ESE 531: Digital Signal Processing

Lec 16: March 26, 2020

Design of FIR Filters, Optimal Filter Design

Linear Filter Design

- Used to be an art
  - Now, lots of tools to design optimal filters
- For DSP there are two common classes
  - Infinite impulse response IIR
  - Finite impulse response FIR
- Both classes use finite order of parameters for design
- Today we will focus on FIR designs

What is a Linear Filter?

- Attenuates certain frequencies
- Passes certain frequencies
- Affects both phase and magnitude

- IIR (Oppenheim 7.2-7.4)
  - Mostly non-linear phase response
  - Could be linear over a range of frequencies
- FIR
  - Much easier to control the phase
  - Both non-linear and linear phase

Impulse Invariance

- Let,

$$h[n] = Th_c(nT)$$

$$H_c(j\Omega) = 0, \quad |\Omega| \geq \pi / T$$

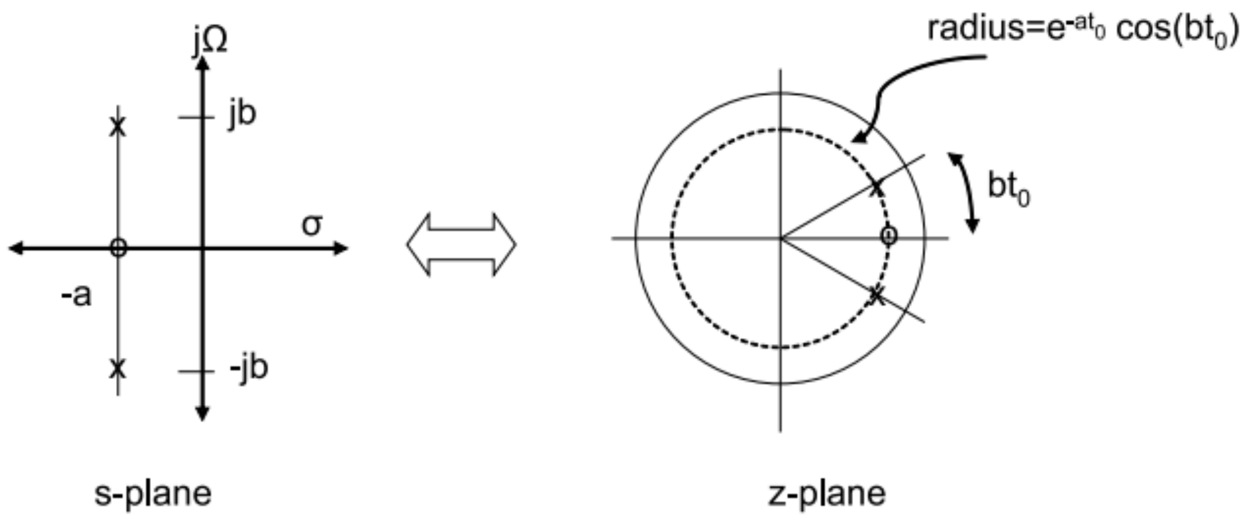
- If sampling at Nyquist Rate then

$$\Omega = wT$$

$$H(e^{j\omega}) = \sum_{k=-\infty}^{\infty} H_c\left[j\left(\frac{\omega}{T} - \frac{2\pi k}{T}\right)\right]$$

$$H(e^{j\omega}) = H_c\left(j\frac{\omega}{T}\right), \qquad |\omega| < \pi$$

S-Plane Mapping to Z-Plane



Bilinear Transformation

- The technique uses an algebraic transformation between the variables *s* and *z* that maps the entire jΩ-axis in the s-plane to one revolution of the unit circle in the z-plane.

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right);$$

$$H(z) = H_c\left(\frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)\right).$$

Bilinear Transformation

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right);$$

- Substituting $s = j\,\Omega$ and $z = e^{j\omega}$
  - Just looking at the $j\Omega$ axis in s-plane (i.e the unit circle in the z-plane)

Bilinear Transformation

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right);$$

- Substituting s = j Ω and z = $e^{j\omega}$

$$s = \frac{2}{T_d}\left(\frac{1 - e^{-j\omega}}{1 + e^{-j\omega}}\right),$$

Bilinear Transformation

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right);$$

- Substituting s = j Ω and z = $e^{j\omega}$

$$s = \frac{2}{T_d}\left(\frac{1 - e^{-j\omega}}{1 + e^{-j\omega}}\right),$$

Euler's formula

$$\frac{2}{T_d}\left[\frac{2e^{-j\omega/2}(j \sin \omega/2)}{2e^{-j\omega/2}(\cos \omega/2)}\right]$$

Bilinear Transformation

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right);$$

- Substituting $s = j\,\Omega$ and $z = e^{j\omega}$

$$s = \frac{2}{T_d}\left(\frac{1 - e^{-j\omega}}{1 + e^{-j\omega}}\right),$$

$$s = j\Omega = \frac{2}{T_d}\left[\frac{2e^{-j\omega/2}(j\sin\omega/2)}{2e^{-j\omega/2}(\cos\omega/2)}\right] = \frac{2j}{T_d}\tan(\omega/2).$$

Bilinear Transformation

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right);$$

- Substituting s = j Ω and z = $e^{j\omega}$

$$s = \frac{2}{T_d}\left(\frac{1 - e^{-j\omega}}{1 + e^{-j\omega}}\right),$$

$$s = j\Omega = \frac{2}{T_d}\left[\frac{2e^{-j\omega/2}(j\sin\omega/2)}{2e^{-j\omega/2}(\cos\omega/2)}\right] = \frac{2j}{T_d}\tan(\omega/2).$$

$$\Omega = \frac{2}{T_d}\tan(\omega/2).$$

~~$\Omega = \omega T$~~

Bilinear Transformation

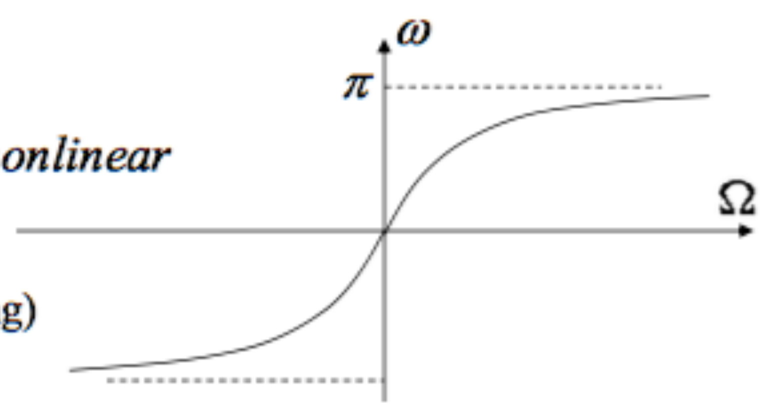
$$\Omega = \frac{2}{T_d}\tan(\omega/2),$$

$$\omega = 2\arctan(\Omega T_d/2).$$

No aliasing, but mapping nonlinear

(Impulse invariance:

linear mapping, but with aliasing)

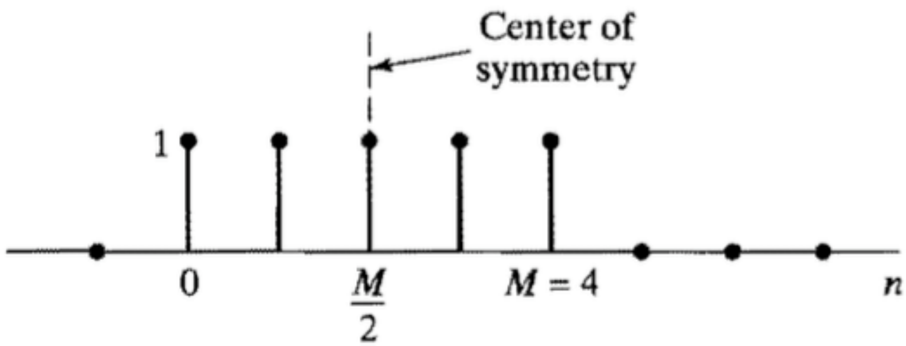


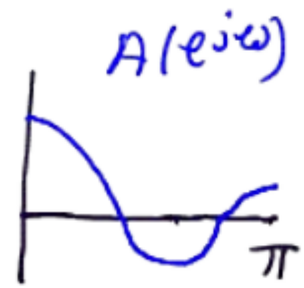
What is a Linear Filter?

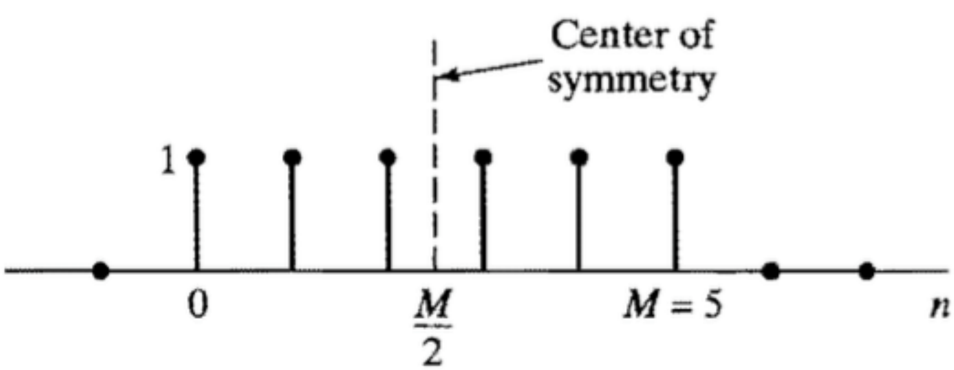
- Attenuates certain frequencies
- Passes certain frequencies
- Affects both phase and magnitude

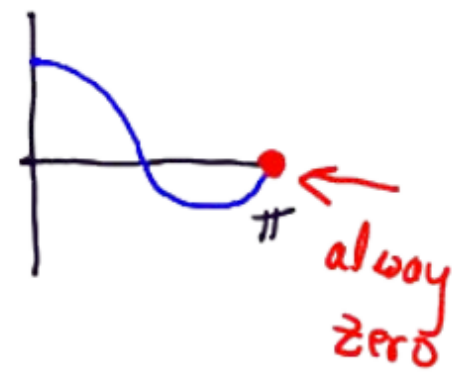
- IIR
  - Mostly non-linear phase response
  - Could be linear over a range of frequencies
- FIR
  - Much easier to control the phase
  - Both non-linear and linear phase

FIR GLP: Type I and II

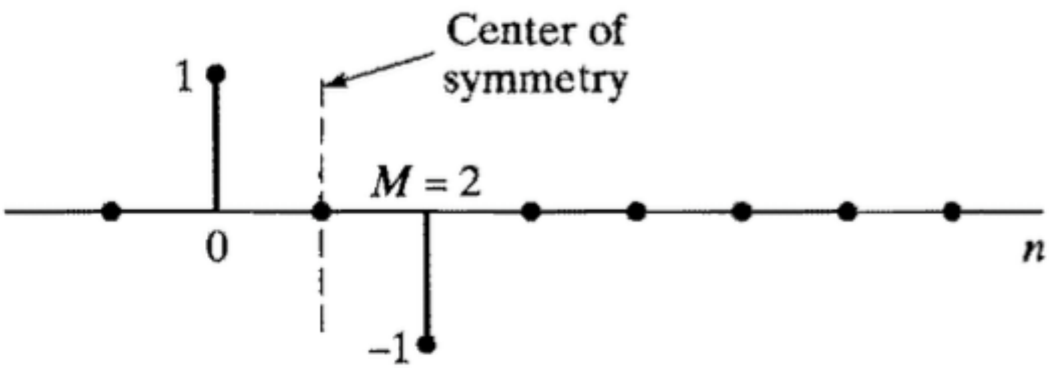


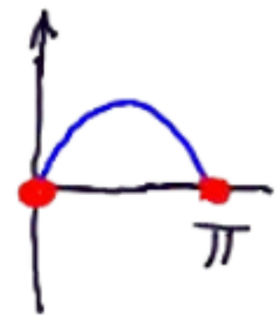


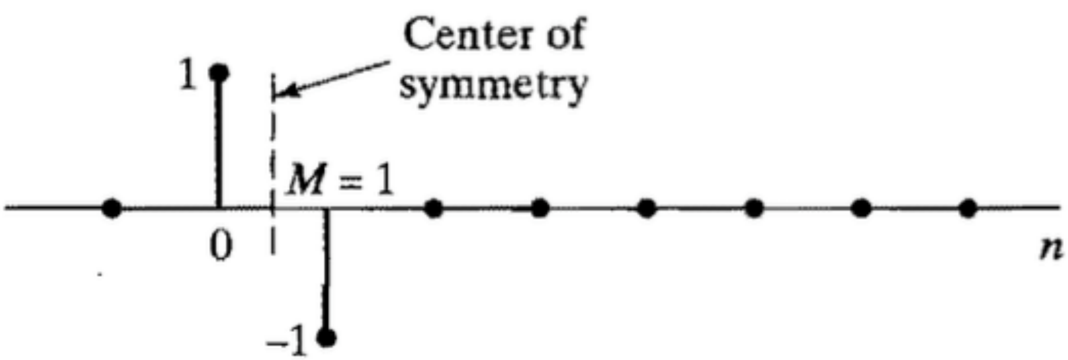


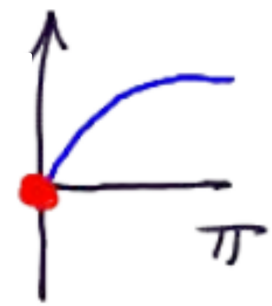


FIR GLP: Type III and IV









FIR Design by Windowing

- Given desired frequency response, $H_d(e^{j\omega})$, find an impulse response

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega$$

ideal

- Obtain the M^{th} order causal FIR filter by truncating/windowing it

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

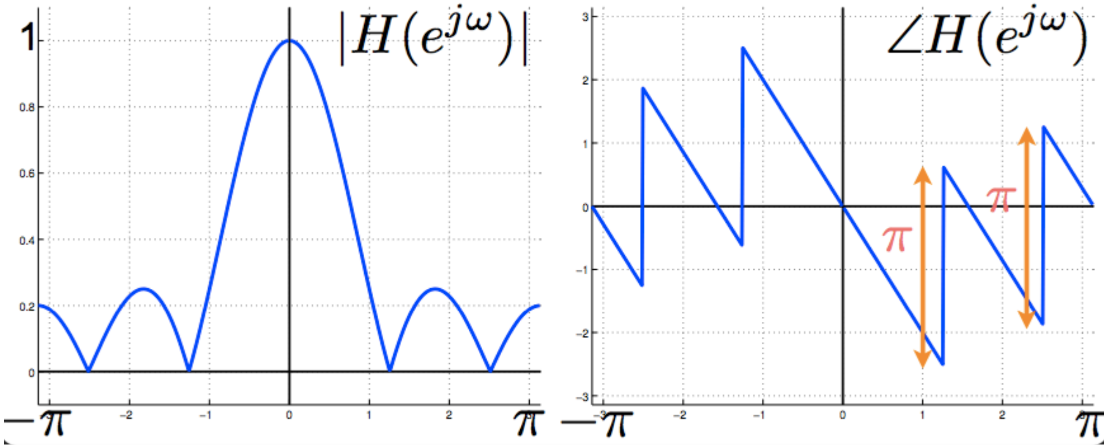
FIR Design by Windowing

- With multiplication in time property,

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

Boxcar Window

$$w[n-M/2] \leftrightarrow W(e^{j\omega}) = e^{-j\omega M/2}\frac{\sin\left((M/2+1/2)\omega\right)}{\sin\left(\omega/2\right)}$$



