DISCRETE FOURIER TRANSFORM AND FILTER DESIGN

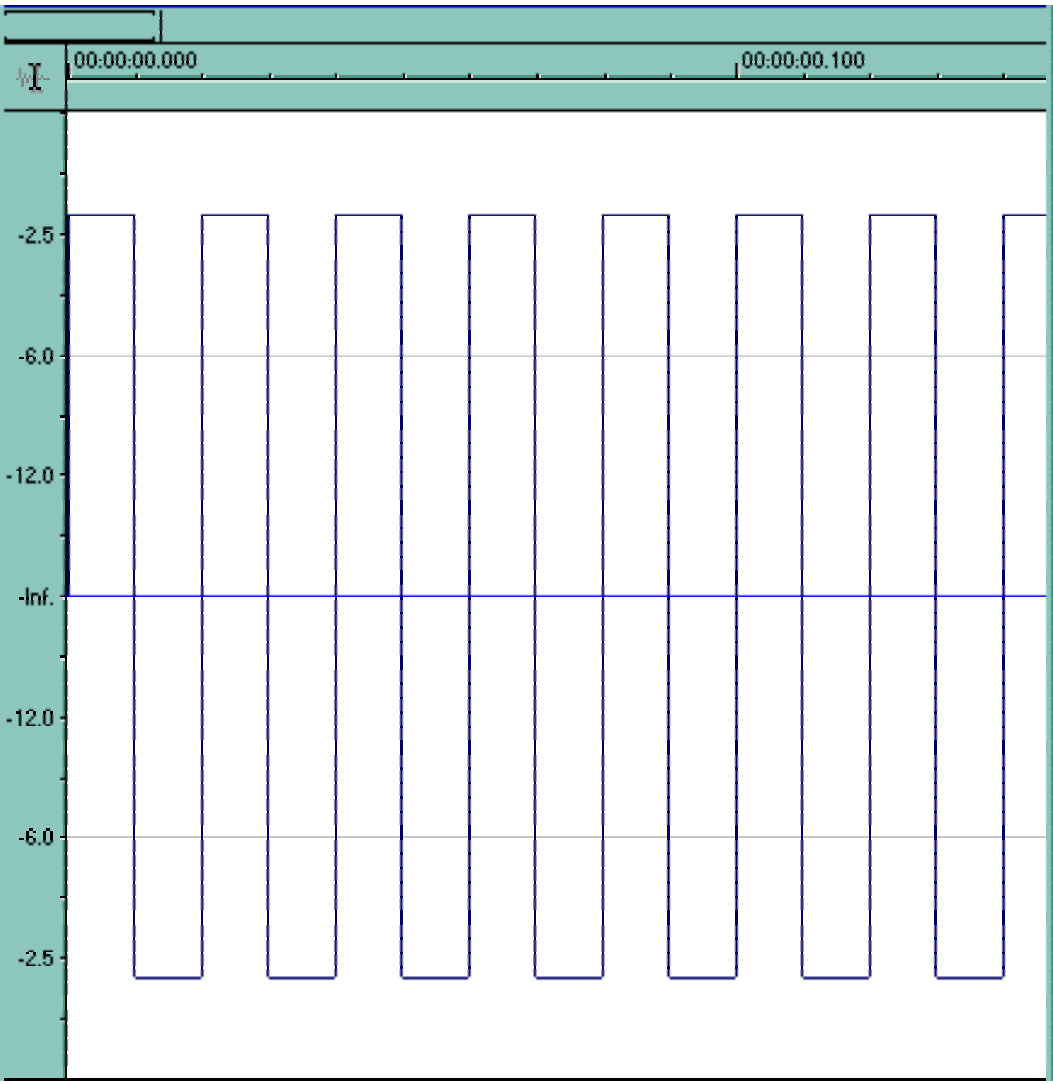
N. C. State University

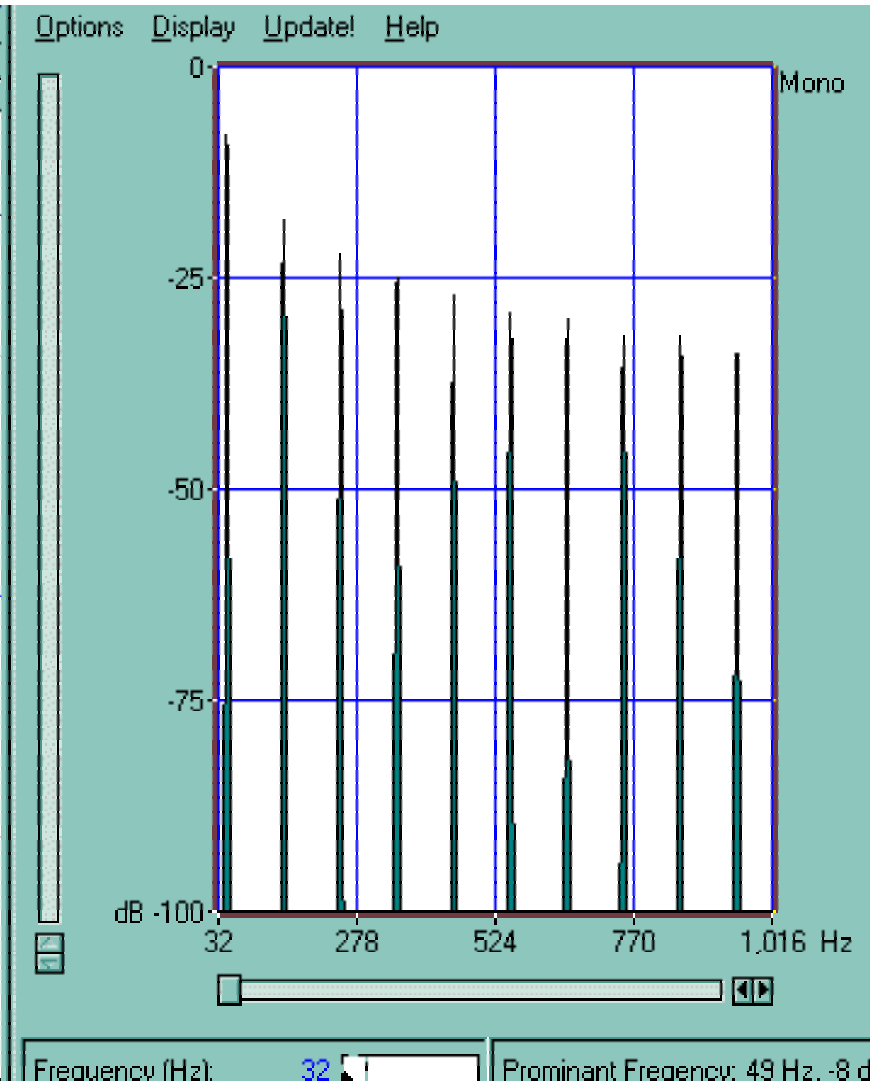
CSC557 ♦ Multimedia Computing and Networking

Fall 2001

Lecture # 03

Spectrum of a Square Wave





Copyrig

Results of Some Filters

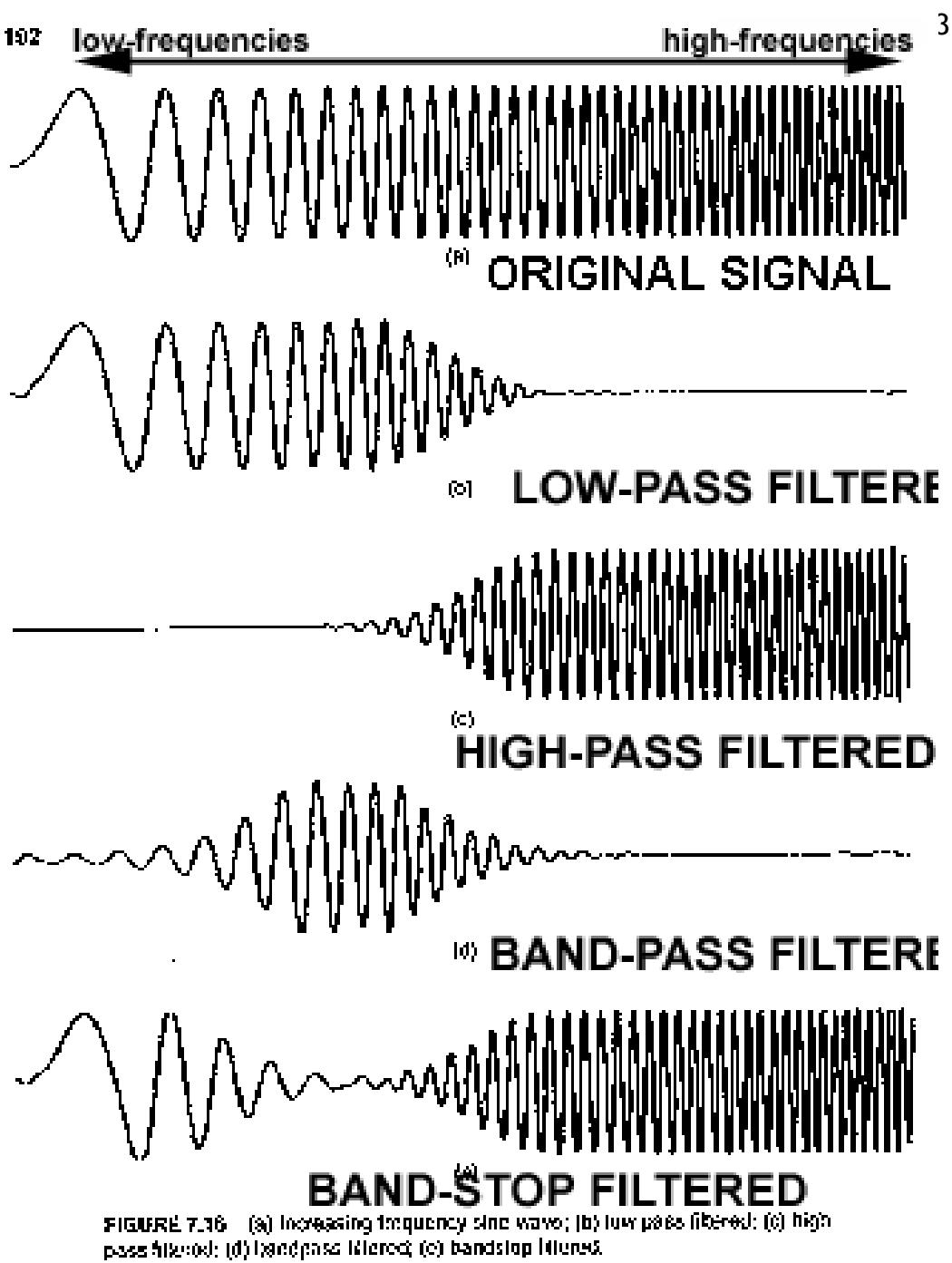


FIGURE 7.36 (a) Increasing frequency sine wave; (b) low pass filtered; (c) high pass filtered; (d) bandpass filtered; (e) bandstop filtered.

Notation

- x[n] and y[n] are discretely sampled signals
- z[n] = x[n] + y[n] = addition of two signals to produce a third signal
- A "system" modifies one signal to produce another

Fundamental Concept of DSP

If

x[n] can be decomposed into a set of additive components

and a system acting on x[n] produces an output y[n]

and the system is applied to each of the additive components individually to produce a set of outputs

Then

adding the outputs together will produce y[n]

