EE327 Digital Signal Processing Convolution Properties

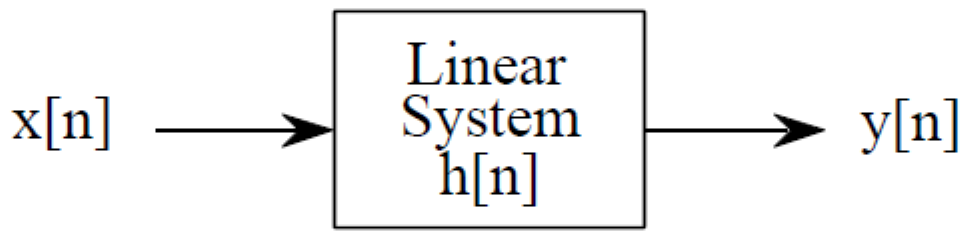
Yasser F. O. Mohammad

REMINDER 1:What is convolution?

- A mathematical operation that takes two signals and produces a third one.
  - $X[n]*Y[n]=Z[n]$
- For us:
  - A way to get the output signal given the input signal and a representation of system function

From now one we will deal only with discrete signals if not otherwise specified

REMINDER 2: How to calculate the output



$$x[n] * h[n] = y[n]$$

- Input length = N
- Impulse Response length = M

- Output length = N+M-1

- For example a 81 points input convolved with a 31 points impulse response gives 111 points output

REMINDER 3: Two ways to understand it

- Input Signal Viewpoint *(Input Side Algorithm)*
  - How each input impulse contributes to the output signal.
  - Good for your understanding

- Output Signal Viewpoint *(Output Side Algorithm)*
  - How each output impulse is calculated from input signal.
  - Good for your calculator

REMINDER 4: Input Side Algorithm

```
100  'CONVOLUTION USING THE INPUT SIDE ALGORITHM
110                                  '
120 DIM X[80]                        'The input signal, 81 points
130 DIM H[30]                        'The impulse response, 31 points
140 DIM Y[110]                       'The output signal, 111 points
150                                  '
160 GOSUB XXXX                       'Mythical subroutine to load X[ ] and H[ ]
170                                  '
180 FOR I% = 0 TO 110                'Zero the output array
190   Y(I%) = 0
200 NEXT I%
210                                  '
220 FOR I% = 0 TO 80                 'Loop for each point in X[ ]
230   FOR J% = 0 TO 30               'Loop for each point in H[ ]
240     Y[I%+J%] = Y[I%+J%] + X[I%]*H[J%]
250   NEXT J%
260 NEXT I%                          '(remember, * is multiplication in programs!)
270                                  '
280 GOSUB XXXX                       'Mythical subroutine to store Y[ ]
290                                  '
300 END
```

REMINDER 5: Output Side Algorithm

```
100  'CONVOLUTION USING THE OUTPUT SIDE ALGORITHM
110                                '
120 DIM X[80]                      'The input signal, 81 points
130 DIM H[30]                      'The impulse response, 31 points
140 DIM Y[110]                     'The output signal, 111 points
150                                '
160 GOSUB XXXX                     'Mythical subroutine to load X[ ] and H[ ]
170                                '
180 FOR I% = 0 TO 110              'Loop for each point in Y[ ]
190   Y[I%] = 0                    'Zero the sample in the output array
200   FOR J% = 0 TO 30             'Loop for each point in H[ ]
210     IF (I%-J% < 0)             THEN GOTO 240
220     IF (I%-J% > 80)            THEN GOTO 240
230     Y(I%) = Y(I%) + H(J%) * X(I%-J%)
240   NEXT J%
250 NEXT I%
260                                '
270 GOSUB XXXX                     'Mythical subroutine to store Y[ ]
280                                '
290 END
```

Common Impulse Responses

- Identity System: $$x[n]*\delta[n]=x[n]$$
- Amplifier/Attenuator: $$x[n]*k\times\delta[n]=k\times x[n]$$
- Delay/Shift: $$x[n]*\delta[n+s]=x[n+s]$$
- Echo: $$x[n]*\left(\delta[n]+\delta[n+s]\right)=x[n]+x[n+s]$$

Discretizing Calculus

- First Difference :

