COM303: Digital Signal Processing

Lecture 13: Optimal FIR Filter design

Overview

- linear phase FIR
- the Parks-McClellan algorithm

Overview

- linear phase FIR
- the Parks-McClellan algorithm

FIR design

FIR: optimal minimax design

FIR filters are a digital signal processing “exclusivity”.
In the 1970s Parks and McClellan developed an algorithm to design optimal FIR filters:

- linear phase
- equiripple error in passband and stopband

algorithm proceeds by **mini**mizing the **max**imum error in passband and stopband

FIR: optimal minimax design

FIR filters are a digital signal processing "exclusivity".
In the 1970s Parks and McClellan developed an algorithm to design optimal FIR filters:

- linear phase
- equiripple error in passband and stopband

algorithm proceeds by **mini**mizing the **max**imum error in passband and stopband

FIR: optimal minimax design

FIR filters are a digital signal processing “exclusivity”.
In the 1970s Parks and McClellan developed an algorithm to design optimal FIR filters:

- linear phase
- equiripple error in passband and stopband

algorithm proceeds by **mini**mizing the **max**imum error in passband and stopband

FIR: optimal minimax design

FIR filters are a digital signal processing “exclusivity”.

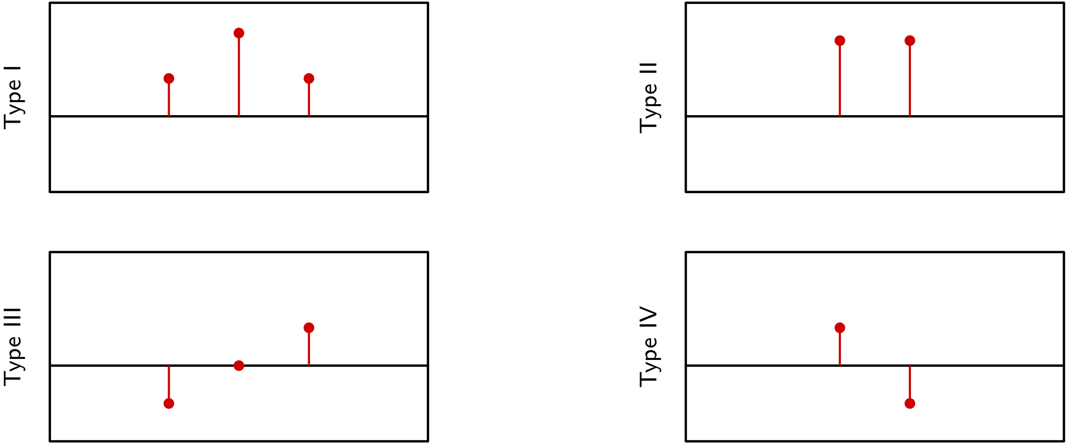
In the 1970s Parks and McClellan developed an algorithm to design optimal FIR filters:

- linear phase
- equiripple error in passband and stopband

algorithm proceeds by **mini**mizing the **max**imum error in passband and stopband

Linear phase in FIRs

Symmetric or antisymmetric impulse responses have linear phase



Linear phase (Type I)

filter length is **odd**: $M = 2L + 1$

$$h[L + n] = h[L - n]$$

zero-centered filter:

$$h_d[n] = h[n + L]$$

$$h_d[n] = h_d[-n]$$

Linear phase (Type I)

filter length is **odd**: $M = 2L + 1$

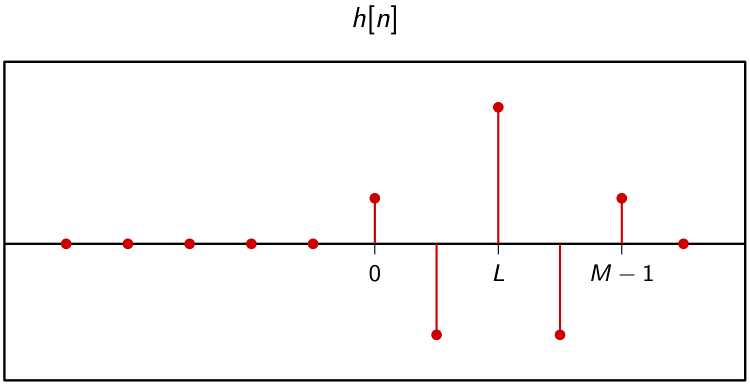
$$h[L + n] = h[L - n]$$

zero-centered filter:

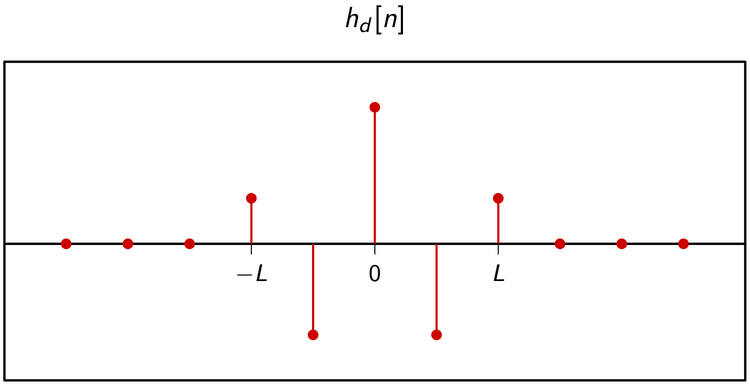
$$h_d[n] = h[n + L]$$

$$h_d[n] = h_d[-n]$$

Linear phase (Type I)



Linear phase (Type I)



Linear phase (Type I)

$$H_d(z) = \sum_{n=-L}^{L} h_d[n] z^{-n}$$

$$= h_d[0] + \sum_{n=1}^{L} h_d[n](z^{n} + z^{-n})$$

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

$$= h_d[0] + 2\sum_{n=1}^{L} h_d[n] \cos \omega n \qquad \in \mathbb{R}$$

Linear phase (Type I)

$$H_d(z) = \sum_{n=-L}^{L} h_d[n] z^{-n}$$

$$= h_d[0] + \sum_{n=1}^{L} h_d[n](z^n + z^{-n})$$

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

$$= h_d[0] + 2\sum_{n=1}^{L} h_d[n] \cos \omega n \quad \in \mathbb{R}$$

Linear phase (Type I)

$$H_d(z) = \sum_{n=-L}^{L} h_d[n] z^{-n}$$

$$= h_d[0] + \sum_{n=1}^{L} h_d[n](z^n + z^{-n})$$

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

$$= h_d[0] + 2\sum_{n=1}^{L} h_d[n] \cos \omega n \qquad \in \mathbb{R}$$

Linear phase (Type I)

$$
\begin{aligned}
H_d(z) &= \sum_{n=-L}^{L} h_d[n] z^{-n} \\
&= h_d[0] + \sum_{n=1}^{L} h_d[n](z^n + z^{-n})
\end{aligned}
$$

$$
\begin{aligned}
H_d(e^{j\omega}) &= h_d[0] + \sum_{n=1}^{L} h_d[n](e^{j\omega n} + e^{-j\omega n}) \\
&= h_d[0] + 2\sum_{n=1}^{L} h_d[n]\cos\omega n \qquad \in \mathbb{R}
\end{aligned}
$$

Linear phase (Type I)

