FIR Filter Design

EE4015 Digital Signal Processing

Dr. Lai-Man Po
Department of Electrical Engineering
City University of Hong Kong

EE4015 Face-to-Face Mid-Term Exam

- The Face-to-Face Mid-Term Exam will be held on November 8, 2022 (Tuesday of Week 11).
- The exam time is 2 hours. Students should arrive at the venue at least 5 minutes before the start of the exam.
- The Mid-Term Exam is an open-note exam. Students can use "Scientific Calculator" and "All Handouts", including exercises and assignments.
  - In addition to hard copies of handouts, students can also use smartphones, tablets or iPads to read notes, but the electronic device must be set to airplane mode. During the exam, you are not allowed to communicate with others and search on the Internet. During the exam, investigators will check from time to time whether your electronic device is in airplane mode.
- Students need to use their own answer sheets (such as A4 paper) to answer the questions.
- The mid-term exam will cover up to week 8 (Lecture L07B will not be covered in Mid-Term Exam).

Content

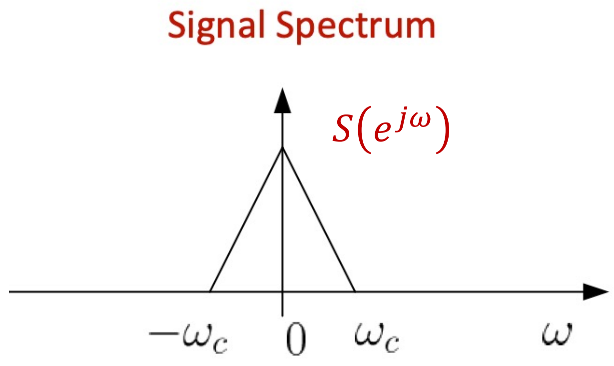
- Digital Filter Design Basic
- Why FIR Filter
- Linear Phase FIR Filter
- FIR Filter Design
    - Window Method
    - Frequency Sampling – Arbitrary Frequency Response
    - Optimal Equiripple Method
        - Parks-McClellan Program with Remez Exchange Algorithm

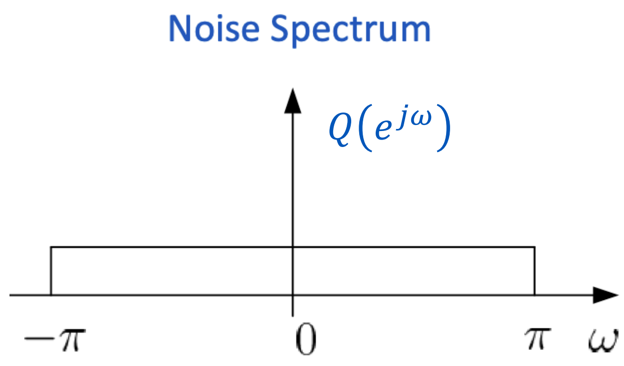
Step in Digital Filter Design

1. **Specification Determination**
2. **Filter Coefficient Calculation**
3. **Implementation**

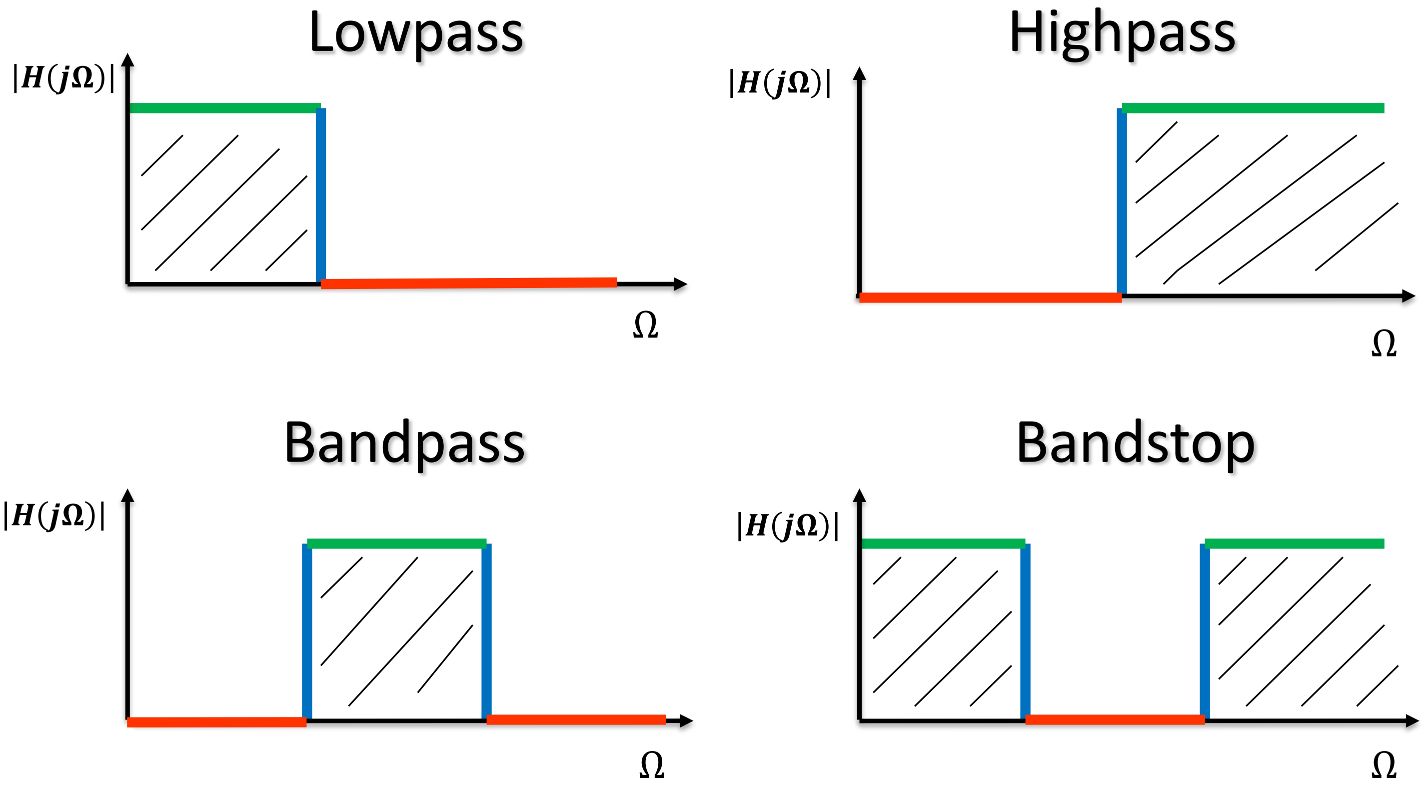
1. Specification Determination

- The first step is to obtain the filter specifications or requirements which are determined by the applications.
  - Suppose we have observations $r[n] = s[n] + q[n]$ where $s[n]$ is the signal of interest and $q[n]$ is the additive noise. The $s[n]$ only has low-frequency components such that $S(e^{j\omega}) = 0$ for $\omega > \omega_c$ while $q[n]$ is of wideband such that $Q(e^{j\omega}) \neq 0$ for the whole frequency range.
  - If our task is to find $s[n]$ from $r[n]$, we can use a lowpass filter with cutoff frequency of ω_c to obtain a noise-reduced version of $s[n]$.





Four Types of Ideal Filters (Continuous-Time)

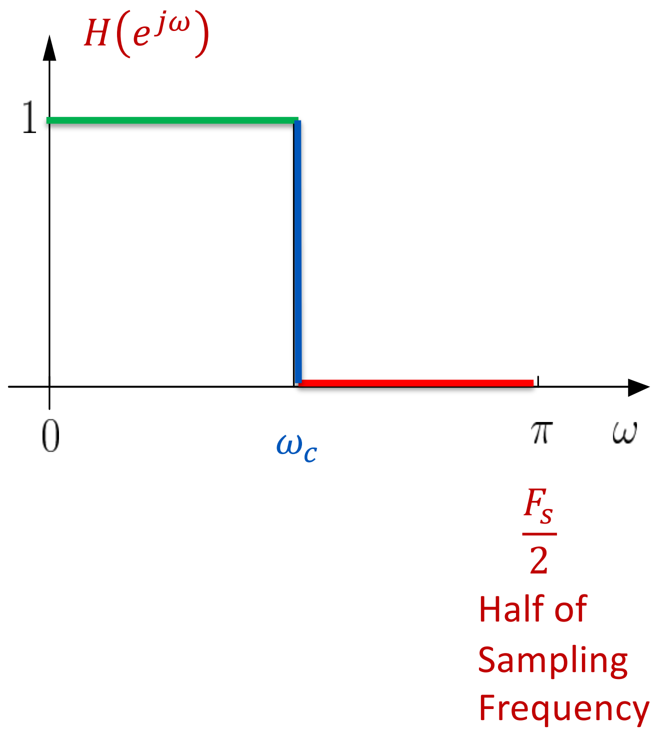


Ideal Discrete-Time Lowpass Filter Specification

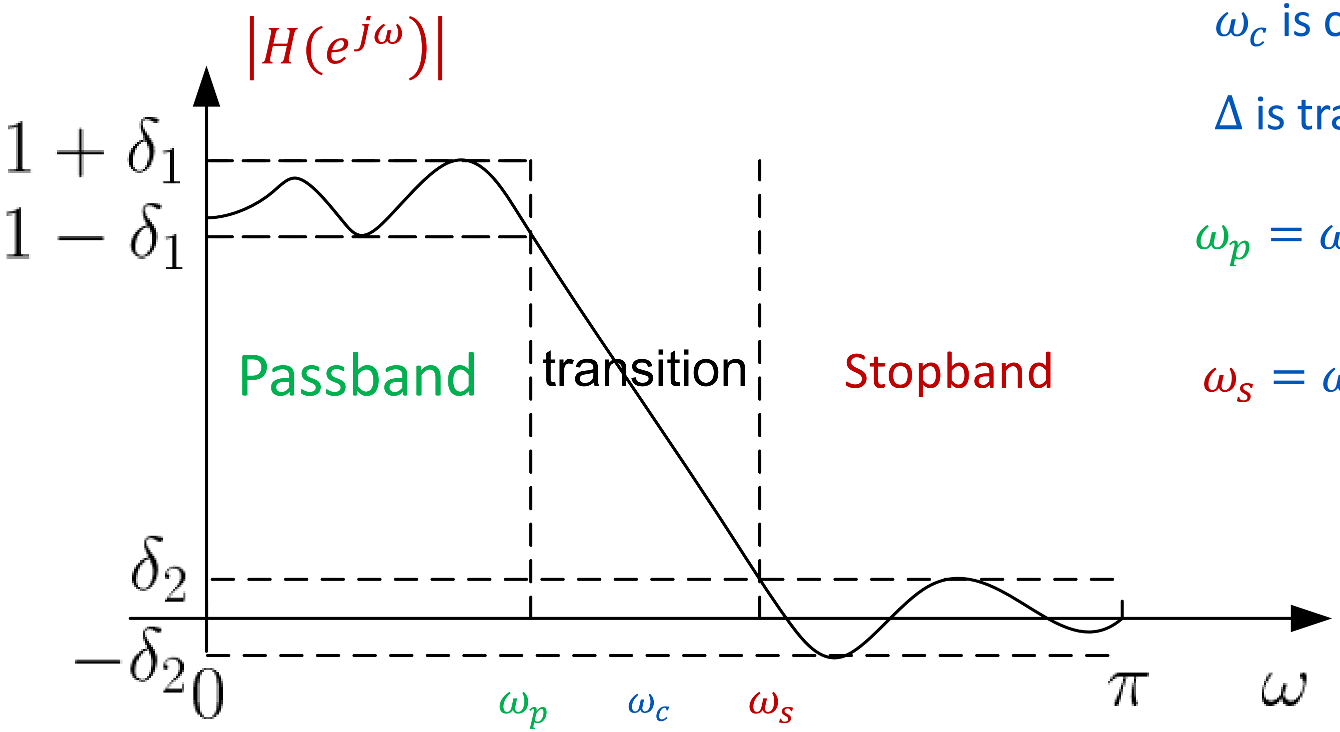
- The filter specification can be described by discrete-time Fourier Transform (DTFT) $H(e^{j\omega})$ as

$$H(e^{j\omega}) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & -\pi < \omega < -\omega_c,\ \omega_c < \omega < \pi \end{cases}$$

- Which specifies both the magnitude and phase.
  - Unity gain for the whole range of $\omega \in (0, \omega_c)$
  - Complete suppression for $\omega \in (\omega_c, \pi)$
  - Step change in frequency response at $\omega = \omega_c$



Practical Magnitude Response Specification



ω_c is cuffoff frequency

Δ is transition bandwidth $\omega_s - \omega_p$

$$\omega_p = \omega_c - \frac{\Delta\omega}{2}$$

δ_1 is the passband ripple

$$\omega_s = \omega_c + \frac{\Delta\omega}{2}$$

δ_2 is the stopband ripple

Passband, Stopband and Transition band

- Passband corresponds to $\omega \in (0, \omega_p)$ where ω_p is the passband frequency and δ_1 is the passband ripple or tolerance which is the maximum allowable deviation from unity in this band
- Stopband corresponds to $\omega \in (\omega_s, \pi)$ where ω_s is the stopband frequency and δ_2 is the stopband ripple or tolerance which is the maximum allowable deviation from zero in this band
- Transition band corresponds to $\omega \in (\omega_p, \omega_s)$ where there are no restrictions on $H(e^{j\omega})$ in this band

2. Filter Coefficients Calculation Method

- We then use digital signal processing techniques to obtain a filter description in terms of transfer function $H(z)$ or impulse response $h[n]$ that fulfills the given specifications

- **FIR Filter Design**
  - Windowing Method
  - Frequency Sampling Method
  - The Optimal Parks-McCellan Method

$$H(z) = h_0 + h_1 z^{-1} + \cdots + h_{M-1} z^{M-1} + h_M z^M$$

- **IIR Filter Design**
  - Mapping from analog filter
  - Impulse Invariant Method
  - Bilinear Transform Method

$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_{M-1} z^{M-1} + b_M z^M}{a_0 + a_1 z^{-1} + \cdots + a_{N-1} z^{N-1} + a_N z^N}$$

Choose Between FIR and IIR Filters

FIR Filters	**IIR Filters**
• Less Efficient	• More Efficient
• No Analog Equivalent	• Analog Equivalent
• Always stable	• May be Unstable
• Linear Phase Response	• Non-Linear Phase Response
• Less Ringing on Glitches	• More Ringing on Glitches
• CAD Design Packages Available	• CAD Design Packages Available
• Decimation Increases Efficiency	• No Efficiency Gained by Decimation

FIR and IIR Filter Selection

- **We choose the methods that best suits the applications.**
  - In most cases, if the FIR properties (e.g. linear phase) are vital then a good candidate is the optimization method
  - If IIR properties (e.g. low complexity) are desirable, then the bilinear method will in most cases suffice.

