

Complex Signals and Negative Frequency

Complex numbers are used in just about every field of science and engineering.[†] For us, complex signal notation is necessary to describe the effects of quadrature and Fourier processing. It may be fair to say that no other topic in digital signal processing causes more initial discomfort for the beginner than *complex* numbers with their *real* and *imaginary* parts. Why, even the terminology is bizarre. So here's where we review the fundamentals of complex numbers, get comfortable with how they're used to represent complex signals, and examine the notion of negative frequency as it relates to complex signal notation.

To start this discussion, it's important for us to be comfortable with the concept of negative frequency because it's essential in understanding the spectral replication effects of periodic sampling, discrete Fourier transforms, and the various quadrature signal processing techniques discussed in Chapter 7. The convention of negative frequency serves as a consistent and powerful mathematical tool in our analysis of signals. In fact, the use of negative frequency is mandatory when we represent *real signals*, such as a sine or cosine wave, in complex notation.

The difficulty in grasping the idea of negative frequency may be, for some, similar to the consternation felt in the parlors of mathematicians in the Middle Ages when they first encountered negative numbers. Until the thirteenth century, negative numbers were considered *fictitious* because numbers were normally used for counting and measuring. So up to that time, negative numbers just didn't make sense [1,2]. In those days, it was valid to ask, "How can you hold something in your hand that is less than nothing?" The idea of subtracting six from four must have seemed meaningless. Math historians suggest that negative num-

[†] That's because complex sinusoids are solutions to the second-order linear differential equations used to describe so much of nature.

bers were first analyzed in Italy. As the story goes, around the year 1200, the Italian mathematician Leonardo da Pisa (known as Fibonacci) was working on a financial problem whose only valid solution involved a negative number. Undaunted, Leo wrote, "This problem, I have shown to be insoluble unless it is conceded that the first man had a debt." Thus negative numbers arrived on the mathematics scene, never again to be disregarded.

Modern men and women can now appreciate that negative numbers have a direction associated with them. The direction is backward from zero in the context that positive numbers point forward from zero. For example, negative numbers can represent temperatures measured in degrees below zero, minutes before the present if the present is considered as zero time, or money we owe the tax collector when our income is considered positive dollars. So the notion of negative quantities is

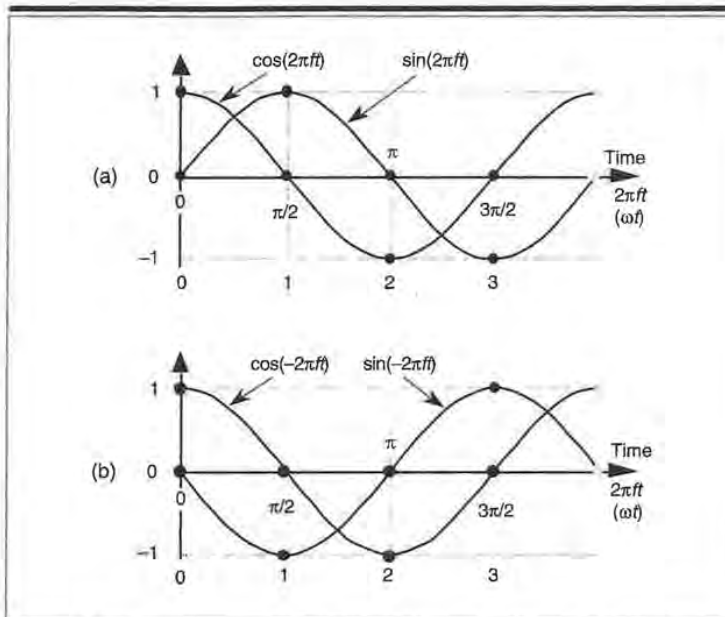


Figure C-1 Positive and negative frequency sinusoids: (a) sine and cosine waves at a positive radian frequency ($2\pi ft$); (b) sine and cosine waves whose arguments are a negative frequency ($-2\pi ft$).

[†] Shortly, we'll represent negative frequency sinusoids graphically using phasor notation and see that the negative frequency phasors are merely rotating in the direction opposite from an arbitrary positive reference direction.

perfectly valid if we just define it properly. As comfortable as we now are with negative numbers, negative frequency remains a troublesome and controversial concept for many engineers [1,2]. This author once encountered a paper in a technical journal which stated: “since negative frequencies cannot exist—.” Well, like negative numbers, negative frequency is a perfectly valid concept as long as we define it properly relative to what we’re used to thinking of as positive frequency. In radians, a negative frequency sinusoid is merely one whose angular argument is negative, as shown in Figure C-1(b).[†]

C.1 Development of Imaginary Numbers

Before we go further in our discussion of negative frequency by way of complex notation, we must first take a deep breath and enter the Twilight Zone of the “ j ” operator and imaginary numbers. You’ve seen the definition $j = \sqrt{-1}$ before. Stated in words, we say that j represents a number which, when multiplied by itself, results in a negative one. Well, this definition causes initial difficulty for the beginner because we know that any number multiplied by itself always results in a positive number. (Unfortunately digital signal processing textbooks typically define j and, then, with justified haste, swiftly carry on with all the ways that the j operator can be used to analyze sinusoidal signals. Readers soon forget about the question: What does $j = \sqrt{-1}$ actually mean? Well, $\sqrt{-1}$ had been on the mathematical scene for some time but wasn’t taken seriously until it had to be used in the sixteenth century to solve cubic equations [3, 4]. Mathematicians reluctantly began to accept the abstract concept of $\sqrt{-1}$ without having to visualize it, as long as its mathematical properties were consistent with the arithmetic of *normal* real numbers. It was Euler’s introduction of the complex plane, used to solve second-order differential equations, that further legitimized the notion of $\sqrt{-1}$ to Europe’s mathematicians in the eighteenth century.[†] Euler showed how complex numbers using the $\sqrt{-1}$ operator had a clean, consistent relationship to the well-known trigonometric functions of sines and cosines [3].

We can get more comfortable with the complex plane representation of imaginary numbers by examining the mathematical properties of the $j = \sqrt{-1}$ operator as shown in Figure C-2(a). Multiplying any number on

[†] Leonhard Euler, born in Switzerland in 1707, is considered by many historians as the world’s greatest mathematician. By the way, in case you don’t speak German, the name *Euler* is pronounced as “oiler.”

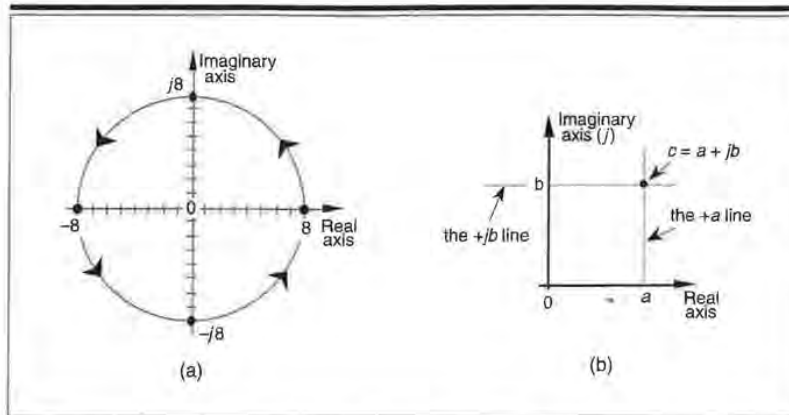


Figure C-2 Numbers on the complex plane: (a) the 90° rotations when a number is multiplied by j ; (b) a complex number, $c = a + jb$, having both real and imaginary parts on the complex plane.

the real axis by j results in an imaginary product that lies on the imaginary axis. The example in Figure C-2(a) shows that if $+8$ is represented by the dot lying on the positive real axis, multiplying $+8$ by j results in an imaginary number, $+j8$, whose position has been rotated 90° counterclockwise from $+8$, putting it on the positive imaginary axis. Similarly, multiplying $+j8$ by j results in another 90° rotation yielding the -8 lying on the negative real axis because $j^2 = -1$. Multiplying -8 by j results in a further 90° rotation giving the $-j8$ lying on the negative imaginary axis. Whenever any number represented by a dot is multiplied by j , the result is a counterclockwise rotation of 90° on the complex plane. (Conversely, multiplication by $-j$ results in a clockwise rotation of -90° on the complex plane.) The complex plane enables us to represent complex numbers having both real and imaginary parts. For example in Figure C-2(b), the complex number $c = a + jb$ is a point lying on the complex plane on neither the real nor the imaginary axis. We locate point c by going $+a$ units along the real axis and up $+b$ units along the imaginary axis. (Just to refresh the reader's memory, Appendix A provides a discussion of the arithmetic of complex numbers.) We can, if we care to, think of all numbers as being complex. Thus, real numbers are complex numbers with no imaginary part, and imaginary numbers are complex numbers with no real part.

Let's pause for a moment here to catch our breath. Don't worry if the ideas of imaginary numbers and the complex plane seem a little mysterious. It's that way for everyone at first—you'll get comfortable with

them the more you use them. (Remember, the $\sqrt{-1}$ operator puzzled Europe's heavyweight mathematicians for hundreds of years.) Not only is the mathematics of complex numbers a bit strange at first, but the terminology is unfortunate. Although the term *imaginary* is an unfortunate one to use, the term *complex* is downright weird. When first encountered, the phrase *complex numbers* makes one think "complicated numbers." This is regrettable because the concept of complex numbers is not really all that complicated.[†]

C.2 Representing Real Signals Using Complex Phasors

OK, now we can turn our attention back to the original discussion of understanding negative frequency when using the mathematics of complex numbers. We start by representing a real sinusoid in the context of the complex plane. To see how this is done, consider the waveform in Figure C-3. Think of a cosine wave defined by $2\cos(\omega t)$ oscillating back and forth on the real axis as time passes. (Here the ω term is frequency in radians/s, and it corresponds to a frequency of $2\pi f$ cycles/s where f is a single cycle/s or 1 Hz.) Now, consider two complex phasors rotating in opposite directions about the time axis, one rotating counterclockwise with a positive angular change ωt and the other phasor rotating with a negative angle, $-\omega t$. The angles, of course, change as a function of time making the $e^{j\omega t}$ and $e^{-j\omega t}$ phasors rotate. Figure C-4 attempts to show, as time passes, one complete rotation of the $e^{j\omega t}$ phasor, for example. That phasor's tip follows a corkscrew path spiraling along and centered about the time axis.

Let's show how the cosine wave is the phasor sum of the two complex phasors. We can visualize the phasors' ωt and $-\omega t$ angles better if we orient the three-dimensional view in Figure C-3 so that we're looking right straight down the time axis in the negative time direction as shown in Figure C-5. The time arrow is coming straight out of the page toward the reader and the two complex phasors, frozen in time, are represented by the bold arrows. Figure C-5 looks reasonably simple, but it contains a wealth of information. Let's first consider the $e^{j\omega t}$ phasor. Using the real and imaginary axis, we can see why the $e^{j\omega t}$ phasor is defined by

[†] The brilliant American engineer Charles P. Steinmetz, who pioneered the use of real and imaginary numbers in electrical circuit analysis in the early twentieth century, refrained from using the term *complex numbers*—he called them "general numbers."

Euler's equation: $\rightarrow$

$$e^{j\omega t} = \cos(\omega t) + j\sin(\omega t) . \quad (C-1)$$

Named in his honor, Eq. (C-1) is now called *Euler's equation*. This expression defines the position of the tip of the $e^{j\omega t}$ arrow in the complex plane. The $\cos(\omega t)$ term in Eq. (C-1) is the phasor's component along the real axis, and the $\sin(\omega t)$ term is the phasor's component along the imaginary, or j , axis. It's very important to realize that the phasor's component projected on the imaginary axis is the real number $\sin(\omega t)$, and *not* $j\sin(\omega t)$. The $+j$ in the second term of Eq. (C-1) merely means that the phasor has a component along

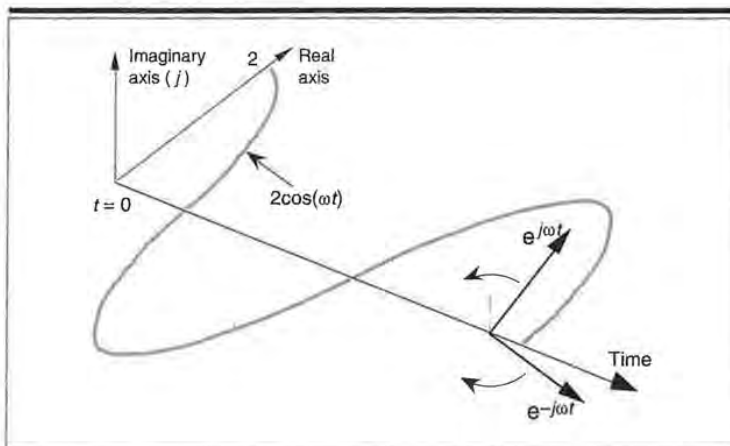


Figure C-3 Cosine represented by two rotating complex phasors.

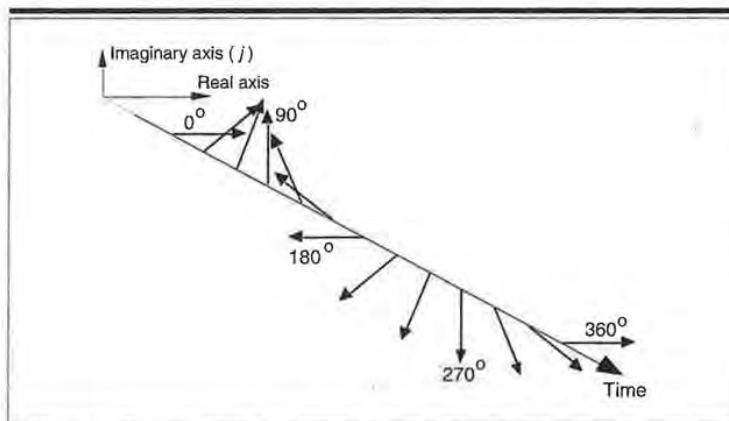


Figure C-4 The rotation of the $e^{j\omega t}$ phasor about the time axis.

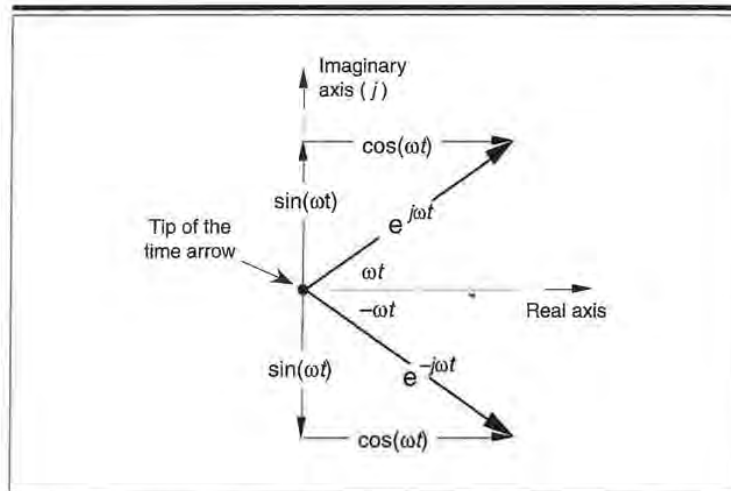


Figure C-5 Complex phasors representing a cosine wave in the complex plane.

the positive j axis whose magnitude is $\sin(\omega t)$. Likewise, the $e^{-j\omega t}$ phasor has a component along the negative j axis whose magnitude is $\sin(\omega t)$. We describe the $e^{-j\omega t}$ phasor with the expression

Another version of Euler's equation: $\rightarrow e^{-j\omega t} = \cos(\omega t) - j\sin(\omega t) .^\dagger \quad (C-2)$

The positive angle ωt is (arbitrarily) defined as the counterclockwise angle between the real axis and the phasor. As time t increases, angle ωt increases and the $e^{j\omega t}$ phasor rotates counterclockwise. The $e^{-j\omega t}$ phasor rotates in the clockwise direction at the same angular frequency as the $e^{j\omega t}$ phasor; that is, when ωt is zero radians, both phasors are at 3:00 o'clock, along the real axis. As times passes, when the $e^{j\omega t}$ phasor is at twelve o'clock, the $e^{-j\omega t}$ phasor is at six o'clock. As the phasors continue rotating, they pass each other, again, exactly at nine o'clock.

Let's look further and see how the sum of the $e^{j\omega t}$ and $e^{-j\omega t}$ phasors represents a real cosine wave. In Figure C-6(a), the phasor angle is $\omega t = \pi/4$ (45 degrees). The bold arrow representing the real cosine wave, $2\cos(\omega t)$, is the phasor sum of the $e^{j\omega t}$ and $e^{-j\omega t}$ phasors. Notice how the cosine wave's arrow

[†] Other versions of Euler's equation, which we can derive from Eqs. (C-1) and (C-2), are $\sin(\omega t) = (e^{j\omega t} - e^{-j\omega t})/2j$, and $\cos(\omega t) = (e^{j\omega t} + e^{-j\omega t})/2$.