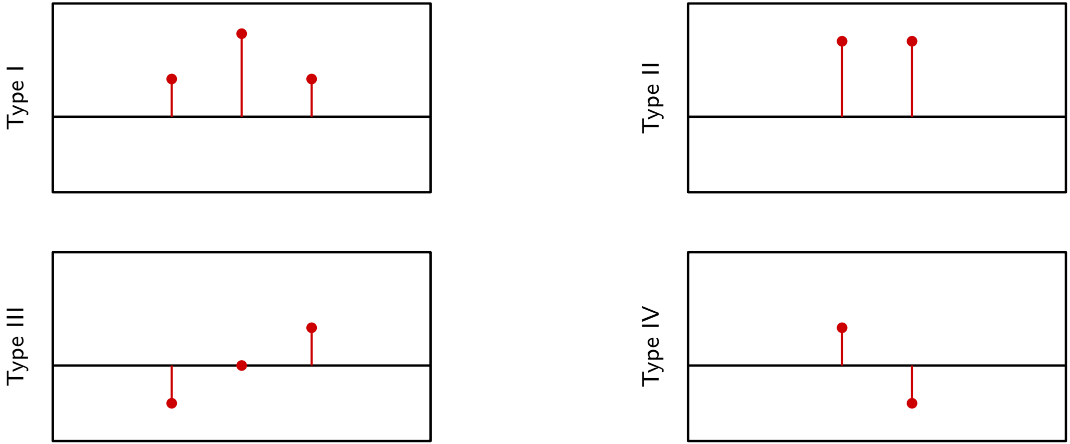
$$H(z) = z^{-L} H_d(z)$$

$$H(e^{j\omega}) = \left[h[L] + 2 \sum_{n=1}^{L} h[n+L] \cos n\omega \right] e^{-j\omega L}$$

Linear Phase FIR Filters

- L: number of points with a "companion"
- even-length FIRs: $M = 2L$ taps
- odd-length FIRs: $M = 2L + 1$ taps
- delay equal to half-length: $C = (M - 1)/2$
- delay is non-integer for even-length filters!

FIR types (again)

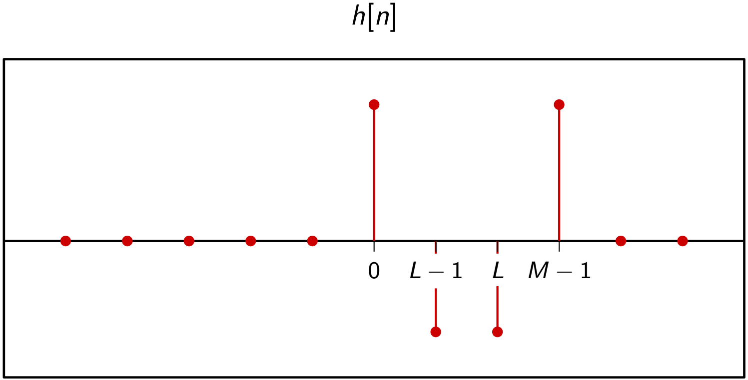


Linear phase (Type II)

filter length is **even**: $M = 2L$

$$h[n] = h[2L - 1 - n]$$

Linear phase (Type II)



Linear phase (Type II)

$$
\begin{aligned}
H(z) = & h[0] && + h[1]z^{-1} && + \dots && + h[L-1]z^{-L+1} + \\
& h[2L-1]z^{-2L+1} && + h[2L-2]z^{-2L+2} && + \dots && + h[L]z^{-L} \\
= & h[0] && + h[1]z^{-1} && + \dots && + h[L-1]z^{-L+1} + \\
& h[0]z^{-2L+1} && + h[1]z^{-2L+2} && + \dots && + h[L-1]z^{-L} \\
= & \sum_{n=0}^{L-1} h[n](z^{-n} + z^{-2L+1+n})
\end{aligned}
$$

Linear phase (Type II)

$$
\begin{aligned}
H(z) = & h[0] && + h[1]z^{-1} && + \dots && + h[L-1]z^{-L+1} + \\
& h[2L-1]z^{-2L+1} && + h[2L-2]z^{-2L+2} && + \dots && + h[L]z^{-L} \\
= & h[0] && + h[1]z^{-1} && + \dots && + h[L-1]z^{-L+1} + \\
& h[0]z^{-2L+1} && + h[1]z^{-2L+2} && + \dots && + h[L-1]z^{-L} \\
= & \sum_{n=0}^{L-1} h[n](z^{-n} + z^{-2L+1+n})
\end{aligned}
$$

Linear phase (Type II)

$$
\begin{aligned}
H(z) = & h[0] && + h[1]z^{-1} && + \dots && + h[L-1]z^{-L+1} + \\
& h[2L-1]z^{-2L+1} && + h[2L-2]z^{-2L+2} && + \dots && + h[L]z^{-L} \\
= & h[0] && + h[1]z^{-1} && + \dots && + h[L-1]z^{-L+1} + \\
& h[0]z^{-2L+1} && + h[1]z^{-2L+2} && + \dots && + h[L-1]z^{-L} \\
= & \sum_{n=0}^{L-1} h[n](z^{-n} + z^{-2L+1+n})
\end{aligned}
$$

Linear phase (Type II)

$$C = (M-1)/2 = (2L-1)/2 = L - 1/2 \quad \text{(non-integer!)}$$

$$H(z) = \sum_{n=0}^{L-1} h[n](z^{-n} + z^{-2C+n})$$

$$= z^{-C} \sum_{n=0}^{L-1} h[n](z^{(C-n)} + z^{-(C-n)})$$

Linear phase (Type II)

$$C = (M-1)/2 = (2L-1)/2 = L - 1/2 \quad \text{(non-integer!)}$$

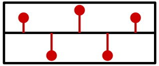
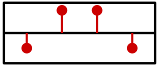
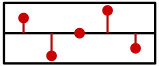
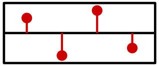
$$\begin{aligned} H(z) &= \sum_{n=0}^{L-1} h[n](z^{-n} + z^{-2C+n}) \\ &= z^{-C} \sum_{n=0}^{L-1} h[n](z^{(C-n)} + z^{-(C-n)}) \end{aligned}$$

Linear phase (Type II)

$$H(e^{j\omega}) = \left[2\sum_{n=0}^{L-1} h[n]\cos(\omega(C-n))\right] e^{-j\omega C}$$

$$C = L - \frac{1}{2}$$

Linear Phase FIR Filters

type	length	sym.	delay	zeros	
I	odd	S	integer		
II	even	S	non-int.		
III	odd	A	integer		
IV	even	A	non-int.		

Zero locations (all types)

- FIRs have only zeros
- $h[n] \in \mathbb{R} \Rightarrow$ if z_0 is a zero, so is z_0^*

Zero locations (Type I)

$$H(z) = z^{-L}\left[h[0] + \sum_n h_d[n](z^n + z^{-n})\right]$$

$$H(z^{-1}) = z^{L}\left[h[0] + \sum_n h_d[n](z^n + z^{-n})\right]$$

$$H(z^{-1}) = z^{2L}H(z)$$

if z_0 is a zero, so is $1/z_0$

Zero locations (Type I)

$$H(z) = z^{-L}\left[h[0] + \sum_n h_d[n](z^n + z^{-n})\right]$$

$$H(z^{-1}) = z^{L}\left[h[0] + \sum_n h_d[n](z^n + z^{-n})\right]$$

$$H(z^{-1}) = z^{2L}H(z)$$

if z_0 is a zero, so is $1/z_0$

Zero locations (Type I)

$$H(z) = z^{-L}\left[h[0] + \sum_n h_d[n](z^n + z^{-n})\right]$$

$$H(z^{-1}) = z^{L}\left[h[0] + \sum_n h_d[n](z^n + z^{-n})\right]$$

$$H(z^{-1}) = z^{2L}H(z)$$

if z_0 is a zero, so is $1/z_0$

Zero locations

if z_0 is a zero, so is $1/z_0$

this is valid for all FIR types (easy to prove)

