Introduction

When we hear the word "signal" we may first think of a phenomenon that occurs continuously over time that carries some information. Most phenomena observed in nature are continuous in time, such as for instance our speech or heart beating, the temperature of the city where we live, the car speed during a given trip, the altitude of the airplane we are traveling in – these are typical continuous-time signals. Engineers are always devising ways to design systems, which are in principle continuous time, for measuring and interfering with these and other phenomena.

One should note that, although continuous-time signals pervade our daily lives, there are also several signals which are originally discrete time; for example, the stock-market weekly financial indicators, the maximum and minimum daily temperatures in our cities, and the average lap speed of a racing car.

If an electrical or computer engineer has the task of designing systems to interact with natural phenomena, their first impulse is to convert some quantities from nature into electric signals through a transducer. Electric signals, which are represented by voltages or currents, have a continuous-time representation. Since digital technology constitutes an extremely powerful tool for information processing, it is natural to think of processing the electric signals generated using it. However, continuous-time signals cannot be processed using computer technology (digital machines), which are especially suited to deal with sequential computation involving numbers. Fortunately, this fact does not prevent the use of digital integrated circuits (which is the technology behind the computer technology revolution we witness today) in signal processing systems designs. This is because many signals taken from nature can be fully represented by their sampled versions, where the sampled signals coincide with the original continuous-time signals at some instants in time. If we know how fast the important information changes, then we can always sample and convert continuous-time information into discrete-time information which, in turn, can be converted into a sequence of numbers and transferred to a digital machine.

The main advantages of digital systems relative to analog systems are high reliability, suitability for modifying the system's characteristics, and low cost. These advantages motivated the digital implementation of many signal processing systems which used to be implemented with analog circuit technology. In addition, a number of new applications became viable once the very large scale integration (VLSI) technology was available. Usually in the VLSI implementation of a digital signal processing system the concern is to reduce power consumption and/or area, or to increase the circuit's speed in order to meet the demands of high-throughput applications.

Currently, a single digital integrated circuit may contain millions of logic gates operating at very high speeds, allowing very fast digital processors to be built at a reasonable cost. This technological development opened avenues to the introduction of powerful general-purpose computers which can be used to design and implement digital signal processing systems. In addition, it allowed the design of microprocessors with special features for signal processing, namely the digital signal processors (DSPs). As a consequence, there are several tools available to implement very complex digital signal processing systems. In practice, a digital signal processing system is implemented either by software on a general-purpose digital computer or DSP, or by using application-specific hardware, usually in the form of an integrated circuit.

For the reasons explained above, the field of digital signal processing has developed so fast in recent decades that it has been incorporated into the graduate and undergraduate programs of virtually all universities. This is confirmed by the number of good textbooks available in this area: Oppenheim & Schafer (1975, 1989); Rabiner & Gold (1975); Peled & Liu (1985); Roberts & Mullis (1987); Ifeachor & Jervis (1993); Jackson (1996); Antoniou (2006); Mitra (2006); Proakis & Manolakis (2007). The present book is aimed at equipping readers with tools that will enable them to design and analyze most digital signal processing systems. The building blocks for digital signal processing systems considered here are used to process signals which are discrete in time and in amplitude. The main tools emphasized in this text are:

- discrete-time signal representations
- discrete transforms and their fast algorithms
- spectral estimation
- design and implementation of digital filters and digital signal processing systems
- multirate systems and filter banks
- wavelets.

Transforms and filters are the main parts of linear signal processing systems. Although the techniques we deal with are directly applicable to processing deterministic signals, many statistical signal processing methods employ similar building blocks in some way, as will be clear in the text.

Digital signal processing is extremely useful in many areas. In the following, we enumerate a few of the disciplines where the topics covered by this book have found application.

(a) *Image processing*: An image is essentially a space-domain signal; that is, it represents a variation of light intensity and color in space. Therefore, in order to process an image using an analog system, it has to be converted into a time-domain signal, using some form of scanning process. However, to process an image digitally, there is no need to perform this conversion, for it can be processed directly in the spatial domain, as a matrix of numbers. This lends the digital image processing techniques enormous power. In fact, in image processing, two-dimensional signal representation, filtering, and transforms play a central role (Jain, 1989).

(b) *Multimedia systems*: Such systems deal with different kinds of information sources, such as image, video, audio, and data. In such systems, the information is essentially represented in digital form. Therefore, it is crucial to remove redundancy from the information, allowing compression, and thus efficient transmission and storage (Jayant & Noll, 1984; Gersho & Gray, 1992; Bhaskaran & Konstantinides, 1997). The Internet is a good application example where information files are transferred in a compressed manner. Most of the compression standards for images use transforms and quantizers.

The transforms, filter banks, and wavelets are very popular in compression applications because they are able to exploit the high correlation of signal sources such as audio, still images, and video (Malvar, 1992; Fliege, 1994; Vetterli and Kovačević, 1995; Strang and Nguyen, 1996; Mallat, 1999).

(c) *Communication systems*: In communication systems, the compression and coding of the information sources are also crucial, since services can be provided at higher speed or to more users by reducing the amount of data to be transmitted. In addition, channel coding, which consists of inserting redundancy in the signal to compensate for possible channel distortions, may also use special types of digital filtering (Stüber, 1996).

Communication systems usually include fixed filters, as well as some self-designing filters for equalization and channel modeling which fall in the class of adaptive filtering systems (Diniz, 2008). Although these filters employ a statistical signal processing framework (Hayes, 1996; Kay, 1988; Manolakis *et al.*, 2000) to determine how their parameters should change, they also use some of the filter structures and in some cases the transforms introduced in this book.

Many filtering concepts take part on modern multiuser communication systems employing code-division multiple access (Verdu, 1998).

Wavelets, transforms, and filter banks also play a crucial role in the conception of orthogonal frequency-division multiplexing (OFDM) (Akansu & Medley, 1999), which is used in digital audio and TV broadcasting.

(d) *Audio signal processing*: In statistical signal processing the filters are designed based on observed signals, which might imply that we are estimating the parameters of the model governing these signals (Kailath *et al.*, 2000). Such estimation techniques can be employed in digital audio restoration (Godsill & Rayner, 1998), where the resulting models can be used to restore lost information. However, these estimation models can be simplified and made more effective if we use some kind of sub-band processing with filter banks and transforms (Kahrs & Brandenburg, 1998). In the same field, digital filters were found to be suitable for reverberation algorithms and as models for musical instruments (Kahrs & Brandenburg, 1998).

In addition to the above applications, digital signal processing is at the heart of modern developments in speech analysis and synthesis, mobile radio, sonar, radar, biomedical engineering, seismology, home appliances, and instrumentation, among others. These developments occurred in parallel with the advances in the technology of transmission, processing, recording, reproduction, and general treatment of signals through analog and digital electronics, as well as other means such as acoustics, mechanics, and optics.

We expect that, with the digital signal processing tools described in this book, the reader will be able to proceed further, not only exploring and understanding some of the applications described above, but developing new ones as well.

1 Discrete-time signals and systems

1.1 Introduction

Digital signal processing is the discipline that studies the rules governing the behavior of discrete signals, as well as the systems used to process them. It also deals with the issues involved in processing continuous signals using digital techniques. Digital signal processing pervades modern life. It has applications in compact disc players, computer tomography, geological processing, mobile phones, electronic toys, and many others.

In analog signal processing, we take a continuous signal, representing a continuously varying physical quantity, and pass it through a system that modifies this signal for a certain purpose. This modification is, in general, continuously variable by nature; that is, it can be described by differential equations.

Alternatively, in digital signal processing, we process sequences of numbers using some sort of digital hardware. We usually call these sequences of numbers digital or discrete-time signals. The power of digital signal processing comes from the fact that, once a sequence of numbers is available to appropriate digital hardware, we can carry out any form of numerical processing on it. For example, suppose we need to perform the following operation on a continuous-time signal:

$$y(t) = \frac{\cosh\left[\ln(|x(t)|) + x^3(t) + \cos^3\left(\sqrt{|x(t)|}\right)\right]}{5x^5(t) + \mathrm{e}^{x(t)} + \tan(x(t))}. \tag{1.1}$$

This would be clearly very difficult to implement using analog hardware. However, if we sample the analog signal $x(t)$ and convert it into a sequence of numbers $x(n)$, it can be input to a digital computer, which can perform the above operation easily and reliably, generating a sequence of numbers $y(n)$. If the continuous-time signal $y(t)$ can be recovered from $y(n)$, then the desired processing has been successfully performed.

This simple example highlights two important points. One is how powerful digital signal processing is. The other is that, if we want to process an analog signal using this sort of resource, we must have a way of converting a continuous-time signal into a discrete-time one, such that the continuous-time signal can be recovered from the discrete-time signal. However, it is important to note that very often discrete-time signals do not come from continuous-time signals, that is, they are originally discrete-time, and the results of their processing are only needed in digital form.

In this chapter, we study the basic concepts of the theory of discrete-time signals and systems. We emphasize the treatment of discrete-time systems as separate entities from continuous-time systems. We first define discrete-time signals and, based on this, we define discrete-time systems, highlighting the properties of an important subset of these systems, namely linearity and time invariance, as well as their description by discrete-time convolutions. We then study the time-domain response of discrete-time systems by characterizing them using difference equations. We close the chapter with Nyquist's sampling theorem, which tells us how to generate, from a continuous-time signal, a discrete-time signal from which the continuous-time signal can be completely recovered. Nyquist's sampling theorem forms the basis of the digital processing of continuous-time signals.

1.2 Discrete-time signals

A discrete-time signal is one that can be represented by a sequence of numbers. For example, the sequence

$$\{x(n), \quad n \in \mathbb{Z}\}, \tag{1.2}$$

where $\mathbb{Z}$ is the set of integer numbers, can represent a discrete-time signal where each number $x(n)$ corresponds to the amplitude of the signal at an instant nT. If $x_{\mathrm{a}}(t)$ is an analog signal, we have that

$$x(n) = x_{\mathrm{a}}(nT), \quad n \in \mathbb{Z}. \tag{1.3}$$

Since n is an integer, T represents the interval between two consecutive points at which the signal is defined. It is important to note that T is not necessarily a time unit. For example, if $x_{\mathrm{a}}(t)$ is the temperature along a metal rod, then if T is a length unit, $x(n) = x_{\mathrm{a}}(nT)$ may represent the temperature at sensors placed uniformly along this rod.

In this text, we usually represent a discrete-time signal using the notation in Equation (1.2), where $x(n)$ is referred to as the nth sample of the signal (or the nth element of the sequence). An alternative notation, used in many texts, is to represent the signal as

$$\{x_{\mathrm{a}}(nT), \quad n \in \mathbb{Z}\}, \tag{1.4}$$

where the discrete-time signal is represented explicitly as samples of an analog signal $x_{\mathrm{a}}(t)$. In this case, the time interval between samples is explicitly shown; that is, $x_{\mathrm{a}}(nT)$ is the sample at time nT. Note that, using the notation in Equation (1.2), a discrete-time signal whose adjacent samples are 0.03 s apart would be represented as

$$\ldots \quad x(0), x(1), x(2), x(3), x(4), \ldots, \tag{1.5}$$

whereas, using Equation (1.4) it would be represented as

$$\ldots \quad x_{\mathrm{a}}(0), x_{\mathrm{a}}(0.03), x_{\mathrm{a}}(0.06), x_{\mathrm{a}}(0.09), x_{\mathrm{a}}(0.12), \ldots \tag{1.6}$$

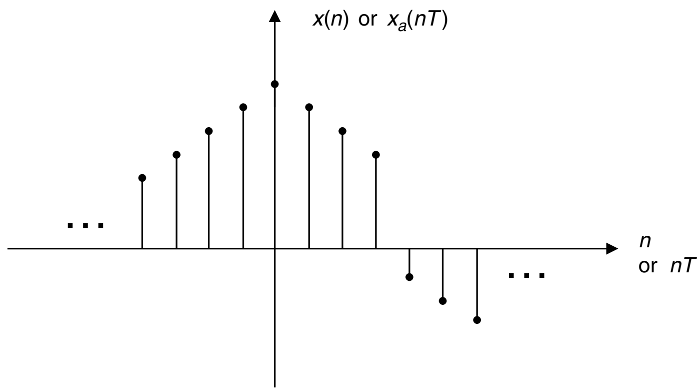

Fig. 1.1. **General representation of a discrete-time signal.**

The graphical representation of a general discrete-time signal is shown in Figure 1.1. In what follows, we describe some of the most important discrete-time signals.

Unit impulse (see Figure 1.2a):

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0. \end{cases} \tag{1.7}$$

Delayed unit impulse (see Figure 1.2b):

$$\delta(n-m) = \begin{cases} 1, & n = m \\ 0, & n \neq m. \end{cases} \tag{1.8}$$

Unit step (see Figure 1.2c):

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0. \end{cases} \tag{1.9}$$

Cosine function (see Figure 1.2d):

$$x(n) = \cos(\omega n). \tag{1.10}$$

The angular frequency of this sinusoid is ω rad/sample and its frequency is $\omega/2\pi$ cycles/sample. For example, in Figure 1.2d, the cosine function has angular frequency $\omega = 2\pi/16$ rad/sample. This means that it completes one cycle, that equals 2π radians, in 16 samples. If the sample separation represents time, then ω can be given in rad/(time unit). It is important to note that

$$\cos[(\omega + 2k\pi)n] = \cos(\omega n + 2kn\pi) = \cos(\omega n) \tag{1.11}$$

for $k \in \mathbb{Z}$. This implies that, in the case of discrete signals, there is an ambiguity in defining the frequency of a sinusoid. In other words, when referring to discrete sinusoids, ω and $\omega + 2k\pi$, $k \in \mathbb{Z}$, are the same frequency.

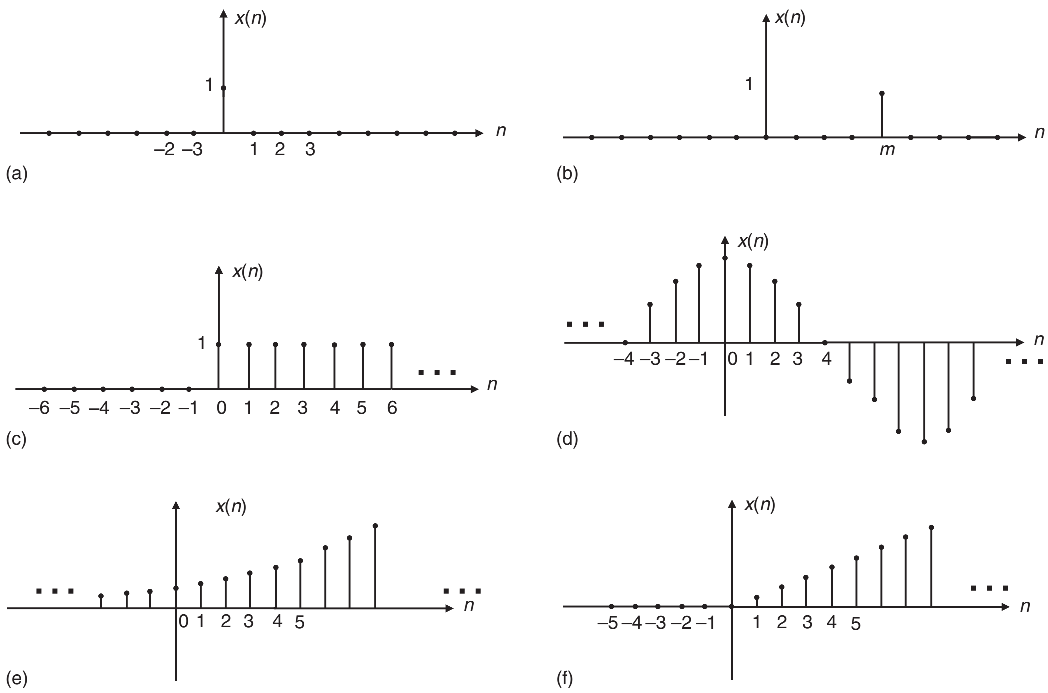

Fig. 1.2. **Basic discrete-time functions: (a) unit impulse; (b) delayed unit impulse; (c) unit step; (d) cosine function with $\omega = 2\pi/16$ rad/sample; (e) real exponential function with $a = 0.2$; (f) unit ramp.**

Real exponential function (see Figure 1.2e):

$$x(n) = \mathrm{e}^{an}. \tag{1.12}$$

Unit ramp (see Figure 1.2f):

$$r(n) = \begin{cases} n, & n \geq 0 \\ 0, & n < 0 \end{cases} \tag{1.13}$$

By examining Figure 1.2b–f, we notice that any discrete-time signal is equivalent to a sum of shifted impulses multiplied by a constant; that is, the impulse shifted by k samples is multiplied by $x(k)$. This can also be deduced from the definition of a shifted impulse in Equation (1.8). For example, the unit step $u(n)$ in Equation (1.9) can also be expressed as

$$u(n) = \sum_{k=0}^{\infty} \delta(n-k). \tag{1.14}$$

Likewise, any discrete-time signal $x(n)$ can be expressed as

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k). \tag{1.15}$$

An important class of discrete-time signals or sequences is that of periodic sequences. A sequence $x(n)$ is periodic if and only if there is an integer $N \neq 0$ such that $x(n) = x(n+N)$ for all n. In such a case, N is called a period of the sequence. Note that, using this definition and referring to Equation (1.10), a period of the cosine function is an integer N such that

$$\cos(\omega n) = \cos[\omega(n+N)], \quad \text{for all } n \in \mathbb{Z}. \tag{1.16}$$

