Digital Signal Processing

Digital Filters

Topics

- Representations of Filter Performance
- Time Domain Parameters
- Frequency Domain Parameters
- Spectral Inversion and Spectral Reversal
- High Pass, Band Pass and Band Reject Filters
- Filter Classification

Purpose of Filters

- Digital Filters are used for two general purposes
  - Signal separation of combined signals
  - Signal restoration (e.g distorted signals)

- Digital filters can have an order of magnitude better performance than analog filters

Purpose of Filters

- Restoration Example
  - Some frequencies ranges of a signal have been attenuated
  - An equalization filter can restore attenuated frequencies.

- Separation Example
  - Removal of 60 Hz “hum” picked up on a microphone on a public address system

Filter Domains

- A Filter’s input and output are often described as being in the time domain.
- Generally, the signal may be in the time domain or spatial domain….. or both.
  - An image (e.g. MPEG movie) has both spatial and time dependent information.

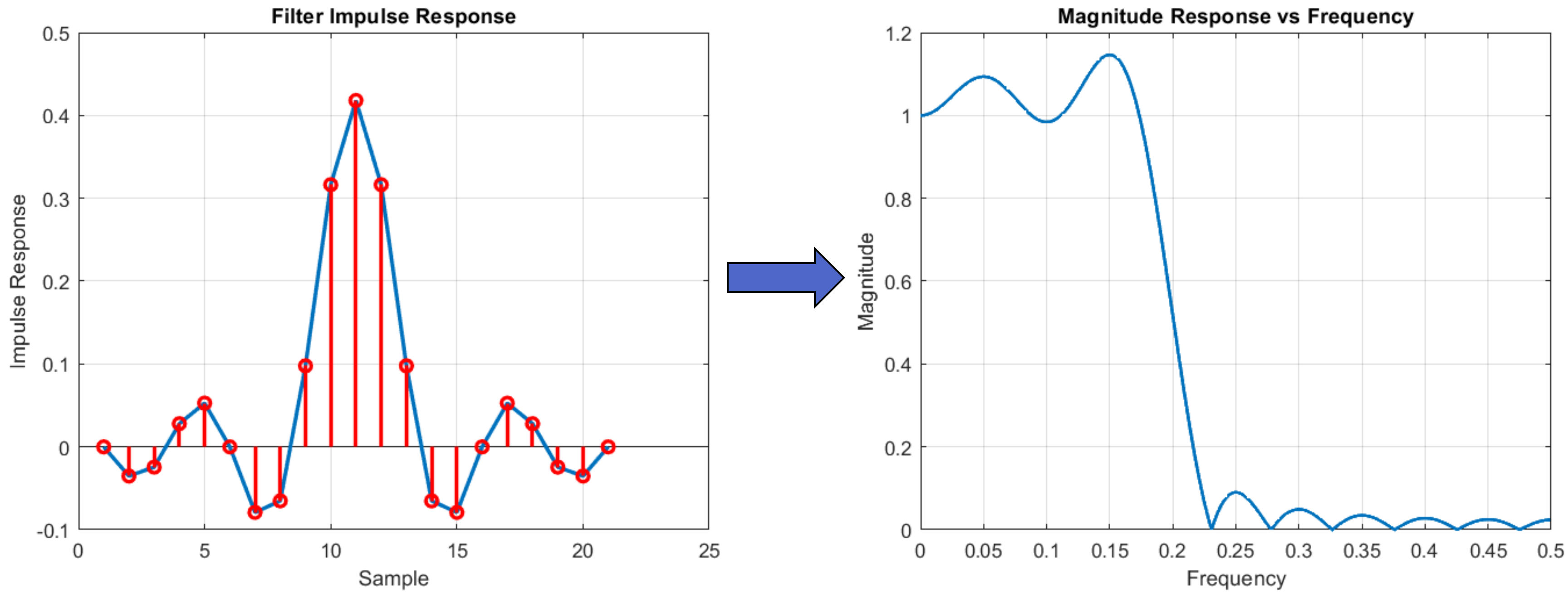
Representing the Filter

- A filter can be described by its
  - Impulse response
  - Step response
  - Frequency response (magnitude and phase)

- Each of these contains complete information about the filter.

Representation of Filters

- The impulse response and frequency response of a filter



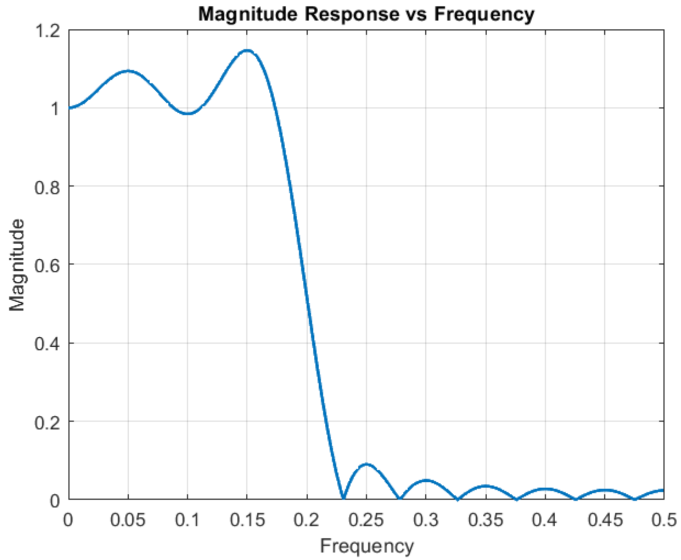
Magnitude and Phase required

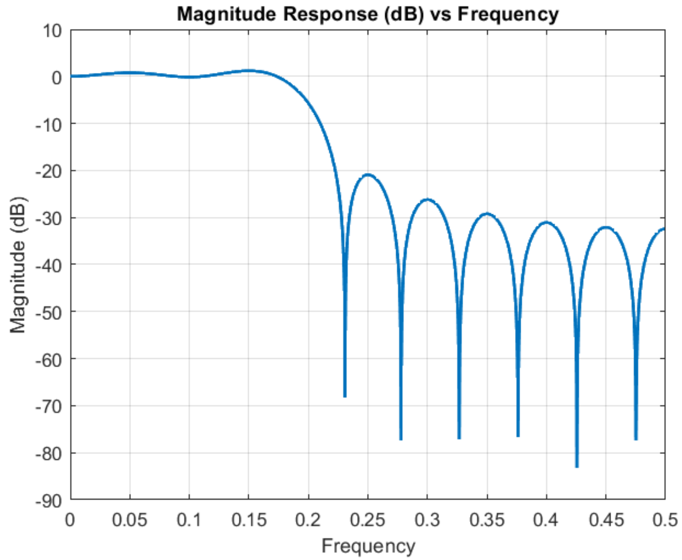
Linear and Decibel Magnitude Plots

- When plotting the frequency response the magnitude is often expressed in decibels

$$G_{dB} = 20\log_{10} G_{num}$$

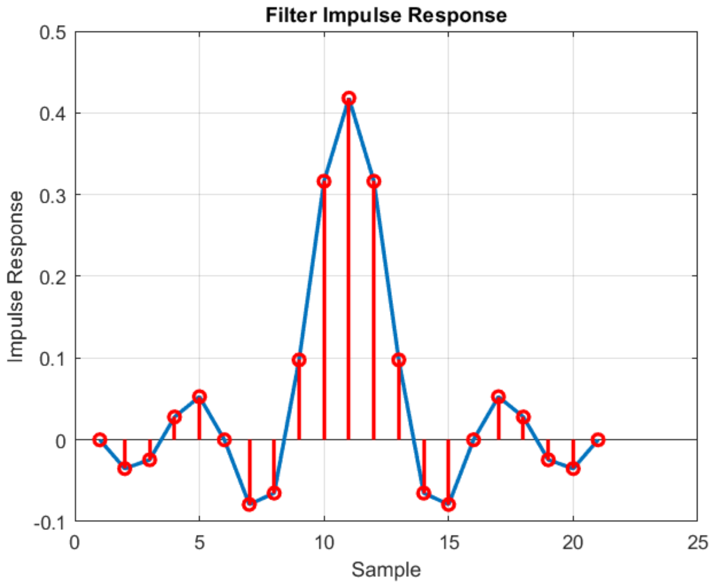
- The numerical gain is referenced to 1 (0 dB)

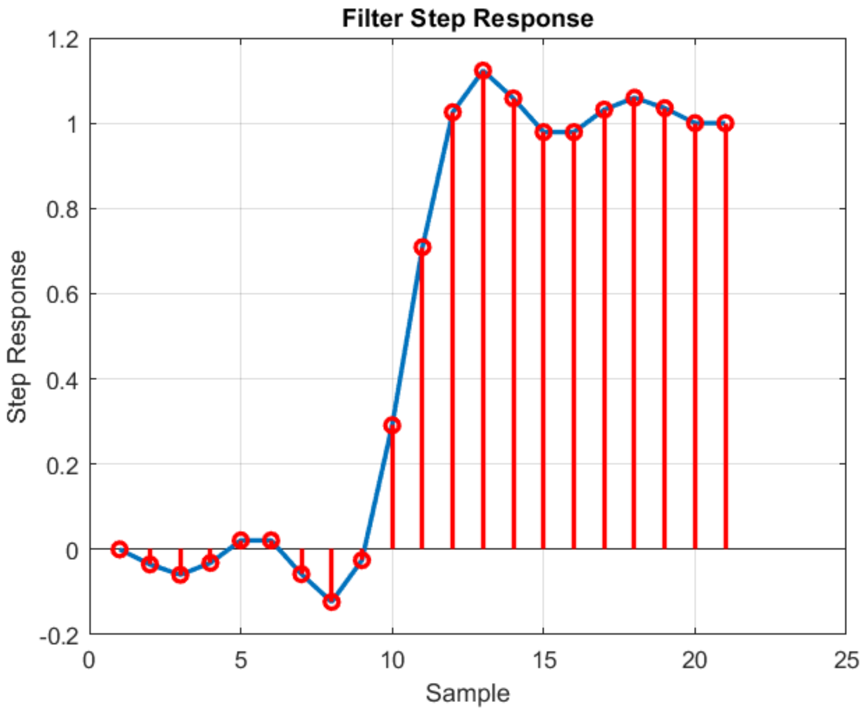


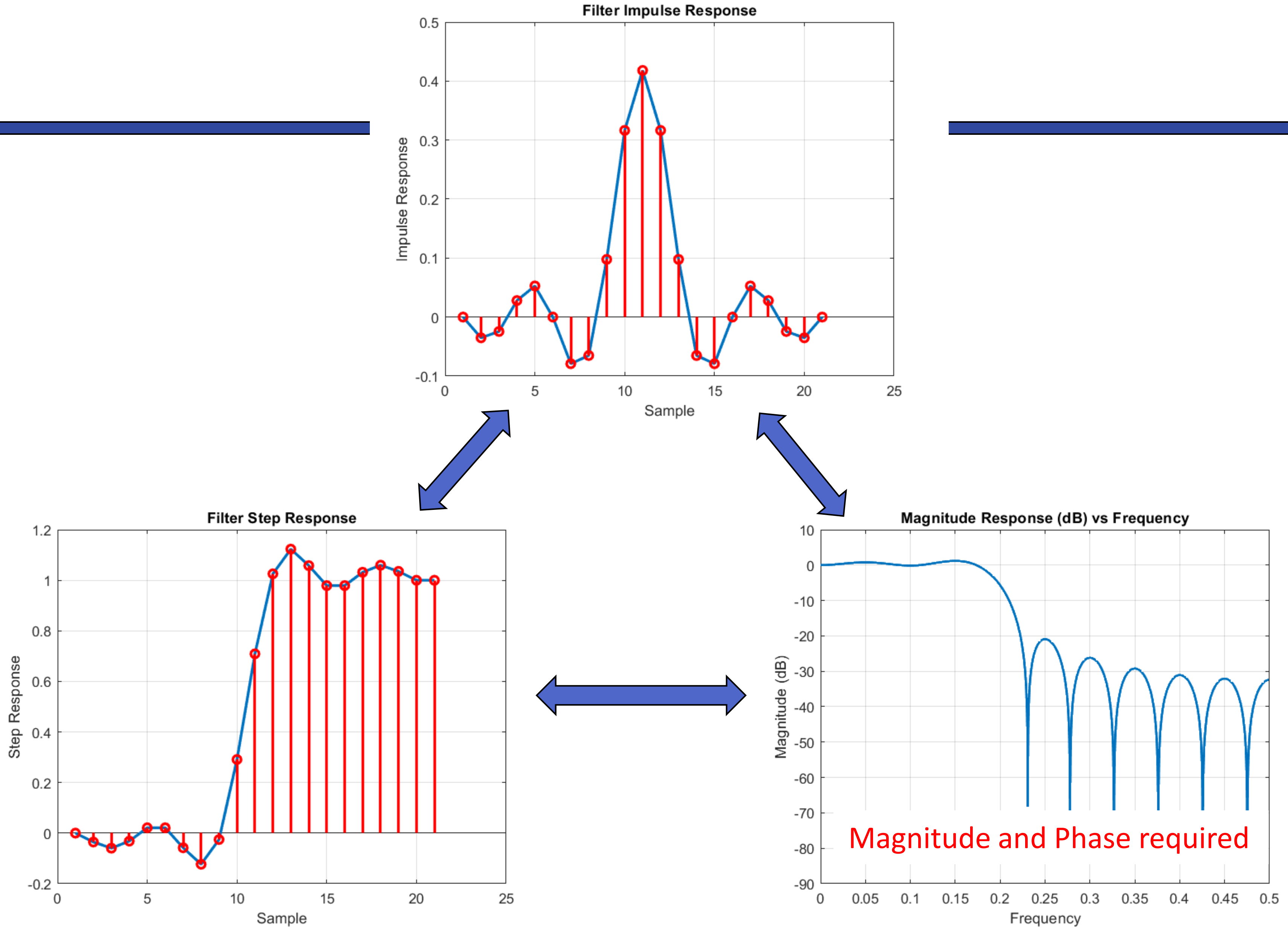


Representation of Filters

- Integrating (discrete integration) allows you to find the step function from the impulse response