Zero locations

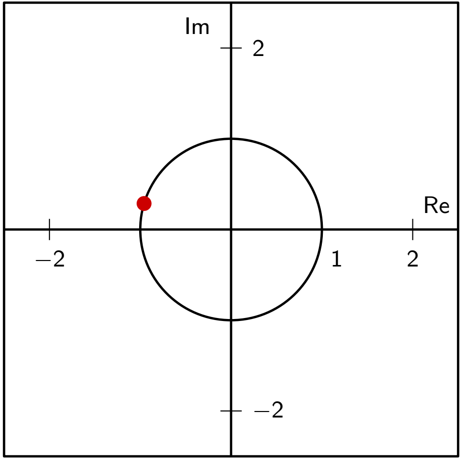
if z_0 is a zero, so is $1/z_0$

this is valid for all FIR types (easy to prove)

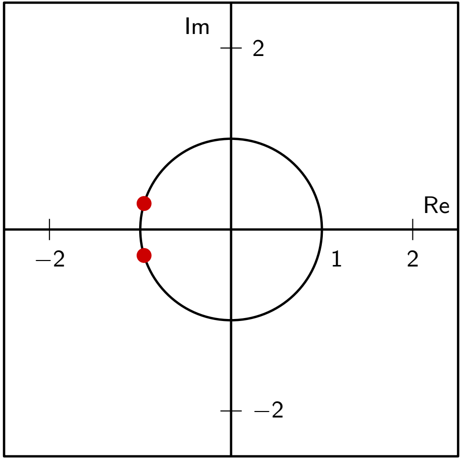
Zero locations (all types)

- if z_0 is a zero, so is z_0^*
- if z_0 is a zero, so is $1/z_0$
- if $z_0 = \rho e^{j\theta}$ is a zero so are:
  - $\rho e^{j\theta}$
  - $(1/\rho) e^{j\theta}$
  - $\rho e^{-j\theta}$
  - $(1/\rho) e^{-j\theta}$

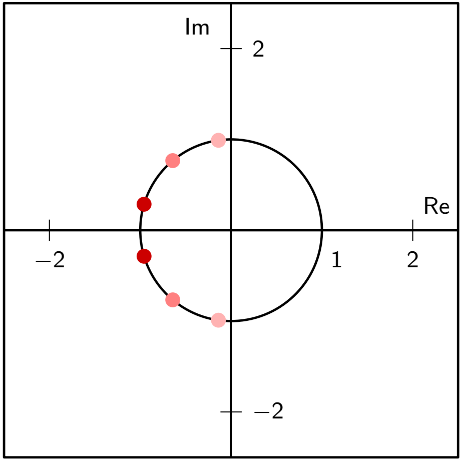
Typical zero plot for linear-phase FIR



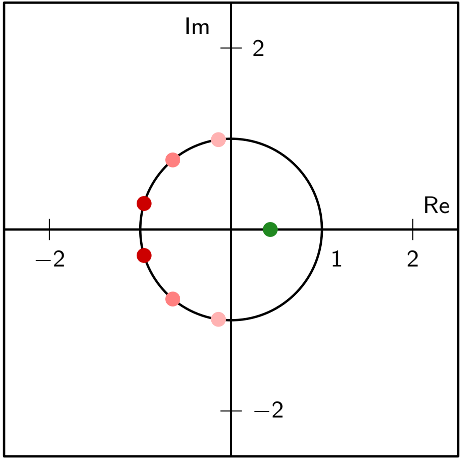
Typical zero plot for linear-phase FIR



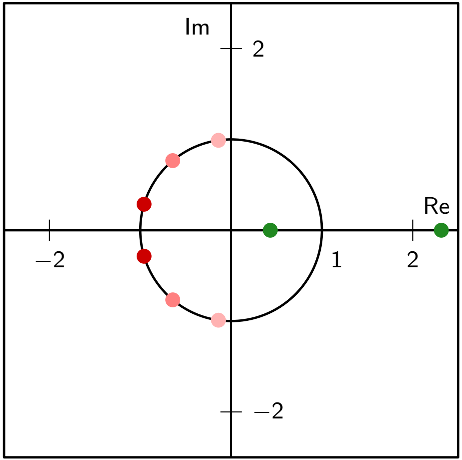
Typical zero plot for linear-phase FIR



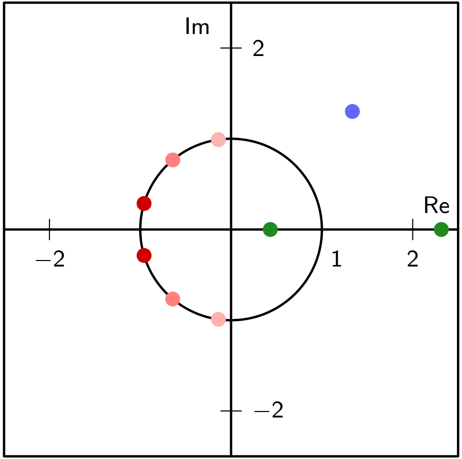
Typical zero plot for linear-phase FIR



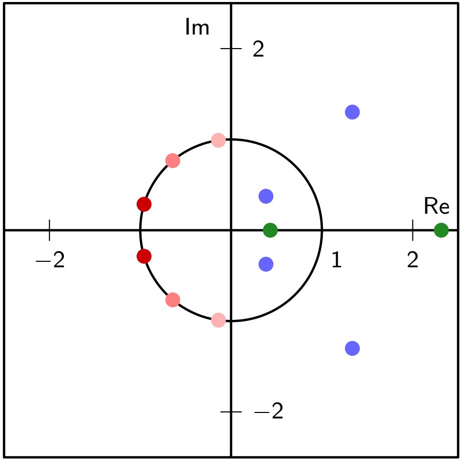
Typical zero plot for linear-phase FIR



Typical zero plot for linear-phase FIR



Typical zero plot for linear-phase FIR



Zero locations (Type II)

$$H(z) = z^{-C} \sum_{n=0}^{L-1} h[n](z^{(C-n)} + z^{-(C-n)})$$

$$H(z^{-1}) = z^{C} \sum_{n=0}^{L-1} h[n](z^{(C-n)} + z^{-(C-n)})$$

$$H(z^{-1}) = z^{2C} H(z)$$

Zero locations (Type II)

$$H(z) = z^{-C} \sum_{n=0}^{L-1} h[n](z^{(C-n)} + z^{-(C-n)})$$

$$H(z^{-1}) = z^{C} \sum_{n=0}^{L-1} h[n](z^{(C-n)} + z^{-(C-n)})$$

$$H(z^{-1}) = z^{2C} H(z)$$

Zero locations (Type II)

$$H(z) = z^{-C} \sum_{n=0}^{L-1} h[n](z^{(C-n)} + z^{-(C-n)})$$

$$H(z^{-1}) = z^{C} \sum_{n=0}^{L-1} h[n](z^{(C-n)} + z^{-(C-n)})$$

$$H(z^{-1}) = z^{2C} H(z)$$

Zero locations (Type II)

$$C = L - 1/2$$

$$H(z^{-1}) = z^{2C} H(z)$$

$$= z^{2L-1} H(z)$$

$$H(-1) = (-1)^{2L-1} H(-1) = -H(-1)$$

$$H(-1) = 0$$

Zero locations (Type II)

$$
\begin{aligned}
C &= L - 1/2 \\
H(z^{-1}) &= z^{2C} H(z) \\
&= z^{2L-1} H(z)
\end{aligned}
$$

$$H(-1) = (-1)^{2L-1} H(-1) = -H(-1)$$

$$H(-1) = 0$$

Zero locations (Type II)

$$
\begin{aligned}
C &= L - 1/2 \\
H(z^{-1}) &= z^{2C} H(z) \\
&= z^{2L-1} H(z)
\end{aligned}
$$

$$H(-1) = (-1)^{2L-1} H(-1) = -H(-1)$$

$$H(-1) = 0$$

Zero locations (Type II)

$$
\begin{aligned}
C &= L - 1/2 \\
H(z^{-1}) &= z^{2C} H(z) \\
&= z^{2L-1} H(z)
\end{aligned}
$$

$$
\begin{aligned}
H(-1) &= (-1)^{2L-1} H(-1) = -H(-1) \\
H(-1) &= 0
\end{aligned}
$$

