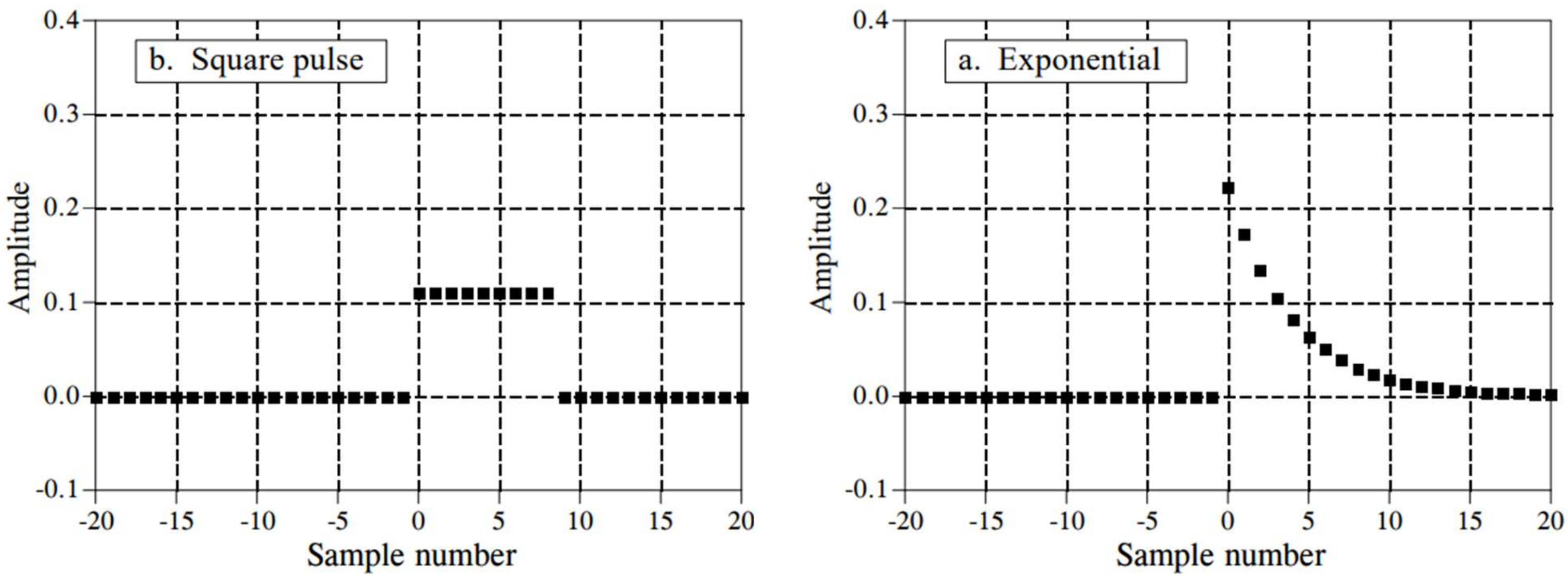
CSE 4631

Chapter 7, 8 - Smith

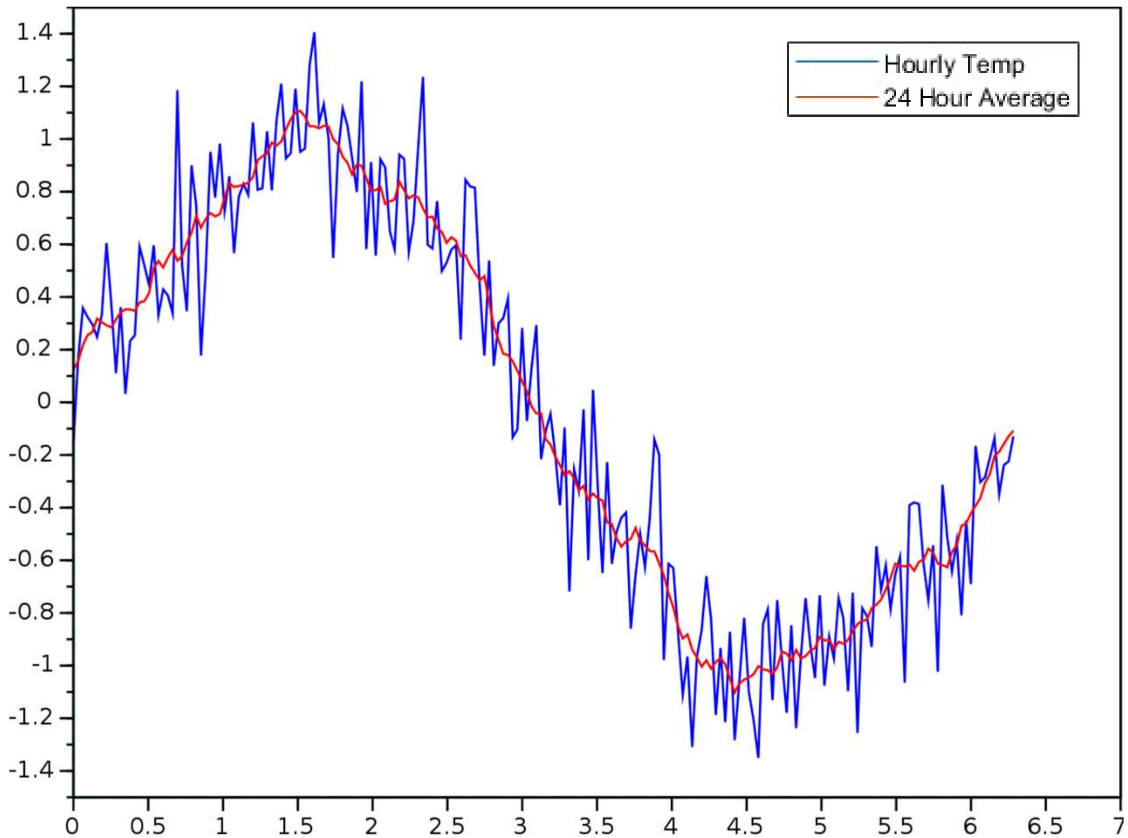
Md. Zahidul Islam
Lecturer, CSE, IUT

Low Pass Filter

- In general, low-pass filter kernels are composed of a group of adjacent positive points.
- This results in each sample in the output signal being a weighted average of many adjacent points from the input signal.
- This averaging smooths the signal, thereby removing high frequency components.



