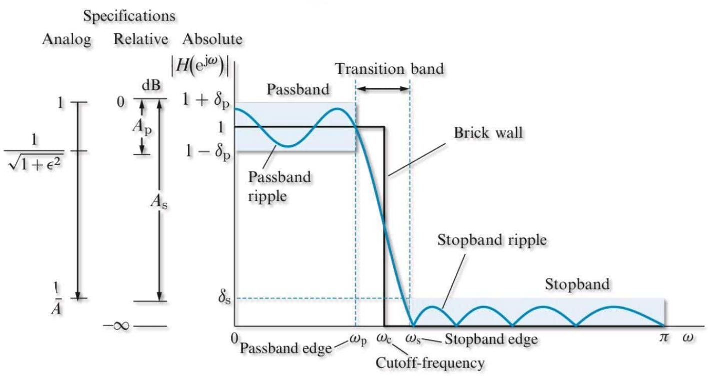
Digital Filter Design Fundamentals



0

Filter Design Problem

We will consider designs of frequency-selective filters. Steps are:

- **Specification**: Specify the desired frequency response function characteristics to address the needs of a specific application. The requirements are often stated in the frequency domain.
- **Approximation**: Approximate the desired specifications (frequency response) by a polynomial or rational frequency response function. The goal is to meet the specifications with minimum computational complexity.
- **Quantization**: Quantize filter coefficients to the required number of bits for a given processor.
- **Verification**: Verify the filter's performance by simulation or testing with real data.
- **Implementation**: Develop structure of the verified filter in hardware, software, or both.

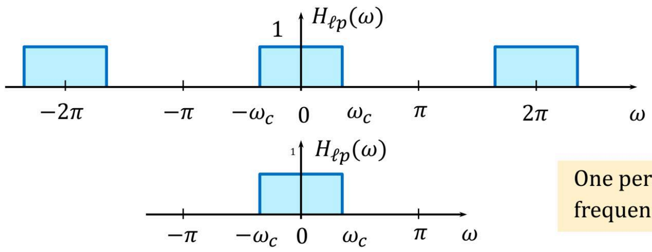
EECE-5666: DSP   Digital Filter Design Fundamentals   1/31

1

Filter Design Problem

- The first step involves obtaining ***specifications*** of the frequency response $H(\omega)$ of the desired frequency-selective filter.
- These specifications mean that we have to provide both the magnitude and phase responses of the desired filter.
- Furthermore, the filter impulse response $h[n]$ must also be ***real*** and ***causal***.
- However, due to Paley-Wiener Theorem, it turns out that causality restricts the simultaneous specification of magnitude and phase responses of a causal filter.
- We will now study this issue of simultaneous specifications before we provide more filter specification details.
- We will then study various sets of specifications used in filter design.

The Ideal Low-Pass Filter



One period of the periodic frequency response

$$H_{\ell p}(\omega) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & \omega_c < |\omega| < \pi \end{cases} \Rightarrow$$

$$h_{\ell p}(n) = \begin{cases} \dfrac{\omega_c}{\pi}, & n = 0 \\ \dfrac{\sin \omega_c n}{\pi n}, & n \neq 0 \end{cases} \Rightarrow \text{IIR}$$

- $h_{\ell p}(n) \neq 0, n < 0 \Rightarrow$ Non−causal
- $\sum_{n=-\infty}^{\infty} |h_{\ell p}(n)| = \infty \Rightarrow$ Unstable
- Non−rational system function $\Rightarrow$ Unrealizable

How can we design "good" practical filters?

The Paley-Wiener Theorem and its Implications

Paley-Wiener Theorem

$$\text{If } \sum_{all\ n} |h(n)|^2 < \infty \text{ and } h(n) = 0, n < 0 \Rightarrow \int_{-\pi}^{\pi} \ln|H(\omega)| d\omega < \infty$$

$$\text{If } \int_{-\pi}^{\pi} |H(\omega)|^2 d\omega < \infty \text{ and } \int_{-\pi}^{\pi} \ln|H(\omega)| d\omega < \infty \Rightarrow \text{We can find } \angle H(\omega) \text{ so that}$$

$$h(n) = \mathcal{F}^{-1}\{|H(\omega)|e^{j\angle H(\omega)}\} = 0 \text{ for } n < 0 \text{ (Causal)}$$

Implications: $|H(\omega)|$ cannot be zero in any finite interval$\Rightarrow$ all ideal filters are non-causal!

Relationship Between $H_R(\omega)$ and $H_I(\omega)$

Decomposition of a real-valued impulse response

$$h(n) = h_e(n) + h_o(n), \qquad h_e(n) = \frac{1}{2}[h(n) + h(-n)], \qquad h_o(n) = \frac{1}{2}[h(n) - h(-n)]$$

If $h(n)$ is causal

$$h(n) = 2h_e(n)u(n) - h_e(0)\delta(n) \Rightarrow h_e(n) \rightarrow h(n)$$

$$h(n) = 2h_o(n)u(n) - h(0)\delta(n) \Rightarrow h_o(n) \rightarrow h(n), \text{ except } h(0)$$

$\Rightarrow h_e(n)$ and $h_o(n)$ are related!

If $\sum_{n=-\infty}^{\infty}|h(n)| < \infty \Rightarrow H(\omega) = H_R(\omega) + jH_I(\omega)$

$$\text{If } h(n) \text{ is real and causal} \Rightarrow \left.\begin{array}{l} \mathcal{F}\{h_e(n)\} = X_R(\omega) \\ \mathcal{F}\{h_o(n)\} = X_I(\omega) \end{array}\right\}$$

$H(\omega)$ is specified by $H_R(\omega)$ or $\{H_I(\omega), h(0)\}$

$H_R(\omega)$ and $H_I(\omega)$ are related!

Implications of Causality

- **Relationship between real-part $h_e(n)$ and odd-part $h_o(n)$ of $h(n)$:**

Every sequence can be written as $h(n) = h_e(n) + h_o(n)$ where

$$h_e(n) = \frac{h(n) + h(-n)}{2} \text{ and } h_o(n) = \frac{h(n) - h(-n)}{2}$$

If $h(n)$ is causal then we have

$$h_e(n) = \begin{cases} \frac{1}{2}h(n), & n \neq 0 \\ h(0)\ , & n = 0 \end{cases} \quad \text{and} \quad h_o(n) = \begin{cases} \frac{1}{2}h(n), & n > 0 \\ -\frac{1}{2}h(n), & n < 0 \\ 0, & n = 0 \end{cases}$$

Hence

$$h(n) = 2h_e(n)u(n) - h_e(n)\delta(n) \qquad \Rightarrow \quad h_e(n) \to h(n)$$
$$h(n) = 2h_o(n)u(n) + h(0)\delta(n) \qquad \Rightarrow \quad h_0(n) \to h(n) \text{ except } h[0]$$

$h_o(n)$ and $h_e(n)$ are related!

6

Implications of Causality

- If $h(n)$ is **causal**

$$h(n) = 2h_e(n)u(n) - h_e(n)\delta(n) \qquad \Rightarrow \quad h_e(n) \to h(n)$$
$$h(n) = 2h_o(n)u(n) + h[0]\delta(n) \qquad \Rightarrow \quad h_0(n) \to h(n) \text{ except } h(0)$$

$h_o(n)$ and $h_e(n)$ are related!

Example: Let $h_e(n) = \{3, 2, 1, \underset{\uparrow}{5}, 1, 2, 3\}$. Then $h_o(n)$ must be equal to

$$h_o(n) = \{-3, -2, -1, \underset{\uparrow}{0}, 1, 2, 3\} \text{ and hence } h(n) = \{0, 0, 0, \underset{\uparrow}{5}, 2, 4, 6\}$$

Exercise: Show that the relationship between $h_o(n)$ and $h_e(n)$ is given by

$$h_o(n) = h_e(n)u[n-1] - h_e(n)u[-n-1] = \text{sgn}(n)h_e(n)$$

where

$$\text{sgn}(n) = \begin{cases} 1, & n > 0 \\ 0, & n = 0 \\ -1, & n < 0 \end{cases}$$

7

Implications of Causality

- Relationship between $H_{\mathrm{R}}(\omega)$ and $H_{\mathrm{I}}(\omega)$:
  - We showed that $h_{\mathrm{e}}(n)$ and $h_{\mathrm{o}}(n)$ are related if $h(n)$ is causal.
  - Furthermore, if $h(n)$ is stable then we also have relationship between the real and imaginary parts of $H(\omega)$. Note

$$\sum_n \left|h(n)\right| < \infty \Rightarrow H(\omega) = H_{\mathrm{R}}(\omega) + \mathrm{j}H_{\mathrm{I}}(\omega) \text{ exists}$$

Since $h(n)$ is **real** and **causal**, we have

$$h(n) \Rightarrow \begin{cases} h_{\mathrm{e}}(n) \overset{\mathcal{F}}{\longleftrightarrow} H_{\mathrm{R}}(\omega) \\ h_{\mathrm{o}}(n) \overset{\mathcal{F}}{\longleftrightarrow} H_{\mathrm{I}}(\omega) \end{cases} \Rightarrow \begin{array}{l} H(\omega) \text{ is specified by} \\ H_{\mathrm{R}}(\omega) \text{ or } \left\{H_{\mathrm{I}}(\omega), h(n)\right\} \end{array}$$

Exercise: If $h(n)$ is causal, stable, and real then show that

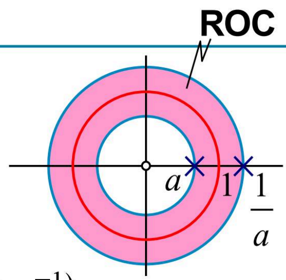
$$H_{\mathrm{R}}(\omega) \xleftrightarrow{\text{Hilbert Transform}} H_{\mathrm{I}}(\omega)$$

where $\mathcal{H}\left[x(t)\right] = \frac{1}{\pi}\int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau}\,\mathrm{d}\tau$ provided the integral converges.

EECE-5666: DSP Digital Filter Design Fundamentals 8/31

8

Implications of Causality



Example: Given a causal, real, and stable system with

$$H_{\mathrm{R}}(\omega) = \frac{1 - a\cos(\omega)}{1 - 2a\cos(\omega) + a^2}, \quad |a| < 1$$

determine $H(\omega)$.

Solution: Using the following identities

$$H_{\mathrm{R}}(\omega) = H_{\mathrm{R}}(z)\big|_{z=\mathrm{e}^{\mathrm{j}\omega}} \text{ and } \cos(\omega) \to \tfrac{1}{2}\left(z + z^{-1}\right)$$

we have

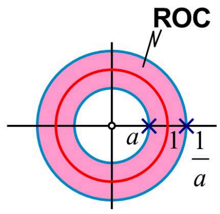
$$\begin{aligned} H_{\mathrm{R}}(z) = \mathcal{Z}\{h_{\mathrm{e}}[n]\} &= \frac{1 - \frac{a}{2}\left(z + z^{-1}\right)}{1 - a\left(z + z^{-1}\right) + a^2} \\ &= \frac{z - \frac{a}{2}\left(z^2 + 1\right)}{(z-a)(1-az)}, \quad |a| < |z| < \left|\tfrac{1}{a}\right| \\ &= \frac{\frac{1}{2} - \frac{1}{a}z^{-1} + \frac{1}{2}z^{-2}}{(1-az^{-1})\left(1 - \frac{1}{a}z^{-1}\right)}, \quad |a| < |z| < \left|\tfrac{1}{a}\right| \end{aligned}$$

EECE-5666: DSP Digital Filter Design Fundamentals 9/31

9

Implications of Causality

Solution: **(continued)**



$$H_R(z) = \frac{\frac{1}{2} - \frac{1}{a}z^{-1} + \frac{1}{2}z^{-2}}{(1-az^{-1})\left(1-\frac{1}{a}z^{-1}\right)}, \quad |a| < |z| < \left|\frac{1}{a}\right|$$

$$= \frac{1}{2} + \frac{\frac{1}{2}\left(a-\frac{1}{a}\right)z^{-1}}{(1-az^{-1})\left(1-\frac{1}{a}z^{-1}\right)}, \quad |a| < |z| < \left|\frac{1}{a}\right|$$

$$= \frac{1}{2} + \frac{\frac{1}{2}}{1-az^{-1}} - \frac{\frac{1}{2}}{1-\frac{1}{a}z^{-1}}, \quad |a| < |z| < \left|\frac{1}{a}\right|$$

After taking the inverse z-transform, we obtain

$$h_e(n) = \tfrac{1}{2}\delta(n) + \tfrac{1}{2}a^n u(n) + \tfrac{1}{2}a^{-n}u(-n-1) = \tfrac{1}{2}\delta(n) + \tfrac{1}{2}a^{|n|}$$

Hence $h(n) = 2h_e(n)u(n) - h_e(n)\delta(n) = a^n u(n) \Rightarrow H(\omega) = \dfrac{1}{1-a\mathrm{e}^{-\mathrm{j}\omega}}$.

Conclusion: For a real, causal, and stable impulse response $h[n]$, $H_R(\omega)$ and $H_I(\omega)$ or $|H(\omega)|$ and $\angle H(\omega)$ ***cannot*** be specified independently.

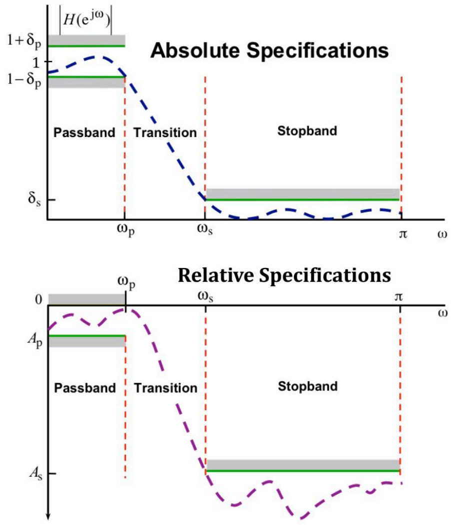
Design of Frequency-Selective Filters

- Constant magnitude and linear phase in the passbands are necessary for distortionless passing of a signal through an LTI filter
- It is impossible to independently specify the magnitude and phase response of a causal LTI system
- Approximate an ideal frequency-selective filter with a realizable filter

Formula used in textbook	$H(z) = \dfrac{\sum_{k=0}^{M-1} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}}$	M = Length of filter
Version used in MATLAB and in these notes is	$H(z) = \dfrac{\sum_{k=0}^{M} b_k z^{-k}}{1+\sum_{k=1}^{N} a_k z^{-k}}$	M = Order of filter

- Filter design approaches
  - Specify magnitude and accept resulting phase or improve phase response by phase equalization
  - Impose linear phase and optimize to satisfy magnitude specifications

Practical Digital Filter Specifications



$$\underset{\text{(Attenuation)}}{\text{dB}} = -20 \log_{10}\left(\frac{\text{Absolute}}{\text{Max Abs}}\right)$$

- Absolute to Relative

$$A_p = 20 \log_{10}\left(\frac{1+\delta_p}{1-\delta_p}\right) > 0(\approx 0)$$

$$A_s = 20 \log_{10}\left(\frac{1+\delta_p}{\delta_s}\right) > 0(\gg 1)$$

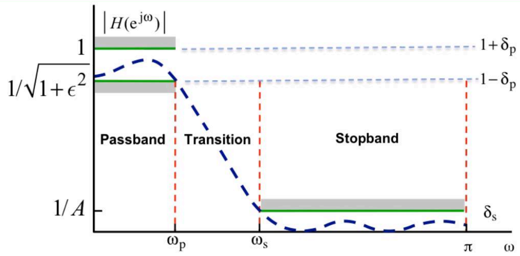
- Relative to Absolute

$$\delta_p = \frac{10^{A_p/20}-1}{10^{A_p/20}+1} \quad (\approx 0)$$

$$\delta_s = \frac{1+\delta_p}{10^{A_s/20}} \quad (\approx 0)$$

12

Practical Analog Filter Specifications



- In practical applications, frequencies are specified in Hz

$$\omega_p = 2\pi\left(F_{\text{pass}}/F_S\right), \quad \omega_s = 2\pi\left(F_{\text{stop}}/F_S\right)$$

- Analog filters are specified using ripple parameter ϵ and attn. parameter A

$$20 \log_{10}\left(\sqrt{1+\epsilon^2}\right) = A_p \quad \text{and} \quad 20 \log_{10}(A) = A_s$$

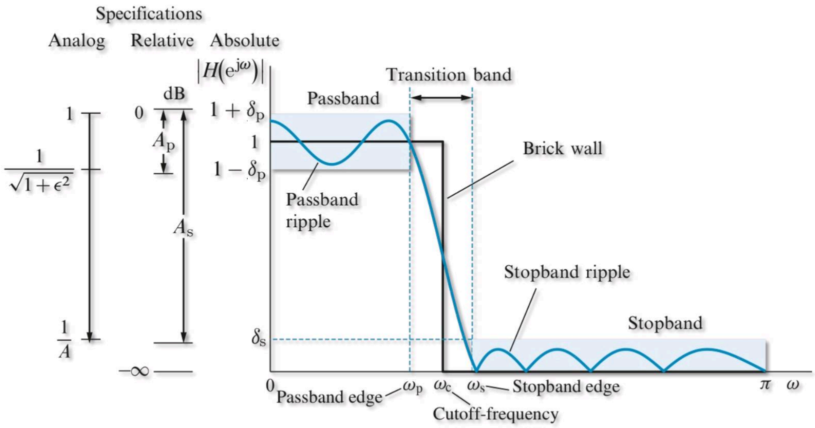
which gives

$$\epsilon = \sqrt{10^{A_p/10}-1} \quad \text{and} \quad A = 10^{A_s/20}$$

13

Practical Digital Filter Specifications

- In summary



Practical Digital Filter Specifications

Example: Conversion of filter coefficients

A lowpass digital filter is specified by the relative specifications:

$$\omega_p = 0.3\pi,\ A_p = 0.5 \text{ dB};\quad \omega_s = 0.5\pi,\ A_s = 40 \text{ dB}.$$

Then the absolute specifications for the filter are given by

$$\begin{aligned} A_p = 0.5 &= 20\log_{10}\left(\frac{1+\delta_p}{1-\delta_p}\right) \Rightarrow \delta_p = 0.0288, \\ A_s = 40 &= 20\log_{10}\left(\frac{1+\delta_p}{\delta_s}\right) \Rightarrow \delta_s = 0.0103. \end{aligned}$$

Similarly the analog filter specifications are given by

$$\epsilon = \sqrt{10^{(0.1A_p)} - 1} = 0.3493 \quad \text{and} \quad A = 10^{(0.05A_s)} = 100.$$

Magnitude vs Amplitude Response

- Frequency response of a linear-phase FIR filter can be written in one of two ways:

$$H(\omega) = |H(\omega)|e^{j\Theta(\omega)} = A(\omega)e^{j\Psi(\omega)}$$

where

$|H(\omega)|$: Magnitude response (real, positive, and even)

$\Theta(\omega)$: Wrapped (broken) phase response

$A(\omega)$: Amplitude response (real and even)

$\Psi(\omega)$: Unwrapped phase response

- Amplitude response is both positive and negative while magnitude response is only non-negative.
- Wrapped phase response is piecewise linear while the unwrapped phase response is truly linear.
- SP Toolbox function: `[A,om,pha] = zerophase(b,a)` computes responses at 512 points along the top of the unit circle.