3. Implementation

- When $H(z)$ or $h[n]$ are known, the filter can then be realized in hardware or software according to a given structure

- **FIR Filter Structures**
  - Direct Form
  - Linear Phase Direct Form
  - Cascade Structure

- **IIR Filter Design**
  - Direct Form (Direct Form I)
  - Canonic Form (Director Form II)
  - Cascade Structure
  - Parallel Structure

Linear-Phase FIR Filter

Stability of FIR Filter

- Transfer function of a causal FIR filter $H(z)$ with length N is:

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{N-1} b_k z^{-k} = \sum_{k=0}^{N-1} h[n] z^{-k}$$

$$y[n] = \underset{h[0]}{b_0} x[n] + \underset{h[1]}{b_1} x[n-1] + \cdots + \underset{h[N]}{b_N} x[n-N]$$

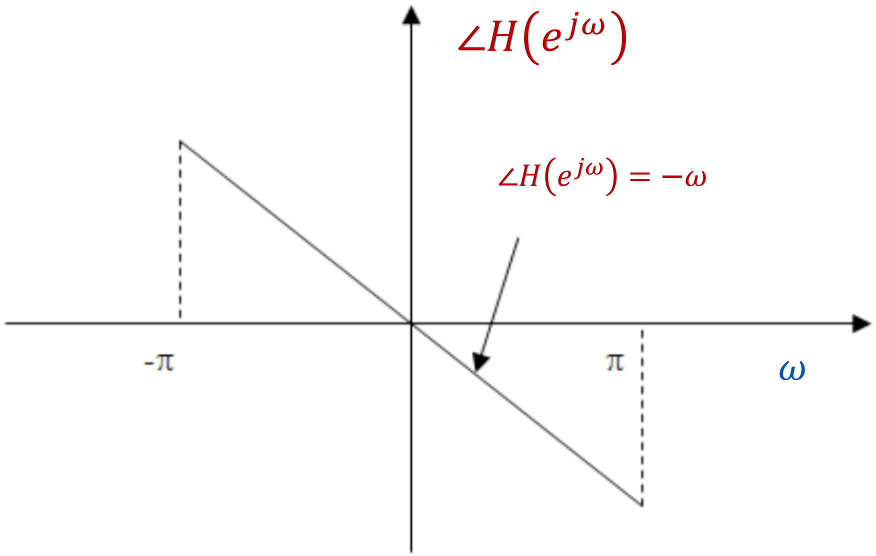
where $h[n]$ is the finite-duration impulse response

- A property of the FIR filter is that it will always be stable as the transfer function $H(\mathrm{z})$ has only zeros with no poles.
  - $\sum_{n=0}^{N-1} |h[n]| < \infty$ for finite N

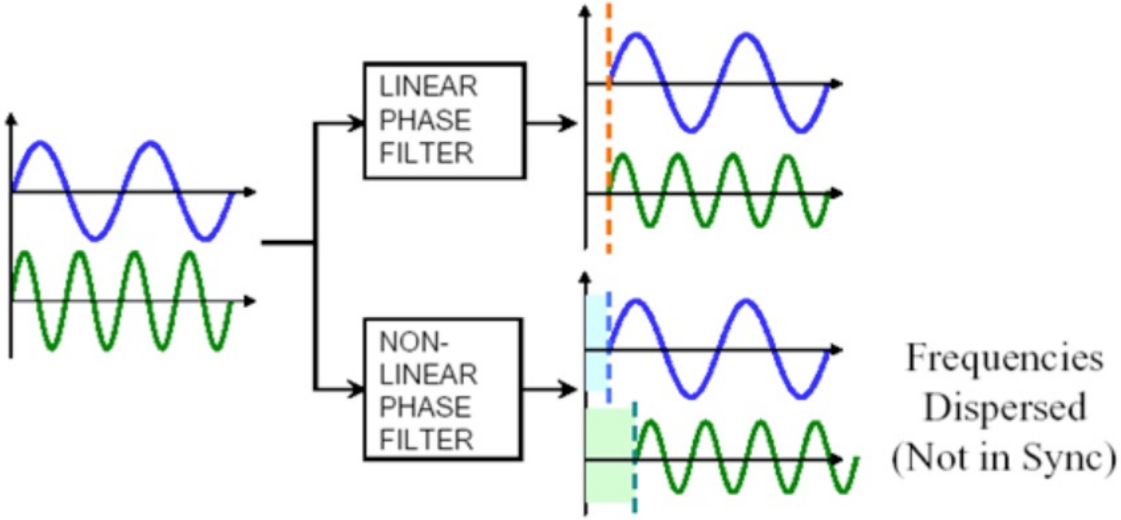
Linear-Phase FIR Filter

- Phase response of FIR system can be exactly linear which results in computation reduction and zero phase distortion.
- A linear-phase filter gives same time delay to all frequency components of the input signal.
- A filter with a nonlinear phase characteristic will cause a phase distortion in the signal that passes through it.
  - This is because the frequency components in the signal will each be delayed by an amount not proportional to frequency, thereby altering their harmonic relationship.
  - Such a distortion is undesirable in many applications, for example music, video etc.

Phase Response of a Linear-Phase Filter

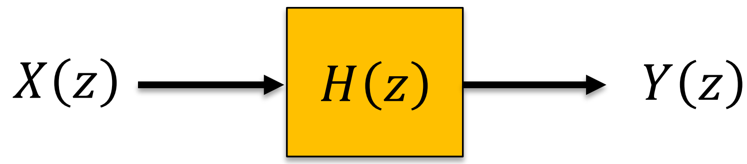


Phase Response of a Linear-Phase Filter



A diagram comparing the performance of a linear phase filter and a non-linear phase filter.

Group Delay



- Frequency response:

Magnitude Response | Phase Response

Phase shift is due to a delay through the system

$$H(z)\Big|_{z=e^{j\omega}} = H\left(e^{j\omega}\right) = \left|H\left(e^{j\omega}\right)\right| \angle H\left(e^{j\omega}\right)$$

- **Group delay** (Delay generally varies with frequency):

$$\tau(\omega) = grad\{H\left(e^{j\omega}\right)\} = -\frac{d\{\angle H\left(e^{j\omega}\right)\}}{d\omega}$$

Negative slope of phase response

- Note: Phase plots normally limited in range to $\pm\pi$
  - Ignore discontinuities when evaluating derivative

Group Delay

- The group delay of the filter provides a useful measure of how the filter modified the phase characteristic of the signal.
- If we consider a signal that consists of several frequency components, the group delay of the filter is the amount of time delay each frequency component of the signal suffers in going through the filter
- The group delay on the other hand is the average time delay the composite signal suffers at each frequency as it passes from the input to the output of the filter.

Linear-Phase Filter

- Basically, a linear-phase filter gives same time delay to all frequency components of the input signal.
- A filter is said to have a linear-phase response if its phase response satisfies one of the following relationships:

$$\angle H\left(e^{j\omega}\right) = -k\ \omega$$

$$\angle H\left(e^{j\omega}\right) = b - k\ \omega$$

- where k and b are constants

Four Types of Causal Linear Phase FIR Systems

- For casual FIR systems, if their impulse response $h[n]$ satisfied the symmetrical property, then the systems will have linear phase responses.
- The symmetrical impulse response property is defined as

$$h[n] = \pm h[M-1-n], \qquad n = 0,1,\dots,M-1$$

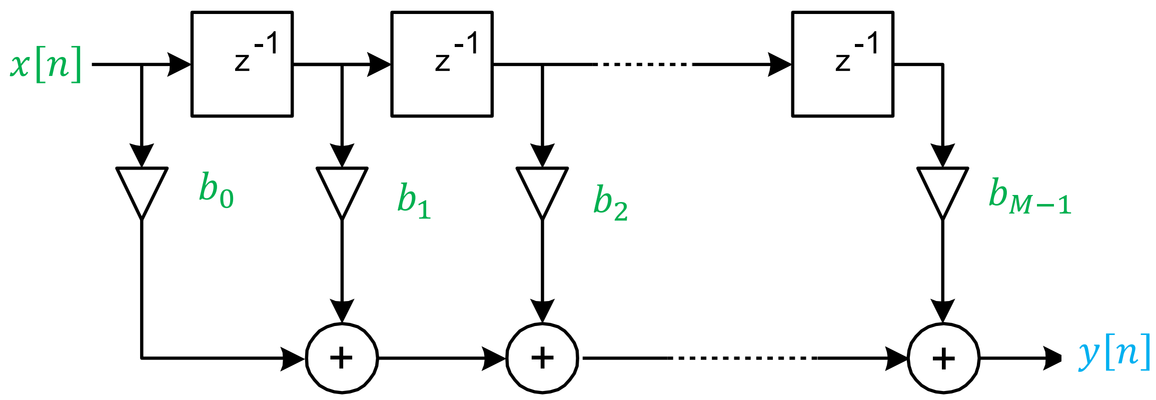
- **There 4 types of linear phase FIR systems**:
  - **Type I : Odd Positive Symmetric** – M is odd and $h[n] = h[M-1-n]$
  - **Type II : Even Positive Symmetric** – M is even and $h[n] = h[M-1-n]$
  - **Type III : Odd Negative Symmetric** – M is odd and $h[n] = -h[M-1-n]$
  - **Type IV : Even Negative Symmetric** – M is even and $h[n] = -h[M-1-n]$

FIR Filter Structures

- Direct Form structure for an FIR filter:

$$H(z) = b_0 + b_1 z^{-1} + \cdots + b_{M-1} z^{M-1} = \sum_{k=0}^{M-1} b_k z^{-k}$$

$$Y(z) = Y(z)X(z) \qquad y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_{M-1} x[n-M+1]$$



Direct Form Structure with Linear-Phase FIR Structures

- Direct form structure for an FIR filter:

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k}$$

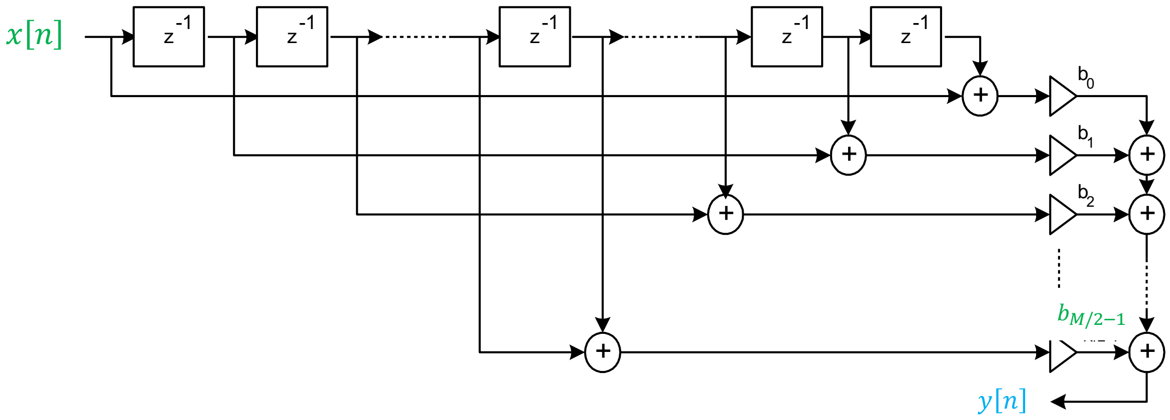
- Linear-Phase structures:
  - M even:

$$H(z) = \sum_{k=0}^{\frac{M}{2}-1} b_k \left(z^{-k} + z^{M-k-1}\right)$$

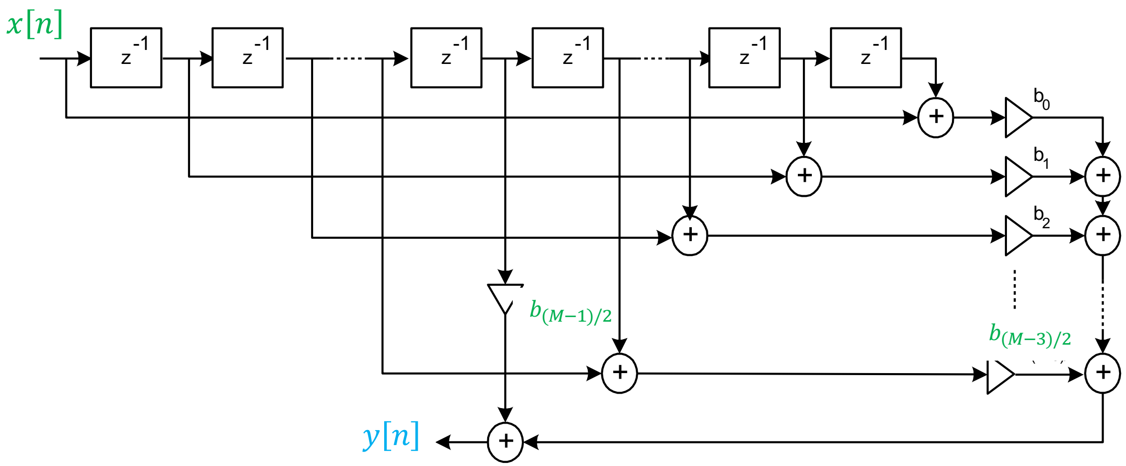
  - M Odd:

$$H(z) = \sum_{k=0}^{\frac{M-1}{2}} b_k \left(z^{-k} + z^{M-k-1}\right) + b_{\frac{M-1}{2}} z^{-\frac{M-1}{2}}$$

Linear-Phase FIR Filter Structures



(a) M even.



(b) M odd.

Design Methods of FIR Filters

Design of FIR Filters

- Basically, FIR filters are easy to understand and design
  - **Window Method**
  - **Frequency Sampling Using Inverse FFT**
    - Arbitrary Frequency Response
  - **Optimal Equiripple Method**
    - Parks-McClellan Program with Remez Exchange Algorithm
    - Computer Aided Method

Window Method

FIR Filter Design by Windowing

- Simplest way of designing FIR filters and start with ideal frequency response $H_d(e^{j\omega})$

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{n=\infty} h_d[n] e^{-j\omega n} \quad \xleftrightarrow{\text{DTFT}} \quad h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega$$

- Choose ideal frequency response as desired response
  - Most ideal impulse responses are of infinite length
  - The easiest way to obtain a causal FIR filter from ideal is

$$h_d[n] = \begin{cases} h_d[n], & 0 \le n \le M \\ 0, & otherwise \end{cases}$$

- More generally

$$h[n] = h_d[n] \cdot w[n] \qquad \text{where} \qquad w[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & otherwise \end{cases}$$

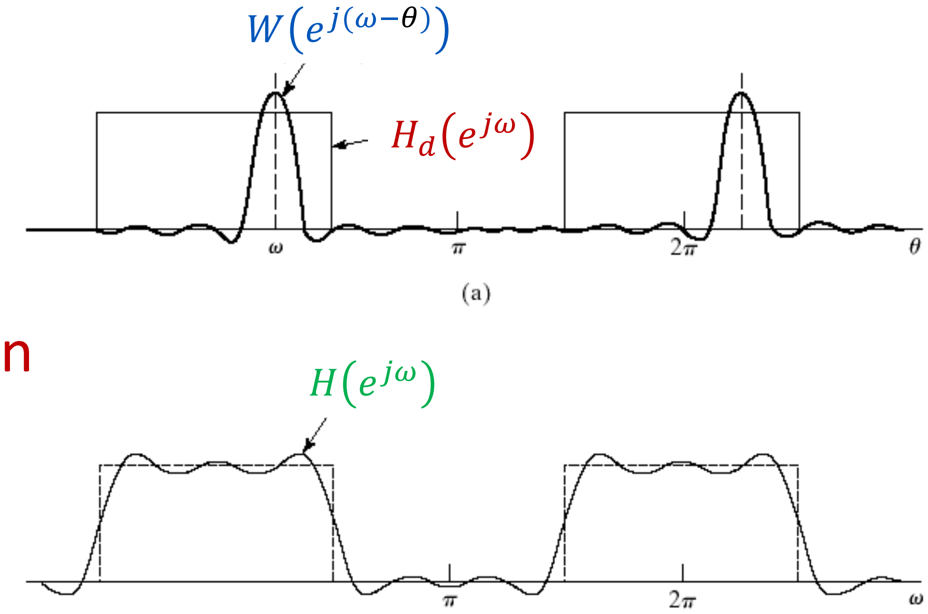
Rectangular Window Function

Windowing in Frequency Domain

- Windowed frequency response

$$H\left(e^{j\omega}\right) = W\left(e^{j\omega}\right) * H_d\left(e^{j\omega}\right)$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\left(e^{j\omega}\right) W\left(e^{j(\omega-\theta)}\right) d\theta$$



- The windowed version is smeared version of desired response
- If $w[n] = 1$ for all n, then $W\left(e^{j\omega}\right)$ is pulse train with 2π period

Rectangular Window

$$w[n] = \begin{cases} 1 & 0 \le n \le M-1 \\ 0 & otherwise \end{cases}$$

$$W(z) = \sum_{n=0}^{M-1} w[n]z^{-n} = 1 + z^{-1} + z^{-2} + \cdots + z^{-(M-1)} = \frac{1-z^{-M}}{1-z^{-1}}$$