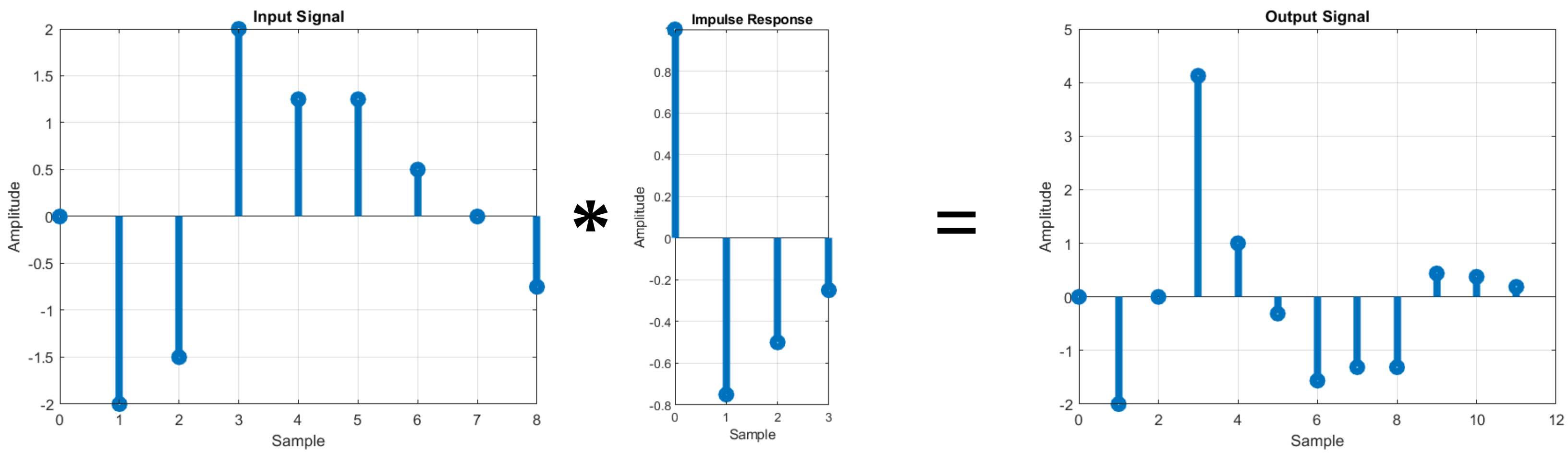




Filter Impulse Response
Impulse Response
Sample
Filter Step Response
Step Response
Sample
Magnitude Response (dB) vs Frequency
Magnitude (dB)
Frequency
Magnitude and Phase required

Implementing Digital Filters
FIR

- The output can be computed using the impulse response and <u>convolution</u>
- Basis for Finite Impulse Response (FIR) filters



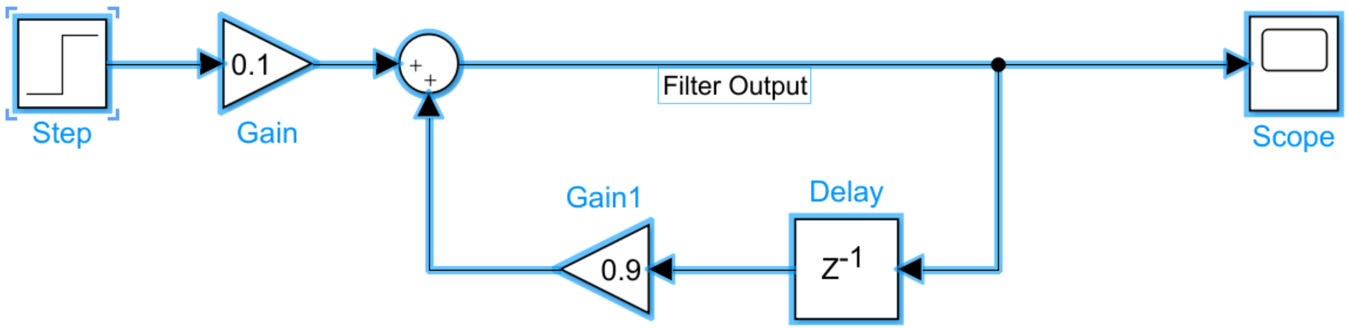
Implementing Digital Filters FIR

- The impulse response is finite in length (Finite Impulse Response)
- The impulse response is called the filter kernel
- Each output is directly synthesized from the input and the impulse response
- Depending on the length of the impulse response computation time can be lengthy

Implementing Digital Filters IIR

- Another approach is to use a recursive equation using old values of the output
- Implement Z-transforms

$$y_k = (1-\alpha)x_k + \alpha y_{k-1} \qquad H(z) = \frac{(1-\alpha)}{1-\alpha z^{-1}}$$

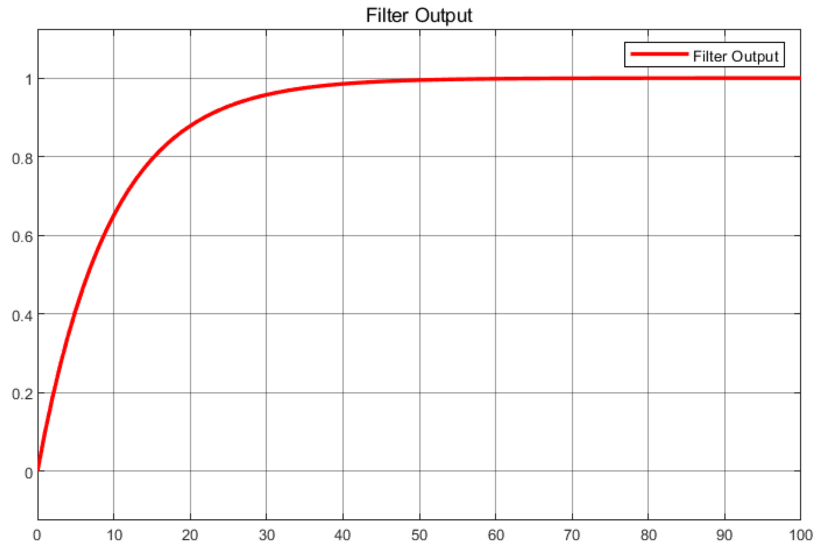


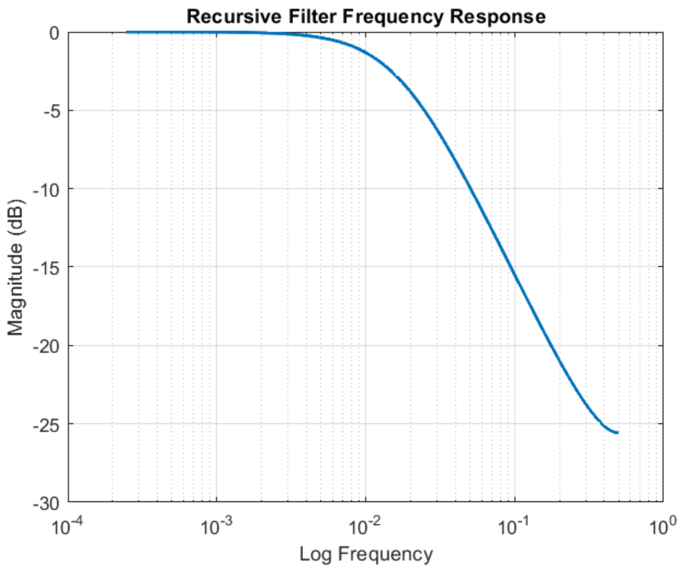
Recursive Filter Example

- Step and Frequency Response

$$y_k = (1 - 0.9)x_k + 0.9y_{k-1}$$

$$H(z) = \frac{(1 - 0.9)}{1 - 0.9z^{-1}}$$





Implementing Digital Filters IIR

- Both the new input samples $x[n]$ and the old output samples $y[n]$ are used in the computation of the next output value

- Infinite Impulse Response (IIR)

- Computation time can be shorter (kernel is shorter)

Filter Characteristics

- We can describe filter characteristics in different domains
- Time domain
- Frequency domain
- Spatial domain (images)

What Describes a Filter in the Time Domain?

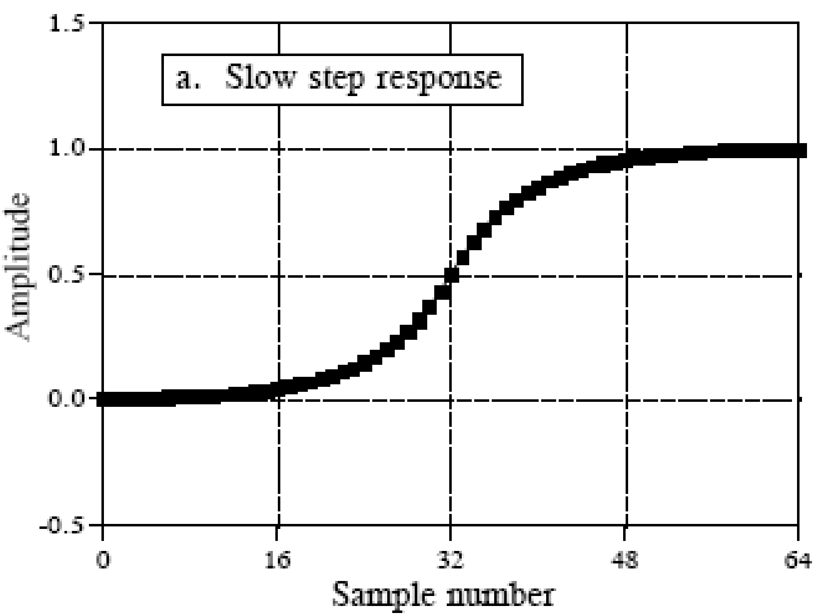
- That depends upon how the information in the signal is encoded and the purpose of the filter.