EECE-5666: DSP Digital Filter Design Fundamentals 16/31

16

Magnitude vs Amplitude Response

Example Consider the impulse response $h(n) = \frac{1}{5}\{\underset{\uparrow}{1}, 1, 1, 1, 1\}$.

Then the frequency response is

$$H(\omega) = \frac{1}{5}\sum_{n=0}^{4} e^{-j\omega n} = \frac{1}{5}\frac{1-e^{-j5\omega/2}}{1-e^{-j\omega}} = \frac{\sin(5\omega/2}{5\sin(\omega/2)}e^{-j2\omega}$$

- Magnitude and wrapped phase response

```
h = [1,1,1,1,1];
[H,om] = freqz(h,1);
magH = abs(H);
WphaH = angle(H)/pi;
```

$$|H(\omega)| = \left|\frac{\sin(5\omega/2}{5\sin(\omega/2)}\right|, \quad -\pi < \omega \leq \pi$$

$$\Theta(\omega) = -2\omega \pm \pi\{\text{when } |H(\omega)| < 0\}$$

- Amplitude and unwrapped phase response

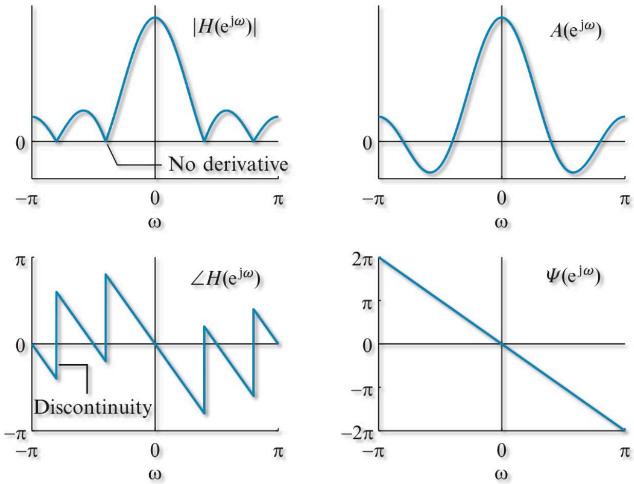
```
[A,om,UnwPhaH] = zerophase(h,1);
```

$$A(\omega) = \frac{\sin(5\omega/2}{5\sin(\omega/2)}, \quad \Psi(\omega) = -2\omega, \qquad -\pi < \omega \leq \pi$$

EECE-5666: DSP Digital Filter Design Fundamentals 17/31

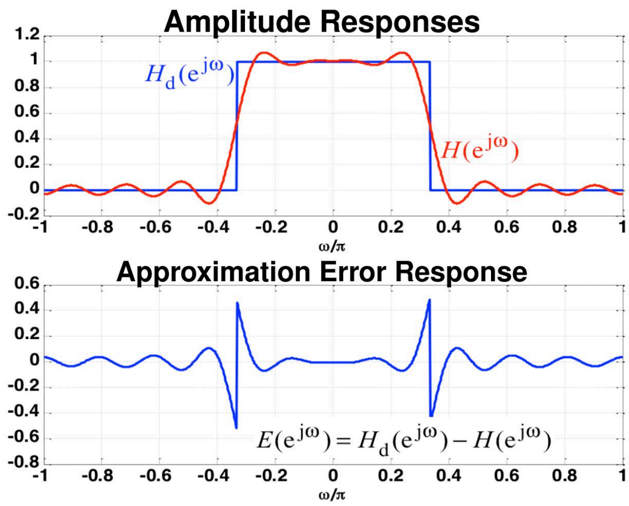
17

Magnitude vs Amplitude Response



18

Approximation Error



19

Approximation Error Criteria

Definitions:

M: Filter order
Book notation: $M-1$

$$H_{\mathrm{d}}(\omega) : \text{Desired ideal frequency response}$$

$$H(\omega) = \frac{\sum_{k=0}^{M} b_k \, \mathrm{e}^{\mathrm{j}k\omega}}{1+\sum_{k=0}^{N} a_k \, \mathrm{e}^{\mathrm{j}k\omega}} : \text{Realizable frequency response}$$

$$E(\omega) = H_{\mathrm{d}}(\omega) - H(\omega) : \text{Approximation error}$$

- To obtain the best possible design, we optimize the approximation error $E(\omega)$ using one of the following three criteria.
  - Minimum mean-squared-error approximation (used in window dsign)
  - Minimax error approximation (used in equiripple filter design)
  - Maximally flat error approximation (used in elliptic filter design)

Design Problem Statement

- The previous specifications are given for a lowpass filter. (See Slide-14)
- Similar specifications can also be given for other types of frequency-selective filters, such as highpass or bandpass.
- However, the most important design parameters are ***frequency-band tolerances*** (or ripples) and ***band-edge frequencies***.
- Whether the given band is a passband or stopband is a relatively minor issue.
- Therefore in describing design techniques, we will concentrate on a lowpass filter design.

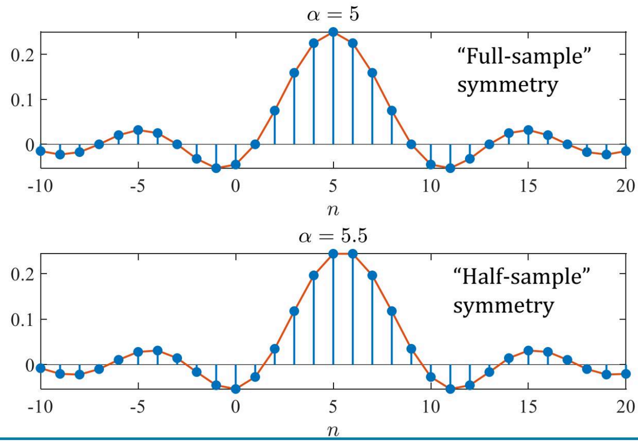
Statement: Design a lowpass filter (i.e., obtain its system function $H(z)$ or its difference equation) that has a passband $[0, \omega_{\mathrm{p}}]$ with tolerance δ_{p} (or A_{p} in dB) and a stopband $[\omega_{\mathrm{s}}, \pi]$ with tolerance δ_{s} (or A_{s} in dB).

Ideal Low-Pass Filter with Linear Phase

$$H_{lp}(\omega) = \begin{cases} e^{-j\alpha\omega}, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases} \Rightarrow h_{lp}(n) = \frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}$$

$$|H_{lp}(\omega)| = 1$$

$$\angle H_{lp}(\omega) = -\alpha\omega$$



Symmetry

$h(\alpha - k) = h(\alpha + k) \Rightarrow$

$h(2\alpha - n) = h(n)$

$\alpha =$ Integer or $\alpha =$ Integer$+\frac{1}{2}$

$2\alpha =$ Integer

Causal IIR filters cannot have linear phase

22

M: Filter order
Book notation: $M - 1$

FIR Filters with Linear Phase

$$y(n) = \sum_{k=0}^{M} h(k)x(n-k) \Rightarrow H(\omega) = \sum_{k=0}^{M} h(k)e^{-j\omega k}$$

- What conditions on the filter coefficients guarantee linear phase $\angle H(\omega) = -\alpha\omega$?
- Do these conditions impose any constraints on the frequency response $H(\omega)$?

$M = 4$ (even) | Even symmetry | $h(0) = h(4), h(1) = h(3)$ | $\alpha = \frac{M}{2} = 2$

$H(\omega) = h(0) + h(1)e^{-j\omega} + h(2)e^{-j2\omega} + h(3)e^{-j3\omega} + h(4)e^{-j4\omega}$

$H(\omega) = e^{-j2\omega}\left[h(0)e^{j2\omega} + h(1)e^{j\omega} + h(2) + h(3)e^{-j\omega} + h(4)e^{-j2\omega}\right]$

$H(\omega) = [2h(0)\cos 2\omega + 2h(1)\cos\omega + h(2)]\, e^{-j2\omega} = A(\omega)e^{j\Phi(\omega)}, \;\; \Phi(\omega) = -2\omega$

$A(\omega) \ge 0 \Rightarrow |H(\omega)| = A(\omega)$ and $\Theta(\omega) = \angle H(\omega) = \Phi(\omega) =$ Linear Phase

$A(\omega) < 0 \Rightarrow |H(\omega)| = -A(\omega)$ and $\Theta(\omega) = \angle H(\omega) = \Phi(\omega) + \pi =$ Generalized Linear Phase

23

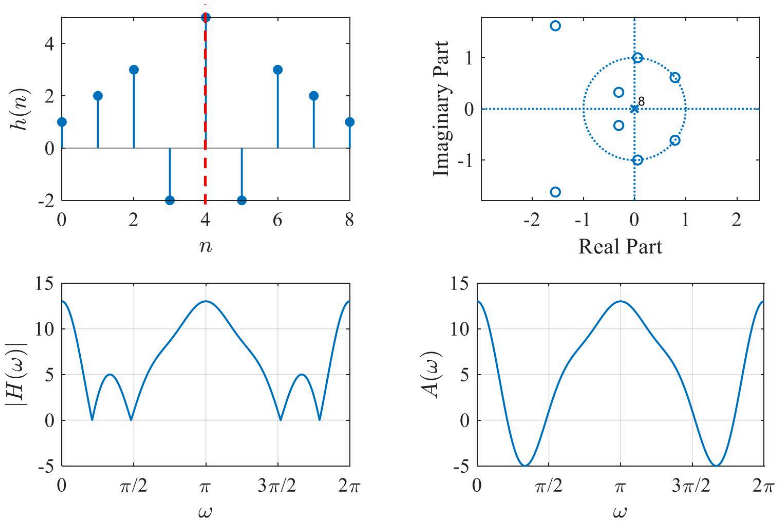
M: Filter order
Book notation: $M-1$

Type-I or Case-1 FIR Filters

$M = 8$

$\alpha = \dfrac{M}{2} = 4$

$$A(\omega) = \sum_{k=0}^{M/2} a(k)\cos\omega k$$



EECE-5666: DSP Digital Filter Design Fundamentals 24/31

24

M: Filter order
Book notation: $M-1$

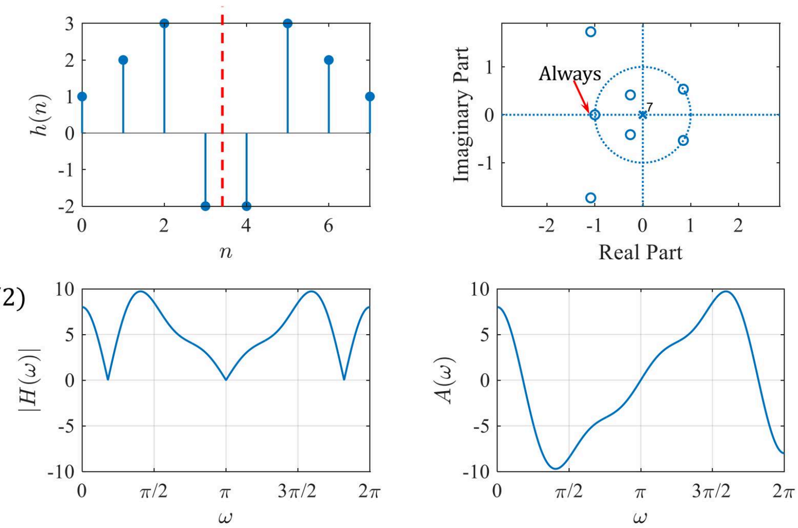
Type-II or Case-2 FIR Filters

$M = 7$

$\alpha = \dfrac{M}{2} = 3.5$

$$A(\omega) = \sum_{k=1}^{\frac{M+1}{2}} b(k)\cos\omega(k-1/2)$$

Unsuitable for HP filters



EECE-5666: DSP Digital Filter Design Fundamentals 25/31

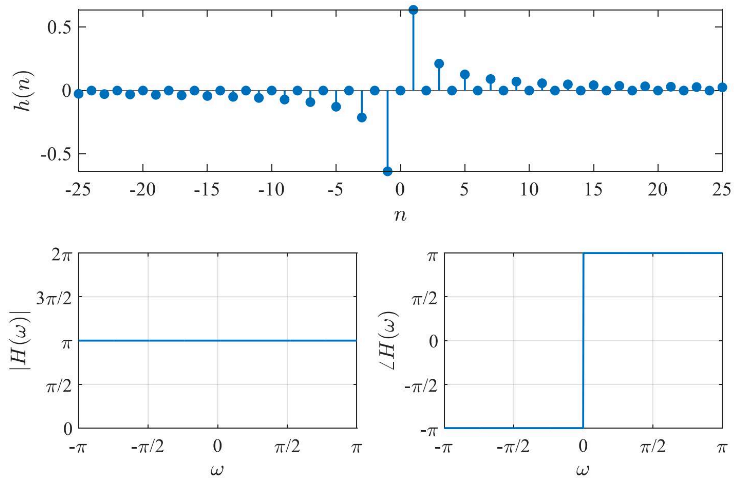
25

Ideal Hilbert Transformer

$$H(\omega)=\begin{cases}-j, & 0<\omega<\pi\\ j, & -\pi<\omega<0\end{cases}$$

$$h(n)=\begin{cases}\dfrac{2}{\pi}\dfrac{\sin^2(\pi n/2)}{n}, & n\neq 0\\ 0, & n=0\end{cases}$$

$$h(n)=\begin{cases}0, n=\text{even}\\ \dfrac{2}{\pi n}, n=\text{odd}\end{cases}$$



EECE-5666: DSP Digital Filter Design Fundamentals 26/31

26

M: Filter order
Book notation: $M-1$

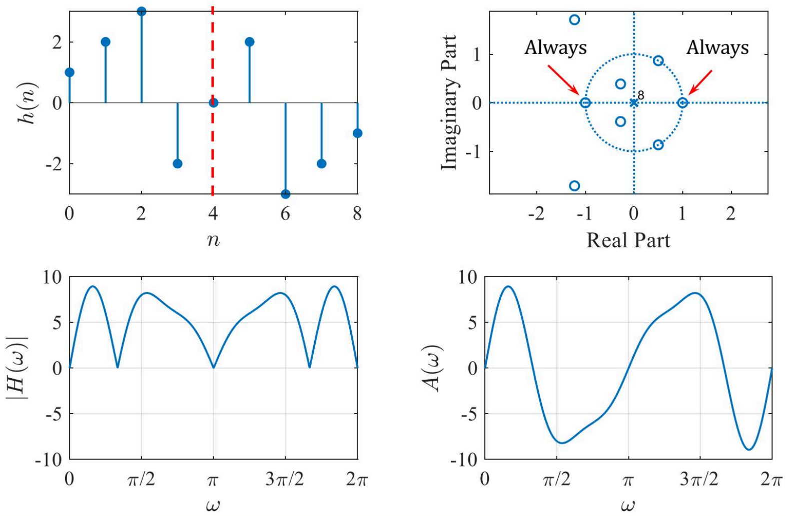
Type-III or Case-3 FIR Filters

$M=8$

$\alpha=\dfrac{M}{2}=4$

$$A(\omega)=\sum_{k=0}^{M/2}c(k)\sin\omega k$$

Suitable for BP filters and Hilbert transformers



EECE-5666: DSP Digital Filter Design Fundamentals 27/31

27

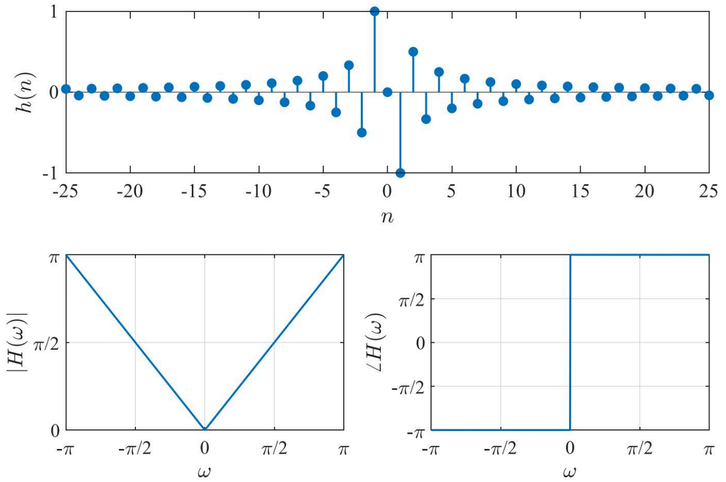
Ideal Differentiator

$$H_d(\omega) = (j\omega)e^{-j\alpha\omega}, \qquad -\pi < \omega \le \pi$$

$$h_d(n) = \frac{\cos[\pi(n-\alpha)]}{n-\alpha} - \frac{\sin[\pi(n-\alpha)]}{\pi(n-\alpha)^2}$$

$$\alpha = 0 \Rightarrow H_d(\omega) = j\omega$$

$$h_d(n) = \begin{cases} 0, n = 0 \\ \dfrac{(-1)^n}{n}, |n| > 0 \end{cases}$$



EECE-5666: DSP Digital Filter Design Fundamentals 28/31

28

Type-IV or Case-4 FIR Filters

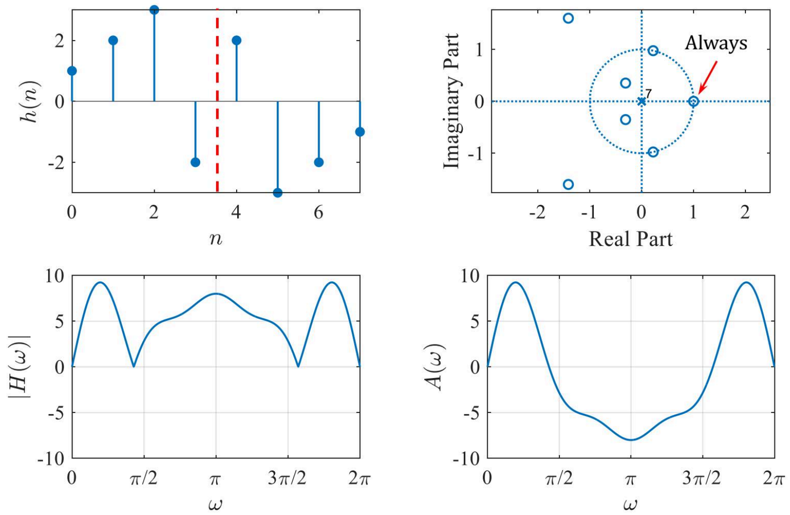
M: Filter order
Book notation: $M-1$

$$M = 7$$

$$\alpha = \frac{M}{2} = 3.5$$

$$A(\omega) = \sum_{k=1}^{\frac{M+1}{2}} d(k)\sin\omega(k - 1/2)$$

Suitable for differentiators



EECE-5666: DSP Digital Filter Design Fundamentals 29/31

29

Generalized Linear Phase

All Type 1, 2, 3, and 4 frequency responses can be expressed in the form:

$$H(\omega) = |H(\omega)|e^{j\Theta(\omega)} = A(\omega)e^{j\Phi(\omega)}$$

$A(\omega) =$ Real-valued

$\Phi(\omega) = -\alpha\omega + \beta$ (Affine function of ω)

$$\Theta(\omega) = -\alpha\omega + \beta + \begin{cases} 0, & A(\omega) \geq 0 \\ \pi, & A(\omega) < 0 \end{cases}$$

$$\tau_g(\omega) = -\frac{d\Theta(\omega)}{d\omega} = \alpha$$

$\Phi(\omega) = -\alpha\omega + \beta \Rightarrow$ Generalized linear phase or constant group delay

Implications

- For frequency selective FIR filters use symmetric impulse response with odd length (or even order)
- For FIR differentiators use anti-symmetric impulse response with even length (odd order)
- For FIR Hilbert transformers use anti-symmetric impulse response with odd length (even order)
- For FIR filter design using optimization techniques use the amplitude response $A(\omega)$ because it is differentiable

EECE-5666: DSP Digital Filter Design Fundamentals 30/31

30

M: Filter order
Book notation: $M - 1$

Types of FIR Filters with Linear Phase

Table 10.1 Properties of impulse response sequence $h[n]$ and frequency response function $H(e^{j\omega}) = A(e^{j\omega})e^{j\Psi(e^{j\omega})}$ of FIR filters with linear phase.

