Jenkins, W.K. “Fourier Series, Fourier Transforms, and the DFT”
Digital Signal Processing Handbook
Ed. Vijay K. Madisetti and Douglas B. Williams
Boca Raton: CRC Press LLC, 1999

1

Fourier Series, Fourier Transforms, and the DFT

W. Kenneth Jenkins
University of Illinois, Urbana-Champaign

1.1 Introduction

Fourier methods are commonly used for signal analysis and system design in modern telecommunications, radar, and image processing systems. Classical Fourier methods such as the Fourier series and the Fourier integral are used for continuous time (CT) signals and systems, i.e., systems in which a characteristic signal, $s(t)$, is defined at all values of t on the continuum $-\infty < t < \infty$. A more recently developed set of Fourier methods, including the discrete time Fourier transform (DTFT) and the discrete Fourier transform (DFT), are extensions of basic Fourier concepts that apply to discrete time (DT) signals. A characteristic DT signal, $s[n]$, is defined only for values of n where n is an integer in the range $-\infty < n < \infty$. The following discussion presents basic concepts and outlines important properties for both the CT and DT classes of Fourier methods, with a particular emphasis on the relationships between these two classes. The class of DT Fourier methods is particularly useful

as a basis for digital signal processing (DSP) because it extends the theory of classical Fourier analysis to DT signals and leads to many effective algorithms that can be directly implemented on general computers or special purpose DSP devices.

The relationship between the CT and the DT domains is characterized by the operations of sampling and reconstruction. If $s_a(t)$ denotes a signal $s(t)$ that has been uniformly sampled every T seconds, then the mathematical representation of $s_a(t)$ is given by

$$s_a(t) = \sum_{n=-\infty}^{\infty} s(t)\delta(t - nT) \tag{1.1}$$

where $\delta(t)$ is a CT impulse function defined to be zero for all $t \neq 0$, undefined at $t = 0$, and has unit area when integrated from $t = -\infty$ to $t = +\infty$. Because the only places at which the product $s(t)\delta(t-nT)$ is not identically equal to zero are at the sampling instances, $s(t)$ in (1.1) can be replaced with $s(nT)$ without changing the overall meaning of the expression. Hence, an alternate expression for $s_a(t)$ that is often useful in Fourier analysis is given by

$$s_a(t) = \sum_{n=-\infty}^{\infty} s(nT)\delta(t - nT) \tag{1.2}$$

The CT sampling model $s_a(t)$ consists of a sequence of CT impulse functions uniformly spaced at intervals of T seconds and weighted by the values of the signal $s(t)$ at the sampling instants, as depicted in Fig. 1.1. Note that $s_a(t)$ is not defined at the sampling instants because the CT impulse function itself is not defined at $t = 0$. However, the values of $s(t)$ at the sampling instants are imbedded as "area under the curve" of $s_a(t)$, and as such represent a useful mathematical model of the sampling process. In the DT domain the sampling model is simply the sequence defined by taking the values of $s(t)$ at the sampling instants, i.e.,

$$s[n] = s(t)|_{t=nT} \tag{1.3}$$

In contrast to $s_a(t)$, which is not defined at the sampling instants, $s[n]$ is well defined at the sampling instants, as illustrated in Fig. 1.2. Thus, it is now clear that $s_a(t)$ and $s[n]$ are different but equivalent models of the sampling process in the CT and DT domains, respectively. They are both useful for signal analysis in their corresponding domains. Their equivalence is established by the fact that they have equal spectra in the Fourier domain, and that the underlying CT signal from which $s_a(t)$ and $s[n]$ are derived can be recovered from either sampling representation, provided a sufficiently large sampling rate is used in the sampling operation (see below).

1.2 Fourier Series Representation of Continuous Time Periodic Signals

It is convenient to begin this discussion with the classical Fourier series representation of a periodic time domain signal, and then derive the Fourier integral from this representation by finding the limit of the Fourier coefficient representation as the period goes to infinity. The conditions under which a periodic signal $s(t)$ can be expanded in a Fourier series are known as the Dirichet conditions. They require that in each period $s(t)$ has a finite number of discontinuities, a finite number of maxima and minima, and that $s(t)$ satisfies the following absolute convergence criterion [1]:

$$\int_{-T/2}^{T/2} |s(t)|\, dt < \infty \tag{1.4}$$

It is assumed in the following discussion that these basic conditions are satisfied by all functions that will be represented by a Fourier series.

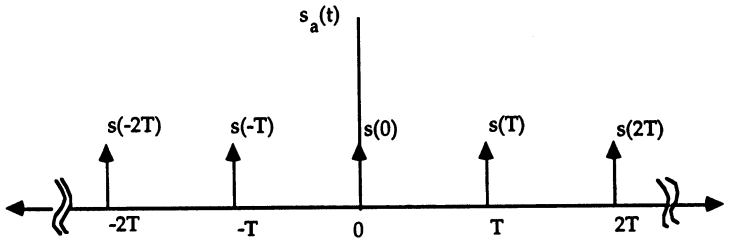

FIGURE 4.1 CT model of a sampled CT signal.

FIGURE 1.1: CT model of a sampled CT signal.

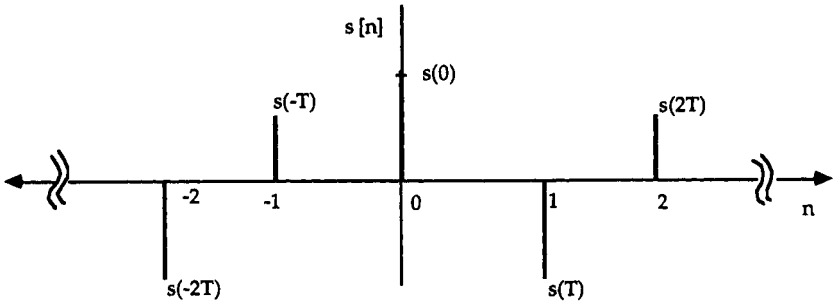

FIGURE 1.2: DT model of a sampled CT signal.

1.2.1 Exponential Fourier Series

If a CT signal $s(t)$ is periodic with a period T, then the classical complex Fourier series representation of $s(t)$ is given by

$$s(t) = \sum_{n=-\infty}^{\infty} a_n e^{jn\omega_0 t} \tag{1.5a}$$

where $\omega_0 = 2\pi/T$, and where the a_n are the complex Fourier coefficients given by

$$a_n = (1/T) \int_{-T/2}^{T/2} s(t) e^{-jn\omega_0 t}\, dt \tag{1.5b}$$

It is well known that for every value of t where $s(t)$ is continuous, the right-hand side of (1.5a) converges to $s(t)$. At values of t where $s(t)$ has a finite jump discontinuity, the right-hand side of (1.5a) converges to the average of $s(t^-)$ and $s(t^+)$, where $s(t^-) \equiv \lim_{\epsilon \to 0} s(t-\epsilon)$ and $s(t^+) \equiv \lim_{\epsilon \to 0} s(t+\epsilon)$.

For example, the Fourier series expansion of the sawtooth waveform illustrated in Fig. 1.3 is characterized by $T = 2\pi$, $\omega_0 = 1$, $a_0 = 0$, and $a_n = a_{-n} = A\cos(n\pi)/(jn\pi)$ for $n = 1, 2, \ldots,$. The coefficients of the exponential Fourier series represented by (1.5b) can be interpreted as the spectral representation of $s(t)$, because the a_n-th coefficient represents the contribution of the $(n\omega_0)$-th frequency to the total signal $s(t)$. Because the a_n are complex valued, the Fourier domain represen-

tation has both a magnitude and a phase spectrum. For example, the magnitude of the a_n is plotted in Fig. 1.4 for the sawtooth waveform of Fig. 1.3. The fact that the a_n constitute a discrete set is consistent with the fact that a periodic signal has a "line spectrum," i.e., the spectrum contains only integer multiples of the fundamental frequency ω_0. Therefore, the equation pair given by (1.5a) and (1.5b) can be interpreted as a transform pair that is similar to the CT Fourier transform for periodic signals. This leads to the observation that the classical Fourier series can be interpreted as a special transform that provides a one-to-one invertible mapping between the discrete-spectral domain and the CT domain. The next section shows how the periodicity constraint can be removed to produce the more general classical CT Fourier transform, which applies equally well to periodic and aperiodic time domain waveforms.

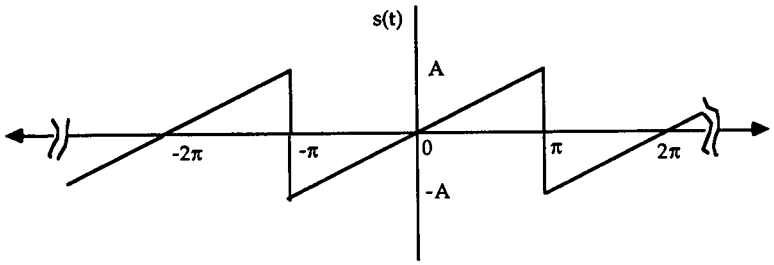

FIGURE 1.3: Periodic CT signal used in Fourier series example.

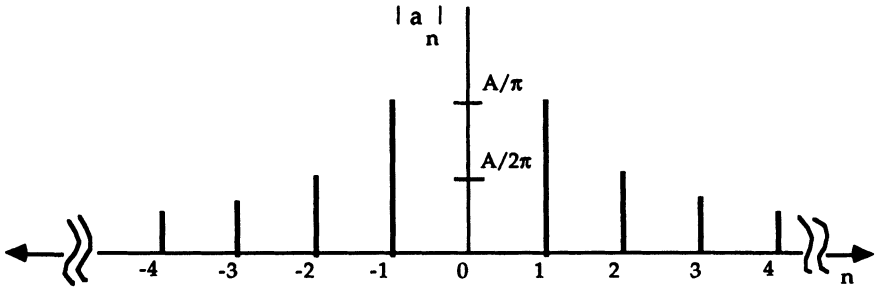

FIGURE 1.4: Magnitude of the Fourier coefficients for example of Figure 1.3.

1.2.2 The Trigonometric Fourier Series

Although Fourier series expansions exist for complex periodic signals, and Fourier theory can be generalized to the case of complex signals, the theory and results are more easily expressed for real-valued signals. The following discussion assumes that the signal $s(t)$ is real-valued for the sake of simplifying the discussion. However, all results are valid for complex signals, although the details of the theory will become somewhat more complicated.

For real-valued signals $s(t)$, it is possible to manipulate the complex exponential form of the Fourier series into a trigonometric form that contains $\sin(\omega_0 t)$ and $\cos(\omega_0 t)$ terms with corresponding real-

valued coefficients [1]. The trigonometric form of the Fourier series for a real-valued signal $s(t)$ is given by

$$s(t) = \sum_{n=0}^{\infty} b_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} c_n \sin(n\omega_0 t) \tag{1.6a}$$

where $\omega_0 = 2\pi/T$. The b_n and c_n are real-valued Fourier coefficients determined by

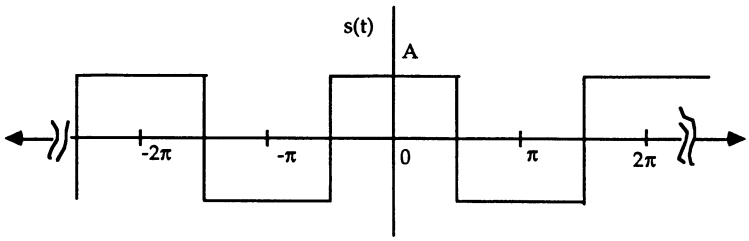

FIGURE 1.5: Periodic CT signal used in Fourier series example 2.