- For time domain encoded information, the key performance measures are:
  - How well the exact time of an event can be determined
    - Risetime – speed of response
  - How accurately a value of amplitude can be determined.
    - Overshoot – minimize this as it clouds amplitude readings
  - How well rising and falling edges match each other
    - Linear phase – it is required for symmetry of step waveform

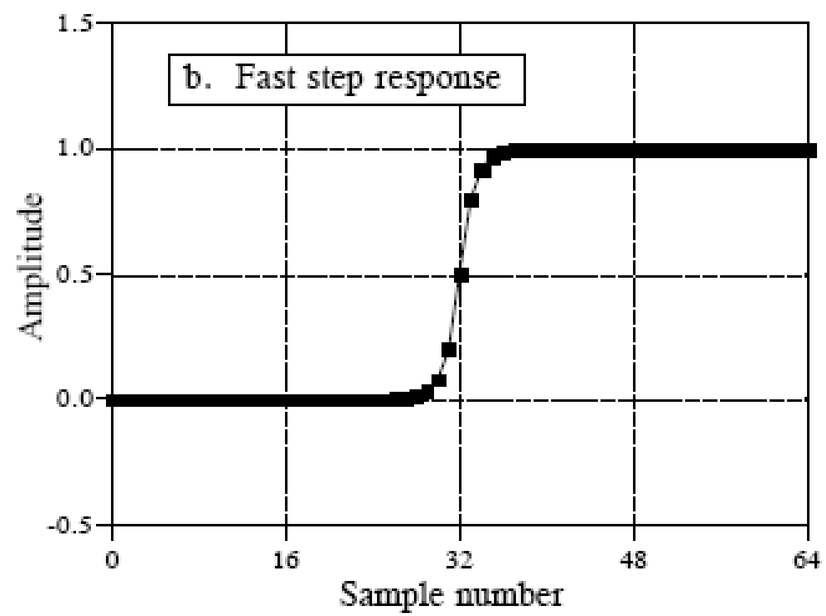
Filter Step Response

- Rise time

Slow



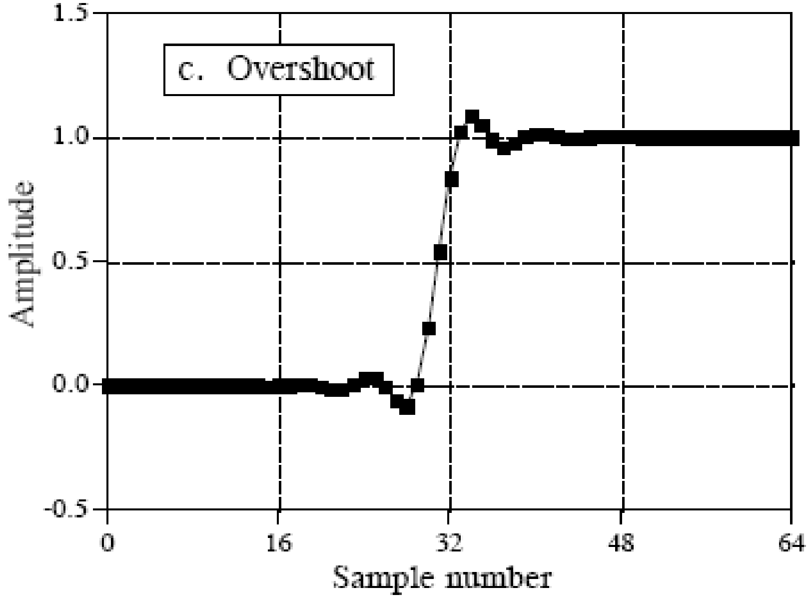
Fast



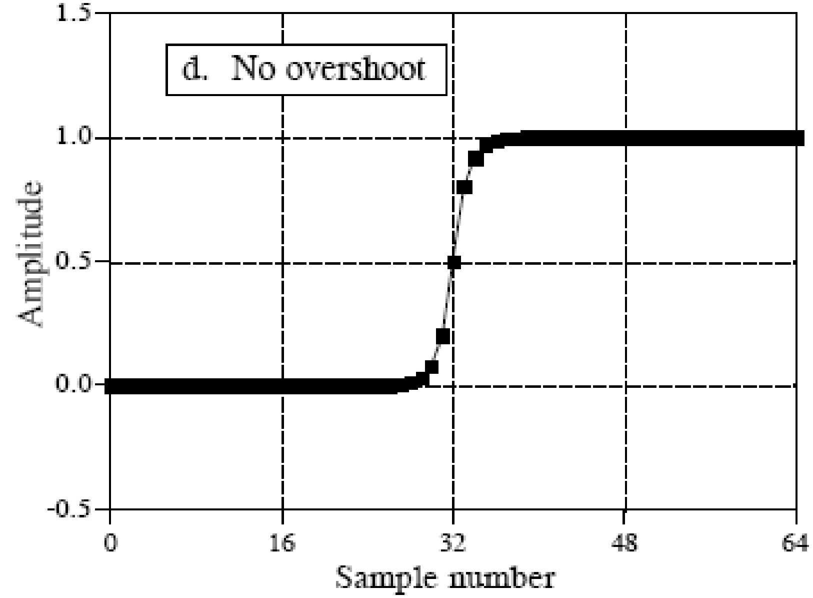
Filter Step Response

- Transient Response

Overshoot



Smooth response

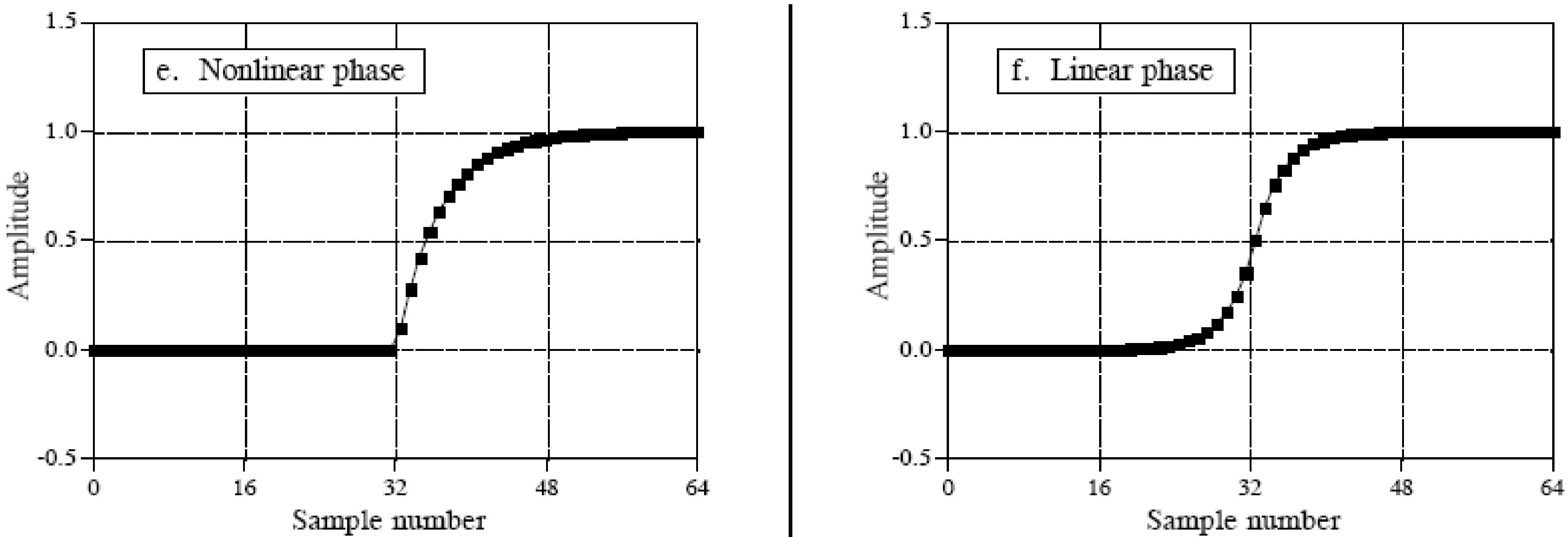


Filter Step Response

- A linear phase filter will have a more “symmetric” step response

Nonlinear Phase

Linear Phase



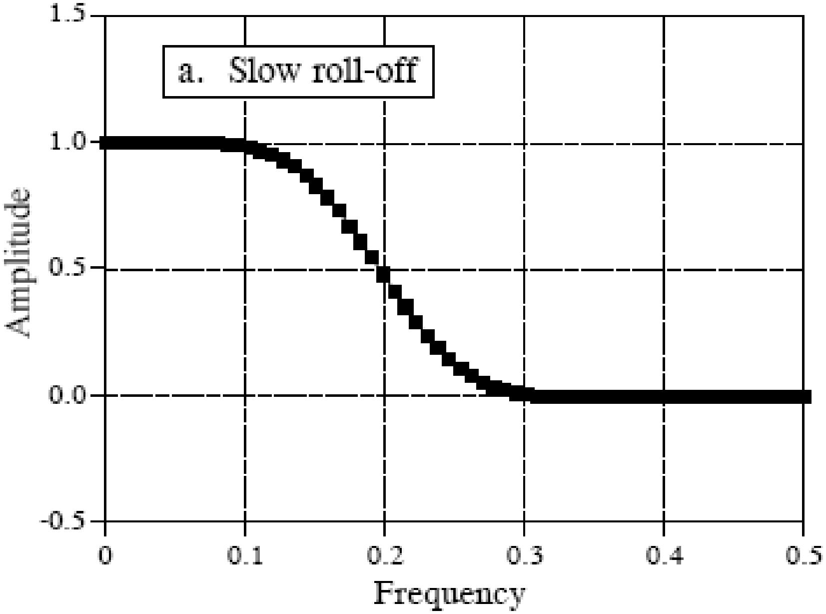
What Describes a Filter in the Frequency Domain?

- For frequency domain encoded information, the key performance measures are:
  - How well the filter separates frequencies
    - Transition bandwidth – fast roll off
  - How accurately amplitude of a signal is passed.
    - Passband ripple - minimize
  - How completely a signal is rejected in the stopband
    - Stopband attenuation–maximize