Moving Avg. Filter – A Low Pass Filter



Low Pass Filter

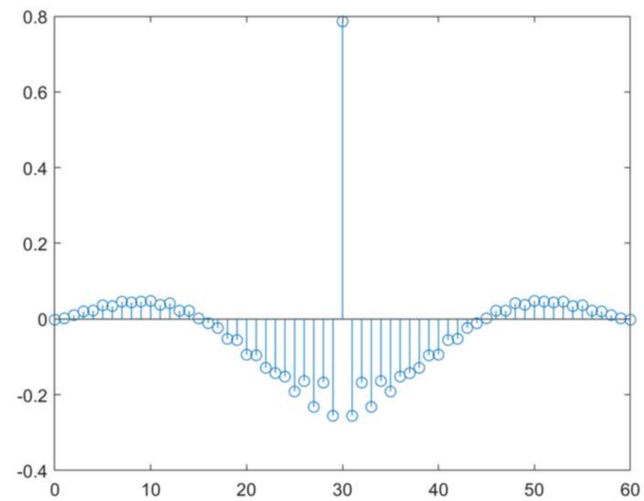
- The cutoff frequency of the filter is changed by making filter kernel wider or narrower.
- If a low-pass filter has a gain of one at DC (zero frequency), then the sum of all of the points in the impulse response must be equal to one.

Has the same blurring effect on images,
(Images are just 2D discrete signals)

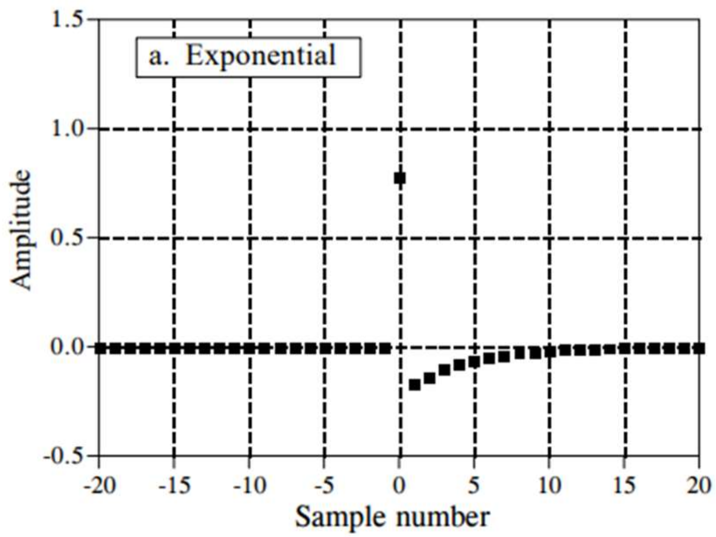
Figure 4: result of Gaussian low pass filter

High Pass Filters

- A delta function impulse response passes the entire signal, while a low-pass impulse response passes only the low frequency components.
- By superposition, a filter kernel consisting of a delta function minus the low-pass filter kernel will pass the entire signal minus the low-frequency components.
- A high-pass filter is born!
- HPF = delta – LPF



High Pass Filters



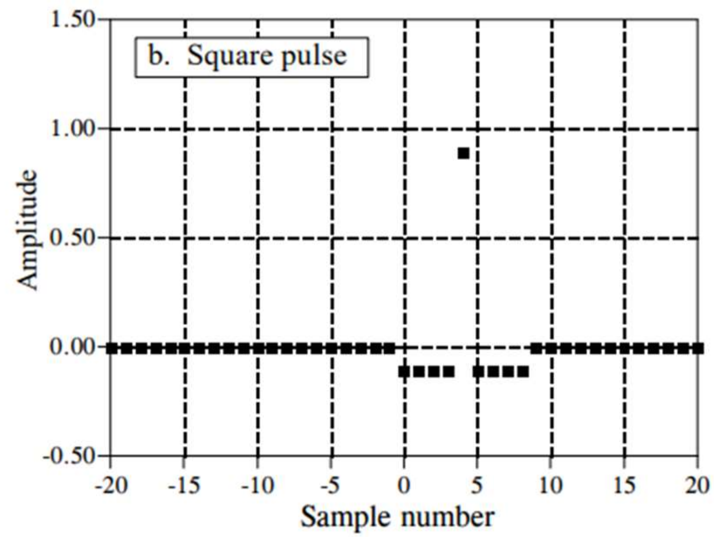
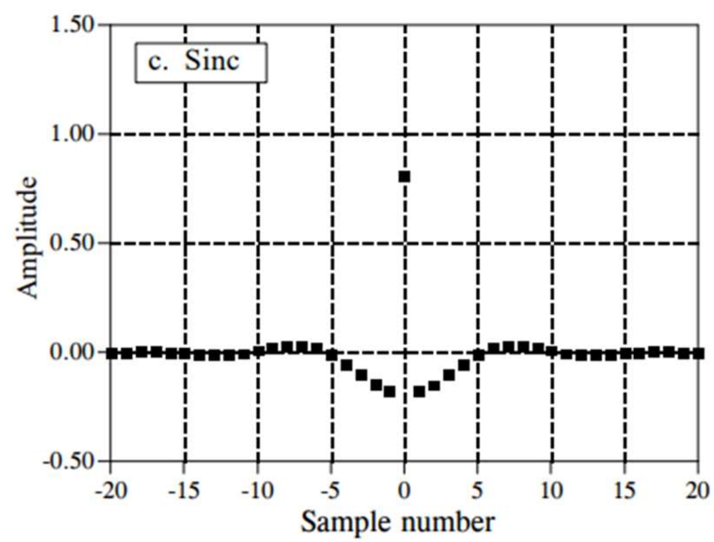
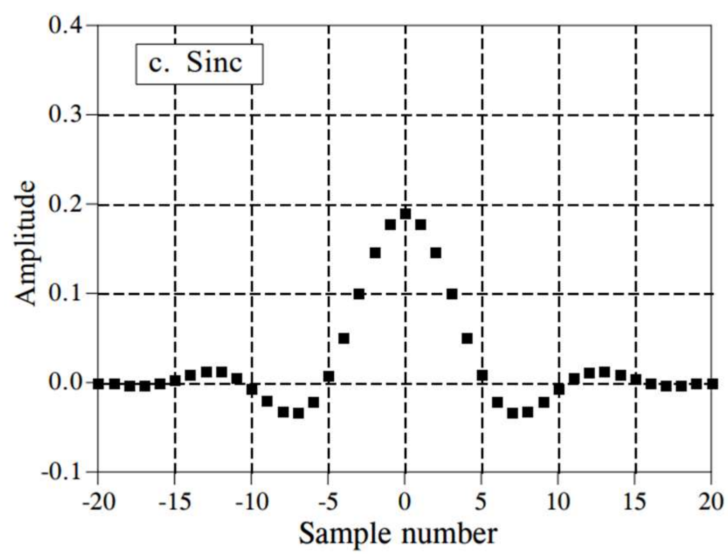


FIGURE 7-5
Typical high-pass filter kernels. These are formed by subtracting the corresponding low-pass filter kernels in Fig. 7-4 from a delta function. The distinguishing characteristic of high-pass filter kernels is a spike surrounded by many adjacent negative samples.

