Administrative

- Quiz2 grades posted on Trunk. Mean 18.6 out of 22
- Matlab3 posted on Trunk. Mean 90.8
  - –Matlab2 grades will be posted by end of week
- Matlab 5 is due next Thursday, but going over it may help with Exam 2 prep; posted on Trunk
- Exam 2 is 1 week from today. Review session- Monday or Tuesday evening?

- Review where we are in class

EE-125: Digital Signal Processing

Finish up Spectrograms

FIR Filter Design 1

Professor Tracey

Tufts

First… some review of spectrum estimation

- Spectrogram examples
  - Same tradeoffs as before: windows with low sidelobes gives wider mainlobe (thus worse resolution)
  - New tradeoff: windows that are short in time give improved time resolution but worse frequency resolution

- Posted a review article on multitaper spectrum estimation on Trunk. Not covered on tests or HW, but is good to know about
  - Can be used for periodograms or smoother spectrograms, when signal is time-varying but random

“Big picture” on FIR filters

- FIR filters are nice because they can have linear phase / constant group delay, though at the cost of needing more coefficients than IIR (as seen in MATLAB3)
- The computational load of FIR filters can be reduced by doing FFT-based filtering (circular convolution = linear convolution) (MATLAB4).
- You’ll design FIR filters in MATLAB6 (the last Matlab)

FIR filter design …

- Basic idea of FIR filter design:
  - Decide on the ideal frequency response $H(\omega)$ (LPF, HPF, etc)
  - Pick a form for the filter that guarantees linear phase FIR
  - Find the coefficient weights that best match $H(\omega)$
- Last step usually involves trading off fast frequency transitions to get lower responses elsewhere
- Three main design approaches:
  - Window method
  - Frequency sampling
  - Optimization of a cost function

