CHAPTER 9

DSP-Based Testing

In Chapter 8, we briefly touched on the advantages of DSP-based testing before beginning our review of sampling theory. In this chapter, we will take a more detailed look at digital signal processing and the power it gives us in testing mixed-signal devices. Although a full study of DSP is beyond the scope of this book, many good texts have been written on the subject.[1–3] In this chapter, we will review the basics of DSP, limiting our discussion to discrete (i.e., sampled) waveforms of finite length. More specifically, this chapter will review the concept of a Fourier series and its application to continuous-time and discrete-time periodic signals. As much of the work related to DSP-based testing involve discrete waveforms, we shall turn our attention to the discrete-time Fourier series (DTFS), discrete Fourier transform (DFT), and their implementation using the fast-Fourier transform (FFT) algorithm. The interplay between time and frequency will also be described. In some test situations, coherency is difficult to achieve, in such situation windowing is used. A detail look at windowing will be provided, from a description of the terminology used to its impact on spectral frequency leakage. Finally, we shall end the chapter with a look at the inverse-FFT and its application to spectral filtering, such as for noise reduction.

9.1 ADVANTAGES OF DSP-BASED TESTING

9.1.1 Reduced Test Time

Coherent DSP-based testing gives us several advantages over traditional measurement techniques. The first advantage of DSP-based testing is reduced test time. Since we can create and measure signals with multiple frequencies at the same time, we can perform many parametric measurements in parallel. If we need to test the frequency response and phase response of a filter, for example, we can perform a series of gain and phase measurements at a dozen or so frequencies simultaneously.

DSP-based testing allows us to send all the filter test frequencies through the device under test (DUT) at the same time. Once we have collected the DUT's output response using a digitizer

or capture memory, DSP allows us to separate each test tone in the output waveform from all the other test tones. We can then calculate a separate gain and phase measurement at each frequency without running many separate tests. We can also measure noise and distortion at the same time that we measure gain and phase shift, further reducing test time.

9.1.2 Separation of Signal Components

Separation of signal components from one another gives us a second huge advantage over non-DSP-based measurements. We can isolate noise and distortion components from one another and from the test tones. This allows for much more accurate and repeatable measurements of the primary test tones and their distortion components.

Using coherent test tones, we are always guaranteed that all the distortion components will fall neatly into separate Fourier spectral bins rather than being smeared across many bins (as is the case with noncoherent signal components). DSP-based testing is a major advantage in the elimination of errors and poor repeatability.

9.1.3 Advanced Signal Manipulations

In this chapter we will see how DSP-based testing allows us to manipulate digitized output waveforms to achieve a variety of results. We can perform interpolations between the samples to achieve better time resolution. We can apply mathematical filters to emphasize or diminish certain frequency components. We can remove noise from signals to achieve better accuracy. All these techniques are made possible by the application of digital signal processing to sampled DUT outputs.

9.2 DIGITAL SIGNAL PROCESSING

9.2.1 DSP and Array Processing

Before we embark on a review of digital signal processing, let us take some time to define exactly what a digital signal is. Then we shall look at the different ways we can process digital signals. There is a slight semantic difference between digital signal processing and array processing. An array, or vector, is a set of similar numbers, such as a record of all the heights of the students in a class. An example of array processing would be the calculation of the average height of the students. A digital signal is also a set of numbers (i.e., voltage samples), but the samples are ordered in time. Digital signal processing is thus a subset of array processing, since it is limited to mathematical operations on *time-ordered* samples. However, since most arrays processed on an ATE tester are in fact digital signals, most automated test equipment (ATE) languages categorize all array processing operations under the umbrella of DSP. So much for semantics!

ATE digital signal processing is often accomplished on a specialized computer called an *array processor*. However, tester computers have become faster over the years, making a separate array processor unnecessary in some of the newer testers. Depending on the sophistication of the tester's operating system, the presence or absence of a separate array processor may not even be apparent to the user.

There are many array processing functions that prove useful in mixed-signal test engineering. One simple example is the averaging function. The average of a series of samples is equivalent to the DC offset of the signal. If we want to measure the RMS noise of a digitized waveform, we must first remove the DC offset. Otherwise the DC offset would add to the RMS calculation, resulting in an erroneous noise measurement. We can compute the DC offset of the digitized waveform using

an averaging function. Subtracting the offset from each sample in the waveform eliminates the DC offset. In MATLAB, we might accomplish the DC removal using the following simple routine:

```
% DC Removal Routine
  % Calculate the DC offset (average) of a waveform, x
     average = sum(x)/length(x);
  % Subtract offset from each waveform sample
     x = x-average;
```

This DC removal routine can be considered a digital signal processing operation, although a dedicated array processor is not needed to perform the calculations. Fortunately, we are able to take advantage of many built-in array processing operations in mixed-signal testers rather than writing them from scratch. These built-in operations are streamlined by the ATE vendor to allow the fastest possible processing time on the available computation hardware. For example, some of the computations may take place in parallel using multiple array processors, thus saving test time. Although tester languages vary widely, the following pseudocode is representative of typical ATE array processing operations:

```
float offset, waveform [256];
offset = average (waveform [1 to 256]);
waveform = waveform – offset;
```

Digital signal processing operations are somewhat more complicated than the simple array processing functions listed. Operations such as the discrete Fourier transform (DFT), fast Fourier transform (FFT), inverse FFT, and filtering operations will require a little more explanation.

9.2.2 Fourier Analysis of Periodic Signals

The tremendous advantage of DSP-based testing is the ability to measure many different frequency components simultaneously, minimizing test time. For example, we can apply a seven-tone multitone signal such as the one in Figure 8.26 to a low-pass filter. The filter will amplify or attenuate each frequency component by a different amount according to the filter's transfer function. It may also shift the phase of each frequency component.

It is easy to see how we can apply a multitone signal to the input of a filter. We simply compute the sample set in Figure 8.26 and apply it through an arbitrary waveform generator (AWG) to the input of the filter. A digitizer can then be used to collect samples from the output of the filter. But how do we then extract the amplitude of the individual frequency components from the composite signal at the filter's output?

The answer is a Fourier analysis. It is a mathematical method that gives us the power to split a composite signal into its individual frequency components. It is based on the fact that we can describe any signal in either the time domain or the frequency domain. Fourier analysis allows us to convert time-domain signal information into a description of a signal as a function of frequency. Fourier analysis allows us to convert a signal from the time domain to the frequency domain and back again without losing any information about the signal in either domain. This powerful capability is at the heart of mixed-signal testing.

Jean Baptiste Joseph Fourier was a clever French mathematician who developed Fourier analysis for the study of heat transfer in solid bodies. His technique was published in 1822. Almost 200 years later, the importance of Fourier's work in today's networked world is astounding. Applications of his method extend to cellular telephones, disk drives, speech recognition systems, radar systems, and mixed-signal testing, to name just a few.

9.2.3 The Trigonometric Fourier Series

Fourier's initial work showed how a mathematical series of sine and cosine terms could be used to analyze heat conduction problems. This became known as the *trigonometric Fourier series* and was probably the first systematic application of a trigonometric series to a problem. At the time of his death in 1830, he had extended his methods to include the Fourier integral, leading to the concept of a *Fourier transform*. The Fourier transform is largely applied to the analysis of aperiodic signals. Mixed-signal test engineering is primarily concerned with the discrete form of the Fourier series and, specifically, coherent sample sets. Therefore, we shall limit our discussion mainly to the Fourier series.

Let $x(t)$ denote a periodic signal with period T such that it satisfies

$$x(t) = x(t+T) \tag{9.1}$$

for all values of $-\infty < t < \infty$. Using a trigonometric Fourier series expansion of this signal, we are able to resolve the signal into an infinite sum of cosine and sine terms according to

$$x(t) = a_0 + \sum_{k=1}^{\infty}\left[a_k \cos\left(k \times 2\pi f_o t\right) + b_k \sin\left(k \times 2\pi f_o t\right)\right] \tag{9.2}$$

The first term in the series a_0 represents the DC or average component of $x(t)$. The coefficients a_k and b_k represents the amplitudes of the cosine and sine terms, respectively. They are commonly referred to as the spectral or Fourier coefficients and are determined from the following integral equations:

$$\begin{aligned} a_0 &= \frac{1}{T}\int_0^T x(t)\,dt \\ a_k &= \frac{2}{T}\int_0^T x(t)\cos\left(k\frac{2\pi}{T}t\right)dt, \qquad k \geq 1 \\ b_k &= \frac{2}{T}\int_0^T x(t)\sin\left(k\frac{2\pi}{T}t\right)dt, \qquad k \geq 1 \end{aligned} \tag{9.3}$$

The frequency of each cosine and sine term is an integer multiple of the fundamental frequency $f_o = 1/T$, referred to as a harmonic. For instance, the quantity kf_o represents the kth harmonic of the fundamental frequency.

A more compact form of the trigonometric Fourier series is

$$x(t) = \sum_{k=0}^{\infty} c_k \cos\left(k \times 2\pi f_o t - \phi_k\right) \tag{9.4}$$

where

$$c_k = \sqrt{a_k{}^2 + b_k{}^2} \tag{9.5}$$

and

$$\phi_k = \begin{cases} \phi_o & \text{if } a_k \geq 0, b_k \geq 0 \\ -\phi_o & \text{if } a_k \geq 0, b_k < 0 \\ \pi - \phi_o & \text{if } a_k < 0, b_k \geq 0 \\ \phi_o - \pi & \text{if } a_k < 0, b_k < 0 \end{cases} \qquad \text{where} \quad \phi_o = \tan^{-1}\left(\frac{|b_k|}{|a_k|}\right) \tag{9.6}$$

This result follows from the trigonometric identity: $\cos(A+B)=\cos(A)\cos(B)-\sin(A)\sin(B)$. We should also note that the phase term is subtracted from the argument of the cosine function. This is known as a *phase lag* representation rather than a *phase lead*. Phase lag is defined as the phase angle corresponding to a zero crossing of some signal minus the zero-crossing phase angle of a reference signal, that is, output phase angle minus reference phase angle. Also, to avoid any numerical difficulty associated with the arctangent function, we define the phase term as $-\pi \le \phi \le \pi$ using a four-quadrant definition.

We prefer the magnitude and phase form representation of the Fourier series as it lends itself more directly to graphical form. Specifically, vertical lines can be drawn at discrete frequency points corresponding to $0, f_o, 2f_o, 3f_o$, and so on, with their heights proportional to the amplitudes of the corresponding frequency components, that is, c_k versus kf_o. Such a plot conveys the *magnitude spectrum*

EXAMPLE 9.1

Determine the Fourier series representation of the 5-V, 10-kHz clock signal shown in Figure 9.1 and plot the corresponding magnitude and phase spectrum.

Solution:

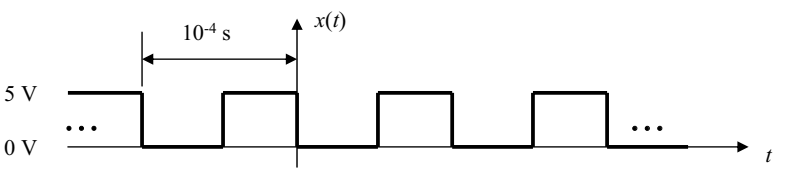
The spectral coefficients are determined according to Eq. (9-3) as follows

$$a_0 = \frac{1}{10^{-4}}\left(\int_0^{0.5\times10^{-4}} 0\,dt + \int_{0.5\times10^{-4}}^{10^{-4}} 5\,dt\right) = \frac{1}{10^{-4}}\,5\,\left(10^{-4} - 0.5\times10^{-4}\right) = 2.5$$

$$b_k = \frac{2}{10^{-4}}\left(\int_0^{0.5\times10^{-4}} 0\,\sin\left(k\,10^4\cdot 2\pi\,t\right)dt + \int_{0.5\times10^{-4}}^{10^{-4}} 5\,\sin\left(k\cdot 10^4\cdot 2\pi\,t\right)dt\right)$$

$$= -\frac{5}{k\pi}\,\cos\left(k\,10^{-4}\cdot 2\pi\,t\right)\Big|_{0.5\times10^{-4}}^{10^{-4}} = \frac{5}{k\pi}\left[\cos(k\pi) - \cos(k\cdot 2\pi)\right] = \frac{5}{k\pi}\left[(-1)^k - 1\right]$$

Figure 9.1. 10-kHz clock signal.



For even values of k, the term $[(-1)^k - 1\ ] = 0$, hence the clock signal consists of only odd harmonics. Therefore,

$$b_k = \begin{cases} 0 & k \text{ even} \\ -\dfrac{10}{k\pi} & k \text{ odd} \end{cases}$$

The Fourier series representation then becomes

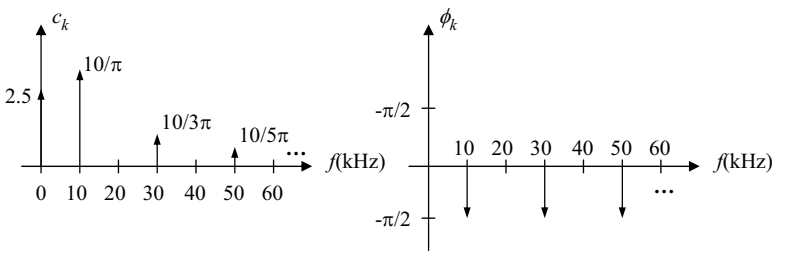
$$x(t) = 2.5 \; - \sum_{k=1,\, k \text{ odd}}^{\infty} \frac{10}{k\pi} \sin\left(k \cdot 2\pi \cdot 10^4 \cdot t\right)$$

or, when written in the form of Eq. (9–4), becomes

$$x(t) = 2.5 \; + \sum_{k=1,\, k \text{ odd}}^{\infty} \frac{10}{k\pi} \cos\left(k \cdot 2\pi \cdot 10^4 \cdot t + \frac{\pi}{2}\right)$$

as $\cos(x + \pi/2) = -\sin(x)$ and $\cos(x - \pi/2) = \sin(x)$. The corresponding magnitude and phase spectrum plots are then as shown in Figure 9.2.

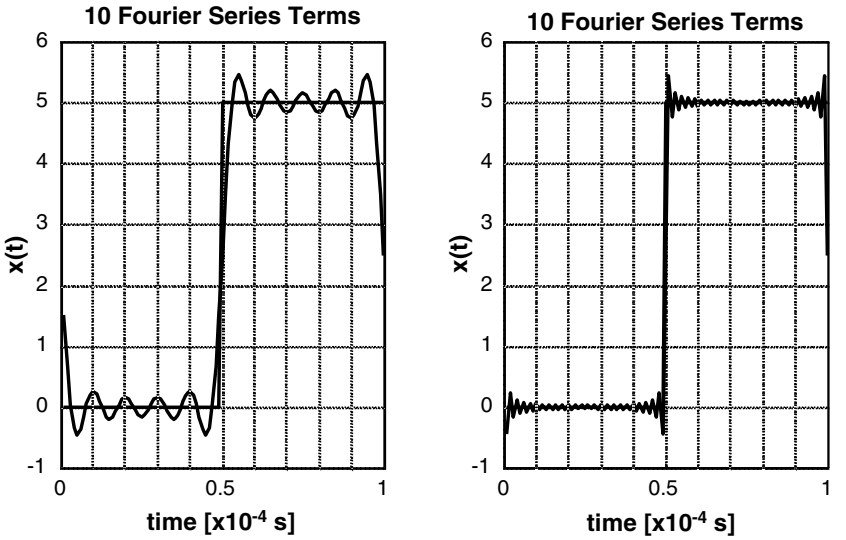
Figure 9.2. Magnitude and phase spectrum plots of 10-kHz clock signal.



of $x(t)$. Likewise, a *phase spectrum* can be drawn in the exact same manner except that the heights of the vertical lines are proportional to the phases ϕ_k of the corresponding frequency components. The following example illustrates these two plots for a 5-V, 10-kHz clock signal.

It is instructive to view the behavior of the Fourier series representation for the clock signal of the previous example consisting of 10 and 50 terms. This we show in Figure 9.3 superimposed on one period of the clock signal. Clearly, as the number of terms in the series is increased, the Fourier series representation more closely resembles the clock signal. Some large amplitude oscillatory behavior occurs at the jump discontinuity. This is known as *Gibb's phenomenon* and is a result of truncating terms in the Fourier series representation.

Figure 9.3. Observing the Fourier series representation of a clock signal for: (a) 10 terms, and (b) 50 terms. Superimposed on the plot is the original clock signal.



9.2.4 The Discrete-Time Fourier Series

As mentioned previously, sampling is an important step in mixed-signal testing. To understand the effects of sampling on an arbitrary periodic signal, consider sampling the general form of the Fourier series representation given in Eq. (9.2). Assuming that the sampling period is T_s, we can write

$$x(t)\big|_{t=nT_s} = a_0 + \sum_{k=1}^{\infty}\left[a_k \cos\left(k \cdot 2\pi f_o \cdot nT_s\right) + b_k \sin\left(k \cdot 2\pi f_o \cdot nT_s\right)\right] \tag{9.7}$$

As $F_s = 1/T_s$, we can rewrite Eq. (9.7) as

$$x(t)\big|_{t=nT_s} = a_0 + \sum_{k=1}^{\infty}\left\{a_k \cos\left[k\left(\frac{2\pi f_o}{F_s}\right)n\right] + b_k \sin\left[k\left(\frac{2\pi f_o}{F_s}\right)n\right]\right\} \tag{9.8}$$

In this form we see that the frequency of the fundamental is a fraction f_o/F_s of 2π. Furthermore, this term no longer has units of radians per second but rather just radians. To distinguish this from our regular notion of frequency, it is commonly referred to as a *normalized frequency,* because it has been normalized by the sampling frequency F_s. Except for the time reference T_s on the left-hand side of the equation, the information about the time scale is lost. This is further complicated by the fact that one usually uses the shorthand notation

$$x[n] = x(t)\big|_{t=nT_s} \tag{9.9}$$

and eliminates the time reference altogether. The discrete-time signal $x[n]$ is simply a sequence of numbers with no reference to the underlying time scale. Hence, the original samples cannot be reconstructed without knowledge of the original sampling frequency. Therefore, a sampling period or frequency must always be associated with a discrete-time signal, $x[n]$.

For much of the work in this textbook, we are concerned with coherent sampling sets, that is, $UTP = NT_s$, or, equivalently, $f_o = 1/UTP = F_f = F_s/N$. Equation (9.8) can then be reduced to

$$x[n] = a_0 + \sum_{k=1}^{\infty} \left\{ a_k \cos\left[k\left(\frac{2\pi}{N}\right) n \right] + b_k \sin\left[k\left(\frac{2\pi}{N}\right) n \right] \right\} \tag{9.10}$$

where the frequency of the fundamental reduces to 1/*N*, albeit normalized by F_s.

As the original continuous-time signal *x*(*t*) is periodic and sampled coherently, *x*[*n*] will be periodic with respect to the sample index *n*, according to

$$x[n] = x[n+N] \tag{9.11}$$

Because the roles of *n* and *k* are interchangeable in the arguments of the sine and cosine terms of Eq. (9.10), it suggests that *x*[*n*] will also repeat with index *k* over the period *N*. Through a detailed trigonometric development, we can rewrite the infinite series given by Eq. (9.10) as the sum of *N/2* trigonometric terms as follows (here it is assumed *N* is even, as is often the case in testing applications):

$$x[n] = \tilde{a}_0 + \sum_{k=1}^{N/2-1} \left\{ \tilde{a}_k \cos\left[k\left(\frac{2\pi}{N}\right) n \right] + \tilde{b}_k \sin\left[k\left(\frac{2\pi}{N}\right) n \right] \right\} + \tilde{a}_{N/2} \cos(\pi n) \tag{9.12}$$

where

$$\tilde{a}_0 = \sum_{m=0}^{\infty} a_{0+mN}, \qquad \tilde{a}_k = \sum_{m=0}^{\infty} \left(a_{k+mN} + a_{N-k+mN} \right), \qquad \tilde{a}_{N/2} = \sum_{m=0}^{\infty} a_{N/2+mN}$$
$$\tilde{b}_k = \sum_{m=0}^{\infty} \left(b_{k+mN} - b_{N-k+mN} \right) \tag{9.13}$$

Here $\tilde{a}_k$ represents the amplitude of the cosine component located at the *k*th Fourier spectral bin. Likewise, $\tilde{b}_k$ is the amplitude of the sine component located at the *k*th Fourier spectral bin. Of course, $\tilde{a}_0$ represents the DC or average value of the sample set. The equations of Eq. (9.13) represent the sum of all aliases terms that arise during the sampling process. Since we are assuming that the original continuous-time signal is frequency band-limited, the sums in Eq. (9.13) will converge to finite values.

