CHAPTER

6

Continuous, Discrete, and Fast Fourier Transform

6.1 INTRODUCTION

In this chapter, the Fourier transform is related to the complex Fourier series presented in Chapter 5 (see overview in Fig. 5.3). The Fourier transform in continuous time (or space) is referred to as the continuous Fourier transform (CFT). In Section 6.3, we develop a discrete version of the Fourier transform (DFT) and explore an efficient algorithm for calculating it. This efficient algorithm is known as the fast Fourier transform (FFT). In the next chapter (Chapter 7) we show an example of the use of the CFT in the reconstruction of images and an application of the DFT for calculation of the power spectra of simulated and recorded signals.

6.2 THE FOURIER TRANSFORM

The Fourier transform is an operation that transforms data from the time (or spatial) domain into the frequency domain. In this section we demonstrate that the transform can be considered as the limiting case of the complex Fourier series.

In Fig. 6.1 we illustrate how one can create an arbitrary function $f(t)$ from a periodic signal $f_{T_0}(t)$ by increasing its period T_0 to ∞. This thought process is the basis on which the CFT is derived from the complex Fourier series. For clarity, we reiterate the expressions for the complex series and its coefficients that we derived in Chapter 5:

$$P(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega t}, \tag{6.1a}$$

Signal Processing for Neuroscientists, Second Edition
https://doi.org/10.1016/B978-0-12-810482-8.00006-0

© 2018 Elsevier Ltd. All rights reserved.

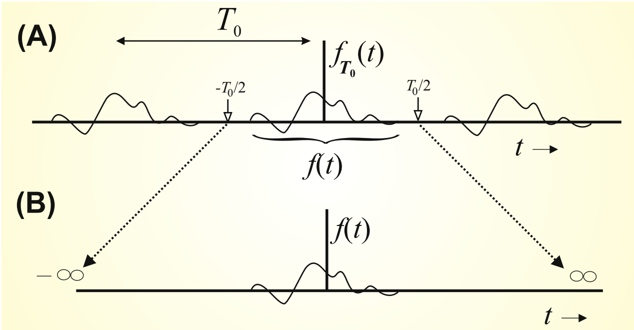

FIGURE 6.1 (A) Periodic function $f_{T_0}(t)$ and (B) function $f(t)$ derived from one cycle.

with the coefficients defined as:

$$c_n = \frac{1}{T}\int_T f(t)e^{-jn\omega t}dt \tag{6.1b}$$

The first step is to establish the coefficient c_n for the series $f_{T_0}(t)$ in Fig. 6.1A. Hereby we integrate over ***one cycle*** of the function $f_{T_0}(t)$, which equals $f(t)$ over the period from $-T_0/2$ to $T_0/2$:

$$c_n = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} f(t)e^{-jn\omega_0 t}dt \tag{6.2}$$

In Eq. (6.2) above, we include the period T_0, and the fundamental angular frequency ω_0. The relationship between these parameters is given by: $T_0 = \frac{1}{f_0} = \frac{2\pi}{\omega_0}$ (with f_0 the fundamental frequency in Hz). So far we are still simply applying the complex Fourier series to $f(t)$ as one cycle of $f_{T_0}(t)$. The second step is to stretch the period parameter T_0 to ∞ (which also means that the fundamental angular frequency $\omega_0 \rightarrow 0$), and to define c_n^* as:

$$c_n^* = \lim_{\substack{T_0 \rightarrow \infty \\ \omega_0 \rightarrow 0}} \int_{-T_0/2}^{T_0/2} f(t)e^{-jn\omega_0 t}dt \tag{6.3}$$

The coefficient c_n^* in Eq. (6.3) is very similar to the limit of c_n in Eq. (6.2) but ***note that we smuggled a 1/T_0 factor out of the expression!*** Stated a bit more formally we say that c_n^* is defined by Eq. (6.3), thereby avoiding a division by $T_0 \rightarrow \infty$. Because $\omega_0 \rightarrow 0$, we may consider the product $n\omega_0$ a continuous scale of the angular frequency ω $\left(\text{i.e., } \lim_{\omega_0 \rightarrow 0} n\omega_0 = \omega\right)$. Further, representing the complex coefficients c_n^* in a function $F(j\omega)$, we may rewrite Eq. (6.3) as:

$$\boxed{F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt} \tag{6.4}$$

Eq. (6.4) is the definition of the continuous Fourier transform (CFT). In some texts $F(\omega)$ is used instead of $F(j\omega)$, subsuming the constant j into the right-hand side. Another common notation substitutes $\omega = 2\pi f$ resulting in: $F(f) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi ft}dt$, which simply represents the results in units of Hz.