Zero locations (Type II)

type-II FIRs always have a zero at $\omega = \pi$

Zero locations (Type III)

$$H(z) = z^{-L}\left[\sum_n h_d[n](z^n - z^{-n})\right]$$

$$H(z^{-1}) = -z^{2L}H(z)$$

$$H(1) = -H(1) \quad \Longrightarrow \quad H(1) = 0$$

$$H(-1) = -H(-1) \quad \Longrightarrow \quad H(-1) = 0$$

Zero locations (Type III)

$$H(z) = z^{-L}\left[\sum_n h_d[n](z^n - z^{-n})\right]$$

$$H(z^{-1}) = -z^{2L}H(z)$$

$$H(1) = -H(1) \implies H(1) = 0$$

$$H(-1) = -H(-1) \implies H(-1) = 0$$

Zero locations (Type III)

$$H(z) = z^{-L}\left[\sum_n h_d[n](z^n - z^{-n})\right]$$

$$H(z^{-1}) = -z^{2L}H(z)$$

$$H(1) = -H(1) \quad \Longrightarrow \quad H(1) = 0$$

$$H(-1) = -H(-1) \quad \Longrightarrow \quad H(-1) = 0$$

Zero locations (Type III)

$$H(z) = z^{-L}\left[\sum_n h_d[n](z^n - z^{-n})\right]$$

$$H(z^{-1}) = -z^{2L}H(z)$$

$$H(1) = -H(1) \quad \Longrightarrow \quad H(1) = 0$$

$$H(-1) = -H(-1) \quad \Longrightarrow \quad H(-1) = 0$$

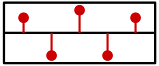
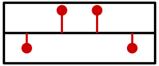
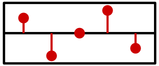
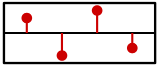
Zero locations (Type III)

type-III FIRs always have a zero at $\omega = 0$ and $\omega = \pi$

Zero locations

Filter Type	Relation	Constraint on Zeros
Type I	$H(z^{-1}) = z^{M-1}H(z)$	No constraints
Type II	$H(z^{-1}) = z^{M-1}H(z)$	Zero at $z = -1$ (i.e. $\omega = \pi$)
Type III	$H(z^{-1}) = -z^{M-1}H(z)$	Zeros at $z = \pm 1$ (i.e. at $\omega = \pi$, $\omega = 0$)
Type IV	$H(z^{-1}) = -z^{M-1}H(z)$	Zero at $z = 1$ (i.e. $\omega = 0$)

Linear Phase FIR Filters

type	length	sym.	delay	zeros	
I	odd	S	integer		
II	even	S	non-int.	$\pm\pi$	
III	odd	A	integer	$0, \pm\pi$	
IV	even	A	non-int.	0	

optimal FIR filter design

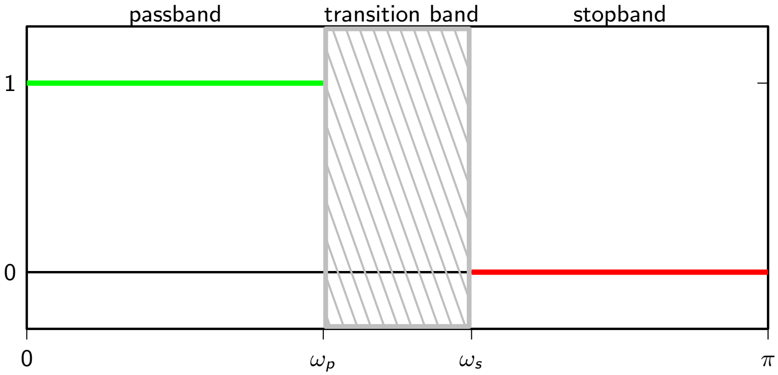
How do we design linear-phase FIRs?

answer: with the Parks-McClellan algorithm

let's work with an example:

- type I
- zero phase (work with $H_d(z)$)
- lowpass characteristc

Remember the realistic specs



Setting up the problem

Intuition #1: z-transform a finite-degree polynomial in z

$$H_d(z) = h_d[0] + \sum_{n=1}^{L} h_d[n](z^n + z^{-n}) = Q_M(z)$$

Intuition #2: Fourier transform also a finite-degree polynomial

$$H_d(e^{j\omega}) = P_L(x) \qquad x = \cos\omega$$

Setting up the problem

Intuition #3: we want

$$P_L(x) \approx D(x)$$

filter design becomes polynomial fitting!

Finding the polynomial

$$H_d(e^{j\omega}) = h_d[0] + 2\sum_{n=1}^{L} h_d[n] \cos \omega n$$

Step 1: Chebyshev polynomials

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2x^2 - 1 \\
&\cdots
\end{aligned}
$$

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$$

Step 1: Chebyshev polynomials

fundamental property:

$$T_n(\cos\omega) = \cos n\omega$$

Step 1: Chebyshev polynomials

$$H_d(e^{j\omega}) = h_d[0] + \sum_{n=1}^{L} 2h_d[n] \cos n\omega$$

$$P(x) = h_d[0] + \sum_{n=1}^{L} 2h_d[n]\, T_n(x)\Big|_{x=\cos\omega}$$

$$H_d(e^{j\omega}) = P(x)\Big|_{x=\cos\omega}$$

Step 1: Chebyshev polynomials

$$H_d(e^{j\omega}) = h_d[0] + \sum_{n=1}^{L} 2h_d[n] \cos n\omega$$

$$P(x) = h_d[0] + \sum_{n=1}^{L} 2h_d[n]\, T_n(x)\Big|_{x=\cos\omega}$$

$$H_d(e^{j\omega}) = P(x)\Big|_{x=\cos\omega}$$

Step 1: Chebyshev polynomials

$$H_d(e^{j\omega}) = h_d[0] + \sum_{n=1}^{L} 2h_d[n] \cos n\omega$$

$$P(x) = h_d[0] + \sum_{n=1}^{L} 2h_d[n]\, T_n(x)\Big|_{x=\cos\omega}$$

$$H_d(e^{j\omega}) = P(x)\Big|_{x=\cos\omega}$$

Step 1: Chebyshev polynomials

$$H_d(e^{j\omega}) = a + 2b\cos\omega + 2c\cos 2\omega + 2d\cos 3\omega$$

$$\begin{aligned} H_d(e^{j\omega}) &= a + 2b\cos\omega + 2c(2\cos^2\omega - 1) + 2d(4\cos^3\omega - 3\cos\omega) \\ &= (a - 2c) + (2b - 6d)\cos\omega + 4c\cos^2\omega + 8d\cos^3\omega \end{aligned}$$

$$= [(a - 2c) + (2b - 6d)x + 4c\,x^2 + 8d\,x^3]_{x=\cos\omega}$$

Step 1: Chebyshev polynomials

$$H_d(e^{j\omega}) = a + 2b\cos\omega + 2c\cos 2\omega + 2d\cos 3\omega$$

$$\begin{aligned} H_d(e^{j\omega}) &= a + 2b\cos\omega + 2c(2\cos^2\omega - 1) + 2d(4\cos^3\omega - 3\cos\omega) \\ &= (a - 2c) + (2b - 6d)\cos\omega + 4c\cos^2\omega + 8d\cos^3\omega \\ &= \left[(a - 2c) + (2b - 6d)x + 4c\,x^2 + 8d\,x^3\right]_{x=\cos\omega} \end{aligned}$$