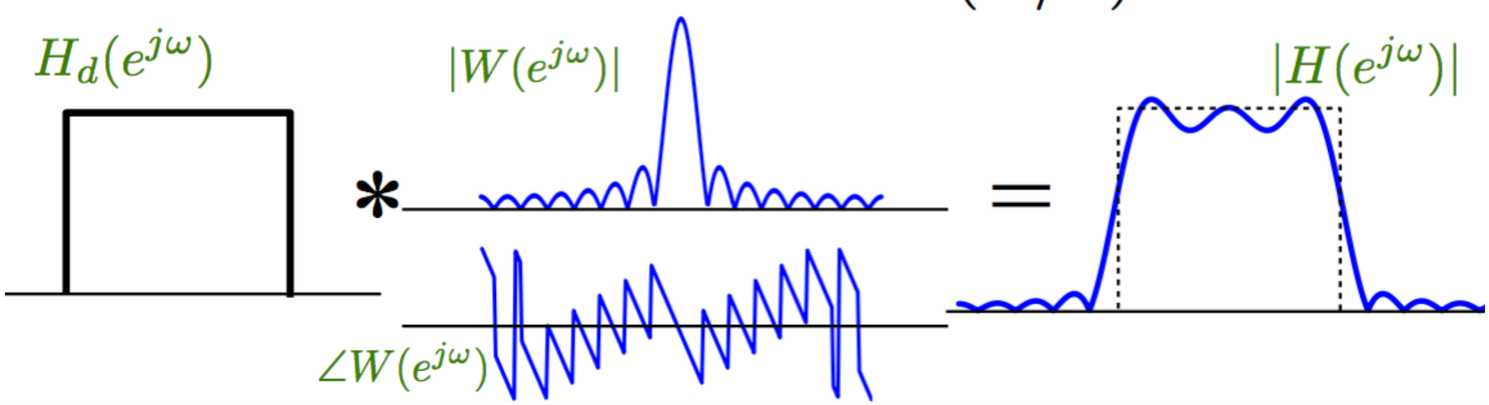
FIR Design by Windowing

- With multiplication in time property,

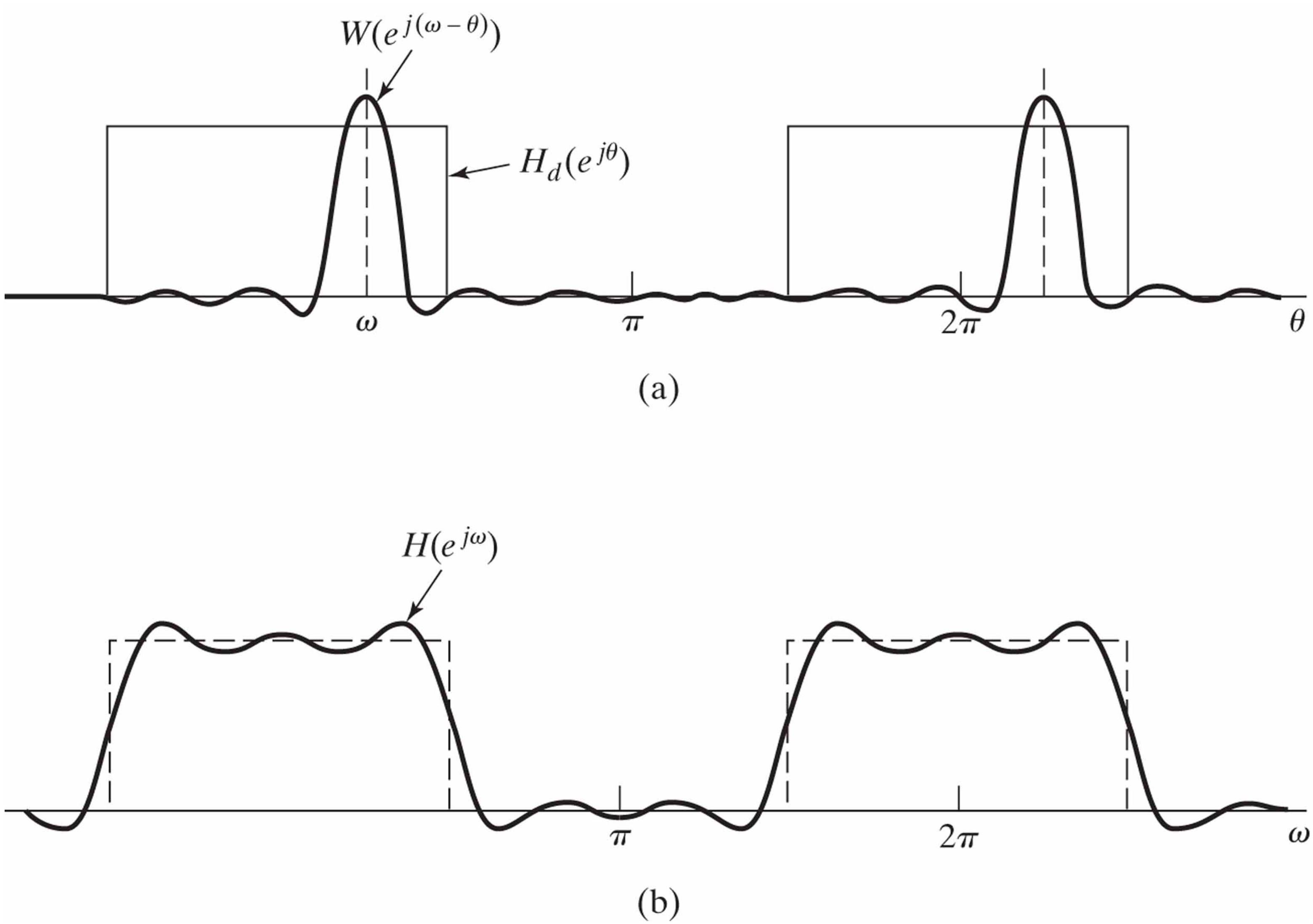
$$H(e^{j\omega}) = H_d(e^{j\omega}) * W(e^{j\omega})$$

- For Boxcar (rectangular) window

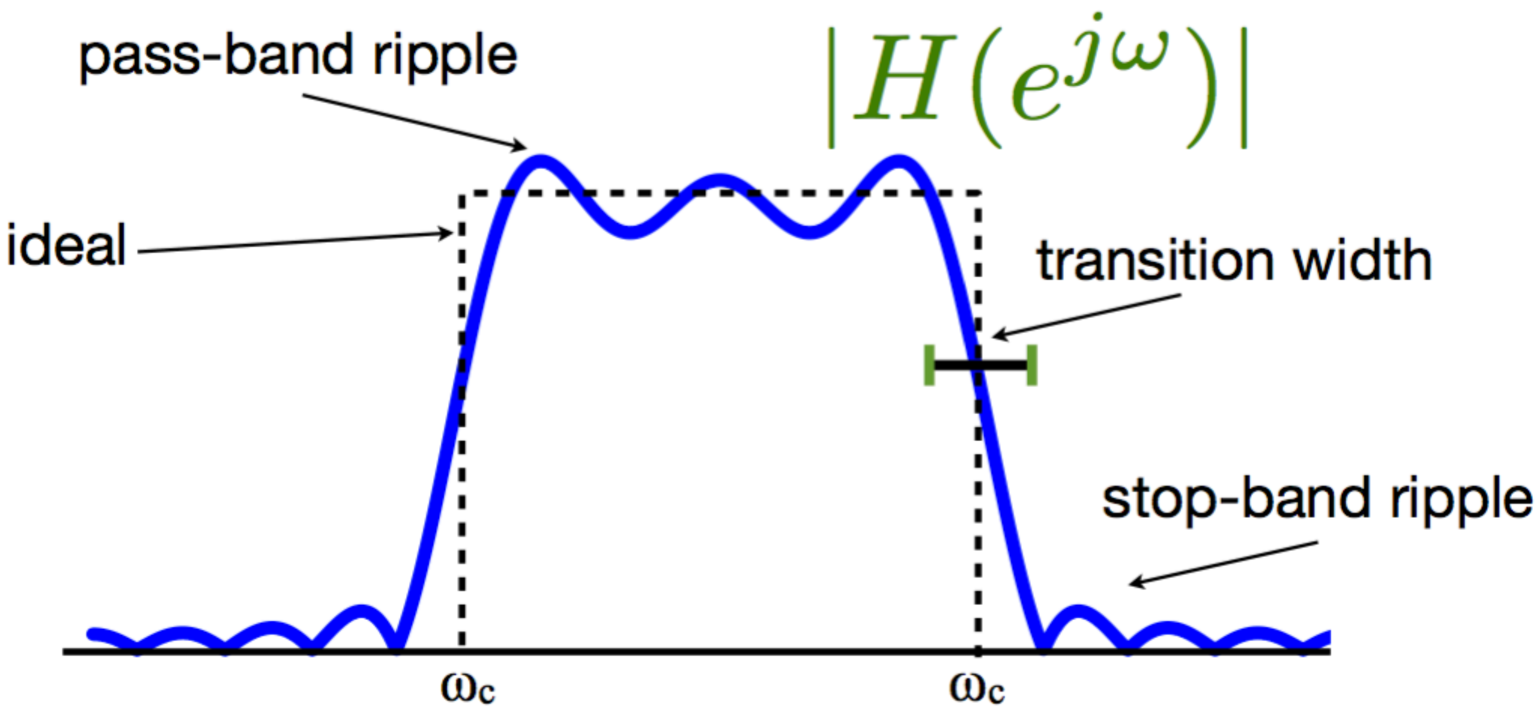
$$W(e^{j\omega}) = e^{-j\omega\frac{M}{2}} \frac{\sin(w(M+1)/2)}{\sin(w/2)}$$



FIR Design by Windowing



FIR Design by Windowing

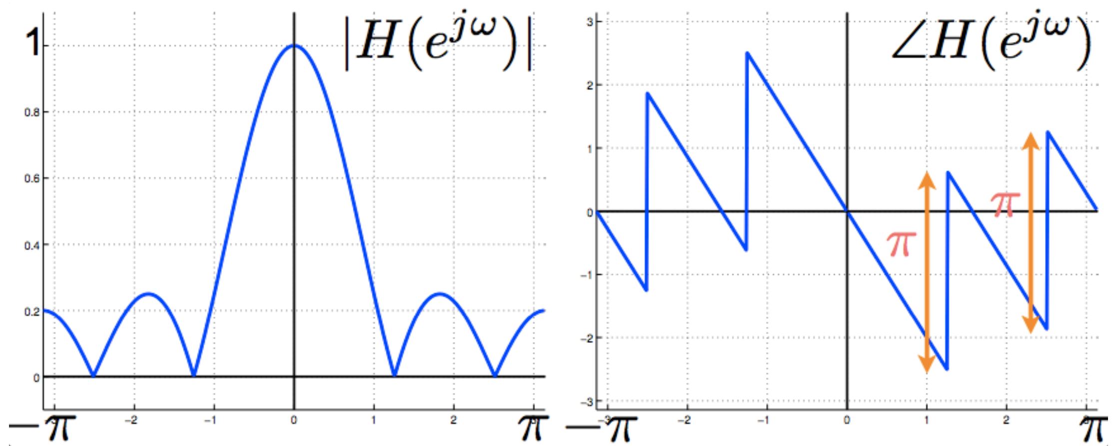


Boxcar Window

Time

$$w[n - M/2] \leftrightarrow W(e^{j\omega}) = e^{-j\omega M/2} \frac{\sin\left((M/2 + 1/2)\omega\right)}{\sin\left(\omega/2\right)}$$

Frequency

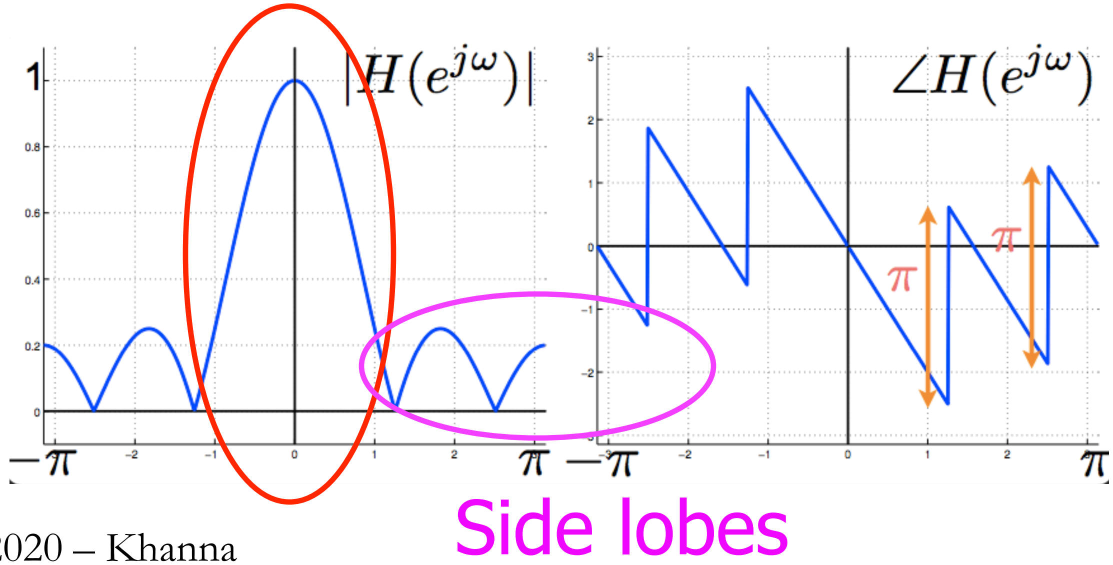


Boxcar Window

Time

$$w[n-M/2] \leftrightarrow W(e^{j\omega}) = e^{-j\omega M/2}\frac{\sin\left((M/2+1/2)\omega\right)}{\sin\left(\omega/2\right)}$$

Frequency



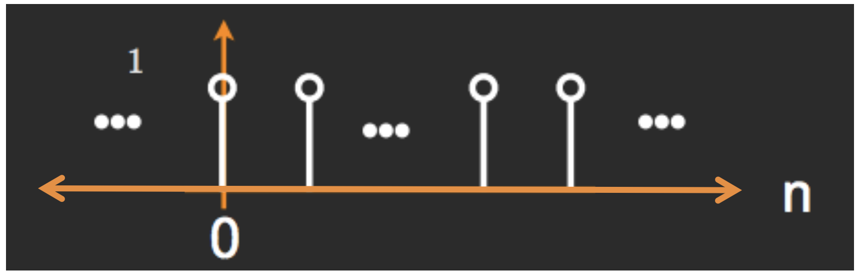
Ideal "Window"?

Time

Frequency

Ideal "Window"?