As before, Eq. (9.12) can be written in a more compact form using magnitude and phase notation as

$$x[n] = \sum_{k=0}^{N/2} \tilde{c}_k \cos\left[k\left(\frac{2\pi}{N}\right) n - \tilde{\phi}_k \right] \tag{9.14}$$

where

$$\tilde{c}_k = \sqrt{\tilde{a}_k^2 + b_k^2} \tag{9.15}$$

and

$$\tilde{\phi}_k = \begin{cases} \tilde{\phi}_o, & \text{if } \tilde{a}_k \geq 0, \tilde{b}_k \geq 0 \\ -\tilde{\phi}_o, & \text{if } \tilde{a}_k \geq 0, \tilde{b}_k < 0 \\ \pi - \tilde{\phi}_o, & \text{if } \tilde{a}_k < 0, \tilde{b}_k \geq 0 \\ \tilde{\phi}_o - \pi, & \text{if } \tilde{a}_k < 0, \tilde{b}_k < 0 \end{cases} \quad \text{where} \quad \tilde{\phi}_o = \tan^{-1}\left(\frac{|\tilde{b}_k|}{|\tilde{a}_k|}\right) \tag{9.16}$$

Equation (9.14) can then be graphically displayed using a magnitude and phase spectrum plot.

The importance of the above expressions cannot be understated, because it relates the spectrum of a sampled signal to the original continuous-time signal. Furthermore, it suggests that a discrete time signal has a spectrum that consists of, at most, *N/2* unique frequencies. Moreover, these frequencies are all harmonically related to the primitive or fundamental frequency of *1/N* radians. This representation is known as a *discrete-time Fourier series (DTFS)* representation of $x[n]$ written in trigonometric form. It will form the basis for all the computer analysis in this text.

Exercises

9.1. Find the trigonometric Fourier series representation of a square-wave $x(t)$ having a period of 2 s and whose behavior is described by

$$x(t) = \begin{cases} +1 & \text{if } 0 < t < 1 \\ -1 & \text{if } 1 < t < 2 \end{cases}$$

ANS. $x(t) = \frac{1}{2} - \frac{2}{\pi}\left(\cos\frac{\pi}{2}t - \frac{1}{3}\cos\frac{3\pi}{2}t + \frac{1}{5}\cos\frac{5\pi}{2}t - + \cdots\right)$

9.2. Express the Fourier series in Exercise 9.1 using cosine terms only.

ANS. $x(t) = \frac{4}{\pi}\left[\cos\left(\pi t - \frac{\pi}{2}\right) + \frac{1}{3}\cos\left(3\pi t - \frac{\pi}{2}\right) + \frac{1}{5}\cos\left(5\pi t - \frac{\pi}{2}\right) + \cdots\right]$

9.3. Find the Fourier series representation of a square-wave $x(t)$ having a period of 4 s and whose behavior is described by

$$x(t) = \begin{cases} 1 & \text{if } -1 < t < 1 \\ 0 & \text{if } 1 < t < 3 \end{cases}$$

ANS. $x(t) = \frac{1}{2} - \frac{2}{\pi}\left(\cos\frac{\pi}{2}t - \frac{1}{3}\cos\frac{3\pi}{2}t + \frac{1}{5}\cos\frac{5\pi}{2}t - + \cdots\right)$

EXAMPLE 9.2

Calculate the DTFS representation of the 10-kHz clock signal of Example 9.1 when sampled at a 100-kHz sampling rate.

Solution:
With a 100-kHz sampling rate, 10 points per period will be collected in one period of the 10-kHz clock signal (i.e., $N = 10$). Using the equations for the spectral coefficients in (9.13), together with the Fourier series result of Example 9.1, we find that the $\tilde{a}_k$ coefficients are as follows:

$$\tilde{a}_0 = \sum_{m=0}^{\infty} a_{0+10m} = a_0 = 2.5$$
$$\tilde{a}_k = \sum_{m=0}^{\infty} \left(a_{k+10m} + a_{10-k+10m}\right) = 0, \qquad k \in \{1,2,3,4\}$$
$$\tilde{a}_5 = \sum_{m=0}^{\infty} a_{5+10m} = 0$$

Subsequently, the $\tilde{b}_k$ coefficients for $k = 1, 2, \ldots, 4$ (by definition, $\tilde{b}_0 = 0$ and $\tilde{b}_5 = 0$) are found as follows:

$$\tilde{b}_k = \sum_{m=0}^{\infty} \left(b_{k+10m} - b_{10-k+10m}\right) = \begin{cases} 0, & k \text{ even} \\ -\dfrac{10}{\pi} \displaystyle\sum_{m=0}^{\infty} \left(\frac{1}{(k+10m)} - \frac{1}{(10-k+10m)} \right), & k \text{ odd} \end{cases}$$

Here the summation involves the difference between two harmonic progressions where no closed-form summation formulas are known to exist. Subsequently, a numerical routine was written that summed the first 100 terms of this series. The result is

$$\tilde{b}_1 = -3.07516, \quad \tilde{b}_2 = 0, \quad \tilde{b}_3 = -0.72528, \quad \tilde{b}_4 = 0$$

According to Eq. (9.12), the discrete-time Fourier series representation for the clock signal then becomes

$$x[n] = 2.5 - 3.07516 \sin\left[1\left(\frac{2\pi}{10}\right)n\right] - 0.72528 \sin\left[3\left(\frac{2\pi}{10}\right)n\right]$$

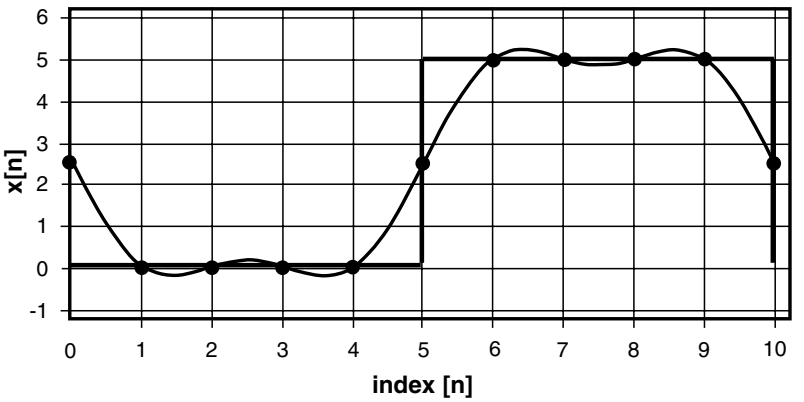
or, using magnitude and phase notation, we write

$$x[n] = 2.5 + 3.07516 \cos\left[1\left(\frac{2\pi}{10}\right)n + \frac{\pi}{2}\right] + 0.72528 \cos\left[3\left(\frac{2\pi}{10}\right)n + \frac{\pi}{2}\right]$$

It is important for the reader to verify the samples produced by the above discrete-time Fourier series. Evaluating $x[n]$ for one complete period (i.e., $n = \{0, 1, 2, \ldots, 9\}$), we obtain $x = \{2.5, 0.0027, 0.0017, 0.0017, 0.0027, 2.5, 4.9973, 4.9983, 4.9983, 4.9973\}$. As expected, all the samples

correspond quite closely with samples from the original signal, as shown in Figure 9.4. The small difference can be contributed to the error that results from including only 100 terms in the summation of the $\tilde{b}_k$ coefficients. Of particular interest is the value that the discrete-time Fourier series assigns to the waveform at the jump discontinuity. In general, the sample value at a jump discontinuity is ambiguous and undefined. Fourier analysis resolves this problem by assigning the sample value of the discontinuity as the midway point of the jump. In this particular case, the midpoint of each jump discontinuity is $(5-0)/2 = 2.5$.

Figure 9.4. Comparing the samples of a DTFS representation and the original clock signal. Also shown is the DTFS as a continuous function of *n*.



Also shown in the plot of Figure 9.4 is a graph of the DTFS representation as a continuous function of sample index, *n*. It is interesting to note that the samples are the intersection of this continuous-time function with the original clock signal.

When analyzing a waveform collected from a DUT we want to know the spectral coefficients $\tilde{a}_k$ and $\tilde{b}_k$ (or $\tilde{c}_k$ and $\tilde{\phi}_k$) so that we can see how much signal power is present at each frequency and deduce the DUT's overall performance. Thus far, we have only considered how to compute the spectral coefficients from a Fourier series expansion of a continuous-time waveform. As we shall see, there is a more direct way to compute the spectral coefficients of a discrete-time Fourier series from N samples of the continuous-time waveform. In fact, we shall outline two methods: one that highlights the nature of the problem in algebraic terms and the other involving a closed-form expression for the coefficients in terms of the sampled values.

To begin, let us consider that we have N samples, denoted $x[n]$ for $n = 0, 1, 2, \ldots, N-1$, Consider that each one of these samples must satisfy the discrete-time Fourier series expansion of Eq. (9.12). This is rather unlike the Fourier series expansion for a continuous-time signal that consists of an infinite number of trigonometric terms.

In practice, the summation must be limited to a finite number of terms, resulting in an approximation error. The discrete-time Fourier series, on the other hand, consists of only N trigonometric terms and, hence, there is no error in its representation. Therefore, we can write directly from Eq. (9.12) at sampling instant $n = 0$.

$$x[0]=\tilde{a}_0+\sum_{k=1}^{N/2-1}\left\{\tilde{a}_k\cos[0]+\tilde{b}_k\sin[0]\right\}+\tilde{a}_{N/2}\cos[0]=\tilde{a}_0+\tilde{a}_1+\cdots+\tilde{a}_{N/2-1}+\tilde{a}_{N/2} \quad \textbf{(9.17)}$$

Next, at $n = 1$, we write

$$\begin{aligned}x[1]&=\tilde{a}_0+\sum_{k=1}^{N/2-1}\left\{\tilde{a}_k\cos\left[k\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_k\sin\left[k\left(\frac{2\pi}{N}\right)\right]\right\}+\tilde{a}_{N/2}\cos[\pi]\\&=\tilde{a}_0+\tilde{a}_1\cos\left[\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_1\sin\left[\left(\frac{2\pi}{N}\right)\right]+\tilde{a}_2\cos\left[2\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_2\sin\left[2\left(\frac{2\pi}{N}\right)\right]+\cdots\\&\quad+\tilde{a}_{N/2-1}\cos\left[\left(\frac{N}{2}-1\right)\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_{N/2-1}\sin\left[\left(\frac{N}{2}-1\right)\left(\frac{2\pi}{N}\right)\right]+\tilde{a}_{N/2}\cos[\pi]\end{aligned} \quad \textbf{(9.18)}$$

Similarly, for $n = 2$, we write

$$\begin{aligned}x[2]&=\tilde{a}_0+\sum_{k=1}^{N/2-1}\left\{\tilde{a}_k\cos\left[2k\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_k\sin\left[2k\left(\frac{2\pi}{N}\right)\right]\right\}+\tilde{a}_{N/2}\cos(2\pi)\\&=\tilde{a}_0+\tilde{a}_1\cos\left[2\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_1\sin\left[2\left(\frac{2\pi}{N}\right)\right]+\tilde{a}_2\cos\left[4\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_2\sin\left[4\left(\frac{2\pi}{N}\right)\right]+\cdots\\&\quad+\tilde{a}_{N/2-1}\cos\left[2\left(\frac{N}{2}-1\right)\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_{N/2-1}\sin\left[2\left(\frac{N}{2}-1\right)\left(\frac{2\pi}{N}\right)\right]+\tilde{a}_{N/2}\cos(2\pi)\end{aligned} \quad \textbf{(9.19)}$$

Continuing for all remaining sampling instants, up to $n = N–1$, we can write the Nth equation as

$$\begin{aligned}x[N-1]&=\tilde{a}_0+\sum_{k=1}^{N/2-1}\left\{\tilde{a}_k\cos\left[(N-1)k\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_k\sin\left[(N-1)k\left(\frac{2\pi}{N}\right)\right]\right\}+\tilde{a}_{N/2}\cos[(N-1)\pi]\\&=\tilde{a}_0+\tilde{a}_1\cos\left[(N-1)\left(\frac{2\pi}{N}\right)\right]+\tilde{b}_1\sin\left[(N-1)\left(\frac{2\pi}{N}\right)\right]+\tilde{a}_2\cos\left[2(N-1)\left(\frac{2\pi}{N}\right)\right]\\&\quad+\tilde{b}_2\sin\left[2(N-1)\left(\frac{2\pi}{N}\right)\right]+\cdots+\tilde{a}_{N/2-1}\cos\left[\left(\frac{N}{2}-1\right)(N-1)\left(\frac{2\pi}{N}\right)\right]\\&\quad+\tilde{b}_{N/2-1}\sin\left[\left(\frac{N}{2}-1\right)(N-1)\left(\frac{2\pi}{N}\right)\right]+\tilde{a}_{N/2}\cos[(N-1)\pi]\end{aligned} \quad \textbf{(9.20)}$$

Finally, on observing the behavior of each one of these equations, we find that all the trigonometric terms have numerical values and the only unknown terms are the spectral coefficients. In essence, we have a system of N linear equations in N unknowns. Straightforward linear algebra can then be used to compute the spectral coefficients. For instance, if we define vectors

$$\mathrm{C}=\begin{bmatrix}a_0 & a_1 & b_1 & a_2 & b_2 & \cdots & a_{N-1}\end{bmatrix}$$

and

$$\mathrm{X}=\begin{bmatrix}x[0] & x[1] & x[2] & \cdots & x[N-1]\end{bmatrix}$$

together with matrix

$$W = \begin{bmatrix} 1 & \cos\left[(1)(0)\left(\frac{2\pi}{N}\right)\right] & \sin\left[(1)(0)\left(\frac{2\pi}{N}\right)\right] & \cdots & \sin\left[\left(\frac{N}{2}-1\right)(0)\left(\frac{2\pi}{N}\right)\right] & \cos\left[\left(\frac{N}{2}\right)(0)\left(\frac{2\pi}{N}\right)\right] \\ 1 & \cos\left[(1)(1)\left(\frac{2\pi}{N}\right)\right] & \sin\left[(1)(1)\left(\frac{2\pi}{N}\right)\right] & \cdots & \sin\left[\left(\frac{N}{2}-1\right)(1)\left(\frac{2\pi}{N}\right)\right] & \cos\left[\left(\frac{N}{2}\right)(1)\left(\frac{2\pi}{N}\right)\right] \\ 1 & \cos\left[(1)(2)\left(\frac{2\pi}{N}\right)\right] & \sin\left[(1)(2)\left(\frac{2\pi}{N}\right)\right] & \cdots & \sin\left[\left(\frac{N}{2}-1\right)(2)\left(\frac{2\pi}{N}\right)\right] & \cos\left[\left(\frac{N}{2}\right)(2)\left(\frac{2\pi}{N}\right)\right] \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & \cos\left[(1)(N-1)\left(\frac{2\pi}{N}\right)\right] & \sin\left[(1)(N-1)\left(\frac{2\pi}{N}\right)\right] & \cdots & \sin\left[\left(\frac{N}{2}-1\right)(N-1)\left(\frac{2\pi}{N}\right)\right] & \cos\left[\left(\frac{N}{2}\right)(N-1)\left(\frac{2\pi}{N}\right)\right] \end{bmatrix}$$

we can write

$$X = WC \tag{9.21}$$

Multiplying both sides by W^{-1}, the spectral coefficients are found as follows

$$C = W^{-1}X \tag{9.22}$$

The next example will illustrate this method on the clock signal samples from Example 9.2.

EXAMPLE 9.3

Consider from Example 9.2 that the clock signal samples are x = {2.5, 0.0, 0.0, 0.0, 0.0, 2.5, 5.0, 5.0, 5.0, 5.0}. Compute the spectral coefficients of the DTFS using linear algebra.

Solution:

Using the procedure described, we can write the following system of linear equations in matrix form:

$$\begin{bmatrix} 2.5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2.5 \\ 5 \\ 5 \\ 5 \\ 5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0.809 & 0.588 & 0.309 & 0.951 & -0.309 & 0.951 & -0.809 & 0.588 & -1 \\ 1 & 0.309 & 0.951 & -0.809 & 0.588 & -0.809 & -0.588 & 0.309 & -0.951 & 1 \\ 1 & -0.309 & 0.951 & -0.809 & -0.588 & 0.809 & -0.588 & 0.309 & 0.951 & -1 \\ 1 & -0.809 & 0.588 & 0.309 & -0.951 & 0.309 & 0.951 & -0.809 & -0.588 & 1 \\ 1 & -1.000 & 0.000 & 1.000 & 0.000 & -1.000 & 0.000 & 1.000 & 0.000 & -1 \\ 1 & -0.809 & -0.588 & 0.309 & 0.951 & 0.309 & -0.951 & -0.809 & 0.588 & 1 \\ 1 & -0.309 & -0.951 & -0.809 & 0.588 & 0.809 & 0.588 & 0.309 & -0.951 & -1 \\ 1 & 0.309 & -0.951 & -0.809 & -0.588 & -0.809 & 0.588 & 0.309 & 0.951 & 1 \\ 1 & 0.809 & -0.588 & 0.309 & -0.951 & -0.309 & -0.951 & -0.809 & -0.588 & -1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \\ a_3 \\ b_3 \\ a_4 \\ b_4 \\ a_5 \end{bmatrix}$$

Using the matrix routines available in MATLAB, we obtain the following spectral coefficients:

$$\begin{bmatrix} \tilde{\mathbf{a}}_0 \\ \tilde{\mathbf{a}}_1 \\ \tilde{\mathbf{b}}_1 \\ \tilde{\mathbf{a}}_2 \\ \tilde{\mathbf{b}}_2 \\ \tilde{\mathbf{a}}_3 \\ \tilde{\mathbf{b}}_3 \\ \tilde{\mathbf{a}}_4 \\ \tilde{\mathbf{b}}_4 \\ \tilde{\mathbf{a}}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{2.500} \\ \mathbf{0} \\ \mathbf{-3.078} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{-0.7265} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} \tilde{c}_0 \\ \tilde{\phi}_0 \\ \tilde{c}_1 \\ \tilde{\phi}_1 \\ \tilde{c}_2 \\ \tilde{\phi}_2 \\ \tilde{c}_3 \\ \tilde{\phi}_3 \\ \tilde{c}_4 \\ \tilde{\phi}_4 \\ \tilde{c}_5 \\ \tilde{\phi}_5 \end{bmatrix} = \begin{bmatrix} 2.500 \\ 0 \\ 3.078 \\ -\pi/2 \\ 0 \\ 0 \\ 0.7265 \\ -\pi/2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Exercises

9.4. Using the summation formulae in Eqs. (9.12) and (9.13), determine the DTFS representation of the Fourier series representation of $x(t)$ given in exercise 9.1. Use 10 points per period and limit the series to 100 terms. Also, express the result in magnitude and phase form.

ANS.
$$1.2301 \sin\left[\left(\frac{2\pi}{10}\right)n\right] + 0.2901 \sin\left[3\left(\frac{2\pi}{10}\right)n\right]$$
$$= 1.2301 \cos\left[\left(\frac{2\pi}{10}\right)n - \frac{\pi}{2}\right] + 0.2901 \cos\left[3\left(\frac{2\pi}{10}\right)n - \frac{\pi}{2}\right]$$

9.5. A sampled signal consists of the following four samples: {0.7071, 0.7071, –0.7071, –0.7071}. Using linear algebra, determine the DTFS representation for these samples. Also, express the result in magnitude and phase form.

ANS.
$$0.7071 \cos\left[\left(\frac{2\pi}{4}\right)n\right] + 0.7071 \sin\left[\left(\frac{2\pi}{4}\right)n\right] = 1.0 \cos\left[\left(\frac{2\pi}{4}\right)n - \frac{\pi}{4}\right]$$

9.6. A sampled signal consists of the following set of samples {0, 1, 2, 3, 2, 1}. Determine the DTFS representation for this sample set using the orthogonal basis method. Also, express the result in magnitude and phase form.

ANS.
$$1.5 - 1.333 \cos\left[\left(\frac{2\pi}{6}\right)n\right] - 0.1667 \cos\left[3\left(\frac{2\pi}{6}\right)n\right]$$
$$= 1.5 + 1.333 \cos\left[\left(\frac{2\pi}{6}\right)n - \pi\right] + 0.1667 \cos\left[3\left(\frac{2\pi}{6}\right)n - \pi\right]$$

On comparison with the results in Example 9.2, we see that they agree reasonably well (the small difference is attributed to the series truncation as explained in the last example). The latter set of magnitude and phase coefficients was derived using Eq. (9.15).

The preceding example highlights the fact that the spectral coefficients of the discrete-time Fourier series are determined by straightforward linear algebraic methods. Another method exists for finding the spectral coefficients and one that is much more insightful as it provides a closed-form solution for each spectral coefficient. To arrive at this solution, we need to consider the orthogonal property of cosines and sines. Specifically, consider the following set of orthogonal basis functions

$$\sum_{n=0}^{N-1}\cos\left[p\left(\frac{2\pi}{N}\right)n\right]\cos\left[k\left(\frac{2\pi}{N}\right)n\right]=\begin{cases}0 & \text{for } p\neq k\\ N/2 & \text{for } p=k\neq 0\\ N & \text{for } p=k=0\end{cases}$$

$$\sum_{n=0}^{N-1}\sin\left[p\left(\frac{2\pi}{N}\right)n\right]\sin\left[k\left(\frac{2\pi}{N}\right)n\right]=\begin{cases}0 & \text{for } p\neq k\\ N/2 & \text{for } p=k\neq 0\end{cases} \qquad \textbf{(9.23)}$$

$$\sum_{n=0}^{N-1}\sin\left[p\left(\frac{2\pi}{N}\right)n\right]\cos\left[k\left(\frac{2\pi}{N}\right)n\right]=0 \quad \text{for all } p$$

Armed with these identities, we multiply Eq. (9.12) with $\cos\left[k\left(\frac{2\pi}{N}\right)n\right]$ and sum n from 0 to N-1 on both sides to obtain*

$$\tilde{a}_k=\begin{cases}\frac{1}{N}\sum_{n=0}^{N-1}x[n]\cos\left[k\left(\frac{2\pi}{N}\right)n\right], & k=0,N/2\\ \frac{2}{N}\sum_{n=0}^{N-1}x[n]\cos\left[k\left(\frac{2\pi}{N}\right)n\right], & k=1,2,\ldots,N/2-1\end{cases} \qquad \textbf{(9.24)}$$

Likewise, we multiply Eq. (9.12) with $\sin\left[k\left(\frac{2\pi}{N}\right)n\right]$ and sum n from 0 to N-1 on both sides to obtain

$$\tilde{b}_k=\frac{2}{N}\sum_{n=0}^{N-1}x[n]\sin\left[k\left(\frac{2\pi}{N}\right)n\right], \quad k=1,2,\ldots,N/2-1 \qquad \textbf{(9.25)}$$

The details are left as an exercise for the reader in problem 9.7.

---

*In many textbooks, a single expression for a_k is usually written as was done for b_k. This is achieved by writing the discrete-time Fourier series as

$$x[n]=\frac{\tilde{a}_0}{2}+\sum_{k=1}^{N/2-1}\left\{\tilde{a}_k\cos\left[k\left(\frac{2\pi}{N}\right)n\right]+\tilde{b}_k\sin\left[k\left(\frac{2\pi}{N}\right)n\right]\right\}+\frac{\tilde{a}_{N/2}}{2}\cos(\pi n)$$

where the end terms in the series are scaled by a factor of 2.

EXAMPLE 9.4

Repeat Example 9.3 but compute the spectral coefficients of the DTFS using the orthogonal basis method.

Solution:
With x = {2.5, 0.0, 0.0, 0.0, 0.0, 2.5, 5.0, 5.0, 5.0, 5.0}, we can compute from Eq. (9.24) the following coefficients:

$$\tilde{a}_0 = \frac{1}{10}[2.5+0+0+0+0+2.5+5+5+5+5] = 2.5$$

$$\tilde{a}_1 = \frac{2}{10}\left\{2.5\cos\left[(1)\left(\frac{2\pi}{10}\right)(0)\right] + 2.5\cos\left[(1)\left(\frac{2\pi}{10}\right)(5)\right] + 5\cos\left[(1)\left(\frac{2\pi}{10}\right)(6)\right] + 5\cos\left[(1)\left(\frac{2\pi}{10}\right)(7)\right] + 5\cos\left[(1)\left(\frac{2\pi}{10}\right)(8)\right] + 5\cos\left[(1)\left(\frac{2\pi}{10}\right)(9)\right]\right\} = 0$$

Continuing, we find $\tilde{a}_2 = 0$, $\tilde{a}_3 = 0$, $\tilde{a}_4 = 0$, and $\tilde{a}_5 = 0$. Likewise, from Eq. (9.25) we have

$$\tilde{b}_1 = \frac{2}{10}\left\{2.5\sin\left[(1)\left(\frac{2\pi}{10}\right)(0)\right] + 2.5\sin\left[(1)\left(\frac{2\pi}{10}\right)(5)\right] + 5\sin\left[(1)\left(\frac{2\pi}{10}\right)(6)\right] + 5\sin\left[(1)\left(\frac{2\pi}{10}\right)(7)\right] + 5\sin\left[(1)\left(\frac{2\pi}{10}\right)(8)\right] + 5\sin\left[(1)\left(\frac{2\pi}{10}\right)(9)\right]\right\} = -3.077$$

$$\tilde{b}_2 = \frac{2}{10}\left\{2.5\sin\left[(2)\left(\frac{2\pi}{10}\right)(0)\right] + 2.5\sin\left[(2)\left(\frac{2\pi}{10}\right)(5)\right] + 5\sin\left[(2)\left(\frac{2\pi}{10}\right)(6)\right] + 5\sin\left[(2)\left(\frac{2\pi}{10}\right)(7)\right] + 5\sin\left[(2)\left(\frac{2\pi}{10}\right)(8)\right] + 5\sin\left[(2)\left(\frac{2\pi}{10}\right)(9)\right]\right\} = 0$$

$$\tilde{b}_3 = \frac{2}{10}\left\{2.5\sin\left[(3)\left(\frac{2\pi}{10}\right)(0)\right] + 2.5\sin\left[(3)\left(\frac{2\pi}{10}\right)(5)\right] + 5\sin\left[(3)\left(\frac{2\pi}{10}\right)(6)\right] + 5\sin\left[(3)\left(\frac{2\pi}{10}\right)(7)\right] + 5\sin\left[(3)\left(\frac{2\pi}{10}\right)(8)\right] + 5\sin\left[(3)\left(\frac{2\pi}{10}\right)(9)\right]\right\} = -0.7265$$

and $\tilde{b}_4 = 0$ and $\tilde{b}_5 = 0$.

Not surprising, the results are identical to those found in Example 9.3.

9.2.5 Complete Frequency Spectrum

One of the most important insights that can be obtained from the closed-form expression for the spectral coefficients is how they behave beyond the range $k = 0,\ldots,N/2$. To begin, consider evaluating Eq. (9.24) over the range $k = N/2,\ldots,N$. On doing so, we write

$$\tilde{a}_k = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1} x[n]\cos\left[k\left(\frac{2\pi}{N}\right)n\right], & k = N/2, N \\ \dfrac{2}{N}\displaystyle\sum_{n=0}^{N-1} x[n]\cos\left[k\left(\frac{2\pi}{N}\right)n\right], & k = N/2+1,\ldots,N-1 \end{cases} \tag{9.26}$$

Here, only the cosine terms are affected by the index k. Next, if we consider the change of variable substitution, $k \to N-k$, together combined with the trigonometric identity $\cos\left[(N-k)(2\pi/N)n\right] = \cos\left[k(2\pi/N)n\right]$, Eq. (9.26) can be rewritten as

$$\tilde{a}_{N-k} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1} x[n]\cos\left[k\left(\frac{2\pi}{N}\right)n\right], & k = 0, N/2 \\ \dfrac{2}{N}\displaystyle\sum_{n=0}^{N-1} x[n]\cos\left[k\left(\frac{2\pi}{N}\right)n\right], & k = 1,\ldots,N/2-1 \end{cases} \tag{9.27}$$

Recognizing that Eq. (9.26) is equivalent to Eq. (9.27) allows us to conclude over the range of $k = 1,\ldots,N-1$ that

$$\tilde{a}_{N-k} = \tilde{a}_k \tag{9.28}$$

Following the same reasoning as above in Eqs. (9.26)–(9.28), together with the trigonometric identity $\sin\left[(N-k)(2\pi/N)n\right] = -\sin\left[k(2\pi/N)n\right]$, one will find

$$\tilde{b}_{N-k} = -\tilde{b}_k \tag{9.29}$$

Converting this result into magnitude and phase form, we find using Eq. (9.15)

$$\begin{aligned} \tilde{c}_{N-k} &= \tilde{c}_k \\ \tilde{\phi}_{N-k} &= -\tilde{\phi}_k \end{aligned} \qquad \text{for all } k \tag{9.30}$$

Here we see that the magnitude spectrum excluding the DC bin has even symmetry about bin $N/2$. Similarly, the phase spectrum exhibits an odd symmetry about $N/2$. These two situations are highlighted in Figure 9.5 for an arbitrary discrete-time periodic signal.

Next, let us consider the periodicity of the spectrum. Consider replacing k in Eq. (9.24) by $k + N$ so that we write

$$\tilde{a}_{k+N} = \begin{cases} \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1} x[n]\cos\left[(k+N)\left(\frac{2\pi}{N}\right)n\right], & k = 0, N/2 \\ \dfrac{2}{N}\displaystyle\sum_{n=0}^{N-1} x[n]\cos\left[(k+N)\left(\frac{2\pi}{N}\right)n\right], & k = 1,\ldots,N/2-1 \end{cases} \tag{9.31}$$

EXAMPLE 9.5

Plot the magnitude and phase spectrum for the clock signal of Example 9.3 over 10 frequency bins.

Solution:
The magnitude spectrum for the clock signal of Example 9.2 is shown in Figure 9.6a and the corresponding phase spectrum appears in Figure 9.6b.

Due to the periodicity of the cosine function, Eq. (9.31) simplifies directly to

$$\tilde{a}_{k+N} = \begin{cases} \frac{1}{N}\sum_{n=0}^{N-1} x[n]\cos\left[k\left(\frac{2\pi}{N}\right)n\right], & k = 0, N/2 \\ \frac{2}{N}\sum_{n=0}^{N-1} x[n]\cos\left[k\left(\frac{2\pi}{N}\right)n\right], & k = 1,\ldots,N/2-1 \end{cases} \tag{9.32}$$

which is equal to $\tilde{a}_{k.}$ Hence, we write

$$\tilde{a}_{k+N} = \tilde{a}_k \quad \text{for all } k \tag{9.33}$$

Following a similar line of reasoning, one can write

$$\tilde{b}_{k+N} = \tilde{b}_k \quad \text{for all } k \tag{9.34}$$

Through the direct application of Eq. (9.15), we can write

$$\begin{aligned} \tilde{c}_{k+N} &= \tilde{c}_k \\ \tilde{\phi}_{k+N} &= \tilde{\phi}_k \end{aligned} \qquad \text{for all } k \tag{9.35}$$

We can therefore conclude from above that the spectrum of a periodic signal $x[n]$ with period N is also periodic with period N. Therefore, combining spectral symmetry, together with its periodicity, the frequency spectrum of a discrete-time periodic signal is defined for all frequencies. Figure 9.7 illustrates the full frequency spectrum of an arbitrary signal. To aid the reader, adjacent periods of the spectrum are indicated with dashed boxes.

At this point in our discussion, we should point out that most test vendors only provide spectral information corresponding to the Nyquist interval, $k = 0,1,\ldots, N/2$. Although less important today, 30 years ago when DSP-based ATE started to appear on the market, memory was expensive. Attempts to minimize memory usage were paramount. This led to the elimination of redundant spectral information.

Figure 9.5. Illustrating the spectral symmetry about $N/2$ for $k = 0,1,\ldots,N-1$: (a) magnitude spectrum and (b) phase spectrum.

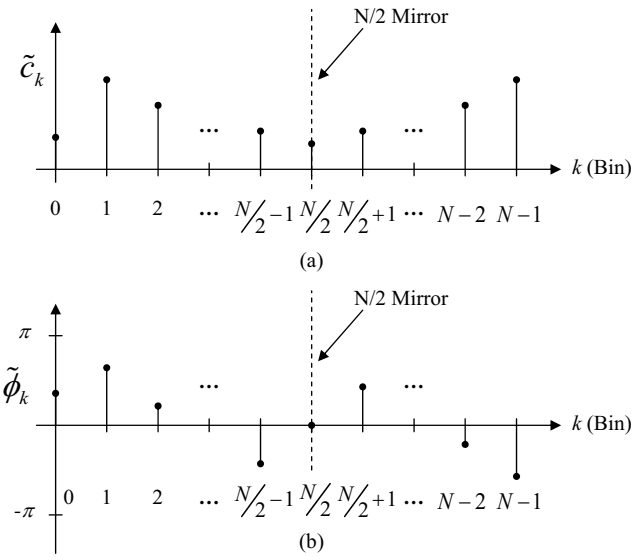


Figure 9.6. (a) Magnitude and (b) phase spectrum for the 10-kHz clock signal over 10 frequency bins.

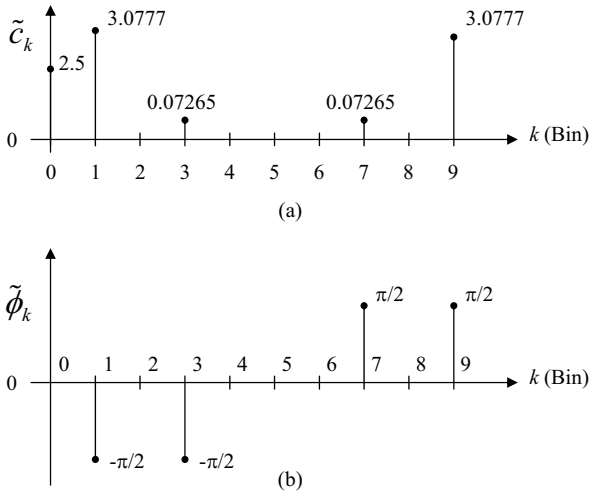
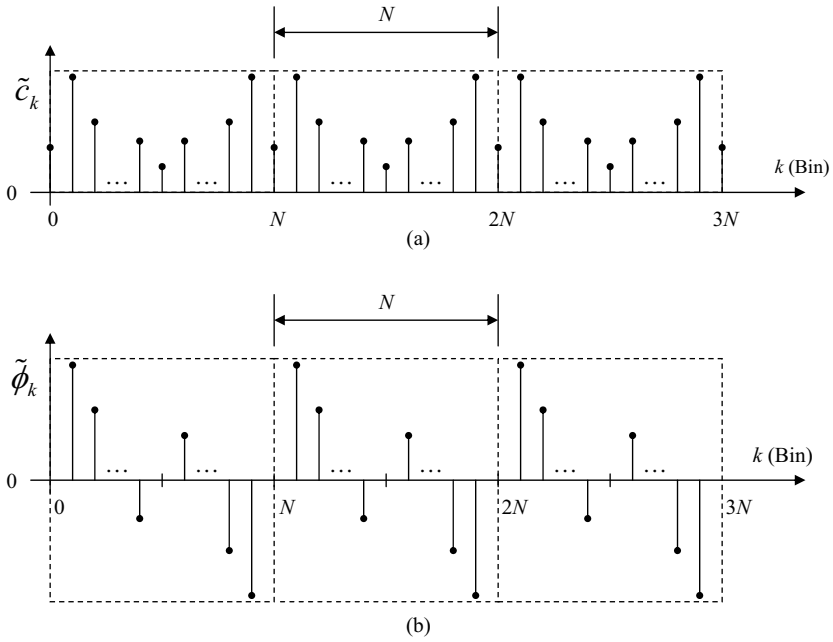


Figure 9.7. Illustrating the spectral periodicity: (a) magnitude spectrum and (b) phase spectrum.



9.2.6 Time and Frequency Denormalization

The time and frequency scale associated with a data sequence $x[n]$ is described in terms of normalized time and frequency, according to the sample indexes, n and k, respectively. To obtain the actual time and frequency scales associated with the original samples, one must perform the operation of time and frequency *denormalization*. To achieve this, knowledge of the sampling period T_s or sampling frequency F_s is required.

To reconstruct the original time scale, one simply multiplies the sample index, n by T_s, according to the translation

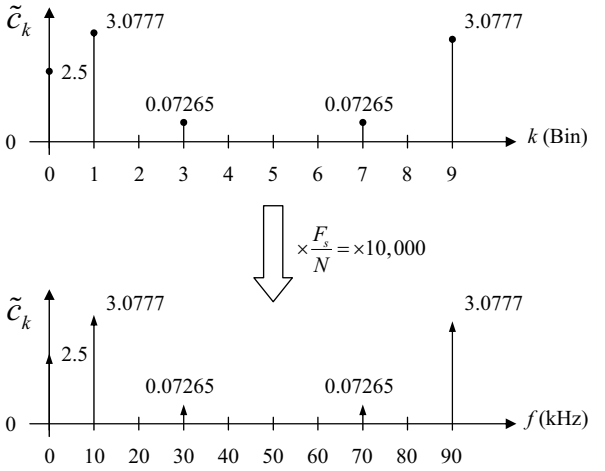
$$n \rightarrow nT_s \tag{9.36}$$

Conversely, the frequency scale is restored when one multiplies the sample index k by F_s/N, according to the translation

$$k \rightarrow k\frac{F_s}{N} \tag{9.37}$$

Figure 9.8 illustrates the frequency denormalization procedure for the magnitude of the spectrum for the clock signal described in Example 9.5. For this particular case, F_s = 100 kHz and N = 10, resulting in a frequency denormalization scale factor of 10 kHz.

Figure 9.8. Illustrating the procedure of denormalizing a frequency axis.



9.2.7 Complex Form of the DTFS

In most DSP textbooks, the DTFS is expressed in complex form using Euler's equation:

$$e^{j\varphi} = \cos(\varphi) + j\sin(\varphi) \tag{9.38}$$

where j is a complex number equal to $\sqrt{-1}$. The main reason for this choice lies with the ease in which the exponential function can be algebraically manipulated in contrast to trigonometric formulas. To convert the DTFS representation in Eq. (9.12) into complex form, consider that the cosine and sine functions can be written as

$$\cos(\varphi) = \frac{e^{j\varphi} + e^{-j\varphi}}{2} \qquad \sin(\varphi) = \frac{e^{j\varphi} - e^{-j\varphi}}{2j} \tag{9.39}$$

When the preceding two formulae are substituted into Eq. (9.12), together with several algebraic manipulations, we get

$$x[n] = \tilde{a}_0 + \sum_{k=1}^{N/2} \left(\frac{\tilde{a}_k - j\tilde{b}_k}{2} \right) e^{jk\left(\frac{2\pi}{N}\right)n} + \sum_{k=N/2}^{N-1} \left(\frac{\tilde{a}_{N-k} + j\tilde{b}_{N-k}}{2} \right) e^{jk\left(\frac{2\pi}{N}\right)n} \tag{9.40}$$

The above expression for the DTFS is more conveniently written in complex form as

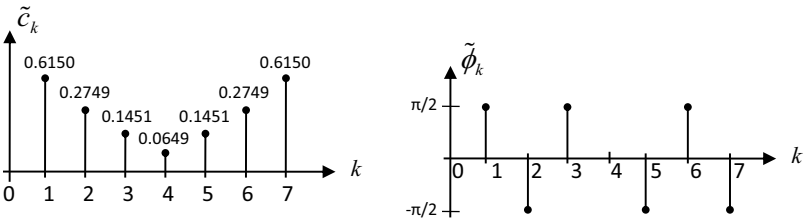
$$x[n] = \sum_{k=0}^{N-1} X(k)\, e^{jk\left(\frac{2\pi}{N}\right)n} \tag{9.41}$$

Exercises

9.7. Plot the magnitude and phase spectrum of the DTFS representation of a sampled signal described by

$$x[n] = 0.6150\ \sin\left[\left(\frac{2\pi}{8}\right)n\right] - 0.2749\ \sin\left[2\left(\frac{2\pi}{8}\right)n\right] + 0.1451\ \sin\left[3\left(\frac{2\pi}{8}\right)n\right] + 0.0649\ \cos\left[4\left(\frac{2\pi}{8}\right)n\right]$$

ANS.

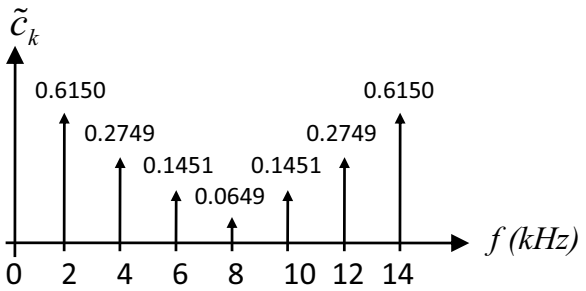


9.8. Plot the frequency-denormalized magnitude spectrum of the following DTFS representation of a sampled signal described by

$$x[n] = 0.6150\sin\left[\left(\frac{2\pi}{8}\right)n\right] - 0.2749\sin\left[2\left(\frac{2\pi}{8}\right)n\right] + 0.1451\sin\left[3\left(\frac{2\pi}{8}\right)n\right] + 0.0649\cos\left[4\left(\frac{2\pi}{8}\right)n\right]$$

assuming a sampling rate of 16kHz.

ANS.



where

$$X(k) = \begin{cases} \tilde{a}_0, & k = 0 \\ \dfrac{\tilde{a}_k - j\tilde{b}_k}{2}, & k = 1,2,\ldots,N/2-1 \\ \tilde{a}_{N/2}, & k = N/2 \\ \dfrac{\tilde{a}_{N-k} + j\tilde{b}_{N-k}}{2}, & k = N/2+1,\ldots,N-1 \end{cases} \tag{9.42}$$

As is evident, the coefficients $X(k)$ in front of each exponential term are, in general, complex numbers. For the most part, the real component of each term relates to one-half the cosine coefficient of the trigonometric series. Conversely, the imaginary part of each term is related to one-half the sine coefficients. The exceptions are the $X(0)$ and $X(N/2)$ terms. These two terms are directly related to the cosine terms with a scale factor of one. The fact that the scale factor is not evenly distributed among each term can be a source of confusion for some.

Alternative forms of Eq. (9.42) can also be written. For instance, we can rewrite Eq. (9.42) in polar form, together with substitutions from Eq. (9.15), as

$$X(k)=\begin{cases}\tilde{c}_0 e^{j0}, & k=0\\ \dfrac{1}{2}\tilde{c}_k e^{-j\tilde{\phi}_k}, & k=1,2,\ldots,N/2-1\\ \tilde{c}_{N/2}e^{j0}, & k=N/2\\ \dfrac{1}{2}\tilde{c}_{N-k}e^{+j\tilde{\phi}_{N-k}}, & k=N/2+1,\ldots,N-1\end{cases} \tag{9.43}$$

9.3 DISCRETE-TIME TRANSFORMS

9.3.1 The Discrete Fourier Transform

In the previous section it was shown how a sequence of N samples repeated indefinitely can be represented *exactly* with a set of $N/2$ harmonically related sinusoidal pairs and a DC component, or in terms of N harmonically related complex exponential functions. In this section, we shall demonstrate that a similar set of harmonically related exponential functions can also be used to represent a sequence of N samples of finite duration. Such signals are known as *discrete-time aperiodic* signals.

Consider an arbitrary sequence $y[n]$ that is of finite duration over the time interval $n = 0$ *to* $N–1$. Next, consider that $y[n] = 0$ outside of this range. A signal of this type is shown in Figure 9.9a. From this aperiodic signal, we can construct a periodic sequence $x[n]$ for which $y[n]$ is one period, as illustrated in Figure 9.9b. This is known as the *periodic extension* of $y[n]$. In mathematical terms, we can describe $y[n]$ as

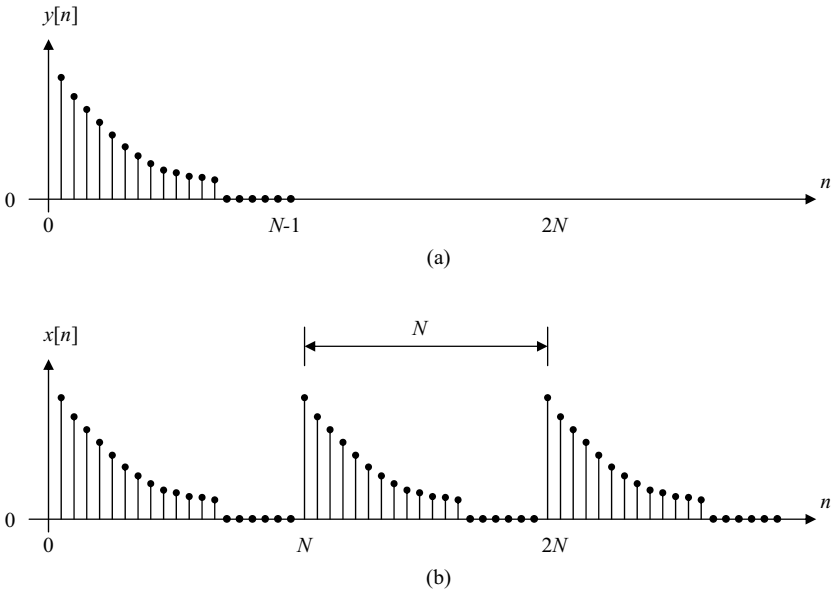
$$y[n]=\begin{cases}x[n], & n=0,1,\ldots,N-1\\ 0, & \text{otherwise}\end{cases} \tag{9.44}$$

If we consider the complex form of the DTFS representation for $x[n]$, we can write $y[n]$ as

$$y[n]=\begin{cases}\displaystyle\sum_{k=0}^{N-1} X(k)\, e^{jk\left(\frac{2\pi}{N}\right)n}, & n=0,1,\ldots,N-1\\ 0, & \text{otherwise}\end{cases} \tag{9.45}$$

As we choose the period N to be larger, $y[n]$ matches $x[n]$ over a longer interval, and as $N \to \infty$, $y[n] = x[n]$ for any finite value of n. Thus, for very large N, the spectrum of $y[n]$ is identical to $x[n]$. However, we note that as $N \to \infty$, the form of the mathematics in Eq. (9.45)

Figure 9.9. (a) Arbitrary signal of finite duration; (b) periodic extension of infinite duration.



changes. The spectral makeup of the discrete-time signal no longer consists of harmonically related discrete frequencies, but rather becomes a continuous function of frequency. Under such conditions, the representation in Eq. (9.45) in the limit becomes known as a *Fourier transform* and is written with an integral operation as follows

$$y[n]=\frac{1}{2\pi}\int_{-\pi}^{\pi} Y\left(e^{j\omega}\right)e^{j\omega n}d\omega \tag{9.46}$$

where

$$Y\left(e^{j\omega}\right)=\sum_{n=-\infty}^{\infty} y[n]\,e^{-j\omega n} \tag{9.47}$$

As in all computer applications, of which mixed-signal testing is just one example, we must limit our discussion to finite values of N and, preferably (because time is always a major concern), to small values of N. This implies that we really have no other choice but to work directly with Eq. (9.45). It has been shown that the spectral coefficients $X(k)$ associated with Eq. (9.45) are directly related to samples of $Y\left(e^{j\omega}\right)$ uniformly spaced according to

$$X(k)=\left.\frac{Y\left(e^{j\omega}\right)}{N}\right|_{\omega=\frac{2\pi}{N}k} \tag{9.48}$$

Substituting Eq. (9.47) into (9.48), and limiting the summation to the maximum possible nonzero values of $y[n]$, $n = 0,1,\ldots, N–1$, we can write

$$X(k)=\frac{1}{N}\sum_{n=0}^{N-1} y[n]e^{-jk\left(\frac{2\pi}{N}\right)n} \tag{9.49}$$

Due to the importance of the interplay between the spectral coefficients of the DTFS representing the periodic extension of the aperiodic signal and its Fourier transform, the set of coefficients

Exercises

9.9. A DTFS representation for a sequence of data is given by

$$x[n]=0.25+1.0\cos\left[\left(\frac{2\pi}{10}\right)n\right]+0.5\sin\left[\left(\frac{2\pi}{10}\right)n\right]$$
$$+0.2\cos\left[3\left(\frac{2\pi}{10}\right)n\right]-0.2\sin\left[3\left(\frac{2\pi}{10}\right)n\right]+0.2\cos\left[5\left(\frac{2\pi}{10}\right)n\right]$$

Express $x[n]$ in complex form.

ANS. $$x[n]=0.25+(0.5-j0.25)e^{j\left(\frac{2\pi}{10}\right)n}+(0.1+j0.1)e^{j3\left(\frac{2\pi}{10}\right)n}$$
$$+0.2e^{j5\left(\frac{2\pi}{10}\right)n}+(0.1-j0.1)e^{j7\left(\frac{2\pi}{10}\right)n}+(0.5+j0.25)e^{j9\left(\frac{2\pi}{10}\right)n}$$

9.10. A DTFS representation expressed in complex form is given by

$$x[n]=2+(1+j1)e^{j\left(\frac{2\pi}{8}\right)n}+(1-j1)e^{j3\left(\frac{2\pi}{8}\right)n}+(1+j1)e^{j5\left(\frac{2\pi}{8}\right)n}+(1-j1)e^{j7\left(\frac{2\pi}{8}\right)n}$$

Express x[n] in trigonometric form.

ANS. $$x[n]=2+2.0\cos\left[\left(\frac{2\pi}{8}\right)n\right]-2.0\sin\left[\left(\frac{2\pi}{8}\right)n\right]+2.0\cos\left[3\left(\frac{2\pi}{8}\right)n\right]+2.0\sin\left[3\left(\frac{2\pi}{8}\right)n\right]$$

9.11. The following vector describes the spectral coefficients of a DTFS expressed in complex rectangular form

$$X=\left[1\quad 0.25+j0.25\quad 4-j1\quad 0\quad 0.3\quad 0\quad 4+j1\quad 0.25-j0.25\right]$$

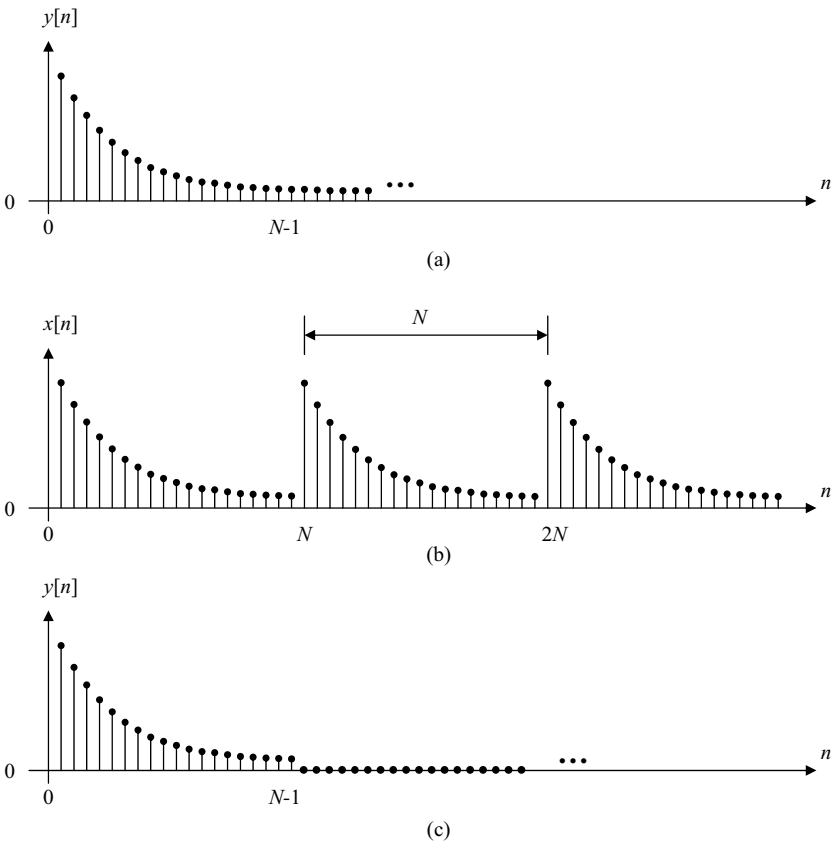
Write the DTFS in trigonometric form using a magnitude and phase notation.

ANS. $$x[n]=1+0.7071\cos\left[\left(\frac{2\pi}{8}\right)n+\frac{\pi}{4}\right]+8.2462\cos\left[2\left(\frac{2\pi}{8}\right)n-0.0780\pi\right]+0.3\cos\left[4\left(\frac{2\pi}{8}\right)n\right]$$

$\{X(0), X(1), \ldots, X(N-1)\}$ in Eq. (9.49) is referred to as the *discrete Fourier transform* (DFT) of $y[n]$. Because the DFT is essentially a special interpretation of a DTFS, the algorithms in Section 9.2.5 can also be used to produce the spectral coefficients of the DFT. However, as explained in the next section, a more efficient algorithm is available to compute the DFT of a discrete-time aperiodic signal or, for that matter, the spectral coefficients of a DTFS. This algorithm is known as the *fast Fourier transform* (FFT) and represents one of the most significant developments in digital signal processing. However, before we move on to this topic, we shall first consider an important degenerate case of the DFT.

Consider the situation where an aperiodic signal has infinite duration or exists with nonzero values over a much longer time than the observation interval of N samples. This is illustrated in Figure 9.10a for an exponentially decaying waveform. Under such conditions it is impossible to represent this signal exactly with a periodic signal having a finite period. Instead, one can only approximate the waveform over the observation interval as shown in Figure 9.10b using a periodic extension of the finite duration signal shown in Figure 9.10c. The error is a form of *time-domain aliasing* and is directly related to the jump discontinuity that occurs at the wraparound point of the periodically extended waveform. The spectral coefficients determined by the DFT would then correspond to samples of the Fourier transform of the signal shown in Figure 9.10c, not Figure 9.10a.

Figure 9.10. (a) Arbitrary signal of infinite duration; (b) periodic extension of infinite duration of short portion of signal; (c) signal approximation of finite duration.



In practice it is important to keep the jump discontinuity to a minimum if the DFT is to reveal the spectral properties of the original waveform. The most common method is to extend the observation interval to large values of N so that the combined energy of the aliased components is made insignificant relative to the energy in the signal. Furthermore, some reduction in the overall observation time can be achieved if the method of windowing is used. Windowing is a mathematical process that alters the shape of the signal over the observation interval and gradually forces it to decay to zero at both ends. This eliminates the discontinuities in the periodically extended waveform at the expense of decreased frequency resolution. The net result is a concentration of the aliasing energy or *frequency leakage* into a few spectral bins instead of having it spread across many different bins.

Because test time is always paramount in a mixed-signal test, one should avoid discontinuities in the periodically extended waveform. In the words of Chapter 8, one should restrict all signals to be coherent rather than noncoherent. We shall delay the introduction of our examples until we first describe the principles of the fast Fourier transform.

9.3.2 The Fast Fourier Transform

In the early 1960s, J. Tukey invented a new algorithm for performing the DFT computations in a much more efficient manner. J. W. Cooley, a programmer at IBM, translated Tukey's algorithm into computer code and the Cooley–Tukey fast Fourier transform (FFT) was born.[4] It is now known that this algorithm actually dates back at least a century. The great German mathematician C. F. Gauss is known to have developed the same algorithm.

To understand the significance of the DFT algorithm, consider in Eq. (9.49) that N complex multiplications and N–1 additions are required to compute each spectral coefficient $X(k)$. For N spectral coefficients, another N multiplications and additions are necessary. Therefore, in total, $(N–1)N$ complex multiplications and additions will be required. As an example, to perform a DFT on 1024 samples, a computer has to perform over one million multiplications. To minimize test time, we would prefer that the DFT computation time be as small as possible; thus obviously we need a more efficient way to perform the multiplications in a DFT. This is where the FFT comes in.

The FFT works by partitioning each of the multiplications and additions in the DFT in such a way that there are many redundant calculations. The redundancy is removed by "folding" the redundant calculations on top of one another and performing each calculation only once. The folding operation forms a so-called *butterfly network* because of the butterfly shapes in the calculation flowchart.[5] There are several different ways to split the calculations and fold the redundancies into one another. The butterfly network can be laid out in a decimation-in-frequency configuration or a decimation-in-time configuration. Fortunately, the details of the FFT algorithm itself are largely unimportant to the test engineer, since the FFT operation is built into the operating system of most mixed-signal testers.

Since many redundant calculations are eliminated by the FFT, it only requires $N \log_2(N)$ complex multiplications. For a 1024-point FFT, only 1024×10 or 10240 complex multiplications are required. Compared to the one million complex multiplications required by the complex version of the DFT, this represents a huge reduction in computation time. The difference between the FFT and DFT becomes more extreme with larger sample sets, since the DFT produces an exponential increase in computations as the sample size increases.

Although the FFT produces the same output as an equivalent DFT, the more common FFTs can only operate on a sample size that is equal to 2n, where n is an integer. For instance, it is not possible to perform a standard Cooley–Tukey FFT on 380 samples, although a DFT would have no problem doing so. The limited choice of sample sizes is the major difference between the DFT and the FFT, other than the difference in computation time. Nevertheless, the savings in test time

is so huge that test engineers usually have no choice but to use the FFT with its limited sample size flexibility. It is quite possible that improvements in computation speeds will eventually make the FFT obsolete in mixed-signal testers, allowing DFTs instead. Until then, the mixed-signal test engineer should be prepared to work with 2n samples for most tests.

9.3.3 Interpreting the FFT Output

The output format of a mixed-signal tester's FFT depends somewhat on the vendor's operating system. In older testers, the format of the FFT output was arranged as an *N*-point array with the DC and Nyquist levels in the first two array elements followed by the cosine/sine pairs for each spectral bin. Today, most testers incorporate commercial DSP chips sets that compute the FFT using complex arithmetic and store the complex numbers in an array beginning with the DC bin followed by successive harmonic bins up to the Nyquist bin. The same format is also used for most numerical software packages such as MATLAB.

One has to be careful, though, when interpreting the FFT output. Many ATE versions of the FFT do not produce peak voltage outputs. Some produce voltage squared (power) outputs, some produce voltage outputs multiplied by the number of samples over 2 (i.e., a 1-V input with 1024 points produces an FFT output of 512 units), and so on. This suggests that the test engineer must become familiar with the FFT routine that they intend to use and determine all the necessary scale factors. In addition, it has been the authors' preference to adjust the scale factors so that the FFT produces RMS levels instead of peak levels. For reasons that will become clear in the next chapter, many test metrics call for the combination of the power of several spectral components. Working with RMS values simplifies this approach.

To better understand the steps involved, let us consider the manner in which MATLAB performs the FFT and the corresponding scale factors needed to produce RMS spectral levels. First, the FFT that is performed in MATLAB is given by the equation

$$Y(k) = \sum_{n=0}^{N-1} y[n]\, e^{-jk\left(\frac{2\pi}{N}\right)n} \tag{9.50}$$

In turn, the complex spectral coefficients $X(k)$ of the DTFS representation of the periodic extension of $y[n]$ are given by

$$X(k) = \frac{Y(k)}{N} \tag{9.51}$$

and the corresponding cosine/sine coefficients are found according to Eq. (9.42) to be

$$\tilde{a}_k = \begin{cases} \mathrm{Re}\{X(k)\}, & k = 0, N/2 \\ 2\,\mathrm{Re}\{X(k)\}, & k = 1, 2, \ldots, N/2 - 1 \end{cases} \tag{9.52}$$

and

$$\tilde{b}_k = \begin{cases} 0, & k = 0, N/2 \\ -2\,\mathrm{Im}\{X(k)\}, & k = 1, 2, \ldots, N/2 - 1 \end{cases} \tag{9.53}$$

where Re{ } and Im{ } denote the real and imaginary parts of a complex number, respectively. Again, we must alert the reader to the different scale factors in front of these terms.

The magnitude and phaserepresentation is determined from Eq. (9.15) to be

$$\tilde{c}_k = \begin{cases} \left|X(k)\right|, & k = 0, N/2 \\ 2\left|X(k)\right|, & k = 1, \ldots, N/2 - 1 \end{cases} \tag{9.54}$$

where $\left|X(k)\right| = \sqrt{\text{Re}\left\{X(k)\right\}^2 + \text{Im}\left\{X(k)\right\}^2}$ and the phase is computed using the four-quadrant formula given below

$$\tilde{\phi}_k = \begin{cases} -\tilde{\phi}_o & \text{if } \text{Re}\left[X(k)\right] \geq 0, \text{Im}\left[X(k)\right] \geq 0, \\ \tilde{\phi}_o & \text{if } \text{Re}\left[X(k)\right] \geq 0, \text{Im}\left[X(k)\right] < 0, \\ \tilde{\phi}_o - \pi & \text{if } \text{Re}\left[X(k)\right] < 0, \text{Im}\left[X(k)\right] \geq 0, \\ \pi - \tilde{\phi}_o & \text{if } \text{Re}\left[X(k)\right] < 0, \text{Im}\left[X(k)\right] < 0, \end{cases} \quad \text{where} \quad \tilde{\phi}_o = \tan^{-1}\left\{\frac{\left|\text{Im}\left[X(k)\right]\right|}{\left|\text{Re}\left[X(k)\right]\right|}\right\} \tag{9.55}$$

In many situations we shall find it more convenient to report the spectral coefficients in terms of their RMS values. To do so, we divide the spectral coefficients $\tilde{c}_k$ (except the DC term described $k = 0$) by $\sqrt{2}$ to obtain

$$\tilde{c}_{k-RMS} = \begin{cases} \tilde{c}_k, & k = 0 \\ \dfrac{\tilde{c}_k}{\sqrt{2}}, & k = 1, \ldots, N/2 \end{cases} \tag{9.56}$$

On substituting Eq. (9.54) into (9.56), we obtain

$$\tilde{c}_{k-RMS} = \begin{cases} \left|X(k)\right|, & k = 0 \\ \sqrt{2}\left|X(k)\right|, & k = 1, \ldots, N/2 - 1 \\ \dfrac{1}{\sqrt{2}}\left|X(k)\right|, & k = N/2 \end{cases} \tag{9.57}$$

The following example will further illustrate this procedure.

