EE123
Digital Signal Processing

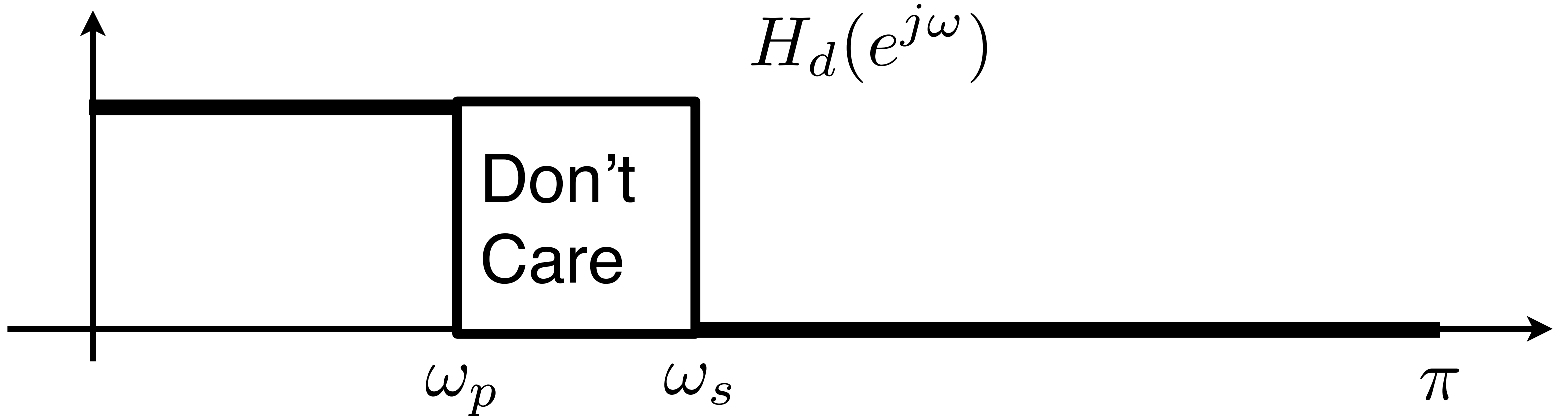
Lecture 21
Optimal Filter Design

M. Lustig, EECS UC Berkeley

Optimal Filter Design

- Window method
  - Design Filters heuristically using windowed sinc functions
- 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.

Optimality



- Least Squares:

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

Variation: weighted least-squares

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

- Chebychev Design (min-max)