Time

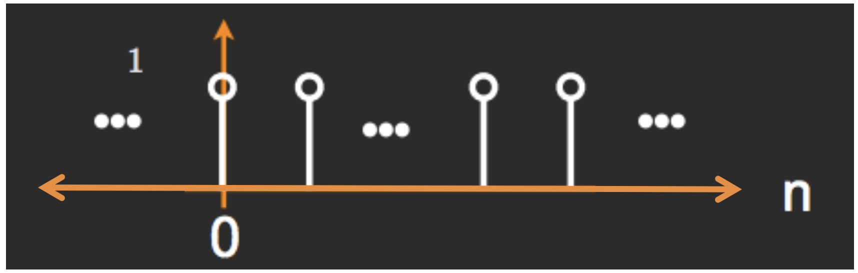


$$w[n] = 1 \leftrightarrow W(e^{j\omega}) = 2\pi\delta[\omega]$$

Frequency

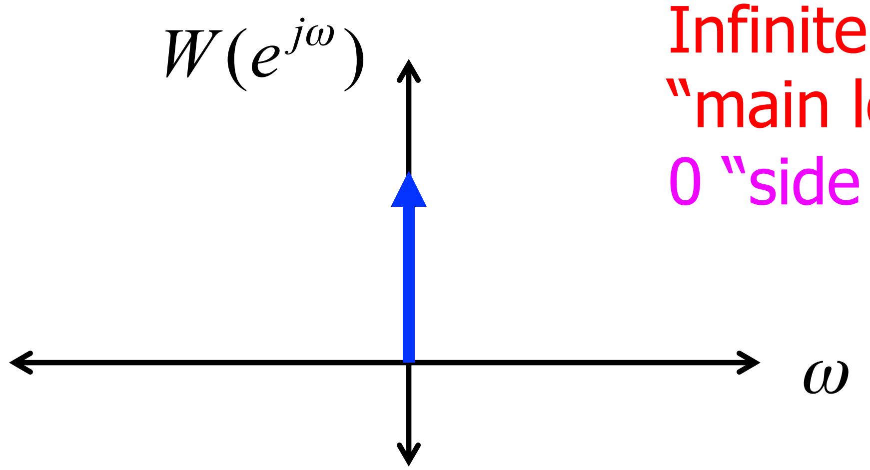
Ideal "Window"?

Time



$$w[n] = 1 \leftrightarrow W(e^{j\omega}) = 2\pi\delta[\omega]$$

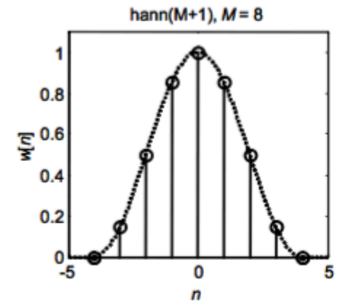
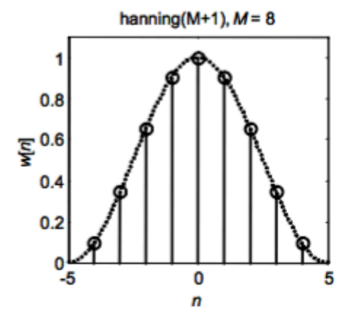
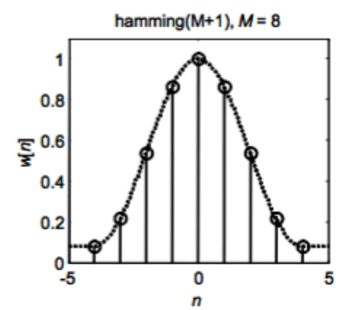
Frequency



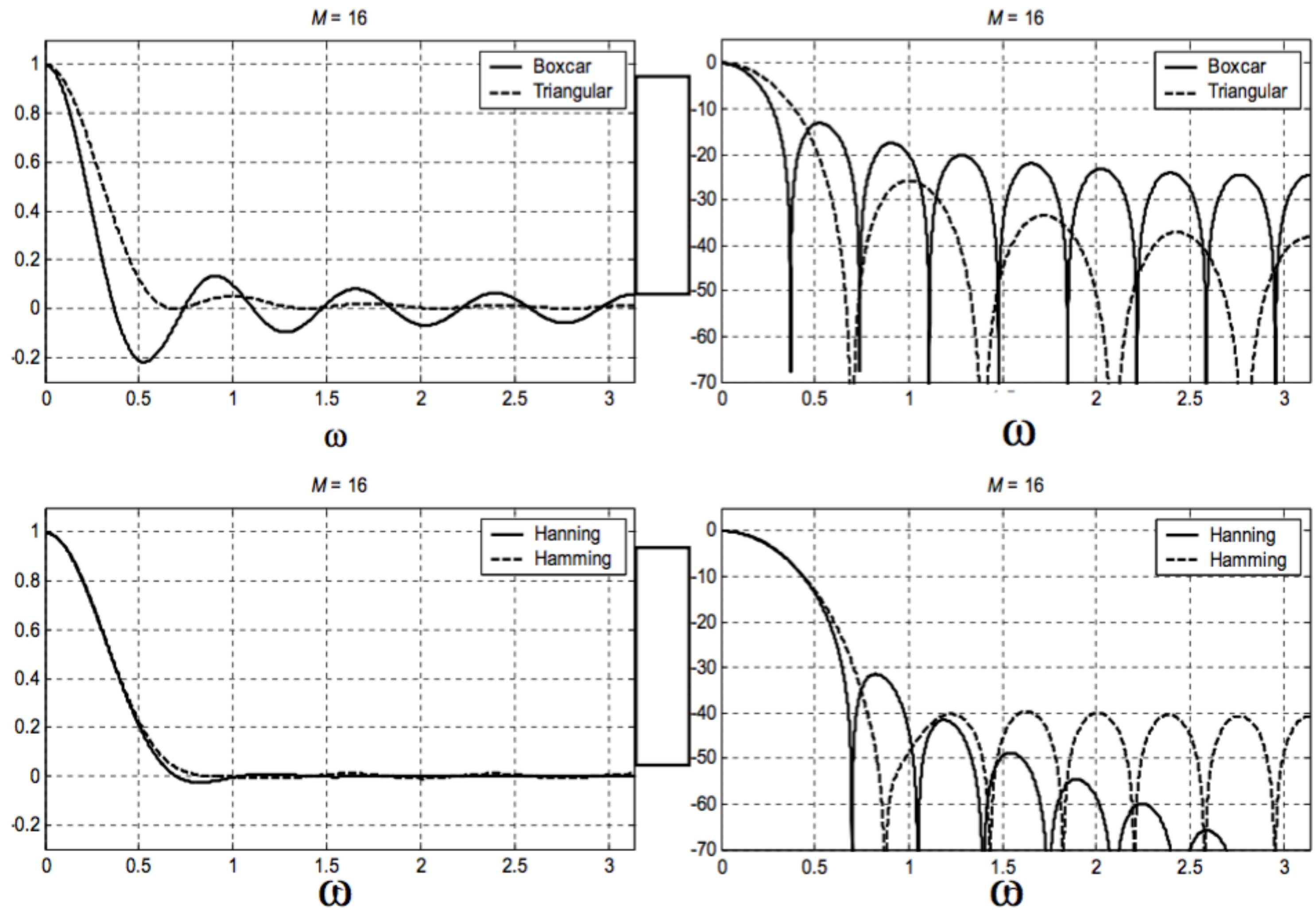
Infinitely narrow "main lobe" width

0 "side lobe" height

Tapered Windows

Name(s)	Definition	MATLAB Command	Graph ($M = 8$)
Hann	$w[n] = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{\pi n}{M/2}\right)\right] & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2 \end{cases}$	`hann(M+1)`	
Hanning	$w[n] = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{\pi n}{M/2+1}\right)\right] & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2 \end{cases}$	`hanning(M+1)`	
Hamming	$w[n] = \begin{cases} 0.54 + 0.46\cos\left(\frac{\pi n}{M/2}\right) & \lvert n\rvert \le M/2 \\ 0 & \lvert n\rvert > M/2 \end{cases}$	`hamming(M+1)`	

Tradeoff – Ripple vs. Transition Width



Commonly Used Windows

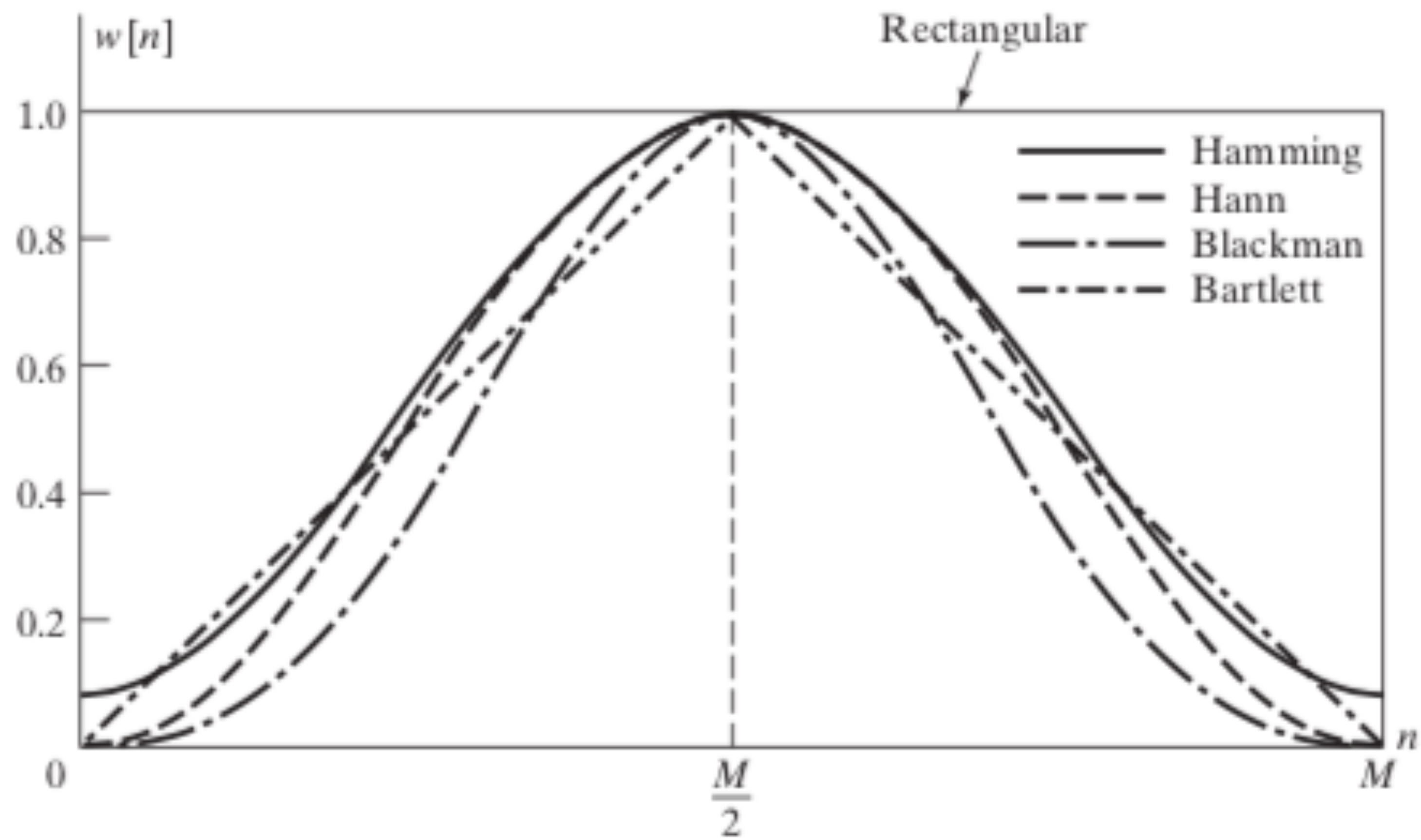
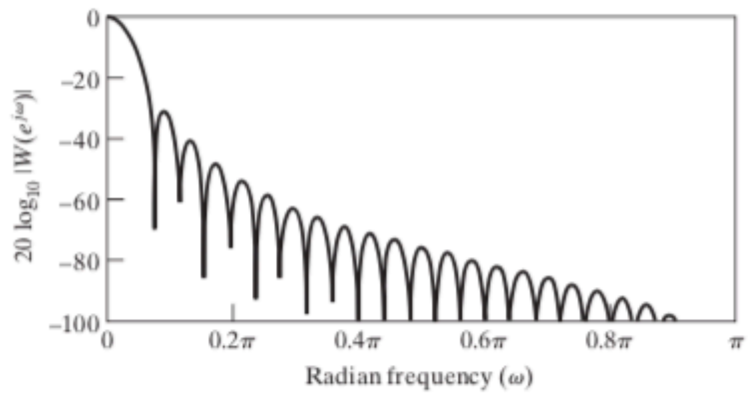


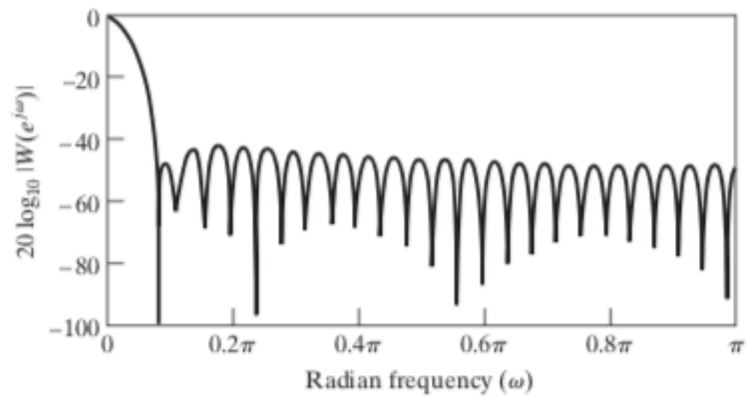
Figure 7.29 Commonly used windows.