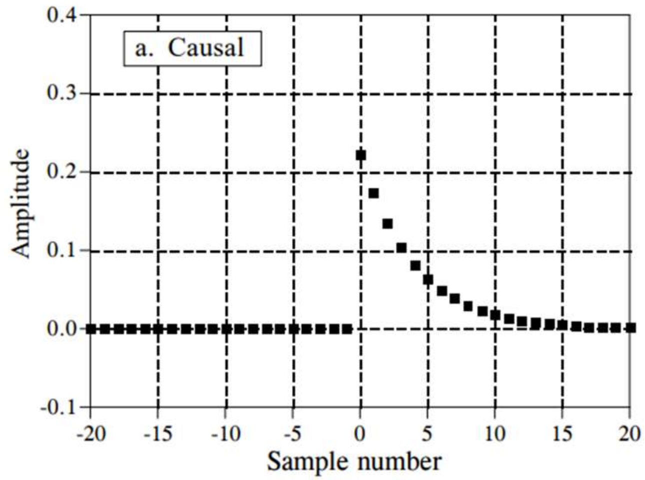


Band-Pass and Band-Reject Filters

- Only lets a certain range of frequencies pass through.
- The sinc function in (c), a curve of the form: sin(x)/(x), is used to separate one band of frequencies from another.



Causal and NonCausal Signals



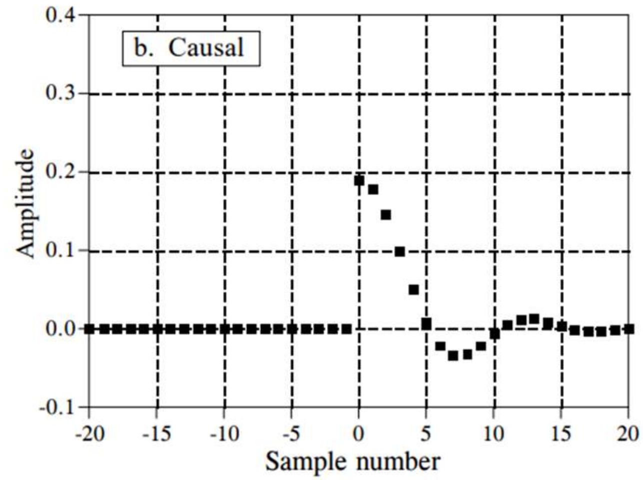
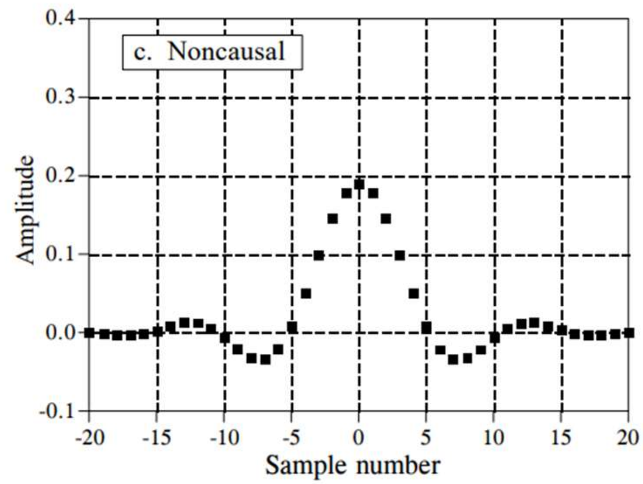
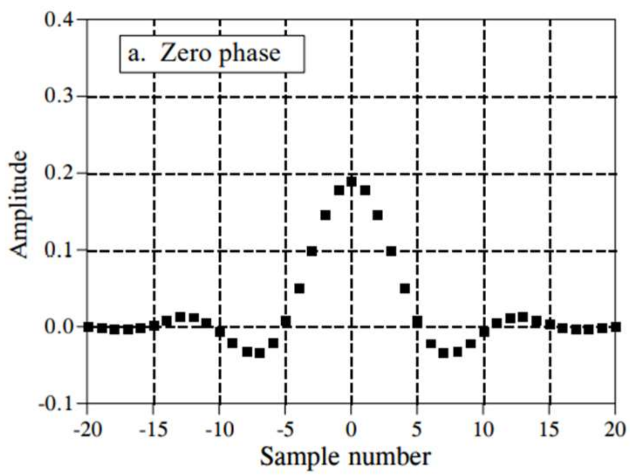


FIGURE 7-6
Examples of causal signals. An impulse response, or any signal, is said to be *causal* if all negative numbered samples have a value of zero. Three examples are shown here. Any noncausal signal with a finite number of points can be turned into a causal signal simply by shifting.



Zero Phase, Linear Phase, and Nonlinear Phase



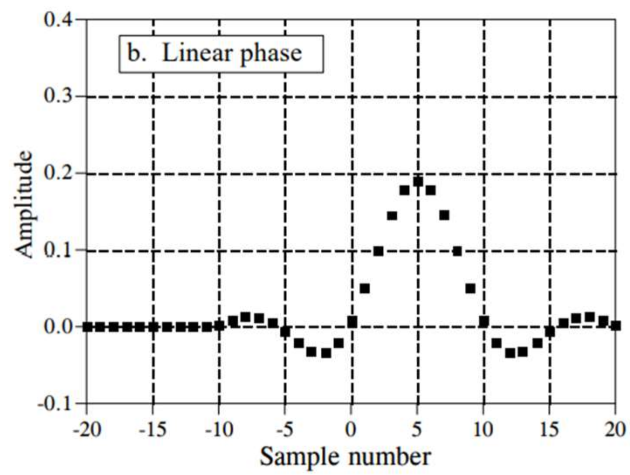
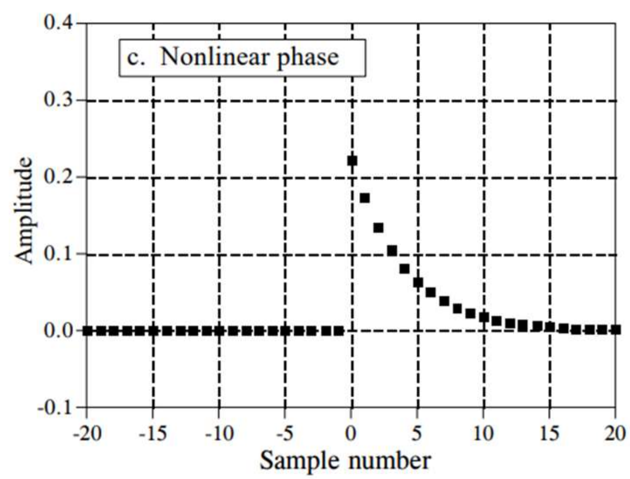