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.ec

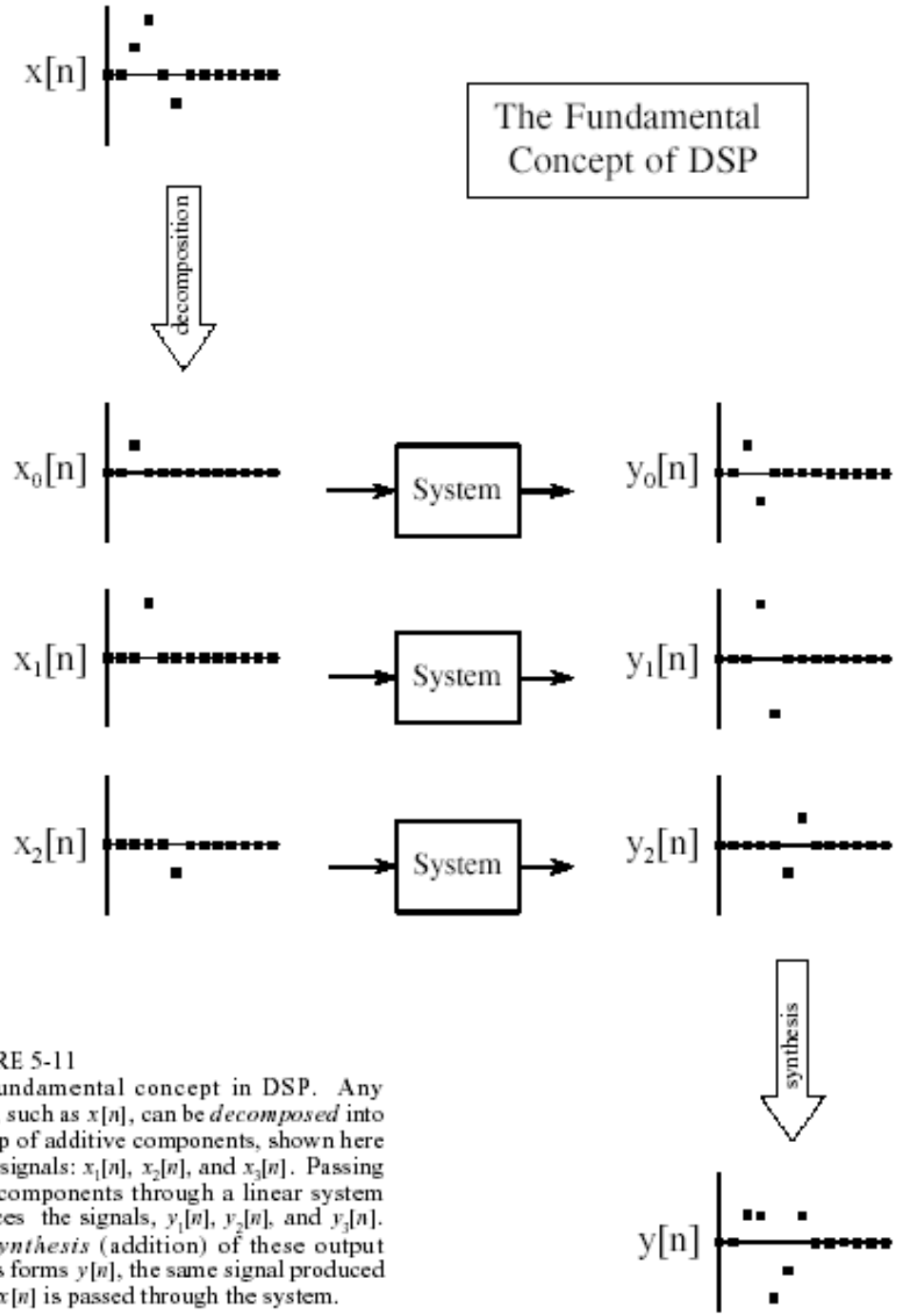


FIGURE 5-11
The fundamental concept in DSP. Any signal, such as $x[n]$, can be *decomposed* into a group of additive components, shown here by the signals: $x_1[n]$, $x_2[n]$, and $x_3[n]$. Passing these components through a linear system produces the signals, $y_1[n]$, $y_2[n]$, and $y_3[n]$. The *synthesis* (addition) of these output signals forms $y[n]$, the same signal produced when $x[n]$ is passed through the system.

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Convolution

- Let a system acting on x[n] be represented as h[n]
- *Convolution* is defined as
  - y[i] = sum from j=0 to M of (x[i-j] • h[j])
- Notation for convolution: y[n] = x[n] * h[n]

Copyright 2001 Douglas S. Reeves (http://reeves.csc.nc

Convolution Example

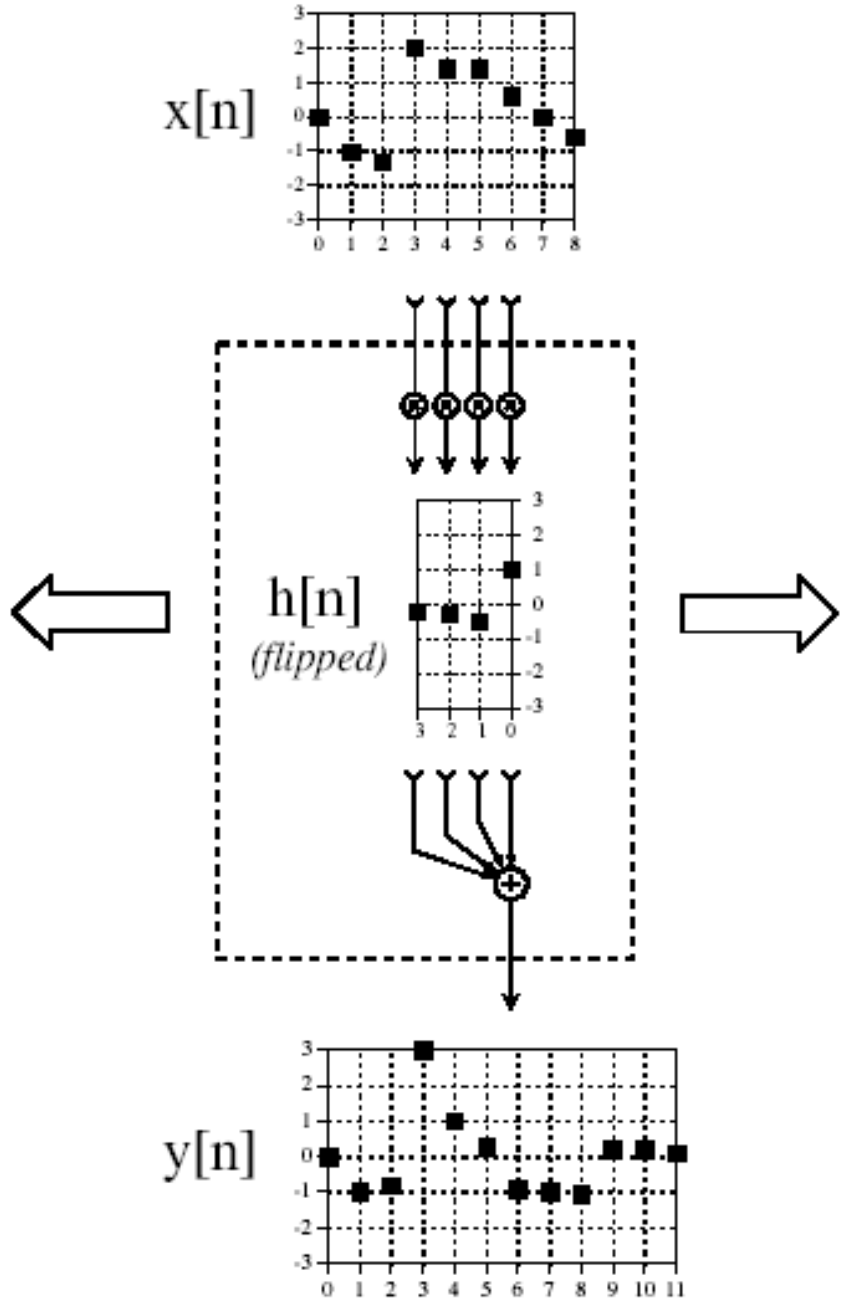


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

FIGURE 6-8

Cont'd

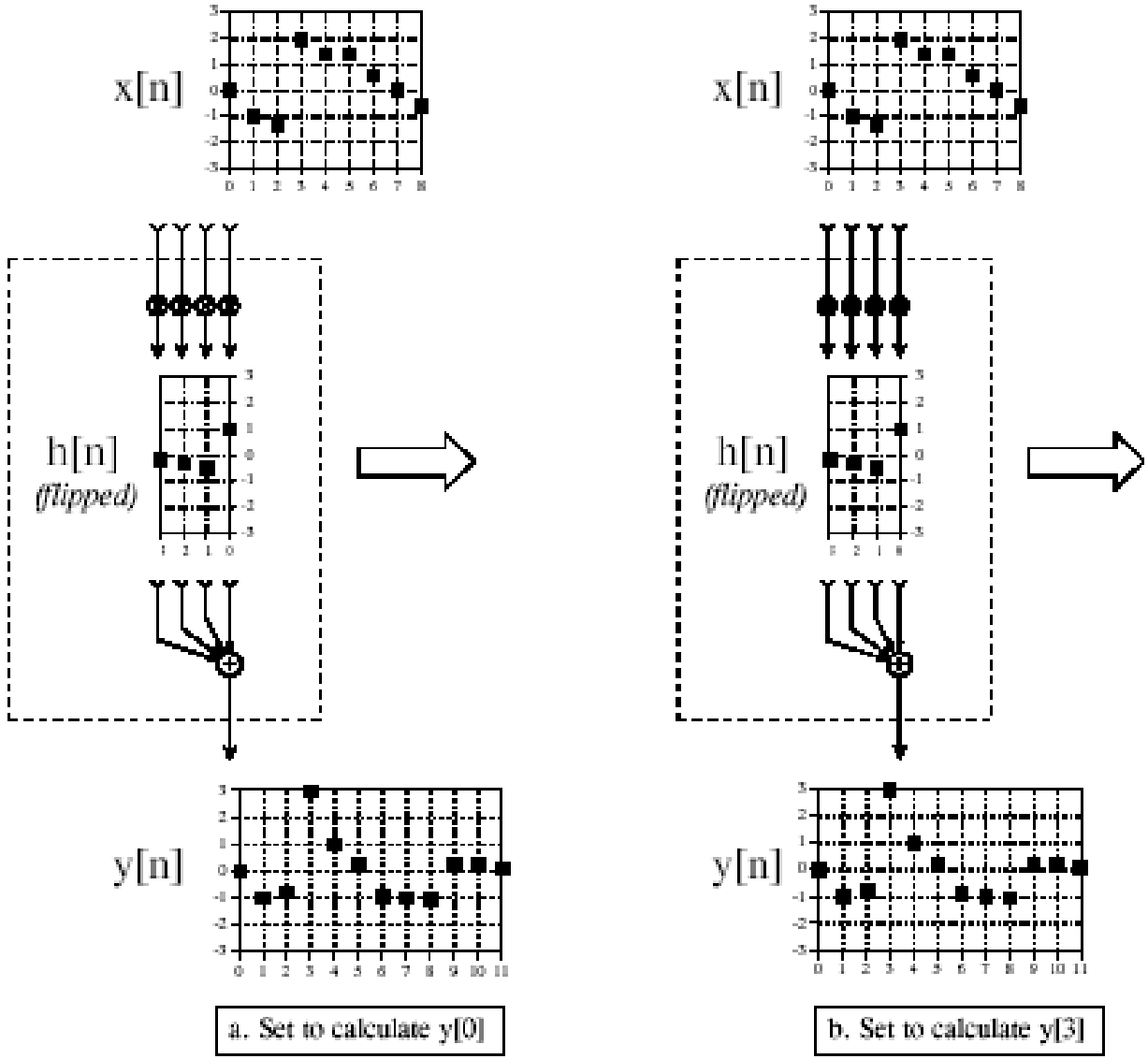


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

FIGURE 6-9
The convolution machine in action. Figures (a) through (d) show the convolution machine set to calculate four different output signal samples, y[0], y[3], y[8], and y[11].

Copyright 2001 Douglas

Cont'd

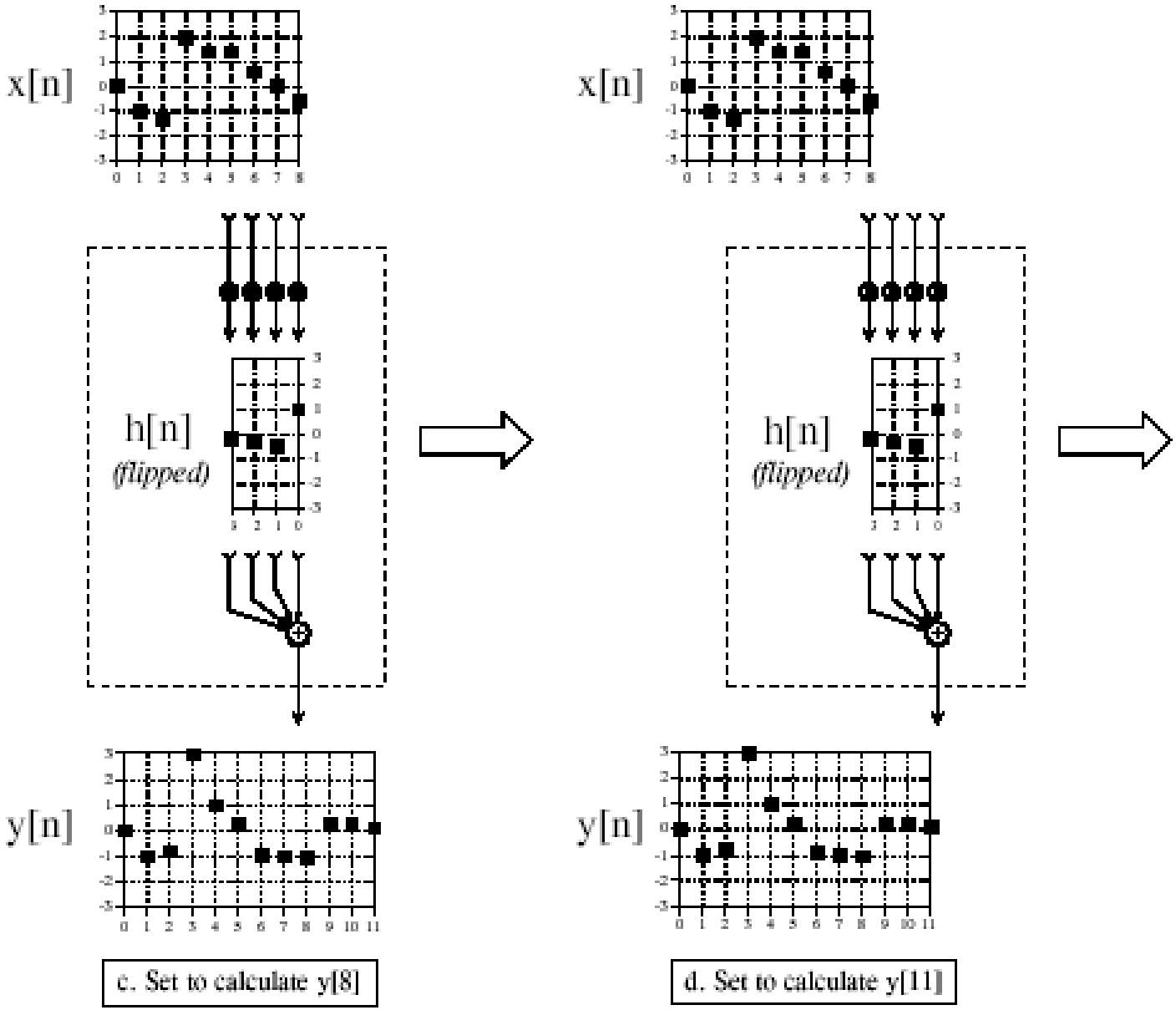


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Figure 6-9 (continued)

