PART IV

Signal and Data Processing

CHAPTER

14

Digital Signal Processing Fundamentals for Radar

Mark A. Richards

Chapter Outline

14.1 INTRODUCTION

Radar technology was first seriously developed during the 1930s and through World War II. Radar systems evolved to use a variety of signal processing techniques to extract useful information from raw radar echoes. Examples include moving target indication (MTI), signal integration to improve the signal-to-noise ratio (SNR), pulse compression for high-range resolution, and angle and delay estimation for range and angle tracking.

As discussed in Chapter 13, early radars used analog processors and techniques to implement these functions. However, the ability to implement far more sophisticated algorithms, supported by the exponential improvements in digital processor speed and capacity, have caused digital signal processing (DSP) methods to displace analog processing in virtually all modern radar systems.

There are many excellent references on basic digital signal processing theory; see, for instance, [1,2]. These texts typically base most of their examples on speech and image signals rather than on radar or communications. A good reference for radar-specific signal processing is [3].

While the fundamental principles of DSP apply equally to all of these signals, some of the details important in radar do not arise in speech and images. One of the most fundamental differences is that radar signals, when detected with a coherent (in-phase/quadrature, or I/Q) receiver such as discussed in Chapter 10, are complex-valued; speech and images are real-valued signals. Care must be taken in correctly extending theoretical results such as formulas for correlation or transform symmetry properties to the complex case. Processors must provide mechanisms for complex arithmetic, and the storage per sample is doubled.

Another major difference is that speech and image signals are baseband; that is, they have no carrier. Radar signals are on a carrier frequency and must be demodulated. This fact requires some care in the application of the Nyquist theorem but also opens the door to the digital intermediate frequency (IF) (also called digital I/Q) methods discussed in Chapter 11.

Radar signals also exhibit high dynamic ranges. The dynamic range of the received signal is often on the order of 40 to 80 dB due to the combined effects of large variations in target radar cross section (RCS) and the R^4 variation in received power as a function of range R. Finally, radar signals often suffer low signal-to-interference ratios, falling well below 0 dB for the raw echoes before any processing is done.

Bandwidth, however, may be the most significant difference. Radar signal bandwidths are very large compared with the signals that characterize many other application areas. Telephone-quality speech signals are conventionally limited to about 4 kHz of bandwidth, while music signals exhibit about a 20 kHz bandwidth. In contrast, instantaneous bandwidths of pulsed radars are routinely on the order of hundreds of kilohertz to a few megahertz, similar to video signals. High-range resolution radar systems, however, can have bandwidths from the tens to hundreds of megahertz, with very high-resolution systems occasionally reaching bandwidths in the low gigahertz!

The wide bandwidth of radar signals requires a high sampling rate, as will be seen in Section 14.2.1. This in turn means that a large number of sampled data values, often millions per second, must be processed in real time. Because of this flood of data, radar DSP was originally limited to relatively simple algorithms, requiring only a few operations per data sample. Thus, older radar processors made heavy use of the fast Fourier transform (FFT) and finite impulse response (FIR) filters. Only relatively recently have they adopted correlation methods common in sonar processing or model-based spectral estimation algorithms such as autoregressive or autoregressive moving average methods, which require the solution of sets of linear or nonlinear equations.

14.2 SAMPLING

The "digital signals" of DSP are obtained by discretizing analog signals in two independent ways, as illustrated in Figure 14-1. Discretization in the independent variable, usually time, is called *sampling*. Discretization in amplitude is called *quantization*. Sampling is discussed first; quantization will be discussed in Section 14.3 Whether a signal is sampled is denoted by enclosing its independent variable in parentheses $(\cdot)$ if it is continuous (nonsampled) and in square brackets $[\cdot]$ if it is discrete (sampled). Thus, $x(t)$ represents a continuous signal, whereas $x[n]$ represents a sampled signal. Also, a subscript a will often be used to denote an analog (continuous independent variable) signal.

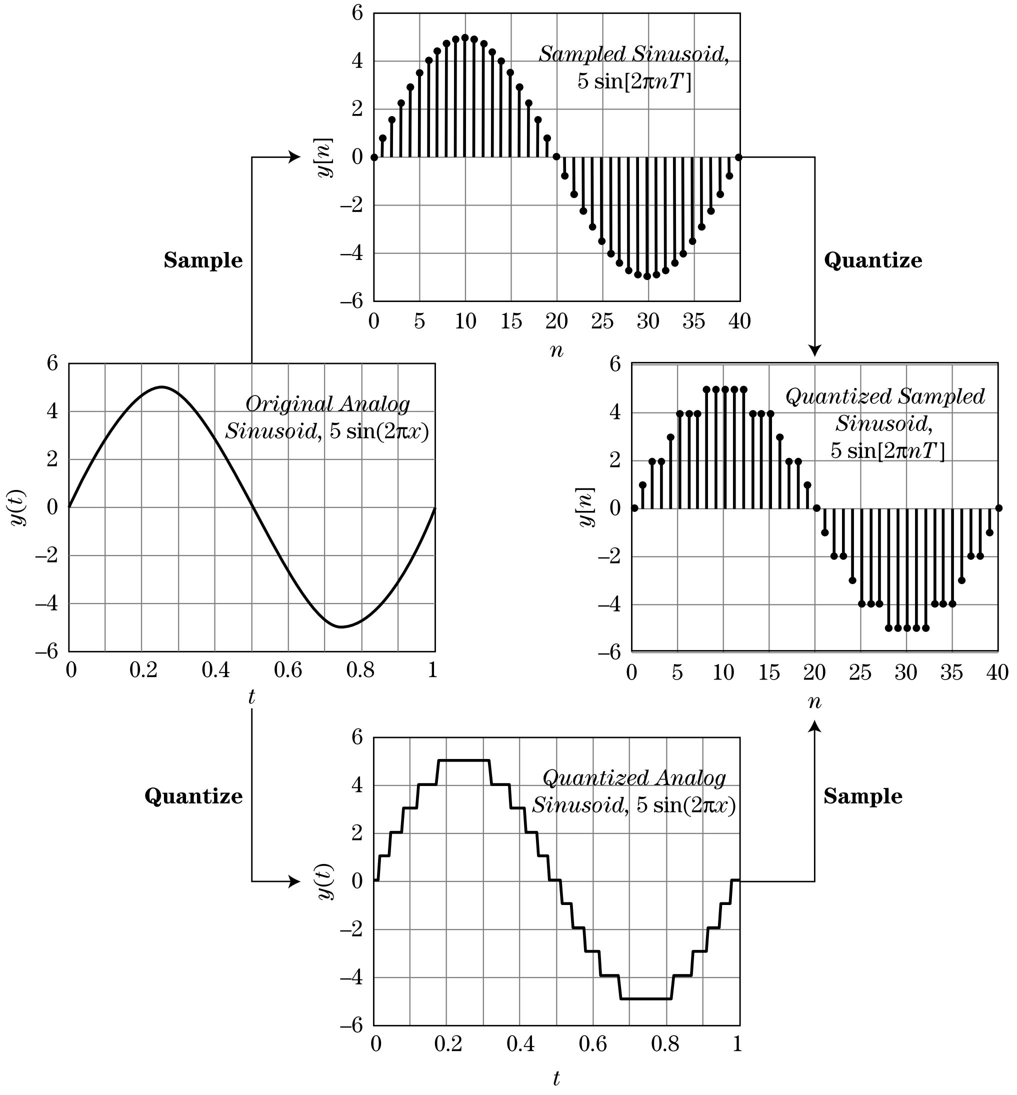


FIGURE 14-1 ■ Illustration of the difference between sampling (discretization in time) and quantization (discretization in amplitude) of an analog time-domain signal.

A sampled signal, $x[n]$, is obtained by taking the value of $x_a(t)$ at multiples of some *sampling interval, T_s*:

$$x[n] = x_a(nT_s), \quad n = -\infty, \ldots, +\infty \tag{14.1}$$

14.2.1 The Nyquist Sampling Theorem

The Nyquist sampling theorem addresses the most important question in sampling: How frequently must samples be taken to adequately represent the analog signal? Consider a signal $x_a(t)$, and assume its Fourier transform $X_a(f)$[1] is band-limited to the

[1] The symbols$f$ and ω are used to represent frequency in cycles per sample (hertz) and radians per second, respectively, for continuous-variable signals. The symbols $\hat{f}$ and $\hat{\omega}$ are used for normalized frequency in cycles per sample and radians per sample for discrete variable signals.

interval $-B/2 \leq f \leq +B/2$ Hz. Now consider the continuous-time signal $x_s(t)$ defined by

$$\begin{aligned} x_s(t) &= x_a(t)\left\{\sum_{n=-\infty}^{+\infty} \delta_D(t - nT_s)\right\} \\ &= \sum_{n=-\infty}^{+\infty} x_a(nT_s)\delta_D(t - nT_s) \\ &= \sum_{n=-\infty}^{+\infty} x[n]\delta_D(t - nT_s) \end{aligned} \tag{14.2}$$

where $\delta_D(\cdot)$ is the Dirac delta (impulse) function. The "sifting property" of the impulse function has been used to obtain the second line in equation (14.2) [4], while the third line was obtained by substituting from equation (14.1). $x_s(t)$ is a sequence of impulse functions occurring at the sample intervals, nT_s, each weighted by a sample of the original analog signal.

Now consider the Fourier transform of $x_s(t)$. An important property of Fourier transforms is that sampling in one domain causes a periodic replication of the signal in the complementary domain. Thus, sampling $x(t)$ in time to obtain $x_s(t)$ (and ultimately $x[n]$) causes a periodic replication of $X_a(f)$. Specifically [1,2],

$$X_s(f) = \frac{1}{T_s}\sum_{k=-\infty}^{+\infty} X_a\left(f - \frac{k}{T_s}\right) = \frac{1}{T_s}\sum_{k=-\infty}^{+\infty} X_a(f - kf_s) \tag{14.3}$$

The spectrum $X_s(f)$ of the sampled signal $x_s(t)$ is thus formed by replicating the original spectrum every $1/T_s = f_s$ Hz. This effect is illustrated in Figure 14-2, which shows an original and sampled signal and the corresponding original and replicated spectra.

The Nyquist criterion follows immediately from the diagram in Figure 14-2 of the replicated spectrum in the lower right. So long as the spectral replicas do not overlap, the original spectrum, and therefore the original signal, can be recovered from the replicated spectrum by an ideal low-pass filter (sinc interpolator) and gain adjustment. This process is shown in Figure 14-3, which also makes clear the necessary conditions on the original spectrum. The replicas will not overlap so long as $x_a(t)$ was band-limited to some finite bandwidth B Hz as shown, and the sampling frequency satisfies

$$f_s = \frac{1}{T_s} > B \tag{14.4}$$

Equation (14.4) is known as the Nyquist criterion; it is the source of the widely quoted statement that a band-limited signal can be recovered from its samples so long as they are taken at a rate equal to "twice the highest frequency" (here, $B/2$ Hz). The minimum sampling rate of B samples/second is called the Nyquist rate. As (14.4) shows, it is equivalent and somewhat more general to restate the Nyquist theorem as requiring that a signal be sampled at a rate equal to or greater than the total width of the original spectrum in hertz.

Most DSP texts present the Nyquist sampling theorem in terms of real-valued, baseband signals, as just described. Coherent radar signals are neither: Before demodulation, they are on a carrier, so the nonzero portion of the spectrum is not centered at 0 Hz,

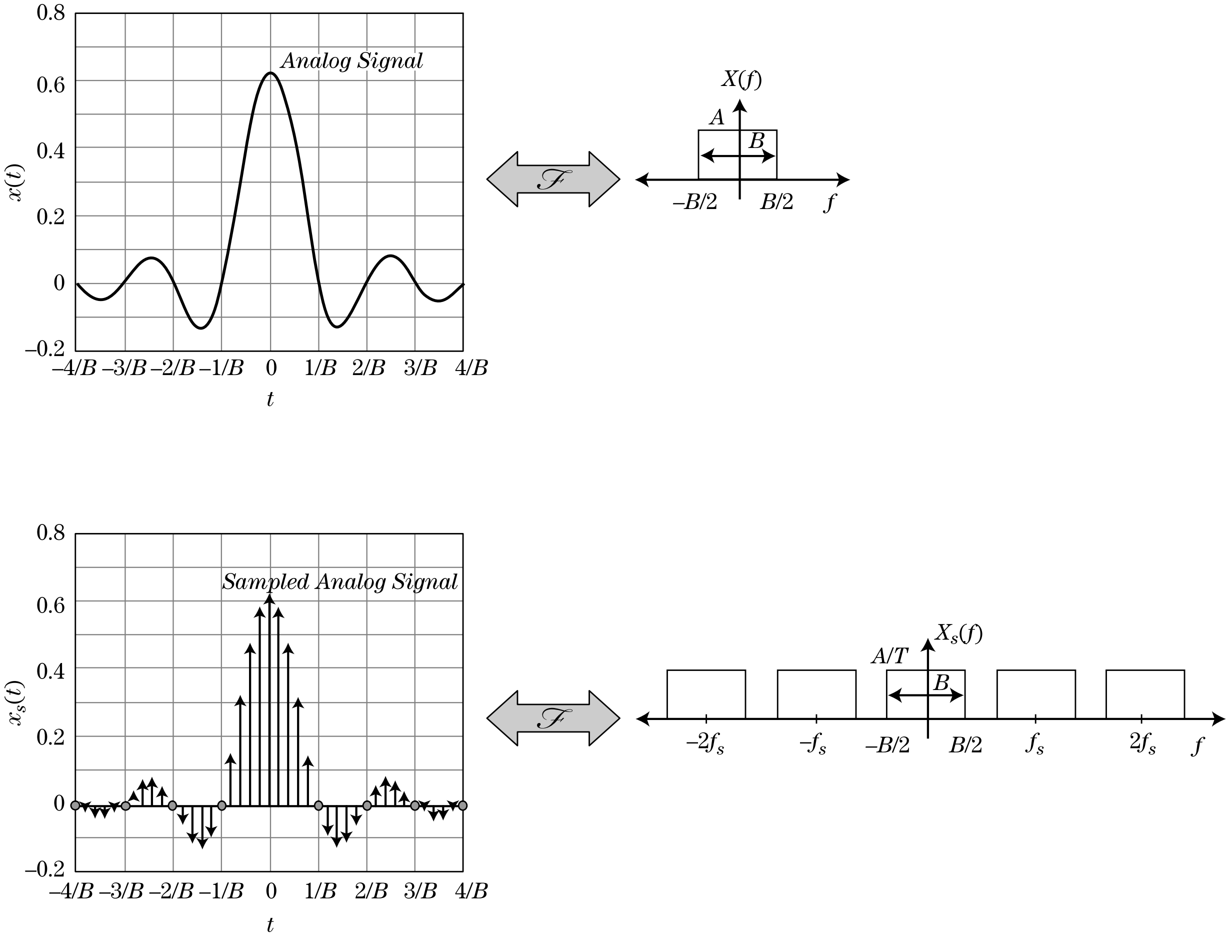


FIGURE 14-2 ■ Effect of sampling on a signal and its spectrum. Top: A band-limited analog signal *x*(*t*) and its Fourier transform *X*(*f*). Bottom: Analog sampled signal $x_s(t)$ and its Fourier transform $X_s(f)$, illustrating replication due to sampling.

and if a coherent receiver is used, the I and Q output signals are combined to form a complex-valued signal. Nonetheless, the Nyquist theorem still applies, because nothing in its derivation relies on the signal being real-valued or having a baseband spectrum. If the signal is complex-valued, this simply means that the samples taken at *B* samples/sec or faster will be complex-valued also. The spectrum replication caused by sampling, which is the basis of the Nyquist theorem (14.3), is a property of Fourier transforms and holds for any signal. Thus, if the original spectrum is still band-limited to a total width of *B* Hz but is centered on a carrier f_0, it is still possible to reconstruct the original signal from samples taken at $f_s = B$ samples/sec or faster. This is illustrated in Figure 14-4, which shows spectrum replication due to sampling a signal having a bandwidth of 40 MHz but not centered at zero Hz. So long as $f_s > 40$ MHz, the replicas shown in the lower part of the figure will not overlap. The difference between this case and the case where the spectrum is centered at zero Hz is that the reconstruction formula will be somewhat different.

This raises another important point. The Nyquist theorem established conditions sufficient to reconstruct an analog signal from its samples. In radar DSP, the analog signal

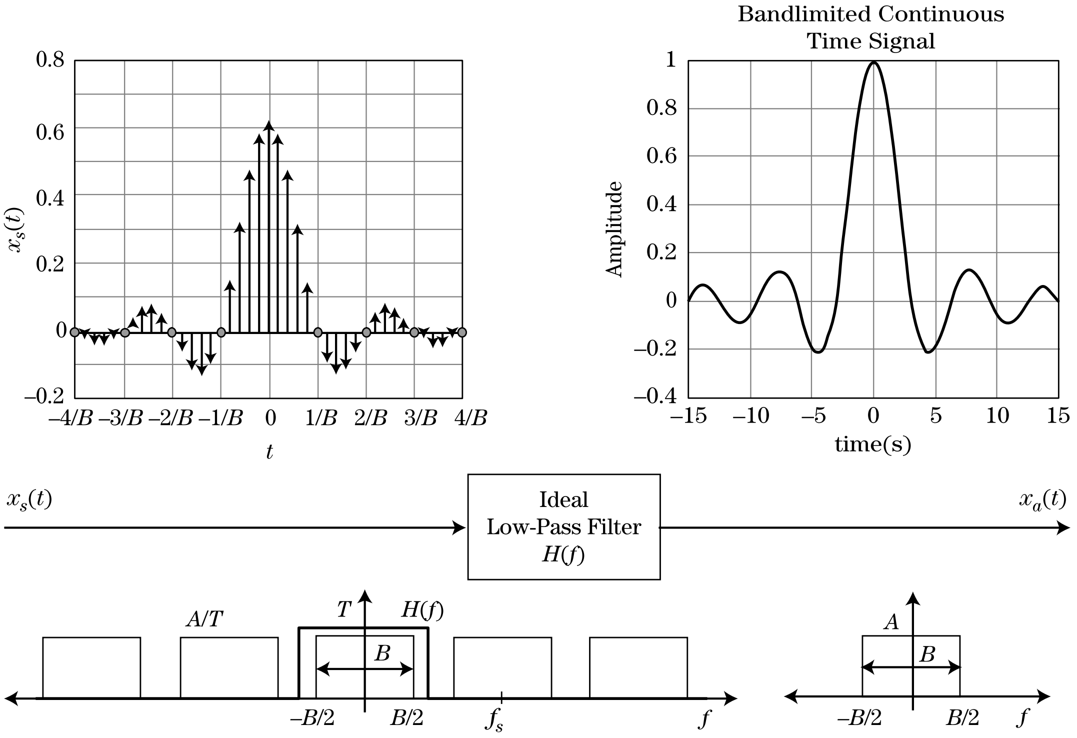


FIGURE 14-3 ■ Reconstruction of analog signal from its sampled equivalent.

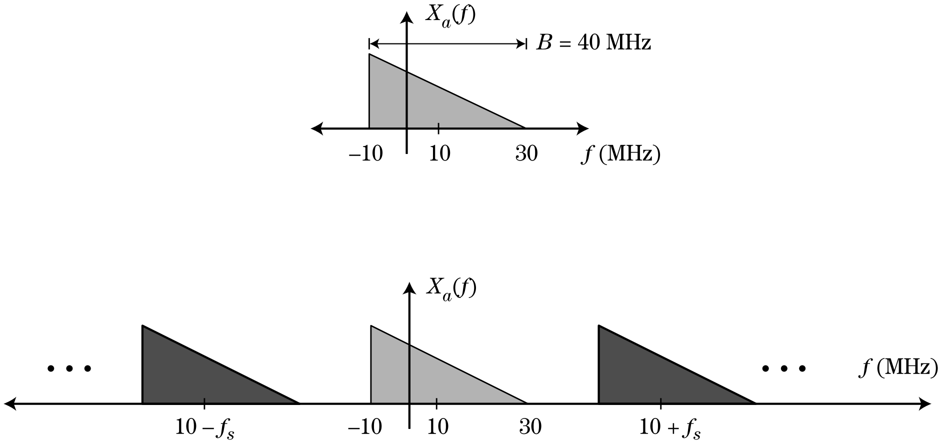


FIGURE 14-4 ■ Effect of sampling on spectrum of a complex, non-baseband signal.

is very rarely, if ever, reconstructed from its samples. Rather, the samples are processed to obtain other information, such as detection reports, track data, or radar images. One could thus ask if the Nyquist criterion is appropriate for radar. In fact, other sampling rules based on other criteria can be developed, such as limiting maximum straddle loss (see Section 14.4.5). However, the ability to reconstruct the original signal implies that all of the information in the signal has been retained, and, for this reason, the Nyquist criterion is generally used to establish required sampling rates.

Finally, note that the Nyquist sampling theorem can be applied in domains other than the time domain. Although the details are not given here, the theorem can be used

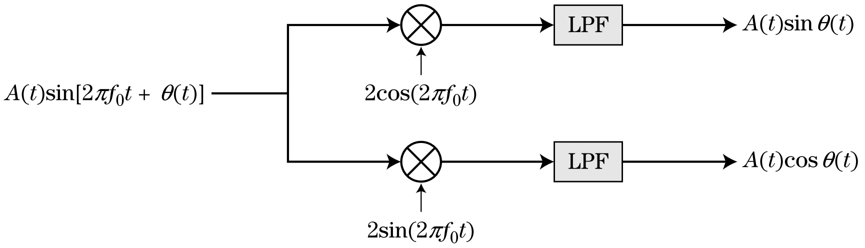


FIGURE 14-5 ■ Coherent (I and Q) receiver.

to establish the minimum-size discrete Fourier transform (DFT) required to represent a finite-length sequence or to determine the density of spatial samples needed in designing an array antenna or an imaging radar.