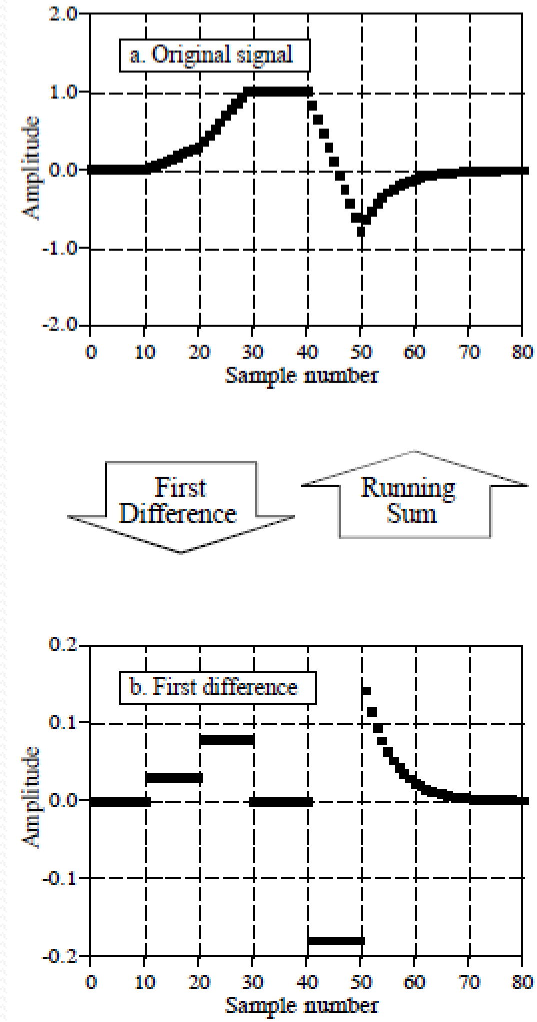
$$y[n] = x[n] - x[n-1]$$

- Discrete equivalent of differentiation
  - Discrete Derivative

- Running Sum:

$$y[n] = \sum_{i=0}^{n} x[i] = x[n] + y[n-1]$$

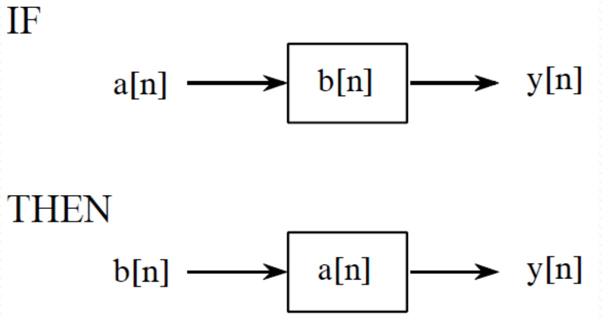
  - Discrete equivalent of integration
  - Discrete Integral



Properties of Convolution

- Commutative Property:

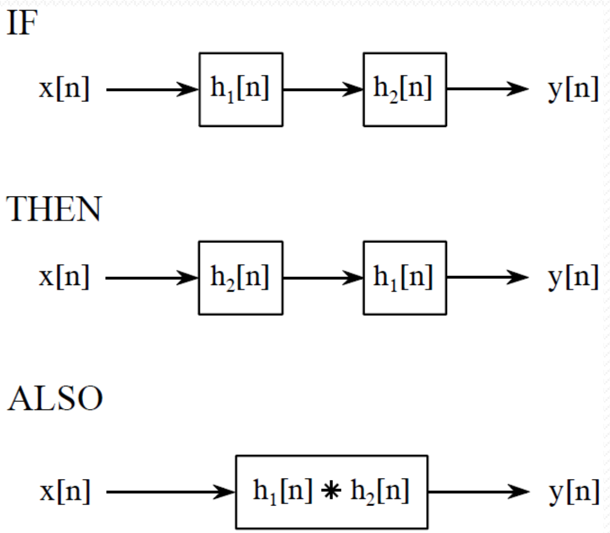
$$a[n]*b[n]=b[n]*a[n]$$



- Associative Property:

$$a[n]*\left(b[n]*c[n]\right)=$$

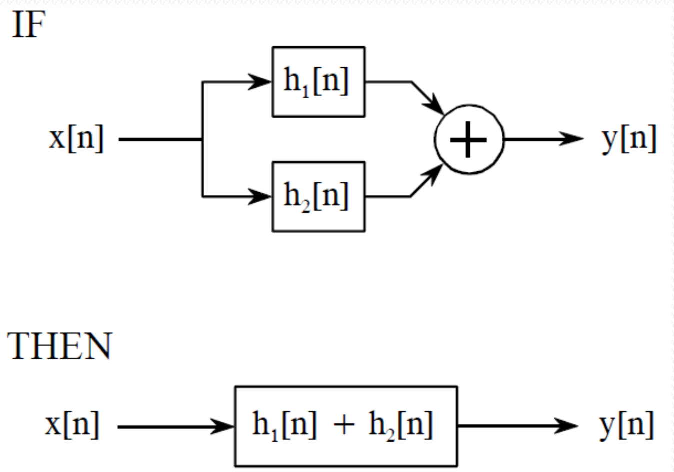
$$\left(a[n]*b[n]\right)*c[n]$$



Properties of Convolution (2)

- Distributive Property:

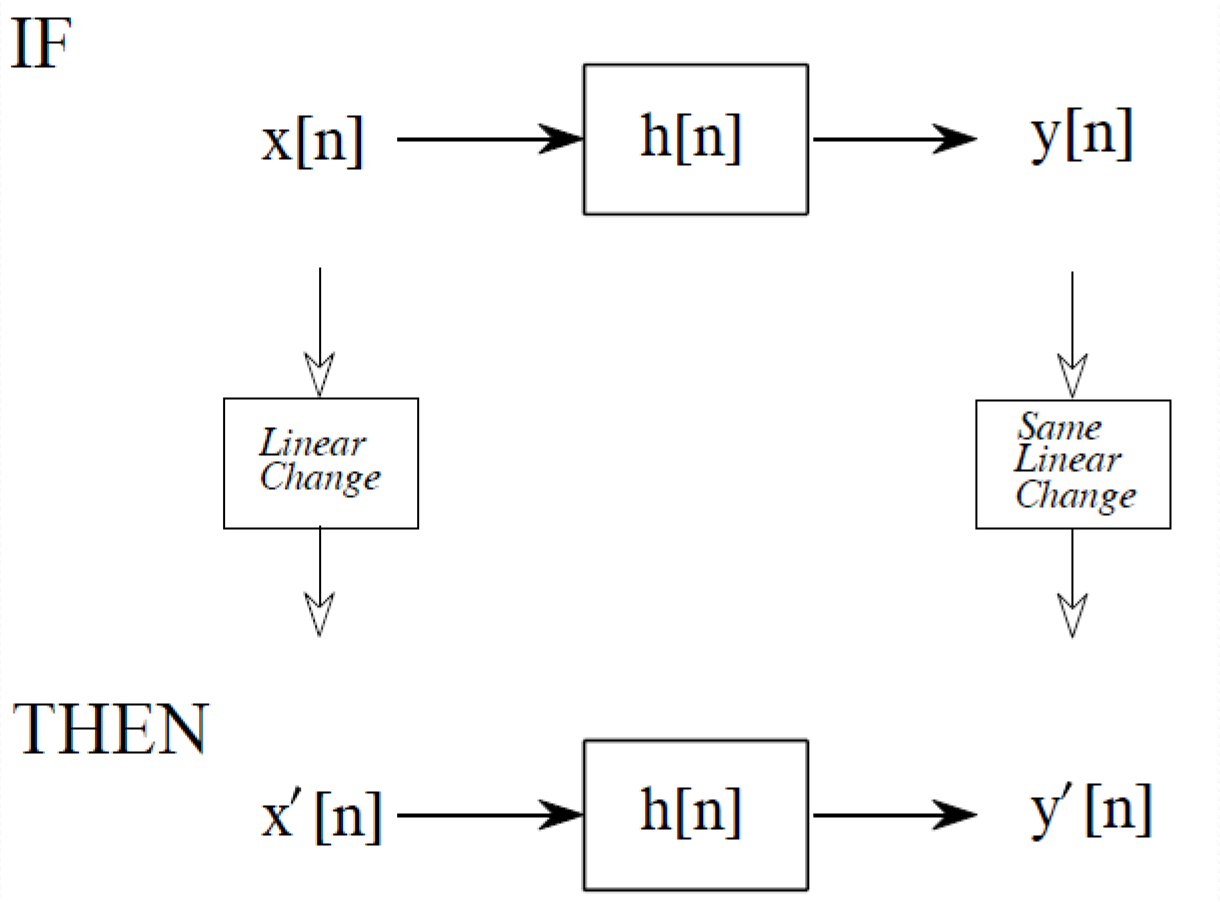
$$a[n]*b[n]+a[n]*c[n]=$$
$$a[n]*\left(b[n]+c[n]\right)$$