Some Examples of Convolution

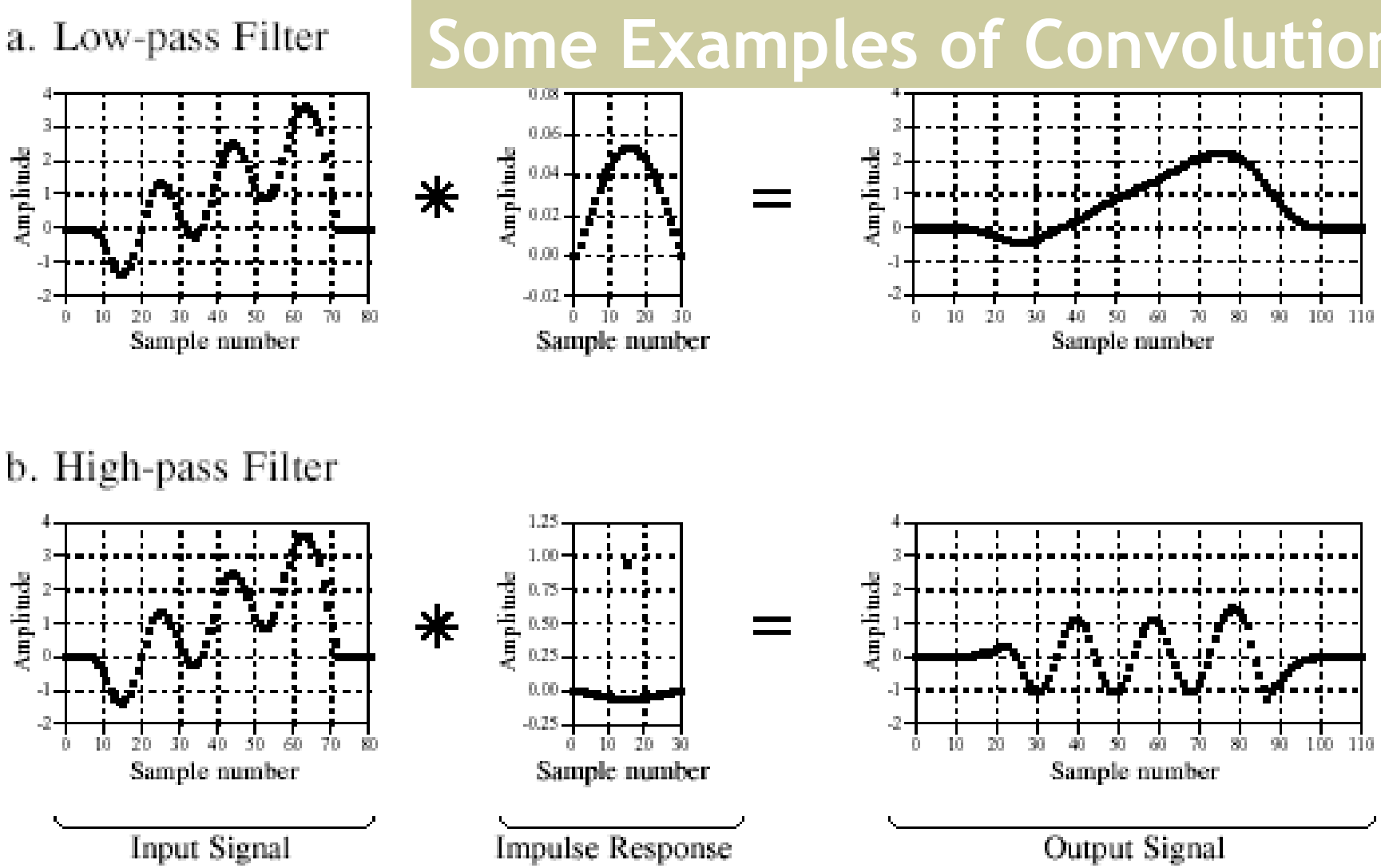


FIGURE 6-3
Examples of low-pass and high-pass filtering using convolution. In this example, the input signal is a few cycles of a sine wave plus a slowly rising ramp. These two components are separated by using properly selected impulse responses.

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Copyrig

Examples (cont'd)

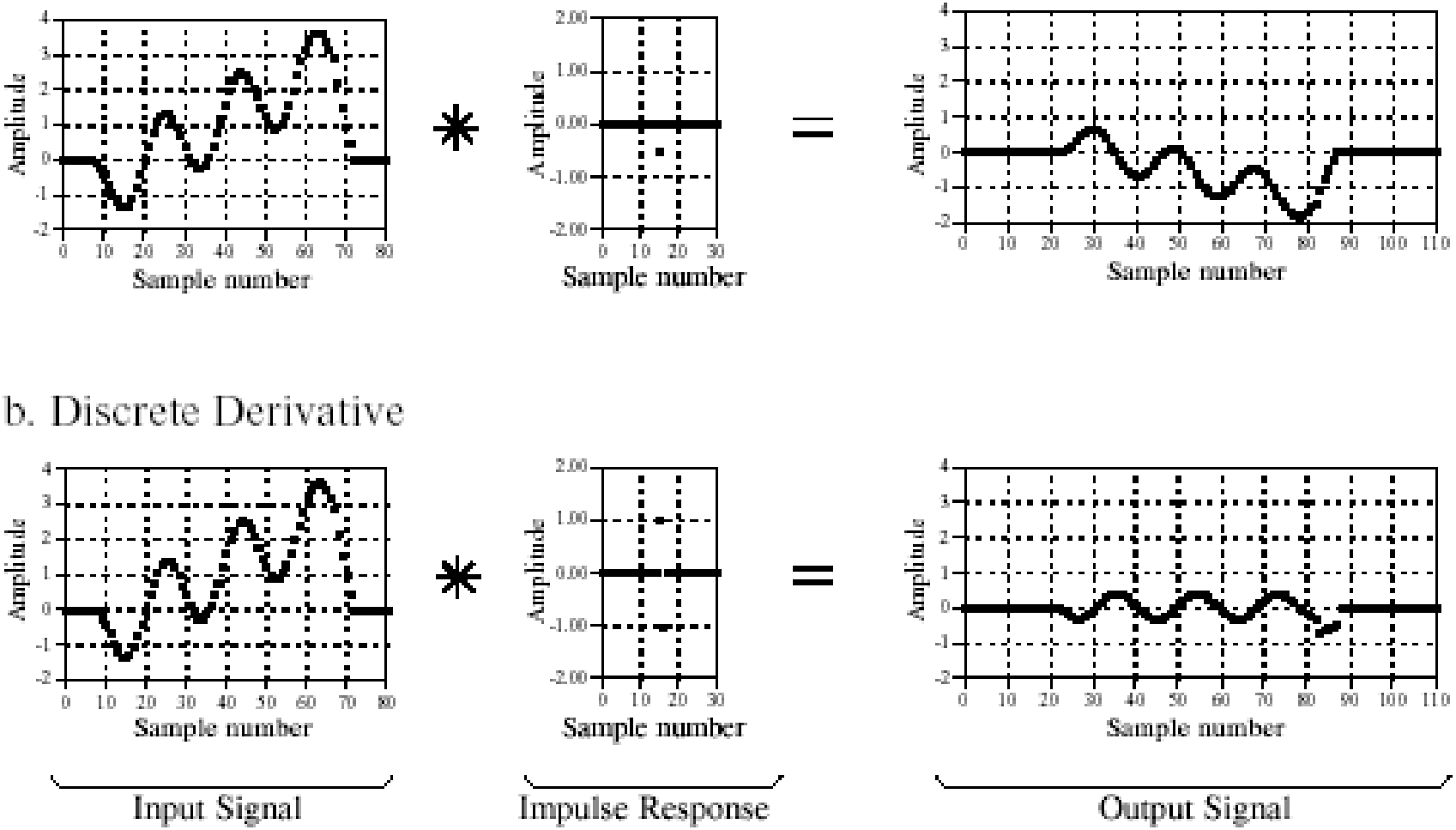


FIGURE 6-4
Examples of signals being processed using convolution. Many signal processing tasks use very simple impulse responses. As shown in these examples, dramatic changes can be achieved with only a few nonzero points.

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Convolution: Boundary Values

- What values do you use for x[i] when i<0 or i>n?
- One solution: "pad with zeros”

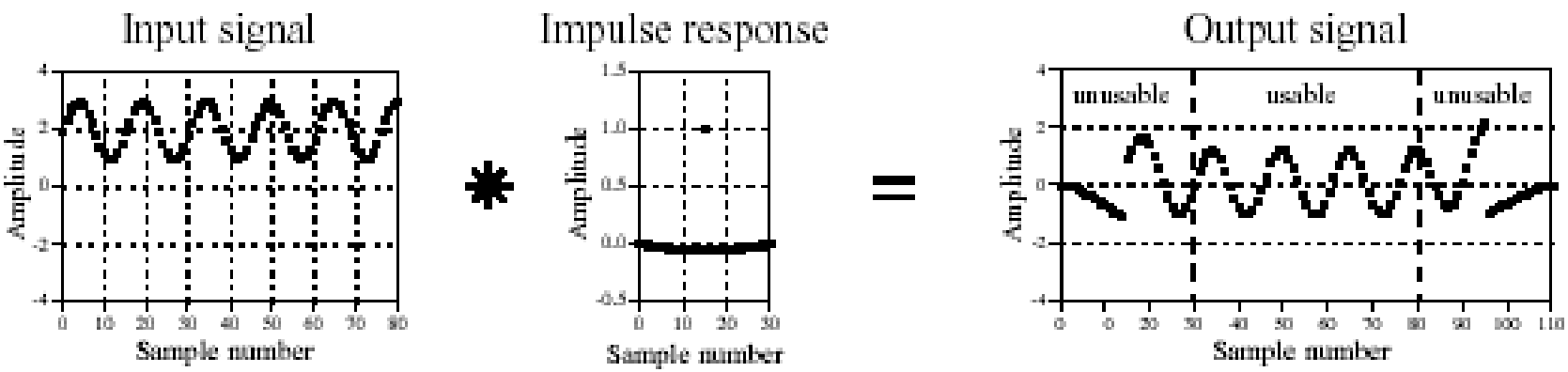


FIGURE 6-10
End effects in convolution. When an input signal is convolved with an *M* point impulse response, the first and last *M*-1 points in the output signal may not be usable. In this example, the impulse response is a high-pass filter used to remove the DC component from the input signal.

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Another View

- Convolution = "sum of weighted components"
- Weighting coefficients or factors = values of the samples of h[n]

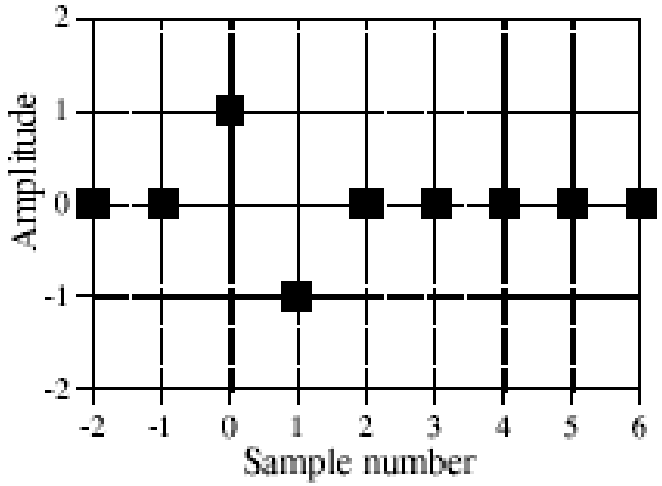
"Calculus-Like" Convolution Operations

- "First difference"
  - each sample of the output = difference between adjacent samples in the input
- "Running sum"
  - each sample in output = sum of all the input samples up to that sample
- First difference and running sum are inverses

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

"Calculus-Like" Convolution Operations

a. First Difference
This is the discrete version of the *first derivative*. Each sample in the output signal is equal to the *difference* between adjacent samples in the input signal. In other words, the output signal is the *slope* of the input signal.



b. Running Sum
The running sum is the discrete version of the *integral*. Each sample in the output signal is equal to the sum of all samples in the input signal to the *left*. Note that the impulse response extends to infinity, a rather nasty feature.

