

*A very*

# Brief Introduction to Signals & Systems

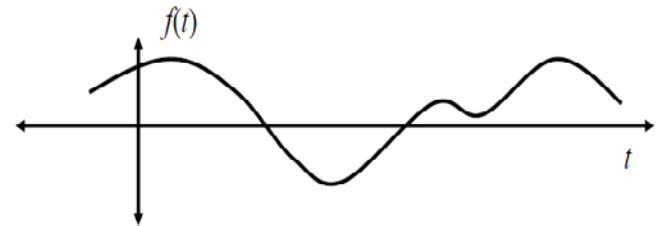
Phani Chavali

# Outline

- Signals & Systems
- Continuous and discrete time signals
- Properties of Systems
- Input- Output relation : **Convolution**
- **Frequency domain representation of signals & systems**
- Analog to digital Conversion
- Sampling – **Nyquist Sampling Theorem**
- Basic Filter Theory
- Types of filters
- **Designing practical filters in Labview and Matlab**

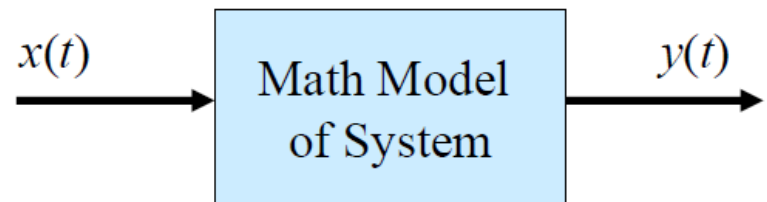
- What is a signal?

- A signal is a function defined on the continuum of time values



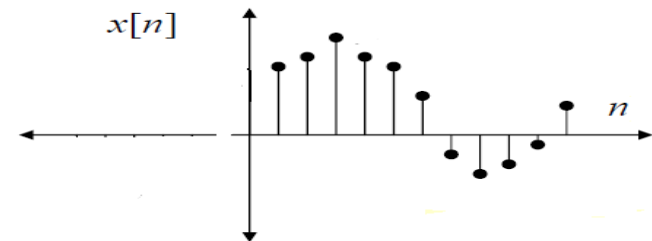
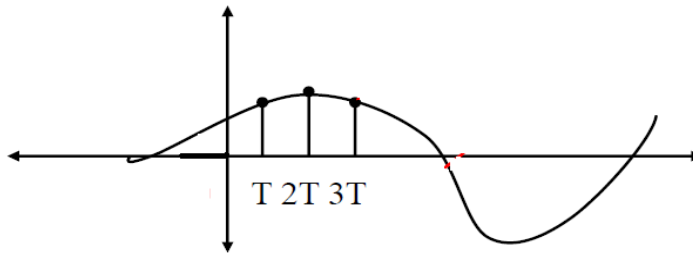
- What is a system ?

- a system is a black box that “takes in” one or more input signals and “produces” one or more output signals



# Continuous time Vs Discrete time Signals

- Most of the modern day systems are discrete time systems. E.g., A computer.
- A computer can't directly process a continuous time signal but instead it needs a stream of numbers, which is a **discrete time signal**.



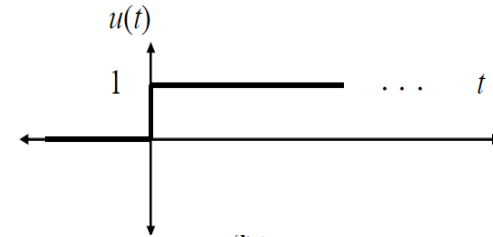
- Discrete time signals are obtained by sampling the continuous time signals
- How fast should we sample the signal?