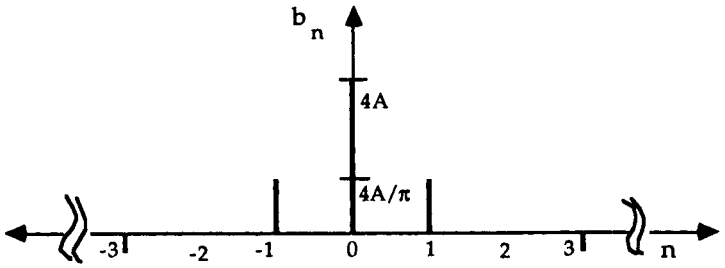

FIGURE 1.6: Fourier coefficients for example of Figure 1.5.

$$\begin{aligned} b_0 &= (1/T)\int_{-T/2}^{T/2} s(t)\,dt \\ b_n &= (2/T)\int_{-T/2}^{T/2} s(t)\cos(n\omega_0 t)\,dt, \qquad n = 1, 2, \ldots, \\ c_n &= (2/T)\int_{-T/2}^{T/2} s(t)\sin(n\omega_0 t)\,dt, \qquad n = 1, 2, \ldots, \end{aligned} \tag{1.6b}$$

An arbitrary real-valued signal $s(t)$ can be expressed as a sum of even and odd components, $s(t) = s_{\text{even}}(t) + s_{\text{odd}}(t)$, where $s_{\text{even}}(t) = s_{\text{even}}(-t)$ and $s_{\text{odd}}(t) = -s_{\text{odd}}(-t)$, and where $s_{\text{even}}(t) = [s(t) + s(-t)]/2$ and $s_{\text{odd}}(t) = [s(t) - s(-t)]/2$. For the trigonometric Fourier series, it can be shown that $s_{\text{even}}(t)$ is represented by the (even) cosine terms in the infinite series, $s_{\text{odd}}(t)$ is represented by the (odd) sine terms, and b_0 is the DC level of the signal. Therefore, if it can be determined by inspection that a signal has DC level, or if it is even or odd, then the correct form of the trigonometric

series can be chosen to simplify the analysis. For example, it is easily seen that the signal shown in Fig. 1.5 is an even signal with a zero DC level. Therefore it can be accurately represented by the cosine series with $b_n = 2A\,\sin(\pi n/2)/(\pi n/2)$, $n = 1, 2, \ldots$, as illustrated in Fig. 1.6. In contrast, note that the sawtooth waveform used in the previous example is an odd signal with zero DC level; thus, it can be completely specified by the sine terms of the trigonometric series. This result can be demonstrated by pairing each positive frequency component from the exponential series with its conjugate partner, i.e., $c_n = \sin(n\omega_0 t) = a_n e^{jn\omega_0 t} + a_{-n} e^{-jn\omega_0 t}$, whereby it is found that $c_n = 2\text{A}\,\cos(n\pi)/(n\pi)$ for this example. In general it is found that $a_n = (b_n - jc_n)/2$ for $n = 1, 2, \ldots$, $a_0 = b_0$, and $a_{-n} = a_n^*$. The trigonometric Fourier series is common in the signal processing literature because it replaces complex coefficients with real ones and often results in a simpler and more intuitive interpretation of the results.

1.2.3 Convergence of the Fourier Series

The Fourier series representation of a periodic signal is an approximation that exhibits mean squared convergence to the true signal. If $s(t)$ is a periodic signal of period T, and $s'(t)$ denotes the Fourier series approximation of $s(t)$, then $s(t)$ and $s'(t)$ are equal in the mean square sense if

$$\text{MSE} = \int_{-T/2}^{T/2} |s(t) - s(t)'|^2\, dt = 0 \tag{1.7}$$

Even with (1.7) satisfied, mean square error (MSE) convergence does not mean that $s(t) = s'(t)$ at every value of t. In particular, it is known that at values of t, where $s(t)$ is discontinuous, the Fourier series converges to the average of the limiting values to the left and right of the discontinuity. For example, if t_0 is a point of discontinuity, then $s'(t_0) = [s(t_0^-) + s(t_0^+)]/2$, where $s(t_0^-)$ and $s(t_0^+)$ were defined previously. (Note that at points of continuity, this condition is also satisfied by the definition of continuity.) Because the Dirichet conditions require that $s(t)$ have at most a finite number of points of discontinuity in one period, the set S_t, defined as all values of t within one period where $s(t) \neq s'(t)$, contains a finite number of points, and S_t is a set of measure zero in the formal mathematical sense. Therefore, $s(t)$ and its Fourier series expansion $s'(t)$ are *equal almost everywhere*, and $s(t)$ can be considered identical to $s'(t)$ for the analysis of most practical engineering problems.

Convergence almost everywhere is satisfied only in the limit as an infinite number of terms are included in the Fourier series expansion. If the infinite series expansion of the Fourier series is truncated to a finite number of terms, as it must be in practical applications, then the approximation will exhibit an oscillatory behavior around the discontinuity, known as the Gibbs phenomenon [1]. Let $s'_N(t)$ denote a truncated Fourier series approximation of $s(t)$, where only the terms in (1.5a) from $n = -N$ to $n = N$ are included if the complex Fourier series representation is used, or where only the terms in (1.6a) from $n = 0$ to $n = N$ are included if the trigonometric form of the Fourier series is used. It is well known that in the vicinity of a discontinuity at t_0 the Gibbs phenomenon causes $s'_N(t)$ to be a poor approximation to $s(t)$. The peak magnitude of the Gibbs oscillation is 13% of the size of the jump discontinuity $s(t_0^-) - s(t_0^+)$ regardless of the number of terms used in the approximation. As N increases, the region that contains the oscillation becomes more concentrated in the neighborhood of the discontinuity, until, in the limit as N approaches infinity, the Gibbs oscillation is squeezed into a single point of mismatch at t_0.

If $s'(t)$ is replaced by $s'_N(t)$ in (1.7), it is important to understand the behavior of the error MSE_N as a function of N, where

$$\text{MSE}_N = \int_{-T/2}^{T/2} |s(t) - s'_N(t)|^2\, dt \tag{1.8}$$

An important property of the Fourier series is that the exponential basis functions $e^{jn\omega_0 t}$ (or $\sin(n\omega_0 t)$ and $\cos(n\omega_0 t)$ for the trigonometric form) for $n = 0, \pm 1, \pm 2, \ldots$ (or $n = 0, 1, 2, \ldots$ for the trigonometric form) constitute an orthonormal set, i.e., $t_{nk} = 1$ for $n = k$, and $t_{nk} = 0$ for $n \neq k$, where

$$t_{nk} = (1/T) \int_{-T/2}^{T/2} (e^{-jn\omega_0 t})(e^{jk\omega_0 t})\, dt \tag{1.9}$$

As terms are added to the Fourier series expansion, the orthogonality of the basis functions guarantees that the error decreases in the mean square sense, i.e., that MSE_N monotonically decreases as N is increased. Therefore, a practitioner can proceed with the confidence that when applying Fourier series analysis more terms are always better than fewer in terms of the accuracy of the signal representations.

1.3 The Classical Fourier Transform for Continuous Time Signals

The periodicity constraint imposed on the Fourier series representation can be removed by taking the limits of (1.5a) and (1.5b) as the period T is increased to infinity. Some mathematical preliminaries are required so that the results will be well defined after the limit is taken. It is convenient to remove the $(1/T)$ factor in front of the integral by multiplying (1.5b) through by T, and then replacing Ta_n by a'_n in both (1.5a) and (1.5b). Because $\omega_0 = 2\pi/T$, as T increases to infinity, ω_0 becomes infinitesimally small, a condition that is denoted by replacing ω_0 with $\Delta\omega$. The factor $(1/T)$ in (1.5a) becomes $(\Delta\omega/2\pi)$. With these algebraic manipulations and changes in notation (1.5a) and (1.5b) take on the following form prior to taking the limit:

$$s(t) = (1/2\pi) \sum_{n=-\infty}^{\infty} a'_n e^{jn\Delta\omega t}\, \Delta\omega \tag{1.10a}$$

$$a'_n = \int_{-T/2}^{T/2} s(t) e^{-jn\Delta\omega t}\, dt \tag{1.10b}$$

The final step in obtaining the CT Fourier transform is to take the limit of both (1.10a) and (1.10b) as $T \to \infty$. In the limit the infinite summation in (1.10a) becomes an integral, $\Delta\omega$ becomes $d\omega$, $n\Delta\omega$ becomes ω, and a'_n becomes the CT Fourier transform of $s(t)$, denoted by $S(j\omega)$. The result is summarized by the following transform pair, which is known throughout most of the engineering literature as the classical CT Fourier transform (CTFT):

$$s(t) = (1/2\pi) \int_{-\infty}^{\infty} S(j\omega) e^{j\omega t}\, d\omega \tag{1.11a}$$

$$S(j\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t}\, dt \tag{1.11b}$$

Often (1.11a) is called the Fourier integral and (1.11b) is simply called the Fourier transform. The relationship $S(j\omega) = \mathcal{F}\{s(t)\}$ denotes the Fourier transformation of $s(t)$, where $\mathcal{F}\{\cdot\}$ is a symbolic notation for the Fourier transform operator, and where ω becomes the continuous frequency variable after the periodicity constraint is removed. A transform pair $s(t) \leftrightarrow S(j\omega)$ represents a one-to-one invertible mapping as long as $s(t)$ satisfies conditions which guarantee that the Fourier integral converges.

From (1.11a) it is easily seen that $\mathcal{F}\{\delta(t - t_0)\} = e^{-j\omega t_0}$, and from (1.11b) that $\mathcal{F}^{-1}\{2\pi\delta(\omega - \omega_0)\} = e^{j\omega_0 t}$, so that $\delta(t - t_0) \leftrightarrow e^{-j\omega t_0}$ and $e^{j\omega_0 t} \leftrightarrow 2\pi\delta(\omega - \omega_0)$ are valid Fourier transform

pairs. Using these relationships it is easy to establish the Fourier transforms of $\cos(\omega_0 t)$ and $\sin(\omega_0 t)$, as well as many other useful waveforms that are encountered in common signal analysis problems. A number of such transforms are shown in Table 1.1.

The CTFT is useful in the analysis and design of CT systems, i.e., systems that process CT signals. Fourier analysis is particularly applicable to the design of CT filters which are characterized by Fourier magnitude and phase spectra, i.e., by $|H(j\omega)|$ and $\arg H(j\omega)$, where $H(j\omega)$ is commonly called the frequency response of the filter. For example, an **ideal transmission channel** is one which passes a signal without distorting it. The signal may be scaled by a real constant A and delayed by a fixed time increment t_0, implying that the impulse response of an ideal channel is $A\delta(t - t_0)$, and its corresponding frequency response is $Ae^{-j\omega t_0}$. Hence, the frequency response of an ideal channel is specified by constant amplitude for all frequencies, and a phase characteristic which is linear function given by ωt_0.

1.3.1 Properties of the Continuous Time Fourier Transform

The CTFT has many properties that make it useful for the analysis and design of linear CT systems. Some of the more useful properties are stated below. A more complete list of the CTFT properties is given in Table 1.2. Proofs of these properties can be found in [2] and [3]. In the following discussion $\mathcal{F}\{\cdot\}$ denotes the Fourier transform operation, $\mathcal{F}^{-1}\{\cdot\}$ denotes the inverse Fourier transform operation, and $*$ denotes the convolution operation defined as

$$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(t-\tau) f_2(\tau)\, d\tau$$

1. Linearity (superposition): $\mathcal{F}\{af_1(t) + bf_2(t)\} = a\mathcal{F}\{f_1(t)\} + b\mathcal{F}\{f_2(t)\}$ (a and b, complex constants)
2. Time shifting: $\mathcal{F}\{f(t - t_0)\} = e^{-j\omega t_0}\mathcal{F}\{f(t)\}$
3. Frequency shifting: $e^{j\omega_0 t} f(t) = \mathcal{F}^{-1}\{F(j(\omega - \omega_0))\}$
4. Time domain convolution: $\mathcal{F}\{f_1(t) * f_2(t)\} = \mathcal{F}\{f_1(t)\}\mathcal{F}\{f_2(t)\}$
5. Frequency domain convolution: $\mathcal{F}\{f_1(t) f_2(t)\} = (1/2\pi)\mathcal{F}\{f_1(t)\} * \mathcal{F}\{f_2(t)\}$
6. Time differentiation: $-j\omega F(j\omega) = \mathcal{F}\{d(f(t))/dt\}$
7. Time integration: $\mathcal{F}\{\int_{-\infty}^{t} f(\tau)\, d\tau\} = (1/j\omega)F(j\omega) + \pi F(0)\delta(\omega)$

The above properties are particularly useful in CT system analysis and design, especially when the system characteristics are easily specified in the frequency domain, as in linear filtering. Note that properties 1, 6, and 7 are useful for solving differential or integral equations. Property 4 provides the basis for many signal processing algorithms because many systems can be specified directly by their impulse or frequency response. Property 3 is particularly useful in analyzing communication systems in which different modulation formats are commonly used to shift spectral energy to frequency bands that are appropriate for the application.

1.3.2 Fourier Spectrum of the Continuous Time Sampling Model

Because the CT sampling model $s_a(t)$, given in (1.1), is in its own right a CT signal, it is appropriate to apply the CTFT to obtain an expression for the spectrum of the sampled signal:

$$\mathcal{F}\{s_a(t)\} = \mathcal{F}\left\{\sum_{n=-\infty}^{\infty} s(t)\delta(t - nT)\right\} = \sum_{n=-\infty}^{\infty} s(nT)e^{-j\omega Tn} \tag{1.12}$$

Because the expression on the right-hand side of (1.12) is a function of $e^{j\omega T}$ it is customary to denote the transform as $F(e^{j\omega T}) = \mathcal{F}\{s_a(t)\}$. Later in the chapter this result is compared to the result of

TABLE 1.1 Some Basic CTFT Pairs

Signal	Fourier Transform	Fourier Series Coefficients (if periodic)	
$\sum_{k=-\infty}^{+\infty} a_k e^{jk\omega_0 t}$	$2\pi \sum_{k=-\infty}^{+\infty} a_k \delta(\omega_k \omega_0)$	a_k	
$e^{j\omega_0 t}$	$2\pi\delta(\omega+\omega_0)$	$a_1 = 1$ $a_k = 0, \quad \text{otherwise}$	
$\cos\omega_0 t$	$\pi[\delta(\omega-\omega_0)+\delta(\omega+\omega_0)]$	$a_1 = a_{-1} = \frac{1}{2}$ $a_k = 0, \quad \text{otherwise}$	
$\sin\omega_0 t$	$\frac{\pi}{j}[\delta(\omega-\omega_0)-\delta(\omega+\omega_0)]$	$a_1 = -a_{-1} = \frac{1}{2j}$ $a_k = 0, \quad \text{otherwise}$	
$x(t) = 1$	$2\pi\delta(\omega)$	$a_0 = 1, \quad a_k = 0, \quad k \neq 0$ (has this Fourier series representation for any choice of $T_0 > 0$	
Periodic square wave $x(t) = \begin{cases} 1, & \lvert t\rvert < T_1 \\ 0, & T_1 < \lvert t\rvert \leq \frac{T_0}{2} \end{cases}$ and $x(t+T_0) = x(t)$	$\sum_{k=-\infty}^{+\infty} \frac{2\sin k\omega_0 T_1}{k} \delta(\omega_k \omega_0)$	$\frac{\omega_0 T_1}{\pi} \sin c\left(\frac{k\omega_0 T_1}{\pi}\right) = \frac{\sin k\omega_0 T_1}{k\pi}$	
$\sum_{n=-\infty}^{+\infty} \delta(t-nT)$	$\frac{2\pi}{T} \sum^{+\infty} k = -\infty \delta\left(\omega - \frac{2\pi k}{T}\right)$	$a_k = \frac{1}{T}$ for all k	
$x(t) = \begin{cases} 1, & \lvert t\rvert < T_1 \\ 0, & \lvert t\rvert > T_1 \end{cases}$	$2T_1 \sin c\left(\frac{\omega T_1}{\pi}\right) = \frac{2\sin\omega T_1}{\omega}$		—
$\frac{W}{\pi}\sin c\left(\frac{Wt}{\pi}\right) = \frac{\sin Wt}{\pi t}$	$X(\omega) = \begin{cases} 1, & \lvert\omega\rvert < W \\ 0, & \lvert\omega\rvert > W \end{cases}$		—
$\delta(t)$	1		—
$u(t)$	$\frac{1}{j\omega} + \pi\delta(\omega)$		—
$\delta(t-t_0)$	$e_{j\omega t_0}$		—
$e^{-at}u(t), \text{Re}\{a\} > 0$	$\frac{1}{a+j\omega}$		—
$te^{-at}u(t), \text{Re}\{a\} > 0$	$\frac{1}{(a+j\omega)^2}$		—
$\frac{t^{n-1}}{(n-1)!}e^{-at}u(t), \quad \text{Re}\{a\} > 0$	$\frac{1}{(a+j\omega)^n}$		—

TABLE 1.2 Properties of the CTFT

Name	If $\mathcal{F} f(t) = F(j\omega)$, then
Definition	$f(j\omega) = \int_{-\infty}^{\infty} f(t) e_{j\omega t}\, dt$
	$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) e^{j\omega t}\, d\omega$
Superposition	$\mathcal{F}[af_1(t) + bf_2(t)] = aF_1(j\omega) + bF_2(j\omega)$
Simplification if:	
(a) $f(t)$ is even	$F(j\omega) = 2\int_0^{\infty} f(t) \cos\omega t\, dt$
(b) $f(t)$ is odd	$F(j\omega) = 2j\int_0^{\infty} f(t) \sin\omega t\, dt$
Negative t	$\mathcal{F} f(-t) = F^*(j\omega)$
Scaling:	
(a) Time	$\mathcal{F} f(at) = \frac{1}{\lvert a\rvert} F\left(\frac{j\omega}{a}\right)$
(b) Magnitude	$\mathcal{F} af(t) = aF(j\omega)$
Differentiation	$\mathcal{F}\left[\frac{d^n}{dt^n} f(t)\right] = (j\omega)^n F(j\omega)$
Integration	$\mathcal{F}\left[\int_{-\infty}^{t} f(x)\, dx\right] = \frac{1}{j\omega} F(j\omega) + \pi F(0)\delta(\omega)$
Time shifting	$\mathcal{F} f(t-a) = F(j\omega) e^{j\omega a}$
Modulation	$\mathcal{F} f(t) e^{j\omega_0 t} = F[j(\omega - \omega_0)]$
	$\{\mathcal{F} f(t) \cos\omega_0 t = \frac{1}{2} F[j(\omega - \omega_0)] + F[j(\omega + \omega_0)]\}$
	$\{\mathcal{F} f(t) \sin\omega_0 t = \frac{1}{2} j[F[j(\omega - \omega_0)] - F[j(\omega + \omega_0)]\}$
Time convolution	$\mathcal{F}^{-1}[F_1(j\omega) F_2(j\omega)] = \int_{-\infty}^{\infty} f_1(\tau) f_2(\tau) f_2(t_\tau)\, d\tau$
Frequency convolution	$\mathcal{F}[f_1(t) f_2(t)] = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(j\lambda) F_2[j(\omega_\lambda)]\, d\lambda$

operating on the DT sampling model, namely $s[n]$, with the DT Fourier transform to illustrate that the two sampling models have the same spectrum.

1.3.3 Fourier Transform of Periodic Continuous Time Signals

We saw earlier that a periodic CT signal can be expressed in terms of its Fourier series. The CTFT can then be applied to the Fourier series representation of $s(t)$ to produce a mathematical expression for the "line spectrum" characteristic of periodic signals.

$$\mathcal{F}\{s(t)\} = \mathcal{F}\left\{\sum_{n=-\infty}^{\infty} a_n e^{jn\omega_0 t}\right\} = 2\pi \sum_{n=-\infty}^{\infty} a_n \delta(\omega - n\omega_0) \tag{1.13}$$

The spectrum is shown pictorially in Fig. 1.7. Note the similarity between the spectral representation of Fig. 1.7 and the plot of the Fourier coefficients in Fig. 1.4, which was heuristically interpreted as a "line spectrum". Figures 1.4 and 1.7 are different but equivalent representations of the Fourier

spectrum. Note that Fig. 1.4 is a DT representation of the spectrum, while Fig. 1.7 is a CT model of the same spectrum.

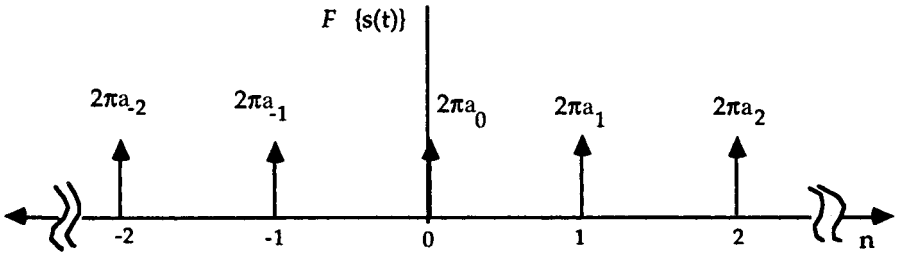

FIGURE 1.7: Spectrum of the Fourier series representation of $s(t)$.

1.3.4 The Generalized Complex Fourier Transform

The CTFT characterized by (1.11a) and (1.11b) can be generalized by considering the variable $j\omega$ to be the special case of $u = \sigma + j\omega$ with $\sigma = 0$, writing (1.11a) in terms of u, and interpreting u as a complex frequency variable. The resulting complex Fourier transform pair is given by (1.14a) and (1.14b)

$$s(t) = (1/2\pi j)\int_{\sigma-j\infty}^{\sigma+j\infty} S(u)e^{jut}\,du \tag{1.14a}$$

$$S(u) = \int_{-\infty}^{\infty} s(t)e^{-jut}\,dt \tag{1.14b}$$

The set of all values of u for which the integral of (1.14b) converges is called the region of convergence (ROC). Because the transform $S(u)$ is defined only for values of u within the ROC, the path of integration in (1.14a) must be defined by σ so that the entire path lies within the ROC. In some literature this transform pair is called the **bilateral Laplace transform** because it is the same result obtained by including both the negative and positive portions of the time axis in the classical Laplace transform integral. [Note that in (1.14a) the complex frequency variable was denoted by u rather than by the more common s, in order to avoid confusion with earlier uses of $s(\cdot)$ as signal notation.] The complex Fourier transform (bilateral Laplace transform) is not often used in solving practical problems, but its significance lies in the fact that it is the most general form that represents the point at which Fourier and Laplace transform concepts become the same. Identifying this connection reinforces the notion that Fourier and Laplace transform concepts are similar because they are derived by placing different constraints on the same general form.

1.4 The Discrete Time Fourier Transform

The discrete time Fourier transform (DTFT) can be obtained by using the DT sampling model and considering the relationship obtained in (1.12) to be the definition of the DTFT. Letting $T = 1$ so that the sampling period is removed from the equations and the frequency variable is replaced with

a normalized frequency $\omega' = \omega T$, the DTFT pair is defined in (1.15a). Note that in order to simplify notation it is not customary to distinguish between ω and ω', but rather to rely on the context of the discussion to determine whether ω refers to the normalized ($T = 1$) or the unnormalized ($T \neq 1$) frequency variable.

$$S(e^{j\omega'}) = \sum_{n=-\infty}^{\infty} s[n]e^{-j\omega' n} \tag{1.15a}$$

$$s[n] = (1/2\pi)\int_{-\pi}^{\pi} S(e^{j\omega'})e^{jn\omega'}d\omega' \tag{1.15b}$$

The spectrum $S(e^{j\omega'})$ is periodic in ω' with period 2π. The fundamental period in the range $-\pi < \omega' \leq \pi$, sometimes referred to as the baseband, is the useful frequency range of the DT system because frequency components in this range can be represented unambiguously in sampled form (without aliasing error). In much of the signal processing literature the explicit primed notation is omitted from the frequency variable. However, the explicit primed notation will be used throughout this section because the potential exists for confusion when so many related Fourier concepts are discussed within the same framework.

By comparing (1.12) and (1.15a), and noting that $\omega' = \omega T$, it is established that

$$\mathcal{F}\{s_a(t)\} = \text{DTFT}\{s[n]\} \tag{1.16}$$

where $s[n] = s(t)_{t=nT}$. This demonstrates that the spectrum of $s_a(t)$, as calculated by the CT Fourier transform is identical to the spectrum of $s[n]$ as calculated by the DTFT. Therefore, although $s_a(t)$ and $s[n]$ are quite different sampling models, they are equivalent in the sense that they have the same Fourier domain representation.

A list of common DTFT pairs is presented in Table 1.3. Just as the CT Fourier transform is useful in CT signal system analysis and design, the DTFT is equally useful in the same capacity for DT systems. It is indeed fortuitous that Fourier transform theory can be extended in this way to apply to DT systems.

In the same way that the CT Fourier transform was found to be a special case of the complex Fourier transform (or bilateral Laplace transform), the DTFT is a special case of the bilateral z-transform with $z = e^{j\omega' t}$. The more general bilateral z-transform is given by

$$S(z) = \sum_{n=-\infty}^{\infty} s[n]z^{-n} \tag{1.17a}$$

$$s[n] = (1/2\pi j)\int_{\mathbf{C}} S(z)z^{n-1}\,dz \tag{1.17b}$$

where $\mathbf{C}$ is a counterclockwise contour of integration which is a closed path completely contained within the region of convergence of $S(z)$. Recall that the DTFT was obtained by taking the CT Fourier transform of the CT sampling model represented by $s_a(t)$. Similarly, the bilateral z-transform results by taking the bilateral Laplace transform of $s_a(t)$. If the lower limit on the summation of (1.17a) is taken to be $n = 0$, then (1.17a) and (1.17b) become the one-sided z-transform, which is the DT equivalent of the one-sided LT for CT signals. The hierarchical relationship among these various concepts for DT systems is discussed later in this chapter, where it will be shown that the family structure of the DT family tree is identical to that of the CT family. For every CT transform in the CT world there is an analogous DT transform in the DT world, and vice versa.

TABLE 1.3 Some Basic DTFT Pairs

Sequence	Fourier Transform
1. $\delta[n]$	1
2. $\delta[n-n_0]$	$e^{-j\omega n_0}$
3. $1 \quad (-\infty < n < \infty)$	$\sum_{k=-\infty}^{\infty} 2\pi\delta(\omega + 2\pi k)$
4. $a^n u[n] \qquad (\lvert a\rvert < 1)$	$\frac{1}{1-ae^{-j\omega}}$
5. $u[n]$	$\frac{1}{1-e^{-j\omega}} + \sum_{k=-\infty}^{\infty} \pi\delta(\omega + 2\pi k)$
6. $(n+1)a^n u[n] \qquad (\lvert a\rvert < 1)$	$\frac{1}{(1-ae^{-j\omega})^2}$
7. $\frac{r^2 \sin\omega_p(n+1)}{\sin\omega_p} u[n] \qquad (\lvert r\rvert < 1)$	$\frac{1}{1-2r\cos\omega_p e^{-j\omega} + r^2 e^{j2\omega}}$
8. $\frac{\sin\omega_c n}{\pi n}$	$Xe^{j\omega} = \begin{cases} 1, & \lvert\omega\rvert < \omega_c \\ 0, & \omega_c < \lvert\omega\rvert \le \pi \end{cases}$
9. $x[n] - \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$	$\frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)} e^{-j\omega M/2}$
10. $e^{j\omega_0 n}$	$\sum_{k=-\infty}^{\infty} 2\pi\delta(\omega - \omega_0 + 2\pi k)$
11. $\cos(\omega_0 n + \phi)$	$\pi \sum_{k=-\infty}^{\infty} [e^{j\phi}\delta(\omega - \omega_0 + 2\pi k) + e^{-j\phi}\delta(\omega + \omega_0 + 2\pi k)]$

1.4.1 Properties of the Discrete Time Fourier Transform

Because the DTFT is a close relative of the classical CT Fourier transform it should come as no surprise that many properties of the DTFT are similar to those presented for the CT Fourier transform in the previous section. In fact, for many of the properties presented earlier an analogous property exists for the DTFT. The following list parallels the list that was presented in the previous section for the CT Fourier transform, to the extent that the same property exists. A more complete list of DTFT pairs is given in Table 1.4. (Note that the primed notation on ω' is dropped in the following to simplify the notation, and to be consistent with standard usage.)

1. Linearity (superposition): $\text{DTFT}\{af_1[n] + bf_2[n]\} = a\text{DTFT}\{f_1[n]\} + b\text{DTFT}\{f_2[n]\}$ (a and b, complex constants)
2. Index shifting: $\text{DTFT}\{f[n-n_0]\} = e^{-j\omega n_0}\text{DTFT}\{f[n]\}$
3. Frequency shifting: $e^{j\omega_0 n} f[n] = \text{DTFT}^{-1}\{F(e^{j(\omega-\omega_0)})\}$
4. Time domain convolution: $\text{DTFT}\{f_1[n] * f_2[n]\} = \text{DTFT}\{f_1[n]\}\text{DTFT}\{f_2[n]\}$
5. Frequency domain convolution: $\text{DTFT}\{f_1[n]f_2[n]\} = (1/2\pi)\text{DTFT}\{f_1[n]\} * \text{DTFT}\{f_2[n]\}$
6. Frequency differentiation: $nf[n] = \text{DTFT}^{-1}\{dF(e^{j\omega})/d\omega\}$

Note that the time-differentiation and time-integration properties of the CTFT do not have analogous counterparts in the DTFT because time domain differentiation and integration are not defined for DT

TABLE 1.4 Properties of the DTFT

Sequence	Fourier Transform
$x[n]$	$X(e^{j\omega})$
$y[n]$	$Y(e^{j\omega})$
1. $ax[n] + by[n]$	$aX(e^{j\omega}) + bY(e^{j\omega})$
2. $x[n - n_d]$ (n_d an integer)	$e^{-j\omega n_d} X(e^{j\omega})$
3. $e^{j\omega_0 n} x[n]$	$X(e^{j(\omega-\omega_0)})$
4. $x[-n]$	$X(e^{-j\omega})$ if $x[n]$ is real $X^*(e^{j\omega})$
5. $nx[n]$	$j\dfrac{dX(e^{j\omega})}{d\omega}$
6. $x[n] * y[n]$	$X(e^{j\omega})Y(e^{j\omega})$
7. $x[n]y[n]$	$\dfrac{1}{2\pi}\displaystyle\int_{-x}^{x} X(e^{j\theta})Y(e^{j(\omega-\theta)})\,d\theta$
Parseval's Theorem	
8. $\displaystyle\sum_{n=-\infty}^{\infty} \lvert x[n]\rvert^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \lvert X(e^{j\omega})\rvert^2\,d\omega$	
9. $\displaystyle\sum_{n=-\infty}^{\infty} x[n]y^*[n] = \frac{1}{2\pi}\inf_{-\pi}^{\pi} X(e^{j\omega})Y^*(e^{j\omega})\,d\omega$	

signals. When working with DT systems practitioners must often manipulate difference equations in the frequency domain. For this purpose property 1 and property 2 are very important. As with the CTFT, property 4 is very important for DT systems because it allows engineers to work with the frequency response of the system, in order to achieve proper shaping of the input spectrum or to achieve frequency selective filtering for noise reduction or signal detection. Also, property 3 is useful for the analysis of modulation and filtering operations common in both analog and digital communication systems.

The DTFT is defined so that the time domain is discrete and the frequency domain is continuous. This is in contrast to the CTFT that is defined to have continuous time and continuous frequency domains. The mathematical dual of the DTFT also exists, which is a transform pair that has a continuous time domain and a discrete frequency domain. In fact, the dual concept is really the same as the Fourier series for periodic CT signals presented earlier in the chapter, as represented by (1.5a) and (1.5b). However, the classical Fourier series arises from the assumption that the CT signal is inherently periodic, as opposed to the time domain becoming periodic by virtue of sampling the spectrum of a continuous frequency (aperiodic time) function [8]. The dual of the DTFT, the discrete frequency Fourier transform (DFFT), has been formulated and its properties tabulated as an interesting and useful transform in its own right [5]. Although the DFFT is similar in concept to the classical CT Fourier series, the formal properties of the DFFT [5] serve to clarify the effects of frequency domain sampling and time domain aliasing. These effects are obscured in the classical treatment of the CT Fourier series because the emphasis is on the inherent "line spectrum" that results from time domain periodicity. The DFFT is useful for the analysis and design of digital filters that are produced by frequency sampling techniques.

1.4.2 Relationship between the Continuous and Discrete Time Spectra

Because DT signals often originate by sampling CT signals, it is important to develop the relationship between the original spectrum of the CT signal and the spectrum of the DT signal that results. First,

the CTFT is applied to the CT sampling model, and the properties listed above are used to produce the following result:

$$\begin{aligned}\mathcal{F}\{s_a(t)\} &= \mathcal{F}\left\{s(t)\sum_{n=-\infty}^{\infty}\delta(t-nT)\right\} \\ &= (1/2\pi)S(j\omega) * \mathcal{F}\left\{\sum_{n=-\infty}^{\infty}\delta(t-nT)\right\}\end{aligned} \tag{1.18}$$

In this section it is important to distinguish between ω and ω', so the explicit primed notation is used in the following discussion where needed for clarification. Because the sampling function (summation of shifted impulses) on the right-hand side of the above equation is periodic with period T it can be replaced with a CT Fourier series expansion as follows:

$$S(e^{j\omega T}) = \mathcal{F}\{s_a(t)\} = (1/2\pi)S(j\omega) * \mathcal{F}\left\{\sum_{n=-\infty}^{\infty}(1/T)e^{j(2\pi/T)nt}\right\}$$

Applying the frequency domain convolution property of the CTFT yields

$$S(e^{j\omega T}) = (1/2\pi)\sum_{n=-\infty}^{\infty} S(j\omega) * (2\pi/T)\delta(\omega - (2\pi/T)n)$$

The result is

$$S(e^{j\omega T}) = (1/T)\sum_{n=-\infty}^{\infty} S(j[\omega - (2\pi/T)n]) = (1/T)\sum_{n=-\infty}^{\infty} S(j[\omega - n\omega_s]) \tag{1.19a}$$

where $\omega_s = (2\pi/T)$ is the sampling frequency expressed in radians per second. An alternate form for the expression of (1.19a) is

$$S(e^{j\omega'}) = (1/T)\sum_{n=-\infty}^{\infty} S(j[(\omega' - n2\pi)/T]) \tag{1.19b}$$

where $\omega' = \omega T$ is the normalized DT frequency axis expressed in radians. Note that $S(e^{j\omega T}) = S(e^{j\omega'})$ consists of an infinite number of replicas of the CT spectrum $S(j\omega)$, positioned at intervals of $(2\pi/T)$ on the ω axis (or at intervals of 2π on the ω' axis), as illustrated in Fig. 1.8. Note that if $S(j\omega)$ is band limited with a bandwidth ω_c, and if T is chosen sufficiently small so that $\omega_s > 2\omega_c$, then the DT spectrum is a copy of $S(j\omega)$ (scaled by $1/T$) in the baseband. The limiting case of $\omega_s = 2\omega_c$ is called the Nyquist sampling frequency. Whenever a CT signal is sampled at or above the Nyquist rate, no aliasing distortion occurs (i.e., the baseband spectrum does not overlap with the higher-order replicas) and the CT signal can be exactly recovered from its samples by extracting the baseband spectrum of $S(e^{j\omega'})$ with an ideal low-pass filter that recovers the original CT spectrum by removing all spectral replicas outside the baseband and scaling the baseband by a factor of T.

