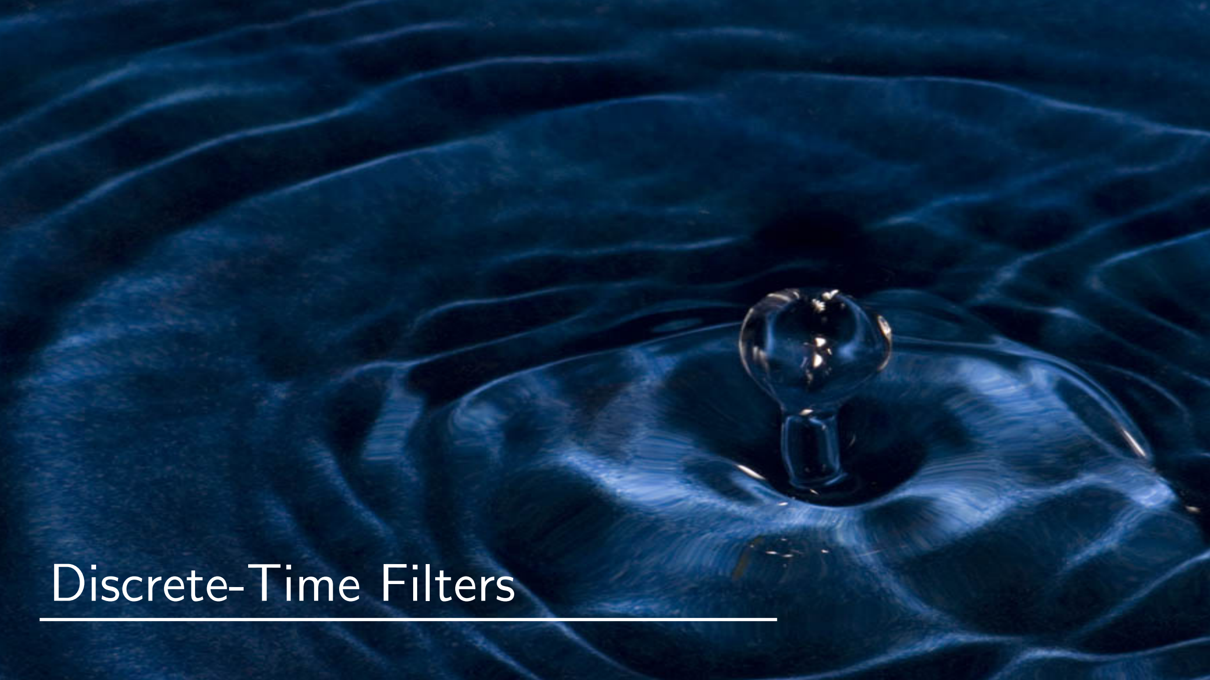


Discrete-Time Filters

Putting LTI Systems to Work

$$x \longrightarrow \boxed{h} \longrightarrow y$$

$$y[n] = x[n] * h[n] = \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

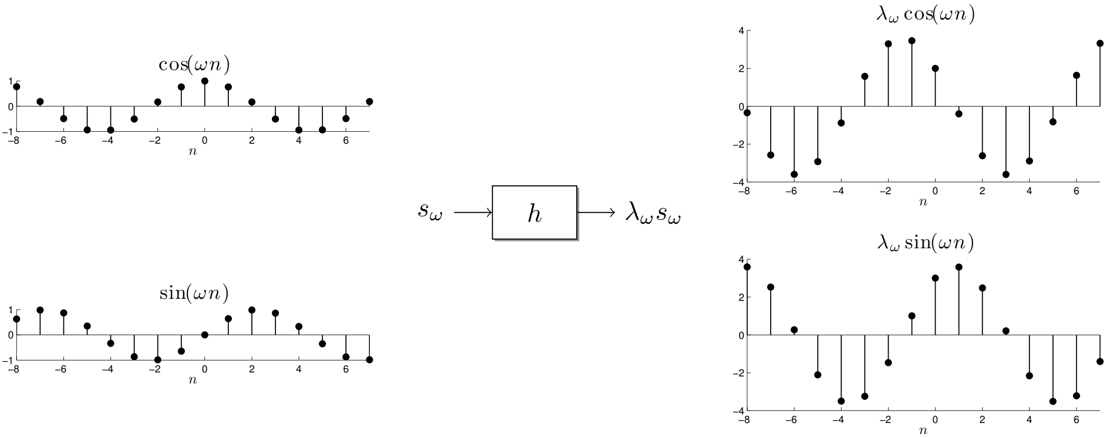
$$Y(z) = H(z)\, X(z), \qquad Y(\omega) = H(\omega)\, X(\omega)$$

- **Goal:** Design a LTI system to perform a certain **task** in some application

- Key questions:
  - What is the **range of tasks** that an LTI system can perform?
  - What are the **parameters** under our control for design purposes?

What Do LTI Systems Do? Recall Eigenanalysis

- LTI system **eigenvectors**: $s_\omega[n] \;=\; e^{j\omega n}$
- LTI system **eigenvalues**: $\lambda_\omega \;=\; H(\omega) \;=\; \sum_{n=-\infty}^{\infty} h[n]\, e^{-j\omega n}$ (frequency response)



LTI Systems Filter Signals

$$x \longrightarrow \boxed{h} \longrightarrow y$$

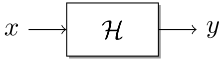
- Important interpretation of $Y(\omega) = H(\omega)\, X(\omega)$

$$x[n] = \int_{-\pi}^{\pi} X(\omega)\, e^{j\omega n}\, \tfrac{d\omega}{2\pi} \longrightarrow \boxed{h} \longrightarrow y[n] = \int_{-\pi}^{\pi} H(\omega)\, X(\omega)\, e^{j\omega n}\, \tfrac{d\omega}{2\pi}$$

- An LTI system processes a signal $x[n]$ by amplifying or attenuating the sinusoids in its Fourier representation (DTFT) $X(\omega)$ by the complex factor $H(\omega)$

- Inspires the terminology that $X(\omega)$ is **filtered** by $H(\omega)$ to produce $Y(\omega)$

Design Parameters of Discrete-Time Filters (LTI Systems)

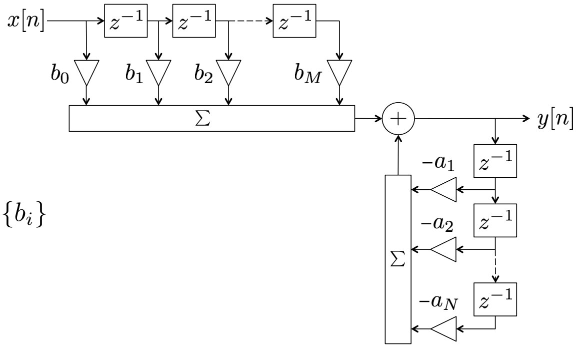


- Impulse response: $h[n]$

- Transfer function: $H(z)$
  - poles and zeros

- Frequency response: $H(\omega)$

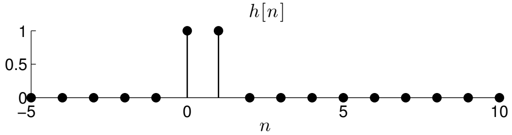
- Moving/recursive average parameters: $\{a_i\}$, $\{b_i\}$



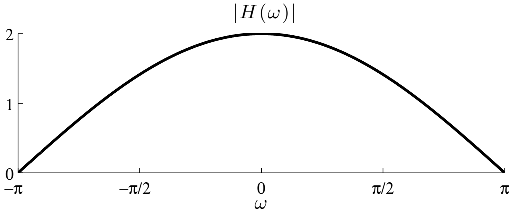
Filters Archetypes: Low-Pass

- Ideal low-pass filter

- Example low-pass impulse response $h[n]$



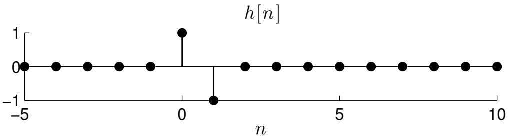
- Example frequency response $|H(\omega)|$



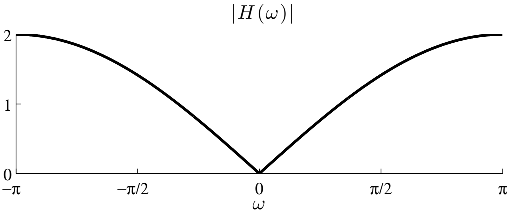
Filters Archetypes: High-Pass

- Ideal high-pass filter

- Example high-pass impulse response $h[n]$



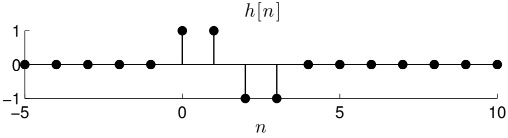
- Example frequency response $|H(\omega)|$