FIGURE 7-7
Examples of phase linearity. Signals that have a left-right symmetry are said to be *linear phase*. If the axis of symmetry occurs at sample number zero, they are additionally said to be *zero phase*. Any linear phase signal can be transformed into a zero phase signal simply by shifting. Signals that do not have a left-right symmetry are said to be *nonlinear phase*. Do not confuse these terms with the *linear* in linear systems. They are completely different concepts.



Mathematical Properties

Commutative Property

The commutative property for convolution is expressed in mathematical form:

EQUATION 7-6
The commutative property of convolution. This states that the order in which signals are convolved can be exchanged.

$$a[n] * b[n] = b[n] * a[n]$$

Associative Property

Is it possible to convolve three or more signals? The answer is yes, and the associative property describes how: convolve two of the signals to produce an intermediate signal, then convolve the intermediate signal with the third signal. The associative property provides that the order of the convolutions doesn't matter. As an equation:

EQUATION 7-7
The associative property of convolution describes how three or more signals are convolved.

$$(a[n] * b[n]) * c[n] = a[n] * (b[n] * c[n])$$

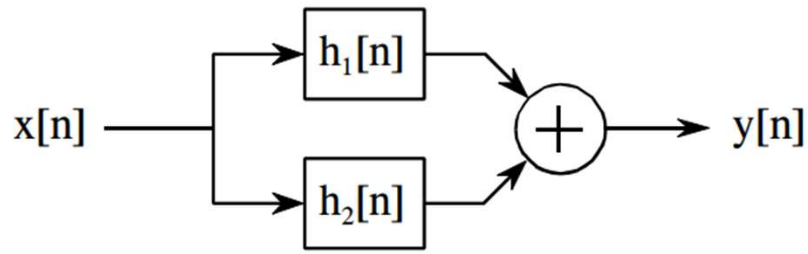
Distributive Property

In equation form, the distributive property is written:

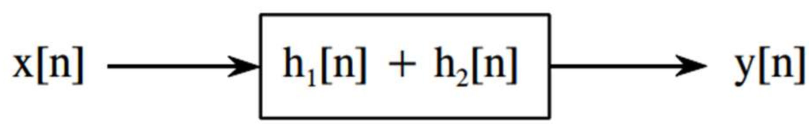
EQUATION 7-8
The distributive property of convolution describes how parallel systems are analyzed.

$$a[n] * b[n] + a[n] * c[n] = a[n] * (b[n] + c[n])$$

IF



THEN



Transference between input and output

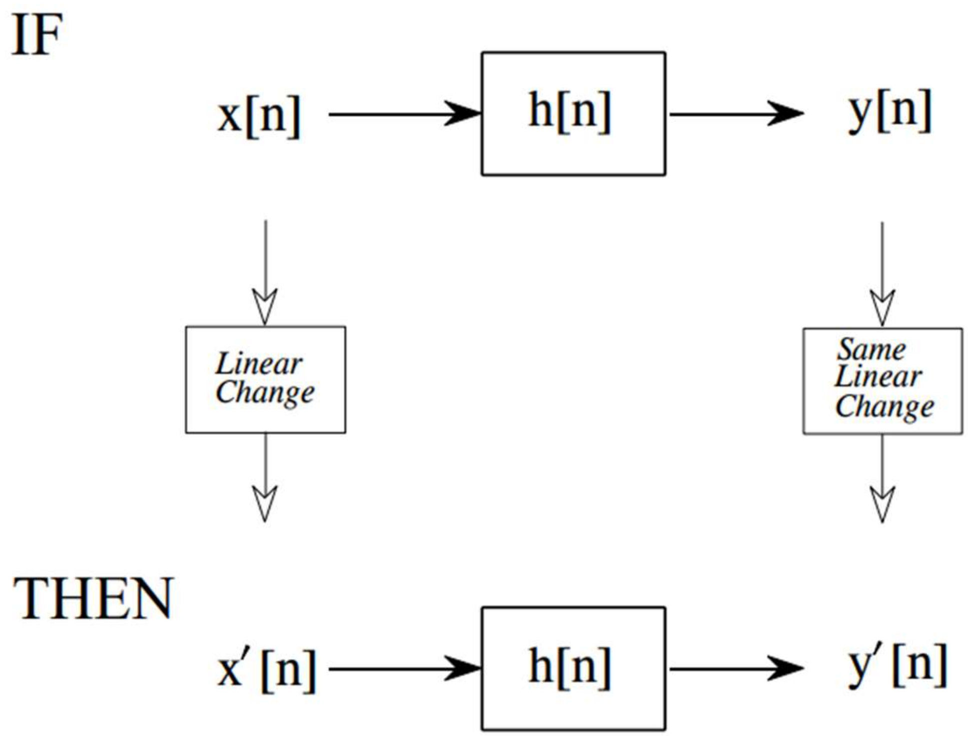
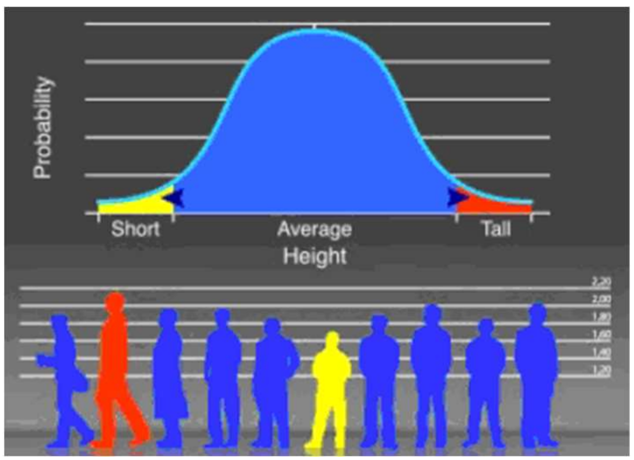


FIGURE 7-11
Tranference between the input and output. This is a way of thinking about a common situation in signal processing. A linear change made to the input signal results in the same linear change being made to the output signal.