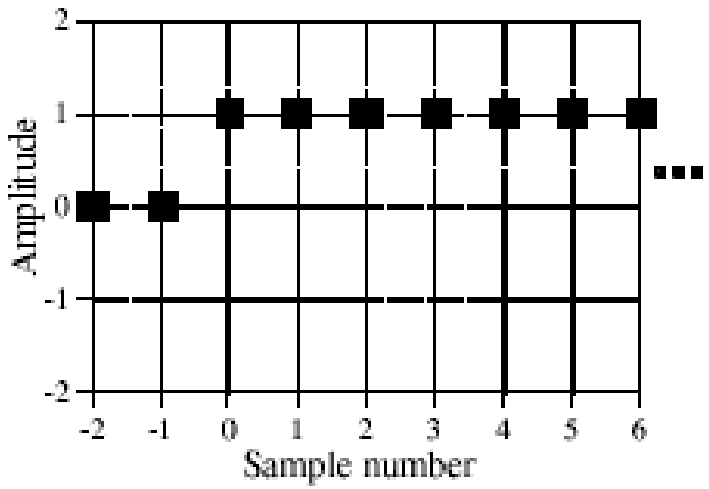


FIGURE 7-2
Impulse responses that mimic calculus operations.

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Copyright 2001 D

(cont'd)

FIGURE 7-3
Example of calculus-like operations. The signal in (b) is the *first difference* of the signal in (a). Correspondingly, the signal is (a) is the *running sum* of the signal in (b). These processing methods are used with discrete signals the same as differentiation and integration are used with continuous signals.

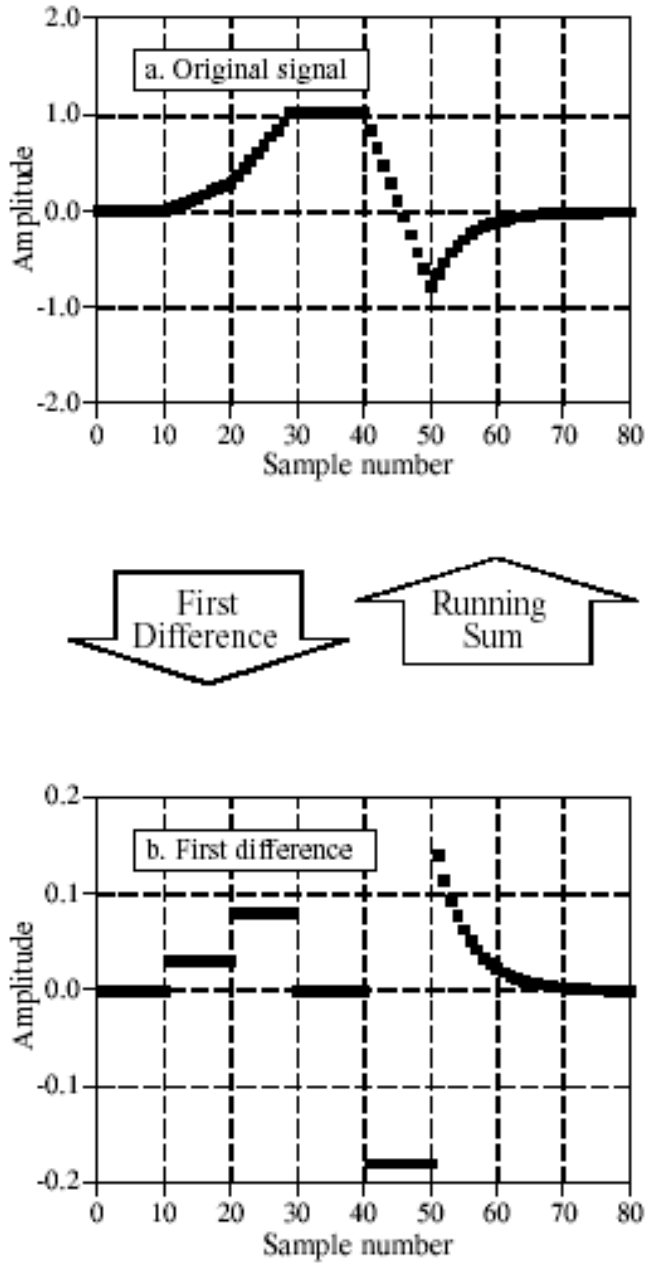


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Associativity of Convolution

- The order of two or more systems performed in series is irrelevant
- Two or more systems performed in series can be replaced by the convolution of the two systems

Correlation

- Purpose: find the "degree of match" between two signals
  - optimal technique for detecting a reference signal or pattern in random noise
- Definition:
  - y[i] = sum from j=0 to M of (x[i+j] • h[j])

Correlation Application

FIGURE 7-13
Key elements of a radar system. Like other echo location systems, radar transmits a short pulse of energy that is reflected by objects being examined. This makes the received waveform a shifted version of the transmitted waveform, plus random noise. Detection of a known waveform in a noisy signal is the fundamental problem in echo location. The answer to this problem is *correlation*.

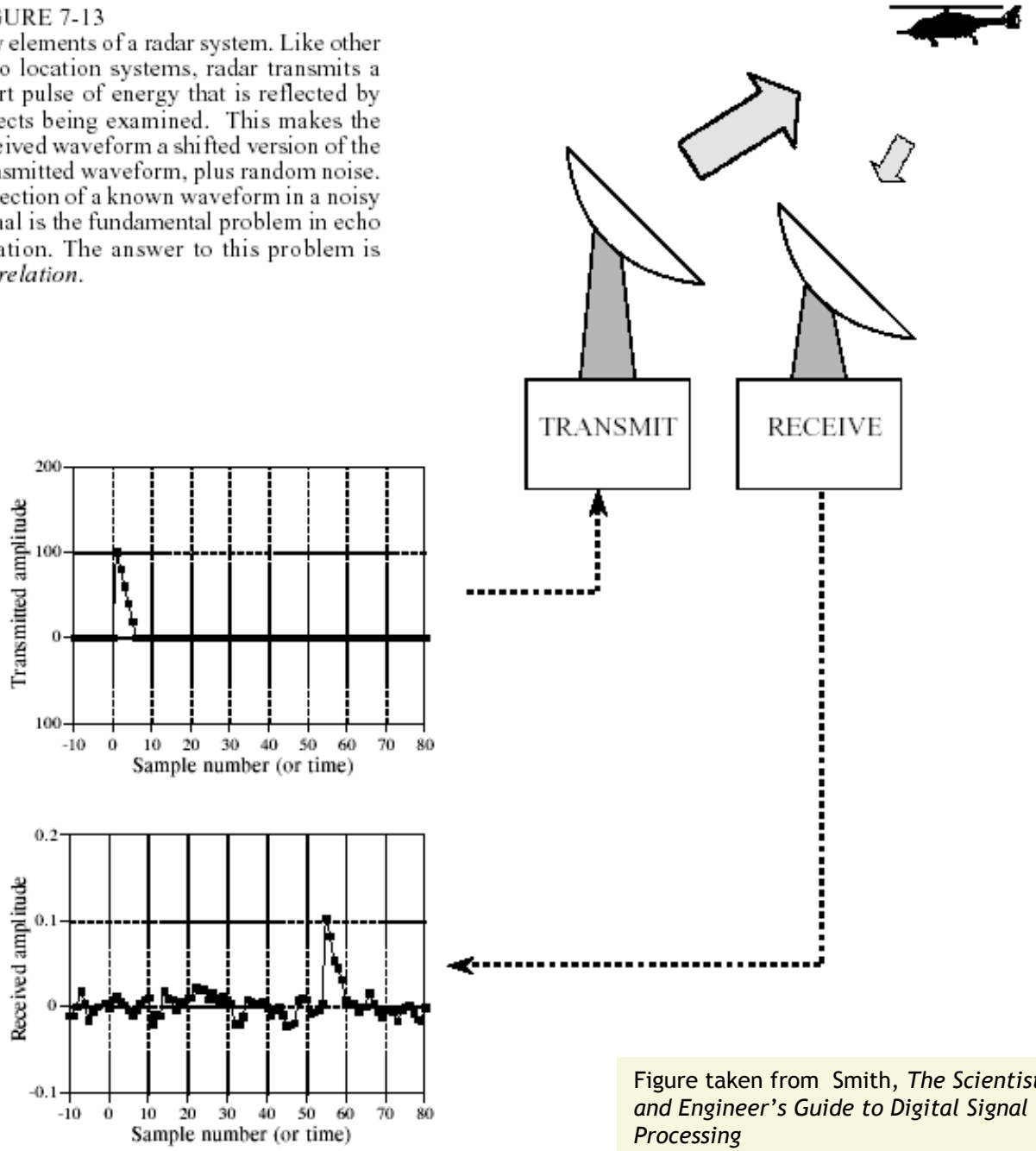


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Illustration of Correlation Computation

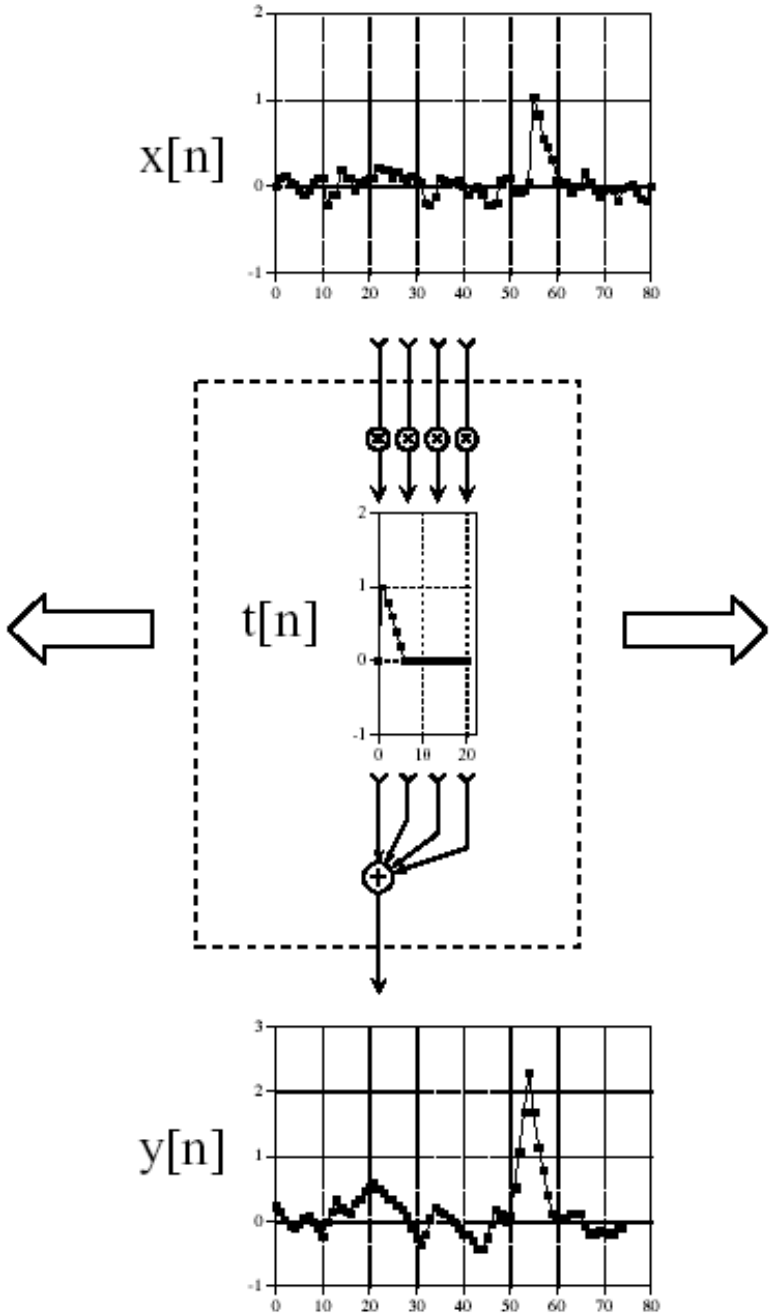


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

FIGURE 7-14

Discrete Fourier Transform (DFT)

- Decompose an N-point signal into (N/2+1) sine waves and (N/2+1) cosine waves
  - to be determined: amplitude of each sine and cosine wave
- The sine and cosine waves form an orthoganal basis function

Illustration of Fourier Transform

FIGURE 8-1a
(see facing page)

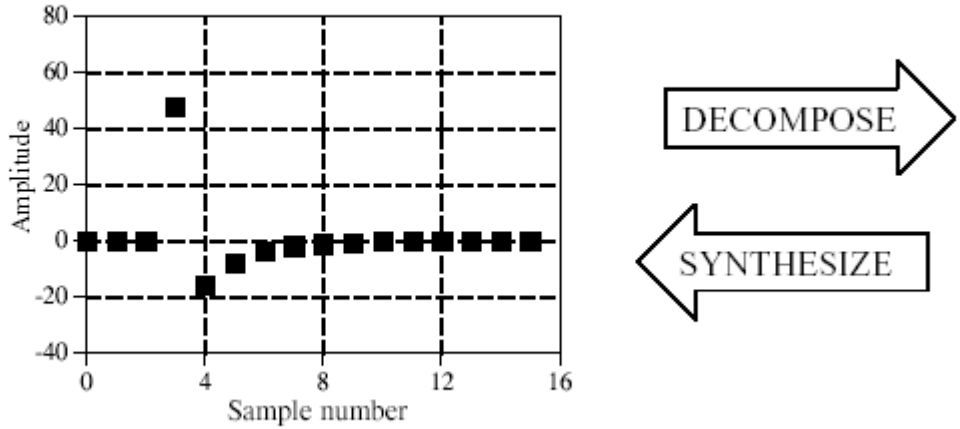


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Illustration (cont'd)