Filter response specifications

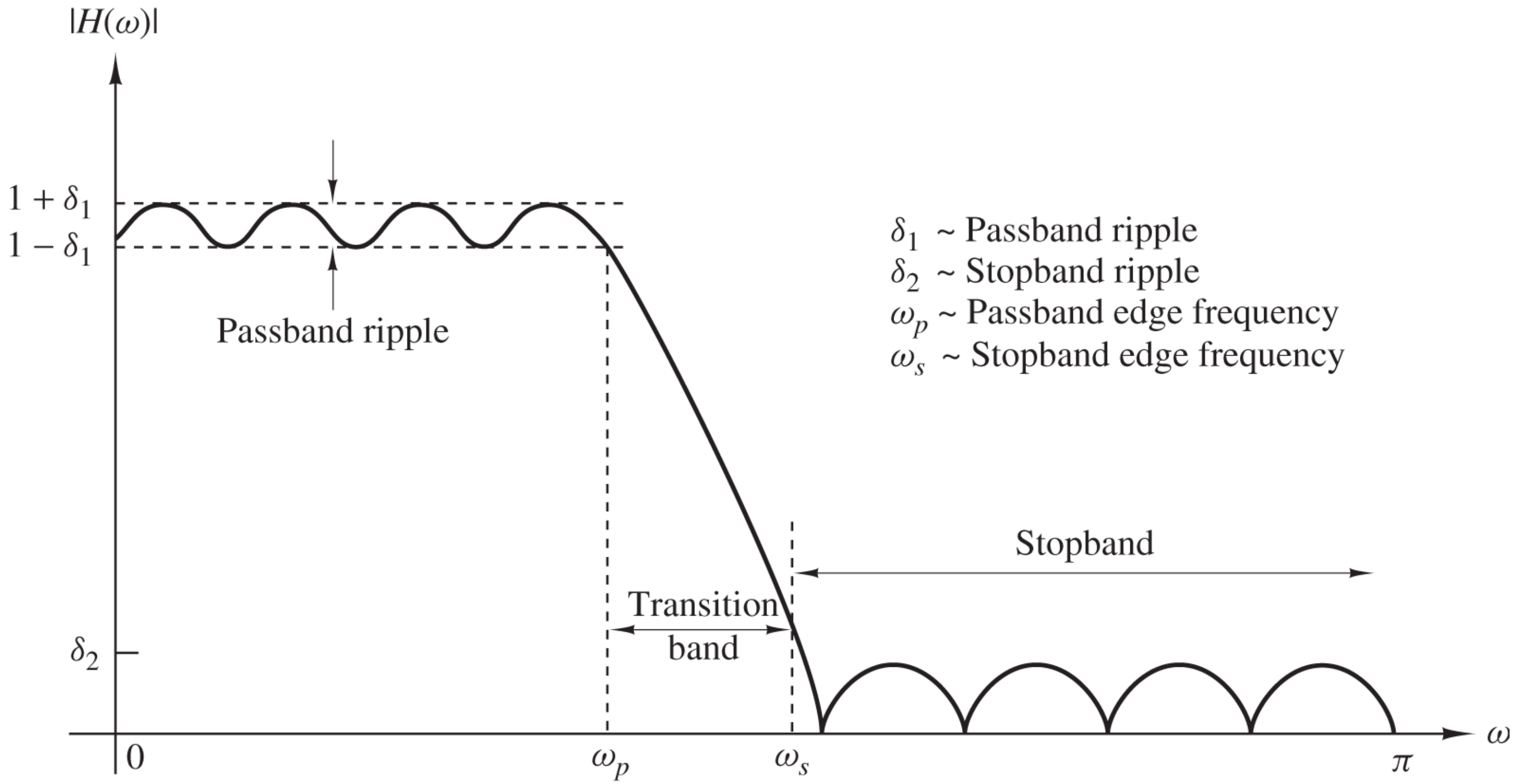


Figure 10.1.2 Magnitude characteristics of physically realizable filters.

Outline

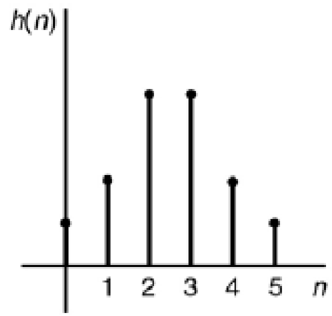
- Linear-phase FIR filters: symmetric and anti-symmetric (P&M section 10.2.1)

- Filter design by windowing (P&M 10.2.2)
  - Rectangular windows
  - Triangle/Bartlett window (mostly for demonstration)
  - Hann (or Hanning)window family (used very often)
  - Many others….Blackman, Kaiser

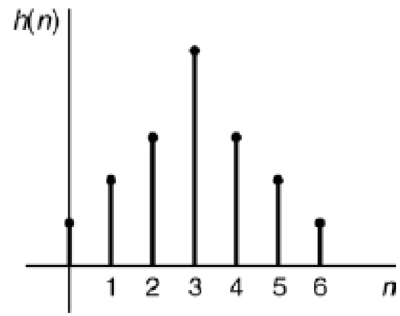
- Filter design by frequency sampling (P&M 10.2.3)
  - Simple versions
  - More optimal

Four basic FIR forms that give linear or generalized linear phase

FIR II: even length, symmetric



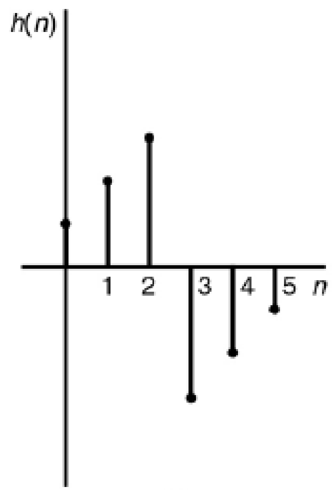
(a)

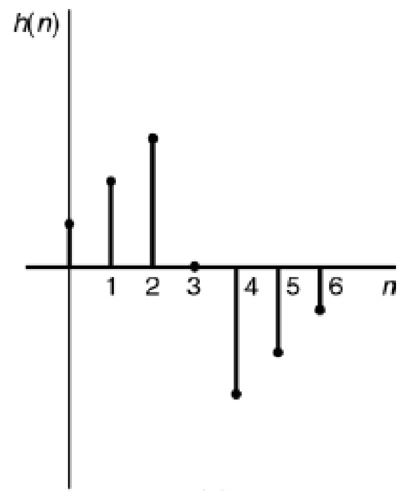


(b)