At this point in our discussion of the FFT, it would be instructive to consider the spectral properties of a coherent sinusoidal signal and a noncoherent sinusoidal signal having equal amplitudes.

Extending the observation interval in Example 9.8 certainly helped to decrease the amount of frequency leakage, which, in turn, helped to improve the measurement accuracy of the noncoherent

EXAMPLE 9.6

Using the FFT routine in MATLAB (or some other equivalent software), compute the spectral coefficients $\{a_k\}$ and $\{b_k\}$ of a multitone signal having the following eight samples; {0.1414, 1.0, –0.1414, –0.8, –0.1414, 1.0, 0.1414, –1.2}. These samples were derived from a signal with the following DTFS representation

$$x[n] = \cos\left[2\left(\frac{2\pi}{8}\right)n - \frac{\pi}{2}\right] + 0.2\ \cos\left[3\left(\frac{2\pi}{8}\right)n - \frac{\pi}{4}\right]$$

Also report the magnitude (in RMS) and the phase of each spectral coefficient.

Solution:
With the samples of the multitone signal described as

$$x = [0.1414,\ 1.0,\ -0.1414,\ -0.8,\ -0.1414,\ 1.0,\ 0.1414,\ -1.2],$$

the FFT routine in MATLAB produces the following output, together with the scaled result:

$$Y = FFT(x) = \begin{bmatrix} 0 \\ 0 \\ 0 - j4.0000 \\ 0.5657 - j0.5657 \\ 0 \\ 0.5657 + j0.5657 \\ 0 + j4.0000 \\ 0 \end{bmatrix} \quad \Rightarrow \quad X = \frac{Y}{8} = \begin{bmatrix} 0 \\ 0 \\ 0 - j0.5000 \\ 0.0707 - j0.0707 \\ 0 \\ 0.0707 + j0.0707 \\ 0 + j0.5000 \\ 0 \end{bmatrix}$$

Subsequently, the cosine/sine spectral coefficients are determined from Eqs. (9.52) and (9.53) to be

$$\begin{bmatrix} \tilde{a}_0 \\ \tilde{a}_1 \\ \tilde{a}_2 \\ \tilde{a}_3 \\ \tilde{a}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.1414 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \tilde{b}_0 \\ \tilde{b}_1 \\ \tilde{b}_2 \\ \tilde{b}_3 \\ \tilde{b}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1.0 \\ 0.1414 \\ 0 \end{bmatrix}$$

Finally, the corresponding magnitude (in RMS) and phase terms (in radians) are as follows:

$$\begin{bmatrix} \tilde{c}_{0-RMS} \\ \tilde{c}_{1-RMS} \\ \tilde{c}_{2-RMS} \\ \tilde{c}_{3-RMS} \\ \tilde{c}_{4-RMS} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.7071 \\ 0.1414 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \tilde{\phi}_0 \\ \tilde{\phi}_1 \\ \tilde{\phi}_2 \\ \tilde{\phi}_3 \\ \tilde{\phi}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \pi/2 \\ \pi/4 \\ 0 \end{bmatrix}$$

EXAMPLE 9.7

Using the FFT routine in MATLAB (or some other equivalent software), compute the spectral coefficients of a coherent and noncoherent sinusoidal signal with parameters $A = 1$, $\phi = 0$, $M = 3$, $N = 64$, and $A = 1$, $\phi = 0$, $M = \pi$ (3.14156), $N = 64$. For each case, plot the RMS magnitude of the spectrum in dB relative to a 1-V RMS reference level.

Solution:
The following MATLAB routine was written to produce 64 samples of the coherent and noncoherent waveforms and to perform the corresponding FFT analysis:

```
% coherent signal definition-x -
N=64; M=3; A=1; P=0;     % signal definition
      for n=1:N,
            x(n)=A*sin(2*pi*M/N*(n-1)+P);
      end;
% noncoherent signal definition–z -
      N=64; M=pi; A=1; P=0;
      for n=1:N,
            z(n)=A*sin(2*pi*M/N*(n-1)+P);
      end;
% perform Fourier analysis
      X=fft(x)/length(x);
      Z=fft(z)/length(z);
```

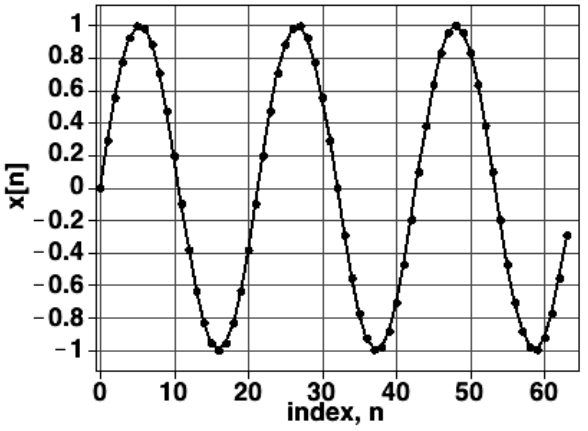
The results of the FFT analysis for the coherent waveform are shown in Figure 9.11. The time-domain waveform is shown on the left, while the corresponding magnitude of the spectrum is shown on the right. The spectrum is expressed in dB relative to a 1-V RMS reference level. When this definition is used, the decibel units are referred to as dBV. Mathematically, it is written as

$$c_{k-RMS}(dBV) = 20\ \log_{10}\left(\frac{c_{k-RMS}}{1-\text{V RMS}}\right) \tag{9.58}$$

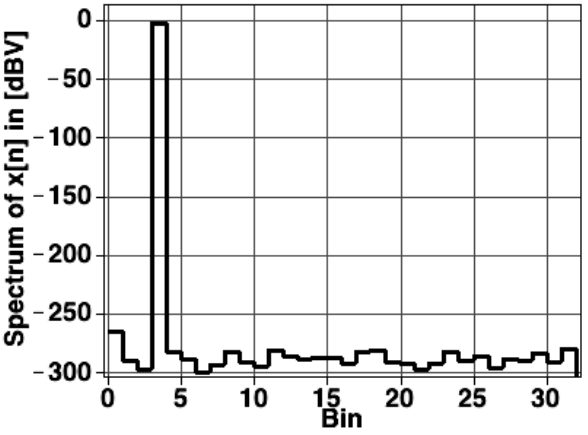
Also, it is customary to plot the frequency-domain data as a continuous curve by interpolating between frequency samples instead of using a line spectrum. In some cases, one uses a zero-order interpolation operation to produce a step or bar graph of the spectrum, as we shall use in the next few examples, or a first-order interpolation operation.

The results for the noncoherent waveform were also found and are shown in Figure 9.12. On comparing the magnitude of the two spectra, we clearly see a significant difference. In the case of the coherent sinusoidal waveform, a single spike occurs in bin 3 with over 300 dB of separation distance from all other spectral coefficients. A closer look at the numbers indicates that the tone has an RMS value of 0.707106 (-3.0103 dBV in the plot) or an amplitude of 1. This is exactly the value specified in the code that is used to generate the coherent sinusoid. In the case of the noncoherent sinusoidal waveform, no single spike occurs. Rather, the single-tone waveform appears to consist of many frequency components. If other signal components were present, then they would be corrupted by the power in these leaked components. What is worse, it is extremely difficult to determine the amplitude of the noncoherent sinusoidal signal with its power smeared across many frequency locations.

Figure 9.11. (a) Coherent waveform time-domain plot and (b) corresponding spectrum plot.



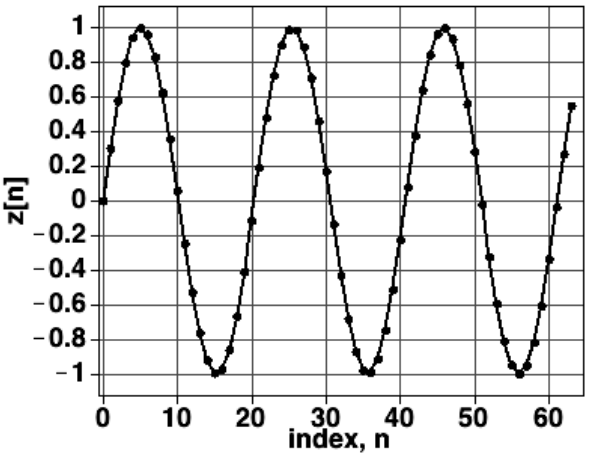
(a)



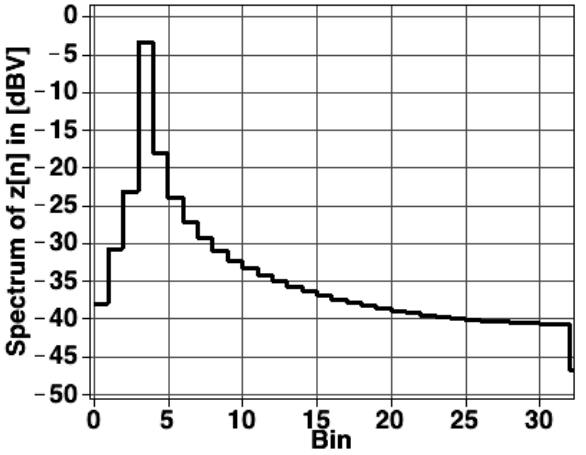
(b)

The most straightforward method to improve the measurement accuracy of a noncoherent waveform is to increase its observation interval. Generally speaking, this approach is used by most benchtop instruments found in one's laboratory, such as spectrum analyzers, multimeters, and digitizing oscilloscopes. Samples taken from the input signal are unrelated to the sampling rate of the instrument, and are therefore noncoherent. Instruments of this type are usually not expected to generate a result in a very short time, such as 25 ms. Rather, they are only required to produce a result every 1 or 2 s, which is usually more than adequate. Consequently, one can construct a less complex, noncoherent measurement system. Our next example will illustrate the effect of a longer observation interval on the spectrum of a noncoherent sinusoidal waveform.

Figure 9.12. (a) Noncoherent waveform time-domain plot and (b) corresponding spectrum plot.



(a)



(b)

EXAMPLE 9.8

Extend the observation interval of the noncoherent waveform of Example 9.7 by collecting 8192 samples instead of 64. Plot the corresponding magnitude of the resulting spectrum. Determine the amplitude of the input signal from its spectrum.

Solution:
The MATLAB code for the noncoherent signal from Example 9.7 was modified as follows:

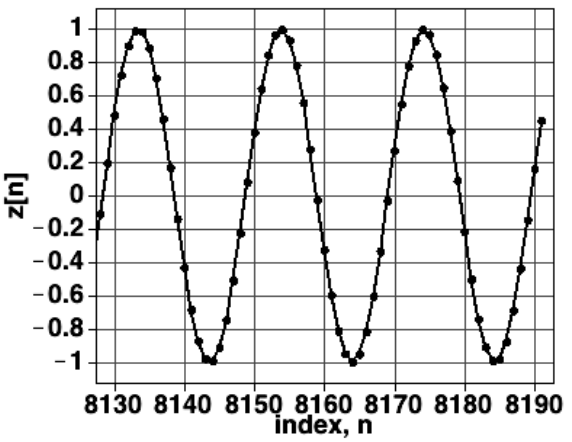
```
% noncoherent signal definition–z -
    NOI=8192;                          % observation interval
```

```
N=64; M=pi; A=1; P=0;                % signal definition
for n=1:NOI,
    z(n)=A*sin(2*pi*M/N*(n-1)+P);
end;
```

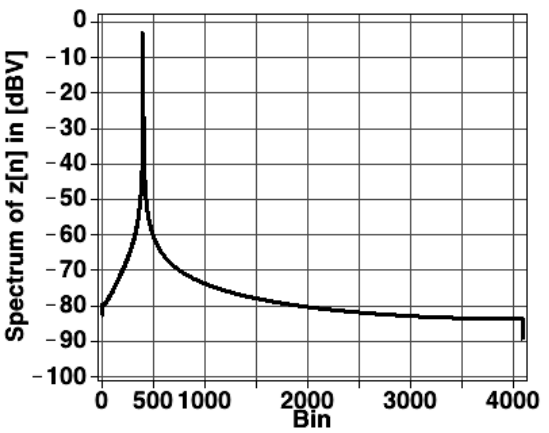
Here we distinguish between N, the number of samples in one UTP, and N_{OI}, the number of samples collected over the entire observation interval. In other words, N_{OI}/N represents the number of UTPs that the signal will complete in the observation interval.

The Fourier analysis was then repeated and the corresponding spectrum was found as shown in Figure 9.13. On the left is the plot of the time-domain waveform over the last 64 samples of the full 8192 samples, as any more samples would fill the graph and mask all detail. On the right is the magnitude of its spectrum.

Figure 9.13. (a) Noncoherent waveform time-domain plot and (b) frequency spectrum with expanded observation interval.



(a)

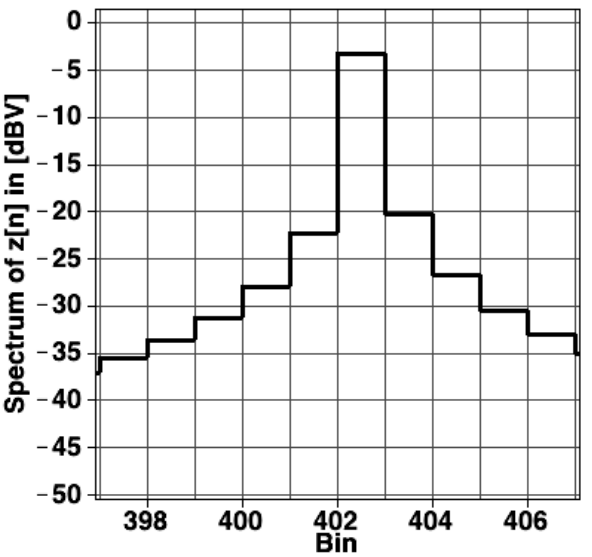


(b)

When we refer back to the spectrum derived in Example 9.7 and compare it to the one derived here, we see that the general shape of the magnitude of the spectrum is much more concentrated around a single frequency and that the frequency leakage components are much smaller. The astute reader may be wondering about the scale of the axis. In this case we are plotting the index or bin of each frequency components from 0 to 8191, whereas in the previous case our frequency index is from 0 to 63. It is important to realize that each bin is equivalent to $Bin\left(F_s/N_{OI}\right)$ Hz. In other words, the frequency range is identical in each case; only the frequency granularity is different.

In order to estimate the amplitude of the input waveform, one cannot rely on the peak value of the spectrum as was done in the coherent waveform case. Rather, we must use several frequency components centered around the peak spectral concentration to estimate the waveform amplitude. To see this more clearly, we provide in Figure 9.14 an expanded view of the spectrum around the spectral peak. Here we see that a peak spectral value of –3.2318 dBV occurs in bin 402 [bin $\left(M/N\right)N_{OI}$]. Ideally, the spectral peak value should be –3.0103 dBV [= $20\log_{10}\left(0.707\text{ V RMS}/1\text{ V RMS}\right)$]. To improve the estimate, we must take into consideration the power associated with the side tones. As the magnitude of these side tones drop off fairly quickly, let us consider that the power associated with the input signal is mainly associated with the power of the five tones before and after the spectral peak. On doing so, the amplitude estimate becomes –3.0355 dBV. Including more side tones into this calculation will only help to improve the estimate. Generally speaking, side tones less than –60 dB below the spectral peak value will improve the accuracy to within 0.1%.

Figure 9.14. Expanded view of the noncoherent tone around the spectral peak.



We could also improve the accuracy of the estimate by further increasing the observation interval. For example, increasing the observation interval to 131,072 samples will improve the amplitude estimate to –3.0109 dB.

waveform. However, as in most production test situations, one is always searching for a faster solution. In Section 9.3.1 the method of windowing was suggested as a possibility. We shall investigate this claim further in the next subsection.

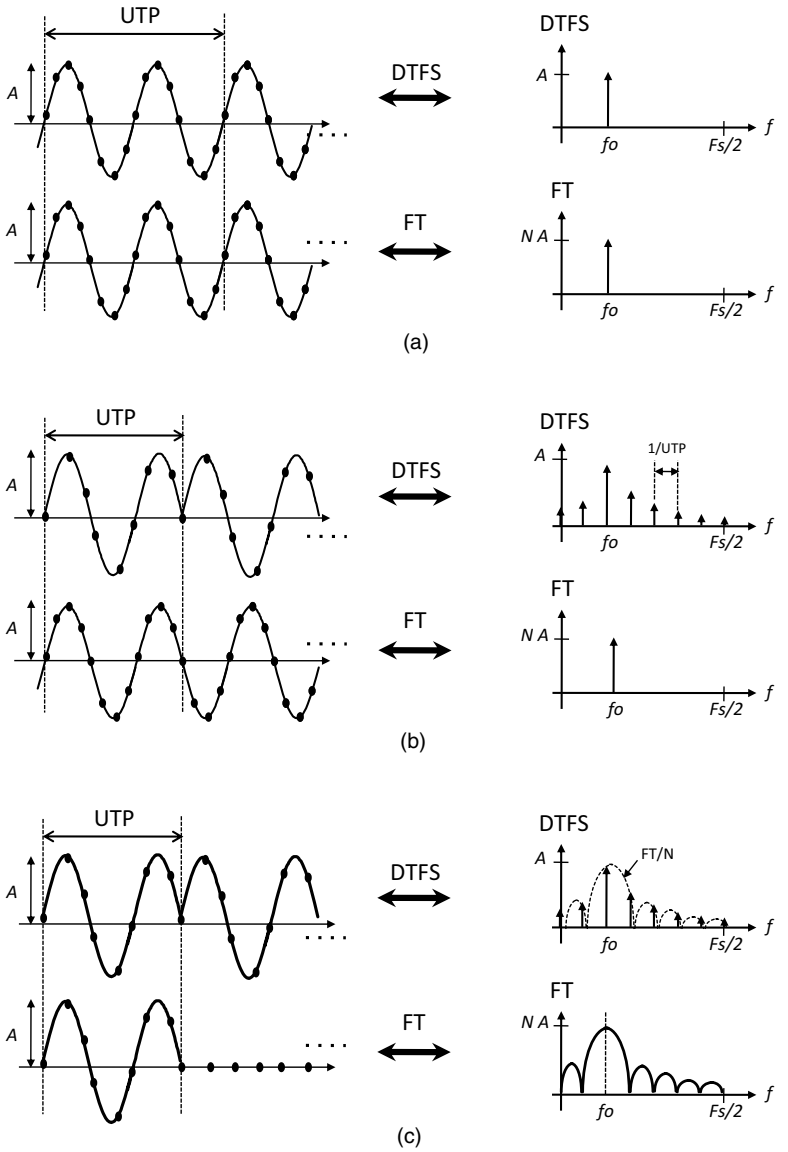
Exercises

9.12.	Evaluate $x[n] = 1.0 \sin\left[3\ (2\pi/16)n + \pi/8\right]$ for n = 0,1,...,15. Using MATLAB, compute the FFT of the 10 samples of $x[n]$ and determine the corresponding $\{a_k\}$ and $\{b_k\}$ spectral coefficients.	ANS. $\{a_3+jb_3\}$ = {0.3827 + j0.9239}; all others are zero.
9.13.	Evaluate $x[n] = (n-3)^2$ for n = 0,1,...,15. Using MATLAB, compute the FFT of the 16 samples of $x[n]$ and determine the corresponding $\{c_k\}$ and $\{\phi_k\}$ spectral coefficients. Express the magnitude coefficients in RMS form.	ANS. $\{c_k\}$ = {41.5, 52.8, 24.3, 16.4, 12.8, 10.8, 9.75, 9.17, 4.50}; $\{c_{k-RMS}\}$ = {41.5, 37.3, 17.2, 11.6, 9.05, 7.67, 6.89, 6.49, 3.18}; $\{\phi_k\}$ = {0, −1.25, −1.70, −1.99, −2.24, −2.47, −2.70, −2.92, −3.14}.
9.14.	A signal has a 0.5-V amplitude. Express its amplitude in units of dBV.	ANS. −9.03 dBV.
9.15.	A signal has a period of 1 ms and is sampled at a rate of 128 kHz. If 128 samples are collected, what is the frequency resolution of the resulting FFT? If the number of samples collected increases to 8192 samples, what is the frequency resolution of the resulting FFT?	ANS. 1 kHz, 15.625 Hz.