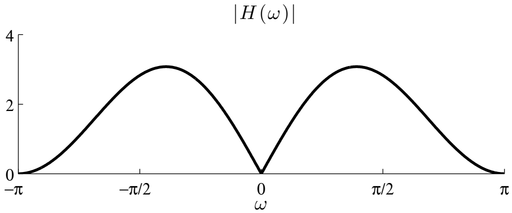
Filters Archetypes: Band-Pass

- Ideal band-pass filter

- Example band-pass impulse response $h[n]$



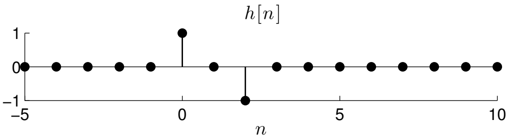
- Example frequency response $|H(\omega)|$



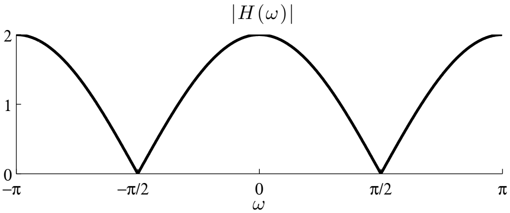
Filters Archetypes: Band-Stop

- Ideal band-stop filter

- Example band-stop impulse response $h[n]$



- Example frequency response $|H(\omega)|$



Summary

- Now that we understand what LTI systems do, we can **design** them to accomplish certain tasks

- An LTI system processes a signal $x[n]$ by amplifying or attenuating the sinusoids in its Fourier representation (DTFT)

- Equivalent design parameters of a discrete-time filter
  - Impulse response: $h[n]$
  - z-transform: $H(z)$ (poles and zeros)
  - Frequency response: $H(\omega)$
  - Moving/recursive average parameters: $\{a_i\}$, $\{b_i\}$

- Archetype filters: Low-pass, high-pass, band-pass, band-stop

- We will emphasize infinite-length signals, but the situation is similar for finite-length signals

Discrete-Time Filter Design

Recall Discrete-Time Filter

$$x \longrightarrow \boxed{h} \longrightarrow y$$

$$y[n] \;=\; x[n] * h[n] \;=\; \sum_{m=-\infty}^{\infty} h[n-m]\, x[m]$$

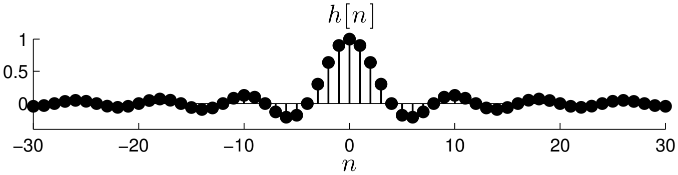
$$Y(z) \;=\; H(z)\, X(z), \qquad Y(\omega) \;=\; H(\omega)\, X(\omega)$$

- A discrete-time filter fiddles with a signal's Fourier representation
- Recall the filter archetypes: ideal low-pass, high-pass, band-pass, band-stop filters

Ideal Lowpass Filter

- Ideal low-pass filter frequency response $H(\omega) = \begin{cases} 1 & -\omega_c \leq \omega \leq \omega_c \\ 0 & \text{otherwise} \end{cases}$

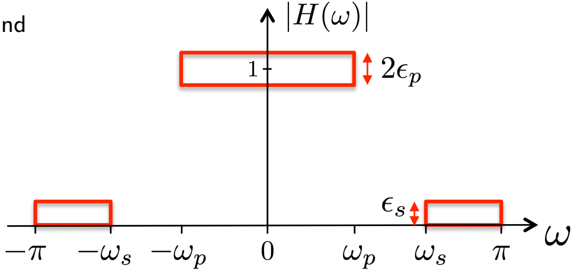
- Impulse response is the infamous "sinc" function $h[n] \;=\; 2\omega_c \frac{\sin(\omega_c n)}{\omega_c n}$



- Problems:
  - System is not BIBO stable! $(\sum_n |h[n]| = \infty)$
  - Infinite computational complexity ($H(z)$ is not a rational function)

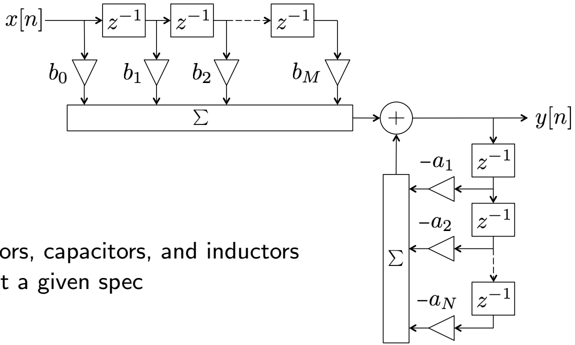
Filter Specification

- Find a filter of minimum complexity that meets a given **specification**
- Example: Low-pass filter
  - Pass-band edge frequency: ω_p
  - Stop-band edge frequency: ω_s
  - Between pass- and stop-bands: transition band
  - Pass-band ripple ϵ_p (often expressed in dB)
  - Stop-band ripple ϵ_s (often expressed in dB)



- Clearly, the tighter the specs, the more complex the filter

Two Classes of Discrete-Time Filters



- Infinite impulse response (IIR) filters
  - Uses both moving and recursive averages
  - $H(z)$ has both poles and zeros
  - Related to "analog" filter design using resistors, capacitors, and inductors
  - Generally have the lowest complexity to meet a given spec

- Finite impulse response (FIR) filters
  - Uses only moving average
  - $H(z)$ has only zeros
  - Unachievable in analog using resistors, capacitors, and inductors
  - Generally higher complexity (than IIR) to meet a given spec
  - But can have linear phase (big plus)

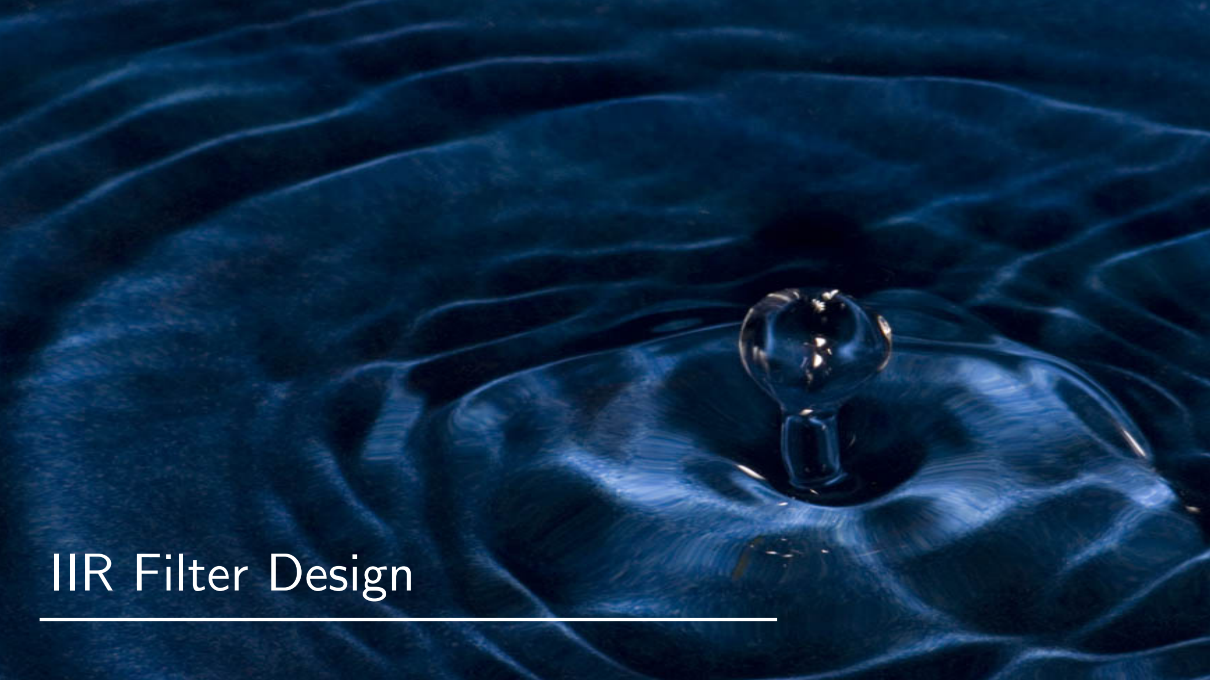
Summary

- A discrete-time filter fiddles with a signal's Fourier representation

- "Ideal" filters are not practical
  - System is not BIBO stable!
  - Infinite computational complexity ($H(z)$ is not a rational function)