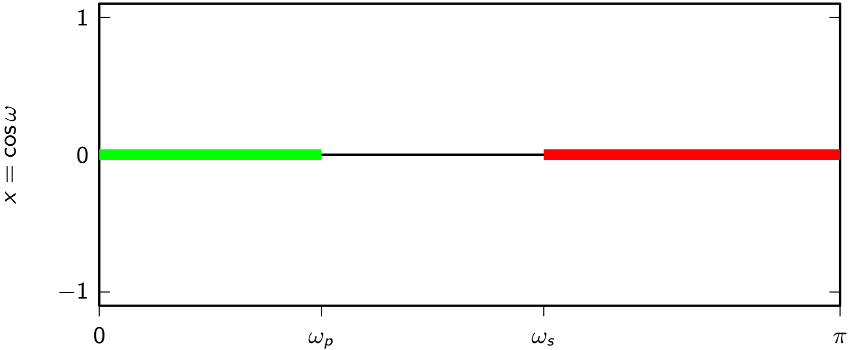
Step 1: Chebyshev polynomials

$$H_d(e^{j\omega}) = a + 2b\cos\omega + 2c\cos 2\omega + 2d\cos 3\omega$$

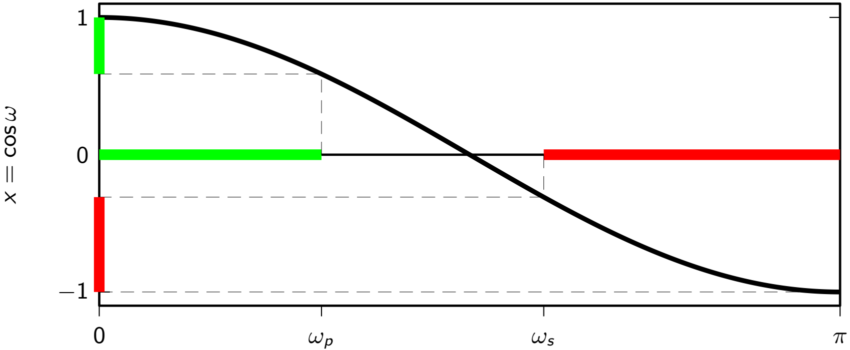
$$\begin{aligned}
H_d(e^{j\omega}) &= a + 2b\cos\omega + 2c(2\cos^2\omega - 1) + 2d(4\cos^3\omega - 3\cos\omega) \\
&= (a - 2c) + (2b - 6d)\cos\omega + 4c\cos^2\omega + 8d\cos^3\omega
\end{aligned}$$

$$= [(a - 2c) + (2b - 6d)x + 4c\,x^2 + 8d\,x^3]_{x=\cos\omega}$$

Step 2: Convert the specs



Step 2: Convert the specs



Step 2: Convert the specs

If $x = \cos\omega$

$$I_p = [0,\ \omega_p] \rightarrow I'_p = [\cos\omega_p,\ 1]$$

$$I_s = [\omega_p,\ \pi] \rightarrow I'_s = [-1,\ \cos\omega_s]$$

Step 2: Convert the specs

If $x = \cos \omega$

$$I_p = [0,\, \omega_p] \rightarrow I'_p = [\cos \omega_p,\, 1]$$

$$I_s = [\omega_p,\, \pi] \rightarrow I'_s = [-1,\, \cos \omega_s]$$

Step 2: Convert the specs

We want

$$P(x) \approx 1 \quad \text{for } x \in I'_p$$

$$P(x) \approx 0 \quad \text{for } x \in I'_s$$

Global error function

$$E(x) = P(x) - D(x)$$

with

$$D(x) = \begin{cases} 1 & \text{for } x \in I'_p \\ 0 & \text{for } x \in I'_s \end{cases}$$

Step 2: Convert the specs

We want

$$P(x) \approx 1 \quad \text{for } x \in I'_p$$
$$P(x) \approx 0 \quad \text{for } x \in I'_s$$

Global error function

$$E(x) = P(x) - D(x)$$

with

$$D(x) = \begin{cases} 1 & \text{for } x \in I'_p \\ 0 & \text{for } x \in I'_s \end{cases}$$

We could try this...

standard fitting of a degree-L polynomial:

- pick $L+1$ points over the two intervals
- build the Vandermode matrix

$$\mathbf{A} = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^L \\ 1 & x_1 & x_1^2 & \dots & x_1^L \\ \vdots & & & & \\ 1 & x_L & x_L^2 & \dots & x_L^L \end{bmatrix}$$

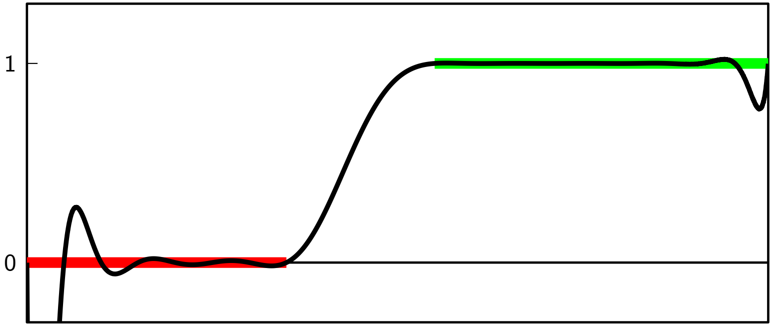
- solve the interpolation problem

$$\mathbf{Ap} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

… but it wouldn't work

- (direct methods numerically unstable)
- interpolation minimizes the MSE but not the maximum error

max error vs MSE



Brilliant idea: minimize max error

$$E = \min_{P(x)} \max_{x \in I_p' \cup I_s'} \{|P(x) - D(x)|\}$$

Alternation Theorem

$P(x)$ is the minimax approximation to $D(x)$ if and only if $P(x) - D(x)$ alternates $L + 2$ times between $+E$ and $-E$ in $I_p' \cup I_s'$

Why Alternation Theorem is key

- check candidates: if $P(x)$ satisfies the AT, we're done
- leads to a numerical algorithm to find $P(x)$: the Remez Exchange

The Remez Algorithm

suppose we *knew* the positions of the alternations; then we could solve

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^L & \epsilon \\ 1 & x_1 & x_1^2 & \dots & x_1^L & -\epsilon \\ & & & \vdots & & \\ 1 & x_L & x_L^2 & \dots & x_L^L & (-1)^L\epsilon \end{bmatrix} \mathbf{p} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

and find both the polynomial coefficients and E

The Remez Algorithm

obviously we don't know the positions of the alternations; but we can start with a guess

- solve the system of equation for the guessed x_i
- check if the solution satisfies the alternation theorem; if so, we're done
- otherwise, find the extrema of the error and use the locations as new guess; repeat

Example

- $M = 9$ $(L = 4)$
- $\omega_p = 0.4\pi$
- $\omega_s = 0.6\pi$

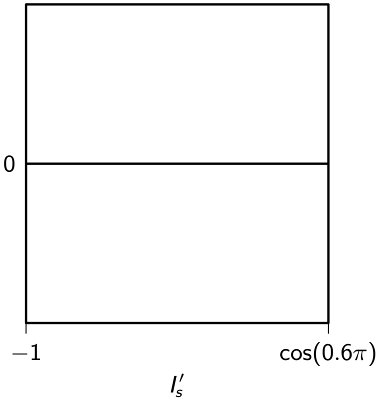
- we need at least $L + 2 = 6$ alternations
- 2 alternations always at band edges (otherwise specs not fulfilled)
- guess the other 4 and apply remez