9.3.4 Windowing

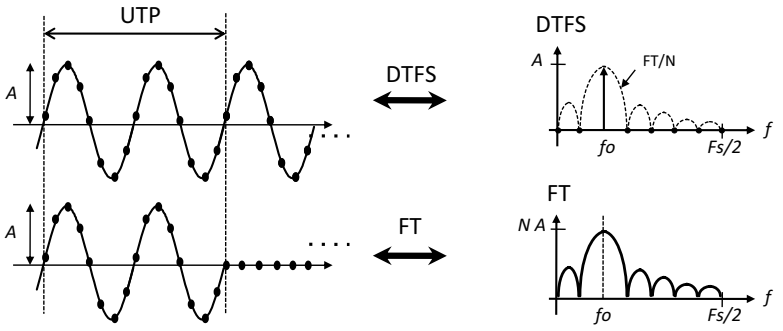
At the heart of a DTFS is the assumption that the signal of interest continues for all time, albeit as a periodic extension of an N-point sample set. If the signal is coherent with the UTP, then the periodic extended waveform will remain coherent over any integer multiple of the UTP. In the limit, as the observation interval (*LxUTP, L* is an integer) goes to infinity, the DTFS representation will have a line spectrum that is identical to that described by the Fourier transform of the infinite-duration sampled sinusoidal signal as illustrated in Figure 9.15a. However, if the sampled signal is not coherent in the UTP, then the resulting DTFS will have a line spectrum that differs significantly from that corresponding to the continuous-running sampled sinusoidal signal. This should not be too surprising given that the periodically extended sampled signal and the continuous-running sampled sinusoidal signals are very different signals, as captured by the illustration on the left-hand side of Figure 9.15b. It should also be of no surprise that the DTFS representation of the noncoherent sampled sinusoidal contains numerous line spectra, as the waveform deviates significantly from an ideal sine wave. Now we learned in Section 9.3.1 that the DTFS representation of a sampled signal represents the sampled spectrum of a finite duration signal, that is, one that lasts for a fixed

Figure 9.15. Comparing the DTFS representation and Fourier transform of various types of sinusoidal signals: (a) coherent with UTP, (b) noncoherent with UTP, and (c) noncoherent vs. aperiodic signal.



duration of a UTP, as depicted in Figure 9.15c. We see from this illustration that the shape of the line spectrum associated with the noncoherent signal is directly related to the Fourier transform of this aperiodic signal. More importantly, the shape of the line spectrum can be traced back to the rectangular function that converts the infinite-duration sampled sine wave to one of finite duration, leading to the term *windowing*.

Figure 9.16. Illustrating the rectangular windowing effect on a coherent signal with UTP and the resulting single-line DTFS representation.



It is interesting to note that the DTFS representation of a coherent sinusoidal signal also represents samples of the Fourier transform of the finite duration signal. However, due to the relationship between the frequency of the signal, f_o, and the sampling frequency F_s, only one point in the entire Nyquist interval ($F_s/2$) is nonzero as shown in Figure 9.16. This, in turn, gives the appearance of a single line spectrum, as well as an appearance identical to that given by its Fourier transform.

Windowing is the operation of converting an infinite duration sampled signal to one of finite duration. Mathematically, if *x[n]* represents a sampled signal of infinite duration and *w[n]* as the samples corresponding to the windowing operation, then the widowed sequence can be written as

$$y[n] = w[n]x[n] \tag{9.59}$$

Windowing is not limited to a rectangular shaped function, because many types of windows have been developed for spectral analysis—*largely to control the shape of the spectral decomposition*. Table 9.1 lists some of the more commonly used window functions and their corresponding N-point sequence. This includes the Hanning (or Hann), Blackman, and Kaiser windows. The Hanning window is used in about 95% of all windowing applications. When in doubt, begin with the Hanning window.

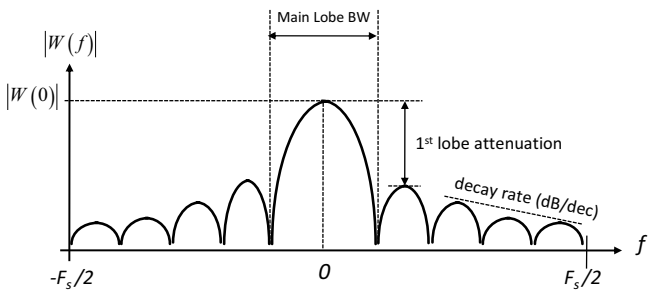
Because multiplication in the time domain corresponds to convolution in the frequency domain, this fact becomes the guiding principle in the design of different window functions. The frequency domain description of a window function is derived through the application of the Fourier transform provided in Eq. (9.47). While window functions have no convenient closed form (except for the rectangular function), window attributes have been collected and tabulated for easy access.[6] Figure 9.17 illustrates the general shape of a window function in the frequency domain. In many ways, the shape of this window resembles that of an amplitude-based filter, and many of the attributes associated with a window are filter-like parameters such as DC frequency gain, main-lobe bandwidth (MLBW), first side-lobe attenuation, rate of attenuation decay and equivalent noise bandwidth (ENBW). One can show that the DC gain of the window, |*W(0)*| is proportional to the sum of the window coefficients as

$$\left|W(0)\right| = \frac{1}{N}\sum_{n=0}^{N-1} w[n] \tag{9.60}$$

Table 9.1. Some Commonly Used Window Types and Their Corresponding *N*-Point Sequence

Window Type	Window Function $w[n]$
Rectangular	$w[n]=\begin{cases}1, & 0\le n\le N-1\\ 0 & \text{otherwise}\end{cases}$
Hanning (Hann)	$w[n]=\begin{cases}0.5-0.5\cos\left(\frac{2\pi}{N-1}n\right) & 0\le n\le N-1\\ 0 & \text{otherwise}\end{cases}$
Blackman	$w[n]=\begin{cases}0.42-0.5\cos\left(\frac{2\pi}{N-1}n\right)+0.08\cos\left(\frac{4\pi}{N-1}n\right) & 0\le n\le N-1\\ 0 & \textit{otherwise}\end{cases}$
Kaiser	$w[n,\beta]=\begin{cases}\frac{I_0\left(\pi\beta\sqrt{1-\left(\frac{2n}{N-1}-1\right)^2}\right)}{I_0(\pi\beta)}, & 0\le n\le N-1\\ 0 & \text{otherwise}\end{cases}$ where $I_0(x)=1+\sum_{k=1}^{\infty}\left[\left(\frac{1}{k!}\frac{x}{2}\right)^k\right]^2\approx 1+\sum_{k=1}^{20}\left[\left(\frac{1}{k!}\frac{x}{2}\right)^k\right]^2$ (zeroth-order modified Bessel function of the first kind)

Figure 9.17. Some attributes of a window function in the frequency domain.



Likewise, the area under the squared-window magnitude response $|W(f)|^2$, denoted by the window shape factor ε^2, can be shown to be equal to

$$\varepsilon^2 = \int_{-F_s/2}^{F_s/2} \left|W(f)\right|^2 df = \frac{1}{N}\sum_{n=0}^{N-1} w^2[n] \tag{9.61}$$

Correspondingly, the ENBW is then defined as

$$\text{ENBW} = \frac{\int_{-F_s/2}^{F_s/2} \left|W(f)\right|^2 df}{\left[W(0)\right]^2} = N\frac{\sum_{n=0}^{N-1} w^2[n]}{\left[\sum_{n=0}^{N-1} w[n]\right]^2} \tag{9.62}$$

Table 9.1 lists commonly used window types, such as rectangular, Hanning, Blackman, and Kaiser, to name just a few. Table 9.2 summarizes their corresponding window metrics. One important window metric that is not self-evident is the processing gain (PG). We'll describe this next.

Consider a set of samples derived from a complex sine wave with frequency f_o and amplitude A, that is,

$$x[n] = Ae^{+j2\pi f_o n} \tag{9.63}$$

Here we make use of a complex-valued sine wave rather than a real-valued one because it simplifies the presentation. Let us further assume these samples are multiplied by some arbitrary window function $w[n]$. The Fourier transform of the widowed signal is then found using Eq. (9.47) to be

$$Y\left(e^{j\omega}\right) = A\sum_{n=0}^{N-1} w[n]e^{+j2\pi f_o n}\ e^{-j\omega n} \tag{9.64}$$

Table 9.2. Some Window Types and Their Corresponding Performance Metrics

Window Type	$G_{COHERENT} = \lvert W(0)\rvert$	$G_{NONCOHERENT} = \sqrt{\int_{-F_s/2}^{F_s/2} \lvert W(f)\rvert^2 df}$	MLBW (bins)	ENBW (bins)	1st Lobe Atten. (dB)	Decay Rate (dB/oct)	PG
Rectangular	1.00	1.00	2	1.00	13	–6	1.00
Hanning (Hann)	0.50	0.612	4	1.50	26	–18	0.666
Blackman	0.42	0.552	6	1.72	50.4	–18	0.579
Kaiser (β = 10)	0.22	0.398	20	3.19	247	–6	0.313

Correspondingly, the FFT samples this frequency domain signal with $\omega = \frac{2\pi}{N}k$, $k = 0, N\text{-}1$, leading to the widowed DTFS spectral coefficient (signified by the subscript *W*) as

$$X_W(k) = \left.\frac{Y\left(e^{j\omega}\right)}{N}\right|_{\omega=\frac{2\pi}{N}k} = \frac{A}{N}\sum_{n=0}^{N-1} w[n]\, e^{j2\pi\left(f_o-\frac{k}{N}\right)n}, \qquad k = 0,1 \tag{9.65}$$

If the input frequency is coherent with the mth bin of the FFT, then Eq. (9.65) reduces to

$$\left|X_W(k)\right| = \begin{cases} \frac{A}{N}\sum_{n=0}^{N-1} w[n], & k = m \\ 0, & k = 0,1,\ldots,N-1, k \neq m \end{cases} \tag{9.66}$$

For the mth bin, we recognize that the magnitude of the spectral coefficient is proportional to the amplitude of the input signal. This allows us to define the *coherent signal gain* of the windowing operation as

$$G_{COHERENT} \triangleq \frac{\left|X_W(m)\right|}{A} = \frac{1}{N}\sum_{n=0}^{N-1} w[n] \tag{9.67}$$

Interestingly enough, the coherent gain is equal to the DC gain of the window.

If we repeat this analysis but this time consider the input as a set of *N* samples derived from a Gaussian distribution with zero mean and standard deviation σ_M, one can show that the expected or average noncoherent power of each FFT bin is

$$E\left\{\left|X_{W-RMS}(k)\right|^2\right\} = \frac{\sigma_M^2}{N^2}\sum_{n=0}^{N-1} w^2[n], \quad k = 0,1,\ldots,N-1 \tag{9.68}$$

In the case of the rectangular window, the average power per bin is $\sigma_{\eta,bin}^2 = \sigma_M^2 / N$, so we define the *noncoherent power gain* for the *m*th bin as

$$G_{NONCOHERENT}^2 \triangleq \frac{E\left\{\left|X_{W-RMS}(m)\right|^2\right\}}{\sigma_{\eta,bin}^2} = \frac{1}{N}\sum_{n=0}^{N-1} w^2[n] \tag{9.69}$$

Here noncoherent power gain is equivalent to the window shape factor ε^2 and we notice that it is less than unity except for a rectangular window. On the surface this would imply that the windowing reduces the noise in each bin and that this would be advantageous. However, the signal level reduces at a faster rate due to the coherence gain, hence the apparent reduction in noise power is further offset by the reduction in signal level. To capture this attribute, we speak in terms of the processing gain (PG), where it is defined as the ratio of the output signal-power-to-noise-power-per-bin to that corresponding to the S/N of the rectangular window, that is,

$$PG \triangleq \frac{[S/N]_{OUT}}{[S/N]_{IN}} = \frac{\frac{\left|X_W(m)\right|^2}{2} \Big/ E\left\{\left|X_{W-RMS}(m)\right|^2\right\}}{\frac{A^2}{2} \Big/ \sigma_{n,bin}^2} = \frac{1}{N}\frac{\left[\sum_{n=0}^{N-1} w[n]\right]^2}{\sum_{n=0}^{N-1} w^2[n]} \tag{9.70}$$

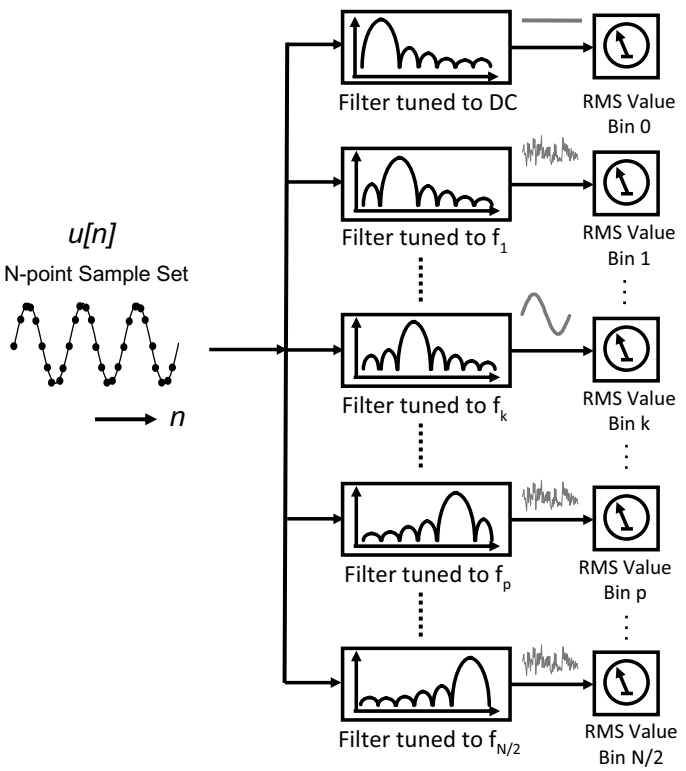
For windows other than rectangular, the sum of window coefficients is always less than the corresponding sum of squared coefficients (as positive and negative coefficients can cancel), PG is always less than one. Therefore, we can conclude that windowing reduces the quality of the spectral coefficient estimation. The astute reader will recognize that the PG is equal to the inverse of the ENBW given in Eq. (9.62). This is reasonable, because a larger noise bandwidth will allow greater noise to contribute to each spectral coefficient estimate.

In order to appreciate the coherent and noncoherent properties of the windowing process associated with the FTT/DTFS spectra estimation process, it is common practice to represent this as a bank of filters as shown in Figure 9.18. Here the center frequency of each filter is tuned to a bin of the FFT and whose frequency domain characteristics are those associated with the window function. Coherent sinusoidal signals will pass directly thorough the center of the appropriate filter with gain $G_{COHERENT}$, for example,

$$X_W(k) = G_{COHERENT} \times X(k) \tag{9.71}$$

where $X(k)$ are the DTFS spectral coefficients that result from a rectangular window. Of course, we could just as easily write this in terms of the c_k spectral coefficients through the application of Eq. (9.54).

Figure 9.18. Representing the windowing-FFT extraction process as a bank of filters whose frequency characteristics are those of the window function tuned to each bin of the FFT.



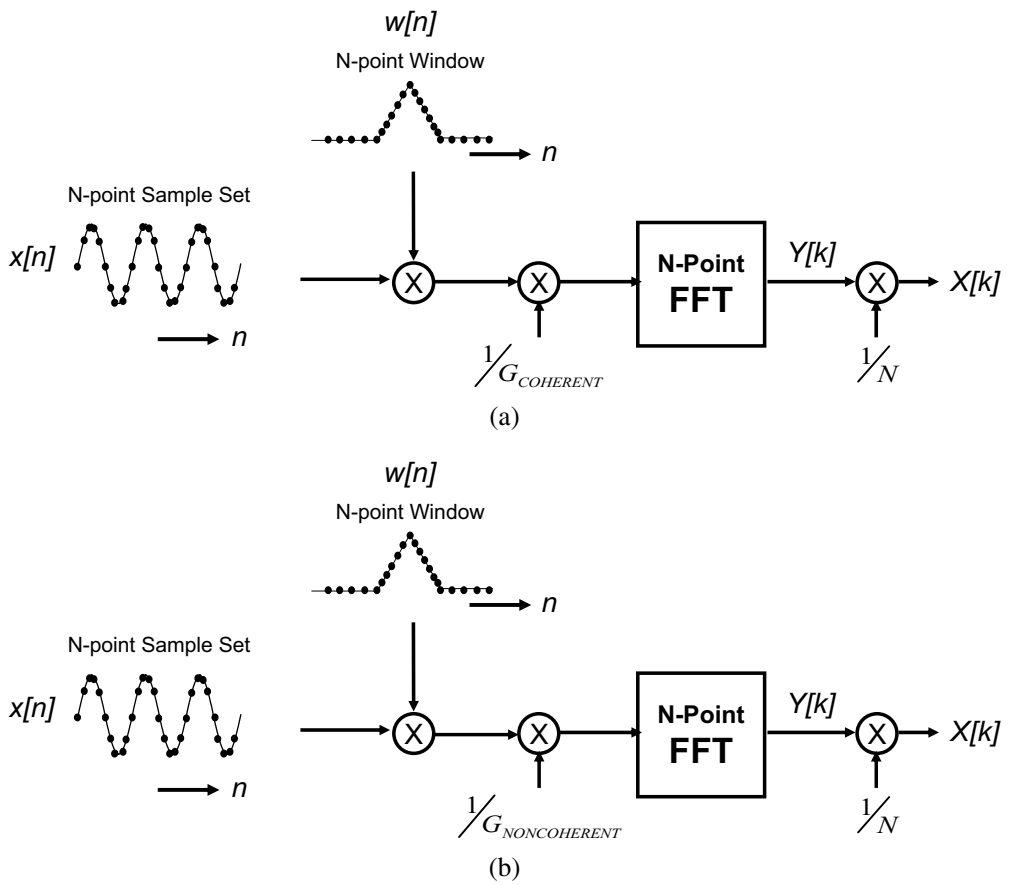
In the case of noncoherent signals, the situation is more complex. If a single noncoherent sine wave signal is processed through a windowing operation within the main-lobe bandwidth of the kth bin, then we describe the spectral coefficient estimate corresponding to bin k as

$$X_W(k) = G_{NONCOHERENT} \times \sqrt{\sum_{l=-MLBW/2}^{MLBW/2} X^2(k+l)} \tag{9.72}$$

If the signal consists of several noncoherent sine waves that are far enough apart so that their main lobes do not overlap, then Eq. (9.72) can be used for each bin k, $k = 0, ..., N-1$. However, if the tones are too close to one another, Eq. (9.72) will be in error, because the spectra will overlap. So unless a window with a main lobe bandwidth less than the frequency separation is found, windowing alone will not improve the spectral coefficient estimates. Widowing combined with an increase observation interval can help to improve this situation. The next few examples will help to illustrate these tradeoffs.

To close this subsection, we illustrate the set up for using windowing with appropriate scale factors for spectral analysis in Figure 9.19. Figure 9.19a illustrates the coherent case and Figure 9.19b illustrates the appropriate scale factor for the noncoherent case. Scaling by the

Figure 9.19. Highlighting the scale factors associated with windowing when (a) coherent signals are being processed; (b) noncoherent signals are being processed.



appropriate gain factor right after the widow operation equalizes the power in the window data with that associated with the original waveform. However, as the power across the main lobe of the window is being combined in a mean-square sense, the noise in each bin is also accumulated resulting in an increase in the noise power associated with the spectral estimate, given by

$$\sigma_{X_k} = \sigma_{n,bin}\sqrt{\text{MLBW}} \tag{9.73}$$

where $\sigma_{n,bin}$ is the average RMS noise level per bin and σ_{X_k} is the noise associated with the kth spectral estimate.

EXAMPLE 9.9

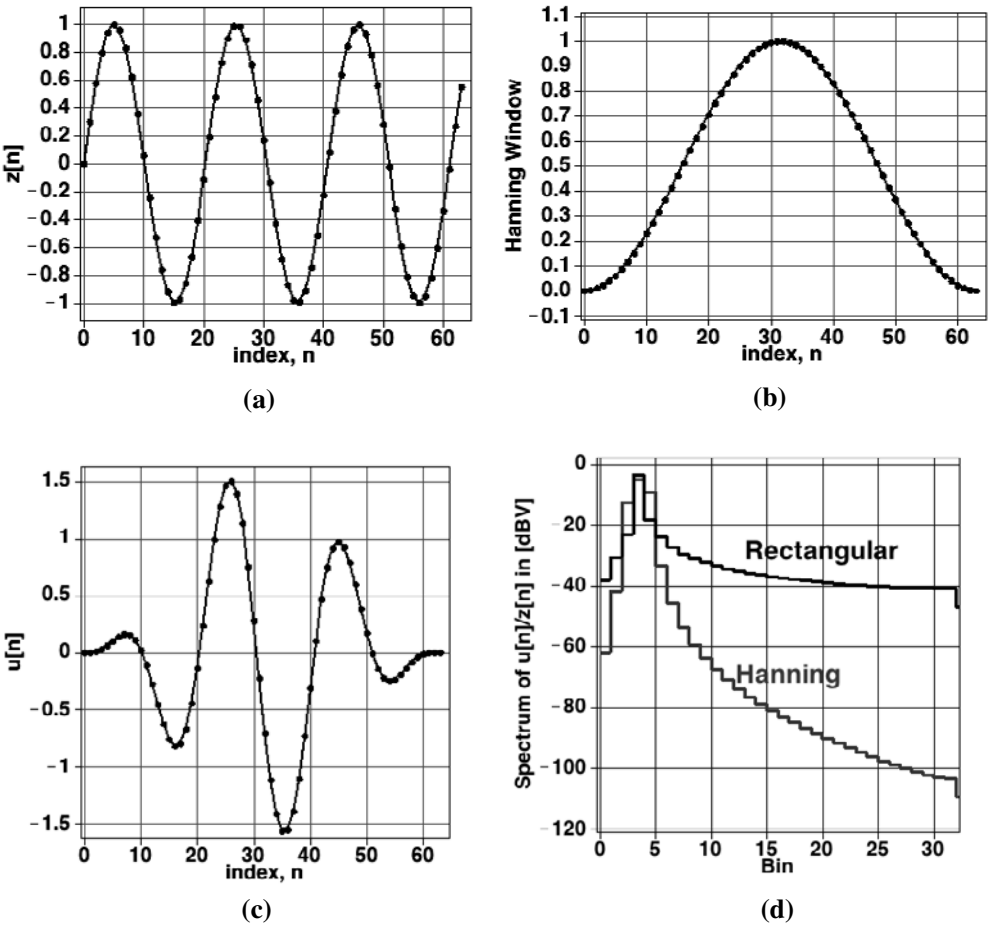
Through the application of a Hanning window, compute the magnitude of the spectra of the noncoherent waveforms described in Examples 9.7 and 9.8 consisting of 64 and 8192 samples, respectively. Compare the spectra with the results from a rectangular window (i.e., the nonwindowed results obtained previously).