- Central Limit Theory:

$$a[n]*a[n]*a[n]*a[n]*\ldots\ldots=N\left(\mu,\sigma\right)$$

Transference Between Input and Output



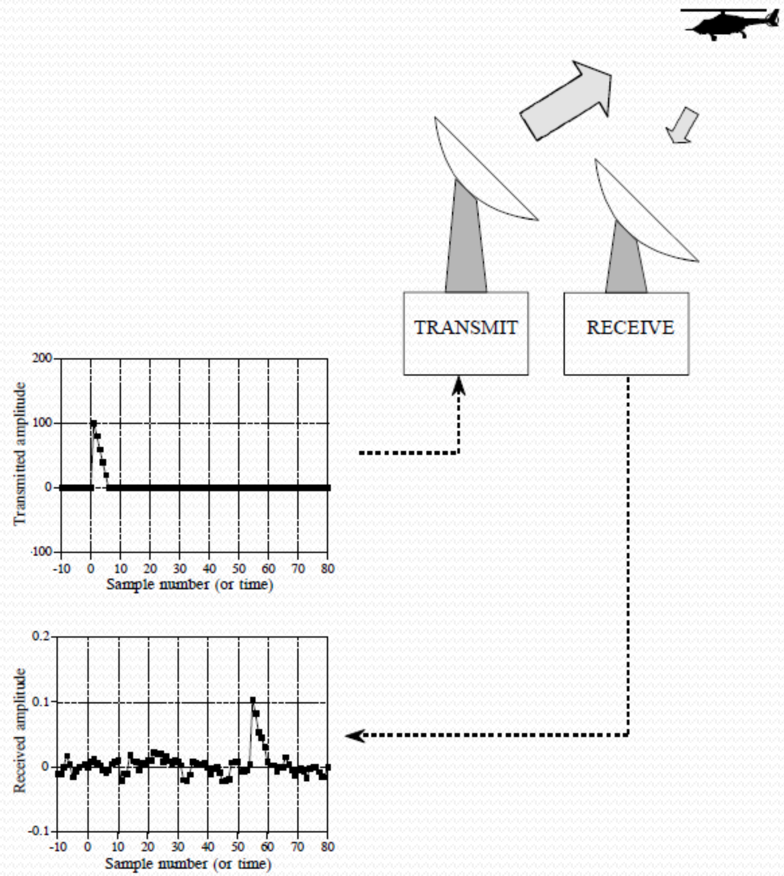
Correlation

- Cross Correlation

$$c_{ab}[j] = \sum_{i=0}^{M-1} a[j]b[i-j]$$

- Self Correlation

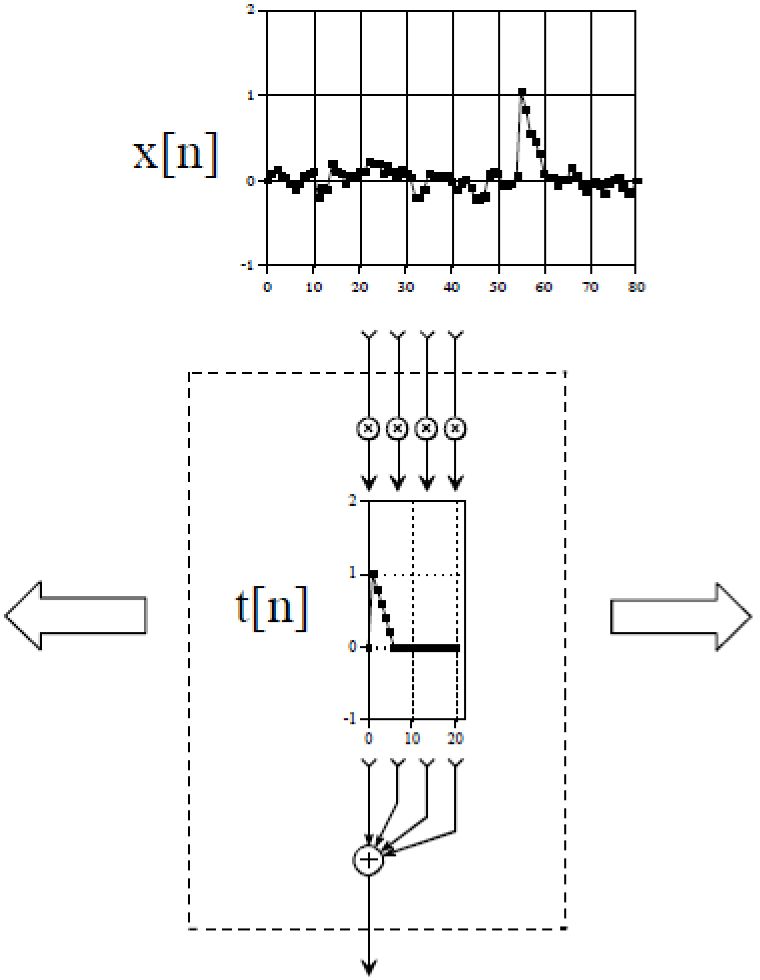
$$c_{a}[j] = \sum_{i=0}^{M-1} a[j]a[i-j]$$



- Measures similarity *(if correctly normalized!!!!)*

Calculating Correlation

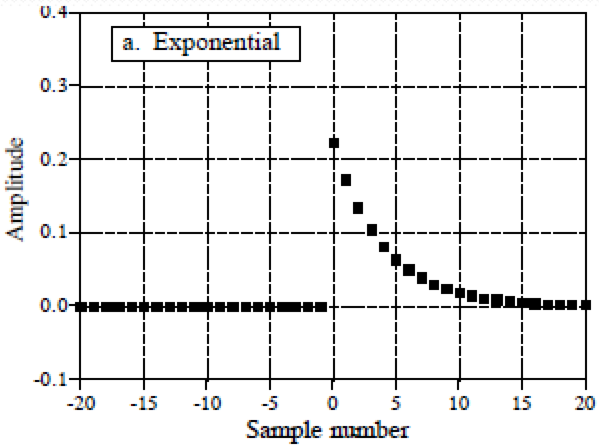
- Convolution without flipping second signal