# Examples

- Signals

- Unit Step function

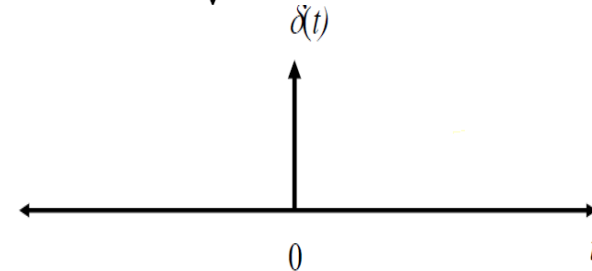
$$u(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$$



- Continuous time impulse function

$$\delta(t) = 0, \text{ for any } t \neq 0$$

$$\int_{-\varepsilon}^{\varepsilon} \delta(t) dt = 1, \text{ for any } \varepsilon > 0$$

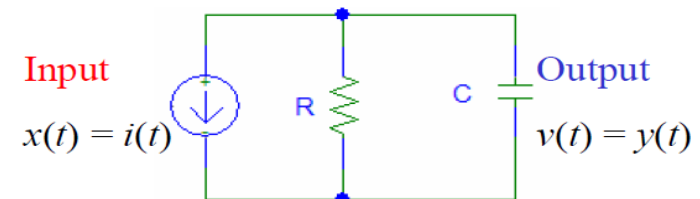


- Discrete time

$$\delta[n] = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$

- Systems

- A simple circuit



$$\sum_{n=-\infty}^{\infty} \delta[n] = 1$$



$$\int_{-\infty}^{\infty} \delta(t) dt = 1$$

$$\sum_{n=-\infty}^{\infty} x[n] \delta[n - n_0] = x[n_0]$$

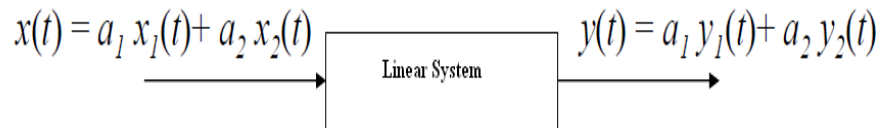
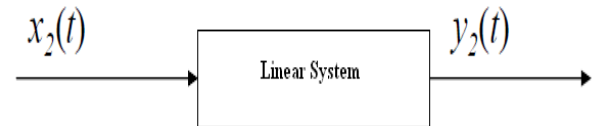
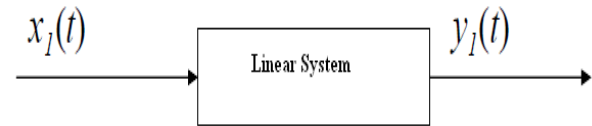


$$\int_{-\infty}^{\infty} x(t) \delta(t - t_0) dt = x(t_0)$$

# Basic System Properties

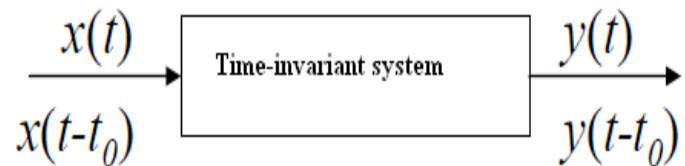
- Linearity

- System is linear if the principle of superposition holds



- Time- Invariance

- The system does not change with time



# Convolution

- Linear & Time invariant (LTI) systems are characterized by their impulse response
- Impulse response is the output of the system when the input to the system is an impulse function
- For Continuous time signals  $y(t) = \int_{-\infty}^{\infty} x(s) h(t - s) ds.$
- For Discrete time signals  $y[n] = \sum_{i=-\infty}^{\infty} x[i]h[n - i]$