1.5 The Discrete Fourier Transform

To obtain the discrete Fourier transform (DFT) the continuous frequency domain of the DTFT is sampled at N points uniformly spaced around the unit circle in the z-plane, i.e., at the points

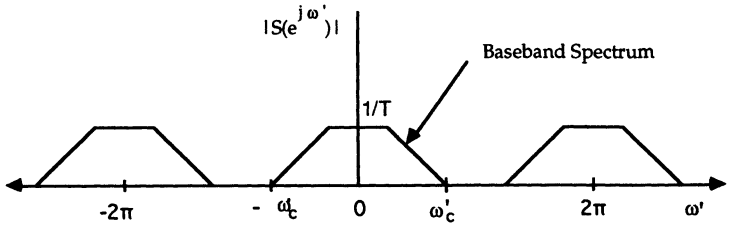

FIGURE 1.8: Illustration of the relationship between the CT and DT spectra.

$\omega_k = (2\pi k/N), k = 0, 1, \ldots, N-1$. The result is the DFT pair defined by (1.20a) and (1.20b). The signal $s[n]$ is either a finite length sequence of length N, or it is a periodic sequence with period N.

$$S[k] = \sum_{n=0}^{N-1} s[n]e^{-j2\pi kn/N} \qquad k = 0, 1, \ldots, N-1 \tag{1.20a}$$

$$s[n] = (1/N)\sum_{k=0}^{N-1} S[k]e^{j2\pi kn/N} \qquad n = 0, 1, \ldots, N-1 \tag{1.20b}$$

Regardless of whether $s[n]$ is a finite length or periodic sequence, the DFT treats the N samples of $s[n]$ as though they are one period of a periodic sequence. This is an important feature of the DFT, and one that must be handled properly in signal processing to prevent the introduction of artifacts. Important properties of the DFT are summarized in Table 1.5. The notation $((k))_N$ denotes k modulo N, and $R_N[n]$ is a rectangular window such that $R_N[n] = 1$ for $n = 0, \ldots, N-1$, and $R_N[n] = 0$ for $n < 0$ and $n \geq N$. The transform relationship given by (1.20a) and (1.20b) is also valid when $s[n]$ and $S[k]$ are periodic sequences, each of period N. In this case n and k are permitted to range over the complete set of real integers, and $S[k]$ is referred to as the discrete Fourier series (DFS). The DFS is developed by some authors as a distinct transform pair in its own right [6]. Whether the DFT and the DFS are considered identical or distinct is not very important in this discussion. The important point to be emphasized here is that the DFT treats $s[n]$ as though it were a single period of a periodic sequence, and all signal processing done with the DFT will inherit the consequences of this assumed periodicity.

1.5.1 Properties of the Discrete Fourier Series

Most of the properties listed in Table 1.5 for the DFT are similar to those of the z-transform and the DTFT, although some important differences exist. For example, property 5 (time-shifting property), holds for *circular* shifts of the finite length sequence $s[n]$, which is consistent with the notion that the DFT treats $s[n]$ as one period of a periodic sequence. Also, the multiplication of two DFTs results in the **circular convolution** of the corresponding DT sequences, as specified by property 7. This latter property is quite different from the **linear convolution** property of the DTFT. Circular convolution is the result of the assumed periodicity discussed in the previous paragraph. Circular convolution is simply a linear convolution of the periodic extensions of the finite sequences being convolved, in which each of the finite sequences of length N defines the structure of one period of the periodic extensions.

For example, suppose one wishes to implement a digital filter with finite impulse response (FIR)

TABLE 1.5 Properties of the DFT

Finite-Length Sequence (Length N)	N-Point DFT (Length N)
1. $x[n]$	$X[k]$
2. $x_1[n], x_2[n]$	$X_1[k], X_2[k]$
3. $ax_1[n] + bx_2[n]$	$aX_1[k] + bX_2[k]$
4. $X[n]$	$Nx[((-k))_N]$
5. $x[((n_m))_N]$	$W_N^{km} X[k]$
6. $W_N^{-ln} x[n]$	$X[((k-l))_N]$
7. $\sum_{m=0}^{N-1} x_1(m)x_2[((n_m))_N]$	$X_1[k]X_2[k]$
8. $x_1[n]x_2[n]$	$\frac{1}{N}\sum_{l=0}^{N-1} X_1(l)X_2[((k-l)_N]$
9. $x^*[n]$	$X^*[((-k))_N]$
10. $x^*[((-n))_N]$	$X^*[k]$
11. $\mathrm{Re}\{x[n]\}$	$X_{ep}[k] = \frac{1}{2}\{X[((k))_N] + K^*[((-k))_N]\}$
12. $j\mathrm{Im}\{x[n]\}$	$X_{op}[k] = \frac{1}{2}\{X[((k))_N] - X^*[((-k))_N]\}$
13. $x_{ep}[n] = \frac{1}{2}\{x[n] + x^*[((-n))_N]\}$	$\mathrm{Re}\{X[k]\}$
14. $x_{op}[n] = \frac{1}{2}\{x[n] - x^*[((-n))_N]\}$	$j\mathrm{Im}\{X[k]\}$
Properties 15–17 apply only when $x[n]$ is real	
15. Symmetry properties	$\begin{cases} X[k] = X^*[((-k))_N] \\ \mathrm{Re}\{X[k]\} = \mathrm{Re}\{X[((-k))_N]\} \\ \mathrm{Im}\{X[k]\} = -\mathrm{Im}\{X[((-k))_N]\} \\ \lvert X[k]\rvert = \lvert X[((-k))_N]\rvert \\ \measuredangle\{X[k]\} = -\measuredangle\{X[((-k))_N]\} \end{cases}$
16. $x_{ep}[n] = \frac{1}{2}\{x[n] + x[((-n))_N]\}$	$\mathrm{Re}\{X[k]\}$
17. $x_{op}[n] = \frac{1}{2}\{x[n] - x[((-n))_N]\}$	$j\mathrm{Im}\{X[k]\}$

$h[n]$. The output $y(n)$ in response to input $s[n]$ is given by

$$y[n] = \sum_{k=0}^{N-1} h[k]s[n-k] \tag{1.21}$$

where $y(n)$ is obtained by transforming $h[n]$ and $s[n]$ into $H[k]$ and $S[k]$ using the DFT, multiplying the transforms point-wise to obtain $Y[k] = H[k]S[k]$, and then using the inverse DFT to obtain $y[n] = \mathrm{DFT}^{-1}\{Y[k]\}$. If $s[n]$ is a finite sequence of length M, then the results of the circular convolution implemented by the DFT will correspond to the desired linear convolution if the block length of the DFT, N_{DFT}, is chosen sufficiently large so that $N_{\mathrm{DFT}} \geq N + M - 1$ and both $h[n]$ and $s[n]$ are padded with zeroes to form blocks of length N_{DFT}.

1.5.2 Fourier Block Processing in Real-Time Filtering Applications

In some practical applications either the value of M is too large for the memory available, or $s[n]$ may not actually be finite in length, but rather a continual stream of data samples that must be processed by a filter at real-time rates. Two well-known algorithms are available that partition $s[n]$ into smaller blocks and process the individual blocks with a smaller-length DFT: (1) overlap-save partitioning and (2) overlap-add partitioning. Each of these algorithms is summarized below.

Overlap-Save Processing

In this algorithm N_{DFT} is chosen to be some convenient value with $N_{\mathrm{DFT}} > N$. The signal $s[n]$ is partitioned into blocks which are of length N_{DFT} and which overlap by $N-1$ data points. Hence, the kth block is $s_k[n] = s[n + k(N_{\mathrm{DFT}} - N + 1)]$, $n = 0, \ldots, N_{\mathrm{DFT}} - 1$. The filter impulse

response is augmented with $N_{\mathrm{DFT}} - N$ zeroes to produce

$$h_{\mathrm{pad}}[n] = \begin{bmatrix} h[n], & n = 0, \ldots, N-1 \\ 0, & n = N, \ldots, N_{\mathrm{DFT}} - 1 \end{bmatrix} \tag{1.22}$$

The DFT is then used to obtain $Y_{\mathrm{pad}}[n] = \mathrm{DFT}\{h_{\mathrm{pad}}[n]\} \cdot \mathrm{DFT}\{s_k[n]\}$, and $y_{\mathrm{pad}}[n] = \mathrm{IDFT}\{Y_{\mathrm{pad}}[n]\}$. From the $y_{\mathrm{pad}}[n]$ array the values that correctly correspond to the linear convolution are saved; values that are erroneous due to wrap-around error caused by the circular convolution of the DFT are discarded. The kth block of the filtered output is obtained by

$$y_k[n] = \begin{bmatrix} y_{\mathrm{pad}}[n], & n = N-1, \ldots, N_{\mathrm{DFT}} - 1 \\ 0, & n = 0, \ldots, N-2 \end{bmatrix} \tag{1.23}$$

For the overlap-save algorithm, each time a block is processed there are $N_{\mathrm{DFT}} - N + 1$ points saved and $N - 1$ points discarded. Each block moves forward by $N_{\mathrm{DFT}} - N + 1$ data points and overlaps the previous block by $N - 1$ points.

Overlap-Add Processing

This algorithm is similar to the previous one except that the kth input block is defined as

$$s_k[n] = \begin{bmatrix} s[n + kL], & n = 0, \ldots, L-1 \\ 0, & n = L, \ldots, N_{\mathrm{DFT}} - 1 \end{bmatrix} \tag{1.24}$$

where $L = N_{\mathrm{DFT}} - N + 1$. The filter function $h_{\mathrm{pad}}[n]$ is augmented with zeroes, as before, to create $h_{\mathrm{pad}}[n]$, and the DFT processing is executed as before. In each block $y_{\mathrm{pad}}[n]$ that is obtained at the output the first $N-1$ points are erroneous, the last $N-1$ points are erroneous, and the middle $N_{\mathrm{DFT}} - 2(N-1)$ points correctly correspond to the linear convolution. However, if the last $N-1$ points from block k are overlapped with the first $N-1$ points from block $k+1$ and added pairwise, correct results corresponding to linear convolution are also obtained from these positions. Hence, after this addition the number of correct points produced per block is $N_{\mathrm{DFT}} - N + 1$, which is the same as that for the overlap-save algorithm. The overlap-add algorithm requires approximately the same amount of computation as the overlap-save algorithm, although the addition of the overlapping portions of blocks is extra. This feature, together with the added delay of waiting for the next block to be finished before the previous one is complete, has resulted in more popularity for the overlap-save algorithm in practical applications.

Block filtering algorithms make it possible to efficiently filter continual data streams in real time because the fast Fourier transform (FFT) algorithm can be used to implement the DFT, thereby minimizing the total computation time and permitting reasonably high overall data rates. However, block filtering generates data in bursts, i.e., a delay occurs during which no filtered data appear, and then an entire block is suddenly generated. In real-time systems buffering must be used. The block algorithms are particularly effective for filtering very long sequences of data that are prerecorded on magnetic tape or disk.

