Chapter 1

Introduction

Digital Signal Processing:

> *That discipline which has allowed us to replace a circuit previously composed of a capacitor and a resistor with two antialiasing filters, an A-to-D and a D-to-A converter, and a general purpose computer (or array processor) so long as the signal we are interested in does not vary too quickly.*
>
> Thomas P. Barnwell, 1974

Signals encountered in real life are often in continuous time, that is, they are waveforms (or functions) on the real line. Their amplitude is usually continuous as well, meaning that it ·can take any real value in a certain range. Signals continuous in time and amplitude are called *analog signals.* There are many kinds of analog signals appearing in various applications. Examples include:

1. Electrical signals: voltages, currents, electric fields, magnetic fields.
2. Mechanical signals: linear displacements, angles, velocities, angular velocities, forces, moments.
3. Acoustic signals: vibrations, sound waves.
4. Signals related to physical sciences: pressures, temperatures, concentrations.

Analog signals are converted to voltages or currents by *sensors,* or *transducers,* in order to be processed electrically. Analog signal processing involves operations such as amplification, filtering, integration, and differentiation, as well as various forms of nonlinear processing (squaring, rectification). Analog processing of electrical signals is typically based on electronic amplifiers, resistors, capacitors, inductors, and so on. Limitations and drawbacks of analog processing include:

1. Accuracy limitations, due to component tolerances, amplifier nonlinearity, biases, and so on.
2. Limited repeatability, due to tolerances and variations resulting from environmental conditions, such as temperature, vibrations, and mechanical shocks.
3. Sensitivity to electrical noise, for example, internal amplifier noise.
4. Limited dynamic range of voltages and currents.
5. Limited processing speeds due to physical delays.

6. Lack of flexibility to specification changes in the processing functions.
7. Difficulty in implementing nonlinear and time-varying operations.
8. High cost and accuracy limitations of storage and retrieval of analog information.

Digital signal processing (DSP) is based on representing signals by numbers in a computer (or in specialized digital hardware), and performing various numerical operations on these signals. Operations in digital signal processing systems include, but are not limited to, additions, multiplications, data transfers, and logical operations. To implement a DSP system, we must be able:

1. To convert analog signals into digital information, in the form of a sequence of binary numbers. This involves two operations: sampling and analog-to-digital (A/D) conversion.
2. To perform numerical operations on the digital information, either by a computer or special-purpose digital hardware.
3. To convert the digital information, after being processed, back to an analog signal. This again involves two operations: digital-to-analog (D/A) conversion and reconstruction.

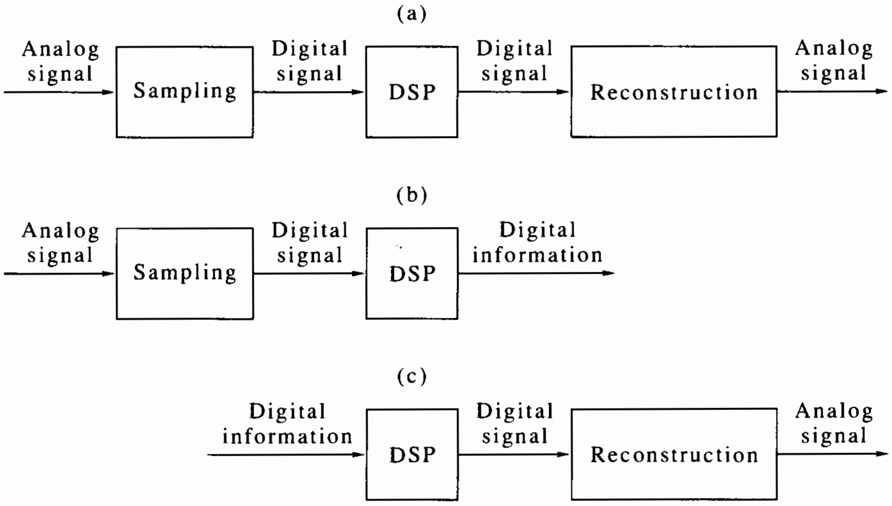

Figure 1.1 Basic DSP schemes: (a) general signal processing system; (b) signal analysis system; (c) signal synthesis system.