Hann



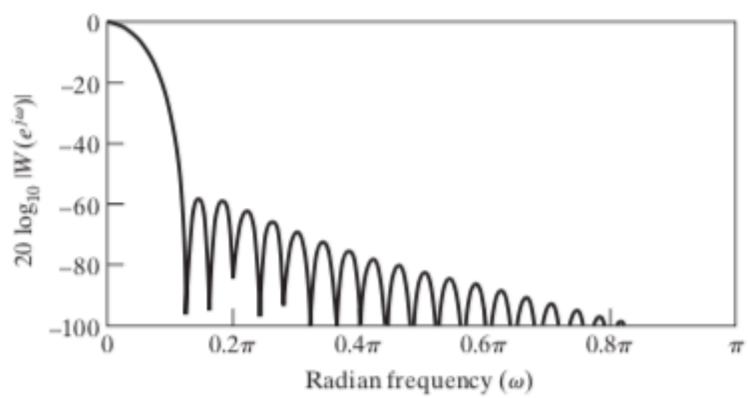
(c)

Hamming



(d)

Blackman



(e)

Kaiser Window

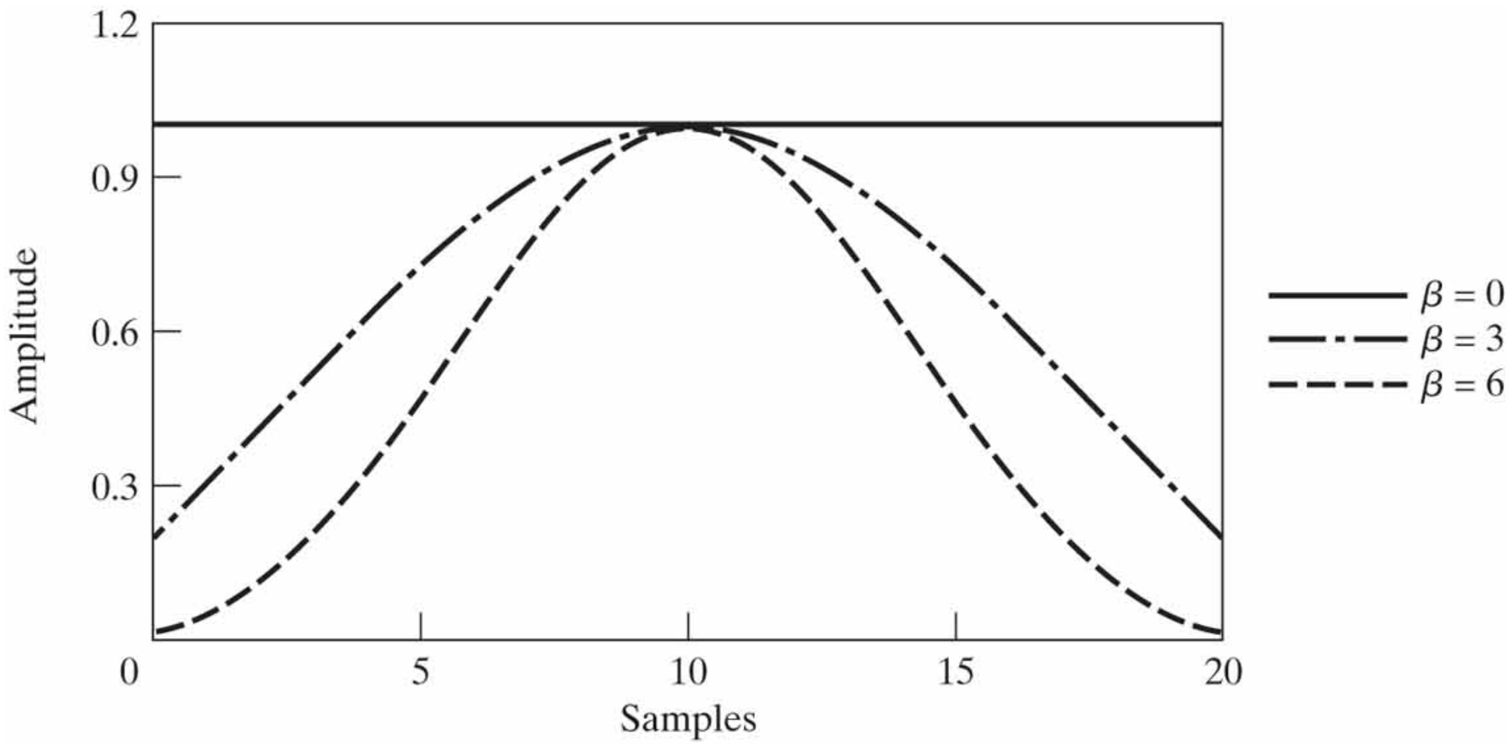
- Near optimal window quantified as the window maximally concentrated around $\omega=0$

$$w[n] = \begin{cases} \dfrac{I_0[\beta(1-[(n-\alpha)/\alpha]^2)^{1/2}]}{I_0(\beta)}, & 0 \le n \le M, \\ 0, & \text{otherwise}, \end{cases}$$

- Two parameters – M and β
- $\alpha=M/2$
- $I_0(x)$ – zeroth order Bessel function of the first kind

Kaiser Window

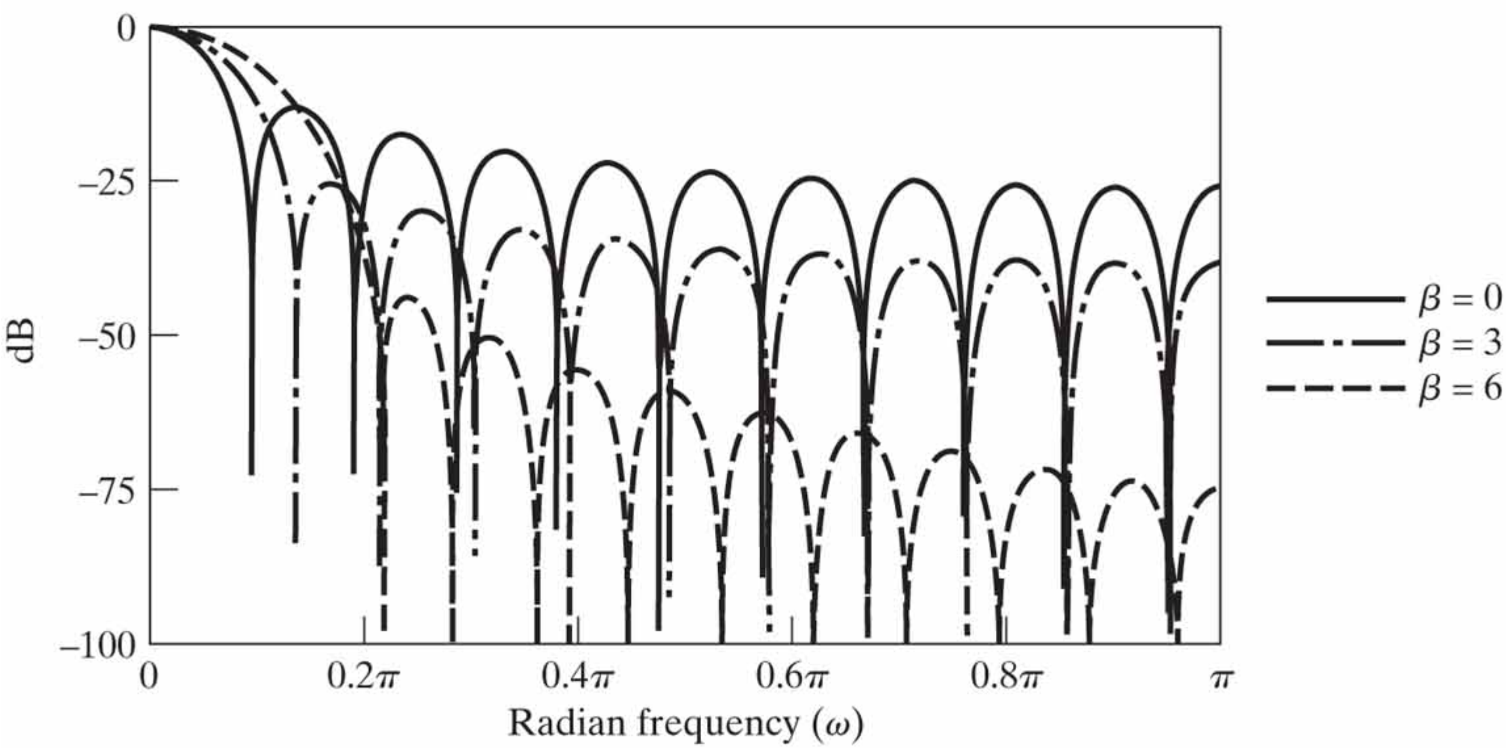
- M=20



(a)

Kaiser Window

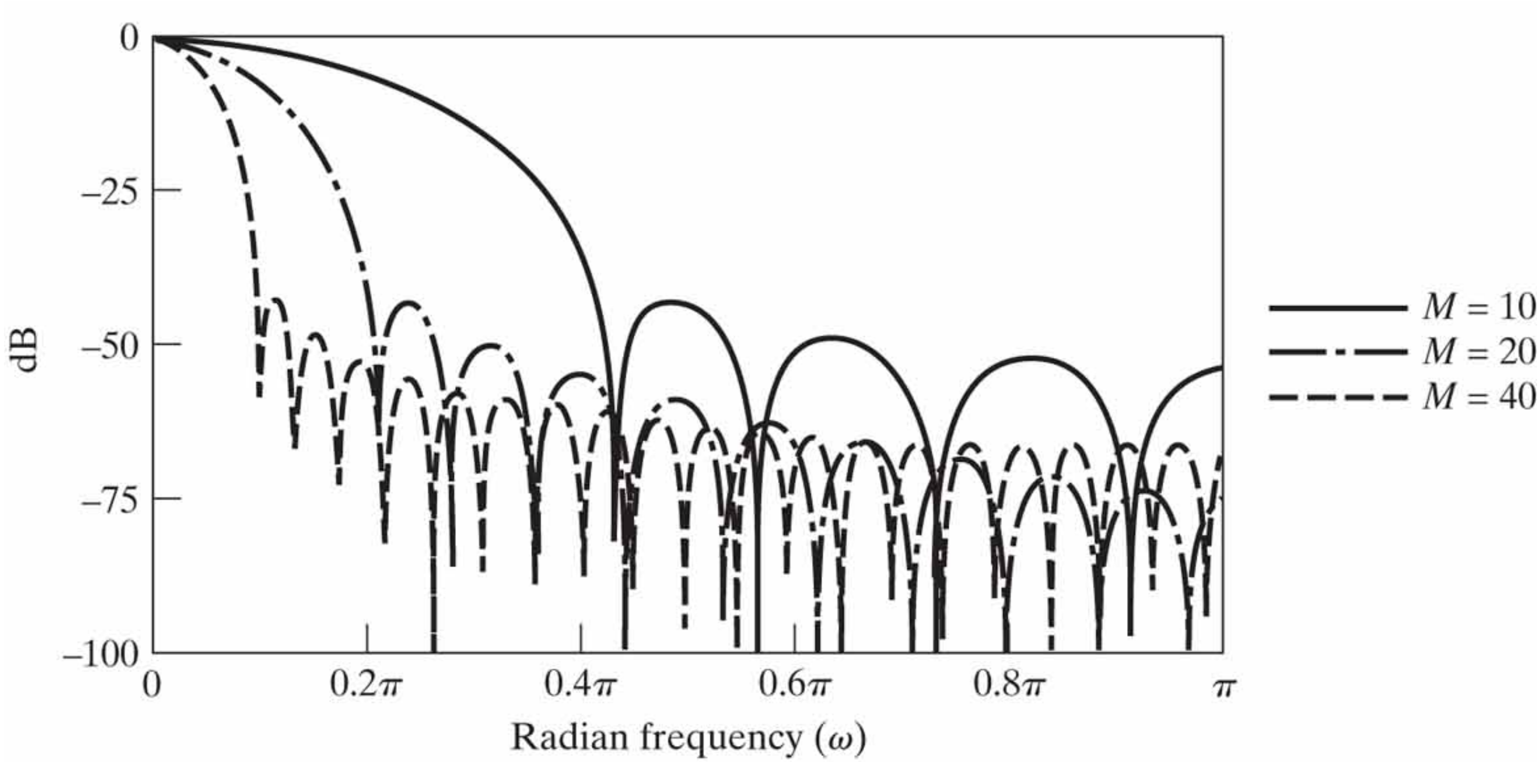
- M=20



(b)

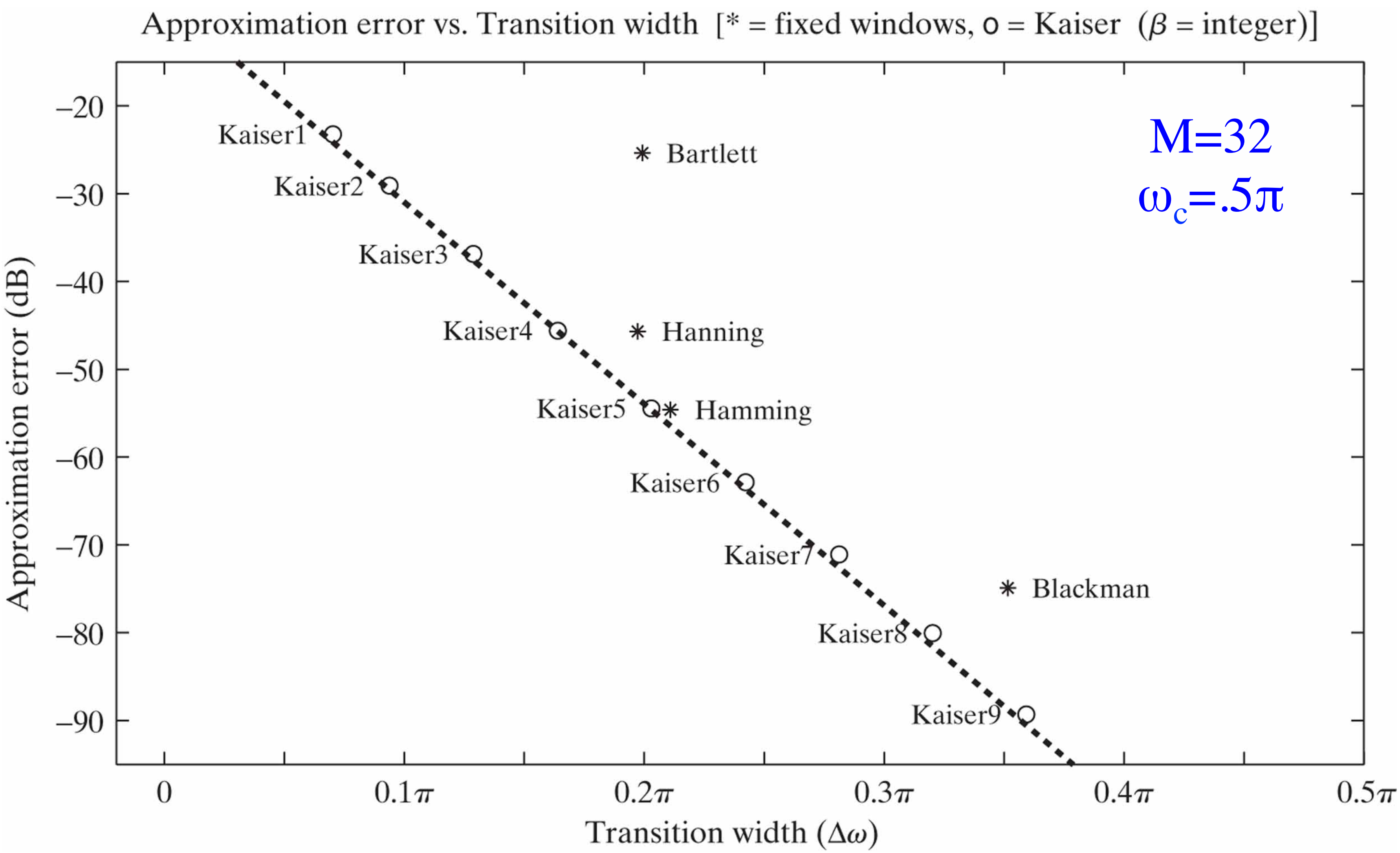
Kaiser Window

- $\beta = 6$



(c)

Approximation Error



FIR Filter Design

- Choose a desired frequency response $H_d(e^{j\omega})$
  - non causal (zero-delay), and infinite imp. response
  - If derived from C.T, choose T and use:

$$H_d(e^{j\omega}) = H_c(j\frac{\Omega}{T})$$

- Window:
  - Length M+1 ⇔ affects transition width
  - Type of window ⇔ transition-width/ ripple

FIR Filter Design

- Choose a desired frequency response $H_d(e^{j\omega})$
  - non causal (zero-delay), and infinite imp. response
  - If derived from C.T, choose T and use:

$$H_d(e^{j\omega}) = H_c(j\frac{\Omega}{T})$$

- Window:
  - Length M+1 ⇔ affects transition width
  - Type of window ⇔ transition-width/ ripple
  - Modulate to shift impulse response
    - Force causality

$$H_d(e^{j\omega})e^{-j\omega\frac{M}{2}}$$

FIR Filter Design

- Determine truncated impulse response $h_1[n]$

$$h_1[n] = \begin{cases} \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{-j\omega\frac{M}{2}}e^{j\omega n} & 0 \le n \le M \\ 0 & \text{otherwise} \end{cases}$$