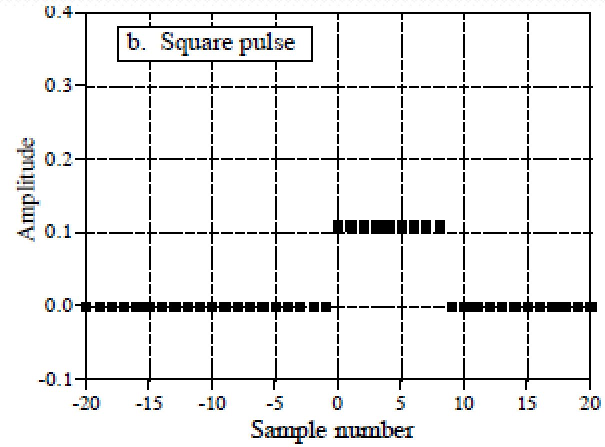


Just Touching Filters

- Major Types
  - Low Pass Filter
  - High Pass Filter

Low Pass Filters





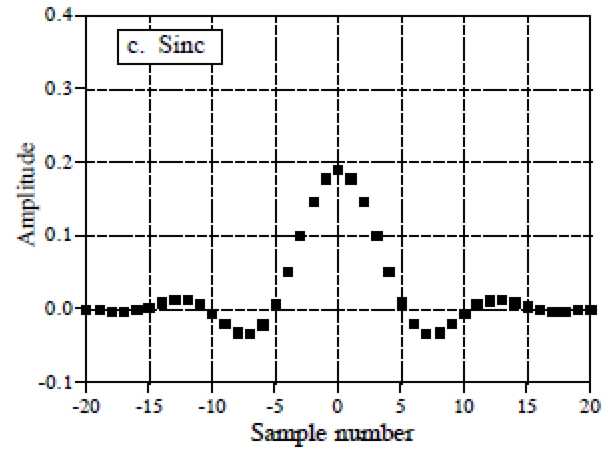


FIGURE 7-4
Typical low-pass filter kernels. Low-pass filter kernels are formed from a group of adjacent positive points that provide an averaging (smoothing) of the signal. As discussed in later chapters, each of these filter kernels is best for a particular purpose. The exponential, (a), is the simplest recursive filter. The rectangular pulse, (b), is best at reducing noise while maintaining edge sharpness. The sinc function in (c), a curve of the form: $\sin(x)/(x)$, is used to separate one band of frequencies from another.