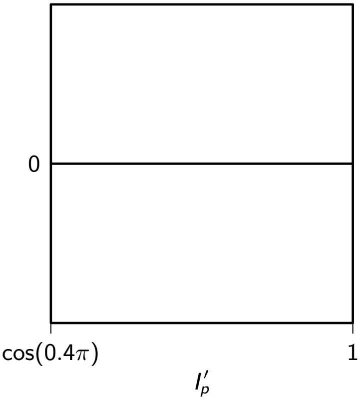
Example

- $M = 9$ ($L = 4$)
- $\omega_p = 0.4\pi$
- $\omega_s = 0.6\pi$

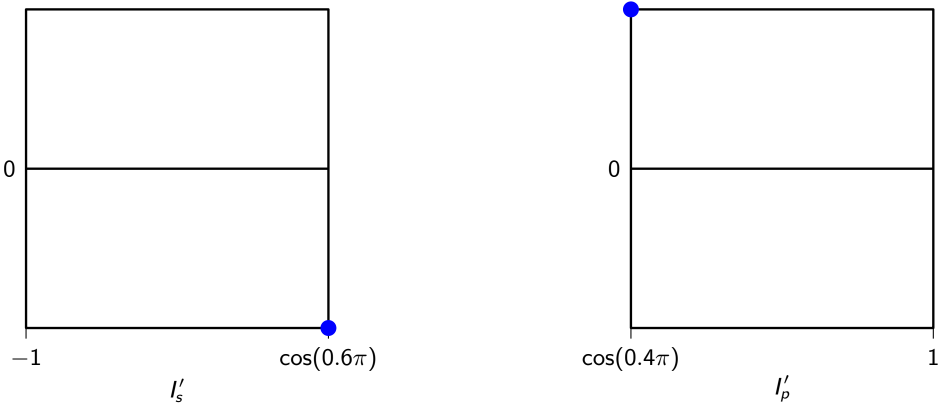
- we need at least $L + 2 = 6$ alternations
- 2 alternations always at band edges (otherwise specs not fulfilled)
- guess the other 4 and apply remez

Remez exchange algorithm

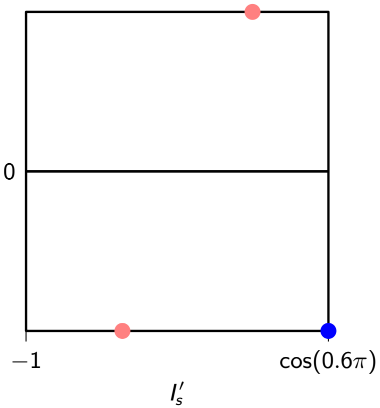




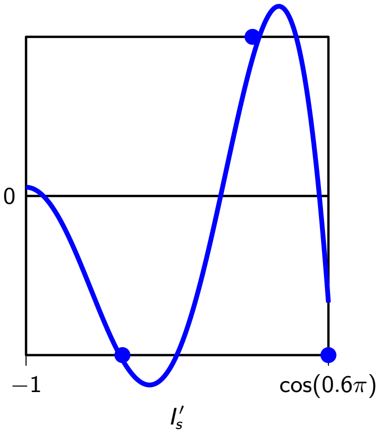
Remez exchange algorithm

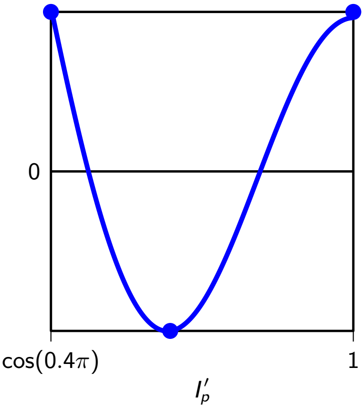


Remez exchange algorithm

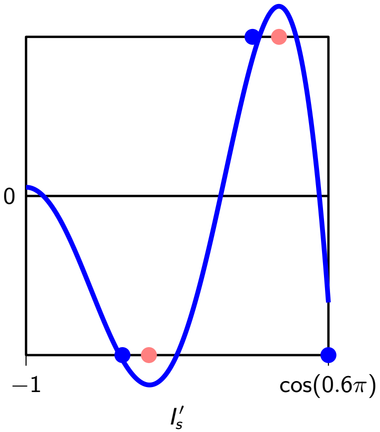


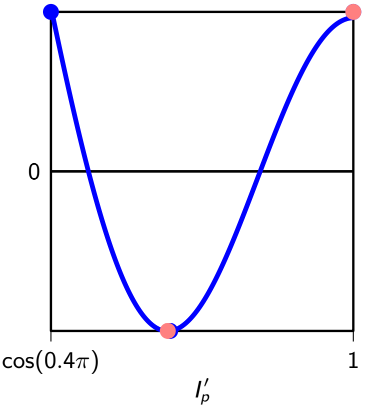
Remez exchange algorithm



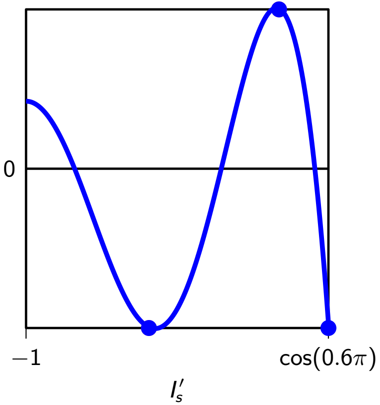


Remez exchange algorithm

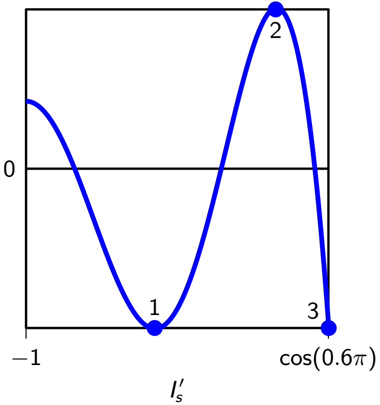




Remez exchange algorithm

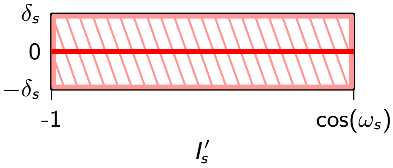


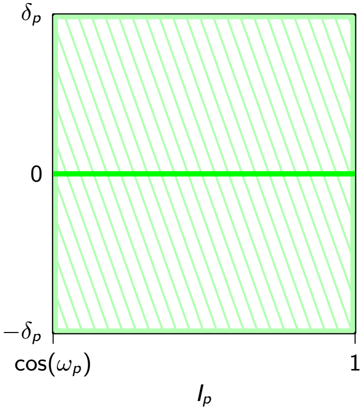
Passband and Stopband Error



Tuning the error

generally, we want to pay more attention to the error in stopband or passband





Goal: fit $E(x)$ within the boxes.