Given Eq. (6.4) we will now show that the inverse transform also follows from the complex Fourier series. Combining Eqs. (6.1b) and (6.4) and using $\omega = n\omega_0$, we have:

$$c_n = \frac{1}{T_0}F(jn\omega_0) \tag{6.5}$$

Note that the 1/T_0 factor is reintroduced! The $1/T_0$ correction maintains the consistency of the transform with its inverse.

Using Eq. (6.1a) for the complex Fourier series:

$$\begin{aligned} f_{T_0}(t) &= \sum_{n=-\infty}^{\infty} \frac{1}{T_0}F(jn\omega_0)e^{jn\omega_0 t}, \text{ using } \frac{1}{T_0} = \frac{\omega_0}{2\pi} \rightarrow \\ f_{T_0}(t) &= \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} F(jn\omega_0)e^{jn\omega_0 t}\omega_0 \end{aligned} \tag{6.6}$$

Now we let $T_0 \rightarrow \infty$ meaning that $\omega_0 \rightarrow 0$. If we change the notation $\omega_0 = \Delta\omega$ to use as a limiting variable, Eq. (6.6) becomes:

$$f(t) = \lim_{\substack{T_0 \rightarrow \infty \\ \omega_0 \rightarrow 0}} f_{T_0}(t) = \lim_{\Delta\omega \rightarrow 0} \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} F(jn\Delta\omega)e^{jn\Delta\omega t}\Delta\omega \tag{6.7}$$

We can interpret the sum in Eq. (6.7) as calculating the area under the continuous function $F(j\omega)\,e^{j\omega t}$ using arbitrarily narrow slices in the limit. This interpretation generates the result for the inverse Fourier transform:

$$\boxed{f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(j\omega)e^{j\omega t}d\omega} \tag{6.8}$$

If $\omega = 2\pi f$ is substituted in Eq. (6.8) we get: $f(t) = \int_{-\infty}^{\infty} F(f)e^{j2\pi ft}df$.

Eqs. (6.4) and (6.8) form a consistent pair for the Fourier transform and the inverse Fourier transform, respectively. With the exception of the $1/T_0$ factor, there is a direct correspondence between the transform and the complex series, whereby the transform can be considered as a series in the limit where $T_0 \rightarrow \infty$. Both the equations for the transform and its inverse apply for continuous time or space. Therefore this flavor of spectral analysis is called the CFT.

6.2.1 Examples of Continuous Fourier Transform Pairs

Eqs. (6.4) and (6.8) can be used to establish Fourier transform pairs; for complicated functions, it is common practice to use tables (e.g., Table 6.1) that summarize the pairs, or software tools to determine integrals. Here we focus on a few simple examples and associated interpretations relevant for signal analysis. First we consider the signal $\delta(t)$, known as the Dirac delta function; its Fourier transform is given by:

$$F(j\omega) = \int_{-\infty}^{\infty} \delta(t)e^{-j\omega t}dt = e^{-j\omega 0} = 1 \tag{6.9}$$

The above integral is evaluated using the sifting property of the delta function (Eq. 2.8). From a signal processing standpoint it is interesting to see that ***the time domain Dirac delta function corresponds to all frequencies in the frequency domain***.

We noticed when discussing the transforms that the equations for the Fourier transform and its inverse are fairly similar. Repeating Eqs. (6.4) and (6.8):

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \Leftrightarrow f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(j\omega)e^{j\omega t}d\omega \tag{6.10}$$

Clearly the transform and its inverse are identical with the exception of the $1/2\pi$ scaling factor and the sign of the imaginary exponent. This means that one can derive the inverse transform from the transform and vice versa by correcting for the scaling and the sign of the variable over

TABLE 6.1 Examples of Fourier Transform Pairs

Time/Spatial Domain $f(t)$	**Frequency Domain** $F(\omega)$
$\delta(t)$	1
1	$2\pi\,\delta(\omega)$
$\cos(\omega_0 t)$	$\pi[\delta(\omega+\omega_0)+\delta(\omega-\omega_0)]$
$\sin(\omega_0 t)$	$j\pi[\delta(\omega+\omega_0)-\delta(\omega-\omega_0)]$

