BINGHAMTON UNIVERSITY
State University of New York

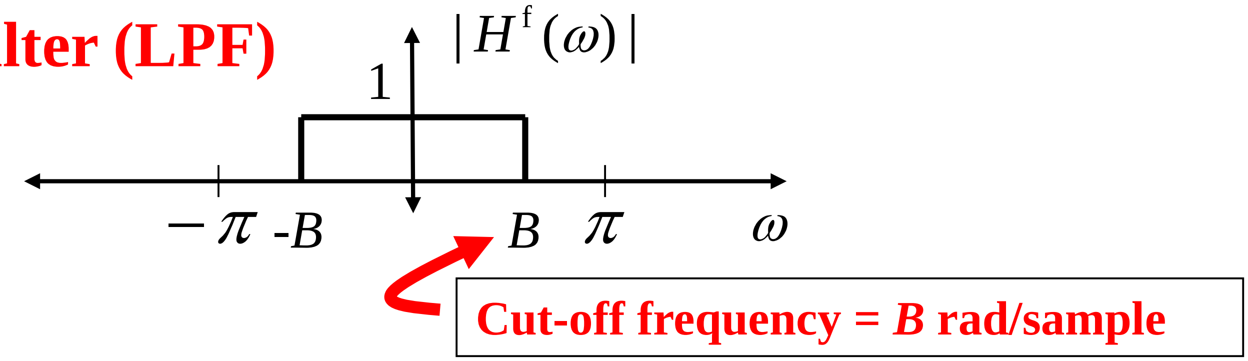
EEO 401
Digital Signal Processing

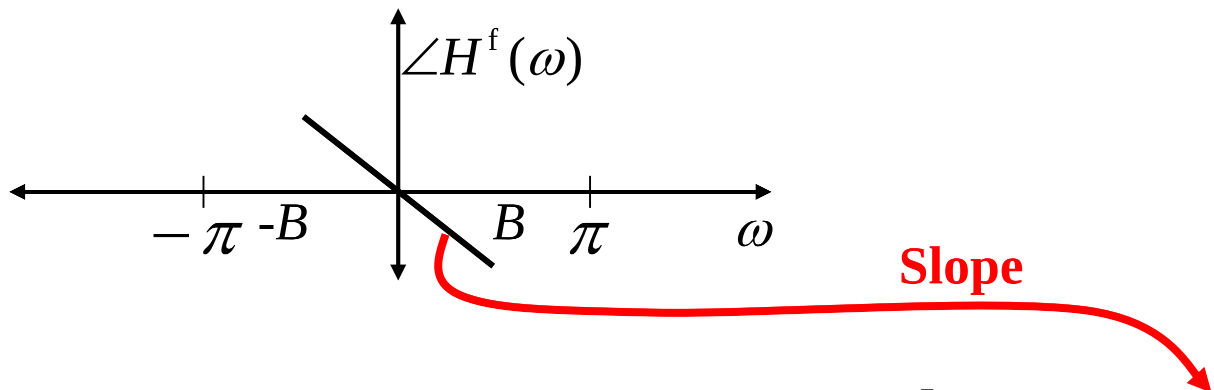
Prof. Mark Fowler

Note Set #29

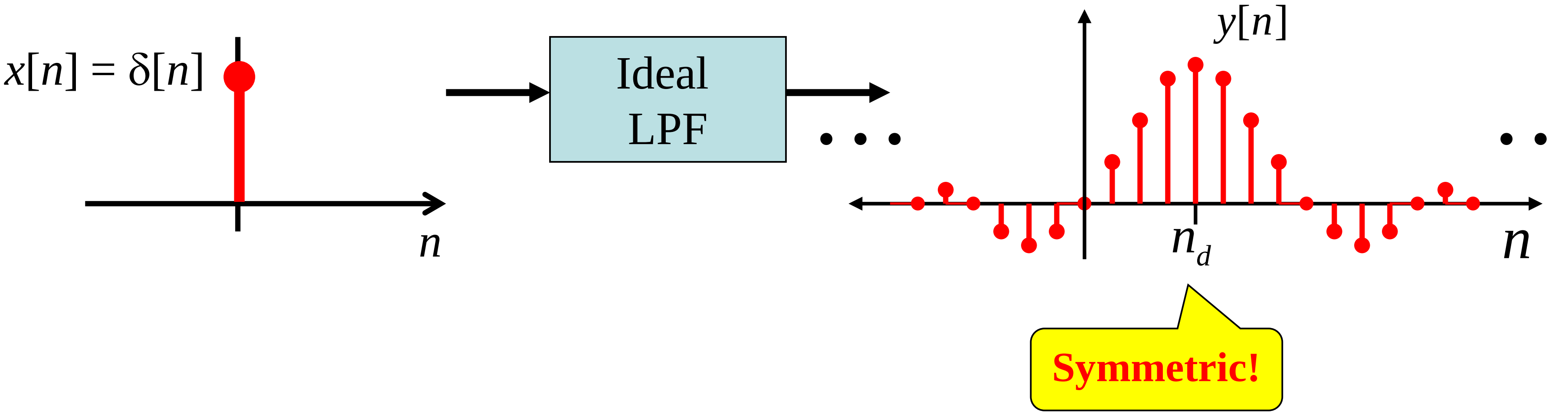
- Linear Phase FIR Design – Windowing Method
- Reading: Sect. 10.2.2 of Proakis & Manolakis

Ideal Lowpass Filter (LPF)