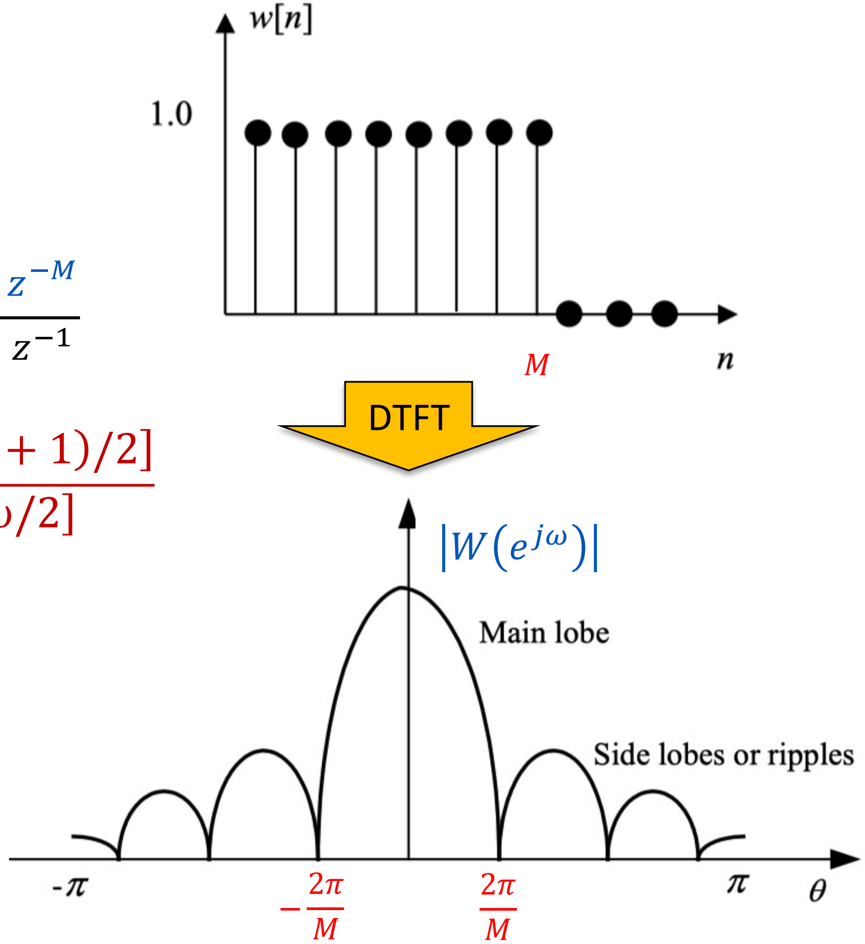
$$W(e^{j\omega}) = W(z)\Big|_{z=e^{j\omega}} = \frac{1-e^{-jM\omega}}{1-e^{-j\omega}} = e^{-j\omega n}\frac{\sin[\omega(M+1)/2]}{\sin[\omega/2]}$$

Magnitude Response

$$|W(e^{j\omega})| = \left|\frac{\sin\frac{M\omega}{2}}{\sin\frac{\omega}{2}}\right|$$

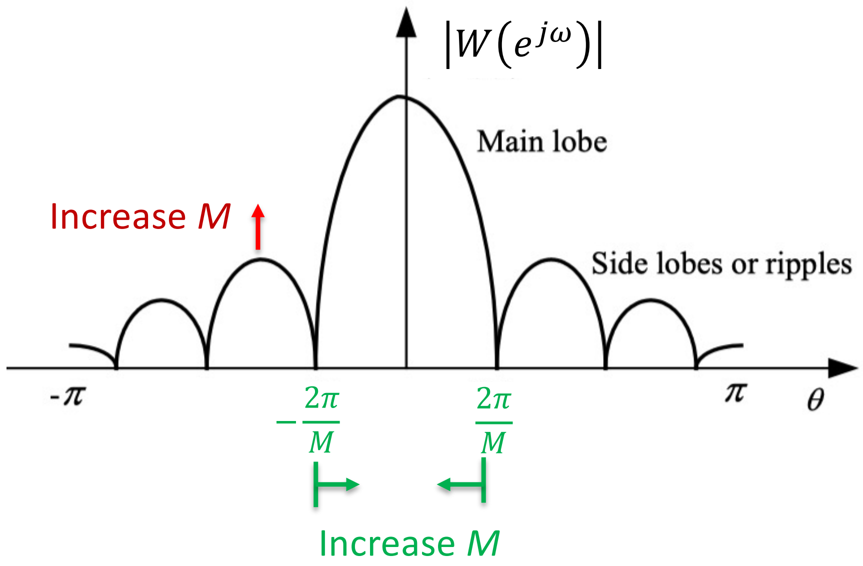
Phase Response

$$\angle W(e^{j\omega}) = -\left(\frac{N-1}{2}\right)\omega$$



Effect of Rectangular Window Side (*M*)

- If *M* increases the width of the main lobe decreases but the peak amplitude of the side lobes grows in a manner such that the area under each lobe is constant while the width of each lobe decreases with *M*.



Design of FIR Filters Using Windows

- The easiest way to obtain an FIR filter is to simply truncate the impulse response of an IIR filter. If $h_d[n]$ represents the impulse response of a desired IIR filter, then an FIR filter with impulse response $h[n]$ can be obtained as follows:

$$h[n] = \begin{cases} h_d[n] & M_1 \leq n \leq M_2 \\ 0 & otherwise \end{cases}$$

- In general $h[n]$ can be thought of as being formed by the product $h_d[n]$ and a window function $w[n]$ as follows:

$$h[h] = h_d[n] \cdot w[n]$$

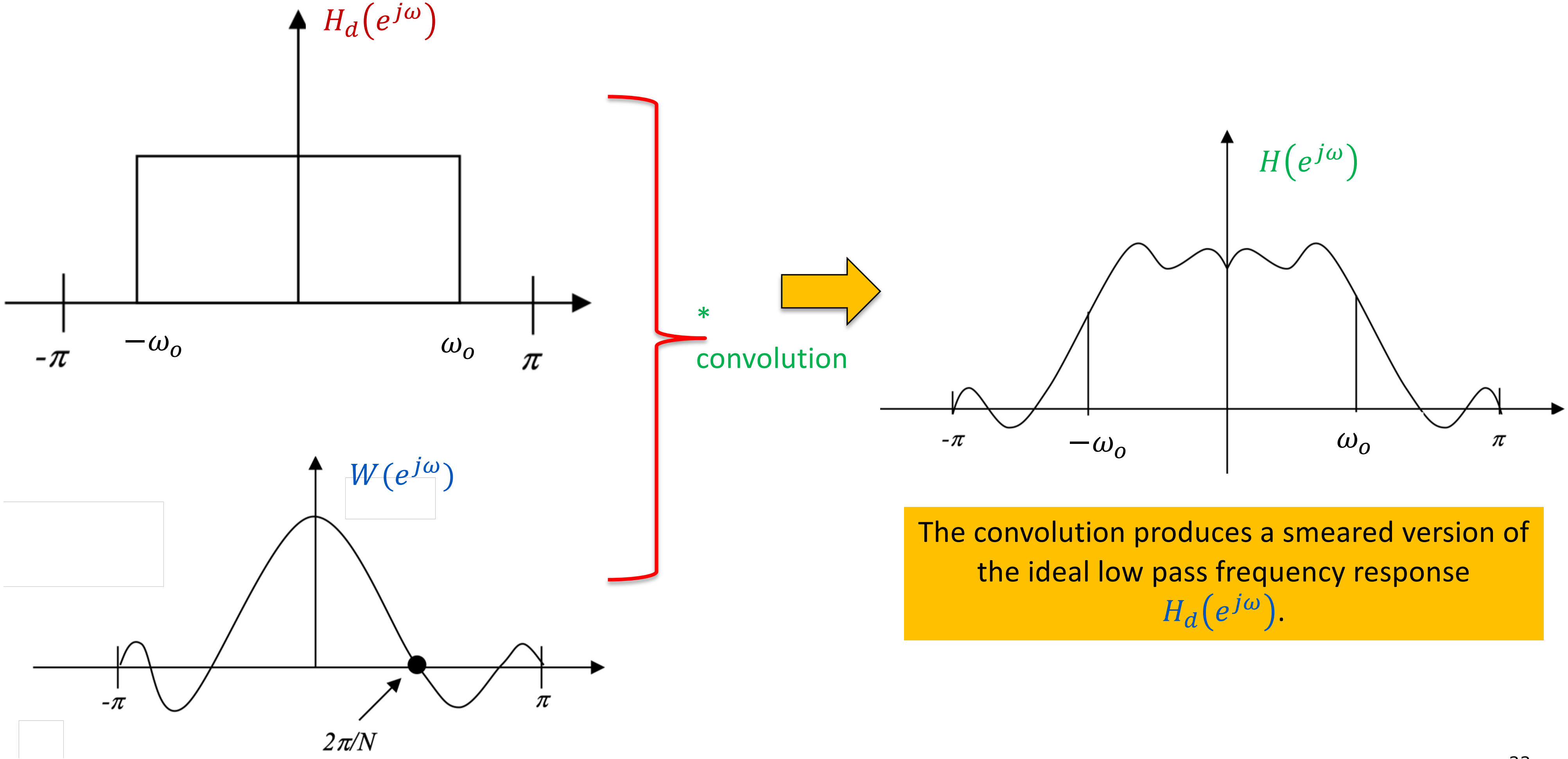
Let it be, for example a rectangular window

$$\Downarrow$$

$$H(e^{j\omega}) = H_d(e^{j\omega}) * W(e^{j\omega})$$

Let the desired filter be an ideal low-pass filter with cut off frequency ω_o

Effect of the Rectangular Window on Frequency Response (1)



The convolution produces a smeared version of the ideal low pass frequency response $H_d(e^{j\omega})$.

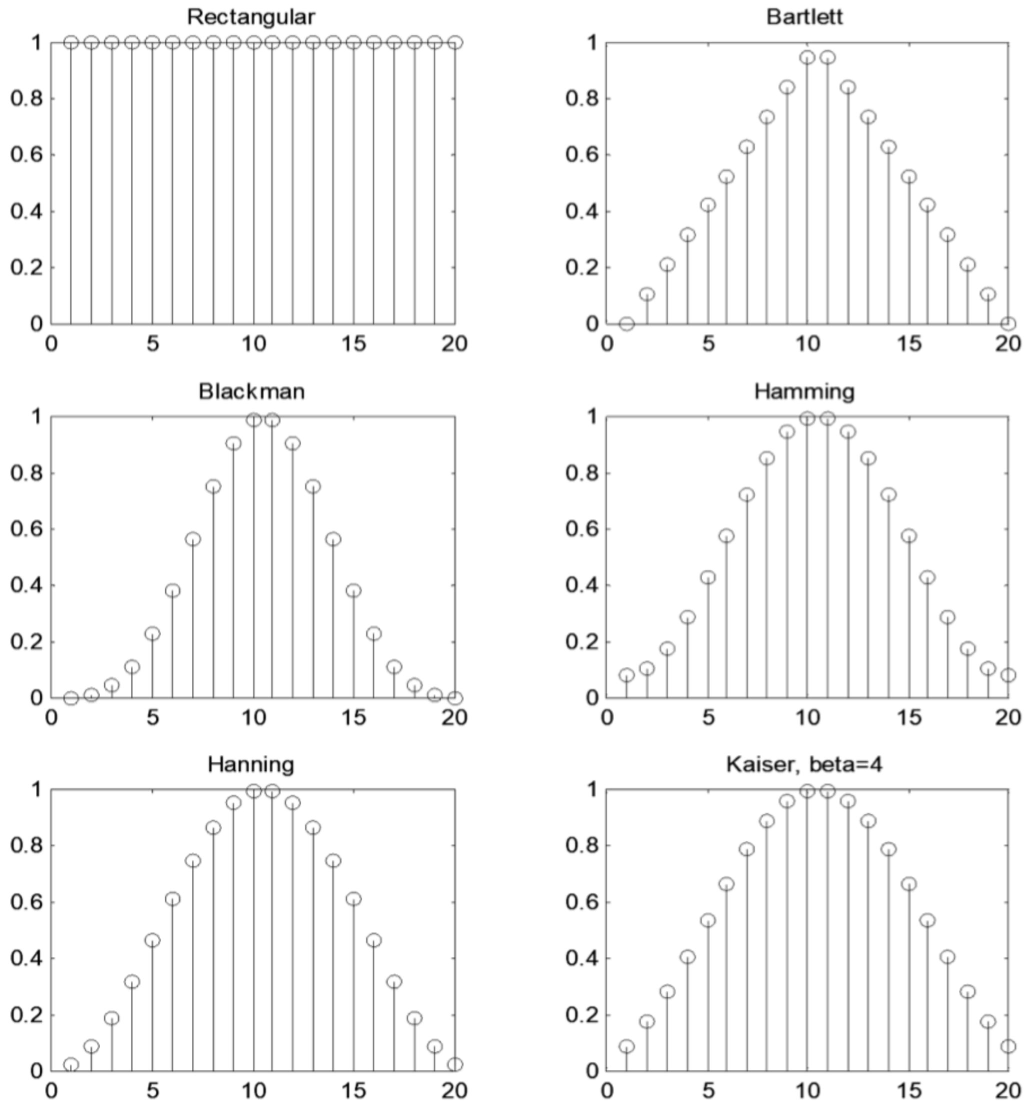
Effect of the Rectangular Window on Frequency Response

- Therefore, it is seen that the convolution produces a smeared version of the ideal low-pass frequency response $H_d(e^{j\omega})$.
- In general, the wider the main lobe of $W(e^{j\omega})$, the more spreading, whereas the narrower the main lobe (larger *M*) the closer $|H(e^{j\omega})|$ comes to $|H_d(e^{j\omega})|$.
- In general, we are left with a trade-off on making *M* large enough so that smearing is minimized, yet small enough to allow reasonable implementation.

Window Functions

Window functions are used **to truncate** a signal to produce a signal of **finite duration**.