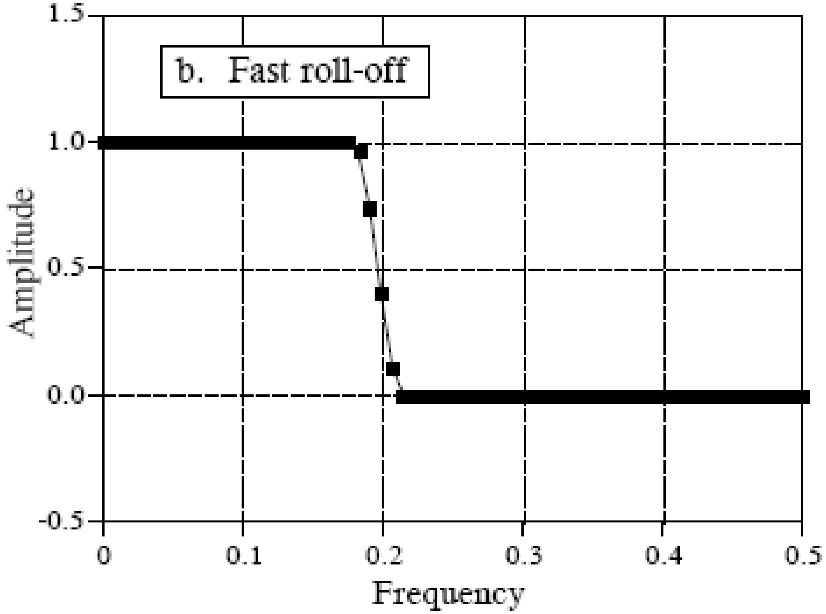
Frequency Domain Parameters

- Frequency Roll off

Slow



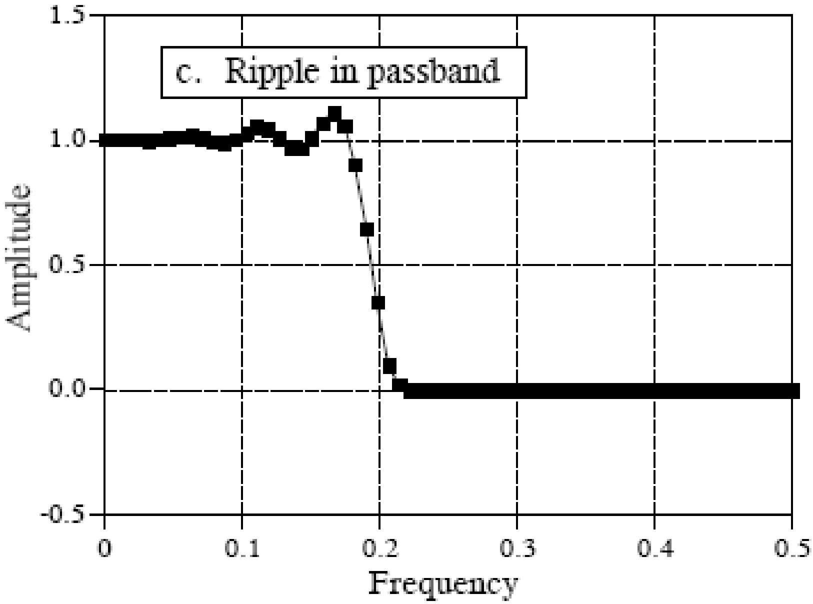
Fast



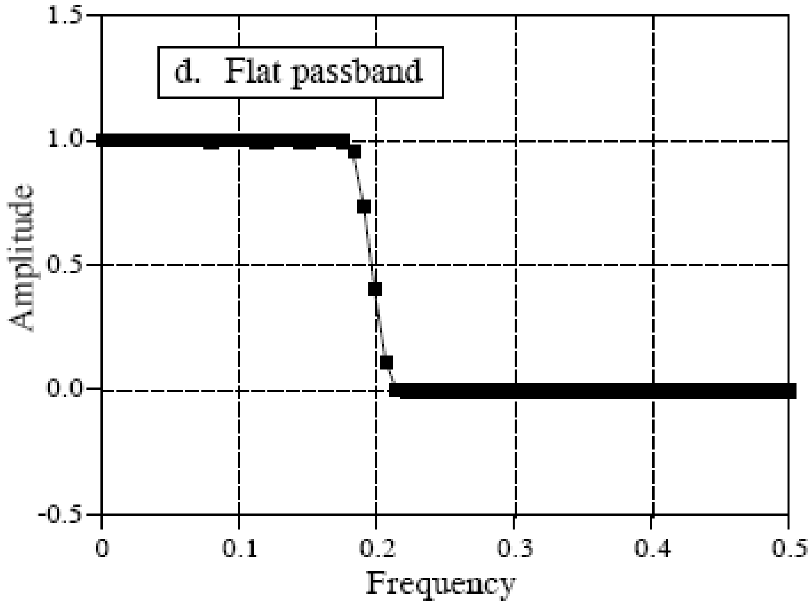
Frequency Domain Parameters

- Pass band Ripple

Ripple in the pass band



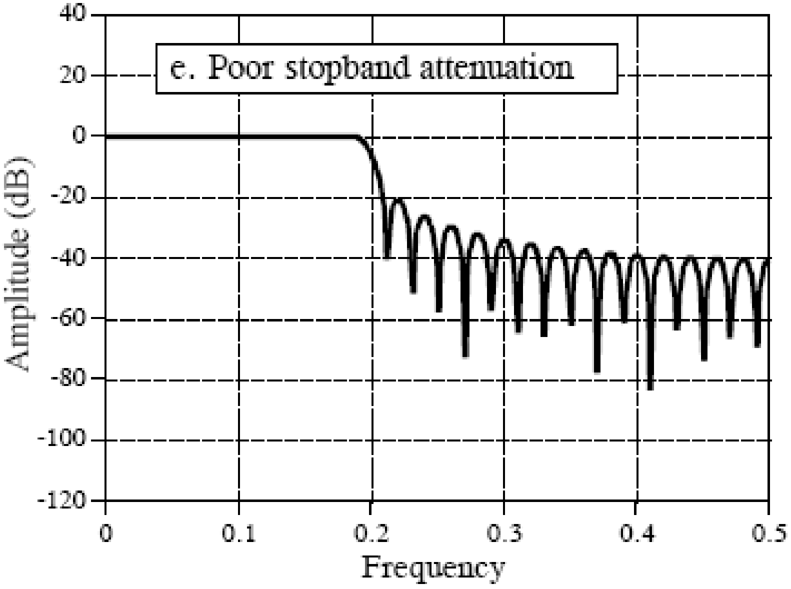
Flat passband response



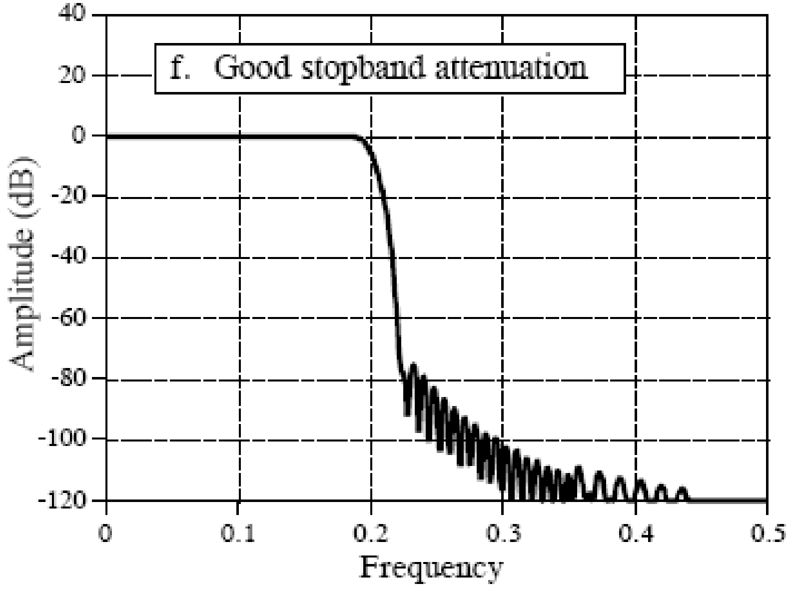
Frequency Domain Parameters

- Stopband attenuation

Less Attenuation



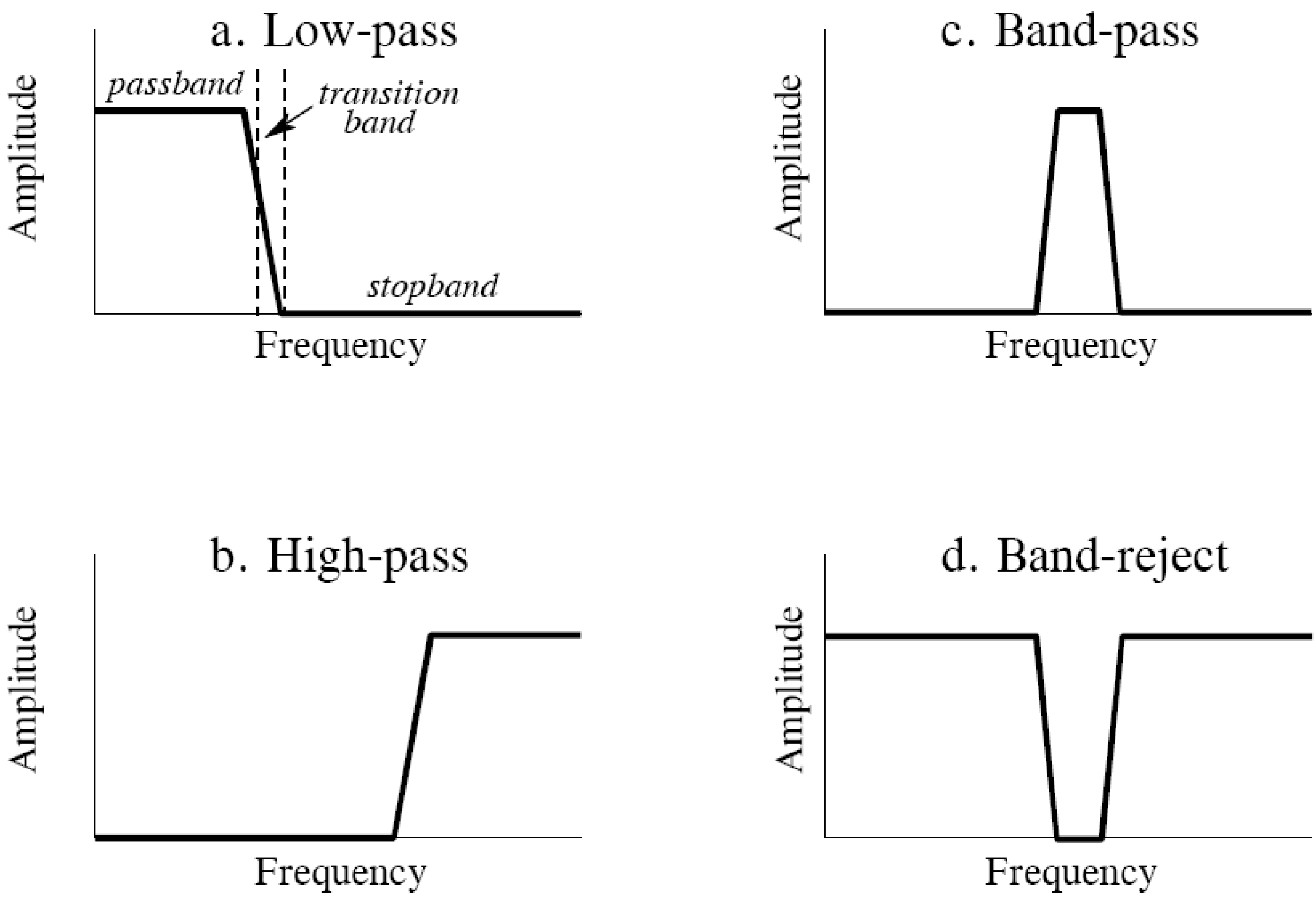
More Attenuation



24

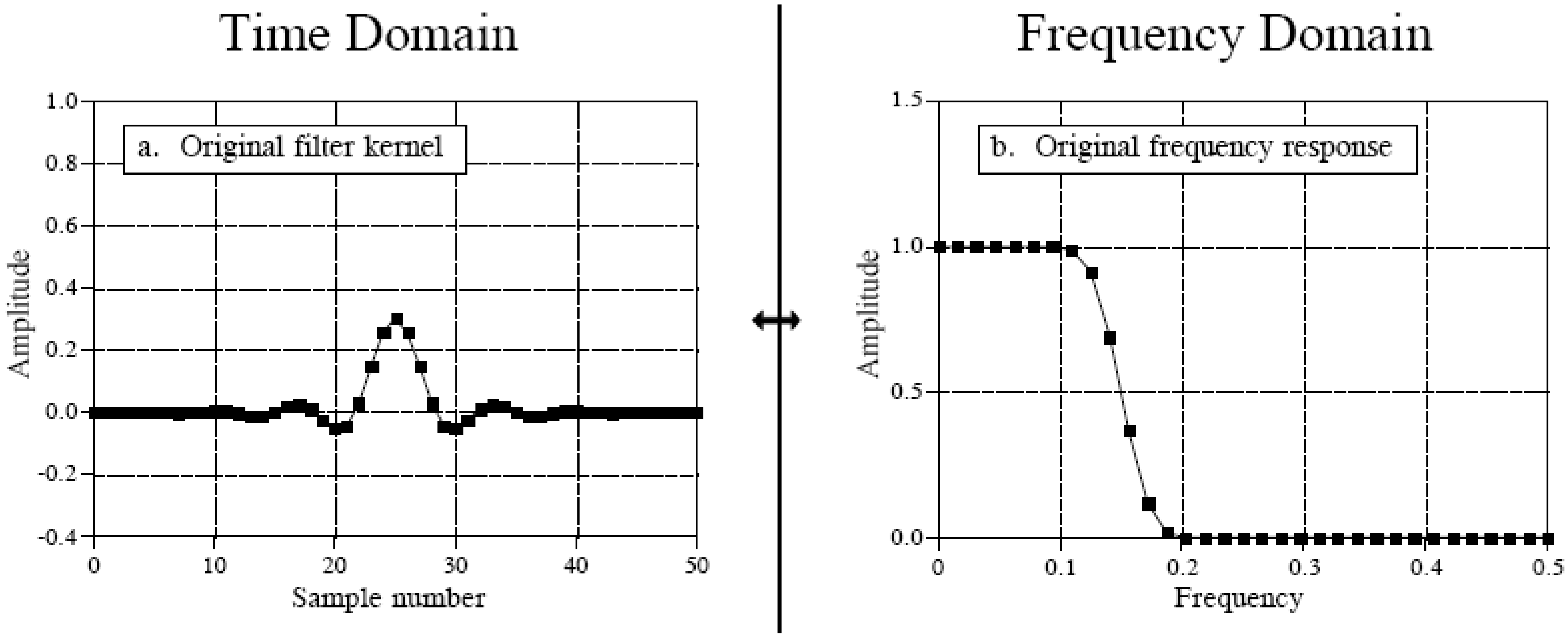
Filter Types

•



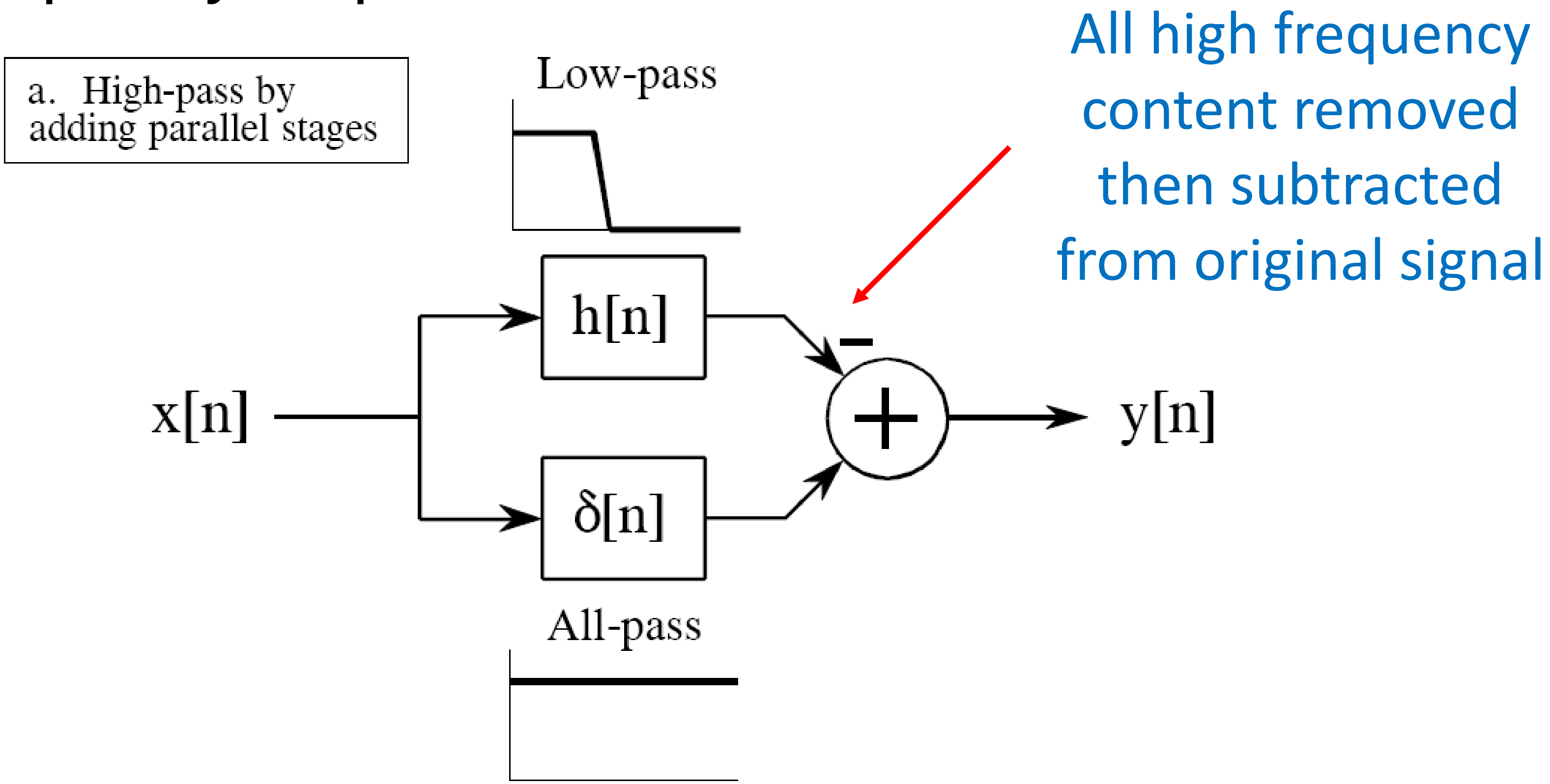
Creating Other Filter Types