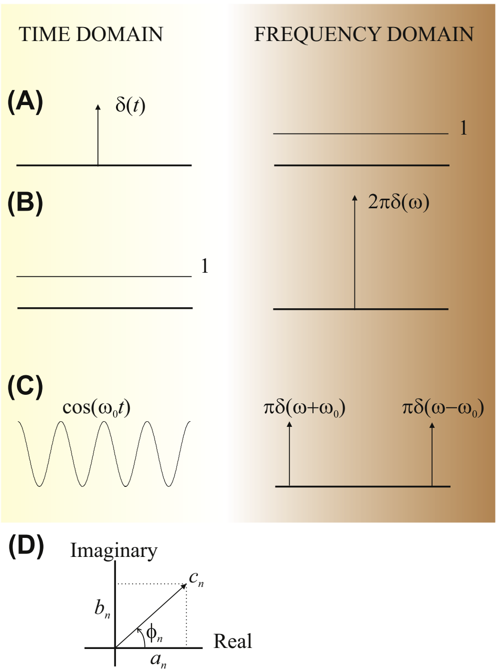

FIGURE 6.2 Common Fourier transform pairs. (A) A Dirac impulse function in the time domain is represented by all frequencies in the frequency domain. (B) This relationship can be reversed to show that a DC component in the time domain generates an impulse function at a frequency of zero. (C) A pure (cosine) wave shows peaks at $\pm\,\omega_0$ in the frequency domain. (D) In general a coefficient c_n being part of $F(j\omega)$ may contain both real (a_n) and imaginary (b_n) parts (represented here in a polar plot).

which one integrates. This is the so-called ***duality property*** which results in some interesting and useful parallels between time and frequency domain representations (Fig. 6.2).

Applying the above Fourier transform and inverse transform relationship to the Dirac impulse $\delta(t)$, one can conclude that the time domain equivalent for a delta function in the frequency domain $\delta(-\omega)$ must be the constant function $f(t) = 1/2\pi$. Because the scaling is a constant (not depending on ω) and $\delta(-\omega) = \delta(\omega)$, one can say that $1 \Leftrightarrow 2\pi\delta(\omega)$ forms a time domain—frequency domain pair; or in a different notation:

$$F(j\omega) = 2\pi\delta(\omega) = \int_{-\infty}^{\infty} 1e^{-j\omega t}dt \qquad (6.11)$$

This outcome can be validated by evaluating the inverse Fourier transform $f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}[2\pi\delta(\omega)]e^{j\omega t}d\omega$. Using the sifting property of $\delta(\omega)$

it can be seen that this expression evaluates to 1. Interpreting this property from a signal processing point of view, this result indicates that an additive constant (1 in this case), or offset, representing ***a time domain DC component corresponds to a peak in the frequency domain at a frequency of zero***.

Another important example is the transform of a cosine function ($\cos\omega_0 t$). This function gives us insight into how a basic pure sinusoid transforms into the frequency (Fourier) domain. Intuitively, we would expect such a function to have a singular representation in the frequency domain. We attack this problem by expressing the cosine as the sum of two complex exponentials (using Euler's relation): $\cos \omega_0 t = \frac{1}{2}\left[e^{-j\omega_0 t} + e^{j\omega_0 t}\right]$. The transform of this function becomes:

$$F(j\omega) = \frac{1}{2}\int_{-\infty}^{\infty}\left[e^{-j\omega_0 t} + e^{j\omega_0 t}\right]e^{-j\omega t}dt$$
$$= \frac{1}{2}\left[\int_{-\infty}^{\infty} e^{-j(\omega+\omega_0)t}dt + \int_{-\infty}^{\infty} e^{-j(\omega-\omega_0)t}dt\right] \tag{6.12}$$

Both integrals in this expression can be evaluated easily using the result for the exponential equation that we obtained above $\int_{-\infty}^{\infty} e^{-j\omega t}dt = 2\pi\delta(\omega)$, replacing ω with $\omega+\omega_0$ and $\omega-\omega_0$, respectively. Thus the Fourier transform of a cosine function (a real and even symmetric function) results in:

$$\cos(\omega_0 t) \Leftrightarrow \pi[\delta(\omega+\omega_0) + \delta(\omega-\omega_0)] \tag{6.13}$$

this is also real and even; that is, a delta function at an angular frequency of ω_0 and another at $-\omega_0$. While the impulse in the positive frequency domain is straightforward, the concept of negative frequency, originating from the complex series representation in Fourier analysis (Chapter 5, Section 5.3), defies common sense interpretations. Following a similar logic as that applied above it can be shown that a real and odd symmetric function results in an imaginary odd function:

$$\sin(\omega_0 t) \Leftrightarrow j\pi[\delta(\omega+\omega_0) - \delta(\omega-\omega_0)] \tag{6.14}$$

More commonly, functions that are not purely odd or even have both real and imaginary parts for each frequency component ω_n, i.e.,:

$$F(j\omega_n) = c_n = a_n + jb_n \tag{6.15}$$

That is, for each frequency component in the time domain, we obtain a complex number in the frequency domain. This number is proportional to the cosine and sine amplitudes.