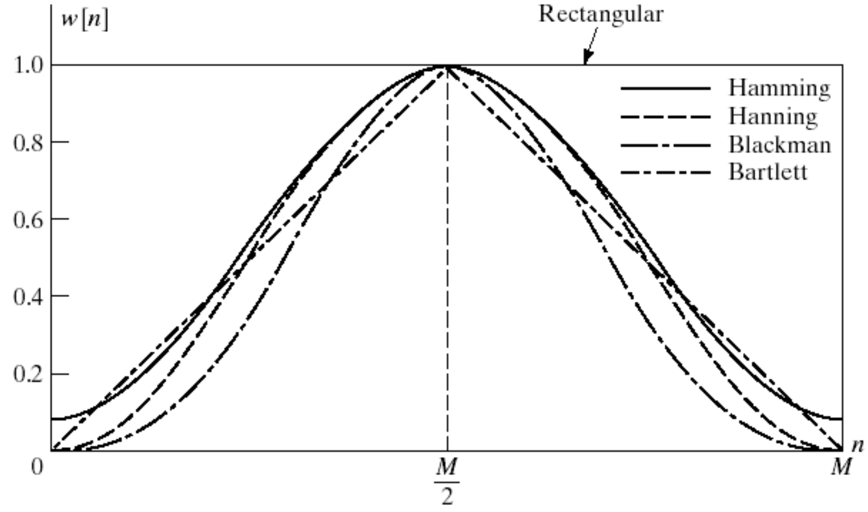
- **Rectangular**
- **Blackman**
- **Bariett**
- **Hanning**
- **Hamming**
- **Kaiser**

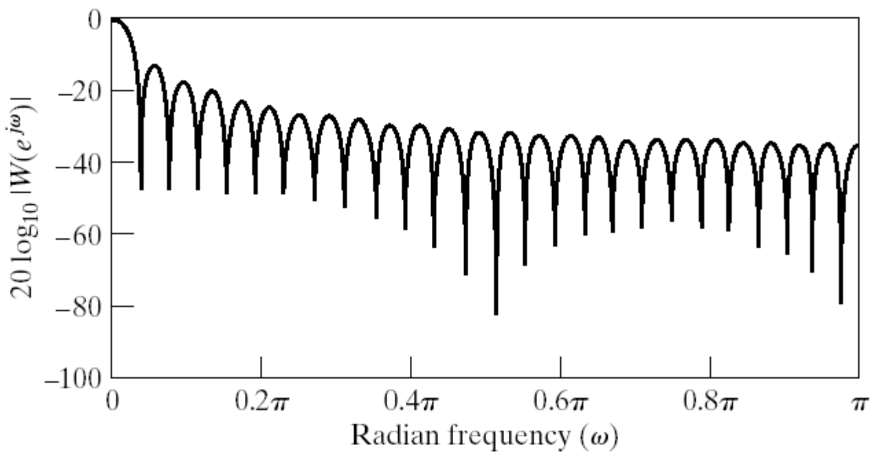


Rectangular Window

- Narrowest main lob
  - $4\pi/(M+1)$
  - Sharpest transitions at discontinuities in frequency
- Large side lobs
  - **-13 dB**
  - Large oscillation around discontinuities
- Simplest window possible

$$w[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & otherwise \end{cases}$$

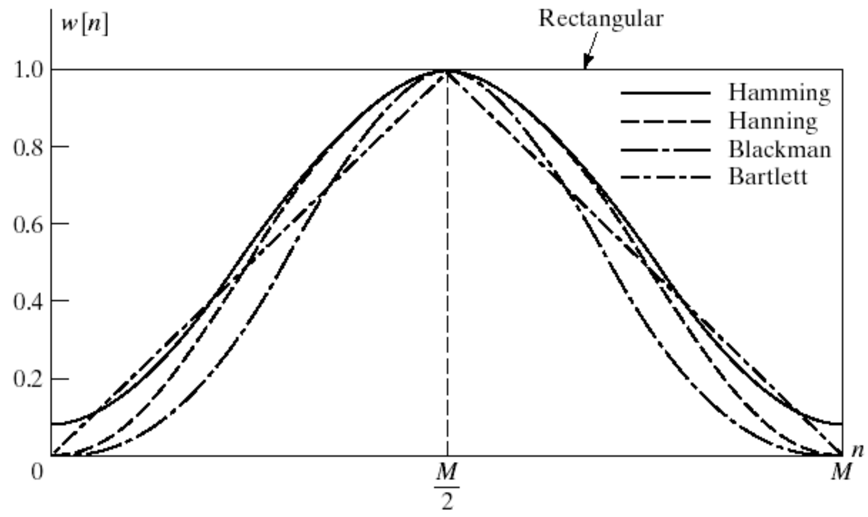


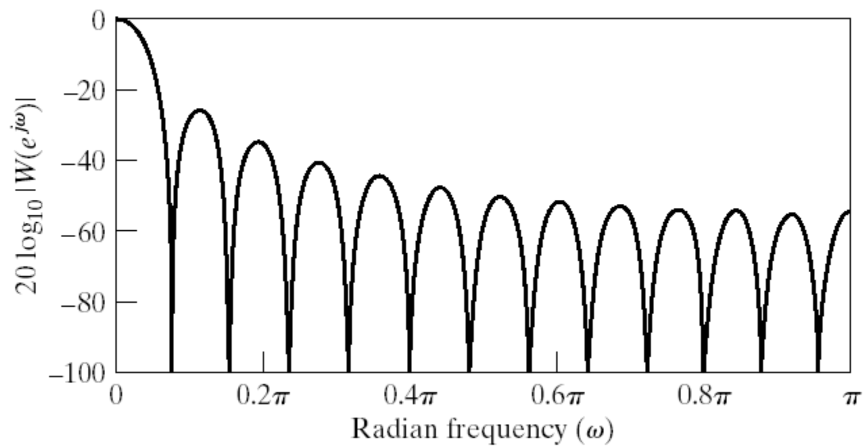


Bartlett Window

- Medium main lob
  - $8\pi/M$
- Side lobs
  - -25 dB
- Hamming window performs better
- Simple equation

$$w[n] = \begin{cases} 2n/M & 0 \le n \le M/2 \\ 2 - 2n/M & \frac{M}{2} \le n \le M \\ 0 & otherwise \end{cases}$$

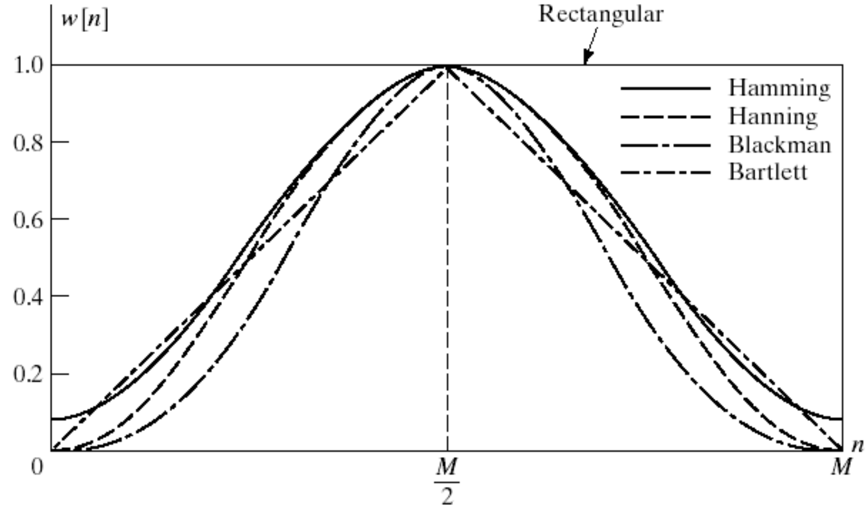


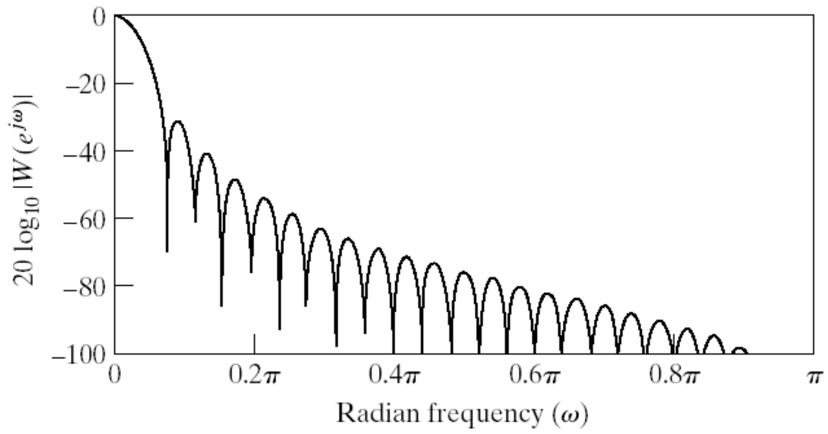


Hanning Window

- Medium main lob
  - $8\pi/M$
- Side lobs
  - **-31 dB**
- Hamming window performs better
- Same complexity as Hamming

$$w[n] = \begin{cases} \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{M}\right)\right], & 0 \le n \le M \\ 0, & otherwise \end{cases}$$

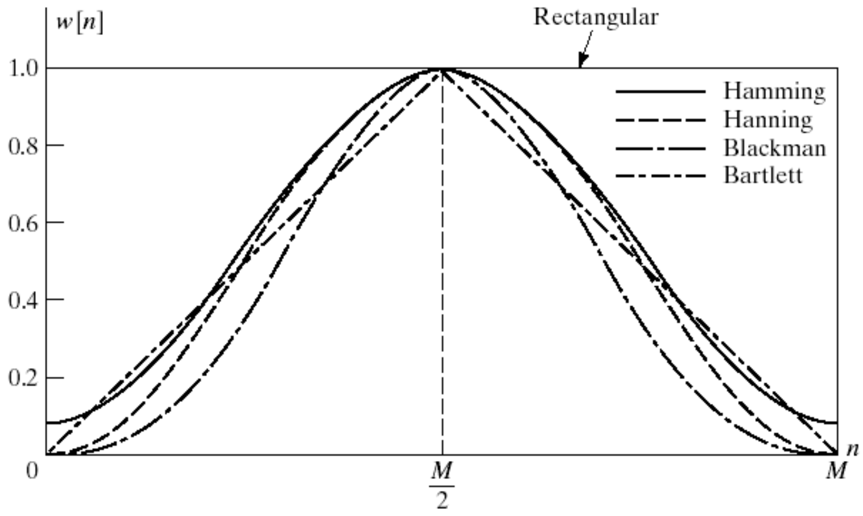


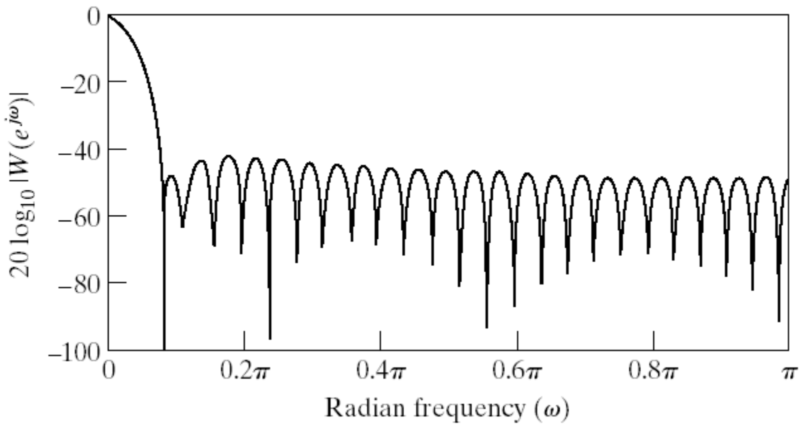


Hamming Window

- Medium main lob
  - $8\pi/M$
- Good side lobs
  - **-41 dB**
- Simpler than Blackman

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{M}\right), & 0 \le n \le M \\ 0, & otherwise \end{cases}$$

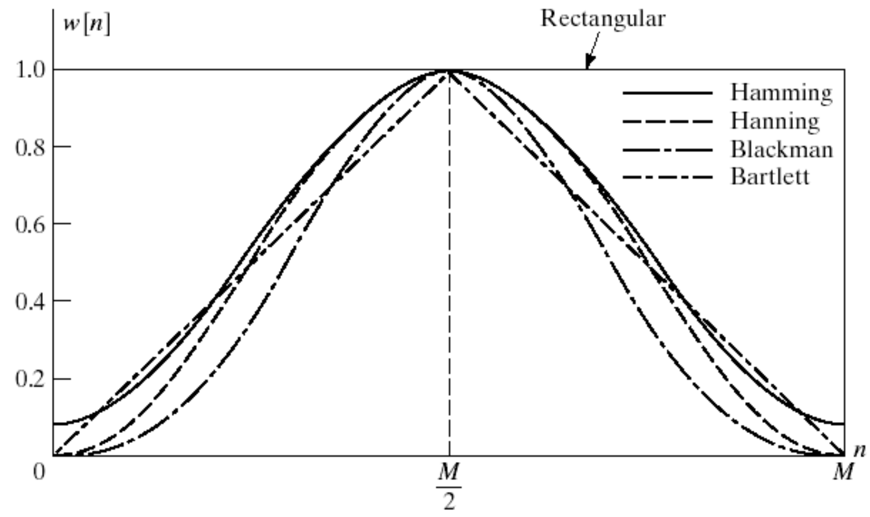


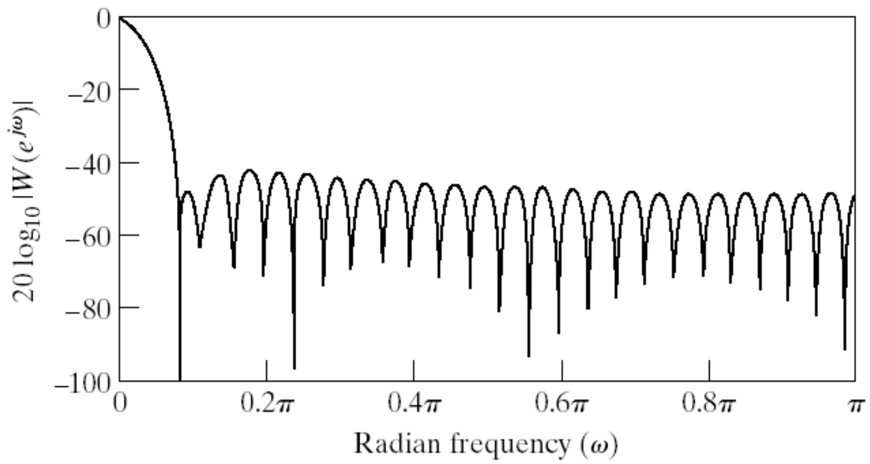


Blackman Window

- Medium main lob
  - $12\pi/M$
- Very good side lobs
  - **-57 dB**
- Complex equation

$$w[n] = \begin{cases} 0.42 - 0.5\cos\left(\frac{2\pi n}{M}\right) + 0.08\cos\left(\frac{4\pi n}{M}\right), & 0 \le n \le M \\ 0, & otherwise \end{cases}$$



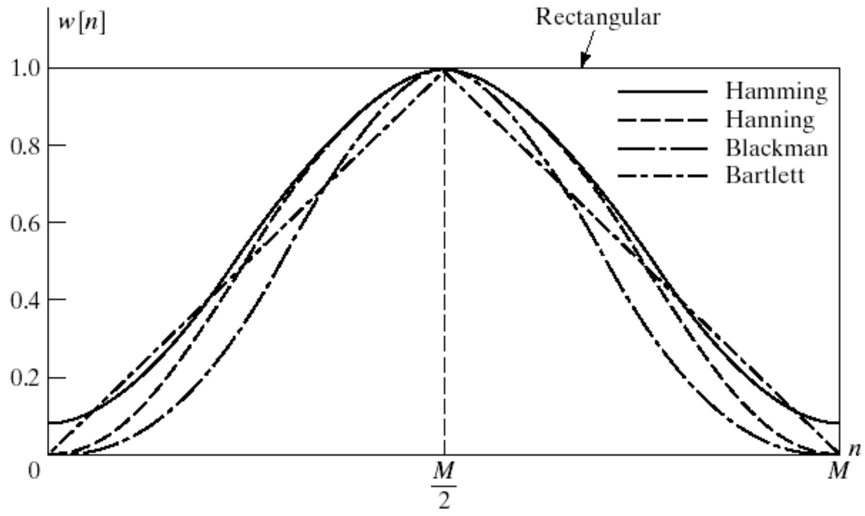


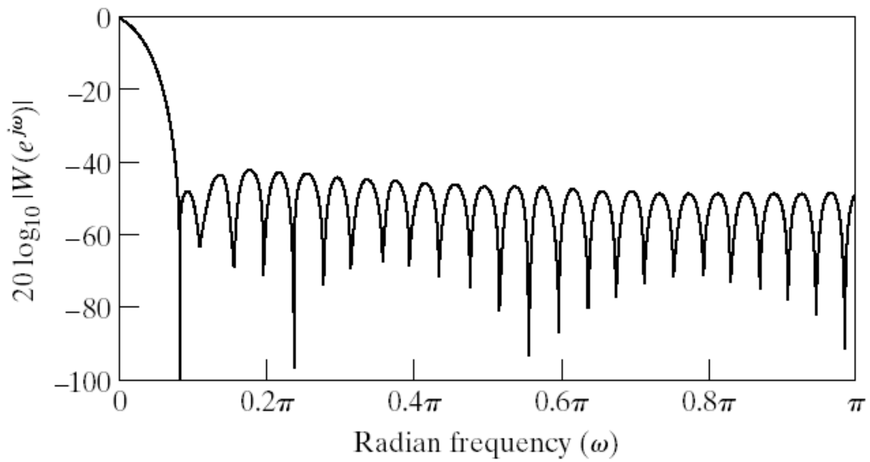
Kaiser Window

- Parameterized equation forming a set of windows
- Parameter to change main-lob width and side-lob area trade-off

$$w[n] = \begin{cases} \dfrac{I_0\left[\beta\sqrt{1-\left(\dfrac{n-M/2}{M/2}\right)^2}\right]}{I_0(\beta)} & 0 \le n \le M \\ 0 & \text{else} \end{cases}$$