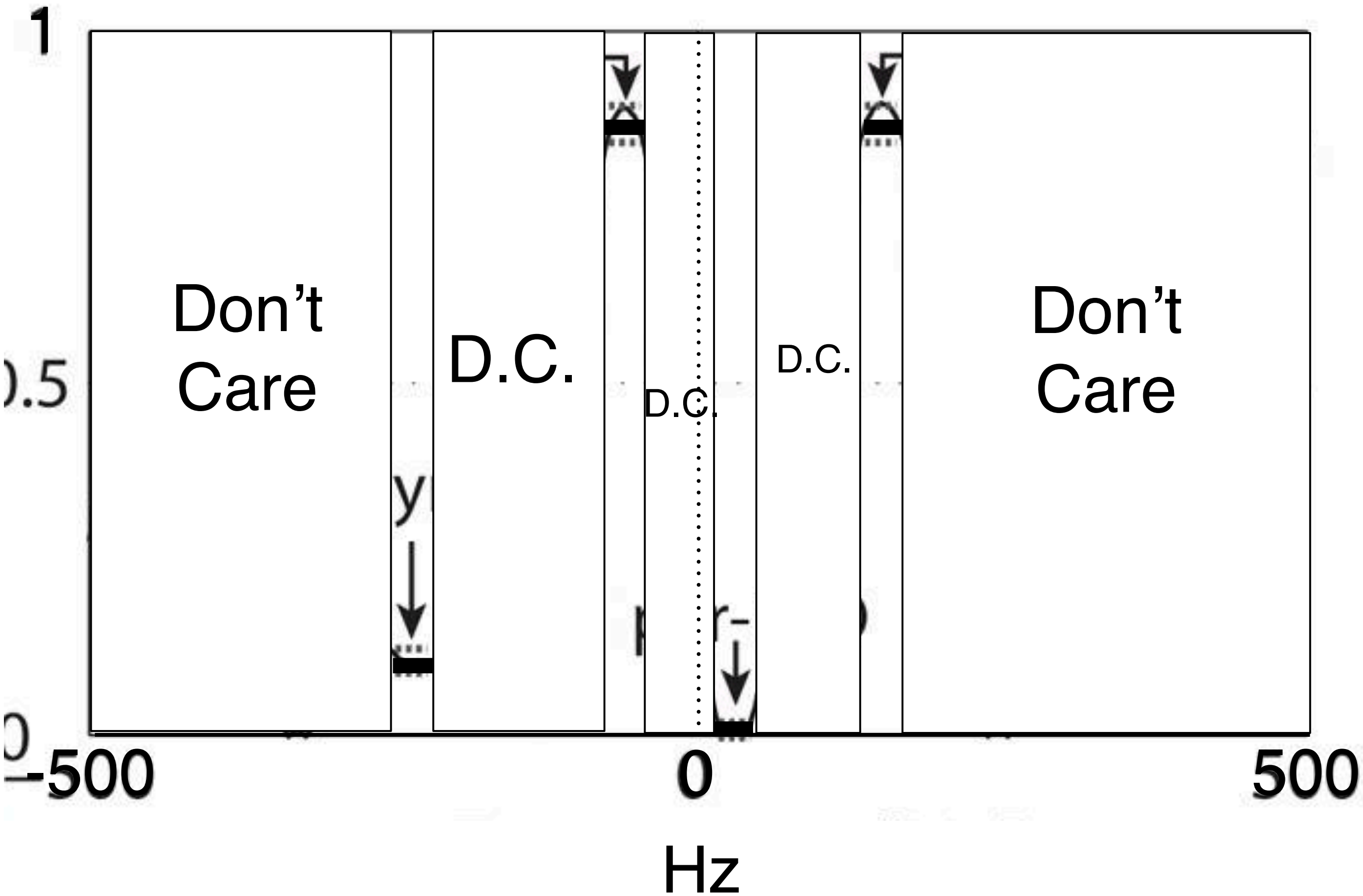
$$\text{minimize}_{\omega \in \text{care}} \quad \max \left| H(e^{j\omega}) - H_d(e^{j\omega}) \right|$$