The Central Limit Theorem

Why the Gaussian probability distribution is observed so commonly in nature?

- Gaussian distribution results when the observed variable is the sum of many random processes



- The Central Limit Theorem has an interesting implication for convolution. If a pulse-like signal is convolved with itself many times, a Gaussian is produced.

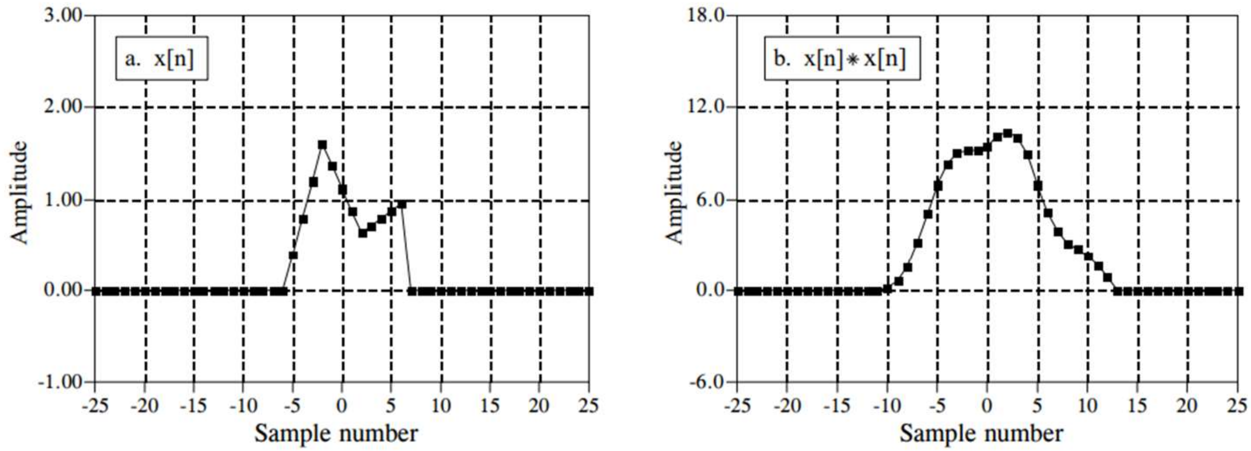
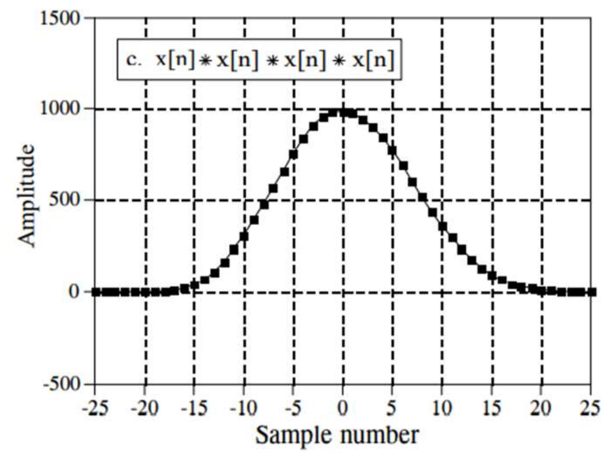


FIGURE 7-12
Example of convolving a pulse waveform with itself. The Central Limit Theorem shows that a Gaussian waveform is produced when an arbitrary shaped pulse is convolved with itself many times. Figure (a) is an example pulse. In (b), the pulse is convolved with itself *once*, and begins to appear smooth and regular. In (c), the pulse is convolved with itself *three* times, and closely approximates a Gaussian.



Correlation

- A measure of similarity between two signals

- Useful to check whether one signal is present in another signal

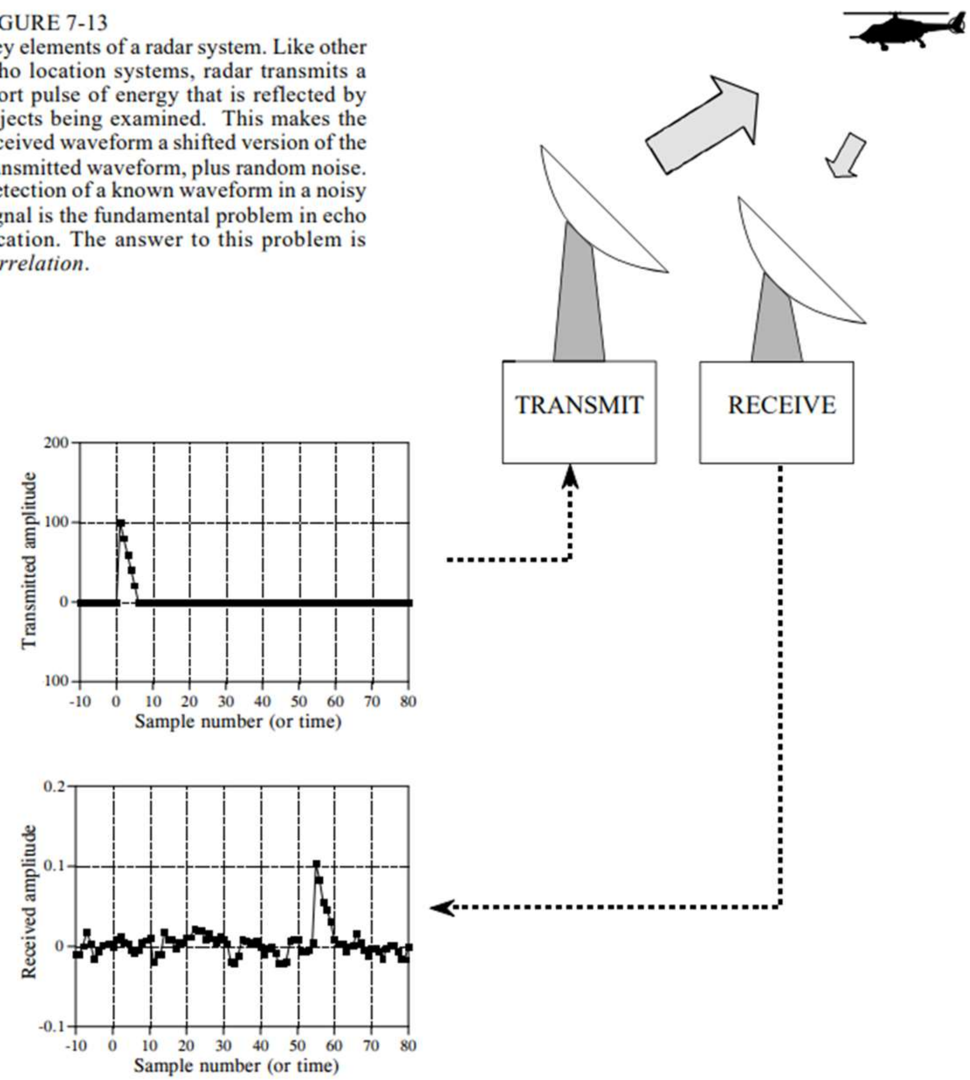
- Correlation is a mathematical operation that is very **similar to convolution**. Just as with convolution, correlation uses two signals to produce a third signal. This third signal is called the **cross-correlation** of the two input signals.