In Chapter 7 we will show how to combine the real and imaginary parts into a metric representing the power for each frequency component in a signal.

6.3 DISCRETE FOURIER TRANSFORM AND THE FAST FOURIER TRANSFORM ALGORITHM

6.3.1 Relationship Between Continuous and Discrete Fourier Transform

The continuous and discrete Fourier transforms and their inverses are related but not identical. For the discrete pair we use a discrete time scale and a discrete frequency scale (Fig. 6.3). Because we want to apply the discrete transform to sampled real-world signals, both the time and frequency scales must also necessarily be finite. Furthermore, we can establish that both scales must be related. For example, in a signal that is observed over a 10-s interval T and sampled at an interval $\Delta t = 1$ ms (0.001 s), these parameters determine the range and precision of the discrete Fourier transform of that signal. It is intuitively clear that in a 10-s interval we cannot reliably distinguish frequencies below a precision of $\Delta f = 1/T = 1/10$ Hz and that the maximum frequency that fits within the sample interval is $1/\Delta t = 1/0.001 = 1000$ Hz. In angular frequency terms,

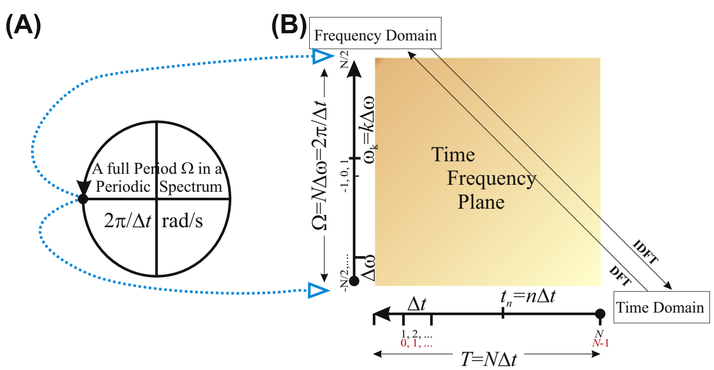

FIGURE 6.3 The time domain and frequency domain scales in the discrete Fourier transform. The Fourier spectrum is periodic, represented by a circular scale in (A). This circular frequency domain scale is mapped onto a line represented by the ordinate in (B). The abscissa in (B) is the time domain scale; note that on the frequency (vertical) axis, the point $-N/2$ is included and $N/2$ is not. Each sample in the time domain represents the preceding sample interval. Depending on which convention is used, the first sample in the time domain is either counted as the zeroth sample (indicated in red) or the first one (indicated in black).

the precision and maximum frequency translate into a step size of $\Delta\omega = 2\pi \times 1/10$ rad/s and a range of $\Omega = 2\pi \times 1000$ rad/s (Fig. 6.3).

> *Note:* From earlier discussions about the Nyquist frequency (Chapter 2) we know that the highest frequency we can observe is actually $2\pi \times 500$ rad/s, ***half*** of Ω. This point will resurface in Chapter 7 when the actual spectra are introduced and we find that these spectra contain two symmetric halves.

The discrete approximation $F_a(j\omega)$ of the CFT $F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt$ sampled over a finite interval including N samples is:

$$F_a(j\omega_k) = \sum_{n=1}^{N} f(t_n)\exp(-j\omega_k t_n)\Delta t \tag{6.16}$$

Discrete time signals are usually created by sampling a continuous time process; each sample thereby represents the signal immediately preceding analog-to-digital conversion. Using this approach, we have a sampled series representing N intervals of length Δt each (Fig. 6.3). For several reasons, it is common practice to use a range for n from 0 to $N-1$ instead of 1 to N. Furthermore, in Eq. (6.16) the time (t) and angular frequency (ω) are represented by discrete variables as indicated in Fig. 6.3. With $t_n = n\,\Delta t$ and $\omega_k = k\,\Delta\omega = k\,2\,\pi/N\Delta t$ (Fig. 6.3), we can write $F_a(j\omega)$ as:

$$F_a(j\omega_k) = \Delta t \sum_{n=0}^{N-1} f(t_n)e^{-j\frac{2\pi}{N}kn} = \Delta t \sum_{n=0}^{N-1} f(t_n)W_N^{kn} \tag{6.17}$$

where W_N^{kn} is a notational simplification of the exponential term. ***Smuggling Δt out of the above expression***, changing $f(t_n)$ to $x(n)$, and $F_a(j\omega_k)$ to $X(k)$ yields the standard definition for the DFT:

$$\boxed{X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn}} \tag{6.18}$$

Similarly, the inverse continuous Fourier transform (ICFT) can be approximated by:

$$f_a(t_n) = \frac{1}{2\pi} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} F_a(j\omega_k)e^{j\omega_k t_n}\Delta\omega \tag{6.19}$$

