DIGITAL FILTER DESIGN
Digital Filter Design

I. Introduction
   - Why Filter
   - Pole/Zero placement

II. IIR Filter Design
    - Impulse Invariance
    - Bilinear transform

III. FIR Filter Design
     - Window methods
     - Parks-McClellan

IV. Statistical Filter Design
    - Wiener Filters

V. Adaptive Filters
   - LMS
WHY FILTERS?

* To separate signal components.

Coffee filter

Separates components based on their particle size

Digital filters separate signal components based on their frequency content.

Ex. Lowpass filter

Consider, \( s[n] \) smooth

\[ s + x \]

\[ H \]

\[ \hat{s} \]

\[ X[n] \) irregular

Signal + noise
ex  "Differentiator"

\[ H(w) \]

Types of Filters:

- Lowpass
- Highpass
- Bandpass
- Bandstop

All can be obtained from lowpass filters

IIR - Infinite Impulse Response

- \( h(n) \) infinite in extent
- \( H(Z) \) has poles

FIR - Finite Impulse Response

- \( h(n) \) finite in extent
- \( H(Z) \) has only zeros
Simple Filter Design by Pole-Zero Placement

Linear Difference Equation Filter:

\[ Y[n] = \sum_{k=0}^{M} b_k x[n-k] - \sum_{k=1}^{N} a_k y[n-k] \]

Transfer Function:

\[ Y(z) = \sum_{k=0}^{M} b_k z^{-k} X(z) - \sum_{k=1}^{N} a_k z^{-k} Y(z) \]

\[ H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \quad a_0 = 1 \]

Factor numerator and denominator polynomials

\[ H(z) = \frac{b_0}{a_0} \prod_{k=1}^{M} \left( 1 - c_k z^{-k} \right) \frac{\prod_{k=1}^{N} \left( 1 - d_k z^{-k} \right)}{\prod_{k=0}^{N} \left( 1 - a_k z^{-k} \right)} = \frac{\text{"zeros"}}{\text{"poles"}} \]

\[ H(z) = \frac{b_0}{a_0} \prod_{k=1}^{M} \left( 1 - c_k e^{-j2\pi F_k} z^{-k} \right) \frac{\prod_{k=1}^{N} \left( 1 - d_k e^{-j2\pi F_k} z^{-k} \right)}{\prod_{k=0}^{N} \left( 1 - a_k e^{-j2\pi F_k} z^{-k} \right)} \]

By cleverly placing poles and zeroes we can obtain a desired frequency response.
Ex. Remove a certain frequency component.

For example, 60 Hz "line noise"

\[ X(t) = s(t) + \sin(2\pi (60) t) \]

Filter \( x(t) \) to remove 60 Hz component. Sample \( x(t) \) at \( \Omega = 240 \text{ Hz} \).

\[ x[n] = s[n] + \sin(2\pi \frac{60}{240} n) \]

Digital Filter Design 1:

\[ \frac{60}{240} = \frac{1}{4} \]

Place Zeros @

\[ z_1 = j, \quad z_2 = -j \]

\[ H(z) = (1 - j z^{-1})(1 + j z^{-1}) \]

\[ = 1 + z^{-2} \]

\[ h_1[n] = s[n] + s[n-2] \rightarrow y[n] = x[n] + x[n-2] \]
\( H_1 \) does the job, but also attenuates frequencies close to 60Hz.

**Note:**

DC Gain of \( H_1(F) \) is 2.

Therefore, use \( h_c(n) = \frac{1}{3} (\sin(n) + \sin(n-2)) \) to get DC gain 1.

Add poles close to zeros. Response still must go to zero at 60Hz, but as we move away from 60Hz pole offsets the effect of the zero.

\[ P_1 = 0.9j, \quad P_2 = \]

\[ H_2(z) = \frac{(1-1z^{-1})(1+jz^{-1})}{(1-0.9jz^{-1})(1+0.9jz^{-1})} = \frac{1+z^{-2}}{1+0.81z^{-2}} \]

\[ y[n] + 0.81y[n-2] = x[n] + x[n-2] \]

\[ y[n] = x[n] + y[n-2] - 0.81y[n-2] \]

Note: Multiply \( H(z) \) by \( \frac{1.81}{z} \) to get unity gain at DC!
Much closer to "ideal" notch at 60 Hz

- \( h_1[n] = \delta[n] + \delta[n-27] \) is FIR

- \( h_2[n] \) impulse response: \( X[n] = \delta[n] \)
  \[
  y[n] = x[n] + x[n-27] + 0.81 y[n-27]
  \]
  IIR (feedback)

- Conjugate symmetry of poles/zerors guarantees real-valued digital filter

This simple approach of placing poles and zeros can work, but it is sort of a trial and error approach.

We can do better!
Given that we want to separate a signal into frequency components, we would clearly like to do the **BEST JOB POSSIBLE**

\[ \Rightarrow \text{Filter Design / Engineering} \]

Usually we will have **specifications** that we need to meet.

**Example:** Lowpass filter \[ \rightarrow A/B \rightarrow H \rightarrow B/A \rightarrow \]

Specifications in CT.

**Passband:** \[ |H_{\text{eff}}(\omega)| \text{ within } \pm 0.01 \text{ of unity gain in the range } 0 \leq f \leq 2\text{kHz} \rightarrow 0 \leq \Omega \leq \]

**Stopband:** \[ |H_{\text{eff}}(\omega)| \leq 0.001 \text{ in the range } \]

\[ f \geq 3\text{kHz} \rightarrow \Omega \geq \]

**Suppose** \[ T = 10^{-4} \rightarrow f_3 = 10\text{kHz} \Rightarrow \Omega_3 = \]

Draw specs in digital freq.

\[ |H(\omega)| \]

\[ 1.01 \]

\[ 0.01 \]

\[ 0.001 \]

\[ 0 \rightarrow \omega \]
IIR Filter Design By
Impulse Invariance

Traditional approach to IIR Filter transforms CT filters (Butterworth, Chebychev) to DT Filters. Why?

1. Continuous-time filter design is advanced, mature, well-understood.
2. We get elegant closed-form formulas.

Properties we desire of the transform:

1. It should map imaginary axis of the s-plane (Laplace) to the unit circle of the z-plane
   (Recall, evaluating Laplace transform $H(s)$ along $jw$-axis produces the CTFT)

2. A stable CT filter should transform to a stable DT filter.

Design steps:
1. Translate DT specs to CT filter
2. Design CT prototype filter
3. Transform filter to discrete-time
CT Filter

Laplace Transform
S-plane, LHP stability
$H(s), H_c(j2)$

DT Filter

Z-transform
Z-plane, unit circle
$H(z), H(w)$

$h_c(t)$ ➔ impulse invariance ➔ $h[n] = h_c(nT)$

$H_c(s)$ ➔ bilinear transform ➔ $H(z) = H_c(s)\bigg|_{s = \frac{z-1}{T(z+1)}}$

Key: Convert from rectangular coordinates to polar coordinates

613
Impulse Invariance Design

Easy - we simply sample the CT filter's impulse response.

\[ h[n] = T_d \cdot h_c(nT_d) \]

A sampling period. (Need not be the same as \( T \) used in A/D and D/A.)

\( h[n] \) is a sampled version of \( h_c(t) \) \( \Rightarrow \) aliasing in frequency domain.

\[
H(\omega) = \sum_{k=-\infty}^{\infty} H_c \left( \frac{\omega + 2\pi k}{T_d} \right)
\]

If \( H_c \) is bandlimited, with \( H_c(\pi) = 0 \)

then

\[
H(\omega) = H_c \left( \frac{\omega}{T_d} \right)
\]

However, this is impossible, since any practical CT filter cannot be perfectly bandlimited. \( \Rightarrow \) Aliasing:

\[
\text{Aliasing}
\]

\( -2\pi \)

\( -\pi \)

\( \pi \)

\( 2\pi \)

\( \omega \)
Aliasing $\Rightarrow$ Need to overdesign CT prototypes

- Given an CT transfer function $H_c(s)$, what is the corresponding DT transfer function?

Expand $H_c(s)$ in partial fractions

$$H_c(s) = \sum_{k=1}^{N} \frac{A_k}{s - s_k} \quad \text{poles @ } s = s_k$$

$\Rightarrow$ $h_c(t) = \sum_{k=1}^{N} A_k e^{s_k t} u(t)$ (Inverse Laplace Transform)

$\Rightarrow$ $h(nT_d) = T_d h_c(nT_d) = T_d \sum_{k=1}^{N} A_k e^{s_k nT_d} u(n)$

$= T_d \sum_{k=1}^{N} A_k (e^{s_k T_d})^n u(n)$

$\Rightarrow$ $H(z) = \sum_{k=1}^{N} \frac{T_d A_k}{1 - e^{s_k T_d} z^{-1}} \quad \text{poles @ } z = e^{s_k T_d}$