- If a signal is correlated with itself, the resulting signal is instead called the **autocorrelation**.

Correlation For Radar

- We want to check whether the first signal is contained within the second signal

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*.

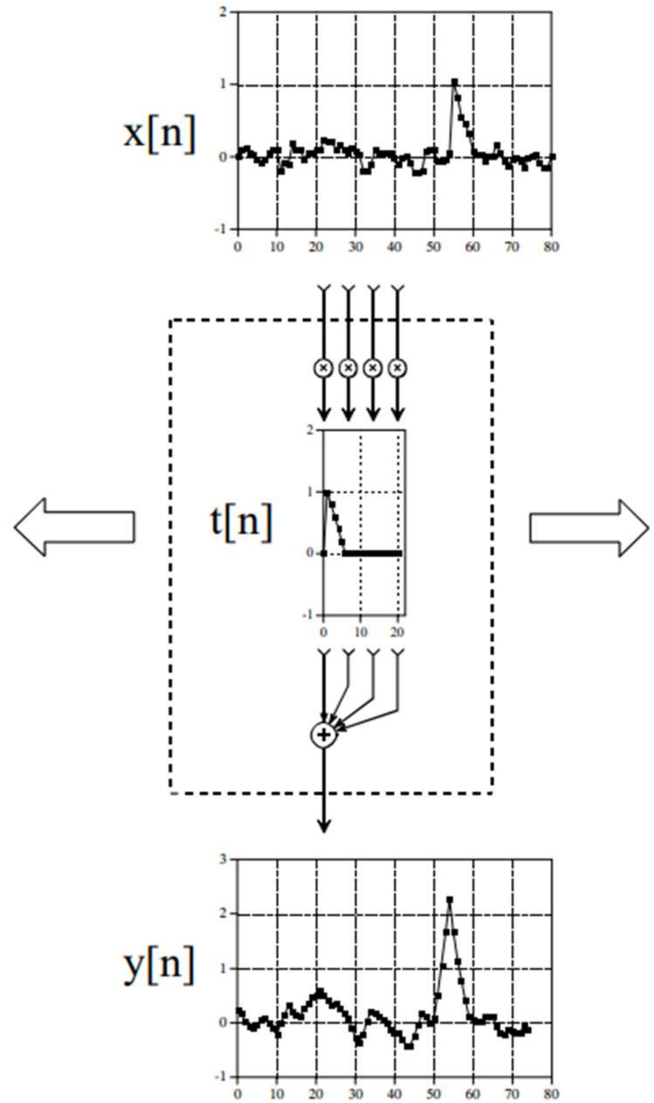


How to calculate correlation?

- Correlation output will have spikes wherever there is a similarity of t[n] with x[n]

Just use the **Output Side algorithm for convolution, but don't flip the impulse response**

Why does it work?



Convolution vs Correlation

- Convolution a[n] * b[n]
- Correlation a[n] * b[-n]

Speed/Complexity

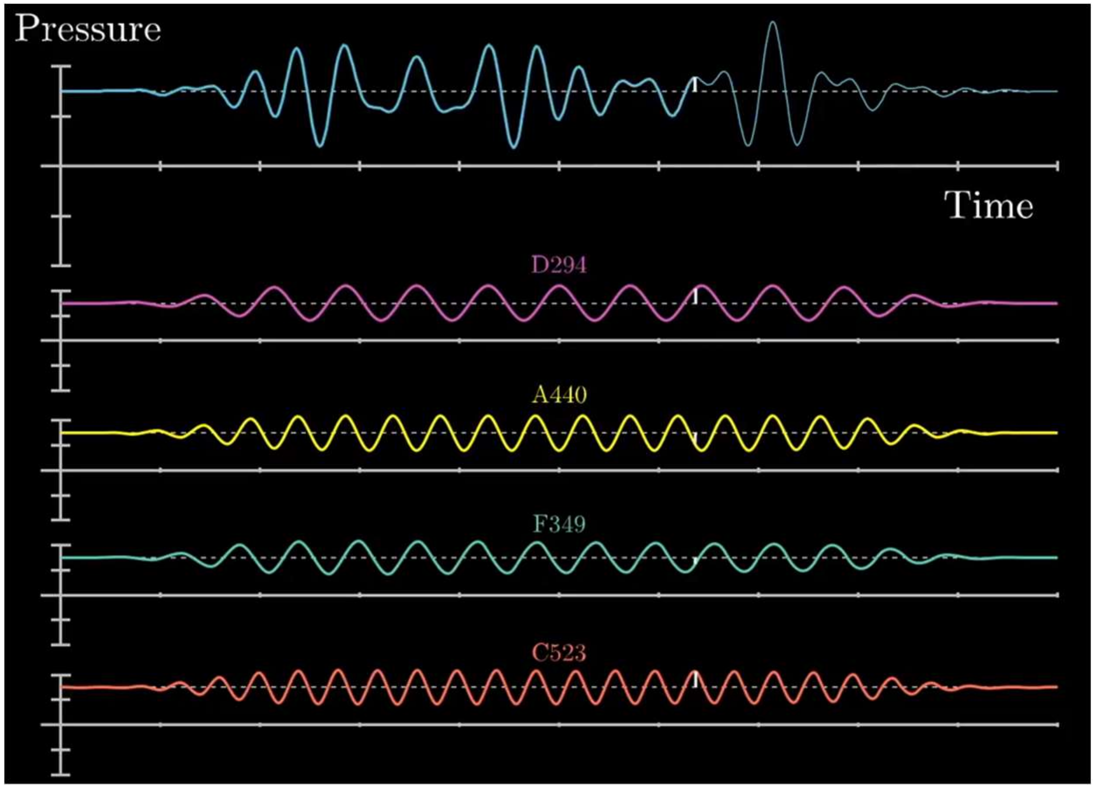
- Normal Convolution NxM
- FFT Convolution NlogN

Brief Overview of Discrete Fourier Transform

- Fourier analysis is a family of mathematical techniques, all based on decomposing signals into sinusoids.