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

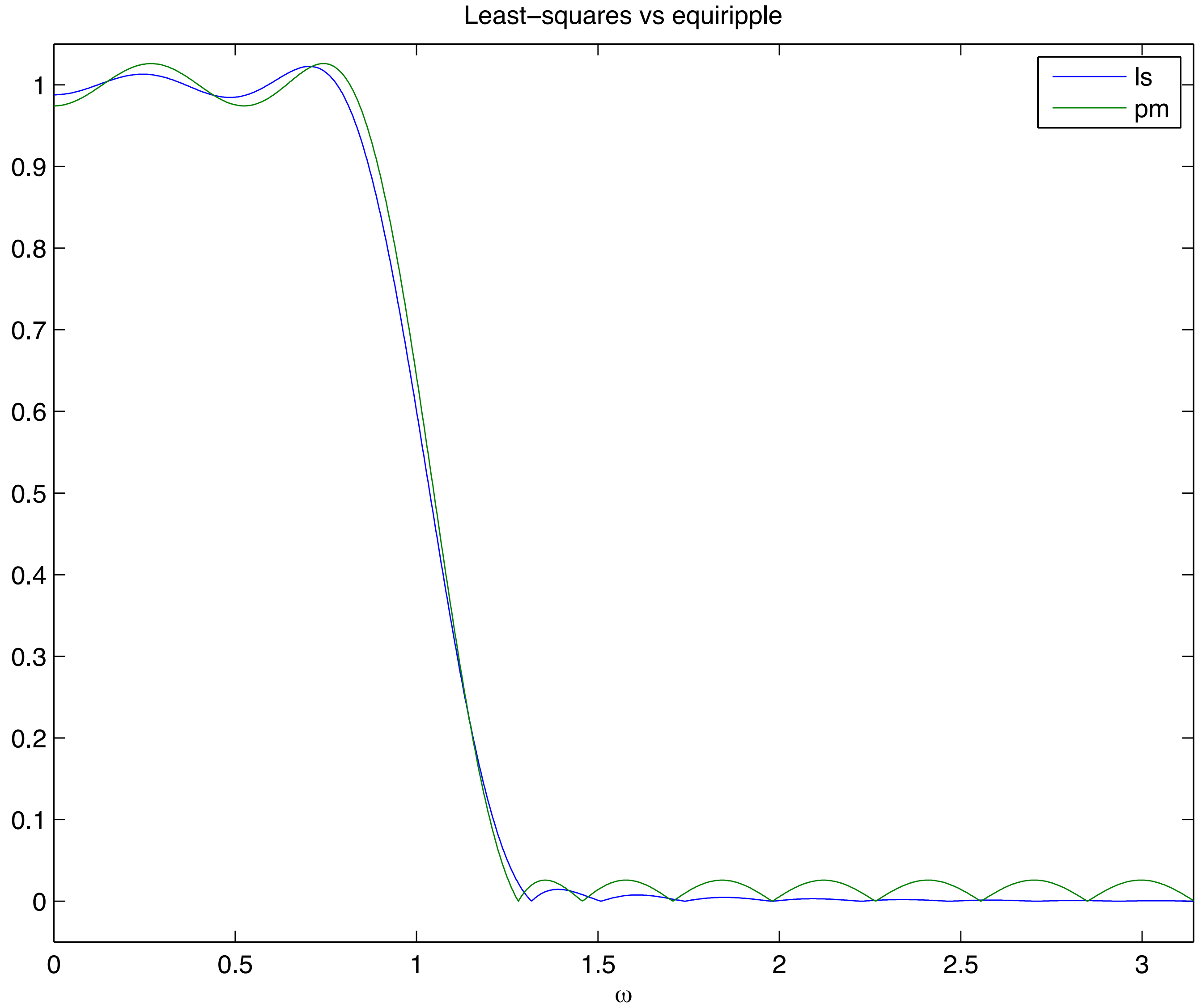
Example of Complex Filter

Larson et. al, “Multiband Excitation Pulses for Hyperpolarized 13C Dynamic Chemical Shift Imaging” JMR 2008;194(1):121-127

Need to design 11 taps filter with following frequency response:



Least-Squares v.s. Min-Max



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 $\tilde{h}[n]$
- 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

- Matrix formulation:

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

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

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

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

Least Squares

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

Solution:

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

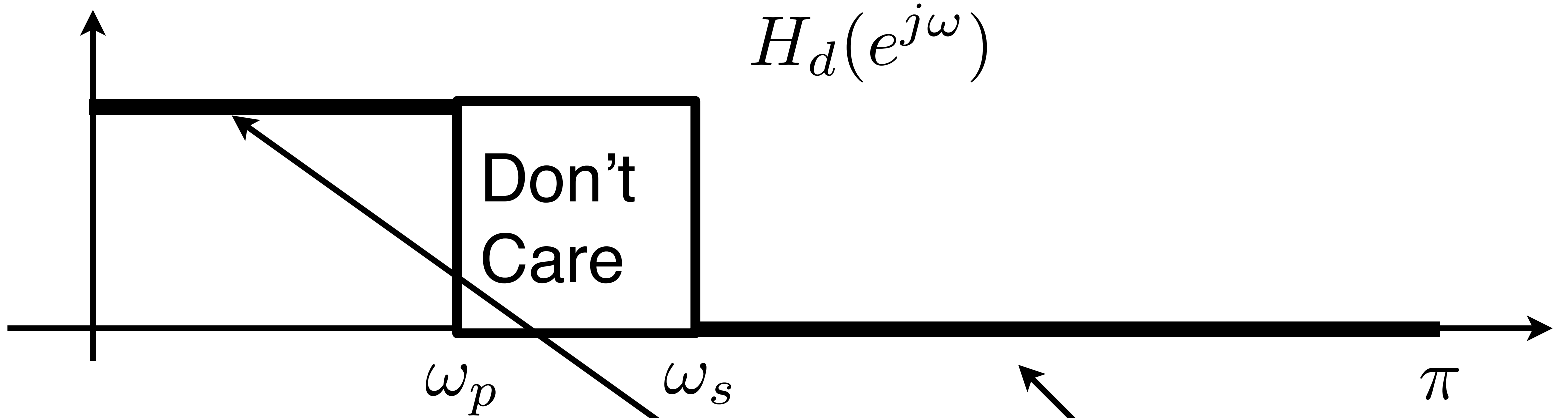
- Result will generally be non-symmetric and complex valued.
- However, if $\tilde{H}(e^{j\omega})$ is real, $\tilde{h}[n]$ should have symmetry!

Design of Linear-Phase L.P Filter

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

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

Least-Squares Linear-Phase Filter



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

$$A = \begin{bmatrix} \quad \\ \quad \end{bmatrix} \qquad b = [\quad , \quad]^T$$

Least-Squares Linear-Phase Filter

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

$$A =$$

$$b = [\;,\;]^T$$

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

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

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

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 $W(\omega)$ is δp in the pass band and δs in stop band

Similarly: $W(\omega)$ is 1 in the pass band and $\delta p/\delta s$ in stop band

Weighted Least-Squares

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

Solution:

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

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

Min-Max optimal Filters

- Chebychev Design (min-max)

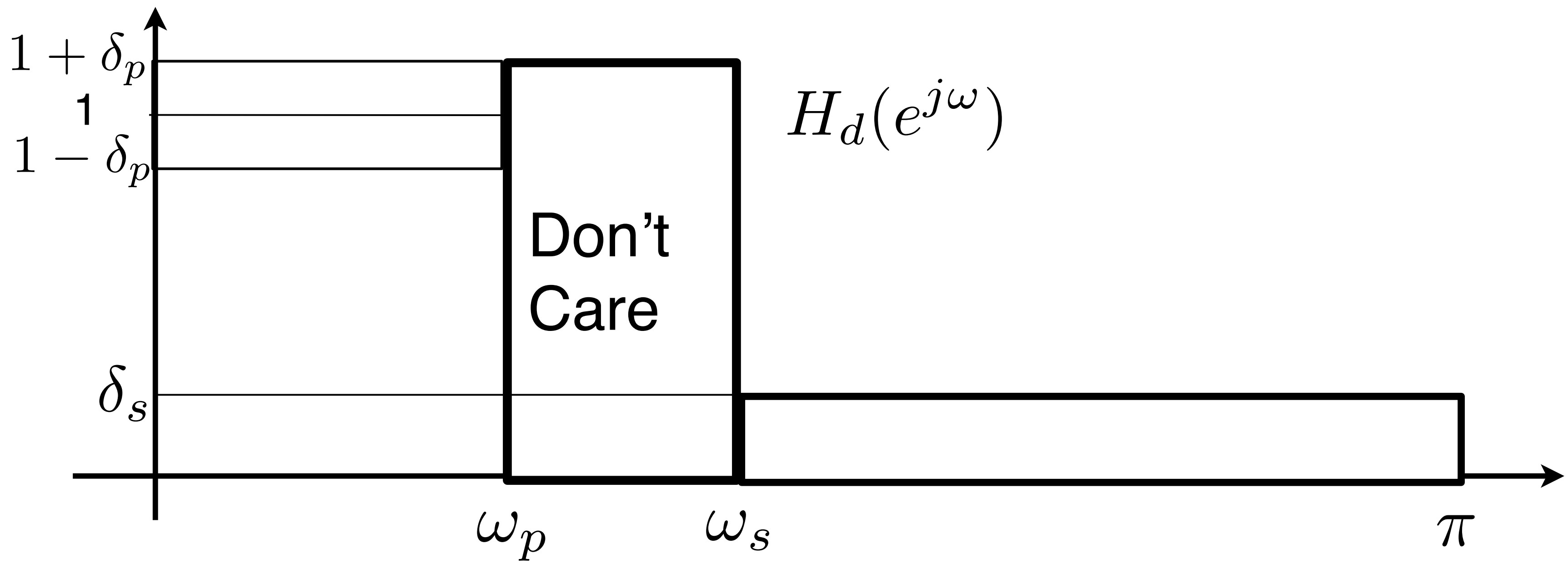
$$\text{minimize}_{\omega \in \text{care}} \quad \max \left| H(e^{j\omega}) - H_d(e^{j\omega}) \right|$$

  – Parks-McClellan algorithm - equi-ripple

  – Also known as Remez exchange algorithms (signal.remez)

  –Also with convex optimization

Specifications



- Filter specifications are given in terms of boundaries

Min-Max Filter Design

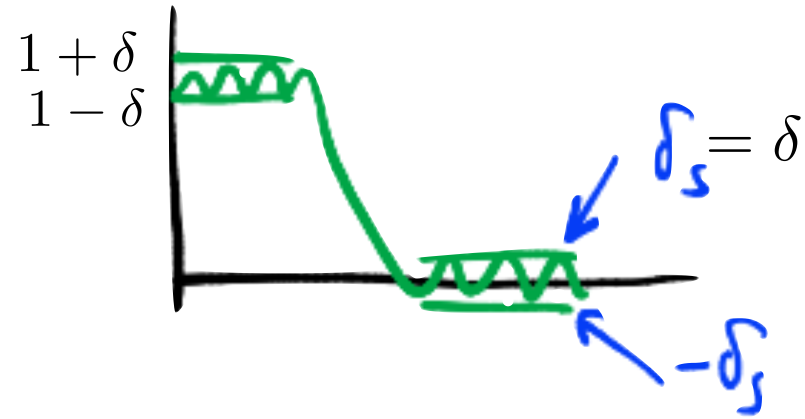
- Minimize:
  - max pass-band ripple

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

  - min-max stop-band ripple

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

Min-max Ripple Design



- Recall, $\tilde{H}(e^{j\omega})$ is symmetric and real
- Given ωp ωs M, find $\delta, \tilde{h}_+$:

$$\text{minimize} \qquad \delta$$

$$\text{Subject to :}$$

$$\begin{aligned}
&1-\delta \le \tilde{H}(e^{j\omega_k}) \le 1+\delta && 0 \le \omega_k \le \omega_p \\
&-\delta \le \tilde{H}(e^{j\omega_k}) \le \delta && \omega_s \le \omega_k \le \pi \\
&\delta > 0
\end{aligned}$$

- Solution is a linear program in $\delta, \tilde{h}_+$
- A well studied class of problems