So the response to a delta (applied at $t = 0$) is: $y[n] = (B/\pi)\,\text{sinc}\left[(B/\pi)(n - n_d)\right]$

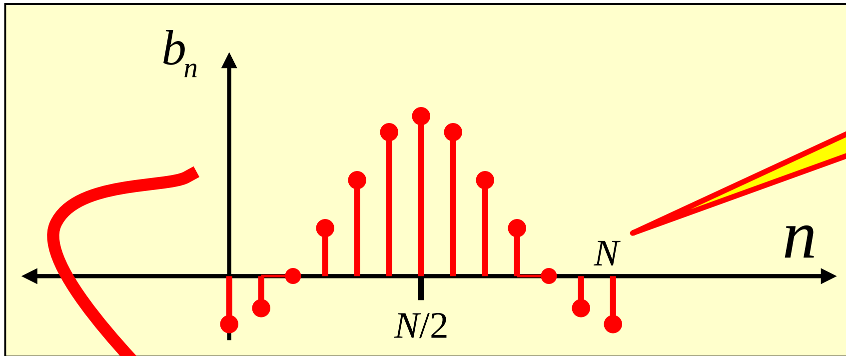


Causal LP FIR Design – Truncated sinc Method

In practice, the best we can do is try to approximate the ideal LPF.

…a simple approach is to define the "b" coefficients in terms of a ***truncated*** shifted sinc function:

Note: Truncation point & Delay are linked



N is filter "order"
$N + 1$ is filter length

B = "Cutoff Frequency"

$$b_n = (B/\pi)\,\text{sinc}\left[(B/\pi)(n - N/2)\right], \quad n = 0, 1, 2, \ \ldots\ , N$$

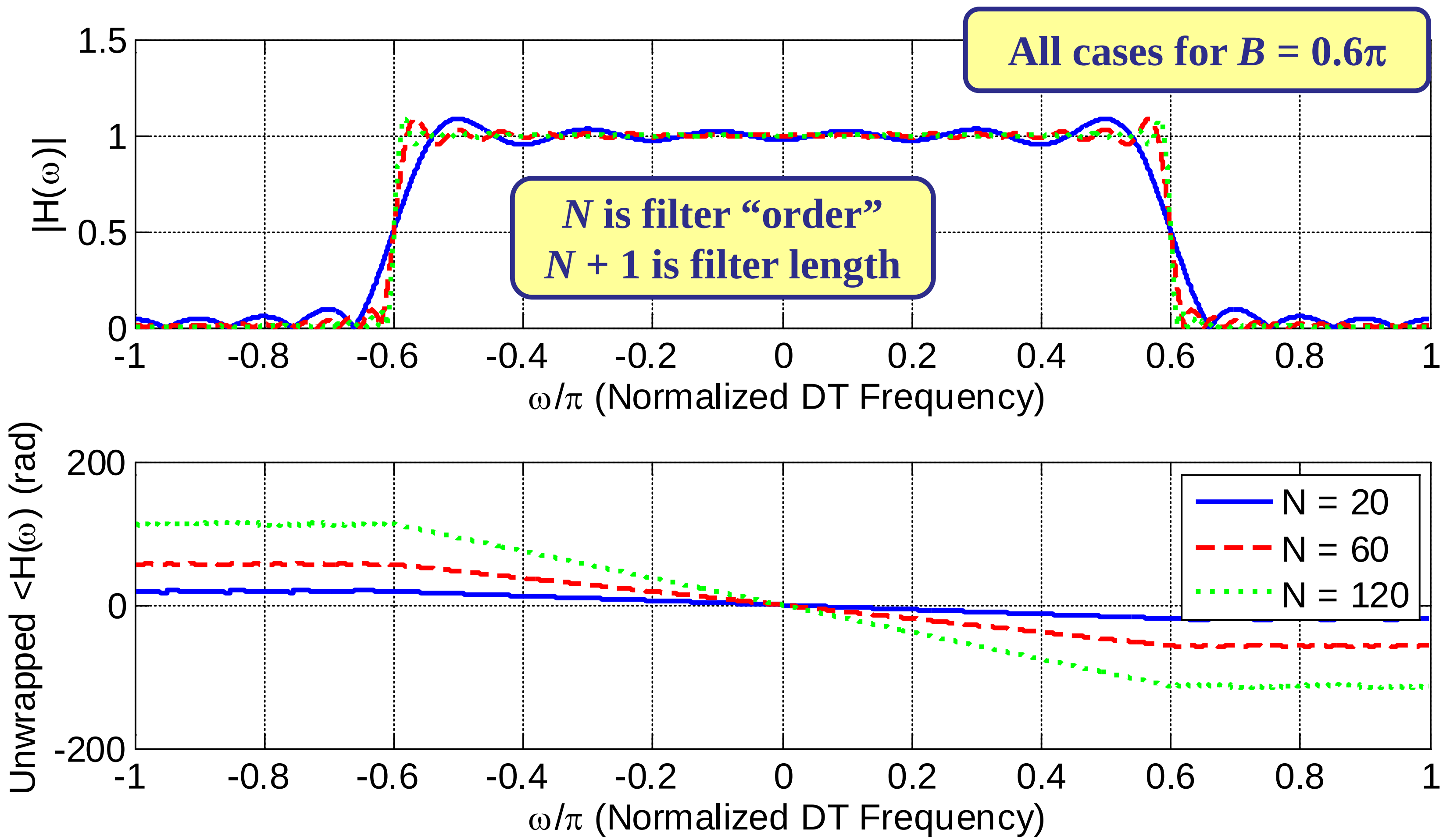
$$y[n] = b_0 x[n] + b_1 x[n-1] + \ldots + b_N x[n-N]$$

Causal, Non-Recursive Filter

$$H^z(z) = b_0 + b_1 z^{-1} + \ldots + b_N z^{-N}$$

$$H^f(\omega) = b_0 + b_1 e^{-j\omega} + \ldots + b_N e^{-j\omega N}$$

Let's see how well this method works… These all have Linear Phase!



<u>Some general insights</u>

Longer lengths for the truncated impulse response:

- **Gives "better" approximation to the ideal filter response**
- **Has steeper phase slope ➔ Longer delay**

MATLAB Code for Previous Plots

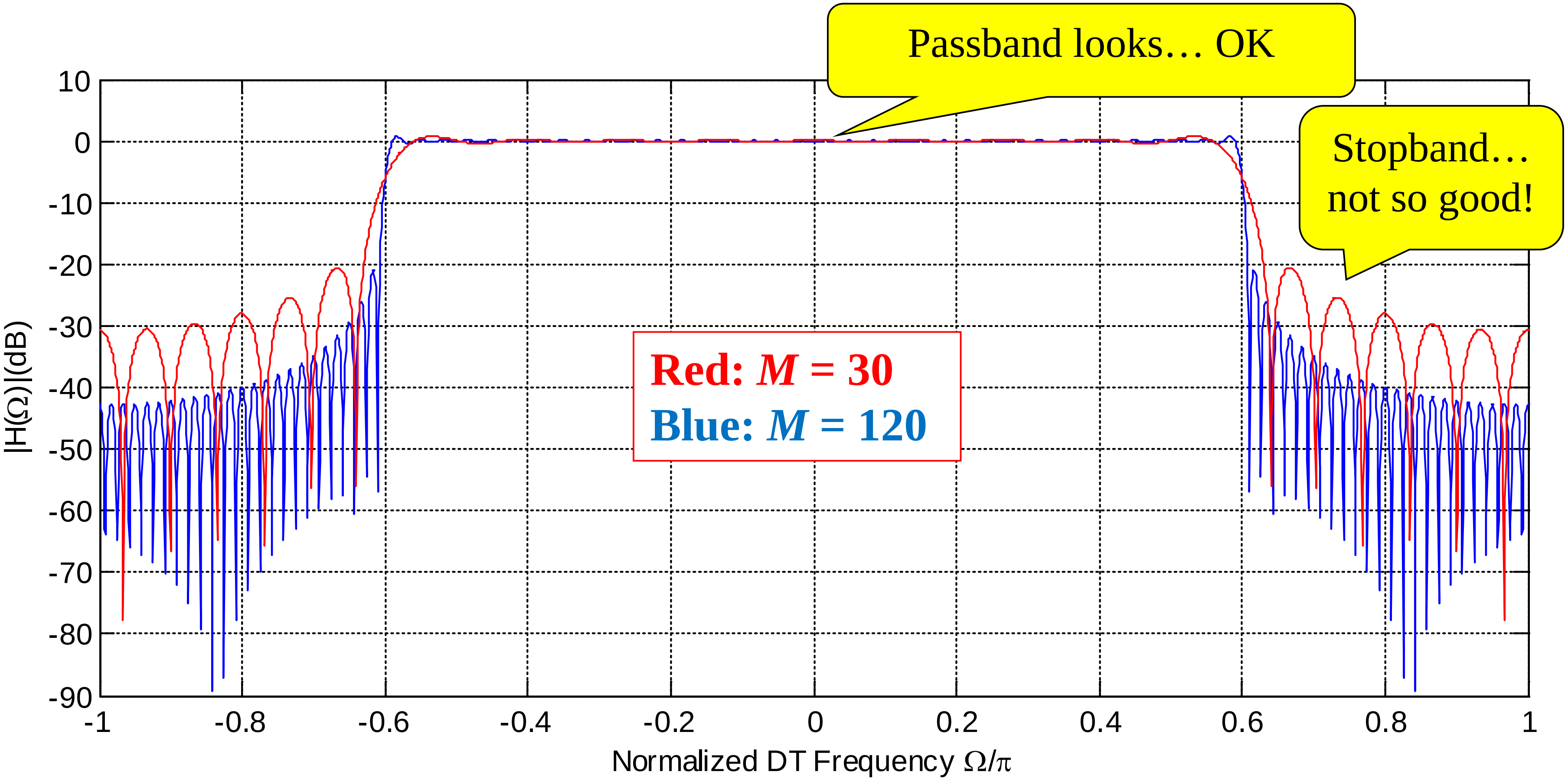
```
B=0.6*pi;
N=20;
n=0:N;h=(B/pi)*sinc((B/pi)*(n-N/2));

PI=fix(1000*pi)/1000;
omega=-PI:0.001:PI;
I = find(omega==0);
H=freqz(h,1,omega);
subplot(2,1,1)
hh=plot(omega/pi,abs(H));
set(hh,'linewidth',2)
hh=gca;
set(hh,'fontsize',14)
hold on
phi = unwrap(angle(H));
phi=phi-phi(I);
```

```
xlabel('\omega/\pi (Normalized DT Frequency)','fontsize',14)
ylabel('|H(\omega)|','fontsize',14)
grid
subplot(2,1,2)
hh=plot(omega/pi,phi);
set(hh,'linewidth',2)
hh=gca;
set(hh,'fontsize',14)
xlabel('\omega/\pi (Normalized DT Frequency)','fontsize',14)
ylabel('Unwrapped <H(\omega) (rad)','fontsize',14)
grid
hold on

< do similar things for N = 60  &  N = 120>

legend('N = 20','N = 60','N = 120')
```

For DT filters… “always” plot in dB but “never” use a log frequency axis!