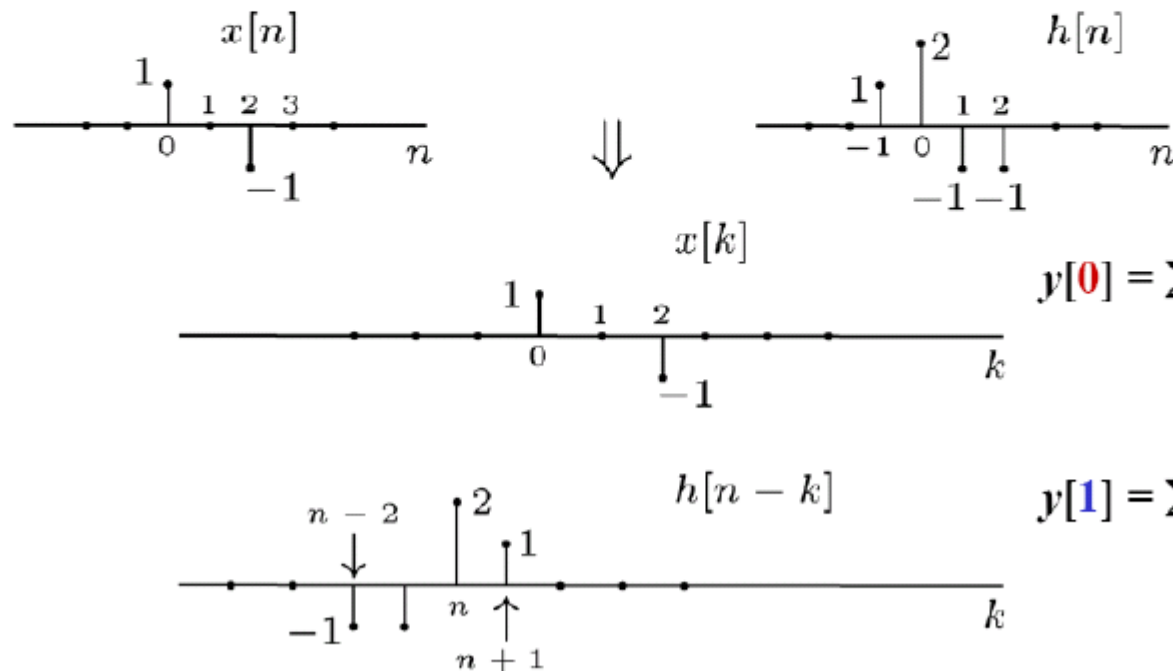


# Visualizing the calculation of $y[n] = x[n] * h[n]$

Choose value of  $n$  and consider it fixed

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

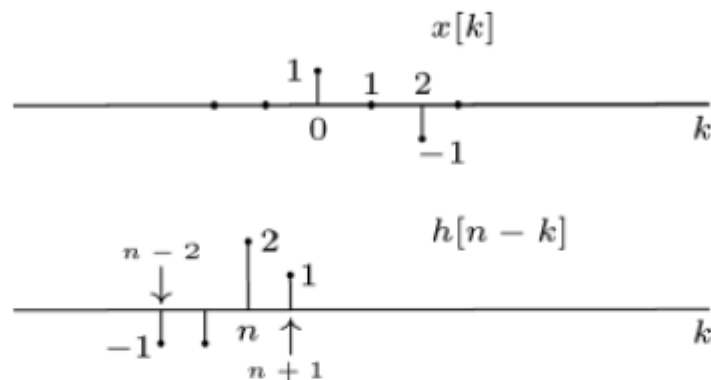
View as functions of  $k$  with  $n$  fixed



$y[0] = \sum$  prod of  
overlap for  
 $n = 0$

$y[1] = \sum$  prod of  
overlap for  
 $n = 1$

## Calculating Successive Values: Shift, Multiply, Sum



$$\begin{aligned}
 y[n] &= 0 \quad \text{for } n < -1 \\
 y[-1] &= 1 \times 1 = 1 \\
 y[0] &= 0 \times 1 + 1 \times 2 = 2 \\
 y[1] &= (-1) \times 1 + 0 \times 2 + 1 \times (-1) = -2 \\
 y[2] &= (-1) \times 2 + 0 \times (-1) + 1 \times (-1) = -3 \\
 y[3] &= (-1) \times (-1) + 0 \times (-1) = 1 \\
 y[4] &= (-1) \times (-1) = 1 \\
 y[n] &= 0 \quad \text{for } n > 4
 \end{aligned}$$

# Frequency domain representation of signals

- In most of the real time applications it will be required to process the signals based on their frequencies
- In such cases, it is easier to represent the signals as a function of the frequency, rather than time
- A Fourier transform provides the mathematical representation