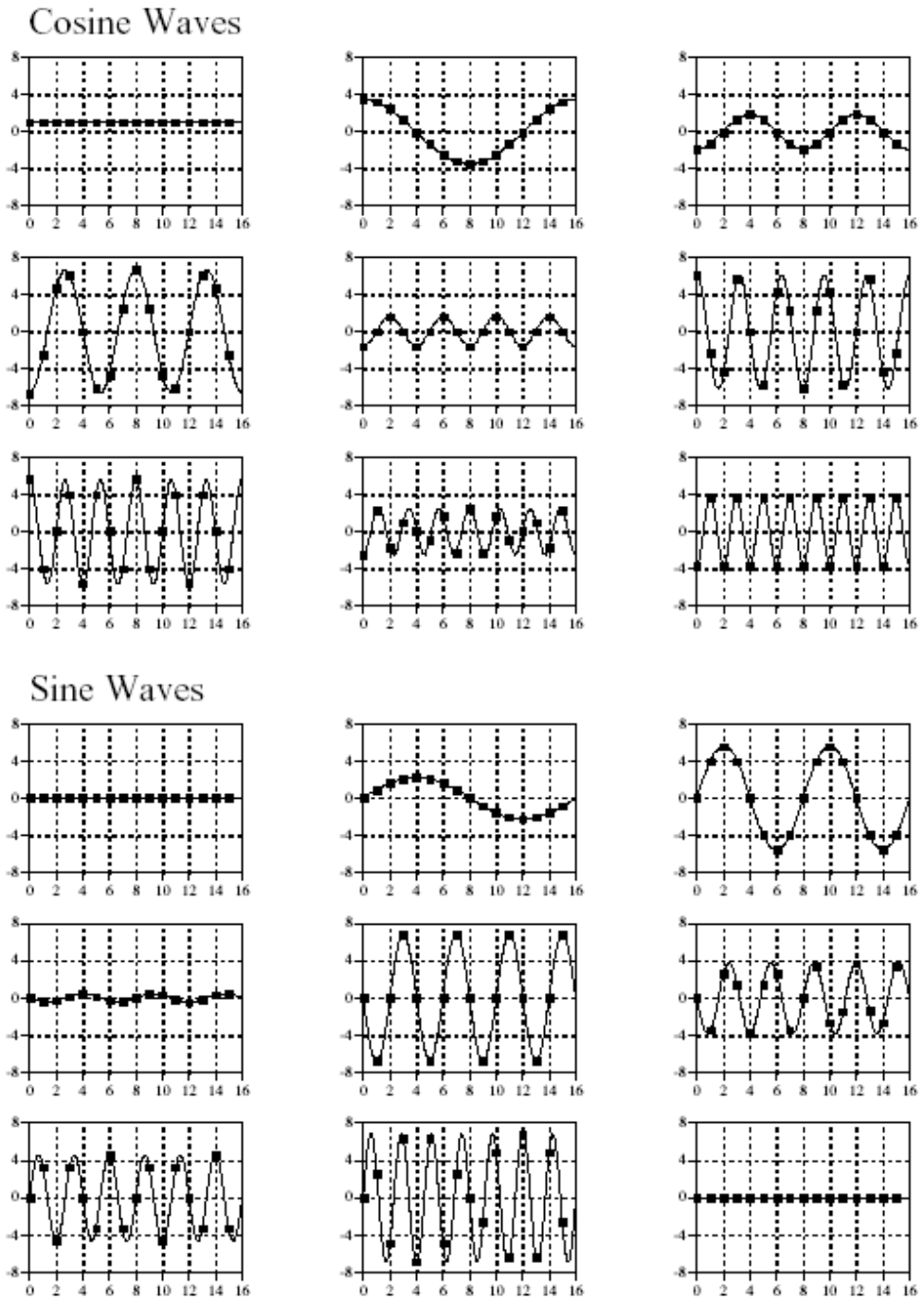


FIGURE 8-1b

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Terminology of DFT

- ReX[N] = real part of the frequency domain representation
- ImX[N] = imaginary part of the frequency domain representation
  - note: ImX[0] = ImX[N/2] = 0, always
- No complex arithmetic in this course!
- Independent variable = i/N = fraction of the sampling frequency

Scaling ReX and ImX

- Scaling needed to get amplitudes of basis functions

EQUATIONS 8-3
Conversion between the sinusoidal amplitudes and the frequency domain values. In these equations, $Re\bar{X}[k]$ and $Im\bar{X}[k]$ hold the amplitudes of the cosine and sine waves needed for synthesis, while $ReX[k]$ and $ImX[k]$ hold the real and imaginary parts of the frequency domain. As usual, N is the number of points in the time domain signal, and k is an index that runs from 0 to $N/2$.

$$Re\bar{X}[k] = \frac{ReX[k]}{N/2}$$

$$Im\bar{X}[k] = -\frac{ImX[k]}{N/2}$$

except for two special cases:

$$Re\bar{X}[0] = \frac{ReX[0]}{N}$$

$$Re\bar{X}[N/2] = \frac{ReX[N/2]}{N}$$

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

  - (see p. 156 of Smith for explanation of why scaling is needed)

- ReX = amplitude of cosine waves
- ImX = amplitudes of sine waves

Synthesis (Inverse DFT)

$$x[i] = \sum_{k=0}^{N/2} Re\bar{X}[k] \cos(2\pi ki/N) + \sum_{k=0}^{N/2} Im\bar{X}[k] \sin(2\pi ki/N)$$

EQUATION 8-2
The synthesis equation. In this relation, $x[i]$ is the signal being synthesized, with the index, i, running from 0 to $N-1$. $Re\bar{X}[k]$ and $Im\bar{X}[k]$ hold the amplitudes of the cosine and sine waves, respectively, with k running from 0 to $N/2$. Equation 8-3 provides the normalization to change this equation into the inverse DFT.

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Analysis (DFT)

- Correlate x[n] with each of the basis functions
  - reminder: find the "degree of match" between x[n] and each of the basis functions

EQUATION 8-4
The analysis equations for calculating the DFT. In these equations, $x[i]$ is the time domain signal being analyzed, and $Re\,X[k]$ & $Im\,X[k]$ are the frequency domain signals being calculated. The index i runs from 0 to $N-1$, while the index k runs from 0 to $N/2$.

basis function

$$Re\,X[k] = \sum_{i=0}^{N-1} x[i] \cos(2\pi k i / N)$$

$$Im\,X[k] = -\sum_{i=0}^{N-1} x[i] \sin(2\pi k i / N)$$

basis function

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Copyright 2001 Douglas S. Reeves (http://reeve

Example: First Part of Correlation

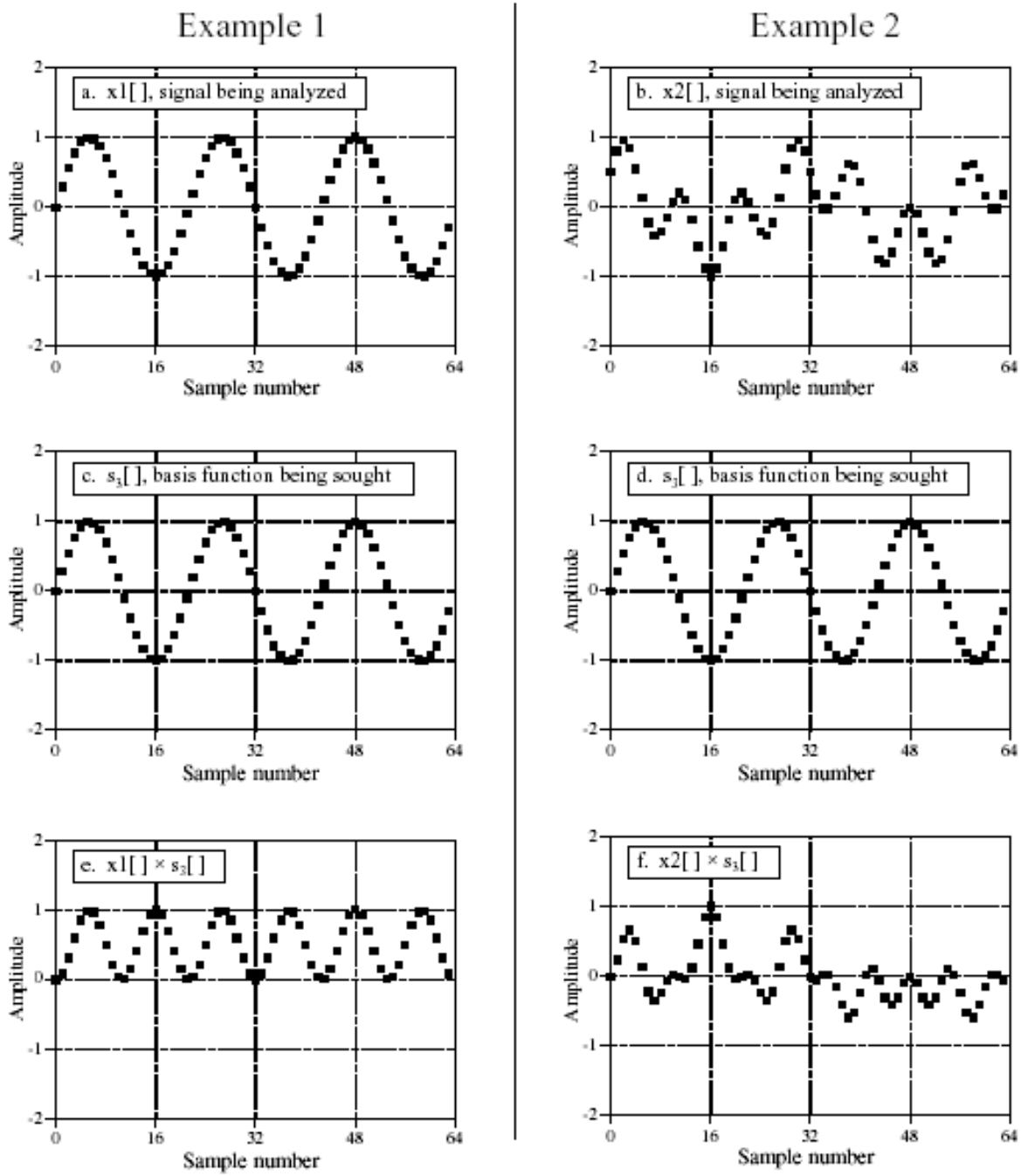


FIGURE 8-8

Polar Notation

- (skip)

Linearity of DFT

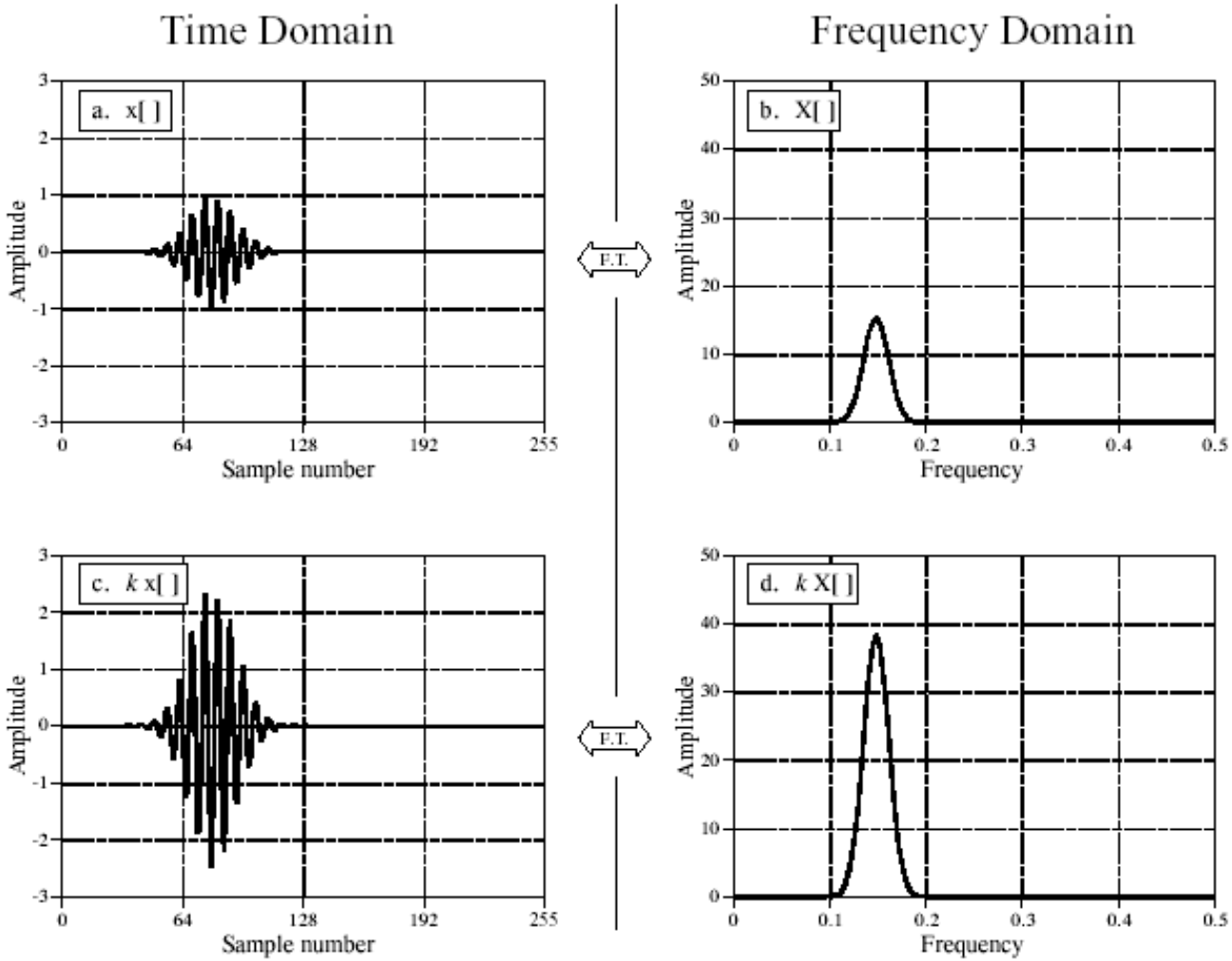


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

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

Copyright 2001 Douglas S. Reeves (http://ree

Linearity (cont'd)

