EE123
Digital Signal Processing

Lecture 20
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
• We will cover FIR designs, briefly mention IIR
What is a linear filter

• Attenuates certain frequencies
• Passes certain frequencies
• Effects 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 Design by Windowing

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

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

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

$$h[n] = \begin{cases} h_d[n] w[n] & 0 \leq n \leq M \\ 0 & \text{otherwise} \end{cases}$$
FIR Design by Windowing

- We already saw that, periodic

\[ H(e^{j\omega}) = H_d(e^{j\omega}) \ast W(e^{j\omega}) \]

- For Boxcar (rectangular) window

\[ W(e^{j\omega}) = e^{-j\omega \frac{M}{2}} \frac{\sin(\omega(M + 1)/2)}{\sin(\omega/2)} \]
FIR Design by Windowing

\[ |H(e^{j\omega})| \]

- pass-band ripple
- transition width
- stop-band ripple
- ideal

\[ \omega_c \]
# Tapered Windows

<table>
<thead>
<tr>
<th>Name(s)</th>
<th>Definition</th>
<th>MATLAB Command</th>
<th>Graph (M = 8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hann</td>
<td>( w[n] = \begin{cases} \frac{1}{2} \left[ 1 + \cos \left( \frac{\pi n}{M/2} \right) \right] &amp;</td>
<td>n</td>
<td>\leq M/2 \ 0 &amp;</td>
</tr>
<tr>
<td>Hanning</td>
<td>( w[n] = \begin{cases} \frac{1}{2} \left[ 1 + \cos \left( \frac{\pi n}{M/2 + 1} \right) \right] &amp;</td>
<td>n</td>
<td>\leq M/2 \ 0 &amp;</td>
</tr>
<tr>
<td>Hamming</td>
<td>( w[n] = \begin{cases} 0.54 + 0.46 \cos \left( \frac{\pi n}{M/2} \right) &amp;</td>
<td>n</td>
<td>\leq M/2 \ 0 &amp;</td>
</tr>
</tbody>
</table>
Tradeoff - Ripple vs Transition Width

Python: scipy.filter.firwin
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 $M+1 \Leftrightarrow$ effect transition width
  - Type of window $\Leftrightarrow$ transition-width/ ripple
  - Modulate to shift impulse response
    \[ H_d(e^{j\omega})e^{-j\omega \frac{M}{2}} \]
FIR Filter Design

- Determine truncated impulse response \( h_1[n] \)

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

- 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 \( M \Rightarrow \) Window length and set

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

\[ h_1[n] = \begin{cases} 
\frac{\sin(\omega_c (n-M/2))}{\pi(n-M/2)} & 0 \leq n \leq M \\
0 & \text{otherwise} 
\end{cases} \]
Example: FIR Low-Pass Filter Design

- The result is a windowed sinc function

\[ h_w[n] = w[n]h_1[n] \]

- High Pass Design:
  - Design low pass \( h_w[n] \)
  - Transform to \( h_w[n](-1)^n \)

- General bandpass
  - Transform to \( 2h_w[n]\cos(\omega_0n) \)
Characterization of Filter Shape

Time-Bandwidth Product, a unitless measure
\[ T(BW) = \frac{(M+1)\omega}{2\pi} \Rightarrow \text{also, total \# of zero crossings} \]

Larger TBW \( \Rightarrow \) More of the “sinc” function

hence, frequency response looks more like a rect function
Frequency Response Profile

Q: What are the lengths of these filters in samples?

TBW=2

\[ 2 = \frac{(M+1)\pi/6}{2\pi} \Rightarrow M=23 \]

TBW=12

\[ 12 = \frac{(M+1)\pi}{2\pi} \Rightarrow M=23 \]

Note that transition is the same!
Alternative 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:  \( \omega_k = k \frac{2\pi}{P} \quad |k = [0, \cdots, P - 1] \)
– Compute \( h_1[n] = \text{IDFT}_P(H_1[k]) \)
– Apply M+1 length window:

\[ h_w[n] = w[n]h_1[n] \]

M. Lustig, EECS UC Berkeley
Example: signal.firwin2

- `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: Design using FFT

- For $M+1=14$
  - $P = 16$ and $P = 1026$
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.
• Least Squares:

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

Variation: weighted least-squares

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

M. Lustig, EECS UC Berkeley
Optimality

- Chebychev Design (min-max)

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

- Parks-McClellan algorithm - equi-ripple
- Also known as Remez exchange algorithms (signal.remez)
Example of Complex Filter


Need to design 11 taps filter with following frequency response: