Basic Design Approaches

- (Classic) IIR filter design: **(Chapter 9)**
  1. Convert the digital filter specifications into an analog prototype lowpass filter specifications
  2. Determine the analog lowpass filter transfer function $|H(\Omega)|$
  3. Transform $|H(\Omega)|$ into the desired digital transfer function $H(z)$
     - Analog approximation techniques are highly advanced
     - They usually yield closed-form solutions
     - Extensive tables are available for analog filter design
     - Many applications require digital simulation of analog systems
- FIR filter design is based on a direct approximation of the specified magnitude response, with the often added requirement that the phase be linear (or sometimes, even minimum)
  - The design of an FIR filter of order M may be accomplished by finding either the length-(M+1) impulse response samples of h[n] or the (N+1) samples of its frequency response $H(\omega)$ **(Chapter 10)**

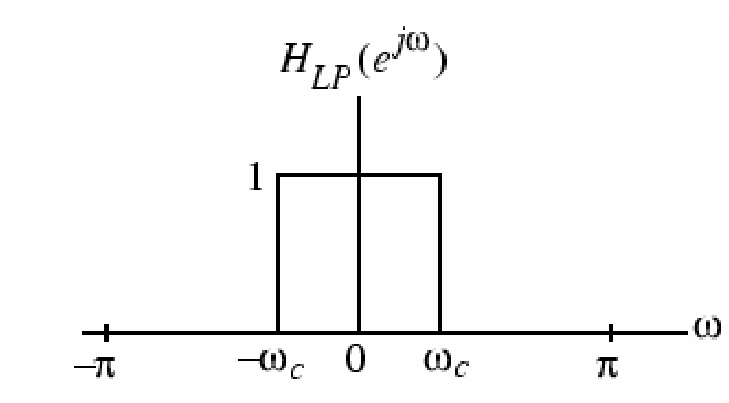
FIR FILTER DESIGN

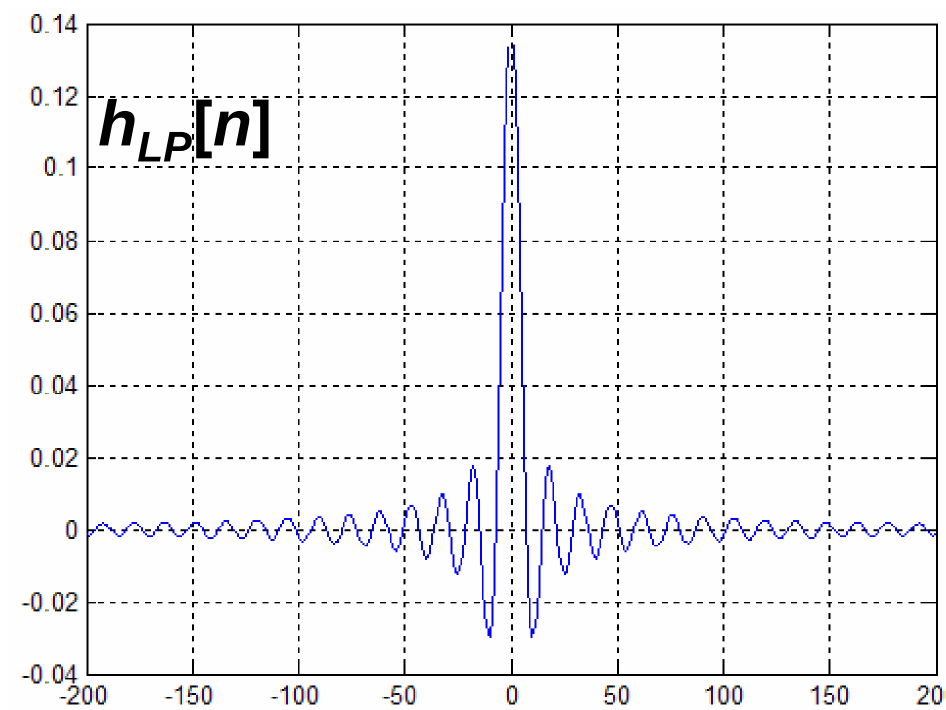
- Let's start with the ideal lowpass filter
  - We know that there are two problems with this filter: infinitely long, and it is non-causal

$$H_{LP}(\omega)=\begin{cases}1, & 0\le|\omega|\le\omega_c\\ 0, & \omega_c\le|\omega|\le\pi\end{cases}$$

$$h_{LP[n]}=\frac{1}{2\pi}\int_{-\omega_c}^{\omega_c}1\cdot e^{j\omega n}d\omega$$

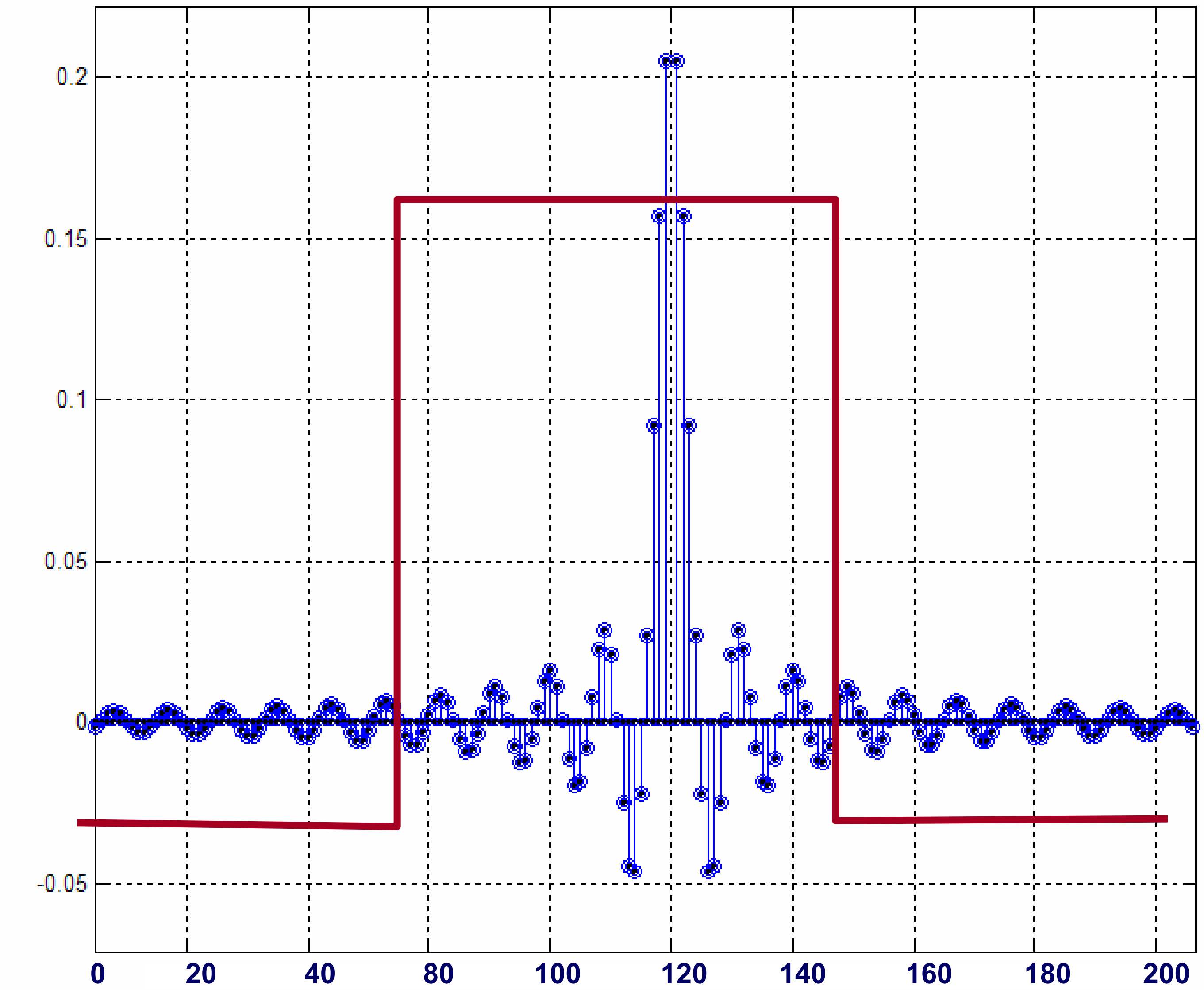
$$=\frac{\sin(\omega_c n)}{\pi n},\ -\infty<n<\infty$$





How can we overcome these two problems?