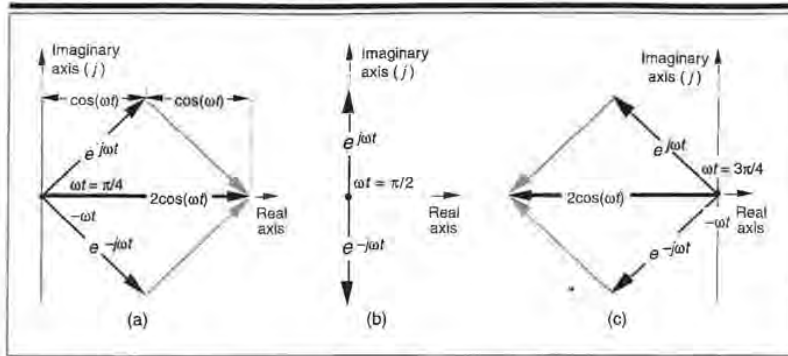


Figure C-6 Complex phasors representing a cosine wave: (a) when $\omega t = \pi/4$; (b) when $\omega t = \pi/2$; (c) when $\omega t = 3\pi/4$.

tip is along the real axis. At a later time, when $\omega t = \pi/2$ radians, the $e^{j\omega t}$ phasor has rotated counterclockwise, the $e^{-j\omega t}$ phasor has rotated clockwise, and the $2\cos(\omega t)$ wave's arrow tip has moved to the left and sits exactly at the zero point on the real axis as shown in Figure C-6(b). We know this is correct because $2\cos(\pi/2) = 0$. As time passes and $\omega t = 3\pi/4$ radians, the $e^{j\omega t}$ and $e^{-j\omega t}$ phasors and their resulting $2\cos(\omega t)$ phasor are shown in Figure C-6(c), which is real and negative. As the complex phasors continue rotating, the $2\cos(\omega t)$ phasor tip merely slides back and forth always remaining on the real axis. Figure C-3 attempts to illustrate this by showing the $2\cos(\omega t)$ wave always lying in the plane of the real axis in our three-dimensional depiction.

Thinking about these phasors, it's clear, now, why the cosine wave can be equated to the sum of two complex phasors by

$$\cos(\omega t) = \frac{e^{j\omega t} + e^{-j\omega t}}{2} \quad (C-3)$$

which is half the sum of Eqs. (C-1) and (C-2). (The example in Figure C-3 used a cosine wave with a peak amplitude of 2 merely to avoid cluttering Figures C-3 through C-6 with a denominator of 2 below each of the $e^{j\omega t}$ and $e^{-j\omega t}$ terms.) If Eq. (C-3) describes a real cosine wave in the complex plane, how do we represent a real sinewave? Well, we can combine Eqs. (C-1) and (C-2) and solve for $\sin(\omega t)$ to get the following standard expression for a sinewave in complex notation:

$$\sin(\omega t) = \frac{e^{j\omega t} - e^{-j\omega t}}{2j} \quad (C-4)$$

To investigate the meaning of this expression, we can multiply both sides of Eq. (C-4) by 2 to get the following alternate expression for a sinewave in terms of complex exponentials:

$$2\sin(\omega t) = \frac{e^{j\omega t} - e^{-j\omega t}}{j} = -j(e^{j\omega t} - e^{-j\omega t}) \quad (\text{C-5})$$

Equation (C-5) looks a little strange, doesn't it? It states that a real sinewave is equal to $-j$ times the sum of two complex numbers! Well, we can decipher Eq. (C-5) by graphically illustrating the complex exponentials as phasors just as we did in Figure C-5. A depiction of just the $(e^{j\omega t} - e^{-j\omega t})$ factor from Eq. (C-5) is the phasors in Figure C-7(a). To account for the $-j$ term in Eq. (C-5), we rotate the phasors in Figure C-7(a) by $\pi/2$, or 90° , in the clockwise direction. So the full $-j(e^{j\omega t} - e^{-j\omega t})$ expression on the right side of Eq. (C-5) is shown in Figure C-7(b). Because the $-e^{-j\omega t}$ phasor rotates clockwise and the $e^{j\omega t}$ phasor rotates counterclockwise, their phasor sum representing $2\sin(\omega t)$ will always lie on the real axis. We know this is true because the $2\sin(\omega t)$ sinewave in Eq. (C-5) is real—like a cosine wave, it has no imaginary part.

To belabor this point a bit further, consider Figure C-8. When the angle $\omega t = 0$, both the $je^{-j\omega t}$ and $-je^{j\omega t}$ phasors lie on the imaginary axis, and their phasor sum sits exactly at the zero point of both axes, as shown in Figure C-8(a). We know this is correct because $2\sin(0) = 0$. At a later time when $\omega t = \pi/4$ radians, the $je^{-j\omega t}$ phasor has rotated clockwise, the $-je^{j\omega t}$ phasor

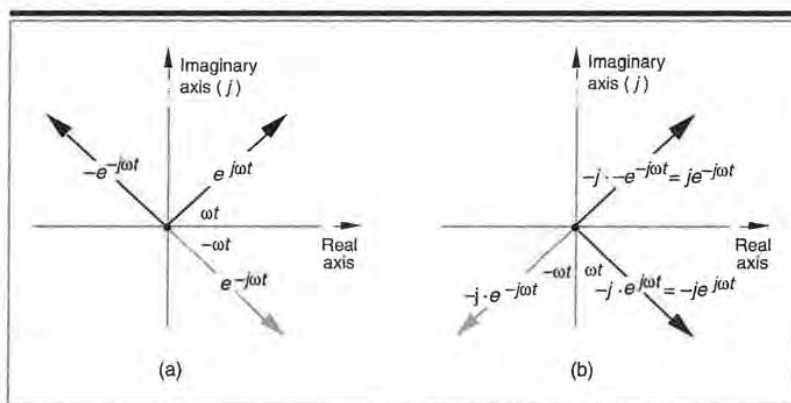


Figure C-7 Complex phasors: (a) phasor representation of $(e^{j\omega t} - e^{-j\omega t})$; (b) phasor representation of $-j(e^{j\omega t} - e^{-j\omega t})$, or $2\sin(\omega t) = -je^{j\omega t} + je^{-j\omega t}$.

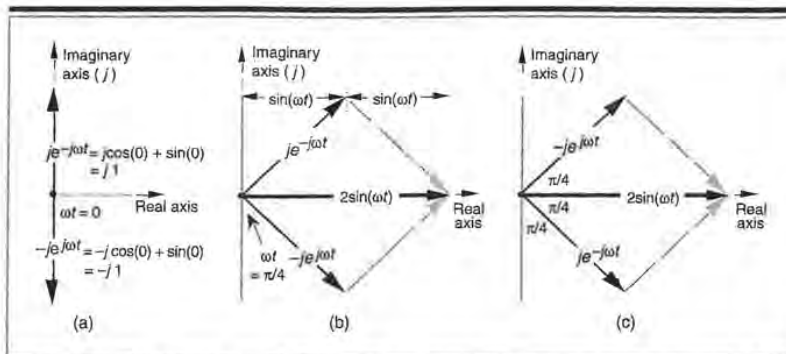


Figure C-8 Complex phasors representing a $2\sin(\omega t)$ sinewave: (a) when $\omega t = 0$; (b) when $\omega t = \pi/4$; (c) when $\omega t = 3\pi/4$.

has rotated counterclockwise, and the $2\sin(\omega t)$ wave's arrow tip has moved to the right to a positive point on the real axis, as shown in Figure C-8(b). As time passes and $\omega t = 3\pi/4$ radians, the $-je^{-j\omega t}$ phasor is now in the northeast quadrant, and $je^{-j\omega t}$ phasor is directed southeast. They and their $2\sin(\omega t)$ phasor sum are shown in Figure C-8(c). As the complex phasors continue rotating, the $2\sin(\omega t)$ phasor oscillates back and forth, always remaining on the real axis.

To keep the reader's mind from spinning like our complex phasors, please realize that the sole purpose of Figures C-6 through C-8 is to validate the complex expressions of the cosine and sinewave given in Eqs. (C-3) and (C-4).

C.3 Representing Real Signals Using Negative Frequencies

OK, let's keep in mind one of the goals of this discussion—why the concept of negative frequency is valid when real signals are represented in complex notation. We can do this by comparing the relationship between a $\cos(\omega_o t)$ wave and a $\sin(\omega_o t)$ wave, using Eqs. (C-3) and (C-4) and combining them graphically on a three-dimensional complex frequency axis as shown in Figure C-9. If we say that the angular frequency is ω_o radians/second $= 2\pi f_o$ Hz, the time-varying composite angle is $\omega_o t$ radians $= 2\pi f_o t$ cycles at time t . Look at Figure C-9 very carefully—it's a snapshot of the complex phasors that comprise a real cosine wave (solid arrows) from Eq. (C-3) and a real sinewave (shaded arrows) from Eq. (C-4) when time equals zero, or $\omega_o t = 0$.

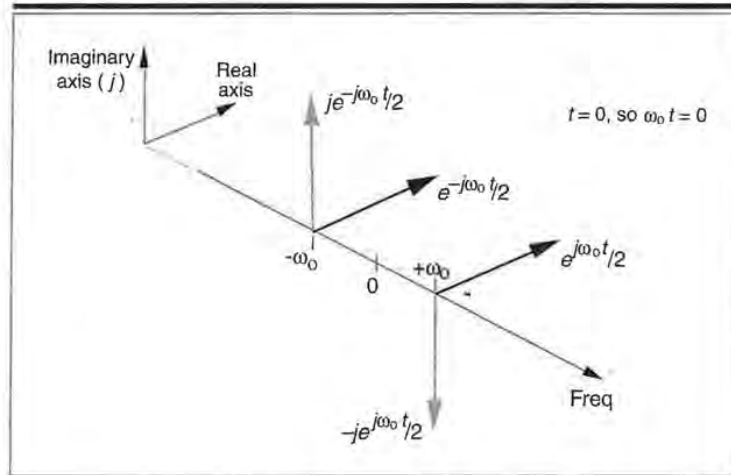


Figure C-9 Relationship of the complex representations of the $\cos(\omega_0 t) = (e^{j\omega_0 t} + e^{-j\omega_0 t})/2$ (solid arrows) and the $\sin(\omega_0 t) = (e^{j\omega_0 t} - e^{-j\omega_0 t})/j2 = (-je^{j\omega_0 t} + je^{-j\omega_0 t})/2$ (shaded arrows), when $\omega_0 t = 0$.

This figure shows that, when we use complex notation, $e^{j\omega t}$ and $e^{-j\omega t}$ are the fundamental constituents of a sinusoid, not $\sin(\omega t)$ or $\cos(\omega t)$ because both $\sin(\omega t)$ and $\cos(\omega t)$ are made up of $e^{j\omega t}$ and $e^{-j\omega t}$ components. On the frequency axis, the notion of negative frequency is seen as those phasors located at $-\omega_0$ radians on the frequency axis. (If we were to take the discrete Fourier transform of a cosine wave and plot the complex results, we'd get exactly those bold phasors in Figure C-9. Likewise, the discrete Fourier transform of a sinewave results in the shaded phasors in Figure C-9.) Figure C-9 also reiterates the fact that a cosine wave is merely a sinewave shifted in time. If we let the phasors in Figure C-9 rotate 90° ($\pi/2$ radians, or a quarter of a cycle), they're oriented as shown in Figure C-10. So the sinewave's shaded phasors, in Figure C-10, are oriented exactly as the cosine wave's phasors in Figure C-9. The relationship between Figures C-9 and C-10 illustrates the two following trigonometric identities:

$$\sin(\theta) = \cos(\theta - \pi/2), \quad (\text{C-6})$$

and

$$\cos(\theta) = -\sin(\theta - \pi/2). \quad (\text{C-7})$$

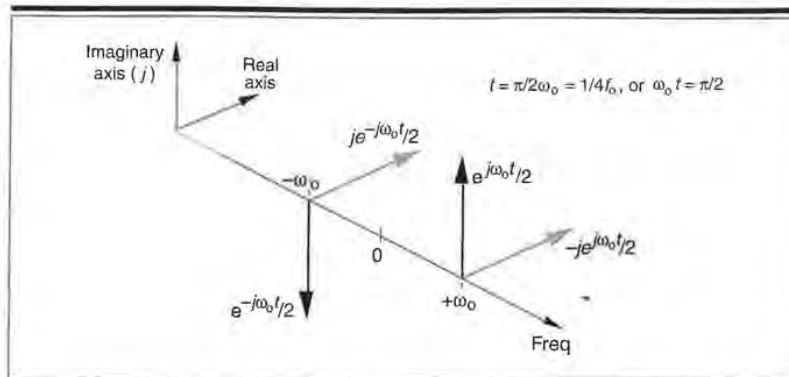


Figure C-10 Relationship of the complex representations of $\cos(\omega_0 t)$ (solid arrows) and $\sin(\omega_0 t)$ (shaded arrows) when $\omega_0 t = \pi/2$.

Putting Eqs. (C-6) and (C-7) into words relative to the two figures above, we can say that the sine phasors in Figure C-10 are equal to the cosine phasors in Figure C-9, and the cosine phasors in Figure C-10 are equal to the negative of the sine phasors in Figure C-9. This tells us that a sinusoidal wave can be described as a sinewave or a cosine wave depending on the time we start looking at it; that is, a delay in the time domain manifests itself as a phase shift in the frequency domain. This effect is very important in digital signal processing, so remember it.[†]

If we use the same axis as before, we can illustrate a sinusoidal wave at an arbitrary phase angle ϕ , $A\cos(\omega_0 t + \phi)$, as shown by the bold phasors in Figure C-11(a). The magnitude of the two bold phasors is, of course, $A/2$, so they'll sum to A when they're aligned along the real axis. Figure C-11(a) shows that the phasors are the *phasor sum* of their real and imaginary parts. Figure C-11(b) shows the real and imaginary components aligned with the real and imaginary axis. Why bother showing the real and imaginary parts as in Figure C-11(b)? Because this signal representation leads us to the graphical form used so often in the digital signal processing literature of quadrature signal processing.

Think for a moment of a real signal comprising seven sinusoids over a bandwidth B centered about a carrier frequency of ω_0 . We could represent that signal's spectrum by the bold phasors in the complex frequency domain of Figure C-12(a). If we had another signal with an unlimited number of sinusoids over the bandwidth B , we could forego drawing

[†] For additional examples of this shifting property, check out Section 3.6.

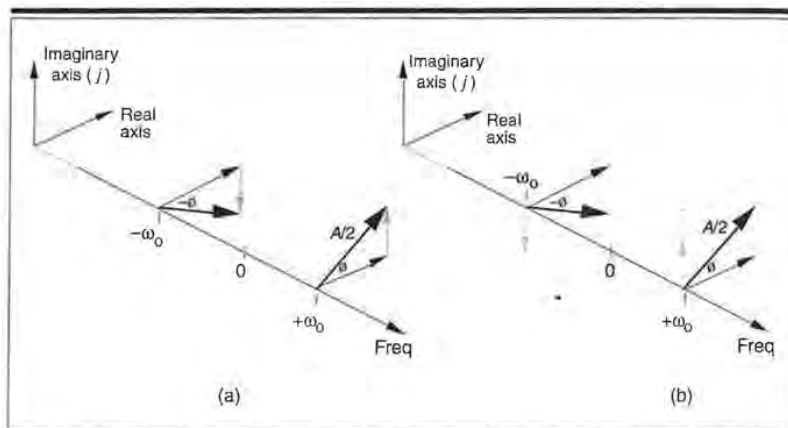


Figure C-11 Two depictions of the complex representation of $A \cos(\omega_0 t + \phi)$ indicated by the bold phasors. The thin solid arrows are the real parts, and the thin shaded arrows are the imaginary parts.

individual phasors and merely show the spectrum by joining the tips of the phasors with the bold lines in Figure C-12(b). This signal has a continuous spectral envelope over the bandwidth B , and that's the typical signal spectrum representation used in the literature of quadrature processing. Figure C-12(b) shows the projection of the signal's constant-amplitude rectangular spectrum on both the real and the imaginary axis.

In quadrature processing, by convention, the real part of the spectrum is called the *in-phase* component, and the imaginary part of the spectrum is called the *quadrature-phase* component. The signal in Figure C-12(b) is real, and, in the time domain, it can be represented digitally by a series of amplitude values that have nonzero real parts and zero-valued imaginary parts. We're not forced to use complex notation to represent it in the time domain—the signal is real. (If it was a continuous physical signal, it could be transmitted over a single conductor.) Real signals always have positive and negative frequency spectral components. For any real signal, the positive and negative frequency components of its in-phase (real) spectrum always have even symmetry about the zero frequency point; that is, the in-phase part's positive and negative frequency components are mirror images of each other. Conversely, the positive and negative frequency components of its quadrature (imaginary) spectrum are always complex conjugates of each other [6]. This means that the phase angle of any given positive quadrature frequency component is the negative of the phase angle of the correspond-

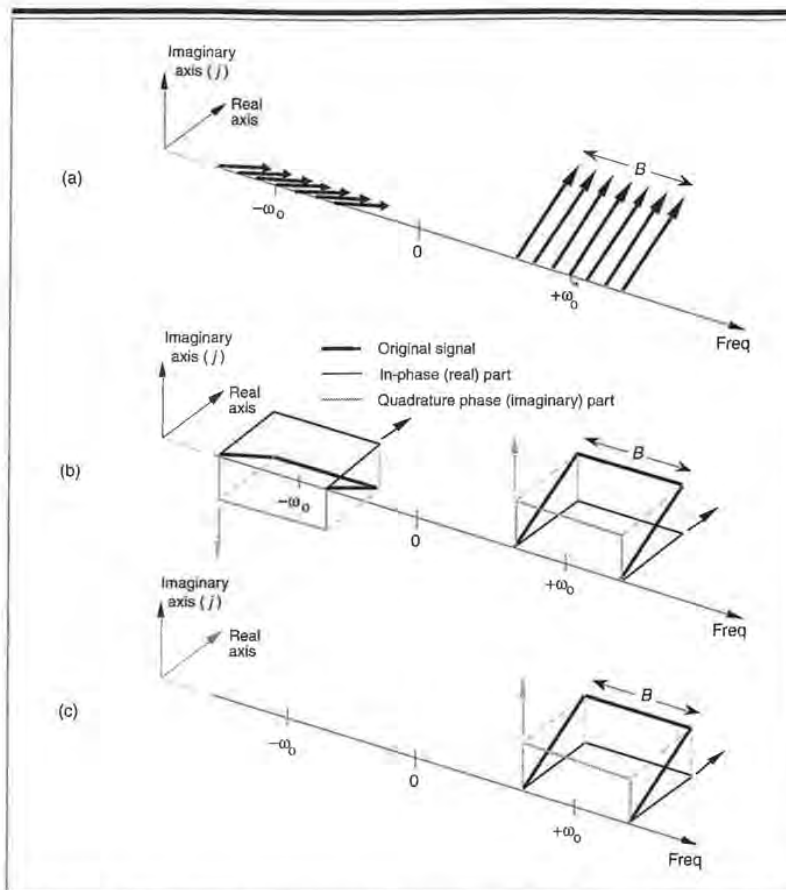


Figure C-12 Quadrature representation of bandpass signals: (a) real signal containing seven sinusoids over bandwidth B ; (b) real signal containing an infinite number of sinusoids over bandwidth B ; (c) complex signal of bandwidth B .

ing quadrature negative frequency component, as shown by the dashed arrows in Figure C-11. This is the invariant nature of real signals when their spectra are represented using complex notation.