14.2.2 Sampling Nonbaseband Signals

Though it is possible to sample a signal while it is still on a carrier, most conventional systems use a coherent receiver such as the one shown in Figure 14-5 to move the signal energy to baseband prior to sampling. The coherent receiver was discussed in Chapter 11. If the spectrum of the baseband pulse $A(t)$ has a two-sided bandwidth of B Hz then the two output signals from the coherent receiver will each be a real-valued baseband signal with a frequency spectrum occupying the range $(-B/2, +B/2)$ Hz. Each can then be conventionally sampled at its Nyquist rate of $f_s > B$ samples/second.

This coherent receiver approach has been very successful in many fielded systems. Nonetheless, it has limitations. The block diagram of Figure 14-5 implies some stringent requirements on the receiver hardware. For example, the two reference oscillators must be exactly 90° out of phase, while the gain and delay through each of the two output channels must be identical across the signal frequency band.[2] This can be difficult to achieve to adequate precision in high-quality systems and has resulted in increasing interest in so-called digital IF or digital I/Q receivers. In these receivers, the signal is demodulated to an IF, but not all the way to baseband, with analog hardware but is then sampled while still on the IF carrier, with the remaining baseband conversion completed digitally. Additional detail on both conventional and digital I/Q coherent receivers are given in Chapter 11.

14.2.3 Vector Representation of Sampled Signals

It is sometimes convenient to represent the sampled signal $x[n]$ as a vector. Suppose $x[n]$ is a finite-length N-sample signal, that is, $x[n] = 0$ for $n < 0$ and $n > N - 1$. Form the N-point column vector

$$\boldsymbol{X} = [x[0] \;\; x[1] \;\; \cdots \;\; x[N-1]]^T \tag{14.5}$$

where the superscript T represents matrix transpose. Uppercase italic boldface variables such as $\boldsymbol{X}$ represent vectors, while uppercase non-italic boldface variables such as $\mathbf{R}$ represent matrices. This vector representation will be used in Section 14.10.

[2] The effects of imperfect gain and phase matching are discussed in Chapter 11.

14.2.4 Data Collection and the Radar Datacube

Advanced radar signal processing often operates on data that are sampled in more than one dimension. Consider a pulsed radar. For each pulse transmitted, a series of complex (I and Q) samples of the echo corresponding to successive range intervals will be collected at the output of the receiver. By the Nyquist criterion, these range samples are collected at a rate equal to the pulse bandwidth or greater; this dimension is often referred to as *fast time*. Range samples are also referred to as *range gates* or *range bins*.

The times of the first and last range samples are determined by the desired starting and stopping ranges R_1 and R_2 and are $2R_1/c$ and $2R_2/c$, respectively.[3] The range extent of the samples, known as the range *swath*, is $R_2 - R_1$, and the number of range samples, L, is the swath time divided by the fast time sampling interval, $2(R_1 - R_2)/cT_s$. Assuming a coherent receiver is used, each sample is a single complex number.

R_1 and R_2 are chosen based on a variety of system and mission considerations, but there are limitations on both. The largest R_2 can be is $R_{max} = cT/2$, where T is the radar's pulse repetition interval (PRI). This case corresponds to continuing to collect samples from one pulse until the next one is transmitted. In monostatic radars, the receiver is turned off during the first τ seconds of the PRI (the time during which the pulse is being transmitted) so that the high transmit power does not leak into and damage the receiver. For these radars, the minimum range is $R_{min} = c\tau/2$ m. The echo from targets at range of R_{min} or less is partially or completely ignored and is said to be *eclipsed*. The maximum swath length is then $c(T - \tau)/2$ m.

This process is repeated for each of M pulses in a *coherent processing interval* (CPI; also sometimes called a *dwell* or simply a pulse burst or group), forming a two-dimensional range–pulse number matrix of sampled data, as shown in Figure 14-6. The pulse number dimension is often called *slow time* because the sampling interval in that dimension, which is the radar's pulse repetition interval T, is much greater than the sampling interval in fast time (range). Note that the slow-time signal formed by examining values from pulse to pulse for a fixed range cell, as illustrated by the shaded row, is also a time series.

Now suppose the radar has multiple receiver channels. Multiple channels most often result from having an antenna with multiple phase centers, such as a monopulse antenna

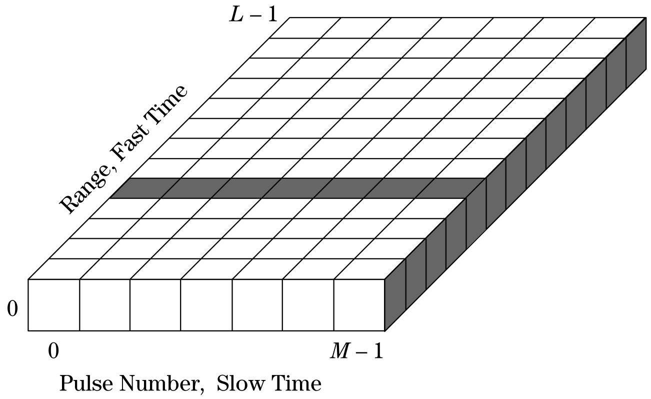


FIGURE 14-6 ■ Fast-time–slow-time matrix of pulse radar data.

[3] The echo signal is passed through a matched filter in many radars, as discussed in Chapter 20. In this case, the minimum and maximum range sampling times are increased by the delay through the matched filter.

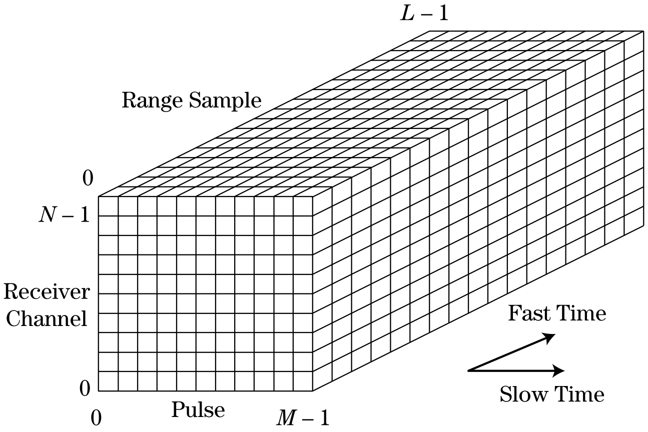


FIGURE 14-7 ■ The radar datacube.

with sum and difference channels or a phased array antenna with a number of subarrays (see Chapter 9). After demodulation and sampling, the data collected over the CPI can be represented by the radar *datacube* shown in Figure 14-7.

Various digital processing algorithms correspond to operating on the datacube in various dimensions. Figure 14-8 illustrates many of the major processing types. For example, pulse compression (see Chapter 20) is linear filtering (or correlation) in the fast-time dimension and can be performed independently for each pulse and receiver channel. Similarly, Doppler processing (MTI or pulse-Doppler; see Chapter 17) operates across multiple samples for a fixed range bin (i.e., in the slow-time dimension) and can be performed independently on each range bin and receiver channel. Synthetic aperture imaging (Chapter 21) is an example of a process that operates on two dimensions of the datacube, in this case both slow and fast time. This view of the datacube is sometimes helpful in thinking about the relationship between differing processing techniques.

Not all radar systems require real-time processing of the data, but, in those that do, any pulse-by-pulse fast-time processing such as pulse compression must be completed in T seconds or less so the processing of the samples of one pulse is completed before the samples of the next pulse arrive. Operations such as beamforming must be completed for each range bin in one PRI or less. Similarly, CPI-level processing such as space-time adaptive processing (STAP) must be completed before the next CPI of data is received, and slow-time processes such as pulse-Doppler processing must be completed for each range bin in one CPI.

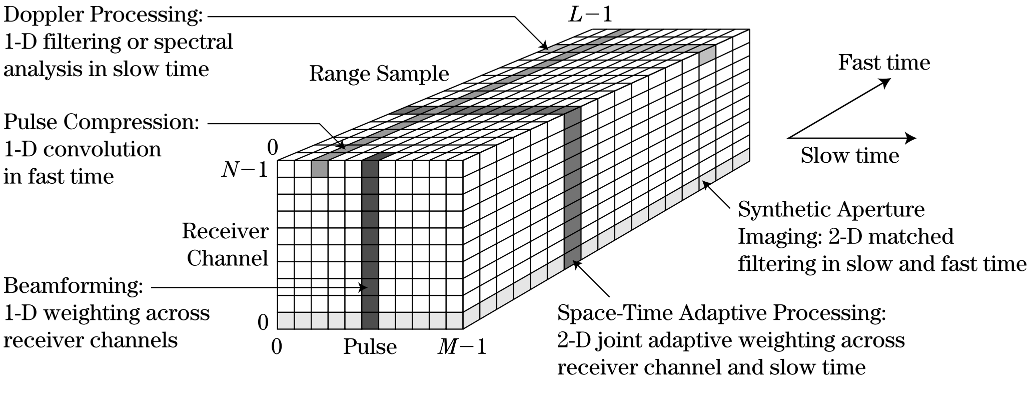

Doppler Processing: 1-D filtering or spectral analysis in slow time
L−1
Range Sample
Fast time
Slow time
Pulse Compression: 1-D convolution in fast time
0
N−1
Receiver Channel
Synthetic Aperture Imaging: 2-D matched filtering in slow and fast time
Beamforming: 1-D weighting across receiver channels
0
0
Pulse
M−1
Space-Time Adaptive Processing: 2-D joint adaptive weighting across receiver channel and slow time

FIGURE 14-8 ■ Datacube showing the dimensions on which various radar signal processing algorithms act.

14.3 QUANTIZATION

The discretization of analog signals in amplitude is called *quantization*. Quantization is necessary so that the amplitude of each sample can be represented with a finite number of bits in a computer or digital hardware.

Binary representations of numeric data can generally be categorized as either fixed point or floating point.[4] In fixed-point representations, the b bits of a binary word are used to represent 2^b distinct and evenly spaced numerical values. In a floating-point representation, some number e of the b total bits is used to represent an exponent, and the remaining $m = b-e$ bits are used to represent the mantissa. Here, fixed-point representations are of primary interest, as this is the representation output by most analog-to-digital converters (ADCs).

The signal voltage value corresponding to each of the 2^b possible binary words in a fixed-point encoding is determined by the arithmetic coding scheme that is used and the quantization step size, Δ. The step size is the change in input value in volts for each increment or decrement of the binary word value. Thus, the value of the data sample is simply Δ times the binary number.

The two most common encodings are called *sign-magnitude* encoding and *two's complement* encoding. In sign-magnitude encoding, the most significant bit of the binary word represents the sign of the data sample; usually a value of zero represents a positive number, while a value of binary one represents a negative number. The remaining $b - 1$ bits encode the magnitude of the sample. Thus, sign-magnitude encoding can represent numbers from $(-2^{b-1} - 1)\Delta$ to $(+2^{b-1} - 1)\Delta$. Note that there are two codes for the value zero, corresponding to $+0$ and -0.

Two's complement is a somewhat more complex encoding that has advantages for the design of digital arithmetic logic. Details are given in [5]. For the present purpose, it is sufficient to note that there is only one code for zero. The extra code value allows the representation of one more negative number, so the range of values becomes $-2^{b-1}\Delta$ to $(+2^{b-1}-1)\Delta$. For any reasonable number of bits b, the difference in range is not significant.

The choice of the number of bits and quantization step size govern the trade-off between the dynamic range and quantization error of the digital signal. Dynamic range is the ratio of the largest representable magnitude to the smallest nonzero magnitude; for the two's complement case, this is

$$DR = \frac{2^{b-1}\Delta}{\Delta} = 2^{b-1} \tag{14.6}$$

For the sign-magnitude case, $DR = 2^{b-1} - 1 \approx 2^{b-1}$ for b more than just a few bits. Expressed in dB, equation (14.6) becomes

$$\begin{aligned} DR\,(\text{dB}) &= 20\log_{10}(2^{b-1}) \\ &= (b-1)20\log_{10}(2) \\ &= 6.02b - 6.02 \ \text{ dB} \end{aligned} \tag{14.7}$$

Equation (14.7) shows that the dynamic range that can be represented at the ADC output without saturation (overflow) increases by 6 dB per bit.

[4] Additional variations, such as block floating point, are also used but are beyond the scope of this chapter.

Conversion from analog, with its infinite allowed amplitude values, to a *b*-bit digital word with its finite number 2^b of possible values, necessarily entails rounding or truncating the analog sample value to one of the allowed quantized values. The difference between the unquantized and quantized samples is the *quantization error*. Although it is in fact a deterministic function of the input data and ADC parameters, the behavior of the quantization error signal is usually complex enough that it is treated as a random variable that is uncorrelated from one sample to the next. Thus, the sequence of quantization errors is modeled as a white random process called the *quantization noise*. Quantization noise is an independent noise signal that adds to the receiver noise already present in the raw analog data. The quantization error for each sample can vary between $\pm\Delta/2$, and it is commonly assumed that errors anywhere in this range are equally likely. Thus, the noise process is modeled as a uniform random process over this range so that the quantization noise power is $\Delta^2/12$ [2].

Assume that the quantizer is calibrated to cover a range of $\pm A_{sat}$ without saturation. Then $\Delta = A_{sat}/2^{b-1}$. Assume also that the input signal to the ADC is modeled as a random process with some power σ^2. This could model the receiver thermal noise, for instance, which is the minimum signal expected at the ADC input. The signal-to-quantization noise ratio (SQNR) is then

$$\begin{aligned} SQNR\,(\text{dB}) &= 10\log_{10}\left(\frac{\sigma^2}{\Delta^2/12}\right) = 10\log_{10}\left(\frac{2^{2b-2}12\sigma^2}{A_{sat}^2}\right) = 10\log_{10}\left(\frac{2^{2b}3\sigma^2}{A_{sat}^2}\right) \\ &= 6.02b - 10\log_{10}\left(\frac{A_{sat}^2}{3\sigma^2}\right) \end{aligned} \tag{14.8}$$

For a given signal power level, the SQNR will improve by 6 dB per bit, similar to the dynamic range.

There remains the issue of how to set the quantizer step size Δ relative to the signal level. If Δ is too small, large input signal values will exceed the range of the ADC and will be clipped to a value of $\pm A_{sat}$, a situation called *saturation*. If Δ is made larger to avoid saturation, then small signal variations may not cause the ADC to count up or down; very small signals may produce only a constant output of zero, thus being suppressed entirely in a condition called *underflow*. In either case, the SQNR will not follow the 6 dB per bit rule. McClellan and Purdy [6] modeled this behavior by deriving the probability density function of the ADC output using the previously given model. Ideally, the output noise level should be the same as the input so that the ADC neither adds nor detracts from the input signal power.

The results of the analysis in [6] are shown in Figure 14-9 as a function of the step size Δ, normalized to the input noise signal standard deviation. This figure suggests that, when the input is the receiver noise, a normalized step size of between ±6 dB, meaning that Δ is between one-half and two times the quiescent noise standard deviation at the ADC input, will result in no more than 1 dB of additional quantization noise at the ADC output, while avoiding saturation or underflow effects. Many fielded systems set Δ in the lower end of this range, effectively devoting one or two bits to representing the receiver noise. Note that a normalized Δ of greater than about 12 dB (four times the noise standard deviation) results in a rapidly falling output power due to underflow because the input signal fails to exceed the relatively large ADC step size. As a result, the output is mostly zero. For a small number of bits, step sizes less than one-half the noise standard deviation result in an output

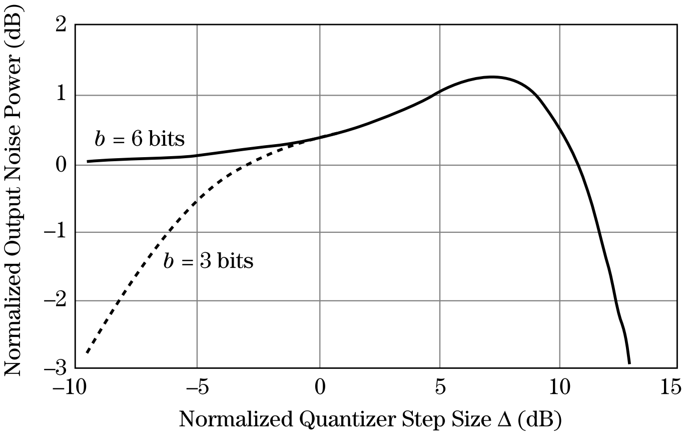


FIGURE 14-9 ■ Ratio of output power to input power for an ADC with a noise signal at its input.

power less than the input power due to saturation, since the small number of ADC levels and the small step size severely limit the range of input signals that can be represented.

Once the data are quantized, a choice must be made between implementing the processing in fixed- or floating-point arithmetic. Generally, floating-point arithmetic requires more digital logic to implement and is therefore slower. However, mathematical algorithms are easier to develop for floating-point arithmetic because numerical overflow is much less likely. Historically, most radar digital processors have relied on fixed-point arithmetic, at least in the early processing stages that tend to be more computationally intensive, because it is faster. Increasing numbers of modern systems are using floating-point processing because the dramatic increases in processor power have made it still possible to implement the desired algorithms in real time.

14.4 FOURIER ANALYSIS

Just as in other signal processing application areas, Fourier analysis is instrumental to radar signal processing, both for analysis and for actual algorithm implementation. In this section, the two major forms of Fourier transforms applicable to sampled signals are reviewed. Their relation to each other and to the Fourier transform of the analog signal is illustrated. Finally, important properties of the Fourier transform of some key signals are illustrated.

14.4.1 The Discrete-Time Fourier Transform

The *discrete-time Fourier transform* (DTFT) of a discrete signal $x[n]$ is defined as

$$X(\hat{\omega}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\hat{\omega}n}, \quad \hat{\omega} \in (-\infty, +\infty)$$

or

$$X(\hat{f}) = \sum_{n=-\infty}^{+\infty} x[n]e^{-j2\pi\hat{f}n}, \quad \hat{f} \in (-\infty, +\infty) \tag{14.9}$$

where the frequency variable is a normalized frequency, $\hat{\omega}$, in rad/sample (not rad/s) or $\hat{f}$, in cycles/sample (not cycles/s, or Hz). It will be seen momentarily why the frequency variable is considered to be normalized. The transform of equation (14.9) will be called the

DTFT even when the independent variable is not time. Note that $X(\hat{\omega})$ is periodic with a period of 2π in $\hat{\omega}$ or a period of 1 in $\hat{f}$. Thus, $X(\hat{\omega})$ is normally examined only in an interval of width 2π, usually $(-\pi, +\pi)$ or $(0, 2\pi)$. When working in units of cycles/sample, $X(\hat{f})$ is typically examined in the interval of $(-0.5, +0.5)$ or $(0, 1)$. The inverse transform is

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\hat{\omega})e^{+j\hat{\omega}n}d\hat{\omega}, \quad n \in (-\infty, +\infty)$$

or

$$x[n] = \int_{-0.5}^{0.5} X(\hat{f})e^{+j2\pi\hat{f}n}d\hat{f}, \quad n \in (-\infty, +\infty) \tag{14.10}$$

Even though the signal $x[n]$ has a discrete independent variable, it is important to realize that the frequency variable of the DTFT ($\hat{\omega}$ or $\hat{f}$) is continuous: the DTFT is defined for all values of frequency. This fact will prove important in understanding the behavior of the discrete Fourier transform, or DFT, in Section 14.4.4.

Consider a discrete signal $x[n]$ obtained by sampling an analog signal $x_a(t)$ at intervals of T_s seconds. To relate the DTFT of $x[n]$ to the Fourier transform of $x_a(t)$, consider the Fourier transform of the sampled signal $x_s(t)$ from (14.2):

$$\begin{aligned} X_s(f) &= \int_{-\infty}^{+\infty} x_s(t)e^{-j2\pi ft}df \\ &= \sum_{n=-\infty}^{+\infty} x[n]\left\{\int_{-\infty}^{+\infty} \delta_D(t-nT_s)e^{-j2\pi ft}df\right\} \\ &= \sum_{n=-\infty}^{+\infty} x[n]e^{-j2\pi fnT_s} \end{aligned} \tag{14.11}$$

Comparing the last line of (14.11) to the definition of the DTFT $X(\hat{f})$ in (14.9) shows that $X(\hat{f}) = X_s(f)$ when $\hat{f} = fT_s = f/f_s$. Using (14.3) then gives the desired relation between the DTFT and the original analog signal spectrum:

$$X(\hat{f}) = \frac{1}{T_s}\sum_{k=-\infty}^{+\infty} X_a(\hat{f}\cdot f_s - kf_s) \tag{14.12}$$

Equation (14.12) shows that the periodic DTFT $X(\hat{f})$ is the replicated (due to sampling) and rescaled (in amplitude and frequency) analog spectrum $X_a(f)$. Features that appear in the original spectrum at some frequency f_0 will appear in the DTFT at the normalized frequency $\hat{f}_0 = f_0/f_s$.

The relation $\hat{f} = f/f_s$ (and equivalently, $\hat{\omega} = \omega/f_s = \omega T_s$) is the reason the DTFT frequency variable $\hat{f}$ is called *normalized* frequency: it is the original analog frequency normalized to the sampling frequency $f_s = 1/T_s$. Since the DTFT is periodic in $\hat{f}$ with a period of 1, it follows that one period of the DTFT is equivalent to f_s Hz.

A particularly important yet simple example is the DTFT of a complex sinusoid, a signal that arises in many areas of radar signal processing. Choose

$$x[n] = Ae^{j2\pi\hat{f}_0 n}, \quad n = 0, \ldots, N-1 \tag{14.13}$$

The DTFT of $x[n]$ is then

$$X\left(\hat{f}\right) = A\sum_{n=0}^{N-1} e^{-j2\pi(\hat{f}-\hat{f}_0)n} = A\frac{1-e^{-j2\pi(\hat{f}-\hat{f}_0)N}}{1-e^{-j2\pi(\hat{f}-\hat{f}_0)}} \tag{14.14}$$

where the last step was obtained by applying the very useful geometric sum formula

$$\sum_{n=N_1}^{N_2} \alpha^n = \frac{\alpha^{N_1} - \alpha^{N_2+1}}{1-\alpha} \tag{14.15}$$

with $\alpha = \exp(-j2\pi(\hat{f} - \hat{f}_0))$, $N_1 = 0$, and $N_2 = N - 1$. Equation (14.14) can be put in a more useful form as follows:

$$\begin{aligned} X(f) &= A\frac{1-e^{-j2\pi(\hat{f}-\hat{f}_0)N}}{1-e^{-j2\pi(\hat{f}-\hat{f}_0)}} = A\frac{e^{-j\pi(\hat{f}-\hat{f}_0)N}(e^{+j\pi(\hat{f}-\hat{f}_0)N} - e^{-j\pi(\hat{f}-\hat{f}_0)N})}{e^{-j\pi(\hat{f}-\hat{f}_0)}(e^{+j\pi(\hat{f}-\hat{f}_0)} - e^{-j\pi(\hat{f}-\hat{f}_0)})} \\ &= Ae^{-j\pi(\hat{f}-\hat{f}_0)(N-1)}\frac{\sin(\pi(\hat{f}-\hat{f}_0)N)}{\sin(\pi(\hat{f}-\hat{f}_0))} \\ &\equiv NAe^{-j\pi(\hat{f}-\hat{f}_0)(N-1)}\operatorname{asinc}(\hat{f}-\hat{f}_0, N) \end{aligned} \tag{14.16}$$

The last line defines the "aliased sinc" or "asinc" function, the discrete-time signal equivalent to the sinc function prevalent in continuous-time Fourier analysis. Still another name for this function is the Dirichlet function. Figure 14-10 illustrates the magnitude of the DTFT of an $N = 20$ sample, unit amplitude complex sinusoid of normalized frequency $\hat{f}_0 = 0.25$. Note that the peak magnitude is equal to N. The asinc function can be evaluated numerically to show that the peak sidelobe is approximately 13.2 dB below the central peak amplitude, that the 3 dB mainlobe width is approximately $0.89/N$ cycles/sample, and that the peak-to-first null mainlobe width (also called the *Rayleigh* width) is $1/N$ cycles/sample.

The inverse dependence between the length of the signal observation, N, and the width of the mainlobe of the DTFT is called the *reciprocal spreading* property of the Fourier transform: the wider a signal is in one domain (time or frequency), the narrower it is in the complementary (frequency or time) domain. This property is illustrated explicitly in Figure 14-11, which compares the DTFTs of 20 and 40 sample complex exponentials of the same frequency $\hat{f}_0 = 0.1$. Only the real part of the complex exponentials is shown.

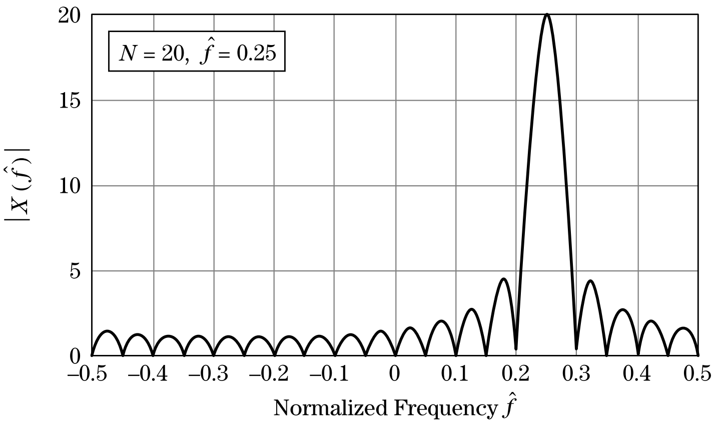


FIGURE 14-10 ■ Magnitude of the DTFT of a pure complex exponential.

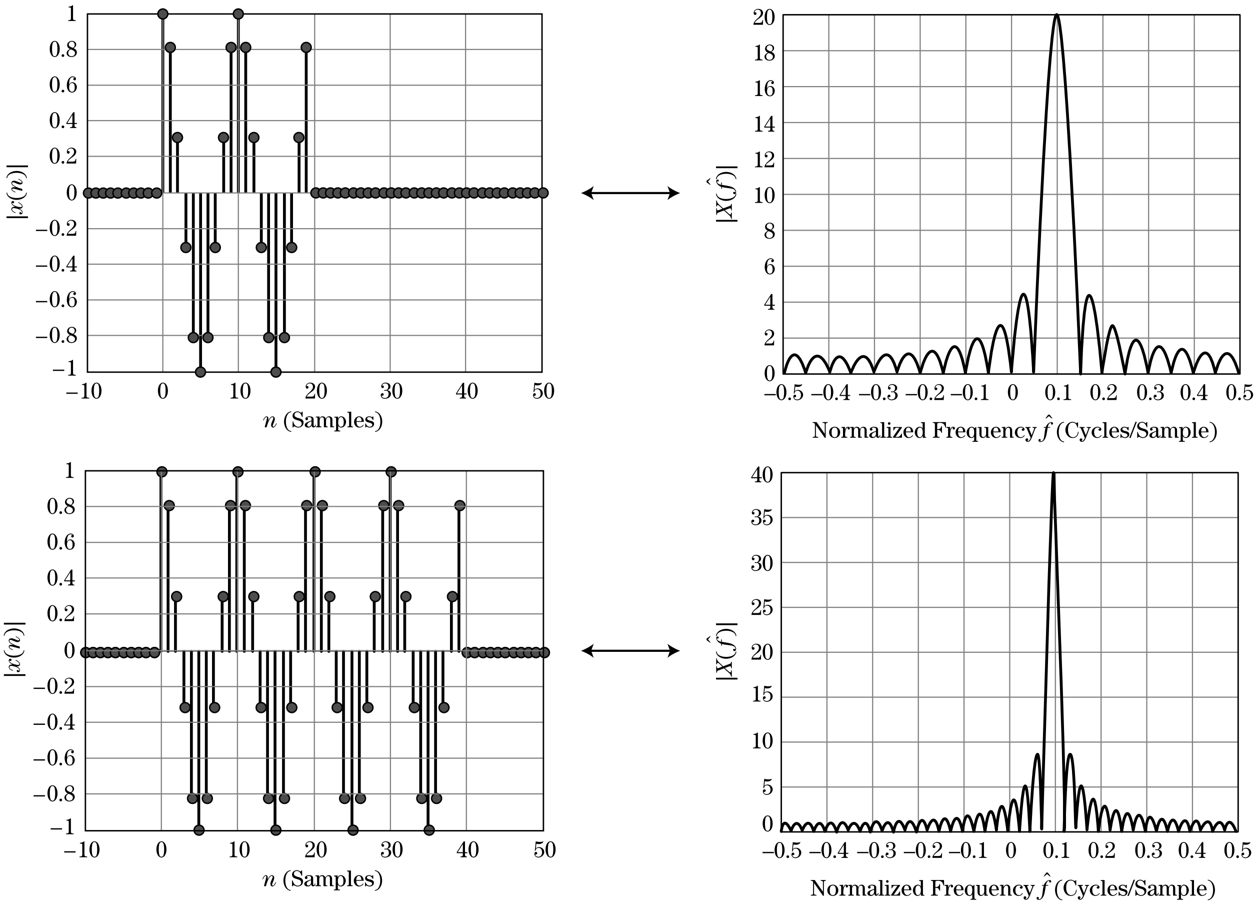


FIGURE 14-11 ■ Illustration of “reciprocal spreading” in the DTFT. (a) $N = 20$ signal samples, and DTFT. (b) $N = 40$ signal samples, and DTFT.

Clearly, doubling the length of the exponential pulse halves the width of the DTFT asinc function.[5]

Specifically, suppose N samples of data are available. The frequency resolution is often taken to be the Rayleigh width of $1/N$ cycles/sample. The $1/N$ criterion is somewhat arbitrary; another common criterion is the two-sided 3 dB width $0.89/N$. The same relationships hold in analog units: if the data are collected at a sampling interval of T_s seconds, the signal duration is NT_s seconds and the DTFT Rayleigh width is $1/NT_s$ Hz whereas the 3 dB width is $0.89/NT_s$ Hz.

14.4.2 Windowing

The –13.2 dB sidelobes of the asinc DTFT in Figure 14-10 are unacceptable in many applications. In radar, multiple targets are frequently observed with received power levels that vary by several tens of decibels due to differences in RCS and range. Consider the Doppler spectrum of a signal containing echoes from two targets. If the peak of a weak

[5] A more formalized expression of this phenomenon is the Fourier uncertainty principle described in [4].

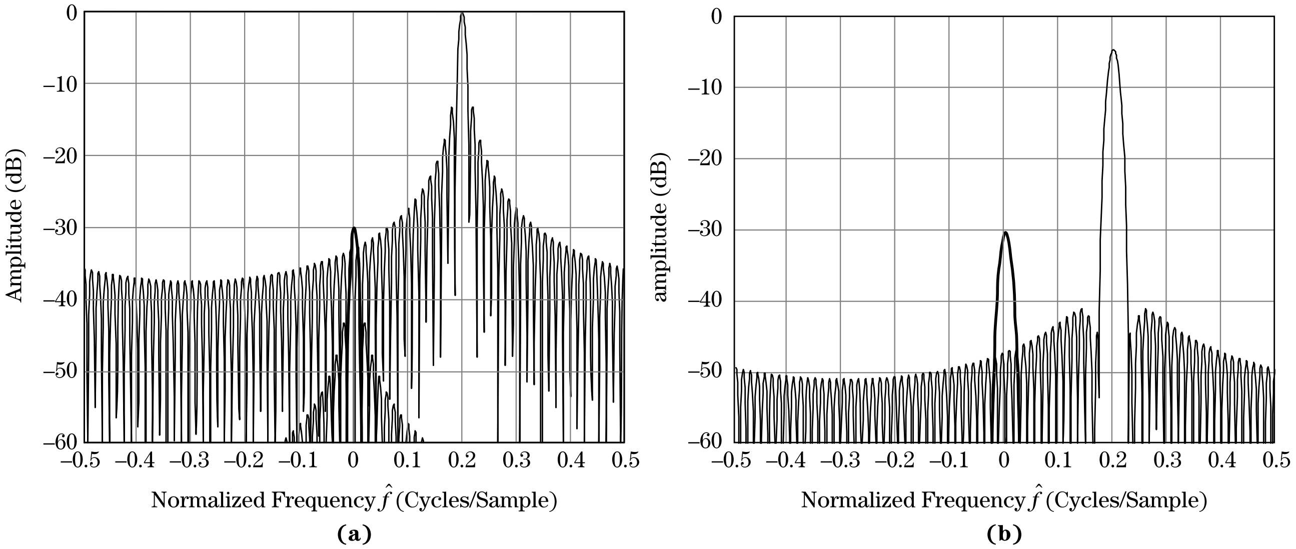


FIGURE 14-12 ■ (a) Masking of a weak target response at $\hat{f} = 0$ by the sidelobes of a 30 dB stronger response at $\hat{f} = 0.2$. (b) Same two responses with Hamming window applied before the DTFT.

target's DTFT is on a par with or is lower than the sidelobes of a strong target's DTFT, the sidelobes of the stronger target will *mask* the DTFT of the weaker target, and only the stronger target will be observed. Figure 14-12a illustrates this effect for a case where the weaker target is 30 dB lower in amplitude than the stronger target.

Windowing the data prior to computing the DTFT is the most common method of ameliorating this problem. Repeating the example of Figure 14-12a with a Hamming window applied to the data before the DTFT results in the spectrum of Figure 14-12b. In this example, the weaker target is easily detectable above the sidelobes of the stronger target.

The windowing procedure is illustrated in Figure 14-13. The upper left plot is a data sequence representing 4 microseconds of a real sinusoid. The lower left sequence is a typical window function, in this case a Hamming window [2]. The two are multiplied point by point to produce the windowed sinusoid shown on the right.

There are many window functions in use; a classic paper by F. J. Harris [8] provides an extensive list of windows and properties.[6] Typical windows are real-valued and are usually smoothly tapered from a maximum value in the center to a lower level (not necessarily zero) at the endpoints. Thus, windows enforce a smooth tapering of the data at the edges of the data sequence.

The principal motivation for windowing is to reduce sidelobes in the DTFT. Figure 14-14 illustrates the effect of a Hamming window on the DTFT of a 20-sample square pulse. The window reduces the energy in the data and its effective time-domain duration by tapering the data near the edges. The reduction in energy is reflected in the DTFT as a reduction in the peak value of the DTFT, in this case by about 5.4 dB. By the reciprocal spreading property, the reduced effective width of the data in the time domain is reflected

[6] Definitions of most common windows are also available at http://www.wikipedia.org and are embedded in many software systems such as MATLAB®.

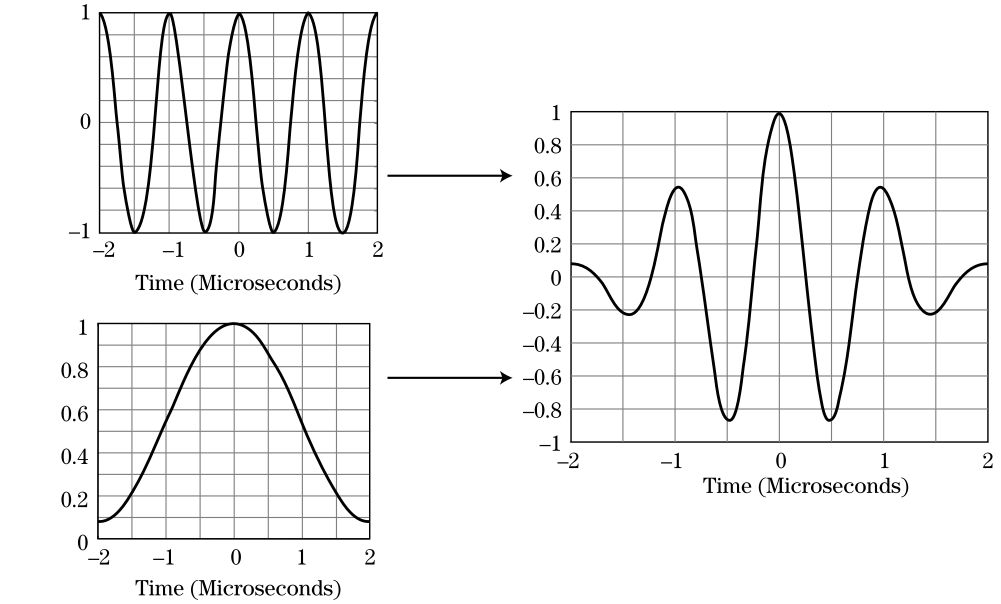


FIGURE 14-13 ■ Illustration of windowing of data.

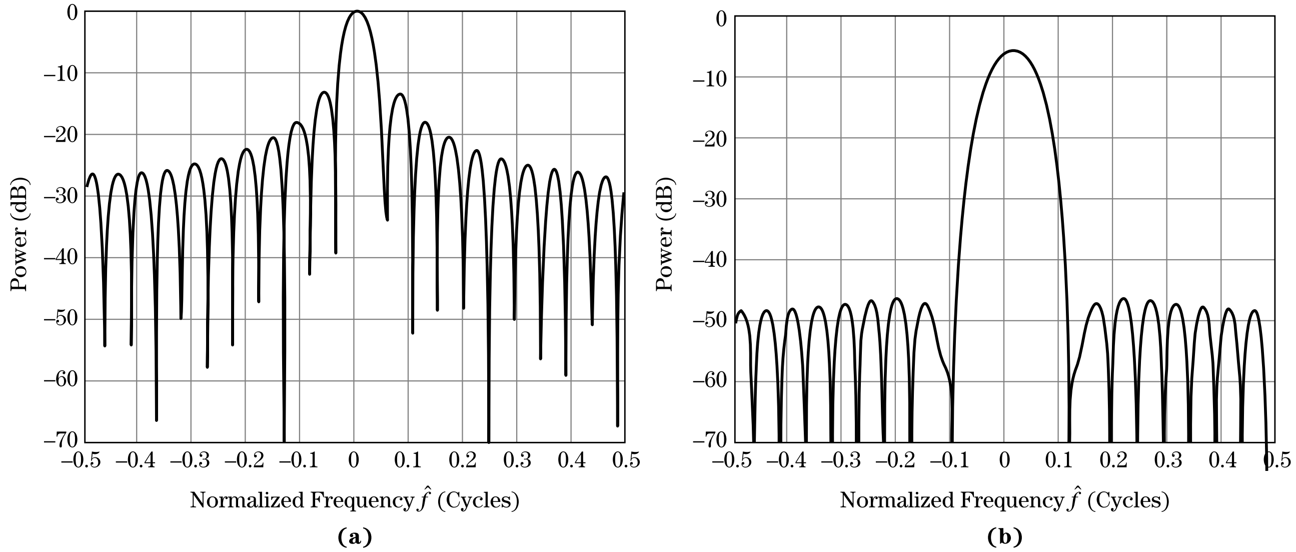


FIGURE 14-14 ■ Effect of a Hamming window on the DTFT of a rectangular pulse. (a) No window. (b) Hamming window.

in a wider mainlobe of its DTFT in the frequency domain. For this example, the 3 dB mainlobe width is increased by 46% compared with the unwindowed case; if measured by the Rayleigh (peak-to-null) width, the increase is 100%. Thus, the frequency resolution of the DTFT has been degraded by 46% to 100%, depending on the metric used. However, these costs of windowing are offset by a dramatic 30 dB reduction in sidelobes, to 43 dB below the DTFT peak. This greatly improves the dynamic range of the DTFT measurement and is the principal reason that windows are routinely used in Fourier analysis.

It is straightforward to quantify some of these window metrics as a function of the window shape. For example, the factor by which the peak power of the DTFT of the windowed data is reduced compared with the DTFT of the unwindowed data is [8]

$$\text{DTFT peak power reduction factor} = \frac{1}{N^2}\left|\sum_{n=0}^{N-1} w[n]\right|^2 \tag{14.17}$$

In (14.17), $w[n]$ is a window function of duration N samples and is assumed to be normalized so that $\max\{w[n]\} = 1$. While this reduction can be significant, the window also tapers any noise present in the data, so the reduction in signal-to-noise ratio is not as great. The factor by which the SNR is reduced due to windowing is given by [8]

$$\text{SNR reduction factor} = \frac{\left|\sum\limits_{n=0}^{N-1} w[n]\right|^2}{N\sum\limits_{n=0}^{N-1} |w[n]|^2} \tag{14.18}$$

In both (14.17) and (14.18) the “reduction” or “loss” is defined to be a number less than 1 on a linear scale and thus a negative number on a decibel scale. The value in dB is obtained by taking $10\log_{10}$ of the values given by the equations.

Most of these metrics are weak functions of the window length, N. As an example, for the Hamming window the loss in signal-to-noise ratio is -1.75 dB for a short ($N = 8$) window, decreasing asymptotically to about -1.35 dB for long windows. Table 14-1 summarizes the five key properties of loss in peak power, peak sidelobe level, 3 dB mainlobe width, SNR loss, and straddle loss (discussed in Section 14.4.5) for several common windows. The values given are for $N = 32$, a typical window length for slow-time pulse-Doppler processing. The “rectangular window” is an untapered window having a constant value of 1.0 for all samples; thus, it actually represents no windowing at all (other than enforcing a finite length to the data) but serves as a reference for the other entries. A much more extensive table, including more metrics as well as the definitions of these and many more types of windows, is given in [8].

TABLE 14-1 ■ Selected Properties of Some Common Window Functions

Window	3 dB Mainlobe Width (Relative to Rectangular Window)	Peak Gain (dB Relative to Peak of Rectangular Windowed Signal)	Peak Sidelobe (dB Relative to Peak of Windowed Signal)	SNR Loss (dB, Relative to Rectangular Windowed Signal)	Maximum Straddle Loss (dB)
Rectangular	1.0	0.0	−13.2	0	3.92
Hann	1.68	−6.3	−31.5	−1.90	1.33
Hamming	1.50	−5.6	−41.7	−1.44	1.68
Kaiser, $\beta = 6.0$	1.63	−6.3	−44.1	−1.80	1.42
Kaiser, $\beta = 8.0$	1.85	−7.5	−57.9	−2.35	1.11
Taylor, 35 dB, $\bar{n} = 5$	1.34	−4.4	−35.2	−0.93	2.11
Taylor, 50 dB, $\bar{n} = 5$	1.52	−5.7	−46.9	−1.49	1.64
Taylor, 50 dB, $\bar{n} = 11$	1.54	−5.8	−49.8	−1.55	1.60
Dolph-Chebyshev (50 dB equiripple)	1.54	−5.6	−50.0	−1.54	1.61
Dolph-Chebyshev (70 dB equiripple)	1.78	−7.2	−70.0	−2.21	1.19

Note: Length $N = 32$ samples in all cases.

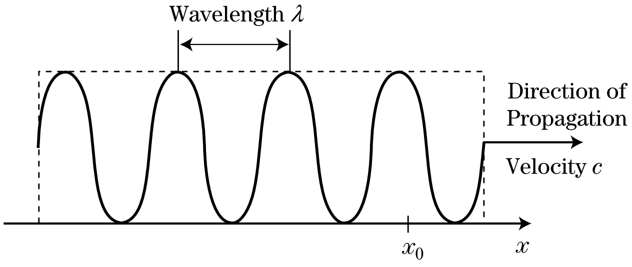


FIGURE 14-15 ■ Temporal and spatial frequency for a sinusoidal pulse.

14.4.3 Spatial Frequency

In discussing the concepts of frequency and Fourier analysis, the signal is usually thought of as being in the time domain. Its Fourier transform is then in the temporal frequency domain, usually just referred to as "frequency." However, in any application such as radar or communications that is concerned with propagating waves, *spatial frequency* is an important concept that will be needed for analyzing spatial sampling, imaging, and space-time adaptive processing. Here a simplified, intuitive introduction to the concept is given. For a more complete discussion, see [3,9].

Consider a radar pulse formed by electromagnetic plane waves propagating in the $+x$ direction with wavelength λ and velocity c as shown in Figure 14-15. An observer at a fixed spatial position x_0 will see successive positive crests of the electric field at a time interval (period) of $T = \lambda/c$ seconds; thus, the temporal frequency of the wave is $f = 1/T = c/\lambda$ Hz or $\omega = 2\pi c/\lambda$ rad/s.

A spatial period can also be defined; it is simply the interval between successive crests of the plane wave in space for a fixed observation time. From the figure, the spatial period of the pulse is obviously λ meters. The spatial frequency is therefore $1/\lambda$ cycles/m or $2\pi/\lambda$ rad/m. It is common to call the latter quantity the *wavenumber* of the pulse and to denote it with the symbol k.[7]

Because position in space and velocity are three-dimensional vector quantities in general, so is the wavenumber. For simplicity of illustration, consider the two-dimensional version of Figure 14-15 shown in Figure 14-16. The plane waves forming the radar pulse are now propagating at an angle θ relative to the $+y$ axis in an x-y plane.[8] The plane wave still has a wavenumber $k = 2\pi/\lambda$ in the direction of propagation. However, projected onto the x axis, the crests are now $\lambda/\sin\theta$ meters apart, and therefore the x component of the wavenumber is $k_x = (2\pi/\lambda)\sin\theta$. Similarly, the y component of the wavenumber is $k_y = (2\pi/\lambda)\cos\theta$. Note that as $\theta \to 0$, $k_x \to 0$ and the wavelength in the x dimension tends to ∞. Similar results hold for the y dimension when $\theta \to \pi/2$.

The extension to three dimensions of space is straightforward. The total wavenumber is related to the components as

$$k = \sqrt{k_x^2 + k_y^2 + k_z^2} \tag{14.19}$$

[7] In keeping with the notation for analog and normalized temporal frequency, the symbol k will be used for analog wavenumber and the symbol $\hat{k}$ for normalized wavenumber.

[8] Incidence angle will often be measured with respect to the normal to the x-axis (i.e., the y-axis) because this is convenient and conventional in analyzing antenna patterns. If the antenna aperture lies in the x dimension, then an incidence angle of $\theta = 0$ indicates a wave propagating normal to the aperture, that is, in the boresight direction.

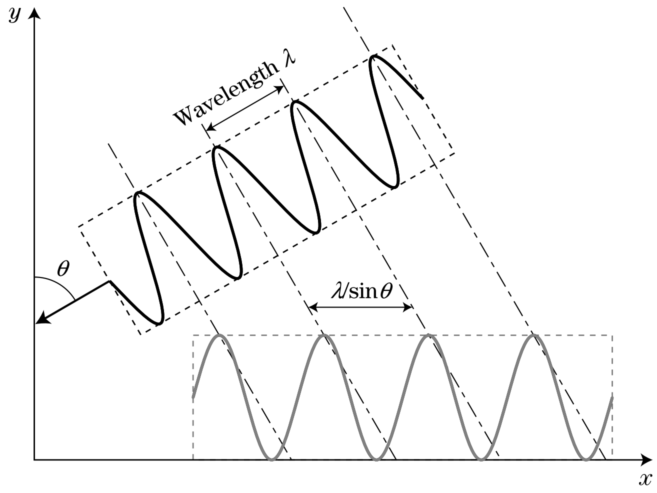


FIGURE 14-16 ■ Spatial frequency and wavelength in the x-direction, expressed in terms of the spatial frequency in the direction of propagation and the incidence angle with respect to the y-axis.

and always equals $2\pi/\lambda$. Note that the temporal frequency remains c/λ Hz regardless of the direction of propagation.

The units of wavenumber are radians per meter. An equivalent but less commonly used cyclical variable is *range frequency, p,* in units of cycles per meter. Thus, $p = k/2\pi$. In monostatic radar, of course, range usually appears as $2R$. The range frequency is defined as $f_r = 2p$. When spatial frequencies are expressed in range frequency units, signal properties such as relationships between bandwidth and resolution, sampling and aliasing, and so forth can be analyzed using the same rules that apply to temporal signals and their Fourier transforms [10].

Equivalent results hold for spatially sampled signals and the corresponding normalized spatial frequencies. If a signal is sampled in space at some interval P_s meters, the DTFT of the spatially sampled data sequence will be in normalized units of cycles/sample or rad/sample. If in rad/sample, the units can be considered to be a normalized wavenumber $\hat{k}$. Normalized wavenumber can be related back to the standard analog wavenumber in rad/m using the relation $\hat{k} = P_s k$. If the DTFT is labeled in cycles/sample $\hat{p}$, then the corresponding spatial frequency p satisfies $\hat{p} = P_s p$.

14.4.4 The Discrete Fourier Transform

Recall that the discrete-time Fourier transform is a function of a continuous normalized frequency variable $\hat{\omega}$ or $\hat{f}$. While the DTFT is an essential tool for analysis, it is not directly useful for computation because of the continuous frequency variable: the DTFT cannot be computed for all $\hat{\omega}$ with a finite number of computations. The solution to this problem is the same as used in the time domain to develop a representation that is sampled in the frequency (as well as in the time or spatial) domain. This representation is called the *discrete Fourier transform* and is not to be confused with the DTFT.

The DFT can be derived by mimicking the steps used in deriving the Nyquist criterion for sampling in time, applying them instead to the DTFT frequency domain. Given a signal $x[n]$ with DTFT $X(\hat{f})$, consider the sampled spectrum $X_s(\hat{f})$, still a function of $\hat{f}$, obtained by multiplying $X(\hat{f})$ by an impulse train of K evenly spaced Dirac impulse functions in the interval [0,1) (repeated periodically as with any DTFT):

$$X_s(\hat{f}) = X(\hat{f})\left\{\frac{1}{K}\sum_{k=-\infty}^{\infty}\delta_D\left(\hat{f}-\frac{k}{K}\right)\right\}$$
$$= \frac{1}{K}\sum_{k=-\infty}^{\infty}X\left(\frac{k}{K}\right)\delta_D\left(\hat{f}-\frac{k}{K}\right) = \frac{1}{K}\sum_{k=-\infty}^{\infty}X[k]\delta_D\left(\hat{f}-\frac{k}{K}\right) \qquad (14.20)$$

where $X[k]$ is defined to be the DTFT samples $X(k/K)$. The inverse transform of $X_s(\hat{f})$, obtained by inserting (14.20) into (14.10) and using the periodicity of $X(\hat{f})$ to shift the limits of integration to the interval (0, 1) for convenience, is

$$x_s[n] = \int_0^1\left\{\frac{1}{K}\sum_{k=-\infty}^{\infty}X[k]\delta_D\left(\hat{f}-\frac{k}{K}\right)\right\}e^{j2\pi n\hat{f}}d\hat{f}$$
$$= \frac{1}{K}\sum_{k=0}^{K-1}X\left(\frac{k}{K}\right)e^{j2\pi nk/K} \qquad (14.21)$$

Note that the finite limits on k resulted from the finite integration limits on $\hat{f}$. To relate $x_s[n]$ to $x[n]$, substitute the definition of $X(\hat{f})$ into (14.21):

$$x_s[n] = \frac{1}{K}\sum_{k=0}^{K-1}\left\{\sum_{m=-\infty}^{\infty}x[m]e^{-j2\pi mk/K}\right\}e^{j2\pi nk/K}$$
$$= \frac{1}{K}\sum_{m=-\infty}^{\infty}x[m]\left\{\sum_{k=0}^{K-1}e^{j2\pi(n-m)k/K}\right\} \qquad (14.22)$$

The inner summation can be evaluated as

$$\sum_{k=0}^{K-1}e^{j2\pi(n-m)k/K} = K\sum_{l=-\infty}^{\infty}\delta[n-m-lK] \qquad (14.23)$$

so that finally

$$x_s[n] = \sum_{l=-\infty}^{\infty}x[n-lK] \qquad (14.24)$$

Equation (14.24) states that if the DTFT of $x[n]$ is sampled in the frequency domain, the signal that corresponds to the inverse transform of the sampled spectrum will be a replicated version of the original signal. This should not be surprising. It was seen early in this chapter that sampling a signal in the time domain caused replication of its spectrum in the frequency domain. In dual fashion, sampling in frequency causes replication in the time domain. The replication period in time is K samples, inversely proportional to the sampling interval of $1/K$ cycles/sample in normalized frequency.

Continuing with the analogy to time-domain sampling, (14.24) shows that $x[n]$ cannot be recovered from the inverse transform of the sampled spectrum $X(k/K)$ unless $x[n]$ is time-limited (analogous to band-limited for time-domain sampling) and the replicas do not overlap. The replicas will not overlap provided that the support of $x[n]$ is limited to $[0, K-1]$. Thus, a signal can be represented by K uniformly spaced samples of its DTFT provided it is of finite-length K samples or less.

Defining the discrete Fourier transform as the sampled DTFT, the K-point DFT and its inverse are

$$\begin{aligned} X[k] &= \sum_{n=0}^{K-1} x[n] e^{-j2\pi nk/K}, \quad k = 0, \ldots, K-1 \\ x[n] &= \frac{1}{K} \sum_{k=0}^{K-1} X[k] e^{j2\pi nk/K}, \quad n = 0, \ldots, K-1 \end{aligned} \tag{14.25}$$

The DFT $X[k]$ is a finite, computable frequency domain representation of the signal $x[n]$. Because the DTFT is periodic, so is the DFT, with period K. Examples of the DFT of specific signals and properties of the DFT can be found in most DSP textbooks, such as [1,2].

The DFT values $X[k]$ are simply samples of the DTFT of $x[n]$:

$$\begin{aligned} X[k] &= X(\hat{f})\big|_{\hat{f}=\frac{k}{K}} \\ &= X(\hat{\omega})\big|_{\hat{\omega}=k\frac{2\pi}{K}} \end{aligned} \tag{14.26}$$

Using the relation $\hat{\omega} = \omega/f_s = \omega T_s$ or $\hat{f} = f/f_s$, the DFT frequency samples can be related to frequency in analog units. Specifically,

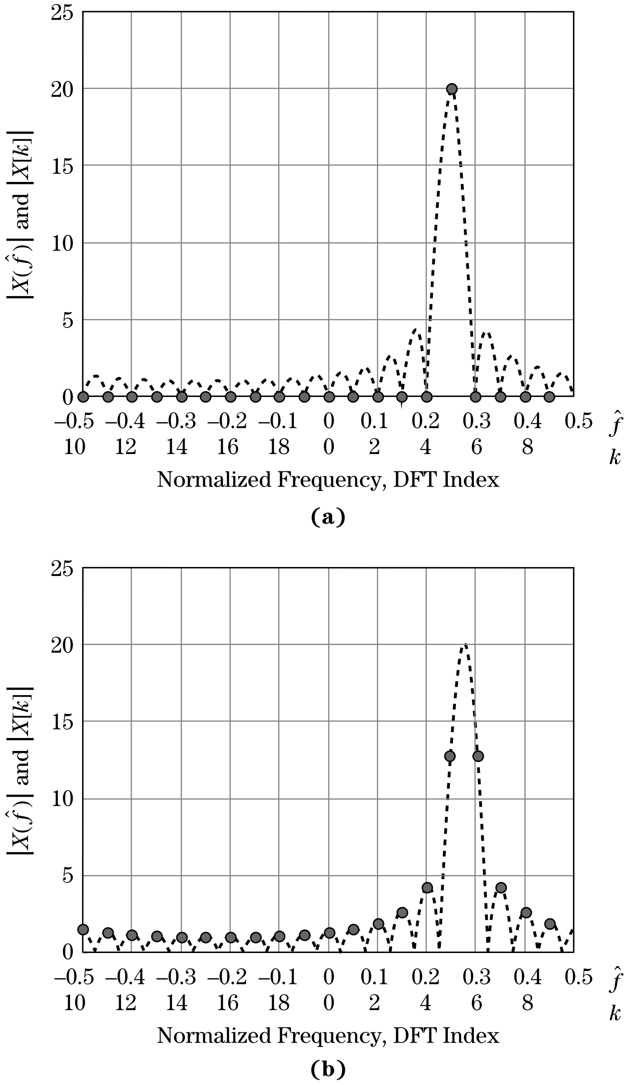
$$\begin{aligned} f_k &= \frac{k}{KT_s} = k\frac{f_s}{K} \text{ Hz}, \quad k = 0, \ldots, K-1 \\ \omega_k &= \frac{2\pi k}{KT_s} = k\frac{2\pi f_s}{K} \text{ rad/s} \quad k = 0, \ldots, K-1 \end{aligned} \tag{14.27}$$

Because of the periodicity of the DTFT, the principal period in $\hat{f}$ can be considered to extend from -0.5 to $+0.5$ instead of 0 to 1, allowing for the representation of negative frequencies. If the DFT is interpreted as representing both positive and negative frequencies, then the samples from $k = 0$ through $\lfloor (K-1)/2 \rfloor$ represent positive frequencies kf_s/K, while the samples from $k = \lceil (K+1)/2 \rceil$ through $K-1$ represent negative frequencies $f_s(k-K)/K$. If $K/2$ is an integer, then the sample at $k = K/2$ corresponds to the Nyquist frequency $f_s/2$.

While an N-point DFT is adequate to represent an N-point discrete sequence, it is often desirable to compute more than N samples of the DTFT of a signal, for example, to obtain a better approximation to the continuous-frequency DTFT or to reduce straddle loss (see Section 14.4.5). Suppose the desired number of frequency samples $K > N$. The K-point DFT of equation (14.25) assumes that $x[n]$ is defined for $0 \le n \le K-1$, but in fact $x[n]$ is only N points long. The solution is to simply append zeroes to the end of the N available data values of $x[n]$, creating a new sequence $x'[n]$ that is K samples long. This process is called *zero padding*. Since no new data values have been added, only zeroes, the DTFT of $x'[n]$ is identical to that of $x[n]$. Computing the K-point DFT simply samples this DTFT at more locations along the frequency axis.

14.4.5 Straddle Loss

The sampling of the DTFT implied by the DFT raises the possibility that the DFT samples may not capture all of the important features of the underlying DTFT. So long as the DFT size, K, is equal to or greater than the signal length, N, the DFT samples capture all of



(a)

(b)

FIGURE 14-17 ■ Illustration of DFT sampling effects and straddle loss. (a) DFT samples falling on the peak and zeroes of the underlying DTFT. (b) DFT samples falling on sidelobes and straddling the peak of the DTFT.

the information needed to reconstruct the DTFT for all frequencies. Nonetheless, the DFT samples may not directly sample important signal features, such as signal peaks. This is illustrated in Figure 14-17. Consider the 20-point complex exponential sequence

$$x[n] = \begin{cases} \exp\left[j2\pi(0.25)n\right], & 0 \leq n \leq 19 \\ 0, & \text{otherwise} \end{cases} \tag{14.28}$$

The dashed line in Figure 14-17a illustrates the magnitude of the DTFT $X(\hat{f})$ (this is the same function shown in Figure 14-10). The solid dots indicate the magnitude of the 20-point DFT of $x[n]$. Note that in this particular case, the DFT samples all fall on zeroes of $X(\hat{f})$, except for the sample at $k = 5$, which falls exactly on the peak of $X(\hat{f})$.

This outcome for the DFT seems ideal: one sample on the peak of the DTFT, with all of the others identically zero.[9] It correctly suggests that the input signal was a single sinusoid

[9] Chapter 17 shows that the DFT is ideal in the sense of maximizing the signal-to-noise ratio for a sinusoid at one of the DFT frequencies in the presence of white noise.

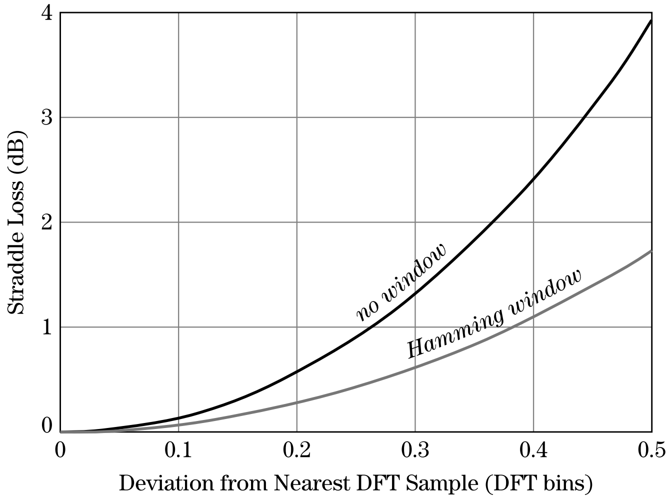


FIGURE 14-18 ■ Worst-case DFT straddle loss for a single sinusoid as a function of the deviation in bins of the sinusoid frequency from the nearest DFT sample frequency. The DFT size equals the data sequence length.

at the frequency corresponding to the nonzero DFT sample. Figure 14-17b shows the same two functions for the case where the exponential frequency is changed from $\hat{f}_0 = 0.25$ to $\hat{f}_0 = 0.275$, a frequency midway between two of the DFT frequency samples. The DTFT has exactly the same functional form but has been shifted on the frequency axis so that it now peaks at $\hat{f} = 0.275$. However, the DFT samples still occur at the same 20 frequencies. These samples now fall approximately on the peak of the DTFT sidelobes instead of the zeroes. More importantly, there is no longer a DFT sample at the peak of the DTFT. Instead, two DFT samples straddle the DTFT peak. Their numerical values are 12.75, in contrast to the peak sample value of 20 in Figure 14-17a.

This example illustrates the sensitivity of the DFT to the alignment of features of the underlying DTFT with the DFT sample frequencies. The two signals in Figures 14-17a and 14-17b are very similar, differing only slightly in frequency. Their DTFTs are identical except for a slight offset in the frequency domain. However, the DFTs look dramatically different.

The apparent reduction in peak signal amplitude in the frequency domain in the second case compared with the first is called *straddle loss*, because it results from the DFT samples straddling, and thus missing, the true peak of the data's DTFT.[10] Straddle loss represents an error in measurement of the amplitude of the underlying spectrum. If the DFT data are being used for signal detection, for example in Doppler processing, this reduction in measured signal amplitude can cause a significant reduction in signal-to-noise ratio and thus in probability of detection. Figure 14-18 shows the straddle loss in decibels for measurement of a single sinusoid as a function of the difference between the signal frequency and the nearest DFT sample frequency, assuming the DFT size equals the signal length (the worst case). For unwindowed data, the loss can be as high as 3.9 dB (the case illustrated in Figure 14-17). The average loss, assuming all values of frequency misalignment are equally likely, is 1.5 dB.

Figure 14-18 also illustrates an underappreciated benefit of windowing. With a Hamming window, the peak straddle loss is reduced from 3.9 dB to 1.7 dB and the average to 0.65 dB. This is a beneficial effect of the widening of the DTFT mainlobe caused by

[10] In many digital signal processing texts, this phenomenon is referred to as *scallop loss* [1].

windowing; the wider DTFT peak does not fall off as fast, so the DFT sample amplitude is not reduced as much when it is not aligned with the DTFT peak. While this does not compensate for the 5.6 dB reduction in peak gain caused by the Hamming window, it does reduce the variability of the apparent peak amplitude as the signal frequency varies.

Even when a peak in the DFT is strong enough to be detected, the misalignment of the DFT frequency samples with the underlying DTFT peak will still result in an underestimate of the signal amplitude and an error in the estimate of the signal frequency. The most obvious strategy for reducing these straddle-induced errors is to increase the DFT size by zero padding the data. This increases the number of frequency samples, reducing their spacing and ensuring that one will fall closer to the DTFT peak. The larger the DFT, the smaller the DFT sample spacing in frequency and the less the maximum straddle loss can be for a given data sequence length. The cost of this procedure is, of course, the increased computation required to obtain the larger DFT. It also interpolates the entire DFT, even if improved detail is desired only in limited regions of the frequency axis.

Increasing the DFT size is equivalent to a band-limited interpolation of the original K-point DFT with a set of K asinc functions [3]. A less accurate but much simpler approach to reducing straddle losses is to perform a local interpolation in the vicinity of each apparent peak [11]. The concept is shown in Figure 14-19 for the example of a quadratic interpolation. The magnitude of the DFT is searched for local peaks. For each local peak found, say, at $k = k_0$, a parabola is fitted through it and the two neighboring samples of $|X[k_0]|$. The equation for the fitted parabola is differentiated and set equal to zero to find the peak of the quadratic, which is then taken as an estimate of the location and amplitude of the true peak. It is possible to develop closed-form equations for the estimated amplitude and peak location without having to explicitly solve for the equation of the interpolating parabola. The result is

$$k' = k_0 + \Delta k = k_0 + \frac{-\frac{1}{2}\{|X[k_0+1]| - |X[k_0-1]|\}}{|X[k_0-1]| - 2|X[k_0]| + |X[k_0+1]|} \tag{14.29}$$

$$\begin{aligned}\left|X\left[k'\right]\right| = \frac{1}{2}\{&(\Delta k - 1)\ \Delta k\ |X[k_0-1]| - 2\,(\Delta k - 1)\,(\Delta k + 1)\,|X[k_0]| \\ &+ (\Delta k + 1)\ \Delta k\ |X[k_0+1]|\}\end{aligned} \tag{14.30}$$

The frequency estimator of equation (14.29) is used more commonly than the amplitude estimator of (14.30). There are a number of similar frequency estimators [11]. These differ

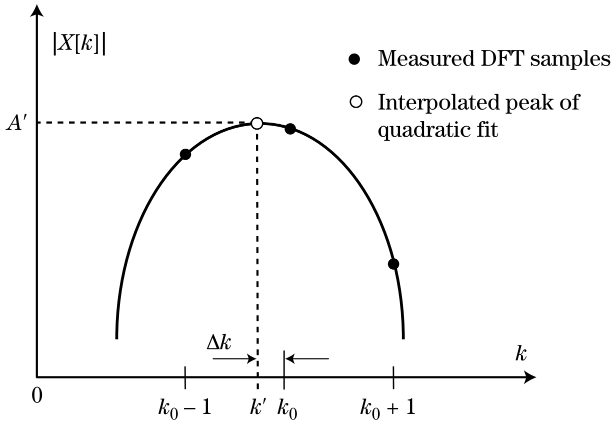


FIGURE 14-19 ■ The concept of quadratic interpolation of a DFT peak.

in the order of the estimate, the coefficients used, and whether they depend on the particular data window used (if any) in the DFT calculation. An example very similar to the previous estimator uses the complex DFT coefficients instead of the magnitude:

$$k' = k_0 + \Delta k = k_0 - \mathrm{Re}\left\{ \frac{Y[k_0+1] - Y[k_0-1]}{2Y[k_0] - Y[k_0-1] - Y[k_0+1]} \right\} \tag{14.31}$$

The amplitude estimator of (14.30) can also be used with this frequency estimator. An improved amplitude estimation technique, at the cost of significantly greater computation, uses (14.29) or (14.31) to estimate the frequency of the peak but then fits an asinc centered at that location to the data [3].

The frequency estimation error using this technique is shown in Figure 14-20. Part a shows the error in the estimate of Δk as a function of the actual value of Δk for both estimators [(14.29) and (14.31)] for the unwindowed, minimum-size DFT case. On this scale, the error using the complex-data version of the estimator is nearly zero. Part b of the figure displays the residual error only for equation (14.31) on a scale magnified by a factor of 1,000.

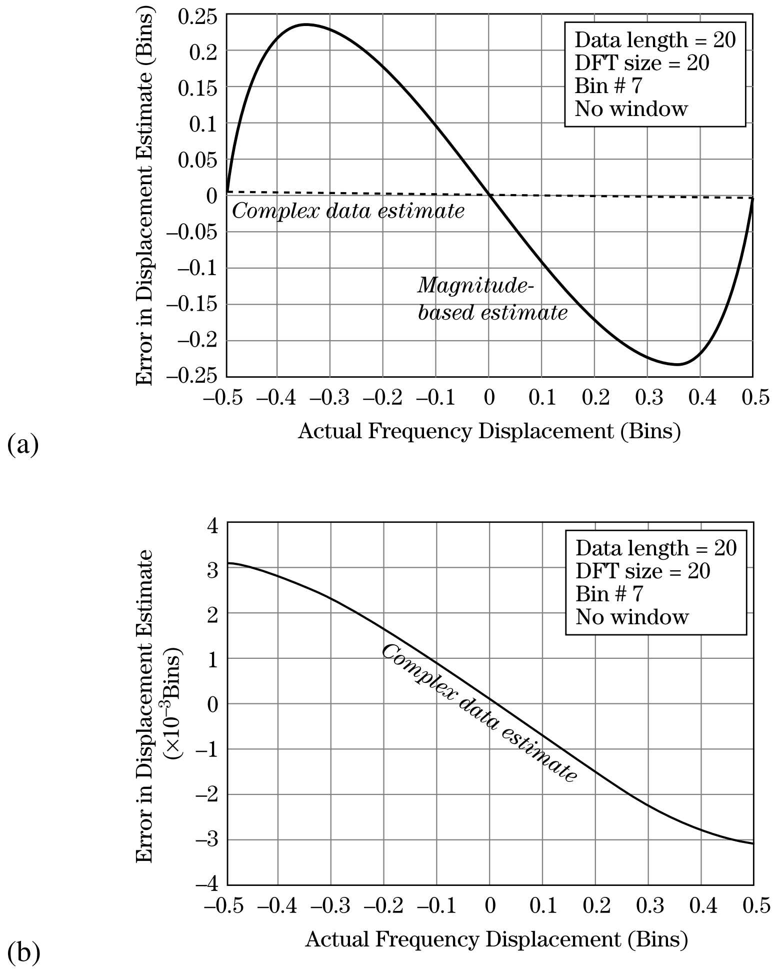


FIGURE 14-20 ■ Error in quadratic interpolation estimation of frequency in the DFT. (a) Residual error in Δk for magnitude- and complex-data estimators. (b) Complex-data estimation error on expanded scale.

14.4.6 The Fast Fourier Transform

The DFT is important because it provides a computable, sampled version of the DTFT. However, it did not become a practical tool for real-time processing until the advent of the fast Fourier transform. The FFT refers to any of a class of algorithms for efficient computation of the DFT; it is not a different mathematical transform. The first widely recognized FFT was the radix-2 algorithm developed by Cooley and Tukey [12]. Used when the DFT length K is a power of two, it reduces the number of complex multiplications required to compute the DFT from K^2 to $(K/2)\log_2 K$, a reduction by a factor of $2K/\log_2 K$. If the data sequence is not a power of 2, it can be zero padded to the required length. The savings in complex additions is a factor of $K/\log_2 K$.

As is typical of fast algorithms, the computational savings due to the FFT algorithm is larger for larger problems (higher values of K) and can become very substantial. These points are illustrated in Figure 14-21. For $K = 16$, the radix-2 Cooley-Tukey FFT achieves a modest but useful savings of 8× compared with the DFT, but for $K = 1{,}024$, the savings is over two orders of magnitude (204.8×, to be exact). The large speedups made possible by publication of the Cooley-Tukey FFT algorithm and its many variants radically increased the variety and order of algorithms that could be applied in real-time radar signal processing.

There are fast algorithms for computing the DFT for other values of K. Cooley-Tukey algorithms are known for radices 4 and 8. The "prime factor algorithm" can be applied when K is a product of prime numbers. Hybrid algorithms exist for other cases. In addition, there are modified algorithms offering additional savings for cases where the data is real-valued or symmetric. A comprehensive summary of DFT and FFT algorithms is available in [13].

14.4.7 Summary of Fourier Transform Relationships

Figure 14-22 summarizes the relationship between the various versions of the Fourier transform discussed in this section. The discrete-time Fourier transform is the appropriate transform for sampled data. It can be derived by sampling the data in the classical Fourier transform for continuous-variable signals. The DTFT frequency variable is continuous, not discrete, but the transform is periodic in frequency. Consequently, the DTFT is primarily an analytic tool.

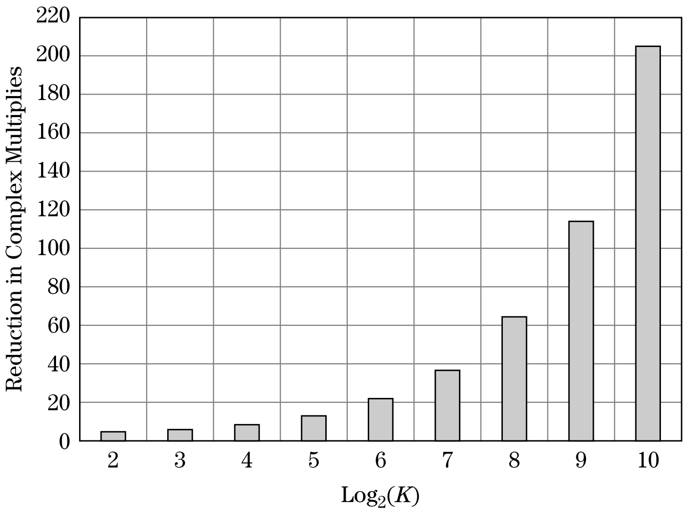


FIGURE 14-21 ■ The $2N/\log_2 N$ reduction in computational load for the radix-2 Cooley-Tukey fast Fourier transform as a function of FFT size.

FIGURE 14-22 ■ Relationship between the various Fourier transforms discussed in this chapter.

Signal Variable	Transform Variable	Transform
continuous	continuous	Fourier Transform (FT)
discrete	continuous	Discrete-Time Fourier Transform (DTFT)
discrete	discrete	Discrete Fourier Transform (DFT)
discrete	discrete	Fast Fourier Transform (FFT)

Sampled data ⇒ replicated spectrum

Sampled spectrum ⇒ finite data sequence

Fast algorithm for DFT

To obtain a computable transform, a transform with a discrete frequency variable needed. Such a transform was obtained by sampling the DTFT in frequency, leading to the discrete Fourier transform. Finally, practical applications require a fast algorithm for actually computing the DFT. This is provided by the various fast Fourier transform algorithms that efficiently compute the DFT for various lengths K. As discussed in Chapter 13, the speed at which an algorithm executes on modern processors often has as much or more to do with the structure of cache memory hierarchies and the efficiency of multiprocessor and multicore data movement as it does with the number of arithmetic operations. The Fastest Fourier Transform in the West (FFTW) library of FFT codes [14] provides efficient implementations of $O(K \log K)$ algorithms for arbitrary values of K on many complex processors.

14.5 THE z TRANSFORM

The z transform of a signal $x[n]$ is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]\, z^{-n} \tag{14.32}$$

where z is a complex variable. In general, this summation may converge only for certain values of z. A sufficient condition for convergence is [2]

$$|X(z)| \leq \sum_{n=-\infty}^{\infty} |x[n]| \left|z^{-n}\right| \leq B_x \tag{14.33}$$

for some finite bound B_x; that is, the summation must be finite. Those areas of the complex z plane (i.e., values of z) where $X(z)$ converges constitute the *region of convergence*, or ROC. The z transform plays the same role in discrete signal analysis that Laplace transforms play in continuous signal analysis, for example, the analysis of system stability.

Note that the DTFT of equation (14.9) is a special case of equation (14.32) for $z = \exp(j\hat{\omega}) = \exp(j2\pi\hat{f})$. In this case, $|z| = 1$, while the argument (phase angle) of z is $\hat{\omega}$. Thus, as $\hat{\omega}$ ranges from 0 to 2π (or $\hat{f}$ from 0 to 1), z traverses the "unit circle" in the complex z plane.

Of particular interest are rational z transforms, where $X(z)$ is the ratio of two finite polynomials in z^{-1}:

$$X(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{14.34}$$

The roots of the denominator polynomial are called the *poles* of $X(z)$, while the roots of the numerator polynomial are called the *zeroes*.

An important property of the z transform is the effect of a delay on the z transform of a signal. If the z transform of $x[n]$ is $X(z)$, then the z transform of $x[n - n_d]$ is $z^{-n_d} X(z)$. In combination with (14.34), this fact leads to one of the three major methods for implementing digital filters, to be described in Section 14.6.3.

14.6 DIGITAL FILTERING

Linear filtering is another major category of digital signal processing pertinent to modern radar. Filters are systems that accept an input signal and perform a useful operation or transformation to generate a modified output signal, as illustrated in Figure 14-23. A digital processor might implement a filter using custom integrated circuits, field programmable gate arrays (FPGAs), or software in a programmable processor.

Linear, shift-invariant (LSI) systems are of greatest interest. A system is linear if, when presented with a linear combination of input signals, the output is the same linear combination of the outputs observed for each input considered separately. Specifically, if the inputs $x_1[n]$ and $x_2[n]$ produce the outputs $y_1[n]$ and $y_2[n]$, then a filter is linear if the input $ax_1[n] + bx_2[n]$ produces the output $ay_1[n] + by_2[n]$. Linearity is sometimes broken down into two component properties, homogeneity and superposition. The system is homogeneous if scaling the input produces a like scaling of the output. It obeys superposition if the response to the sum of two inputs is the sum of the individual outputs. A filter is shift-invariant if delaying or advancing any input signal causes an equal delay or advance in the corresponding output but no other change. Thus, if the input $x[n]$ produces the output $y[n]$, the input $x[n - n_0]$ produces the output $y[n - n_0]$ for any n_0.

LSI filters are completely described by their impulse response $h[n]$, which is the output obtained when the input is the impulse function $\delta[n]$:

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & \text{otherwise} \end{cases} \tag{14.35}$$

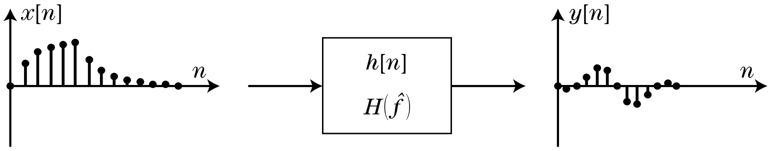


FIGURE 14-23 ■ A linear, shift-invariant system, or filter, is described by its impulse response $h[n]$ and frequency response $H(\hat{f})$.

Given the impulse response of an LSI system, its output can be computed for any input via the *convolution* sum [1,2]:

$$y[n] = \sum_{m=-\infty}^{+\infty} x[m]\,h[n-m] = \sum_{m=-\infty}^{+\infty} h[m]\,x[n-m] \tag{14.36}$$

where the second version of the sum is obtained from the first by a change of variable. Equation (14.36) is useful both for analysis and for actual computation of the filter output. It is often the case that the input, impulse response, or both are of finite duration. If both are finite duration, say, N and L samples, respectively, then (14.36) can be used to show that the duration of the filter output will be limited to $N + L - 1$ samples. This fact has importance for implementing filters using DFTs.

A system is said to be *causal* if, when starting from a state of initial rest, the output is nonzero only for indices n equal to or greater than the first index n_0 at which a nonzero input is applied. In other words, no output occurs before the input is applied. It can be seen from equation (14.36) that an LSI system will be causal if and only if $h[n] = 0$ for all $n < 0$.

It is sometimes convenient to represent linear filtering and convolution by a vector operation. Consider computing the output at time n_0 for a finite impulse response of duration L samples. Define the impulse response and signal column vectors

$$\begin{aligned} \boldsymbol{H} &= [h[0] \quad h[1] \quad \ldots \quad h[L-1]]^T \\ \boldsymbol{X}_{n_0} &= [x[n_0] \quad x[n_0-1] \quad \ldots \quad x[n_0-L+1]]^T \end{aligned} \tag{14.37}$$

Note that the samples of $\mathbf{x}_{n_0}$ are in time-reversed order, while those of $\mathbf{h}$ are in normal order. With these definitions, the output sample $y[n_0]$ can be computed as the vector dot product

$$y[n_0] = \boldsymbol{H}^T \boldsymbol{X}_{n_0} \tag{14.38}$$

Equation (14.38) produces only a single scalar value of the output sequence $y[n]$. To produce additional output samples, the definition of the signal vector $\mathbf{x}_{n_0}$ must be updated for the desired output sample.

14.6.1 Spectral Representations of LSI Systems

The impulse response $h[n]$ is a complete description of an LSI system, and the convolution sum can be used to compute the output for any input if $h[n]$ is known. However, it is sometimes more convenient to work in the frequency domain or the z domain. It is straightforward to show that the z transform of the convolution equation (14.36) gives the relation

$$Y(z) = H(z)X(z) \tag{14.39}$$

$H(z)$ is the z transform of $h[n]$ and is referred to as the *system function* or *transfer function* of the filter. Since the DTFT is simply the z transform evaluated on the unit circle in the z plane, it also follows that

$$Y(\hat{f}) = H(\hat{f})X(\hat{f}) \tag{14.40}$$

$H(\hat{f})$, the DTFT of the impulse response, is called the system's *frequency response*.

14.6.2 Digital Filter Characteristics and Design

Digital filters are divided into two broad classes, *finite impulse response* filters, (also called *nonrecursive* filters) and *infinite impulse response* (IIR) filters (also called *recursive* filters), depending on whether the impulse response $h[n]$ is of finite or infinite duration. Any digital filter is fully described by any one of its impulse response, frequency response, or system function. Either class can implement common filters such as low-pass, band-pass, and high-pass designs, but each class has important advantages and disadvantages.

14.6.2.1 FIR Filters

FIR filters[11] have a finite duration impulse response. If the impulse response is nonzero for $0 \le n \le M-1$, the filter *order* is $M-1$, and the length of the impulse response is M samples. The convolution sum (14.36) is then a weighted combination of a finite number M of input samples. So long as the input data remains bounded (finite) in value, so will the output. Consequently, FIR filters are always stable.

In radar signal processing, much of the information of interest is in the phase of the received signals. Phase information is essential to Doppler processing, pulse compression, synthetic aperture imaging, and space-time adaptive processing, for example. It is important that the signal processor not distort the phase information. The frequency domain relation (14.40) suggests that this would require that the phase of $H(\hat{f})$ be identically zero. However, it is sufficient to restrict $H(\hat{f})$ to have a linear phase function. Linear phase corresponds to pure delay in the time domain [2] but no other distortion. FIR filters can be designed to have exactly linear phase. If the impulse response is either exactly symmetric or exactly antisymmetric,

$$\begin{gathered} h[n] = h[M-1-n] \\ \text{or} \\ h[n] = -h[M-1-n] \end{gathered} \tag{14.41}$$

then the frequency response $H(\hat{f})$ will have linear phase [2]. As an added benefit, symmetric or antisymmetric filters also offer computational savings compared with an arbitrary FIR filter. Conventional IIR filters do not have linear phase.[12] Thus, FIR filters are popular in many radar applications.

There are many algorithms for designing FIR filters to meet a set of specifications, usually given in the form of a desired frequency response. Only two are briefly mentioned here. Much more detail is given in most DSP texts such as [1,2]. Whatever the algorithm, the resulting filter is represented by its impulse response $h[n]$.

The window method is a particularly simple technique that is well suited to general band-pass/band-stop frequency responses. The user specifies the desired ideal frequency response $H_i(\hat{f})$ and then computes the corresponding impulse response $h_i[n]$ using the inverse DTFT formula (14.10). In general, this impulse response will be infinite in duration.

[11] The acronym in the term *FIR filter* is often pronounced as the word *fir* (like the type of tree), resulting in the term *fir filter.*

[12] It is possible to achieve zero-phase filtering using IIR filters by filtering the signal twice, once in the forward and once in the reverse direction. Also, some advanced design techniques can achieve almost-linear phase IIR filters. Only FIR filters can readily achieve exactly linear phase without multiple filter passes.

A finite impulse response is then obtained by multiplying $h_i[n]$ by a finite length window function $w[n]$:

$$h[n] = w[n]h_i[n] = w[n] \cdot \text{DTFT}^{-1}\{H_i(\hat{f})\} \tag{14.42}$$

Because $h[n]$ is only an approximation to $h_i[n]$, the final realized frequency response $H(\hat{f})$ will be an approximation to $H_i(\hat{f})$. The choice of window function offers the designer a trade-off between the transition bandwidth between passbands and stopbands (a measure of the sharpness of the frequency response cutoffs) versus the sidelobe level in the stopbands.

FIR filters designed by the window method have generally equal peak approximation errors in the various passbands and stopbands. In addition, ripples in the frequency response are largest near the band edges. The Parks-McClellan design method provides more control over the frequency response at the expense of greater computational effort. The algorithm minimizes the maximum error between the actual and ideal frequency response for a given filter order. The resulting filters have an equiripple characteristic: the approximation error is spread evenly throughout the passband and stopbands rather than being concentrated at the edges. In addition, the Parks-McClellan algorithm allows the user to specify the relative amount of error in the various passbands and stopbands instead of being limited to equal error in each band as with the window method. For a given peak sidelobe level, the Parks-McClellan design will have a sharper transition between passbands and stopbands than a filter designed with the window method.

14.6.2.2 IIR Filters

IIR digital filters do not have the guaranteed stability of FIR filters; rather, they must be designed to be stable. Furthermore, IIR filters cannot readily be designed to have exactly linear phase. However, IIR filters generally achieve a more selective frequency response (sharper cutoff, lower sidelobes) for a given filter order than a comparable FIR filter. Linear phase is not required in some radar applications, such as MTI filters [15] and some digital IF sampling approaches [16]. For such applications, IIR filters can provide a given level of frequency selective filtering at less computational cost than FIR filters, provided attention is paid to the filter stability.

IIR filters are designed by either applying a transformation to one of the many well-documented analog filter designs or by using one of many direct computer-aided optimization techniques. For example, the *bilinear transformation* converts an analog filter specified by the analog system function $H_a(s)$ into a digital filter specified by the system function $H(z)$ by applying the substitution

$$s = \frac{2}{T_s}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \tag{14.43}$$

to $H_a(s)$, where T_s is the sampling period of the digital filter. The bilinear transformation has many useful properties, such as preserving the stability of analog filters in the corresponding digital design and being applicable to all types (low-pass, band-pass, high-pass, and arbitrary frequency response) of filter designs. Furthermore, rational analog transfer functions map into rational digital transfer functions of the form (14.34).

Because the impulse response is infinite in duration by definition, IIR filters are normally not represented by $h[n]$, since that would require an infinite number of coefficients. Instead, IIR filters are represented by their discrete system function $H(z)$, which will have only a finite number of coefficients.

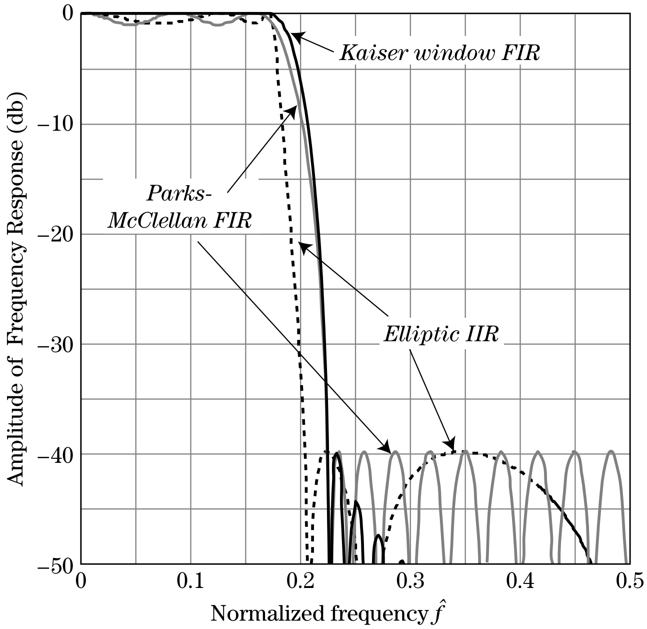


FIGURE 14-24 ■ Comparison of three classes of digital filters with similar design requirements (see text for details).

14.6.2.3 Filter Design Comparison Example

Some of the advantages and disadvantages of IIR and FIR filters can be summarized by comparing designs that result from comparable specifications. Consider a low-pass filter with a normalized frequency cutoff of $\hat{f} = 0.2$ cycles/sample, a transition bandwidth $\Delta\hat{f} = 0.1$ cycles/sample, a passband gain that does not exceed unity (0 dB) with a ripple of 1 dB, and a stopband attenuation of 40 dB. Figure 14-24 compares the frequency responses of an elliptic IIR filter and two FIR filters: one designed with the window technique using a Kaiser window and the other with the Parks-McClellan algorithm. Routines for the automated design of all of these filters classes, and several more, are readily available in MATLAB® and other mathematical software packages.

Elliptic filters are a class of recursive designs that have equiripple errors in both the passband and stopband; this behavior is clearly evident in Figure 14-24. A sixth-order elliptic filter achieves the desired specifications and can be implemented at a cost of 11 multiplications and 10 additions per output sample. As designed, the filter is stable, but if the coefficients were quantized for a fixed-point implementation, care would be required to ensure that stability is maintained.

A Parks-McClellan filter is an FIR design that also exhibits equiripple behavior in both the passband and stopband. A 29th order filter meets the same specifications as the elliptic filter. The FIR coefficients are symmetric, so that the frequency response of the filter inherently exhibits exactly linear phase over the entire frequency spectrum. By taking advantage of this symmetry, the filter can be implemented at a computational cost of 15 multiplications and 28 additions per output sample, still significantly more than for the elliptic design. The transition bandwidth, which is the change in frequency over which the frequency response transitions from the passband to the desired attenuation in the stopband, is significantly wider for the Parks-McClellan design than for the elliptic filter.

Finally, a 43rd-order Kaiser window design also meets the stopband attenuation requirement. This filter is also exactly linear phase. It requires 22 multiplications and 42 additions per output sample. Like all filters designed via the window method, this design is not equiripple in either the stopband or passband. Because the passband and stopband ripples are approximately equal, the passband appears smooth when plotted on a dB scale. In fact, the Kaiser window filter in Figure 14-24 exhibits a passband ripple of about 0.15 dB, while the Parks-McClellan and elliptic filters were designed to have a passband ripple of approximately 1 dB. The stopband ripples of the window design filter decay as approximately $1/f$. The Kaiser window gives the narrowest transition bandwidth of any window design for a given filter order and stopband attenuation. In this example, the transition bandwidth is slightly narrower than the Parks-McClellan design, though still significantly wider than that of the elliptic filter.

14.6.3 Implementing Digital Filters

There are three basic methods for implementing digital filters: time-domain convolution, time-domain difference equations, and frequency-domain convolution, also called *fast convolution*. In addition, there are a number of variants for long signals, such as the overlap-add and overlap-save techniques [2]; however, these build on the three basic methods.

The first method is the time-domain convolution of equation (14.36). This method requires that the impulse response $h[n]$ of the filter be known but is very straightforward to implement. Convolution relies on a series of multiply-accumulate (add) operations. If the impulse response is M samples long, then implementation of convolution requires M multiplications and $M - 1$ additions per output sample. In radar, the data are often complex-valued, and the filter coefficients are sometimes complex-valued as well, so these are complex multiplications and additions. Note that direct convolution is practical only for FIR filters, for which M is finite.

The second method uses the convolution theorem of Fourier transforms given in (14.40). When using the DTFT, this technique is applicable in principle to both FIR and IIR filters. However, the DTFT, while good for analysis, is not a computable transform due to the continuous frequency variable. Thus, the discrete frequency DFT version is used instead. This states that the product of two DFTs, say, $Y[k] = H[k]X[k]$, is the DFT of the *circular* convolution of the corresponding sequences $x[n]$ and $h[n]$ [2]. "Circular convolution" is the name given the result of convolving[13] periodic sequences; recall that sampling the DTFT to create the DFT implies a periodic replication of the input signal as derived in equation (14.24). Circular convolution produces results at the beginning and end of the output sequence that differ from the standard convolution result, sometimes significantly.

The standard convolution, not a circular convolution, is usually required. Despite the replication effects, this can be accomplished using DFTs if the DFT size is chosen correctly. Specifically, if the lengths of the sequences $x[n]$ and $h[n]$ are N_x and N_h samples, respectively, then their linear convolution has a length of $N_x + N_h - 1$ samples. The circular convolution produced by the inverse of the product of their DFTs is identical to the linear convolution provided that the DFT size $K \geq N_x + N_h - 1$. Thus, for finite-length data sequences, an FIR filter can be implemented using

$$y[n] = \mathrm{DFT}_K^{-1}\{H[k]X[k]\} \tag{14.44}$$

[13] Note that the base verb for describing the operation of convolution is "to convolve," not "to convolute."

where DFT_K^{-1} represents an inverse DFT of length K and K satisfies the previous constraint.

The process of FIR filtering using multiplication of DFTs is referred to as *fast convolution*. Whether it is in fact faster than direct convolution depends on the parameters involved. Calculation of $y[n]$ using equation (14.44) requires computation of two K-point DFTs, a K-point vector multiplication to form the product $H[k]X[k]$, and a K-point inverse DFT. Thus, the number of (complex, in general) multiplications and additions is approximately $(3(K/2)\log K + K)$, where K is approximately the sum of the lengths of $x[n]$ and $h[n]$. While this is often fewer than in direct convolution, there are exceptions, particularly when one of the sequences (typically $h[n]$) is much shorter than the other. If the same filter will be applied to many different data sequences, the filter frequency response $H[k]$ need be computed only once; the number of additional operations per data sequence filtered is then reduced to $(K\log K + K)$.

Very long or continuously running input sequences can be filtered using FIR filters and fast convolution applied to the overlap-add or overlap-save methods previously mentioned. The basic idea of both methods is to divide a long input signal into shorter segments, to filter each segment individually, and then to use linearity to reassemble the filtered segments and form the filtered output. Filtering of the individual segments can be done using time-domain or fast convolution. Details are given in [2] and in most DSP texts.

The final implementation method is the one most used for IIR filters. It begins with equation (14.39), which relates the z transforms of the filter input and output and the system function of the filter. For a rational $H(z)$ of the form of (14.34), this becomes

$$
\begin{aligned}
Y(z) &= \frac{\sum\limits_{k=0}^{M} b_k z^{-k}}{1+\sum\limits_{k=1}^{N} a_k z^{-k}} X(z) \\
\Rightarrow \left(1+\sum_{k=1}^{N} a_k z^{-k}\right) Y(z) &= \left(\sum_{k=0}^{M} b_k z^{-k}\right) X(z)
\end{aligned} \tag{14.45}
$$

Applying the z transform property that the z transform of $x[n-n_d]$ is $z^{-n_d}X(z)$ to each term on both sides of (14.45) and rearranging gives a *difference equation* for computing the filter output $y[n]$:

$$
y[n] = \sum_{k=0}^{M} b_k x[n-k] - \sum_{k=1}^{N} a_k y[n-k] \tag{14.46}
$$

Difference equations are the discrete equivalent of differential equations in continuous analysis.

Equation (14.46) expresses each output sample as the weighted sum of M input samples and N previous output samples. Thus, IIR filters, unlike FIR filters, incorporate feedback of previous outputs to the current output. For this reason, as mentioned earlier, they are also called recursive digital filters; FIR filters are nonrecursive. The presence of feedback is also the reason that IIR filters can become unstable if improperly designed. Despite the infinite impulse response, the difference equation approach allows computation of each output sample using a finite number of operations, namely, approximately $(M+N)$ multiplications and additions per output sample.

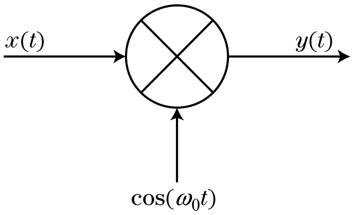


FIGURE 14-25 ■ A modulator.

14.6.4 Shift-Varying and Nonlinear Systems

FIR and IIR filters are linear, shift-invariant systems. Though not nearly so ubiquitous as LSI systems, nonlinear and shift-varying operations are commonly used in certain radar signal processing operations, and their use can be expected to grow over time.

The most important example of a shift-varying operation in radar is simple modulation, that is, multiplication of a signal by a sinusoid to shift its Fourier transform on the frequency axis. Modulation is illustrated in Figure 14-25. Suppose the input to the system is $x(t)$. The output is $y_1(t) = x(t)\cos(\omega_0 t)$. If the input to the system is delayed by t_d seconds, becoming $x(t - t_d)$, the output will be $y_2(t) = x(t - t_d)\cos(\omega_0 t)$. Because $y_2(t) \neq y_1(t - t_d)$, the system is not shift-invariant. This occurs in this case because delaying the input does not also delay the time origin of the oscillator used in the modulator. Note that the modulator is still linear, however. It is also worth noting that the output of an LSI system can contain energy only at frequencies at which there was energy in the input signal. Modulators, which move signal energy to new locations on the frequency axis, must therefore be non-LSI.

Modulators are so common in coherent receivers that they rarely draw any notice. A more interesting example of a shift-varying operation is synthetic aperture radar (SAR) image formation. SAR processing is discussed in Chapter 21. The core operation is a *matched filter* (see Section 14.9) implemented as a two-dimensional correlation using one of a variety of algorithms. However, in general (and in practice in many newer systems), it is a shift-varying correlation because the response of the SAR to a point target is a function of the range to that target. Focusing at different ranges consequently requires different matched filter correlation kernels. In lower (coarser)-resolution SAR systems, the variation in range may be insignificant so that the system is effectively LSI. However, in modern high-resolution SAR systems, the shift-varying nature is not negligible, leading to complicated correlation algorithms.

The previous examples, while shift-varying, are still linear. *Median filters*, and more generally *order statistics filters*, are a class of nonlinear operations that are used in radar for such functions as detection and image enhancement. A median filter is a sliding-window operation, very similar to an FIR filter. The difference is that instead of computing the output as a sum of the input samples weighted by the FIR filter impulse response, the output is the median of the input samples. Thus, equation (14.38) is replaced by

$$y[n_0] = \text{median}(\boldsymbol{X}_{n_0}) \qquad (14.47)$$

A simple way to find the median of a list of N data values is to sort the list into numerical order and then to simply choose the middle element of the list (or the average of the two middle elements if N is even). An order statistics filter is a generalization of this idea in which the m-th element of the ordered list is output. For instance, the second-order statistic filter would output the second-largest element of the list. A median filter is simply the $N/2$ order statistic filter.

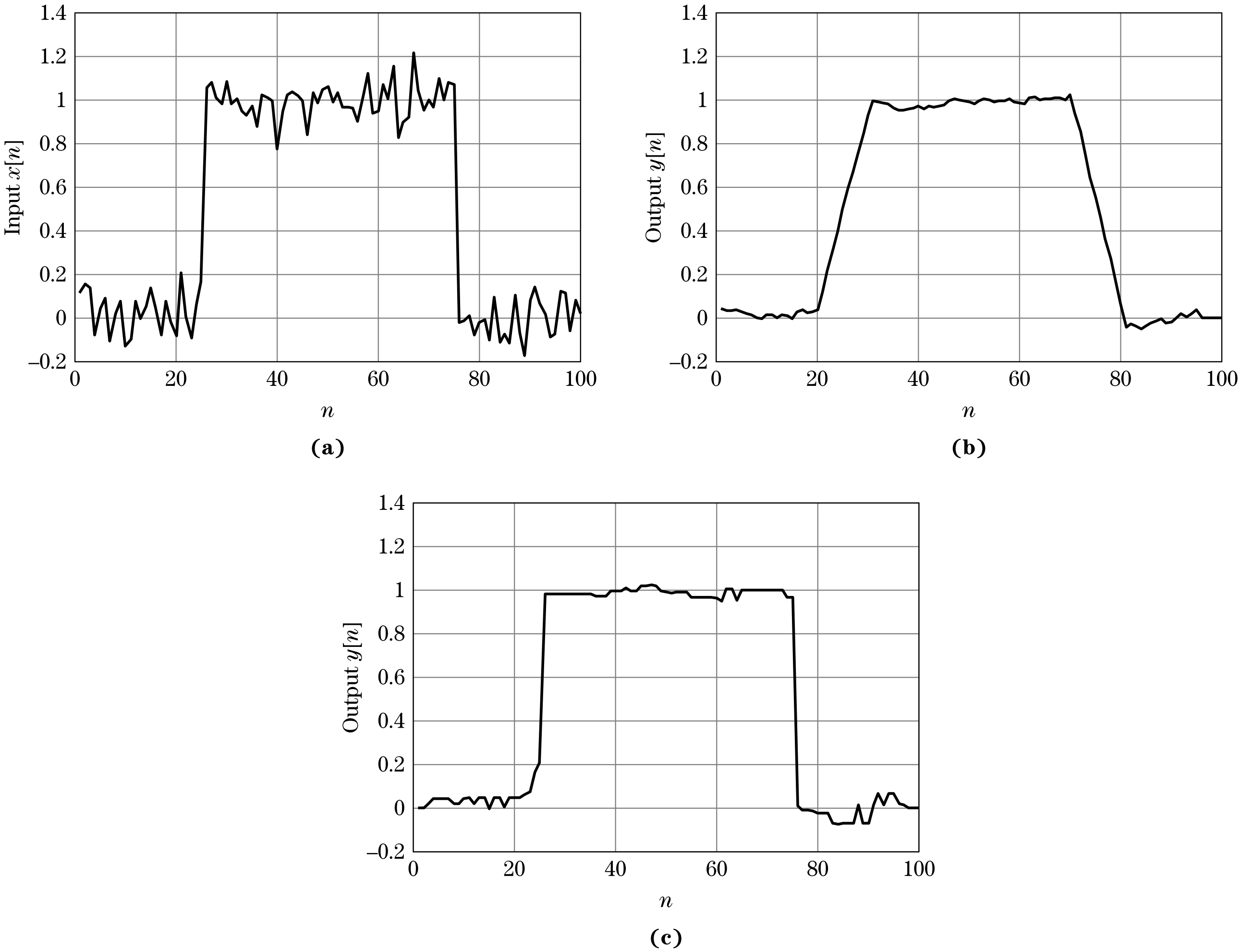


FIGURE 14-26 ■ Comparison of FIR and median filter. (a) 50-point square pulse with noise added. (b) Output of 11-point moving average FIR filter. (c) Output of 11-point median filter.

Order statistic filters are nonlinear. They exhibit the property of homogeneity, but not of superposition. To see an example of this, consider a median filter acting on the two three-point input sequences $\boldsymbol{X}_1 = \{1,4,0\}$ and $\boldsymbol{X}_2 = \{5,3,3\}$. The output of the median filter in response to the first sequence is $y_1 = 1$; this is found by sorting the list $\{1,4,0\}$ into numerical order, giving the new sequence $\{0,1,4\}$, and then taking the middle value of the list. Similarly, $y_2 = 3$. If a new sequence $\boldsymbol{X} = \boldsymbol{X}_1 + \boldsymbol{X}_2 = \{6,7,3\}$ is formed, the output of the median filter will be $y = 6 \neq y_1 + y_2$. Thus, the median filter does not exhibit the superposition property and is therefore not linear.

Figure 14-26 compares a median filter with a moving average filter in smoothing a noisy square pulse. Figure 14-26b is the output of an 11-point moving average filter, which is simply an FIR filter in which all of the filter coefficients have the same value of $1/L = 1/11$. While the filter greatly reduces the noise, it also smears the vertical edges of the pulse over an 11-sample transition region. Figure 14-26c is the output of a median filter for the same input. This nonlinear filter reduces the noise by a similar amount without introducing any smearing of the edges. A two-dimensional version of this filter is sometimes useful for reducing noise in imagery without blurring the edges of objects in the image.

The other major area of radar signal processing where order statistic filters are commonly used is in constant false alarm rate (CFAR) detection. This topic will be discussed in Chapter 16 in detail. The key operation is estimating the interference mean from a finite set of interference samples; this mean is then used to set a detection threshold. Both averaging (FIR filters) and order statistic filters can be applied to this task; that is, either the sample mean of the available data or its median (or other order statistic) can be used to estimate the true interference mean. Order statistic CFARs may be more useful when the data are likely to contain "outliers."

Nonlinear systems are more difficult to analyze than LSI systems. In particular, the concepts of impulse response and convolution do not apply, and Fourier analysis is much less informative. Another interesting aspect of order statistics filters in particular, and many nonlinear systems, is that they put much greater emphasis on sorting and counting operations than on the multiply-accumulate operations of LSI systems. This has a significant impact on processor design.

14.7 RANDOM SIGNALS

Many signals of interest are modeled as *random processes*. The most obvious example is noise. However, many deterministic signals are so complicated that they can be reasonably modeled only as random processes. Examples in radar include reflections from complex targets or from clutter, as described in Chapters 8 and 5, respectively. Here it is sufficient to consider only stationary random processes; a more general description is given in [2] and in many other DSP and random process textbooks.

14.7.1 Probability Density Functions

A stationary discrete random process is a discrete function $x[n]$ where the value x_n of x at each index n is a random variable described by some *probability density function* (PDF) $p_{x_n}(x)$, and the relationship between values x_n and x_m of $x[n]$ at different times n and m is governed by a set of joint PDFs. The stationarity restriction implies that the PDF of x_n is the same for every n; that is, the statistics do not change with time. It also implies that the joint PDF of x_n and x_m depends only on the spacing $n - m$, not on the absolute indices n and m.

Formally, a PDF describes the probability that $x_n = x[n]$ will take on a value in a particular range:

$$\text{Probability that } x_1 \le x_n \le x_2 = \int_{x_1}^{x_2} p_{x_n}(x)dx \tag{14.48}$$

For example, Figure 14-27 shows an example of a Gaussian PDF. The probability that the random variable x_n will be between 0.5 and 1.5 is the integral of the PDF from $x = 0.5$ to $x = 1.5$, which is simply the gray shaded area. It follows that the PDF can be thought of as indicating the relative likelihood that x_n will take on various values. The higher the value of the PDF at some point $x = x_0$, the more likely that x_n will take on a value near x_0. In the Gaussian example of Figure 14-27, x_n is more likely to take on values near zero than to take on values near 2, for instance.

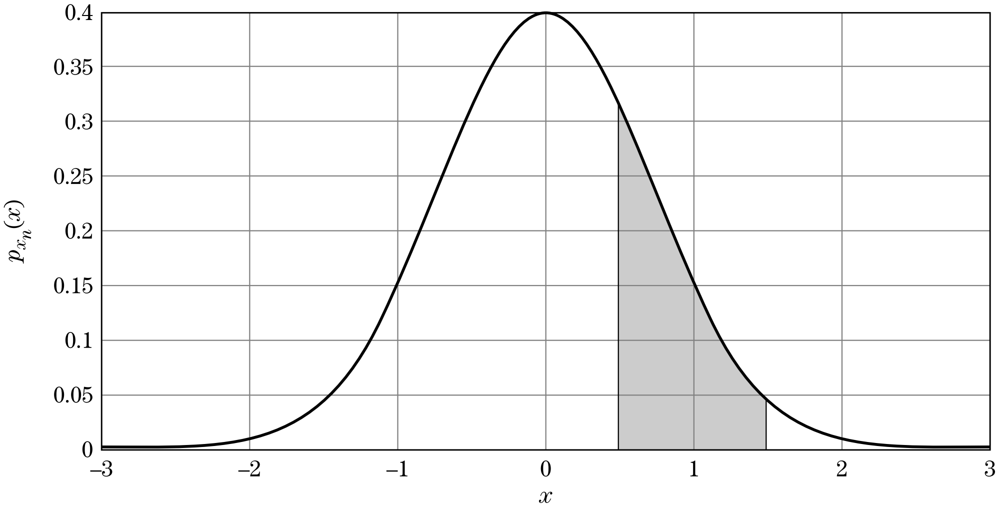


FIGURE 14-27 ■ Use of PDF to compute probabilities.

14.7.2 Moments and Power Spectrum

Random processes are also characterized by their *moments*. Most common are the mean, mean square, and variance:

$$\text{mean: } \bar{x} = m_x = \mathrm{E}\{x\} \equiv \int_{-\infty}^{\infty} x p_x(x)\; dx$$

$$\text{mean square: } \overline{x^2} = p_x = \mathrm{E}\{x^2\} \equiv \int_{-\infty}^{\infty} x^2 p_x(x)\; dx$$

$$\text{variance: } \sigma_x^2 = \mathrm{E}\{(x-\bar{x})^2\} \equiv \int_{-\infty}^{\infty} (x-\bar{x})^2\, p_x(x)\; dx \tag{14.49}$$

The notation E{·} means "expected value" and is defined by the integrals shown. For any random variable, regardless of PDF, it is true that $\sigma_x^2 = \overline{x^2} - (\bar{x})^2$, which also means that the mean square and variance are equal if the mean of the random process is zero. For stationary signals, these moments are independent of the index n. That is, the mean, mean square, and variance are the same at all points within the sequence.

The mean is the average value of a process. The mean square is the average value of the square of the process and is thus the average power of the process. The variance is a measure of how wide a range of values the process takes on around its mean. A small variance means that most of the values are concentrated near the mean; a large variance means that the process takes on values over a wide range about the mean.

The moments in equation (14.49) describe behavior of $x[n]$ at a single index, n. Correlation functions are moments that relate values of random processes at different indices. The *cross-correlation* function compares values at two different times in two different random processes x and y, while the *autocorrelation* function compares values of the same random process at two different times. They are defined as

$$\begin{aligned}&\text{cross-correlation: } \phi_{xy}[m] = \mathrm{E}\{x[n]y^*[n+m]\}\\&\text{autocorrelation: } \phi_{xx}[m] = \mathrm{E}\{x[n]x^*[n+m]\}\end{aligned} \tag{14.50}$$

Stationarity ensures that these correlation functions depend only on the interval m between the two samples, called the *correlation lag* or just *lag*, and not on the absolute indices n and $n+m$. Note that $\phi_{xx}[0] = \overline{x^2}$. Thus, the zero lag of the autocorrelation function gives the power in a random process. If the process is zero mean, $\phi_{xx}[0] = \sigma_x^2$.

An alternative definition of the cross- and autocorrelation is

$$\begin{aligned}&\text{cross-correlation: } \bar{\phi}_{xy}[m] = \mathrm{E}\{x[n]y^*[n-m]\}\\ &\text{autocorrelation: } \bar{\phi}_{xx}[m] = \mathrm{E}\{x[n]x^*[n-m]\}\end{aligned} \tag{14.51}$$

Comparing with equation (14.50) shows that $\bar{\phi}_{xy}[m] = \phi_{xy}[-m]$. $\bar{\phi}_{xy}[m]$ exhibits the same basic properties as $\phi_{xy}[m]$, but results derived using one definition often differ from the same results derived using the other by reflections, conjugations, or both; readers must be alert to differences in definitions when comparing results from different sources.

The autocorrelation indicates how well knowledge of a random process at one index predicts its value at another index. Specifically, it can be shown that if the autocorrelation function is normalized to have a maximum value of 1, then the minimum mean square error (MMSE) linear estimate of $x[n+m]$ given the value of $x[n]$ is [17]

$$\widehat{x[n+m]} = \bar{x} + \phi_{xx}[m](x[n] - \bar{x}) \tag{14.52}$$

Thus, if $\phi_{xx}[1] = 1$ (maximum correlation), the value of $x[n]$ at index n_0 is also the best MMSE linear estimate of what its value will be at time $n_0 + 1$. If $\phi_{xx}[1] = 0$, then $x[n_0]$ provides no information regarding the value of $x[n_0 + 1]$, so the best estimate is simply the mean $\bar{x}$. Intermediate values of $\phi_{xx}[m]$ indicate that the estimated value at $n_0 + m$ will be partially related to, and partially independent of, the value at n_0.

The auto- and cross-correlation functions have numerous applications in radar signal processing. As discussed in Chapter 20, the autocorrelation and cross-correlation functions are integral to understanding matched filtering and signal detection. They are also important in designing a variety of adaptive algorithms such as adaptive beamforming and STAP.

The DTFT of a random process does not exist mathematically. Instead, frequency domain representations of a random process are obtained by considering the *power spectrum* $S_{xx}(\hat{f})$, which is the DTFT of the autocorrelation function. The symmetry properties of the autocorrelation function are such that the power spectrum is guaranteed to be real and nonnegative for all frequency. Thus, it is typically interpreted as a measure of the relative weight of various frequency components in the random process. For instance, if $\phi_{xx}[m] = \overline{x^2}$ for all m, then $S_{xx}(\hat{f}) = 2\pi\overline{x^2}\delta_D(\hat{f})$, an impulse at the origin. This indicates that the only frequency component is the direct current (DC), component, so that the random process is actually a constant for all m. A more important and realistic example is given next.

14.7.3 White Noise

Many random processes in radar are modeled as zero mean *white noise*. In many cases the noise is also assumed to have a Gaussian PDF. This model is common because it is a realistic model for such ubiquitous signals as thermal noise.

White noise is any random process whose autocorrelation function is of the form

$$\phi_{xx}[m] = \sigma_n^2\delta[m] \quad \text{(white noise)} \tag{14.53}$$

Note that this implies a zero mean process. It follows that the power spectrum is a constant for all frequencies:

$$S_{xx}(\hat{f}) = \sigma_n^2 \quad \text{(white noise)} \tag{14.54}$$

Equation (14.54) states that all frequency components are present in the random process with equal weight. Such a process is called "white noise" in analogy to white light, which contains all colors in equal measure. Because the autocorrelation function is zero for all m except $m = 0$, a white noise process is "totally random" in that knowing the value of a white noise process at one index provides no information predicting its value at any other index.

14.7.4 Time Averages

In processing real data, the precise PDFs and expected values are not known but instead are estimated from the data by computing various averages of a measured data sequence $x[n]$ that is assumed to be a realization of a random process. Given an L-point finite data sequence, common estimators are as follows:[14]

$$\begin{aligned}
&\text{mean: } \hat{m}_x = \frac{1}{L}\sum_{n=0}^{L-1} x[n] \\
&\text{mean square: } \hat{p}_x = \frac{1}{L}\sum_{n=0}^{L-1} |x[n]|^2 \\
&\text{autocorrelation: } \hat{\phi}_{xx}[m] = \frac{1}{L}\sum_{n=0}^{L-1} x[n]x^*[n+m]
\end{aligned} \tag{14.55}$$

The cross-correlation is estimated analogously to the autocorrelation. Also, alternative normalizations of the cross- and autocorrelation are sometimes used that differ in handling the end effects for finite data sequences; see [2] for details.

Note that each of these quantities is the sum of random variables or products of random variables, so each estimated moment is itself a random variable with some PDF and some mean, mean square, and variance. An estimator is *asymptotically unbiased* if the expected value of the estimated moment equals the true expected value of the corresponding moment of the random process as the amount of data available for the estimate becomes infinite, that is, as $L \to \infty$. It is *consistent* if it is asymptotically unbiased and the variance of the estimate approaches zero as $L \to \infty$. Each of the previous estimators is unbiased, and the mean and mean square estimators are consistent (provided that $\sigma_x^2 < \infty$). For example, $E\{\hat{m}_x\} = m_x$ and $\lim_{L\to\infty} \{\text{var}(\hat{m}_x)\} = 0$.

[14] These estimators are valid only if the random process is *wide sense stationary* and *ergodic*. The former property says that the mean and mean square are independent of time and that the autocorrelation depends only on the lag m, not the particular absolute times n and $n + m$. The latter property guarantees that time averages of a single realization of the random process equal the corresponding ensemble averages. See [17] for a very accessible introduction to these topics.

14.8 INTEGRATION

Integration is the process of combining multiple samples of a signal, each contaminated by noise or other interference, to "average down the noise" and obtain a single combined signal-plus-noise sample that has a higher SNR than the individual samples. This high-SNR sample is then used in a detection or tracking algorithm to obtain better performance than was possible with the individual low-SNR samples. Integration can be either coherent, meaning that the signal phase information is used, or noncoherent, meaning that only the magnitude of the signal is processed.

14.8.1 Coherent Integration

Assume a measured signal x consists of a signal component, $Ae^{j\phi}$, and a noise component, w. For example, x could be a single range sample from a single pulse, or a particular Doppler spectrum sample $X[k]$, or a single pixel from a complex SAR image. If the measurement that gave x is repeated N times, a sequence of measurements $x[n]$ can be formed. The noise $w[n]$ in each sample is assumed independent and identically distributed (i.i.d.) with variance σ_w^2, but the signal component is the same in each sample. The SNR of each individual sample is therefore $SNR_1 = A^2 / \sigma_w^2$. Now consider the SNR of the integrated signal

$$\begin{aligned} x_N &= \sum_{n=0}^{N-1} (x[n] + w[n]) = \sum_{n=0}^{N-1} (Ae^{j\phi} + w[n]) \\ &= NAe^{j\phi} + \sum_{n=0}^{N-1} w[n] \end{aligned} \tag{14.56}$$

Because the signal samples all add in phase with one another, the amplitude of the coherently integrated signal component is now NA and the signal power will be $(NA)^2$. The power in the noise component is

$$\begin{aligned} \mathrm{E}\left\{\left|\sum_{n=0}^{N-1} w[n]\right|^2\right\} &= \mathrm{E}\left\{\left(\sum_{n=0}^{N-1} w[n]\right)\left(\sum_{l=0}^{N-1} w^*[l]\right)\right\} \\ &= \sum_{n=0}^{N-1}\sum_{l=0}^{N-1} \mathrm{E}\{w[n]w^*[l]\} \\ &= N\sigma_w^2 \end{aligned} \tag{14.57}$$

where in the last step the common assumptions that the noise process w is zero mean, white, and stationary has been used for simplicity. The SNR of the coherently integrated data is therefore

$$SNR_N = \frac{N^2 A^2}{N\sigma_w^2} = N\frac{A^2}{\sigma_w^2} = N \cdot SNR_1 \tag{14.58}$$

Thus, coherent integration of N data samples increases the SNR by a factor of N. This increase is called the *integration gain*.

Many radar signal processing operations are coherent integration in disguise. Suppose the signal component of the series of samples consists of echoes from a target moving with constant radial velocity, resulting in a Doppler frequency of $\hat{f}_d$ on a normalized frequency scale. The signal samples are then

$$x[n] = Ae^{j(2\pi \hat{f}_d n + \phi)}, \; n = 0, \ldots, N-1 \tag{14.59}$$

If the data are coherently integrated as is, the signal samples will not combine in phase with one another (except when $\hat{f}_d = 0$), and the factor of N gain in signal amplitude will not be realized. However, if the sum

$$\begin{aligned} x &= \sum_{n=0}^{N-1} x[n] e^{-j2\pi \hat{f}_d n} = \sum_{n=0}^{N-1} (Ae^{j\phi} e^{+j2\pi \hat{f}_d n}) e^{-j2\pi \hat{f}_d n} \\ &= Ae^{j\phi} \sum_{n=0}^{N-1} (1) = NAe^{j\phi} \end{aligned} \tag{14.60}$$

is formed, then the signal component of the sum is the same as in equation (14.56). The integration of equation (14.60) includes a phase rotation or phase compensation of the data so that the signal components will add in phase. This phase compensation will have no effect on the noise power, so there will again be an N-fold increase in SNR. Obviously, it is necessary to know $\hat{f}_d$ or, equivalently, to repeat the computation for several values of $\hat{f}_d$ to search for the one that works best, to achieve this gain.

Finally, suppose that $\hat{f}_d$ happens to be of the form k/K for some integer k. Then the summation in the first step of (14.60) is identical to the definition of the DFT given in (14.25). Thus, the DFT implements ideal coherent integration of complex sinusoids with frequencies equal to the DFT frequencies. Furthermore, a K-point DFT effectively computes K coherent integrations at once, each matched to a different signal frequency. The SNR of the DFT sample that actually corresponds to an input signal frequency will be N times that of the input data. In this manner, the DFT provides a mechanism to test multiple candidate frequencies to maximize the integration gain of an input signal; the FFT algorithm enables this search to be done very quickly.[15]

14.8.2 Noncoherent Integration

Noncoherent integration does not use phase information in the data. Thus, (14.56) is replaced by

$$|x| = \sum_{n=0}^{N-1} |x[n] + w[n]| = \sum_{n=0}^{N-1} |Ae^{j\phi} + w[n]| \tag{14.61}$$

or

$$|x|^2 = \sum_{n=0}^{N-1} |x[n] + w[n]\,|^2 = \sum_{n=0}^{N-1} |Ae^{j\phi} + w[n]|^2 \tag{14.62}$$

In these equations, $x[n]$ and $w[n]$ are the outputs of a coherent receiver, so they are complex (magnitude and phase) values. However, in noncoherent integration the phase information is discarded by taking the magnitude of the data.

Equation (14.61) is called noncoherent integration with a *linear* detector, while (14.62) uses a *square law* detector. Other detector laws (e.g., logarithmic) are also sometimes used. Noncoherent integration also provides an integration gain in SNR, but it is much more difficult to compute because of the cross-products between signal and noise implied by

[15] In older systems, multiple Doppler frequencies were tested for the presence of a target by subdividing the Doppler spectrum using an explicit bank of band-pass filters, or *filterbank*, within a detector at the output of each filter. The relationship between DFTs and filterbanks is discussed in Chapter 17.

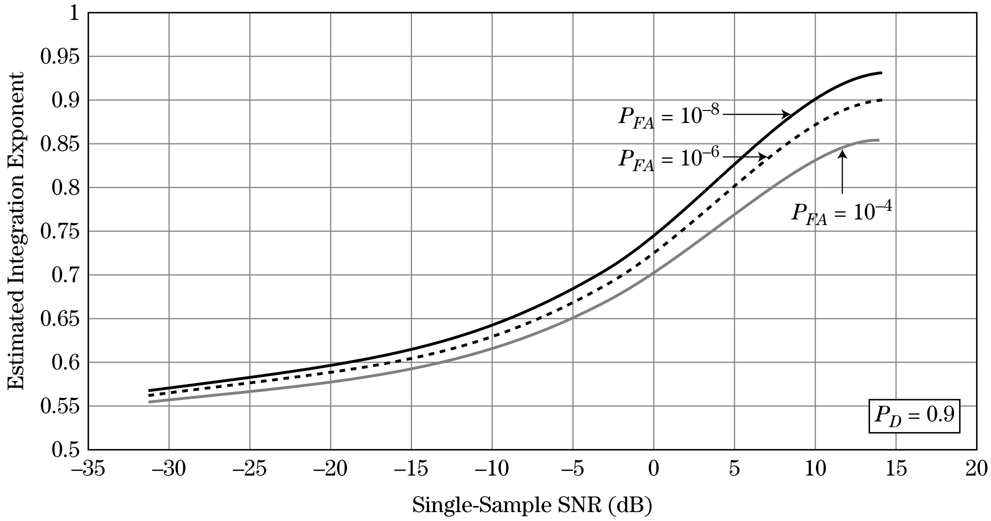


FIGURE 14-28 ■ Estimated noncoherent integration exponent for detection of a constant in complex Gaussian noise with a linear detector. Approximation based on Albersheim's equation [18].

the magnitude operation. The result depends on the particular statistics of the interference and signal and on the detector law. Noncoherent integration gain is usually quantified by computing the preintegration SNR required such that, after integration of N samples, the integrated sample x meets specified performance requirements such as particular probabilities of detection and false alarm, or angular tracking accuracy. The ratio of the predetection SNR required when there is no integration to that required when N samples are integrated is the integration gain. The result is usually expressed in the form of an integration exponent, that is, as N^{α} for some exponent α. For coherent integration, $\alpha = 1$.

Figure 14-28 shows an estimate of the noncoherent integration exponent as a function of SNR for the radar detection problem. Albersheim's equation [18], which provides an estimate of the single-sample SNR needed to achieve specified detection probabilities when N samples are noncoherently integrated with a linear detector, was adapted to estimate α for the case where a detection probability of 90% and several values of false alarm probability were specified.[16] The noncoherent integration is in the range of 0.85 to 0.95 for high single-sample SNRs but falls to nearly $\sqrt{N}$ for very low single-sample SNRs. Lower false alarm probabilities or higher detection probabilities produce still higher values of α at high SNR, but the low-SNR limit remains approximately 0.5. Thus, noncoherent integration is less efficient than coherent integration. This result should be expected because, in discarding the data phase, noncoherent integration does not take advantage of all of the information in the data.

14.9 | CORRELATION AS A SIGNAL PROCESSING OPERATION

The *deterministic cross-correlation* of two signals $x[n]$ and $y[n]$ is defined as

$$c_{xy}[m] = \sum_{n=-\infty}^{+\infty} x[n]\, y^*[n+m] \tag{14.63}$$

[16] See Chapters 3 and 15 for a discussion of Albersheim's equation.

Analogously to the statistical correlation, the index m is called the correlation *lag*. If $x[n] = y[n]$, $c_{xx}[m]$ is called the deterministic *autocorrelation* of $x[n]$. Equation (14.63) is the deterministic equivalent of the statistical concept of correlation for two stationary random processes discussed in Section 14.7.2. Deterministic correlation is clearly a linear operation; if $s[n] = x[n] + w[n]$, $c_{sx}[m] = c_{xx}[m] + c_{wx}[m]$. Note that

$$c_{xx}[0] = \sum_{n=-\infty}^{+\infty} x[n]\,x^*[n] = \sum_{n=-\infty}^{+\infty} |x[n]|^2 = E_x \tag{14.64}$$

where E_x is the energy in the signal $x[n]$. Furthermore, it is shown in many DSP and random process texts that the $m = 0$ lag is the peak of the autocorrelation function:

$$c_{xx}[m] \le c_{xx}[0] \tag{14.65}$$

This property does not hold for the cross-correlation between two different signals. Another important property of the cross-correlation function is the symmetry property

$$c_{xy}[m] = c_{yx}^*[-m] \tag{14.66}$$

which reduces to $c_{xx}[m] = c_{xx}^*[-m]$ in the autocorrelation case.

As with the statistical cross-correlation, an alternative definition of the deterministic cross-correlation is

$$\bar{c}_{xy}[m] = \sum_{n=-\infty}^{+\infty} x[n]\,y^*[n-m] \tag{14.67}$$

The two definitions are related by $\bar{c}_{xy}[m] = c_{xy}[-m]$. Again, readers must be alert to differences in definitions when comparing results from different sources.

Correlation is a measurement of similarity between two signals. Its primary use in signal processing is to detect the presence of a known signal in high noise and to estimate the known signal's location in the noisy data. Figure 14-29 illustrates this idea. Figure 14-29a shows a noisy signal $s[n]$ composed of zero mean white noise $w[n]$ and an 18-sample square pulse $x[n]$ of amplitude 2 located between $n = 10$ and $n = 27$. Figure 14-29b shows the cross-correlation $c_{sx}[m]$ of the noise signal and a replica of the noise-free pulse. The peak at lag 10 indicates a high correlation between the reference pulse and the samples of $s[n]$ that begin at $n = 10$; this is exactly the portion of $s[n]$ where the square pulse is located in the noise.

Correlation is another signal processing operation that can be related to coherent integration. Let $x[n] = A[n]e^{j\phi[n]}$ be an arbitrary complex-valued signal. The autocorrelation peak occurs at lag $m = 0$. From (14.64),

$$\begin{aligned} c_{xx}[0] &= \sum_{n=-\infty}^{+\infty} x[n]\,x^*[n] = \sum_{n=-\infty}^{+\infty} \left\{A[n]e^{j\phi[n]}\right\}\left\{A^*[n]e^{-j\phi[n]}\right\} \\ &= \sum_{n=-\infty}^{+\infty} |A[n]|^2 = E_x \end{aligned} \tag{14.68}$$

Equation (14.68) shows that the autocorrelation peak occurs because the signal is multiplied by its own conjugate, forming a zero-phase product. Each term in the sum is thus real and adds in phase to produce the autocorrelation peak. In other words, the $m = 0$ lag coherently integrates the samples of $x[n]$.