1.5.3 Fast Fourier Transform Algorithms

The DFT is typically implemented in practice with one of the common forms of the FFT algorithm. The FFT is not a Fourier transform in its own right, but simply a computationally efficient algorithm that reduces the complexity of computing the DFT from order $\{N^2\}$ to order $\{N \log_2 N\}$. When N is large, the computational savings provided by the FFT algorithm is so great that the FFT makes real-time DFT analysis practical in many situations that would be entirely impractical without

it. Fast Fourier transform algorithms abound, including decimation-in-time (D-I-T) algorithms, decimation-in-frequency (D-I-F) algorithms, bit-reversed algorithms, normally ordered algorithms, mixed-radix algorithms (for block lengths that are not powers of 2), prime factor algorithms, and Winograd algorithms [7]. The D-I-T and the D-I-F radix-2 FFT algorithms are the most widely used in practice. Detailed discussions of various FFT algorithms can be found in [3, 6, 7], and [10].

The FFT is easily understood by examining the simple example of $N = 8$. The FFT algorithm can be developed in numerous ways, all of which deal with a nested decomposition of the summation operator of (1.20a). The development presented here is called an **algebraic development** of the FFT because it follows straightforward algebraic manipulation. First, the summation indices (k, n) in (1.20a) are expressed as explicit binary integers, $k = k_2 4 + k_1 2 + k_0$ and $n = n_2 4 + n_1 2 + n_0$, where k_i and n_i are bits that take on the values of either 0 or 1. If these expressions are substituted into (1.20a), all terms in the exponent that contain the factor $N = 8$ can be deleted because $e^{-j2\pi l} = 1$ for any integer l. Upon deleting such terms and regrouping the remaining terms, the product nk can be expressed in either of two ways:

$$nk = (4k_0)n_2 + (4k_1 + 2k_0)n_1 + (4k_2 + 2k_1 + k_0)n_0 \quad (1.25a)$$

$$nk = (4n_0)k_2 + (4n_1 + 2n_0)k_1 + (4n_2 + 2n_1 + n_0)k_0 \quad (1.25b)$$

Substituting (1.25a) into (1.20a) leads to the D-I-T FFT, whereas substituting (1.25b) leads to the D-I-F FFT. Only the D-I-T FFT is discussed further here. The D-I-F and various related forms are treated in detail in [6].

The D-I-T FFT decomposes into $\log_2 N$ stages of computation, plus a stage of bit reversal,

$$x_1[k_0, n_1, n_0] = \sum_{n_2=0}^{1} s[n_2, n_1, n_0] W_8^{4k_0 n_2} \quad \text{(stage 1)} \quad (1.26a)$$

$$x_2[k_0, k_1, n_0] = \sum_{n_1=0}^{1} x[k_0, n_1, n_0] W_8^{(4k_1+2k_0)n_2} \quad \text{(stage 2)} \quad (1.26b)$$

$$x_3[k_0, k_1, k_2] = \sum_{n_0=0}^{1} x[k_0, k_1, n_0] W_8^{(4k_2+2k_1+k_0)n_0} \quad \text{(stage 3)} \quad (1.26c)$$

$$S[k_2, k_1, k_0] = x_3[k_0, k_1, k_2] \quad \text{(bit reversal)} \quad (1.26d)$$

In each summation above one of the n_i is summed out of the expression, while at the same time a new k_i is introduced. The notation is chosen to reflect this. For example, in stage 3, n_0 is summed out, k_2 is introduced as a new variable, and n_0 is replaced by k_2 in the result. The last operation, called bit reversal, is necessary to correctly locate the frequency samples $X[k]$ in the memory. It is easy to show that if the samples are paired correctly, an **in-place computation** can be done by a sequence of butterfly operations. The term "in-place" means that each time a butterfly is to be computed, a pair of data samples is read from memory, and the new data pair produced by the butterfly calculation is written back into the memory locations where the original pair was stored, thereby overwriting the original data. An in-place algorithm is designed so that each data pair is needed for only one butterfly, and thus the new results can be immediately stored on top of the old in order to minimize memory requirements.

For example, in stage 3 the $k = 6$ and $k = 7$ samples should be paired, yielding a "butterfly" computation that requires one complex multiply, one complex add, and one subtract:

$$x_3(1,1,0) = x_2(1,1,0) + W_8^3 x_2(1,1,1) \quad (1.27a)$$

$$x_3(1,1,1) = x_2(1,1,0) - W_8^3 x_2(1,1,1) \quad (1.27b)$$

Samples $x_2(6)$ and $x_2(7)$ are read from the memory, the butterfly is executed on the pair, and $x_3(6)$ and $x_3(7)$ are written back to the memory, overwriting the original values of $x_2(6)$ and $x_2(7)$. In general, $N/2$ butterflies are found in each stage and there are $\log_2 N$ stages, so the total number of butterflies is $(N/2)\log_2 N$. Because one complex multiplication per butterfly is the maximum, the total number of multiplications is bounded by $(N/2)\log_2 N$ (some of the multiplies involve factors of unity and should not be counted).

Figure 1.9 shows the signal flow graph of the D-I-T FFT for $N = 8$. This algorithm is referred to

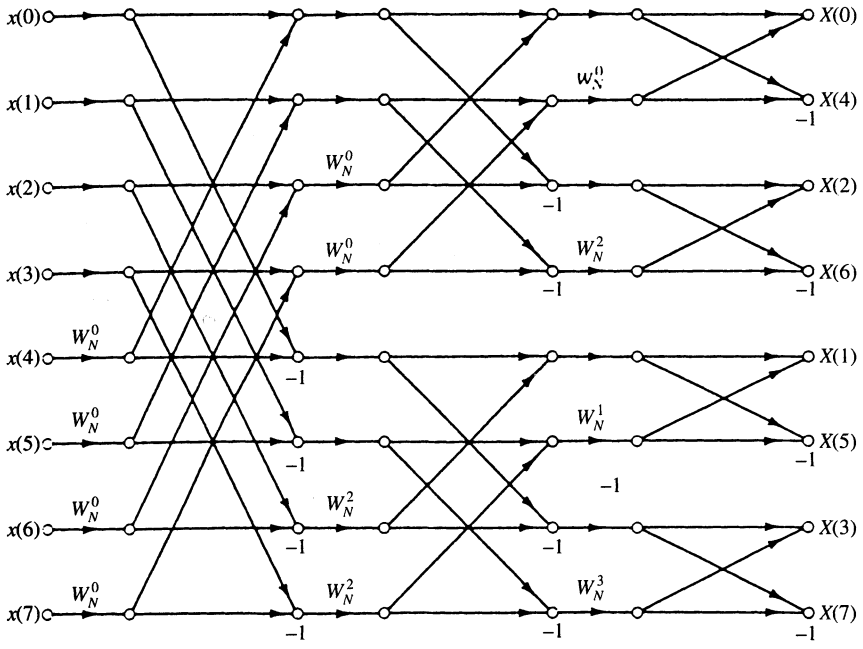

FIGURE 1.9: D-I-T FFT algorithm with normally ordered inputs and bit-reversed outputs.

as an in-place FFT with normally ordered input samples and bit-reversed outputs. Minor variations that include both bit-reversed inputs and normally ordered outputs and non-in-place algorithms with normally ordered inputs and outputs are possible. Also, when N is not a power of 2, a mixed-radix algorithm can be used to reduce computation. The mixed-radix FFT is most efficient when N is highly composite, i.e., $N = p_1^{r_1} p_2^{r_2} \cdots p_L^{r_L}$, where the p^i are small prime numbers and the r^i are positive integers. It can be shown that the order of complexity of the mixed radix FFT is order $\{N(r_1(p_1-1)+r_2(p_2-1)+\cdots+r_L(p_L-1)\}$. Because of the lack of uniformity of structure among stages, this algorithm has not received much attention for hardware implementation. However, the mixed-radix FFT is often used in software applications, especially for processing data recorded in laboratory experiments in which it is not convenient to restrict the block lengths to be powers of 2. Many advanced FFT algorithms, such as higher-radix forms, the mixed-radix form, the prime-factor

algorithm, and the Winograd algorithm are described in [9]. Algorithms specialized for real-valued data reduce the computational cost by a factor of two. A radix-2 D-I-T FFT program, written in C language, is listed in Table 1.6.

1.6 Family Tree of Fourier Transforms

It is now possible to illustrate the functional relationships among the various forms of Fourier transforms that have been discussed in the previous sections. The family tree of CT Fourier transform is shown in Fig. 1.10, where the most general, and consequently the most powerful, Fourier transform is the classical complex Fourier transform (or equivalently, the bilateral Laplace transform). Note that the complex Fourier transform is identical to the bilateral Laplace transform, and it is at this level that the classical Laplace transform and Fourier transform techniques become identical. Each special member of the CT Fourier family is obtained by impressing certain constraints on the general form, thereby producing special transforms that are simpler and more useful in practical problems where the constraints are met.

The analogous family of DT Fourier techniques is presented in Fig. 1.11, in which the bilateral z-transform is analogous to the complex Fourier transform, the unilateral z-transform is analogous to the classical (one-sided) Laplace transform, the DTFT is analogous to the classical Fourier (CT) transform, and the DFT is analogous to the classical (CT) Fourier series.

1.7 Selected Applications of Fourier Methods

1.7.1 Fast Fourier Transform in Spectral Analysis

An FFT program is often used to perform spectral analysis on signals that are sampled and recorded as part of laboratory experiments, or in certain types of data acquisition systems. Several issues must be addressed when spectral analysis is performed on (sampled) analog waveforms that are observed over a finite interval of time.

Windowing

The FFT treats the block of data as though it were one period of a periodic sequence. If the underlying waveform is not periodic, then harmonic distortion may occur because the periodic waveform created by the FFT may have sharp discontinuities at the boundaries of the blocks. This effect is minimized by removing the mean of the data (it can always be reinserted) and by windowing the data so the ends of the block are smoothly tapered to zero. A good rule of thumb is to taper 10% of the data on each end of the block using either a cosine taper or one of the other common windows shown in Table 1.7. An alternate interpretation of this phenomenon is that the finite length observation has already windowed the true waveform with a rectangular window that has large spectral sidelobes (see Table 1.7). Hence, applying an additional window results in a more desirable window that minimizes frequency domain distortion.

