CHAPTER

10

Fourier Transform Properties

The time and frequency domains are alternative ways of representing signals. The Fourier transform is the mathematical relationship between these two representations. If a signal is modified in one domain, it will also be changed in the other domain, although usually not in the same way. For example, it was shown in the last chapter that *convolving* time domain signals results in their frequency spectra being *multiplied*. Other mathematical operations, such as addition, scaling and shifting, also have a matching operation in the opposite domain. These relationships are called *properties* of the Fourier Transform, how a mathematical change in one domain results in a mathematical change in the other domain.

Linearity of the Fourier Transform

The Fourier Transform is *linear*, that is, it possesses the properties of *homogeneity* and *additivity*. This is true for all four members of the Fourier transform family (Fourier transform, Fourier Series, DFT, and DTFT).

Figure 10-1 provides an example of how homogeneity is a property of the Fourier transform. Figure (a) shows an arbitrary time domain signal, with the corresponding frequency spectrum shown in (b). We will call these two signals: $x[\ ]$ and $X[\ ]$, respectively. *Homogeneity* means that a change in amplitude in one domain produces an identical change in amplitude in the other domain. This should make intuitive sense: when the amplitude of a time domain waveform is changed, the amplitude of the sine and cosine waves making up that waveform must also change by an equal amount.

In mathematical form, if $x[\ ]$ and $X[\ ]$ are a Fourier Transform pair, then $kx[\ ]$ and $kX[\ ]$ are also a Fourier Transform pair, for any constant k. If the frequency domain is represented in *rectangular* notation, $kX[\ ]$ means that both the real part and the imaginary part are multiplied by k. If the frequency domain is represented in *polar* notation, $kX[\ ]$ means that the magnitude is multiplied by k, while the phase remains unchanged.

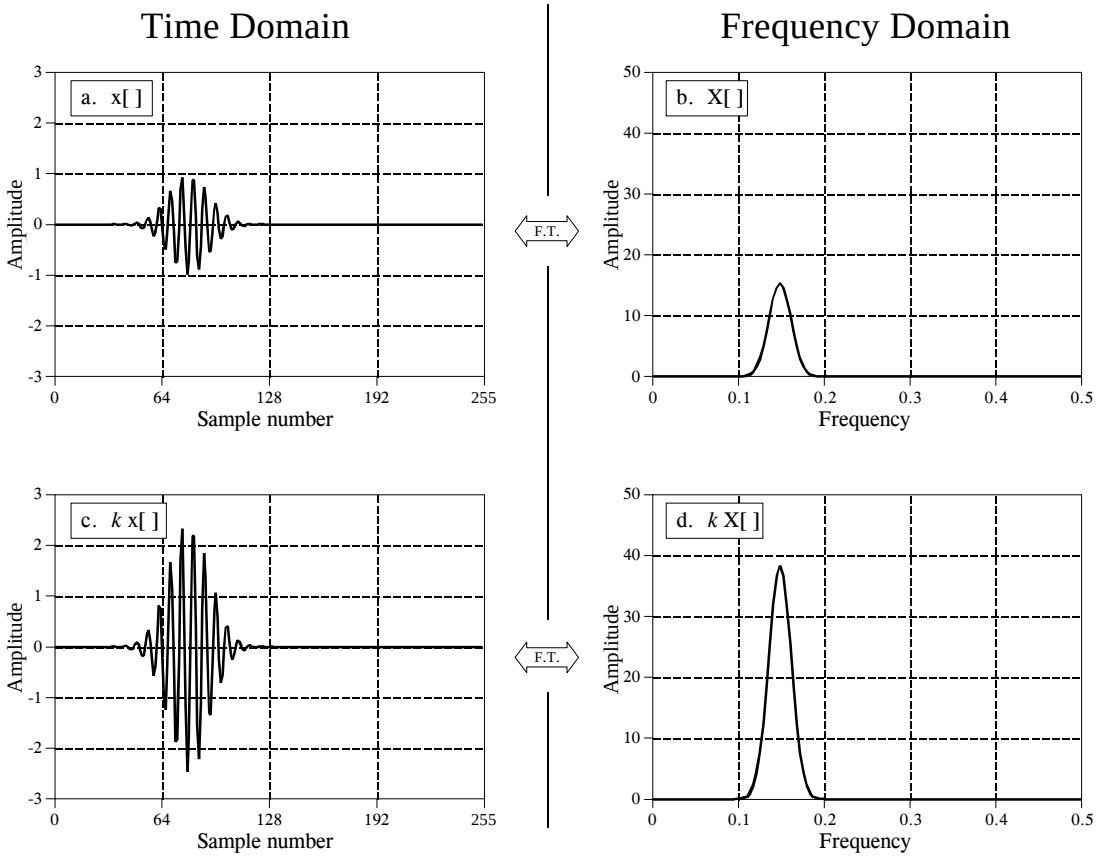

FIGURE 10-1
Homogeneity of the Fourier transform. If the amplitude is changed in one domain, it is changed by the same amount in the other domain. In other words, *scaling* in one domain corresponds to *scaling* in the other domain.

Additivity of the Fourier transform means that *addition* in one domain corresponds to *addition* in the other domain. An example of this is shown in Fig. 10-2. In this illustration, (a) and (b) are signals in the time domain called $x_1[\ ]$ and $x_2[\ ]$, respectively. Adding these signals produces a third time domain signal called $x_3[\ ]$, shown in (c). Each of these three signals has a frequency spectrum consisting of a real and an imaginary part, shown in (d) through (i). Since the two time domain signals *add* to produce the third time domain signal, the two corresponding spectra *add* to produce the third spectrum. Frequency spectra are added in rectangular notation by adding the real parts to the real parts and the imaginary parts to the imaginary parts. If: $x_1[n] + x_2[n] = x_3[n]$, then: $ReX_1[f] + ReX_2[f] = ReX_3[f]$ and $ImX_1[f] + ImX_2[f] = ImX_3[f]$. Think of this in terms of cosine and sine waves. All the cosine waves add (the real parts) and all the sine waves add (the imaginary parts) with no interaction between the two.

Frequency spectra in polar form cannot be directly added; they must be converted into rectangular notation, added, and then reconverted back to

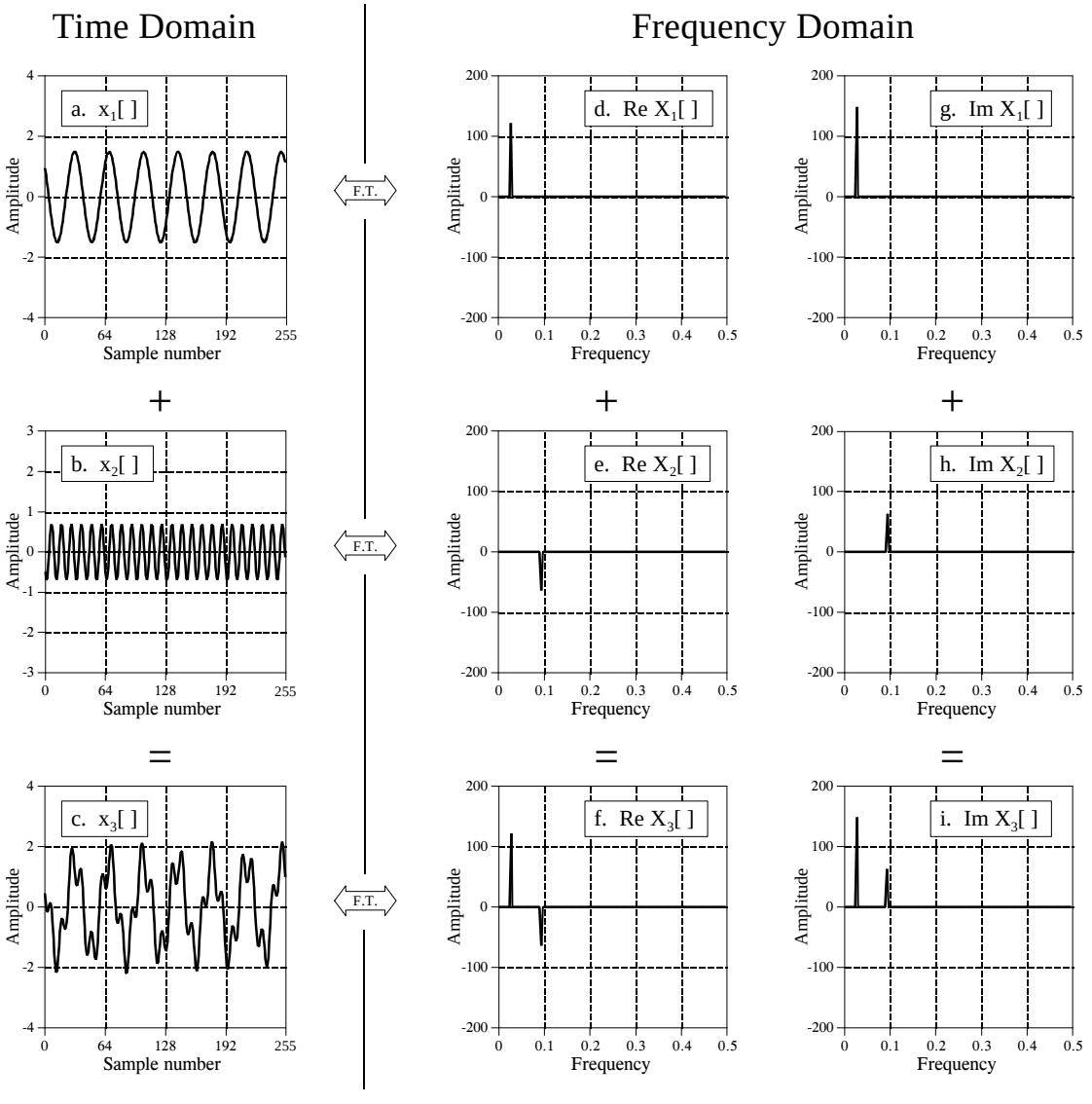

FIGURE 10-2
Additivity of the Fourier transform. Adding two or more signals in one domain results in the corresponding signals being added in the other domain. In this illustration, the time domain signals in (a) and (b) are added to produce the signal in (c). This results in the corresponding real and imaginary parts of the frequency spectra being added.

polar form. This can also be understood in terms of how sinusoids behave. Imagine adding two sinusoids having the same frequency, but with different amplitudes (A_1 and A_2) and phases (ϕ_1 and ϕ_2). If the two phases happen to be same ($\phi_1 = \phi_2$), the amplitudes will add ($A_1 + A_2$) when the sinusoids are added. However, if the two phases happen to be exactly opposite ($\phi_1 = -\phi_2$), the amplitudes will *subtract* ($A_1 - A_2$) when the sinusoids are added. The point is, when sinusoids (or spectra) are in polar form, they *cannot* be added by simply adding the magnitudes and phases.