Solution:
Let us begin by investigating the spectral properties of the noncoherent waveform of Example 9.7, consisting of 64 samples (Figure 9.12). A MATLAB script was written to perform the windowing operation. The code is

```
% noncoherent signal definition–z -
    NOI=64;                    % observation interval
    N=64; M=pi; A=1; P=0;              % signal definition
    for n=1:NOI,
        z(n)=A*sin(2*pi*M/N*(n-1)+P);
    end;
% Hanning windowing operation
    for n=1:NOI,
        w(n)=0.5*(1-cos(2*pi*(n-1)/(N-1)));
    end;
    Gnoncoherent=sqrt(sum(w.*w)/NOI);
    u=1/Gnoncoherent * z .* w;
```

The results of running the above code is shown in Figure 9.20. Specifically, in Figure 9.20a the noncoherent sine wave $z[n]$ is shown plotted over 64 points. In Figure 9.20b, 64 points associated with the window function $w[n]$ are shown. The corresponding window data sequence, with the appropriate noncoherent gain factor accounted for, is shown in Figure 9.20c. As can be seen from Table 9.2, the noncoherent gain for the Hanning window is 0.612. Here we see that the window effectively squeezes the endpoints of the noncoherent waveform toward zero, forcing the endpoints of the waveform to meet smoothly. An FFT was then performed on the modified waveform from which the magnitude of the spectrum was derived. The windowed spectrum is shown in Figure 9.20d. Superimposed on the plot is the spectrum of the original noncoherent waveform derived in Example 9.6 without windowing (i.e., rectangular windowing). As is evident, the Hanning windowed spectrum is much more concentrated about a single frequency than the rectangular windowed spectrum.

Figure 9.20. Windowing results (a) original noncoherent waveform, (b) Hanning window, (c) windowed data, and (d) spectrum magnitude.



In order to estimate the amplitude of the signal present in the windowed data, an expanded view of the spectrum about its peak is shown in Figure 9.21. As side tones are present about the peak value of the spectrum, we must consider these tones in the estimate of the signal amplitude. Performing a square-root-of-sum-of-squares calculation involving the RMS value of the signals present in bins 1 to 5 (within the main lobe bandwidth of the window), we obtain a combined RMS value of 0.7070567. This, in turn, implies an amplitude estimate of 0.99993. For all intents and purposes, this is unity and was found with only 64 samples. Unless improved frequency resolution is necessary, preprocessing with the Hanning window is just as effective as the coherent measurement in this example. However, repeatability of measurements is degraded in windowed systems in the presence of random noise, because the processing gain is less than unity.

If we increase the number of samples to 8192 and repeat the windowing operation, we find the spectrum much more closely concentrated about a single frequency. The results are shown

Figure 9.21. Expanded view of the windowed, noncoherent tone around the spectral peak.

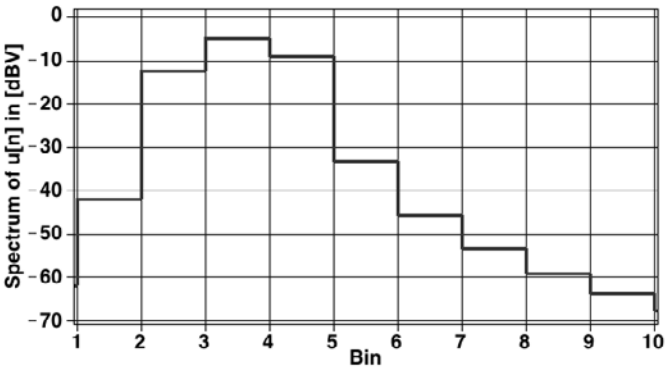
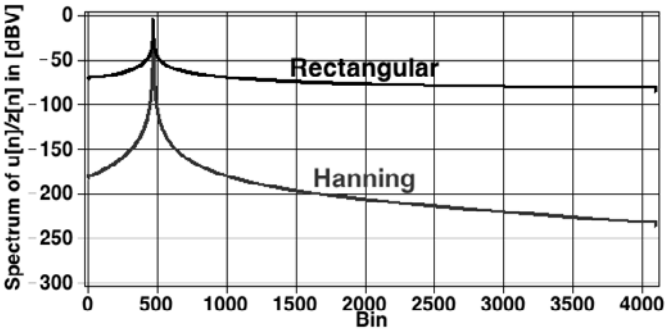


Figure 9.22. Spectral comparison of rectangular and Hanning windows.



in Figure 9.22, together with the spectrum resulting from a rectangular window. Clearly, the Hanning window has a very narrow spectrum, very much like the coherent case. It would be suitable for making measurements in situations where more than one tone is present in the input signal. The drawback, of course, is that a longer observation interval is necessary and that side tones have to be dealt with.

In the previous example, windowing was used to improve the measurement of a noncoherent sinusoidal signal. In fact, an accurate estimate was obtained without extending the observation interval over and above that of a coherent sinusoidal signal. This example may incorrectly give the reader the impression that windowing can resolve the frequency leakage problem associated with noncoherent signaling with no added time expense. As we shall see in this next example consisting of a noncoherent multitone signal, this is indeed not the case.

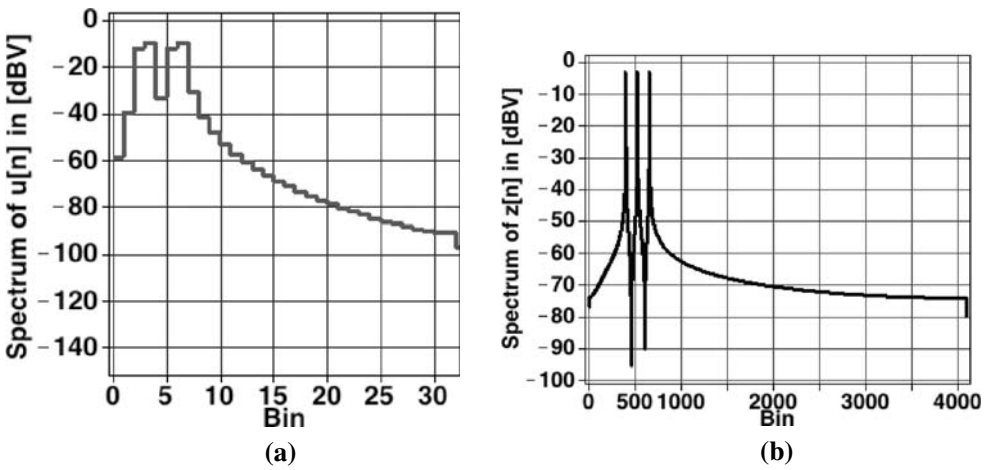
EXAMPLE 9.10

Determine the magnitude of the spectrum of a multitone signal consisting of three noncoherent tones with parameters, $A = 1$, $\phi = 0$, $M = \pi$, $N = 64$; $A = 1$, $\phi = 0$, $M = \pi + 1$, $N = 64$; and $A = 1$, $\phi = 0$, $M = \pi + 2$, $N = 64$.

Solution:
The following MATLAB routine was written to generate the three-tone multitone signal with a Hanning window over a 64-sample observation interval:

```
% noncoherent 3-tone multitone signal definition
    NOI=64;             % observation interval
    N=64;               % signal definition
    M1=pi; A1=1; P1=0;
    M2=pi+1; A2=1; P2=0;
    M3=pi+2; A3=1; P3=0;
    for n=1: NOI,
        z(n)=A1*sin(2*pi*M1/N*(n-1)+P1) +
             A2*sin(2*pi*M2/N*(n-1)+P2) +
             A3*sin(2*pi*M3/N*(n-1)+P3);
    end;
% Hanning windowing operation
    for n=1:NOI,
        w(n)=0.5*(1-cos(2*pi*(n-1)/(N-1)));
    end;
    Gnoncoherent=sqrt(sum(w.*w)/NOI);
    u=1/Gnoncoherent * z .* w;
```

Figure 9.23. Windowed three-tone multitone spectra with (a) *NOI* = 64 and (b) 8192.



The routine was executed and the corresponding magnitude of the frequency spectrum was derived using the frequency-domain conversion routine of Example 9.7. The result is shown in Figure 9.23 on the left-hand side. The plot extends over the Nyquist frequency range of 32 points.

Surprisingly, the spectrum appears to have only two spectral peaks. It is as if only two tones were present in the input signal. This is a direct result of frequency leakage. If the observation interval is increased, say, to 8192 samples, then better frequency resolution is obtained and a clear separation of each frequency component is evident. This is shown on the right-hand side of Figure 9.23. With the side tones around each spectral peak below 100 dB, each tone can be accurately measured with a relative error of no more than 0.001% using the method of the previous example.

9.4 THE INVERSE FFT

9.4.1 Equivalence of Time- and Frequency-Domain Information

A discrete Fourier transform produces a frequency-domain representation of a discrete-time waveform. This was shown to be equivalent to a discrete-time Fourier series representation of a periodically extended waveform. The transformation is lossless, meaning that all information about the original signal is maintained in the transformation. Since no information is lost in the transformation from the time domain to the frequency domain, it seems logical that we should be able to take a frequency-domain signal back into the time domain to reconstruct the original signal. Indeed, this is possible and can be seen directly from Eqs. (9.21) and (9.22). If we substitute the expression for the frequency-domain coefficients given by Eq. (9.22) back into (9.21), we clearly see that we obtain our original information

$$\mathbf{X} = \mathbf{W}\left[\mathbf{W}^{-1}\mathbf{X}\right] = \mathbf{W}\mathbf{W}^{-1}\mathbf{X} = \mathbf{X} \tag{9.74}$$

In practice, the form of the mathematics used to perform the frequency-to-time operation is very similar to that used to perform the time-to-frequency operation. In fact, the same FFT algorithm can be used, except for some possible array rearrangements and some predictable scale factor changes. When the FFT is used to perform the frequency-to-time transformation, it is referred to as an *inverse-FFT*. The calculations are so similar that some testers perform an inverse FFT using the same syntax as the forward FFT. A flag is set to determine whether the FFT is forward or inverse.

It is worth noting that the magnitude of a spectrum alone cannot be converted back into the time domain. Phase information at each test frequency must be combined with the magnitude information in the form of a complex number using either rectangular or polar notation. The specific format will depend on the vendor's operating system. Our next example will illustrate this procedure using MATLAB.

Exercises

9.16. The results of an FFT analysis of a noncoherent sinusoid indicate the following significant dBV values around the spectral peak: -38.5067, -36.3826, -33.6514, -29.7796, -22.9061, -7.7562, -25.3151. Estimate the amplitude of the corresponding tone.

ANS. 0.596.

9.17. By what factor does the repeatability of a set of spectral measurements made on a coherent signal decrease with the application of a Blackman window with signal power correction?

ANS. Sample set scaled by $1/G_{COHERENT} = 2.38$ followed by a RMS noise change of $G_{NONCOHERENT} = 0.552$ due to window shape, resulting in an increase RMS noise level of $G_{NONCOHERENT}/G_{COHERENT} = 1.31$ $\left(\sqrt{ENBW}\right)$. Repeatability therefore suffers by a factor of 1.31.

9.18. If a sampled sine wave corresponding to a bin of an FFT has an amplitude of 1 V, what is the expected amplitude observed through a Hanning window without correction?

ANS. 0.5 (= $G_{COHERENT}$).

9.19. If the average RMS noise voltage per bin associated with an N-point rectangular window FFT is 100 μV, what is the expected RMS noise level per bin if a Hanning window is used without correction assuming the input signal is noncoherent with the UTP? What would this noise level per bin be after correction? What is the noise level associated with a signal within the main lobe of the window?

ANS. 61.2 μV (observed directly); 100 μV (with correction); 200 μV.

EXAMPLE 9.11

A discrete-time signal is described by its DTFS representation as

$$x[n] = 1 + 2\cos\left[2\left(\frac{2\pi}{8}\right)n + \frac{\pi}{4}\right] + 0.5\cos\left[3\left(\frac{2\pi}{8}\right)n\right]$$

Determine the time-domain samples using MATLAB's inverse FFT routine.

Solution:
The inverse FFT is performed in MATLAB using a vector Y that contains the samples of the Fourier transform $Y(k)$ in complex form. As described in Section 9.3, these are related to the coefficients of the DTFS according to

$$Y(k) = NX(k)$$

To obtain $X(k)$, we first note from $x[n]$ that the spectral coefficients of the DTFS written in trigonometric form are immediately obvious as

$$\begin{bmatrix} \tilde{c}_0 \\ \tilde{c}_1 \\ \tilde{c}_2 \\ \tilde{c}_3 \\ \tilde{c}_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 0.5 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \tilde{\phi}_0 \\ \tilde{\phi}_1 \\ \tilde{\phi}_2 \\ \tilde{\phi}_3 \\ \tilde{\phi}_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\pi/4 \\ 0 \\ 0 \end{bmatrix}$$

Subsequently, $X = [X(0)\ X(1) \ldots X(N-1)]$ and $Y = [Y(0)\ Y(1) \ldots Y(N-1)]$ with $N = 8$ are determined from Eq. (9.43), and from their interrelationship described, to be

$$\mathrm{X} = \begin{bmatrix} 1 \\ 0 \\ 0.7071 + j0.7071 \\ 0.25 \\ 0 \\ 0.25 \\ 0.7071 - j0.7071 \\ 0 \end{bmatrix} \quad \Rightarrow \quad \mathrm{Y} = 8 \quad \mathrm{X} = \begin{bmatrix} 8 \\ 0 \\ 5.6568 + j5.6568 \\ 2 \\ 0 \\ 2 \\ 5.6568 - j5.6568 \\ 0 \end{bmatrix}$$

Submitting the Y vector to MATLAB's inverse FFT routine, we obtain the following time-domain samples:

$$\boldsymbol{x} = IFFT\{\mathrm{Y}\} = \begin{bmatrix} 2.9142 \\ -0.7678 \\ -0.4142 \\ 2.7678 \\ 1.9141 \\ -0.0606 \\ -0.4142 \\ 2.0606 \end{bmatrix}$$

Of course, these time-domain samples agree with those obtain by evaluating $x[n]$ directly at the sampling instances.

Exercises

9.20. Using MATLAB, compute the IFFT of the following sequence of complex numbers: {1.875, 0.75–*j*0.375, 0.625, 0.75+*j*0.375}.

ANS. {1.0, 0.5, 0.25, 0.125}.

9.21. For the following signal $x[n]$, compute the RMS value of each frequency component (beginning with DC):

$$x[n] = 1 + 2\cos\left[2\left(\frac{2\pi}{8}\right)n + \frac{\pi}{4}\right] + 0.5\cos\left[3\left(\frac{2\pi}{8}\right)n\right]$$

Using Parseval's theorem, compute the RMS value of $x[n]$.

ANS. 1, 1.414, 0.3536; 1.7676.

9.4.2 Parseval's Theorem

The RMS value X_{RMS} of a discrete time periodic signal $x[n]$ is defined as the square root of the sum of the individual samples squared, normalized by the number of samples N, according to

$$X_{RMS} = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]} \tag{9.75}$$

Parseval's theorem for discrete-time periodic signals states that the RMS value X_{RMS} of a periodic signal $x[n]$ described by a DTFS written in trigonometric terms is given by

$$X_{RMS} = \sqrt{c_0^2 + \frac{1}{2}\sum_{k=1}^{N/2} c_k^2} \tag{9.76}$$

When the magnitude of each spectral coefficient is expressed as an RMS value, Eq. (9.76) can be rewritten as a square-root-of-sum-of-squares calculation given by

$$X_{RMS} = \sqrt{\sum_{k=0}^{N/2} c_{k-RMS}^2} \tag{9.77}$$

In this text, we shall make greatest use of this form of Parseval's theorem. It is the easiest form to remember, because there are no extra scale factors to keep track of.

The importance of Parseval's theorem is that it allows the computation of the RMS value associated with all aspects of a signal such as distortion, noise, and so on, to be made directly from the DTFS description. For example, the RMS noise level associated with a signal is the square root of sum of squares of all bins that do not contain signal-related power or DC given by

$$X_n = \sqrt{\sum_{k=1, k\neq signal}^{N/2} c_{k-RMS}^2} \tag{9.78}$$

9.4.3 Frequency-Domain Filtering

A useful feature of DSP-based testing is the ability to apply arbitrary filter functions to a set of collected samples, simulating electronic filters in traditional analog instrumentation. Filtering can be applied either in the time domain or in the frequency domain. Time-domain filtering is accomplished by convolving the sampled waveform by the impulse response of the desired filter. Frequency-domain filtering is performed by multiplying the signal spectrum by the desired filter's frequency response. Frequency-domain filtering is faster to implement than time-domain filtering, because the spectrum of a signal is already available in the computer. As such, time-domain filtering is rarely used in mixed-signal testing applications.

Filter functions are selected to provide a pre-described frequency response such as a low-pass or bandpass behavior. For instance, a low-pass filter is used to suppress the high-frequency content of a signal while allowing the low-frequency energy to pass relatively unchanged. In another application, filters are used to alter the phase of a signal to improve its transient behavior (i.e., reduce ringing). In general, the z-domain transfer function $H(z)$ of an arbitrary discrete-time filter is described by

$$H(z) = \frac{a_0 + a_1 z^1 + \cdots + a_N z^N}{1 + b_1 z^1 + \cdots + b_N z^N} \tag{9.79}$$

From the convolution property of discrete-time Fourier transforms, the transform $Y_{out}\left(e^{j\omega T_s}\right)$ of the filter output is related to the transform $Y_{in}\left(e^{j\omega T_s}\right)$ of the filter input according to

$$Y_{out}\left(e^{j\omega T_s}\right) = H\left(e^{j\omega T_s}\right) Y_{in}\left(e^{j\omega T_s}\right) \tag{9.80}$$

If we limit the input to discrete frequencies

$$\omega = \frac{2\pi}{N} k, \qquad k = 0, 1, \ldots, N-1 \tag{9.81}$$

Eq. (9.80) can be rewritten as

$$Y_{out}\left(e^{j\frac{2\pi}{N}kT}\right) = H\left(e^{j\frac{2\pi}{N}kT_s}\right) Y_{in}\left(e^{j\frac{2\pi}{N}kT_s}\right) \tag{9.82}$$

or, with the shorthand notation of section 9.3.1, for $k = 0, 1, \ldots, N-1$, as

$$Y_{out}(k) = H(k) Y_{in}(k) \tag{9.83}$$

or in vector form as

$$\mathbf{Y}_{out} = \mathbf{H} \cdot \mathbf{Y}_{in} \tag{9.84}$$

where the dot operation represents an inner product. If we define the DTFS transforms for the input and output signals, that is, $\mathbf{Y}_{in} = \mathrm{FFT}\left[\mathbf{y}_{in}/N\right]$ and $\mathbf{Y}_{out} = \mathrm{FFT}\left[y_{in}/N\right]$, then Eq. (9.81) can be written as

$$\mathrm{FFT}\left[\frac{\mathbf{y}_{out}}{N}\right] = \mathbf{H} \cdot \mathrm{FFT}\left[\frac{\mathbf{y}_{in}}{N}\right] \tag{9.85}$$

Taking the inverse FFT of each side, we obtain the time-domain samples of the output signal as

$$\mathbf{y}_{out} = N \cdot \text{IFFT}\left\{\mathbf{H} \cdot \text{FFT}\left[\frac{\mathbf{y}_{in}}{N}\right]\right\} \tag{9.86}$$

which is further reduced to

$$\mathbf{y}_{out} = \text{IFFT}\left\{\mathbf{H} \cdot \text{FFT}\left[\mathbf{y}_{in}\right]\right\} \tag{9.87}$$

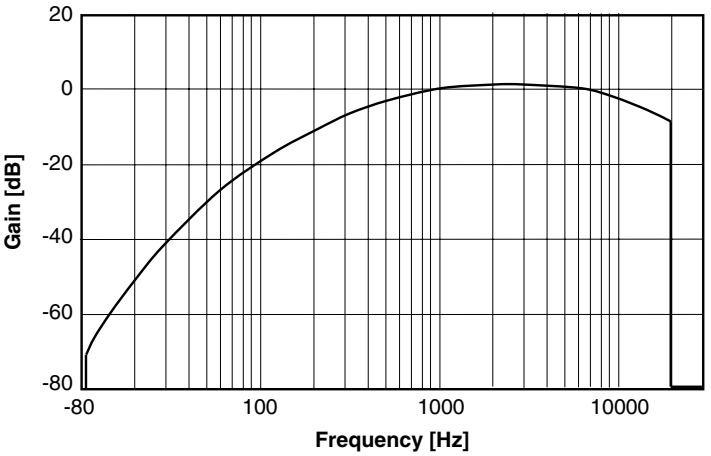
as $\alpha \cdot FFT\left[x\right] = FFT\left[\alpha \cdot x\right]$. The procedure to filter a signal in the frequency domain is now clear. We first perform an FFT operation on the input signal to bring it into the frequency domain. Next, each frequency component of the input is scaled by the corresponding filter response $H(k)$ to produce the filtered frequency-domain output. Finally, an inverse FFT can be performed on the filtered output to produce the filtered time-domain signal. Often, this last step can be eliminated because we can extract all desired information from the filtered spectrum in the frequency domain.