There are four basic schemes of digital signal processing, as shown in Figure 1.1:

1. A general DSP system is shown in part a. This system accepts an analog input signal, converts it to a digital signal, processes it digitally, and converts it back to analog. An example of such a system is digital recording and playback of music. The music signal is sensed by microphones, amplified, and converted to digital. The digital processor performs such tasks as filtering, mixing, and reverberation control. Finally, the digital music signal is converted back to analog, in order to be played back by a sound system.
2. A signal analysis system is shown in part b. Such systems are used for applications that require us to extract only certain information from the analog signal. As an example, consider the Touch-Tone system of telephone dialing. A Touch-Tone

dial includes 12 buttons arranged in a 4×3 matrix. When we push a button, two sinusoidal signals (tones) are generated, determined by the row and column numbers of the button. These two tones are added together and transmitted through the telephone lines. A digital system can identify which button was pressed by determining the frequencies of the two tones, since these frequencies uniquely identify the button. In this case, the output information is a number between 1 and 12.

3. A signal synthesis system is shown in part c. Such systems are used when we need to generate an analog signal from digital information. As an example, consider a text-to-speech system. Such a system receives text information character by character, where each character is represented by a numerical code. The characters are used for constructing syllables; these are used for generating artificial digital sound waveforms, which are converted to analog in order to be played back by a sound system.
4. A fourth type of DSP system is purely digital, accepting and yielding digital information. Such a system can be regarded as a degenerate version of any of the three aforementioned types.

As we see, Thomas Barnwell's definition of DSP (quoted in the beginning of the chapter), although originally meant as ironic, is essentially correct today as it was when first expressed, in the early days of DSP. However, despite the relative complexity of DSP systems, there is much to gain for this complexity. Digital signal processing has the potential of freeing us from many limitations of analog signal processing. In particular:

1. Computers can be made accurate to any desired degree (at least theoretically), by choosing their word length according to the required accuracy. Double precision can be used when single precision is not sufficient, or even quadruple precision, etc.
2. Computers are perfectly repeatable, as long as they do not malfunction (due to either hardware or software failure).
3. The sensitivity of computers to electrical noise is extremely low (but not nil, as is commonly believed; electrical noise can give rise to bit errors, although rarely).
4. Use of floating point makes it possible, by choosing the word length, to have a practically infinite dynamic range.
5. Speed is a limiting factor in computers as well as in analog devices. However, advances in technology (greater CPU and memory speeds, parallel processing) push this limit forward continually.
6. Changes in processing functions can be made through programming. Although programming (or software development in general) is usually a difficult task, its implementation (by loading the new software into the computer storage devices) is relatively easy.
7. Implementing nonlinear and time-varying operations (e.g., in adaptive filtering) is conceptually easy, since it can be accomplished via programming, and there is usually no need to build special hardware.
8. Digital storage is cheap and flexible.
9. Digital information can be encrypted for security, coded against errors, and compressed to reduce storage and transmission costs.

Digital signal processing is not free of drawbacks and limitations of its own:

1. Sampling inevitably leads to loss of information. Although this loss can be minimized by careful sampling, it cannot be completely avoided.
2. A/D and D/A conversion hardware may be expensive, especially if great accuracy and speed are required. It is also never completely free of noise and distortions.
3. Although hardware becomes cheaper and more sophisticated every year, this is not necessarily true for software. On the contrary, software development and testing appear more and more often to be the main bottleneck in developing digital signal processing applications (and in the digital world in general).
4. In certain applications, notably processing of RF signals, digital processing still cannot meet speed requirements.

The theoretical foundations of digital signal processing were laid by Jean Baptiste Joseph Fourier who, in 1807, presented to the Institut de France a paper on what we call today *Fourier series*.[1] Major theoretical developments in digital signal processing theory were made in the 1930s and 1940s by Nyquist and Shannon, among others (in the context of digital communication), and by the developers of the z-transform (notably Zadeh and Ragazzini in the West, and Tsypkin in the East). The history of applied digital signal processing (at least in the electrical engineering world) began around the mid-1960s with the invention of the fast Fourier transform (FFT). However, its rapid development started with the advent of microprocessors in the 1970s. Early DSP systems were designed mainly to replace existing analog circuitry, and did little more than mimicking the operation of analog signal processing systems. It was gradually realized that DSP has the potential for performing tasks impractical or even inconceivable to perform by analog means. Today, digital signal processing is a clear winner over analog processing. Whereas analog processing is—and will continue to be—limited by technology, digital processing appears to be limited only by our imagination.[2]

We cannot do justice to all applications of DSP in this short introduction, but we name a few of them without details:

Biomedical applications: analysis of biomedical signals, diagnosis, patient monitoring, preventive health care, artificial organs.