In spite of being linear, the Fourier transform is *not* shift invariant. In other words, a shift in the time domain *does not* correspond to a shift in the frequency domain. This is the topic of the next section.

Characteristics of the Phase

In mathematical form: if $x[n] \leftrightarrow Mag\,X[f]$ & $Phase\,X[f]$, then a shift in the time domain results in: $x[n+s] \leftrightarrow Mag\,X[f]$ & $Phase\,X[f] + 2\pi sf$, (where f is expressed as a fraction of the sampling rate, running between 0 and 0.5). In words, a shift of s samples in the time domain leaves the magnitude unchanged, but adds a linear term to the phase, $2\pi sf$. Let's look at an example of how this works.

Figure 10-3 shows how the phase is affected when the time domain waveform is shifted to the left or right. The magnitude has not been included in this illustration because it isn't interesting; it is not changed by the time domain shift. In Figs. (a) through (d), the waveform is gradually shifted from having the peak centered on sample 128, to having it centered on sample 0. This sequence of graphs takes into account that the DFT views the time domain as *circular*; when portions of the waveform exit to the right, they reappear on the left.

The time domain waveform in Fig. 10-3 is symmetrical around a vertical axis, that is, the left and right sides are mirror images of each other. As mentioned in Chapter 7, signals with this type of symmetry are called *linear phase*, because the phase of their frequency spectrum is a *straight line*. Likewise, signals that don't have this left-right symmetry are called *nonlinear phase*, and have phases that are something other than a straight line. Figures (e) through (h) show the phase of the signals in (a) through (d). As described in Chapter 7, these phase signals are *unwrapped*, allowing them to appear without the discontinuities associated with keeping the value between π and $-\pi$.

When the time domain waveform is shifted to the right, the phase remains a straight line, but experiences a *decrease* in slope. When the time domain is shifted to the left, there is an *increase* in the slope. This is the main property you need to remember from this section; a shift in the time domain corresponds to changing the slope of the phase.

Figures (b) and (f) display a unique case where the phase is entirely zero. This occurs when the time domain signal is *symmetrical* around sample *zero*. At first glance, this symmetry may not be obvious in (b); it may appear that the signal is symmetrical around sample 256 (i.e., $N/2$) instead. Remember that the DFT views the time domain as circular, with sample zero inherently connected to sample N-1. Any signal that is symmetrical around sample zero will also be symmetrical around sample $N/2$, and vice versa. When using members of the Fourier Transform family that do not view the time domain as periodic (such as the DTFT), the symmetry must be around sample zero to produces a zero phase.

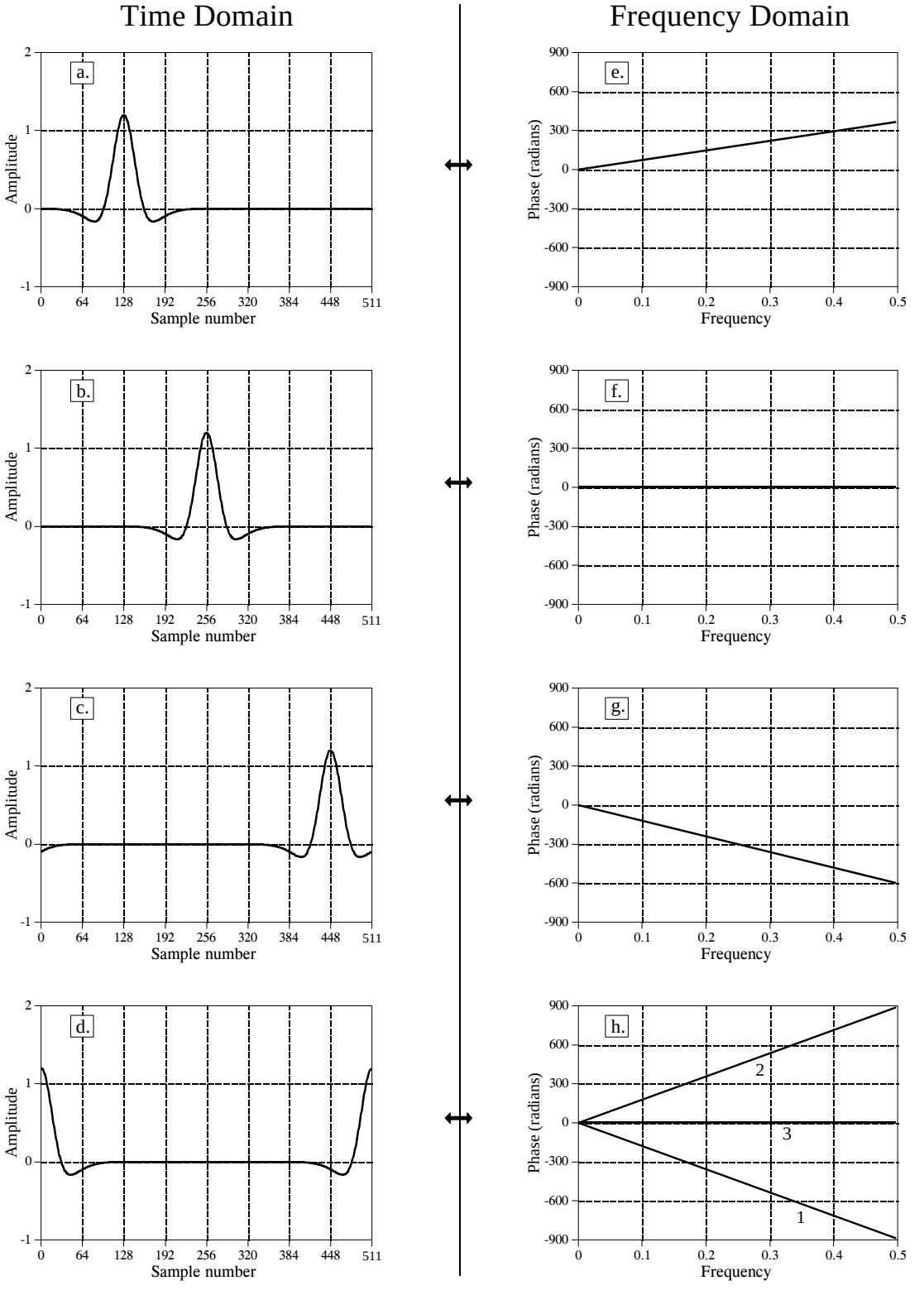

FIGURE 10-3
Phase changes resulting from a time domain shift.

Figures (d) and (h) shows something of a riddle. First imagine that (d) was formed by shifting the waveform in (c) slightly more to the right. This means that the phase in (h) would have a slightly more negative slope than in (g). This phase is shown as line 1. Next, imagine that (d) was formed by starting with (a) and shifting it to the left. In this case, the phase should have a slightly more positive slope than (e), as is illustrated by line 2. Lastly, notice that (d) is symmetrical around sample $N/2$, and should therefore have a zero phase, as illustrated by line 3. Which of these three phases is correct? They all are, depending on how the π and 2π phase ambiguities (discussed in Chapter 8) are arranged. For instance, every sample in line 2 differs from the corresponding sample in line 1 by an integer multiple of 2π, making them equal. To relate line 3 to lines 1 and 2, the π ambiguities must also be taken into account.

To understand why the phase behaves as it does, imagine shifting a waveform by *one* sample to the right. This means that all of the sinusoids that compose the waveform must also be shifted by *one* sample to the right. Figure 10-4 shows two sinusoids that might be a part of the waveform. In (a), the sine wave has a very low frequency, and a one sample shift is only a small fraction of a full cycle. In (b), the sinusoid has a frequency of one-half of the sampling rate, the highest frequency that can exist in sampled data. A one sample shift at this frequency is equal to an entire 1/2 cycle, or π radians. That is, when a shift is expressed in terms of a phase change, it becomes *proportional* to the frequency of the sinusoid being shifted.

For example, consider a waveform that is symmetrical around sample zero, and therefore has a zero phase. Figure 10-5a shows how the phase of this signal changes when it is shifted left or right. At the highest frequency, one-half of the sampling rate, the phase increases by π for each one sample shift to the left, and decreases by π for each one sample shift to the right. At zero frequency there is no phase shift, and all of the frequencies between follow in a straight line.

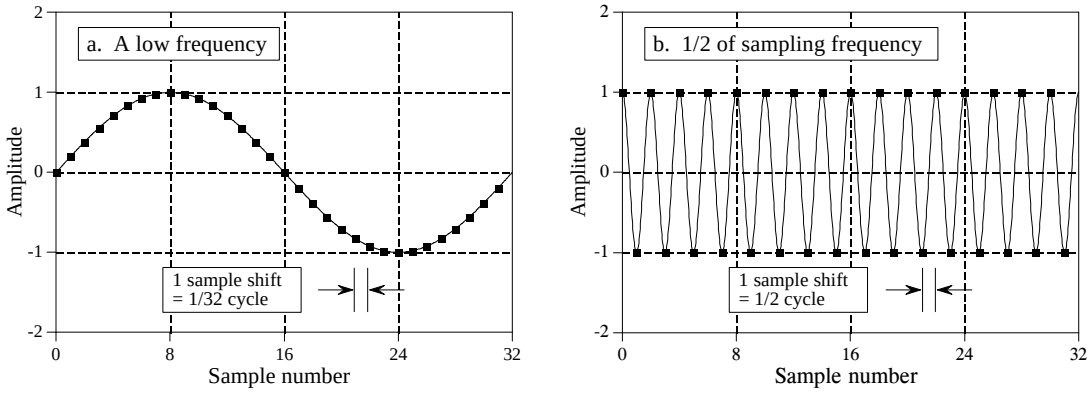

FIGURE 10-4
The relationship between samples and phase. Figures (a) and (b) show low and high frequency sinusoids, respectively. In (a), a one sample shift is equal to 1/32 of a cycle. In (b), a one sample shift is equal to 1/2 of a cycle. This is why a shift in the waveform changes the phase more at high frequencies than at low frequencies.