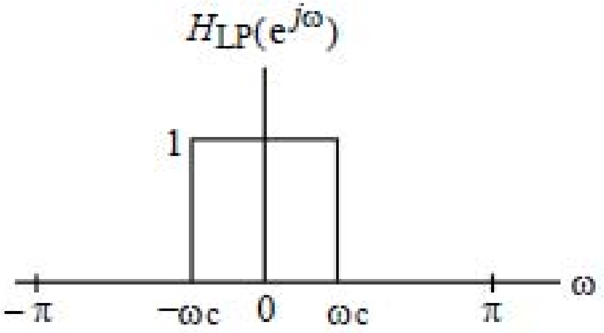
FIR Filter Design



FIR Filter Design

- This is the basic, straightforward approach to FIR filter design:
  - Start with an ideal filter that meets the design criteria, say a filter H(ω)
  - Take the inverse DTFT of this $H(\omega)$ to obtain h[n].
    - This h[n] will be double infinitely long, and non-causal ➔ unrealizable
  - Truncate using a window, say a rectangle, so that M+1 coefficients of h[n] are retained, and all the others are discarded.
    - We now have a finite length (order M) filter, $h_t[n]$, however, it is still non-causal
  - Shift the truncated h[n] to the right (i.e., delay) by M/2 samples, so that the first sample now occurs at n=0.
    - The resulting impulse response, $h_t[n-M/2]$ is a causal, stable, FIR filter, which has an almost identical magnitude response and a phase factor or $e^{-jM/2}$ compared to the original filter, due to delay introduced.

FIR (LOWPASS) FILTER DESIGN



Unrealizable!

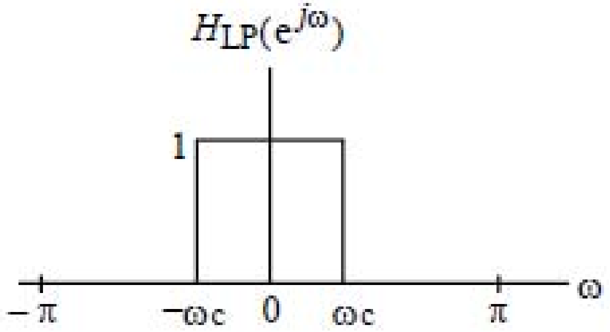
$$h_{LP}[n] = \frac{\sin(\omega_c n)}{\pi n}, \quad -\infty < n < \infty$$

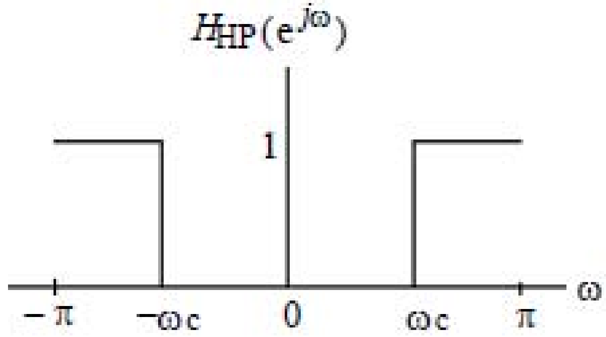
Realizable!

WINDOWING: zero coefficients outside $-M/2 \le n \le M/2$ and a shift to the right yields finite series with length $M+1$

$$h_{LP}[n] = \begin{cases} \dfrac{\sin(\omega_c (n - M/2))}{\pi(n - M/2)}, & 0 \le n \le M, n \ne \dfrac{M}{2} \\ \dfrac{\omega_c}{\pi}, & n = \dfrac{M}{2} \end{cases}$$

FIR Highpass Design

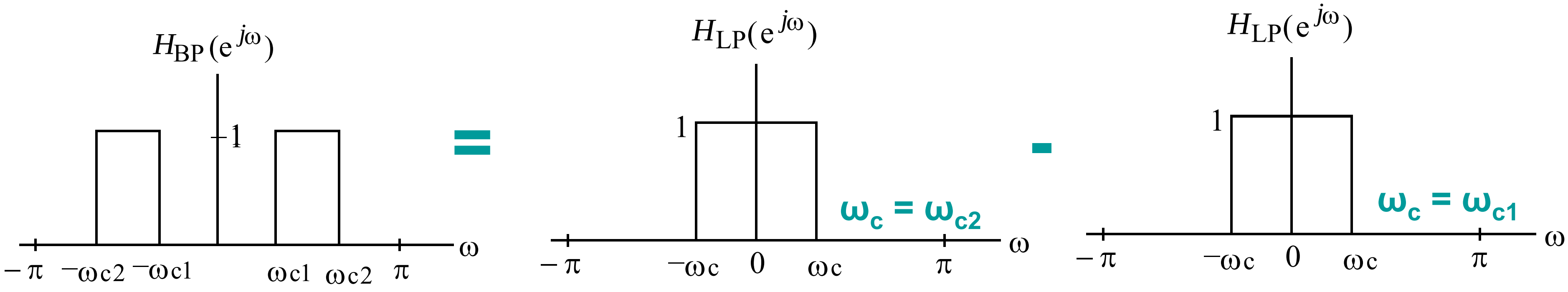




$$H_{HP}(\omega) = 1 - H_{LP}(\omega) \quad \Longleftrightarrow \quad h_{HP}[n] = \delta[n] - h_{LP}[n]$$

$$h_{HP}[n] = \begin{cases} -\dfrac{\sin(\omega_c n)}{\pi n}, & |n| > 0 \\ 1\text{-}\ \dfrac{\omega_c}{\pi}, & n = 0 \end{cases}$$

FIR BPF / BSF Design



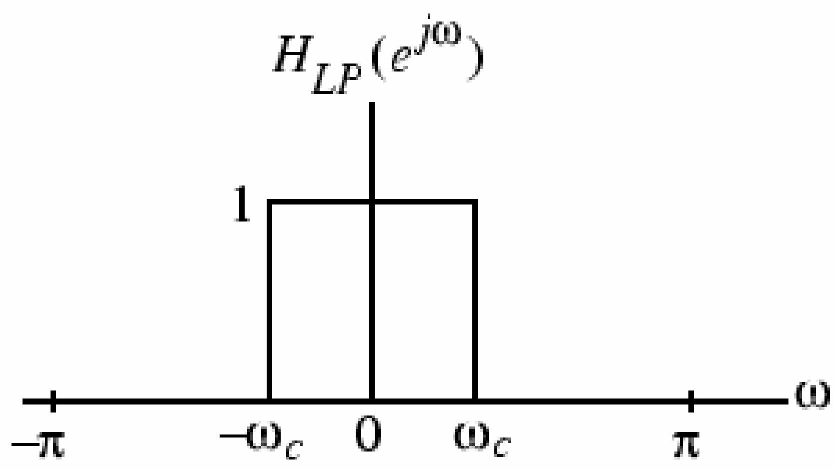
$$H_{BP}(\omega) = H_{LP}(\omega)\Big|_{\omega_c=\omega_{c2}} - H_{LP}(\omega)\Big|_{\omega_c=\omega_{c1}} \iff h_{BP}[n] = h_{LP}[n]\Big|_{\omega_c=\omega_{c2}} - h_{LP}[n]\Big|_{\omega_c=\omega_{c1}}$$

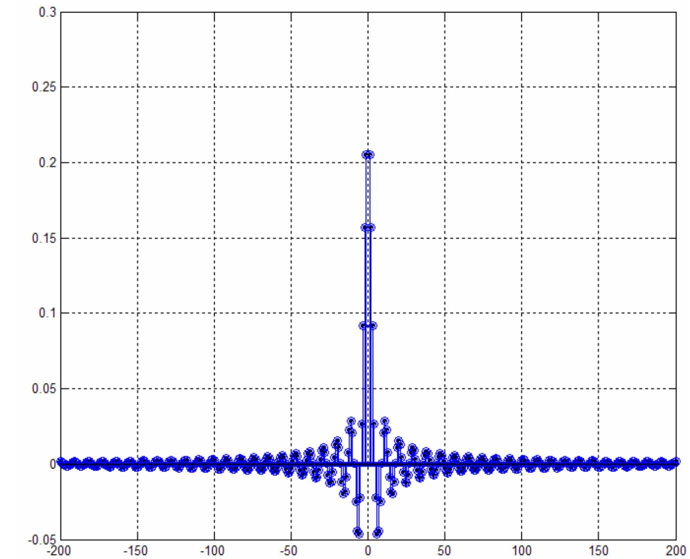
$$h_{BP[n]} = \begin{cases} \dfrac{\sin\left(\omega_{c_2}(n-M/2)\right)}{\pi(n-M/2)} - \dfrac{\sin\left(\omega_{c_1}(n-M/2)\right)}{\pi(n-M/2)}, & 0 < n < M,\ n \neq \dfrac{M}{2} \\ \dfrac{\omega_{c_2}}{\pi} - \dfrac{\omega_{c_1}}{\pi}, & n = \dfrac{M}{2} \end{cases}$$