C.4 Complex Signals and Quadrature Mixing

Complex signals, on the other hand, are a different story. Figure C-12(c) shows the spectrum of an arbitrary complex signal of bandwidth B . In

the time domain, it can only be represented by amplitude values that have nonzero real and nonzero imaginary parts. We're forced to use complex notation to represent it in the time domain—the signal is complex. (As a continuous physical signal, a complex signal could only be transmitted using two conductors.) Complex signals aren't forced to have the symmetric real and the conjugate symmetric imaginary spectral components characteristic of real signals—complex signals can have just positive (or just negative) frequency components, as shown in Figure C-12(c).

We can summarize all that we've covered in this chapter by considering an interesting topic—the spectrum translation effects of quadrature mixing [7]. Figure C-13(a) shows the triangular spectrum of an arbitrary real signal centered about ω_0 radians/s. In quadrature mixing, we translate the original signal's spectrum down to the zero frequency point before digitization. This is typically done by multiplying the original signal by a cosine wave of ω_0 radians/s, in the time domain, to get the in-phase spectral component of the input signal as shown in Figure C-13(b). It's important to see that Figure C-13(b) is the product of Figure C-13(a) and the solid cosine phasors in Figure C-9. The $e^{-j\omega_0 t}/2$ complex cosine phasor from Figure C-9 translates both Figure C-13(a)'s original input signal spectral bandwidths toward the negative frequency direction, as shown in Figure C-13(b). Likewise, the $e^{j\omega_0 t}/2$ complex cosine phasor from Figure C-9 translates both original input signal spectral bandwidths toward the positive frequency direction, also shown in Figure C-13(b). Although not obvious in the figure, the magnitude in Figure C-13(b) is half the magnitude of that in Figure C-13(a) because of the 1/2 scale factors in the cosine expressions in Eq. (C-3) and Figure C-9.

To get the quadrature-phase spectral components of our input signal, we multiply the input signal by a sinewave of ω_0 radians/s to get the spectrum shown in Figure C-13(c). Again, Figure C-13(c) is the product of Figure C-13(a) and the dashed sine phasors in Figure C-9. The $je^{-j\omega_0 t}/2$ dashed complex sine phasor from Figure C-9 translates both Figure C-13(a)'s original input signal spectral bandwidths toward the negative frequency direction, but aligned with the imaginary $+j$ axis as shown in Figure C-13(c). The $-je^{j\omega_0 t}/2$ dashed complex sine phasor from Figure C-9 translates both original input signal spectral bandwidths toward the positive frequency direction aligned with the imaginary $-j$ axis, also shown in Figure C-13(c). Figure C-13(d) shows both the in-phase and quadrature spectral bands combined on the complex frequency plane axis. (Examples of this quadrature mixing scheme are discussed in Sections 7.1 and 7.2.)

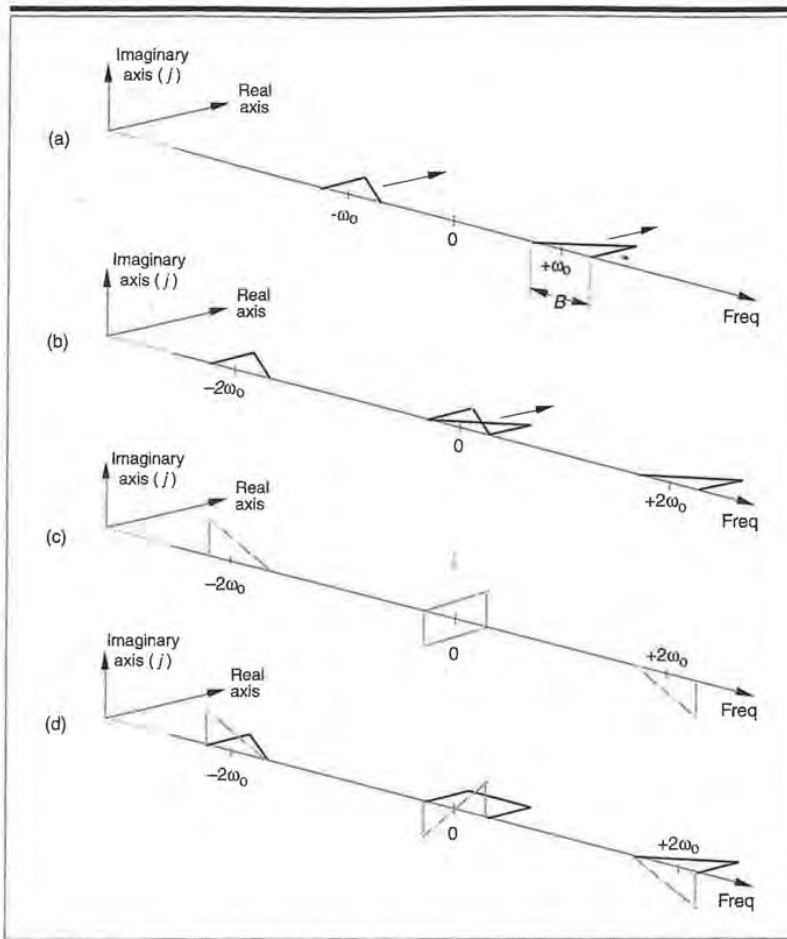


Figure C-13 Quadrature mixing of a real bandpass signal with even symmetric spectral components: (a) original real input signal of bandwidth B ; (b) in-phase spectral components of the input signal; (c) quadrature-phase spectral components of the input signal; (d) combined in-phase and quadrature-phase components.

Providing another example to demonstrate the frequency translation that takes place in complex signal mixing, Figure C-14 shows the spectral results of quadrature mixing when the input signal is real and has conjugate positive and negative frequency components. Section 10.1, with its discussion of digital mixing sequences, provides several practical examples of

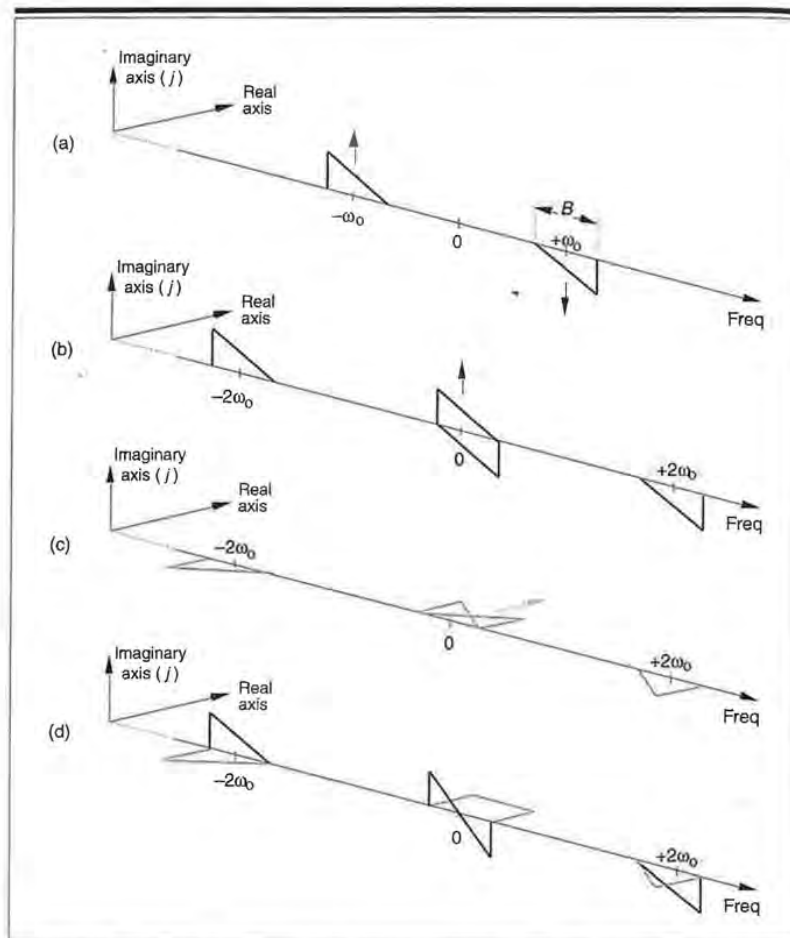


Figure C-14 Quadrature mixing of a real bandpass signal with complex conjugate symmetric spectral components: (a) original real input signal of bandwidth B ; (b) in-phase spectral components of the input signal; (c) quadrature-phase spectral components of the input signal; (d) combined in-phase and quadrature-phase components.

the spectral translation effects of quadrature mixing. That concludes our introduction to complex signals and negative frequency. For further material on these topics, the reader is encouraged to review reference [8], a terrific series of papers detailing how complex signals and negative frequencies relate to modern-day digital communications signals.

References

- [1] Lewis, L. J., et al. *Linear Systems Analysis*, McGraw-Hill Inc., New York, 1969, pp. 193.
- [2] Schwartz, M. *Information, Transmission, Modulation, and Noise*, McGraw-Hill Inc., New York, 1970, pp. 35.
- [3] Struik, D. *A Concise History of Mathematics*, Dover Publications Inc., New York, 1967, pp. 92.
- [4] Bergamini, D. *Mathematics*, Life Science Library, Time Inc., New York, 1963, pp. 152.
- [5] Hsu, H. *Fourier Analysis*, Simon and Schuster, New York, 1970, pp. 75.
- [6] Chester, D. B., and Phillips, G. "Single Chip Digital Down Converter Simplifies RF DSP Applications," *RF Design*, November 1992.
- [7] Boutin, N., "Complex Signals," *RF Design*, December 1989.

Mean, Variance, and Standard Deviation

In our studies, we're often forced to consider noise functions. These are descriptions of noise signals that we cannot explicitly describe with a time-domain equation. Noise functions can be quantified, however, in a worthwhile way using the statistical measures of mean, variance, and standard deviation. Although we only touch, here, on the very broad and important field of statistics, we will describe why, how, and when to use these statistical indicators, so that we can add them to our collection of signal analysis tools. First we'll determine how to calculate these statistical values for a series of discrete data samples, cover an example using a continuous analytical function, and conclude this appendix with a discussion of the probability density functions of several random variables that are common in the field of digital signal processing. So let's proceed by sticking our toes in the chilly waters of the mathematics of statistics to obtain a few definitions.

D.1 Statistical Measures

Consider a continuous sinusoid having a frequency of f_o Hz with a peak amplitude of A_p expressed by the equation

$$x(t) = A_p \sin(2\pi f_o t). \quad (\text{D-1})$$

Equation (D-1) completely specifies $x(t)$ —that is, we can determine $x(t)$'s exact value at any given instant. For example, when time $t = 1/4f_o$, we know that $x(t)$'s amplitude will be A_p and, at the later time $t = 1/2f_o$, $x(t)$'s amplitude will be zero. On the other hand, we have no definite way to express

the successive values of a random function or of random noise.[†] There's no equation like Eq. (D-1) available to predict future noise-amplitude values for example. (That's why they call it random noise.) Statisticians have, however, developed powerful mathematical tools to characterize several properties of random functions. The most important of these properties have been given the names *mean*, *variance*, and *standard deviation*.

Mathematically, the *mean*, or *average*, of N separate values of a sequence x , denoted x_{ave} , is defined as [1]

$$x_{\text{ave}} = \frac{1}{N} \sum_{n=1}^N x(n) = \frac{x(1) + x(2) + x(3) + \dots + x(N)}{N} \quad (\text{D-2})$$

Equation (D-2), already familiar to most people, merely states that the average of a sequence of N numbers is the sum of those numbers divided by N . Graphically, the average can be depicted as that value about which a series of sample values cluster, or congregate, as shown in Figure D-1. If the eight values depicted by the dots in Figure D-1 represent some measured quantity and we applied those values to Eq. (D-2), the average of the series is 5.17, as shown by the dotted line.

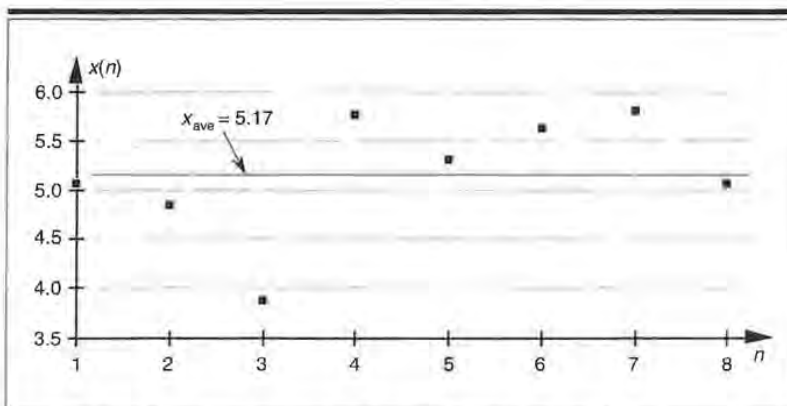


Figure D-1 Average of a sequence of eight values.

[†] We define *random noise* to be unwanted, unpredictable, disturbances contaminating a signal or a data sequence of interest.

Now that we've defined *average*, another key definition is the variance of a sequence, σ^2 , defined as

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N [x(n) - x_{\text{ave}}]^2$$

$$= \frac{[x(1) - x_{\text{ave}}]^2 + [x(2) - x_{\text{ave}}]^2 + [x(3) - x_{\text{ave}}]^2 + \dots + [x(N) - x_{\text{ave}}]^2}{N} \quad (\text{D-3})$$

Sometimes, in the literature, we'll see σ^2 defined with a $1/(N-1)$ factor before the summation instead of the $1/N$ factor in Eq. (D-3). There are subtle statistical reasons why the $1/(N-1)$ factor sometimes gives more accurate results [2]. However, when N is greater than, say 20, as it will be for our purposes, the difference between the two factors will have no practical significance.

Variance is a very important concept because it's the yardstick with which we measure, for example, the effect of quantization errors and the usefulness of signal-averaging algorithms. It gives us an idea how the aggregate values in a sequence fluctuate about the sequence's average and provides us with a well-defined quantitative measure of those fluctuations. (Because the positive square root of the variance, the standard deviation, is typically denoted as σ in the literature, we'll use the conventional notation of σ^2 for the variance.) Equation (D-3) looks a bit perplexing if you haven't seen it before. Its meaning becomes clear if we examine it carefully. The $x(1) - x_{\text{ave}}$ value in the bracket, for example, is the difference between the $x(1)$ value and the sequence average x_{ave} . For any sequence value $x(n)$, the $x(n) - x_{\text{ave}}$ difference, which we denote as $\Delta(n)$, can be either positive or negative, as shown in Figure D-2. Specifically, the differences $\Delta(1)$, $\Delta(2)$, $\Delta(3)$, and $\Delta(8)$ are negative because their corresponding sequence values are below the sequence average shown by the dotted line. If we replace the $x(n) - x_{\text{ave}}$ difference terms in Eq. (D-3) with $\Delta(n)$ terms, the variance can be expressed as

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N [\Delta(n)]^2, \text{ where } \Delta(n) = x(n) - x_{\text{ave}} \quad (\text{D-4})$$

The reader might wonder why the squares of the differences are summed, instead of just the differences themselves. If we just add the differences, some of the negative $\Delta(n)$ s will cancel some of the positive $\Delta(n)$ s

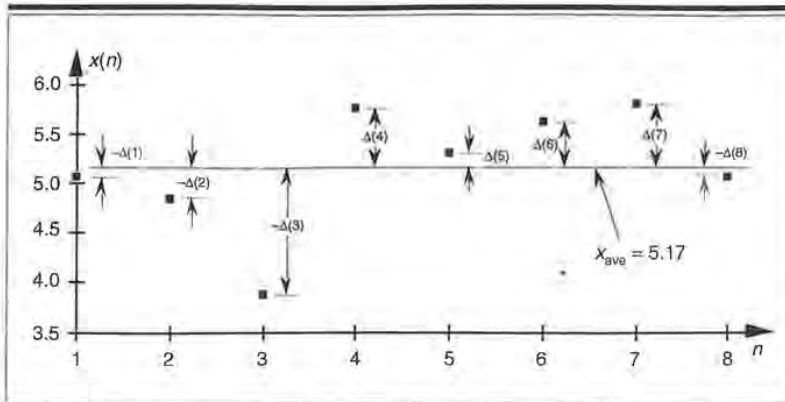


Figure D-2 Difference values $\Delta(n)$ of the sequence in Figure D-1.

resulting in a sum that may be too small. For example, if we add the $\Delta(n)$ s in Figure D-2, the positive $\Delta(6)$ and $\Delta(7)$ values and the negative $\Delta(3)$ value will just about cancel each other out and we don't want that. Because we need an unsigned measure of each difference, we use the difference-squared terms as indicated by Eq. (D-4). In that way, individual $\Delta(n)$ difference terms will contribute to the overall variance regardless of whether the difference is positive or negative. Plugging the $\Delta(n)$ values from the example sequence in Figure D-2 into Eq. (D-4), we get a variance value of 0.34. Another useful measure of a signal sequence is the square root of the variance known as the *standard deviation*. Taking the square root of Eq. (D-3) to get the standard deviation σ ,

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{1}{N} \sum_{n=1}^N [x(n) - x_{\text{ave}}]^2} \quad (\text{D-5})$$

So far, we have three measurements to use in evaluating a sequence of values: the average x_{ave} , the variance σ^2 , and the standard deviation σ . Where x_{ave} indicates about what constant level the individual sequence values vary, σ^2 is a measure of the magnitude of the noise fluctuations about the average x_{ave} . If the sequence represents a series of random signal samples, we can say that x_{ave} specifies the average, or constant, value of the signal. The variance σ^2 is the magnitude squared, or power, of the fluctuating component of the signal. The standard deviation, then, is an indication of the magnitude of the fluctuating component of the signal.