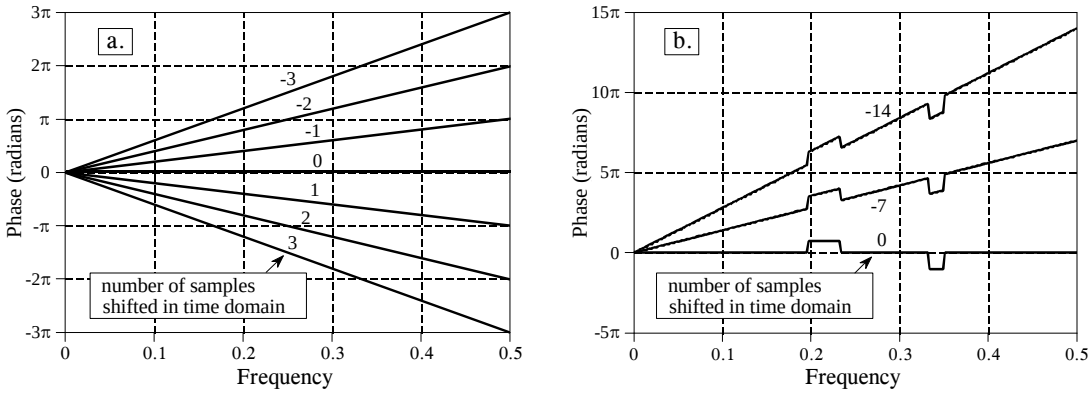

FIGURE 10-5
Phases resulting from time domain shifting. For each sample that a time domain signal is shifted in the positive direction (i.e., to the right), the phase at frequency 0.5 will decrease by π radians. For each sample shifted in the negative direction (i.e., to the left), the phase at frequency 0.5 will increase by π radians. Figure (a) shows this for a linear phase (a straight line), while (b) is an example using a nonlinear phase.

All of the examples we have used so far are *linear* phase. Figure 10-5b shows that *nonlinear* phase signals react to shifting in the same way. In this example the nonlinear phase is a straight line with two rectangular pulses. When the time domain is shifted, these nonlinear features are simply superimposed on the changing slope.

What happens in the *real* and *imaginary parts* when the time domain waveform is shifted? Recall that frequency domain signals in rectangular notation are nearly impossible for humans to understand. The real and imaginary parts typically look like random oscillations with no apparent pattern. When the time domain signal is shifted, the wiggly patterns of the real and imaginary parts become even more oscillatory and difficult to interpret. Don't waste your time trying to understand these signals, or how they are changed by time domain shifting.

Figure 10-6 is an interesting demonstration of what information is contained in the *phase*, and what information is contained in the *magnitude*. The waveform in (a) has two very distinct features: a rising edge at sample number 55, and a falling edge at sample number 110. Edges are very important when information is encoded in the *shape* of a waveform. An edge indicates *when* something happens, dividing whatever is on the left from whatever is on the right. It is time domain encoded information in its purest form. To begin the demonstration, the DFT is taken of the signal in (a), and the frequency spectrum converted into polar notation. To find the signal in (b), the phase is replaced with random numbers between -π and π, and the inverse DFT used to reconstruct the time domain waveform. In other words, (b) is based only on the information contained in the *magnitude*. In a similar manner, (c) is found by replacing the magnitude with small random numbers before using the inverse DFT. This makes the reconstruction of (c) based solely on the information contained in the *phase*.

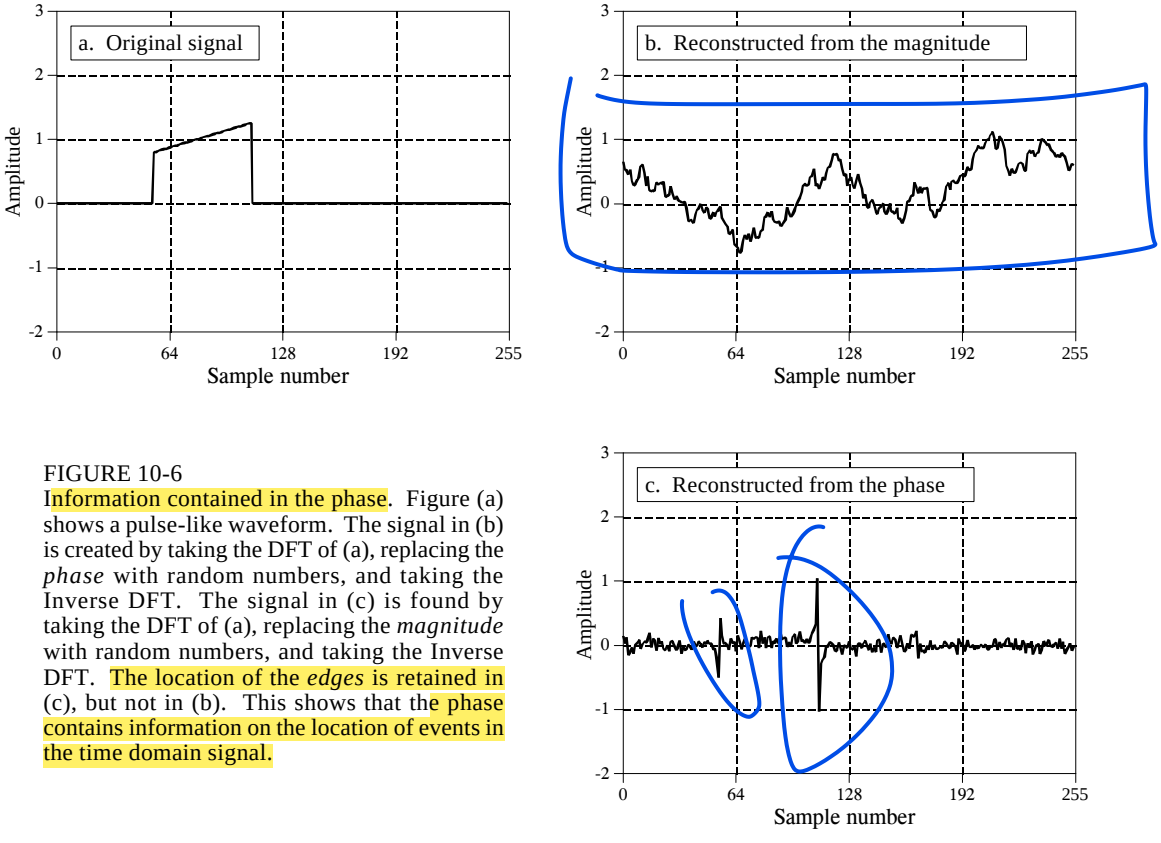

FIGURE 10-6
Information contained in the phase. Figure (a) shows a pulse-like waveform. The signal in (b) is created by taking the DFT of (a), replacing the *phase* with random numbers, and taking the Inverse DFT. The signal in (c) is found by taking the DFT of (a), replacing the *magnitude* with random numbers, and taking the Inverse DFT. The location of the *edges* is retained in (c), but not in (b). This shows that the phase contains information on the location of events in the time domain signal.

The result? The locations of the edges are clearly present in (c), but totally absent in (b). This is because an edge is formed when many sinusoids *rise* at the same location, possible only when their *phases* are coordinated. In short, much of the information about the shape of the time domain waveform is contained in the *phase*, rather than the *magnitude*. This can be contrasted with signals that have their information encoded in the frequency domain, such as audio signals. The magnitude is most important for these signals, with the phase playing only a minor role. In later chapters we will see that this type of understanding provides strategies for designing filters and other methods of processing signals. Understanding how information is represented in signals is always the first step in successful DSP.

Why does left-right symmetry correspond to a zero (or linear) phase? Figure 10-7 provides the answer. Such a signal can be decomposed into a left half and a right half, as shown in (a), (b) and (c). The sample at the center of symmetry (zero in this case) is divided equally between the left and right halves, allowing the two sides to be perfect mirror images of each other. The magnitudes of these two halves will be *identical*, as shown in (e) and (f), while the phases will be opposite in sign, as in (h) and (i). Two important concepts fall out of this. First, every signal that is symmetrical between the left and right will have a linear phase *because* the nonlinear phase of the left half exactly cancels the nonlinear phase of the right half.