Note that the upper summation limit does not include $N/2$; due to the circular scale of ω, $-N/2$ and $N/2$ are the same (Fig. 6.3). Changing the range of the summation from $-N/2 \rightarrow (N/2)-1$ into $0 \rightarrow N-1$ and $\Delta\omega = 2\pi/N\Delta t$ (see the axis in Fig. 6.3), Eq. (6.19) yields:

$$f_a(t_n) = \frac{1}{2\pi}\sum_{k=0}^{N-1} F_a(j\omega_k)e^{j\omega_k t_n}\frac{2\pi}{N\Delta t} = \frac{1}{N\Delta t}\sum_{k=0}^{N-1} F_a(j\omega_k)e^{j\omega_k t_n} \tag{6.20}$$

We now use $t_n = n\Delta t$ and $\omega_k = \frac{k2\pi}{N\Delta t}$ (see Fig. 6.3), ***smuggle Δt back***, change $f_a(t_n)$ to $x(n)$, and $F_a(j\omega_k)$ to $X(k)$, thereby obtaining the expression for the discrete inverse Fourier transform:

$$\boxed{x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi}{N}kn} = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-kn}} \tag{6.21}$$

To keep the transform ⇔ inverse transform pair consistent, the division by Δt is corrected in the inverse discrete Fourier transform (IDFT), where the expression is multiplied by Δt.

6.3.2 The Twiddle Factor

The weighting factor introduced as W_N in the above formulae plays an important role in the practical development of DFT algorithms including the optimized one known as the FFT. The efficiency of this algorithm relies crucially on the fact that this factor, also known as the "twiddle" ***factor,*** is periodic.

6.3.3 Discrete Fourier Transform Versus a Base-2 Fast Fourier Transform

The basic idea used to optimize the DFT algorithm involves using the periodicity in the twiddle factor to combine terms and therefore reduce the number of computationally demanding multiplication steps required for a given number of samples (Cooley and Tukey, 1965). Specifically, the standard formulation of the DFT of a time series with N values requires N^2 multiplications for a time series with N points; whereas, the FFT requires only $N\log_2(N)$ multiplications.

For example, consider a four-point time series: $x(0)$, $x(1)$, $x(2)$, $x(3)$ and its DFT $X(0)$, $X(1)$, $X(2)$, $X(3)$. For $N = 4$ each of the X values is calculated with:

$$X(k) = \sum_{n=0}^{3} x(n)W_4^{kn} \tag{6.22}$$

If we were to perform the DFT directly from Eq. (6.22) we would have four multiplications for each $X(k)$; since we have $X(0)$–$X(3)$ this leads to a total of $4 \times 4 = 16$ multiplications. However, since the expression in Eq. (6.22) is a summation, we can split the problem into odd and even components:

$$X(k) = \sum_{r=0}^{1} x(2r)W_4^{2rk} + \sum_{r=0}^{1} x(2r+1)W_4^{(2r+1)k} \tag{6.23}$$

The second twiddle factor can be separated into two factors:

$$W_4^{(2r+1)k} = W_4^{2rk}W_4^k \tag{6.24}$$

Expanding the summations for $X(0)-X(3)$ and combining terms we get:

$$\begin{aligned}
X(0) &= x(0)W_4^0 + x(2)W_4^0 + x(1)W_4^0W_4^0 + x(3)W_4^0W_4^0 \\
&= \left[x(0) + x(2)W_4^0\right] + W_4^0\left[x(1) + x(3)W_4^0\right] \\
X(1) &= x(0)W_4^0 + x(2)W_4^2 + x(1)W_4^1W_4^0 + x(3)W_4^1W_4^2 \\
&= \left[x(0) + x(2)W_4^2\right] + W_4^1\left[x(1) + x(3)W_4^2\right] \\
X(2) &= x(0)W_4^0 + x(2)W_4^4 + x(1)W_4^2W_4^0 + x(3)W_4^2W_4^4 \\
&= \left[x(0) + x(2)W_4^4\right] + W_4^2\left[x(1) + x(3)W_4^4\right] \\
X(3) &= x(0)W_4^0 + x(2)W_4^6 + x(1)W_4^3W_4^0 + x(3)W_4^3W_4^6 \\
&= \left[x(0) + x(2)W_4^6\right] + W_4^3\left[x(1) + x(3)W_4^6\right]
\end{aligned} \tag{6.25}$$