Communication: encoding and decoding of digital communication signals, detection, equalization, filtering, direction finding.

Digital control: servomechanism, automatic pilots, chemical plants.

General signal analysis: spectrum estimation, parameter estimation, signal modeling, signal classification, signal compression.

Image processing: filtering, enhancement, coding, compression, pattern recognition.

Instrumentation: signal generation, filtering.

Multimedia: generation, storage, and transmission of sound, still images, motion pictures, digital TV, video conferencing.

Music applications: recording, playback and manipulation (mixing, special effects), synthesis of digital music.

Radar: radar signal filtering, target detection, position and velocity estimation, tracking, radar imaging.

Sonar: similar to radar.

Speech applications: noise filtering, coding, compression, recognition, synthesis of artificial speech.

Telephony: transmission of information in digital form via telephone lines, modem technology, cellular phones.

Implementation of digital signal processing varies according to the application. Off-line or laboratory-oriented processing is usually done on general purpose computers using high-level software (such as C, or more recently MATLAB). On-line or field-oriented processing is usually performed with microprocessors tailored to DSP applications. Applications requiring very high processing speeds often use special-purpose very-large-scale integration (VLSI) hardware.

1.1 Contents of the Book

Teaching of digital signal processing begins at the point where a typical signals and systems course ends. A student who has learned signals and systems knows the basic mathematical theory of signals and their relationships to linear time-invariant systems: convolutions, transforms, frequency responses, transfer functions, concepts of stability, simple block-diagram manipulations, and more. Modern signals and systems curricula put equal emphases on continuous-time and discrete-time signals. Chapter 2 reviews this material to the extent needed as a prerequisite for the remainder of the book. This chapter also contains two topics less likely to be included in a signals and systems course: real Fourier series (also called Fourier cosine and sine series) and basic theory of random signals.

Sampling and reconstruction are introduced in Chapter 3. Sampling converts a continuous-time signal to a discrete-time signal, reconstruction performs the opposite conversion. When a signal is sampled it is irreversibly distorted, in general, preventing its exact restoration in the reconstruction process. Distortion due to sampling is called *aliasing*. Aliasing can be practically eliminated under certain conditions, or at least minimized. Reconstruction also leads to distortions due to physical limitations on realizable (as opposed to ideal) reconstructors. These subjects occupy the main part of the chapter.

Chapter 3 also includes a section on physical aspects of sampling: digital-to-analog and analog-to-digital converters, their operation, implementation, and limitations.

Three chapters are devoted to frequency-domain digital signal processing. Chapter 4 introduces the discrete Fourier transform (DFT) and discusses in detail its properties and a few of its uses. This chapter also teaches the discrete cosine transform (DCT), a tool of great importance in signal compression. Chapter 5 concerns the fast Fourier transform (FFT). Chapter 6 is devoted to practical aspects of frequency-domain analysis. It explains the main problems in frequency-domain analysis, and teaches how to use the DFT and FFT for solving these problems. Part of this chapter assumes knowledge of random signals, to the extent reviewed in Chapter 2.

Chapter 7 reviews the z-transform, difference equations, and transfer functions. Like Chapter 2, it contains only material needed as a prerequisite for later chapters.

Three chapters are devoted to digital filtering. Chapter 8 introduces the concept of filtering, filter specifications, magnitude and phase properties of digital filters, and review of digital filter design. Chapters 9 and 10 discuss the two classes of digital filters: finite impulse response (FIR) and infinite impulse response (IIR), respectively. The focus is on filter design techniques and on properties of filters designed by different techniques.

The last four chapters contain relatively advanced material. Chapter 11 discusses filter realizations, introduces state-space representations, and analyses finite word length effects. Chapter 12 deals with multirate signal processing, including an introduction to filter banks. Chapter 13 concerns the analysis and modeling of random signals. Finally, Chapter 14 describes selected applications of digital signal processing: compression, speech modeling, analysis of music signals, digital communication, analysis of biomedical signals, and special DSP hardware.

