INTRODUCTION TO FIR FILTERS

CHAPTER 3.1
WHY FIR FILTERS?

- Always stable
  - Finite length
- Linear phase property (delay) is guaranteed
  - Even/odd symmetry
- Finite precision errors are less severe
  - No feedback
- FIR filtering is efficient for implementation

- Modern filter design is FIR design
**TRANSIENT RESPONSE**

- Remember the LTI system

\[ x(n) \rightarrow h(n) \rightarrow y(n) \]

- Transient response
  - Rising-time – how fast output can change (changing rate)
  - Settling-time – how long to settle to stable value
  - Overshoot – if output goes over the desired value
**STEADY STATE RESPONSE**

\[ x(n) \rightarrow h(n) \rightarrow y(n) \]

- **Magnitude response**
  \[ |Y(\omega)| = |H(\omega)||X(\omega)| \]

- **Phase response**
  \[ \Phi_Y(\omega) = \Phi_H(\omega) + \Phi_X(\omega) \]

- **Linear phase filters**
  \[ \Phi_H(\omega) = -\alpha\omega, \text{ or } \pi - \alpha\omega \]
  - All frequencies delayed by same amount
  - Simple phase relationship indicates a time shift by \( \alpha \)
    - \( y(n) = x(n - \alpha) \)

- **Group delay**
  \[ T_d(\omega) = -\frac{d\Phi_H(\omega)}{d\omega} \]
  - Constant group delay for linear phase \( \rightarrow \) no phase distortion
FILTER TYPES

- **Ideal filters**

  ![Magnitude responses of four different ideal filters](image)

  - Defined in terms of magnitude response
  - Note: only $[0, \pi]$ given because with real filter coefficients $H(\omega)$ is even symmetric across $\omega = 0$
  - Remember this is $2\pi$ periodic
  - Bandstop with a narrow band is called a notch filter
  - Allpass filter has $|H(\omega)| = 1, \forall \omega$
FILTER SPECIFICATIONS

- Defined by magnitude response
- Must give a tolerance scheme
  - Cannot practically make ideal filters with sharp transitions

- $\omega_p$ - passband edge frequency
- $\omega_s$ - stopband edge frequency
- $\delta_p$ - passband ripple
  - $A_p = 20 \log_{10} \left( \frac{1+\delta_p}{1-\delta_p} \right) dB$
- $\delta_s$ - stopband attenuation
  - $A_s = -20 \log_{10} \delta_s dB$

Figure 3.4 Magnitude response and performance measurement of lowpass filter

- $1 - \delta_p \leq |H(\omega)| \leq 1 + \delta_p \quad 0 \leq \omega \leq \omega_p$
- $|H(\omega)| \leq \delta_s \quad \omega_s \leq \omega \leq \pi$
LINEAR PHASE FIR FILTERS

- Systems have symmetry which can be exploited

- Even
  - \( b_l = b_{L-1-l}, \ l = 0,1,...,L-1 \)

- Odd
  - \( b_l = -b_{L-1-l}, \ l = 0,1,...,L-1 \)

- Group delay is constant
  
  \[ T_d(\omega) = M = \begin{cases} \frac{L}{2} & L \text{ even} \\ \frac{L-1}{2} & L \text{ odd} \end{cases} \]

- Less multiplications are required because coefficients are shared

Figure 3.5 Signal-flow diagram of symmetric FIR filter; L is an even number
DESIGN OF FIR FILTERS

CHAPTER 3.2
FIR FILTER DESIGN

- Determine filter coefficients to meet specifications
  - $a_k, b_k$ coefficients

Figure 3.4 Magnitude response and performance measurement of lowpass filter
FOURIER SERIES (WINDOWING) METHOD

- Find a desired impulse response from desired frequency response
  - \( H_d(\omega) = \sum_{n=-\infty}^{\infty} h_d(n)e^{-j\omega n} \)
  - \( h_d(n) = \int_{-\pi}^{\pi} H_d(\omega)e^{j\omega n} d\omega \)
- Notice the impulse response is in general infinite
  - Can make this finite only taking some of the samples (truncate)
    - \( h(n) = \begin{cases} h_d(n) & -M \leq n \leq M \\ 0 & \text{else} \end{cases} \)
    - This can be made causal by shifting to the right by \( M \) samples
      - \( b_l = h(l - M), \ l = 0, \ldots, 2M \)
- Notice that \( h(n) \) can be thought of as FS coefficients for \( H_d(\omega) \)
  - More coefficients, better approximation
Example 3.5

\[ H_d(\omega) = \begin{cases} 
1 & |\omega| \leq \omega_c \\
0 & \text{else} 
\end{cases} \]