9.4.4 Noise Weighting

Noise weighting is one common example of DSP-based filtering in mixed-signal test programs. Weighting filters are called out in many telecom and audio specifications because the human ear is more sensitive to noise in some frequency bands than others. The magnitude of the frequency response of the A-weighting filter is shown in Figure 9.24. It is designed to approximate the frequency response of the average human ear. For matters related to hearing, phase variations have little effect on the listener and are therefore ignored in noise-related tests. By weighting the noise from a telephone or audio circuit before measuring its RMS level, we can get a more accurate idea of how good or bad the telephone or audio equipment will sound to the consumer.

In traditional bench instruments, the weighting filter is applied to the analog signal before it is passed to an RMS voltmeter. In DSP-based testing, we can perform the same filtering function mathematically, after the unweighted DUT signal has been sampled. Application of a mathematical filter to a sampled waveform means the ATE tester does not have to include a physical A-weighting filter in its measurement instruments. The resulting reduction in tester complexity reduces tester cost and improves reliability. Application of the A-weighting filter is a simple matter of multiplying its magnitude by the magnitude of the FFT of the signal under test.

Figure 9.24. A-weighting filter magnitude response.



Exercises

9.22. The gain and phase of a particular system at 1 kHz is 0.8 and -π/4, respectively. Determine the spectral coefficient of the DTFS that corresponds to the output signal at 1 kHz when excited by a signal with a spectral coefficient described by 0.25 – *j*0.35. Express the result in rectangular and polar form.

ANS. –0.0566 – *j*0.3394, 0.3441e+j1.7341.

A very simple form of noise filtering can be used to measure the noise over a particular bandwidth. For example, if a specification calls for a noise level of 10 μV RMS over a band of 100 Hz to 1 kHz, then we can simply apply a brick wall bandpass filter to the FFT results, eliminating all noise components that do not fall within this frequency range. The remaining noise can then be measured by performing an inverse FFT followed by an RMS calculation. The same results can be also achieved by adding up the signal power in all spectral bins from 100 Hz to 1 kHz. To do this we simply square the RMS value of each frequency component from 100 Hz to 1 kHz, add them all together, and then take the square root of the total to obtain the RMS value of the noise according to

$$V_{N-rms} = \sqrt{\sum_{k=B_L}^{B_U} c_{k-RMS}^2} \tag{9.88}$$

Here B_L and B_U are the spectral bins corresponding to the lower and upper frequencies of the brick wall filter (excluding any DC component). In this particular case, B_L and B_U correspond to the 100 Hz and 1 kHz frequencies, respectively.

9.5 SUMMARY

Coherent DSP-based testing allows the mixed-signal test engineer to perform AC measurements in a few tens of milliseconds. These same measurements might otherwise take hundreds or thousands of milliseconds using traditional analog bench instruments. The AWG, digitizer, source memory, and capture memory of a mixed-signal ATE tester allow us to translate signals between continuous time and sampled time. Digital signal processing operations such as the FFT and inverse FFT allow us to perform operations that are unavailable using traditional non-DSP measurement methodologies. Time-domain interpolations, frequency-domain filtering, and noise reduction functions are just a few of the powerful operations that DSP-based testing makes available to the accomplished mixed-signal test engineer.

We are fortunate in mixed-signal ATE to be able to use coherent sampling systems to bypass mathematical windowing. Bench instruments such as spectrum analyzers and digitizing oscilloscopes must use windowing extensively. The signals entering a spectrum analyzer or oscilloscope are generally noncoherent, since they are not synchronized to the instrument's sampling rate. However, a spectrum analyzer is not usually expected to produce an accurate reading in only 25 ms; thus it can overcome the repeatability problem inherent in windowing by simply averaging results or collecting additional samples. ATE equipment must be fast as well as accurate; thus windowing is normally avoided whenever possible. Fortunately, mixed-signal testers give us

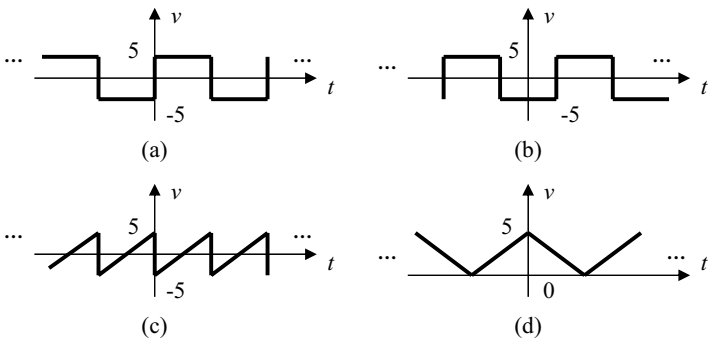
control of both the signal source and sampling processes during most tests. Synchronization of input waveforms and sampling processes affords us the tremendous accuracy/cost advantage of coherent DSP-based testing.

Despite its many advantages, DSP-based testing also places a burden of knowledge upon the mixed-signal test engineer. Matthew Mahoney, author of "DSP-Based Testing of Analog and Mixed-Signal Circuits,"[7] once told an amusing story about a frustrated student in one of his DSP-based testing seminars. Exasperated by the complexity of digital signal processing, the distressed student suddenly exclaimed "But this means we must know something!" Indeed, compared to the push-the-button/read-the-answer simplicity of bench equipment, DSP-based testing requires us to know a whole lot of "something."

The next two chapters will explore the use of DSP-based measurements in testing analog and sampled channels. In Chapter 10, "Analog Channel Testing," we will explore the various types of DSP-based tests that are commonly performed on nonsampled channels such as amplifiers and analog filters. Chapter 11, "Sampled Channel Testing," will then extend these DSP-based testing concepts to sampled channels such as ADCs, DACs, and switched capacitor filters.

PROBLEMS

9.1 Find the trigonometric Fourier series representation of the functions displayed in (a)–(d). Assume that the period in all cases is 1 ms.



9.2. Find the trigonometric Fourier series representation of a sawtooth waveform $x(t)$ having a period of 2 s and whose behavior is described by $x(t) = t$ if $-1 < t < 1$. Using MATLAB, numerically compare your FS representation with $x(t)$.

9.3. Express the Fourier series in problem 9.2 using cosine terms only.

9.4. Using the summation formulae in Eq. (9.13), determine the DTFS representation of the Fourier series representation of $x(t)$ given in Exercise 9.1. Use 10 points per period and limit the series to 100 terms. Also, express the result in magnitude and phase form.

9.5. A sampled signal consists of the following six samples: {0.1, 0.1, 1, –3, 4, 0}. Using linear algebra, determine the DTFS representation for these samples. Also, express the result in magnitude and phase form. Using MATLAB, numerically compare the samples from your DTFS representation with the actual samples given here.

9.6. A sampled signal consists of the following set of samples {0, –1, 2.4, 4, –0.125, 3.4}. Determine the DTFS representation for this sample set using the orthogonal basis method. Also, express the result in magnitude and phase form. Using MATLAB, numerically compare the samples from your DTFS representation with the actual samples given here.

9.7. Derive Eqs. (9.24) and (9.25) from first principles. Begin by multiplying the DTFS representation given in Eq. (9.12) by $\cos\left[k\left(2\pi/N\right)n\right]$, then sum n on both sides from 0 to N–1. Reduce the expression by using the set of trigonometric identities given in Eq. (9.23). Repeat using $\sin\left[k\left(2\pi/N\right)n\right]$.

9.8. Plot the magnitude and phase spectrum of the following FS representations of $x(t)$:

$$\text{(a)}\quad x(t)=\frac{2}{\pi}\left(\sin\pi t-\tfrac{1}{2}\sin 2\pi t+\tfrac{1}{3}\sin 3\pi t-\tfrac{1}{4}\sin 4\pi t+-\cdots\right)$$

$$\text{(b)}\quad x(t)=\frac{2}{\pi}\left[\cos\left(\pi t-\frac{\pi}{2}\right)+\tfrac{1}{2}\cos\left(2\pi t+\frac{\pi}{2}\right)+\tfrac{1}{3}\cos\left(3\pi t-\frac{\pi}{2}\right)+\tfrac{1}{4}\cos\left(4\pi t+\frac{\pi}{2}\right)+\cdots\right]$$

$$\text{(c)}\quad x(t)=-1.5+\sum_{k=1,\,k\ \text{odd}}^{\infty}\frac{1}{k\pi}\cos\left(k\ 2\pi\times 10^{3}\ t+\pi/2\right)$$

9.9. Plot the magnitude and phase spectrum of the following DTFS representations of $x[n]$:

$$\text{(a)}\quad x[n]=0.6150\ \sin\left[\left(\frac{2\pi}{10}\right)n\right]-0.2749\ \sin\left[2\left(\frac{2\pi}{10}\right)n\right]+0.1451\ \sin\left[3\left(\frac{2\pi}{10}\right)n\right]-0.0649\ \sin\left[4\left(\frac{2\pi}{10}\right)n\right]$$

$$\text{(b)}\quad x[n]=1+\cos\left[\left(\frac{2\pi}{8}\right)n\right]+\sin\left[\left(\frac{2\pi}{8}\right)n\right]+\cos\left[2\left(\frac{2\pi}{8}\right)n\right]+\sin\left[2\left(\frac{2\pi}{8}\right)n\right]-\cos\left[3\left(\frac{2\pi}{8}\right)n\right]-\sin\left[3\left(\frac{2\pi}{8}\right)n\right]$$

9.10. A DTFS representation expressed in complex form is given by

$$x[n]=1+0.2e^{j\left[\left(\frac{2\pi}{8}\right)n-\frac{\pi}{4}\right]}+0.3e^{j\left[3\left(\frac{2\pi}{8}\right)n+\frac{\pi}{3}\right]}+0.3e^{j\left[5\left(\frac{2\pi}{8}\right)n-\frac{\pi}{3}\right]}+0.2e^{j\left[7\left(\frac{2\pi}{8}\right)n+\frac{\pi}{4}\right]}$$

Express $x[n]$ in trigonometric form. Plot the magnitude and phase spectra.

9.11. A DTFS representation expressed in complex form is given by

$$x[n]=1+(1.8-j1.9)e^{j\left[2\left(\frac{2\pi}{8}\right)n\right]}+(0.75+j0.25)e^{j\left[3\left(\frac{2\pi}{8}\right)n\right]}+(0.75-j0.25)e^{j\left[5\left(\frac{2\pi}{8}\right)n\right]}+(1.8+j1.9)e^{j\left[6\left(\frac{2\pi}{8}\right)n\right]}$$

Express $x[n]$ in trigonometric form and plot the corresponding magnitude and phase spectra.

9.12. Derive the polar form of Eq. (9.43) from (9.42).

9.13. Sketch the periodic extension for the following sequence of points:

(a) [0, 0.7071, 1.0, 0.7071, 0.0, –0.7071, –1.0000, –0.7071]

(b) [0, 0.7071, 1.0, 0.7071, 0.0, –0.7071, –1.0, –0.7071, 0.0, 0.7071]

9.14. Using the FFT algorithm in MATLAB or some other software package, verify your answers to Problems 9.4, 9.5, and 9.6.

9.15. Given

$$x[n] = 0.25 + 0.5\cos\left[\left(\frac{2\pi}{8}\right)n\right] + 0.1\sin\left[\left(\frac{2\pi}{8}\right)n\right] + 2.1\cos\left[2\left(\frac{2\pi}{8}\right)n\right] - 0.9\cos\left[3\left(\frac{2\pi}{8}\right)n\right] - 0.1\sin\left[3\left(\frac{2\pi}{8}\right)n\right]$$

Express $x[n]$ in complex form.
Using MATLAB, write a script that samples $x[n]$ for $n = 0,1,\ldots,7$.
Compute the FFT of the samples found in part (b) and write the corresponding DTFS representation in complex form. How does it compare with $x[n]$ found in part (a)?

9.16. Given

$$x[n] = (0.2 - j0.4)e^{j\left[\left(\frac{2\pi}{10}\right)n\right]} + (0.25 + j0.25)e^{j\left[3\left(\frac{2\pi}{10}\right)n\right]} + (0.25 - j0.25)e^{j\left[7\left(\frac{2\pi}{10}\right)n\right]} + (0.2 + j0.4)e^{j\left[9\left(\frac{2\pi}{10}\right)n\right]}$$

(a) Express $x[n]$ in trigonometric form.
(b) Using MATLAB, write a script that samples $x[n]$ for $n = 0,1,\ldots, 9$.
(c) Compute the FFT of the samples found in part (b) and write the corresponding DTFS representation in trigonometric form. How does it compare with $x[n]$ found in part (a)?

9.17. Evaluate $x[n] = n^3 - 2n^2 - 2n + 1$ for $n = 0,1,\ldots,15$. Using MATLAB, compute the FFT of the 16 samples of $x[n]$ and determine the corresponding $\{c_k\}$ and $\{\phi_k\}$ spectral coefficients. Express the magnitude coefficients in RMS form.

9.18. Investigate the effects of increasing the observation window on the spectrum of a noncoherent sinusoidal signal. Consider generating a sinusoidal signal using parameters $A = 1$, $\phi = 0$, $M = 9.9$, and $N = 64$. Next, compare the magnitude spectrum of this signal when the following number of samples are collected: (a) 64 samples, (b) 512 samples, (c) 1024 samples, (d) 8192 samples. In all cases, estimate the amplitude of the sinusoidal signal.

9.19. An observation window consists of 64 points. Plot the frequency response of (a) rectangular window, (b) Hanning window, (c) Blackman window, and (d) Kasier ($\phi = 10$) window all on the same graph. *Hint*: Determine the frequency response of a window function using the Fourier transform definition of Eq. (9.47).

9.20. Compute the window shape factors for the rectangular, Hanning, Blackman, and Kasier ($\beta = 10$) windows using the window function coefficients.

9.21. Repeat Problem 9.18 but view the data first through a Blackman window. Estimate the amplitude of the sinusoidal signal.

9.22. Repeat Problem 9.18 but view the data first through a Kasier ($\beta = 10$) window. Estimate the amplitude of the sinusoidal signal.

9.23. A signal has a period of 128 μs and is sampled at a rate of 1 MHz. If 128 samples are collected, what is the frequency resolution of the resulting FFT? If the number of samples collected increases to 8192 samples, what is the frequency resolution of the FFT?

9.24. Repeat Example 9.9, but this time use a Blackman window. By what factor does the accuracy of the calculation improve over the rectangular window?

9.25. Repeat Example 9.10, but this time use a Blackman window.

9.26. Using the FFT and IFFT routines found in MATLAB, together with the samples described in Exercise 9.12, verify that IFFT(FFT(x)) = x.

9.27. Using the trigonometric identities described in Eq. (9.23), derive the trigonometric form of Parseval's theorem given in Eq. (9.76).

9.28. Using the trigonometric form of Parseval's theorem in Eq. (9.76) as a starting point, derive the corresponding complex form of the theorem given in Eq. (9.77).

9.29. The complex coefficients of a spectrum of a sampled signal are

$$\{X_k\} = \{0.5,\ 0.2 - j0.4,\ 0,\ 0.25 + j0.25,\ 0,\ 0.25 - j0.25,\ 0,\ 0.2 + j0.4\}$$

What is the RMS value of this signal?

9.30. The magnitude coefficients of a spectrum of a sampled signal are
$\{c_k\} = \{0.1,\ 0.3,\ 0,\ 0.05,\ 0,\ 0.001\}$
What is the RMS value of this signal?

9.31. Find the coherent sample set of $x[n]$ using MATLAB's IFFT routine assuming its spectrum is described by the following:

(a) $$x[n] = 0.25 + 0.5\ \cos\left[\left(\frac{2\pi}{8}\right)n\right] + 0.1\ \sin\left[\left(\frac{2\pi}{8}\right)n\right] + 2.1\ \cos\left[2\left(\frac{2\pi}{8}\right)n\right] - 0.9\ \cos\left[3\left(\frac{2\pi}{8}\right)n\right] - 0.1\ \sin\left[3\left(\frac{2\pi}{8}\right)n\right]$$

(b) $$x[n] = 1 + 0.2e^{j\left[\left(\frac{2\pi}{8}\right)n - \frac{\pi}{4}\right]} + 0.3e^{j\left[3\left(\frac{2\pi}{8}\right)n + \frac{\pi}{3}\right]} + 0.3e^{j\left[5\left(\frac{2\pi}{8}\right)n - \frac{\pi}{3}\right]} + 0.2e^{j\left[7\left(\frac{2\pi}{8}\right)n + \frac{\pi}{4}\right]}$$

(c) $$x[n] = (0.2 - j0.4)e^{j\left[\left(\frac{2\pi}{10}\right)n\right]} + (0.25 + j0.25)e^{j\left[3\left(\frac{2\pi}{10}\right)n\right]} + (0.25 - j0.25)e^{j\left[7\left(\frac{2\pi}{10}\right)n\right]} + (0.2 + j0.4)e^{j\left[9\left(\frac{2\pi}{10}\right)n\right]}$$

9.32. Verify the samples in Problem 9.31 by evaluating the function at each sampling instant.

9.33. A signal with a DTFS representation given by

$$x[n] = 0.1 + 2\ \cos\left[\left(\frac{2\pi}{16}\right)n - \frac{\pi}{5}\right] + 0.5\ \cos\left[5\left(\frac{2\pi}{16}\right)n + \frac{\pi}{5}\right]$$

passes through a system with the following transfer function

$$H(z) = \frac{0.0020 + 0.00402z^{1} + 0.0020z^{2}}{1 - 1.5610z^{1} + 0.64135z^{2}}$$

What is the DTFS representation of the output signal?

9.34. A signal with noise is described by the following DTFS representation:

$$x[n] = 0.25 + 2\cos\left[\left(\frac{2\pi}{8}\right)n\right] + 10^{-5}\cos\left[2\left(\frac{2\pi}{8}\right)n\right]$$
$$-10^{-7}\cos\left[3\left(\frac{2\pi}{8}\right)n\right] - 10^{-5}\sin\left[3\left(\frac{2\pi}{8}\right)n\right] + 10^{-5}\cos\left[4\left(\frac{2\pi}{8}\right)n\right]$$

What is the RMS value of the noise signal that appears between bins 2 and 4?

REFERENCES

1. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, March 1989, ISBN 013216292X.
2. A. V. Oppenheim et al., *Signals and Systems*, Prentice Hall, Englewood Cliffs, NJ, 1997, ISBN 0138147574.
3. W. McC. Siebert, *Circuits, Signals, and Systems*, The MIT Press, Cambridge, MA, 1985, ISBN 0262192292.
4. R. W. Ramirez, *The FFT Fundamentals and Concepts*, Prentice Hall, Englewood Cliffs, NJ, 1985, ISBN 0133143864.
5. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications,* 3rd Edition, Prentice Hall, Englewood Cliffs, NJ, 1996, ISBN 0133737624.
6. F. J. Harris., *On the use of windows for harmonic analysis with the discrete fourier transform*, *Proceedings of the IEEE*, **66** (1), pp. 51–83, January 1978.
7. M. Mahoney, *Tutorial DSP-Based Testing of Analog and Mixed-Signal Circuits*, The Computer Society of the IEEE, Washington, D.C., 1987, ISBN 0818607858.