FIR I: odd length, symmetric

FIR IV: even length, antisymmetric





FIR III: odd length, antisymmetric

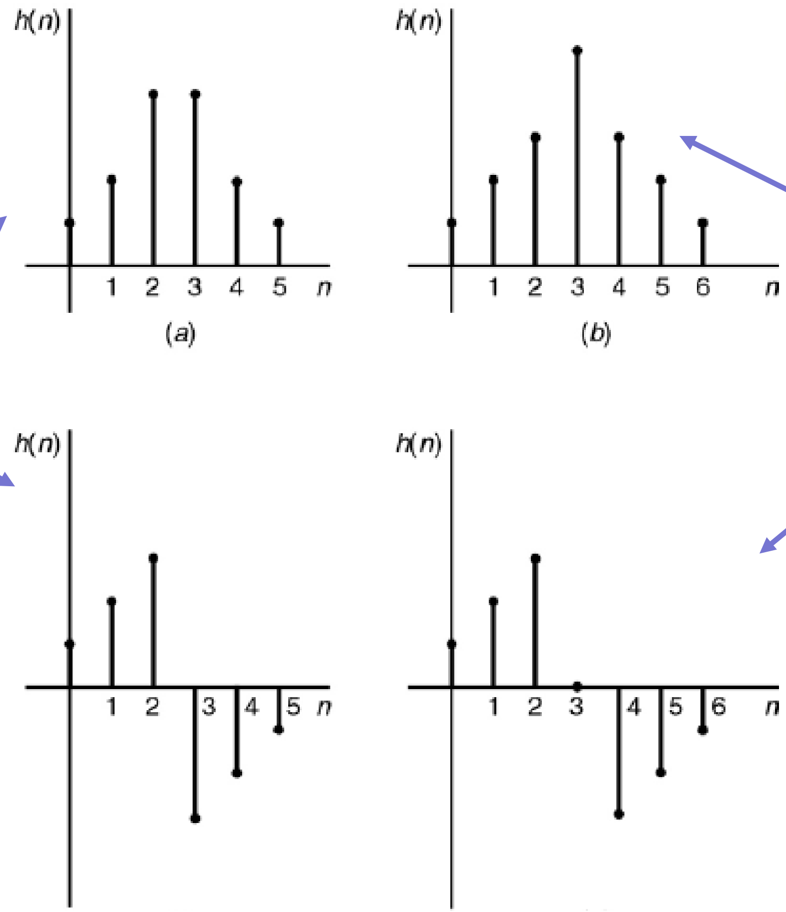
Note for this case that h[M/2]=0

Four basic FIR forms that give linear or generalized linear phase

FIR II: even length, symmetric

Symmetric around “sample” 2.5

FIR IV: even length, antisymmetric



FIR I: odd length, symmetric

Symmetric around sample 3

FIR III: odd length, antisymmetric

Note for this case that h[M/2]=0

Filter restrictions

- Symmetric filters (Types I and II) naturally have $H(\omega)$ that involves $\cos(\omega*\text{something})$
- Antisymmetric filters (Types III and IV) naturally have $H(\omega)$ that involves $\sin(\omega*\text{something})$

- Type II filters always have a zero at $z=-1$, so can't be used to design a highpass

- Type III filters always have a zeros at $z=-1$ and $z = 1$, so can't be used to design a highpass, lowpass, or bandstop

- Type IV (even, antisymmetric) always has zero at $z = 1$, so can't be used to design a lowpass

Filter restrictions

- Symmetric filters (Types I and II) naturally have $H(\omega)$ that involves $\cos(\omega\text{*something})$
- Antisymmetric filters (Types III and IV) naturally have $H(\omega)$ that involves $\sin(\omega\text{*something})$

- Type II filters always have a zero at z=-1, so can't be used to design a highpass

- Type III filters always have a zeros at z=-1 and z = 1, so can't be used to design a highpass, lowpass, or bandstop

- Type IV (even, antisymmetric) always has zero at z = 1, so can't be used to design a lowpass

This is why Matlab's fir1 changed the filter order when you tried to design a 51'st order (52 point long) highpass

FIR filter design …

- Basic idea of FIR filter design:
  - Decide on the ideal frequency response $H(\omega)$
  - Pick a form for the filter that guarantees linear phase FIR
  - Find the coefficient weights that best match $H(\omega)$
- Last step usually involves trading off fast frequency transitions to get lower responses elsewhere
- Three main design approaches:
  - Window method
  - Frequency sampling
  - Optimization of a cost function

Window design considerations

- The main issues are <u>main lobe width</u> and <u>sidelobe height</u>
- Main lobe determine how 'smoothed' the desired response is
  - Thus ideally, main lobe would be very narrow
  - This generally implies higher M
- The sidelobe determines 'ringing' in frequency
  - High sidelobes mean less attenuation of undesired frequencies
  - High frequency sidelobes result from sharp discontinuities in time

- Rectangular window: narrowest mainlobe, but high sidelobes
- Other windows; wider mainlobe but lower sidelobes
  - All are linear phase
  - All have smaller mainlobe as M increases (just like rect)
  - Design is generally trial-and-error

Some possible filters

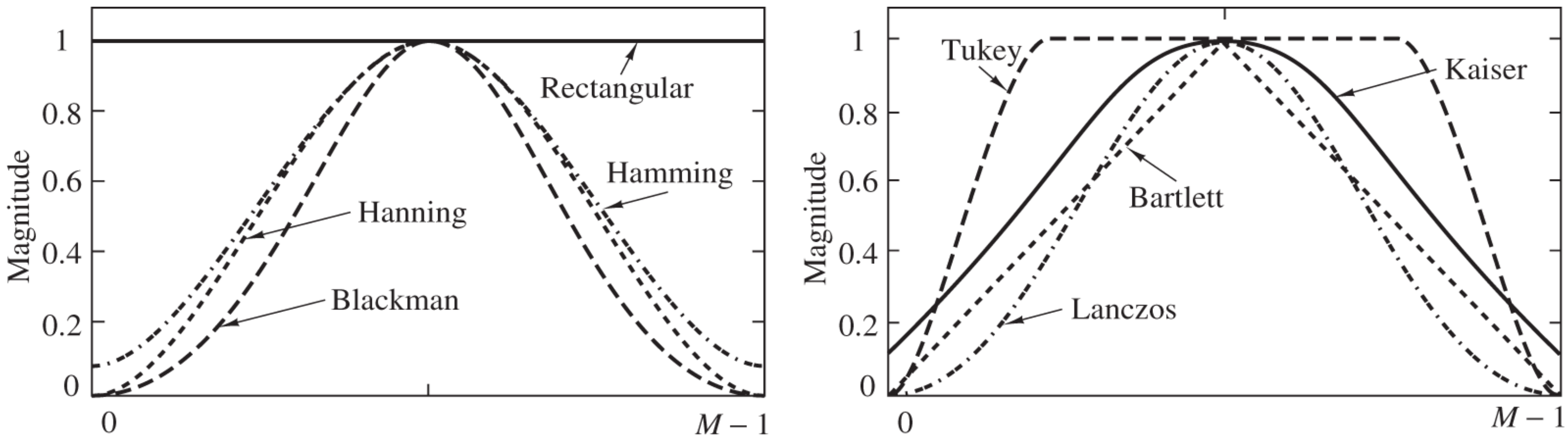


Figure 10.2.3 Shapes of several window functions.

[For lots of details on window types, see https://ccrma.stanford.edu/~jos/sasp/sasp.html](https://ccrma.stanford.edu/~jos/sasp/sasp.html)

In detail....

- Triangle/Bartlett filter – basically convolve two boxcars
- Hamming family: multiply rectangular window by one period of a cosine.
  - General form is:
    $w(n) = rect(n)*[\alpha+2\beta\cos(2\pi n/M)]$
  - If $\alpha=0.5$ and $\beta=0.25$, we get a *Hanning* or *raised cosine*
  - If we chose α,β to cancel the biggest sidelobe, we get *Hamming* – quick transition to low sidelobes, then flat
- If we add multiple cosines, we get the Blackman family of windows.
- Many, many windows to choose from