1.2 Notational Conventions

In this section we introduce the notational conventions used throughout this book. Some of the concepts should be known, whereas others are likely to be new. All concepts will be explained in detail later; the purpose of this section is to serve as a convenient reference and reminder.

1. **Signals**
   (a) We denote the real line by $\mathbb{R}$, the complex plane by $\mathbb{C}$, and the set of integers by $\mathbb{Z}$.
   (b) In general, we denote temporal signals by lowercase letters.
   (c) Continuous-time signals (i.e., functions on the real line) are denoted with their arguments in round parentheses; for example: $x(t)$, $y(t)$, and so on.
   (d) Discrete-time signals (i.e., sequences, or functions on the integers) are denoted with their arguments in square brackets; for example: $x[n], y[n]$, and so on.
   (e) Let the continuous-time signal $x(t)$ be defined on the interval $[0, T]$. Its periodic extension on the real line is denoted as
   $$\tilde{x}(t) = x(t \bmod T).$$
   (f) Let the discrete-time signal $x[n]$ be defined for $0 \le n \le N-1$. Its periodic extension on the integers is denoted by
   $$\tilde{x}[n] = x[n \bmod N].$$

2. **Convolutions**
   (a) The *convolution* of continuous-time signals is (whenever the right side exists)
   $$\{x * y\}(t) = \int_{-\infty}^{\infty} x(\tau)y(t-\tau)d\tau, \quad t \in \mathbb{R}. \tag{1.1}$$
   Convolution is an operator acting on a pair of continuous-time signals x and y, and producing a continuous-time signal $x * y$, whose value at time t is $\{x * y\}(t)$.
   (b) The *discrete-time convolution* of discrete-time signals is (whenever the right side exists)
   $$\{x * y\}[n] = \sum_{m=-\infty}^{\infty} x[m]y[n-m], \quad n \in \mathbb{Z}. \tag{1.2}$$
   Discrete-time convolution is an operator acting on a pair of discrete-time signals x and y, and producing a discrete-time signal $x * y$, whose value at time n is $\{x * y\}[n]$.
   (c) The *circular convolution* of finite-duration, discrete-time signals $\{x[n], y[n], 0 \le n \le N-1\}$ is
   $$\{x \circledast y\}[n] = \sum_{m=0}^{N-1} x[m]y[(n-m) \bmod N], \quad 0 \le n \le N-1. \tag{1.3}$$

Circular convolution can be regarded as an operator acting on a pair of N-dimensional vectors x and y, and producing an N-dimensional vector $x \circledast y$, whose nth component is $\{x \circledast y\}[n]$.

3. **Transforms**

For transforms we use two types of notation: a script letter to denote the *transform operator*, and an uppercase letter modified by a superscript to denote the resulting function. The superscript specifies the transform in question. An uppercase superscript indicates that the transformed signal is in continuous time, whereas a lowercase superscript indicates that the transformed signal is in discrete time.

(a) The *two-sided Laplace transform* of the continuous-time signal $x(t)$ is[3]

$$X^{\mathrm{L}}(s) = \{\mathcal{L}x\}(s) = \int_{-\infty}^{\infty} x(t)e^{-st}dt, \quad s \in \mathbb{C}, \tag{1.4}$$

whenever the right side exists.

(b) The *Fourier transform* of the continuous-time signal $x(t)$ is

$$X^{\mathrm{F}}(\omega) = \{\mathcal{F}x\}(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt, \quad \omega \in \mathbb{R}, \tag{1.5}$$

whenever the right side exists. The inverse relationship is

$$x(t) = \{\mathcal{F}^{-1}X^{\mathrm{F}}\}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X^{\mathrm{F}}(\omega)e^{j\omega t}d\omega, \quad t \in \mathbb{R}. \tag{1.6}$$

We have, when both Laplace and Fourier transforms exist,

$$X^{\mathrm{F}}(\omega) = X^{\mathrm{L}}(j\omega). \tag{1.7}$$

(c) The *Fourier series* of the continuous-time signal $x(t)$ on the interval $[0, T]$ (or of its periodic extension) is

$$X^{\mathrm{S}}[k] = \{\mathcal{S}x\}[k] = \frac{1}{T}\int_{0}^{T} x(t)\exp\left(-\frac{j2\pi kt}{T}\right)dt, \quad k \in \mathbb{Z}. \tag{1.8}$$

The inverse relationship is

$$x(t) = \{\mathcal{S}^{-1}X^{\mathrm{S}}\}(t) = \sum_{k=-\infty}^{\infty} X^{\mathrm{S}}[k]\exp\left(\frac{j2\pi kt}{T}\right), \quad 0 \le t \le T. \tag{1.9}$$

(d) The *two-sided z-transform* of the discrete-time signal $x[n]$ is

$$X^{\mathrm{z}}(z) = \{\mathcal{Z}x\}(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}, \tag{1.10}$$

whenever the right side exists.

