Filters

Phani Chavali

Filters

- 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

Impulse Response & Frequency response

- The elements h[n] are called taps;depending on whether h[n] is a finite sequence or an infinte sequence, we call the filters as finite impulse response (FIR) or infinte impulse response (IIR) filters.
- Given the impulse response, the output of the filter y[n] for any input x[n] can be expressed as

  $$y[n]=x[n]*h[n]=\sum x[k]h[n-k]$$

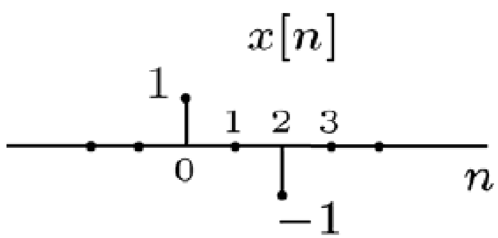
  - Reverse h in time around zero
  - y[n] is the dot product of x reversed h shifted n places to the left.
- The Fourier transform of the signal x[n] is called as Frequency transform.
  - Periodic with period 2π

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

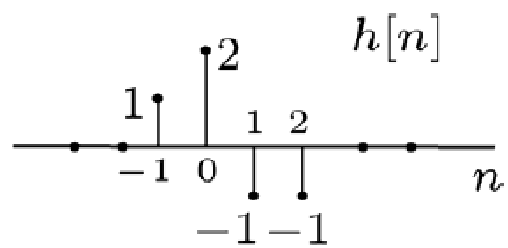
Choose value of n and consider it fixed

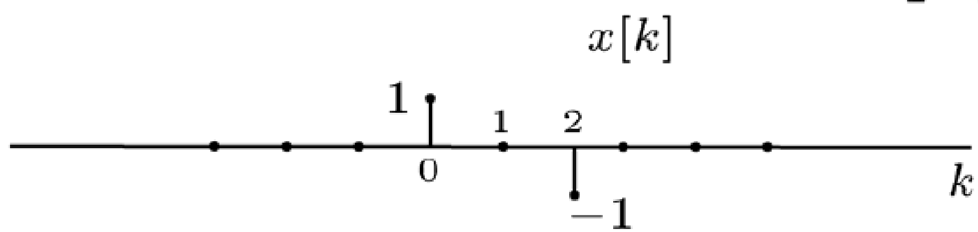
$$y[n] = \sum_{k=\infty}^{\infty} x[k]h[n-k]$$

View as functions of k with n fixed

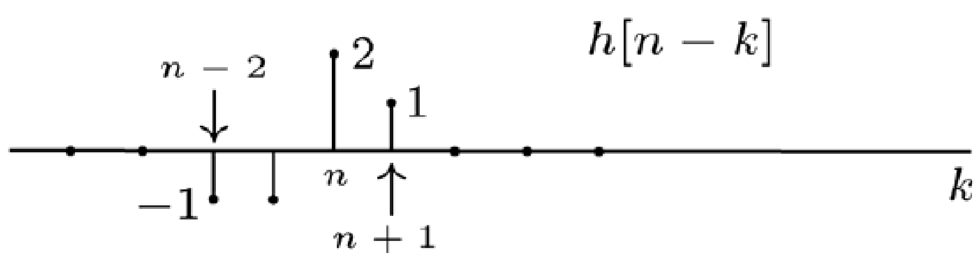


$\Downarrow$



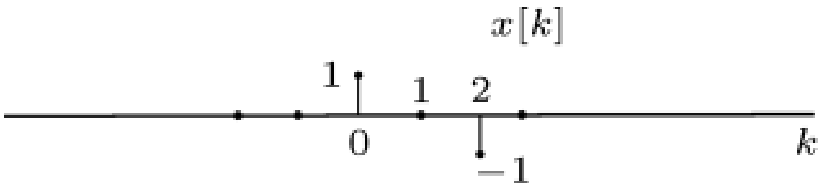


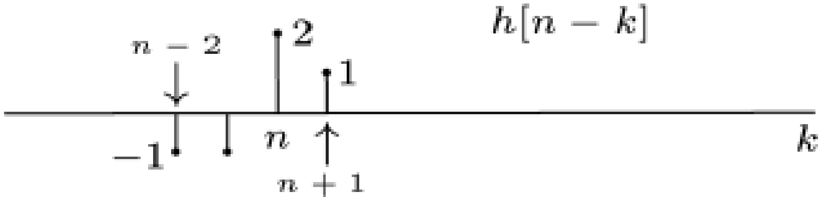
$y[0] = \Sigma$ prod of overlap for $n = 0$



