further. Indeed the material covered in this chapter is unique in the sense that every discrete multi-scale transform is linked to a potential application.
- Chapter 10 on Frames discusses the use of overcomplete signal representations employing frames and their duals, frame operators, inverse frames, and frame bounds. The signal analysis and synthesis using frame representation is also addressed in detail. In particular, the chapter emphasis is on frames generated from a fixed prototype signal by using translations, modulations and dilations, and analyzes frames of translates, Gabor frames and wavelet frames. The chapter also presents signal analysis techniques based on the Gaborgram and time-frequency analysis using frames and the matching pursuit algorithm.
- Chapter 11 on Parametric Estimation exploits the key engineering concept related to modeling signals and systems, so that the model provides understanding of the underlying phenomena and possibly their control. This chapter emphasizes the parametric models leading to simple estimation of the parameters while exposing the main characteristics of the signals or systems under study. By employing both statistical and deterministic formulations, the chapter explains how to generate auto-regressive and moving average models which are then applied to solve problems related to spectrum estimation, prediction, and filtering.
- Chapter 12 on Adaptive Filtering presents a comprehensive description of the current trends as well as some open problems in this area. The chapter discusses the basic building blocks utilized in adaptive filtering setup as well as its typical applications. Then, the chapter discusses what are the optimal solutions following by the derivation of the main algorithms. The authors confront for the first time two standard approaches to access the performance of the adaptive filtering algorithms. The chapter closes by discussing some current research topics and open problems for further investigation.

References

[1] A. Antoniou, On the Roots of Digital Signal Processing, IEEE Circuits Syst. Mag. 7 (1) (2007) 8–18; A. Antoniou, IEEE Circuits Syst. Mag. 7 (4) (2007) 8–19.

[2] P.S.R. Diniz, E.A. Silva, S.L. Netto, Digital Signal Processing: System Analysis and Design, second ed., Cambridge University Press, Cambridge, UK, 2010.

[3] P.S.R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, fourth ed., Springer, New York, NY, 2013.

[4] P.S.R. Diniz, W.A. Martins, M.V.S. Lima, Block Transceivers: OFDM and Beyond, Morgan & Claypool Publishers, Ft, Collins, CO., 2012.

[5] A. Antoniou, Digital Signal Processing: Signals, Systems, and Filters, McGraw-Hill, New York, NY, 2006.

[6] R.E. Crochiere, L.R. Rabiner, Multirate Digital Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[7] A.V. Oppenheim, A.S. Willsky, A.H. Nawab, Signals and Systems, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1997.

[8] A.V. Oppenheim, R.W. Schaffer, Discrete-Time Signal Processing, third ed., Prentice-Hall, Englewood Cliffs, NJ, 2010.
[9] P.P. Vaidyanathan, Multirate Systems and Filter Banks, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[10] M. Vetterli, J. Kovačević, Wavelets and Subband Coding, Prentice-Hall, Englewood Cliffs, NJ, 1995.
[11] J.G. Proakis, D.G. Manolakis, Digital Signal Processing: Principles, Algorithms, and Applications, fourth ed., Prentice-Hall, Englewood Cliffs, NJ, 2007.
[12] J.G. Proakis, D.G. Manolakis, Applied Digital Signal Processing, Cambridge University Press, Cambridge, UK, 2011.
[13] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way, third ed., Academic Press, Burlington, MA, 2009.
[14] S.M. Kay, Modern Spectral Estimation: Theory and Application, Prentice-Hall, Englewood Cliffs, NJ, 1988.
[15] J.H. McClellan, R.W. Schafer, M.A. Yoder, DSP First: A Multimedia Approach, third ed., Prentice-Hall, Englewood Cliffs, NJ, 1998.
[16] S.K. Mitra, Digital Signal Processing: A Computer-Based Approach, third ed., McGraw-Hill, New York, NY, 2006.
[17] A. Papoulis, S. UnniKrishna Pillai, Probability, Random Variables, and Stochastic Processes, fourth ed., McGraw-Hill, New York, NY, 2002.
[18] J.J. Shynk, Probability, Random Variables, and Random Processes: Theory and Signal Processing Applications, John Wiley & Sons, Hoboken, NJ, 2013.
[19] T. Saramäki, Finite-impulse response filter design, in: S.K. Mitra, J.F. Kaiser (Eds.), Handbook of Digital Signal Processing, John Wiley & Sons, New York, NY, pp. 155–177.
[20] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons, Chichester, UK, 1980.
[21] D.G. Luenberger, Introduction to Linear and Nonlinear Programming, second ed., Addison-Wesley, Boston, MA, 1984.
[22] A. Antoniou, W.-S. Lu, Practical Optimization: Algorithms and Engineering Applications, Springer, New York, NY, 2013.
[23] D. Tse, P. Viswanath, Fundamentals of Wireless Communication, Cambridge University Press, Cambridge, UK, 2005.
[24] P.P. Vaidyanathan, S.-M. Phoong, Y.-P. Lin, Signal Processing and Optimization for Transceiver Systems, Cambridge University Press, Cambridge, UK, 2010.
[25] H.L. Van Trees, Optimum Array Processing, John Wiley & Sons, New York, NY, 2002.
[26] S. Haykin, Neural Networks and Learning Machines, Prentice-Hall, Englewood Cliffs, NJ, 2009.
[27] S. Haykin, Adaptive Filter Theory, fourth ed., Prentice-Hall, Englewood Cliffs, NJ, 2002.
[28] A.H. Sayed, Adaptive Filters, John Wiley & Sons, Hoboken, NJ, 2008.