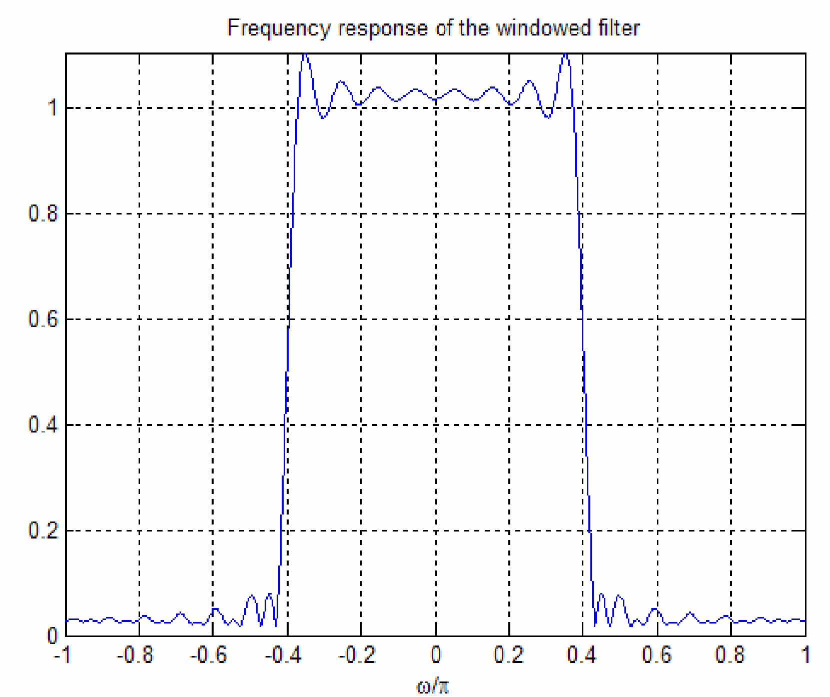
Similarly,

$$H_{BS}(\omega) = 1 - H_{BP}(\omega) \iff h_{BS}[n] = \delta[n] - h_{BP}[n]$$

HOWEVER…

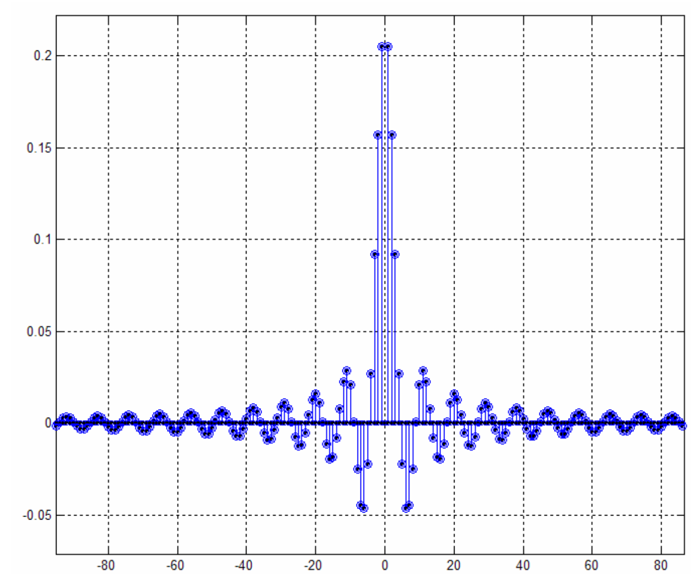






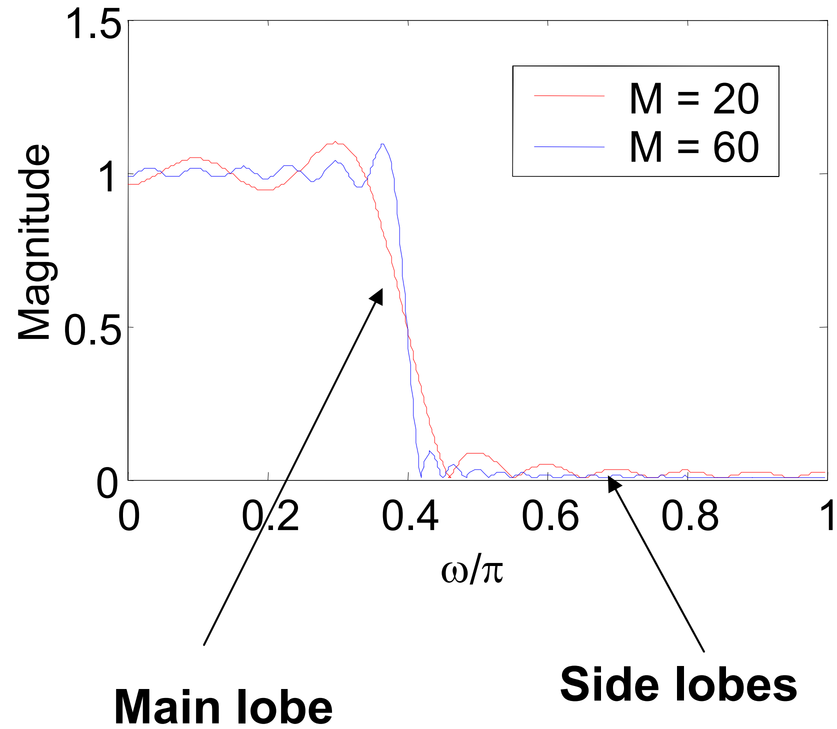
?

What happened…?



Gibbs Phenomenon

- Truncating the impulse response of an ideal filter to obtain a realizable filter, creates oscillatory behavior in the frequency domain.
  - The Gibbs Phenomenon
- We observe the following:
  - As M↑, the number of ripples ↑ however, ripple widths ↓
  - The height of the largest ripples remain constant, regardless of the filter length
  - As M↑, the height of all other ripples ↓
  - The main lobe gets narrower as M↑, that is, the drop-off becomes sharper
  - Similar oscillatory behavior can be seen in all types of truncated filters



Gibbs Phenomenon

- Why is this happening?
- The Gibbs phenomenon is simply an artifact of the windowing operation.
  - Multiplying the ideal filter's impulse response with a rectangular window function is equivalent to convolving the underlying frequency response with a sinc

$$h_t[n] = h_d[n] \cdot w[n] \Leftrightarrow H_t(\omega) = H_d(\omega) * W(\omega)$$

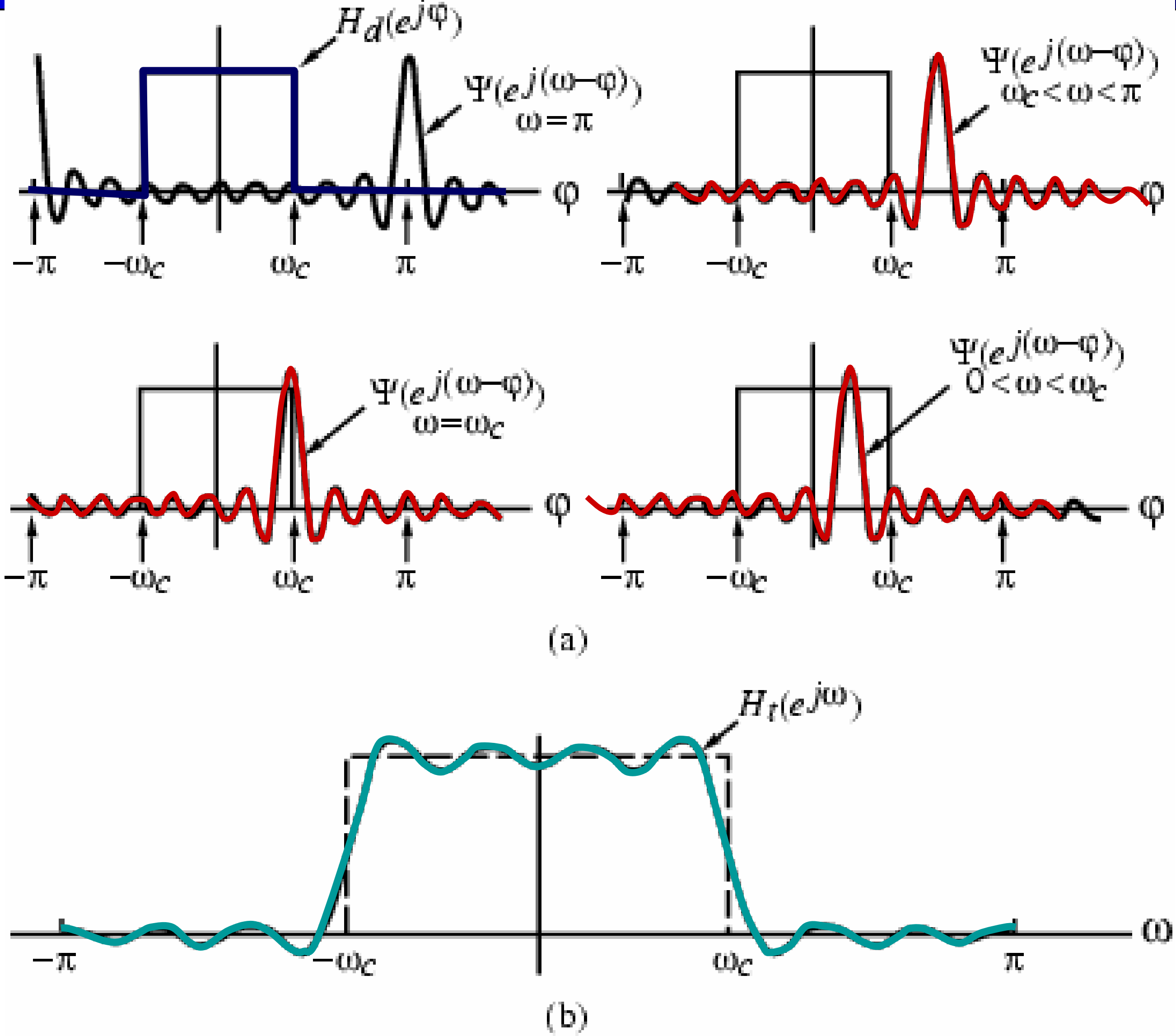
Truncated filter impulse response ($h_t[n]$)

Desired filter impulse response ($h_d[n]$)

Windowing function ($w[n]$)

  - However, we want $\boldsymbol{H_t(\omega)}$ to be as close as possible to $\boldsymbol{H_d(\omega)}$, which can only be possible if the $\boldsymbol{W(\omega)=\delta(\omega)} \leftrightarrow \boldsymbol{w[n]}=1$, an infinite window.
    - We have conflicting requirements: On one hand, we want a narrow window, so that we have a short filter; on the other hand, we want the truncated filter as closely match as possible ideal filter's frequency response, which in turn requires an infinitely long window!
  - This convolution results in the oscillations, particularly dominant at the edges.

Gibbs Phenomenon



H_d: Ideal filter frequency response

Ψ: Rectangular window frequency response

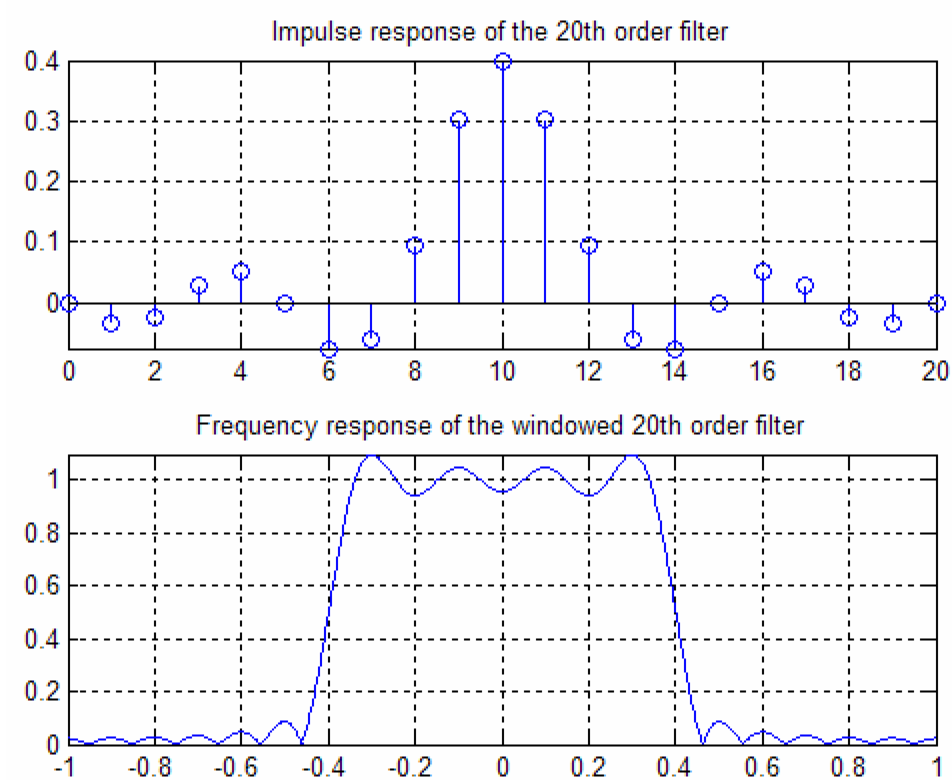
H_t: Truncated filter's frequency response