- Apply window

$$h_w[n] = w[n]h_1[n]$$

- Check:
  - Compute $H_w(e^{j\omega})$, if does not meet specs increase M or change window

Example: FIR Low-Pass Filter Design

$$H_d(e^{j\omega}) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & \text{otherwise} \end{cases}$$

Choose M ⇒ Window length and set

$$H_1(e^{j\omega}) = H_d(e^{j\omega})e^{-j\omega\frac{M}{2}}$$

$$h_1[n] = \begin{cases} \frac{\sin(\omega_c(n-M/2))}{\pi(n-M/2)} & 0 \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\omega_c}{\pi}\text{sinc}(\frac{\omega_c}{\pi}(n-M/2))$$

Example: FIR Low-Pass Filter Design

- The result is a windowed sinc function

$$h_w[n] = w[n]h_1[n]$$

$$h_1[n] = \frac{\omega_c}{\pi}\text{sinc}(\frac{\omega_c}{\pi}(n - M/2))$$

- High Pass Design:
  - Design low pass
  - Transform to $h_w[n](-1)^n = h_w[n]e^{j\pi n}$

- General bandpass
  - Transform to $2h_w[n]\cos(\omega_0 n)$ or $2h_w[n]\sin(\omega_0 n)$

Design through FFT

- To design order M filter:
- Over-Sample/discretize the frequency response at P points where P >> M (P=15M is good)

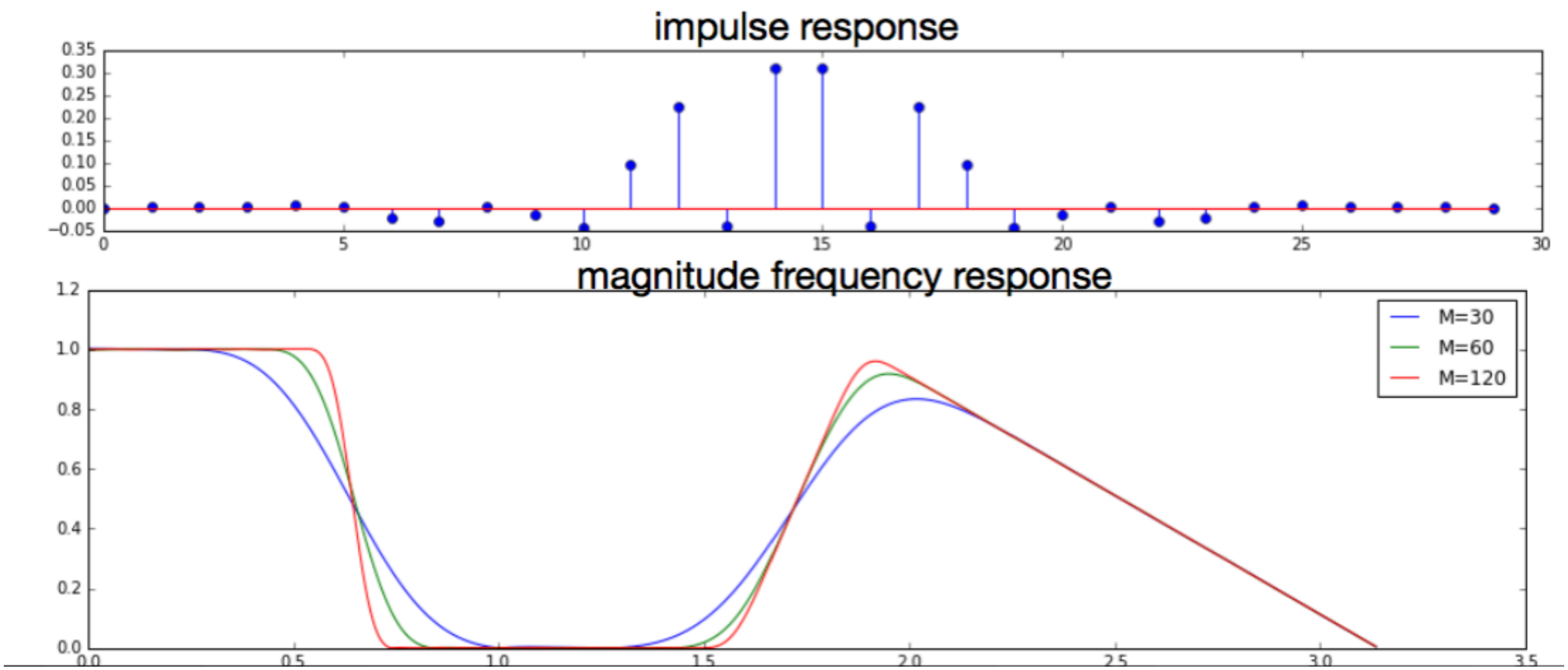
$$H_1(e^{j\omega_k}) = H_d(e^{j\omega_k})e^{-j\omega_k \frac{M}{2}}$$

- Sampled at: $\omega_k = k\frac{2\pi}{P} \qquad |k = [0, \cdots, P-1]$
- Compute $h_1[n] = \text{IDFT}_P(H_1[k])$
- Apply M+1 length window:

$$h_w[n] = w[n]h_1[n]$$

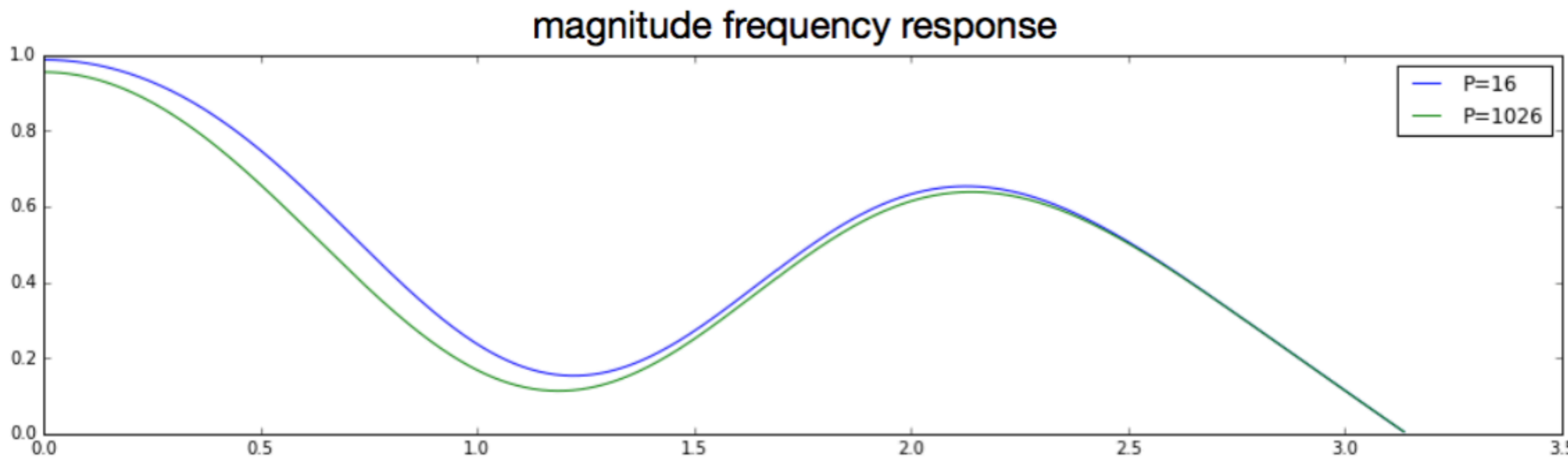
Example

- signal.firwin2(M+1,omega_vec/pi, amp_vec)
- taps1 = signal.firwin2(30, [0.0,0.2,0.21,0.5, 0.6, 1.0], [1.0, 1.0, 0.0,0.0,1.0,0.0])



Example

- For M+1=14
  - P = 16 and P = 1026



Optimal Filter Design

- Window method
  - Design Filters heuristically using windowed sinc functions
  - Choose order and window type
  - Check DTFT to see if filter specs are met
- Optimal design
  - Design a filter h[n] with $H(e^{j\omega})$
  - Approximate $H_d(e^{j\omega})$ with some optimality criteria - or satisfies specs.

Mathematical Optimization

(mathematical) optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- $x = (x_1, \ldots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

optimal solution $x^\star$ has smallest value of f_0 among all vectors that satisfy the constraints

Examples

portfolio optimization

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: overall risk or return variance

device sizing in electronic circuits

- variables: device widths and lengths
- constraints: manufacturing limits, timing requirements, maximum area
- objective: power consumption

data fitting

- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or prediction error

Solving Optimization Problems

general optimization problem

- very difficult to solve
- methods involve some compromise, *e.g.*, very long computation time, or not always finding the solution

exceptions: certain problem classes can be solved efficiently and reliably

- least-squares problems
- linear programming problems
- convex optimization problems

Least-Squares Optimization

$$\text{minimize} \quad \|Ax - b\|_2^2$$

solving least-squares problems

- analytical solution: $x^\star = (A^TA)^{-1}A^Tb$
- reliable and efficient algorithms and software
- computation time proportional to n^2k ($A \in \mathbf{R}^{k\times n}$); less if structured
- a mature technology

using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility (*e.g.*, including weights, adding regularization terms)

Linear Programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, m \end{array}$$

solving linear programs

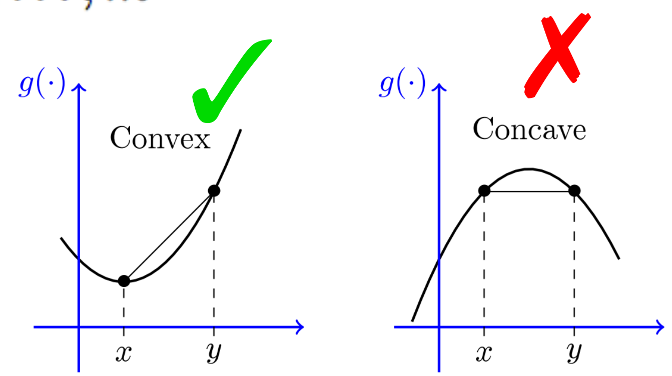
- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m \geq n$; less with structure
- a mature technology

using linear programming

- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs ($e.g.$, problems involving ℓ_1- or ℓ_∞-norms, piecewise-linear functions)

Convex Optimization

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$



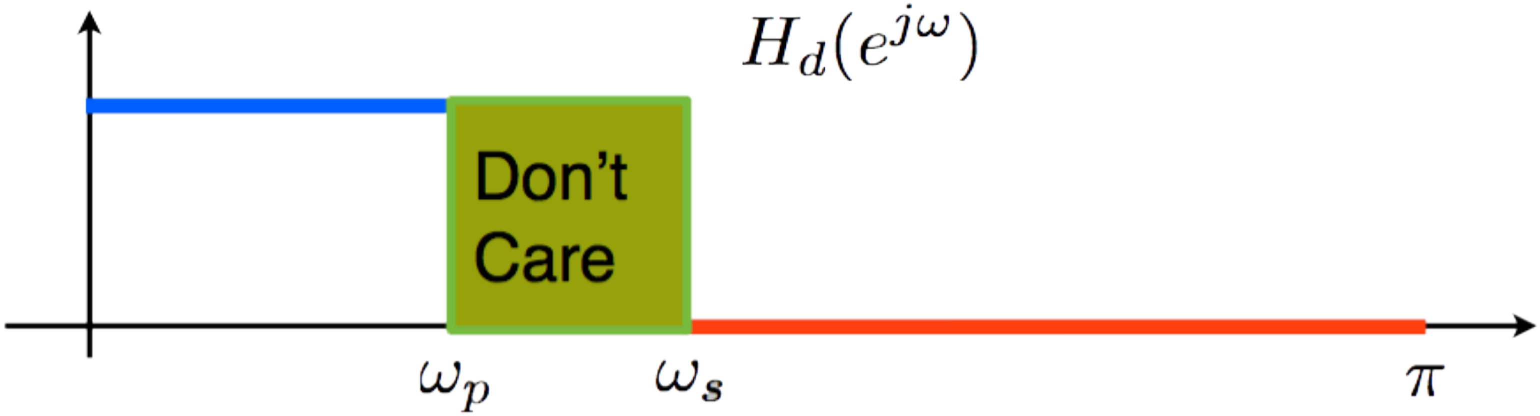
- objective and constraint functions are convex:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

if $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$

- includes least-squares problems and linear programs as special cases

Optimality – Least Squares



- Least Squares:

$$\text{minimize} \quad \int_{\omega \in \text{care}} |H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

Design Through Optimization

- Idea: Sample/discretize the frequency response

$$H(e^{j\omega}) \Rightarrow H(e^{j\omega_k})$$

- Sample points are fixed $\omega_k = k\frac{\pi}{P}$

$$-\pi \leq \omega_1 < \cdots < \omega_p \leq \pi$$

- M+1 is the filter order
- P >> M + 1 (rule of thumb P=15M)
- Yields a (good) approximation of the original problem

Example: Least Squares

- Target: Design M+1= 2N+1 filter
- First design non-causal $\tilde{H}(e^{j\omega})$ and hence $\tilde{h}[n]$

Example: Least Squares

- Target: Design M+1= 2N+1 filter
- First design non-causal $\tilde{H}(e^{j\omega})$ and hence $\tilde{h}[n]$

- Then, shift to make causal

$$h[n] = \tilde{h}[n - M/2]$$

$$H(e^{j\omega}) = e^{-j\frac{M}{2}}\tilde{H}(e^{j\omega})$$

Example: Least Squares

$$\tilde{h} = \left[\tilde{h}[-N], \tilde{h}[-N+1], \cdots, \tilde{h}[N]\right]^T$$

$$b = \left[H_d(e^{j\omega_1}), \cdots, H_d(e^{j\omega_P})\right]^T$$

$$A = \begin{bmatrix} e^{-j\omega_1(-N)} & \ldots & e^{-j\omega_1(+N)} \\ \vdots & & \\ e^{-j\omega_P(-N)} & \ldots & e^{-j\omega_P(+N)} \end{bmatrix}$$

$$\text{argmin}_{\tilde{h}} \quad ||A\tilde{h} - b||^2$$

Least-Squares

$$\text{argmin}_{\tilde{h}} \quad ||A\tilde{h} - b||^2$$

Solution:

$$\tilde{h} = (A^*A)^{-1}A^*b$$

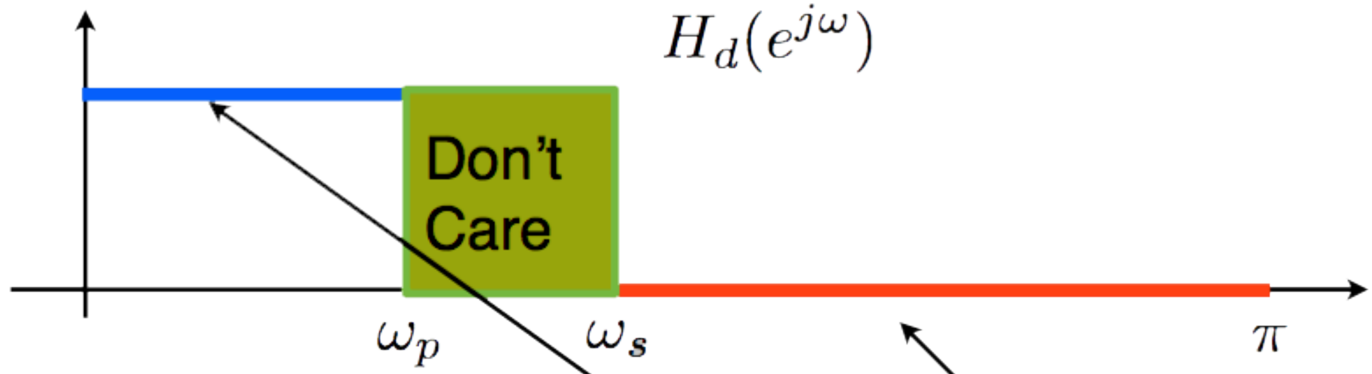
- Result will generally be non-symmetric and complex valued.
- However, if $\tilde{H}(e^{j\omega})$ is real, $\tilde{h}[n]$ should have symmetry!