This “truncated sinc” approach is a very simplistic approach and does not yield the best possible filters… as we can see even better in the dB plot below!

There are ways to make this design method better….



FIR Design – Windowed Truncation Method

The truncation method has some potential but does not give really good designs.

When we studied DFT processing of truncated signals we saw that tapered windows helped and they can help here too… for the very same reason!

Suppose you have a desired frequency response for which you'd like to design a realizable FIR filter: $H_d^{\mathrm{f}}(\omega)$

The corresponding impulse response is: $h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d^{\mathrm{f}}(\omega) e^{j\omega n} d\omega$

This is likely to be non-causal as well as infinite in duration…

So… we apply a general window to a delayed version to get the FIR taps:

$$h[n] = h_d[n - n_d]w[n]$$

where $w[n]$ is non-zero only over $n = 0, 1, 2, \ldots, M - 1$

Effect of Rectangular Window

DTFT of <u>delayed</u> $h_d[n]$

$$h[n] = h_d[n - n_d]w[n] \Rightarrow H^f(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \tilde{H}_d^f(v)W^f(\omega - v)d\omega$$

Convolution… causes smearing of the desired freq response

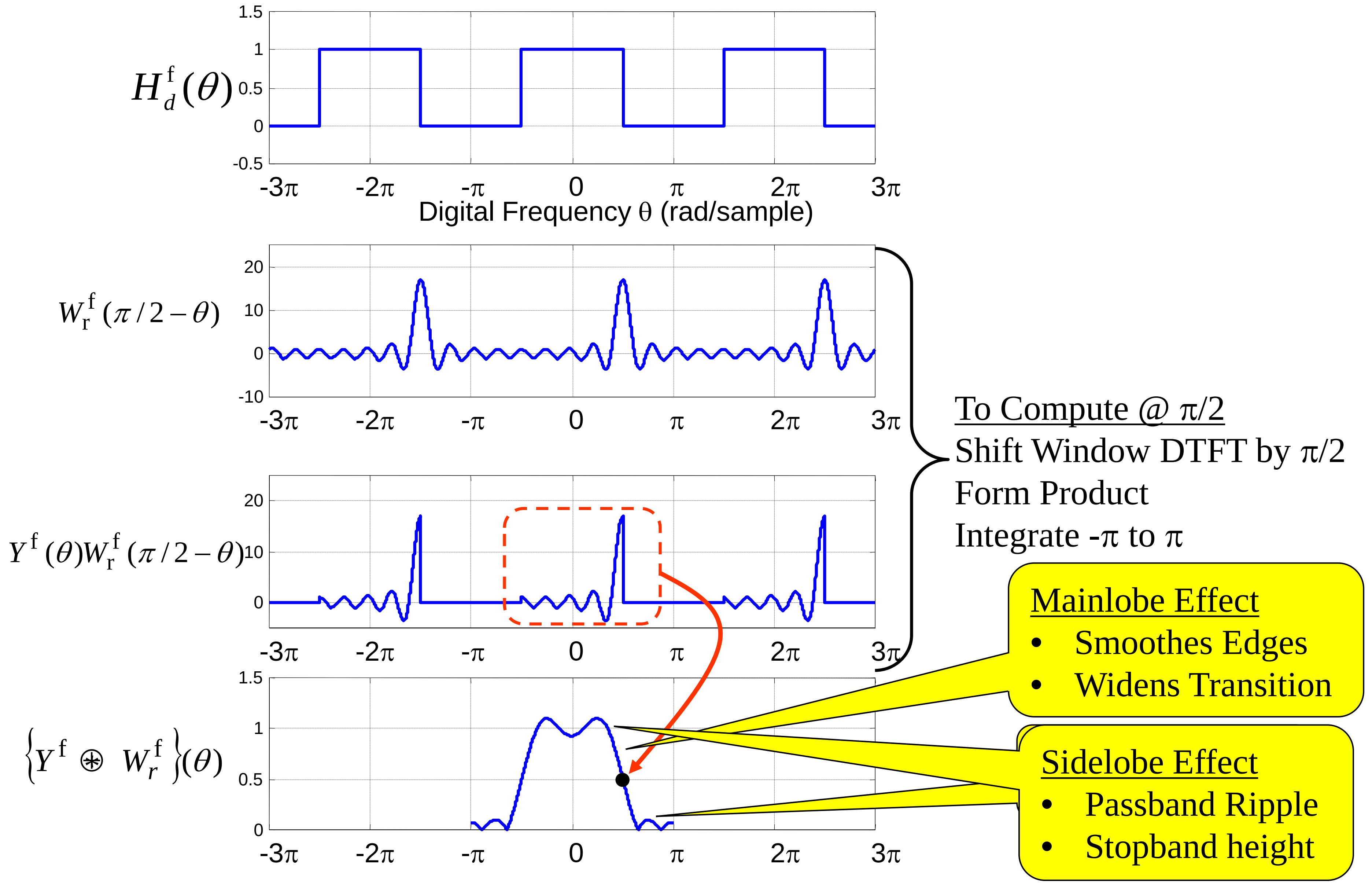
The thing that causes the smearing is the DTFT of the window function:

$$W^f(\omega) = \sum_{n=0}^{M-1} w[n]e^{-j\omega n}$$

For the rectangular window used in the pure truncation version of this:

$$W_r^f(\omega) = \sum_{n=0}^{M-1} e^{-j\omega n} = e^{-j\omega(M-1)/2}\frac{\sin(\omega M/2)}{\sin(\omega/2)}$$