D.2 Standard Deviation, or RMS, of a Continuous Sinewave

For sinewaves, electrical engineers have taken the square root of Eq. (D-3), with $x_{\text{ave}} = 0$, and defined a useful parameter, called the *rms* value, that's equal to the standard deviation. For discrete samples, that parameter, $x(n)_{\text{rms}}$, is defined as

$$x(n)_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{n=1}^N x(n)^2} \quad (\text{D-6})$$

The $x(n)_{\text{rms}}$ in Eq. (D-6), obtained by setting $x_{\text{ave}} = 0$ in Eq. (D-5), is the square root of the mean (average) of the squares of the sequence $x(n)$. For a continuous sinusoid $x(t) = A_p \sin(2\pi ft) = A_p \sin(\omega t)$ whose average value is zero, $x(t)_{\text{rms}}$ is $x(t)_{\text{rms-sinewave}}$ defined as

$$\begin{aligned} x(t)_{\text{rms-sinewave}} &= \sqrt{\frac{1}{2\pi} \int_0^{2\pi} [A_p \sin(\omega t)]^2 d(\omega t)} \\ &= \sqrt{\frac{A_p^2}{2\pi} \int_0^{2\pi} \frac{1}{2} [1 - \cos(\omega t)] d(\omega t)} \\ &= \sqrt{\frac{A_p^2}{2\pi} \left[\frac{\omega t}{2} - \frac{1}{2} (\sin(\omega t)) \right]_0^{2\pi}} = \sqrt{\frac{A_p^2}{2\pi} \left[\frac{2\pi}{2} \right]} \\ &= \frac{A_p}{\sqrt{2}} \quad (\text{D-7}) \end{aligned}$$

This $x(t)_{\text{rms-sinewave}} = A_p / \sqrt{2}$ expression is a lot easier to use for calculating average power dissipation in circuit elements than taking the integral of more complicated expressions for instantaneous power dissipation. The variance of a sinewave is, of course, the square of Eq. (D-7), or $A_p^2/2$.

We've provided the equations for the mean (average) and variance of a sequence of discrete values, introduced an expression for the standard

deviation or rms of a sequence, and given an expression for the rms of a continuous sinewave. The next question is "How can we characterize random functions for which there are no equations to predict their values and we have no discrete sample values with which to work?" The answer is that we must use probability density functions.

D.3 The Mean and Variance of Random Functions

To determine the mean or variance of a random function, we use what's called the *probability density function*. The probability density function (PDF) is a measure of the likelihood of a particular value occurring in some function. We can explain this concept with simple examples of flipping a coin or throwing dice as illustrated in Figure D-3(a) and (b). The result of flipping a coin can only be one of two possibilities: heads or tails. Figure D-3(a) indicates this PDF and shows that the probability (likelihood) is equal to one-half for both heads and tails. That is, we have an equal chance that the coin side facing up will be heads or tails. The sum of those two probability values is one, meaning that there's a 100% probability that either a head or a tail will occur.

Figure D-3(b) shows the probability of a particular sum of the upper faces when we throw a pair of dice. This probability function is not uniform because, for example, we're six times more likely to have the die faces add to seven than add to two (snake eyes). We can say that after tossing the dice a large number of times, we should expect that $6/36 = 16.7$ percent of those tosses would result in sevens, and $1/36 = 2.8$ percent of the time we'll get snake eyes. The sum of those eleven probability values

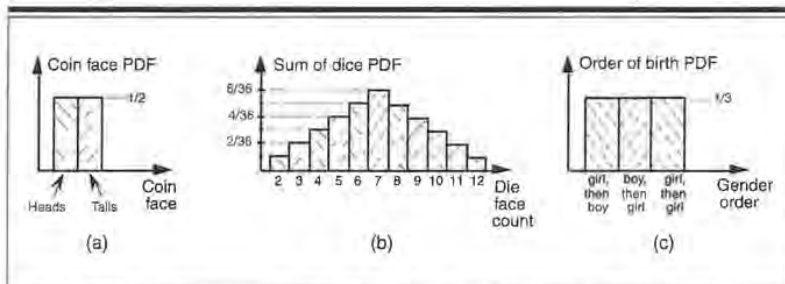


Figure D-3 Simple probability density functions: (a) the probability of flipping a single coin; (b) the probability of a particular sum of the upper faces of two die; (c) the probability of the order of birth of the girl and her sibling.

in Figure D-3(b) is also one, telling us that this PDF accounts for all (100%) of the possible outcomes of throwing the dice.

The fact that PDFs must account for all possible results is emphasized in an interesting way in Figure D-3(c). If a woman says, "Of my two children, one is a girl. What's the probability that she has a sister?" Be careful now—curiously enough, the answer to this controversial question is not a 50–50 chance. There are more possibilities to consider than just the girl having a brother or a sister. We can think of all the possible combinations of birth order of two children such that one child is a girl. Because we don't know the gender of the first-born child, there are three gender order possibilities: girl, then boy; boy, then girl; and girl, then girl as shown in Figure D-3(c). So the possibility of the daughter having a sister is 1/3 instead of 1/2! (Believe it.) Again, the sum of those three 1/3rd probability values is one.

Two important features of PDFs are illustrated by the examples in Figure D-3: PDFs are always positive, and the areas under their *curves* must be equal to unity. The very concept of PDFs make them a positive *likelihood* that a particular result will occur, and the fact that some result must occur is equivalent to saying that there's a probability of one (100% chance) that we'll have a result. For continuous probability density functions, $p(f)$, we indicate these two characteristics by

$$\text{PDF values are never negative: } \rightarrow \quad p(f) \geq 0, \quad (\text{D-8})$$

and

$$\text{The sum of all the PDF values is one: } \rightarrow \quad \int_{-\infty}^{\infty} p(f) df = 1. \quad (\text{D-8'})$$

In Section D.1 we illustrated how to calculate the average (mean) and variance of discrete samples. We can also determine these statistical measures for a random function if we know the PDF of the function. Using μ_f to denote the average of a random function of f , then, μ_f is defined as

$$\mu_f = \int_{-\infty}^{\infty} f \cdot p(f) df, \quad (\text{D-9})$$

and the variance of f is defined as [3]

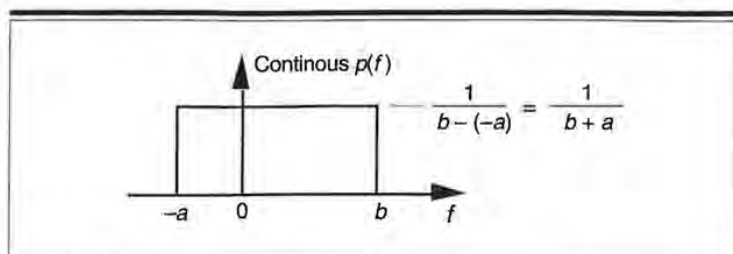


Figure D-4 Continuous, uniform probability density function.

$$\sigma_f^2 = \int_{-\infty}^{\infty} (f - \mu_f)^2 \cdot p(f) df = \int_{-\infty}^{\infty} f^2 \cdot p(f) df - \mu_f^2. \quad (\text{D-10})$$

In digital signal processing, we'll encounter continuous probability density functions that are uniform in value similar to the examples in Figure D-3. In these cases, it's easy to use Eqs. (D-9) and (D-10) to determine their average and variance. Figure D-4 illustrates a uniform, continuous PDF indicating a random function whose values have an equal probability of being anywhere in the range from $-a$ to b . From Eq. (D-8), we know that the area under the curve must be unity (i.e., the probability is 100% that the value will be somewhere under the curve). So the amplitude of $p(f)$ must be the area divided by the width, or $p(f) = 1/(b + a)$. From Eq. (D-9), the average of this $p(f)$ is given by

$$\begin{aligned} \mu_f &= \int_{-\infty}^{\infty} f \cdot p(f) df = \int_{-\infty}^{\infty} f \cdot \frac{1}{b+a} df \\ &= \frac{1}{b+a} \int_{-a}^b f df = \frac{1}{b+a} \cdot \left[\frac{f^2}{2} \right]_{-a}^b = \frac{b^2 - a^2}{2(b+a)} \\ &= \frac{(b+a)(b-a)}{2(b+a)} = \frac{b-a}{2} \end{aligned} \quad (\text{D-11})$$

which happens to be the midpoint in the range from $-a$ to b . The variance of the PDF in Figure D-4 is given by

$$\begin{aligned}
\sigma_f^2 &= \int_{-\infty}^{\infty} f^2 \cdot p(f) df - \mu_f^2 = \int_{-a}^b f^2 \cdot \frac{1}{b+a} df - \frac{(b-a)^2}{4} \\
&= \frac{1}{b+a} \cdot \left[\frac{f^3}{3} \right]_{-a}^b - \frac{(b+a)^2}{4} = \frac{1}{3(b+a)} \cdot (b^3 + a^3) - \frac{(b-a)^2}{4} \\
&= \frac{(b+a)(b^2 - ab + a^2)}{3(b+a)} - \frac{b^2 - 2ab + a^2}{4} \\
&= \frac{b^2 + 2ab + a^2}{12} = \frac{(b+a)^2}{12} \quad (D-12)
\end{aligned}$$

We use the results of Eqs. (D-11) and (D-12) in Chapter 9 to analyze the errors induced by quantization from analog-to-digital converters and the effects of finite word lengths of hardware registers.

D.4 The Normal Probability Density Function

A probability density function that's so often encountered in nature deserves our attention. This function is so common that it's actually called the *normal* probability density function.[†] This function, whose shape is shown in Figure D-5, is important because random data having this distribution is very useful in testing both software algorithms and hardware processors. The normal PDF is defined mathematically by

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-x_{ave})^2 / 2\sigma^2} \quad (D-13)$$

The area under the curve equals one, and the percentages at the bottom of Figure D-5 tell us that, for random functions having a normal distribution, there's a 68.27 percent chance that any particular value of x will differ from the mean by $\leq \sigma$. Likewise, 99.73 percent of all the x data values will be within 3σ of the mean x_{ave} .

[†] The *normal probability density function* is sometimes called the Gaussian function. A scheme for generating data to fit this function is discussed in Section 10.12.

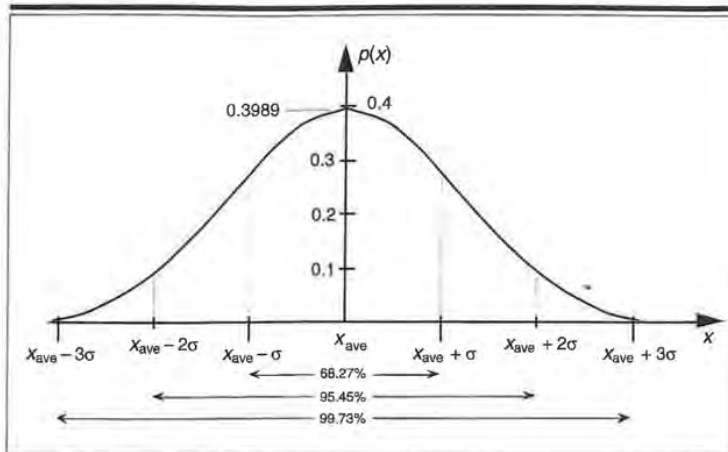


Figure D-5 A normal PDF with mean = x_{ave} and standard deviation = σ .

References

- [1] Papoulis, A. *Probability Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1965, pp. 266–68.
- [2] Miller, I., and Freund, J. *Probability and Statistics for Engineers*, 2nd Ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1977, pp. 118.
- [3] Bendat, J., and Piersol, A. *Measurement and Analysis of Random Data*, John Wiley and Sons, New York, 1972, pp. 61.

Decibels (dB and dBm)

This appendix introduces the logarithmic function used to improve the magnitude resolution of frequency-domain plots when we evaluate signal spectrums, digital filter magnitude responses, and window function magnitude responses. When we use a logarithmic function to plot signal levels in the frequency domain, the vertical axis unit of measure is *decibels*.

E.1 Using Logarithms to Determine Relative Signal Power

In discussing decibels, it's interesting to see how this unit of measure evolved. When comparing continuous (analog) signal levels, early specialists in electronic communications found it useful to define a measure of the difference in powers of two signals. If that difference was treated as the logarithm of a ratio of powers, it could be used as a simple additive measure to determine the overall gain or loss of cascaded electronic circuits. The positive logarithms associated with system components having gain could be added to the negative logarithms of those components having loss to quickly determine the overall gain or loss of the system. With this in mind, the difference between two signal power levels (P_1 and P_2), measured in bels, was defined as the base 10 logarithm of the ratio of those powers, or

$$\text{Power difference} = \log_{10} \left(\frac{P_1}{P_2} \right) \text{ bels.}^\dagger \quad (\text{E-1})$$

[†] The dimensionless unit of measure *bel* was named in honor of Alexander Graham Bell.

The use of Eq. (E-1) led to another evolutionary step because the unit of bel was soon found to be inconveniently large. For example, it was discovered that the human ear could detect audio power level differences of one-tenth of a bel. Measured power differences smaller than one bel were so common that it led to the use of the decibel (bel/10), effectively making the unit of bel obsolete. The decibel (dB), then, is a unit of measure of the relative power difference of two signals defined as

$$\text{Power difference} = 10 \cdot \log_{10} \left(\frac{P_1}{P_2} \right) \text{ dB} \quad (\text{E-2})$$

The logarithmic function $10 \cdot \log_{10}$, plotted in Figure E-1, doesn't seem too beneficial at first glance, but its application turns out to be very useful. Notice the large change in the function's value, when the power ratio (P_1/P_2) is small, and the gradual change when the ratio is large. The effect of this nonlinearity is to provide greater resolution when the ratio P_1/P_2 is small, giving us a good way to recognize very small differences in the power levels of signal spectrums, digital filter responses, and window function frequency responses.

Let's demonstrate the utility of the logarithmic function's variable resolution. First, remember that the power of any frequency-domain sequence representing signal magnitude $|X(m)|$ is proportional to

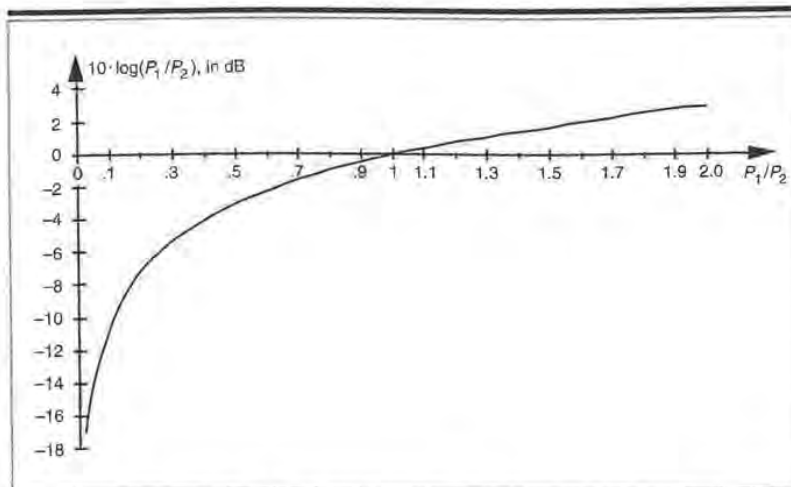


Figure E-1 Logarithmic decibel function of Eq. (E-2).