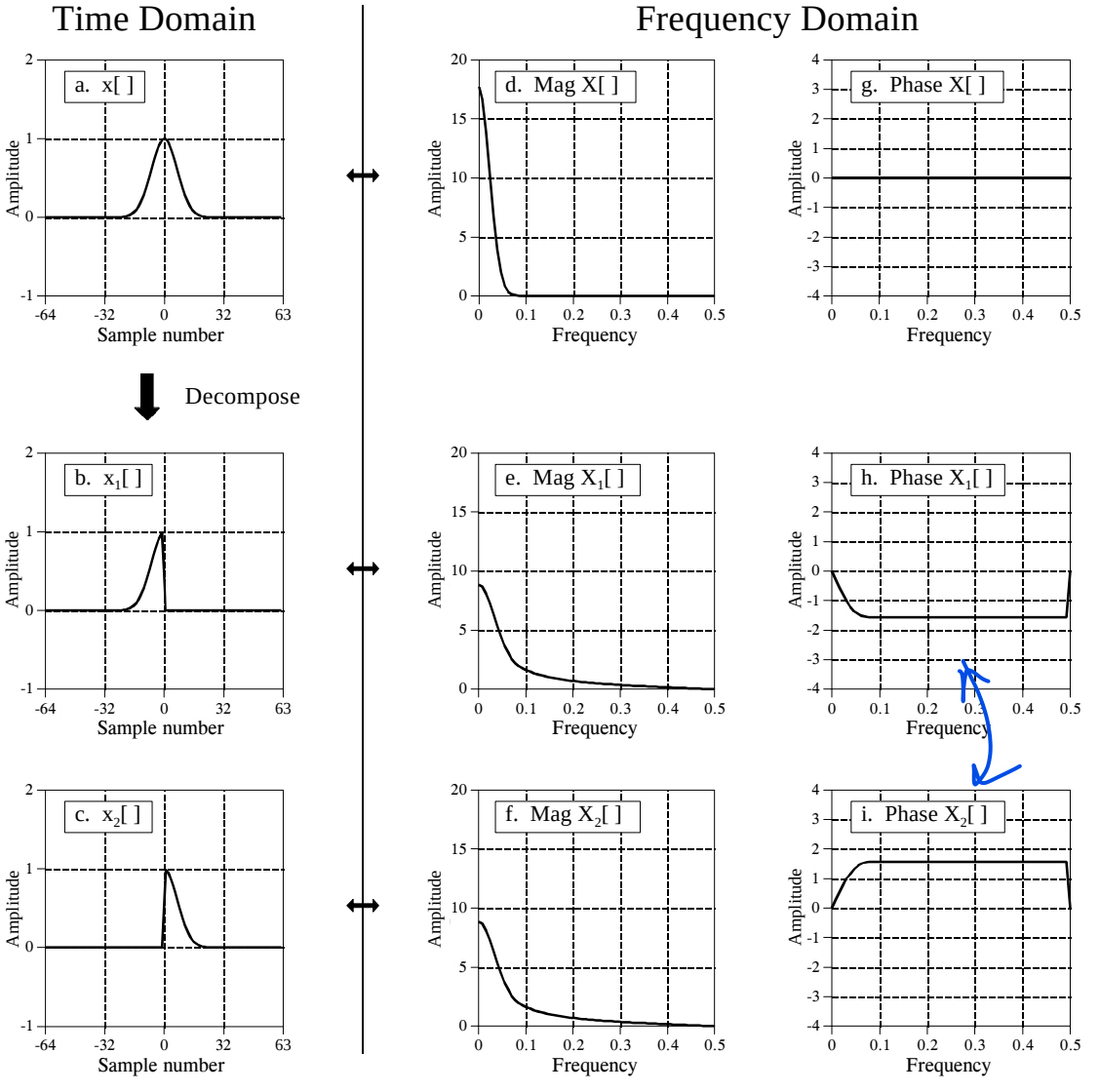

FIGURE 10-7
Phase characteristics of left-right symmetry. A signal with left-right symmetry, shown in (a), can be decomposed into a right half, (b), and a left half, (c). The magnitudes of the two halves are identical, (e) and (f), while the phases are the negative of each other, (h) and (i).

Second, imagine flipping (b) such that it becomes (c). This left-right flip in the time domain does nothing to the magnitude, but changes the sign of every point in the phase. Likewise, changing the sign of the phase flips the time domain signal left-for-right. If the signals are continuous, the flip is around zero. If the signals are discrete, the flip is around sample zero *and* sample $N/2$, simultaneously.

Changing the sign of the phase is a common enough operation that it is given its own name and symbol. The name is **complex conjugation**, and it is

represented by placing a star to the upper-right of the variable. For example, if $X[f]$ consists of $Mag\,X[f]$ and $Phase\,X[f]$, then $X^*[f]$ is called the *complex conjugate* and is composed of $Mag\,X[f]$ and $-Phase\,X[f]$. In rectangular notation, the complex conjugate is found by leaving the real part alone, and changing the sign of the imaginary part. In mathematical terms, if $X[f]$ is composed of $Re\,X[f]$ and $Im\,X[f]$, then $X^*[f]$ is made up of $Re\,X[f]$ and $-Im\,X[f]$.

Here are several examples of how the complex conjugate is used in DSP. If $x[n]$ has a Fourier transform of $X[f]$, then $x[-n]$ has a Fourier transform of $X^*[f]$. In words, flipping the time domain left-for-right corresponds to changing the sign of the phase. As another example, recall from Chapter 7 that correlation can be performed as a convolution. This is done by flipping one of the signals left-for-right. In mathematical form, $a[n] \ast b[n]$ is convolution, while $a[n] \ast b[-n]$ is correlation. In the frequency domain these operations correspond to $A[f] \times B[f]$ and $A[f] \times B^*[f]$, respectively. As the last example, consider an arbitrary signal, $x[n]$, and its frequency spectrum, $X[f]$. The frequency spectrum can be changed to *zero phase* by multiplying it by its complex conjugate, that is, $X[f] \times X^*[f]$. In words, whatever phase $X[f]$ happens to have will be canceled by adding its opposite (remember, when frequency spectra are multiplied, their phases are added). In the time domain, this means that $x[n] \ast x[-n]$ (a signal convolved with a left-right flipped version of itself) will have left-right symmetry around sample zero, regardless of what $x[n]$ is.

To many engineers and mathematicians, this kind of manipulation *is* DSP. If you want to be able to communicate with this group, get used to using their language.

Periodic Nature of the DFT

Unlike the other three Fourier Transforms, the DFT views *both* the time domain and the frequency domain as *periodic*. This can be confusing and inconvenient since most of the signals used in DSP are *not* periodic. Nevertheless, if you want to use the DFT, you must conform with the DFT's view of the world.

Figure 10-8 shows two different interpretations of the time domain signal. First, look at the upper signal, the time domain viewed as N points. This represents how digital signals are typically acquired in scientific experiments and engineering applications. For instance, these 128 samples might have been acquired by sampling some parameter at regular intervals of *time*. Sample 0 is distinct and separate from sample 127 because they were acquired at *different* times. From the way this signal was formed, there is no reason to think that the samples on the left of the signal are even related to the samples on the right.

Unfortunately, the DFT doesn't see things this way. As shown in the lower figure, the DFT views these 128 points to be a single period of an infinitely long periodic signal. This means that the left side of the acquired signal is

connected to the right side of a duplicate signal. Likewise, the right side of the acquired signal is connected to the left side of an identical period. This can also be thought of as the right side of the acquired signal wrapping around and connecting to its left side. In this view, sample 127 occurs next to sample 0, just as sample 43 occurs next to sample 44. This is referred to as being **circular**, and is identical to viewing the signal as being *periodic*.

The most serious consequence of time domain periodicity is **time domain aliasing.** To illustrate this, suppose we take a time domain signal and pass it through the DFT to find its frequency spectrum. We could immediately pass this frequency spectrum through an Inverse DFT to reconstruct the original time domain signal, but the entire procedure wouldn't be very interesting. Instead, we will modify the frequency spectrum in some manner before using the Inverse DFT. For instance, selected frequencies might be deleted, changed in amplitude or phase, shifted around, etc. These are the kinds of things routinely done in DSP. Unfortunately, these changes in the frequency domain can create a time domain signal that is too long to fit into

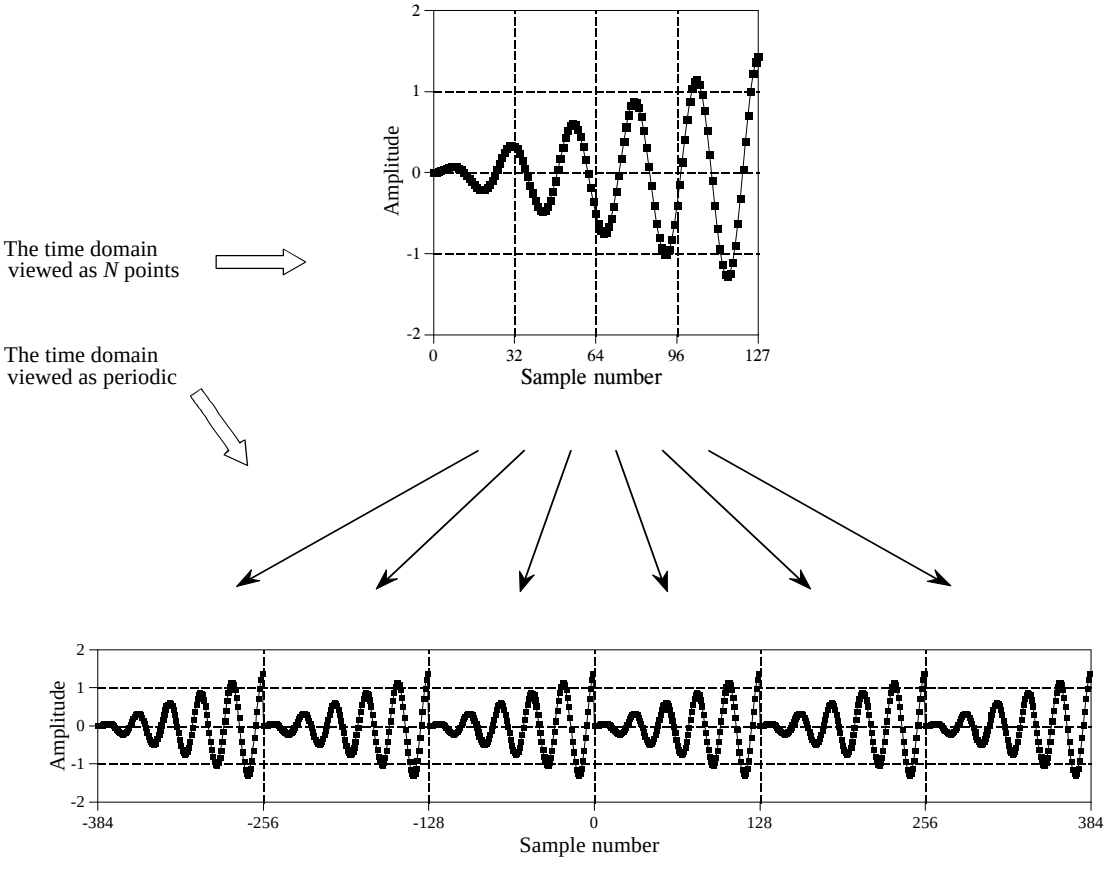

FIGURE 10-8
Periodicity of the DFT's time domain signal. The time domain can be viewed as *N* samples in length, shown in the upper figure, or as an infinitely long periodic signal, shown in the lower figure.

a single period. This forces the signal to spill over from one period into the adjacent periods. When the time domain is viewed as *circular*, portions of the signal that overflow on the right suddenly seem to reappear on the left side of the signal, and vice versa. That is, the overflowing portions of the signal *alias* themselves to a new location in the time domain. If this new location happens to already contain an existing signal, the whole mess adds, resulting in a loss of information. Circular convolution resulting from frequency domain multiplication (discussed in Chapter 9), is an excellent example of this type of aliasing.

Periodicity in the frequency domain behaves in much the same way, but is more complicated. Figure 10-9 shows an example. The upper figures show the magnitude and phase of the frequency spectrum, viewed as being composed of $N/2 + 1$ samples spread between 0 and 0.5 of the sampling rate. This is the simplest way of viewing the frequency spectrum, but it doesn't explain many of the DFT's properties.