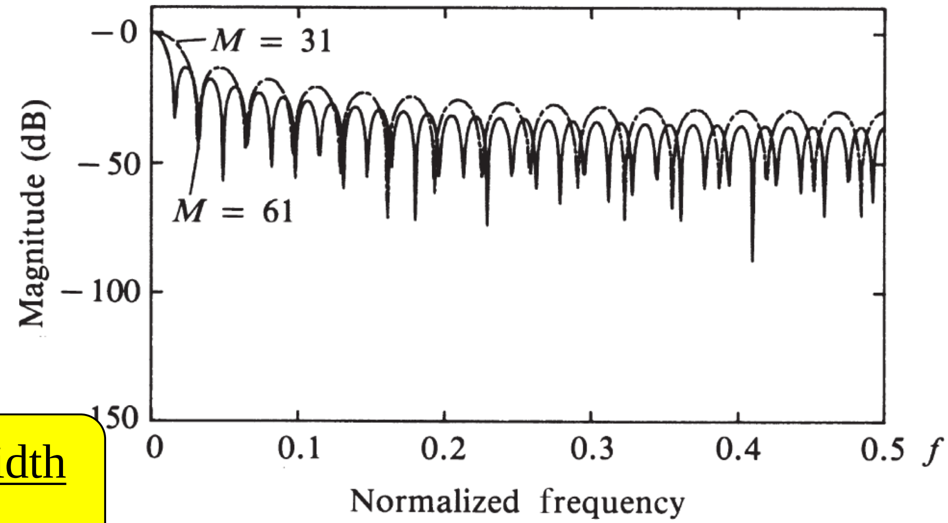
sinc-like!


$H_d^f(\theta)$
Digital Frequency θ (rad/sample)
$W_r^f(\pi/2-\theta)$
$Y^f(\theta)W_r^f(\pi/2-\theta)$
$\{Y^f \circledast W_r^f\}(\theta)$
-3π -2π -π 0 π 2π 3π
To Compute @ π/2
Shift Window DTFT by π/2
Form Product
Integrate -π to π
Mainlobe Effect
• Smoothes Edges
• Widens Transition
Sidelobe Effect
• Passband Ripple
• Stopband height

Increased Length Helps

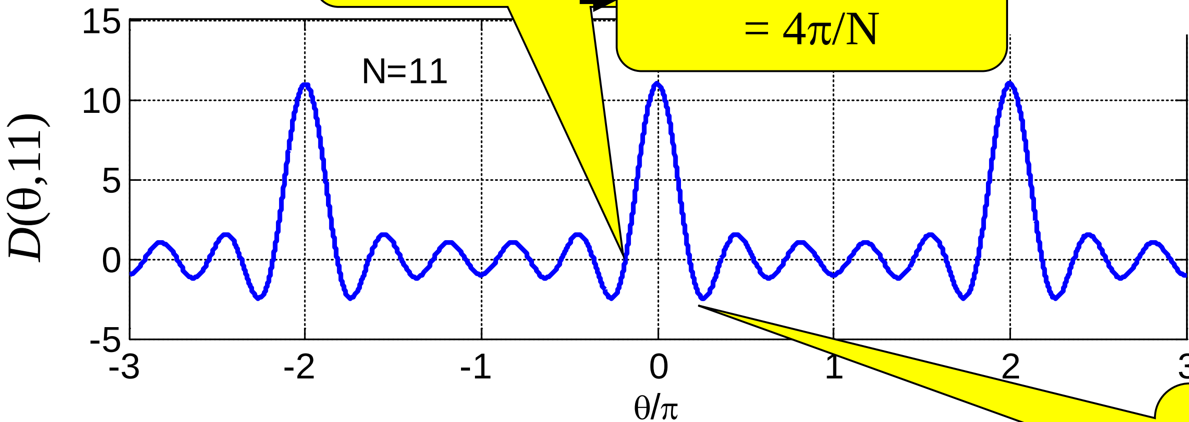
- Mainlobe Gets Narrower as $N\uparrow$
- Sidelobes "Get Lower" as $N\uparrow$
- Height of Mainlobe = N