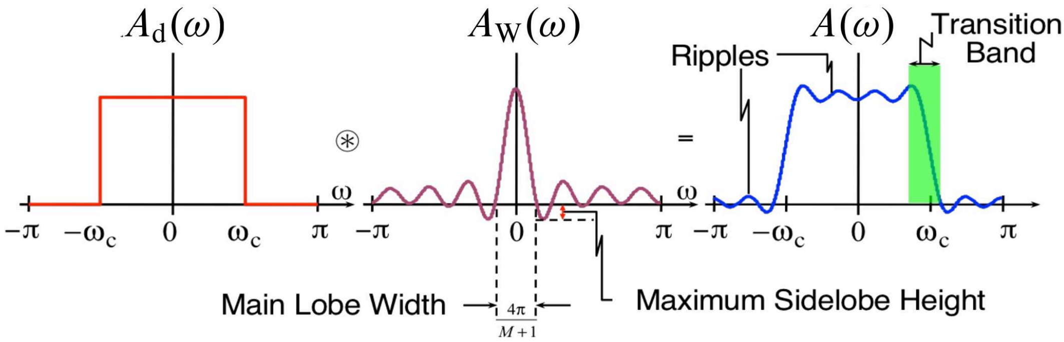
Type	$h[k]$	M	$A(e^{j\omega})$	$A(e^{j\omega})$	$\Psi(e^{j\omega})$
I	even	even	$\sum_{k=0}^{M/2} a[k]\cos\omega k$	even–no restriction	$-\frac{\omega M}{2}$
II	even	odd	$\sum_{k=1}^{\frac{M+1}{2}} b[k]\cos\left[\omega\left(k-\frac{1}{2}\right)\right]$	even $A(e^{j\pi}) = 0$	$-\frac{\omega M}{2}$
III	odd	even	$\sum_{k=1}^{M/2} c[k]\sin\omega k$	odd $A(e^{j0}) = 0$ $A(e^{j\pi}) = 0$	$\frac{\pi}{2} - \frac{\omega M}{2}$
IV	odd	odd	$\sum_{k=1}^{\frac{M+1}{2}} d[k]\sin\left[\omega\left(k-\frac{1}{2}\right)\right]$	odd $A(e^{j0}) = 0$	$\frac{\pi}{2} - \frac{\omega M}{2}$

EECE-5666: DSP Digital Filter Design Fundamentals 31/31

31

FIR Filter Design

Window Design Method



0

Window Design of FIR Filters with Linear Phase

- Consider an ideal lowpass filter as the desired filter

$$H_d(\omega) = \begin{cases} (1)e^{-j\omega}, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases} \quad \Rightarrow \quad h_d(n) = \frac{\sin\big(\omega_c(n-\alpha)\big)}{\pi(n-\alpha)}$$

- For $\alpha = M/2$, $h_d(n) = h_d[M-n]$, $0 \le n \le M \Rightarrow$ Even symmetry
- Let $w(n) = w[M-n]$ be a **window** or truncation function which is nonzero over $0 \le n \le M$ and symmetric with respect to $M/2$.
- Then a **causal**, **linear-phase**, **FIR** filter is obtained as

$$h(n) = h_d(n)w(n) = h[M-n], \quad 0 \le n \le M$$

- This truncation minimizes the **mean-squared error** between $h_d(n)$ and $h(n)$.
- The frequency response of $h(n)$ is given by

$$H(\omega) = \sum_{n=0}^{M} h_d(n)w(n)e^{-j\omega n} = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\theta)W((\omega-\theta))d\theta$$

which is a **periodic convolution** $H(\omega) \circledast W(\omega)$.

1

Truncation : Rectangular Windowing

- Consider a rectangular window applied to an ideal LPF to obtain the FIR filter

$$h(n) = h_d(n)w(n), \quad h_d(n) = h_{lp}(n) = \frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}, \quad w(n) = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

Then

$$W(\omega) = \sum_{0}^{M} e^{-j\omega m} = \underbrace{\frac{\sin(\omega(M+1)/2]}{\sin(\omega/2)}}_{A_W(\omega)}, e^{-j\omega M/2}$$

while

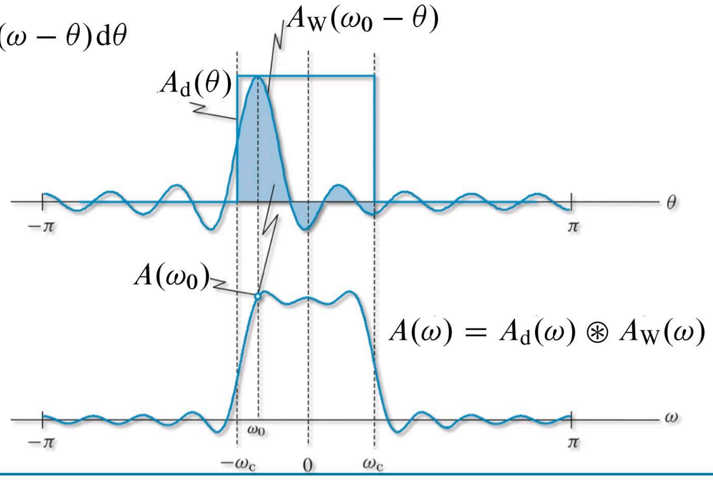
$$H_d(\omega) = \begin{cases} 1 \cdot e^{-j\alpha\omega}, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases} = e^{-j\alpha\omega} \underbrace{\begin{cases} 1, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases}}_{A_d(\omega)}$$

and $H(\omega) = A(\omega)e^{j\Psi(\omega)} = A(\omega)e^{-j\omega M/2}$. Hence

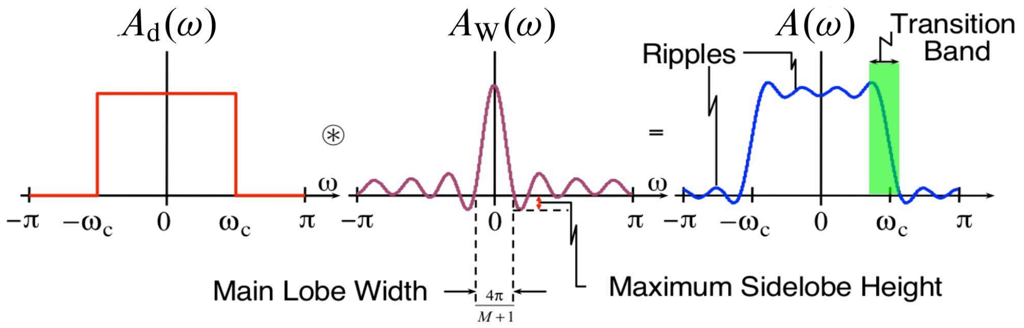
$$H(\omega) = H_d(\omega) \circledast W(\omega) \quad \Rightarrow \quad A(\omega) = A_d(\omega) \circledast A_W(\omega)$$

Effects of Windowing in the Frequency Domain

$$A(\omega) = A_d(\omega) \circledast A_W(\omega)$$
$$= \frac{1}{2\pi}\int_0^{2\pi} A_d(\theta)A_W(\omega-\theta)d\theta$$



Effects of Windowing in the Frequency Domain



- The finite length $L = M + 1$ of the window results in a main lobe with width proportional to $1/L$.
- The convolution **smears** the ideal response.
- The main lobe produces the transition band.
- The side lobes produce pass band and stop band ripples with the same maximum size

4

Ripples and Transition Bandwidth

- A careful but approximate analysis of ripple values and transition bandwidth for rectangular window can be done (see Appendix) for which

$$A_d(\omega) = \begin{cases} 1, & |\omega| \le \omega_c \\ 0, & \omega_c \le |\omega| \le \pi \end{cases} \quad \text{and } A_W(\omega) = \frac{\sin(\omega L/2)}{\sin(\omega/2)}; \; L = M + 1$$

- Then using

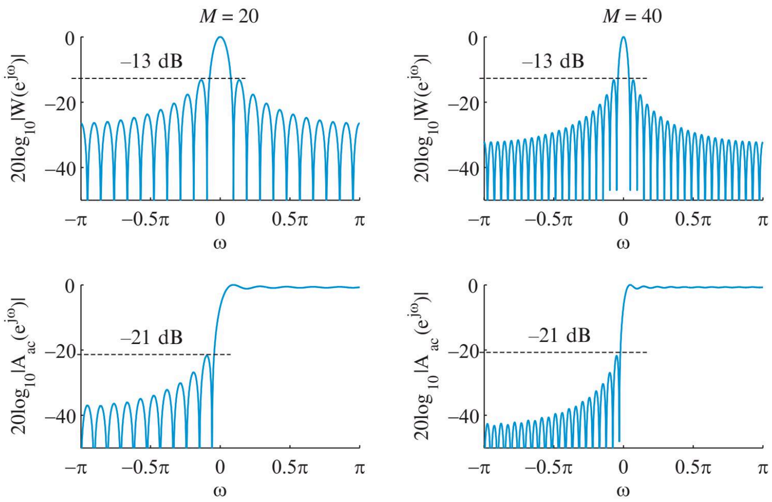
$$A(\omega) = A_d(\omega) \circledast A_W(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} A_d(\theta) A_W((\omega - \theta))\mathrm{d}\theta$$
$$= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} A_W((\omega - \theta))\mathrm{d}\theta$$

we can show that for $L \gg 1$

1. $\delta_p \approx \delta_s \approx 0.0895$, irrespective of order M,
2. Passband ripple $A_p = 20\log_{10}(1.0895) = 0.75$ dB
3. Stopband ripple $A_s = 20\log_{10}(1/0.0895) = 21$ dB (not sufficient)
4. Transition bandwidth $\Delta\omega \triangleq \omega_s - \omega_p \approx \frac{1.8\pi}{L} < \frac{4\pi}{L}$

5

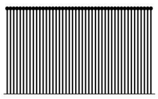
Rectangular Window



6

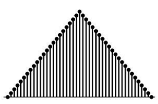
Fixed (Attenuation) Windows

Rectangular



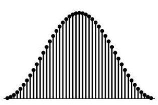
$$w[n] = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

Bartlett (triangular)



$$w[n] = \begin{cases} 2n/M, & 0 \le n \le M/2, \quad M \text{ even} \\ 2 - 2n/M, & M/2 < n \le M \\ 0, & \text{otherwise} \end{cases}$$

Hann

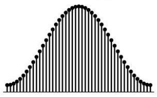


$$w[n] = \begin{cases} 0.5 - 0.5\cos(2\pi n/M), & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

7

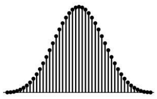
Fixed (Attenuation) Windows

Hamming



$$w[n] = \begin{cases} 0.54 - 0.46\cos(2\pi n/M), & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

Blackman

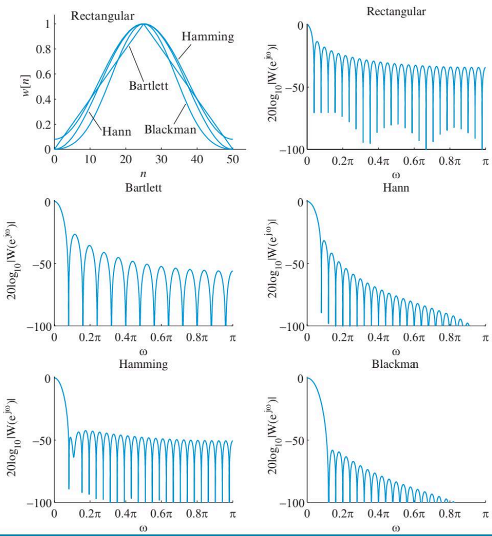


$$w[n] = \begin{cases} 0.42 - 0.5\cos(2\pi n/M) + 0.08\cos(4\pi n/M), & 0 \le n \le M \\ 0. & \text{otherwise} \end{cases}$$

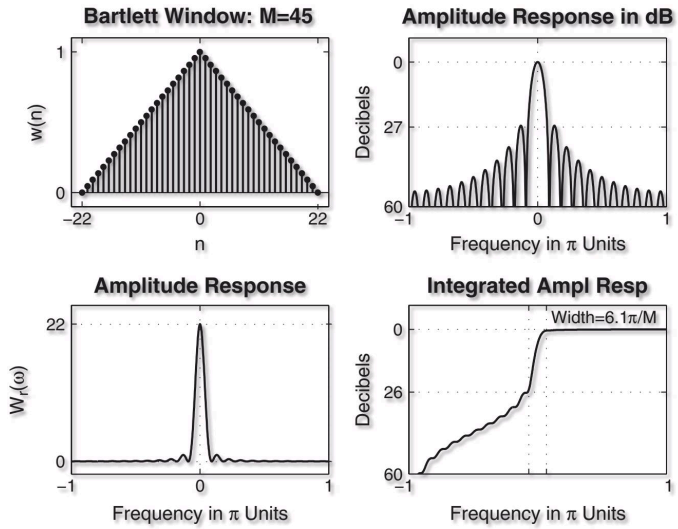
- For non-rectangular windows, even the approximate analysis to determine ripple values and transition bandwidth is not possible. Hence we numerically compute

$$A_{\text{ac}}(\omega) \triangleq \int_{-\pi}^{\omega} A_{\text{W}}(\theta)\,\mathrm{d}\theta : \text{ Integrated or accumulated Ampl. Resp.}$$

Fixed (Attenuation) Window Responses



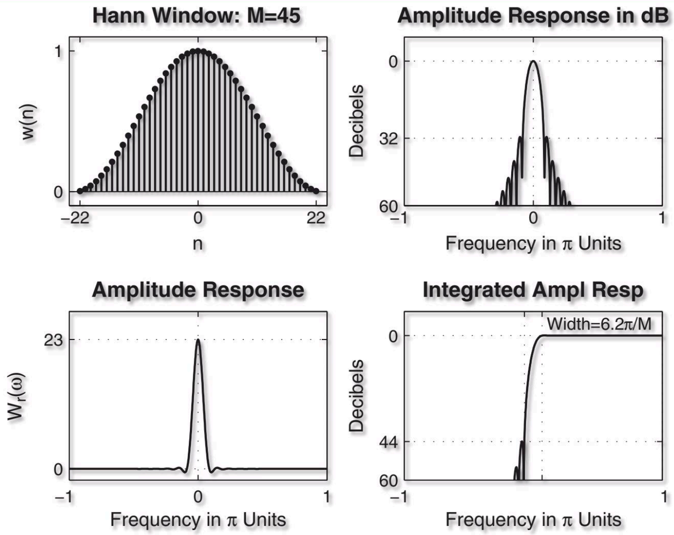
Fixed Window Integrated Amplitude Responses



EECE-5666: DSP FIR Filter Design - Window method 10/33

10

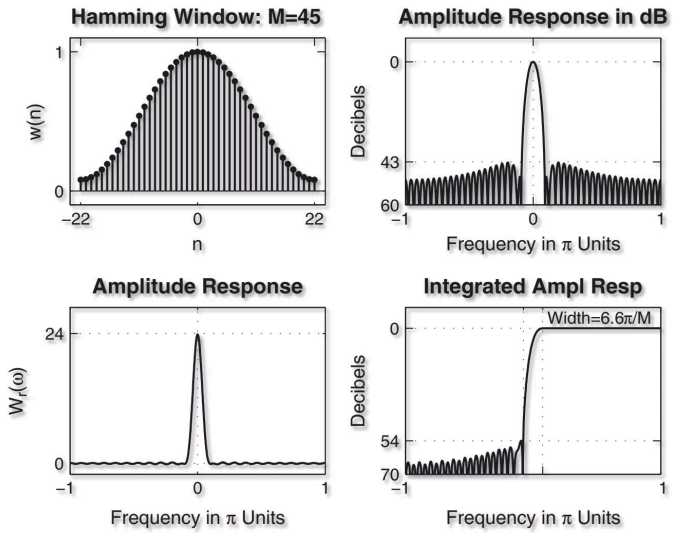
Fixed Window Integrated Amplitude Responses



EECE-5666: DSP FIR Filter Design - Window method 11/33

11

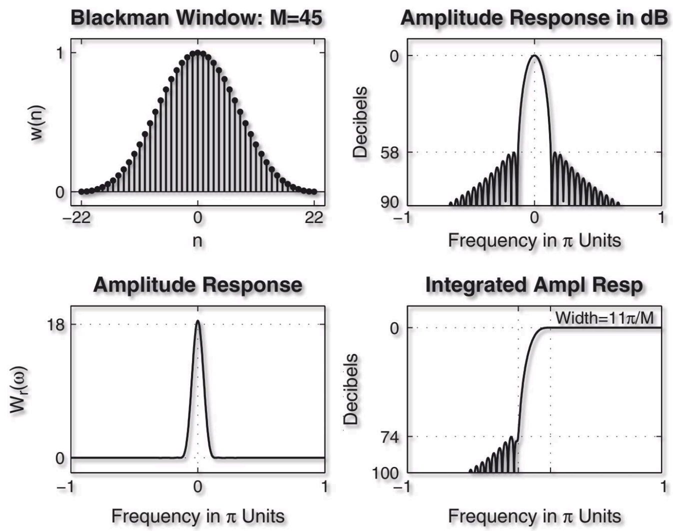
Fixed Window Integrated Amplitude Responses



EECE-5666: DSP FIR Filter Design - Window method 12/33

12

Fixed Window Integrated Amplitude Responses



EECE-5666: DSP FIR Filter Design - Window method 13/33

13

Comparison of Fixed (Attenuation) Windows

Table 10.3 Properties of commonly used windows ($L = M + 1$).

Window name	Side lobe level (dB)	Approx. $\Delta\omega$	Exact $\Delta\omega$	$\delta_p \approx \delta_s$	A_p (dB)	A_s (dB)
Rectangular	-13	$4\pi/L$	$1.8\pi/L$	0.09	0.75	21
Bartlett	-25	$8\pi/L$	$6.1\pi/L$	0.05	0.45	26
Hann	-31	$8\pi/L$	$6.2\pi/L$	0.0063	0.055	44
Hamming	-41	$8\pi/L$	$6.6\pi/L$	0.0022	0.019	53
Blackman	-57	$12\pi/L$	$11\pi/L$	0.0002	0.0017	74

- Ripple (stopband attenuation) does not depend on cutoff frequency ω_c or filter order M
- Transition band $\Delta\omega$ does and is proportional to $1/L$ or $1/(M+1)$

14

MATLAB Functions for Fixed Windows

Rectangular window: `w = rectwin(L)` returns an `L`-point rectangular window values in array `w`.