Note that with the exception of the first line in Eq. (6.25) we used $W_4^0 = 1$. In the first equation we kept W_4^0 in the expression solely to emphasize the similarity between all the expressions for $X(k)$. Further, we exploit the fact that the twiddle factors (W) represent periodic exponentials (Fig. 6.4B), i.e.,

$$\begin{aligned}
W_4^4 &= \exp\left\{j\left(-\frac{2\pi}{4}\right)4\right\} = 1 = W_4^0 \text{ and} \\
W_4^6 &= \exp\left\{j\left(-\frac{2\pi}{4}\right)6\right\} = -1 = W_4^2
\end{aligned} \tag{6.26}$$

In a MATLAB® script representing Eq. (6.25) may look like:

```
X(0+1)=(x(0+1)+x(2+1)*W0)+W0*(x(1+1)+x(3+1)*W0);
X(1+1)=(x(0+1)+x(2+1)*W2)+W1*(x(1+1)+x(3+1)*W2);
X(2+1)=(x(0+1)+x(2+1)*W0)+W2*(x(1+1)+x(3+1)*W0);
X(3+1)=(x(0+1)+x(2+1)*W2)+W3*(x(1+1)+x(3+1)*W2);
```

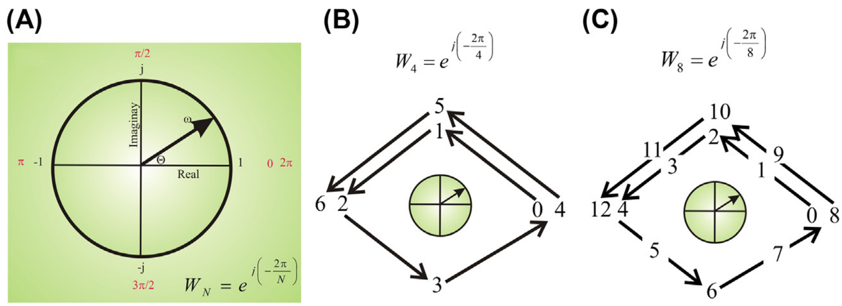

FIGURE 6.4 Periodicity of the twiddle factor W_N. (A) The values of Θ are indicated in red and the real and imaginary components in black. For instance $-1 + j0$ is associated with $\Theta = \pi$, $0 + j$ is associated with $\Theta = \pi/2$, etc. It can be seen that for $\Theta = 0$ or 2π, the values are identical $(1 + j0)$ due to the periodicity of W_N. In (B) and (C) concrete examples are provided for the periodicity of a four-point (W_4) and eight-point (W_8) algorithm. The numbers correspond to the powers of the twiddle factor (e.g., $0 \rightarrow W_4^0$; $1 \rightarrow W_4^1$; $2 \rightarrow W_4^2$, etc.): in case of $N = 4$, a cycle is completed in four steps; whereas for $N = 8$ the cycle is completed in eight steps. In the first case (B): $W_4^0 = W_4^4$, $W_4^1 = W_4^5$, and $W_4^2 = W_4^6$. In the second example (C): $W_8^0 = W_8^8$, $W_8^2 = W_8^{10}$, etc.

In this script, the time series is **x**, its transform is **X**, and the twiddle factors are **W0−W3.** Parenthetically, all indices are augmented with one because MATLAB® does not allow zero indices for vectors.

You may note that there are still 12 multiplications here, an improvement over the $4 \times 4 = 16$ multiplications for the brute force approach, but more than the expected $4\log_2(4) = 8$ multiplications. However, if we take advantage of the repeated multiplications in the above algorithm (i.e., `x(2+1)*W0`, `x(2+1)*W2`, `x(3+1)*W0`, and `x(3+1)*W2`) we end up with $12 - 4 = 8$ multiplications.

It is easier to see the algorithm flow in a diagram where the nodes are variables and the lines represent the operations on those variables (Fig. 6.5). In this example the input variables (left) are added in the output. In two of the cases, the input is multiplied by a twiddle factor (i.e., W_x and W_y in Fig. 6.5).