$S$-plane

$\text{Re}(s)$

$\text{Im}(s)$

$Z$-plane

$G15$
IIR Filter Design by Impulse Invariance

5 Golden Steps

1. Given specs for $H(\omega)$, choose $T_d$ (arbitrary) & translate specs to $H_c(\omega/T_d) = H(\omega)$

2. Design $H_c(s)$: Butterworth, Chebyshev, ...

3. Find $H_c(s)$ & do partial fractions (get poles, $s_k$)

4. Transform poles through

   $s_k \rightarrow e^{s_k T_d}$

   $s$-plane $\rightarrow$ $z$-plane

   $\frac{A_k}{s - s_k} \rightarrow \frac{T_d A_k}{1 - e^{s_k T_d} z^{-1}}$

   Compute $H(z)$

5. Check performance of $H(\omega)$. If not up to spec, redesign $H_c(s)$ in step 2 with more conservative specs.
Continuous-Time Butterworth Filters

+ Appendix B in OSB

- Magnitude response is **maximally flat** in the passband
- Magnitude response is **monotone**

\[
|H_c(\omega)|^2 = \frac{1}{1 + (\omega/\omega_c)^{2N}}, \quad \omega \sim \text{half power frequency}
\]

What happens as \( N \) increases?

| \[ \text{sharper cut-off} \] |
In the $s$-plane:

$$H_c(s) H_c(-s) = \frac{1}{1 + (\frac{s}{j \Omega_c})^{2N}}$$

$\Rightarrow$ poles are at

$$s_k = (-1)^{\frac{1}{2N}} (j \Omega_c) = \Omega_c e^{j \frac{\pi}{2N} (2k + N - 1)}$$

$k = 0, 1, \ldots, 2N - 1$

$\Rightarrow$ poles lie on a circle of radius $\Omega_c$

ex. $N = 3$ (Third order filter)

For a stable, causal filter, we take only the left half-plane poles.

* See Matlab function
  
  `butter(____, 's')`
Design Example: Impulse Invariance

Specs on $H(z)$:

**Passband**: $0.89125 \leq |H(\omega)| \leq 1$, 
$0 \leq |\omega| \leq 0.2\pi$

**Stopband**: $|H(\omega)| \leq 0.17783$, $0.3\pi \leq |\omega| \leq \pi$

![Graph of $|H(\omega)|$]

**Step 1**: Set $T_d = 1$

$\Rightarrow$ Specs for $H_c(z)$ are identical to those for $H(\omega)$.

**Step 2**: Design $H_c(z)$. Choose Butterworth.

What are design parameters?

$\omega_c$, $N$

Choose to meet passband constraint exactly.
Since magnitude of Butterworth is monotonic if freq

\[ |H_c(j\omega)| \geq 0.89125 \]

\[ \Rightarrow \frac{1}{1 + \left(\frac{0.2\pi}{\omega_c}\right)^{2N}} = (0.89129)^2 \]

(using monotonicity property of Butterworth)

\[ |H_c(j\omega)| \leq 0.17783 \]

\[ \Rightarrow \frac{1}{1 + \left(\frac{0.8\pi}{\omega_c}\right)^{2N}} = (0.17783)^2 \]

\[ 1 + equation \rightarrow N = 5.9858, \omega_c = 0.70474 \]

\[ \Rightarrow \text{round } N \text{ to nearest integer} \]

\[ N = 6 \quad (3 \text{ pole pairs}) \]

Pole pair 1: \( s_1, s_2 = -0.182 \pm j(0.679) \)

Pole pair 2: \( s_3, s_4 = -0.497 \pm j(0.497) \)

Pole pair 3: \( s_5, s_6 = -0.679 \pm j(0.182) \)
**Step 3:** Where are the poles?

\[ |s_k| = 0.7029 \]

\[ H_c(s) = \frac{0.1209}{(s^2 + 0.3640s + 0.4945)(s^2 + 0.995 + 0.49)(s^2 + 1.3855 + 0.049)} \]

**Step 4:**

Partial fraction expansion and transform

\[ \frac{A_k}{s-s_k} \rightarrow \frac{T_d A_k}{1-e^{s_k T_d}}z^{-1} \]

\[ H(z) = \frac{0.2871 - 0.4466z^{-1}}{1 - 1.2971z^{-1} + 0.6949z^{-2}} + \frac{-2.1428 + 1.1455z^{-1}}{1 - 1.0691z^{-1} + 0.3697z^{-2}} + \frac{1.857 - 0.6303z^{-1}}{1 - 0.9972z^{-1} + 0.2570z^{-2}} \]
STEP 5: Plot \( H(\cdot) \) and verify it meets specs.

Are specs met?

What if they weren't met?

Why wouldn't they be met?
Impulse Invariance

Motivated by a desire to match the impulse response if CT filter is band limited, then approximation will be close in the frequency domain.

Problem: Aliasing

⇒ Difficult to control errors in design process

Q: How to design a highpass or band stop filter this way?

In practice, II design is not used very often due to these difficulties.
IIR FILTER DESIGN BY BILINEAR TRANSFORMATION

Recall: **Impulse Invariance design**
- maps poles & zeros in $s$-plane ($\sim H_c(s)$)
- poles & zeros in $z$-plane ($\sim H(\omega)$)

Problem: $H(\omega)$ is an aliased version of $H_c(s)$.

Why: Because we get $H(\omega)$ by wrapping $H_c(s)$ around the unit circle (an infinite # of times) and adding the result together.

The Bilinear transformation avoids aliasing by compressing the entire $s$-axis ($j\omega$-axis in $s$-plane) into one revolution of the unit circle, through a **Frequency Axis Warping**
<table>
<thead>
<tr>
<th><strong>BILINEAR TRANSFORMATION (BLT)</strong></th>
<th>[ S = 2 \left( \frac{z-1}{z+1} \right) ]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INVERSE BLT</strong></td>
<td>[ z = \frac{1 + s/2}{1 - s/2} ]</td>
</tr>
</tbody>
</table>

Substitute \( s = \sigma + j \omega \)

\[ z = \frac{1 + \sigma/2 + j\omega/2}{1 - \sigma/2 + j\omega/2} \]

**Property 1:** If \( \sigma < 0 \), then \(|z| < 1\)

If \( \sigma > 0 \), then \(|z| > 1\).

\[ \Rightarrow \]

Left-half-plane \( \Rightarrow \) inside unit circle

Stability \( \Rightarrow \) stability.
**Property 2:**

BLT maps \( j\omega \)-axis of \( s \)-plane to unit circle in \( z \)-plane.

Substitute \( s = j\omega \)

\[
\Rightarrow z = \frac{1 + j\omega}{1 - j\omega} \Rightarrow e^{j\omega} = \frac{1 + j\omega}{1 - j\omega}
\]

\[
\Rightarrow j\omega = \frac{1 - e^{-j\omega}}{1 + e^{-j\omega}} = \frac{2je^{-j\omega/2}}{2e^{-j\omega/2}/2} = \frac{\sin \omega/2}{\cos \omega/2}
\]

\[\Rightarrow j\omega = 2\tan \frac{\omega}{2}\]

A maps entire \( j\omega \)-axis \((-\infty, \infty)\) to 1-period \( \omega \)-axis \([-\pi, \pi]\)

**KEY:** Design a CT Filter \( H_c(s) \) where \( H(\omega) = H_c(z) \) with \( z = 2(\frac{\omega}{2\pi})^{\frac{\pi}{2\pi}} \)
Where does BLT map poles and zeros?

Given an s-plane zero or pole $s_k$, it is mapped to

$$Z_k = \frac{1 + s_k/z}{1 - s_k/z}$$

Ex.

$$H_c(s) = \frac{1}{(s-s_1)(s-s_2)(s-s_3)}$$

will be mapped to

$$H(z) = \frac{\text{Const.}(z+1)^3}{(z - (\frac{1+s_1/z}{1-s_1/z}))(z - (\frac{1+s_2/z}{1-s_2/z}))(z - (\frac{1+s_3/z}{1-s_3/z}))}$$

---

**S-plane**  
**Z-plane**
Consider a single pole:

\[ H_c(s) = \frac{1}{(s-s_i)} \]

\[ l l(Z) = H_c(s) \bigg|_{s=2}\left(\frac{2Z-1}{Z+1}\right) \]

\[ = \frac{1}{\left(2\frac{Z-1}{Z+1} - s_i\right)} \]

\[ = \frac{Z+1}{(2Z-1) - s_i(2+1)} \]

\[ = \frac{(Z+1)}{(2-s_i)Z - (2+s_i)} \]

\[ = \frac{\left(\frac{1}{2-s_i}\right)(Z+1)}{Z - \left(\frac{1+s_i/2}{1-s_i/2}\right)} \]
IIR FILTER DESIGN

BY BLT

4 Golden Steps

DT specs

1. Pre-warp to obtain CT specs

2. Design CT filter to meet these specs

3. Get $H_c(s)$

4. BLT: $H(z) = H_c(s) \mid_{s=2(z-1)/(z+1)}$

or re-map poles & zeros

Pole or zero

$z_k = \frac{1 + s_k/z}{1 - s_k/z}$
Ex. Butterworth Design in II example using BLT instead.

Resulting Filter:

Compare with II design and be able to explain the differences.
Advantages of BLT Design

1. Simple 1-to-1 mapping between s and z-planes avoids aliasing.

2. Can be used for bandpass and highpass designs as well as lowpass.

3. Don't necessarily need to use partial fraction expansion. Simply transform:

\[ s \rightarrow 2 \left( \frac{z^2 - 1}{z + 1} \right) \]
IIR FILTERS

- Impulse Invariance
- Bilinear Transformation

Advantage:

Simple designs base on CT Filters.
Frequency response (magnitude) requirements easily met.

Disadvantage:

Nonlinear phase
Example: \( y(n) = \frac{1}{2} y(n-1) + x(n) \)

\[ Y(z) = \frac{1}{2} z^{-1} Y(z) + X(z) \]

\[ H(z) = \frac{1}{1 - \frac{1}{2} z^{-1}} \]

\[ H(\omega) = \frac{1}{1 - \frac{1}{2} e^{-j\omega}} \]

\[
H(\omega) = \frac{1}{(1 - \frac{1}{2} \cos \omega) + j \frac{1}{2} \sin \omega}
\]

\[
= \frac{(1 - \frac{1}{2} \cos \omega) - j \frac{1}{2} \sin \omega}{|1 - \frac{1}{2} \cos \omega|^2 + |\frac{1}{2} \sin \omega|^2}
\]

\[
\angle H(\omega) = \tan^{-1} \left( \frac{\frac{1}{2} \sin \omega}{1 - \frac{1}{2} \cos \omega} \right)
\]
Phasc Response is Important

Q: What does an LTI filter do to a signal?

A: In frequency domain

\[ X \xrightarrow{\text{H}} Y \]

\[ Y(\omega) = H(\omega) \cdot X(\omega) \]

\[ = \left[ |H(\omega)| \cdot e^{j \Delta H(\omega)} \right] \cdot X(\omega) \]

- Attenuates or accentuates amplitude of frequency component of \( X \) at frequency \( \omega \).
- Shifts the phase of frequency component of \( X \) at frequency \( \omega \).

A IIR filter designs only attack magnitude of frequency response.

We can break things into 3 cases:

1. **Zero Phase**: \( \Delta H(\omega) = 0 \)
2. **Linear Phase**: \( \Delta H(\omega) = \pi \omega \)
3. **Nonlinear Phase**: Everything else
To see what the various phase types can do to signals, let's look at an example of an all-pass system: \( |H(\omega)| = 1 \) for all.

And a simple pulse signal \( x[n] \).
Zero phase: $\Delta H(\omega) = 0 \Rightarrow H(\omega) = 1$

Linear phase: $\Delta H(\omega) = -20\pi \Rightarrow H(\omega) = e^{-i0\omega}$

Simple delay: $\tau = \frac{20\pi}{2\pi} = 10$
Nonlinear Phase:

\[ H(\omega) = -10i \omega \left| \frac{1}{2} \right. \text{sign}(\omega) \]

\[ \Rightarrow H(\omega) = e^{-j10i \omega \left| \frac{1}{2} \right. \text{sign}(\omega)} \]

Nonlinear phase

Output

Distorted pulse!

Bottom line: Nonlinear phase response distorts output waveform
Linear Phase $\implies$ simple delay

Nonlinear Phase $\implies$ distortion of waveform

**IIR Filters** have nonlinear phase and therefore can distort waveforms

Distortion is often very minimal in passband (although hard to design for this)

ex.

[Graph showing magnitude and phase responses of a filter]

approx. linear in passband

nonlinear in stopband but who cares? We're attenuating those four components anyway.
Ex. 6-pole Butterworth
LOWPASS IIR Filter

Note: phase is close to linear in passband.
FIR LOWPASS FILTER

Shifted and truncated sinc

Magnitude

Phase

Note: Gibbs' phenomena

Note: Linear phase
FIR Filter Design

- Does not involve mapping CT prototypes to DT
- Easy to design FIR filters with linear phase
- Good numerical properties

We will look at two methods for FIR filter design

1. Windowing
   - Just truncate ideal filter impulse response to make it FIR.

2. Optimal Design
   - Designs optimize specified criteria
     (Parks-McClellan method)
FIR Filters

\[ H(z) = \sum_{n=0}^{N-1} h[n] z^{-n} \]

\[ = h[0] + h[1] z^{-1} + \cdots + h[N-1] z^{-(N-1)} \]

↑ polynomial in \( z^{-1} \) (not rational)

\( h[n] \) ~ impulse response of FIR filter

\[ y[n] = \sum_{k=0}^{N-1} h[k] x[n-k] \]

- no feedback!
- function of present and past inputs only

Implementation:

\[ y[n] \]

\[ \rightarrow \text{delay} \]

\[ \rightarrow \text{delay} \]

\[ \text{...} \]

\[ \rightarrow \text{delay} \]

\[ y[n-1] \]
Symmetric FIR Filters

Consider a symmetric impulse response

\[ h(n) \]

\[
H(\omega) = \sum_{n=-L}^{L} h(n) e^{-j\omega n}
\]

\[
= h(0) + \sum_{n=1}^{L} h(n) (e^{-j\omega n} + e^{j\omega n})
\]

\[
= h(0) + \sum_{n=1}^{L} h(n) \cdot 2 \cos(\omega n)
\]

\[ \text{pure real} \]
\[ \Rightarrow \text{zero phase}! \]

Problem: **Noncausal**

Solution: Shift \( h(n) \) to get causality

\[ h(n) * \delta[n-L] = \tilde{h}(n) \]
Note: \[ \hat{H}(\omega) = (h[n] + \sum_{n=1}^{\infty} h[n] \cos(\omega n)) e^{-j\omega L} \]

| \[ |H(\omega)| = |\hat{H}(\omega)| \]  
\[ \Delta H(\omega) = 0 \quad \Delta \hat{H}(\omega) = \Delta e^{-j\omega L} = -L \]  

There are four ways, in general, to construct a linear phase filter:

- Two kinds of lengths: even / odd
- Two kinds of symmetry: even / odd

**Type I**

\[ \text{EVEN symmetry} \quad \text{ODD length} \quad \frac{L+1}{2L+1} \]

| \[ |H(\omega)| = |h[0]| + \sum_{n=1}^{L} h[n] \cos(\omega n) \]  
\[ \Delta H(\omega) = -L \]
**TYPE 2**

Even symmetry + EVEN length $\frac{2m}{2m-1}$

$$|H(\omega)| = 2 \sum_{k=1}^{m} h[k] \cos(\omega (k-\frac{1}{2}))$$

$$\Delta H(\omega) = \Delta e^{-j\omega m} = -m$$

**TYPE 3**

ODD symmetry + ODD length $\frac{2L}{2L+1}$

$$|H(\omega)| = 2 \sum_{k=1}^{L} h[k] \sin(\omega k)$$

$$\Delta H(\omega) = -L$$

**TYPE 4**

ODD symmetry + EVEN length $\frac{2M}{2M-1}$

$$|H(\omega)| = 2 \sum_{k=1}^{M} h[k] \sin(\omega (k-\frac{1}{2}))$$

$$\Delta H(\omega) = -M$$
FIR FILTER DESIGN - WINDOW METHOD

A Lo-TECH!

Simple motivation: Say we want to design a lowpass filter.

1. Start with the ideal filter response $H_d(\omega)$

2. Compute IDTFT of $H_d(\omega) \rightarrow h_d[n]$ and shift to make it causal

3. $h_d[n]$ is infinite extent, so truncate and shift to make it causal
MAJOR DRAWBACK

Truncation introduces Gibbs phenomena!

To see this, we can view

\[ h[n] = \begin{cases} 
  h_d[n-L], & 0 \leq n \leq N-1 \\
  0, & \text{otherwise} 
\end{cases} \]

\[ = h_d[n-L] \cdot W[n] \]

Rectangular window

\[ W[n] = \begin{cases} 
  1, & 0 \leq n \leq 2L \\
  0, & \text{otherwise} 
\end{cases} \]

Sinc function

\[ W(\omega) = \frac{e^{j\omega/2}}{2\pi} \cdot \frac{\sin(\omega (2L+1))}{\sin \omega/2} \]

\[ \Rightarrow H(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega) e^{-j\omega L} W(\omega - \lambda) d\lambda \]

What we want delay for causality

"smears" or "blurs" desired freq. response

\[ |W(\omega)| \]

\[ \frac{2\pi}{N} \]

13dB
Convolving with \( W(\omega) \) has two detrimental effects:

1. Main lobe width smears out sharp edges in \( H_d(\omega) \).

2. Sidelobes add ripples (Gibb's phenomena).

Why not just make \( W(\omega) = \delta(\omega) \)?

\[ \implies \hat{w}[n] = 1 \quad \text{for all } n \]

\[ \implies \text{No windowing (truncation) at all!} \]

Not very useful.
Every finite length window has DTFT with
1. Main lobe of some width
2. Sidelobes

We want to minimize mainlobe width and minimize sidelobe height.

Recall from our discussions on windowing (E-30 to E-33) these are conflicting requirements, leading to

mainlobe width / sidelobe trade-off

implies engineering

Unfortunately, this trade-off means that there really isn't a best window.

The choice of window is
more or less a subjective one?

Do you want a better main lobe or better sidelobes?
Example Design

Bottom Line:

1. Window designs are not optimal in any sense (save the rectangular $\delta^2$)

2. Need a closed form expression for $H_d(\omega)$

3. Window smearing makes placement of transition points difficult (need to iterate design process)

4. Easy, but kind of stupid $\Rightarrow$ don't use window design.

5. Ripple is the same in passband & stopband
OPTIMAL FIR FILTER DESIGN

Objective: Design the "best" filter for a given problem

⇒ optimization

Needed: a numerical quantity or criterion that measures performance.

Sensible Measure: Error in frequency response.

How "close" is \( H(\omega) \) to \( H_d(\omega) \)?

How can we measure close?

Criterion 1: (Energy of Error)

\[
\text{minimize} \quad \int_{-\pi}^{\pi} \left| H(\omega) - H_d(\omega) \right|^2 d\omega
\]

Criterion 2: (Maximum Error)

\[
\text{minimize} \quad \max_{-\frac{1}{2} \leq \omega \leq \frac{1}{2}} \left| H(\omega) - H_d(\omega) \right|
\]

Choose FIR filter \( h[n] \) that

minimizes criterion 1 or 2.
CRITERION 1: Minimize Energy of Error

To start with:

- \( h[n] \) is fixed length 2L+1
- centered at origin
  (we can take care of shift (for causality) later)

\[
\begin{array}{c}
\ldots \quad H[n] \quad \ldots
\end{array}
\]

Objective:

Find \( h[n] \) to minimize

\[
\pi \int_{-\pi}^{\pi} |H_d(\omega) - H(\omega)|^2
\]

\[
H_d(\omega) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-j\omega n}
\]

\[
H(\omega) = \sum_{n=-\infty}^{\infty} h[n] e^{-j\omega n}
\]

\[\rightarrow\]

\[
H_d(\omega) - H(\omega) = \sum_{n=-\infty}^{\infty} (h_d[n] - h[n]) e^{-j\omega n}
\]

\[
= \sum_{n=-\infty}^{\infty} d[n] e^{-j\omega n}
\]

\[d[n] = h_d[n] - h[n].\]
So we see,

the FIR filter that minimizes
the energy of the error is

\[ h[n] = \begin{cases} 
  1, & 0 \leq n \leq L-1 \\
  0, & \text{otherwise} 
\end{cases} \]

**Note:**

1. We can shift \( h[n] \) above to get causality (and linear phase)

2. This shows that the optimal filter minimizing the energy criterion is nothing but the rectangular window truncation!

\( \rightarrow \) We have already seen this is not such a great filter design.

\( \rightarrow \) Maybe energy criterion isn't such a hot idea!
CRITERION 2: "Minimize maximum error"

\[ \min_{h \in \text{passband or stopband}} \max_{\text{frequencies in}} |H_d(\omega) - H(\omega)| \]

- IDEA makes good engineering sense
  (minimize the max error; make response \( H(\omega) \) meet specs to within \( \pm 8 \) at all frequencies with \( \delta \) as small as possible)

- Results in good designs (they work well in real-world DSP problems)

- Also, spreads error out over passband and stopband rather than concentrating at the band edges like a window design.
Let's consider Type I (symmetric, odd-length) FIR filters. Recall that the DTFT of a Type I filter is

\[ H(\omega) = h[0] + 2 \sum_{k=1}^{m} h[k] \cos(k \omega) \]

\[ \equiv \sum_{k=0}^{m} \alpha_k \cos(k \omega) \]

(\text{Chebyshev Polynomial})

\[ \text{DTFT} \quad H(\omega) = \sum_{k=0}^{M} \alpha_k \cos(k \omega) \]

\[ \text{Simple relationship between } \alpha_k \text{'s and } \]

\[ \begin{align*}
  h[0] &= \alpha_0 \\
  h[k] &= \frac{1}{2} \alpha_k & \text{for } k > 0
\end{align*} \]

\text{(Note: Noncausal. We will match desired magnitude response, then shift to get causality.)}

\textbf{Objectives:} Choose \( \alpha_k \)’s to minimize

\[ \max_{\omega \in \text{passband} \cup \text{stopband}} \left| H_d(\omega) - \sum_{k=0}^{M} \alpha_k \cos(k \omega) \right| \]

\textbf{Question:} How?
The Alternation Theorem: (from mathematical Approximation Theory)

Consider the weighted approximation error

$$E(w) = W(w) \left| H_d(w) - \sum_{k=0}^{M} \alpha_k \cos(kw) \right|$$

$W(F)$ represents a weighting function that can be used to select different error bounds in passband and stopband.

The Alternation Theorem tells us that there are at least $M+2$ frequencies, $k, k=1, 2, \ldots, M+2$, called extremal frequencies where

1. The error alternates between two equal maxima and minima (extrema)
   $$E(\omega_k) = -E(\omega_{k+1}) \quad k=1, 2, \ldots, M+1$$

2. The error at the frequencies $\omega_k$ equals the maximum absolute error
   $$|E(\omega_k)| = \max \{ |E(\omega)| \} \quad k=1, 2, \ldots, M+2$$
Now, \( \int |H_d(w) - H(w)|^2 \, dF \)

\[ = \int \left| \sum_{n=-\infty}^{\infty} d[n] e^{-jwn} \right|^2 dw \]

\[ = \int \left( \sum_{n=-\infty}^{\infty} d[n] e^{-jwn} \right) \left( \sum_{m=-\infty}^{\infty} d[m] e^{-jwm} \right)^* \, dw \]

\[ = \int \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} d[n] d[m] e^{-jwn} (n-m) \, dw \]

\[ = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} d[n] d[m] \int \frac{1}{\pi} e^{-j\omega(n-m)} \, d\omega \]

\[ = \left\{ \begin{array}{ll}
1, & n=m \\
0, & n \neq m
\end{array} \right. \]

because of periodicity

\[ = \sum_{n=-\infty}^{\infty} d[n]^2 \]

\[ = \sum_{n=-\infty}^{\infty} |h_d[n] - h_c[n]|^2 \]

\[ = \sum_{n=-L}^{L} |h_d[n] - h_c[n]|^2 + \sum_{n=L+1}^{\infty} |h_d[n]|^2 \]

\[ \text{this term is minimum} \]

\[ (0) \text{ when } h_c[n] = h_d[n] \]

\[ \text{can't do anything about the third term} \]
Theorem: there exist at least \( m+2 \) extremal frequencies where we achieve the maximum error.

Let's find the extremal frequencies and choose \( \alpha \) to minimize the error at these points.

Requires iterative methods.

Most popular method is the **Parks-McClellan Algorithm** which uses the **Remez Exchange Algorithm**.

Resulting FIR filter design is called an **Equiripple Approximation**.
Example

Lowpass Design

Specs:

\[ 0.99 \leq |H(w)| \leq 1.01, \quad |w| \leq 0.4\pi \]
\[ |H(w)| \leq 0.001, \quad 0.6\pi \leq |w| \leq \pi \]

\[ \Rightarrow \delta_1 = 0.01, \quad \delta_2 = 0.001, \quad \omega_p = 0.4\pi \]
\[ \omega_s = 0.6\pi \]
\[ K = \frac{\delta_1}{\delta_2} = 10 \]

Note:

\[ \delta_1 \neq \delta_2 \Rightarrow \text{use weight function } K = 10 \]

Matlab Command:

\[ h = \text{remez}(N-1, f, m, w) \]

\[ \text{filter length } N \]

\[ \begin{bmatrix} 0 \ & \omega_p & \pi \ & 1 \end{bmatrix} \]

\[ \begin{bmatrix} 0 \ & 0.4 \ & 0.6 \ & 1 \end{bmatrix} \]

\[ \text{Increase } N \text{ until } \delta_1 \text{ and } \delta_2 \text{ specs are met} \]
\[ h = \text{remez}(N-1, f, m, w) \]

**Resulting Design**
How the Remez Exchange Works

1. Start with trial set of frequencies 
   \( w_1, w_2, \ldots, w_{m+2} \)

2. Force satisfaction of the alternation conditions at \( w_1, \ldots, w_{m+1} \)
   
   Set 
   \[
   E(w_k) = \epsilon \\
   E(w_{k+1}) = -\epsilon
   \]
   \( k = 1, \ldots, m+1 \)

   (we don't know max error, so \( \epsilon > 0 \)
   is picked arbitrarily)

3. Now we have \( m+1 \) unknown coefficients \( a_k \), and unknown true value of \( \epsilon \). We have \( m+2 \) equations:
   \[
   (-1)^k \epsilon = W(w_k) | H_d(w_k) - H(w_k) |
   \]
   \( k = 1, 2, \ldots, m+2 \)

   \( \Rightarrow \) solve for \( a_k \)'s and \( \epsilon \).

4. Compute true extremal frequencies for new \( a_k \)'s.
   These may not satisfy alternation condition.
   So go back and repeat (2) and (3)

(Repeat until convergence)

\* Check out built-in Matlab command \( \text{remez} \)