$|X(m)|$ squared. For convenience, the proportionality constant is assumed to be one, so we say the power of $|X(m)|$ is

$$\text{discrete power spectrum of } X(m) = |X(m)|^2. \quad (\text{E-3})$$

Although equation (E-3) may not actually represent power (in watts) in the classical sense, it's the squaring operation that's important here, because it's analogous to the traditional magnitude squaring operation used to determine the power of continuous signals. (Of course, if $X(m)$ is complex, we can calculate the power spectrum sequence using $|X(m)|^2 = X_{\text{real}}(m)^2 + X_{\text{imag}}(m)^2$.) Taking 10 times the log of Eq. (E-3) allows us to express a power spectrum sequence $X_{\text{dB}}(m)$ in dB as

$$X_{\text{dB}}(m) = 10 \cdot \log_{10}(|X(m)|^2) \text{ dB}. \quad (\text{E-4})$$

Because $\log(x^2) = \log(x) + \log(x) = 2\log(x)$, we can eliminate the squaring operation in Eq. (E-4) by doubling the factor of 10 and represent the power spectrum sequence by the expression

$$X_{\text{dB}}(m) = 20 \cdot \log_{10}(|X(m)|) \text{ dB}. \quad (\text{E-5})$$

Without the need for the squaring operation, Eq. (E-5) is a more convenient way, than Eq. (E-4), to calculate the $X_{\text{dB}}(m)$ power spectrum sequence from the $X(m)$ sequence.

Equations (E-4) and (E-5), then, are the expressions used to convert a linear magnitude axis to a logarithmic magnitude-squared, or power, axis measured in dB. What we most often see in the literature are normalized log magnitude spectral plots where each value of $|X(m)|^2$ is divided by the first $|X(0)|^2$ power value (for $m = 0$) as

$$\text{normalized } X_{\text{dB}}(m) = 10 \cdot \log_{10}\left(\frac{|X(m)|^2}{|X(0)|^2}\right) = 20 \cdot \log_{10}\left(\frac{|X(m)|}{|X(0)|}\right) \text{ dB}. \quad (\text{E-6})$$

The division by the $|X(0)|^2$ or $|X(0)|$ value always forces the first value in the normalized log magnitude sequence $X_{\text{dB}}(m)$ equal to 0 dB.[†] This makes it easy for us to compare multiple log magnitude spectral plots. To illustrate, let's look at the frequency-domain representations of the Hanning

[†] That's because $\log_{10}(|X(0)| / |X(0)|) = \log_{10}(1) = 0$.

and triangular window functions. The magnitudes of those frequency-domain functions are plotted on a linear scale in Figure E-2(a) where we've arbitrarily assigned their peak values to be two. Comparing the two linear scale magnitude sequences, $W_{\text{Hanning}}(m)$ and $W_{\text{triangular}}(m)$, we can see some minor differences between their magnitude values. If we're interested in the power associated with the two window functions, we square the two magnitude functions and plot them on a linear scale as in Figure E-2(b). The difference between the two window function's power sequences is impossible to see above the frequency of, say, $m = 8$ in Figure E-2(b). Here's where the dB scale helps us out. If we plot the normalized log magnitude versions of the two magnitude-squared sequences on a logarithmic dB scale using Eq. (E-6), the difference between the two functions will become obvious.

Normalization, in the case of the Hanning window, amounts to calculating the log magnitude sequence normalized over $|W_{\text{Hanning}}(0)|$ as

$$W_H(m) = 10 \cdot \log_{10} \left(\frac{|W_{\text{Hanning}}(m)|^2}{|W_{\text{Hanning}}(0)|^2} \right) = 20 \cdot \log_{10} \left(\frac{|W_{\text{Hanning}}(m)|}{|W_{\text{Hanning}}(0)|} \right) \text{ dB} \quad (\text{E-7})$$

The normalized log magnitude sequences are plotted in Figure E-2(c). We can now clearly see the difference in the magnitude-squared window functions in Figure E-2(c) as compared to the linear plots in Figure E-2(b). Notice how normalization forced the peak values for both log magnitude functions in Figure E-2(c) to be zero dB. (The dots in Figure E-2 are connected by lines to emphasize the sidelobe features of the two log magnitude sequences.)

Although we've shown the utility of dB plots using window function frequency responses as examples, the dB scale is equally useful when we're plotting signal-power spectrums or digital filter frequency responses. We can further demonstrate the dB scale using a simple digital filter example. Let's say we're designing an 11-tap highpass FIR filter whose coefficients are shown in Figure E-3(a). If the center coefficient $h(5)$ is -0.48 , the filter's frequency magnitude response $|H_{-0.48}(m)|$ can be plotted as the white dots on the linear scale in Figure E-3(b). Should we change $h(5)$ from -0.48 to -0.5 , the new frequency magnitude response $|H_{-0.5}(m)|$ would be the black dots in Figure E-3(b). It's difficult to see much of a difference between $|H_{-0.48}(m)|$ and $|H_{-0.5}(m)|$ on a linear scale. If we used Eq. (E-6) to calculate two normalized log magnitude sequences, they could be plotted as shown in Figure E-3(c) where the filter sidelobe effects of changing $h(5)$ from -0.48 to -0.5 are now easy to see.

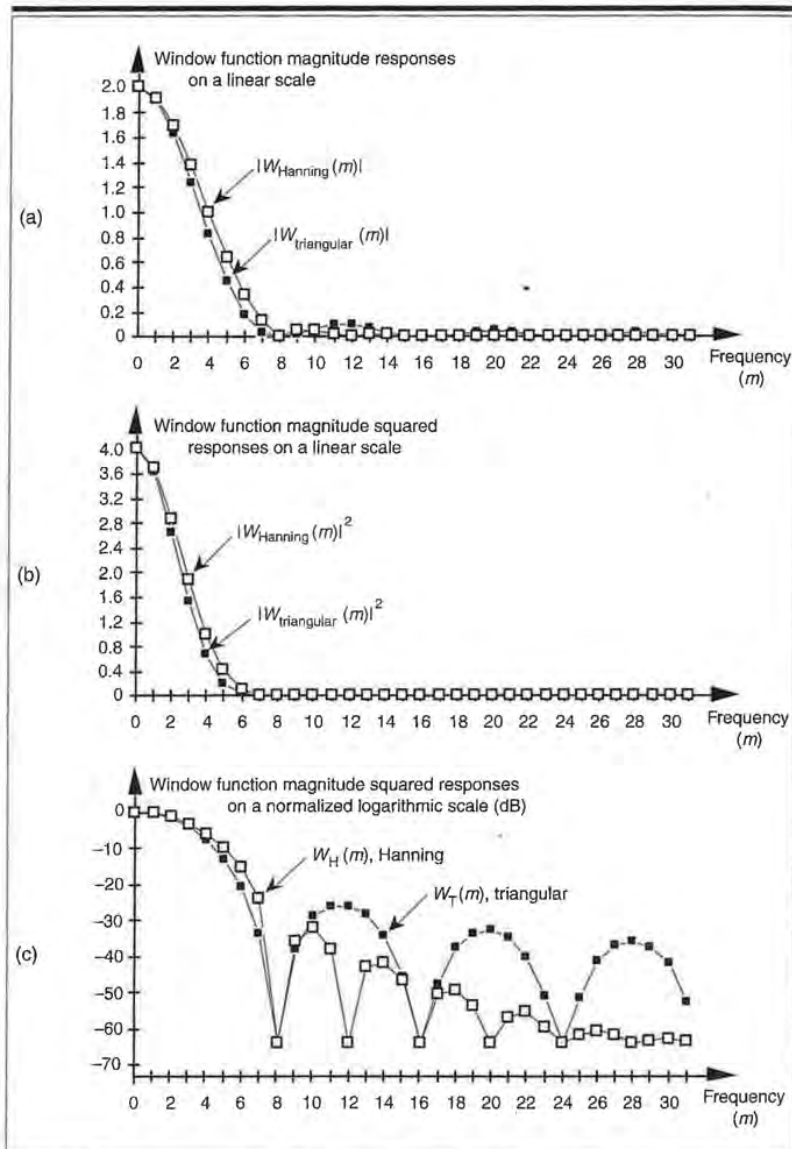


Figure E-2 Hanning (white squares) and triangular (black squares) window functions in the frequency domain: (a) magnitude responses using a linear scale; (b) magnitude-squared responses using a linear scale; (c) log magnitude responses using a normalized dB scale.

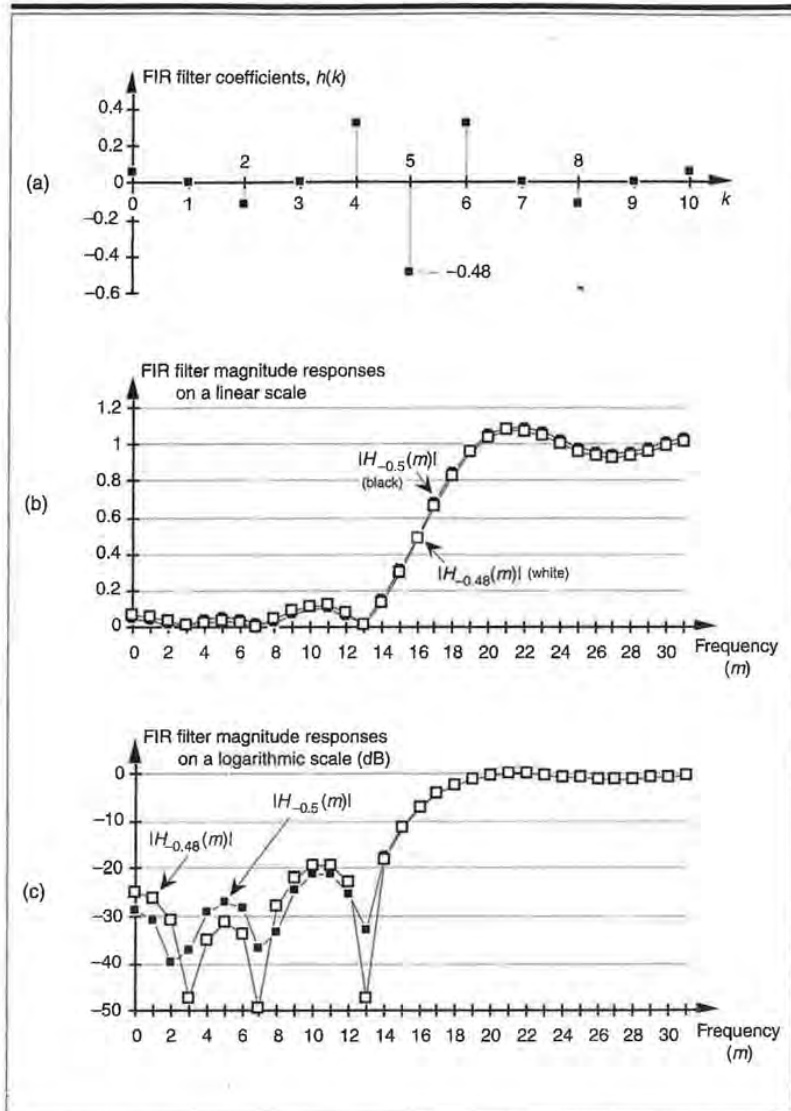


Figure E-3 FIR filter magnitude responses: (a) FIR filter time-domain coefficients; (b) magnitude responses using a linear scale, $|H_{-0.48}(m)|$ is shown by the white squares and $|H_{-0.5}(m)|$ is shown by the black squares; (c) log magnitude responses using the dB scale.

E.2 Some Useful Decibel Numbers

If the reader finds him- or herself using dB scales on a regular basis, there are a few constants worth committing to memory. A power difference of 3 dB corresponds to a power factor of 2; that is, if the magnitude-squared ratio of two different frequency components is 2, then from Eq. (E-2),

$$\text{power difference} = 10 \cdot \log_{10} \left(\frac{2}{1} \right) = 10 \cdot \log_{10}(2) = 3.01 \approx 3 \text{ dB} . \quad (\text{E-8})$$

Likewise, if the magnitude-squared ratio of two different frequency components is 1/2, then the relative power difference is -3 dB because

$$\text{power difference} = 10 \cdot \log_{10} \left(\frac{1}{2} \right) = 10 \cdot \log_{10}(0.5) = -3.01 \approx -3 \text{ dB} . \quad (\text{E-9})$$

Table E-1 lists several magnitude and power ratios vs. dB values worth remembering. Keep in mind that decibels indicate only relative power

Table E-1 Some Useful Logarithmic Relationships

Magnitude Ratio	Magnitude-Squared Power (P_1/P_2) Ratio	Relative dB (approximate)	
$10^{-1/2}$	10^{-1}	-10	$\leftarrow P_1$ is one-tenth P_2
2^{-1}	$2^{-2} = 1/4$	-6	$\leftarrow P_1$ is one-fourth P_2
$2^{-1/2}$	$2^{-1} = 1/2$	-3	$\leftarrow P_1$ is one-half P_2
2^0	$2^0 = 1$	0	$\leftarrow P_1$ is equal to P_2
$2^{1/2}$	$2^1 = 2$	3	$\leftarrow P_1$ is twice P_2
2^1	$2^2 = 4$	6	$\leftarrow P_1$ is four times P_2
$10^{1/2}$	$10^1 = 10$	10	$\leftarrow P_1$ is ten times P_2
10^1	$10^2 = 100$	20	$\leftarrow P_1$ is one hundred times P_2
$10^{3/2}$	$10^3 = 1000$	30	$\leftarrow P_1$ is one thousand times P_2

relationships. For example, if we're told that signal A is 6 dB above signal B, we know that the power of signal A is four times that of signal B, and that the magnitude of signal A is twice the magnitude of signal B. We may not know the absolute power of signals A and B in watts, but we do know that the power ratio is $P_A/P_B = 4$.

E.3 Absolute Power Using Decibels

Let's discuss another use of decibels that the reader may encounter in the literature. It's convenient for practitioners in the electronic communications field to measure continuous signal-power levels referenced to a specific absolute power level. In this way, they can speak of absolute power levels in watts while taking advantage of the convenience of decibels. The most common absolute power reference level used is the milliwatt. For example, if P_2 in Eq. (E-2) is a reference power level of one milliwatt, then

$$\text{absolute power of } P_1 = 10 \cdot \log_{10} \left(\frac{P_1}{P_2} \right) = 10 \cdot \log_{10} \left(\frac{P_1 \text{ in watts}}{1 \text{ milliwatt}} \right) \text{ dBm. (E-10)}$$

The dBm unit of measure in Eq. (E-10) is read as "dB relative to a milliwatt." Thus, if a continuous signal is specified as having a power of 3 dBm, we know that the signal's absolute power level is 2 times one milliwatt, or 2 milliwatts. Likewise, a -10 dBm signal has an absolute power of 0.1 milliwatts.[†]

The reader should take care and not inadvertently use dB and dBm interchangeably. They mean very different things. Again, dB is a relative power level relationship, and dBm is an absolute power level in milliwatts.

[†] Other absolute reference power levels can be used. People involved with high-power transmitters sometimes use a single watt as their reference power level. Their unit of power using decibels is the dBW, read as "dB relative to a watt." In this case, for example, 3 dBW is equal to a 2 watt power level.

Digital Filter Terminology

The first step in becoming familiar with digital filters is to learn to speak the language used in the filter business. Fortunately, the vocabulary of digital filters corresponds very well with the mother tongue used for continuous (analog) filters—so we don't have to unlearn anything that we already know. This appendix is an introduction to the terminology of digital filters.

Allpass filter — an IIR filter whose magnitude response is unity over its entire frequency range, but whose phase response is variable. Allpass filters are typically appended in a cascade arrangement following a standard IIR filter, $H_1(z)$, as shown in Figure F-1.

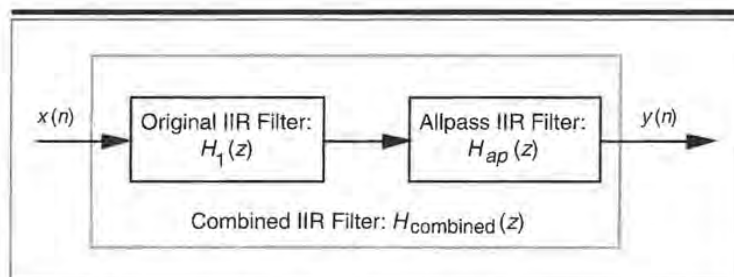


Figure F-1 Typical use of an allpass filter.

An allpass filter, $H_{ap}(z)$, can be designed so that its phase response compensates for, or *equalizes*, the nonlinear phase response of an original IIR filter [1–3]. Thus, the phase response of the combined filter, $H_{combined}(z)$, is more linear than the original $H_1(z)$, and this is particularly desirable in communications systems. In this context, an allpass filter is sometimes called a *phase equalizer*.

Attenuation — an amplitude loss, usually measured in dB, incurred by a signal after passing through a digital filter. Filter attenuation is the ratio, at a given frequency, of the signal amplitude at the output of the filter divided by the signal amplitude at the input of the filter, defined as

$$\text{attenuation} = 20 \cdot \log_{10} \left(\frac{a_{\text{out}}}{a_{\text{in}}} \right) \text{ dB} . \quad (\text{F-1})$$

For a given frequency, if the output amplitude of the filter is smaller than the input amplitude, the ratio in Eq. (F-1) is less than one, and the attenuation is a negative number.

Band Reject Filter — a filter that rejects (attenuates) one frequency band and passes both a lower and a higher frequency band. Figure F-2(a) depicts the frequency response of an ideal band reject filter. This filter type is sometimes called a *notch filter*.