The lower two figures show how the DFT views this frequency spectrum as being periodic. The key feature is that the frequency spectrum between 0 and 0.5 appears to have a *mirror image* of frequencies that run between 0 and -0.5. This mirror image of **negative frequencies** is slightly different for the magnitude and the phase signals. In the magnitude, the signal is flipped left-for-right. In the phase, the signal is flipped left-for-right, *and* changed in sign. As you recall, these two types of symmetry are given names: the magnitude is said to be an **even** signal (it has *even* symmetry), while the phase is said to be an **odd** signal (it has *odd* symmetry). If the frequency spectrum is converted into the real and imaginary parts, the *real part* will always be *even*, while the *imaginary part* will always be *odd*.

Taking these negative frequencies into account, the DFT views the frequency domain as periodic, with a period of 1.0 times the sampling rate, such as -0.5 to 0.5, or 0 to 1.0. In terms of sample numbers, this makes the length of the frequency domain period equal to N, the same as in the time domain.

The periodicity of the frequency domain makes it susceptible to **frequency domain aliasing**, completely analogous to the previously described time domain aliasing. Imagine a time domain signal that corresponds to some frequency spectrum. If the time domain signal is modified, it is obvious that the frequency spectrum will also be changed. If the modified frequency spectrum cannot fit in the space provided, it will push into the adjacent periods. Just as before, this aliasing causes two problems: frequencies aren't where they should be, and overlapping frequencies from different periods add, destroying information.

Frequency domain aliasing is more difficult to understand than time domain aliasing, since the periodic pattern is more complicated in the frequency domain. Consider a single frequency that is being forced to move from 0.01 to 0.49 in the frequency domain. The corresponding negative frequency is therefore moving from -0.01 to -0.49. When the positive frequency moves

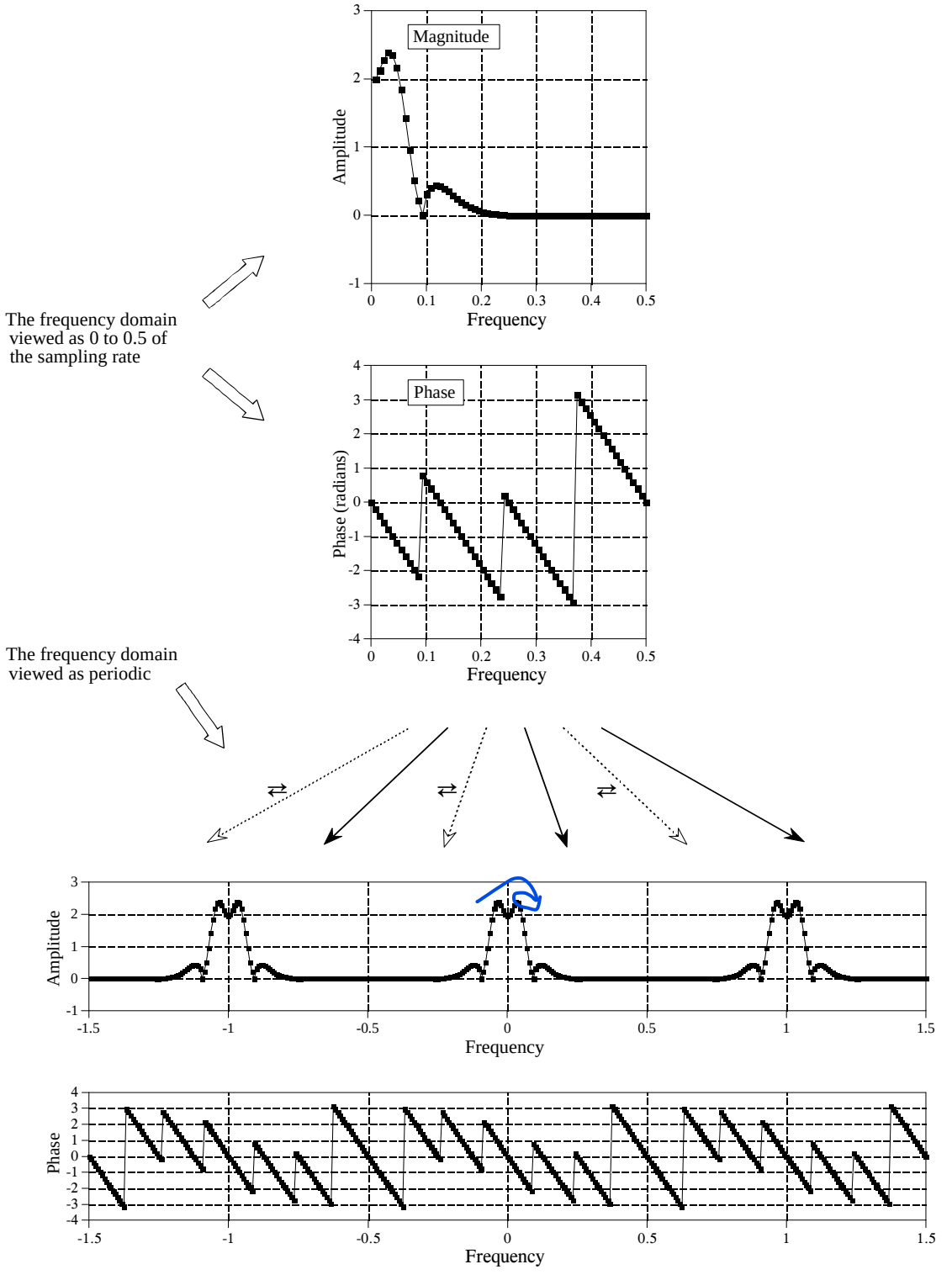

FIGURE 10-9
Periodicity of the DFT's frequency domain. The frequency domain can be viewed as running from 0 to 0.5 of the sampling rate (upper two figures), or an infinity long periodic signal with every other 0 to 0.5 segment flipped left-for-right (lower two figures).

across the 0.5 barrier, the negative frequency is pushed across the -0.5 barrier. Since the frequency domain is periodic, these same events are occurring in the other periods, such as between 0.5 and 1.5. A clone of the positive frequency is crossing frequency 1.5 from left to right, while a clone of the negative frequency is crossing 0.5 from right to left. Now imagine what this looks like if you can only see the frequency band of 0 to 0.5. It appears that a frequency leaving to the *right*, reappears on the *right*, but moving in the opposite direction.

Figure 10-10 illustrates how aliasing appears in the time and frequency domains when only a single period is viewed. As shown in (a), if one end of a time domain signal is too long to fit inside a single period, the protruding end will be *cut off* and *pasted* onto the other side. In comparison, (b) shows that when a frequency domain signal overflows the period, the protruding end is *folded over*. Regardless of where the aliased segment ends up, it adds to whatever signal is already there, destroying information.

Let's take a closer look at these strange things called *negative frequencies*. Are they just some bizarre artifact of the mathematics, or do they have a real world meaning? Figure 10-11 shows what they are about. Figure (a) is a discrete signal composed of 32 samples. Imagine that you are given the task of finding the frequency spectrum that corresponds to these 32 points. To make your job easier, you are told that these points represent a discrete cosine wave. In other words, you must find the frequency and phase shift (f and θ) such that $x[n] = \cos(2\pi nf/N + \theta)$ matches the given samples. It isn't long before you come up with the solution shown in (b), that is, $f = 3$ and $\theta = -\pi/4$.

If you stopped your analysis at this point, you only get 1/3 credit for the problem. This is because there are two other solutions that you have missed. As shown in (c), the second solution is $f = -3$ and $\theta = \pi/4$. Even if the idea of a *negative frequency* offends your sensibilities, it doesn't

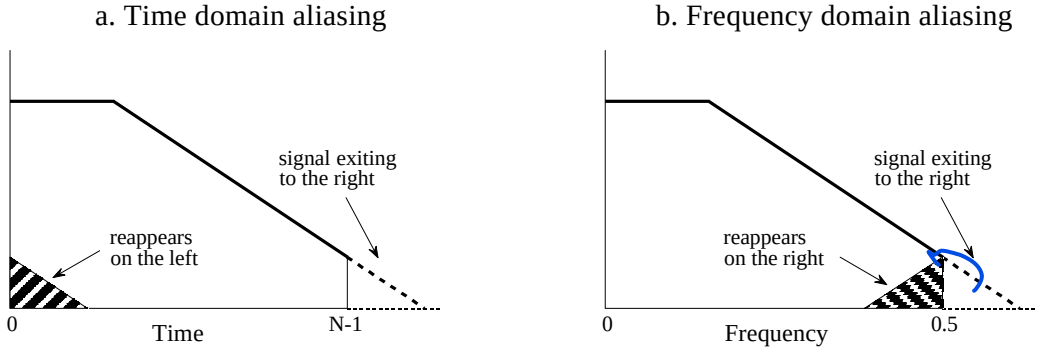

FIGURE 10-10
Examples of aliasing in the time and frequency domains, when only a single period is considered. In the time domain, shown in (a), portions of the signal that exits to the right, reappear on the left. In the frequency domain, (b), portions of the signal that exit to the right, reappear on the right as if they had been folded over.