Bartlett window: `w = bartlett(L)` returns an `L`-point Bartlett (triangular) window values in array `w`

Hann window: `w = hann(L)` returns an `L`-point Hann window values in array `w`

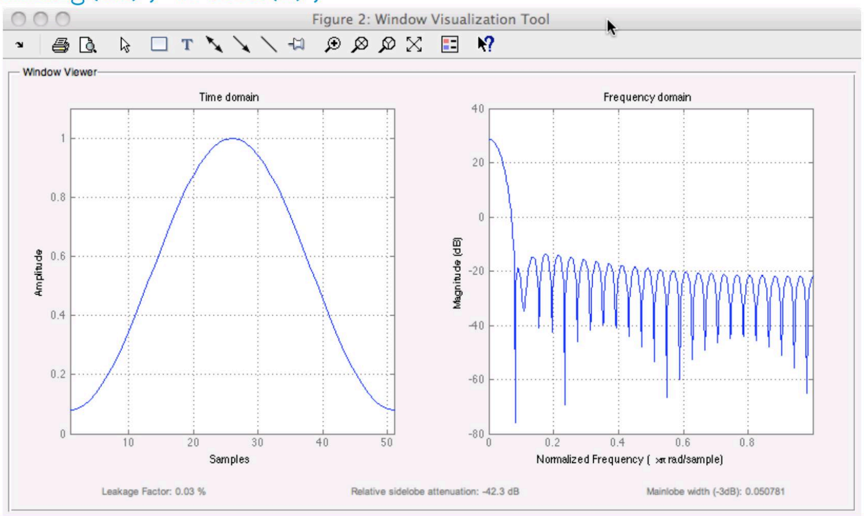
Hamming Window: `w = hamming(L)` returns an `L`-point Hamming window values in array `w`

Blackman window: `w = blackman(L)` returns an `L`-point Blackman window values in array `w`

15

Window Visualization Tool

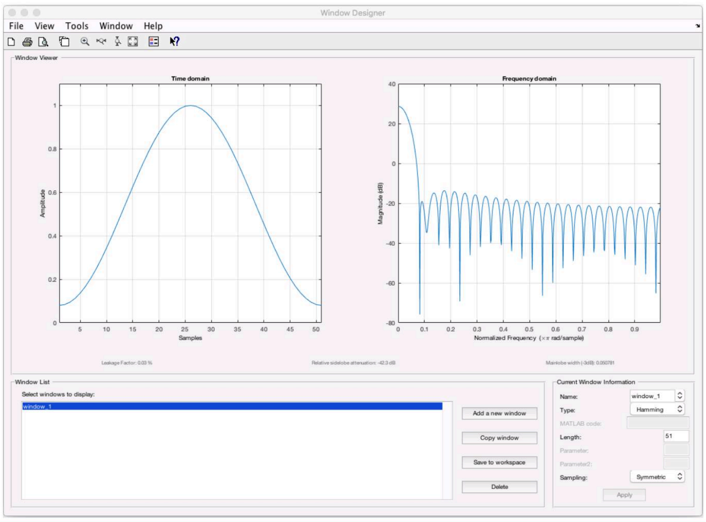
```
>> w = hamming(51); wvtool(w);
```



16

Window Designer App

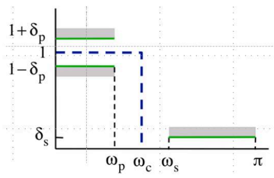
```
>> windowdesigner
```



17

Design Procedure of LP FIR Filters Using Fixed Windows

1. Set $\delta = \min(\delta_p, \delta_s)$ because the window design method inherently gives $\delta_p = \delta_s$
2. Determine the cutoff frequency of the underlying ideal LPFL $\omega_c = (\omega_p + \omega_s)/2$
3. Determine the transition band $\Delta\omega$ and the stopband attenuation of the filter:
$$\Delta\omega = \omega_s - \omega_p, \quad A_s = -20\log_{10}(\delta)$$
4. Select a window function that gives the smallest stopband attenuation that is more than or equal to A_s.
5. For the chosen window function $w(n)$, determine the window length $L = M + 1$ that satisfies the exact transition-width from the table in Slide-14.
6. Compute the impulse response $h(n) = h_d(n)w(n)$
$$h(n) = \left\{ \frac{\sin\left[\omega_c\left(n - \frac{M}{2}\right)\right]}{\pi\left(n - \frac{M}{2}\right)} \right\} w(n)$$



Lowpass Filter Design Example: Hamming Window

Design a lowpass linear-phase filter to satisfy: $\omega_p = 0.25\pi$, $\omega_s = 0.35\pi$, $A_p = 0.1$dB, $A_s = 50$dB

1. Convert relative dB specifications to absolute specs:
$$A_p = 0.1\text{dB} \Rightarrow \delta_p = 0.0058; \qquad A_s = 50\text{dB} \Rightarrow \delta_s = 0.0032; \quad \Rightarrow \quad \delta = 0.0032$$
2. Set LPF cutoff frequency and compute the transition bandwidth
$$\omega_c = (\omega_p + \omega_s)/2 = 0.3\pi \quad \text{and} \quad \Delta\omega = \omega_s - \omega_p = 0.1\pi$$
3. Determine the transition band $\Delta\omega$ and the stopband attenuation of the filter:
$$\Delta\omega = \omega_s - \omega_p, \quad A_s = -20\log_{10}(\delta)$$
4. From the table on Slide-14, we select the Hamming window. Then using the transition width formula $\Delta\omega \approx 6.6\pi/L = 0.1\pi \quad \Rightarrow \quad L = 66$
5. We choose length $L = 67$ or order $M = 66$ to obtain Type-I filter which gives $\alpha = M/2 = 33$
6. Compute the impulse response $h(n) = h_d(n)w(n)$

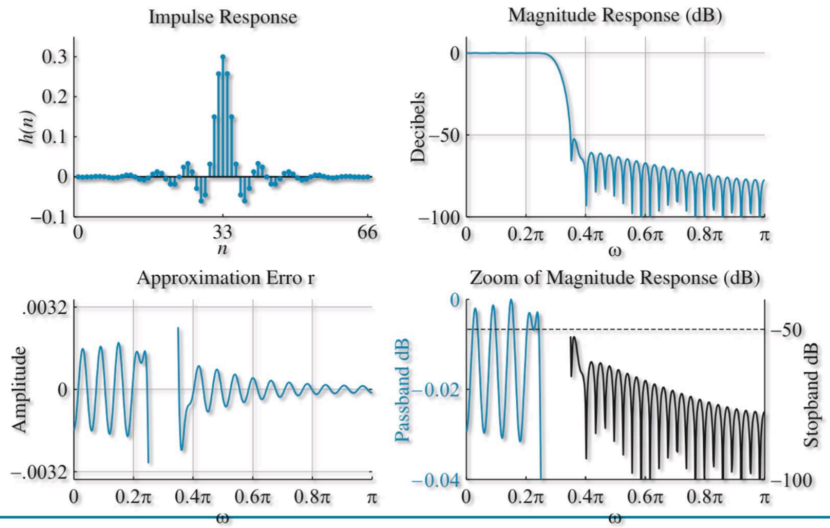
MATLAB Functions from ADSP Toolbox

```
function hd = ideallp(omc,M);
% Ideal LowPass filter computation
% --------------------------------
% [hd] = ideal_lp(omc,M)
%  hd = ideal impulse response between 0 to M-1
% omc = cutoff frequency in radians
%   M = order of the ideal filter
%
n = (0:M)'-M/2;
h = (omc/pi)*ones(size(n));
I = find(n);
h(I) =sin(omc*n(I))./(pi*n(I));
```


Lowpass Filter Design Example: Hamming Window

```
>> wp = 0.25*pi; ws = 0.35*pi; Ap = 0.1; As = 50;
>> deltap = (10^(Ap/20)-1)/(10^(Ap/20)+1);
>> deltas = (1+deltap)/(10^(As/20));
>> delta = min(deltap,deltas);
>> A = -20*log10(delta);
>> Deltaw = ws-wp; omegac = (ws+wp)/2;
>> L = ceil(6.6*pi/Deltaw)+1; % Window length
>> M=L-1;                     % Window order
>> n = 0:M; hd = ideallp(omegac,M);
>> h = hd.*hamming(L)';
```

Lowpass Filter Design Example: Hamming Window



Kaiser Window

- Unlike fixed windows, this window has an adjustable parameter, β, which controls the interplay between transition bandwidth and the stopband attenuation, given by

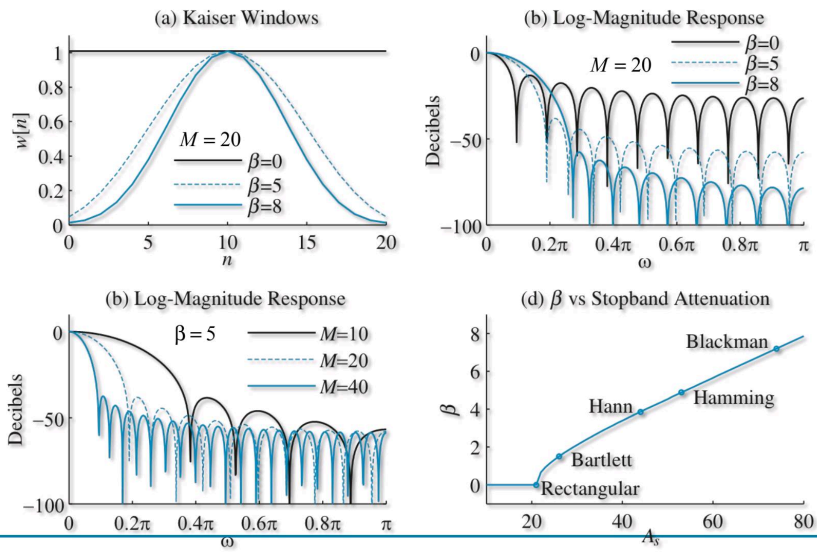
$$w[n] = \frac{I_0\left[\beta\sqrt{1-\left(1-\frac{2n}{M}\right)^2}\right]}{I_0[\beta]}, \quad 0 \le n \le M$$

where

$$I_0(x) = 1 + \sum_{k=0}^{\infty}\left[\frac{(x/2)^k}{k!}\right]^2$$

- If $\beta = 5.658$ then $\Delta\omega = 7.7\pi/M$ and $A_s = 60$ dB
- If $\beta = 4.538$ then $\Delta\omega = 5.8\pi/M$ and $A_s = 50$ dB
- `w = kaiser(L,beta)` returns the `beta`-valued `L`-point Kaiser window in array `w`

Kaiser Window

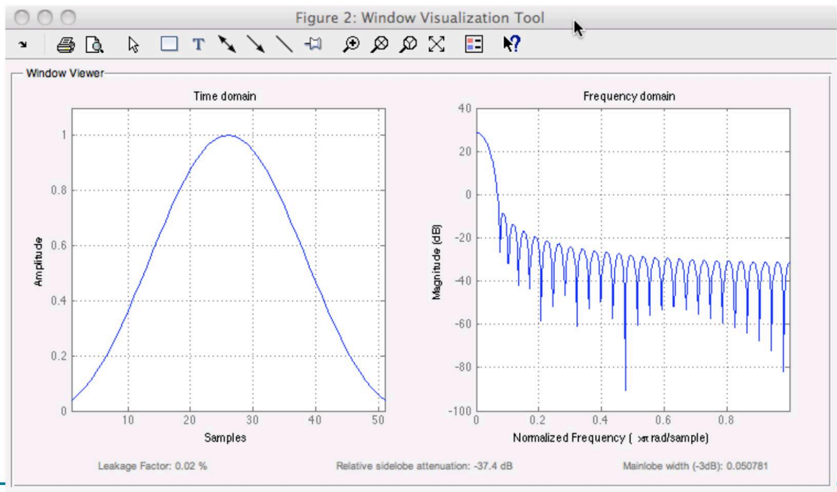


EECE-5666: DSP FIR Filter Design - Window method 24/33

24

Window Visualization Tool

```
>> w = kaiser(51,2.5); wvtool(w);
```

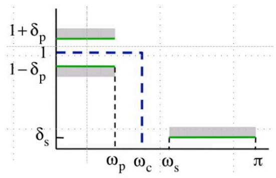


EECE-5666: DSP FIR Filter Design - Window method 25/33

25

Lowpass Filter Design Procedure: Kaiser Window

1. Set $\delta = \min(\delta_p, \delta_s)$ because the window design method inherently gives $\delta_p = \delta_s$
2. Determine the cutoff frequency of the underlying ideal LPFL $\omega_c = (\omega_p + \omega_s)/2$
3. Determine the transition band $\Delta\omega$ and the stopband attenuation of the filter:
$$\Delta\omega = \omega_s - \omega_p, \quad A_s = -20\log_{10}(\delta)$$
4. Determine the parameters of the Kaiser window $w_K(n)$ using empirical design equations:
$$\beta = \begin{cases} 0, & A_s < 21 \\ 0.5842(A_s - 21)^{0.4} + 0.07886(A_s - 21), & 21 \le A_s \le 50 \\ 0.1102(A_s - 8.7), & A_s > 50 \end{cases}$$
$$M = (A_s - 8)/(2.28\Delta\omega$$
5. Compute the impulse response $h(n) = h_d(n)w_K(n)$
$$h(n) = \left\{ \frac{\sin\left[\omega_c\left(n - \frac{M}{2}\right)\right]}{\pi\left(n - \frac{M}{2}\right)} \right\} w_K(n)$$



26

Lowpass Filter Design Example: Kaiser Window

Design a lowpass linear-phase filter to satisfy: $\omega_p = 0.25\pi$, $\omega_s = 0.35\pi$, $A_p = 0.1$dB, $A_s = 50$dB

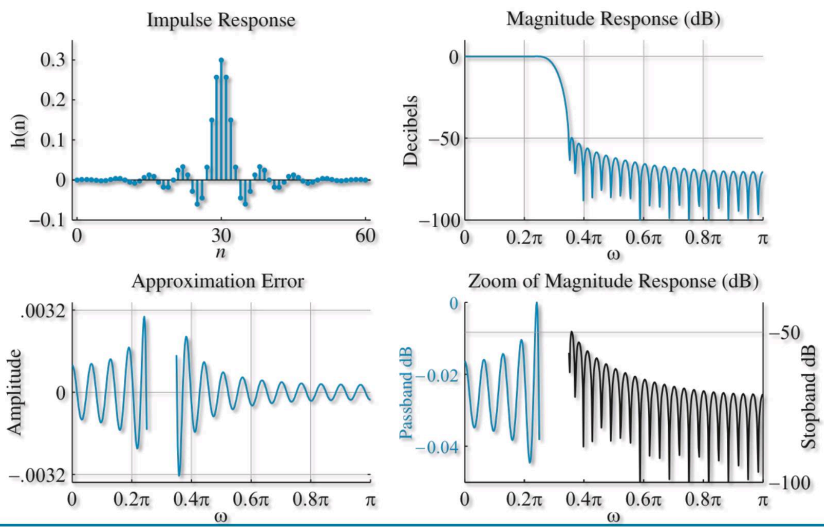
1. Convert relative dB specifications to absolute specs:
$$A_p = 0.1\text{dB} \Rightarrow \delta_p = 0.0058; \qquad A_s = 50\text{dB} \Rightarrow \delta_s = 0.0032; \quad \Rightarrow \quad \delta = 0.0032$$
2. Set LPF cutoff frequency and compute the transition bandwidth
$$\omega_c = (\omega_p + \omega_s)/2 = 0.3\pi \quad \text{and} \quad \Delta\omega = \omega_s - \omega_p = 0.1\pi$$
3. Determine the transition band $\Delta\omega$ and the stopband attenuation of the filter:
$$\Delta\omega = \omega_s - \omega_p, \quad A_s = -20\log_{10}(\delta)$$
4. Determine Kaiser window parameters and length:
$$A_s = 50\text{dB} \Rightarrow \beta = 0.5842(A_s - 21)^{0.4} + 0.07886(A_s - 21) = 4.528$$
$$\Delta\omega = 0.1\pi \Rightarrow M = (A_s - 8)/(2.28\Delta\omega) = 59$$
We choose order $M = 60$ or length $L = 61$ to obtain Type-I filter which gives $\alpha = M/2 = 30$
5. Compute the impulse response $h(n) = h_d(n)w_K(n)$

27

Lowpass Filter Design Example: Kaiser Window

```
>> beta = 0.5842*(A-21)^(0.4)+0.07886*(A-21); % Kaiser beta
>> M = ceil((A-8)/(2.285*Deltaw))+1;          % window order
>> L = M+1;                                   % window length
>> alpha = M/2;
>> n = 0:M;
>> hd = wc*sinc(wc*(n-alpha));
>> h = hd.*kaiser(L,beta)';
```


Lowpass Filter Design Example: Kaiser Window



Bandpass Filter Design Example: Kaiser Window

Design a bandpass linear-phase filter using a Kaiser window to satisfy:]vspace*-0.1in

$$\begin{aligned} |H(\omega)| &\le 0.01, & |\omega| &\le 0.2\pi \\ 0.99 \le |H(\omega)| &\le 1.01, & 0.3\pi \le |\omega| &\le 0.7\pi \\ |H(\omega)| &\le 0.01, & 0.78\pi \le |\omega| &\le \pi \end{aligned}$$

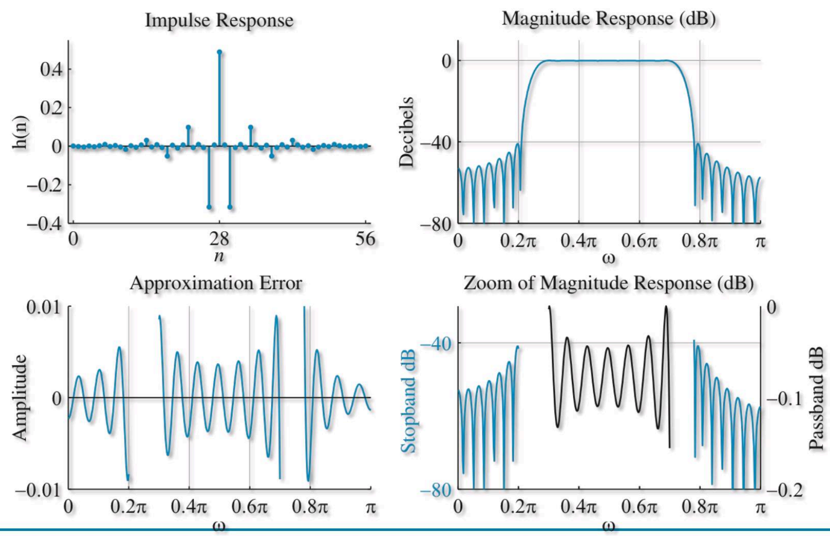
1. There are three band ripple parameters and two transition bandwidths
2. Window design approach can only satisfy one ripple parameter and one transition bandwidth
3. We will design for the minimum ripple parameter, which for this example is set to $\delta = 0.01$
4. We will also design for the minimum transition bandwidth, which for this example is

$$\Delta\omega = \min(0.1\pi, 0.08\pi) = 0.08\pi$$

5. Now the Kaiser window parameters are: $\beta = 3.3.953$ and $M = 56 \Rightarrow \alpha = 28$
6. Compute the impulse response $h(n) = h_{\mathrm{d}}(n) w_{\mathrm{K}}(n)$

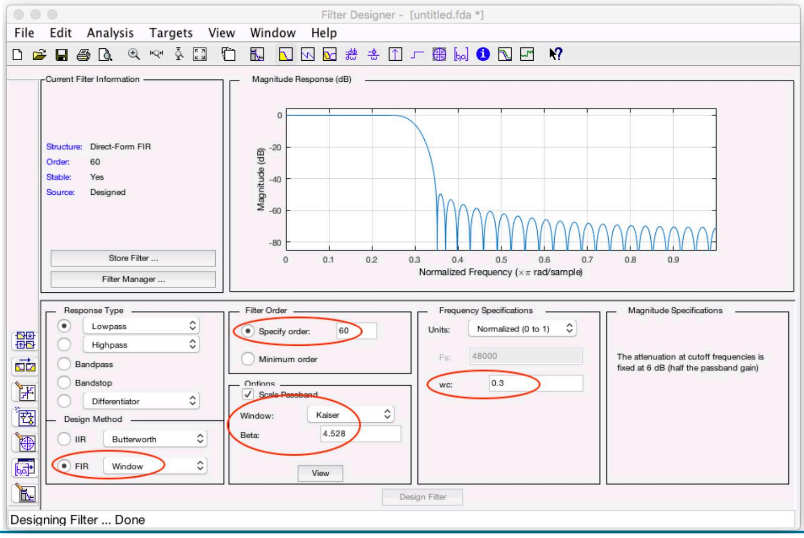
30

Bandpass Filter Design Example: Kaiser Window



31

Filter Designer App



MATLAB Window-Based FIR Filter Design

Syntax

```
b = fir1(M,Wn)
b = fir1(M,Wn,'ftype')
b = fir1(M,Wn,window)
b = fir1(M,Wn,'ftype',window)
b = fir1(...,'normalization')
[M,Wn,BTA,filtype] = kaiserord(f,A,dev,Fs)
```

```
ftype:
'low'
'high'
'bandpass'
'stop'
```

Description

`fir1` implements the classical method of windowed linear-phase FIR digital filter design. It designs filters in standard lowpass, highpass, bandpass, and bandstop configurations. By default, the filter is normalized so that the magnitude response of the filter at the center frequency of the passband is 0 dB.