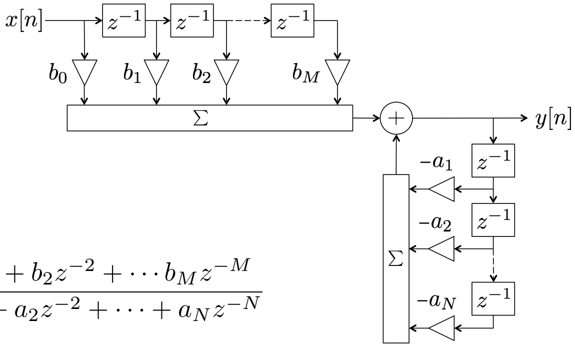
- Filter design: Find a filter of minimum complexity that meets a given spec

- Two different types of filters (IIR, FIR) mean two different types of filter design



IIR Filter Design

IIR Filters



- Use both moving and recursive averages
- Transfer function has both **poles** and **zeros**

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}}$$

$$= z^{N-M} \frac{(z-\zeta_1)(z-\zeta_2)\cdots(z-\zeta_M)}{(z-p_1)(z-p_2)\cdots(z-p_N)}$$

- We **design** an IIR filter by specifying the locations of its poles and zeros in the z-plane
- Generally can satisfy a spec with lower complexity than FIR filters

IIR Filters from Analog Filters

- In contrast to FIR filter design, IIR filters are typically designed by a two-step procedure that is slightly ad hoc

- **Step 1:** Design an **analog** filter (for resistors, capacitors, and inductors) using the Laplace transform $H_L(s)$ (this theory is well established but well beyond the scope of this course)

- **Step 2:** Transform the analog filter into a discrete-time filter using the **bilinear transform** (a conformal map from complex analysis)

$$s \;=\; c\,\frac{z-1}{z+1}$$

- The discrete-time filter's transfer function is given by

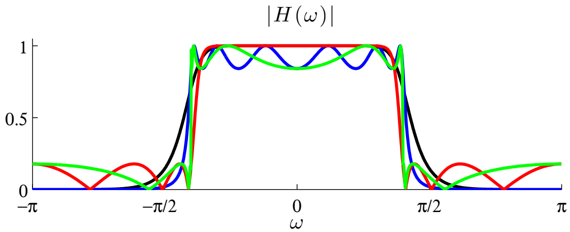
$$H(z) \;=\; H_L(s)\big|_{s=c\frac{z-1}{z+1}}$$

Three Important Classes of IIR Filters

- Butterworth filters
  - `butter` command in Matlab
  - No ripples (oscillations) in $|H(\omega)|$
  - Gentlest transition from pass-band to stop-band for a given order

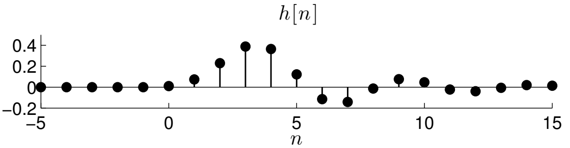
- Chebyshev filters
  - `cheby1` and `cheby2` commands in Matlab
  - Ripples in either pass-band or stop-band

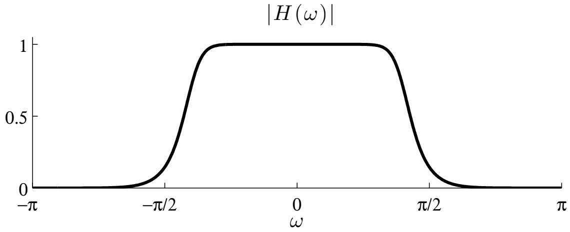
- Elliptic filters
  - `ellip` command in Matlab
  - Ripples in both pass-band and stop-band
  - Sharpest transition from pass-band to stop-band for a given order (use with caution!)

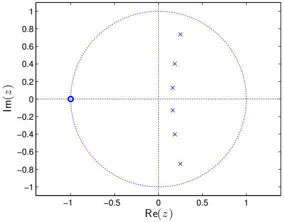


Butterworth IIR Filter

- "Maximally flat" frequency response
  - Largest number of derivatives of $|H(\omega)|$ equal to 0 at $\omega = 0$ and π
- N zeros and N poles
  - Zeros are all at $z = -1$
  - Poles are located on a circle inside the unit circle
- Example: $N = 6$ using `butter` command in Matlab





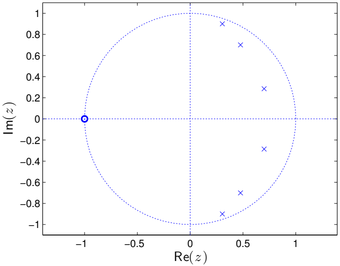


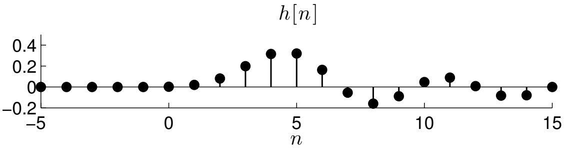
Chebyshev Type 1 IIR Filter

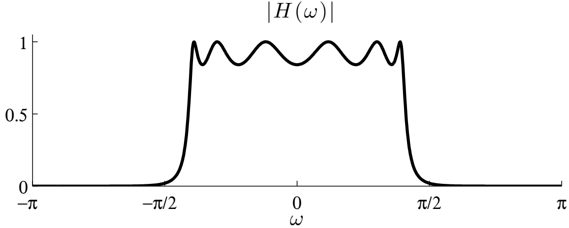
- Ripples/oscillations (of equal amplitude) in the pass-band and not in the stop-band

- N zeros and N poles
  - Zeros are all at $z = -1$
  - Poles are located on an ellipse inside the unit circle

- Example: $N = 6$ using `cheby1` command in Matlab





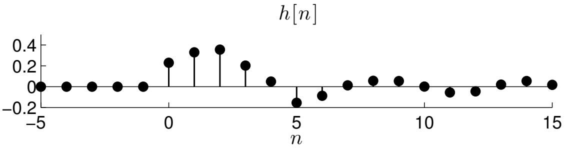


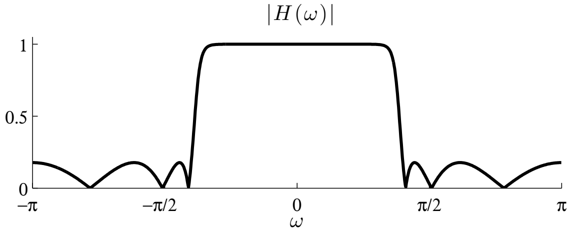
Chebyshev Type 2 IIR Filter

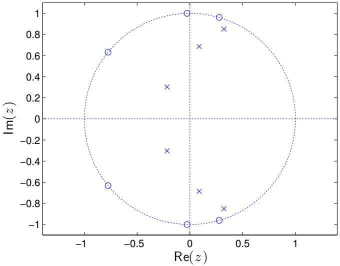
- Ripples/oscillations (of equal amplitude) in the stop-band and not in the pass-band

- N zeros and N poles
  - Zeros are distributed on unit circle
  - Poles are located on an ellipse inside the unit circle

- Example: $N = 6$ using `cheby2` command in Matlab

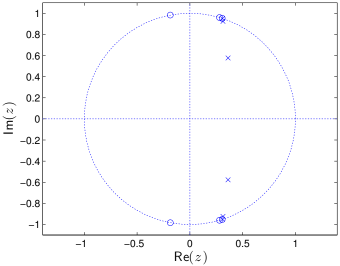


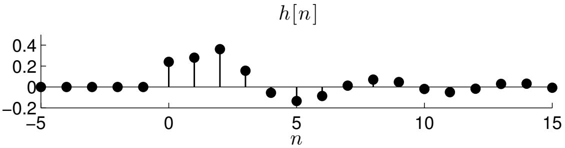


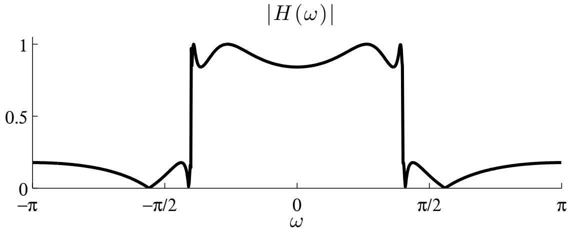


Elliptic IIR Filter

- Ripples/oscillations in both the stop-band and pass-band
- N zeros and N poles
  - Zeros are clustered on unit circle near ω_p
  - Poles are clustered close to unit circle near ω_p
- Example: $N = 6$ using `ellip` command in Matlab