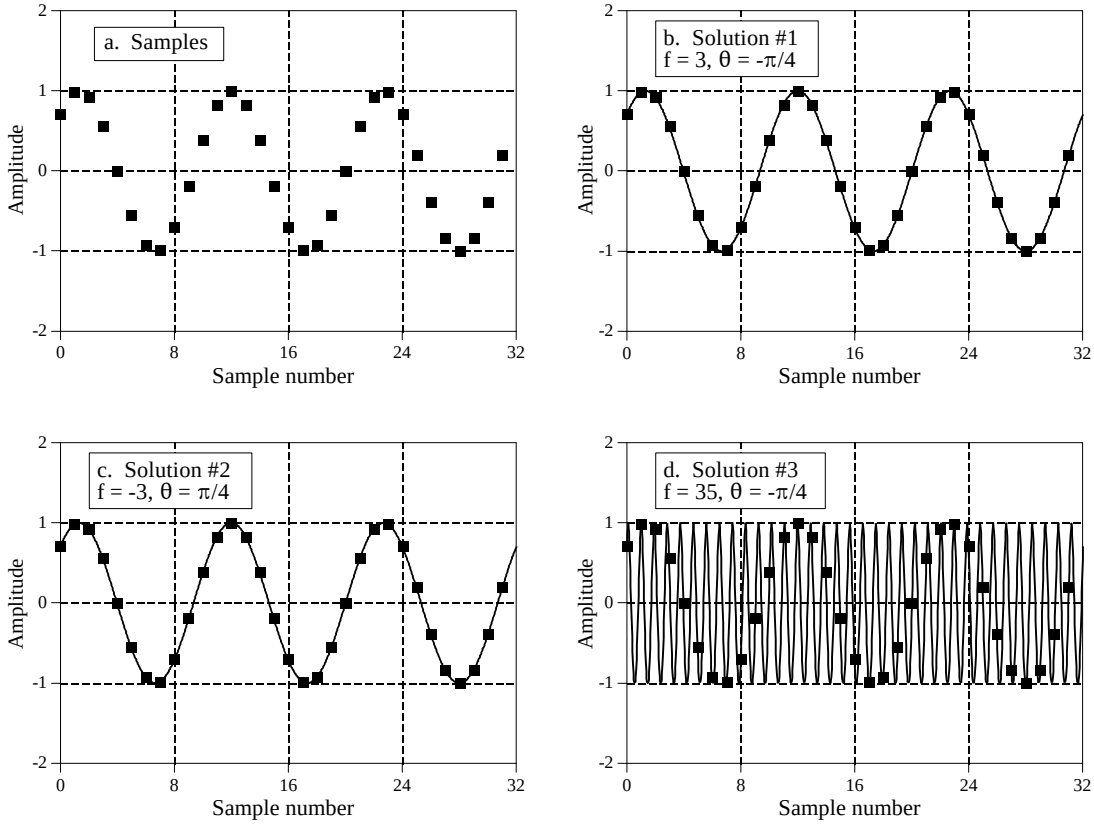

FIGURE 10-11
The meaning of negative frequencies. The problem is to find the frequency spectrum of the discrete signal shown in (a). That is, we want to find the frequency and phase of the sinusoid that passed through all of the samples. Figure (b) is a solution using a *positive* frequency, while (c) is a solution using a *negative* frequency. Figure (d) represents a family of solutions to the problem.

change the fact that it is a mathematically valid solution to the defined problem. Every *positive* frequency sinusoid can alternately be expressed as a *negative* frequency sinusoid. This applies to continuous as well as discrete signals

The third solution is not a single answer, but an infinite family of solutions. As shown in (d), the sinusoid with $f = 35$ and $\theta = -\pi/4$ passes through all of the discrete points, and is therefore a correct solution. The fact that it shows oscillation between the samples may be confusing, but it doesn't disqualify it from being an authentic answer. Likewise, $f = \pm 29$, $f = \pm 35$, $f = \pm 61$, and $f = \pm 67$ are all solutions with multiple oscillations between the points.

Each of these three solutions corresponds to a different section of the frequency spectrum. For discrete signals, the first solution corresponds to frequencies between 0 and 0.5 of the sampling rate. The second solution

results in frequencies between 0 and -0.5. Lastly, the third solution makes up the infinite number of duplicated frequencies below -0.5 and above 0.5. If the signal we are analyzing is continuous, the first solution results in frequencies from zero to positive infinity, while the second solution results in frequencies from zero to negative infinity. The third group of solutions does not exist for continuous signals.

Many DSP techniques do not require the use of negative frequencies, or an understanding of the DFT's periodicity. For example, two common ones were described in the last chapter, *spectral analysis*, and the *frequency response* of systems. For these applications, it is completely sufficient to view the time domain as extending from sample 0 to N-1, and the frequency domain from zero to one-half of the sampling frequency. These techniques can use a simpler view of the world because they never result in portions of one period moving into another period. In these cases, looking at a single period is just as good as looking at the entire periodic signal.

However, certain procedures can *only* be analyzed by considering how signals overflow between periods. Two examples of this have already been presented, *circular convolution* and *analog-to-digital conversion*. In circular convolution, multiplication of the frequency spectra results in the time domain signals being convolved. If the resulting time domain signal is too long to fit inside a single period, it overflows into the adjacent periods, resulting in *time domain aliasing*. In contrast, analog-to-digital conversion is an example of *frequency domain aliasing*. A nonlinear action is taken in the time domain, that is, changing a continuous signal into a discrete signal by sampling. The problem is, the spectrum of the original analog signal may be too long to fit inside the discrete signal's spectrum. When we force the situation, the ends of the spectrum protrude into adjacent periods. Let's look at two more examples where the periodic nature of the DFT is important, *compression & expansion* of signals, and *amplitude modulation*.

Compression and Expansion, Multirate methods

As shown in Fig. 10-12, a *compression* of the signal in one domain results in an *expansion* in the other, and vice versa. For continuous signals, if $X(f)$ is the Fourier Transform of $x(t)$, then $1/k \times X(f/k)$ is the Fourier Transform of $x(kt)$, where k is the parameter controlling the expansion or contraction. If an event happens *faster* (it is compressed in time), it must be composed of *higher* frequencies. If an event happens *slower* (it is expanded in time), it must be composed of *lower* frequencies. This pattern holds if taken to either of the two extremes. That is, if the time domain signal is compressed so far that it becomes an *impulse*, the corresponding frequency spectrum is expanded so far that it becomes a *constant value*. Likewise, if the time domain is expanded until it becomes a constant value, the frequency domain becomes an impulse.

Discrete signals behave in a similar fashion, but there are a few more details. The first issue with discrete signals is *aliasing*. Imagine that the

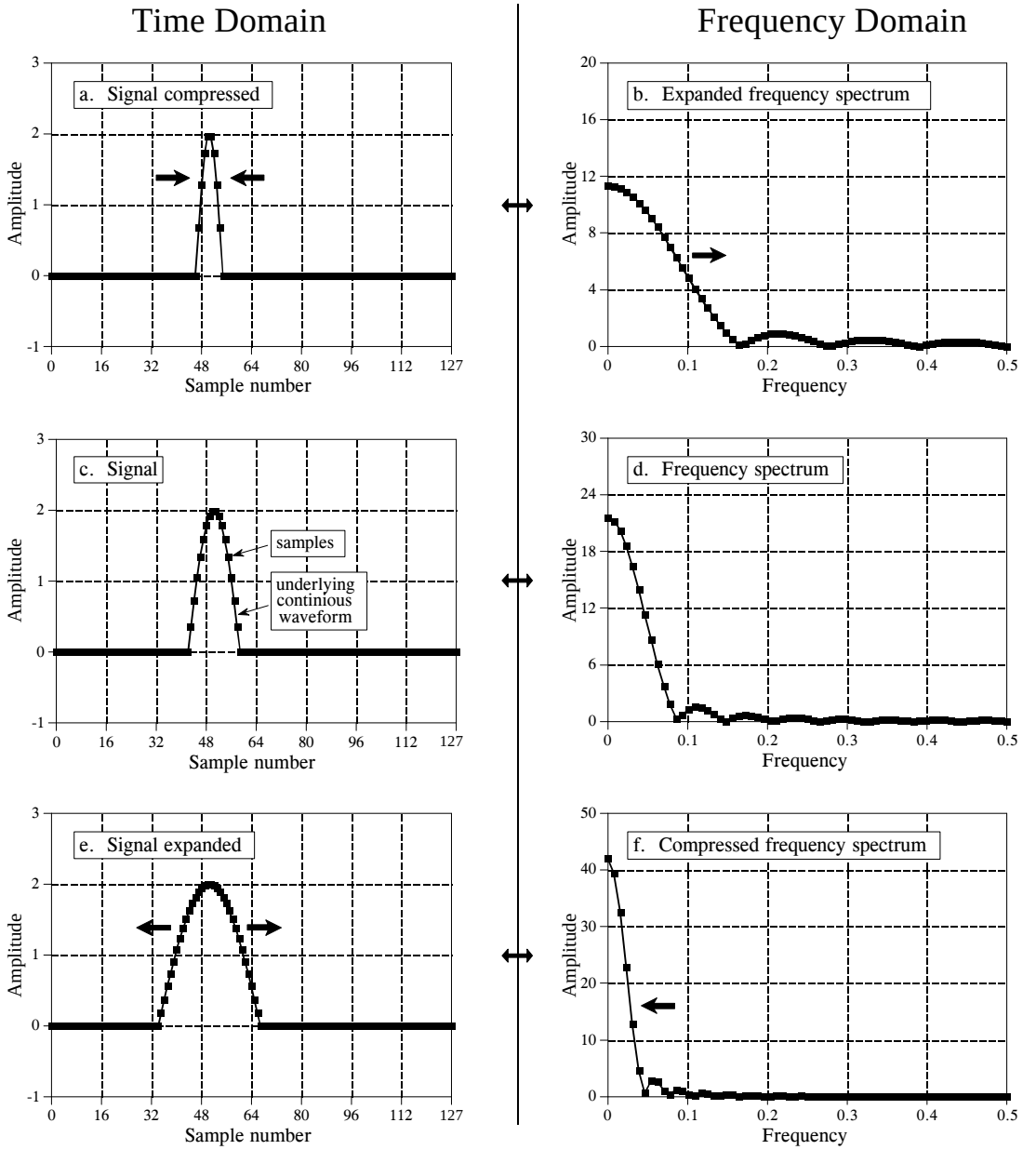

FIGURE 10-12
Compression and expansion. Compressing a signal in one domain results in the signal being expanded in the other domain, and vice versa. Figures (c) and (d) show a discrete signal and its spectrum, respectively. In (a) and (b), the time domain signal has been compressed, resulting in the frequency spectrum being expanded. Figures (e) and (f) show the opposite process. As shown in these figures, discrete signals are expanded or contracted by expanding or contracting the underlying continuous waveform. This underlying waveform is then resampled to find the new discrete signal.

pulse in (a) is compressed several times more than is shown. The frequency spectrum is expanded by an equal factor, and several of the humps in (b) are pushed to frequencies beyond 0.5. The resulting aliasing breaks the simple expansion/contraction relationship. This type of aliasing can also happen in the

time domain. Imagine that the frequency spectrum in (f) is compressed much harder, resulting in the time domain signal in (e) expanding into neighboring periods.

A second issue is to define exactly what it means to compress or expand a discrete signal. As shown in Fig. 10-12a, a discrete signal is compressed by compressing the underlying *continuous* curve that the samples lie on, and then resampling the new continuous curve to find the new discrete signal. Likewise, this same process for the expansion of discrete signals is shown in (e). When a discrete signal is compressed, events in the signal (such as the width of the pulse) happen over a *fewer* number of samples. Likewise, events in an expanded signal happen over a *greater* number of samples.