`kaiserord` determines Kaiser window parameters

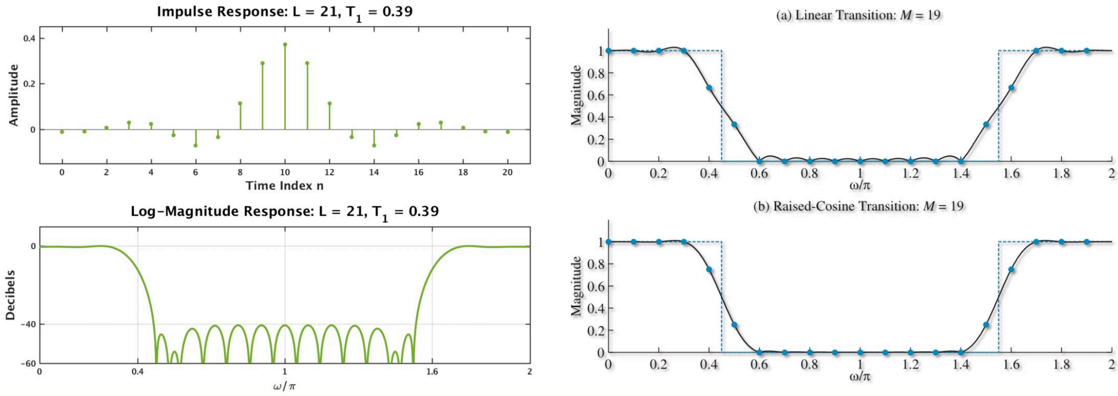
```
normalization:
'scale' (default)
'noscale'
```

Example

```
h = fir1(30,2*0.2,'low') % Note that fc=Fc/Fs
fvtool(h);
```

FIR Filter Design

Frequency Sampling Method



0

Frequency Sampling Design Technique

- The important factor in the FIR window design was the availability of the ideal impulse response in the form of a closed-form mathematical formula.
- In some applications, an ideal frequency response is available, but its inverse may not have a closed-form expression.
- In such situations we begin with
  - the ideal lowpass filter frequency response,
  - sample it at N equispaced frequencies on the unit circle, and
  - use these samples to obtain an FIR filter impulse response.
- The resulting filter frequency response will not be exactly same as the ideal response except at the sampling points.
- We then investigate design approaches that will mitigate some norm of the approximation error.
- We will use the resulting frequency-sampling design techniques to design other frequency selective filters also.

1

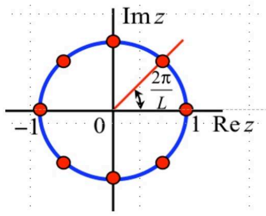
Foundation of Frequency Sampling

- Let us assume that a desired frequency response $H_{\mathrm{d}}(\omega)$ is available but the desired impulse response $h_{\mathrm{d}}(n)$ is not available
- We sample $H\mathrm{d}(\omega)$ at L equispaced points on the unit citcle

$$H_{\mathrm{d}}(k) \triangleq H_{\mathrm{d}}(\omega_k)\big|_{\omega_k=\frac{2\pi k}{L}}, \quad k = 0, 1, 2, \ldots, L-1$$

- From the DFT properties, we note that the IDFT of $H_{\mathrm{d}}(k)$ is related to $h_{\mathrm{d}}(n)$ by the aliasing formula

$$\tilde{h}(n) \triangleq \frac{1}{L}\sum_{k=0}^{L-1} H_{\mathrm{d}}(k) W_L^{-kn} = \sum_{r=-\infty}^{\infty} h_{\mathrm{d}}(n - rL)$$



which is periodic with fundamental period L.
- An L-point FIR filter can now be obtained via L-point windowing operation

$$h(n) = \tilde{h}(n)w(n)$$

or

$$H(\omega) = \frac{1}{L}\sum_{k=0}^{L-1} H_{\mathrm{d}}(k) W\left(\omega - \frac{2\pi k}{L}\right) \qquad \text{(an interpolation)}$$

2

Foundation of Frequency Sampling

$$h(n) = \tilde{h}(n)w(n) \quad \Rightarrow \quad H(\omega) = \frac{1}{L}\sum_{k=0}^{L-1} H_{\mathrm{d}}\left(\omega - \frac{2\pi k}{L}\right)$$

- Thus, the frequency sampling method creates FIR filter by ***aliasing*** while the window design method creates by ***truncating*** ideal filter.
- In this approach, the designed FIR filter may or may not be linear-phase. For linear-phase, we can impose the restriction $h(n) = \pm h(L-1-n)$.
- Also the window function $w(n)$ can be non-rectangular in which case the resulting frequency response will not go through the chosen frequency samples; however, the errors are very small.
- **Approaches**: We will consider the following design techniques
  - Basic (rectangular window design) approach
  - Optimum design approach
  - Smooth transition band approach
  - Non-rectangular window design approach

3

Basic Approach

- For a linear-phase FIR filter of length $L = M - 1$; M : order, we have

$$h(n) = \pm h(L-1-n), \quad n = 0,1,1,\dots,L-1 \quad \Rightarrow \quad H(k) = A(k)\mathrm{e}^{\mathrm{j}\Psi(k)}$$

where
$$A(k) = \begin{cases} A_{\mathrm{d}}\left(\frac{2\pi}{L}(0)\right), & k = 0 \\ A_{\mathrm{d}}\left(\frac{2\pi}{L}(L-k)\right), & k = 1,\dots,L-1 \end{cases}$$

and
$$\Psi(k) = \begin{cases} -\left(\frac{L-1}{2}\right)\left(\frac{2\pi k}{L}\right), & k = 0,\dots,\lfloor\frac{L-1}{2}\rfloor \\ +\left(\frac{L-1}{2}\right)\frac{2\pi}{L}(L-k), & k = \lfloor\frac{L-1}{2}\rfloor + 1,\dots,L-1 \end{cases}$$ Type-I & II

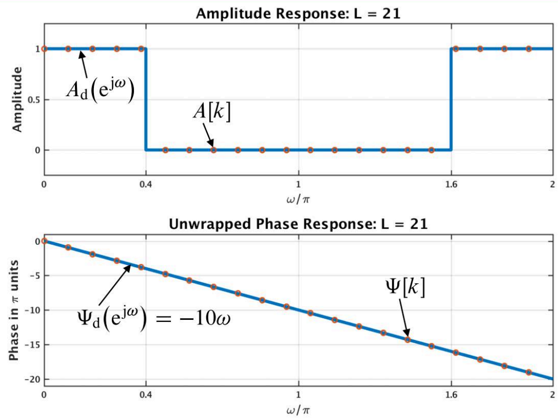
or
$$\Psi(k) = \begin{cases} \pm\frac{\pi}{2} - \left(\frac{L-1}{2}\right)\left(\frac{2\pi k}{L}\right), & k = 0,\dots,\lfloor\frac{L-1}{2}\rfloor \\ \mp\frac{\pi}{2} + \left(\frac{L-1}{2}\right)\frac{2\pi}{L}(L-k), & k = \lfloor\frac{L-1}{2}\rfloor + 1,\dots,L-1 \end{cases}$$ Type-III & IV

- Finally,

$$h(n) = \mathrm{IDFT}\big[H(k)\big]$$

Example of the Basic Approach

- Consider the design of a LP filter from an ideal LPF with $\omega_{\mathrm{c}} = 0.4\pi$.
- Let us choose $L = 21$ so that the cutoff frequency 0.4π has no sample
- Then $\alpha = \frac{L-1}{2} = 10$ and $\Psi_{\mathrm{d}}(\omega) = -10\omega$
- Now we can create amplitude and phase samples



Type-I Linear-Phase Response

Example of the Basic Approach

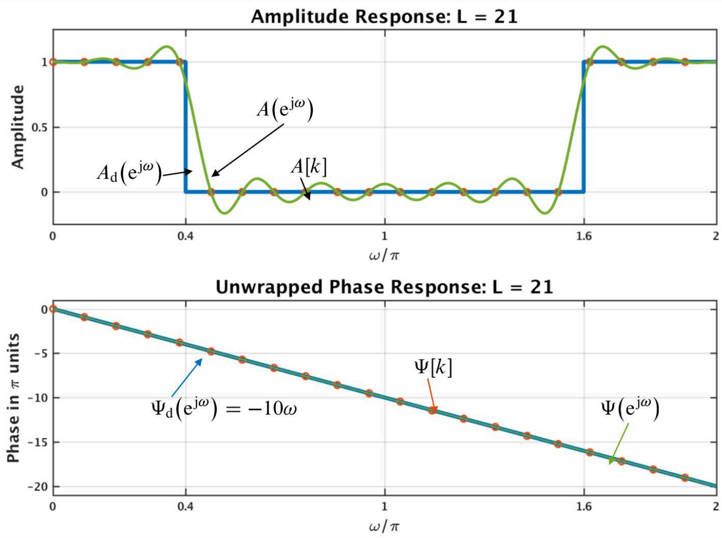
- Using these frequency samples we determine the impulse response and then the frequency response to verify the design.

```
>> L = 21; M = L-1;
>> Dom = 2*pi/L;
>> alpha = M/2;
>> k = 0:M;
>> omk = Dom*k;
>> Ad = [ones(1,5),zeros(1,12),ones(1,4)];
>> Psid = -alpha*Dom*k;
>> Ak = Ad.*exp(1j*Psid);
>> h = real(ifft(Ak,L));
>> om = linspace(0,2,1001)*pi;
>> [A,~,Psi] = zerophase(h,1,om);
```

EECE-5666: DSP FIR Filter Design - Frequency Sampling Method 6/36

6

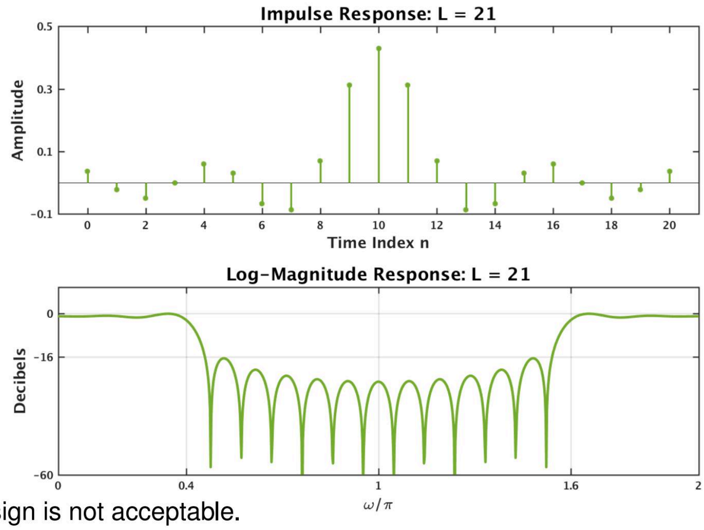
Example of the Basic Approach



EECE-5666: DSP FIR Filter Design - Frequency Sampling Method 7/36

7

Example of the Basic Approach



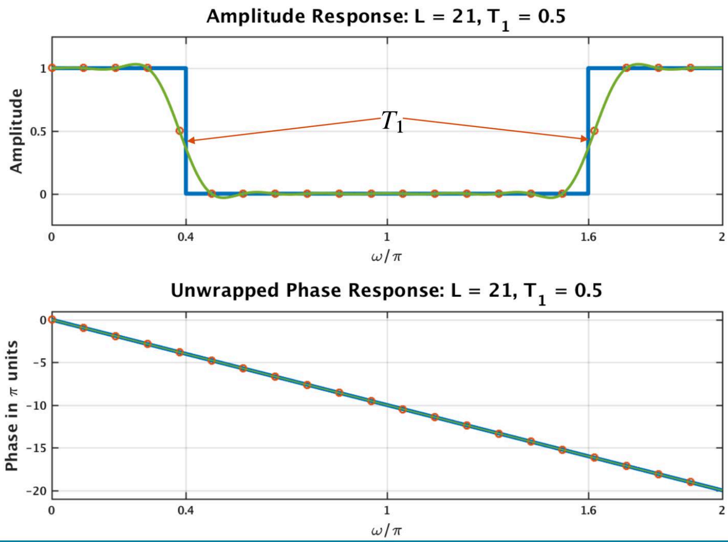
- Clearly this design is not acceptable.

Halfway Transition Band

- For improvement, we let value T_1 of one sample near the cutoff frequency a variable value. Let $T_1 = 0.5$.
- Following the same procedure, we design a new filter.

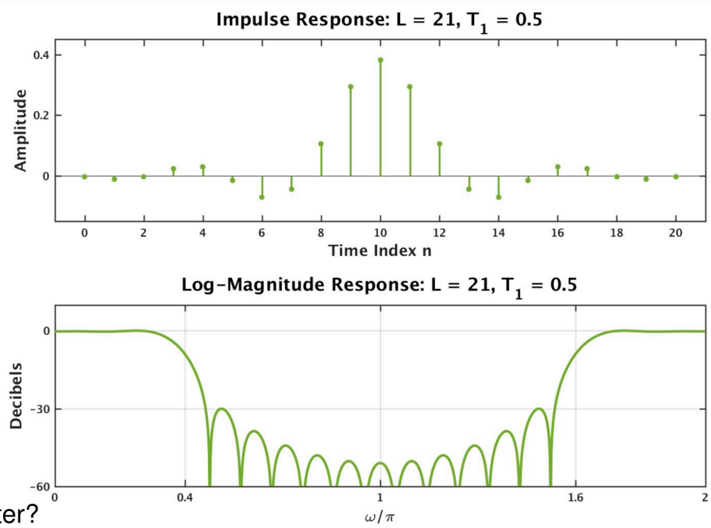
```
>> L = 21; M = L-1; Dom = 2*pi/L;
>> alpha = M/2;
>> k = 0:M; omk = Dom*k;
>> T1 = 0.5;
>> Ad = [ones(1,4),T1,zeros(1,12),T1,ones(1,3)];
>> Psid = -alpha*Dom*k;
>> Ak = Ad.*exp(1j*Psid);
>> h = real(ifft(Ak,L));
>> om = linspace(0,2,1001)*pi;
>> [A,~,Psi] = zerophase(h,1,om);
```

Halfway Transition Band



10

Halfway Transition Band



- Can we do better?

11

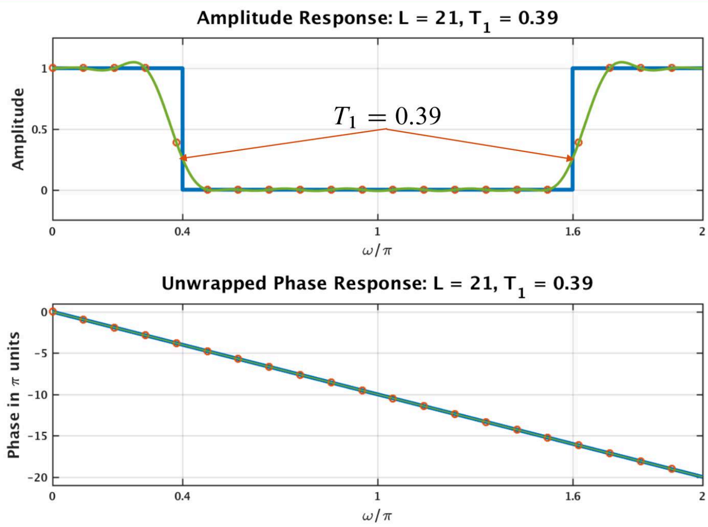
Optimum Design Approach

- By continuously varying T_1 value we obtain the largest possible dB attenuation
- The optimum value of T_1 was found to be 0.39.

```
>> L = 21; M = L-1; Dom = 2*pi/L;
>> alpha = M/2;
>> k = 0:M; omk = Dom*k;
>> T1 = 0.39;
>> Ad = [ones(1,4),T1,zeros(1,12),T1,ones(1,3)];
>> Psid = -alpha*Dom*k;
>> Ak = Ad.*exp(1j*Psid);
>> h = real(ifft(Ak,L));
>> om = linspace(0,2,1001)*pi;
>> [A,~,Psi] = zerophase(h,1,om);
```

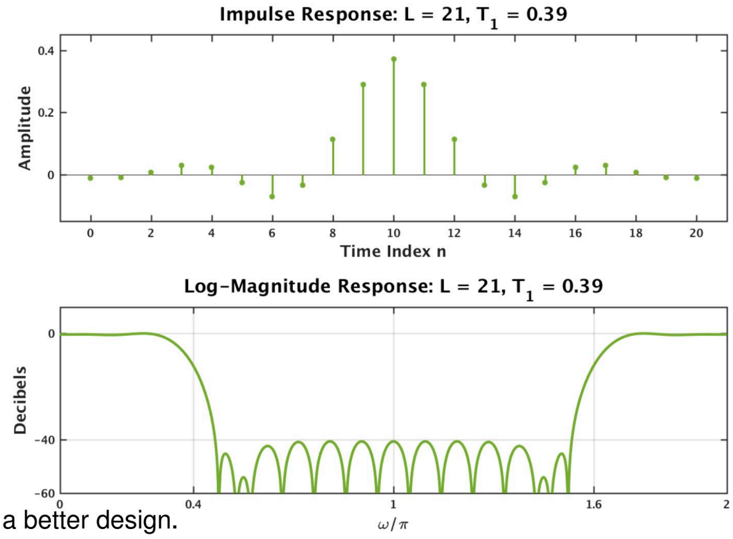

12

Optimum Design Approach



13

Optimum Design Approach



- Clearly this is a better design.