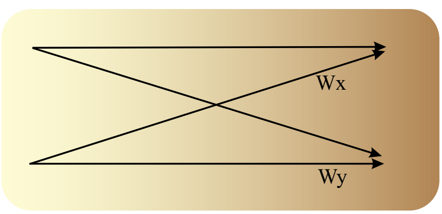

FIGURE 6.5 A flow diagram that forms the basis of the fast Fourier transform (FFT) algorithm. The base diagram is known as the **FFT butterfly.**

6.3.4 Examples

1. A four-point example

   The following listing is an example of a four-point FFT MATLAB® Script (Fig. 6.6). To clearly show the specific features of the FFT algorithm, the program has not been optimized to avoid redundant operations.

```
% pr6_1.m
% A four point FFT
clear

x(0+1)=0;             % input time series x(n)
x(1+1)=1;             % Indices are augmented by 1
x(2+1)=1;             % MATLAB indices start at 1
x(3+1)=0;             % instead of 0

W4=exp(j*2*pi/4);     % the W4 twiddle factor
W0=W4^0;              % and the 0-3rd powers
W1=W4^1;
W2=W4^2;
W3=W4^3;

X(0+1)=(x(0+1)+x(2+1)*W0)+W0*(x(1+1)+x(3+1)*W0);
X(1+1)=(x(0+1)+x(2+1)*W2)+W1*(x(1+1)+x(3+1)*W2);
X(2+1)=(x(0+1)+x(2+1)*W0)+W2*(x(1+1)+x(3+1)*W0);
X(3+1)=(x(0+1)+x(2+1)*W2)+W3*(x(1+1)+x(3+1)*W2);

% Check with MATLAB fft command
fx=fft(x);

figure
hold
plot(X);
plot(fx,'r+');
```

2. An eight-point example

Evaluate the diagram in Fig. 6.7 with the **MATLAB®** ***script pr6_2.m***. While the signal vector indices seem rather arbitrarily ordered, a binary representation can make this indexing more straightforward; Table 6.2

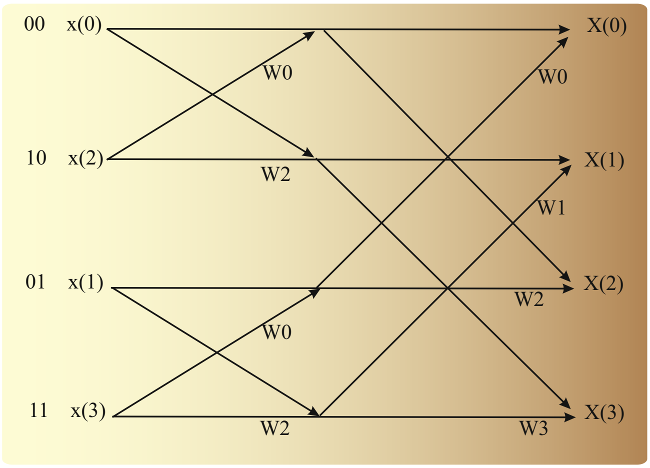

FIGURE 6.6 Diagram for a four-point fast Fourier transform. See the following for the diagram's implementation in MATLAB® script. Here we used $W_4^0 = W_4^4$ and $W_4^2 = W_4^6$ (Fig. 6.4) to optimize the algorithm. Note that we only indicate the superscript value of the twiddle factor in the diagram: $W_4^0 \rightarrow W0$, $W_4^2 \rightarrow W2$, etc.

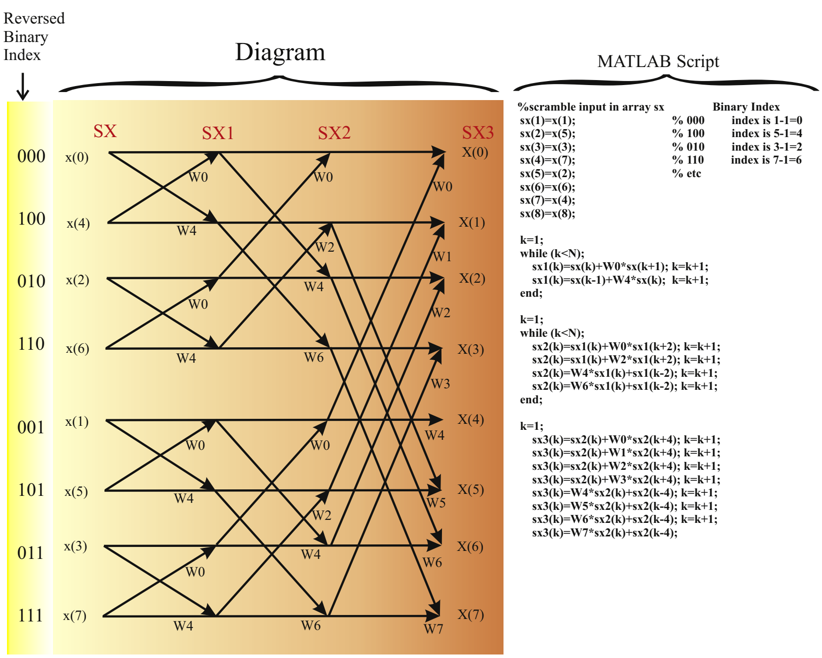

FIGURE 6.7 A diagram of an eight-point fast Fourier transform.