Window_LPF.m

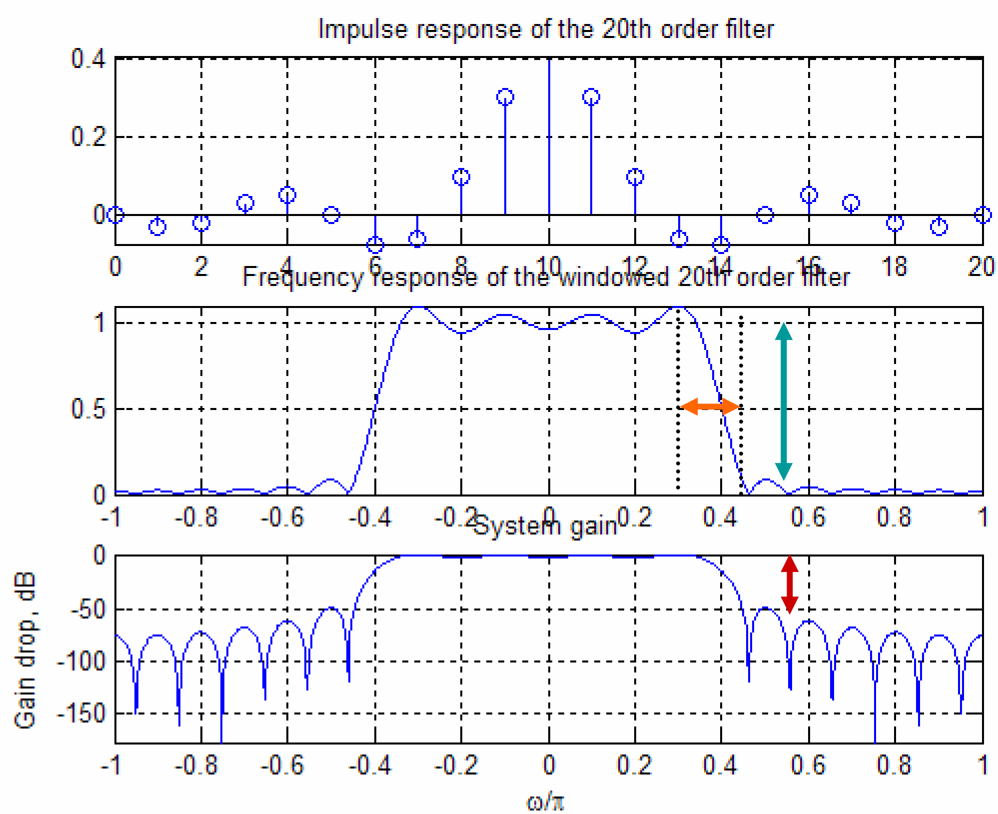
GIBBS PHENOMENON DEMO

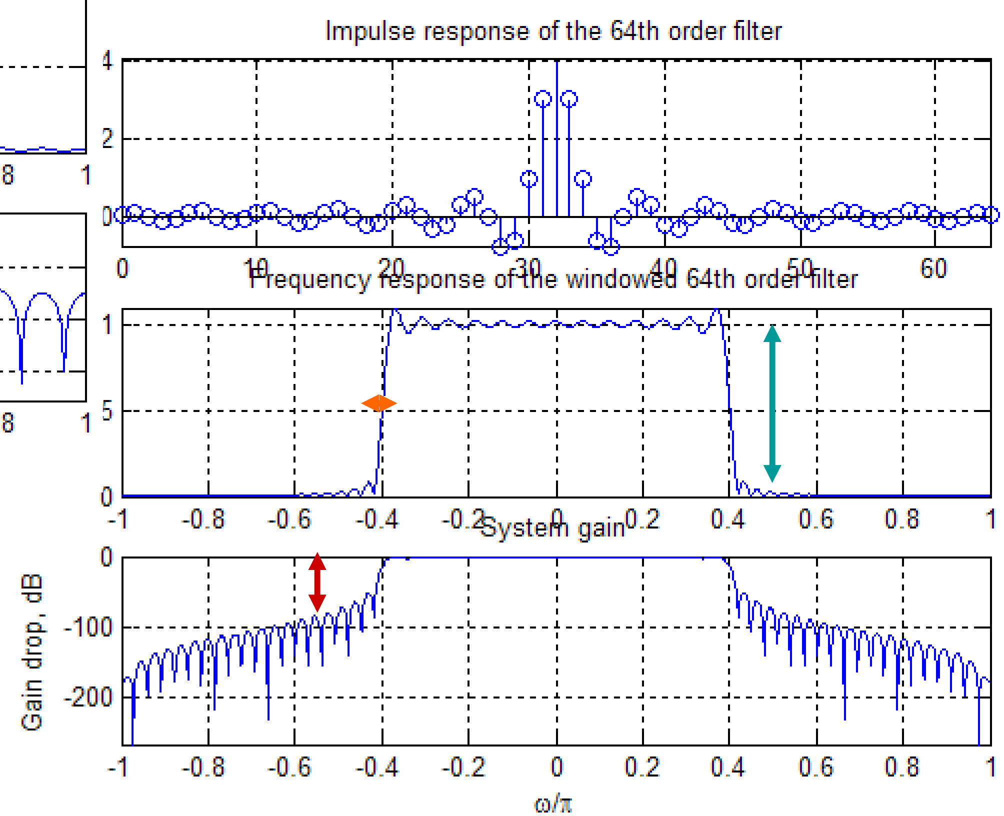
$$h_{LP}[n] = \begin{cases} \dfrac{\sin\left(\omega_C \left(n - M/2\right)\right)}{\pi\left(n - M/2\right)}, & 0 \le n \le M,\ n \ne \dfrac{M}{2} \\ \dfrac{\omega_C}{\pi}, & n = \dfrac{M}{2} \end{cases}$$

```
% FIR Lowpass filter

M=input('Enter the order of filter:');
wc=input('Enter the cutoff frequency in terms of radians:');

n=0:M;
h_LP=sin(wc*(n-M/2))./(pi*(n-M/2));
h_LP(ceil(M/2+1))=wc/pi;

subplot(211)
stem(n, h_LP)
axis([0 M min(h_LP) max(h_LP)])
title(['Impulse response of the ',num2str(M),…
'th order filter']);
subplot(212)
H_LP=fft(h_LP, 1024);
w=linspace(-pi, pi, 1024);
plot(w/pi, abs(fftshift(H_LP)))
title(['Frequency response of the windowed ', …
num2str(M), 'th order filter']);
grid
axis([-1 1 0 max(abs(H_LP))])
```



Effect of Filter Length





Transition band / main lobe width

Stopband attenuation

In dB

- Here's what we want:
  - Quick drop off → Narrow transition band
    - Narrow main lobe
    - Increased stopband attenuation

    Conflicting requirements
  - Reduce the height of the side-lobe which causes the ripples
  - Reduce Gibb's phenomenon (ringing effects, all ripples)
  - Minimize the order of the filter.
- Gibb's phenomenon can be reduced (but not eliminated) by using a ***smoother window*** that gently tapers off to zero, rather than the brick wall behavior of the rectangular filter.
  - Several window functions are available, which usually trade-off main-lobe width and stopband attenuation.
    - Rectangular window has the narrowest main-lobe width, but poor side-lobe attenuation.
    - Tapered window causes the height of the sidelobes to diminish, with a corresponding increase in the main lobe width resulting in a wider transition at the cutoff frequency.

Commonly Used Windows

1. Rectangular

$$w(n) = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

2. Bartlett

$$w(n) = 1 - \frac{2\left|n - \frac{M}{2}\right|}{M}$$

3. Blackman

$$w(n) = 0.42 - 0.5 \cos \frac{2\pi n}{M} + 0.08 \cos \frac{4\pi n}{M}$$

4. Hamming

$$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{M}$$

5. Hann

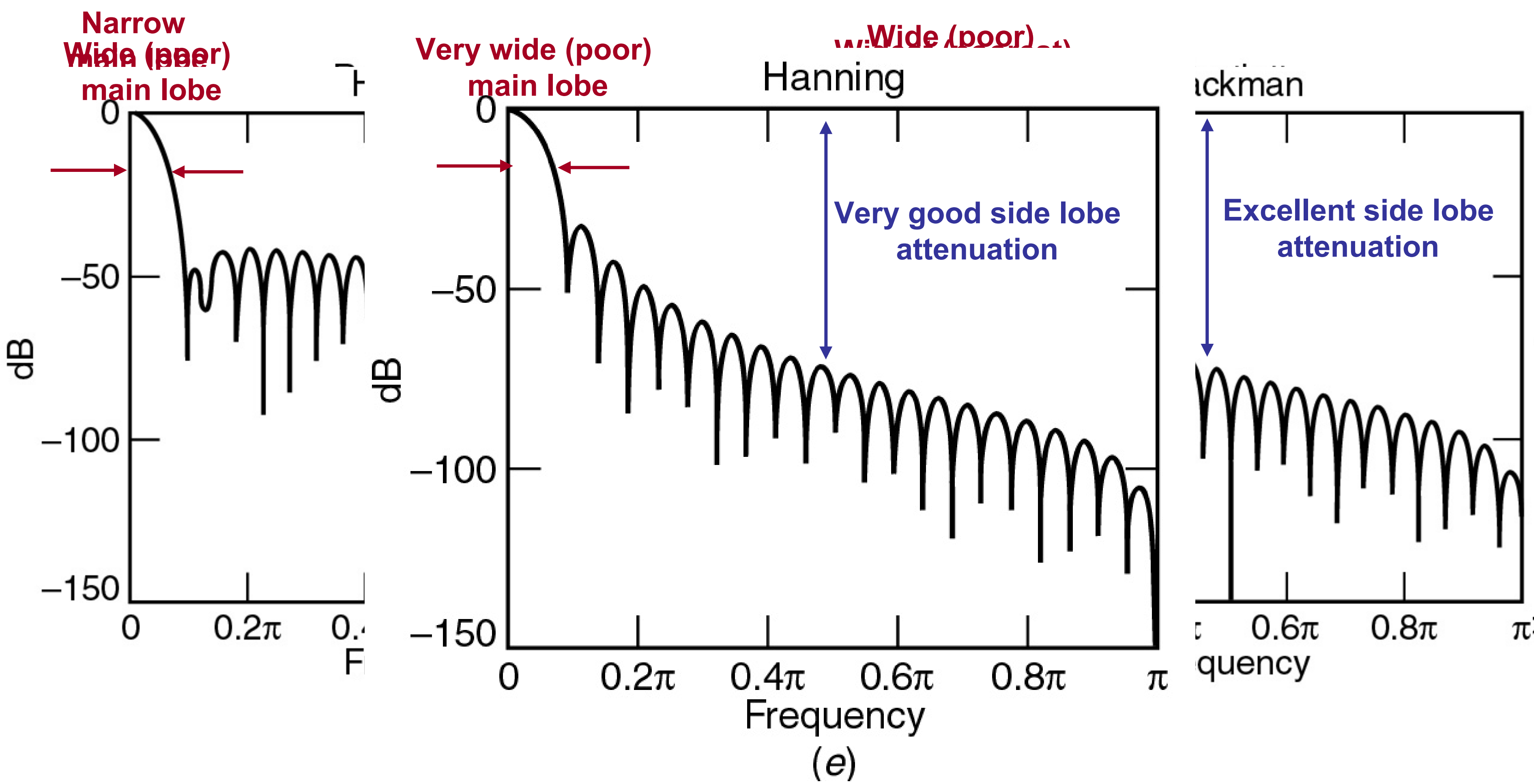
$$w(n) = 0.5 - 0.5 \cos \frac{2\pi n}{M}$$

6. Kaiser

$$I_0 \left\{ \beta \sqrt{1 - \left(\frac{n - M/2}{M/2} \right)^2} \right\}$$

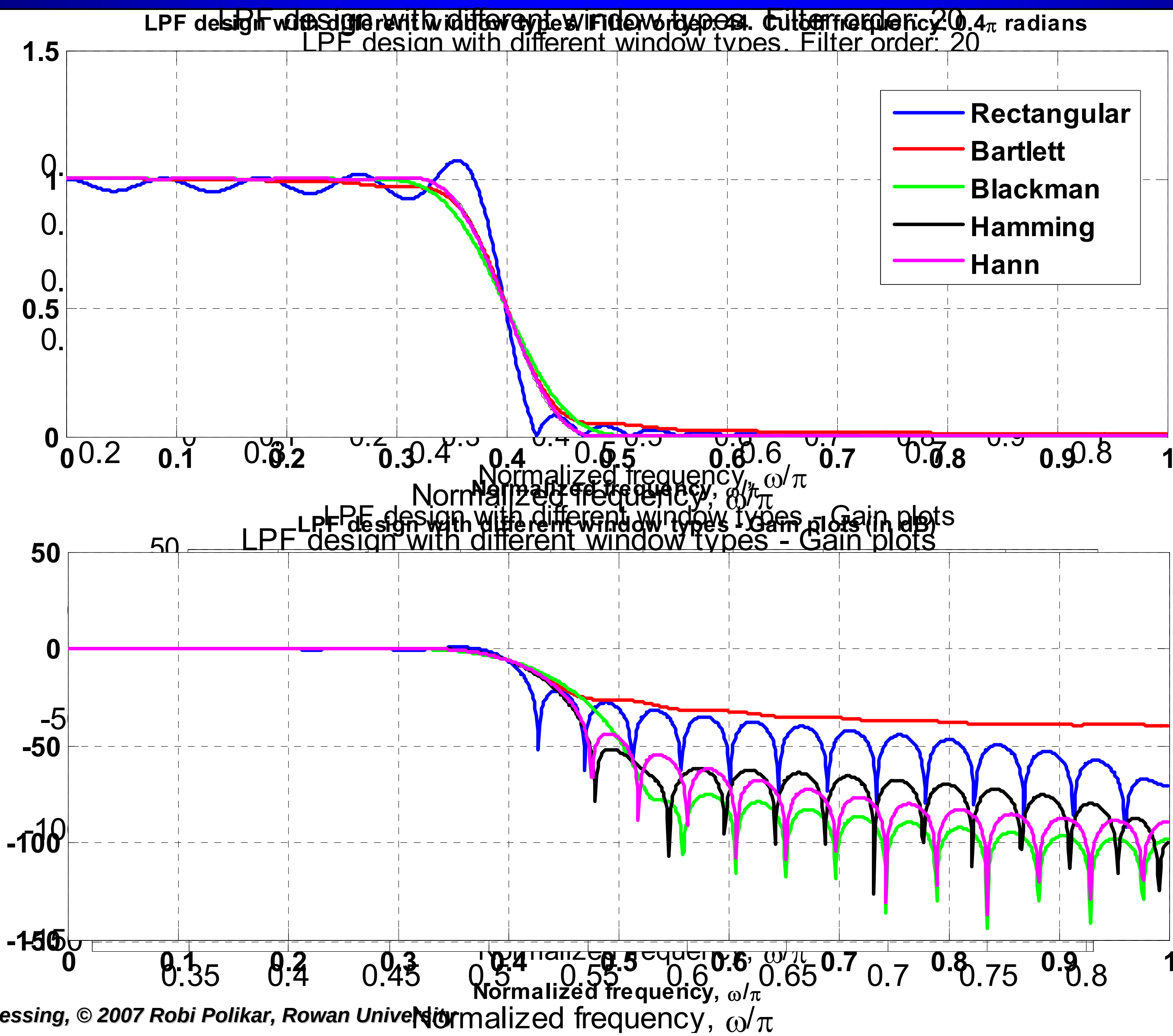
$0<n<M$
(length M+1)
In your text:
$-M<n<M$
(length 2M+1)

Commonly Used Windows



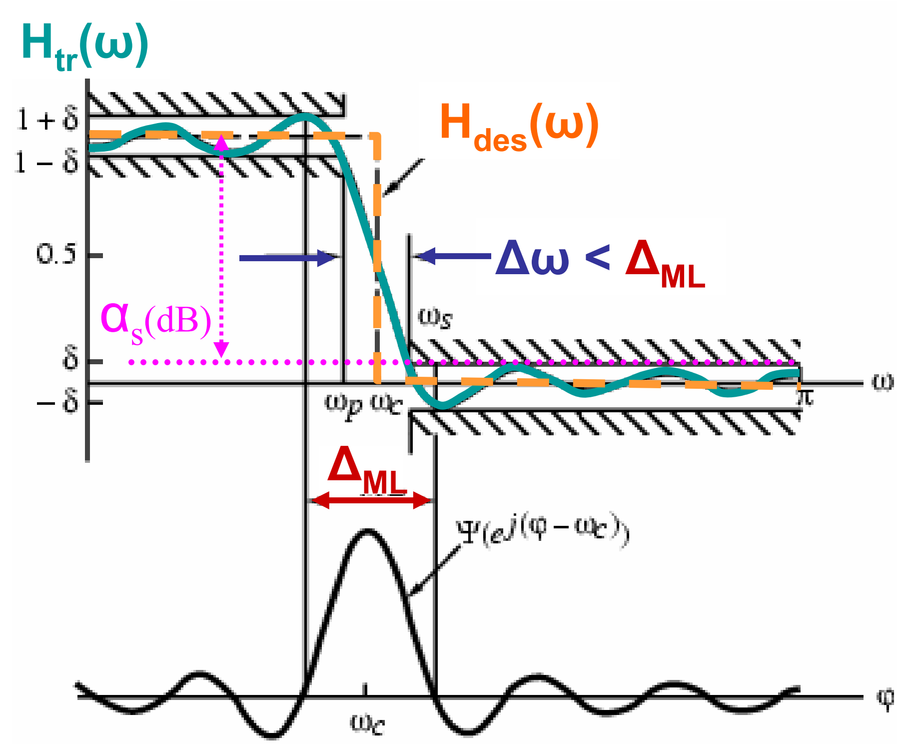
Compare_windows.m

Comparing Windows



FIXED WINDOW FUNCTIONS

- All windows shown so far are ***fixed window*** functions
  - Magnitude spectrum of each window characterized by a main lobe centered at $\omega = 0$ followed by a series of sidelobes with decreasing amplitudes
  - Parameters predicting the performance of a window in filter design are:
    - Main lobe width (Δ_{ML}) : the distance b/w nearest zero-crossings on both sides or transition bandwidth ($\Delta\omega = \omega_s - \omega_p$)
    - Relative sidelobe level (A_{sl}): difference in dB between the amp. of the largest sidelobe and the main lobe (or sidelobe attenuation (α_s))
  - For a given window, both parameters all completely determined once the filter order ***M*** is set.



Fixed Window Functions

Windows	Mainlobe width	Sidelobe attenuation (dB)	Min. stopband Attenuation	Transition Bandwidth Δω
Rectangular	$4\pi/M$	−13	**20.9**	**0.92π/(M/2)**
Bartlett	$8\pi/M$	−27	**See book**	**See book**
Harning	$8\pi/M$	−32	**43.9**	**3.11 π/(M/2)**
Hamming	$8\pi/M$	−43	**54.5**	**3.32 π/(M/2)**
Blackman	$12\pi/M$	−58	**75.3**	**5.56 π/(M/2)**

- How to design:
  - Set $\omega_c = (\omega_p + \omega_s)/2$
  - Choose window type based on the specified sidelobe attenuation (A_{sl}) or minimum stopband attenuation (α_s)
  - Choose ***M*** according to the transition band width (Δω) and/or mainlobe width (Δ_{ML}). Note that this is the only parameter that can be adjusted for fixed window functions. Once a window type and M is selected, so are A_{sl}, α_s , and Δ_{ML}
    - Ripple amplitudes cannot be custom designed.
  - Adjustable windows have a parameter that can be varied to trade-off between main-lobe width and side-lobe attenuation.

Kaiser Window

- The most popular adjustable window

$$w[n] = \frac{I_0\left\{\beta\sqrt{1-\left(\frac{n-M/2}{M/2}\right)^2}\right\}}{I_0(\beta)}, \quad 0 \le n \le M$$

where β is an adjustable parameter to trade-off between the main lobe width and sidelobe attenuation, and $I_0\{x\}$ is the modified zeroth-order Bessel function of the first kind:

$$I_0(x) = 1 + \sum_{k=1}^{\infty}\left[\frac{(x/2)^k}{k!}\right]^2$$

In practice, this infinite series can be computed for a finite number of terms for a desired accuracy. In general, 20 terms is adequate.

$$I_0(x) \cong 1 + \sum_{k=1}^{20}\left[\frac{(x/2)^k}{k!}\right]^2$$

FIR Design Using Kaiser Window

- Given the following:
  - ω_p - passband edge frequency and ω_s - stopband edge frequency
  - δ_p - peak ripple value in the passband and δ_s - peak ripple value in the stopband
- Calculate:

1. Minimum ripple in dB: $\alpha_s = -20\log_{10}(\delta_s)\ or\ -20\log_{10}\left(\min\{\delta_s, \delta_p\}\right)$
2. Normalized transition bandwidth: $\Delta\omega = \omega_s - \omega_p$
3. Window parameters:

$$\beta = \begin{cases} 0.1102(\alpha_s - 8.7), & \alpha_s > 50\,\text{dB} \\ 0.5842(\alpha_s - 21)^{0.4} + 0.07886(\alpha_s - 21), & 21 \le \alpha_s \le 50\,\text{dB} \\ 0, & \alpha_s \le 21\,\text{dB} \end{cases}$$

4. Filter length, M+1:

$$M + 1 = \begin{cases} \dfrac{\alpha_s - 7.95}{2.285\Delta\omega} + 1, & \alpha_s > 21 \\ \dfrac{5.79}{\Delta\omega}, & \alpha_s < 21 \end{cases}$$

Design specs for Kaiser window in your book is different. This one, while may seem more complicated is actually easier to follow.

5. Determine the corresponding Kaiser window
6. Obtain the filter by multipling the ideal filter $h_I[n]$ with w[n]

$$w[n] = \frac{I_0\left\{\beta\sqrt{1 - \left(\frac{n - M/2}{M/2}\right)^2}\right\}}{I_0(\beta)}, \quad 0 \le n \le M$$

Example

➲ Design an FIR filter with the following characteristics:

- $\omega_p = 0.3\pi$ $\omega_s = 0.5\pi$, $\delta_s = \delta_p = 0.01$ ➔ $\alpha = 40\text{dB}$, $\Delta\omega = 0.2\pi$

$$\beta = 0.5842(\alpha_s - 21)^{0.4} + 0.07886(\alpha_s - 21), \; 21 \le \alpha_s < 50$$

$$\beta = 0.5842(19)^{0.4} + 0.07886 \times 19 = 3.3953$$

$$M + 1 = \frac{32.05}{2.285(0.2\pi)} + 1 = 23.2886 \approx 24$$

$$h_{LP[n]} = \begin{cases} \dfrac{\sin(\omega_c(n - M/2))}{\pi(n - M/2)}, & 0 < n < M, \; n \ne \dfrac{M}{2} \\ \dfrac{\omega_c}{\pi}, & n = \dfrac{M}{2} \end{cases} \Rightarrow h_{LP[n]} = \begin{cases} \dfrac{\sin(0.4\pi(n-12))}{\pi(n-12)}, & 0 < n < 23, \; n \ne 12 \\ 0.4, & n = 12 \end{cases}$$

$$h_t[n] = h_{LP}[n] \cdot w[n], \quad -12 \le n \le 12$$

Complete Cycle for FIR Filter Design using Windows

- Depending on your specs, determine what kind of window you would like to use.
  - For all window types, except Kaiser, once you choose the window, the only other parameter to choose is filter length M.
    - For Kaiser window, determine M and beta, based on the specs using the given expressions.
- Compute the window coefficients w[n] for the chosen window.
- Compute filter coefficients (taps)
  - Determine the ideal impulse response $h_I[n]$ from the given equations for the type of magnitude response you need (lowpass, highpass, etc.)
  - Multiply window and ideal filter coefficients to obtain the realizable filter coefficients (also called ***taps*** or ***weights***): $h[n]=h_I[n].w[n]$
- Convolve your signal with the new filter coefficients $y[n]=x[n]*h[n]$.