14

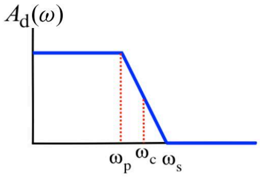
Alternative Approaches

- **Optimum Approach:** There are two issues.
  1. With one variable sample in the transition band, the minimum achievable stopband attenuation is around 40 dB.
  2. To achieve more than 40 dB attenuation, we need to vary more than two samples, which is very cumbersome.
- **Suboptimum Approaches:** These approaches are esy to implement and work very well in practice.
  - Smooth transition band approach: In this approach, we specify the shape of the transition band response.
    - Linear Transition band
    - Raised-cosine transition band
  - Nonrectangular window design approach: In this case, we multiply the impulse response in the basic approach by previously discussed window functions. Also used in MATLAB

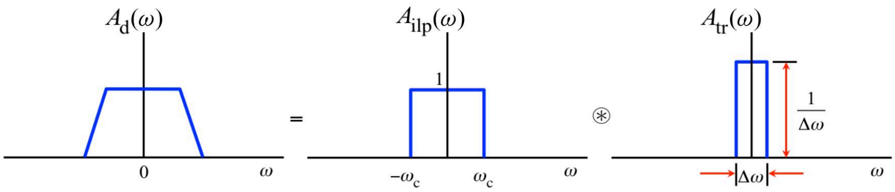
15

Smooth Transition Band Approaches

- Linear Transition band



$$A_d(\omega) = \begin{cases} 1, & 0 \le |\omega| \le \omega_p \\ \dfrac{\omega_s - \omega}{\omega_s - \omega_p}, & \omega_p < |\omega| < \omega_s \;; \\ 0, & \omega_s \le |\omega| \le \pi \end{cases} \quad \Delta\omega = \omega_s - \omega_p, \quad \omega_c = \frac{\omega_p + \omega_s}{2}$$

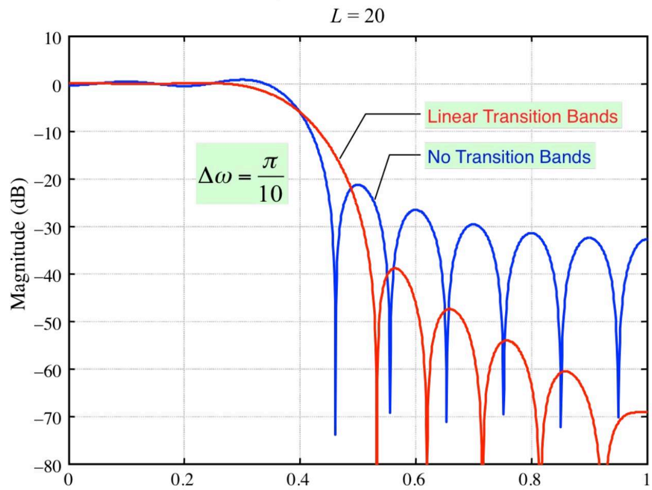


$$\Psi_d(\omega) = 0 \quad \Rightarrow \quad h_d(n) = 2\pi \left(\frac{\sin(\omega_c n)}{\pi n}\right)\left(\frac{\sin(n\Delta\omega/2)}{\pi n \Delta\omega}\right)$$

16

Smooth Transition Band Approaches

- **Linear Transition band**: Example



17

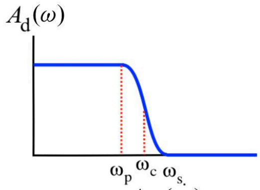
Smooth Transition Band Approaches

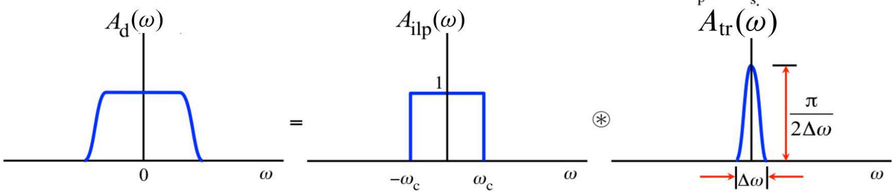
- Further smoothing

$$A_{\mathrm{d}}(\omega) = \left[A_{\mathrm{ilp}}(\omega) \circledast A_{\mathrm{tr}}(\omega)\right] \circledast A_{\mathrm{tr}}(\omega) \quad \Rightarrow \quad \text{Second-order spline}$$

 – Replace linear with raised-cosine roll-off

$$\frac{\omega_{\mathrm{s}} - \omega}{\omega_{\mathrm{s}} - \omega_{\mathrm{p}}} \Rightarrow 0.5 - 0.5\cos\left(\pi \frac{\omega_{\mathrm{s}} - \omega}{\omega_{\mathrm{s}} - \omega_{\mathrm{p}}}\right)$$



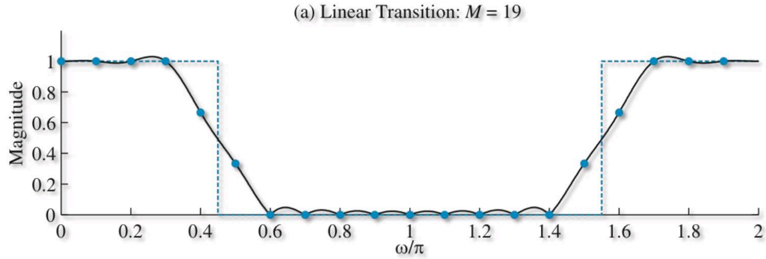


 – Straight-line or raised-cosine roll-off sufficient for many applications

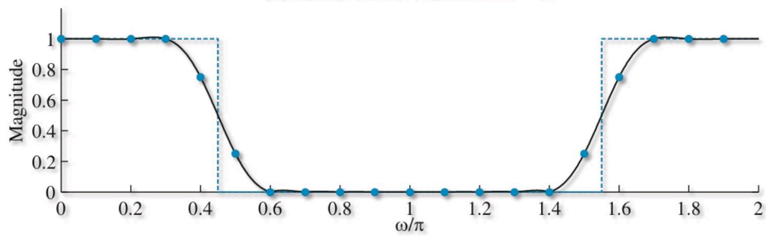
18

Smooth Transition Band Approaches

(a) Linear Transition: $M = 19$



(b) Raised-Cosine Transition: $M = 19$

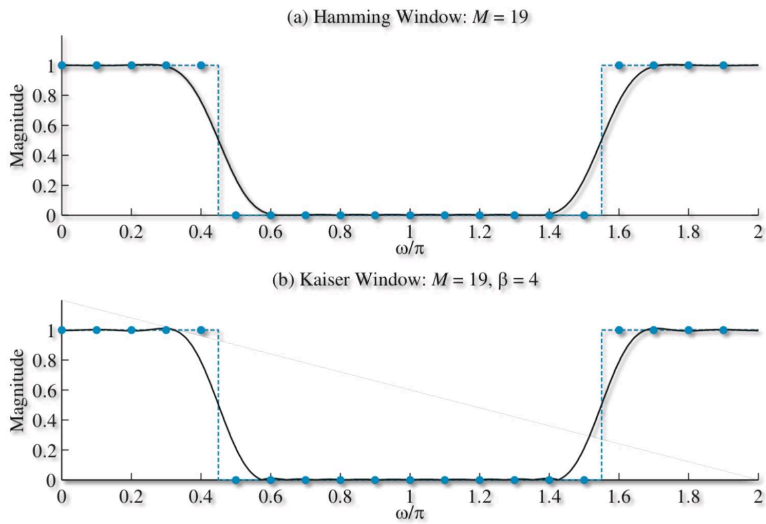


19

Non-Rectangular Window Design Approach

- In this approach, given the samples of the ideal response around the unit circle, we sample $H_{\mathrm{d}}(\omega)$ at L equispaced points on the unit citcle
$$H_{\mathrm{d}}(k) \triangleq A_{\mathrm{d}}(\omega_k)e^{j\Psi_{\mathrm{d}}(\omega_k)}\Big|_{\omega_k=\frac{2\pi k}{L}}, \quad k = 0, 1, 2, \ldots, L-1$$
- We perform the IDFT of $H_{\mathrm{d}}(k)$ to obtain
$$\tilde{h}(n) \triangleq \frac{1}{L}\sum_{k=0}^{L-1} H_{\mathrm{d}}(k)W_L^{-kn} = \sum_{r=-\infty}^{\infty} h_{\mathrm{d}}(n - rL)$$
which is periodic with fundamental period L.
- An L-point FIR filter can now be obtained via L-point windowing operation
$$h(n) = \tilde{h}(n)w_{\mathrm{NR}}(n)$$
where $w_{\mathrm{NR}}(n)$ is a non-rectangular window.
- **Drawback:** The resulting frequency response $H(\omega)$ may not pass through the frequency samples $H_{\mathrm{d}}(k)$, especially near band edges.

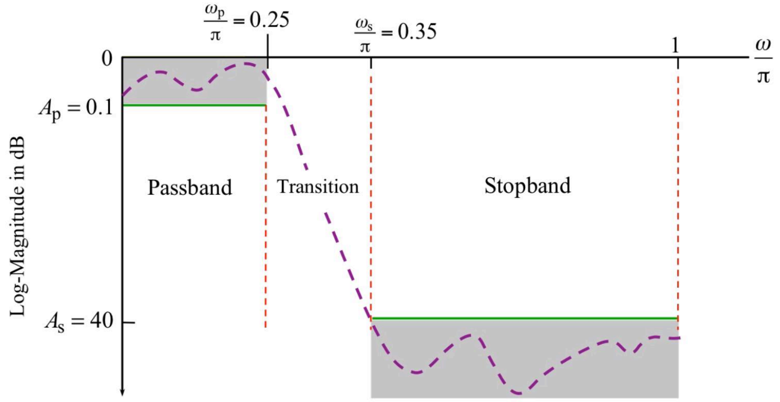
Non-Rectangular Window Design Approach



Lowpass Filter Design Example

Consider the following specifications of a lowpass filter:

$$\omega_p = 0.25\pi, \quad \omega_s 0.35\pi, \quad A_p = 0.1\,\text{dB}, \quad A_s = 40\,\text{dB}$$



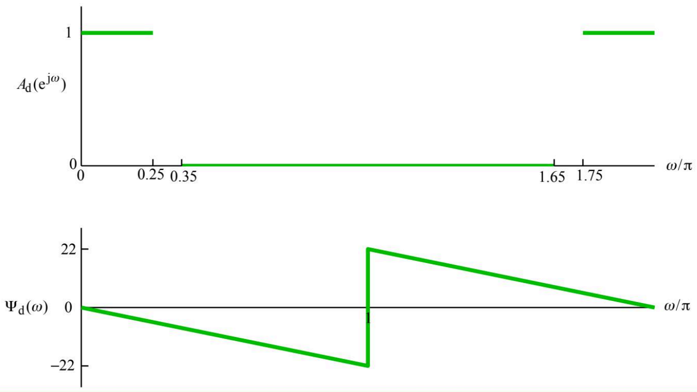
EECE-5666: DSP FIR Filter Design - Frequency Sampling Method 22/36

22

Lowpass Filter Design Example

$$\omega_p = 0.25\pi, \quad \omega_s 0.35\pi, \quad A_p = 0.1\,\text{dB}, \quad A_s = 40\,\text{dB}$$

- We should have two or more samples in the transition band for a good response. Hence we choose $M = 44$ or $L = 45$.



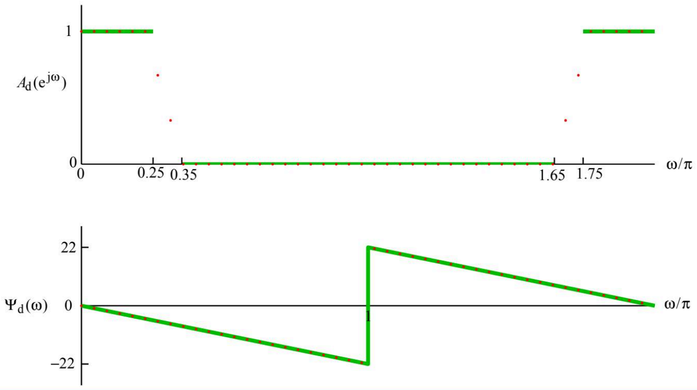
EECE-5666: DSP FIR Filter Design - Frequency Sampling Method 23/36

23

Lowpass Filter Design Example

$$\omega_p = 0.25\pi, \quad \omega_s 0.35\pi, \quad A_p = 0.1\,\text{dB}, \quad A_s = 40\,\text{dB}$$

- We should have two or more samples in the transition band for a good response. Hence we choose $M = 44$ or $L = 45$.



24

Lowpass Filter Design Example

$$\omega_p = 0.25\pi, \quad \omega_s 0.35\pi, \quad A_p = 0.1\,\text{dB}, \quad A_s = 40\,\text{dB}$$

- Raised-cosine design:

```
% Design Parameters
>> M = 44; L = M+1; % Impulse response length
>> alpha = M/2; Q = floor(alpha); % phase delay parameters
>> om = linspace(0,2*pi,1001); % Frequency array
>> k = 0:M; % Frequency sample index
>> psid = -alpha*2*pi/L*[(0:Q),-(L-(Q+1:M))]; % Desired Phase
>> Dw = 2*pi/L; % Width between frequency samples
% Design
>> k1 = floor(wp/Dw); % Index for sample nearest to PB edge
>> k2 = ceil(ws/Dw);  % Index for sample nearest to SB edge
>> w = ((k2-1):-1:(k1+1))*Dw; % Freq in the transition band
>> A = 0.5+0.5*cos(pi*(ws-w)/(ws-wp)); % Transition band samples
>> B = fliplr(A); % Transition band samples for omega >pi
>> Ad = [ones(1,k1+1),A,zeros(1,L-2*k2+1),B,ones(1,k1)]; % Desired Ampl
>> Hd = Ad.*exp(1j*psid); % Desired Freq Resp Samples
>> hd = real(ifft(Hd)); % Desired Impulse response
>> h = hd.*rectwin(L)'; % Actual Impulse response
>> H = freqz(h,1,om); % Frequency response of the actual filter
```


25

Lowpass Filter Design Example

$\omega_p = 0.25\pi, \quad \omega_s 0.35\pi, \quad A_p = 0.1\,\text{dB}, \quad A_s = 40\,\text{dB}$

- The minimum stopband attenuation is

```
>> maxmag = max(abs(H)); dw = 2*pi/1000;
>> Asd = min(-20*log10(abs(H(ceil(ws/dw):501))/maxmag))
Asd = 33.1507
```

Certainly not enough!

- Hamming window design:

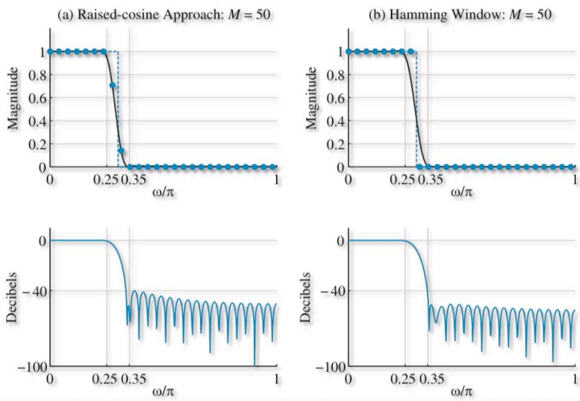
```
>> M = 44; L = M+1; % Impulse response length
>> alpha = M/2; Q = floor(alpha); % phase delay parameters
>> om = linspace(0,2*pi,1001); % Frequency array
>> k = 0:M; % Frequency sample index
>> psid = -alpha*2*pi/L*[(0:Q),-(L-(Q+1:M))]; % Desired Phase
>> Dw = 2*pi/L; % Width between frequency samples
% Design
>> omc = (wp+ws)/2; % Cutoff frequency
>> k1 = floor(omc/Dw); % Left sample index nearest to cutoff edge
>> k2 = ceil(omc/Dw);  % Right sample index nearest to cutoff edge
>> Ad = [ones(1,k1+1),zeros(1,L-2*k2+1),ones(1,k1)]; % Desired Amplitude
>> Hd = Ad.*exp(1j*psid); % Desired Freq Resp Samples
>> hd = real(ifft(Hd));   % Desired Impulse response
>> h = hd.*hamming(L)';   % Hamming window Impulse response
>> H = freqz(h,1,om);     % Frequency response of the actual filter
```

Resulting minimum stopband attenuation: 37.4 dB. Again not enough!

Lowpass Filter Design Example

$\omega_p = 0.25\pi, \quad \omega_s 0.35\pi, \quad A_p = 0.1\,\text{dB}, \quad A_s = 40\,\text{dB}$

- The required stopband attenuation of 40 dB was obtained by increasing the filter order to $M = 50$ or length $L = 51$.



MATLAB Functions in SP Toolbox

- Matlab does not provide classic (that is, the basic approach that we discussed) frequency sampling design functions
- It however does provide the frequency sampling approach based on non-rectangular windows in the function `fir2`.
- This approach uses the basic naïve frequency sampling approach followed by windowing of the resulting impulse response to minimize the Gibbs phenomenon.
- This then avoids the issue of choosing optimum samples in the transition bands.
- A side effect of this windowing approach is that the resulting frequency response does not go through the original frequency samples.
- This however is not a serious problem because the response is close to the actual samples for large M and the stopband attenuation is the main issue.
- The `fdatool` GUI function does not provide any options for the frequency sampling design method.

MATLAB Functions in SP Toolbox

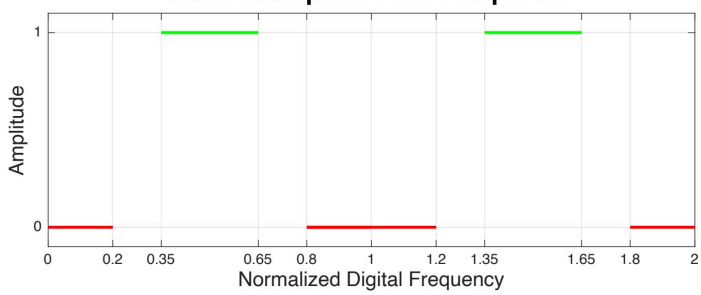
- `h = fir2(M,f,A)` designs an Mth-order $(M = L - 1)$ lowpass FIR filter and returns the impulse response in vector `h`.
  - The desired magnitude response of the filter is supplied in vectors `f` and `A`, which must be of the same length.
  - The vector `f` contains normalized frequencies in the range from 0 to 1, where 1 corresponds to π rad/sample.
  - The first value of `f` must be 0 and the last value 1.
  - The vector `A`, contains the samples of the desired Amplitude response at the values specified in `f`.
  - The desired frequency response is then interpolated onto a dense, evenly spaced grid of length 512.
  - This syntax uses the default `Hamming window`.
- `h = fir2(M,f,A,window)` uses the vector window of length $L = M + 1$ obtained from one of the specified Matlab window function.
- `h = fir2(M,f,A,npt)` or `h = fir2(M,f,A,npt,window)` specifies the number of points, `npt`, for the grid onto which `fir2` interpolates the freq. resp.