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

Calculating Successive Values: Shift, Multiply, Sum



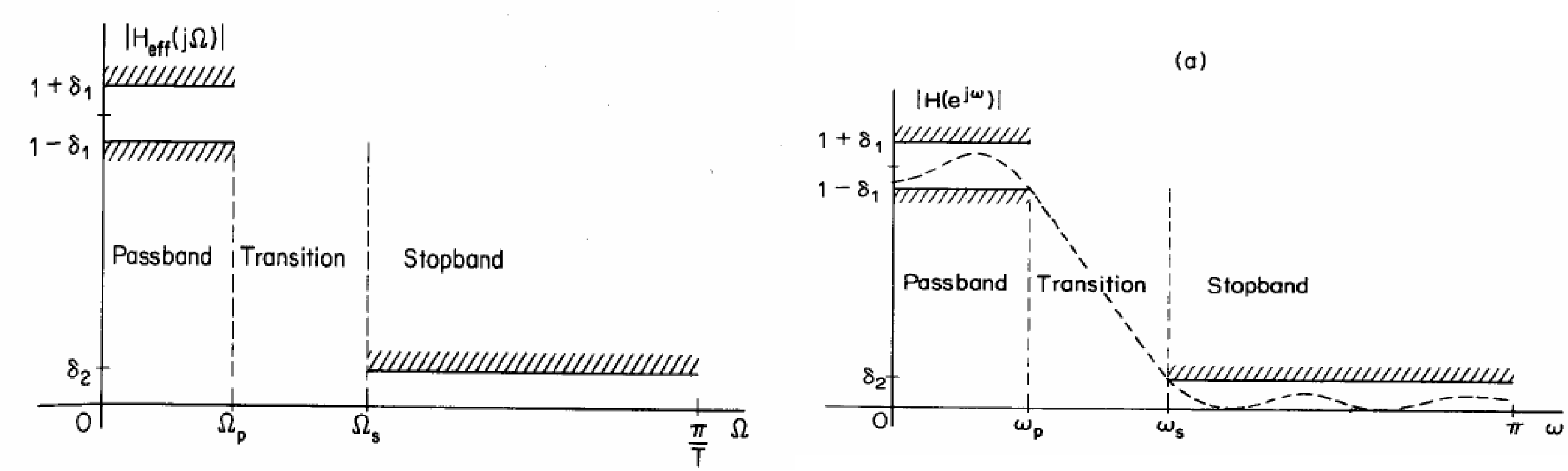


$$
\begin{aligned}
y[n] &= 0 \quad \text{for } n < -1 \\
y[-1] &= 1 \times 1 = 1 \\
y[0] &= 0 \times 1 + 1 \times 2 = 2 \\
y[1] &= (-1) \times 1 + 0 \times 2 + 1 \times (-1) = -2 \\
y[2] &= (-1) \times 2 + 0 \times (-1) + 1 \times (-1) = -3 \\
y[3] &= (-1) \times (-1) + 0 \times (-1) = 1 \\
y[4] &= (-1)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

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 Schafer)

- Ω_p is the Passband frequency
- Ω_s is the Stopband frequency
- δ_1 is the Passband Ripple
- δ_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.

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

IIR filter equation

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

FIR filter equation

- Several techniques for designing filters (both FIR & IIR)
  - We don't learn the design techniques in this class. We use Matlab as 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 N'th order lowpass FIR digital filter and returns the filter coefficients in length N+1 vector B. B = FIR1(N,Wn,'high') designs an N'th 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