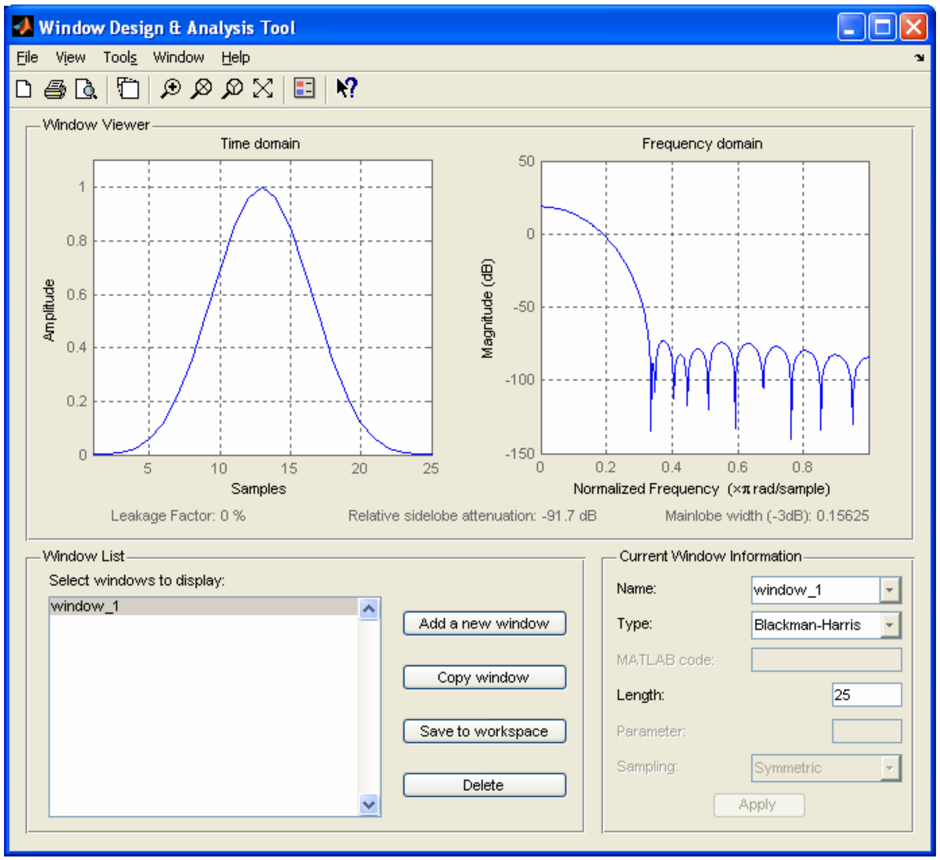
Demo: FIR_window.m

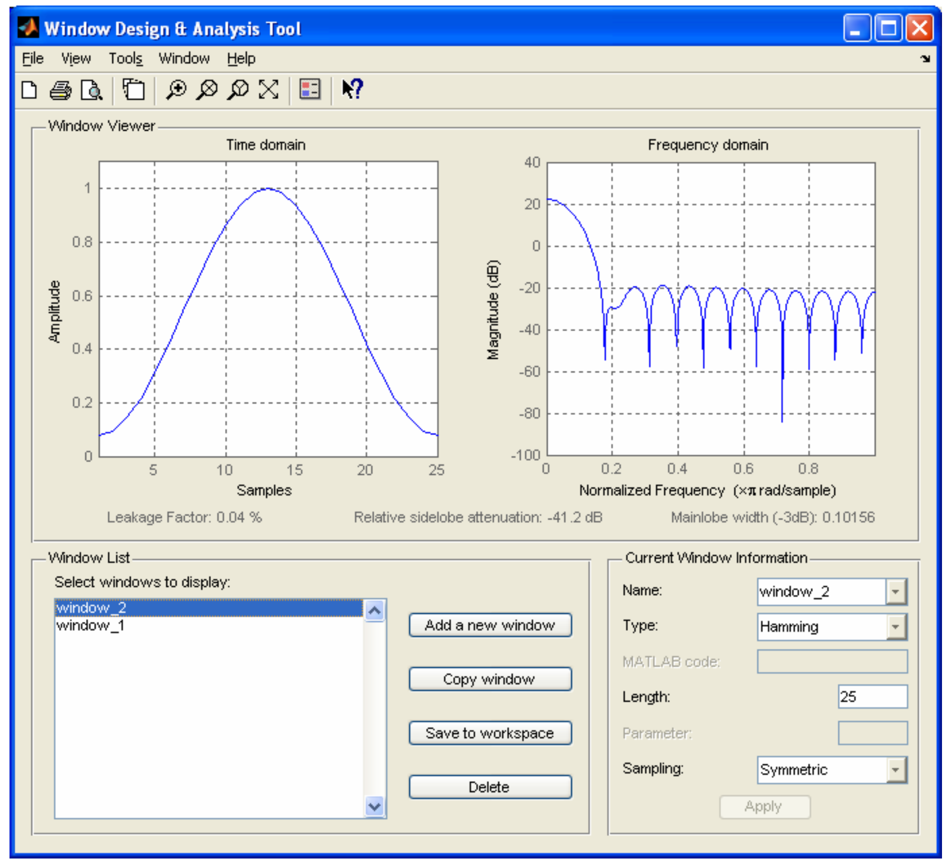
IN MATLAB

- The following functions create *N*-point windows for the corresponding functions:
  - `rectwin(N)`
  - `hamming(N)`
  - `hanning(N)`
  - `bartlett(N)`
  - `kaiser(N, beta)`
  - `blackman(N)`
- Try this: h=hamming(40); [H w]=freqz(h,1, 1024); plot(w, abs(H))
  - Compare for various windows. Also plot gain in dB
- The function **`window`** ***window design and analysis tool*** provides a GUI to custom design several window functions.
- The function **`fdatool`** ***filter design and analysis tool*** provides a GUI to custom design several types of filters from the given specs.
- The function **`sptool`** ***signal processing tool***, provides a GUI to custom design, view and apply to custom created signals. It also provides a GUI for spectral analysis.

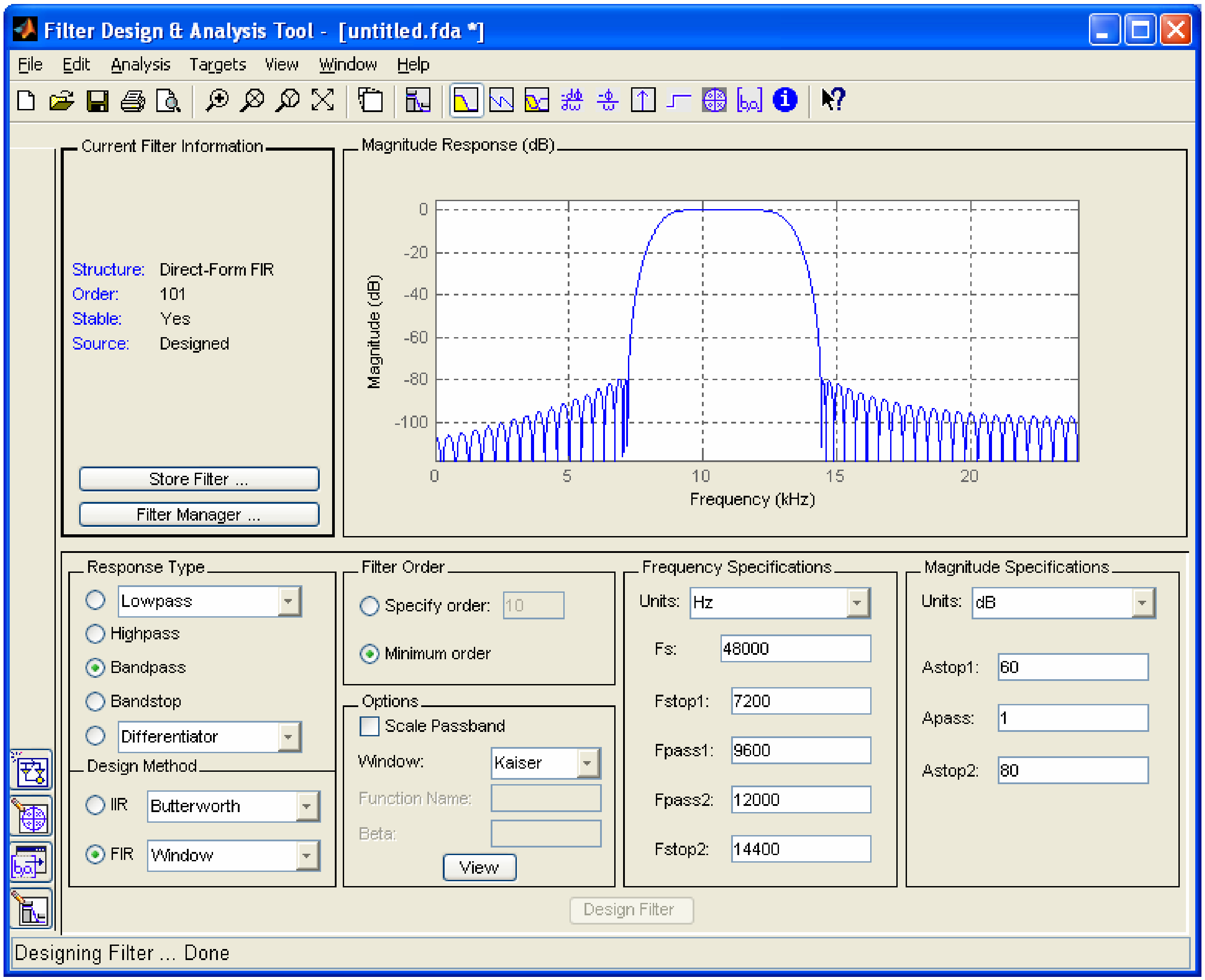
Window Design & Analysis Tool

>> window





Filter Design & Analysis Tool



In Matlab: fir1

fir1 FIR filter design using the window method.

b = fir1(N,Wn) designs an N'th order lowpass FIR digital filter and returns the filter coefficients in length N+1 vector B. The cut-off frequency Wn must be between 0 < Wn < 1.0, with 1.0 corresponding to half the sample rate. The filter B is real and has linear phase. The normalized gain of the filter at Wn is -6 dB.

b = fir1(N,Wn,'high') designs an N'th order highpass filter. You can also use B = fir1(N,Wn,'low') to design a lowpass filter.

If Wn is a two-element vector, Wn = [W1 W2], FIR1 returns an order N bandpass filter with passband W1 < W < W2. You can also specify **b = fir1(N,Wn,'bandpass')**. If Wn = [W1 W2], **b = fir1(N,Wn,'stop')** will design a bandstop filter.

If Wn is a multi-element vector, Wn = [W1 W2 W3 W4 W5 ... WN], FIR1 returns an order N multiband filter with bands 0 < W < W1, W1 < W < W2, ..., WN < W < 1.

b = fir1(N,Wn,'DC-1') makes the first band a passband.

b = fir1(N,Wn,'DC-0') makes the first band a stopband.

b = fir1(N,Wn,WIN) designs an N-th order FIR filter using the N+1 length vector WIN to window the impulse response. *If empty or omitted, FIR1 uses a Hamming window of length N+1.* For a complete list of available windows, see the help for the WINDOW function. If using a Kaiser window, use the following

b = fir1(N,Wn,kaiser(n+1,beta))