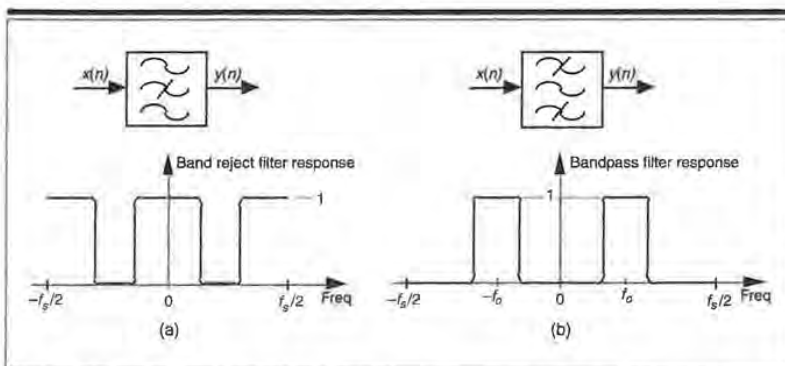


Figure F-2 Filter symbols and frequency responses: (a) band reject filter; (b) bandpass filter.

Bandpass Filter — a filter, as shown in Figure F-2(b), that passes one frequency band and attenuates frequencies above and below that band.

Bandwidth — the frequency width of the passband of a filter. For a low-pass filter, the bandwidth is equal to the cutoff frequency. For a band-

pass filter, the bandwidth is typically defined as the frequency difference between the upper and lower 3 dB points.

Bessel function — a mathematical function used to produce the most linear phase response of all IIR filters with no consideration of the frequency magnitude response. Specifically, filter designs based on Bessel functions have maximally constant group delay.

Butterworth function — a mathematical function used to produce maximally flat filter magnitude responses with no consideration of phase linearity or group delay variations. Filter designs based on a Butterworth function have no amplitude ripple in either the passband or the stopband. Unfortunately, for a given filter order, Butterworth designs have the widest transition region of the most popular filter design functions.

Cascaded filters — a filtering *system* where multiple individual filters are connected in series; that is, the output of one filter drives the input of the following filter as illustrated in Figures F-1 and 6-38(a).

Center Frequency (f_o) — the frequency lying at the midpoint of a bandpass filter. Figure F-2(b) shows the f_o center frequency of a bandpass filter.

Chebyshev function — a mathematical function used to produce passband or stopband ripples constrained within fixed bounds. There are families of Chebyshev functions based on the amount of ripple, such as 1 dB, 2 dB, and 3 dB of ripple. Chebyshev filters can be designed to have a frequency response with ripples in the passband and flat passbands (Chebyshev Type I), or flat passbands and ripples in the stopband (Chebyshev Type II). Chebyshev filters cannot have ripples in both the passband and the stopband. Digital filters based upon Chebyshev functions have steeper transition region roll-off but more nonlinear-phase response characteristics than, say, Butterworth filters.

Cutoff Frequency — the upper passband frequency for low-pass filters, and the lower passband frequency for highpass filters. A cutoff frequency is determined by the -3 dB point of a filter magnitude response relative to a peak passband value. Figure F-3 illustrates the f_c cutoff frequency of a low-pass filter.

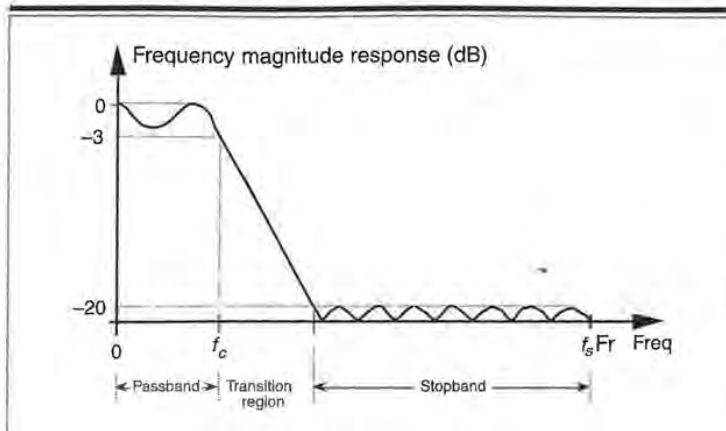


Figure F-3 A low-pass digital filter frequency response. The stopband relative amplitude is -20 dB.

Decibels (dB) — a unit of attenuation, or gain, used to express the relative voltage or power between two signals. For filters, we use decibels to indicate cutoff frequencies (-3 dB) and stopband signal levels (-20 dB) as illustrated in Figure F-3. Appendix E discusses decibels in more detail.

Decimation filter — a low-pass digital FIR filter where the output sample rate is less than the filter's input sample rate. As discussed in Section 7.3, to avoid aliasing problems, the output sample rate must not violate the Nyquist criterion.

Digital filter — computational process, or algorithm, transforming a discrete sequence of numbers (the input) into another discrete sequence of numbers (the output) having a modified frequency-domain spectrum. Digital filtering can be in the form of a software routine operating on data stored in computer memory or can be implemented with dedicated hardware.

Elliptic function — a mathematical function used to produce the sharpest roll-off for a given number of filter taps. However, filters designed by using elliptic functions, also called *Cauer filters*, have the poorest phase linearity of the most common IIR filter design functions. The ripple in the passband and stopband are equal with elliptic filters.

Envelope delay — *see* group delay.

Filter coefficients — the set of constants, also called *tap weights*, used to multiply against delayed signal sample values within a digital filter structure. Digital filter design is an exercise in determining the filter coefficients that will yield the desired filter frequency response. For an FIR filter, by definition, the filter coefficients are the impulse response of the filter.

Filter order — a number describing the highest exponent in the numerator or denominator of the z-domain transfer function of a digital filter. For FIR filters, there is no denominator in the transfer function, and the filter order is merely the number of taps used in the filter structure. For IIR filters, the filter order is equal to the number of delay elements in the filter structure. Generally, the larger the filter order, the better the frequency magnitude response performance of the filter.

Finite impulse response (FIR) filter — a class of digital filters that has only zeros on the z-plane. The key implications of this are that FIR filters are always stable and have linear phase responses (as long as the filter's coefficients are symmetrical). For a given filter order, FIR filters have a much more gradual transition region roll-off than digital IIR filters.

Frequency magnitude response — a frequency-domain description of how a filter interacts with input signals. The frequency magnitude response in Figure F-3 is a curve of filter attenuation (in dB) vs. frequency. Associated with a filter's magnitude response is a phase response.

Group delay — the derivative of a filter's phase with respect to frequency, $G = \Delta\theta/\Delta f$, or the slope of a filter's $H_o(m)$ phase response curve. The concept of group delay deserves additional explanation beyond a simple definition. For an ideal filter, the phase will be linear and the group delay would be constant. Group delay, whose unit of measure is time in seconds, can also be thought of as the propagation time delay of the envelope of an amplitude-modulated signal as it passes through a digital filter. (In this context, group delay is often called *envelope delay*.) Group delay distortion occurs when signals at different frequencies take different amounts of time to pass through a filter. If the group delay is denoted G , then the relationship between group delay, a $\Delta\theta$ increment of phase, and a Δf increment of frequency is

$$G = \frac{\Delta\theta_{\text{degrees}} / 360}{\Delta f} = \frac{\Delta\theta_{\text{radians}} / 2\pi}{\Delta f} \text{ seconds.} \quad (\text{F-2})$$

If we know a linear phase filter's phase shift ($\Delta\theta$) in degrees/Hz, or radians/Hz, we can determine the group delay in seconds using

$$G \cdot \Delta f = G \cdot 1 = G = \frac{\Delta\theta_{\text{degrees/Hz}}}{360} = \frac{\Delta\theta_{\text{radians/Hz}}}{2\pi} \text{ seconds.} \quad (\text{F-3})$$

To demonstrate Eq. (F-3) and illustrate the effect of a nonlinear phase filter, let's assume that we've digitized a continuous waveform comprising four frequency components defined by

$$x(t) = \sin(2\pi \cdot 1 \cdot t) + \sin(2\pi \cdot 3 \cdot t)/3 + \sin(2\pi \cdot 5 \cdot t)/5 + \sin(2\pi \cdot 7 \cdot t)/7. \quad (\text{F-4})$$

The $x(t)$ input comprises the sum of 1-Hz, 3-Hz, 5-Hz, and 7-Hz sinewaves, and its discrete representation is shown in Figure F-4(a). If we applied the discrete sequence representing $x(t)$ to the input of an ideal 4-tap linear-phase low-pass digital FIR filter with a cutoff frequency of greater than 7 Hz, and whose phase shift is 0.25 radians/Hz, the filter's output sequence would be that shown in Figure F-4(b).

Because the filter's phase shift is 0.25 radians/Hz, Eq. (F-3) tells us that the filter's constant group delay G in seconds is

$$G = \frac{\Delta\theta_{\text{radians/Hz}}}{2\pi} = \frac{0.25}{2\pi} = 0.04 \text{ seconds.} \quad (\text{F-5})$$

With a constant group delay of 0.04 seconds, the 1-Hz input sinewave is delayed at the filter output by 0.25 radians, the 3-Hz sinewave is delayed by 0.75 radians, the 5-Hz sinewave by 1.25 radians, and the 7-Hz sinewave by 1.75 radians. Notice how a linear-phase (relative to frequency) filter results in an output that's merely a time shifted version of the input as seen in Figure F-4(b). The amount of time shift is the group delay of 0.04 seconds. Figure F-4(c), on the other hand, shows the distorted output waveform if the filter's phase was nonlinear, for whatever reason, such that the phase shift was 3.5 radians instead of the ideal 1.75 radians at 7 Hz. Notice the distortion of the beginning of the output waveform envelope in Figure F-4(c) compared to Figure F-4(b). The point here is that, if the desired informa-

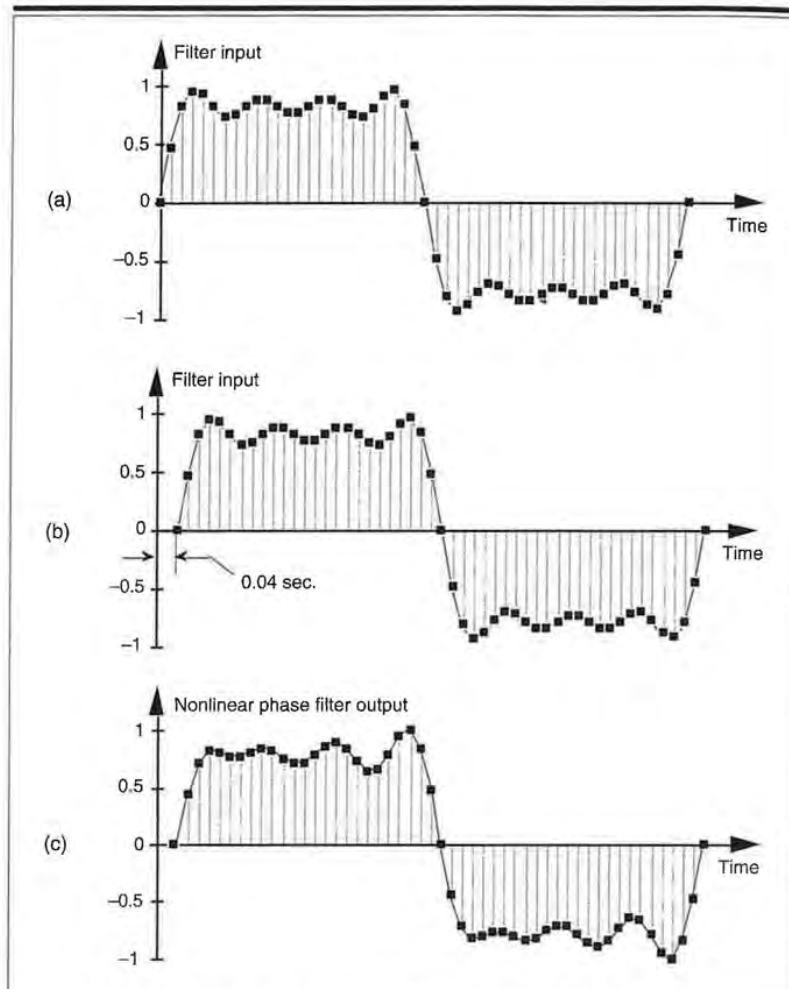


Figure F-4 Filter time-domain response examples: (a) filter input sequence; (b) linear-phase filter output sequence that's time shifted by 0.04 seconds, duplicating the input sequence; (c) distorted output sequence due to a filter with a nonlinear phase.

tion is contained in the envelope of a signal that we're passing through a filter, we'd like that filter's passband phase to be as linear as possible with respect to frequency. In other words, we'd prefer the filter's

group delay to vary as little as possible in the passband. (Additional aspects of nonlinear-phase filters are discussed in Section 5.8.)

Half-Band filter — a type of FIR filter whose transition region is centered at one quarter of the sampling rate, or $f_s/4$. Specifically, the end of the passband and the beginning of the stopband are equally spaced about $f_s/4$. Due to their frequency-domain symmetry, half-band filters are often used in decimation filtering schemes because half of their time-domain coefficients are zero. This reduces the number of necessary filter multiplications, as described in Section 5.7.

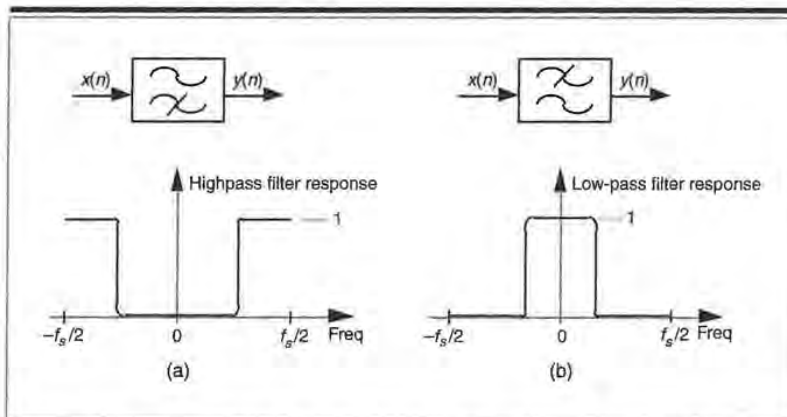


Figure F-5 Filter symbols and frequency responses: (a) highpass filter; (b) low-pass filter.

Highpass filter — a filter that passes high frequencies and attenuates low frequencies, as shown in Figure F-5(a). We've all experienced a kind of highpass filtering in our living rooms. Notice what happens when we turn up the treble control (or turn down the bass control) on our home stereo systems. The audio amplifier's normally flat frequency response changes to a kind of analog highpass filter giving us that sharp and *tinny* sound as the high-frequency components of the music are being accentuated.

Impulse response — a digital filter's time-domain output sequence when the input is a single unity-valued sample (impulse) preceded and followed by zero-valued samples. A digital filter's frequency-domain

response can be calculated by taking the discrete Fourier transform of the filter's time-domain impulse response [4].

Infinite impulse response (IIR) filter — a class of digital filters that may have both zeros and poles on the z -plane. As such, IIR filters are not guaranteed to be stable and almost always have nonlinear phase responses. For a given filter order (number of IIR feedback taps), IIR filters have a much steeper transition region roll-off than digital FIR filters.

Linear-phase filter — a filter that exhibits a constant change in phase angle (degrees) as a function of frequency. The resultant filter phase plot vs. frequency is a straight line. As such, a linear-phase filter's group delay is a constant. To preserve the integrity of their information-carrying signals, linear phase is an important criteria for filters used in communication systems.

Low-pass filter — a filter that passes low frequencies and attenuates high frequencies as shown in Figure F-5(b). By way of example, we experience low-pass filtering when we turn up the bass control (or turn down the treble control) on our home stereo systems giving us that dull, muffled sound as the low-frequency components of the music are being intensified.

Notch filter — *see* band reject filter.

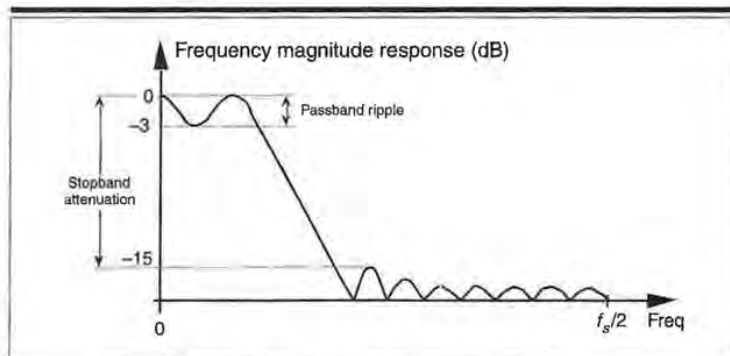


Figure F-6 Low-pass digital filter frequency response showing passband ripple and stopband attenuation.

Passband — the frequency range over which a filter passes signal energy. Usually defined as the frequency range where the filter's frequency response is equal to or greater than -3 dB, as depicted in Figure F-3.

Passband ripple — fluctuations, or variations, in the frequency magnitude response within the passband of a filter. Passband ripple, measured in dB, is illustrated in Figure F-6.

Phase response — the difference in phase, at a particular frequency, between an input sinewave and the output sinewave at that frequency. The phase response, sometimes called *phase delay*, is usually depicted by a curve showing the filter's phase shift vs. frequency. Section 5.8 discusses digital filter phase response in more detail.

Phase wrapping — an artifact of arctangent software routines, used to calculate phase angles, that causes apparent phase discontinuities. When a true phase angle is in the range of -180° to -360° , some software routines automatically convert those angles to their equivalent positive angles in the range of 0° to $+180^\circ$. Section 5.8 illustrates an example of phase wrapping when the phase of an FIR filter is calculated.

Quadrature filter — a dual-path digital filter operating on complex signals, as shown in Figure F-7. One filter operates on the in-phase $i(n)$ data, and the other filter processes the quadrature-phase $q(n)$ signal data. Quadrature filtering is normally performed with low-pass filters.