This happens only if there is $k \in \mathbb{N}$ such that $\omega N = 2\pi k$. The smallest period is then

$$N = \min_{\substack{k \in \mathbb{N} \\ (2\pi/\omega)k \in \mathbb{N}}} \left\{ \frac{2\pi}{\omega} k \right\}. \tag{1.17}$$

Therefore, we notice that not all discrete cosine sequences are periodic, as illustrated in Example 1.1. An example of a periodic cosine sequence with period $N = 16$ samples is given in Figure 1.2d.

Example 1.1. Determine whether the discrete signals above are periodic; if they are, determine their periods.

(a) $x(n) = \cos\left[(12\pi/5)n\right]$
(b) $x(n) = 10\sin^2\left[(7\pi/12)n + \sqrt{2}\right]$
(c) $x(n) = 2\cos(0.02n + 3)$.

Solution

(a) In this case, we must have

$$\frac{12\pi}{5}(n+N) = \frac{12\pi}{5}n + 2k\pi \quad \Rightarrow \quad N = \frac{5k}{6}. \tag{1.18}$$

This implies that the smallest N results for $k = 6$. Then the sequence is periodic with period $N = 5$. Note that in this case

$$\cos\left(\frac{12\pi}{5}n\right) = \cos\left(\frac{2\pi}{5}n + 2\pi n\right) = \cos\left(\frac{2\pi}{5}n\right) \tag{1.19}$$

and thus we have also that the frequency of this sinusoid, besides being $\omega = 12\pi/5$, is also $\omega = 2\pi/5$, as indicated by Equation (1.11).

(b) In this case, periodicity implies that

$$\sin^2\left[\frac{7\pi}{12}(n+N) + \sqrt{2}\right] = \sin^2\left(\frac{7\pi}{12}n + \sqrt{2}\right) \tag{1.20}$$

and then

$$\sin\left[\frac{7\pi}{12}(n+N) + \sqrt{2}\right] = \pm\sin\left(\frac{7\pi}{12}n + \sqrt{2}\right) \tag{1.21}$$

such that

$$\frac{7\pi}{12}(n+N) = \frac{7\pi}{12}n + k\pi \quad \Rightarrow \quad N = \frac{12k}{7}. \tag{1.22}$$

The smallest N results for $k = 7$. Then this discrete-time signal is periodic with period $N = 12$.

(c) The periodicity condition requires that

$$\cos[0.02(n+N)+3] = \cos(0.02n+3) \tag{1.23}$$

such that

$$0.02(n+N) = 0.02n + 2k\pi \quad \Rightarrow \quad N = 100k\pi. \tag{1.24}$$

Since no integer N satisfies the above equation, the sequence is not periodic.

△

1.3 Discrete-time systems

A discrete-time system maps an input sequence $x(n)$ to an output sequence $y(n)$, such that

$$y(n) = \mathcal{H}\{x(n)\}, \tag{1.25}$$

where the operator $\mathcal{H}\{\cdot\}$ represents a discrete-time system, as shown in Figure 1.3. Depending on the properties of $\mathcal{H}\{\cdot\}$, the discrete-time system can be classified in several ways, the most basic ones being either linear or nonlinear, either time invariant or time variant, and causal or noncausal. These classifications will be discussed in what follows.

1.3.1 Linearity

Let us suppose that there is a system that accepts as input a voice signal and outputs the voice signal modified such that its acute components (high frequencies) are enhanced. In such a system, it would be undesirable if, in the case that one increased the voice tone at the input, the output became distorted instead of enhanced. Actually, one tends to expect

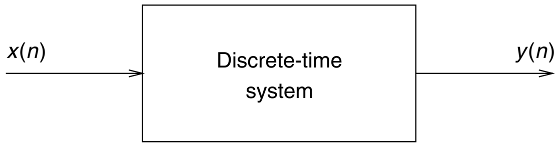

Fig. 1.3. Representation of a discrete-time system.