High Pass Filters

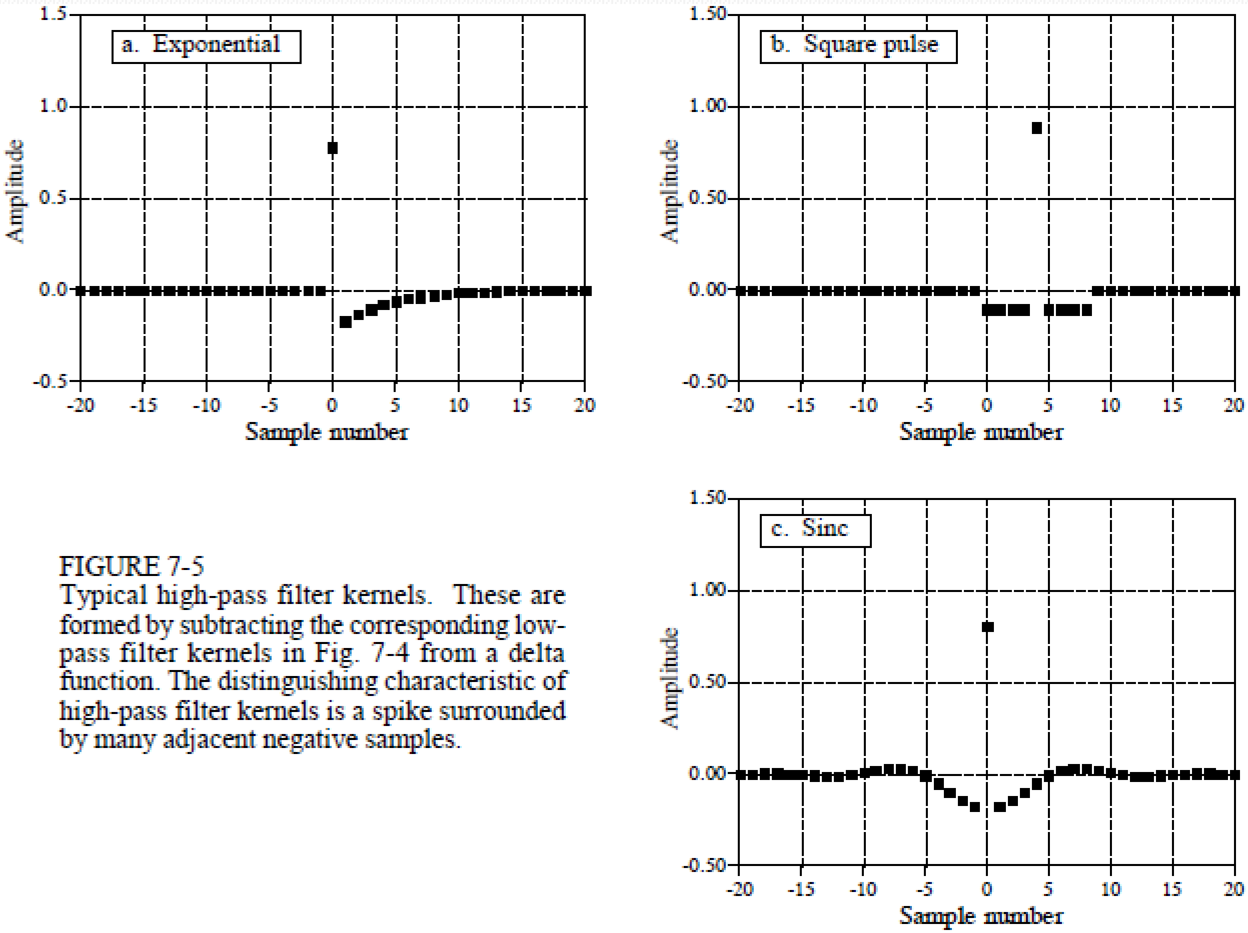


FIGURE 7-5
Typical high-pass filter kernels. These are formed by subtracting the corresponding low-pass filter kernels in Fig. 7-4 from a delta function. The distinguishing