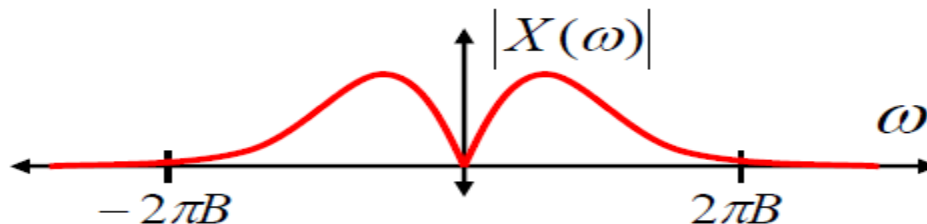
$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t} d\omega$$

**Fourier Transform**

**Inverse Fourier Transform**

# Bandwidth of the signal

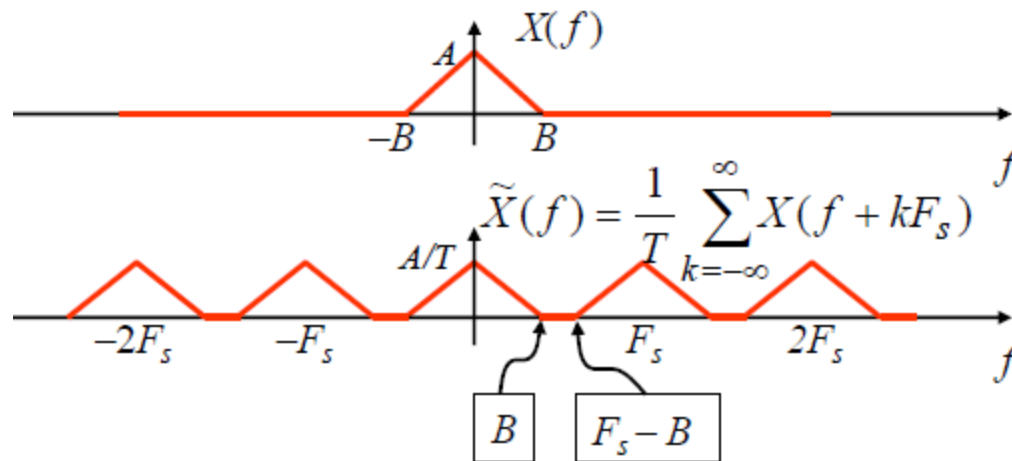
- For a lot of signals –like audio –they fill up the lower frequencies but then decay as  $\omega$  gets large



- We say the signal's BW =  $B$  in Hz if there is "negligible" content for  $|\omega| > 2\pi B$

# Nyquist Sampling Theorem

- For band limited analog signals, sampling frequency should be at least twice the bandwidth to avoid aliasing.



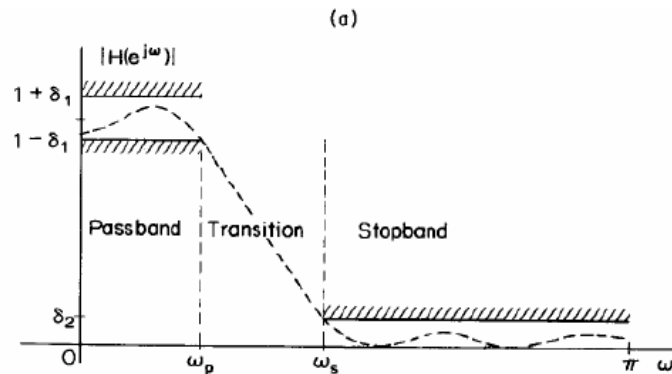
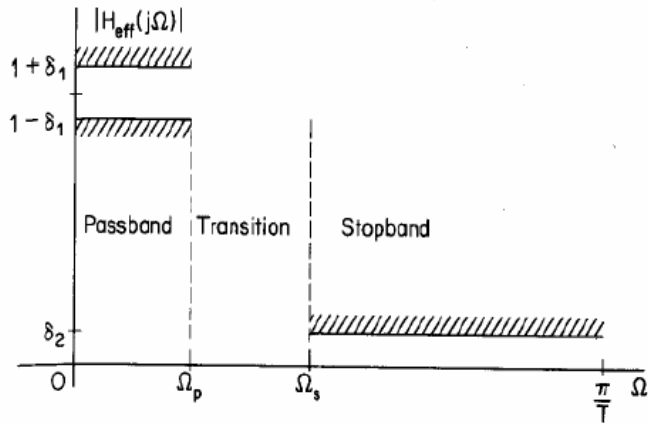
# Filters – Introduction