- $I_0(\bullet)$ represents zeroth-order modified Bessel function of 1st kind





Determining Kaiser Window Parameters

- Given filter specifications Kaiser developed empirical equations
  - Given the peak approximation error δ or in dB as $A=-20\log10\ \delta$
  - and transition band width
- The shape parameter β should be

$$\beta = \begin{cases} 0.1102(A-8.7) & A > 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21) & 20 \le A \le 50 \\ 0 & A < 21 \end{cases}$$

- The filter order M is determined approximately by

$$M = \frac{A-8}{2.285\Delta\omega}$$

Design Procedure of Window Method (1)

- An ideal low-pass filter with linear phase of slope-β and cut-off ω_c can be characterized in the frequency domain by

Linear Phase Response

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\beta\omega}, & |\omega| \le \omega_c \\ 0., & \omega_c < |\omega| \le \pi \end{cases}$$

- The corresponding impulse response $h_d[n]$ can be obtained by taking the inverse Fourier transform of $H_d(e^{j\omega})$ and easily shown to be

$$h_d[n] = \frac{\sin[\omega_c(n-\beta)]}{\pi(n-\beta)}$$

Design Procedure of Window Method (2)

- A causal FIR filter with impulse response $h[n]$ can be obtained by multiplying $h_d[n]$ by a window $w[n]$ beginning at the origin and ending at $M-1$ as follows :

$$h[n] = \frac{\sin[\omega_c(n-\beta)]}{\pi(n-\beta)} w[n]$$

- For $h[n]$ to be a linear phase , β must be selected so that the resulting $h[n]$ is symmetric.
- As $\sin[\omega_c(n-\beta)]/\pi(n-\beta)$ is symmetric about $n=\beta$ and the window is symmetric about $n=(M-1)/2$

$$\beta = \frac{N-1}{2}$$

← Symmetric about β

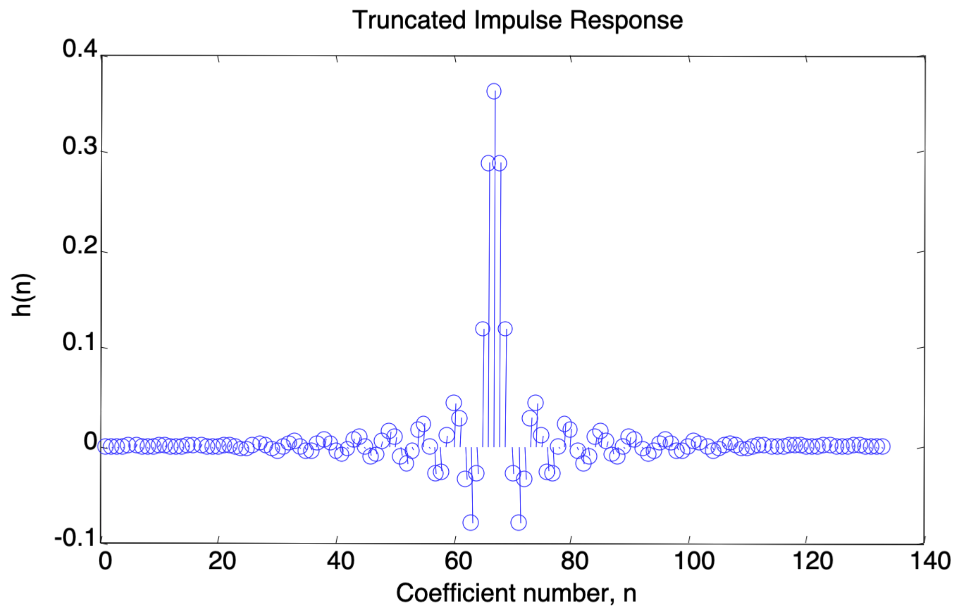
Window Function Selection

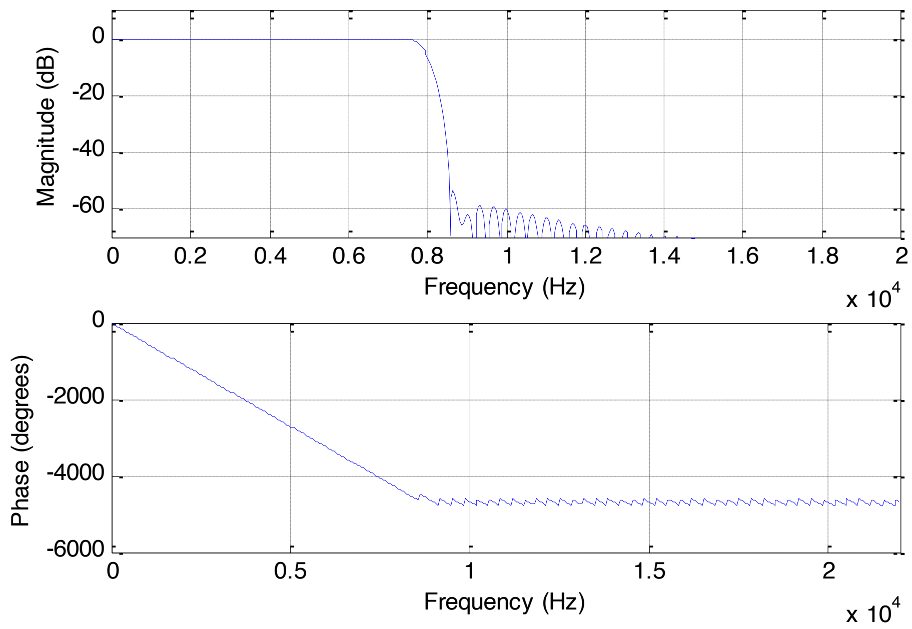
- In the second stage of this method, we need to select a window function based on the passband or attenuation specifications, then determine the filter length based on the required width of the transition band.

Window Type	Normalised Transition Width (Δf(Hz))	Passband Ripple(dB)	Stopband Attenuation (dB)
Rectangular	$\frac{0.9}{N}$	0.7416	21
Hanning	$\frac{3.1}{N}$	0.0546	44
Hamming	$\frac{3.3}{N}$	0.0194	53
Blackman	$\frac{5.5}{N}$	0.0017	74
Kaiser	$\frac{2.93}{N} \rightarrow \beta = 4.54$ $\frac{5.71}{N} \rightarrow \beta = 8.96$	0.0274 0.000275	50 90

- For example, using Hamming Window: $M = \frac{3.3}{\Delta f} = \frac{3.3}{1200-1400} \cdot 8000 = 132$

Truncated Window Function and Frequency Response





FIR Filter Design : Window Method Example

(a) Determine the impulse response $h_d[n]$ of the lowpass filter whose frequency response is given by

$$H_d(e^{j\omega}) = \begin{cases} 1 & 0 \le |\omega| \le \dfrac{\pi}{3} \\ 0 & \dfrac{\pi}{3} < |\omega| \le \pi \end{cases}$$

(b) To obtain a finite impulse response from $h_d[n]$ a rectangular window of length N = 9 is used. Compute the coefficients of the FIR filter with a linear-phase characteristic ands with this finite impulse response.

Solution for (a)

- We first determine the impulse response $h_d[n]$ of the lowpass filter by inverse DTFT

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega = \frac{1}{2\pi}\int_{-\frac{\pi}{3}}^{\frac{\pi}{3}} e^{j\omega n}d\omega = \frac{1}{2}\left[\frac{e^{j\omega n}}{jn}\right]_{-\frac{\pi}{3}}^{\frac{\pi}{3}} = \frac{1}{\pi}\left[\frac{\sin\left(n\frac{\pi}{3}\right)}{n}\right]$$

$$h_d[n] = \begin{cases} \dfrac{1}{3} & n = 0 \\ \dfrac{\sin\left(n\frac{\pi}{3}\right)}{n\pi} & n \neq 0 \end{cases}$$

Solution for (b)

- For a linear phase filter, $h_d[n]$ is positive symmetric at $n = 0$ and the window is symmetric at $n = (N-1)/2 = (9-1)/2 = 4$

$$h[n] = h_d[n-4] \cdot w[n] \qquad n = 0,1,\ldots,8$$

- The coefficients are

n	0	1	2	3	4	5	6	7	8
Filter Coefficients	$\frac{-\sqrt{3}}{8\pi}$	0	$\frac{\sqrt{3}}{4\pi}$	$\frac{\sqrt{3}}{2\pi}$	0.333	$\frac{\sqrt{3}}{2\pi}$	$\frac{\sqrt{3}}{4\pi}$	0	$\frac{-\sqrt{3}}{8\pi}$

Symmetry about n=4

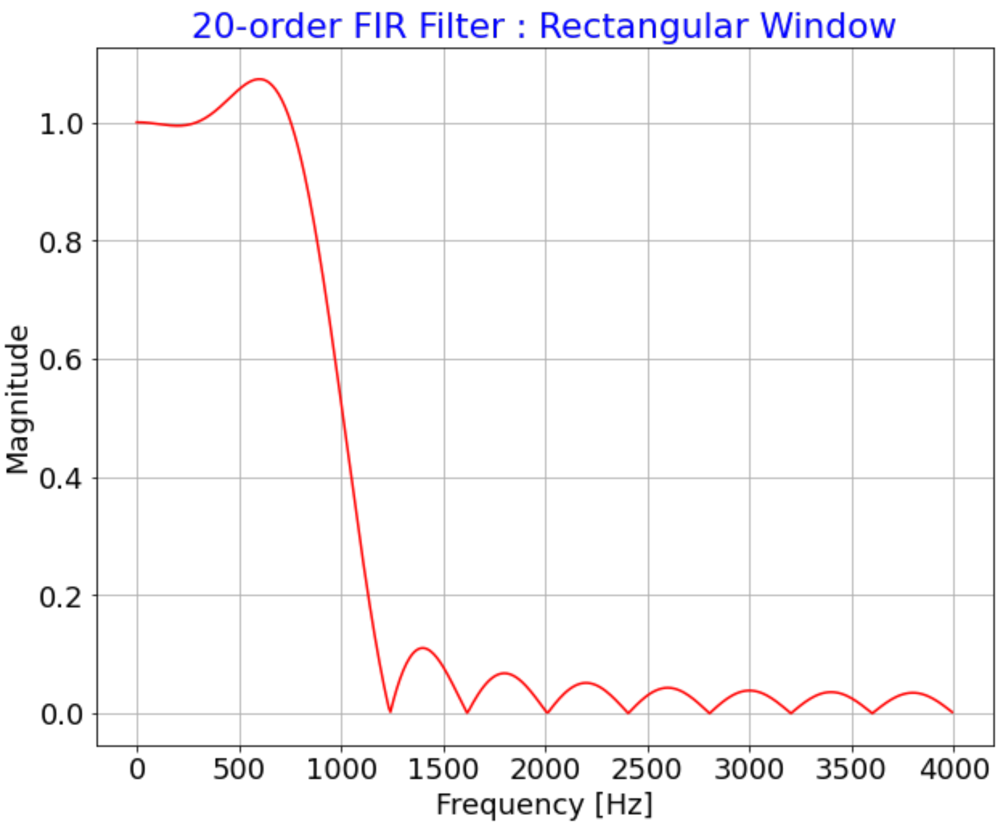
Python FIR Design Using Window Method

Filter specifications:

- Sampling Frequency 8kHz : $F_s = 8000$
- Cutoff frequency at 1kHz : $f_c = 1000$
- Transition bandwidth 800Hz : $\Delta f = 800$
- Use Window Functions : Rectangular, Hanning, Hamming and Blackman
  - Rectangular window : $M = \frac{0.9}{\Delta f} F_s = 0.9 \cdot 10 = 9$ This estimated order is too low. 20 order is required for rectangular.
  - Hanning window : $M = \frac{3.1}{\Delta f} F_s = 31$
  - Hamming window : $M = \frac{3.3}{\Delta f} F_s = 33$
  - Blackman window : $M = \frac{5.5}{\Delta f} F_s = 55$

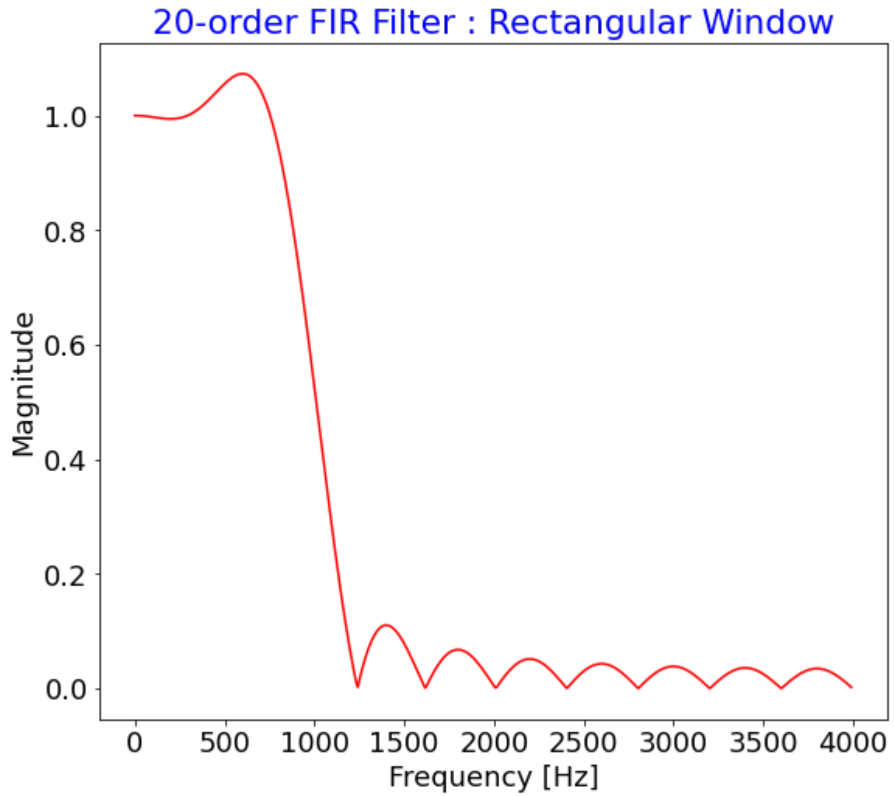
Python Code: FIR Filter using Window Method

```
# Lowpass Design using signal.firwin()
import matplotlib.pyplot as plt
from scipy import signal
import numpy as np

Fs = 8000 # Sampling Rate 8KHz
M = 20 # Filter Order = 20
f_c = 1000 # Cutoff Frequency 1000

h = signal.firwin(M, f_c, window='boxcar', fs=Fs)

w, h = signal.freqz(h, [1], fs=Fs)

# Plot the Magnitude Response
fig = plt.figure()
ax1 = fig.add_subplot(1, 1, 1)
ax1.set_title('9-order FIR Filter : Rectangular Window',
color='b')
ax1.plot(w, np.abs(h), 'r')
ax1.set_ylabel('Magnitude')
ax1.set_xlabel('Frequency [Hz]')
ax1.grid()
plt.show()
```