characteristic of high-pass filter kernels is a spike surrounded by many adjacent negative samples.

Causal Systems (from Impulse Response)

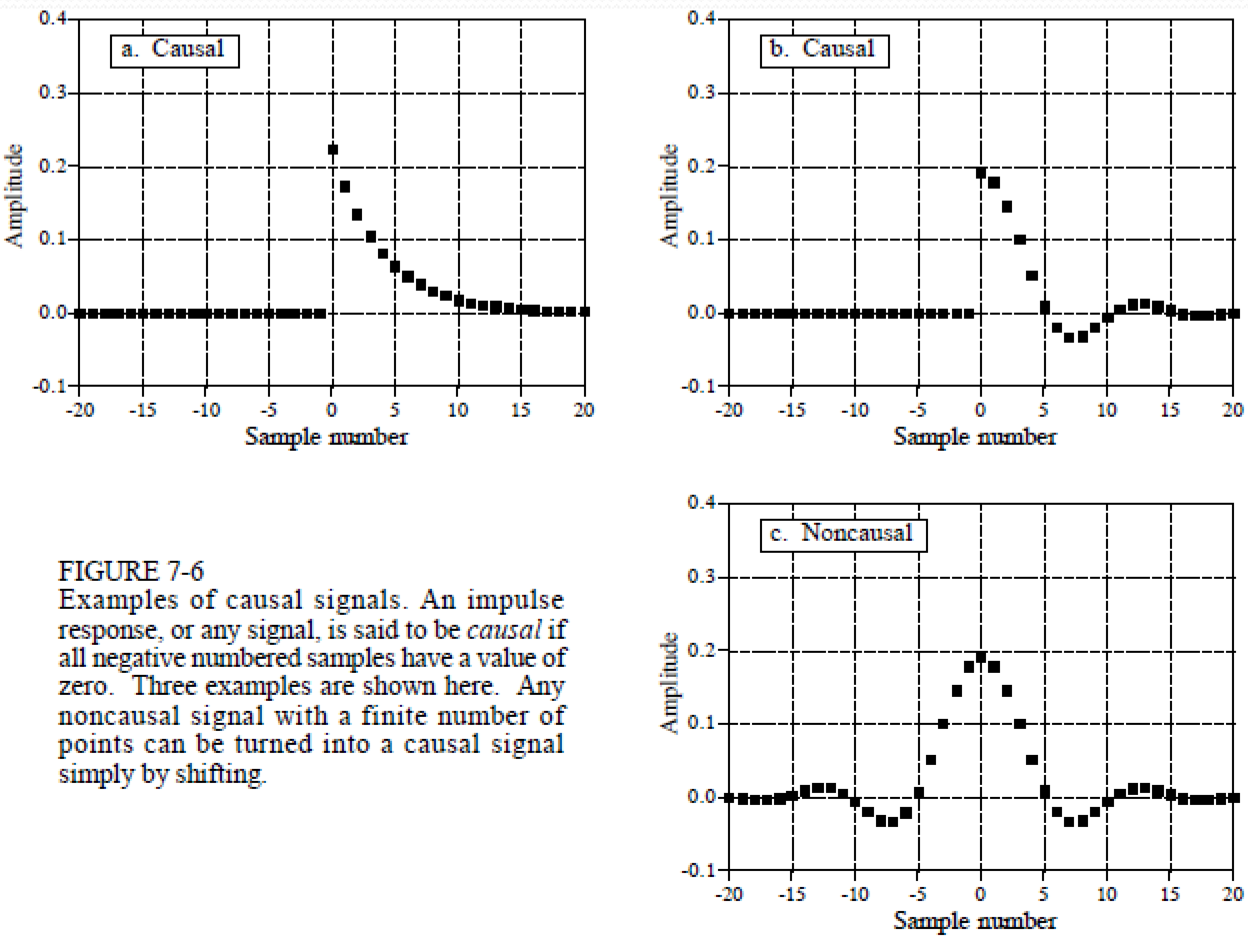
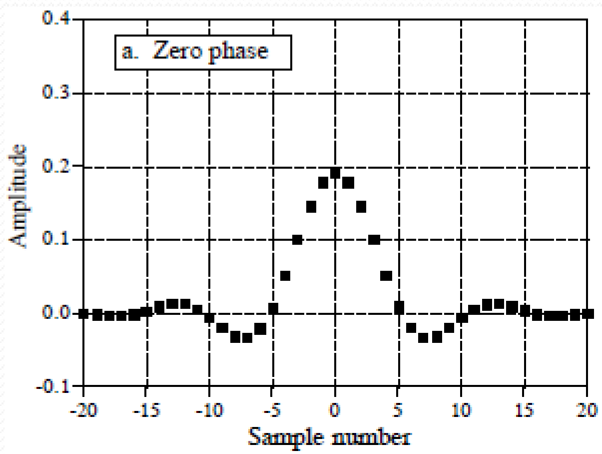