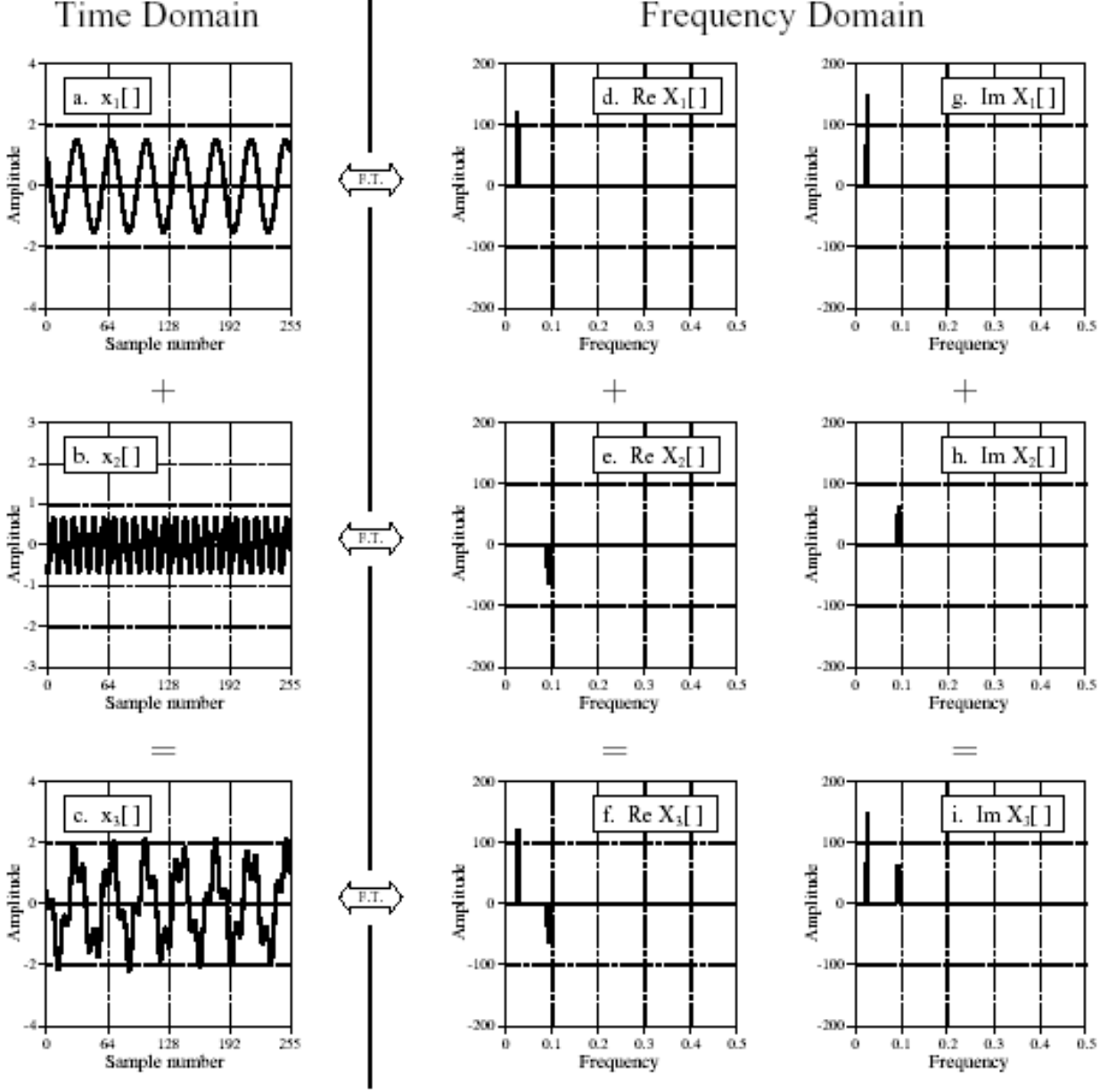


Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

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

Compression and Expansion of Time-Varying Signals

- Equivalent to resampling at a higher or lower rate
- *Interpolation* = adding samples
- *Decimation* = removing samples

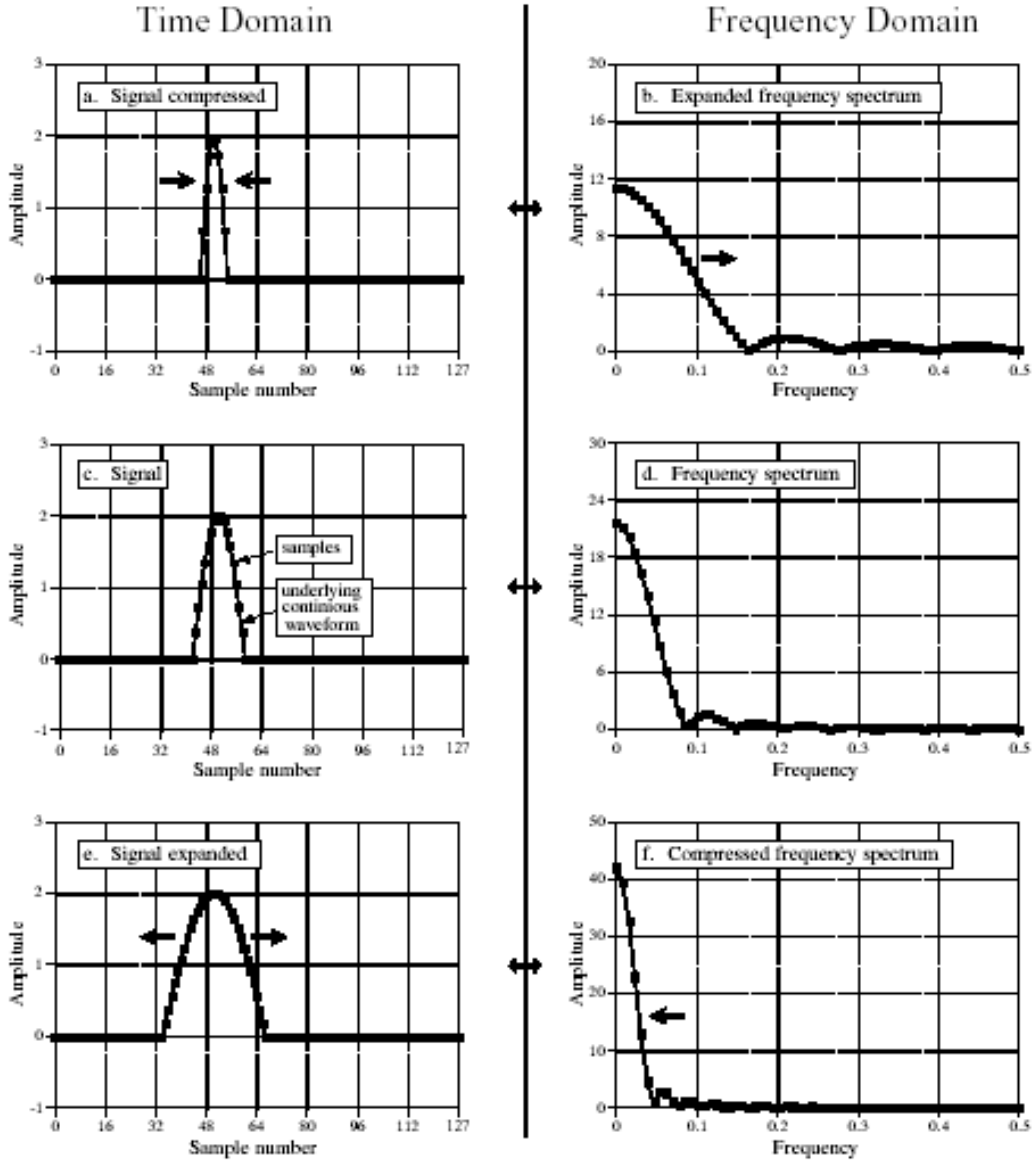


FIGURE 10-12
Compression and expansion. Compressing a signal in one domain results in the signal being expanded in the other domain, and vice versa. Figures (c) and (d) show a discrete signal and its spectrum, respectively. In (a) and (b), the time domain signal has been compressed, resulting in the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

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Expansion (cont'd)

- Resampling from the actual (analog) signal -- no problem!
- Resampling from a discrete (digital) signal -- problems!
  - How do you interpolate between samples -- linear? curve fit?
  - Better idea: convert to frequency domain, pad with zeros, synthesize back to time domain at new rate

Copyright 2001 Douglas S.

Expansion (cont'd)

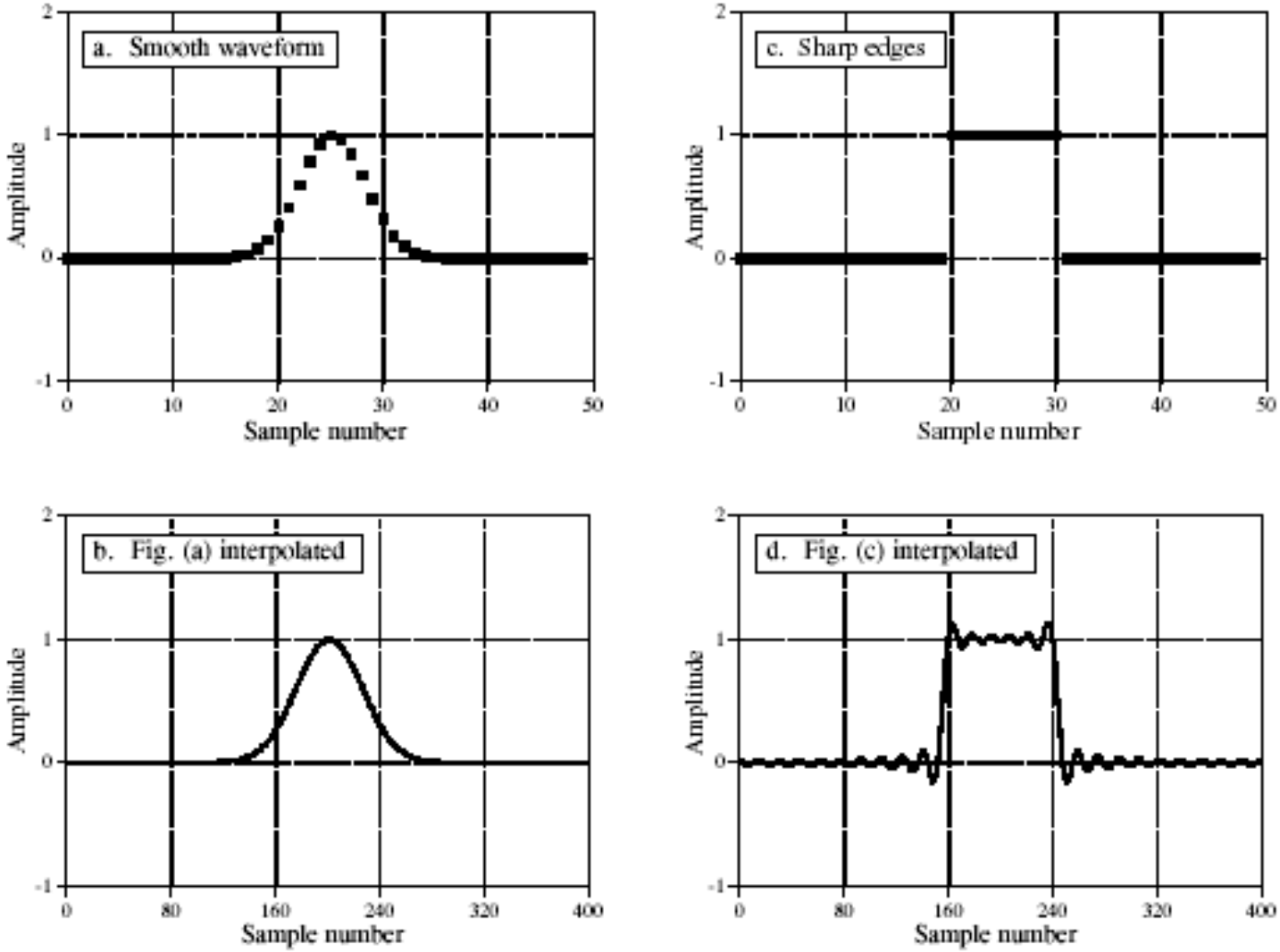


FIGURE 10-13
Interpolation by padding the frequency domain. Figures (a) and (c) each consist of 50 samples. These are interpolated to 400 samples by padding the frequency domain with zeros, resulting in (b) and (d), respectively. (Figures (b) and (d) are discrete signals, but are drawn as continuous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