Use FT equation or in a Table of common pairs

\[ h_d(n) = \sin \left( \frac{\omega_c n}{\pi} \right) = \frac{\omega_c}{\pi} \text{sinc} \left( \frac{\omega_c n}{\pi} \right) \]

Window the impulse response and shift to make causal

\[ b_l = \begin{cases} 
\frac{\omega_c}{\pi} \text{sinc} \left( \frac{\omega_c (l-M)}{\pi} \right) & 0 \leq l \leq L - 1 \\
0 & \text{else} 
\end{cases} \]

Example 3.7

Design a LP filter with \( \omega_c = 0.4\pi \) with \( L = 61 \).

\[ M = \frac{L-1}{2} = 30 \]

\[ b_l = 0.4 \text{sinc}(0.4(l - 30)), \quad l = 0, 1, \ldots, 60 \]
Notice the rippling effect known as Gibbs phenomenon

Windowing is equivalent to multiplication in time domain

\[ h(n) = h_d(n)w(n) \]

Rectangular window

\[ w(n) = \begin{cases} 1 & -M \leq n \leq M \\ 0 & \text{else} \end{cases} \]

Multiplication in time is convolution in frequency domain

\[ H(\omega) = \frac{1}{2\pi} H_d(\omega) * W(\omega) \]

\[ W(\omega) = \frac{\sin\left(\frac{(2M+1)\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \]

Figure 7.28 Magnitude of the Fourier transform of a rectangular window \((M = 7)\).
\[ H(\omega) = \frac{1}{2\pi} H_d(\omega) \ast W(\omega) \]

- Ideal frequency response is smoothed by window DTFT

Figure 7.27  (a) Convolution process implied by truncation of the ideal impulse response. (b) Typical approximation resulting from windowing the ideal impulse response.

Figure 7.28  Magnitude of the Fourier transform of a rectangular window \((M = 7)\).
RECTANGULAR WINDOW

- \[ W(\omega) = \frac{\sin\left(\frac{(2M+1)}{2} \omega\right)}{\sin\frac{\omega}{2}} \]

- This window spectrum has ripples which causes ripples in \( H(\omega) \) at sharp transitions
  - Can’t make perfectly sharp edges

- Mainlobe – centered at \( \omega = 0 \)
  - Care about width

- Sidelobes – all other ripples
  - Care about height

- Gibbs phenomenon can be managed by smoothing the window edges
  - Results in lower sidelobe height and increased mainlobe width
  - Larger transition width at discontinuity but less ringing

*Figure 3.11* Magnitude responses of the rectangular windows for \( M = 8 \) (top) and \( M = 20 \) (bottom).
**WINDOWING DESIGN CONSIDERATIONS**

- \( H(\omega) = \frac{1}{2\pi} H_d(\omega) * W(\omega) \)
  - Ideal frequency response is smoothed by window DTFT

- The quality of the FIR approximation is dependent on two factors
  - The width of the main lobe
  - The peak side-lobe amplitude

- Want narrow main-lobe width with small side lobe amplitude
  - More impulse-like
  - Cannot optimize both at the same time

- \( N\Delta f = c \)
  - \( N \) – length of filter
  - See Shaum’s DSP notes

- Increasing length of window the decreases the width of the mainlobe
  - Decreases width of the transition band

- Peak sidelobe amplitude is practically independent of length only depends on shape of window
  - Decrease in sidelobe amplitude results in greater mainlobe width
WINDOW FUNCTIONS

- Many windows have been designed to trade off mainlobe width and sidelobe height
- All have smooth transitions at edge of window

### Table 9-1: Some Common Windows

<table>
<thead>
<tr>
<th>Window</th>
<th>$w(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>$w(n) = \begin{cases} 1 &amp; 0 \leq n \leq N \ 0 &amp; \text{else} \end{cases}$</td>
</tr>
<tr>
<td>Hanning$^1$</td>
<td>$w(n) = \begin{cases} 0.5 - 0.5 \cos \left( \frac{2\pi n}{N} \right) &amp; 0 \leq n \leq N \ 0 &amp; \text{else} \end{cases}$</td>
</tr>
<tr>
<td>Hamming</td>
<td>$w(n) = \begin{cases} 0.54 - 0.46 \cos \left( \frac{2\pi n}{N} \right) &amp; 0 \leq n \leq N \ 0 &amp; \text{else} \end{cases}$</td>
</tr>
<tr>
<td>Blackman</td>
<td>$w(n) = \begin{cases} 0.42 - 0.5 \cos \left( \frac{2\pi n}{N} \right) + 0.08 \cos \left( \frac{4\pi n}{N} \right) &amp; 0 \leq n \leq N \ 0 &amp; \text{else} \end{cases}$</td>
</tr>
</tbody>
</table>

$^1$In the literature, this window is also called a Hann window or a von Hann window.
**WINDOW PERFORMANCE**


<table>
<thead>
<tr>
<th>Window</th>
<th>Side-Lobe Amplitude (dB)</th>
<th>Transition Width (Δf)</th>
<th>Stopband Attenuation (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rectangular</td>
<td>-13</td>
<td>0.9/N</td>
<td>-21</td>
</tr>
<tr>
<td>Hanning</td>
<td>-31</td>
<td>3.1/N</td>
<td>-44</td>
</tr>
<tr>
<td>Hamming</td>
<td>-41</td>
<td>3.3/N</td>
<td>-53</td>
</tr>
<tr>
<td>Blackman</td>
<td>-57</td>
<td>5.5/N</td>
<td>-74</td>
</tr>
</tbody>
</table>

http://www.labbookpages.co.uk/audio/firWindowing.html
FIR DESIGN STEPS

- Select window type to satisfy stopband attenuation requirements
- Determine window size $L$ based on transition width ($N$ in Schaum’s)
- Calculate window values/coefficients
- Calculate impulse response of desired filter
  - Truncate to fixed length $L$
  - Shift to make causal
- Calculate final filter coefficients as product of window and desired response
  - $b_l = h_d[l - M]w[l]$
APPLICATIONS: INTERPOLATION AND DECIMATION FILTERS

CHAPTER 3.4
UPSAMPLING/INTERPOLATION

- Increase the sampling rate of a signal by factor $L$
- Accomplished by inserting zeros into a sequence and then lowpass filtering
- Zero insertion is upsampling
  
  - $x_u(n) = \begin{cases} x\left(\frac{n}{L}\right) & n = 0, \pm L, \pm 2L, \ldots \\ 0 & \text{else} \end{cases}$
  
  - Resulting signal has more samples but gaps between values
- LP filtering is interpolation to fill gaps
  
  - LP filter using gain $L$ and cutoff $= \pi/L$
  - Gain of $L$ to “spread” sample energy to neighbor zeros

Figure 4.24 Frequency-domain illustration of interpolation.
- Reduce the sampling rate of a signal by factor $M$
- Accomplished by dropping samples
  - $x_d(n) = x(nM)$
- Remember bandwidth is controlled by sampling rate
  - Both sampling rate and bandwidth decrease by factor $M$
  - This may result in aliasing of the signal
- Avoid aliasing by pre-filtering signal with LP filter with cutoff $= \pi/M$ before decimation

Figure 4.21  (a)–(c) Downsampling with aliasing. (d)–(f) Downsampling with prefiltering to avoid aliasing.
Conversion to arbitrary sample rate is possible

\[ R = \frac{U}{D} \]

- Must find appropriate upsample factor \( U \) and downsample factor \( D \)

First perform interpolation followed by decimation

- Minimize reduction in signal bandwidth
  - No fear of aliasing in upsample
- Downsampling first could result in loss of high frequency content

Can combine interpolation LP filter with LP for decimation

- Cutoff should be minimum of either operation

Use Matlab interp.m, decimate.m, and upfindn.m/resample.m