FIGURE 7-6
Examples of causal signals. An impulse response, or any signal, is said to be *causal* if all negative numbered samples have a value of zero. Three examples are shown here. Any noncausal signal with a finite number of points can be turned into a causal signal simply by shifting.

Zero, Linear and nonlinear Phase Systems

Symmetric around 0

Symmetric around another point



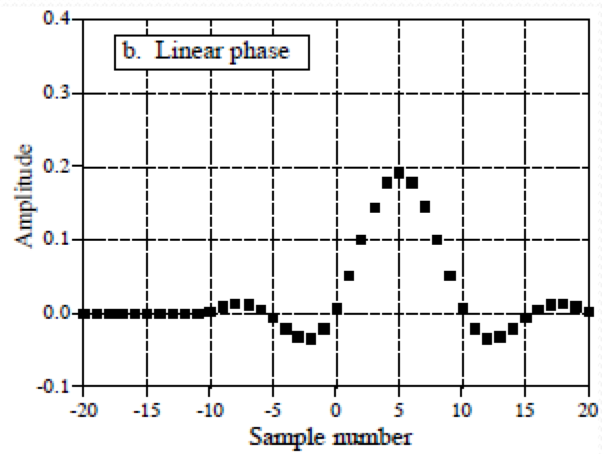
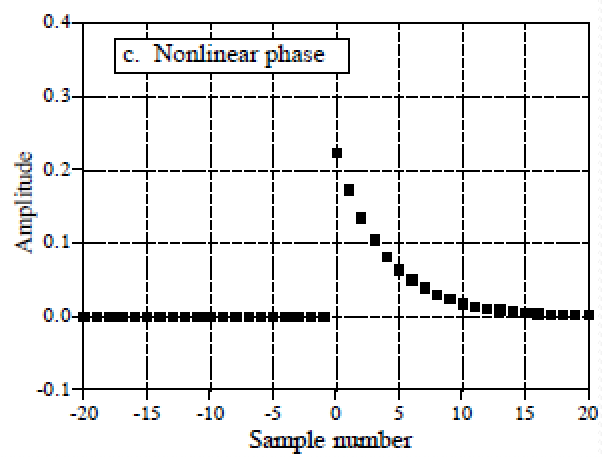


FIGURE 7-7
Examples of phase linearity. Signals that have a left-right symmetry are said to be *linear phase*. If the axis of symmetry occurs at sample number zero, they are additionally said to be *zero phase*. Any linear phase signal can be transformed into a zero phase signal simply by shifting. Signals 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.



Asymmetric