- Starting with the impulse response of a lowpass filter, create other filter responses



Spectral Inversion
LPF to HPF or HPF to LPF

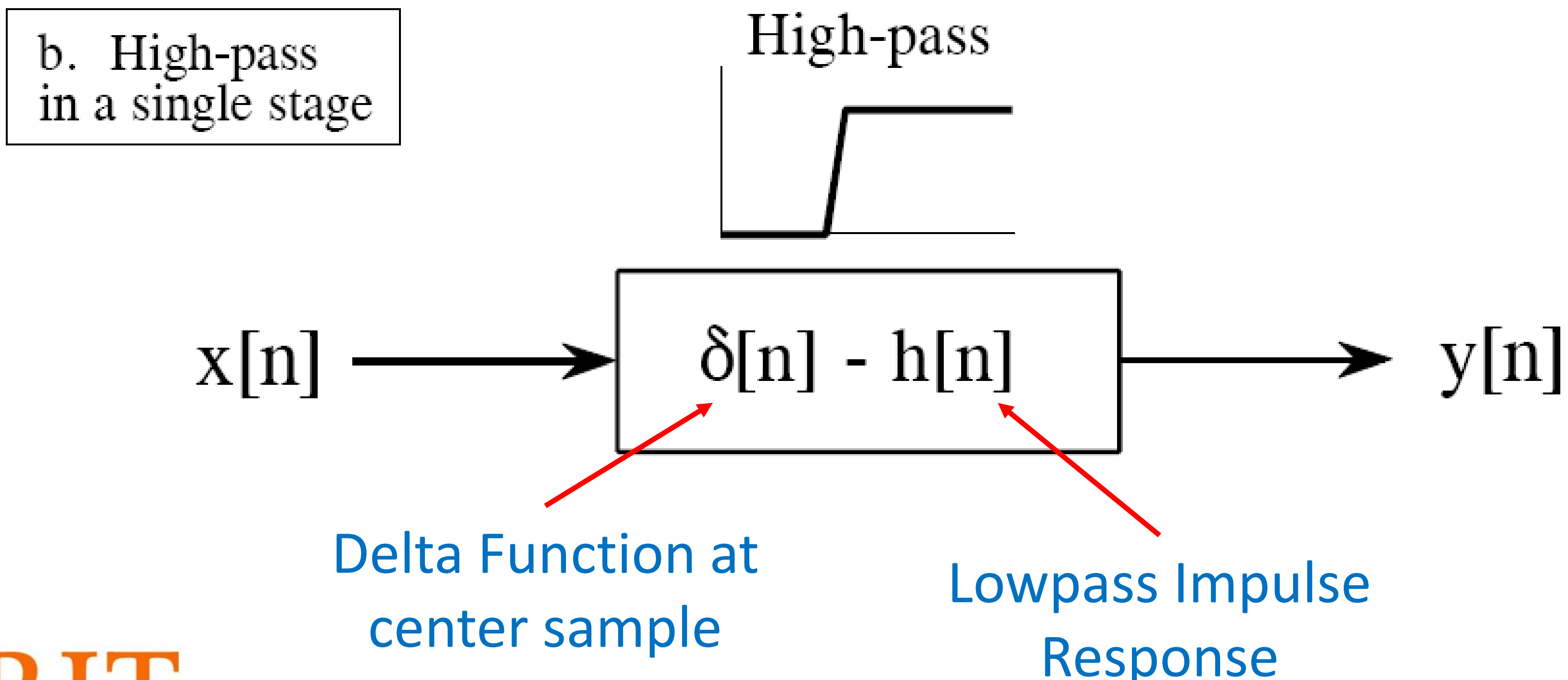
- Conceptually if I subtract the low passed signal from the original signal, I'll retain the high frequency response



27

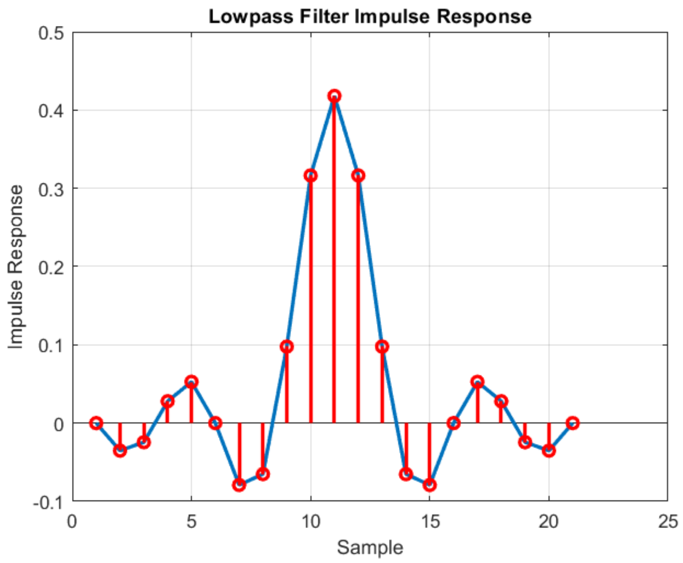
Spectral Inversion

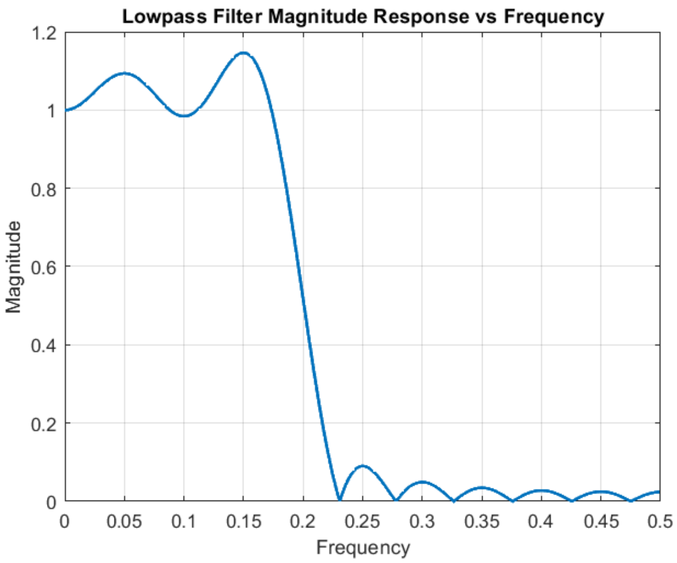
- Mathematically the same as inverting all the samples and subtracting from $\delta[n]$ where n is the middle sample



Spectral Inversion

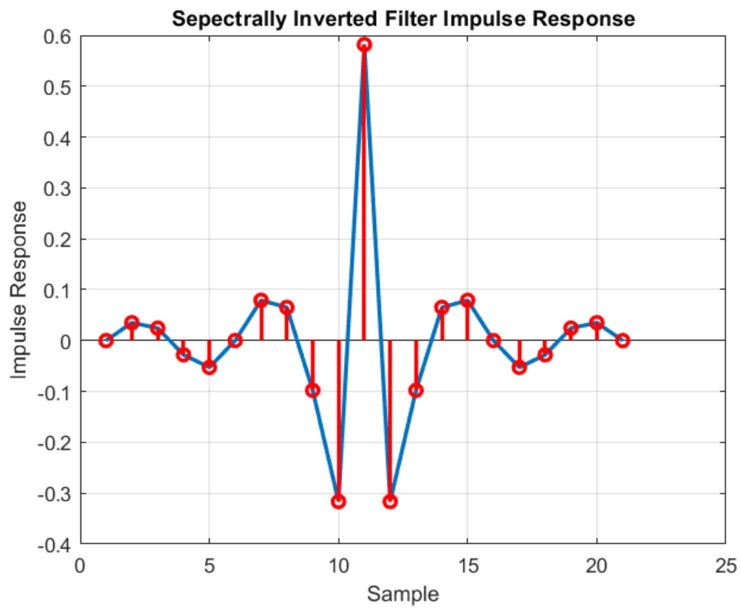
- Mathematically the same as inverting all the samples and subtracting from $\delta[n]$ where n is the middle sample