➔ **Looks more like delta as $N\uparrow$**

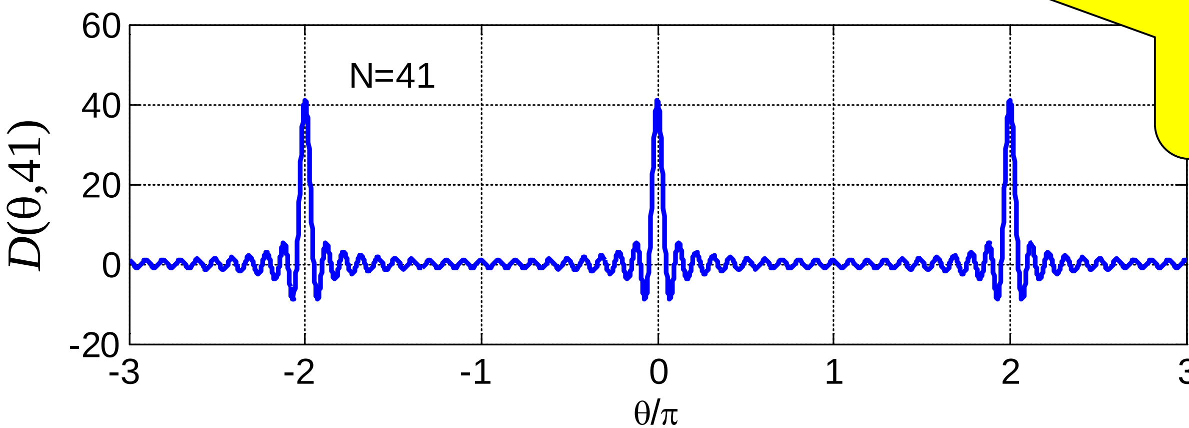


Nearest Zero
@ $\theta=2\pi/N$

Mainlobe Width
$= 4\pi/N$

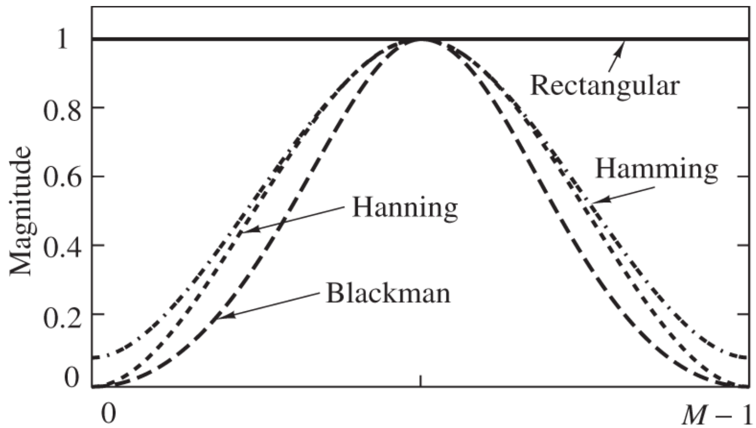


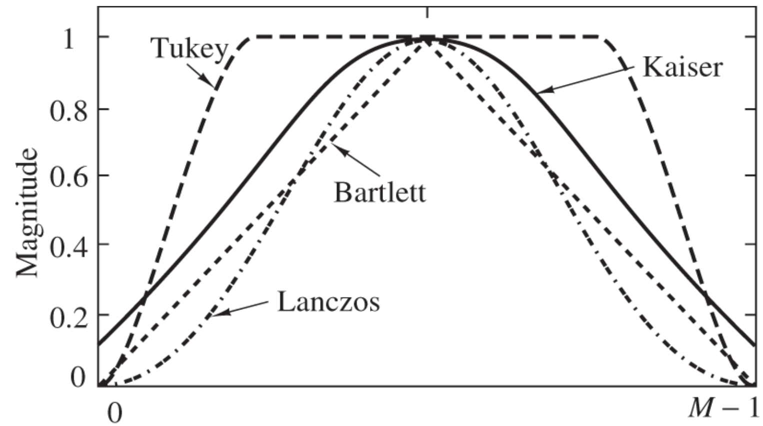
Largest Sidelobe
–13 dB w.r.t. ML peak
(For Any N)



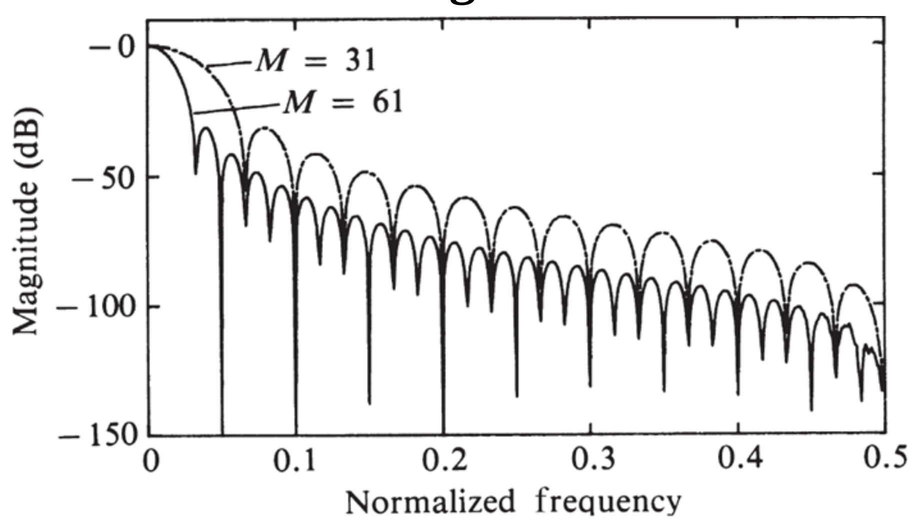
Window Shape Helps More

With the rectangular window the sidelobes are a problem… always no lower than -13 dB.   Causes excessive stopband height!!