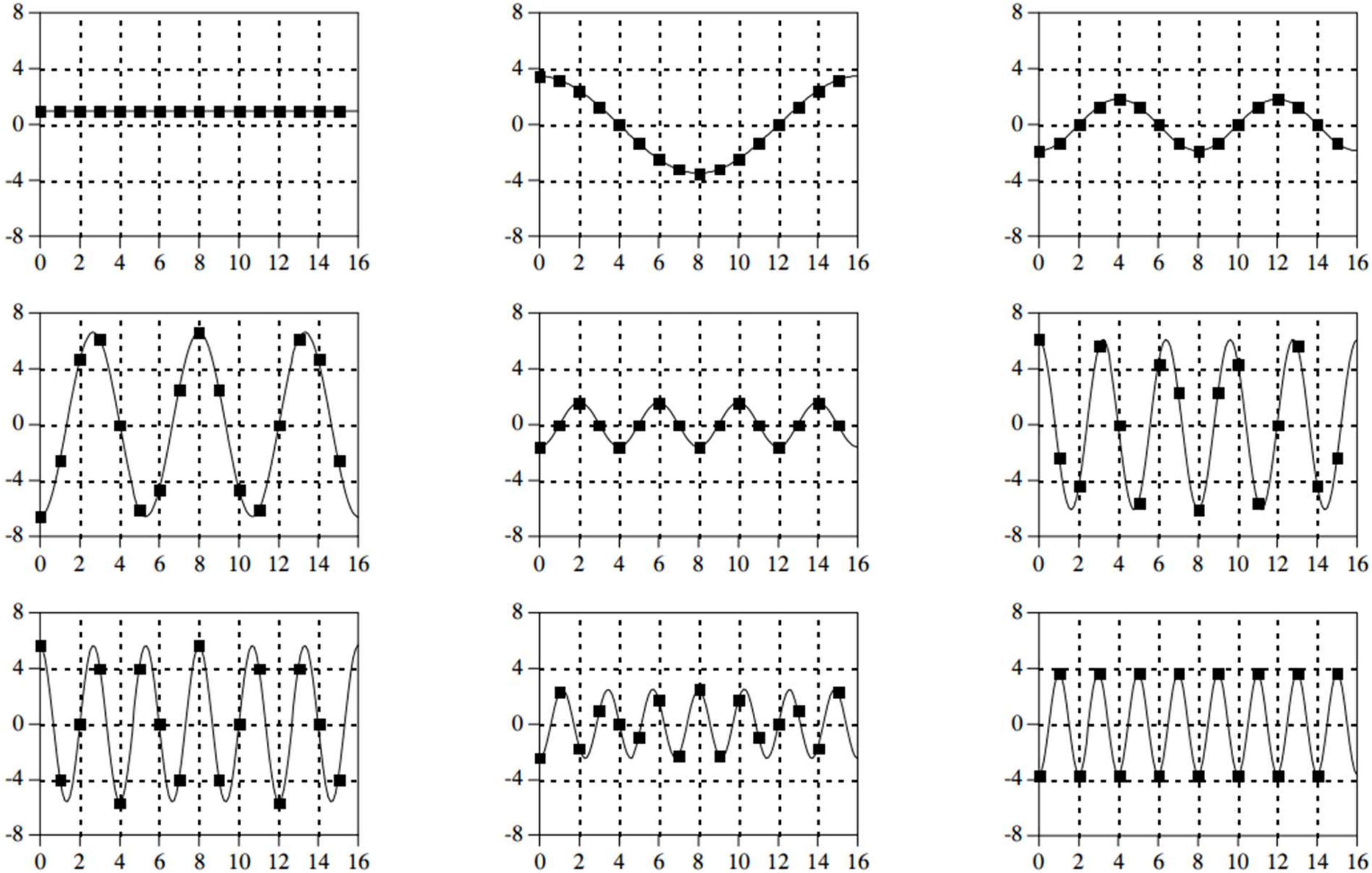
Why are sinusoids used instead of, for instance, square or triangular waves? Remember, there are an infinite number of ways that a signal can be decomposed. The goal of decomposition is to end up with something *easier* to deal with than the original signal. For example, impulse decomposition allows signals to be examined one point at a time, leading to the powerful technique of convolution. The component sine and cosine waves are simpler than the original signal because they have a property that the original signal does not have: *sinusoidal fidelity*. As discussed in Chapter 5, a sinusoidal input to a system is guaranteed to produce a sinusoidal output. Only the amplitude and phase of the signal can change; the frequency and wave shape must remain the same. Sinusoids are the only waveform that have this useful property. While square and triangular decompositions are *possible*, there is no general reason for them to be *useful*.

- DFT expresses any discrete time periodic signal into a sum of some sine and cosine signals