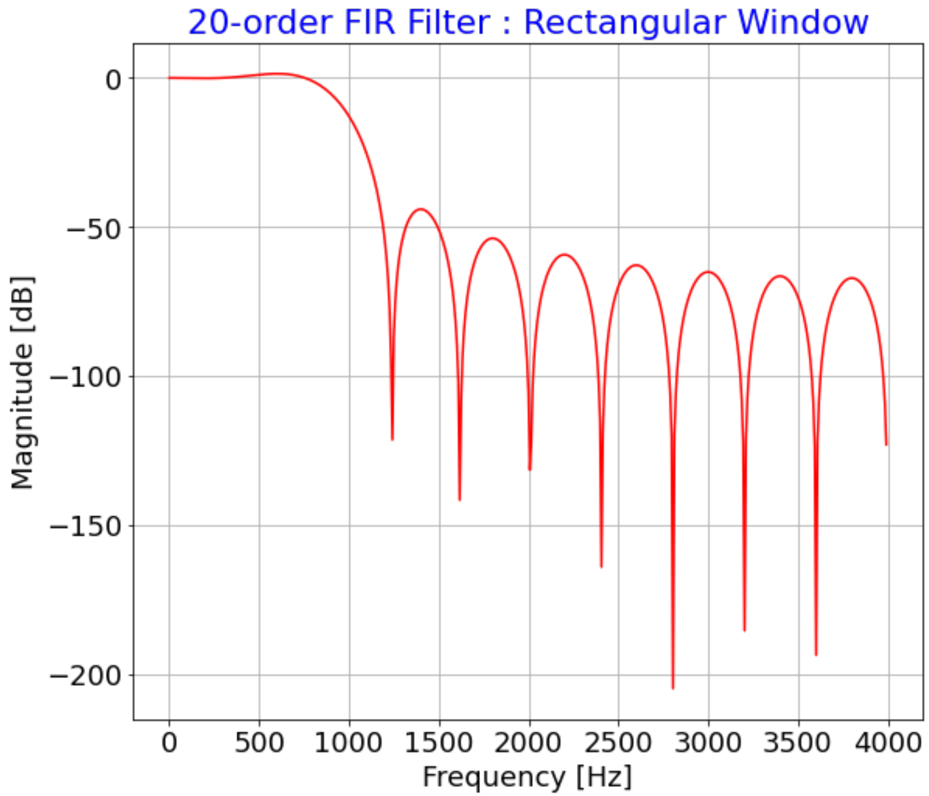


https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.firwin.html

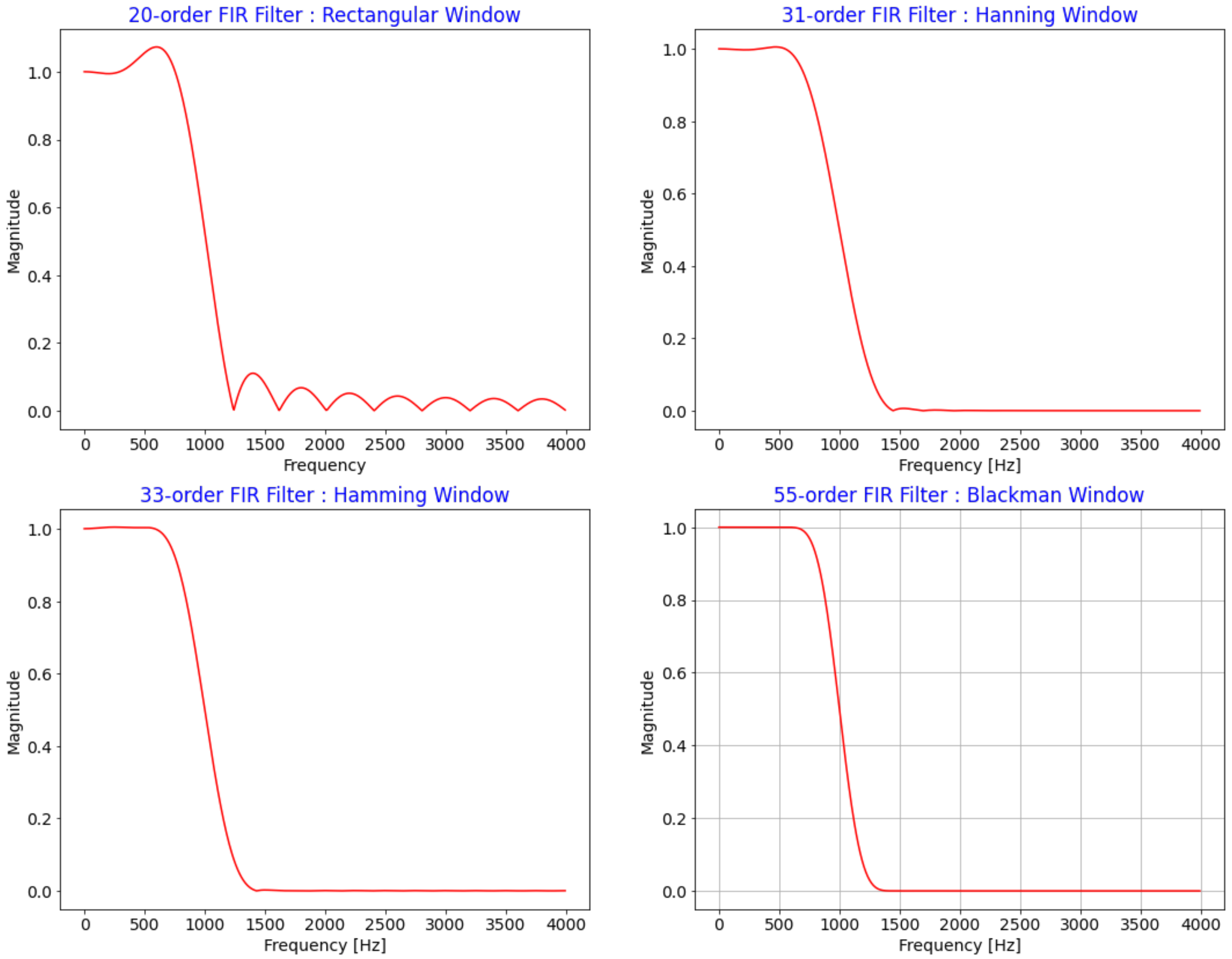
Rectangular Window



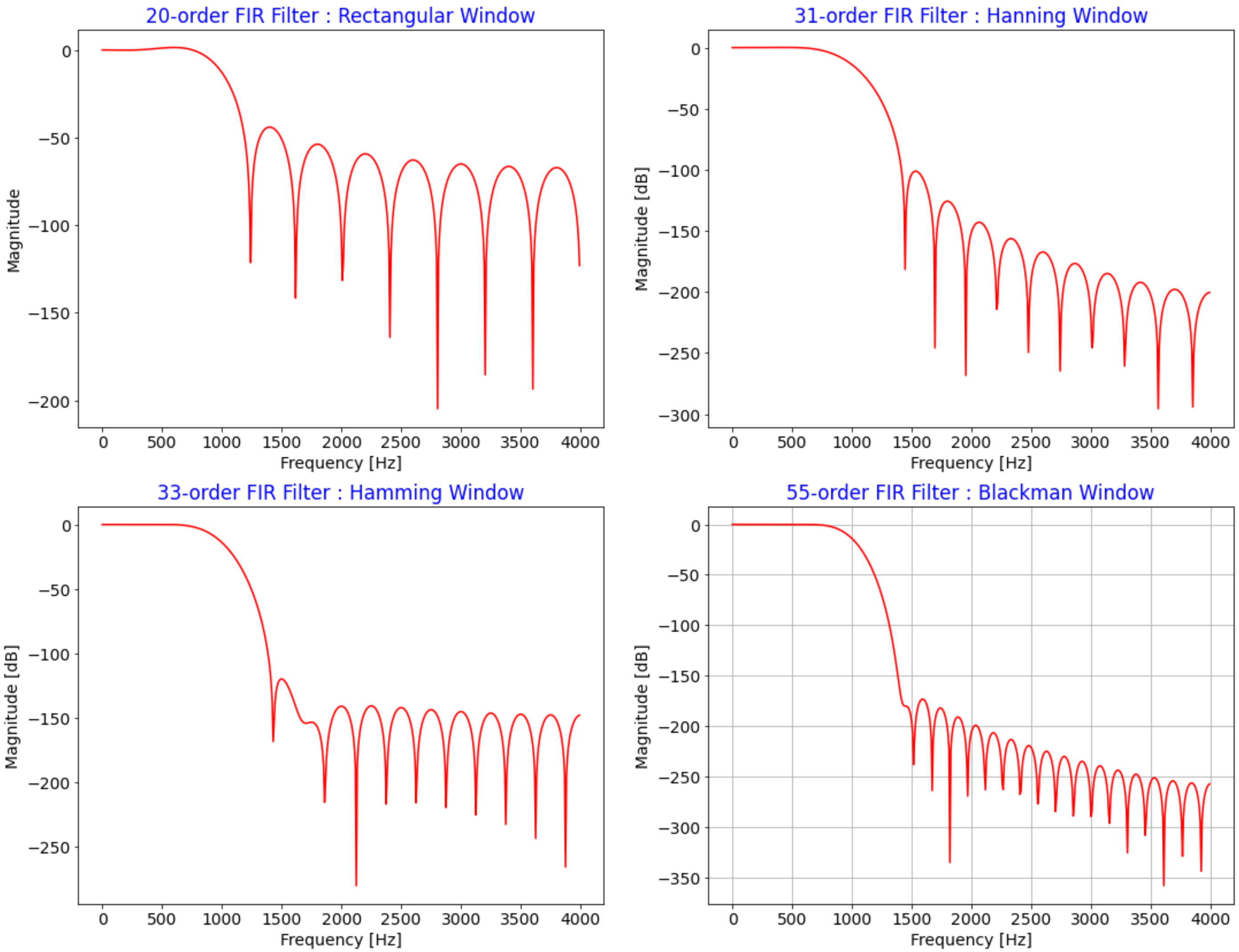

```
ax1.plot(w, np.abs(h), 'r')
```




```
ax1.plot(w, 20*np.log(h), 'r')
```



20-order FIR Filter : Rectangular Window
Magnitude
Frequency
31-order FIR Filter : Hanning Window
Magnitude
Frequency [Hz]
33-order FIR Filter : Hamming Window
Magnitude
Frequency [Hz]
55-order FIR Filter : Blackman Window
Magnitude
Frequency [Hz]

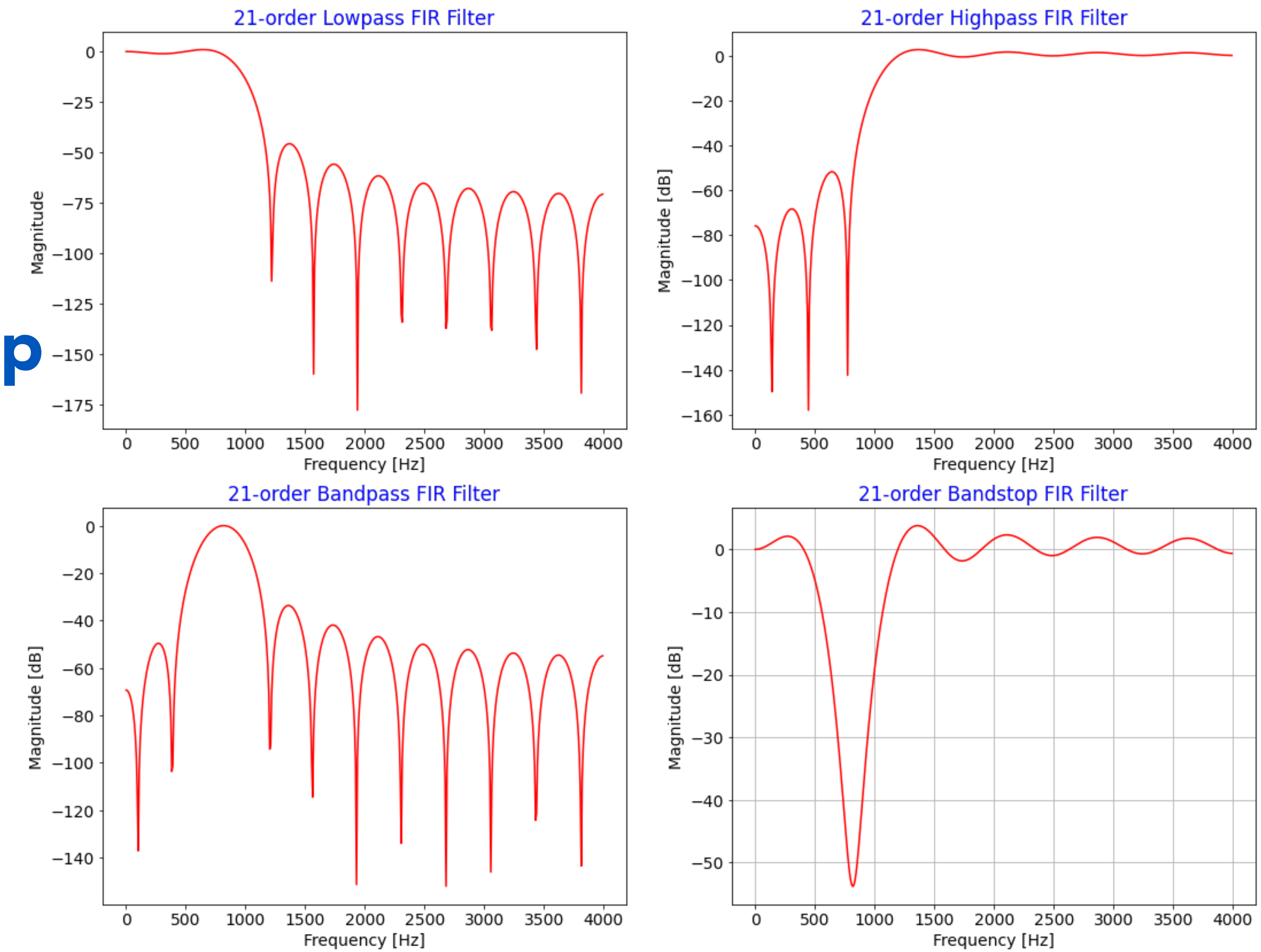


 https://colab.research.google.com/drive/1h8hS76WAOjFZkf0O_-31BBcj32412xUp?usp=sharing

Lowpass, Higpass, Bandpass, and Bandstop FIR Filters

```
h1 = signal.firwin(21, f_c, window='boxcar', fs=Fs)
w, h1 = signal.freqz(h1, [1], fs=Fs)

h2 = signal.firwin(21, f_c, window='boxcar', pass_zero=False, fs=Fs)
w, h2 = signal.freqz(h2, [1], fs=Fs)

f1 = 600
f2 = 1000

h3 = signal.firwin(21, [f1,f2], window='boxcar', pass_zero=False, fs=Fs)
w, h3 = signal.freqz(h3, [1], fs=Fs)

h4 = signal.firwin(21, [f1,f2], window='boxcar', fs=Fs)
w, h4 = signal.freqz(h4, [1], fs=Fs)
```

Lowpass, Higpass, Bandpass, and Bandstop FIR Filters



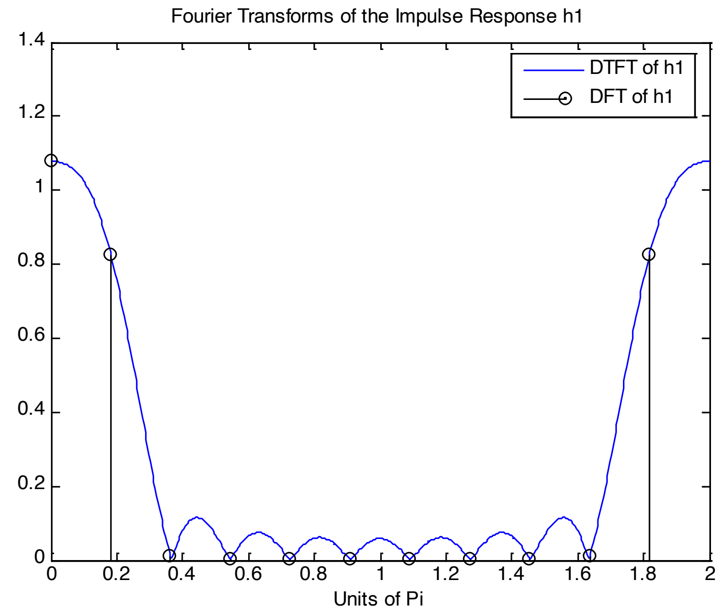
Frequency Sampling Method

Frequency Sampling Method

- The basic idea is to utilize the Discrete Fourier Transform (DFT), which corresponds to samples of the desired frequency response $H_d(e^{j\omega})$, to produce $h[n]$:

$$H[k] = H_d(e^{j\omega})\Big|_{\omega=\frac{2\pi}{M}k}, \qquad k = 0,1,\ldots,M-1$$

- The DFT ($H[k]$) is equal to $H_d(e^{j\omega})$ sampled at M distinct frequencies between $\omega \in (0,2\pi)$ with a uniform frequency spacing of $2\pi/N$



FIR Filter Impulse Response Obtained by IDFT

- Basically, the DFT coefficients $H[k]$ can be considered as *M*-sampled version of DTFT for a *M*-point finite impulse response $h[n]$ between the discrete frequency ω range $[0,2\pi]$

$$H[k] = \sum_{n=0}^{M-1} h[n]e^{-j\frac{2\pi}{M}kn} \qquad k = 0,1,2,\dots,M-1$$

- Therefore, given the frequency response $H[k]$, the impulse response $h[n]$ can be computed from the inverse DFT (IDFT) of the frequency response

$$h[n] = \frac{1}{M}\sum_{k=0}^{N-1} H[k]e^{j\frac{2\pi}{M}kn} \qquad n = 0,1,2,\dots,M-1$$

Ideal Lowpass Lowpass Filter

1. An ideal lowpass filter with linear phase of slope-k and cutoff ω_c can be characterized in the frequency domain by

Linear Phase Response

$$H_d(e^{j\omega}) = \begin{cases} e^{-jk\omega} & |\omega| \leq \omega_c \\ 0 & \omega_c < |\omega| \leq \pi \end{cases}$$

2. The corresponding impulse response $h_d[n]$ can be obtained by taking the Inverse Fourier Transform of $H_d(e^{j\omega})$ and easily shown to be

$$h_d[n] = \frac{\sin[\omega_c(n-k)]}{\pi(n-k)}$$

Procedure of Frequency Sampling Method

- **Step 1**: Determine the critical discrete-time frequency ω_c
- **Step 2**: Determine the filter order M, where M is even
- **Step 3**: Construct a vector of $M+1$ real-valued frequency response values evenly spaced from $\omega = 0$ to 2π.