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Digital Filters

- "Thousands of times better performance than analog filters"
- Implementing a filter = convoluting a signal with the filter kernel

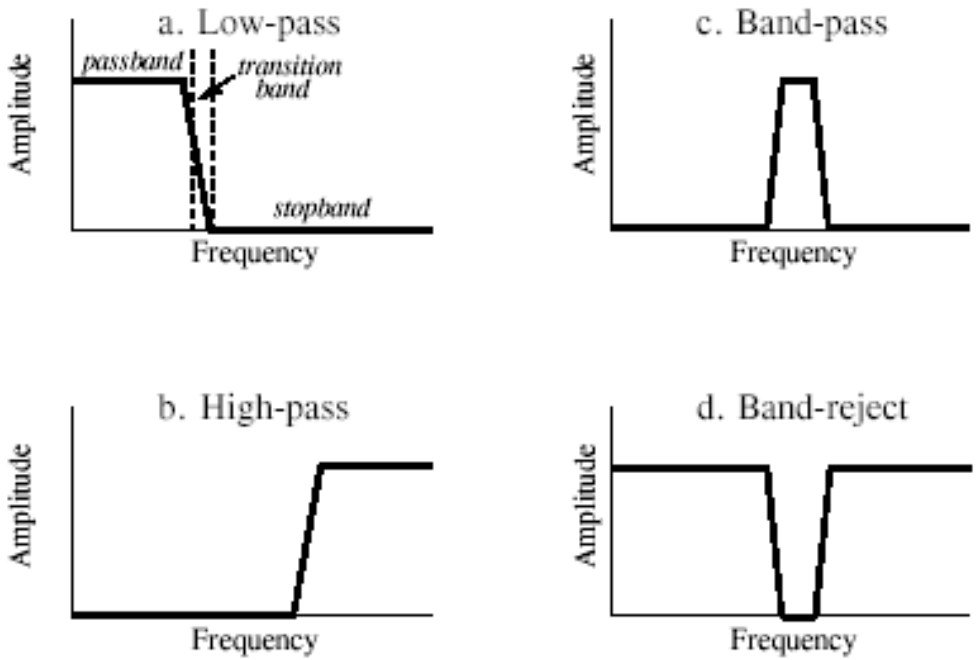


FIGURE 14-3
The four common frequency responses. Frequency domain filters are generally used to pass certain frequencies (the *passband*), while blocking others (the *stopband*). Four responses are the most common: low-pass, high-pass, band-pass, and band-reject.

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.

Filter Characteristics

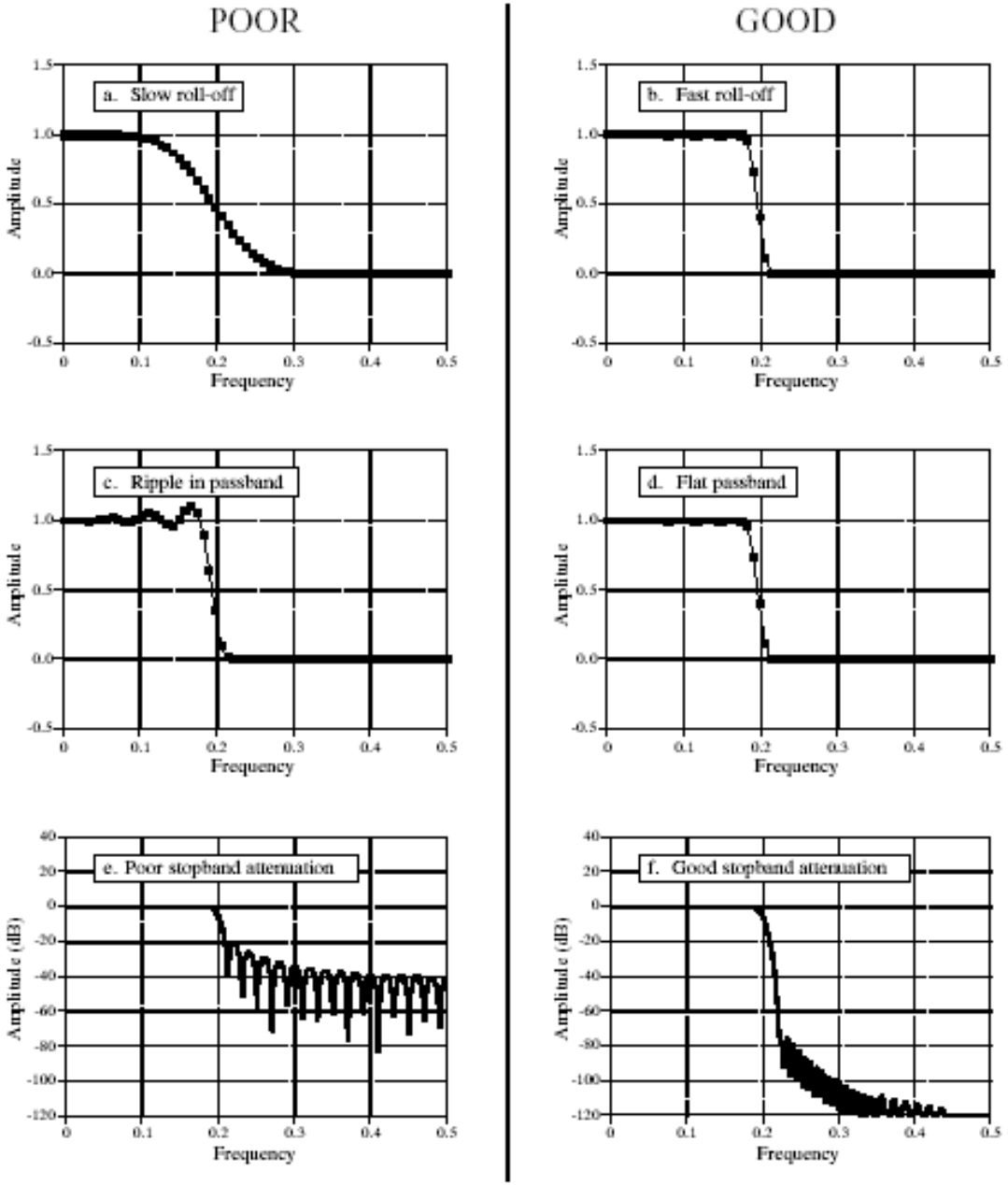
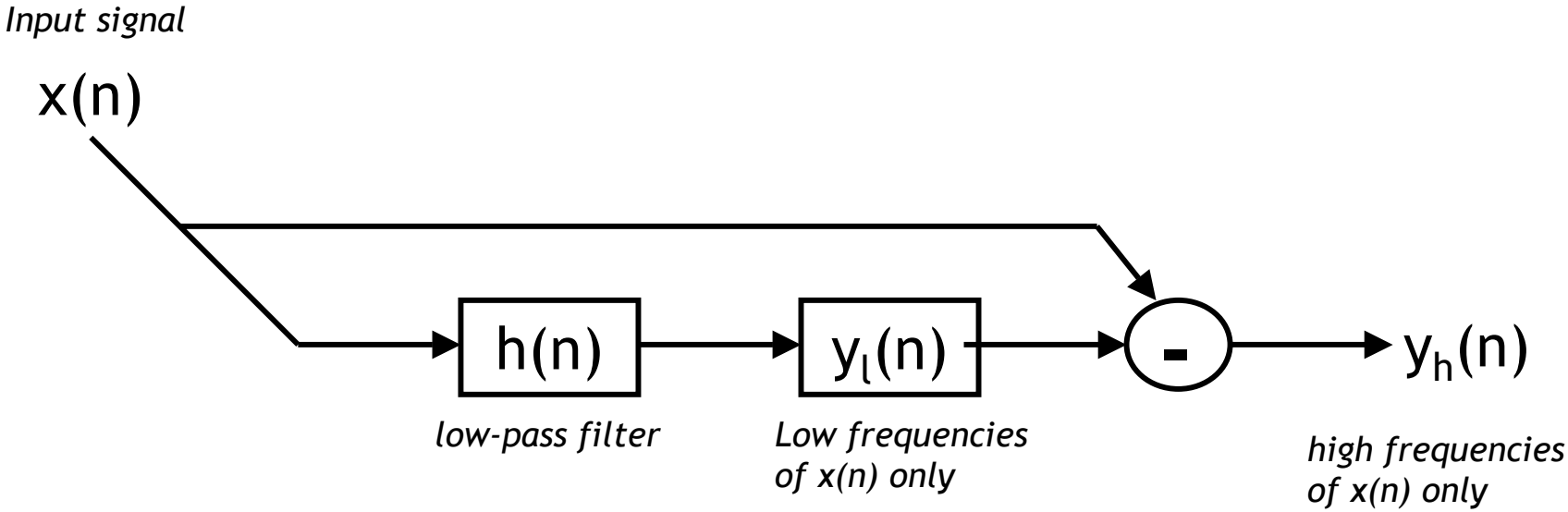


FIGURE 14-4
Parameters for evaluating *frequency domain* performance. The frequency responses shown are for low-pass filters. Three parameters are important: (1) roll-off sharpness, shown in (a) and (b), (2) passband ripple, shown in (c) and (d), and (3) stopband attenuation, shown in (e) and (f).

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Filter Design

- High pass, band pass, and band reject can be derived from low pass filters



- An exercise for the reader: how can you construct band pass and band reject filters from this?

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

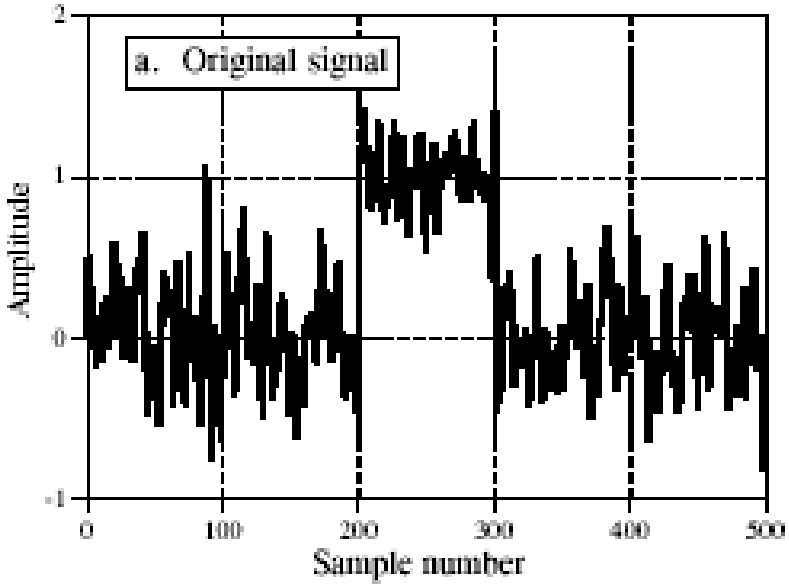
Moving Average Filter Design

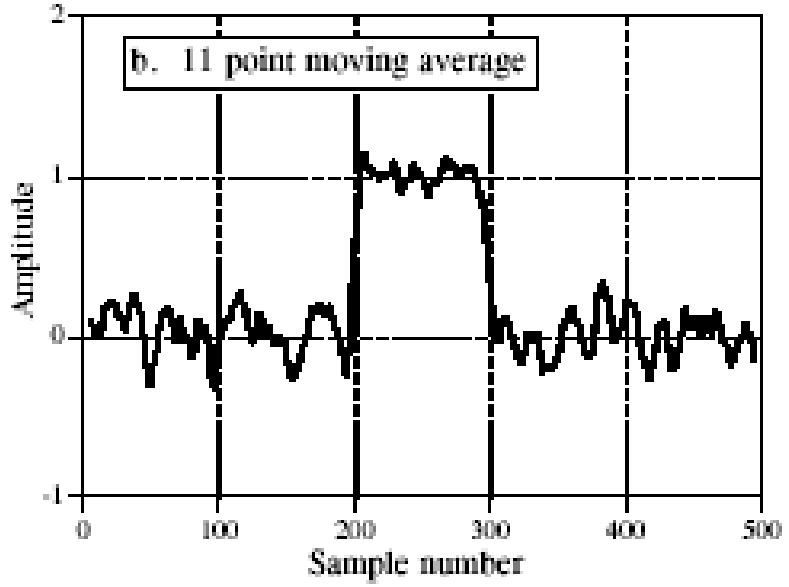
- M = *order* of the filter
  - Higher M = better filter quality, lower M = less computation

EQUATION 15-1
Equation of the moving average filter. In this equation, $x[\ ]$ is the input signal, $y[\ ]$ is the output signal, and M is the number of points used in the moving average. This equation only uses points on *one side* of the output sample being calculated.