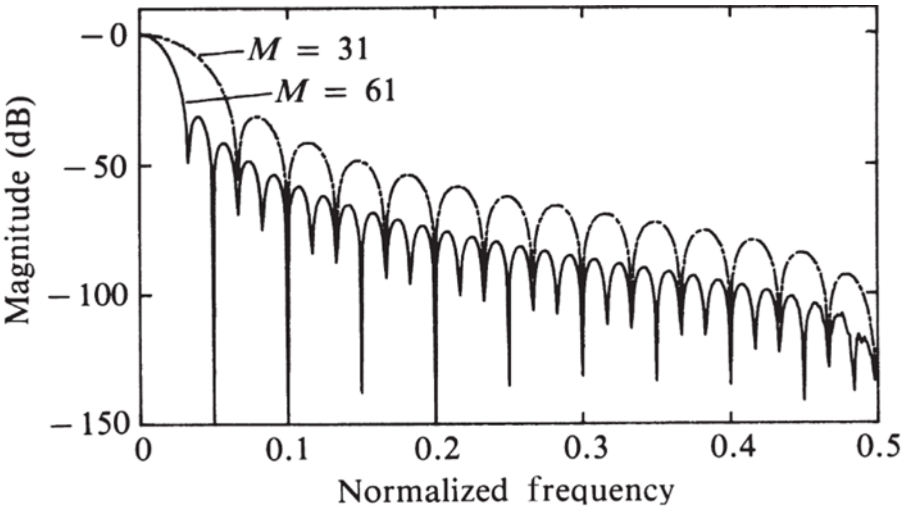




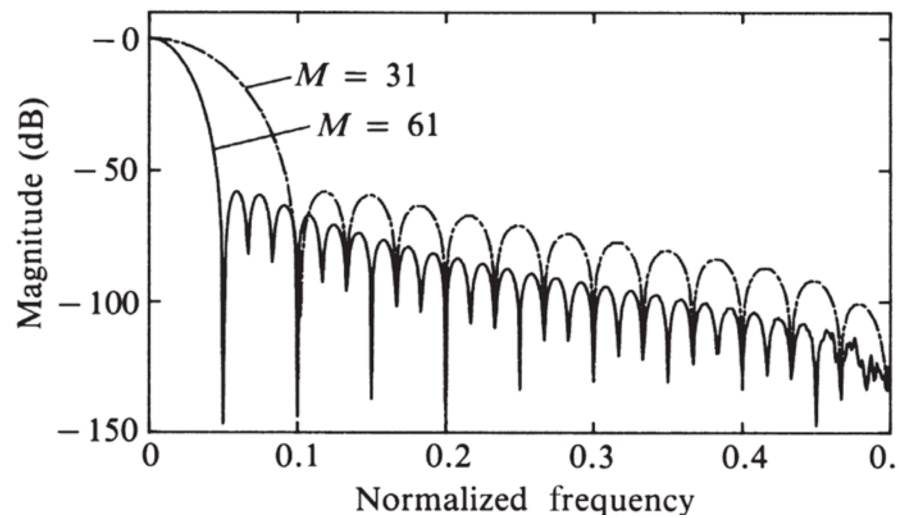
Hanning Window



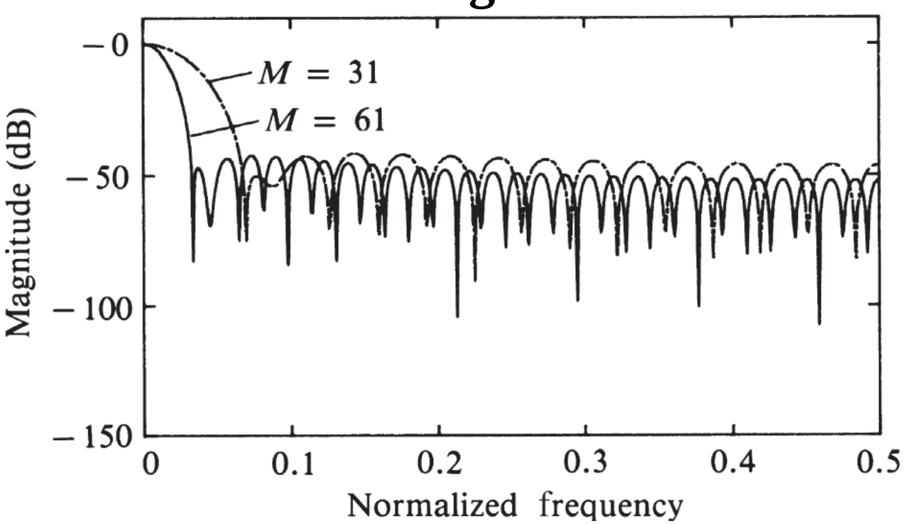
Hanning Window DTFT



Blackman Window DTFT

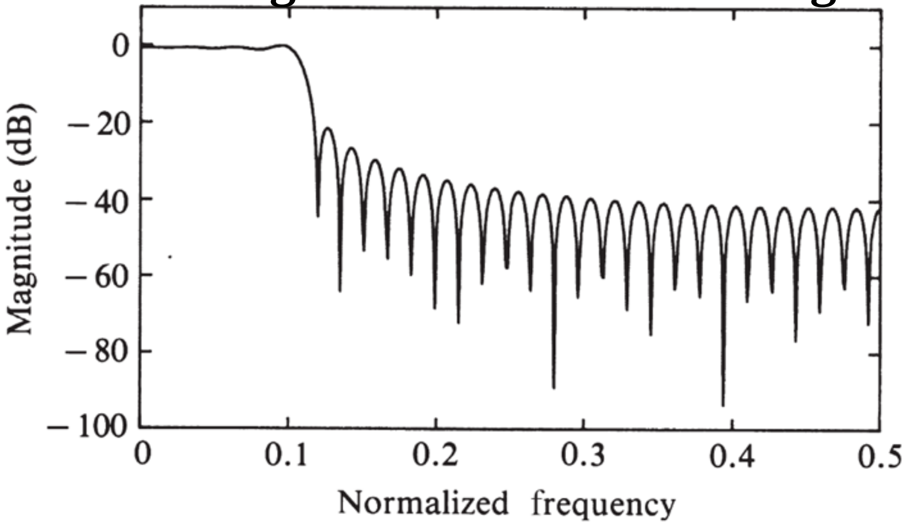


Hamming Window DTFT

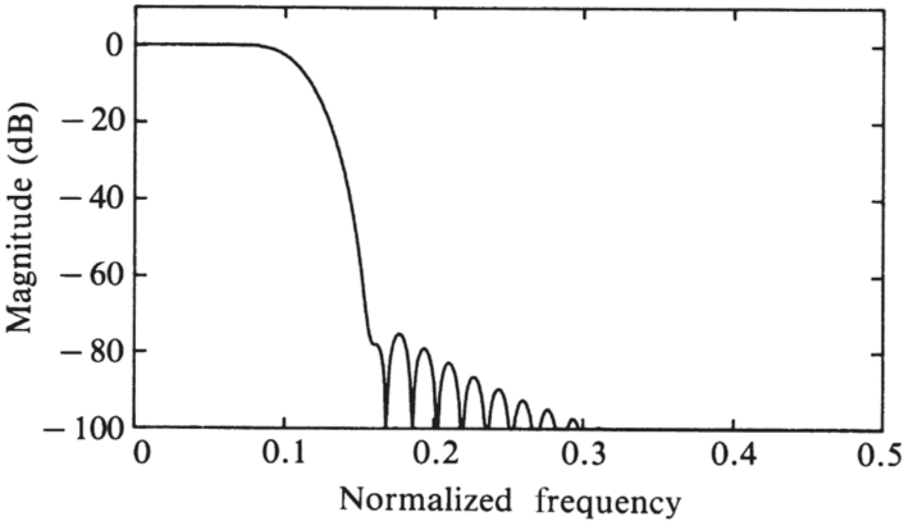


Filter Length $M = 61$

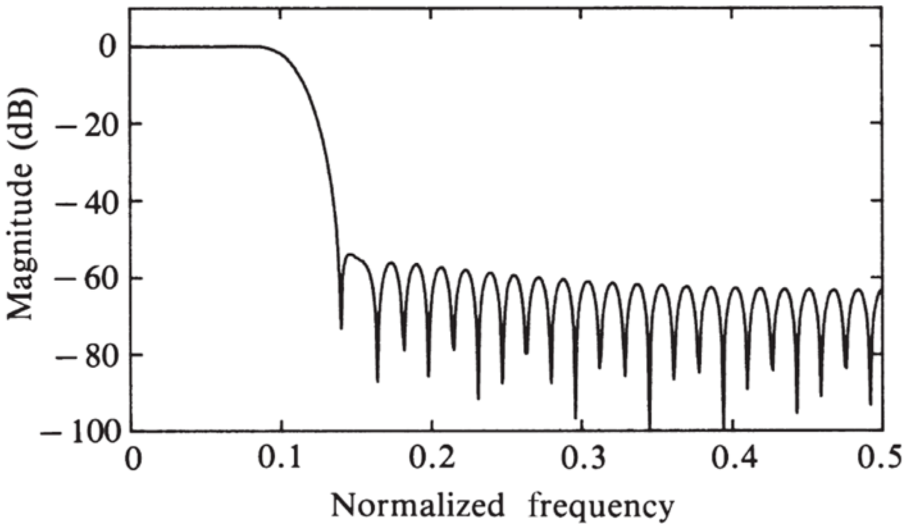
Rectangular Window FIR Design



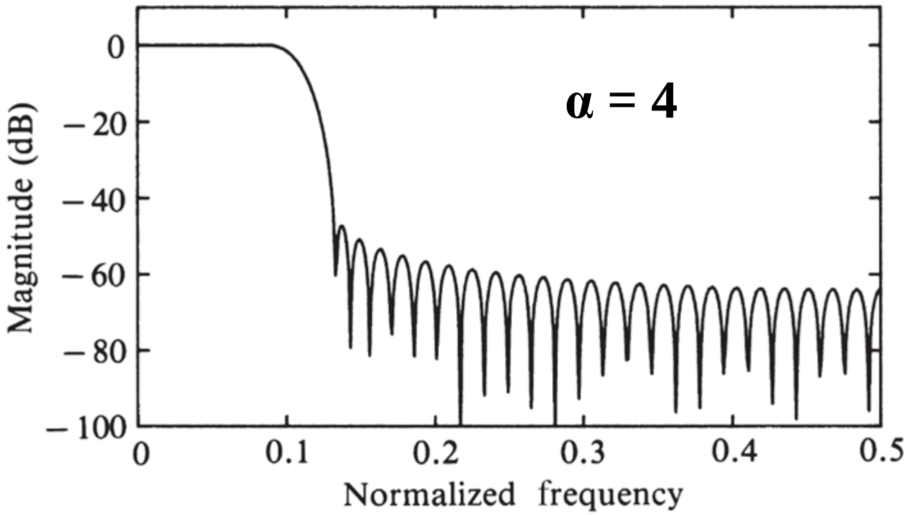
Blackman Window FIR Design



Hamming Window FIR Design



Kaiser Window FIR Design



MATLAB fir1

>> help 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.

If Wn is a two-element vector, Wn = [W1 W2], with $W1 < W2$.

B = fir1(N,Wn,‘bandpass') will design a bandpass filter.

B = fir1(N,Wn,'stop') will design a bandstop filter.

If Wn is a multi-element vector, Wn = [W1 W2 W3 W4 W5 ... WN], gives order N multiband filter

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.

Hamming is Default!