Tuning the error

The Alternation Theorem works also with a weighting function:

$$W(x) = \begin{cases} 1 & \text{for } x \in I'_p \\ \delta_p/\delta_s & \text{for } x \in I'_s \end{cases}$$

The updated minimization problem:

$$\min \max_{x \in I'_p \cup I'_s} \{|W(x)[P(x) - D(x)]|\}$$

Tuning the error

The Alternation Theorem works also with a weighting function:

$$W(x) = \begin{cases} 1 & \text{for } x \in I'_p \\ \delta_p/\delta_s & \text{for } x \in I'_s \end{cases}$$

The updated minimization problem:

$$\min \max_{x \in I'_p \cup I'_s} \{|W(x)[P(x) - D(x)]|\}$$

Parks-McClellan Algorithm; the full recipe for lowpass

User data:

Parks-McClellan Algorithm; the full recipe for lowpass

User data:

- filter length $M = 2L + 1$

Parks-McClellan Algorithm; the full recipe for lowpass

User data:

- filter length $M = 2L + 1$
- ω_p and ω_s

Parks-McClellan Algorithm; the full recipe for lowpass

User data:

- filter length $M = 2L + 1$
- ω_p and ω_s
- stopband -to-passband tolerance ratio δ_s / δ_p

Parks-McClellan Algorithm; the full recipe for lowpass

User data:

- filter length $M = 2L + 1$
- ω_p and ω_s
- stopband -to-passband tolerance ratio δ_s / δ_p

Run Parks-McClellan algorithm; obtain:

- M filter coefficients

Parks-McClellan Algorithm; the full recipe for lowpass

User data:

- filter length $M = 2L + 1$
- ω_p and ω_s
- stopband -to-passband tolerance ratio δ_s/δ_p

Run Parks-McClellan algorithm; obtain:

- M filter coefficients
- stopband and passband tolerances δ_s and δ_p

Parks-McClellan Algorithm; the full recipe for lowpass

User data:

- filter length $M = 2L + 1$
- ω_p and ω_s
- stopband -to-passband tolerance ratio δ_s/δ_p

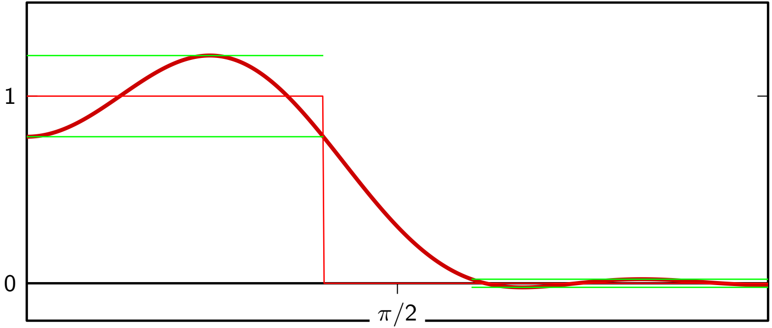
Run Parks-McClellan algorithm; obtain:

- M filter coefficients
- stopband and passband tolerances δ_s and δ_p
- If error too big, increase M and retry.

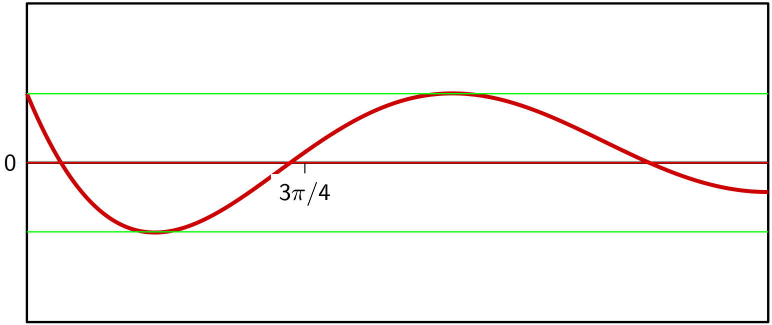
Example revisited

- $M = 9$ $(L = 4)$
- $\omega_p = 0.4\pi$
- $\omega_s = 0.6\pi$
- $\delta_s/\delta_p = 1/10$

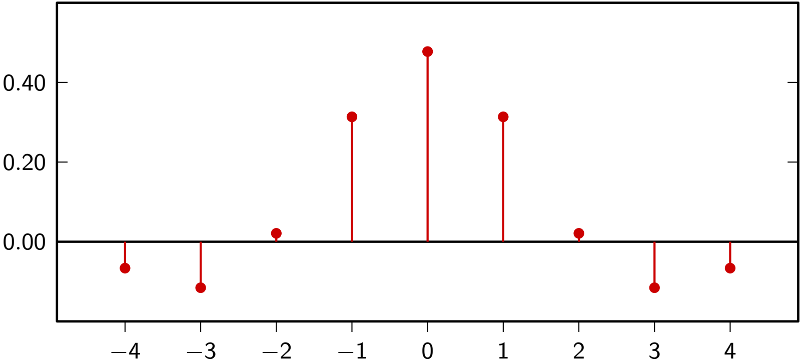
Final Result



Final Result (stopband)



Final Result (Impulse Response)



Alternations

- Max number of alternations is $L+3$:
  - polynomial degree L has $L-1$ local extrema
  - ω_p and ω_s are always alternations
  - sometimes $\omega = 0$
  - sometimes $\omega = \pi$
- look at the *in band* alternations, that gives you $L-1$

Minimax lowpass filter (recap)

Magnitude response:

- equiripple in passband and stopband

Design parameters:

- order N (number of taps)
- passband edge ω_p
- stopband edge ω_s
- ratio of passband to stopband error δ_p/δ_s

Design test criterion:

- passband max error
- stopband max error

Minimax lowpass filter (recap)

Magnitude response:

- equiripple in passband and stopband

Design parameters:

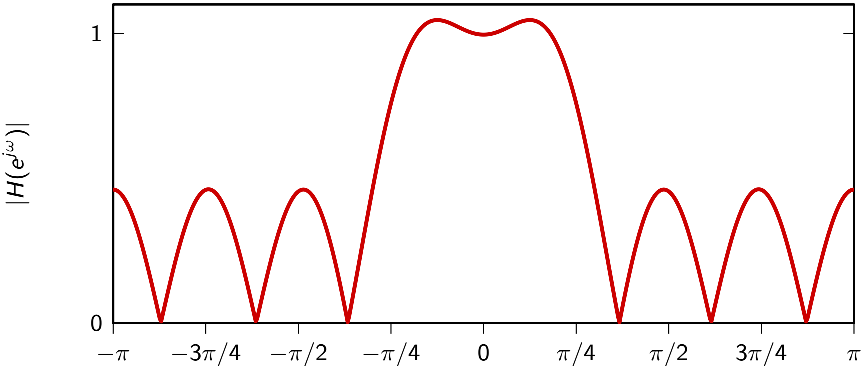
- order N (number of taps)
- passband edge ω_p
- stopband edge ω_s
- ratio of passband to stopband error δ_p/δ_s

Design test criterion:

- passband max error
- stopband max error

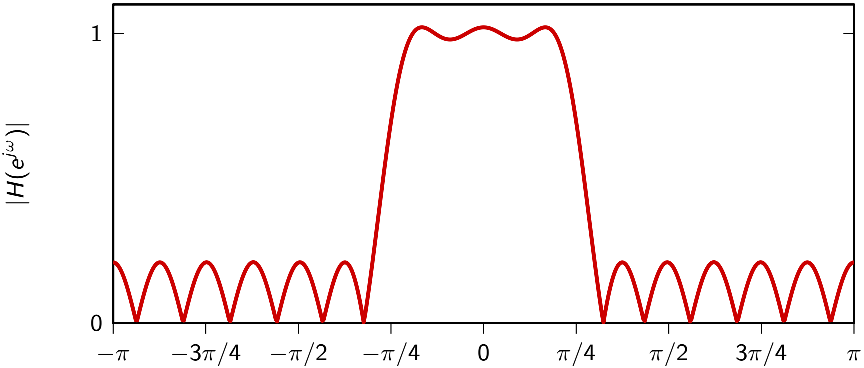
Minimax lowpass example

$$N = 9, \omega_p = 0.2\pi, \omega_s = 0.3\pi, \delta_p/\delta_s = 10$$



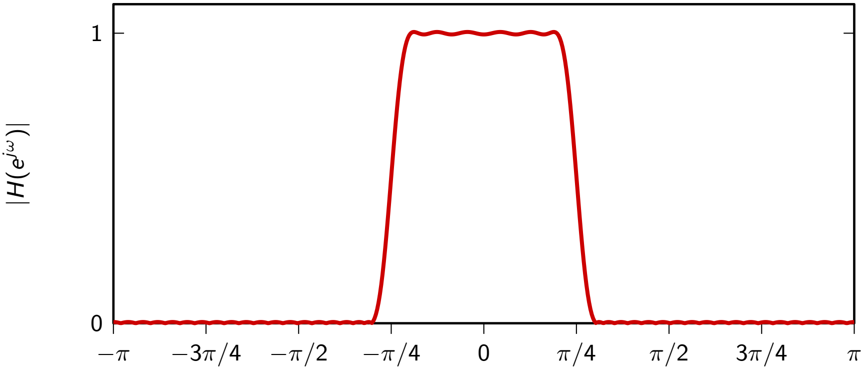
Minimax lowpass example

$$N = 19, \omega_p = 0.2\pi, \omega_s = 0.3\pi, \delta_p/\delta_s = 10$$



Minimax lowpass example

$$N = 51, \omega_p = 0.2\pi, \omega_s = 0.3\pi, \delta_p/\delta_s = 1$$



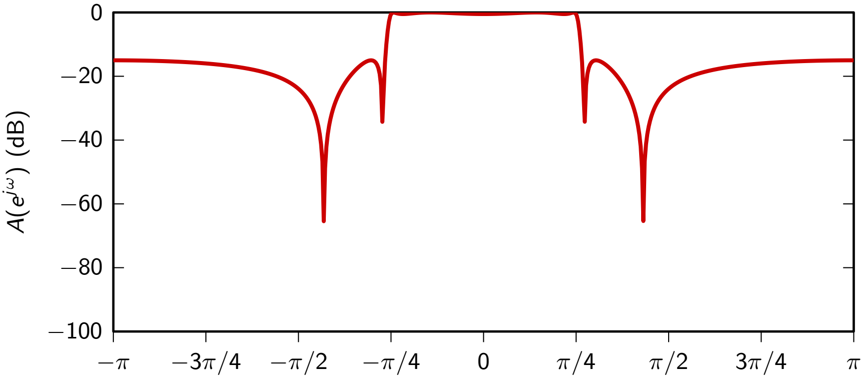
Magnitude response in decibels

- filter max passband magnitude G
- filter attenuation expressed in decibels as:

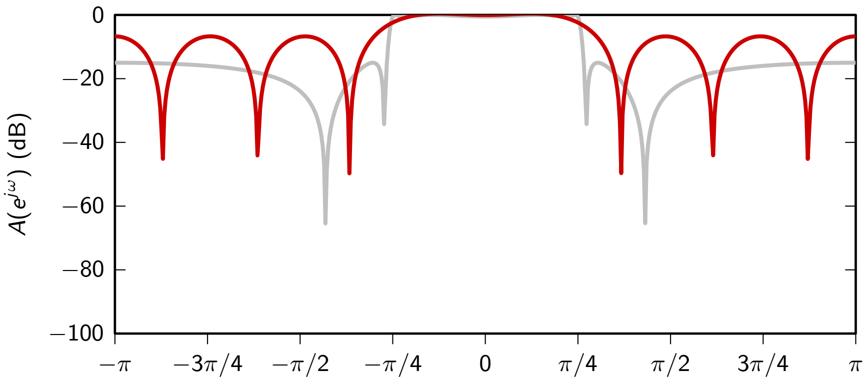
$$A_{\mathrm{dB}} = 20\log_{10}(|H(e^{j\omega})|/G)$$

- useful to compare attenuations between filters

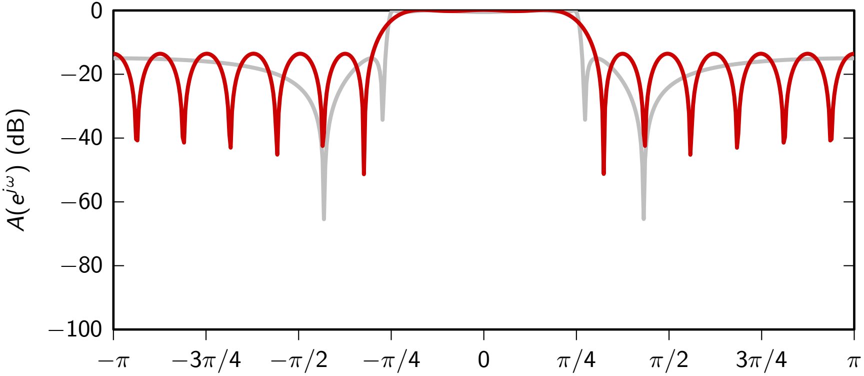
4th-order elliptic lowpass, $\omega_c = \pi/4$, log scale



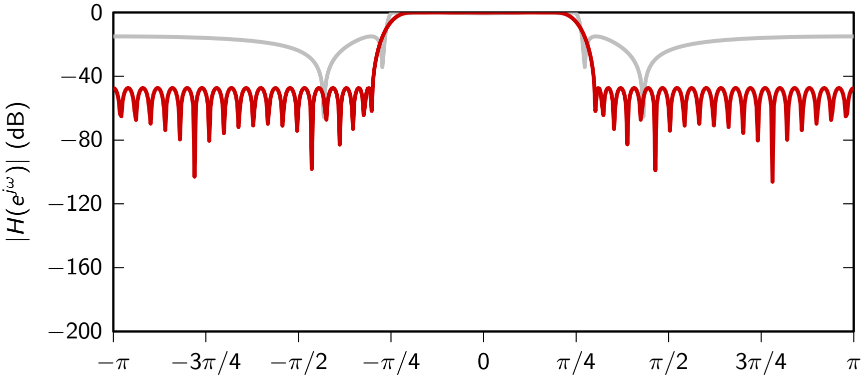
9-tap minimax lowpass, $\omega_c = \pi/4$, log scale



19-tap minimax lowpass, $\omega_c = \pi/4$, log scale



51-tap minimax lowpass, $\omega_c = \pi/4$, log scale



Life beyond lowpass

The IIR and FIR methods we just described can be used to design more general filter types than lowpass, with only minor modifications

- IIR bandpass and highpass can be obtain by modulating the lowpass response
- optimal FIR bandpass and highpass can be designed by the Parks-McClellan algorithm
- optimal FIR can also be designed with piecewise linear magnitude response
- the literature on filter design is vast: this is just the tip of the iceberg!

Life beyond lowpass

The IIR and FIR methods we just described can be used to design more general filter types than lowpass, with only minor modifications

- IIR bandpass and highpass can be obtain by modulating the lowpass response
- optimal FIR bandpass and highpass can be designed by the Parks-McClellan algorithm
- optimal FIR can also be designed with piecewise linear magnitude response
- the literature on filter design is vast: this is just the tip of the iceberg!

Life beyond lowpass

The IIR and FIR methods we just described can be used to design more general filter types than lowpass, with only minor modifications

- IIR bandpass and highpass can be obtain by modulating the lowpass response
- optimal FIR bandpass and highpass can be designed by the Parks-McClellan algorithm
- optimal FIR can also be designed with piecewise linear magnitude response
- the literature on filter design is vast: this is just the tip of the iceberg!

Life beyond lowpass

The IIR and FIR methods we just described can be used to design more general filter types than lowpass, with only minor modifications

- IIR bandpass and highpass can be obtain by modulating the lowpass response
- optimal FIR bandpass and highpass can be designed by the Parks-McClellan algorithm
- optimal FIR can also be designed with piecewise linear magnitude response
- the literature on filter design is vast: this is just the tip of the iceberg!