(e) The *Fourier transform* of the discrete-time signal $x[n]$ is

$$X^{\mathrm{f}}(\theta) = \{\mathcal{F}x\}(\theta) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\theta n}, \quad \theta \in \mathbb{R}, \tag{1.11}$$

whenever the right side exists. The inverse relationship is

$$x[n] = \{\mathcal{F}^{-1}X^{\mathrm{f}}\}[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X^{\mathrm{f}}(\theta)e^{j\theta n}d\theta, \quad n \in \mathbb{Z}. \tag{1.12}$$

We have, when both z- and Fourier transforms exist,

$$X^{\mathrm{f}}(\theta) = X^{\mathrm{z}}(e^{j\theta}). \tag{1.13}$$

(f) The *discrete Fourier transform* of the finite-duration, discrete-time signal $\{x[n],\ 0 \le n \le N-1\}$ is

$$X^{\mathrm{d}}[k] = \{\mathcal{D}x\}[k] = \sum_{n=0}^{N-1} x[n]\exp\left(-\frac{j2\pi kn}{N}\right), \quad 0 \le k \le N-1. \tag{1.14}$$

The inverse relationship is

$$x[n] = \{\mathcal{D}^{-1}X^{\mathrm{d}}\}[n] = \frac{1}{N}\sum_{k=0}^{N-1} X^{\mathrm{d}}[k]\exp\left(\frac{j2\pi kn}{N}\right), \quad 0 \le n \le N-1. \quad (1.15)$$

1.3 Summation Rules

The notation

$$\sum_{n=1}^{N} a[n] = a[1] + a[2] + \cdots + a[N]$$

should be familiar to readers of this book. Digital signal processing uses the summation notation extensively. We therefore present here a few rules for proper handling of summations.

1. In the sum

$$\sum_{n=n_1}^{n_2} a[n,k],$$

the variables n and k play completely different roles. The variable n is *dummy*: It does not exist outside the sum, and it can be renamed arbitrarily without affecting the value of the expression. In mathematical logic, it is known as a *bound variable*. The variable k is *free*: The result depends on it and it cannot be renamed, lest it change the value of the expression. The first rule of summation is therefore: Always be sure to distinguish between free variables and bound variables in the expression you are reading or writing. Never use the same symbol for a free variable and a dummy variable in the same expression. For example, the expression

$$a[n]\sum_{n=n_1}^{n_2} b[n],$$

which is perfectly valid mathematically, will create havoc when the product is expanded. Use

$$a[n]\sum_{m=n_1}^{n_2} b[m]$$

instead, and now you can safely expand it to

$$\sum_{m=n_1}^{n_2} a[n]b[m].$$

2. A sum can be broken up into several sums, and several sums can be joined into a single sum as in

$$\sum_{n=n_1}^{n_3} a[n] = \sum_{n=n_1}^{n_2} a[n] + \sum_{n=n_2+1}^{n_3} a[n],$$

provided n_2 is between n_1 and n_3.

3. If the upper limit is smaller than the lower limit, the sum is zero by definition, that is

$$\sum_{n=n_1}^{n_2} a[n] = 0, \quad \text{if} \quad n_2 < n_1.$$

We call this an *empty sum*. Note that this convention is *not* shared by integrals, where $\int_a^b x(t)dt = -\int_b^a x(t)dt$.

4. Zero terms do not affect the sum. Thus, if $a[n] = 0$ for $n_2 + 1 \le n \le n_3$, then

$$\sum_{n=n_1}^{n_2} a[n] = \sum_{n=n_1}^{n_3} a[n].$$

5. Here is how a change of variables appears:

$$\sum_{n=n_1}^{n_2} a[n+k] = \sum_{m=n_1+k}^{n_2+k} a[m] = \sum_{n=n_1+k}^{n_2+k} a[n].$$

In passing from the first form to the second, we made the substitution $m = n + k$. Note how this affects the summation limits. Since m is a dummy variable, we can revert from m to n and use the third form.