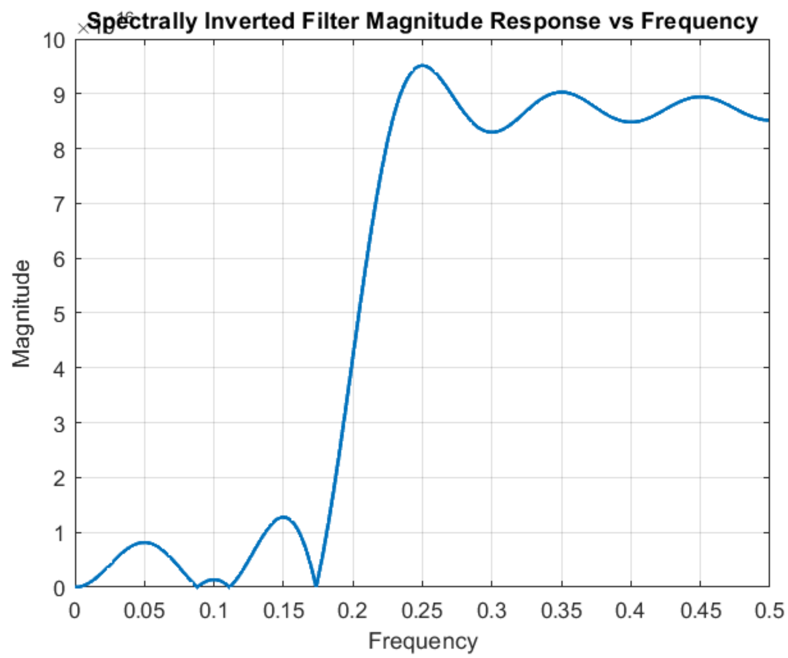




Spectral Inversion

$$\delta \; - h[n]$$

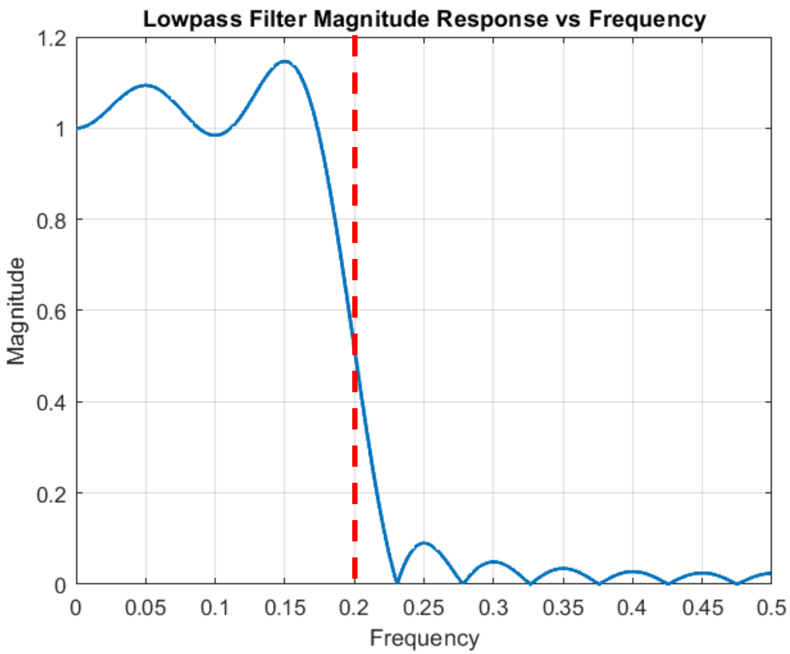


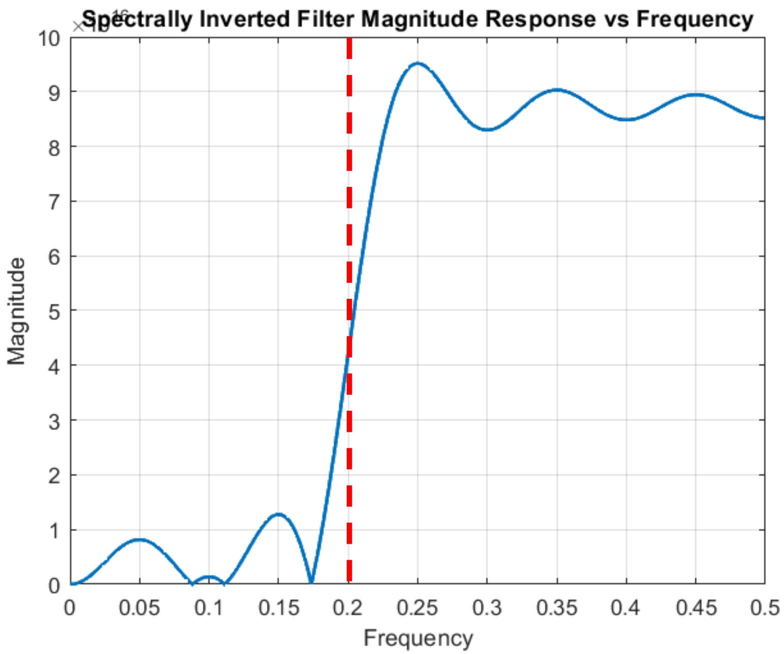


Spectral Inversion

- The responses have the same “corner” frequency

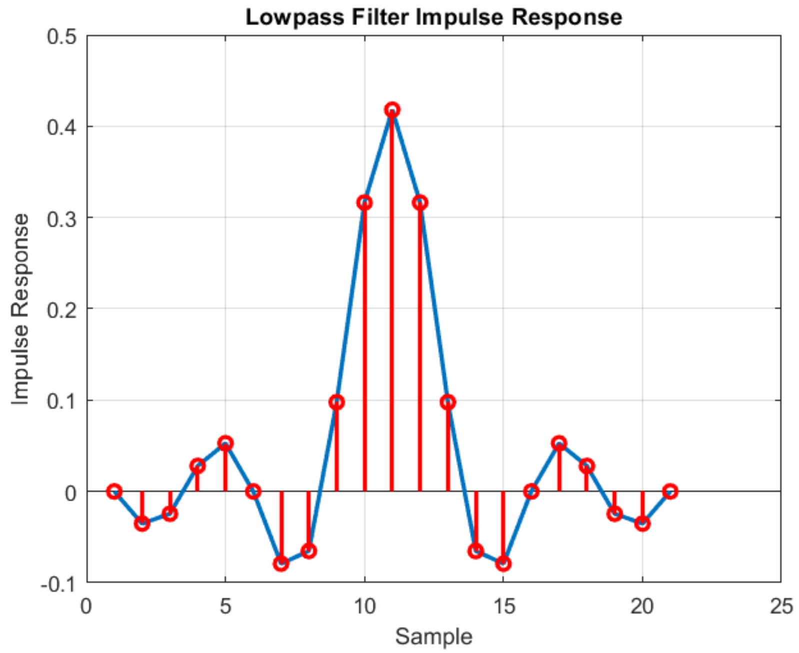
$$f_c = 0.2$$

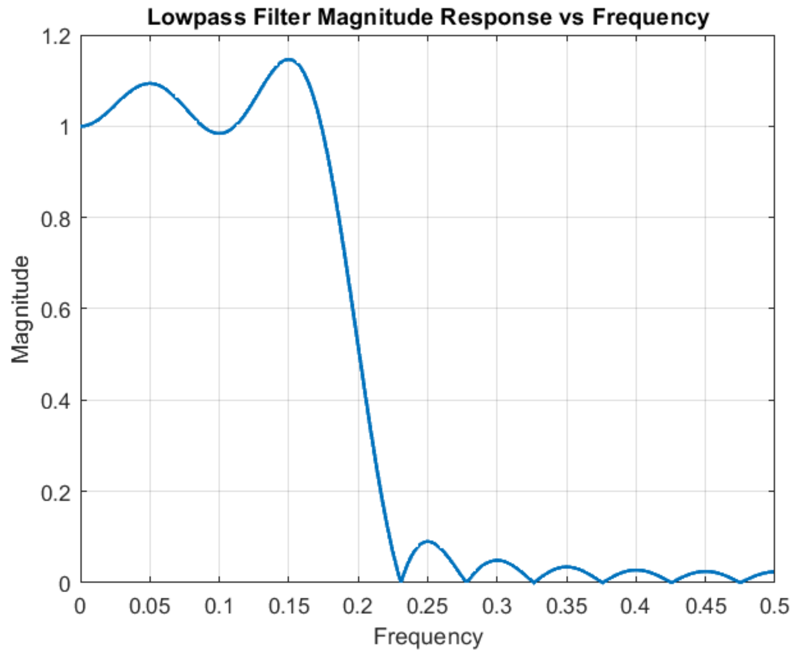




Spectral Reversal

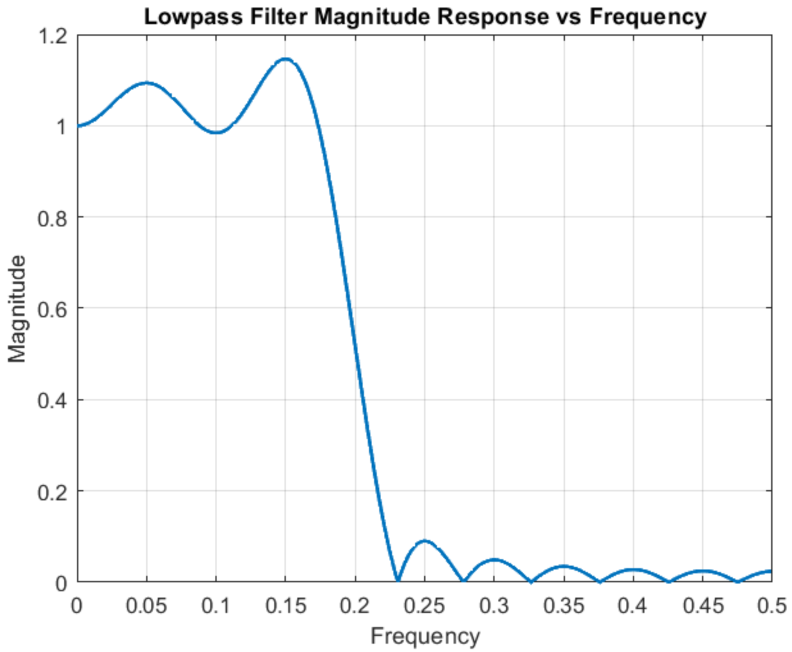
- Again, starting with a lowpass filter response
- The spectrum can be reversed or "flipped"

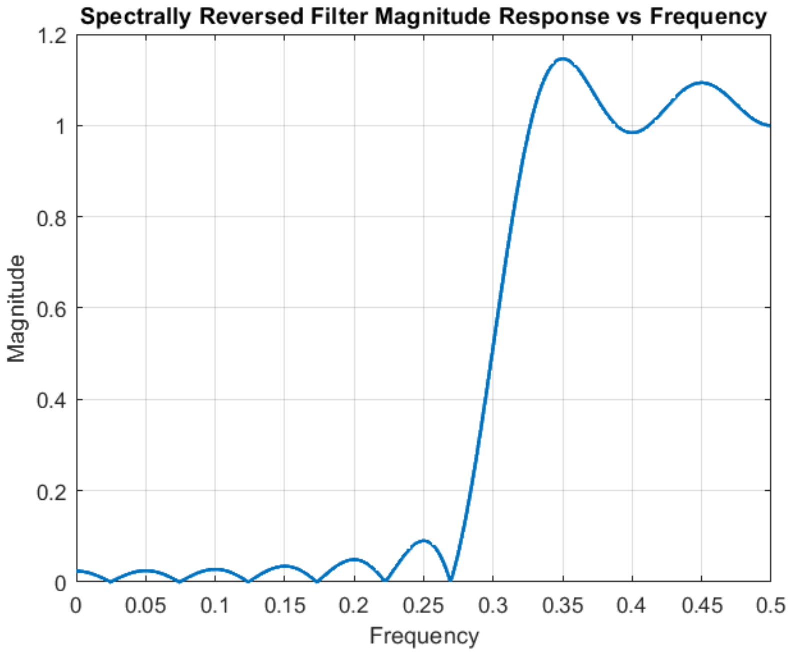




Spectral Reversal

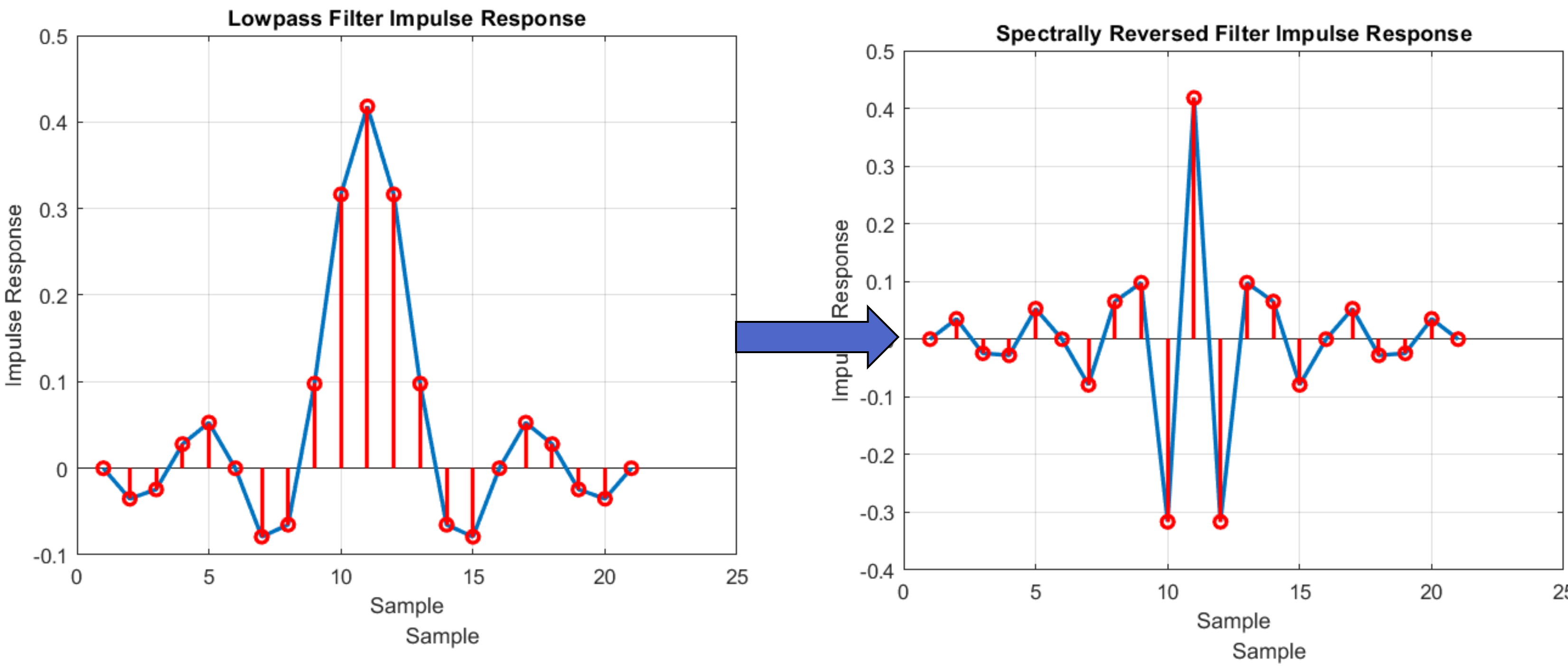
- "Flip" the spectrum left for right.
- Creates a HPF with a different corner frequency from the LPF filter





Spectral Reversal

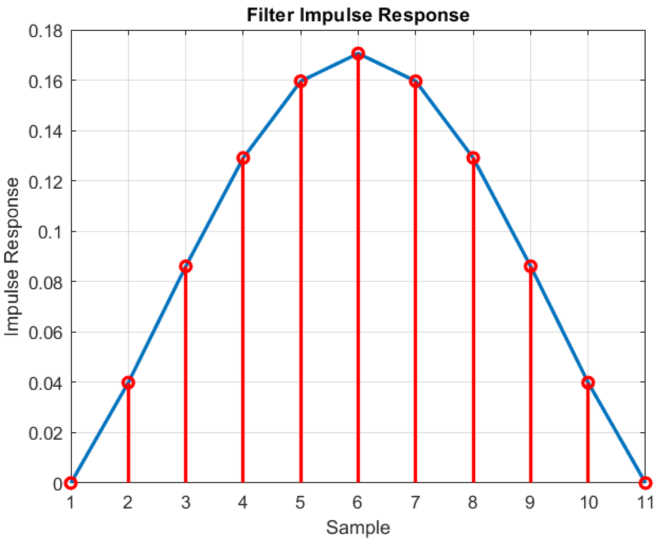
- Mathematically achieved by inverting the sign of every other sample



In Class Problem
Spectral Reversal/Inversion

- A low pass filter has an impulse response of

Sample	h[n]
0	0
1	0.0399
2	0.0861
3	0.1291
4	0.1596
5	0.1706
6	0.1596
7	0.1291
8	0.0861
9	0.0399
10	0



$$h_{inv}[n] = \delta[n] - h[n]$$

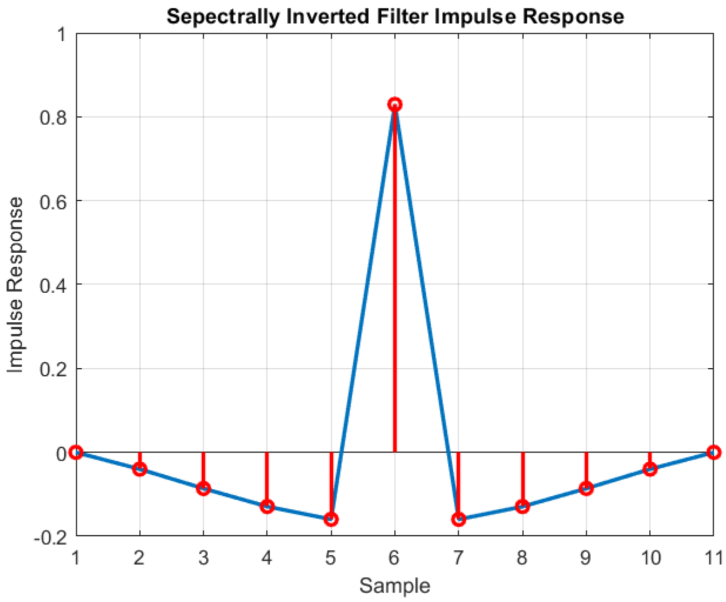
$$h_{rev}[n] = alternate \pm h[n]$$

- Find the impulse response of the spectral reversed and spectrally inverted signal

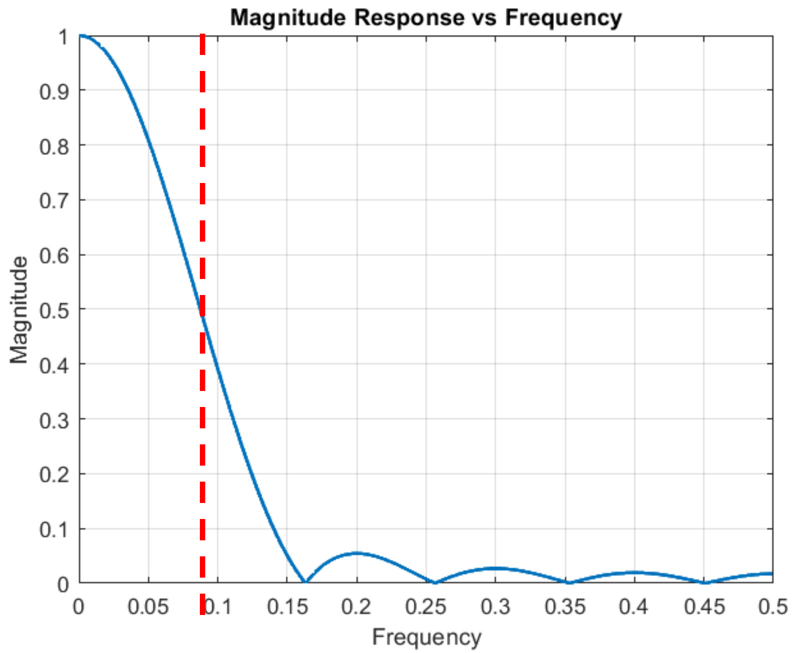
In Class Problem
Spectral Reversal/Inversion

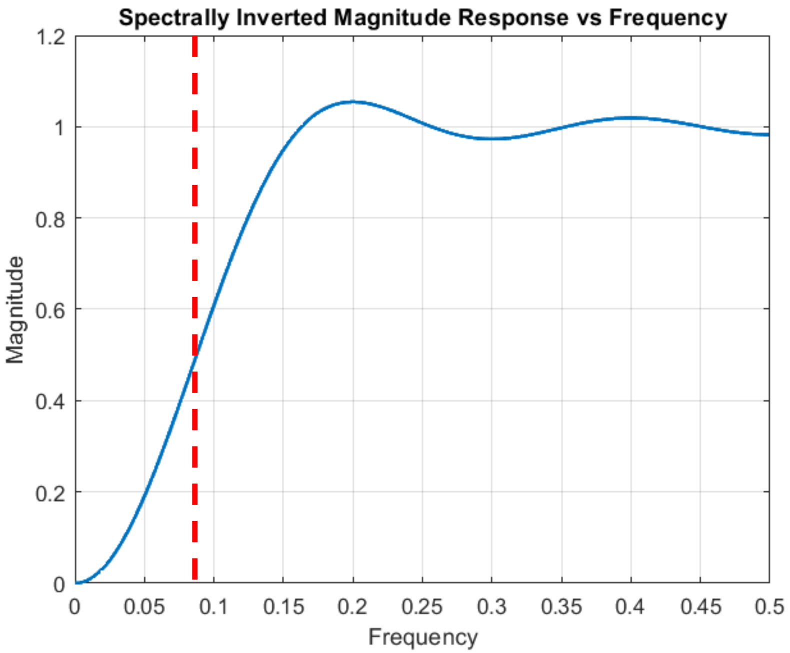
- To find the spectrally inverted signal align the delta function with the middle sample and subtract the low pass impulse response

Sample	h[n]		Delta	Hinv[n]
0	0		0	0
1	0.0399		0	-0.0399
2	0.0861		0	-0.0861
3	0.1291		0	-0.1291
4	0.1596		0	-0.1596
5	0.1706		1	0.8294
6	0.1596		0	-0.1596
7	0.1291		0	-0.1291
8	0.0861		0	-0.0861
9	0.0399		0	-0.0399
10	0		0	0