Design of Linear-Phase L.P Filter

- Suppose:
  - $\tilde{H}(e^{j\omega})$ is real-symmetric
  - M is even (M+1 length)
- Then:
  - $\tilde{h}[n]$ is real-symmetric around midpoint
- So:

$$\begin{aligned}\tilde{H}(e^{j\omega}) &= \tilde{h}[0] + \tilde{h}[1]e^{-j\omega} + \tilde{h}[-1]e^{+j\omega} \\ &\quad + \tilde{h}[2]e^{-j2\omega} + \tilde{h}[-2]e^{+j2\omega} \ldots \\ &= \tilde{h}[0] + 2\cos(\omega)\tilde{h}[1] + 2\cos(2\omega)\tilde{h}[2] + \cdots\end{aligned}$$

Least-Squares Linear Phase Filter



Given M, ω_P, ω_s find the best LS filter:

$$A = \begin{bmatrix} 1 & \cdots & 2\cos(\frac{M}{2}\omega_1) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_p) \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_s) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_P) \end{bmatrix}$$

$$b = [1, 1, \cdots, 1, 0, 0, \cdots, 0]^T$$

Capital P

Least-Squares Linear Phase Filter

Given M, ω_P, ω_s find the best LS filter:

$$A = \begin{bmatrix} 1 & \cdots & 2\cos(\frac{M}{2}\omega_1) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_p) \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_s) \\ \vdots & & \\ 1 & \cdots & 2\cos(\frac{M}{2}\omega_P) \end{bmatrix} \qquad b = [1, 1, \cdots, 1, 0, 0, \cdots, 0]^T$$

$$\tilde{h}_+ = [\tilde{h}[0], \cdots, \tilde{h}[\frac{M}{2}]]^T = (A^*A)^{-1}A^*b$$

$$\tilde{h} = \begin{cases} \tilde{h}_+[n] & n \geq 0 \\ \tilde{h}_+[-n] & n < 0 \end{cases}$$

$$h[n] = \tilde{h}[n - M/2]$$

Extension:

- LS has no preference for pass band or stop band
- Use weighting of LS to change ratio

want to solve the discrete version of:

$$\text{minimize} \quad \int_{-\pi}^{\pi} W(\omega)|H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

where W(ω) is δp in the pass band and δs in stop band

Similarly: W(ω) is 1 in the pass band and δp/δs in stop band

Weighted Least-Squares

$$\text{argmin}_{\tilde{h}_+} \quad (A\tilde{h}_+ - b)^* W^2 (A\tilde{h}_+ - b)$$

Solution:

$$\tilde{h}_+ = (A^* W^2 A)^{-1} W^2 A^* b$$

$$W = \begin{bmatrix} 1 & & & & & & 0 \\ & 1 & & & & & \\ & & \cdots & & & & \\ & & & \frac{\delta_p}{\delta_s} & & & \\ & & & & \cdots & & \\ 0 & & & & & \frac{\delta_p}{\delta_s} & \end{bmatrix}$$

Optimality – min-max

- Chebychev Design (min-max)

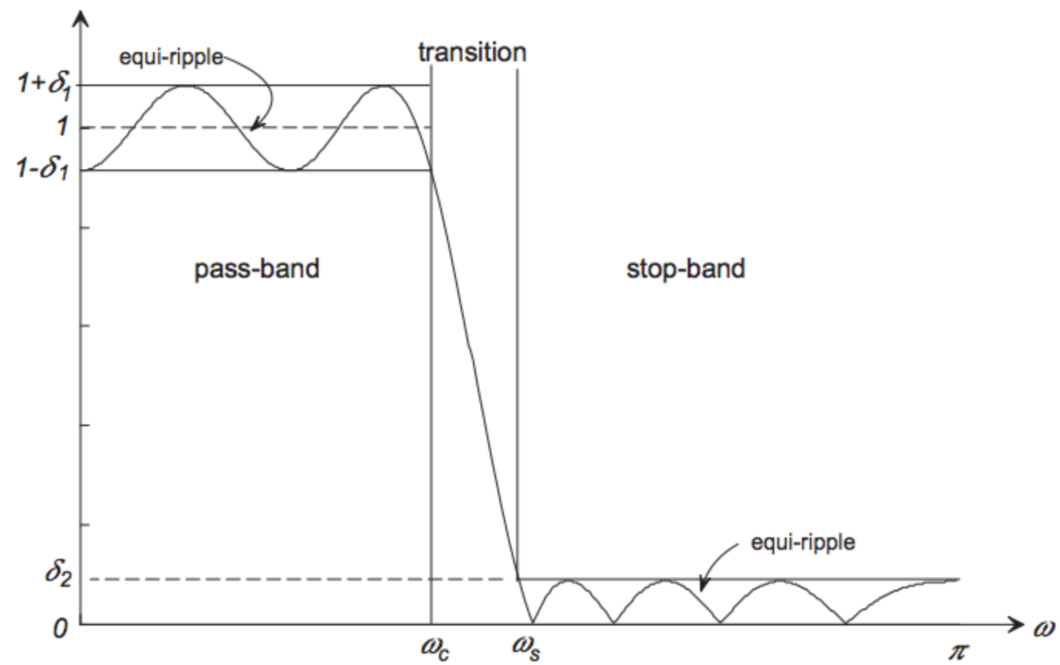
$$\text{minimize}_{\omega \in \text{care}} \quad \max |H(e^{j\omega}) - H_d(e^{j\omega})|$$

  - Parks-McClellan algorithm - equiripple
  - Also known as Remez exchange algorithms (signal.remez)
  - Can also use convex optimization

Parks-McClellan

- Allows for multiple pass- and stop-bands.
- Is an equi-ripple design in the pass- and stop-bands, but allows independent weighting of the ripple in each band.
- Allows specification of the band edges.

Parks-McClellan: LP Filter



- For the low-pass filter shown above the specifications are

$$
\begin{array}{rcccl}
1-\delta_1 & < & H(e^{j\omega}) & < & 1+\delta_1 \quad \text{in the pass-band } 0 < \omega \leq \omega_c \\
-\delta_2 & < & H(e^{j\omega}) & < & \delta_2 \quad \text{in the stop-band } \omega_s < \omega \leq \pi.
\end{array}
$$

Parks-McClellan: LP Filter

- Need to determine M+1 (length of the filter) and the filter coefficients $\{h_n\}$

Parks-McClellan: LP Filter

- Need to determine M+1 (length of the filter) and the filter coefficients $\{h_n\}$

- If we assume M even and even symmetry FIR filter (Type I), then

$$A_e(e^{j\omega}) = h_e[0] + \sum_{n=1}^{L} 2h_e[n]\cos(\omega n).$$

$$H(e^{j\omega}) = A_e(e^{j\omega})e^{-j\omega M/2}.$$

Parks-McClellan: LP Filter

- Define approximation error function

$$E(\omega) = W(\omega)[H_d(e^{j\omega}) - A_e(e^{j\omega})],$$

$$W(e^{j\omega}) = \begin{cases} \delta_2/\delta_1 & \text{in the pass-band} \\ 1 & \text{in the stop-band} \\ 0 & \text{in the transition band.} \end{cases}$$

Parks-McClellan: LP Filter

- Define approximation error function

$$E(\omega) = W(\omega)[H_d(e^{j\omega}) - A_e(e^{j\omega})],$$

- Apply min-max or Chebyshev criteria

$$\min_{\{h_e[n]:0\le n\le L\}} \left(\max_{\omega\in F} |E(\omega)| \right),$$

Min-Max Filter Design

- Constraints:
  - min-max pass-band ripple

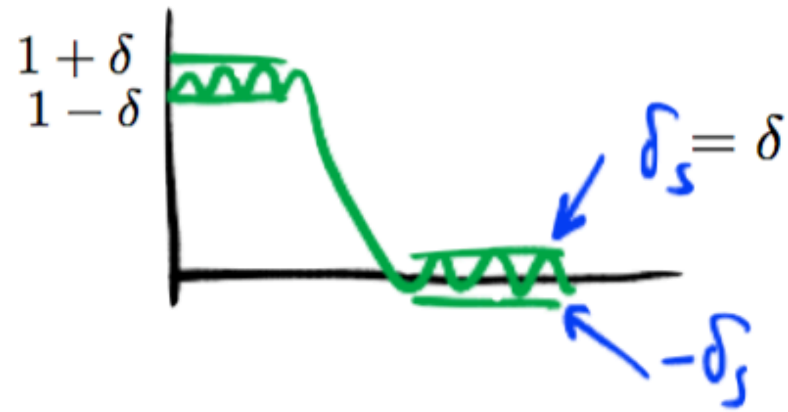
$$1 - \delta_p \leq |H(e^{j\omega})| \leq 1 + \delta_p, \qquad 0 \leq w \leq \omega_p$$

  - min-max stop-band ripple

$$|H(e^{j\omega})| \leq \delta_s, \qquad \omega_s \leq w \leq \pi$$

Min-Max Ripple Design

- Given ω_p, ω_s, M, find δ , $\tilde{h}_+$



$$
\begin{aligned}
&\text{minimize} \quad && \delta \\
&\text{Subject to :} && \\
& && 1-\delta \le \tilde{H}(e^{j\omega_k}) \le 1+\delta \qquad && 0 \le \omega_k \le \omega_p \\
& && -\delta \le \tilde{H}(e^{j\omega_k}) \le \delta && \omega_s \le \omega_k \le \pi \\
& && \delta > 0 &&
\end{aligned}
$$

- Formulation is a linear program with solution δ , $\tilde{h}_+$
- A well studied class of problems with good solvers

Min-Max Ripple via LP

$$\text{minimize} \quad \delta$$

$$\text{subject to :}$$

$$1 - \delta \preceq A_p \tilde{h}_+ \preceq 1 + \delta$$

$$-\delta \preceq A_s \tilde{h}_+ \preceq \delta$$

$$\delta > 0$$

$$A_p = \begin{bmatrix} 1 & 2\cos(\omega_1) & \cdots & 2\cos(\frac{M}{2}\omega_1) \\ & \vdots & & \\ 1 & 2\cos(\omega_p) & \cdots & 2\cos(\frac{M}{2}\omega_p) \end{bmatrix}$$

$$A_s = \begin{bmatrix} 1 & 2\cos(\omega_s) & \cdots & 2\cos(\frac{M}{2}\omega_s) \\ & \vdots & & \\ 1 & 2\cos(\omega_P) & \cdots & 2\cos(\frac{M}{2}\omega_P) \end{bmatrix}$$

capital P

Parks-McClellan

- The method is based on reformulating the problem as one in polynomial approximation, using Chebyshev polynomials

$$A_e(e^{j\omega}) = \sum_{k=0}^{L} a_k (\cos\omega)^k,$$

$$A_e(e^{j\omega}) = P(x)|_{x=\cos\omega}, \qquad P(x) = \sum_{k=0}^{L} a_k x^k.$$

Parks-McClellan – Alternation Theorem

- The algorithm uses Chebyshev's alternation theorem to recognize the optimal solution.

Define the error $E(x)$ as above, namely

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

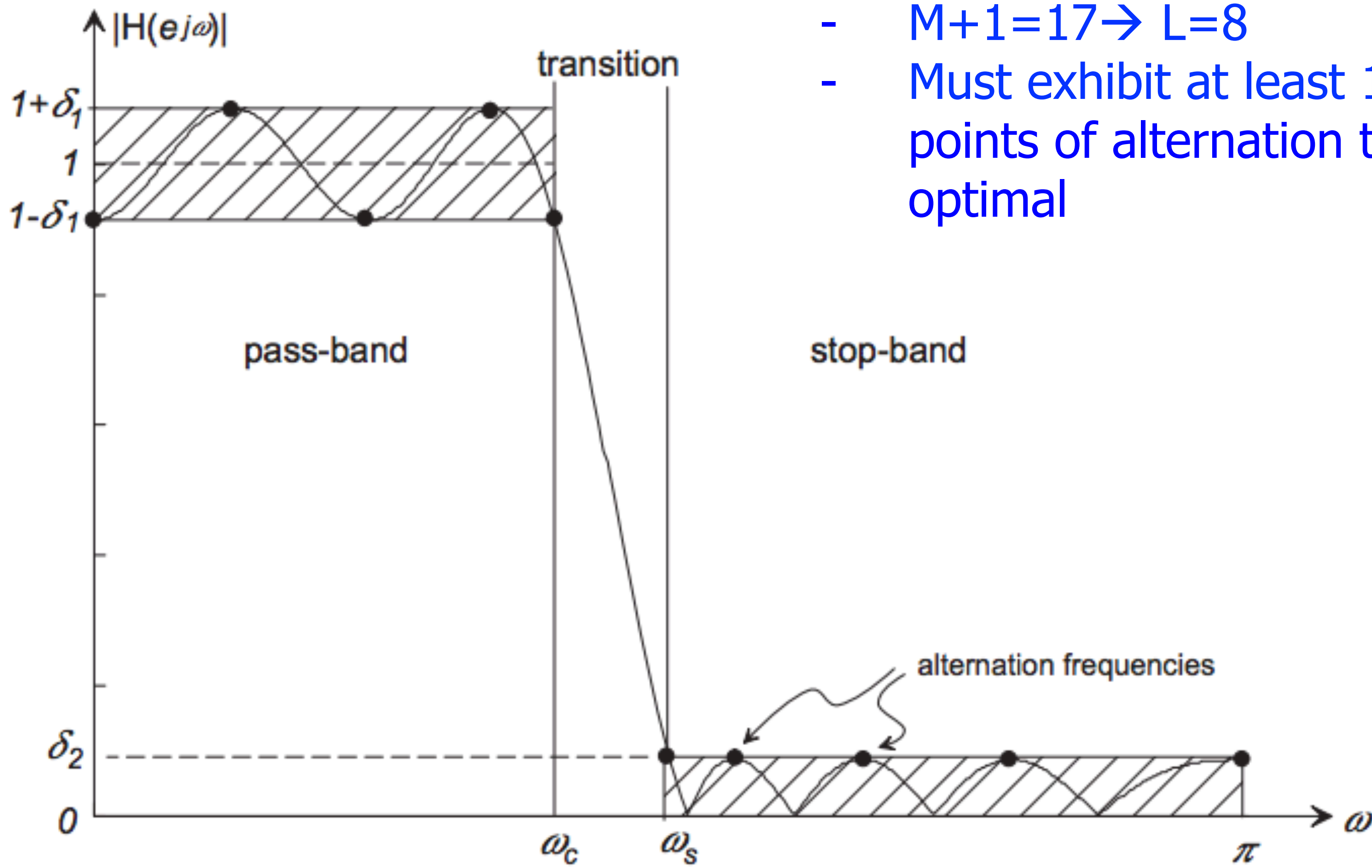
and the maximum error as

$$\|E(e^{j\omega})\|_\infty = \text{argmax}_{x\in\Omega}\left|E(e^{j\omega})\right|$$