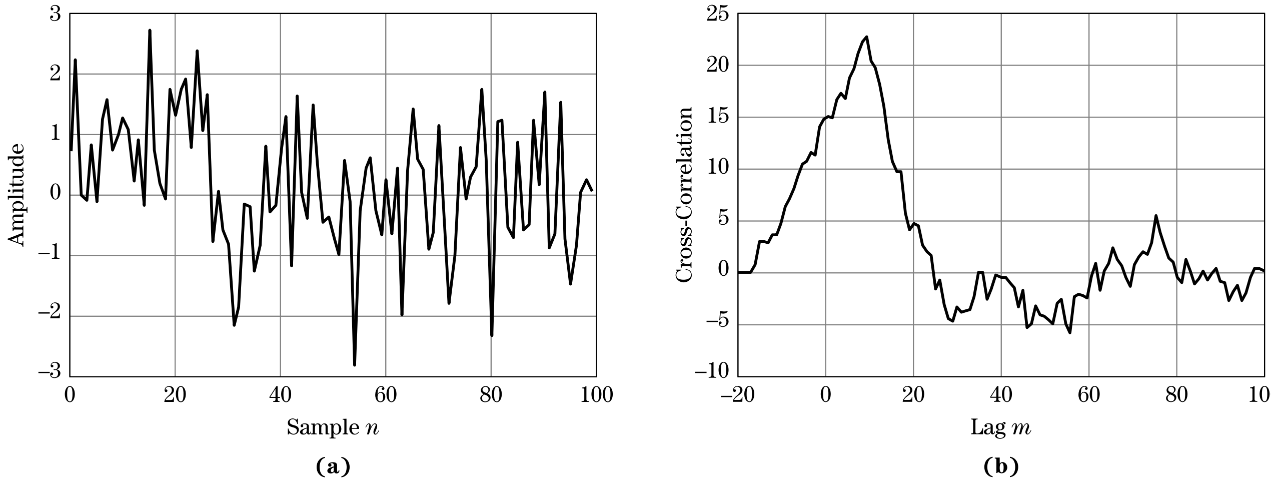


FIGURE 14-29 ■ Use of cross-correlation to locate a signal buried in noise. (a) Signal containing a square pulse in samples 10 through 17, plus noise in all samples. (b) Cross-correlation of the noisy data with a replica of the pulse.

14.10 MATCHED FILTERS

Chapters 15 and 18 show that the signal-to-noise ratio is the major determinant of performance in such applications as detection and tracking. In detection, for example, the probability of detection increases monotonically with SNR for a fixed probability of false alarm. In tracking, the error variance of track measurements (whether in range, angle, or Doppler) is inversely proportional to SNR; thus, as SNR increases, the error variance decreases. Consequently, maximizing the SNR of radar data is a fundamental goal of radar design, affecting everything from transmitter power and antenna size to the signal processing operations applied to the data.

The radar signal processor can contribute to the maximization of SNR by filtering or transforming the data in some way that removes the noise while reinforcing the desired signal component. This intuitive notion can be formalized by determining the LSI filter that will maximize the SNR. This is an optimization problem that can be stated and solved in many ways, all of which of course lead to the same answer. The approach here is based on the vector notation for signals and for convolution from Sections 14.2.3 and 14.6. Recall the vector expression for the output of an FIR filter in equation (14.38). The power in the output sample $y[n]$ is

$$|y|^2 = y^* y^T = \boldsymbol{H}^H \boldsymbol{X}^* \boldsymbol{X}^T \boldsymbol{H} \tag{14.69}$$

where the superscript H indicates a Hermitian (conjugate transpose) operation and the subscript n has been dropped for notational brevity. Now consider a finite-length signal consisting of signal and noise components: $\boldsymbol{X} = \boldsymbol{S} + \boldsymbol{W}$. The goal is to find the filter coefficient vector $\boldsymbol{H}$ that maximizes the SNR. Since the filter is linear, its effect on the signal power and noise power can be computed separately. The signal component power is, from (14.69), just $\boldsymbol{H}^H \boldsymbol{S}^* \boldsymbol{S}^T \boldsymbol{H}$. The noise power at the particular time index n is, similarly, $\boldsymbol{H}^H \boldsymbol{W}^* \boldsymbol{W}^T \boldsymbol{H}$. However, the average noise power is much more relevant. The expected value of the filtered noise sample is

$$|y|^2_{noise} = \boldsymbol{H}^H \mathbf{R_I} \boldsymbol{H} \tag{14.70}$$

where the linearity of the expected value operation has been used to bring it inside the vector products with $\boldsymbol{H}$ and the noise *covariance matrix* has been defined as

$$\mathbf{R_I} = \mathrm{E}\left\{\boldsymbol{W} * \boldsymbol{W}^T\right\} \tag{14.71}$$

Covariance matrices have several special properties. For instance, they are Toeplitz matrices, meaning they are constant along each diagonal, they have Hermitian symmetry, meaning $\mathbf{R_I}^H = \mathbf{R_I}$, and they are positive definite [19].

The SNR is just the ratio of the signal and noise powers at the filter output:

$$SNR = \frac{\boldsymbol{H}^H \boldsymbol{S}^* \boldsymbol{S}^T \boldsymbol{H}}{\boldsymbol{H}^H \mathbf{R_I} \boldsymbol{H}} \tag{14.72}$$

To find the filter coefficient vector $\boldsymbol{H}$ that maximizes SNR, the Schwarz inequality is needed in the form

$$\left|\boldsymbol{P}^H \boldsymbol{Q}\right|^2 \le \|\boldsymbol{P}\|^2 \|\boldsymbol{Q}\|^2 \tag{14.73}$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are two arbitrary vectors, and the norm of $\boldsymbol{P}$ is defined as $||\boldsymbol{P}|| = \boldsymbol{P}^H \boldsymbol{P}$. Because $\mathbf{R_I}$ is positive definite it can be factored as follows:

$$\mathbf{R_I} = \mathbf{A}^H \mathbf{A} \tag{14.74}$$

Contriving to choose $\boldsymbol{P} = \mathbf{A}\boldsymbol{H}$ and $\boldsymbol{Q} = (\mathbf{A}^H)^{-1}\boldsymbol{S}^*$ in the Schwarz inequality gives $\boldsymbol{P}^H \boldsymbol{Q} = \boldsymbol{H}^H \boldsymbol{S}^*$ and, with some basic matrix algebra, the Schwarz inequality gives

$$\begin{aligned} \boldsymbol{H}^H \boldsymbol{S}^* \boldsymbol{S}^T \boldsymbol{H} &\le \|\mathbf{A}\boldsymbol{H}\|^2 \left\| \left(\mathbf{A}^H\right)^{-1} \boldsymbol{S}^* \right\|^2 \\ &= \left(\boldsymbol{H}^H \mathbf{R_I} \boldsymbol{H}\right) \left(\boldsymbol{S}^T \mathrm{R}_{\mathbf{I}}^{-1} \boldsymbol{S}^*\right) \end{aligned} \tag{14.75}$$

Dividing both sides by $\boldsymbol{H}^H \mathbf{R_I} \boldsymbol{H}$ gives

$$\frac{\boldsymbol{H}^H \boldsymbol{S}^* \boldsymbol{S}^T \boldsymbol{H}}{\boldsymbol{H}^H \mathbf{R_I} \boldsymbol{H}} = \chi \le \boldsymbol{S}^T \mathbf{R_I}^{-1} \boldsymbol{S}^* \tag{14.76}$$

What choice of $\boldsymbol{H}$ will achieve this maximum SNR? The maximum is achieved when $\boldsymbol{P} = k\boldsymbol{Q}$ for some k. This condition becomes $\mathbf{A}\boldsymbol{H} = k\left(\mathbf{A}^H\right)^{-1} \boldsymbol{S}^*$, so that

$$\boldsymbol{H} = k\mathbf{R_I}^{-1}\, \boldsymbol{S}^* \tag{14.77}$$

The effect of $\mathbf{R_I}^{-1}$ is to whiten the interference prior to matched filtering with $\boldsymbol{S}^*$.

Equation (14.77) is of fundamental importance in radar signal processing; it is the basis not only of waveform matched filtering but also of many techniques in adaptive beamforming, ground moving target indication, and space-time adaptive processing.

While stationary interference has been assumed, the noise has not been assumed to be white. Thus, equation (14.77) is fairly general. If in fact the noise is white, an important special case, then $\mathbf{R_I} = \sigma_n^2 \mathbf{I}$ where $\mathbf{I}$ is an identity matrix. If k is chosen to equal $1/\sigma_n^2$, then

$$\boldsymbol{H} = \boldsymbol{S}^* \tag{14.78}$$

Thus, the optimum filter coefficients when the interference is white noise are just the conjugate of the samples of the desired signal $\boldsymbol{S}$. In this case, $\boldsymbol{H}$ is called the *matched filter*, because the coefficients are "matched" to the signal the filter is designed to detect. If that signal changes, then the filter coefficients must also be changed to maintain the maximum possible SNR.

The matched filter can also be derived for continuous-time signals using the appropriate version of the Schwarz inequality. The result is that the impulse response of the filter should satisfy

$$h(t) = ks^*(T_M - t) \tag{14.79}$$

where T_M is the time instant at which the SNR is maximized [3]. If the signal s is of finite length τ, then $T_M \geq \tau$ for causality. The derivation of equation (14.79) does not assume a finite impulse response; however, $h(t)$ will be infinite only if the desired signal s to which it is matched is infinite in duration.

Filtering a signal $s(t)$ with its matched filter from (14.79) corresponds to computing the continuous-time autocorrelation function $c_{ss}(l)$. To see this, write the convolution of $s(t)$ and its matched filter impulse response:

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} h(u)s(t-u)du \\ &= \int_{-\infty}^{\infty} s^*(T_M - u)s(t-u)du \\ &= \int_{-\infty}^{\infty} s(v)s^*(v + T_M - t)dv \end{aligned} \tag{14.80}$$

The last step, obtained with the substitution $v = t - u$, is easily recognized as the autocorrelation of s evaluated at lag $T_M - t$. Thus, the signal component of the time waveform at the output of the matched filter is actually the autocorrelation function of that signal. The matched filter peak $y(T_M)$ is then $c_{ss}(0) = E_s$, where E_s is the total energy in the signal s. As previously discussed, the zero autocorrelation lag is equivalent to coherently integrating the signal s. The output of the vector matched filter $y = \boldsymbol{H}^T\boldsymbol{S}$ with the optimum coefficients $\boldsymbol{H} = \boldsymbol{S}^*$ is also easily seen to be equivalent to coherently integrating the signal vector samples, producing an output equal to the signal vector energy E_s. Once again, coherent integration is at the heart of a key signal processing operation.

Equation (14.76) shows that, in the case of white noise, the actual value of the peak SNR is

$$SNR_{\max} = \frac{E_s}{\sigma_n^2} \tag{14.81}$$

The same result is obtained in the continuous-time case. Equation (14.81) shows the important fact that, when the interference is white and a matched filter is used, the maximum SNR depends only on the signal energy, not on its detailed shape. This fact will be critical to the development of pulse compression and to its use in gaining independent control of detection performance and range resolution in Chapter 20.

There is a direct discrete-time equivalent to the analog matched filter of (14.79), which in turn gives a discrete correlation equivalent to (14.80):

$$\begin{aligned} h[n] &= ks^*[N_0 - n] \\ y[n] &= \sum_{m=-\infty}^{\infty} h[m]s[n-m] \\ &= \sum_{m=-\infty}^{\infty} s[m]s^*[m + N_0 - n] \end{aligned} \tag{14.82}$$

The discrete version exhibits the same properties (e.g., value of peak output, coherent integration effect) as the continuous version.

14.11 FURTHER READING

While there are many excellent texts devoted to digital signal processing fundamentals, few if any focus on aspects specific to radar and similar sensor systems. At this writing, the most recent text focused specifically on radar signal processing basics is *Fundamentals of Radar Signal Processing* by M. A. Richards [3]. Good background texts in general digital signal processing are too numerous to mention; two of the more successful texts are those by Mitra [1] and Oppenheim and Schafer [2]. Hayes has produced *Schaum's Outline of Digital Signal Processing* [20], which is an excellent concise reference and source of sample problems. Finally, Lyons's *Understanding Digital Signal Processing* [21], offers a very practical-minded review of DSP concepts a well as many "tricks of the trade."

14.12 REFERENCES

[1] Mitra, S.K., *Digital Signal Processing: A Computer-Based Approach*, 2d ed., McGraw-Hill, New York, 2001.

[2] Oppenheim, A.V., and Schafer, R.W., *Discrete-Time Signal Processing*, 3d ed., Prentice-Hall, Englewood Cliffs, NJ, 2009.

[3] Richards, M.A., *Fundamentals of Radar Signal Processing*, McGraw-Hill, New York, 2005.

[4] Bracewell, R.N., *The Fourier Transform and Its Applications*, 3d ed., McGraw-Hill, New York, 1999.

[5] Ercegovec, M., and Lang, T., *Digital Arithmetic*, Morgan Kauffman, San Francisco, CA, 2003.

[6] McClellan, J.H., and Purdy, R.J., "Applications of Digital Signal Processing to Radar," Chapter 5 in *Applications of Digital Signal Processing*, Ed. A.V. Oppenheim, Prentice-Hall, Englewood Cliffs, NJ, 1978.

[7] Papoulis, A., *The Fourier Integral and Its Applications*, McGraw-Hill, New York, 1987.

[8] Harris, F.J., "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proceedings of the IEEE,* vol. 66, no. 1, pp. 51–83, January 1978.

[9] Johnson, D.H., and Dudgeon, D.E., *Array Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[10] Richards, M.A., "Relationship between Temporal and Spatial Frequency in Radar Imaging," Technical Memorandum. July 7, 2007. Available at http://www.radarsp.com.

[11] Jacobsen, E., and Kootsookos, P., "Fast Accurate Frequency Estimators," *IEEE Signal Processing Magazine*, pp. 123–125, May 2007.

[12] Cooley, J.W., and Tukey, J.W., "An Algorithm for the Machine Computation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, pp. 297–301, April 1965.

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

[14] "FFTW [Fastest Fourier Transform in the West]," n.d. Available at http://www.fftw.org.

[15] Shrader, W.W., and Gregers-Hansen, V., "MTI Radar," Chapter 2 in *Radar Handbook*, 3d ed., Ed. M.I. Skolnik, McGraw-Hill, New York, 2008.

[16] Rader, C.M., "A Simple Method for Sampling In-Phase and Quadrature Components," *IEEE Transactions Aerospace and Electronic Systems*, vol. AES-20, no. 6, pp. 821–824, November 1984.

[17] Kay, S., *Intuitive Probability and Random Processes Using MATLAB®* , Springer, New York, 2006.

[18] Albersheim, W.J., "Closed-Form Approximation to Robertson's Detection Characteristics," *Proceedings of the IEEE*, vol. 69, no. 7, p. 839, July 1981.

[19] Watkins, D.S., *Fundamentals of Matrix Computations*, Wiley, New York, 1991.

[20] Hayes, M.H., *Schaum's Outline of Digital Signal Processing*, McGraw-Hill, New York, 1998.

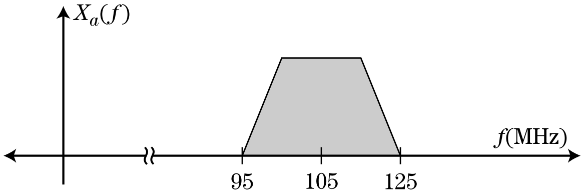
[21] Lyons, R.G., *Understanding Digital Signal Processing*, 2d ed., Prentice-Hall, New York, 2004.

14.13 PROBLEMS

1. Consider an electromagnetic wave with a temporal frequency of 1 GHz, propagating in the $+x$ direction. What is the numerical value of spatial frequency k_x of this wave, in radians per meter, and also in cycles/m?

2. How many pulses must be coherently integrated to produce an integration gain sufficient to increase the detection range for a given target from 5 to 50 miles?

3. In some cases, the spectrum replication property of sampling can be used as a substitute for demodulation. Given a signal $x_a(t)$ with the spectrum shown, what is the lowest sampling rate that will ensure that one of the spectrum replicas is centered at $f = 0$ with no aliasing?



4. Consider a pulsed radar with a fast-time sampling rate equal to the pulse bandwidth $B = 10$ MHz, a PRF of 5 kHz, and a single receiver channel. If the range swath extends from 5 km to 50 km and the duration of the CPI is 4 ms, what is the total number of complex samples in the fast-time–slow-time data matrix?

5. The saturation voltage of an ADC is set at four times the input noise standard deviation, $A_{sat} = 4\sigma$. How many bits, *b,* are required in the ADC to provide an SQNR of at least 70 dB?

6. What is the 3 dB Doppler resolution of a system that collects 40 pulses of data at a PRF of 10 kHz and applies a Hamming window to the slow-time data before computing the Doppler spectrum?

7. Consider a sequence of 20 slow-time data samples collected at a PRF of 2 kHz. If a 1,000-point DFT of this sequence is computed, what is the spacing between DFT frequency samples in Hz?

8. Explicitly compute the SNR loss of equation (14.18) as a function of K for a triangular window of odd length $K + 1$ (so K is even) defined according to

$$w[k] = \begin{cases} 2k/K, & 0 \le k \le K/2 \\ 2 - 2k/K, & K/2 \le k \le K \\ 0. & \text{otherwise} \end{cases}$$

Numerically evaluate the result for $K = 4$ and $K = 20$, and give the results in dB. What is the asymptotic value of the loss in dB as $K \rightarrow \infty$ (give the result in both linear and dB units)? The following fact may be useful (careful about the limits!):

$$\sum_{k=1}^{n} k = n\,(n+1)/2$$

Another hint: Sum just the first half of the triangle, and then use symmetry to get the sum of the whole function, being careful not to double-count any samples.

9. An X-band (10 GHz) pulse-Doppler radar collects a fast-time–slow-time matrix of 30 pulses by 200 range bins per pulse. This is converted to a range–Doppler matrix by applying a Hamming window and then a 64-point fast Fourier transform to each slow-time row. Suppose that there is a target with a constant radial velocity of 30 m/s approaching the radar at a range corresponding to range bin #100. The PRF is 6,000 samples/s. There is no ground clutter. Ignore thermal noise as well. For which FFT sample index k_0 is $|Y[k_0]|$ the largest? What velocity in m/s does this sample correspond to? What is the error between the apparent velocity based on the largest FFT sample, and the actual velocity?

10. Continuing Problem 9, in terms of the window function $w[m]$, what is the peak value of the DTFT (not DFT) of the windowed data in range bin #100, assuming that each slow-time sample has an amplitude of 1 before windowing? What is the numerical value of this peak? (Use a computing environment or spreadsheet program such as MATLAB® or Excel® to compute this value.) Now suppose the peak value of the magnitude of the FFT of the data $|Y[k_0]| = 15.45$. What is the straddle loss in dB?

11. Suppose also that $|Y[k_0 - 1]| = 11.61$ and $|Y[k_0 + 1]| = 14.61$. Use the quadratic interpolation technique of (14.29) to estimate the velocity of the target and the peak amplitude of the DTFT. Compute the loss in the new amplitude value relative to the true amplitude. Compare the new values of velocity error and straddle loss with those found in Problems 9 and 10.

12. One drawback in some situations of the K-point FFT is that it computes all of the DFT samples, even if only one or a few are required. The DFT of (14.25) can be used to compute individual frequency domain samples. In terms of K, what is the maximum value of L such that the number of complex multiplies to compute L samples of the K-point DFT using (14.25) is less than the number of complex multiplies to compute all K DFT samples using the FFT? What is the numerical value of L (must be an integer) for $K = 256$?

13. Fast convolution is to be used to convolve a 1,000-point input signal $x[n]$ with a filter impulse response $h[n]$ that is 100 samples long. What is the length of the filter output $y[n]$? Assuming the FFT size is restricted to be a power of 2, what is the minimum size FFTs that can be used?

14. Median filters can be used to filter “impulse noise,” meaning isolated pulses of energy, from an otherwise relatively constant signal. What is the widest impulse in samples that will be removed by an 11-point median filter?

15. Show that the mean estimator of (14.45) is unbiased and consistent, that is, $\mathrm{E}\{\hat{m}_x\} = m_x$ and $\lim_{L\rightarrow\infty} \{\mathrm{var}\,(\hat{m}_x)\} = 0$.

16. If 100 samples, each having an SNR of 0 dB, are coherently integrated, what will be the integrated SNR? What is the integration gain in dB? Assuming a receiver designed to achieve $P_D = 0.9$ and $P_{FA} = 10^{-8}$, use Figure (14-28) to estimate the number of 0 dB SNR samples that must be noncoherently integrated to achieve the same integration gain.

17. This problem applies the vector matched filter design equation (14.77) to the problem of detecting a sinusoid in zero mean complex white Gaussian noise with power σ_n^2. Let the filter order N be arbitrary. Show that the interference covariance matrix for this case is an identify matrix, $\mathbf{R_I} = \sigma_n^2\mathbf{I}$. Let the desired signal be a pure sinusoid of normalized radian frequency $\hat{\omega}_0$, so that $\boldsymbol{S} = \left[1 \; \exp(j\hat{\omega}_0) \; \exp(j2\hat{\omega}_0) \; \cdots \; \exp(j(N-1)\hat{\omega}_0)\right]^T$. Find the matched filter coefficient vector $\boldsymbol{H}$. Show that when applied to a vector of data $\boldsymbol{X} = \boldsymbol{S} + \boldsymbol{W}$, the filter coherently integrates the signal components of the data.

18. A pulsed radar observes a target in a particular range bin that is approaching with a constant radial velocity of v m/s. In this situation, the slow-time signal observed in that range bin will be of the form $s[m] = A\exp(j\hat{\omega}_0 m), m = 0, 1, \ldots, M-1$ for some number of pulses M. What is the impulse response of the discrete-time matched filter for this signal, assuming it is designed to produce the peak SNR at time $m = M$? For a general input signal $x[m]$, show that the output of this matched filter at the peak is, to within a scale factor, the DTFT of the input evaluated at frequency $\hat{\omega}_0$.