Min-Max Ripple via Linear Programming

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

$$\text{subject to :}$$

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

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

$$\delta > 0$$

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

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

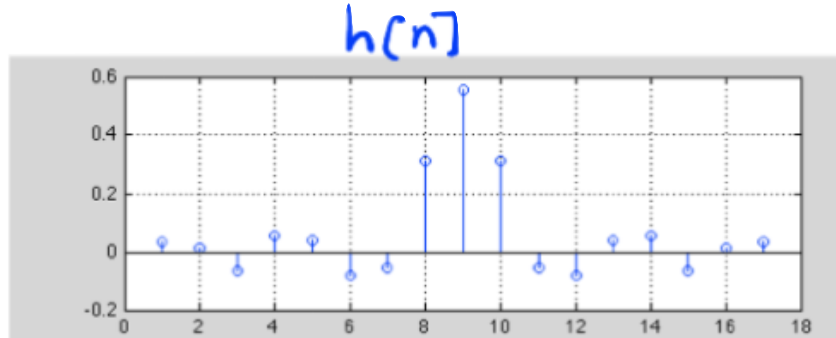
capital P

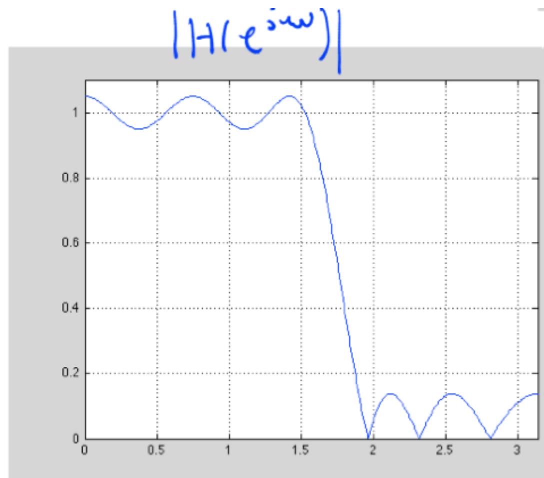
Convex Optimization

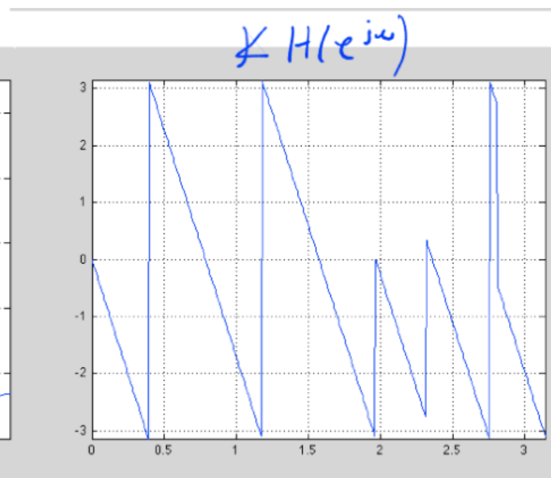
- Many tools and Solvers
- Tools:
  - CVX (Matlab) http://cvxr.com/cvx/
  - CVXOPT, CVXMOD (Python)
- Engines:
  - Sedumi (Free)
  - MOSEK (commercial)
- Take EE127!

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)];
```







Variations:

- Convex Problems:
  - –Fix δ_s optimize for δ_p
  - –Fix δ_p optimize for δ_s
  - –Linear constraints on h[n]
- Quasi-Convex (feasible through bisection)
  - –Fix δ_p, δ_s, M, minimize $\Delta\omega=\omega_s-\omega_p$
  - –Fix δ_p, δ_s, $\Delta\omega=\omega_s-\omega_p$, minimize M

Bisection Example: Minimize M

- given δ_p, δ_s, $\Delta\omega=\omega_s-\omega_p$ Initialize problem with:
  - Set M_{min} to be small and hence infeasible
  - Set M_{max} to be large and hence feasible
  - Set M = floor($M_{max}/2 + M_{min}/2$)
- Given M, δ_p, $\Delta\omega=\omega_s-\omega_p$ solve for minimum δ_s
  - If δ_s violates constrains, set Mmin = M
  - if δ_s within constraints, set Mmax = M
  - Set M = floor($M_{max}/2 + M_{min}/2$)
  - Repeat till M is tight