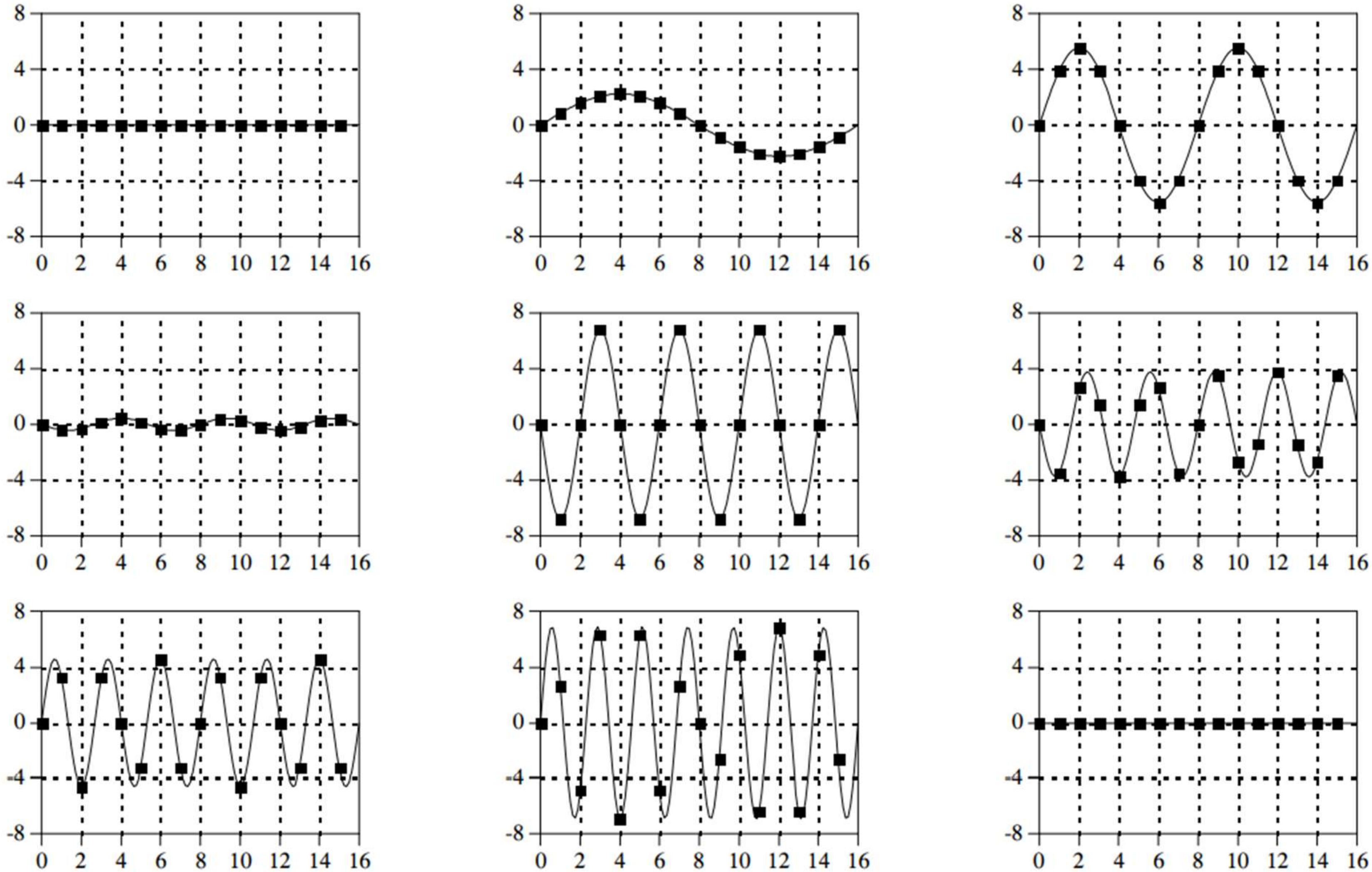


The signal in the figure is continuous, not discrete time periodic; but the idea is the same

Cosine Waves



Sine Waves



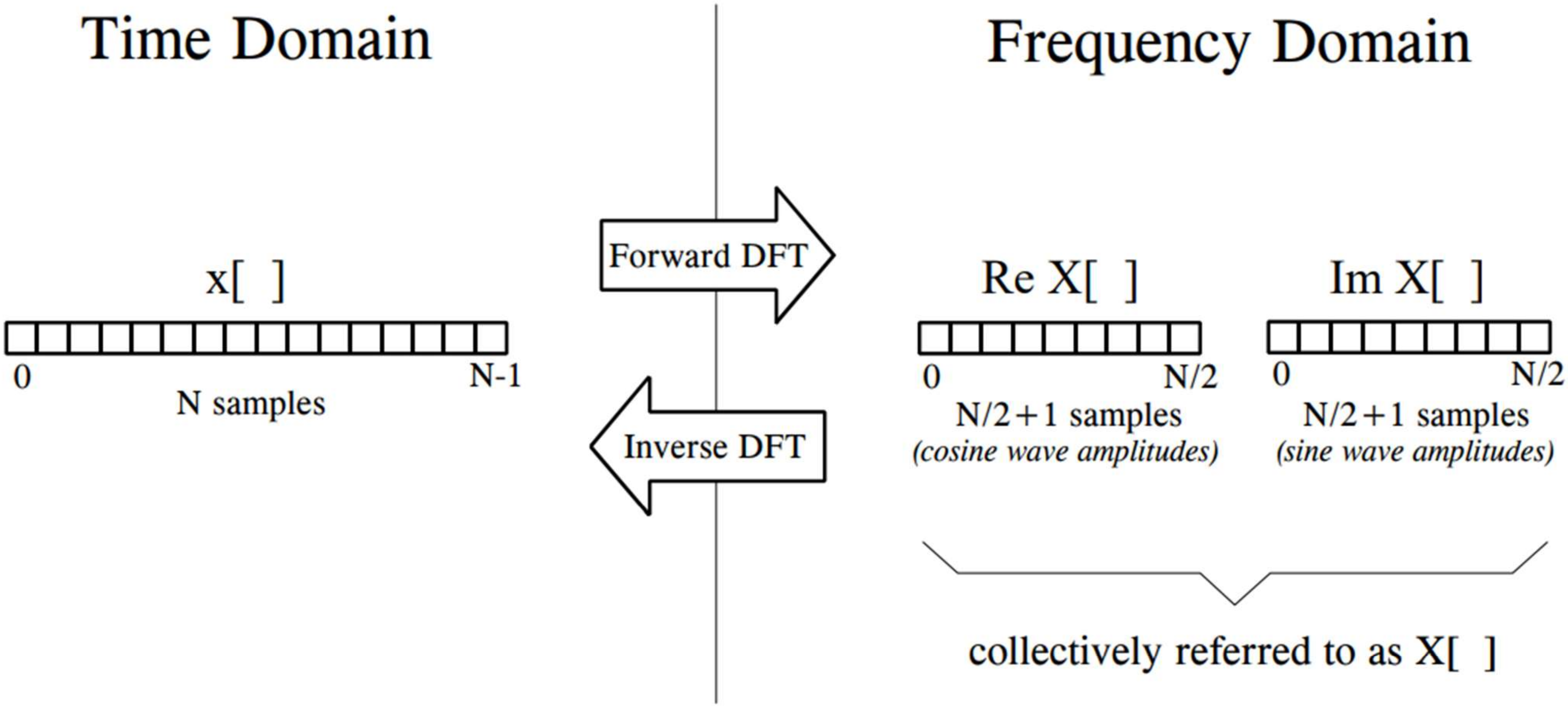


FIGURE 8-3
DFT terminology. In the time domain, $x[\ ]$ consists of N points running from 0 to $N-1$. In the frequency domain, the DFT produces two signals, the real part, written: $Re\,X[\ ]$, and the imaginary part, written: $Im\,X[\ ]$. Each of these frequency domain signals are $N/2+1$ points long, and run from 0 to $N/2$. The Forward DFT transforms from the time domain to the frequency domain, while the Inverse DFT transforms from the frequency domain to the time domain. (Take note: this figure describes the **real DFT**. The **complex DFT**, discussed in Chapter 31, changes N complex points into another set of N complex points).

In the next class,

We will see more details on Discrete Fourier Transform

Thank You.