Spectrally Inverted Frequency Response

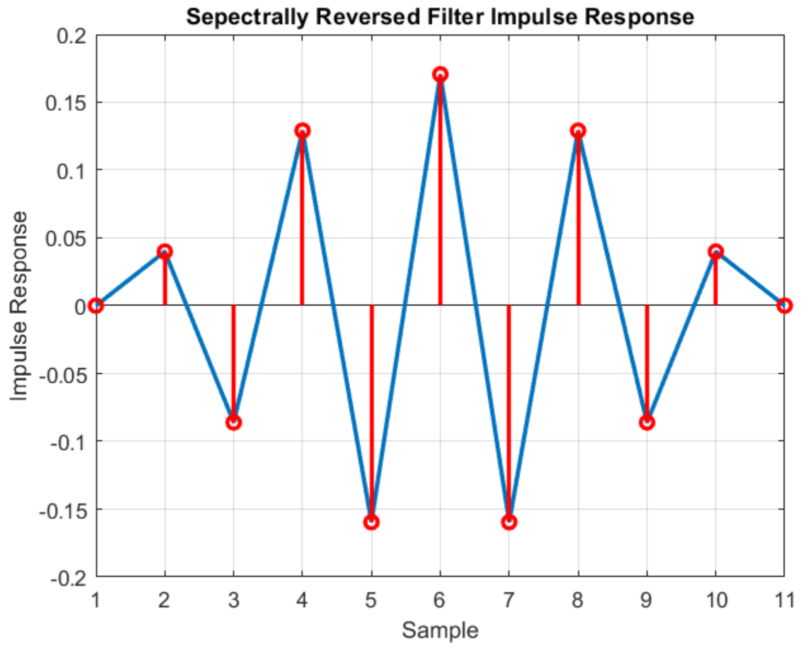




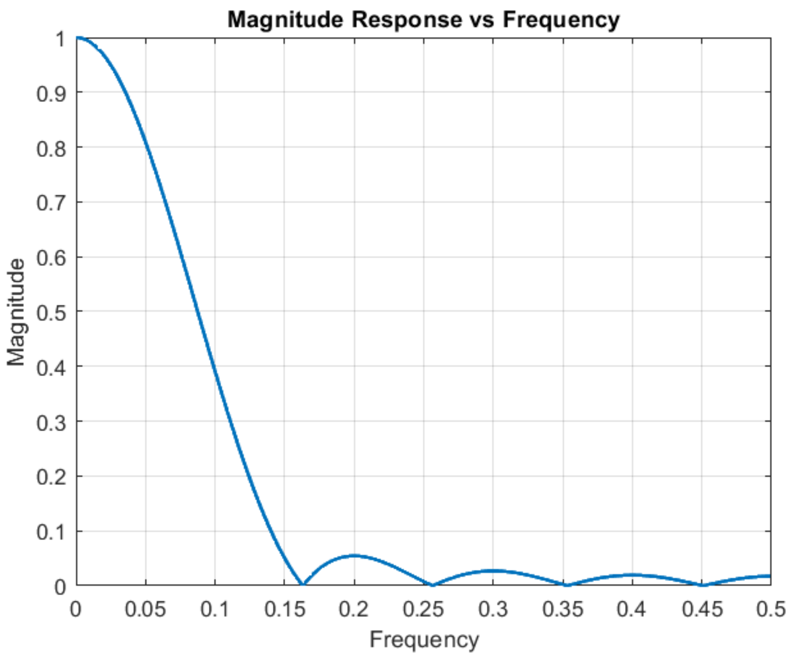
Spectrally Reversed Impulse Response

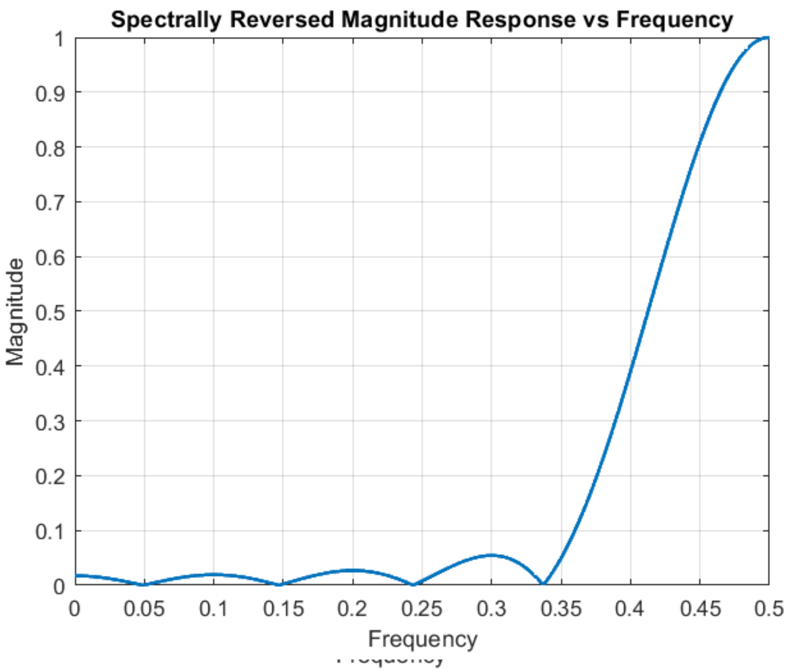
- To find the spectrally reversed signal reverse the sign of every other sample

Sample	h[n]		Sign	Hinv[n]
0	0		-1	0
1	0.0399		1	0.0399
2	0.0861		-1	-0.0861
3	0.1291		1	0.1291
4	0.1596		-1	-0.1596
5	0.1706		1	0.1706
6	0.1596		-1	-0.1596
7	0.1291		1	0.1291
8	0.0861		-1	-0.0861
9	0.0399		1	0.0399
10	0		-1	0



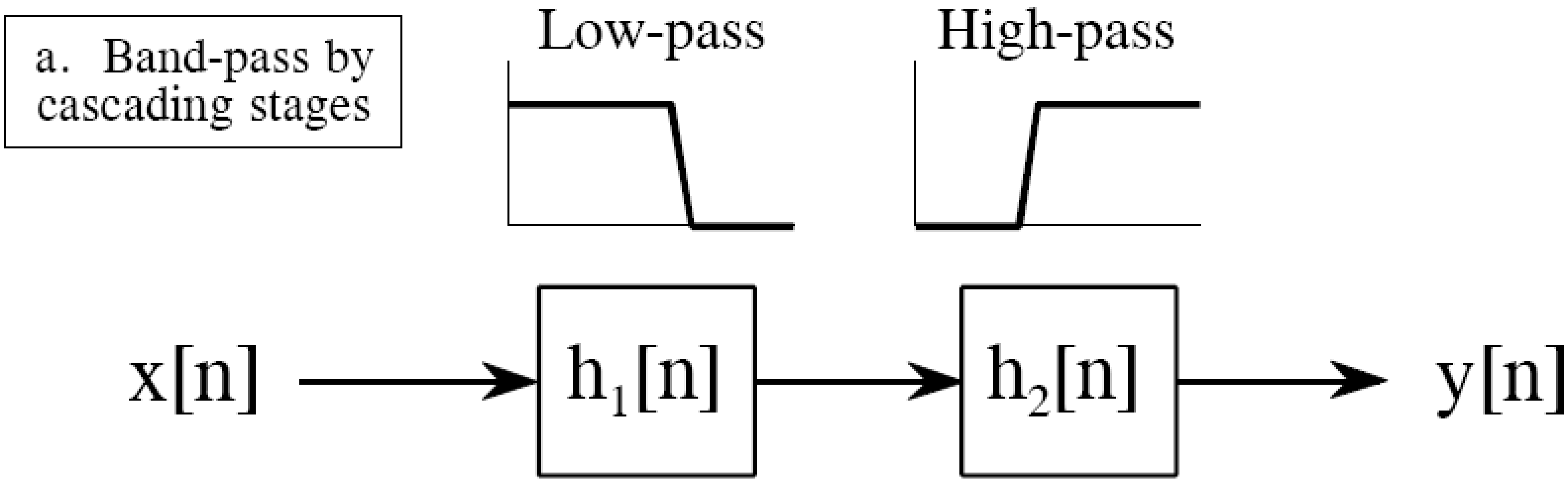
Spectrally Reversed Frequency Response





Creating a Bandpass Filter

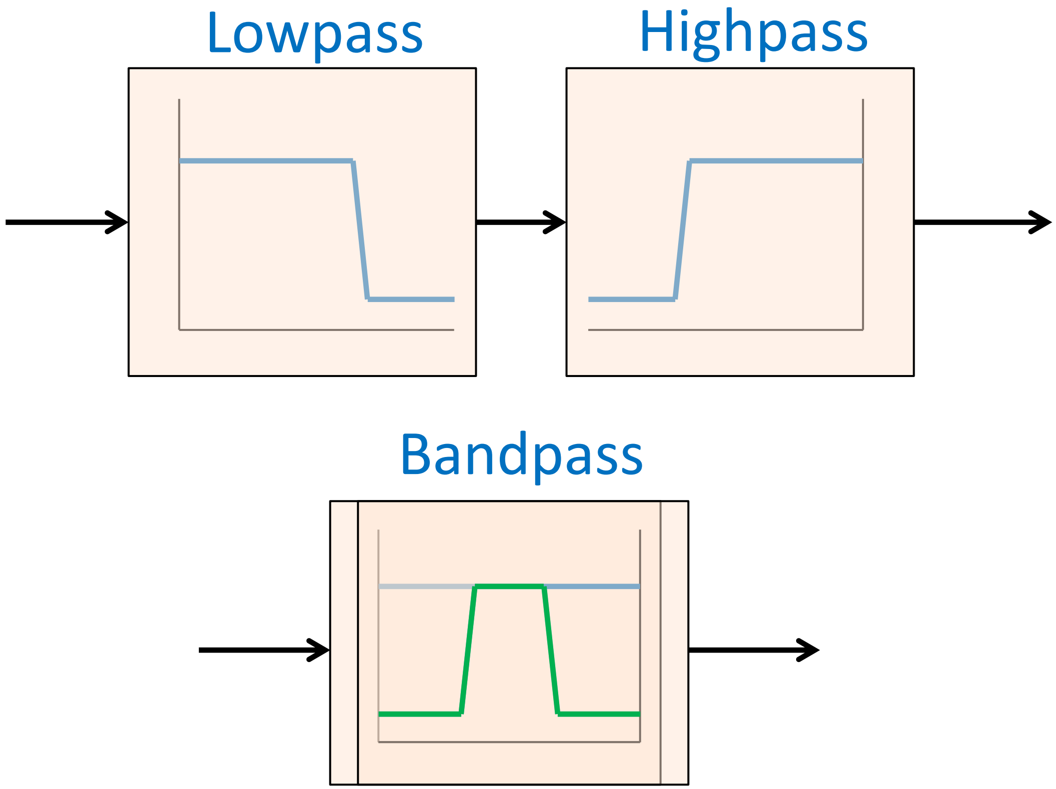
- Use the spectrally reversed or inverted filter cascaded with a low pass to create a bandpass filter



40

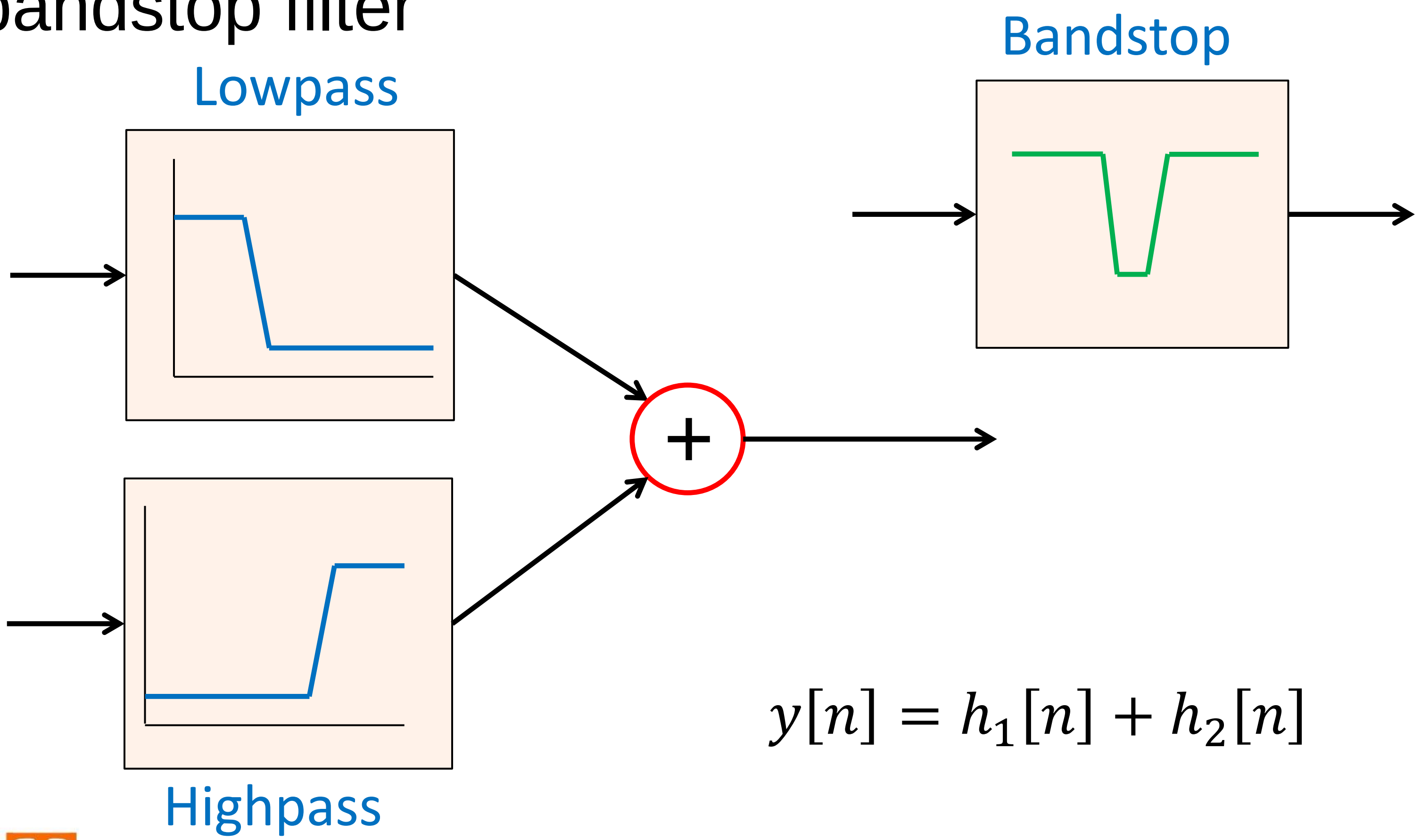
Creating a Bandpass Filter

- Cascading a low pass and a high pass filter creates a band pass filter



Creating a Band Stop (or Band reject)Filter

- Lowpass and highpass in parallel create a bandstop filter



$$y[n] = h_1[n] + h_2[n]$$

Filter Classification

FILTER IMPLEMENTED BY:

FILTER USED FOR:	Convolution *Finite Impulse Response (FIR)*	Recursion *Infinite Impulse Response (IIR)*
Time Domain *(smoothing, DC removal)*	Moving average (Ch. 15)	Single pole (Ch. 19)
Frequency Domain *(separating frequencies)*	Windowed-sinc (Ch. 16)	Chebyshev (Ch. 20)
Custom *(Deconvolution)*	FIR custom (Ch. 17)	Iterative design (Ch. 26)

43