A necessary and sufficient condition that $H(e^{j\omega})$ is the unique Lth-order polynomial minimizing $\|E(e^{j\omega})\|_\infty$ is that $E(e^{j\omega})$ exhibit at least $L+2$ extremal frequencies, or "alternations", that is there must exist at least $L+2$ values of ω, $\omega_k \in \Omega$, $k = [0, 1, \ldots, L+1]$, such that $\omega_0 < \omega_1 < \ldots < \omega_{L+1}$, and such that

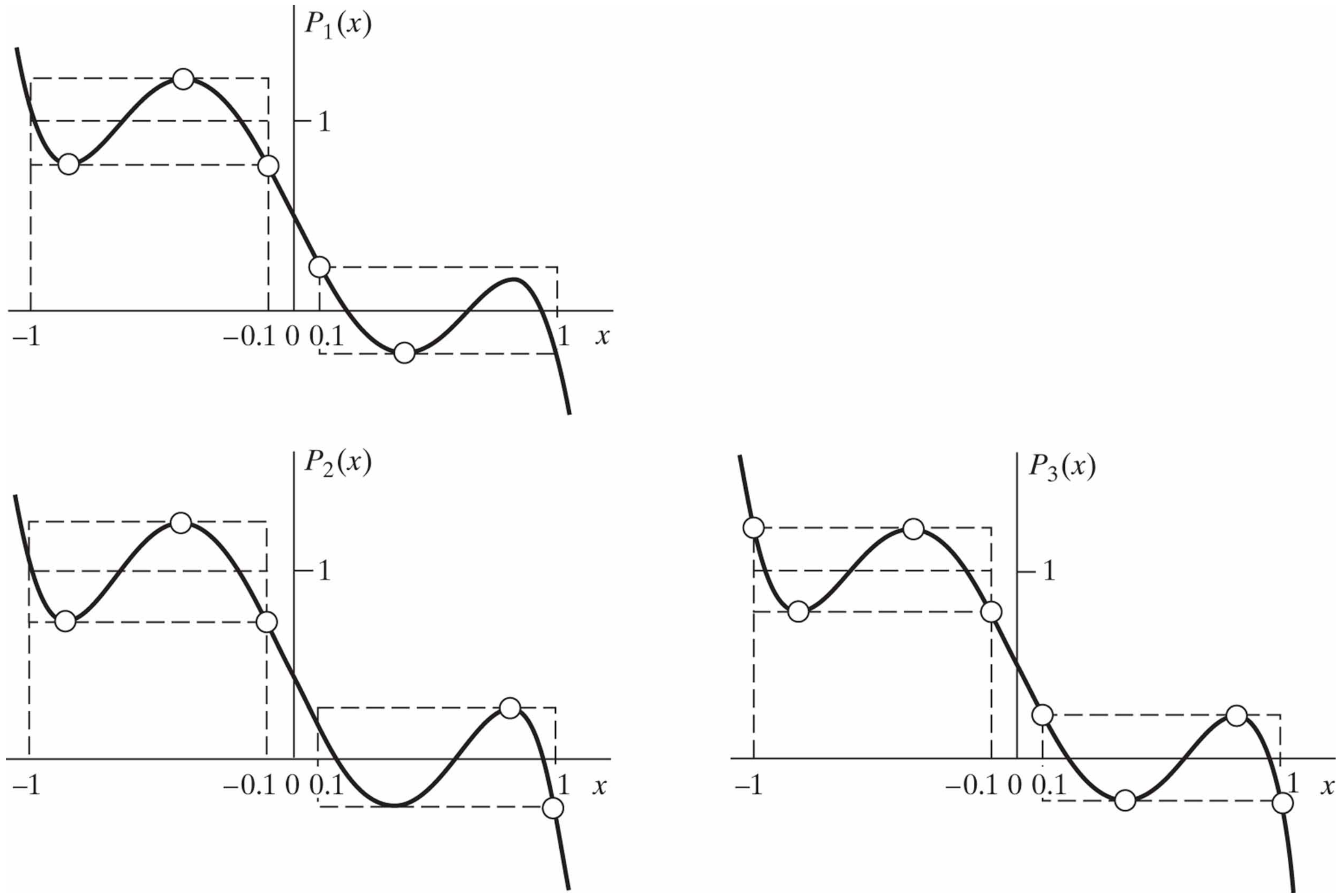
$$E(e^{j\omega_k}) = -E(e^{j\omega_{k+1}}) = \pm\left(\|E(e^{j\omega})\|_\infty\right).$$

Parks-McClellan – Alternation Theorem

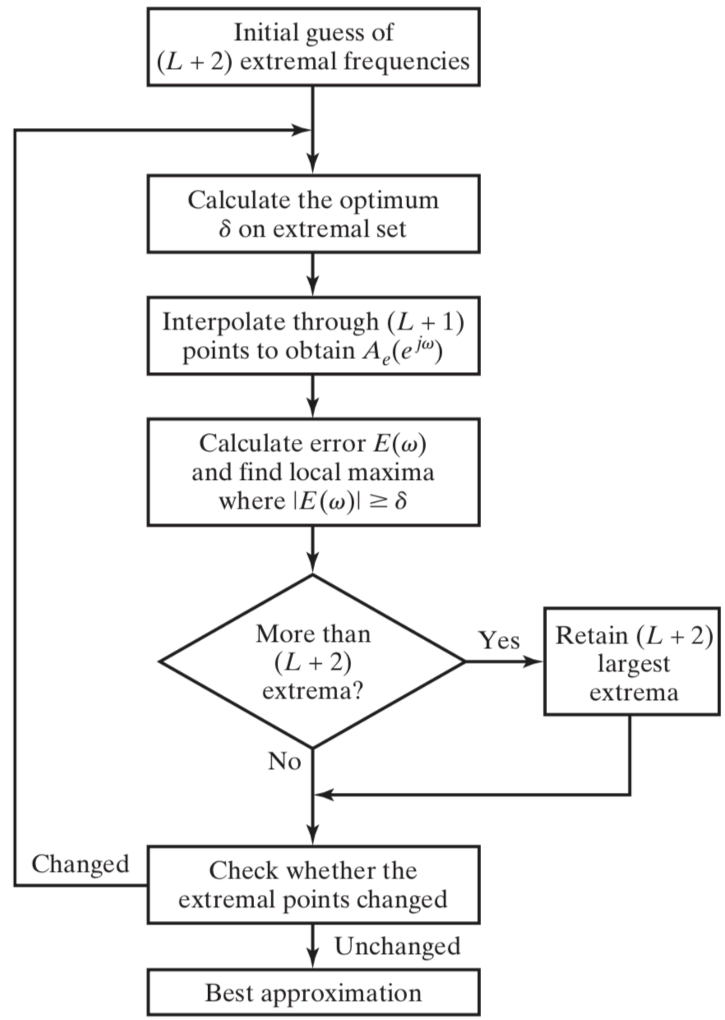


- M+1=17→ L=8
- Must exhibit at least 10 points of alternation to be optimal

Alternation Theorem Example – 5th order



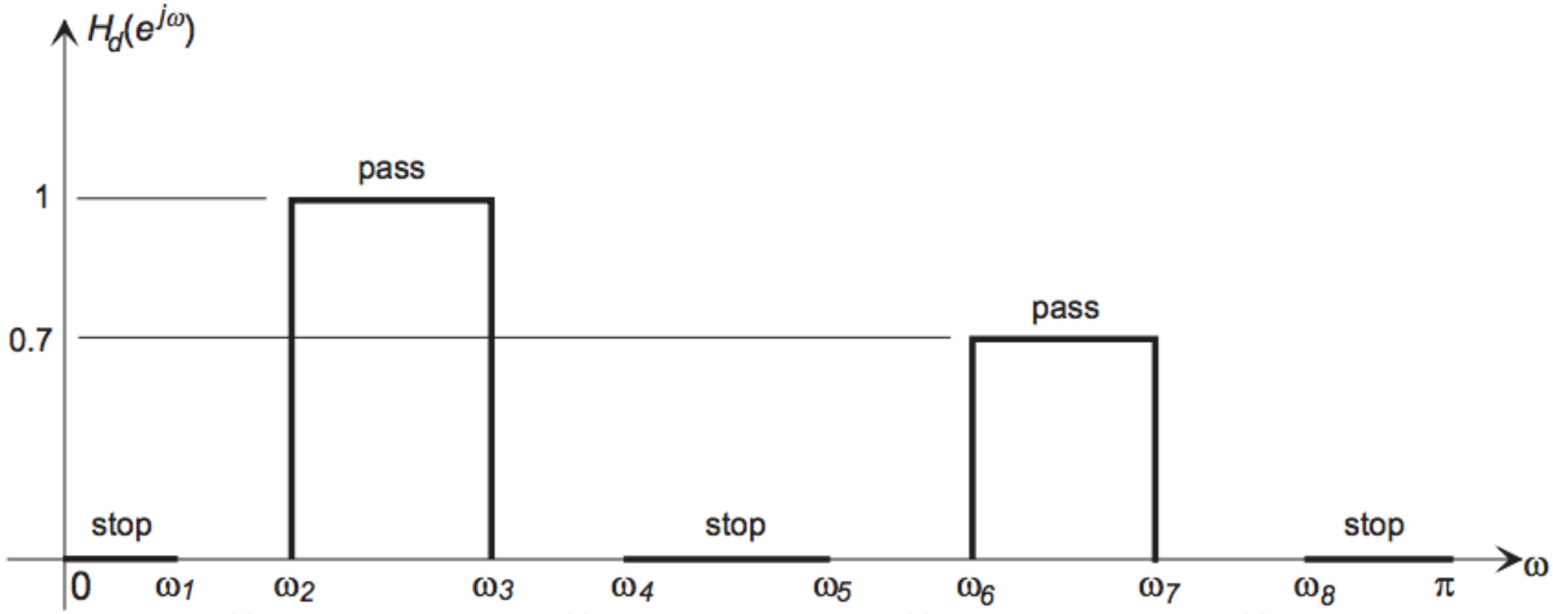
Parks–McClellan algorithm



MATLAB Parks-McClellan Function

- `b = firpm(M,F,A,W)`
  - `b` is the array of filter coefficients (impulse response)
  - `M` is the filter order (`M+1` is the length of the filter),
  - `F` is a vector of band edge frequencies in ascending order
  - `A` is a set of filter gains at the band edges
  - `W` is an optional set of relative weights to be applied to each of the bands

MATLAB Parks-McClellan Function



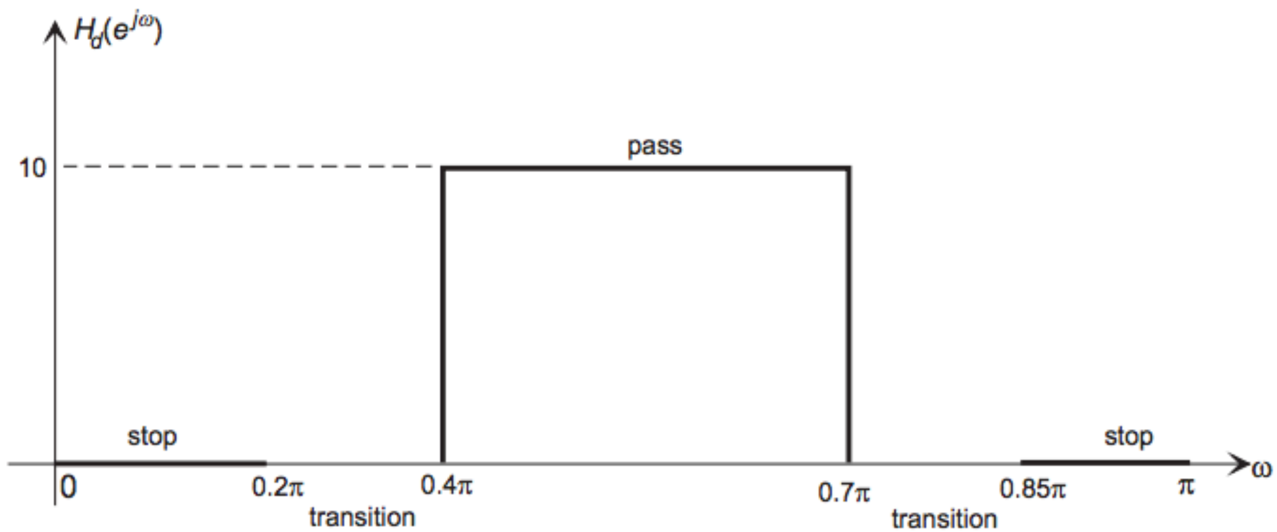
$$\mathtt{F} = [0 \quad \omega_1 \quad \omega_2 \quad \omega_3 \quad \omega_4 \quad \omega_5 \quad \omega_6 \quad \omega_7 \quad \omega_8 \quad 1]$$

$$\mathtt{A} = [0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0.7 \quad 0.7 \quad 0 \quad 0]$$

$$\mathtt{W} = [10 \quad 1 \quad 10 \quad 1 \quad 10]$$

MATLAB Example

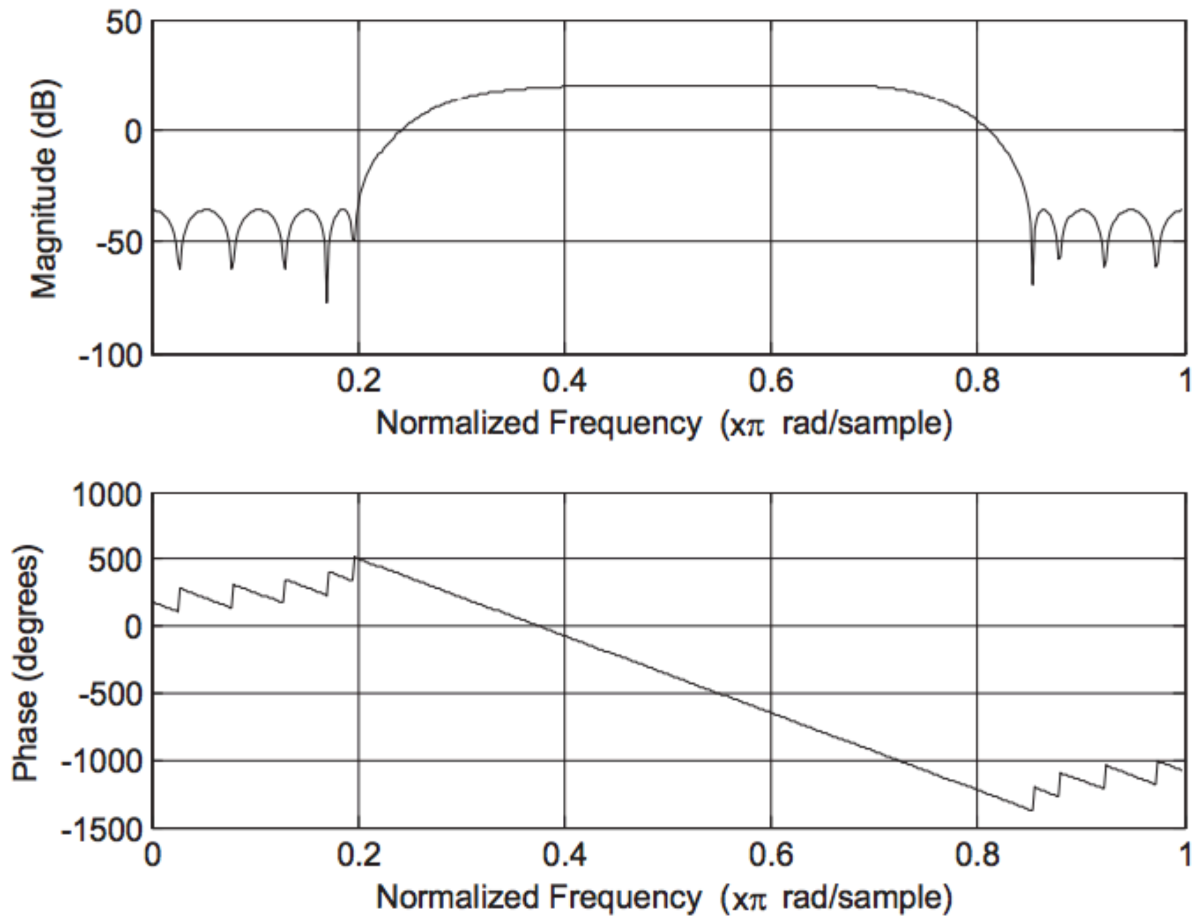
- Design a 33 length PM band-pass filter and weight the stop-band ripple 10x more than the pass-band ripple




```
h=firpm(32,[0 0.2 0.4 0.7 0.85 1],[0 0 10 10 0 0],[10 1 10])
freqz(h,1)
```