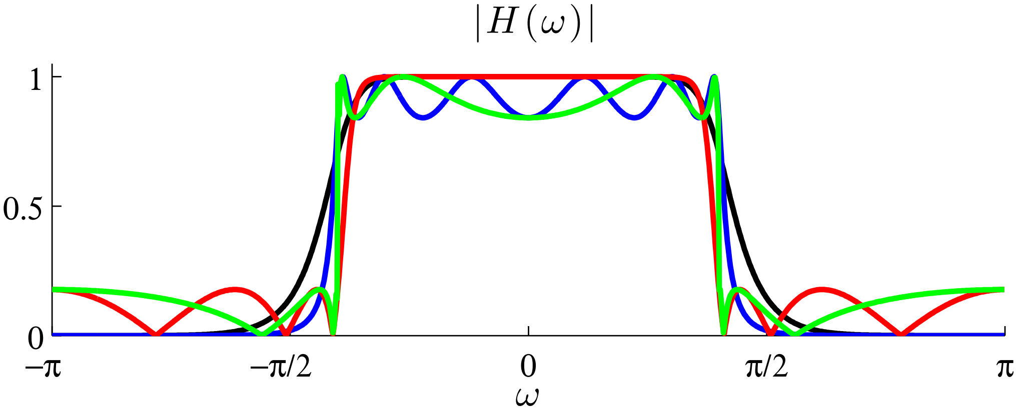






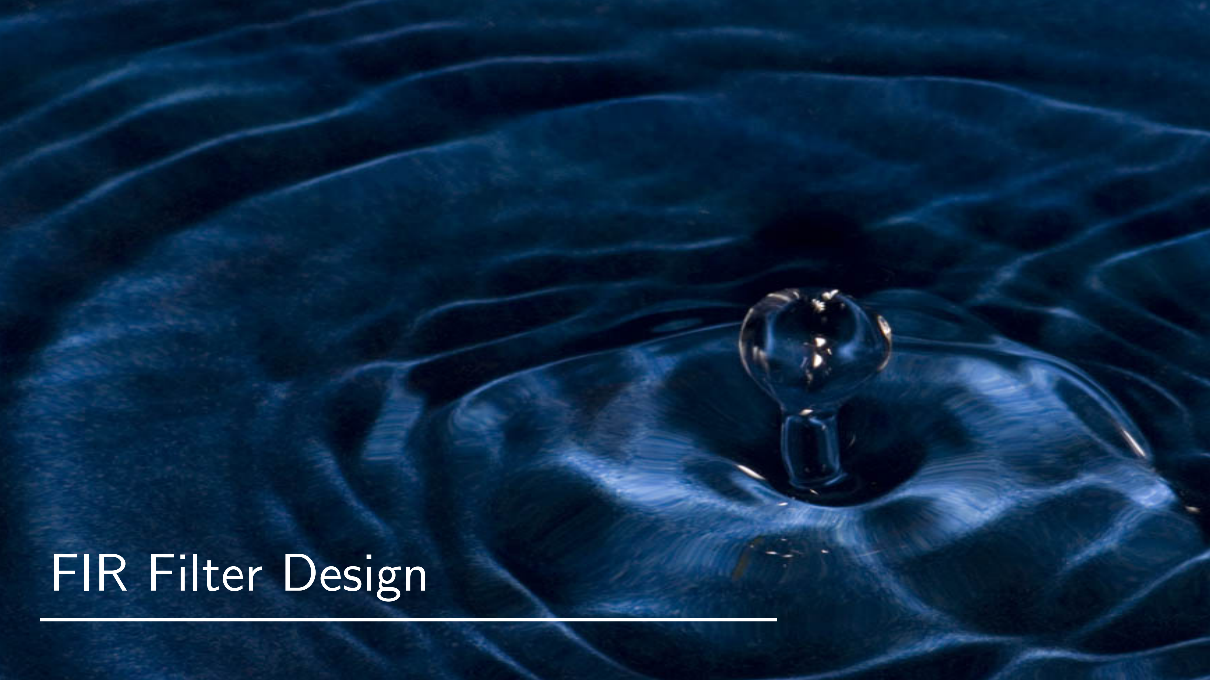
IIR Filter Comparison

- Butterworth (black), Chebyshev 1 (blue), Chebyshev 2 (red), Elliptic (green)



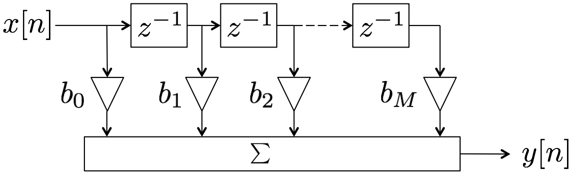
Summary

- IIR filters use use both moving and recursive averages and have both poles and zeros
- Typically designed by transforming an analog filter design (for use with resistors, capacitors, and inductors) into discrete-time via the bilinear transform
- Four families of IIR filters: Butterworth, Chebyshev (1,2), Elliptic
- Useful Matlab commands for choosing the filter order N that meets a given spec: `butterord`, `cheby1ord`, `cheby2ord`, `ellipord`



FIR Filter Design

FIR Filters



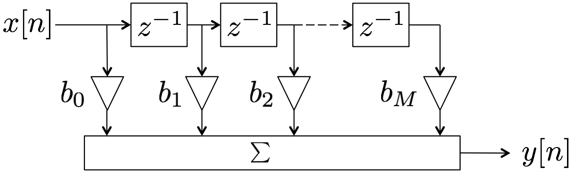
- Use only a moving average
- Transfer function has only **zeros** (and trivial poles at $z = 0$)

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots b_M z^{-M}$$

$$= z^{-M}(z - \zeta_1)(z - \zeta_2) \cdots (z - \zeta_M)$$

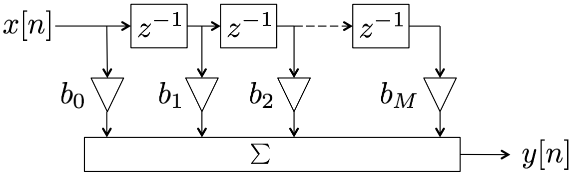
- We **design** an FIR filter by specifying the values of the **taps** $b_0, b_1, \ldots, b_M$ (this is equivalent to specifying the locations of the zeros in the z-plane)
- Generally require a higher complexity to meet a given spec than an IIR filter

FIR Filters Are Interesting



- FIR filters are **specific to discrete-time**; they cannot be built in analog using R, L, C
- FIR filters are always BIBO stable
- FIR filters can be designed to **optimally** meet a given spec
- Unlike IIR filters and all analog filters, FIR filters can have (generalized) **linear phase**
  - A nonlinear phase response $\angle H(\omega)$ distorts signals as they pass through the filter
  - Recall that a linear phase shift in the DTFT is equivalent to a simple time shift in the time domain

Impulse Response of an FIR Filter



- Easy to see by inputting $x[n] = \delta[n]$ that the **impulse response** of an FIR filter consists of the taps weights

n	$\cdots$	-2	-1	0	1	2	$\cdots$	M	$M+1$	$M+2$	$\cdots$
$h[n]$	$\cdots$	0	0	b_0	b_1	b_2	$\cdots$	b_M	0	0	$\cdots$

- Note: Filter **order** $= M$; filter **length** $= M + 1$

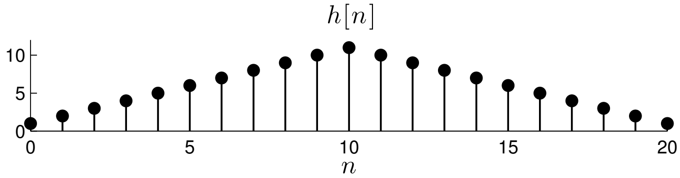
Symmetric FIR Filters

- Unlike IIR filters, FIR filters can be causal and have (generalized) linear phase

- Linear phase filters must have a symmetric impulse response
  - Four cases: even/odd length, even/odd symmetry
  - Different symmetries can be useful for different filter types (low-pass, high-pass, etc.)

- We will focus here on low-pass filters with **odd-length**, **even-symmetric** impulse response
  - Odd length: $M+1$ is odd (M is even)
  - Even symmetric (around the center of the filter): $h[n] = h[M-n], \;\; n = 0, 1, \ldots, M$

- Example: Length $M+1 = 21$



Frequency Response of a Symmetric FIR Filter (1)

- Compute frequency response when $h[n]$ is **odd-length** and **even-symmetric** $(h[n] = h[M-n])$

$$
\begin{aligned}
H(\omega) &= \sum_{n=0}^{M} h[n]\, e^{-j\omega n} = \sum_{n=0}^{M/2-1} h[n]\, e^{-j\omega n} + h[M/2]\, e^{-j\omega M/2} + \sum_{n=M/2+1}^{M} h[n]\, e^{-j\omega n} \\
&= \sum_{n=0}^{M/2-1} h[n]\, e^{-j\omega n} + h[M/2]\, e^{-j\omega M/2} + \sum_{n=M/2+1}^{M} h[M-n]\, e^{-j\omega n} \\
&= \sum_{n=0}^{M/2-1} h[n]\, e^{-j\omega n} + h[M/2]\, e^{-j\omega M/2} + \sum_{r=0}^{M/2-1} h[r]\, e^{-j\omega (M-r)} \\
&= h[M/2]\, e^{-j\omega M/2} + \sum_{n=0}^{M/2-1} h[n] \left(e^{-j\omega n} + e^{j\omega (n-M)} \right)
\end{aligned}
$$

Frequency Response of a Symmetric FIR Filter (2)

- Compute frequency response when $h[n]$ is **odd-length** and **even-symmetric** $(h[n] = h[M-n])$

$$
\begin{aligned}
H(\omega) &= h[M/2]\, e^{-j\omega M/2} + \sum_{n=0}^{M/2-1} h[n] \left(e^{-j\omega n} + e^{j\omega(n-M)} \right) \\
&= h[M/2]\, e^{-j\omega M/2} + \sum_{n=0}^{M/2-1} h[n]\, e^{-j\omega M/2} \left(e^{-j\omega(n-M/2)} + e^{j\omega(n-M/2)} \right) \\
&= \left(h[M/2] + \sum_{n=0}^{M/2-1} 2\, h[n] \cos(\omega(n - M/2)) \right) e^{-j\omega M/2} \\
&= A(\omega)\, e^{-j\omega M/2}
\end{aligned}
$$

Generalized Linear Phase FIR Filters

- Frequency response when $h[n]$ is **odd-length** and **even-symmetric** ($h[n] = h[M-n]$)

$$H(\omega) \;=\; A(\omega)\, e^{-j\omega M/2}$$

  with

$$A(\omega) \;=\; h[M/2] \;+\; \sum_{n=0}^{M/2-1} 2\, h[n] \cos(\omega(n - M/2))$$

- $A(\omega)$ is called the **amplitude** of the filter; it plays a role like $|H(\omega)|$ since

$$|H(\omega)| \;=\; |A(\omega)|$$

  However, $A(\omega)$ is not necessarily ≥ 0

- $e^{-j\omega M/2}$ is a **linear phase shift**
  $H(\omega)$ has linear phase except when $A(\omega)$ changes sign, in which case its phase jumps by π rad

FIR Filter Design

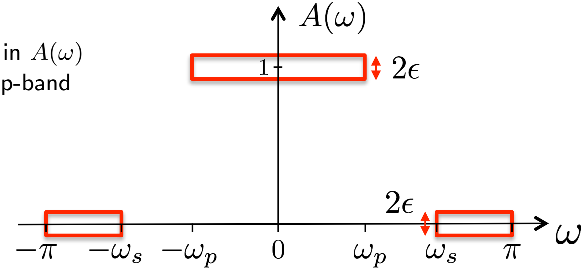
- Frequency response when $h[n]$ is **odd-length** and **even-symmetric** $(h[n] = h[M-n])$

$$H(\omega) \;=\; A(\omega)\, e^{-j\omega M/2}$$

  with

$$A(\omega) \;=\; h[M/2] \;+\; \sum_{n=0}^{M/2-1} 2\, h[n] \cos(\omega(n - M/2))$$

- Design of $H(\omega)$ is equivalent to the design of $A(\omega)$; spec changes slightly
  - Stop-band spec now allows negative values in $A(\omega)$
  - For simplicity, same ϵ in both pass- and stop-band (this is easy to generalize)



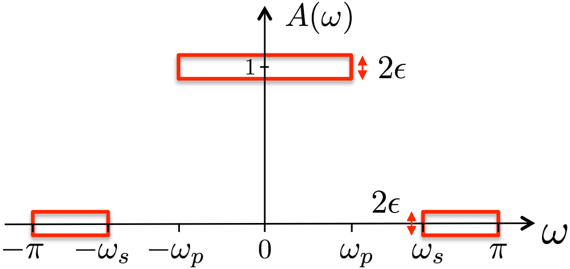
Optimal FIR Filter Design

- **Goal:** Find the **optimal** $A(\omega)$ (in terms of shortest length $M+1$) that meets the specs

$$A(\omega) \;=\; h[M/2] \;+\; \sum_{n=0}^{M/2-1} 2\,h[n]\cos(\omega(n-M/2))$$

- Parameters under our control: The $M/2+1$ filter taps $h[n]$, $n=0,1,\ldots,M/2$

- Problem solved by James McClellan and Thomas Parks at Rice University (1971) "Parks-McClellan Filter Design"



Key Ingredients of Optimal FIR Filter Design

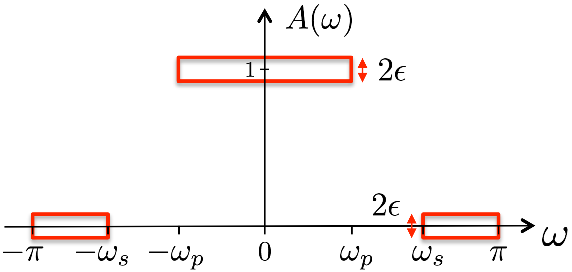
- **Goal:** Find the **optimal** $A(\omega)$ (in terms of shortest length $M+1$) that meets the specs

$$A(\omega) \;=\; h[M/2] \;+\; \sum_{n=0}^{M/2-1} 2\,h[n]\cos(\omega(n-M/2))$$

- **Ripples:** $A(\omega)$ oscillates $M/2$ times in the interval $0 \le \omega \le \pi$

- **Equiripple property:** The oscillations of the optimal $A(\omega)$ are all the same size

- **Alternation Theorem:** The optimal $A(\omega)$ will touch the error bounds $M/2+2$ times in the interval $0 \le \omega \le \pi$



Remez Exchange Algorithm for Optimal FIR Filter Design

- **Goal:** Find the **optimal** $A(\omega)$ (in terms of shortest length $M+1$) that meets the specs

$$A(\omega) \;=\; h[M/2] \;+\; \sum_{n=0}^{M/2-1} 2\,h[n]\cos(\omega(n-M/2))$$

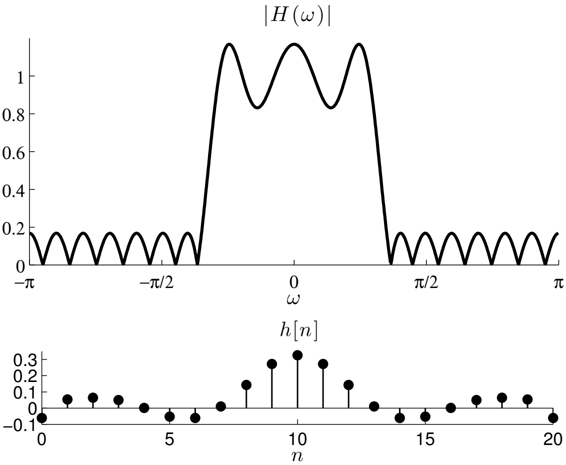
- **Alternation Theorem:** The optimal $A(\omega)$ will touch the error bounds $M/2+2$ times in the interval $0 \le \omega < \pi$

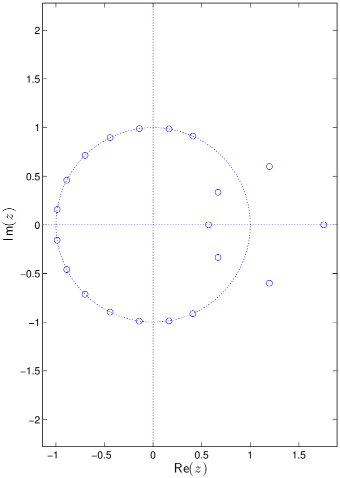
- Parks and McClellan proposed the **Remez Exchange Algorithm** to find the $h[n]$ such that $A(\omega)$ satisfies the alternation theorem

- Matlab command `firpm` and `firpmord` (be careful with the parameters)

Example 1: Optimal FIR Filter Design (1)

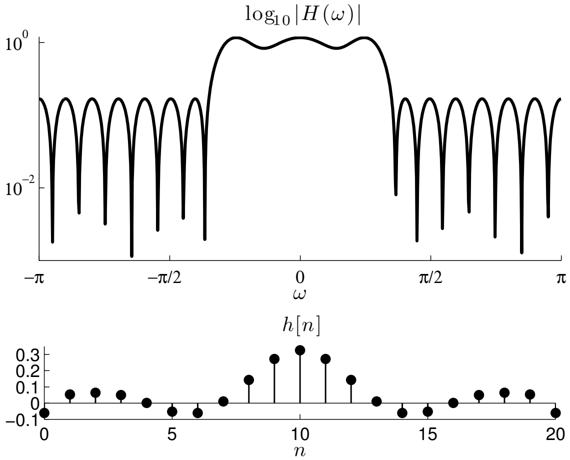
- Optimal low-pass filter of length $M+1=21$ with $\omega_p=0.30\pi$, $\omega_s=0.35\pi$
- Note the $M/2+2=12$ alternations

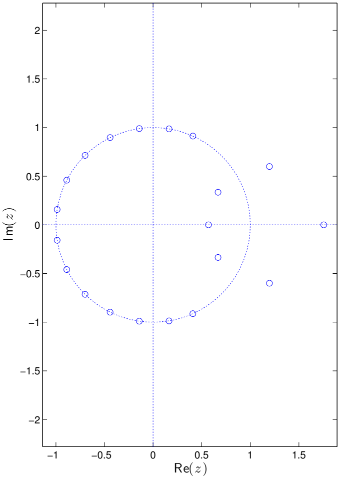




Example 1: Optimal FIR Filter Design (2)

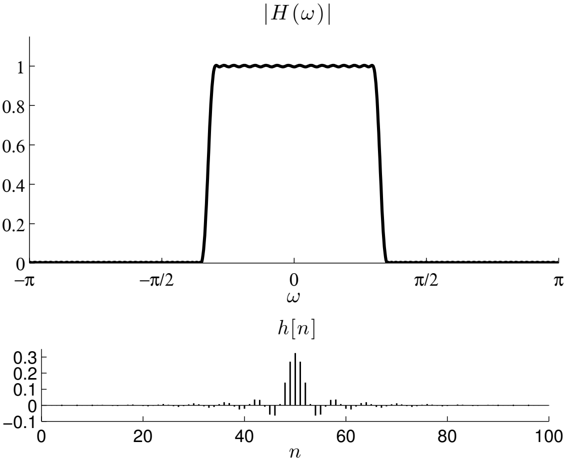
- Optimal low-pass filter of length $M+1=21$ with $\omega_p = 0.30\pi$, $\omega_s = 0.35\pi$
- Note the $M/2+2=12$ alternations

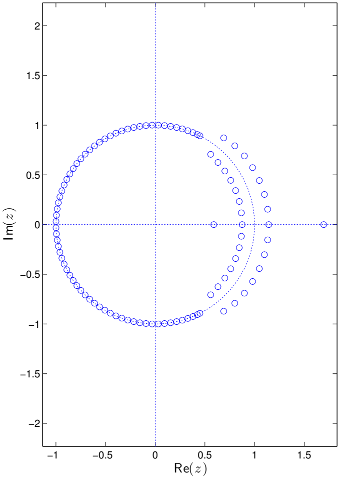




Example 2: Optimal FIR Filter Design (1)

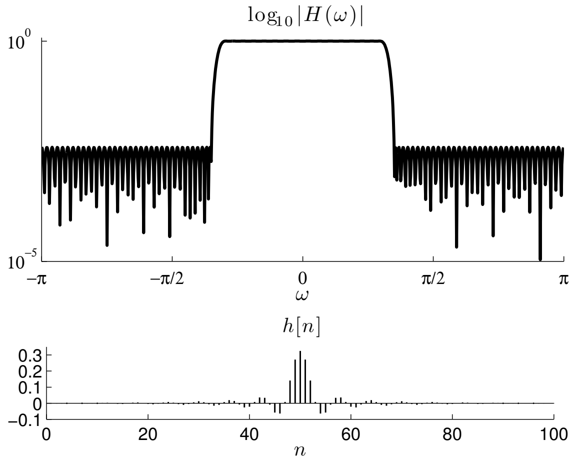
- Optimal low-pass filter of length $M+1 = 101$ with $\omega_p = 0.30\pi$, $\omega_s = 0.35\pi$
- Note the $M/2+2 = 52$ alternations

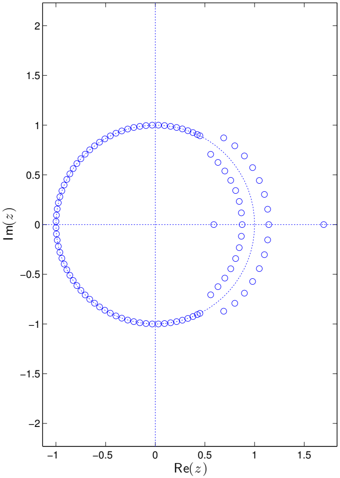




Example 2: Optimal FIR Filter Design (2)

- Optimal low-pass filter of length $M + 1 = 101$ with $\omega_p = 0.30\pi$, $\omega_s = 0.35\pi$
- Note the $M/2 + 2 = 52$ alternations



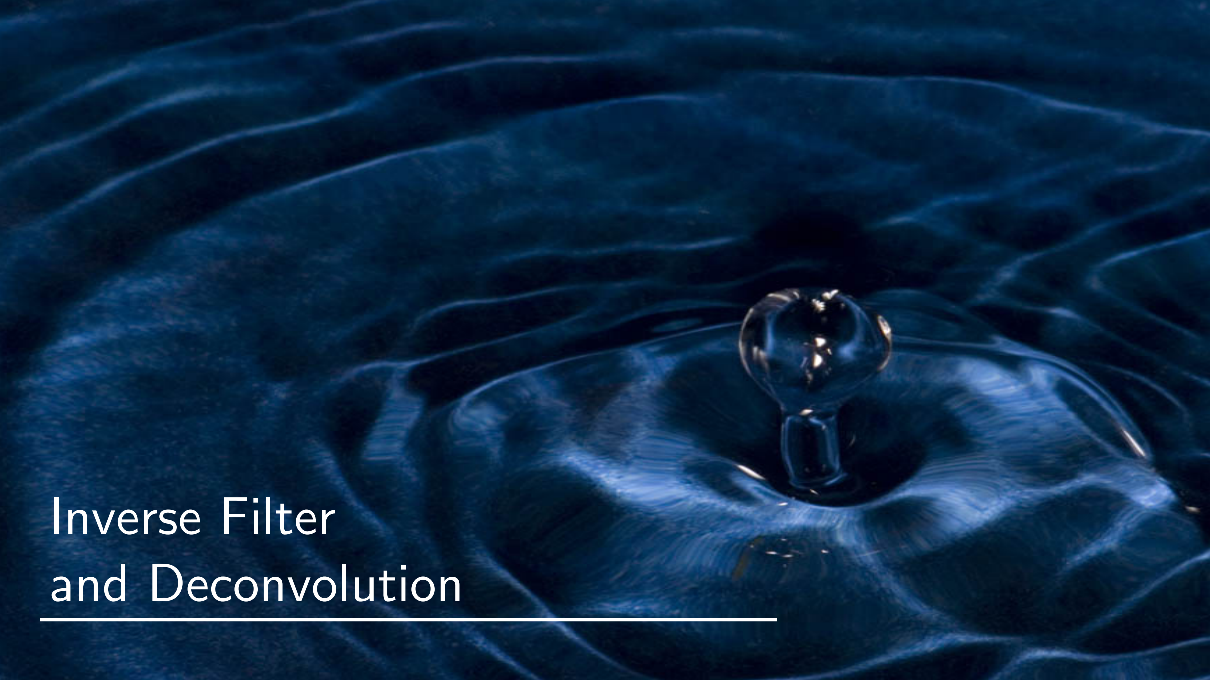


Matlab Example: Optimal FIR Filter Design

- Process a chirp signal through an optimal low-pass filter with
  - Length $M + 1 = 101$
  - $\omega_p = \pi/3$
  - $\omega_s = \pi/2$

Summary

- FIR filters correspond to a moving average and have **only zeros** (no poles)
- FIR filters are specific to discrete-time; they cannot be built in analog using R, L, C
- Symmetrical FIR filters have (generalized) linear phase, which is impossible with IIR or analog filters
- Design **optimal** FIR filters using the Parks-McClellan algorithm (Remez exchange algorithm)
- FIR filters are always BIBO stable and very numerically stable (to coefficient quantization, etc.)
- Generally require a higher complexity to meet a given spec than an IIR filter, but the benefits can outweigh the computational cost



Inverse Filter and Deconvolution

LTI Signal Degradations

- In many important applications, we do not observe the signal of interest x but rather a version y processed by an LTI system with impulse response g

$$x \longrightarrow \boxed{\;g\;} \longrightarrow y$$

- Examples:
  - Digital subscriber line (DSL) communication (long wires)
  - Echos in audio signals
  - Camera blur due to misfocus or motion (2D)
  - Medical imaging (CT scans), ...

- **Goal:** Ameliorate the degradation by passing y through a second LTI system in the hopes that we can "cancel out" the effect of the first such that $\widehat{x} = x$

$$x \longrightarrow \boxed{\;g\;} \longrightarrow y \quad \longrightarrow \boxed{\;h\;} \longrightarrow \widehat{x}$$

LTI Signal Degradations in the z-Transform Domain

- **Goal:** Ameliorate the degradation by passing y through a second LTI system in the hopes that we can "cancel out" the effect of the first such that $\widehat{x} = x$

$$x \longrightarrow \boxed{\;g\;} \longrightarrow y \qquad \longrightarrow \boxed{\;h\;} \longrightarrow \widehat{x}$$

- Easy to understand using z-transform

$$\widehat{X}(z) \;=\; H(z)\,Y(z) \;=\; H(z)\,G(z)\,X(z)$$

- Therefore, in order to have $\widehat{x} = x$, and thus $\widehat{X}(z) = X(z)$, we need

$$H(z)\,G(z) \;=\; 1 \quad \text{or} \quad H(z) \;=\; \frac{1}{G(z)}$$

- $H(z) = \frac{1}{G(z)}$ is called the **inverse filter**, and this process is called **deconvolution**

Inverse Filter – Poles and Zeros

- If the degradation filter $G(z)$ is a rational function with zeros $\{\zeta_i\}$ and poles $\{p_j\}$

$$G(z) = z^{N-M} \frac{(z-\zeta_1)(z-\zeta_2)\cdots(z-\zeta_M)}{(z-p_1)(z-p_2)\cdots(z-p_N)}$$

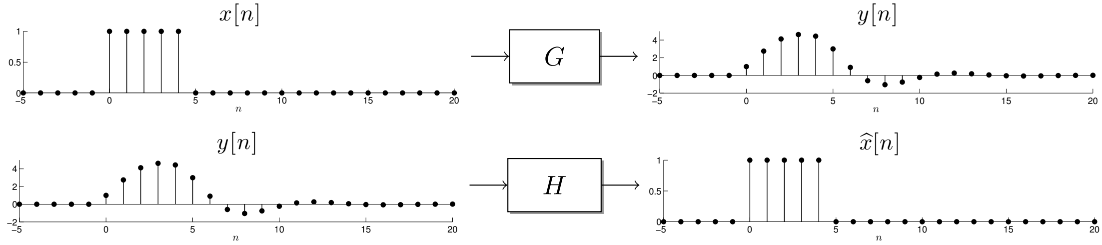
  then the inverse filter $H(z)$ is a rational function with <u>zeros</u> $\{p_j\}$ and <u>poles</u> $\{\zeta_i\}$

$$H(z) = \frac{1}{G(z)} = z^{M-N} \frac{(z-p_1)(z-p_2)\cdots(z-p_N)}{(z-\zeta_1)(z-\zeta_2)\cdots(z-\zeta_M)}$$

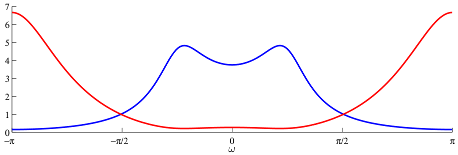
- Assuming that $G(z)$ and $H(z)$ are <u>causal</u>, if any of the zeros of $G(z)$ are outside the unit circle, then $H(z)$ is **not BIBO stable**, which means that the inverse filter does not exist

- When $G(z)$ is causal and all of its zeros are inside the unit circle, we say that it has **minimum phase**; in this case an exact inverse filter $H(z)$ exists

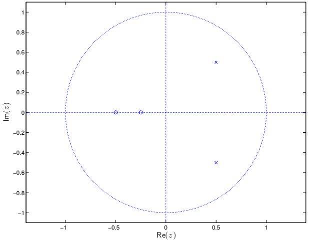
Example: Exact Inverse Filter



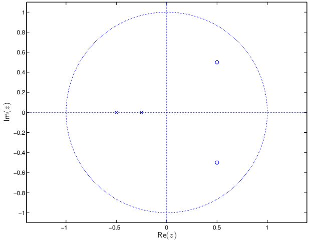
$G(\omega)$ (blue) and $H(\omega) = \frac{1}{G(\omega)}$ (red)



$G(z)$



$H(z)$



Approximate Inverse Filter

- When $G(z)$ is non-minimum phase, an exact inverse filter does not exist, because $\frac{1}{G(z)}$ has one or more poles outside the unit circle

- We can still find an **approximate** inverse filter by **regularizing** $\frac{1}{G(z)}$; for example

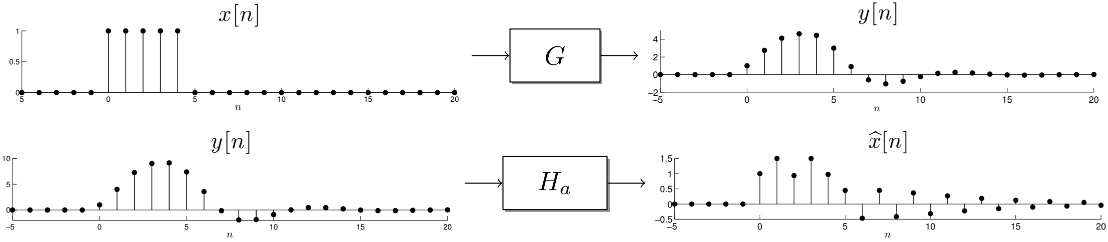
$$H_a(z) = \frac{1}{G(z) + r}$$

  where r is a constant (technically this is called Tikhonov regularization)

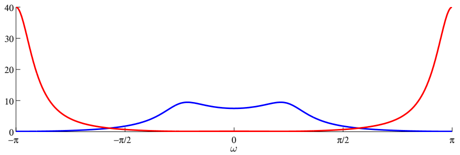
- Typically we try to choose the smallest r such that $H_a(z)$ is BIBO stable

- We no longer have $\widehat{x} = x$, but rather $\widehat{x} \approx x$

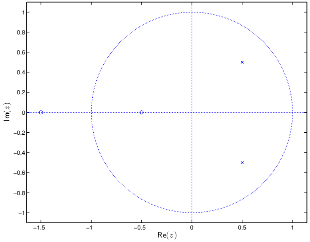
Example: Approximate Inverse Filter



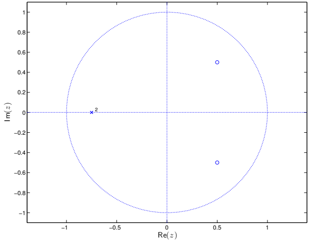
$G(\omega)$ (blue) and $H_a(\omega) = \frac{1}{G(\omega)+\frac{1}{16}}$ (red)



$G(z)$



$H_a(z)$



Summary

- **Deconvolution:** Ameliorate the degradation from an LTI system G by passing the degraded signal through a second LTI system H in the hopes that we can "cancel out" the effect of the first such that $\widehat{x} = x$

$$x \longrightarrow \boxed{\; g \;} \longrightarrow y \qquad \longrightarrow \boxed{\; h \;} \longrightarrow \widehat{x}$$

- **Inverse filter:** Poles/zeros of $G(z)$ become zeros/poles of $H(z)$

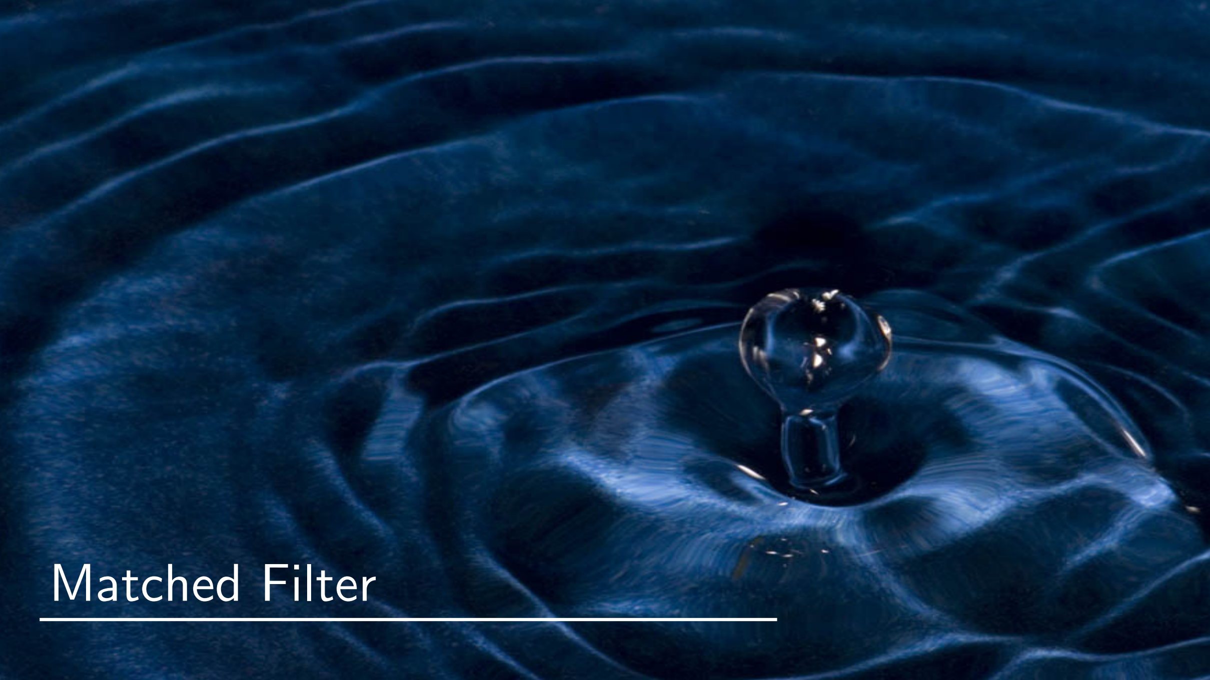
$$H(z) \;=\; \frac{1}{G(z)}$$

- Best case: When $G(z)$ is causal and all of its zeros are inside the unit circle, we say that it has **minimum phase**; in this case an exact inverse filter $H(z)$ exists

- Puzzler: What do we do when $N \neq M$ in $G(z)$?

- Advanced topics beyond the scope of this course: blind deconvolution, adaptive filters (LMS alg.)

Matched Filter



Inner Product and Cauchy Schwarz Inequality

- Recall the **inner product** (or dot product) between two signals x, y (whether finite- or infinite-length)

$$\langle y, x \rangle \;=\; \sum_{n} y[n]\, x[n]^*$$

- Recall the **Cauchy-Schwarz Inequality** (CSI)

$$0 \;\leq\; |\langle y, x \rangle| \;\leq\; \|y\|_2\, \|x\|_2$$

- Interpretation: The inner product $\langle y, x \rangle$ measures the **similarity** of y and x
  - Large value of $|\langle y, x \rangle| \;\Rightarrow\; y$ and x very similar
  - Small value of $|\langle y, x \rangle| \;\Rightarrow\; y$ and x very disimilar

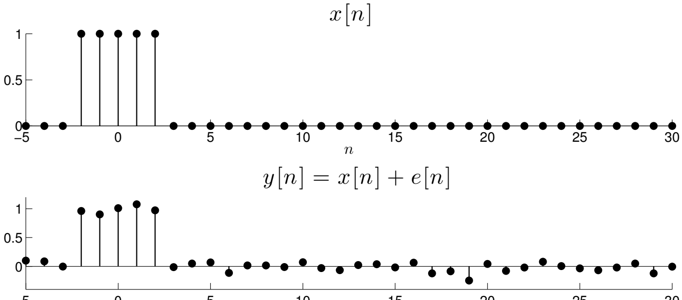
Signal Detection Using Inner Product

- We can determine if a **target signal** x of interest is present within a given signal y simply by computing the inner product and comparing it to a threshold $t > 0$

$$|d| \;=\; |\langle y, x\rangle| \begin{cases} \geq & t \quad \text{signal is present} \\ < & t \quad \text{signal is not present} \end{cases}$$

(Aside: In certain useful cases, this is the optimal way to detect a signal)

- Example: Detect the square pulse x in a noisy version $y = x + e$; we calculate $|d| = |\langle y, x\rangle| = 4.92$

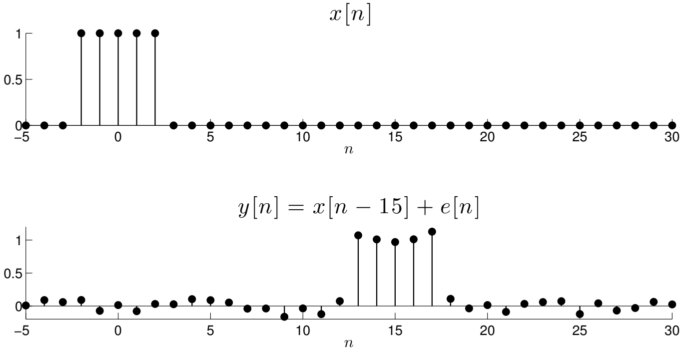


Signal Detection With Unknown Shift

- In many important applications, the target is **time-shifted** by some unknown amount ℓ

$$y[n] \;=\; x[n-\ell] + e[n]$$

- Example: Square pulse with shift $\ell = 15$



Solution: Signal Detection With Unknown Shift

- In many important applications, the target is **time-shifted** by some unknown amount ℓ

$$y[n] \;=\; x[n-\ell] + e[n]$$

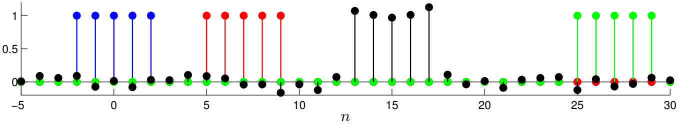
- **Solution:** Compute inner product between y and **shifted target signal** $x[n-m]$ for all $m \in \mathbb{Z}$

$$|d[m]| \;=\; \big|\langle y[n], x[n-m]\rangle\big|$$

- In statistics, $d[m]$ is called the **cross-correlation**; it provides both
  - The detection statistic to compare against the threshold t for each value of shift m
  - An estimate for ℓ (the m the maximizes $d[n]$)

- Example: Square pulse with shift $\ell = 15$ and $m = 0, 7, 27$

$$y[n] = x[n-15] + e[n]$$



Matched Filter

- Useful interpretation of the cross correlation: Let $\widetilde{x}[n] = x^*[-n]$; then

$$d[m] \;=\; \langle y[n], x[n-m]\rangle \;=\; \sum_n y[n]\, x^*[n-m] \;=\; \sum_n y[n]\, \widetilde{x}[m-n]$$

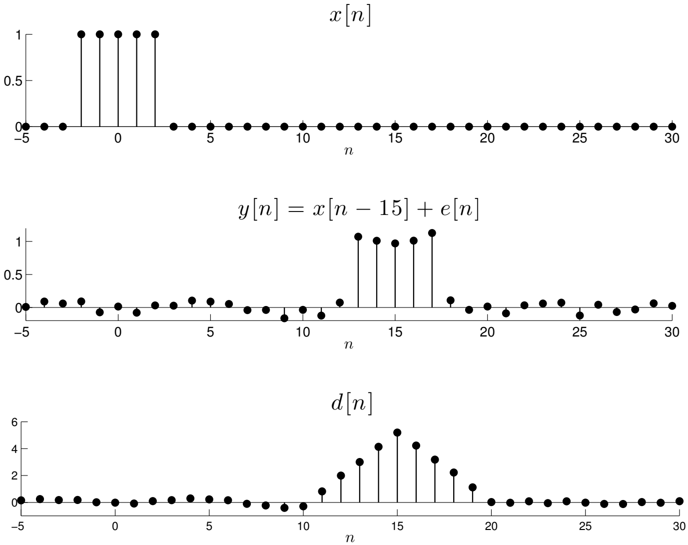
- In words, the cross-correlation $d[m]$ equals the **convolution** of $y[n]$ with the time-reversed and conjugated target signal $\widetilde{x}[n] = x^*[-n]$

- $\widetilde{x}[n] = x^*[-n]$ is the **impulse response** of the **matched filter**

$$y \longrightarrow \boxed{x^*[-n]} \longrightarrow d$$

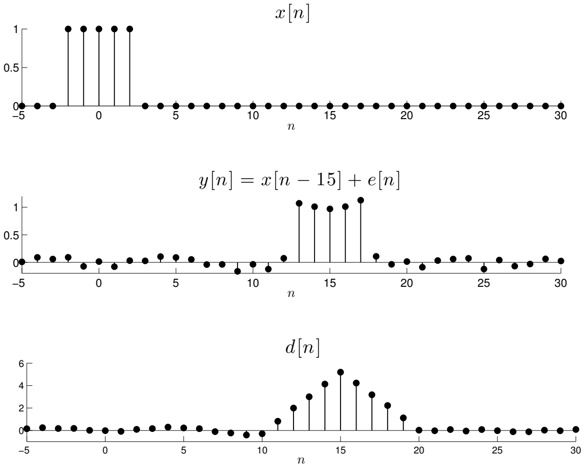
Example: Matched Filter

- Square pulse shifted by $\ell = 15$: $y[n] \; = \; x[n-15] + e[n]$



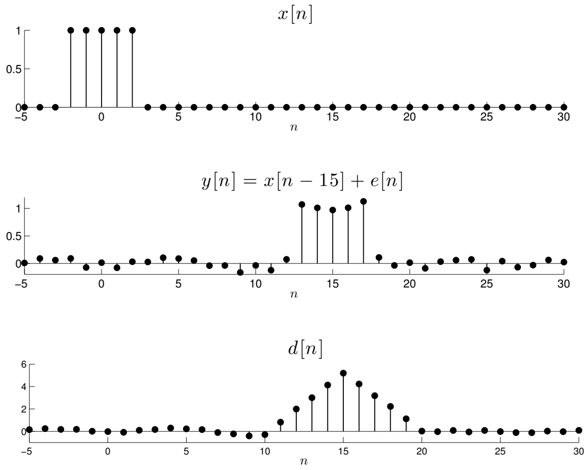
Application: Radar Imaging (1)

- In a radar system, the time delay ℓ is linearly proportional to $2\times$ the distance between the antenna and the target



Application: Radar Imaging (2)

- In a radar system, the time delay ℓ is linearly proportional to $2\times$ the distance between the antenna and the target



Summary

- Inner product and Cauchy Schwarz Inequality provide a natural way to detect a target signal x embedded in another signal y
  - Compare magnitude of inner product to a threshold

- When the target signal is time-shifted by an unknown time-shift ℓ, compute the **cross-correlation**: inner products at all possible time shifts

- Cross-correlation can be interpreted as the convolution of the signal y with a time-reversed and conjugated version of x: the **matched filter**

- Matched filter is ubiquitous in signal processing: radar, sonar, communications, pattern recognition ("Where's Waldo?"), ...

Acknowledgements