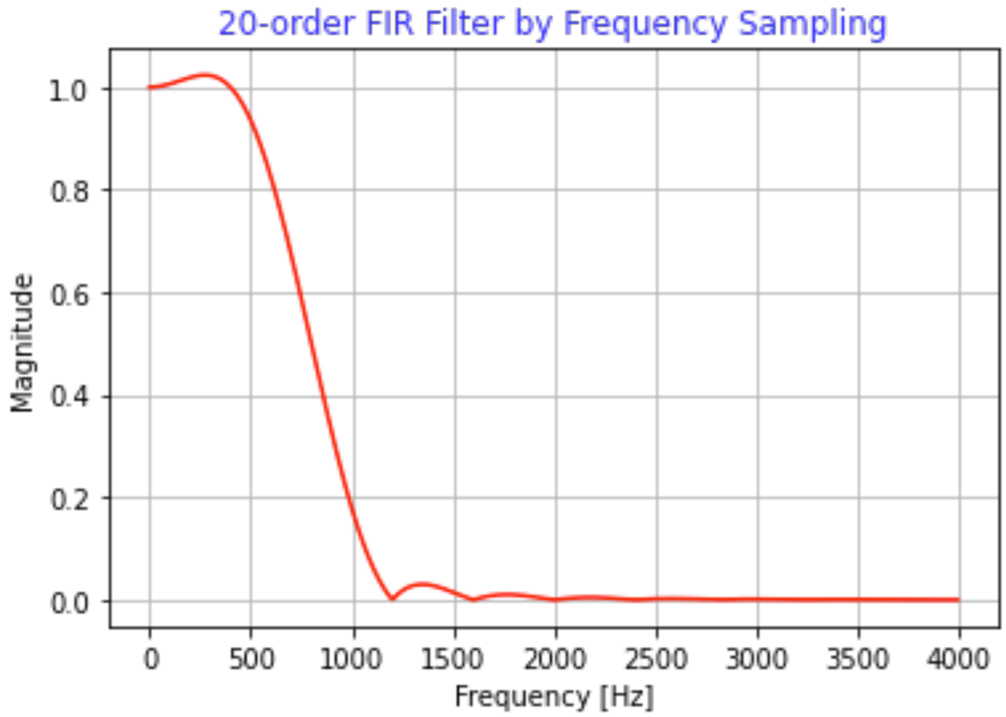
Python : Frequency Sampling Method

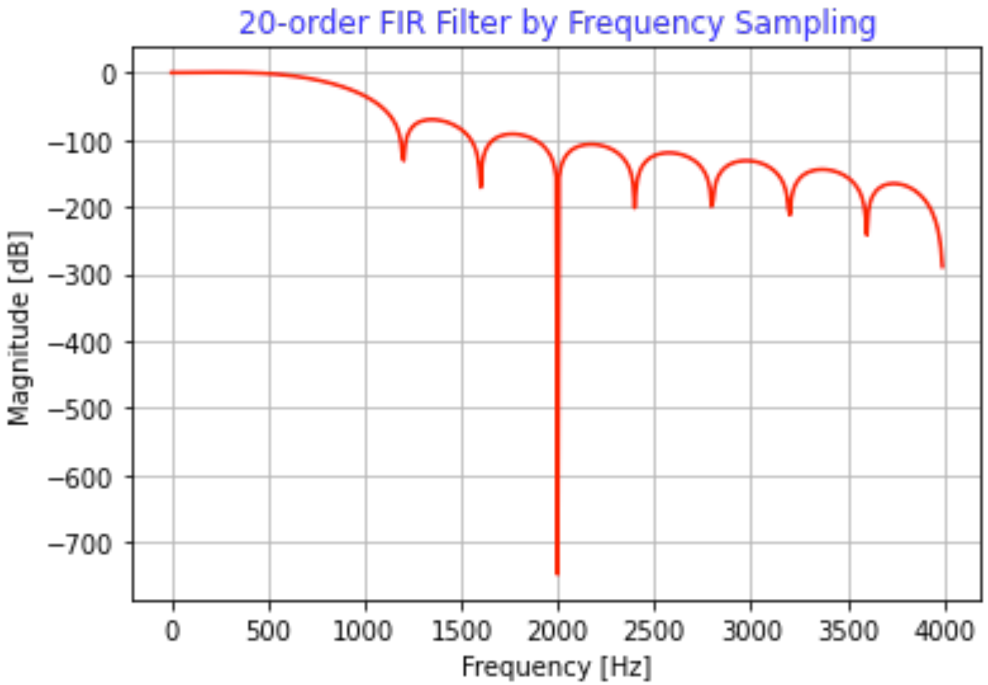
```
# Frequency sampling design of linear phase FIR filter
import matplotlib.pyplot as plt
from scipy import signal
import numpy as np

Fs = 8000 # Sampling Rate = 8kHz
N = 20 # Filter Order
# Create index n vector
n = np.arange(0, N,1)
# Frequency vector
w = n*2*np.pi/N
# Linear Phase = -k*w
k = np.floor((N-1)/2)

# Define Ideal Lowpass Magnitude Response
M = np.ones(N)
M[2:18]=0
D = M*np.exp(-1j*k*w)

# Compute the impulse response h[n] by IFFT
h = np.fft.ifft(D)
h = np.real(h)
```

```
# Plot the Magnitude Response
w, h = signal.freqz(h, [1], fs=2000)

fig = plt.figure()
ax1 = fig.add_subplot(1, 1, 1)
ax1.set_title('Lowpass Filter Desing by
Frequency Sampling', color='b')

ax1.plot(w, abs(h), 'r')
ax1.set_ylabel('Magnitude')
ax1.set_xlabel('Frequency [Hz]')
ax1.grid()

plt.axis('tight')
plt.show()
```

https://colab.research.google.com/drive/1h8hS76WAOjFZkf0O_-31BBcj32412xUp?usp=sharing

Magnitude Response





The Optimal Equiripple Method

Optimal Equiripple Method

- Most popular technique for designing FIR filters
- Also called: **Parks-Mclelan**, or **Remez** (Matlab)
- Published in 1972
- Originally written in Fortran
- Widely Used especially in Matlab and Python - scipy.signal.remez( )

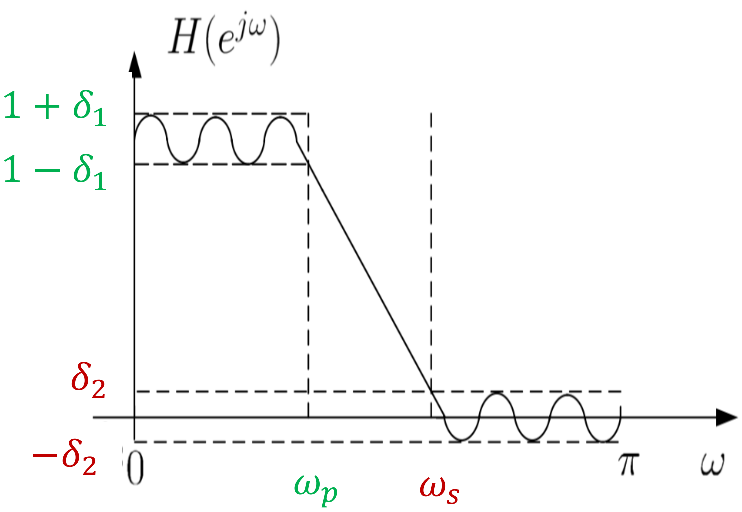
Why use Optimal Equiripple Method?

- Input parameters correspond to filter design specifications
- Creates optimal design for the given parameters
- Can design many categories of FIR filters
  - Simple lowpass / highpass / bandpass / bandstop
  - Multi-band
  - Hilbert transforms
  - Differentiators

Basic Idea of Optimal Equiripple Method

To evenly distribute the ripples in both passband and stopband

- Required filter length will be shorter than that of the window method. Its $H(e^{j\omega})$ exactly meets the passband or stopband ripple specification at one frequency
- Allow $\delta_1 \neq \delta_2$
- Passband ω_p and stopband ω_s frequencies specified
  - Although $\omega_p = \omega_c - \Delta\omega/2$ and $\omega_s = \omega_c + \Delta\omega/2$ implicitly implied in the window method



How does the Optimal Equiripple Method Work?

- Guess the positions of the extrema are evenly spaced in the passband and stopband
- Perform polynomial interpolation and re-estimate positions of the local extrema
- Move extrema to new positions and iterate until the extrema stop shifting. (Remez algorithm)

- The impulse response of optimal equiripple design is determined from:

$$h[n] = \min_{\{\tilde{h}[n]\}} \left(\max_{0 \le \omega \le \omega_p, \omega_s \le \omega \le \pi} \left| E\left(e^{j\omega}\right) \right| \right)$$

where

$$E\left(e^{j\omega}\right) = W\left(e^{j\omega}\right) \left(H_d\left(e^{j\omega}\right) - \widetilde{H}\left(e^{j\omega}\right) \right)$$

- which corresponds to a minmax optimization problem
- $E\left(e^{j\omega}\right)$ is the frequency-domain error between the desired and actual responses weighted by $W\left(e^{j\omega}\right)$
- $W\left(e^{j\omega}\right)$ is the weighting function incorporates all specification parameters, namely, δ_1, δ_2, ω_p and ω_s, into the design process

How does the method operate?

- For example, in lowpass filter design, $W(e^{j\omega})$ has the form of:

$$W(e^{j\omega}) = \begin{cases} \delta_1/\delta_2\,, & 0 \le \omega \le \omega_p \\ 1, & \omega_s < \omega \le \pi \end{cases}$$

- When $\delta_1 > \delta_2$, there is a larger weighting at the stopband. On the other hand, $\delta_1 < \delta_2$ implies a larger weighting at the passband
- To solve for the minmax problem, we can make use of the Parks-McClellan algorithm which requires iterations. We can employ

$$M = \left\lceil \frac{-10\log_{10}(\delta_1\delta_2) - 13}{2.324(\omega_s - \omega_p)} \right\rceil$$

- to get its initial estimate and then compute . If the tolerance specifications are not met, we increment M until the maximum deviations are bounded by δ_1 and δ_2.

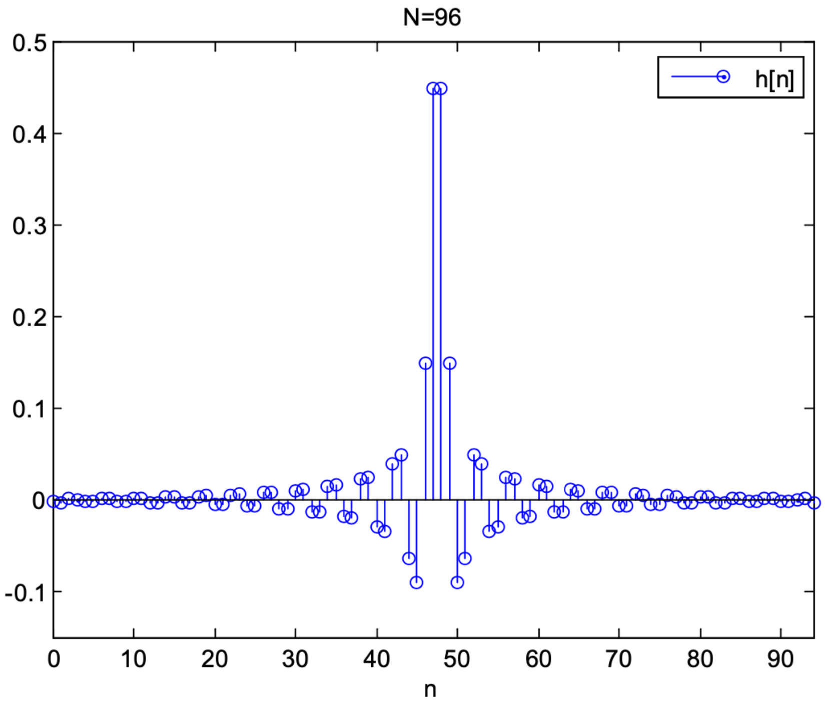
Optimal Equiripple FIR Design Example

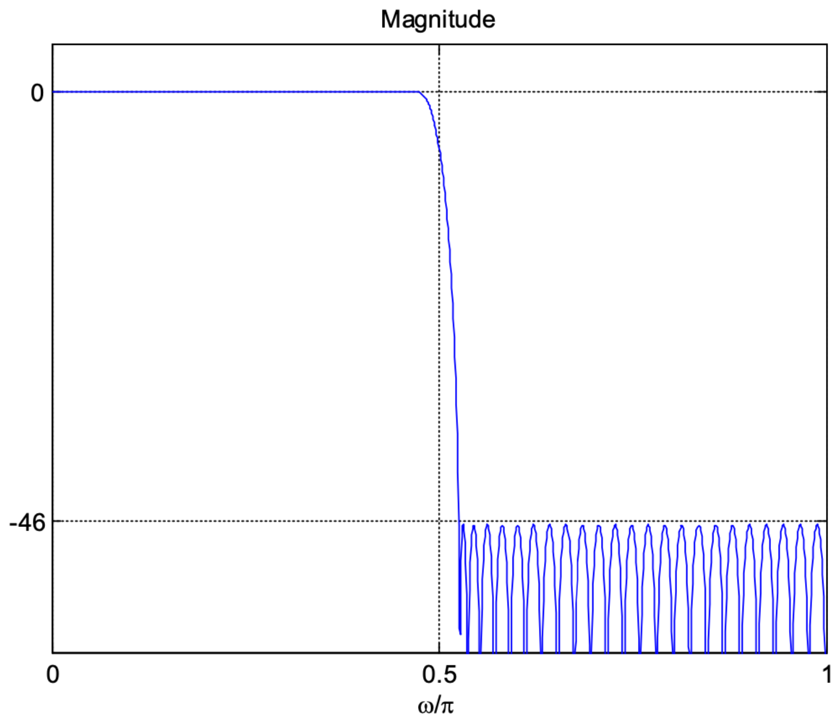
- Use the optimal equiripple method to design a linear-phase and causal FIR filter which approximates an ideal lowpass filter whose passband frequency is $\omega_p = 0.475\pi$ and stopband frequency is $\omega_s = 0.525\pi$.
- The maximum allowable tolerance is $\delta_1 = \delta_2 = 0.005$ in both passband and stopband.
- An initial value of M is computed as

$$M = \left\lceil \frac{-10\log_{10}(0.005 \cdot 0.005) - 13}{2.324(0.525\pi - 0.425\pi)} \right\rceil = 91$$