6. Here is how a change of variables with reversal of direction appears:

$$\sum_{n=n_1}^{n_2} a[k-n] = \sum_{m=k-n_2}^{k-n_1} a[m] = \sum_{n=k-n_2}^{k-n_1} a[n].$$

Carefully observe the change of summation limits in this case.

7. Double sums typically appear in two situations:
   (a) When expanding a product of sums as in

$$\left[\sum_{n=n_1}^{n_2} a[n]\right]\left[\sum_{m=m_1}^{m_2} b[m]\right] = \sum_{n=n_1}^{n_2}\sum_{m=m_1}^{m_2} a[n]b[m].$$

   (b) When substituting an entity involving one sum in another sum as in

$$\sum_{n=n_1}^{n_2} a[n]\left[\sum_{m=m_1}^{m_2} b[m,n]\right] = \sum_{n=n_1}^{n_2}\sum_{m=m_1}^{m_2} a[n]b[m,n].$$

Be sure to use different dummy variables in the two sums. Never use the same dummy variable in two different sums appearing in the same expression. For example, never try to expand the left side of 7a in the form $\left[\sum_{n=n_1}^{n_2} a[n]\right]\left[\sum_{n=m_1}^{m_2} b[n]\right]$. Change the dummy variable in one of the sums, as shown above.

8. Convolution of finite sequences is often confusing to beginners, due to the need to set the summation limits properly. Consider the convolution of the sequences $\{x[n],\ 0 \le n \le N_1 - 1\}$ and $\{y[n],\ 0 \le n \le N_2 - 1\}$. In theory, the convolution is

$$z[n] = \sum_{m=-\infty}^{\infty} x[m]y[n-m].$$

However, due to the finite length of the two operands, the terms in the sum are nonzero only when

$$0 \le m \le N_1 - 1, \quad 0 \le n - m \le N_2 - 1.$$

This is the same as

$$0 \le m \le N_1 - 1, \quad n - N_2 + 1 \le m \le n,$$

which can also be written as

$$\max\{0, n - N_2 + 1\} \le m \le \min\{N_1 - 1, n\}.$$

In summary, the convolution of the two sequences is given by

$$z[n] = \sum_{m=\max\{0,n-N_2+1\}}^{\min\{N_1-1,n\}} x[m]y[n-m]. \tag{1.16}$$

1.4 Summary and Complements

1.4.1 Summary

In this chapter we introduced the concept of digital signal processing, compared digital and analog processing methodologies, mentioned a few applications of DSP, and presented a brief summary of the contents of the book. We also introduced the system of notation used in this book and included some guidelines for the use of summation.

Of the many textbooks on digital signal processing, some of the better known are Oppenheim and Schafer [1975, 1989], Parks and Burrus [1987], Roberts and Mullis [1987], Proakis and Manolakis [1992], Antoniou [1993], Kuc [1988], Strum and Kirk [1989], Haddad and Parsons [1991], and Jackson [1996].

1.4.2 Complements

1. [p. 4] This paper was not published at that time, due to the unyielding opposition of J. L. Lagrange.
2. [p. 4] We do not wish to form the impression that analog signal processing has become obsolete. Analog signal processing is used, for example (1) for relatively simple tasks, in which analog implementation is the most economical, (2) when speed requirements imposed by the frequency of the signal render digital processing impractical, (3) when interfacing analog signals to digital processors (more on this will be said in Chapter 3). Also, some recent electronic devices operate on analog signals (such as voltages and charges), but in discrete time; examples include charge-coupled devices and switched-capacitor circuits.
3. [p. 7] The *one-side Laplace transform* is defined as

$$X_+^{\mathrm{L}}(s) = \{\mathcal{L}x\}(s) = \int_0^{\infty} x(t)e^{-st}dt, \quad s \in \mathbb{C}. \tag{1.17}$$

   Note that the two Laplace transforms of $x(t)$ coincide if and only if $x(t) = 0$ for $t < 0$. Also, (1.7) does not hold in general for $X_+^{\mathrm{L}}(s)$.