TABLE 6.2 A 3-Bit Binary Set of Numbers to Explain Scrambling in Fast Fourier Transform Input

Binary-Decimal	MATLAB® Index	Inverted Binary-Decimal	MATLAB® Index
000 = 0	1	000 = 0	1
001 = 1	2	100 = 4	5
010 = 2	3	010 = 2	3
011 = 3	4	110 = 6	7
100 = 4	5	001 = 1	2
101 = 5	6	101 = 5	6
110 = 6	7	011 = 3	4
111 = 7	8	111 = 7	8

provides an overview of binary numbers and how they relate to this index scrambling procedure.

In the example in Fig. 6.7, the input time series is $x(0)$, $x(1)$, …, $x(7)$. Note that the input of the algorithm is the lower case vector x, and the output is represented as capital X. First the input is ***scrambled*** to obtain the input to the FFT algorithm: i.e., $SX(0) = x(0)$, $SX(1) = x(4)$, …, $SX(7) = x(7)$. The scrambling process can be presented by reversing the index in binary format. In our time series we have eight data points ($x(0)$–$x(7)$); to represent indices from 0 to 7 we need 3 bits ($2^3 = 8$). We use the reverse binary values to scramble the input, for example, SX with index 1 (in binary 3 bit representation 001) becomes x with index 4 (in binary representation 100). An overview for all indices can be found in Table 6.2. Note that in the MATLAB® script example all indices are increased by one ($SX(0) \rightarrow SX(1)$ etc.) because MATLAB® cannot work with a zero index. After the input is scrambled, the rest of the diagram shows the flow of the calculations. For instance, working backward in the diagram from $X(3)$:

$$X(3) = SX2(3) + W3 \times SX2(7)$$

with:

$$SX2(3) = W6 \times SX1(3) + SX1(1) \ \text{ and } \ SX2(7) = W6 \times SX1(7) + SX1(5)$$

with:

$$SX1(1) = SX(0) + W4 \times SX(1)\ldots \ \text{etc.}$$

As you can see the creation of the algorithm from the diagram is tedious but not difficult. The associated part of the MATLAB® script in Fig. 6.7 reflects the diagram with all indices increased by one. The purpose of this example script is to make the FFT operation transparent; therefore this example script is neither optimized for efficiency nor particularly elegant (from a programmer's perspective).

EXERCISES

6.1. Show that when frequency f is used instead of angular frequency ω ($\omega = 2\pi f$), that $F(f) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi ft}dt$ and $f(t) = \int_{-\infty}^{\infty} F(f)e^{j2\pi ft}df$ are equivalent to Eqs. (6.4) and (6.8).

6.2. Given Eqs. (6.4) and (6.8),
   a. Determine the Fourier transform of $\delta(t)$
   b. Determine the inverse transform of $\delta(\omega)$
   c. Interpret your findings.

6.3. Compute the CFT (show all the steps) of
   (Hint: see Section 6.2.1. Examples of CFT pairs)
   a. $\exp(j\omega_0 t)$
   b. $\cos(\omega_0 t)$
   c. $\sin(\omega_0 t)$
   d. $\sin(\omega_0 t) + \cos(\omega_0 t)$
   e. $\cos(\omega_0 t + \phi)$
   f. Are the functions even, odd, neither, real, imaginary, complex?
   g. Are their CFTs even, odd, neither, real, imaginary, complex?

6.4. We sample an epoch of T seconds at a rate of s samples/second and we use DFT to compute a spectrum with a resolution of Δf Hz and a range of F Hz.
   Complete the "—" in Table E6.4.

TABLE E6.4 DFT Time- and Frequency Domain Parameters

T (s)	s (samples/s)	Δf (Hz)	F (Hz)
10	10,000	—	—
—	—	1	50
—	—	0.5	10
5	—	0.2	—
5	100	—	—
—	1000	—	500
—	10,000	1	—

6.5. Using a sampled time series $x(n)$ with N samples we can determine the DFT $X(k)$ using the equation:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \quad \text{and} \quad W_N^{kn} = \exp\left(-j\left[\frac{2\pi}{N}\right]kn\right)$$

a. Show that $X(0) = N\overline{x}$ with $\overline{x} =$ the mean of the time series
b. Explain why $X(k)$ is a periodic function
c. Determine the value of the period P for which $X(k) = X(k + P)$.

Reference

Cooley, J.W., Tukey, J.W., 1965. An algorithm for machine calculation of complex Fourier series. Math Comput. 19, 297–301.