Bandpass Filter Design Example

Design a FIR bandpass filter that satisfies:

Lower stopband edge:	$\omega_{s_1} = 0.2\pi,$	$A_s = 60$ dB
Lower passband edge:	$\omega_{p_1} = 0.35\pi,$	$A_p = 1$ dB
Upper passband edge:	$\omega_{p_2} = 0.65\pi,$	$A_p = 1$ dB
Upper stopband edge:	$\omega_{s_2} = 0.8\pi,$	$A_s = 40$ dB

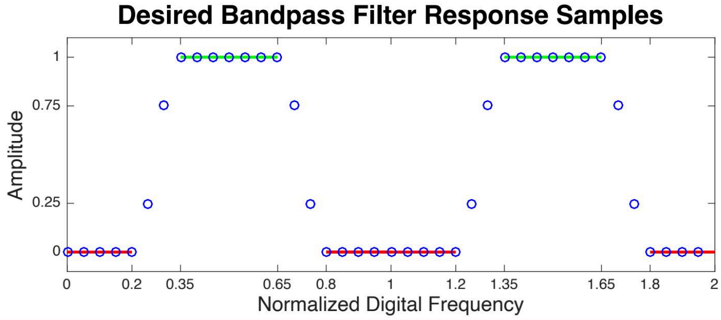
Ideal Bandpass Filter Response



Bandpass Filter Design Example

- We will use the raised cosine samples in the transition band.
- Let us choose $L = 40$ samples on the unit circle. Hence the length of this filter is $L = 40$ or the order is $M = L - 1 = 39$. So this is Type-2 LP FIR filter.
- The samples of the amplitude response are shown below.
- Note that the phase response is linear, i.e., $\Psi(\omega) = -\frac{39}{2}\omega = 19.5\omega$.

Desired Bandpass Filter Response Samples

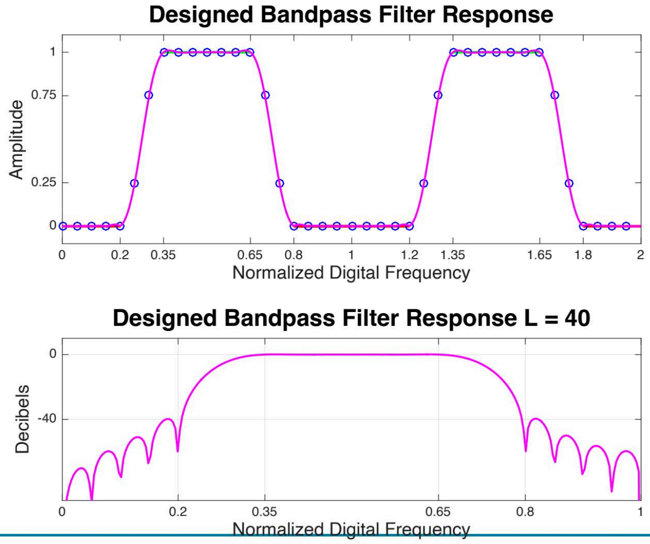


Bandpass Filter Design Example

```
fs1 = 0.2; fp1 = 0.35; fp2 = 0.65; fs2 = 0.8;
L = 40; % This will result in 2 samples in the transition band
Delf = 2/L; % Normalized frequency resolution
ks1 = floor(fs1/Delf)+1; % Index of the lower stopband edge
kp1 = ceil(fp1/Delf)+1; % Index of the lower stopband edge
kp2 = floor(fp2/Delf)+1; % Index of the lower stopband edge
ks2 = ceil(fs2/Delf)+1; % Index of the lower stopband edge
kltrb = (ks1+1:kp1-1); % Lower transition band sample indexes
kutrb = (kp2+1:ks2-1); % upper transition band sample indexes
Altrb = 0.5-0.5*cos((fs1-(kltrb-1)*Delf)./(fs1-fp1)*pi);
Autrb = 0.5-0.5*cos((fs2-(kutrb-1)*Delf)./(fs2-fp2)*pi);
A1 = [zeros(1,ks1),Altrb,ones(1,kp2-kp1+1),Autrb,zeros(1,L/2-ks2+2)];
A = [A1(1:end-1),fliplr(A1(2:end))];
alpha = (L-1)/2; Psi = (-alpha*(-L/2:L/2-1)*Delf*pi);
Psi = fftshift(Psi);
Hd = A.*exp(1j*Psi);
h = real(ifft(Hd));
fvtool(h);
```


32

Bandpass Filter Design Example

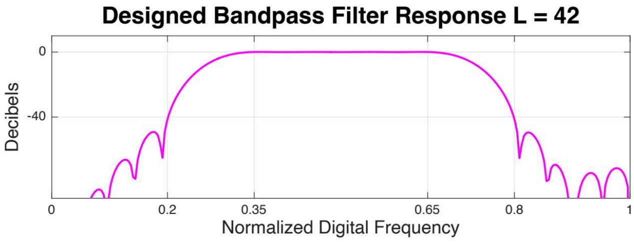


Almost there

33

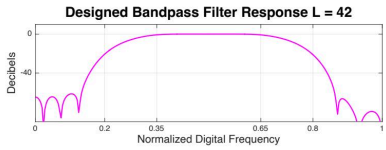
Bandpass Filter Design Example

- So the length was increased by two; $L = 42$ or $M = 41$ which is also a Type-II FIR linear-phase filter.



- Use of the `fir2` function:

```
L2 = 42; M2 = L2-1;
f2 = [0,fs1,fp1,fp2,fs2,1];
A2 = [0,0,1,1,0,0];
win = hamming(L2);
h2 = fir2(M2,f2,A2,win);
```



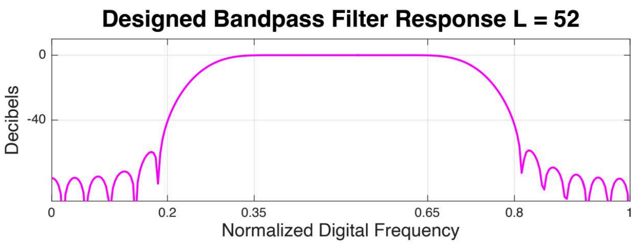
Clearly unacceptable

34

Bandpass Filter Design Example

- We need to increase the filter length L and band-edge frequencies:

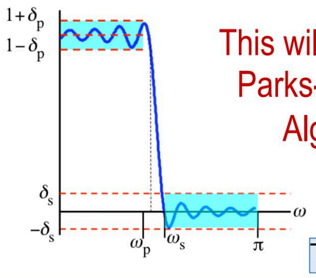
```
L2 = 52; M2 = L2-1;
f2 = [0,fs1+0.04,fp1-0.04,fp2+0.04,fs2-0.04,1];
A2 = [0,0,1,1,0,0];
win = hamming(L2);
h2 = fir2(M2,f2,A2,win);
```



35

Key Features of Window/FS Methods

- Bandedge frequencies ω_p and ω_s cannot be assigned **precisely**
- Band tolerances (ripples) are **always equal**: $\delta_p = \delta_s$
- The approximation error is **non-uniformly** distributed
- The mean-squared aproximation error $\varepsilon_M^2 \triangleq \frac{1}{2\pi}\int_{2\pi} |H(_M(\omega) - H_d(\omega)|^2 \, d\omega$ is minimum only for the rectangualr window



This will lead to the Parks-McClellan Algorithm

Question Can we distribute the approximation error uniformly (equiripple approximation) and obtain a "better" filter with the same order M or a filter withe same quaility but smaller M?

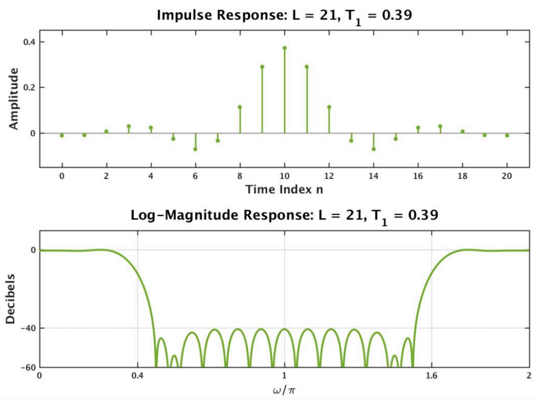
Answer YES! Use numerical optimization techniques

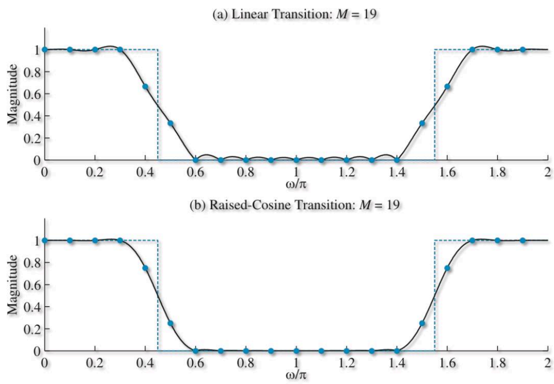
The important quantity is the **maximum** absolute error and not ε_M^2

36

FIR Filter Design

Frequency Sampling Method

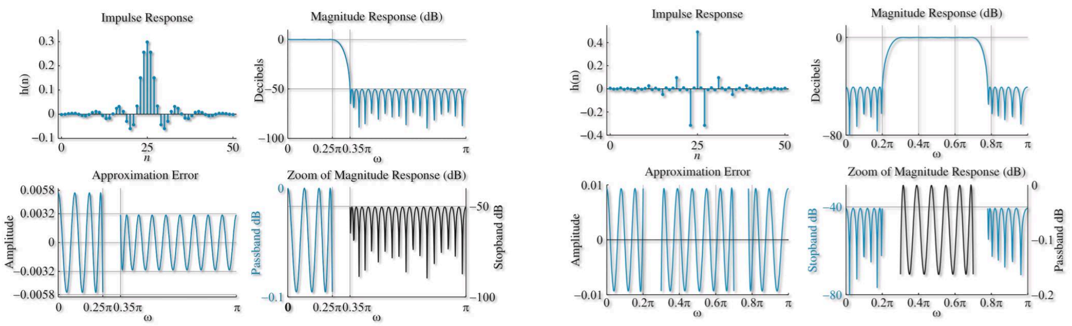




37

FIR Filter Design

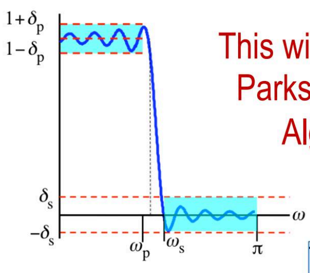
Parks-McClellan Algorithm



0

Key Features of Window/FS Methods

- Bandedge frequencies ω_p and ω_s cannot be assigned **precisely**
- Band tolerances (ripples) are **always equal**: $\delta_p = \delta_s$
- The approximation error is **non-uniformly** distributed
- The mean-squared aproximation error $E^2(\omega) \triangleq \frac{1}{2\pi}\int_{2\pi} |H_M((\omega) - H_d(\omega)|^2 \, d\omega$ is minimum only for the rectangualr window



This will lead to the Parks-McClellan Algorithm

Question Can we distribute the approximation error uniformly (equiripple approximation) and obtain a "better" filter with the same order M or a filter withe same quaility but smaller M?

Answer YES! Use numerical optimization techniques

The important quantity is the **maximum** absolute error and not ε_M^2

EECE-5666: DSP FIR Filter Design - PM Algorithm 1/30

1

Approximation Error Criteria

- $H_d(\omega)$: Desired frequency response
- $H(\omega) = \sum_{n=0}^{M} h(n)e^{-j\omega n}$: Approximated (or designed) frequency response, which is a trigonometric polynomial for linear-phase FIR filters. For example, let $h(n) = \{\underset{\uparrow}{1}, 2, 3, 2, 1\}$, which is a Type-1 FIR filter. Then

$$H(\omega) = 1 + 2e^{-j\omega} + 3e^{-j2\omega} + 2e^{-j3\omega} + e^{-j4\omega} = e^{-j2\omega}\underbrace{\big(3 + 4\cos(\omega) + 2\cos(2\omega)\big)}_{A(\omega)\text{: Trig Poly}}$$

- $E(\omega) = H_d(\omega) - H(\omega)$: Approximation error over $\mathcal{B}$, a union of desired bands.
- $E_2 \triangleq \sqrt{\frac{1}{2\pi}\int_{\mathcal{B}} \big|E(\omega)\big|^2 d\omega}$: Mean Squared Error (MSE) or L_2 norm (minimizes error energy)
- $E_\infty \triangleq \max_{\omega\in\mathcal{B}} |E(\omega)|$: Chebyshev or L_∞ or uniform norm (minimize maximum error)

We will focus on this error

Frequency Responses of LP-FIR Revisited

Type	$h[k]$	M	$A(e^{j\omega})$	$A(e^{j\omega})$	$\Psi(e^{j\omega})$
I	even	even	$\sum_{k=0}^{M/2} a[k]\cos\omega k$	even–no restriction	$-\frac{\omega M}{2}$
II	even	odd	$\sum_{k=1}^{\frac{M+1}{2}} b[k]\cos\left[\omega\left(k-\frac{1}{2}\right)\right]$	even $A(e^{j\pi}) = 0$	$-\frac{\omega M}{2}$
III	odd	even	$\sum_{k=1}^{M/2} c[k]\sin\omega k$	odd $A(e^{j0}) = 0$ $A(e^{j\pi}) = 0$	$\frac{\pi}{2} - \frac{\omega M}{2}$
IV	odd	odd	$\sum_{k=1}^{\frac{M+1}{2}} d[k]\sin\left[\omega\left(k-\frac{1}{2}\right)\right]$	odd $A(e^{j0}) = 0$	$\frac{\pi}{2} - \frac{\omega M}{2}$

Frequency Responses of LP-FIR Revisited

Fact Using simple trigonometric identities, we can express $A(\omega)$ as a product of a *fixed function* of ω, called $Q(\omega)$ and *function of cosines*, called $P(\omega)$.

Example Let $h(n) = [2, 3, 3, 2]$ which is a Type-2 LP-FIR filter with order $M = 3$. Then

$$\begin{aligned} H(\omega) &= \sum_{n=0}^{M} h(n)\mathrm{e}^{-\mathrm{j}\omega n} = 2 + 3\mathrm{e}^{-\mathrm{j}\omega} + 3\mathrm{e}^{-\mathrm{j}2\omega} + 2\mathrm{e}^{-\mathrm{j}3\omega} \\ &= \mathrm{e}^{-\mathrm{j}3\omega/2}\left(2\mathrm{e}^{\mathrm{j}3\frac{\omega}{2}} + 3\mathrm{e}^{\mathrm{j}\frac{\omega}{2}} + 3\mathrm{e}^{-\mathrm{j}\frac{\omega}{2}} + 3\mathrm{e}^{-\mathrm{j}\omega} + 2\mathrm{e}^{-\mathrm{j}3\frac{\omega}{2}}\right) \\ &= \mathrm{e}^{-\mathrm{j}3\omega/2}\underbrace{\left[6\cos\left(\tfrac{\omega}{2}\right) + 4\cos\left(3\tfrac{\omega}{2}\right)\right]}_{A(\omega)=\sum_{n=1}^{2} b(n)\cos\left((2n-1)\frac{\omega}{2}\right)} \end{aligned}$$

Using $\cos\left(3\frac{\omega}{2}\right) = 2\cos\left(\frac{\omega}{2}\right)\cos(\omega) - \cos\left(\frac{\omega}{2}\right) = \cos\left(\frac{\omega}{2}\right)\left[2\cos(\omega) - 1\right]$, we can express

$$\begin{aligned} A(\omega) &= 6\cos\left(\tfrac{\omega}{2}\right) + 4\cos\left(\tfrac{\omega}{2}\right)\left[2\cos(\omega) - 1\right] \\ &= \underbrace{\cos\left(\tfrac{\omega}{2}\right)}_{Q(\omega)}\underbrace{\left[2\cos(0\omega) + 8\cos(1\omega)\right]}_{P(\omega)=\sum_{n=0}^{1}\tilde{b}(n)\cos(\omega n)} \triangleq Q(\omega)P(\omega) \end{aligned}$$

$b(n) = \{6,4\}$
$\tilde{b}(n) = \{2,8\}$

EECE-5666: DSP FIR Filter Design - PM Algorithm 4/30

4

Frequency Responses of LP-FIR Revisited

$$H(\omega) = \sum_{n=0}^{M} h(n)\mathrm{e}^{-\mathrm{j}\omega n} = \mathrm{e}^{\mathrm{j}(\beta-\alpha\omega)}A(\omega) = \mathrm{e}^{\mathrm{j}(\beta-\alpha\omega)}Q(\omega)P(\omega)$$

Table 10.2 Unified representation and uses of FIR filters with linear phase.

Type	M	$Q(\mathrm{e}^{\mathrm{j}\omega})$	$P(\mathrm{e}^{\mathrm{j}\omega})$	$H(\mathrm{e}^{\mathrm{j}\omega}) = 0$	Uses
I	even	1	$\sum_{k=0}^{M/2}\tilde{a}[k]\cos\omega k$		LP, HP, BP, BS, multiband filters
II	odd	$\cos(\omega/2)$	$\sum_{k=0}^{\frac{M-1}{2}}\tilde{b}[k]\cos\omega k$	$\omega = \pi$	LP, BP
III	even	$\sin\omega$	$\sum_{k=0}^{M/2}\tilde{c}[k]\cos\omega k$	$\omega = 0, \pi$	differentiators, Hilbert transformers
IV	odd	$\sin(\omega/2)$	$\sum_{k=0}^{\frac{M-1}{2}}\tilde{d}[k]\cos\omega k$	$\omega = 0$	differentiators, Hilbert transformers

$P(\omega) = \sum_0^R p(n)\cos(\omega n)$ can be expanded in terms of Chebyshev polynomials

EECE-5666: DSP FIR Filter Design - PM Algorithm 5/30

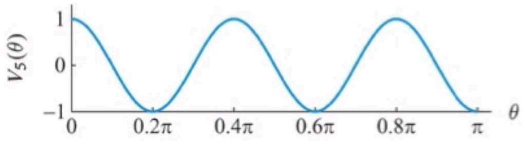
5

Chebyshev Polynomials

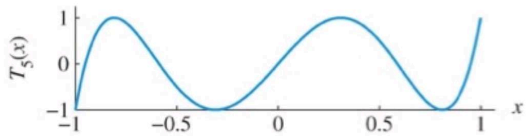
Chebyshev polynomials are "cosines in disguise"

$x = \cos\theta \in [-1,1]$

$V_m(\theta) = \cos(m\theta), \qquad \theta \in [0,\pi]$



$T_m(x) = \cos[m\cos^{-1}(x)], \qquad x \in [-1,1]$



Using the trigonometric formula

$$\cos(n+1)\theta + \cos(n-1)\theta = 2\cos\theta\cos n\theta \Rightarrow$$

$$\cos\theta = x$$

$$\cos 2\theta = 2\cos^2\theta - 1$$

$$\cos 3\theta = 2\cos\theta\cos 2\theta - \cos\theta$$
$$= 4\cos^3\theta - 3\cos\theta$$

$$\cos 4\theta = 2\cos\theta\cos 3\theta - \cos 2\theta$$
$$= 8\cos^4\theta - 8\cos^2\theta + 1$$

$T_0(x) = 1$

$T_1(x) = x \qquad T_3(x) = 4x^3 - 3x$

$T_2(x) = 2x^2 - 1 \qquad T_4(x) = 8x^4 - 8x^2 + 1$

$$\sum_{k=0}^{L} a_k \cos k\theta = \sum_{k=0}^{L} c_k T_k(\cos\theta)$$

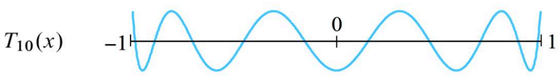
6

Properties of Chebyshev Polynomials

The leading coefficient of $T_m(x)$ is $2^{m-1},\ m \geq 1$

Symmetry property: $T_m(-x) = (-1)^m T_m(x)$

$T_m(x)$ has m zeros and $m+1$ extrema in $[-1,1]$



$T_m(x)$ and $T_n(x)$ are orthogonal in $[-1,1]$

Chebyshev's theorem on Minimax Approximation Optimality

Of all polynomials of degree m with coefficient of x^m equal to 1, the normalized Chebyshev polynomial $2^{-(m-1)}T_m(x)$ has the least maximum amplitude on the interval $[-1,1]$

Why Chebyshev polynomials?

$$A(\omega) = 2h(0)\cos 2\omega + h(1)\cos\omega + h(2)$$
$$= a_0 + a_1\cos\omega + a_2\cos 2\omega$$

$$A(\omega) = [a_0T_0(x) + a_1T_1(x) + a_2T_2(x)]\Big|_{x=\cos\omega}$$

- This allows us to express the amplitude response $A(\omega)$ of FIR filters with linear phase in terms of Chebyshev polynomials.
- This relationship establishes the connection between Chebyshev polynomial approximation and filter design.