An equivalent way of looking at this procedure is to keep the underlying continuous waveform the same, but resample it at a different sampling rate. For instance, look at Fig. 10-13a, a discrete Gaussian waveform composed of 50 samples. In (b), the same underlying curve is represented by 400 samples. The change between (a) and (b) can be viewed in two ways: (1) the sampling rate has been kept constant, but the underlying waveform has been expanded to be eight times wider, or (2) the underlying waveform has been kept constant, but the sampling rate has increased by a factor of eight. Methods for changing the sampling rate in this way are called **multirate** techniques. If more samples are added, it is called **interpolation**. If fewer samples are used to represent the signal, it is called **decimation**. Chapter 3 describes how multirate techniques are used in ADC and DAC.

Here is the problem: if we are given an arbitrary discrete signal, how do we know what the underlying continuous curve is? It depends on if the signal's information is encoded in the *time domain* or in the *frequency domain*. For time domain encoded signals, we want the underlying continuous waveform to be a smooth curve that passes through all the samples. In the simplest case, we might draw straight lines between the points and then round the rough corners. The next level of sophistication is to use a curve fitting algorithm, such as a spline function or polynomial fit. There is not a single "correct" answer to this problem. This approach is based on minimizing irregularities in the *time domain* waveform, and completely ignores the frequency domain.

When a signal has information encoded in the frequency domain, we ignore the time domain waveform and concentrate on the frequency spectrum. As discussed in the last chapter, a finer sampling of a frequency spectrum (more samples between frequency 0 and 0.5) can be obtained by padding the time domain signal with zeros before taking the DFT. Duality allows this to work in the opposite direction. If we want a finer sampling in the time domain (interpolation), pad the frequency spectrum with zeros before taking the Inverse DFT. Say we want to interpolate a 50 sample signal into a 400 sample signal. It's done like this: (1) Take the 50 samples and add zeros to make the signal 64 samples long. (2) Use a 64 point DFT to find the frequency spectrum, which will consist of a 33 point real part and a 33 point imaginary part. (3) Pad the right side of the frequency spectrum

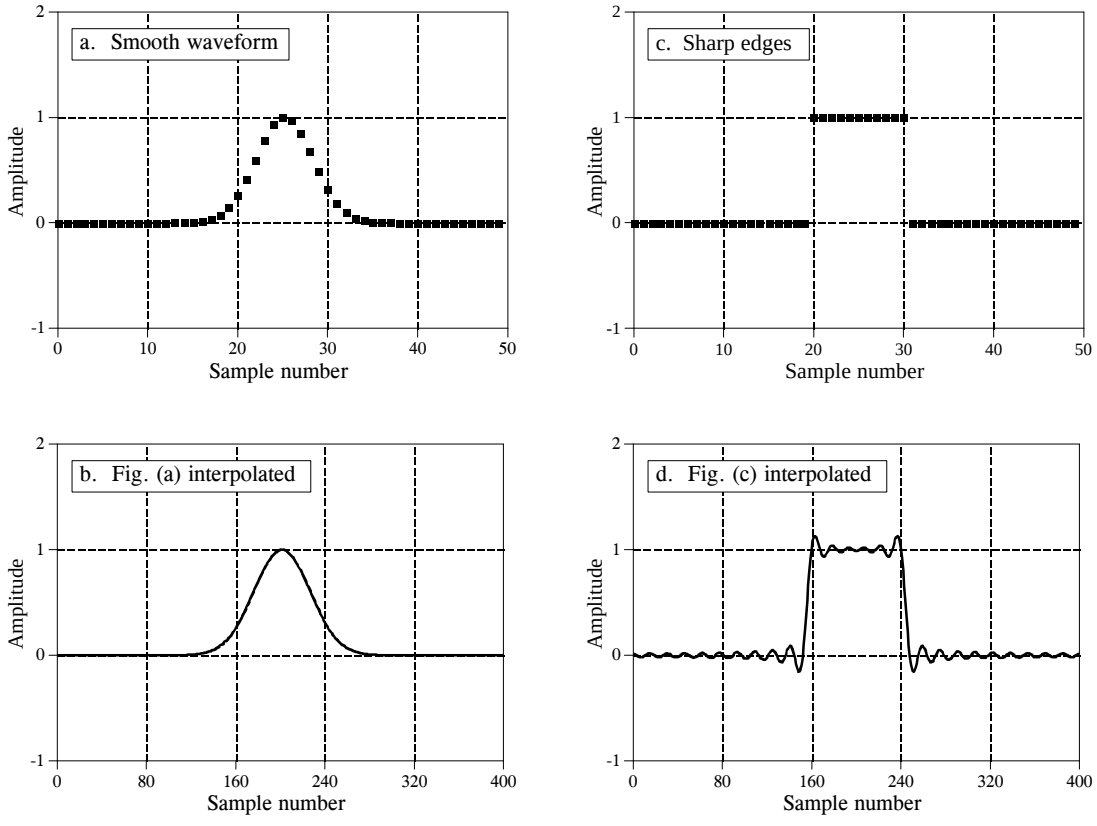

FIGURE 10-13
Interpolation by padding the frequency domain. Figures (a) and (c) each consist of 50 samples. These are interpolated to 400 samples by padding the frequency domain with zeros, resulting in (b) and (d), respectively. (Figures (b) and (d) are discrete signals, but are drawn as continuous lines because of the large number of samples).

(both the real and imaginary parts) with 224 zeros to make the frequency spectrum 257 points long. (4) Use a 512 point Inverse DFT to transform the data back into the time domain. This will result in a 512 sample signal that is a high resolution version of the 64 sample signal. The first 400 samples of this signal are an interpolated version of the original 50 samples.

The key feature of this technique is that the interpolated signal is composed of *exactly* the same frequencies as the original signal. This may or may not provide a well-behaved fit in the time domain. For example, Figs. 10-13 (a) and (b) show a 50 sample signal being interpolated into a 400 sample signal by this method. The interpolation is a smooth fit between the original points, much as if a curve fitting routine had been used. In comparison, (c) and (d) show another example where the time domain is a mess! The oscillatory behavior shown in (d) arises at edges or other discontinuities in the signal. This also includes any discontinuity between sample zero and N-1, since the time domain is viewed as being circular. This overshoot at discontinuities is called the *Gibbs effect*, and is discussed in Chapter 11. Another frequency domain interpolation technique is presented in Chapter 3, adding zeros between the time domain samples and low-pass filtering.

Multiplying Signals (Amplitude Modulation)

An important Fourier transform property is that *convolution* in one domain corresponds to *multiplication* in the other domain. One side of this was discussed in the last chapter: time domain signals can be convolved by multiplying their frequency spectra. Amplitude modulation is an example of the reverse situation, multiplication in the time domain corresponds to convolution in the frequency domain. In addition, amplitude modulation provides an excellent example of how the elusive *negative* frequencies enter into everyday science and engineering problems.

Audio signals are great for short distance communication; when you speak, someone across the room hears you. On the other hand, radio frequencies are very good at propagating long distances. For instance, if a 100 volt, 1 MHz sine wave is fed into an antenna, the resulting radio wave can be detected in the next *room*, the next *country*, and even on the next *planet*. **Modulation** is the process of merging two signals to form a third signal with desirable characteristics of both. This always involves nonlinear processes such as multiplication; you can't just add the two signals together. In radio communication, modulation results in radio signals that can propagate long distances *and* carry along audio or other information.

Radio communication is an extremely well developed discipline, and many modulation schemes have been developed. One of the simplest is called **amplitude modulation**. Figure 10-14 shows an example of how amplitude modulation appears in both the time and frequency domains. Continuous signals will be used in this example, since modulation is usually carried out in analog electronics. However, the whole procedure could be carried out in discrete form if needed (the shape of the future!).

Figure (a) shows an audio signal with a DC bias such that the signal always has a positive value. Figure (b) shows that its frequency spectrum is composed of frequencies from 300 Hz to 3 kHz, the range needed for voice communication, plus a spike for the DC component. All other frequencies have been removed by analog filtering. Figures (c) and (d) show the **carrier wave**, a pure sinusoid of much higher frequency than the audio signal. In the time domain, amplitude modulation consists of *multiplying* the audio signal by the carrier wave. As shown in (e), this results in an oscillatory waveform that has an instantaneous amplitude proportional to the original audio signal. In the jargon of the field, the *envelope* of the carrier wave is equal to the modulating signal. This signal can be routed to an antenna, converted into a radio wave, and then detected by a receiving antenna. This results in a signal identical to (e) being generated in the radio receiver's electronics. A *detector* or *demodulator* circuit is then used to convert the waveform in (e) back into the waveform in (a).

Since the time domain signals are multiplied, the corresponding frequency spectra are convolved. That is, (f) is found by convolving (b) & (d). Since the spectrum of the carrier is a shifted delta function, the spectrum of the

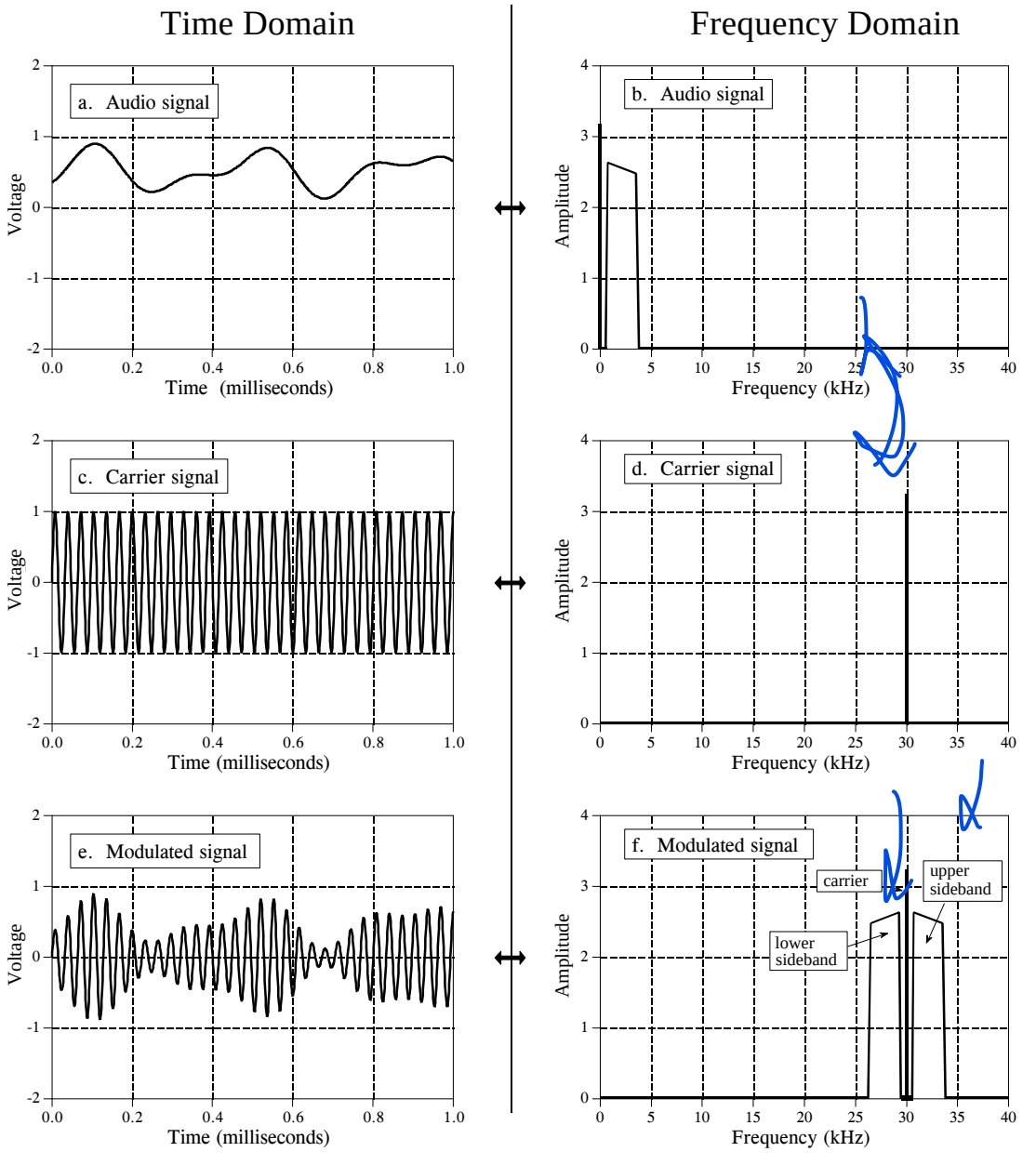

FIGURE 10-14
Amplitude modulation. In the time domain, amplitude modulation is achieved by multiplying the audio signal, (a), by the carrier signal, (c), to produce the modulated signal, (e). Since multiplication in the time domain corresponds to convolution in the frequency domain, the spectrum of the modulated signal is the spectrum of the audio signal shifted to the frequency of the carrier.

modulated signal is equal to the audio spectrum *shifted* to the frequency of the carrier. This results in a modulated spectrum composed of three components: a **carrier wave**, an **upper sideband**, and a **lower sideband**.

These correspond to the three parts of the original audio signal: the DC component, the positive frequencies between 0.3 and 3 kHz, and the negative

frequencies between -0.3 and -3 kHz, respectively. Even though the negative frequencies in the original audio signal are somewhat elusive and abstract, the resulting frequencies in the lower sideband are as real as you could want them to be. The ghosts have taken human form!

Communication engineers live and die by this type of frequency domain analysis. For example, consider the frequency spectrum for television transmission. A standard TV signal has a frequency spectrum from DC to 6 MHz. By using these frequency shifting techniques, 82 of these 6 MHz wide channels are stacked end-to-end. For instance, channel 3 is from 60 to 66 MHz, channel 4 is from 66 to 72 MHz, channel 83 is from 884 to 890 MHz, etc. The television receiver moves the desired channel back to the DC to 6 MHz band for display on the screen. This scheme is called **frequency domain multiplexing**.

The Discrete Time Fourier Transform

The Discrete Time Fourier Transform (DTFT) is the member of the Fourier transform family that operates on *aperiodic*, *discrete* signals. The best way to understand the DTFT is how it relates to the DFT. To start, imagine that you acquire an N sample signal, and want to find its frequency spectrum. By using the DFT, the signal can be decomposed into $N/2 + 1$ sine and cosine waves, with frequencies equally spaced between zero and one-half of the sampling rate. As discussed in the last chapter, padding the time domain signal with zeros makes the period of the time domain *longer*, as well as making the spacing between samples in the frequency domain *narrower*. As N approaches infinity, the time domain becomes *aperiodic*, and the frequency domain becomes a *continuous* signal. This is the DTFT, the Fourier transform that relates an *aperiodic*, *discrete* signal, with a *periodic*, *continuous* frequency spectrum.

The mathematics of the DTFT can be understood by starting with the synthesis and analysis equations for the DFT (Eqs. 8-2, 8-3 and 8-4), and taking N to infinity:

EQUATION 10-1
The DTFT analysis equation. In this relation, $x[n]$ is the time domain signal with n running from 0 to $N-1$. The frequency spectrum is held in: $ReX(\omega)$ and $ImX(\omega)$, with ω between 0 and pi.

$$ReX(\omega) = \sum_{n=-\infty}^{+\infty} x[n] \cos(\omega n)$$

$$ImX(\omega) = -\sum_{n=-\infty}^{+\infty} x[n] \sin(\omega n)$$

EQUATION 10-2
The DTFT synthesis equation.

$$x[n] = \frac{1}{\pi} \int_0^{\pi} ReX(\omega) \cos(\omega n) - ImX(\omega) \sin(\omega n)\, d\omega$$

There are many subtle details in these relations. First, the time domain signal, $x[n]$, is still discrete, and therefore is represented by *brackets*. In comparison, the frequency domain signals, $Re\,X(\omega)$ & $Im\,X(\omega)$, are continuous, and are thus written with *parentheses*. Since the frequency domain is continuous, the synthesis equation must be written as an integral, rather than a summation.

As discussed in Chapter 8, frequency is represented in the DFT's frequency domain by one of three variables: k, an index that runs from 0 to $N/2$; f, the fraction of the sampling rate, running from 0 to 0.5; or ω, the fraction of the sampling rate expressed as a natural frequency, running from 0 to π. The spectrum of the DTFT is continuous, so either f or ω can be used. The common choice is ω, because it makes the equations shorter by eliminating the always present factor of 2π. Remember, when ω is used, the frequency spectrum extends from 0 to π, which corresponds to DC to one-half of the sampling rate. To make things even more complicated, many authors use Ω (an upper case omega) to represent this frequency in the DTFT, rather than ω (a lower case omega).

When calculating the inverse DFT, samples 0 and $N/2$ must be divided by two (Eq. 8-3) before the synthesis can be carried out (Eq. 8-2). This is not necessary with the DTFT. As you recall, this action in the DFT is related to the frequency spectrum being defined as a *spectral density*, i.e., amplitude per unit of bandwidth. When the spectrum becomes continuous, the special treatment of the end points disappear. However, there is still a normalization factor that must be included, the $2/N$ in the DFT (Eq. 8-3) becomes $1/\pi$ in the DTFT (Eq. 10-2). Some authors place these terms in front of the *synthesis* equation, while others place them in front of the *analysis* equation. Suppose you start with some time domain signal. After taking the Fourier transform, and then the Inverse Fourier transform, you want to end up with what you started. That is, the $1/\pi$ term (or the $2/N$ term) must be encountered somewhere along the way, either in the forward or in the inverse transform. Some authors even split the term between the two transforms by placing $1/\sqrt{\pi}$ in front of both.

Since the DTFT involves infinite summations and integrals, it cannot be calculated with a digital computer. Its main use is in theoretical problems as an alternative to the DFT. For instance, suppose you want to find the frequency response of a system from its impulse response. If the impulse response is known as an *array of numbers*, such as might be obtained from an experimental measurement or computer simulation, a DFT program is run on a computer. This provides the frequency spectrum as another *array of numbers*, equally spaced between 0 and 0.5 of the sampling rate.

In other cases, the impulse response might be know as an *equation*, such as a sinc function (described in the next chapter) or an exponentially decaying sinusoid. The DTFT is used here to mathematically calculate the frequency domain as another *equation*, specifying the entire continuous curve between 0 and 0.5. While the DFT could also be used for this calculation, it would only provide an equation for *samples* of the frequency response, not the entire curve.

Parseval's Relation

Since the time and frequency domains are equivalent representations of the same signal, they must have the same *energy*. This is called Parseval's relation, and holds for all members of the Fourier transform family. For the DFT, Parseval's relation is expressed:

EQUATION 10-3
Parseval's relation. In this equation, $x[i]$ is a time domain signal with i running from 0 to N-1, and $X[k]$ is its *modified* frequency spectrum, with k running from 0 to $N/2$. The *modified* frequency spectrum is found by taking the DFT of the signal, and dividing the first and last frequencies (sample 0 and $N/2$) by the square-root of two.

$$\sum_{i=0}^{N-1} x[i]^2 = \frac{2}{N} \sum_{k=0}^{N/2} Mag\, X[k]^2$$

The left side of this equation is the total energy contained in the time domain signal, found by summing the energies of the N individual samples. Likewise, the right side is the energy contained in the frequency domain, found by summing the energies of the $N/2 + 1$ sinusoids. Remember from physics that *energy* is proportional to the *amplitude squared*. For example, the energy in a spring is proportional to the displacement squared, and the energy stored in a capacitor is proportional to the voltage squared. In Eq. 10-3, $X[f]$ is the frequency spectrum of $x[n]$, with one slight modification: the first and last frequency components, $X[0]$ & $X[N/2]$, have been divided by $\sqrt{2}$. This modification, along with the $2/N$ factor on the right side of the equation, accounts for several subtle details of calculating and summing energies.

To understand these corrections, start by finding the frequency domain representation of the signal by using the DFT. Next, convert the frequency domain into the amplitudes of the sinusoids needed to reconstruct the signal, as previously defined in Eq. 8-3. This is done by dividing the first and last points (sample 0 and $N/2$) by 2, and then dividing all of the points by $N/2$. While this provides the amplitudes of the sinusoids, they are expressed as a *peak* amplitude, not the *root-mean-square* (rms) amplitude needed for energy calculations. In a sinusoid, the peak amplitude is converted to rms by dividing by $\sqrt{2}$. This correction must be made to all of the frequency domain values, *except* sample 0 and $N/2$. This is because these two sinusoids are unique; one is a constant value, while the other alternates between two constant values. For these two special cases, the *peak* amplitude is already equal to the *rms* value. All of the values in the frequency domain are squared and then summed. The last step is to divide the summed value by N, to account for each sample in the frequency domain being converted into a sinusoid that covers N values in the time domain. Working through all of these details produces Eq. 10-3.

While Parseval's relation is interesting from the physics it describes (conservation of energy), it has few practical uses in DSP.