Zero Padding

An improved spectral analysis is achieved if the block length of the FFT is increased. This can be done by taking more samples within the observation interval, increasing the length of the observation interval, or augmenting the original data set with zeroes. First, it must be understood that the finite observation interval results in a fundamental limit on the spectral resolution, even before the signals are sampled. The CT rectangular window has a $(\sin x)/x$ spectrum, which is convolved with the true spectrum of the analog signal. Therefore, the frequency resolution is limited by the width of the mainlobe in the $(\sin x)/x$ spectrum, which is inversely proportional to the length

TABLE 1.6 An In-Place D-I-T FFT Program in C Language

```
/*****************************************************
*     fft: in-place radix-2 DFT of a complex input
*
*     input:
*        n:      length of FFT: must be a power of two
*        m:      n = 2**m
*     input/output:
*        x:      float array of length n with real part of data
*        y:      float array of length n with image part of data
*****************************************************/
fft(n,m,x,y)
tnt       n,m;
float     x[ ], y[ ]:
{
          int    i,j,k,nl,n2:
          float  c,s,e,a,t1,t2;

          j = 0;                        /*BIT-REVERSE  */
          n2 = n/2;
          for (i=1; 1 < n-1; i++)             /*bit-reverse counter */
          {
           nl = n1/2;
           while ( j >= n1)
            {
             j = j - n1;
             nl = n1/2;
            }
           j = j + nl;
           if (i < j)                          /*swap data */
            {
             t1 = x[i]; x[i] = x[j]; x[j] = t1;
             t1 = y[i]; y[i] = y[j]; y[j] = t1;
            }
           }
          n1 = 0; n2 = 1;               /* FFT */
          for (i = 0; i < m; i++)             /*state loop */
           {
            n1 = n2;  n2 = n2 + n2;
            e = -6.283185307179586/n2;
            a = 0.0;

            for (j=0; j < n1; j++)            /*flight loop */
             {
              c = cos(a); s=sin (a);
              a = a + e;

              for (k=j; k < n; k=k+n2)        /*butterfly loop */
               {
                t1 = c*x[k+n1] - s*y[k+n1];
                t2 = s*x[k+n1] + c*y[k+n1];
                x[k+n1] = x[k] - t1;
                y[k+n1] = y[k] - t2;
                x[k] = x[k] + t1;
                y[k] = y[k] + t2;
               }
             }
           }
           return;
}
```

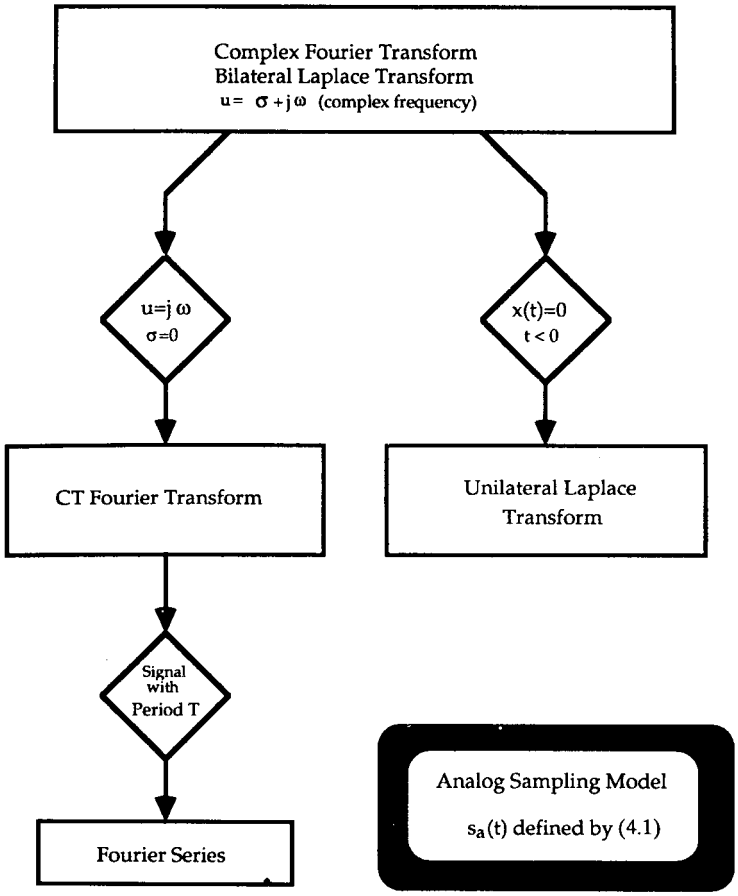

FIGURE 1.10: Relationships among CT Fourier concepts.

of the observation interval. Sampling causes a certain degree of aliasing, although this effect can be minimized by sampling at a high enough rate. Therefore, lengthening the observation interval increases the fundamental resolution limit, while taking more samples within the observation interval minimizes aliasing distortion and provides a better definition (more sample points) on the underlying spectrum.

Padding the data with zeroes and computing a longer FFT does give more frequency domain points (improved spectral resolution), but it does not improve the fundamental limit, nor does it alter the effects of aliasing error. The resolution limits are established by the observation interval and the sampling rate. No amount of zero padding can improve these basic limits. However, zero padding is a useful tool for providing more spectral definition, i.e., it allows a better view of the (distorted) spectrum that results once the observation and sampling effects have occurred.

Leakage and the Picket Fence Effect

An FFT with block length N can accurately resolve only frequencies $\omega_k = (2\pi/N)k, k = 0, \ldots, N-1$ that are integer multiples of the fundamental $\omega_1 = (2\pi/N)$. An analog waveform that is sampled and subjected to spectral analysis may have frequency components between the harmonics. For example, a component at frequency $\omega_{k+1/2} = (2\pi/N)(k+1/2)$ will appear scattered throughout

TABLE 1.7 Common Window Functions

Name	Function	Peak Side-Lobe Amplitude (dB)	Mainlobe Width	Minimum Stopband Attenuation (dB)
Rectangular	$\omega(n) = 1. \quad 0 \le n \le N-1$	-13	$4\pi/N$	-21
Bartlett	$\omega(n) = \begin{cases} 2/N, & 0 \le n \le (N-1)/2 \\ 22n/N, & (N-1)/2 \le n \le N-1 \end{cases}$	-25	$8\pi/N$	-25
Hanning	$\omega(n) = (1/2)[1 - \cos(2\pi n/N)]$ $0 \le n \le N-1$	-31	$8\pi/N$	-44
Hamming	$\omega(n) = 0.54 - 0.46\cos(2\pi n/N)$, $0 \le n \le N-1$	-43	$8\pi/N$	-53
Backman	$\omega(n) = 0.42 - 0.5\cos(2\pi n/N) + 0.08\cos(4\pi n/N), \quad 0 \le n \le N-1$	-57	$12\pi/N$	-74

the spectrum. The effect is illustrated in Fig. 1.12 for a sinusoid that is observed through a rectangular window and then sampled at N points. The **picket fence effect** means that not all frequencies can be seen by the FFT. Harmonic components are seen accurately, but other components "slip through the picket fence" while their energy is "leaked" into the harmonics. These effects produce artifacts in the spectral domain that must be carefully monitored to assure that an accurate spectrum is obtained from FFT processing.

1.7.2 Finite Impulse Response Digital Filter Design

A common method for designing FIR digital filters is by use of windowing and FFT analysis. In general, window designs can be carried out with the aid of a hand calculator and a table of well-known window functions. Let $h[n]$ be the impulse response that corresponds to some desired frequency response, $H(e^{j\omega})$. If $H(e^{j\omega})$ has sharp discontinuities, such as the low-pass example shown in Fig. 1.13, then $h[n]$ will represent an infinite impulse response (IIR) function. The objective is to time limit $h[n]$ in such a way as to not distort $H(e^{j\omega})$ any more than necessary. If $h[n]$ is simply truncated, a ripple (Gibbs phenomenon) occurs around the discontinuities in the spectrum, resulting in a distorted filter (Fig. 1.13).

Suppose that $w[n]$ is a window function that time limits $h[n]$ to create an FIR approximation, $h'[n]$; i.e., $h'[n] = w[n]h[n]$. Then if $W(e^{j\omega})$ is the DTFT of $w[n]$, $h'[n]$ will have a Fourier transform given by $H'(e^{j\omega}) = W(e^{j\omega}) * H(e^{j\omega})$, where $*$ denotes convolution. Thus, the ripples in $H'(e^{j\omega})$ result from the sidelobes of $W(e^{j\omega})$. Ideally, $W(e^{j\omega})$ should be similar to an impulse so that $H'(e^{j\omega})$ is approximately equal to $H(e^{j\omega})$.

Special Case. Let $h[n] = \cos n\omega_0$, for all n. Then $h[n] = w[n]\cos n\omega_0$, and

$$H'(e^{j\omega}) = (1/2)W(e^{j(\omega+\omega_0)}) + (1/2)W(e^{j(\omega-\omega_0)}) \tag{1.28}$$

as illustrated in Fig. 1.14. For this simple class, the center frequency of the bandpass is controlled by ω_0, and both the shape of the bandpass and the sidelobe structure are strictly determined by the choice of the window. While this simple class of FIRs does not allow for very flexible designs, it is a simple technique for determining quite useful low-pass, bandpass, and high-pass FIRs.

General Case. Specify an ideal frequency response, $H(e^{j\omega})$, and choose samples at selected values of ω. Use a long inverse FFT of length N' to find $h'[n]$, an approximation to $h[n]$, where if N is the desired length of the final filter, then $N' \gg N$. Then use a carefully selected window to truncate $h'[n]$ to obtain $h[n]$ by letting $h[n] = \omega[n]h'[n]$. Finally, use an FFT of length N' to find $H'(e^{j\omega})$. If $H'(e^{j\omega})$ is a satisfactory approximation to $H(e^{j\omega})$, the design is finished. If not, choose a new $H(e^{j\omega})$ or a new $w[n]$ and repeat. Throughout the design procedure it is important to choose $N' = kN$, with k an integer that is typically in the range of 4 to 10. Because this design technique is a

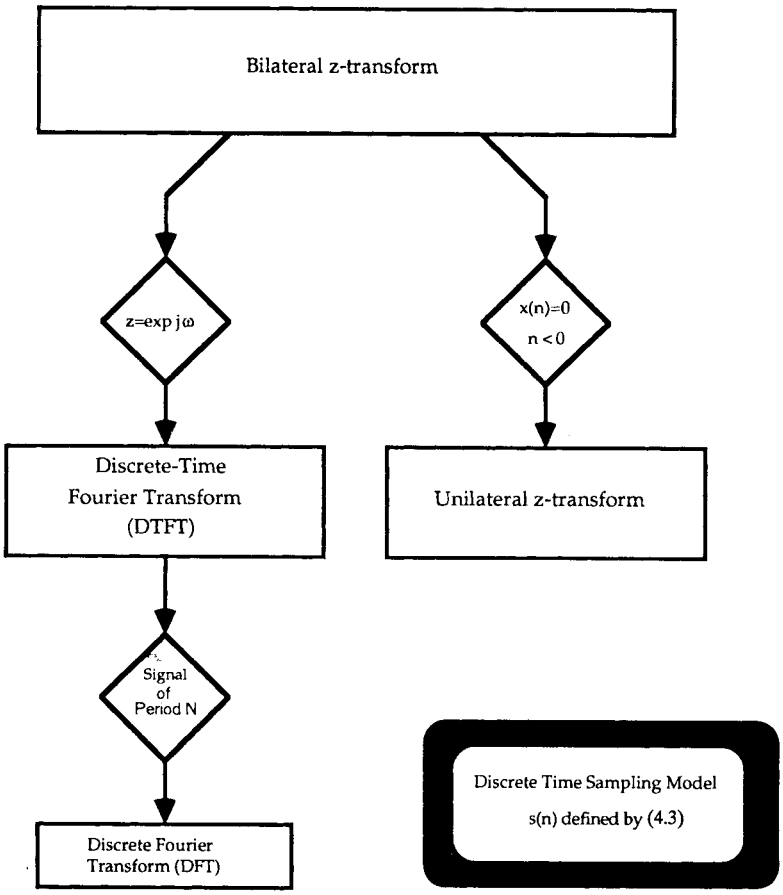

FIGURE 1.11: Relationships among DT concepts.

trial and error procedure, the quality of the result depends to some degree on the skill and experience of the designer. Table 1.7 lists several well-known window functions that are often useful for this type of FIR filter design procedure.