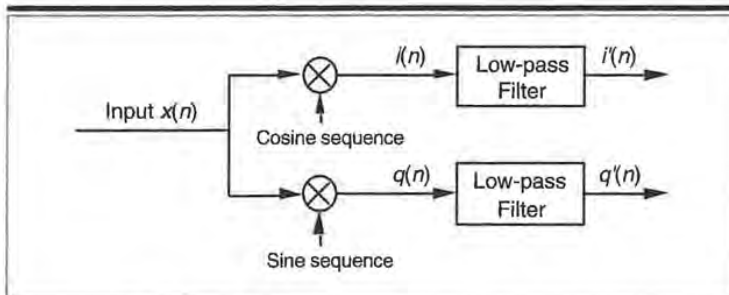


Figure F-7 Two low-pass filters used to implement quadrature filtering.

Relative attenuation — attenuation measured relative to the largest magnitude value. The largest signal level (minimum attenuation) is typically assigned the reference level of zero dB, as depicted in Figures F-2 and F-5, making all other magnitude points on a frequency-response curve negative dB values.

Ripple — refers to fluctuations (measured in dB) in the passband, or stopband, of a filter's frequency-response curve. Elliptic and Chebyshev-based filters have equiripple characteristics in that their ripple is constant across their passbands. Bessel and Butterworth derived filters have no ripple in their passband responses. Ripples in the stopband response are sometimes called *out-of-band ripple*.

Roll-off — a term used to describe the steepness, or slope, of the filter response in the transition region from the passband to the stopband. A particular digital filter may be said to have a roll-off of 12 dB/octave, meaning that the second-octave frequency would be attenuated by 24 dB, and the third-octave frequency would be attenuated by 36 dB, and so on.

Shape factor — a term used to indicate the steepness of a filter's roll-off. Shape factor is normally defined as the ratio of a filter's passband width divided by the passband width plus the transition region width. The smaller the shape factor value, the steeper the filter's roll-off. For an ideal filter with a transition region of zero width, the shape factor is unity. The term *shape factor* is also used to describe analog filters.

Stopband — that band of frequencies attenuated by a digital filter. Figure F-3 shows the stopband of a low-pass filter. Although the stopband attenuation in Figure F-3 is -20 dB, not all filters have stopband lobes of equal amplitude. Figure F-6 shows that stopband attenuation is measured between the peak passband amplitude and the largest stopband lobe amplitude.

Structure — refers to the block diagram showing how a digital filter is implemented. A recursive filter structure is one in which feedback takes place and past filter output samples are used, along with past input samples, in calculating the present filter output. IIR filters are almost always implemented with recursive filter structures. A nonrecursive filter structure is one in which only past input samples are used in calculating the present filter output. FIR filters are almost

always implemented with nonrecursive filter structures. See Chapter 6 for examples of various digital filter structures.

Tap weights — see filter coefficients.

Tchebyshev function — see Chebyshev.

Transfer function — a mathematical expression of the ratio of the output of a digital filter divided by the input of the filter. Given the transfer function, we can determine the filter's frequency magnitude and phase responses.

Transition region — the frequency range over which a filter transitions from the passband to the stopband. Figure F-3 illustrates the transition region of a low-pass filter. The transition region is sometimes called the *transition band*.

Transversal filter — in the field of digital filtering, *transversal filter* is another name for FIR filters implemented with the nonrecursive structures described in Chapter 5.

References

- [1] Rabiner, L. R., and Gold, B. *The Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1975, pp. 206, 273, and 288.
- [2] Oppenheim, A. V., and Schaffer, R. W. *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989, pp. 236 and 441.
- [3] Laakso, T. I., et al. "Splitting the Unit Delay," *IEEE Signal Processing Magazine*, January 1972, pp. 46.
- [4] Pickerd, J. "Impulse-Response Testing Lets a Single Test Do the Work of Thousands," *EDN*, April 27, 1995.

INDEX

A

Absolute value, 9
 A/D converters. *See* Analog-to-digital (A/D) converters
 Addition process, 10
 for complex numbers, 446–47
 in finite impulse response filters, 158–63
 Aliased sinc function, 101
 Aliasing, 23–29
 in infinite impulse response filters, 255–56
 All ones rectangular function, 108–11, 113–15
 Allpass filter, 494
 α Max+ β Min algorithm, 400–404
 Amplitude, defined, 8–9
 Amplitude response, discrete Fourier transform
 complex input, 119–23
 real cosine input, 125
 Analog, use of term, 2
 Analog filters, use of term, 253
 Analog signal processing, examples of, 2
 Analog-to-digital (A/D) converters
 coherent sampling, 297n, 433
 crest factor, 364n
 quantization errors, 357–65
 testing technique, 432–35
 Anti-aliasing filters, 32
 Argand diagram, 444n
 Attenuation
 defined, 495
 relative, 503
 stopband, 189
 Averaging
 applications, 319
 coherent, 320–27
 exponential, 341–48
 filtering aspects of time-domain, 340–41

in finite impulse response filters, 158–63
 incoherent, 327–30
 multiple fast Fourier transforms, 133, 330–40, 429–30

B

Band-limited signal, 29
 Bandpass design, for finite impulse response filters, 191–93
 Bandpass filter, 495
 Bandpass sampling, 32–42
 spectral inversion in, 43–46
 summary of, 46
 Band reject filter, 495
 Bandwidth, 495–96
 Bartlett windows, 82
 Bessel functions, 189, 496
 Bias, 356
 Bilinear transform design, of
 infinite impulse response filters, 223, 272–84
 Binary data formats
 defined, 349
 fixed-point, 349–56
 fixed-point effects on word length, 357–75
 floating-point, 375–82
 fractional, 352–53
 hexadecimal, 351
 octal, 351
 offset, 356
 place value system for, 350
 precision and dynamic range, 356–57
 sign-magnitude, 353
 two's complement, 356, 366–68
 Bits
 defined, 349–50
 effective number of, 365
 Bit reversal, fast Fourier transform
 input/output data index, 145–46

Blackman-Harris windows, 409
 Blackman windows, 184–87, 409
 Block diagrams, use of, 10, 161
 Block floating-point, 381–82
 Boxcar windows, 82
 Butterfly structures, radix-2 FFT, 146–54
 Butterworth function, 496

C

Carrier frequency, 34
 Cascade filters, 290–92, 496
 Cauer filters, 497
 Causal systems, 225
 Center frequency, 496
 Central Limit Theorem, 438–39
 Chebyshev function, 496
 Chebyshev windows, 87, 187–91
 Coefficient quantization, 286–87
 Coefficients, for finite impulse response filters, 166
 constant, 173, 498
 Coherent averaging, 320–27
 Coherent sampling, 297n, 433
 Commutative property, 20
 Complex conjugate, 63–64
 Complex demodulation, 297n
 Complex down-conversion, 297n
 Complex frequency, 225
 for M th-order infinite impulse response filter, 245–46
 Complex input, discrete Fourier transform frequency response to, 119–23
 Complex numbers
 addition and subtraction of, 446–47
 applications of, 453–54
 arithmetic representation of, 444–46
 conjugation of, 427, 448
 division of, 448–49
 fast multiplication of, 411–12
 graphical representation of, 443–44
 inverse of, 449
 logarithm to the base 10 of, 451–52
 multiplication of, 411–12, 447–48
 natural logarithms of, 451
 negative frequency and, 458–74
 raised to a power, 450
 roots of, 450–51

Complex plane, 460–61
 Complex signals
 quadrature mixing and, 471–74
 without multiplication, 385–400
 Conditional stability, 236
 Conjugate/conjugation, 63–64
 of complex numbers, 427, 448
 Constant-coefficient transversal finite impulse response filters, 173
 Constant-geometry algorithm, 154
 Continuous-signal processing
 frequency in, 5–6
 use of term, 2
 Convolution, 20
 discrete, 204–16
 equation, 165
 fast, 435–36
 in finite impulse response filters, 163–74
 theorem, 209–16
 in time-domain, 206–9
 Cooley, J., 129
 Cosines of consecutive angles, calculation of, 436–38
 Crest factor, 364n
 Critical Nyquist, 28
 Cutoff frequency, 496

D

Data formats
 binary precision/dynamic range, 356–57
 defined, 349
 fixed-point binary, 349–56
 floating-point, 375–82
 dB. *See* Decibels
 dBm. *See* Decibels
 Deadband effects, 288
 DEC floating-point, 378
 Decibels
 absolute power using, 493
 defined, 486, 497
 magnitude and power ratios, 492–93
 signal power determined by logarithms, 486–91
 Decimation
 combined with interpolation, 314–16
 filters, 306, 497

- resampling by, 304–9
 - use of term, 304
 - Decimation-in-frequency algorithm, 148–53, 154
 - Decimation-in-time, 145, 149–53
 - Demodulation, I/Q or Weaver, 299–301
 - Difference equation, 221, 222
 - Digital filters. *See* Finite impulse response filters; Infinite impulse response (IIR) filters
 - Digital mixing, quadrature sampling and, 301–3, 389–93
 - Direct Form I structure, 244
 - Direct Form II structure, 252
 - Dirichlet, Peter, 101
 - Dirichlet kernel, 100–103
 - all ones, 108–11, 114
 - symmetrical, 105–8
 - Discrete convolution, 204–16
 - Discrete Fourier transform (DFT), 21
 - defined, 50
 - equation, description of, 50–63
 - example using, 54–63
 - frequency axis, 67–68
 - frequency response to complex input, 119–23
 - frequency response to real cosine input, 123–25
 - inverse, 70–71
 - leakage, 71–80
 - linearity, 65
 - magnitudes, 66–67
 - processing gain, 93–97
 - purpose of, 49
 - rectangular functions, 97–119
 - relationship of fast Fourier transform to, 130–31
 - scalloping loss, 88–89
 - shifting theorem, 68–70
 - single-bin frequency response to real cosine input, 125–27
 - symmetry, 63–65
 - windows, 80–87
 - zero stuffing, 89–93
 - Discrete linear systems, 13–18
 - Discrete-time expression, 4
 - Discrete-time Fourier transform (DTFT), defined, 93
 - Discrete-time signal
 - connecting dots in, problem with, 3
 - example of, 2–3
 - frequency in, 6
 - input and output shown, 5
 - use of term, 2
 - Discrete-time waveform, methods for describing, 8
 - Dither sequence, 288
 - Division of complex numbers, 448–49
 - Dolph-Chebyshev windows, 187
 - Double-memory algorithm, 153–54
 - Down-conversion, complex, 297n
 - Downsampling, resampling by, 304–9
 - Dynamic range
 - fixed-point binary data format, 356–57
 - floating-point, 379–81
- E**
- Effective number of bits (ENOB), 365
 - Elliptic function, 497
 - Envelope delay, 202, 204, 498
 - Euler, Leonhard, 460
 - Euler's equation, 50, 100, 249, 267, 275, 298, 463, 464
 - Even symmetry, 64
 - Exponential averaging, 341–48
- F**
- Fast convolution, 435–36
 - Fast Fourier transform (FFT)
 - averaging multiple, 133, 330–40, 429–30
 - calculating the inverse of, using the forward, 425–29
 - constant-geometry algorithm, 154
 - decimation-in-frequency, 148–53, 154
 - decimation-in-time, 145, 149–53
 - development of, 129
 - double-memory algorithm, 153–54
 - fast finite impulse response filtering using, 435–36
 - hints for using, 131–36
 - in-place algorithm, 153
 - input/output data index bit reversal, 145–46
 - interpreting results, 134–36

- overflow errors, 369–70
- performing two N-point real, 412–20
- performing 2N-point real, 420–25
- radix-2 FFT algorithm, 136–45
- radix-2 FFT butterfly structures, 146–54
- relationship to discrete Fourier transform, 130–31
- software programs, 136
- Fibonacci, 459
- Filtering/ filters
 - See also* Finite impulse response (FIR) filters; Infinite impulse response (IIR) filters
 - allpass, 494
 - aspects of time-domain averaging, 340–41
 - background of, 157
 - bandpass, 495
 - band reject, 495
 - cascade, 290–92, 496
 - Cauer, 497
 - coefficients, 498
 - decimation, 306, 497
 - description of process, 157–58
 - interpolation, 310, 314
 - multirate, 314
 - nonrecursive, 158, 253
 - notch, 495
 - order, 498
 - parallel, 290–92
 - polyphase, 313
 - prototype, 254
 - recursive, 253
 - structure, 161
 - time-varying, 313
 - transversal, 162
- Finite impulse response (FIR) filters
 - averaging with, 158–63
 - bandpass design, 191–93
 - coefficients, 166
 - convolution in, 163–74
 - defined, 498
 - fast Fourier transform used for fast, 435–36
 - 5-tap, 162–63
 - Fourier series design of, 175–91
 - frequency response in, 163, 181
 - half-band, 197–98
 - highpass design, 193–94
 - impulse response, 165–66
 - infinite impulse response filters
 - compared to, 292–93
 - low-pass design, 174–91
 - nonrecursive filters, 158, 253
 - nonzero input values in, 158
 - optimal design method, 194
 - other names for, 158
 - Parks-McClellan design, 194
 - phase response in, 199–204
 - Remez Exchange design, 194–97
 - structure simplified, 430–32
 - window design of, 175–91
- First-order sampling, 35
- 5-tap finite impulse response filters, 162–63
- Fixed-point binary data formats
 - defined, 349
 - effects on word length, 357–75
 - fractional, 352–53
 - hexadecimal, 351
 - octal, 351
 - offset, 356
 - place value system for, 350
 - precision and dynamic range, 356–57
 - sign-magnitude, 353
 - two's complement, 356
- Floating-point binary data format
 - block, 381–82
 - DEC, 378
 - description of, 375–82
 - dynamic range, 379–81
 - fraction of, 376
 - hidden bit, 376
 - IBM, 378
 - IEEE P754, 378
 - MIL-STD 1750A, 378
 - overflow oscillations/limit cycles and, 289
- Folding frequency, 28, 30
- Fourier series design of finite impulse response filters, 175–91
- Fractional binary data format, 352–53
- Frequency

- center, 496
 - cutoff, 496
 - differences between continuous and discrete systems, 5–6
 - Frequency axis, 67–68
 - in Hz, 112
 - normalized angle variable and, 113
 - in radians/seconds, 113
 - rectangular function and, 111–13
 - Frequency domain
 - defined, 6
 - listing of sequences, 8
 - windowing in, 407–9
 - Frequency response, 21
 - discrete Fourier transform, to complex input, 119–23
 - discrete Fourier transform, to real cosine input, 123–25
 - in finite impulse response filters, 163, 181
 - for Mth-order infinite impulse response filter, 245–46
 - for N-stage infinite impulse response filter, 251
 - single-bin, to real cosine input, 125–27
 - Frequency translation
 - sampling and, 33
 - without multiplication, 385–400
- G**
- Gain. *See* Processing gain or loss
 - Gaussian distribution, 325
 - General rectangular function, 98–105
 - Geometric series, 99, 455–57
 - Gibb's phenomenon, 184
 - Gradual underflow, 379
 - Group delay, 201–2, 204, 498–500
- H**
- Half-band finite impulse response filters, 197–98, 501
 - Half Nyquist, 28
 - Hamming windows, 81, 82, 83, 84, 407–9
 - Hanning/Hann windows, 81, 82, 83, 84–86, 407–9
 - Harmonic sampling, 33
 - Heaviside, Oliver, 223
 - Hertz, 3
 - Hexadecimal binary data
 - format, 351
 - Hidden bit, floating-point, 376
 - Highpass design, for finite impulse response filters, 193–94
 - Homogeneity property, 13
- I**
- IBM floating-point, 378
 - IEEE P754, 378
 - IF sampling, 33
 - Imaginary numbers, development of, 460–62
 - Impulse invariance design, for infinite impulse response filters, 223, 254–55
 - Design Method 1 example, 262–65
 - Design Method 1 steps, 256–58
 - Design Method 2 example, 265–72
 - Design Method 2 steps, 259–62
 - Impulse response
 - defined, 501
 - for finite impulse response filters, 165–66
 - Incoherent averaging, 327–30
 - Infinite impulse response (IIR) filters, 26
 - aliasing, 255–56
 - bilinear transform design, 223, 272–84
 - cascade/parallel combinations, 290–92
 - defined, 502
 - Direct Form I structure, 244
 - Direct Form II structure, 252
 - finite impulse response filters
 - compared to, 292–93
 - impulse invariance design, 223, 254–72
 - Laplace transform, 223–38
 - nonzero input values in, 220–21
 - optimized design, 223, 284–86