- Filtering is the most common signal processing procedure.
  - Used as echo cancellers, equalizers, front end processing in RF receivers
  - Used for modifying input signals by passing certain frequencies and attenuating others.
- Characterized by the impulse response like other Linear & Time Invariant systems.
- Both Analog and Digital Filters can be used.
- Analog
  - Uses analog electronic circuits made up of components like resistors and capacitors
  - Used widely for video enhancement in TV's
- Digital
  - Uses a general purpose processor for implementation
  - Used widely in many applications these days because of the flexibility they offer in design and implementation

# Types of Filters

- High pass filter
  - Attenuates the low frequency components of a signal and allows high frequency components
- Low pass filter
  - Attenuates the high frequency component and allows low frequency component
- Band pass filter
  - Allows a particular frequency band and attenuates the rest of the frequency components.
- Band stop filter
  - Attenuates the frequency components in a particular band and allows the other frequencies.

# Filter Design



(From Discrete-Time Signal Processing, Oppenheim and Schaffer)

- $\Omega_p$  is the Passband frequency
- $\Omega_s$  is the Stopband frequency
- $\delta_1$  is the Passband Ripple
- $\delta_2$  is the Stopband Attenuation



# FIR Vs IIR Filters

- Several factors influence the choice of FIR / IIR filters like linear phase, stability, hardware required to build etc.

$$y[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k y[n-k] \quad \text{IIR filter equation}$$

$$y[n] = \sum_{k=0}^M b_k x[n-k] \quad \text{FIR filter equation}$$

- Several techniques for designing filters (both FIR & IIR)
- We don't learn the design techniques in this class. We use Matlabas a design tool
- IIR filter types
  - Butterworth : Maximally flat
  - Chebycheff : Equi-ripple in pass band (type 1) & stop band (type 2)
  - Elliptical : Sharp transition region

# Some Matlab Commands

- **plot**
  - PLOT(Y) plots the columns of Y versus their index. PLOT(X,Y) plots vector Y versus vector X.
- **fir1**
  - B = FIR1(N,Wn) designs an Nth order lowpass FIR digital filter and returns the filter coefficients in length N+1 vector B. B = FIR1(N,Wn,'high') designs an Nth order highpass filter.
- **butter**
  - [B,A] = BUTTER(N,Wn) designs an Nth order lowpass digital Butterworth filter and returns the filter coefficients in length N+1 vectors B (numerator) and A (denominator).
- **cheby1**
  - [B,A] = CHEBY1(N,R,Wp) designs an Nth order lowpass digital Chebyshev filter with R decibels of peak-to-peak ripple in the passband. CHEBY1 returns the filter coefficients in length N+1 vectors B (numerator) and A (denominator). Use R=0.5 as a starting point, if you are unsure about choosing R
- See also cheby2 & ellip
- **filter**
  - Y = FILTER(B,A,X) filters the data in vector X with the filter described by vectors A and B to create the filtered data Y where A and B are as in direct form II structure

# Task

- Create a signal which is sum of two sinusoids with frequencies 5Hz and 15 Hz.
- Plot  $x(t)$  and  $X(f)$ . Use time and frequency as x-axis while plotting, not the sample number.
- Create an FIR low pass filter with cutoff frequency 6Hz and plot the response of the filter. Change the order of filter and see how the frequency response changes.
- Pass the signal  $x(t)$  through the filter and plot the output.
- Create an FIR high pass filter with cutoff frequency 12 Hz and plot the response of the filter. Repeat for different orders.
- Pass the signal  $x(t)$  through the filter and plot the output.
- Repeat the experiment with an IIR filters of same order and see the performance difference