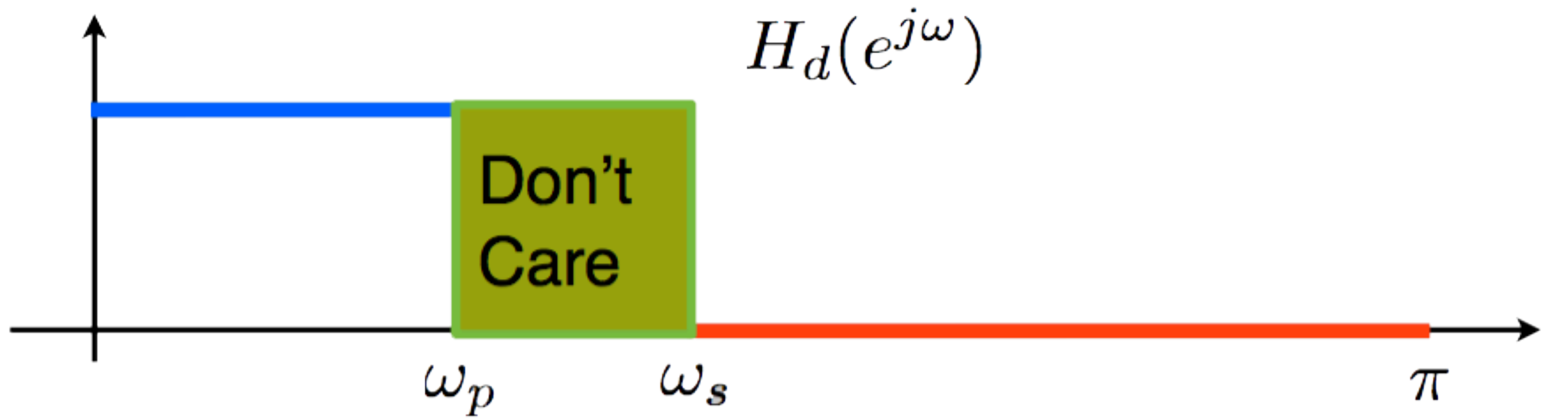
MATLAB Example

- Des … op-
  ban




```
h=firp…                          1 10])
freqz(…
```

Optimality – Least Squares



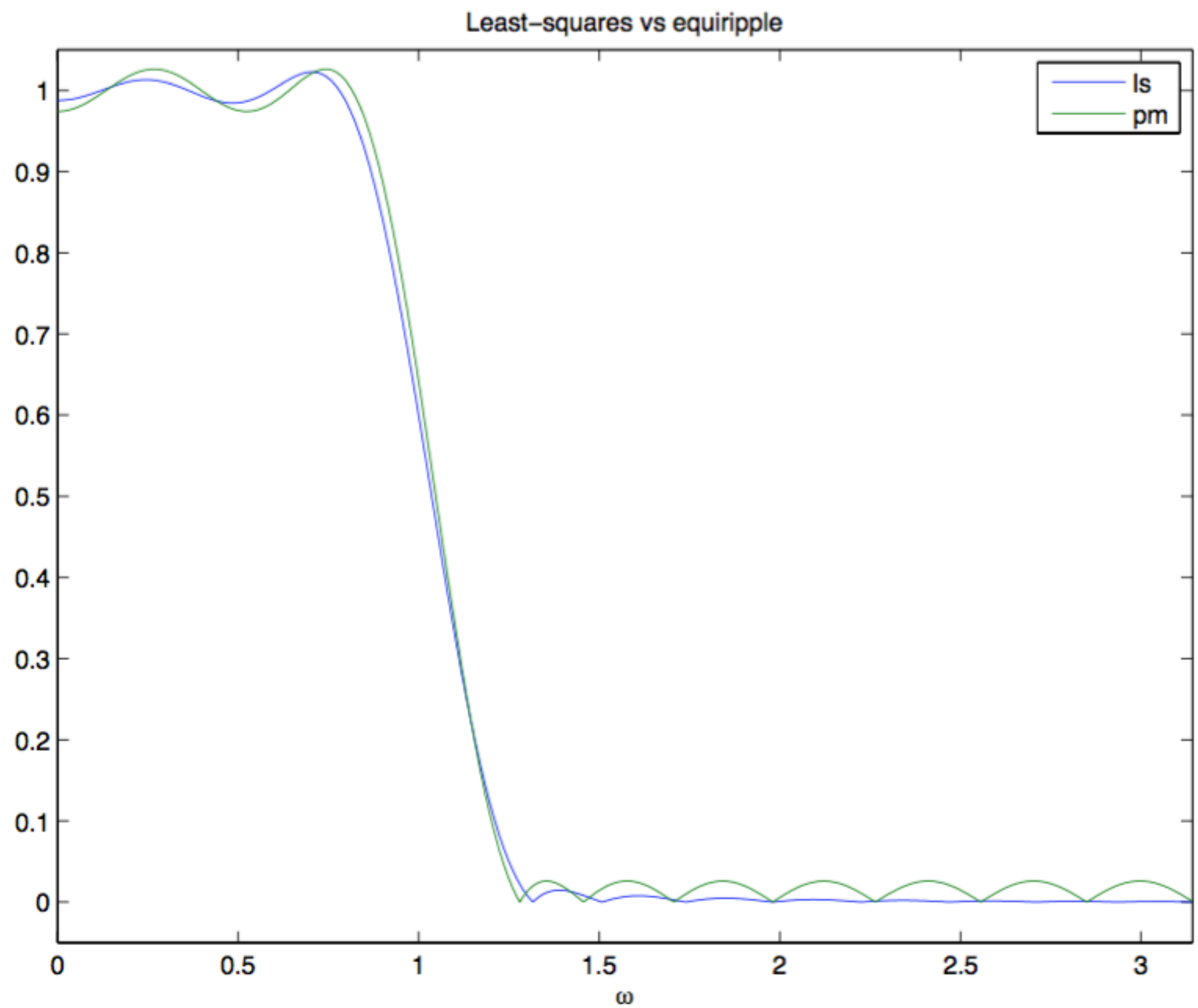
- Least Squares:

$$\text{minimize} \quad \int_{\omega \in \text{care}} |H(e^{j\omega}) - H_d(e^{j\omega})|^2 d\omega$$

- Parks-McClellan

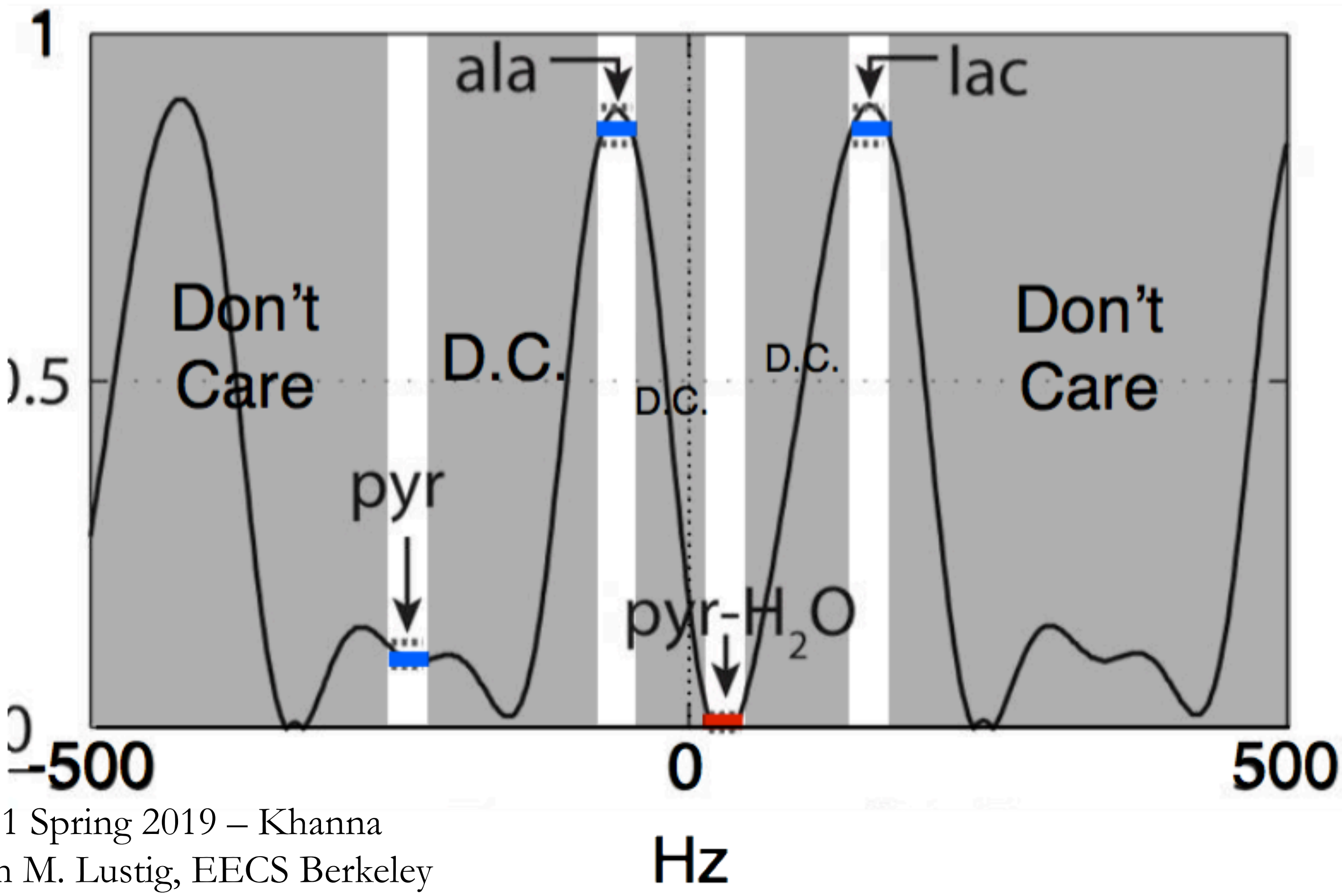
$$\min_{\{h_e[n]:0 \leq n \leq L\}} \left(\max_{\omega \in F} |E(\omega)| \right),$$

Least-Squares vs. Min-Max



Example of Complex Filter

- Larson et. al, "Multiband Excitation Pulses for Hyperpolarized 13C Dynamic Chemical Shift Imaging" JMR 2008;194(1):121-127
- Need to design length 11 filter with following frequency response:

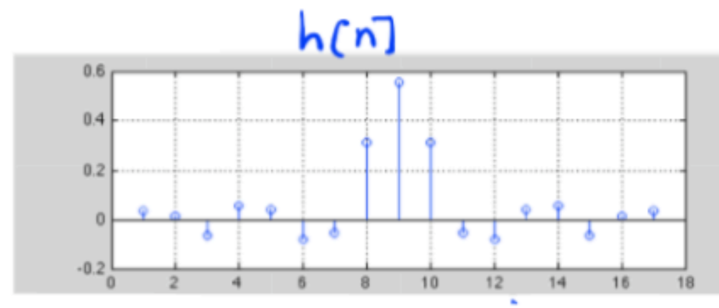


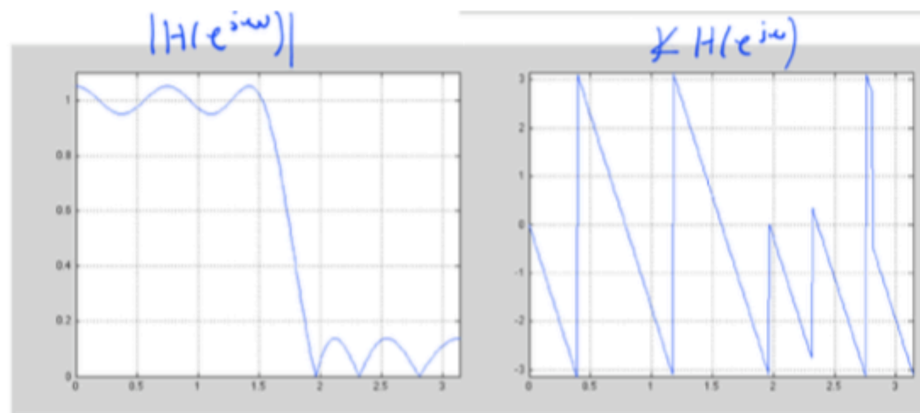
Convex Optimization

- Many tools and Solvers
- Tools:
  - CVX (Matlab) http://cvxr.com/cvx/
  - CVXOPT, CVXMOD (Python)
- Engines:
  - Sedumi (Free)
  - MOSEK (commercial)

Using CVX (in Matlab)

```
M = 16;
wp = 0.5*pi;
ws = 0.6*pi;
MM = 15*M;
w = linspace(0,pi,MM);

idxp = find(w <=wp);
idxs = find(w >=ws);

Ap = [ones(length(idxp),1) 2*cos(kron(w(idxp)',
[1:M/2]))];
As = [ones(length(idxs),1) 2*cos(kron(w(idxs)',
[1:M/2]))];

% optimization
cvx_begin
   variable hh(M/2+1,1);
   variable d(1,1);

   minimize(d)
   subject to
      Ap*hh <=1+d;
      Ap*hh >=1-d;
      As*hh <  d;
      As*hh > -d;
      ds>0;
cvx_end
h = [hh(end:-1:1) ; hh(2:end)];
```





Admin

- HW 6
  - Due Sunday 3/29