1.7.3 Fourier Analysis of Ideal and Practical Digital-to-Analog Conversion

From the relationship characterized by (1.19b) and illustrated in Fig. 1.8, CT signal $s(t)$ can be recovered from its samples by passing $s_a(t)$ through an ideal lowpass filter that extracts only the baseband spectrum. The ideal lowpass filter, shown in Fig. 1.15, is a zero-phase CT filter whose magnitude response is a constant of value T in the range $-\pi < \omega' \leq \pi$, and zero elsewhere. The impulse response of this "reconstruction filter" is given by $h(t) = T\text{sinc}((\pi/T)t)$, where $\text{sinc}x = (\sin x)/x$. The reconstruction can be expressed as $s(t) = h(t) * s_a(t)$, which, after some mathematical manipulation, yields the following classical reconstruction formula

$$s(t) = \sum_{n=-\infty}^{\infty} s(nT)\text{sinc}((\pi/T)(t - nT)) \tag{1.29}$$

Note that the signal $s(t)$ is exactly recovered from its samples only if an infinite number of terms is

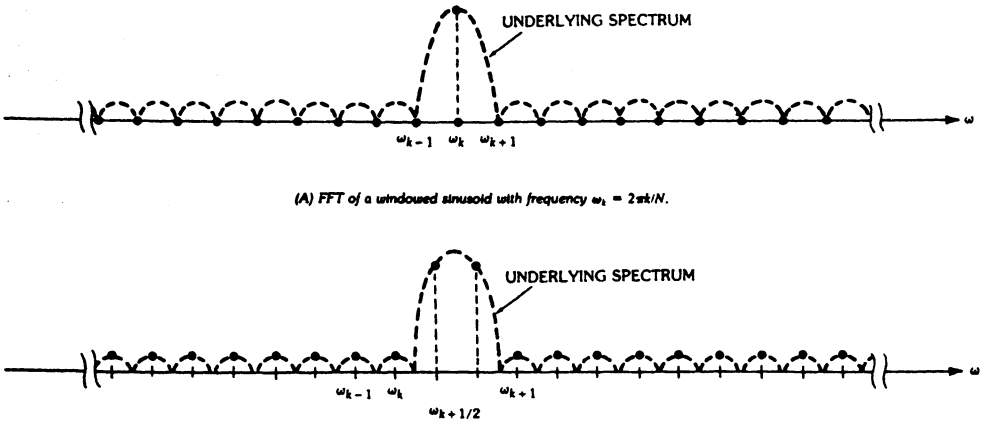

FIGURE 1.12: Illustration of leakage and the picket-fence effects.

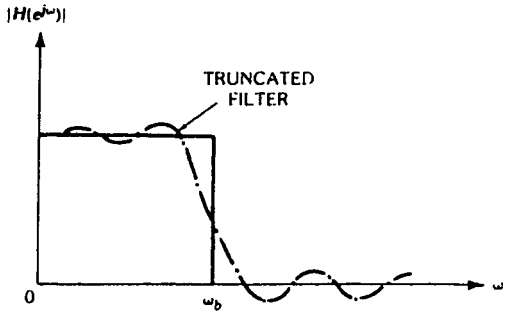

FIGURE 1.13: Gibbs effect in a low-pass filter caused by truncating the impulse response.

included in the summation of (1.29). However, good approximation of $s(t)$ can be obtained with only a finite number of terms if the lowpass reconstruction filter $h(t)$ is modified to have a finite interval of support, i.e., if $h(t)$ is nonzero only over a finite time interval. The reconstruction formula of (1.29) is an important result in that it represents the inverse of the sampling operation. By this means Fourier transform theory establishes that as long as CT signals are sampled at a sufficiently high rate, the information content contained in $s(t)$ can be represented and processed in either a CT or DT format. Fourier sampling and reconstruction theory provides the theoretical mechanism for translation between one format or the other without loss of information.

A CT signal $s(t)$ can be perfectly recovered from its samples using (1.29) as long as the original sampling rate was high enough to satisfy the Nyquist sampling criterion, i.e., $\omega_s > 2\omega_B$. If the sampling rate does not satisfy the Nyquist criterion the adjacent periods of the analog spectrum will overlap, causing a distorted spectrum. This effect, called **aliasing distortion**, is rather serious because it cannot be corrected easily once it has occurred. In general, an analog signal should always be prefiltered with an CT low-pass filter prior to sampling so that aliasing distortion does not occur.

Figure 1.16 shows the frequency response of a fifth-order elliptic analog low-pass filter that meets industry standards for prefiltering speech signals. These signals are subsequently sampled at an 8-kHz sampling rate and transmitted digitally across telephone channels. The band-pass ripple is less than ± 0.01 dB from DC up to the frequency 3.4 kHz (too small to be seen in Fig. 1.16), and the stopband

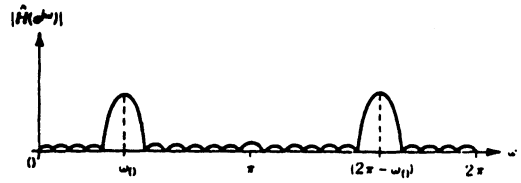

FIGURE 1.14: Design of a simple bandpass FIR filter by windowing.

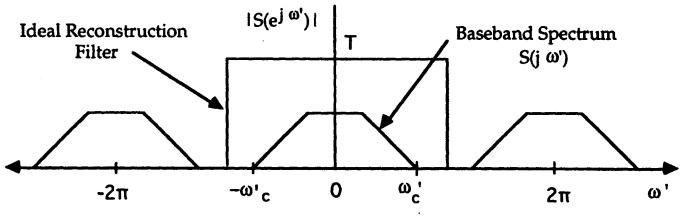

FIGURE 1.15: Illustration of ideal reconstruction.

rejection reaches at least -32.0 dB at 4.6 kHz and remains below this level throughout the stopband.

Most practical systems use digital-to-analog converters for reconstruction, which results in a staircase approximation to the true analog signal, i.e.,

$$\hat{s}(t) = \sum_{n=-\infty}^{\infty} s(nT)\{u(t-nT) - u[t-(n+1)]\}, \tag{1.30}$$

where $\hat{s}(t)$ denotes the reconstructed approximation to $s(t)$, and $u(t)$ denotes a CT unit step function. The approximation $\hat{s}(t)$ is equivalent to a result obtained by using an approximate reconstruction filter of the form

$$H_a(j\omega) = 2Te^{-j\omega T/2}\sin c(\omega T/2) \tag{1.31}$$

The approximation $\hat{s}(t)$ is said to contain "sin x/x distortion," which occurs because $H_a(j\omega)$ is not an ideal low-pass filter. $H_a(j\omega)$ distorts the signal by causing a droop near the passband edge, as well as by passing high-frequency distortion terms which "leak" through the sidelobes of $H_a(j\omega)$. Therefore, a practical digital to analog converter is normally followed by an analog postfilter

$$H_p(j\omega) = \begin{bmatrix} H_a^{-1}(j\omega), & 0 \le |\omega| < \pi/T \\ 0, & \omega \text{ otherwise} \end{bmatrix} \tag{1.32}$$

which compensates for the distortion and produces the correct $\hat{s}(t)$, i.e., the correctly constructed CT output. Unfortunately, the postfilter $H_p(j\omega)$ cannot be implemented perfectly, and, therefore, the actual reconstructed signal always contains some distortion in practice that arises from errors in approximating the ideal postfilter. Figure 1.17 shows a digital processor, complete with analog-to-digital and digital-to-analog converters, and the accompanying analog pre- and postfilters necessary for proper operation.

1.8 Summary

This chapter presented many different Fourier transform concepts for both continuous time (CT) and discrete time (DT) signals and systems. Emphasis was placed on illustrating how these various

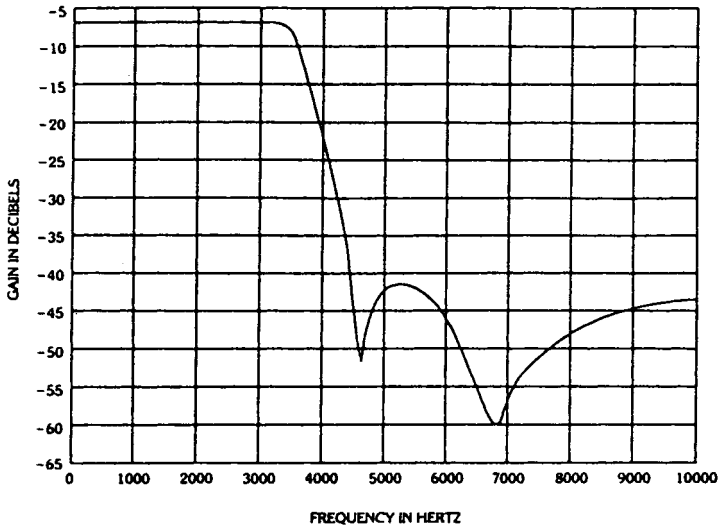

FIGURE 1.16: A fifth-order elliptic analog anti-aliasing filter used in the telecommunications industry with an 8-kHz sampling rate.

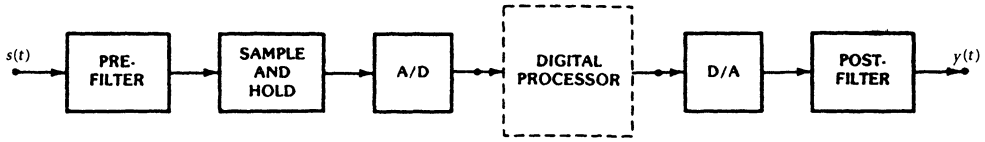

FIGURE 1.17: Analog pre- and postfilters required at the analog to digital and digital to analog interfaces.

forms of the Fourier transform relate to one another, and how they are all derived from more general complex transforms, the complex Fourier (or bilateral Laplace) transform for CT, and the bilateral z-transform for DT. It was shown that many of these transforms have similar properties which are inherited from their parent forms, and that a parallel hierarchy exists among Fourier transform concepts in the CT and the DT worlds. Both CT and DT sampling models were introduced as a means of representing sampled signals in these two different "worlds," and it was shown that the models are equivalent by virtue of having the same Fourier spectra when transformed into the Fourier domain with the appropriate Fourier transform. It was shown how Fourier analysis properly characterizes the relationship between the spectra of a CT signal and its DT counterpart obtained by sampling. The classical reconstruction formula was obtained as an outgrowth of this analysis. Finally, the discrete Fourier transform (DFT), the backbone for much of modern digital signal processing, was obtained from more classical forms of the Fourier transform by simultaneously discretizing the time and frequency domains. The DFT, together with the remarkable computational efficiency provided by the fast Fourier transform (FFT) algorithm, has contributed to the resounding success that engineers and scientists have experienced in applying digital signal processing to many practical scientific problems.

References

[1] VanValkenburg, M.E., *Network Analysis*, 3rd ed., Englewood Cliffs, NJ: Prentice-Hall, 1974.

[2] Oppenheim, A.V., Willsky, A.S., and Young, I.T., *Signals and Systems*, Englewood Cliffs, NJ: Prentice-Hall, 1983.

[3] Bracewell, R.N., *The Fourier Transform*, 2nd ed., New York: McGraw-Hill, 1986.

[4] Oppenheim, A.V. and Schafer, R.W., *Discrete-Time Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1989.

[5] Jenkins, W.K. and Desai, M.D., The discrete-frequency Fourier transform, *IEEE Trans. Circuits Syst.*, vol. CAS-33, no. 7, pp. 732–734, July 1986.

[6] Oppenheim, A.V. and Schafer, R.W., *Digital Signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.

[7] Blahut, R.E., *Fast Algorithms for Digital Signal Processing*, Reading, MA: Addison-Wesley, 1985.

[8] Deller, J.R., Jr., Tom, Dick, and Mary discover the DFT, *IEEE Signal Processing Mag.*, vol. 11, no. 2, pp. 36–50, Apr. 1994.

[9] Burrus, C.S. and Parks, T.W., *DFT/FFT and Convolution Algorithms*, New York: John Wiley and Sons, 1985.

[10] Brigham, E.O., *The Fast Fourier Transform*, Englewood Cliffs, NJ: Prentice-Hall, 1974.