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i-j]$$

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*





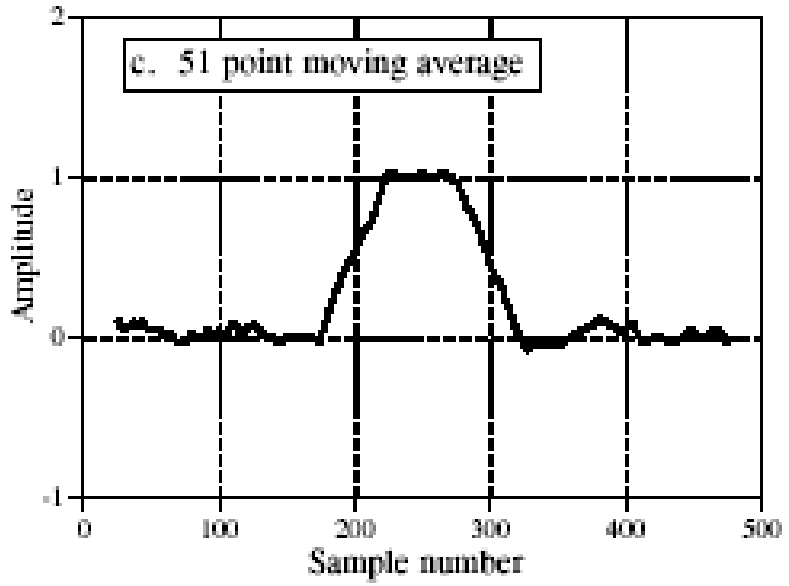


FIGURE 15-1
Example of a moving average filter. In (a), a rectangular pulse is buried in random noise. In (b) and (c), this signal is filtered with 11 and 51 point moving average filters, respectively. As the number of points in the filter increases, the noise becomes lower; however, the edges becoming less sharp. The moving average filter is the *optimal* solution for this problem, providing the lowest noise possible for a given edge sharpness.

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

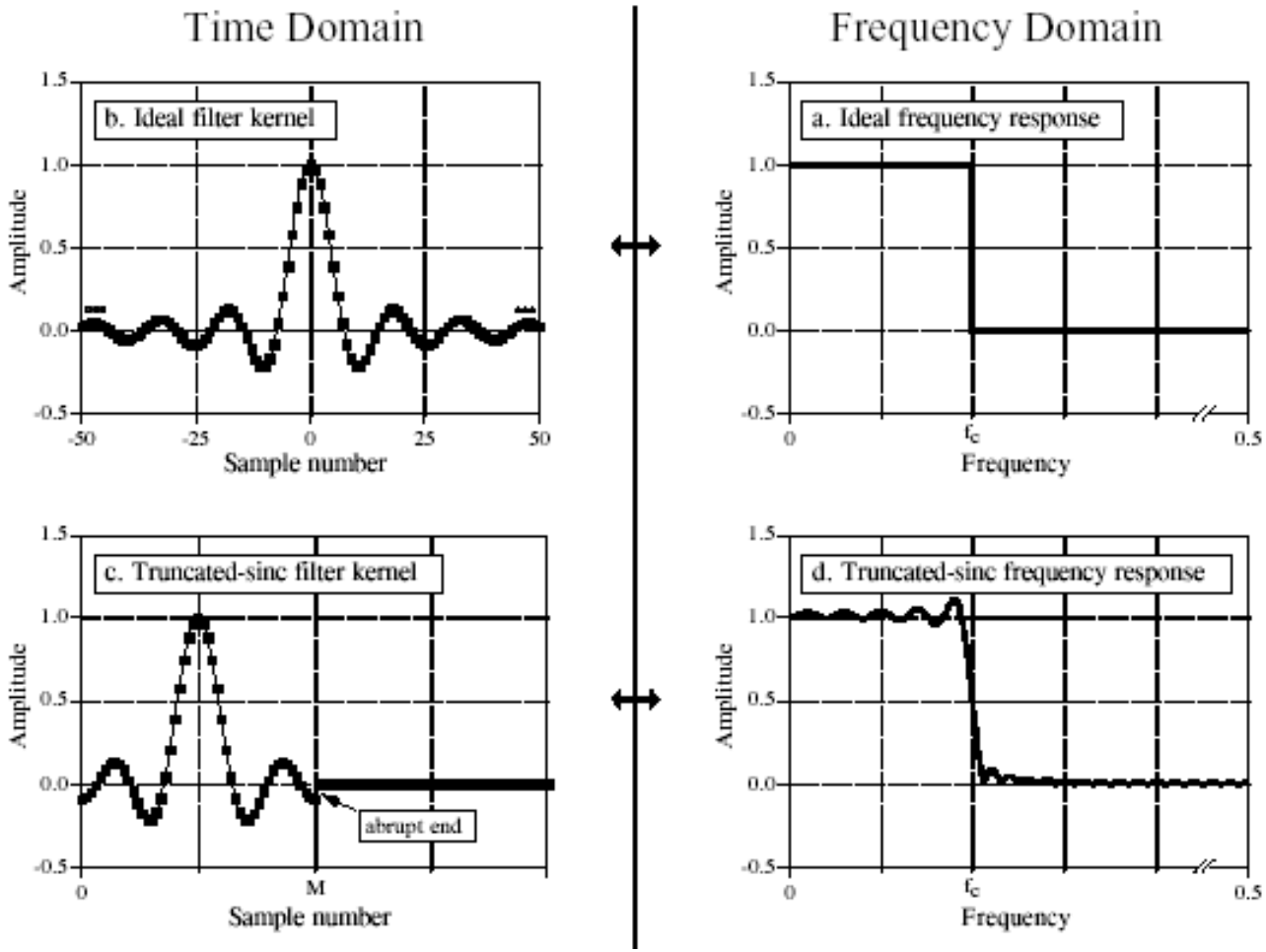
Ideal Filter Design

- Start with frequency transform of the desired response

1. Do IDFT to time domain
2. Terminate the infinite series at M+1 points
3. Shift to range 0 to M
4. Multiply by Blackman window to compensate for finite effects

Ideal Filter Design Illustration

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*



Copyright 2001 Douglas S. Reeves (http://reeves.csc.ncsu.edu)

Illustration (cont'd)

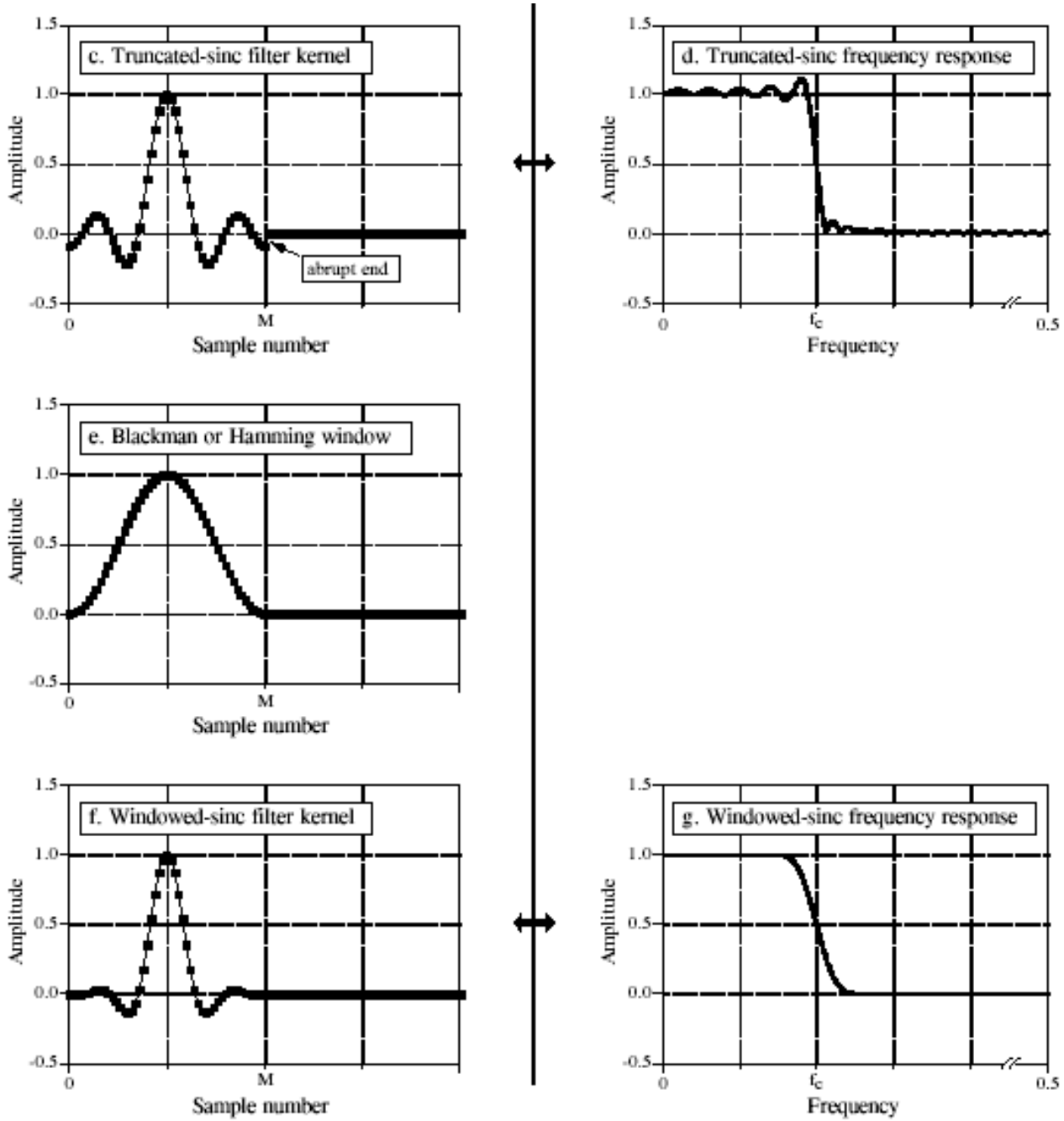


FIGURE 16-1

Figure taken from Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*

Filter Design (Bottom Line)

- f_c is fraction of sampling rate
  - E.g., if sampling rate is 8000 samples/sec, and you want a cutoff frequency of 2000 cycles/sec, then f_c = 2000/8000 = .25

$$h[i] = K \frac{\sin(2\pi f_c (i - M/2))}{i - M/2} \left[0.42 - 0.5 \cos\left(\frac{2\pi i}{M}\right) + 0.08 \cos\left(\frac{4\pi i}{M}\right) \right]$$

EQUATION 16-4
The windowed-sinc filter kernel. The cutoff frequency, f_C, is expressed as a fraction of the sampling rate, a value between 0 and 0.5. The length of the filter kernel is determined by M, which must be an even integer. The sample number i, is an integer that runs from 0 to M, resulting in $M+1$ total points in the filter kernel. The constant, K, is chosen to provide unity gain at zero frequency. To avoid a divide-by-zero error, for $i = M/2$, use $h[i] = 2\pi K$.

Sources of Information

- Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*
  - *Chapter 6*
    - *Delta function and impulse response – skim only*
    - *Convolution – read*
    - *Input side algorithm – not needed*
    - *Output side algorithm – read*
    - *Sum of weighted inputs – read*
  - *Chapter 7*
    - *Delta function – not needed*
    - *Calculus-like operations – read*
    - *Low-pass and high-pass filters – read*
    - *Causal and non-causal signals – not needed*
    - *Zero phase etc. – not needed*
    - *Mathematical properties – read*
    - *Correlation – read*
    - *Speed – not needed*

Sources of Information

- Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*
  - *Chapter 8*
    - *Family of Fourier Transforms – read*
    - *Notation and format of the Real DFT – read*
    - *Frequency domain independent variable – read*
    - *DFT basis functions – read*
    - *Synthesis, calculating the inverse DFT – read*
    - *Analysis, calculating the DFT – read*
    - *Duality – not needed*
    - *Polar notation – skim*
  - *Chapter 9*
    - *Spectral analysis of signals – read only the starting part*
    - *Frequency response of systems – not needed*
    - *Convolution via the frequency domain – useful, not needed*

Sources of Information

- Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*
  - *Chapter 10*
    - *Linearity of fourier transform – read*
    - *Characteristics of phase – not needed*
    - *Periodic nature of DFT – skim*
    - *Compression and expansion, multirate – read*
    - *Multiplying signals (modulation) – skip*
  - *Chapter 11—13 – not needed*
  - *Chapter 14*
    - *Filter basics – read*
    - *How information is represented – skip*
    - *Time domain parameters – skim only*
    - *Frequency domain parameters – read*
    - *High-pass, band-pass, and band-reject filters – skim only*
    - *Filter classification – skim only*

Sources of Information

- Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*
  - *Chapter 15*
    - *Implementation by convolution - read*
    - *Noise reduction vs. step response – skim only*
    - *Frequency response – skim only*
    - *Rest of chapter… - skip*
  - *Chapter 16*
    - *Strategy for the windowed sinc filter – skim only*
    - *Designing the filter – read*
    - *Rest of chapter… - skip*

And More Sources of Information

- [Pierce81] *Signals: The Telephone and Beyond*
- [McClellen98] *DSP First: A Multimedia Approach*