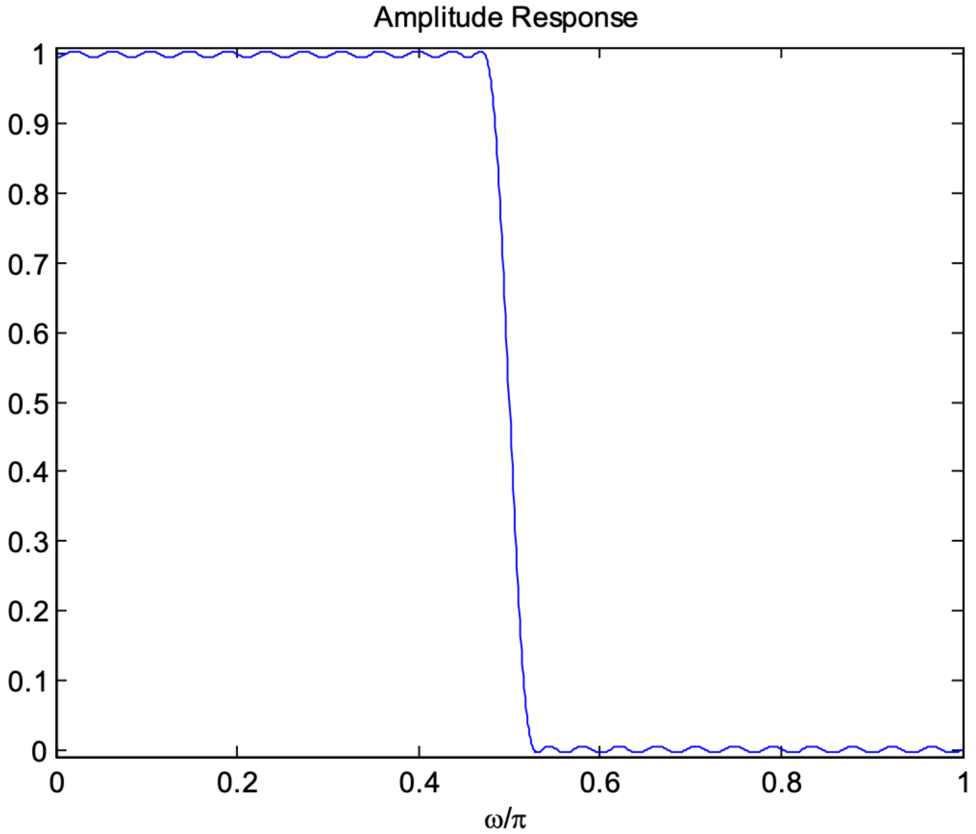
- Starting with $M = 91$ in the Parks-McClellan algorithm, we increment its value until $M = 96$ so that the tolerance specifications are met.

Impulse Response and Magnitude Response of Optimal Equiripple Lowpass Filter

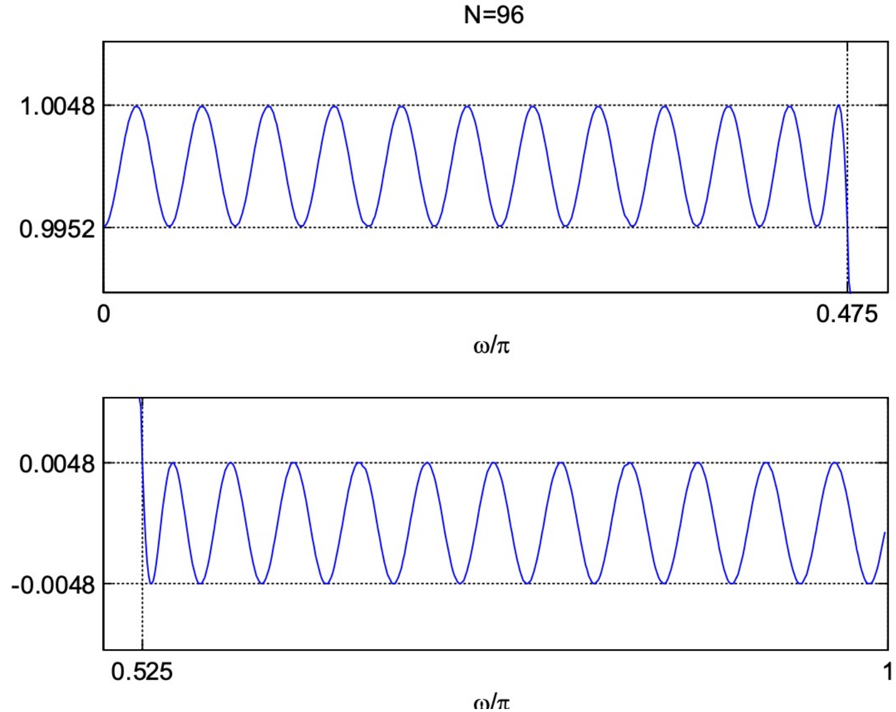




Amplitude response of optimal equiripple lowpass filter



Zoomed amplitude responses



Python Equiripple FIR Design Example

Filter specifications:

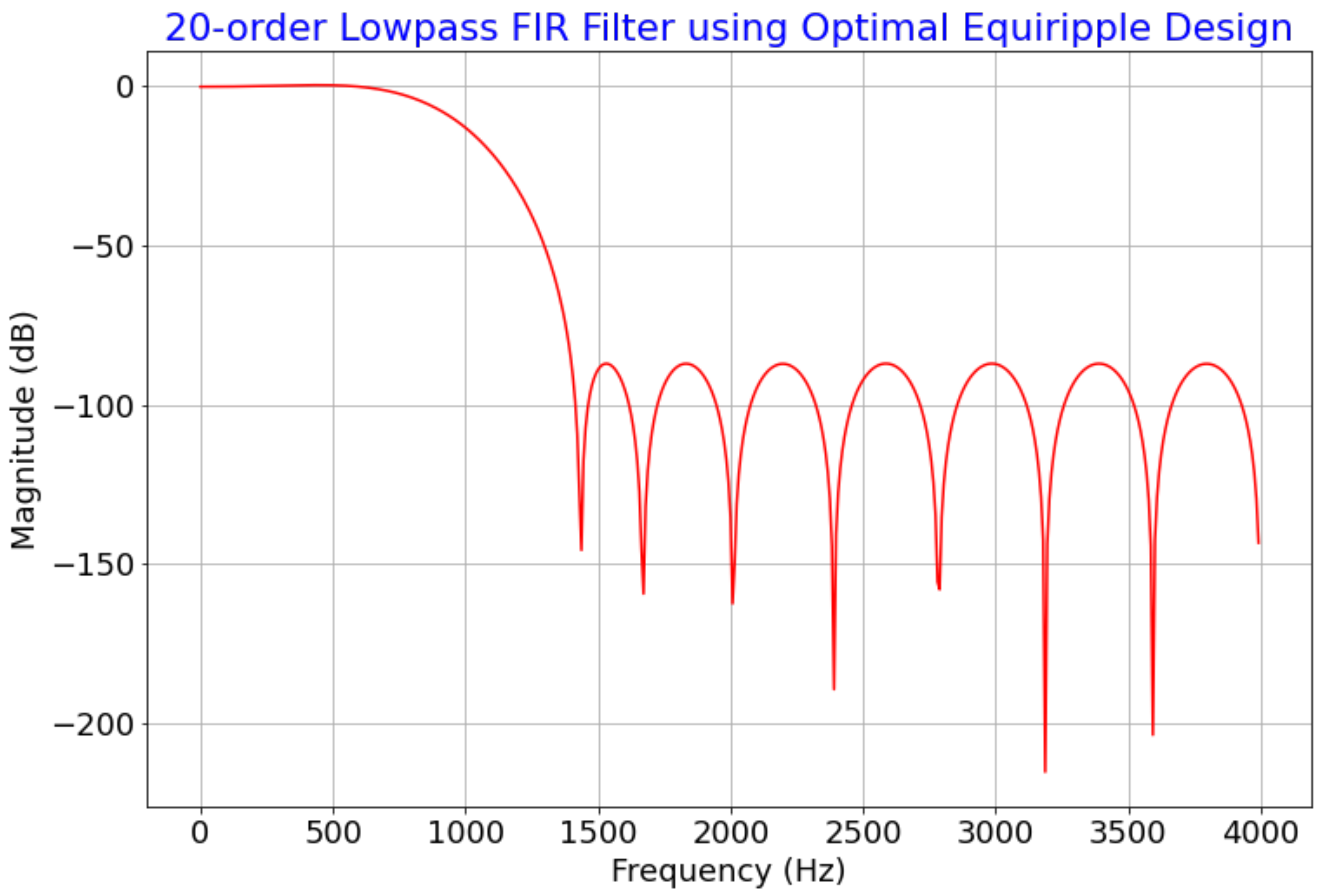
- Sampling Frequency 8kHz : $F_s = 8000$
- Order 20 lowpass filter : $M = 20$
- Cutoff frequency at 1kHz : $f_c = 1000$
- Transition widthband 800Hz : $\Delta f = 800$
    - $f_p = 600$ and $f_s = 1400$

- By specifying the order and the transition width, the forced parameter is the pass-band and stop-band ripple

Python Code : scipy.signal.remez( )

```
from scipy import signal
import numpy as np
import matplotlib.pyplot as plt

Fs = 8000 # Sampling Frequency
fc = 1000/Fs
trans = 800/Fs # This defines the transition bandwidth
f = [0, fc-(trans/2), fc+(trans/2), 0.5 ]
a = [1, 0]
M = 20
lowpass = signal.remez(M, f, a)
freq, response = signal.freqz(lowpass)
magnitude = np.abs(response)
plt.plot(freq/(2*np.pi)*Fs, 20*np.log(magnitude), 'b-') # freq in Hz
plt.title('Lowpass FIR Filter using Optimal Equiripple Design')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude (dB)')
plt.grid()
plt.show()
```

https://colab.research.google.com/drive/1h8hS76WAOjFZkf0O_-31BBcj32412xUp?usp=sharing



To improve the ripple performance, either the order must be increased, or the transition width must be increased

A Multiple Band-Pass Filter Example

Filter specifications:

- Sampling frequency 8 kHz
- Band-pass at:
  - 300 to 400 Hz
  - 600 to 700 Hz
  - 2000 to 3000 Hz
- Transition width 25 Hz
- -40 dB ripple in both the pass-bands and stop-bands

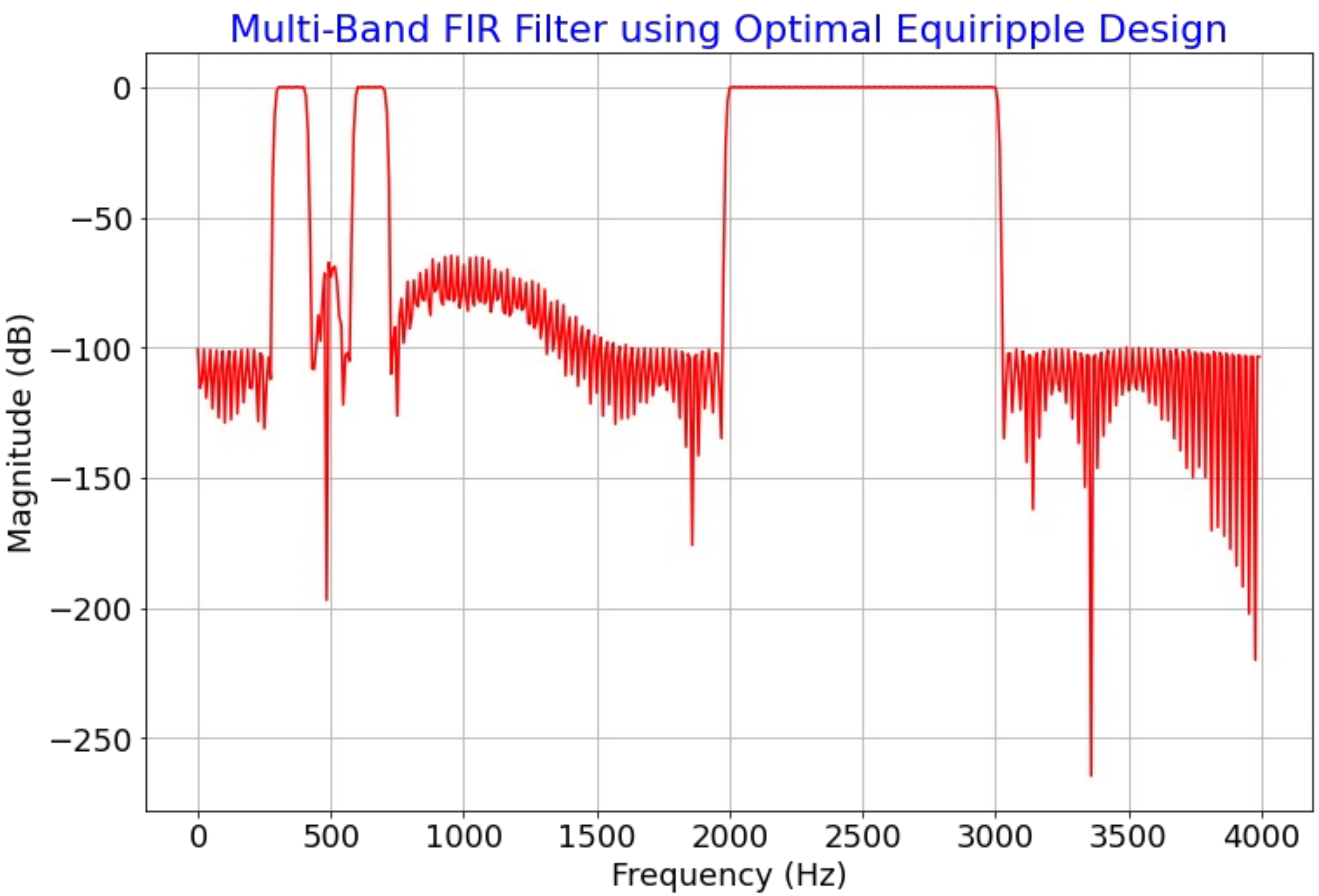
- With this design the forced parameter is the filter order; in this case it was found by experimentation to be 700.

Python Code : Multi-Band FIR Filter Design

```
from scipy import signal
import matplotlib.pyplot as plt

Fs = 8000 # Sampling Frequency
f1 = 300/Fs
f2 = 400/Fs
f3 = 600/Fs
f4 = 700/Fs
f5 = 2000/Fs
f6 = 3000/Fs
trans = 25/Fs # This defines the transition width
f = [0, f1-trans, f1, f2, f2+trans, f3-trans, f3, f4, f4+trans, f5-trans, f5, f6, f6+trans, 0.5 ]
a = [0, 1, 0, 1, 0, 1, 0 ]
M = 700 # This is the filter order which was found by trial and error

mb_pass = signal.remez(M, f, a)

freq, response = signal.freqz(mb_pass)
magnitude = np.abs(response)

plt.plot(freq/(2*np.pi)*Fs, 20*np.log(magnitude), 'r-') # freq in Hz
plt.title('Multi-Band FIR Filter using Optimal Equiripple Design', color='b')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude (dB)')
plt.grid()
plt.show()
```

https://colab.research.google.com/drive/1h8hS76WAOjFZkf0O_-31BBcj32412xUp?usp=sharing

Multiple Band-Pass Results



Stopband Response can be improved by either higher filter order or a relaxed transition specification

Summary of FIR Filter Design

- FIR filters allow the design of **linear-phase filters**, which eliminate the possibility of signal phase distortion.
- Three methods of linear-phase FIR design were discussed:
  - **Window Method**
  - **Frequency Sampling Method**
  - **Optimal Equiripple Method**
    - Or called Parks-McClellan Method