- pitfalls in building, 286–89
 - reasons for using, 219–20
 - recursive filters, 253
 - structure, 221–23, 251–54
 - z-transform, 238–51
 - In-place algorithm, 153
 - Input/output data index bit reversal, 145–46
 - Integration processing gain, 93, 97, 325, 331
 - coherent, 334–37
 - incoherent, 331–33
 - Intermodulation distortion, 18
 - Interpolation
 - combined with decimation, 314–16
 - filter, 310, 314
 - resampling by, 309–13
 - Inverse discrete Fourier transform (IDFT), 70–71
 - Inverse fast Fourier transform, calculating the, 425–29
 - Inverse of complex numbers, 449
 - Inverse of general rectangular function, 115–17
 - Inverse of symmetrical rectangular function, 117–19
 - I/Q demodulation, 299–301
 - Iterative optimization, 284
- J**
- j operator, 460–62
- K**
- Kaiser windows, 87, 187–91
 - Kelvin, Lord, 50
- L**
- Laplace transform
 - bilateral/two-sided, 225
 - for continuous time-domain, 224–25
 - description of, 223–30
 - development of, 223
 - one-sided/causal, 225
 - tables, 224
 - Laplace transform transfer function, 229–30
 - in bilinear transform design, 272–73, 280
 - in cascade filters, 290
 - in impulse invariance design
 - Method 1, 262–65
 - in impulse invariance design
 - Method 2, 265–66
 - in parallel filters, 290
 - second-order, 233
 - used to determine stability and frequency response in continuous systems, 230–38
- Leakage
 - discrete Fourier transform, 71–80
 - wraparound, 78–79
- Least significant bit (lsb), 350
- L'Hospital's rule, 101
- Limit cycles, 288
- Linear, use of term, 13
- Linear averaging, 320
- Linearity, discrete Fourier transform, 65
- Linear-phase filter, 502
- Linear time-invariant (LTI) systems, 12
 - analyzing, 20–21
 - commutative property, 20
 - discrete, 13–18
 - example of a linear system, 14–15
 - example of a nonlinear system, 15–18
 - homogeneity property, 13
 - time-invariant systems, 18–19
- Loading factor, 361–63
- Logarithms, signal power determined by, 486–91
- Logarithms and complex numbers
 - to base 10, 451–52
 - to base 10 using natural, 452
 - natural, 451
- Loss. *See* Processing gain or loss
- Low-pass finite impulse response filter design, 174–91, 502
- Low-pass signals, sampling, 29–32
- M**
- Magnitude(s)
 - approximation (vector), 400–406
 - of C (modulus of C), 445

- defined, 8, 9
 - discrete Fourier transform, 66–67
 - Mantissa, use of term, 375
 - Mean
 - defined, 319, 477
 - of random functions, 481–84
 - MIL-STD 1750A floating-point, 378
 - Mixing
 - complex signals and quadrature, 471–74
 - complex signals without multiplication, 385–400
 - convolution and, 393–95
 - cosine mixing sequence, 389–91
 - magnitude and phase and, 395
 - quadrature-phase mixing sequence, 391
 - quadrature sampling and digital, 301–3, 389–93
 - summary of, 400
 - Modulus of C , 445
 - Most significant bit (msb), 350
 - M th-order infinite impulse response filter
 - frequency response for, 245–46
 - time domain expression for, 244
 - z -domain expression for, 244–45
 - z -domain transfer expression for, 245
 - Multiplication
 - of complex numbers, 411–12, 447–48
 - frequency translation without, 385–400
 - process, 12
 - Multirate filter, 314
- N**
- Negative frequency, 299, 300
 - complex numbers and, 458–74
 - real signals represented by, 467–71
 - Noise, 319
 - definition of random, 477
 - Nonlinear system, example of, 15–18
 - Nonrecursive filters, 158, 253
 - Normal distribution, 325
 - of random data, 438–40
 - Normalization, 376
 - Normalized angle variable, 113
 - Normal probability density function, 484–85
 - Notch filters, 495
 - N -stage infinite impulse response filter, frequency response for, 251
 - Nyquist criterion, 30, 301
- O**
- Octal binary data format, 351
 - Odd symmetry, 64
 - Offset binary data format, 356
 - One-sided/causal system, 225
 - Optimal design, for finite impulse response filters, 194
 - Optimization design, for infinite impulse response filters, 223, 284–86
 - Out-of-band ripple, 504
 - Overflow errors, 286, 287–88
 - fast Fourier transform, 369–70
 - fixed-point effects on, 365–70
 - rounding, 373–75
 - truncation, 370–73
 - two's complement, 356, 366–68
 - vector-magnitude approximation and, 404–5
 - Overflow oscillations, 288–89
 - Overlapped windows, 410–11
- P**
- Parallel filters, 290–92
 - Parks-McClellan design, for finite impulse response filters, 194
 - Parzen windows, 82
 - Passband, defined, 502
 - Passband phase angle resolution, 202–3
 - Passband ripple, 195, 503
 - Phase angle/argument of C , 445
 - Phase delay, 503
 - Phase equalizer, 494
 - Phase response
 - defined, 503
 - in finite impulse response filters, 199–204
 - Phase wrapping, 201, 503
 - Phasor outputs of fast Fourier transform, 337–38

- Phasors for representing real signals, complex, 462–67
 - Picket fence effect, 88
 - Place value system, 350
 - Polyphase filters, 313
 - Postdetection averaging, 327
 - Power, signal
 - absolute power using decibels, 493
 - defined, 9–10
 - determined by logarithms, 486–91
 - Power spectrum, 54
 - Predetection averaging, 320
 - Prewarp, 283
 - Probability density function (PDF)
 - A/D converter quantization errors and, 360–61
 - mean and variance of random functions and, 481–84
 - normal, 484–85
 - rounding and, 373–74
 - truncation and, 370–71
 - Processing gain or loss
 - discrete Fourier transform, 93–97
 - integration, 93, 97, 325, 331–37
 - window, 83, 409–11
 - Prototype filter, 254
- Q**
- Quadratic factorization formula, 234
 - Quadrature filter, 503
 - Quadrature heterodyning, 297n
 - Quadrature mixing, complex signals and, 471–74
 - Quadrature sampling
 - description of, 297–301
 - digital mixing and, 301–3, 389–93
 - other names for, 297n
 - Quantization, coefficient/errors, 286–87, 292
 - fixed-point effects on A/D converter, 357–65
- R**
- Radian, negative frequency and, 460
 - Radix-2 FFT algorithm, derivation of, 136–45
 - Radix-2 FFT butterfly structures, 146–54
 - Raised to a power, complex numbers, 450
 - Random data, normally distributed, 438–40
 - Random functions, mean and variance of, 481–84
 - Real cosine input, discrete Fourier transform frequency response to, 123–25
 - single-bin, 125–27
 - Real discrete Fourier transform inputs, 63
 - Real numbers, graphical representation of, 443–44
 - Real sampling, 35
 - Real signals
 - complex phasors for representing, 462–67
 - negative frequencies for representing, 467–71
 - Rectangular functions
 - all ones, 108–11, 113–15
 - definition of general, 98–105
 - forms of, 97
 - inverse of general, 115–17
 - inverse of symmetrical, 117–19
 - symmetrical, 105–8
 - time and frequency axes and, 111–13
 - Rectangular windows, 80, 82, 83
 - Recursive filters, 253
 - Relative attenuation, 503
 - Remez Exchange design, for finite impulse response filters, 194–97
 - Replications, spectral, 29–32
 - Resampling, 303–16
 - applications, 303
 - decimation, 304–9
 - decimation combined with interpolation, 314–16
 - defined, 304
 - interpolation, 309–13
 - Ripples
 - Blackman window and, 184–87
 - defined, 504
 - finite impulse response filters and, 173

- low-pass finite impulse response filters and, 179–91
- out-of-band, 504
- passband, 195
- stopband, 195
- rms averaging, 327
 - of a sinewave, 480–81
- Roll-off, 504
- Roots of complex numbers, 450–51
- Roundoff errors, 286, 288
 - fixed-point effects on, 373–75
- Roundoff noise, 359

S

- Sample rate converter, 314
- Sampling
 - coherent, 297n, 433
 - digital resampling, 303–16
 - quadrature, 297–303
- Sampling, periodic
 - aliasing, 23–29
 - bandpass, 32–42
 - low-pass, 29–32
 - spectral inversion in bandpass, 43–46
 - summary of bandpass, 46
 - translation, 33
 - under-, 30–31, 33
- Scalar averaging, 327
- Scalloping loss, 88–89
- Shape factor, 504
- Shark's tooth pattern, 28
- Shifting theorem, discrete Fourier transform, 68–70
- Shift-invariant systems, 19
- Sidelobes, 74, 103
 - Blackman window and, 184–87
 - finite impulse response filters and, 173
 - low-pass finite impulse response filters and, 179–91
- Signal processing
 - analog, 2
 - digital, 2
 - operational symbols, 10–12
 - use of term, 2
- Signal-to-noise ratio (SNR)
 - coherent averaging, 324–25
 - incoherent averaging, 329–30
 - word length and A/D converters, 359–64
- Sign-magnitude binary data format, 353
- Sinc function, 73–74, 108–10, 123
- Sines of consecutive angles, calculation of, 436–38
- Single-bin frequency response to real cosine input, 125–27
- Software programs, fast Fourier transform, 136
- Spectral inversion in bandpass sampling, 43–46
- Spectral replications, 29–32
- Square roots, obtaining, 400
- Stability
 - conditional, 236
 - Laplace transfer function and, 230–38
 - z-transform and, 240–42
- Standard deviation, 320, 479
 - of a sinewave, 480–81
- Starburst pattern, 145
- Statistical measures, 476–79
- Step response, 343
- Stopband, defined, 504
- Stopband attenuation, 189
- Stopband ripple, 195
- Structure
 - defined, 504
 - Direct Form I, 244
 - Direct Form II, 252
 - filtering, 161
 - infinite impulse response filter, 221–23, 251–54
 - simplified finite impulse response filter, 430–32
- Sub-Nyquist sampling, 33
- Subtraction process, 10
 - for complex numbers, 446–47
- Summation process, 10–12, 322–23
- Symbols, in signal processing, 10–12
- Symmetrical rectangular function, 105–8
- Symmetry
 - discrete Fourier transform, 63–65

even, 64
odd, 64

T

Tap weights, 498
Tchebyshev windows, 187
Time delay, z -transform and, 242–45
Time-domain averaging, filtering,
340–41
Time domain expression for M th-order
infinite impulse response filter,
244
Time-domain signals, 5
discrete convolution in, 206–9
Laplace transform for continuous,
224–25
Time-invariant systems
analyzing, 20–21
commutative property, 20
example of, 18–19
Time-varying filters, 313
Transfer function, 229–30
in bilinear transform design, 272–73,
280
in cascade filters, 290
defined, 505
in impulse invariance design
Method 1, 262–65
in impulse invariance design
Method 2, 265–66
in parallel filters, 290
second-order, 233
stability and frequency response in
continuous systems using,
230–38
Transition region/band, 505
Translation without multiplication, fre-
quency, 385–400
Transmultiplexing, 385
Transversal filter, 162, 505
Triangular windows, 80, 82, 83
Truncation errors, 370–73
vector-magnitude approximation
and, 405–6
Tukey, J., 129

Two N -point real FFTs algorithm,
412–20
2 N -point real FFTs algorithm, 420–25
Two's complement, 356, 366–68

U

Underflow, gradual, 379
Undersampling, 30–31, 33
Unit circle, 239
Unit delay, 12
Unit impulse response, 20

V

Variance, 319–20
defined, 478
of random functions, 481–84
Vector, use of term, 444–45
Vector averaging, 320
Vector demodulation, 297 n
Vector-magnitude approximation,
400–406
Video averaging, 327

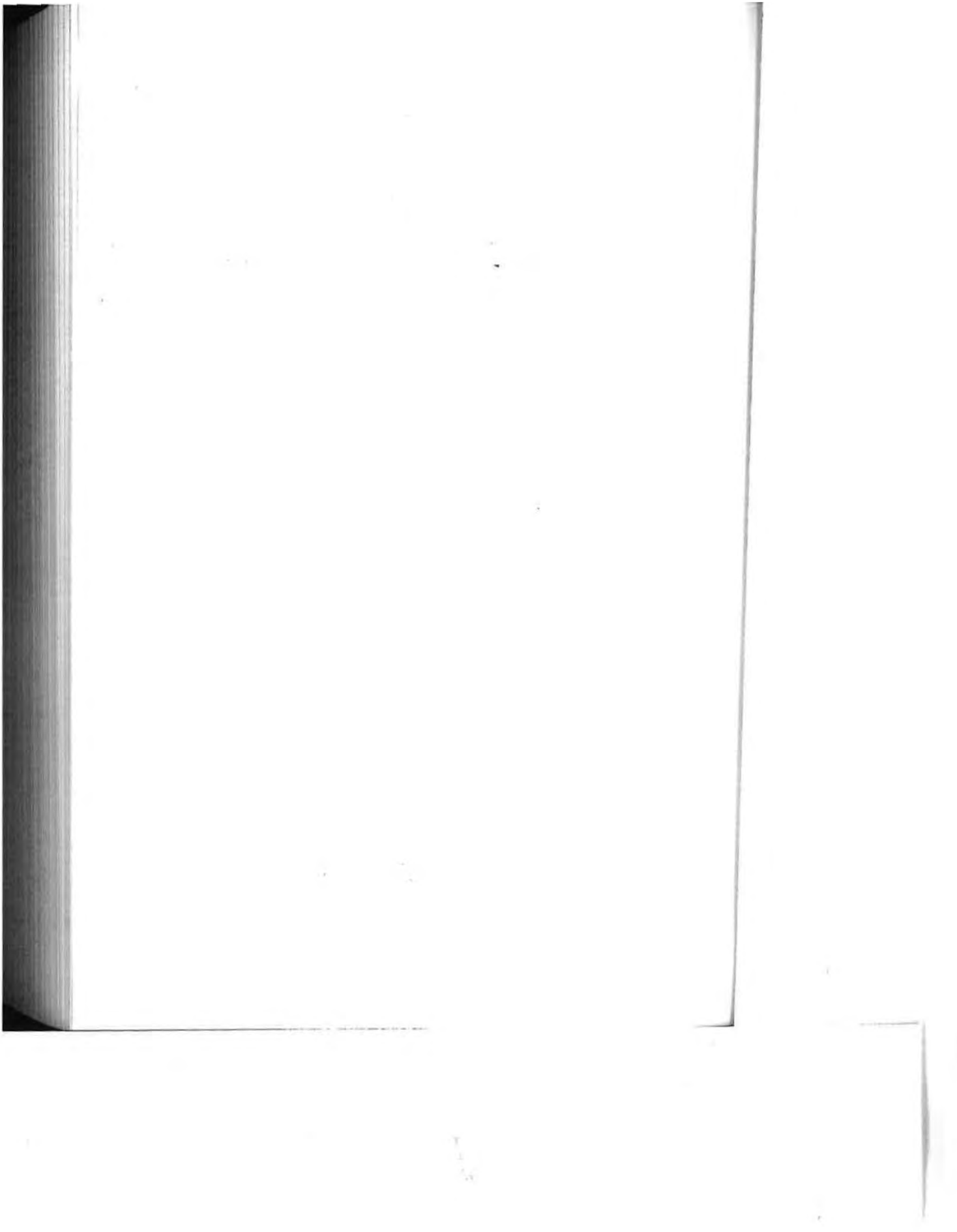
W

Warping, frequency, 276–77, 278
Weaver demodulation, 299–301
Weighting factor, 341, 343
Windows
Bartlett, 82
Bessel functions, 189
Blackman, 184–87, 409
Blackman-Harris, 409
boxcar, 82
Chebyshev, 87, 187–91
discrete Fourier transform, 80–87
Dolph-Chebyshev, 187
dual window functions, 409–10
for finite impulse response filters,
175–91
frequency domain and, 407–9
Hamming, 81, 82, 83, 84, 407–9
Hanning/Hann, 81, 82, 83, 84–86,
407–9

Kaiser, 87, 187–91
 overlapped, 410–11
 Parzen, 82
 processing gain or loss, 83, 409–11
 purpose of, 86
 raised cosine, 82
 rectangular, 80, 82, 83
 selection of, 86
 Tchebyshev, 187
 triangular, 80, 82, 83
 tricks for using, 406–11
 Word length, defined, 350
 Word length, fixed-point effects on
 A/D converter quantization errors,
 357–65
 overflow, 365–70
 rounding, 373–75
 truncation, 370–73
 Word length errors, infinite impulse
 response filters and finite, 286
 Wrapping, phase, 201

Z

z-domain expression for M th-order
 infinite impulse response filter,
 244–45
 z-domain transfer expression for M th-
 order infinite impulse response
 filter, 245
 Zero-frequency IF format, 300
 Zero padding, 340
 Zero stuffing, discrete Fourier trans-
 form, 89–93
 z-transform
 background of, 238
 defined, 238
 description of, 238–40
 infinite impulse response filters and,
 242–51
 polar form, 239
 stability and, 240–42
 time delay, 242–45



This is undoubtedly the most accessible book on digital signal processing (DSP) available to the beginner. Using intuitive explanations and well-chosen examples, this book gives you the tools to develop a fundamental understanding of DSP theory.

The author covers the essential mathematics by explaining the meaning and significance of the key DSP equations. Comprehensive in scope, and gentle in approach, the book will help you achieve a thorough grasp of the basics and move gradually to more sophisticated DSP concepts and applications.

The book begins with a complete explanation of the often misunderstood topic of periodic sampling. The introduction to the important discrete Fourier transform, and its fast Fourier transform (FFT) implementation, is the most lucid and illuminating explanation available anywhere. You will also find extensive information on both finite impulse response (FIR) and infinite impulse response (IIR) digital filters, as well as coverage of the benefits of signal averaging. In addition, the book demystifies the abstruse topics of the Convolution theorem and complex signals. The practical uses of various binary number formats are also carefully described and compared. Finally, a collection of tricks-of-the-trade used by professionals to make DSP algorithms more efficient will help you apply DSP concepts successfully.

Richard G. Lyons is a Systems Engineer with the Systems Integration Group at TRW, Inc. He has been involved with the design and testing of digital signal processing systems for the past fifteen years. He is the author of numerous articles on the topic and is a member of the IEEE and Eta Kappa Nu, the electrical engineering honor society.

Cover design by Amanda L. Peters

Cover art by Mind of the Machine

♻️ Text printed on recycled paper

▲ ADDISON-WESLEY

Addison-Wesley Corporate & Professional
is an imprint of Addison Wesley Longman, Inc.