7

Design of Optimum Equiripple FIR Filters

The amplitude response of Type I-IV FIR filters is

$$A(\omega) = \sum_{k=0}^{L} a_k \cos \omega k = \sum_{k=0}^{L} c_k T_k(\cos \omega)$$

This makes possible the use of min-max optimization to optimize the criterion

$$\min_{c_k} \left[\max_{\omega \in \mathcal{B}} |E(\omega)| \right]$$

where

$$E(\omega) \triangleq W(\omega)\{A_d(\omega) - A(\omega)\}$$

The weighting function $W(\omega)$ compensates for $\delta_p \neq \delta_s$ by scaling bands accordingly.

This min-max problem can be solved using the Remez exchange algorithm

The Remez exchange algorithm exploits the famous Alteration Theorem, which gives the necessary and sufficient condition for an optimal real Chebyshev solution as one that has at least $M + 1$ alternating external points (i.e., points where the error is maximal).

The design of optimum equiripple FIR filters using the Remez algorithm and FORTRAN code for its implementation was introduced by Parks-McClellan (1973).

Alternation Theorem

- Let $\mathcal{B}$ be a union of closed subsets (in our case $\mathcal{B} = [0, \omega_p] \cup [\omega_s, \pi]$).
- For a positive weighting function $W(\omega)$, a necessary and sufficient condition for

$$A(\omega) = \sum_{k=0}^{R} a(k) \cos(k\omega)$$

to be the unique function that minimizes the maximum value of the weighted error $|E(\omega)|$ over the set $\mathcal{B}$ is that the $E(\omega)$ have at least $R + 2$ alternations.

- That is, there must be at least $R + 2$ extremal frequencies, $\omega_0 < \omega_1 < \cdots < \omega_{R+1}$ over the set $\mathcal{B}$ such that the maximum error alternates, or

$E(\omega_k) = -E(\omega_{k=1}), k = 0,1, \ldots, R$ and $|E(\omega_k)| = \max_{\omega \in \mathcal{B}} |E(\omega)|, k = 0, \ldots, R + 1$

- Thus, according to the alternation theorem the optimum filter is equiripple.
- The alternation theorem characterizes the best solution but does not provide a way to obtain it.

Alternation Theorem

- From the alternation theorem, it follows that
$$W(\omega)[A_d(\omega) - A(\omega)] = (-1)^k \varepsilon, \qquad k = 0,1,\dots,R+1$$
or
$$A_d(\omega) - A(\omega) = (-1)^k \varepsilon / W(\omega)$$
where $\varepsilon = \pm \max_{\omega\in\mathcal{B}} |E(\omega)|$ is the maximum absolute weighted error.
- These equations can be written in matrix form in terms of the unknowns $\{a(k)\}_0^R$ and ε as
$$\begin{bmatrix} 1 & \cos(\omega_0) & \cdots & \cos(R\omega_0) & 1/W(\omega_0) \\ 1 & \cos(\omega_1) & \cdots & \cos(R\omega_1) & -1/W(\omega_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\omega_R) & \cdots & \cos(R\omega_R) & (-1)^R/W(\omega_R) \\ 1 & \cos(\omega_{R+1}) & \cdots & \cos(R\omega_{R+1}) & (-1)^{R+1}/W(\omega_{R+1}) \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a(R) \\ \varepsilon \end{bmatrix} = \begin{bmatrix} A_d(\omega_0) \\ A_d(\omega_1) \\ \vdots \\ A_d(\omega_R) \\ A_d(\omega_{R+1}) \end{bmatrix}$$
- Knowing the extremal frequencies $\{\omega_k\}_0^{R+1}$, these equations can be solved for $\{a(k)\}_0^R$ and ε.
- Parks and McClellan provided an efficient iterative procedure to obtain these extremal frequencies. This algorithm is now known as the Parks-McClellan (PM) algorithm.

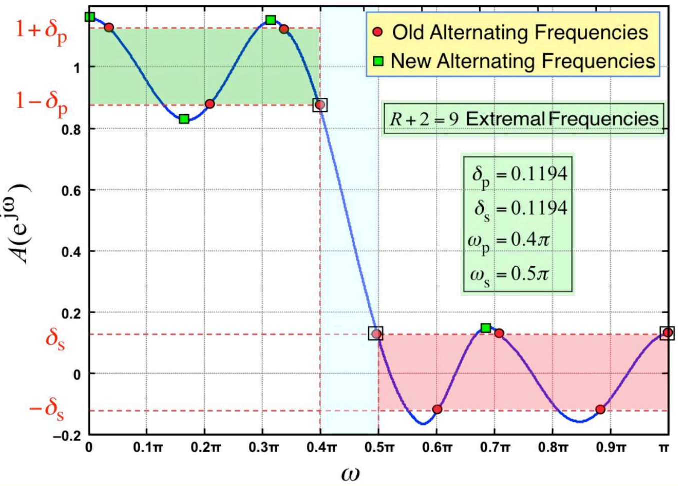
Parks-McClellan Algorithm for Equiripple Filter Design

1. Guess an initial set of extremal frequencies.
2. Solve the matrix equation from the previous slide and obtain the extremal frequencies $\{\omega_k\}_0^{R+1}$ and ε.
3. Using the Lagrange polynomial interpolation formula, interpolate between extremal frequencies and evaluate the weighted error over the set $\mathcal{B}$.
4. Select a new set of extremal frequencies by choosing the $R+2$ frequencies for which the interpolated absolute error function is maximum.
5. If the extremal frequencies have changed then repeat the iteration from step-2.

- The parameter R is related to the order M of the filter for each of the linear-phase filter types I through IV.
- The approximate order calculation is provided by Kaiser using the formula
$$M \approx \frac{-10\log_{10}(\delta_p\delta_s) - 13}{14.6\Delta f}$$

Remez Algorithm Step Before the Optimum

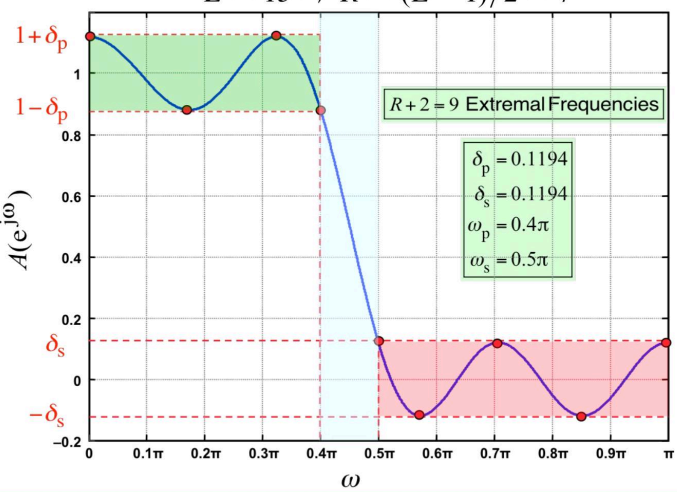
$M = 15 \Rightarrow R = (M-1)/2 = 7$



12

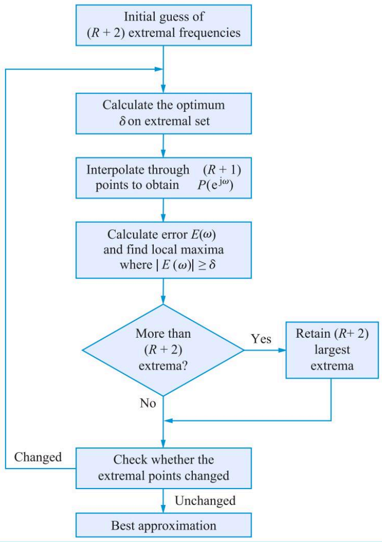
Illustration of Equiripple Property

$L = 15 \Rightarrow R = (L-1)/2 = 7$



13

The Parks-McClellan Algorithm



14

Design Using Parks-McClellan Method

- Fix: $M, \omega_p, \omega_s, \delta_p/\delta_s$ vary $h[0], h[1], ..., h[M]$ systematically so that we obtain an *equiripple* approximation within passband or stopband (Remez)
- *Use* a weighting function to impose the maximum approximation errors to be different in different bands
- It is not possible to specify M in advance $\Rightarrow$ trial and error, i.e., start with

$$M = \frac{-10\log_{10}(\delta_p \delta_s) - 13}{2.324\Delta\omega}, \ \Delta\omega = \omega_s - \omega_p$$

and increase M until the desired filter is obtained. Empirical Formula

Note: For Kaiser $M = \frac{A-8}{2.285\Delta\omega}$, $A = -20\log_{10}\delta$

$\Rightarrow \delta_p, \delta_s$, PMFIR 5 dB better for the same M

PM method provides the best FIR filter design technique for LP, HP, Hilbert Transformers, and differentiators

15

MATLAB Functions

The Parks-McClellan algorithm is available in Matlab as a function called `firpm`:

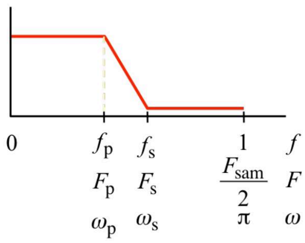
- Basic syntax: `[h,delta] = firpm(M,f,a)` designs an $M = L - 1$ order FIR lowpass digital filter.
  - frequency response is specified by the arrays `f` and `a`.
  - The filter coefficients are returned in array `h` of length L.
  - The achieved maximum ripple δ is returned in `delta`.
  - The array `f` contains band-edge frequencies in units of π.
  - The array `a` contains the desired magnitude response specified in `f`.
  - The weighting function used in each band is equal to unity, which means that the tolerances in every band are the same.
- Weighting function Syntax: `[h,delta] = firpm(M,f,a,W)`
  similar to the preceding case except that the array `W` specifies the weighting function in each band.
- General syntax: `[h,delta] = firpm(M,f,a,W,ftype)`
  - The parameter type is one of `'hilbert'` or `'differentiator'`.

MATLAB Functions

- To estimate the order $M = L - 1$, SP Toolbox in MATLAB provides a function called `firpmord`:
  `[M,f0,a0,W] = firpmord(f,a,tol)`
  - The function computes the window order `M`, the normalized frequency band edges in `f0`, amplitude response in `a0`, and the band weights in `W`.
  - The vector `f` is a vector of normalized band edges and `a` is a vector specifying the desired amplitude on the bands defined by `f`.
  - The length of `f` is two less than twice the length of `a`; i.e., `f` does not contain `0` or `1`.
  - The vector `tol` specifies absolute (that is, not dB) tolerances in each band.
  - The estimated parameters can now be used in the `firpm` function.

Examples of `firpm` parameters

- Lowpass Design



`h=firpm(M,f,a,W)`

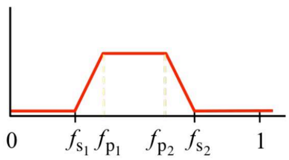
$h = \left[h[0], h[1], \ldots, h[M]\right]$

$$\Rightarrow f = \begin{bmatrix} 0 & f_p & f_s & 1 \end{bmatrix}$$

$$a = [\underbrace{1 \quad 1}_{} \quad \underbrace{0 \quad 0}_{}]$$

$$W = [\ w_p \qquad w_s\ ]$$

- Bandpass Design



$$f = \begin{bmatrix} 0 & f_{s_1} & f_{p_1} & f_{p_2} & f_{s_2} & 1 \end{bmatrix}$$

$$a = [\underbrace{0 \quad 0}_{} \quad \underbrace{1 \quad 1}_{} \quad \underbrace{0 \quad 0}_{}]$$

$$W = [\ w_{s_1} \qquad w_p \qquad w_{s_2}\ ]$$

`h=firpm(M,f,a,W,'bandpass')`

18

Example: Lowpass Filter Design

- Consider the lowpass filter design done earlier using window design with specs

$$\omega_p = 0.25\pi, \qquad \omega_s = 0.35\pi, \qquad A_p = 0.1 \text{ dB}, \qquad A_s = 50 \text{ dB}$$

- First, we need to compute δ_p and δ_s from A_p and A_s:

$$A_p = 20\log\left(\frac{1+\delta_p}{1-\delta_p}\right) = 0.1 \quad \Rightarrow \quad \delta_p = 0.0058;$$

$$A_s = 20\log\left(\frac{1+\delta_p}{\delta_s}\right) = 50 \quad \Rightarrow \quad \delta_s = 0.0032$$

- Next, we determine the approximate filter order using the `firpmord` function
- Then, we determine the optimum FIR filter for the given order using the `firpm` function
- Finally, we check the obtained maximum ripple δ with δ_p, and increase order M (if $\delta > \delta_p$) or decrease M (if $\delta < \delta_p$) and obtain the required optimum FIR equiripple filter.

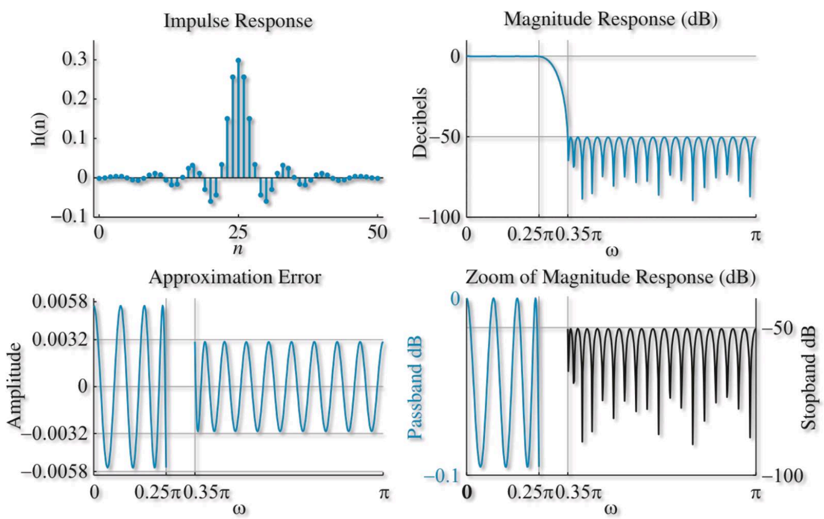
19

Example: Lowpass Filter Design (continued)

```
>> % Given Specifications
>> wp = 0.25*pi; ws = 0.35*pi; Ap = 0.1; As = 50;
>> % Passband and Stopband Ripple Calculations
>> delp = (10^(Ap/20)-1)/(10^(Ap/20)+1);
>> dels = (1+delp)/(10^(As/20));
>> % Estimated Filter order using FIRPMORD function
>> [M,fo,ao,W] = firpmord([wp,ws]/pi,[1,0],[delp,dels]);
>> M
M =
    48
>> % Filter Design using FIRPM function
>> [h,delta] = firpm(M,fo,ao,W); err, delp
delta =
    0.0071
delp =
    0.0058
```

- Clearly, $\delta > \delta_p$. Hence we increase M and repeat the procedure. The optimum M was found to be 50 compared to 60 for Kaiser window.

Example: Lowpass Filter Design (continued)



Example: Bandpass Filter Design

- Consider a bandpass filter design done earlier using Kaiser window with specs

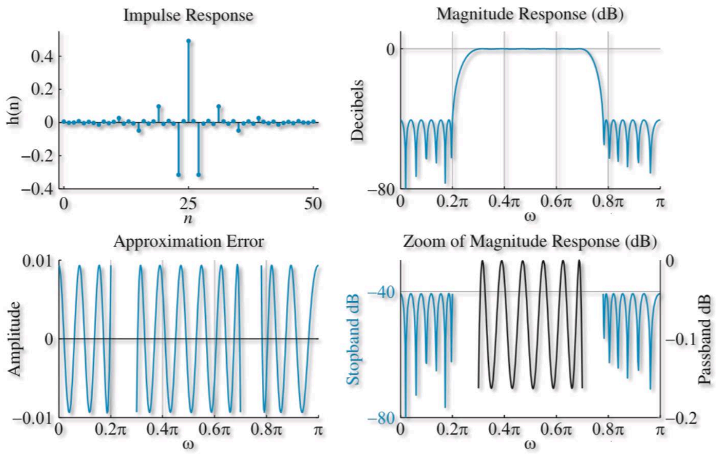
$$\begin{aligned} \left|H(e^{j\omega})\right| &\le 0.01, & |\omega| &\le 0.2\pi \\ 0.99 \le \left|H(e^{j\omega})\right| &\le 1.01, & 0.3\pi \le |\omega| &\le 0.7\pi \\ \left|H(e^{j\omega})\right| &\le 0.01. & 0.78\pi \le |\omega| &\le \pi \end{aligned}$$

```
>> ws1 = 0.2*pi; deltas1 = 0.01; % Lower stopband:
>> wp1 = 0.3*pi; wp2 = 0.7*pi; deltap = 0.01; % Passband
>> ws2 = 0.78*pi; deltas2 = 0.01; % Upper stopband
>> % Estimated Filter order using FIRPMORD function
>> f = [ws1,wp1,wp2,ws2]/pi; % Band-edge array
>> a = [0,1,0]; % Band-edge desired gain
>> dev = [deltas1,deltap,deltas2]; % Band tolerances
>> [M,fo,ao,W] = firpmord(f,a,dev); M
M = 49
>> % Filter Design using FIRPM function
>> [h,delta] = firpm(M,fo,ao,W);
>> delta, deltas1
delta = 0.0108
deltas1 = 0.0100
```


22

Example 10.8: Bandpass Filter (continued)

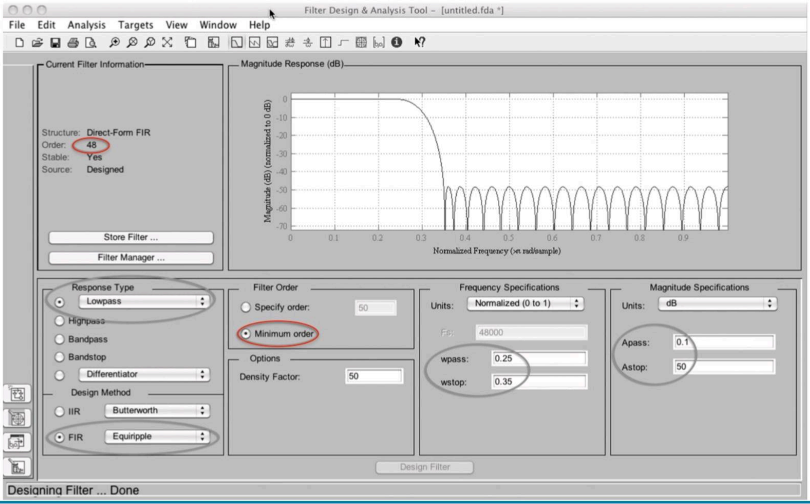
- Again we have to increase M and the optimum order is $M = 50$ while the order is $M = 56$ for the Kaiser window design.



23

`filterdesigner` Tool for Optimum FIR Filter Design

- Setup for obtaining initial filter order M

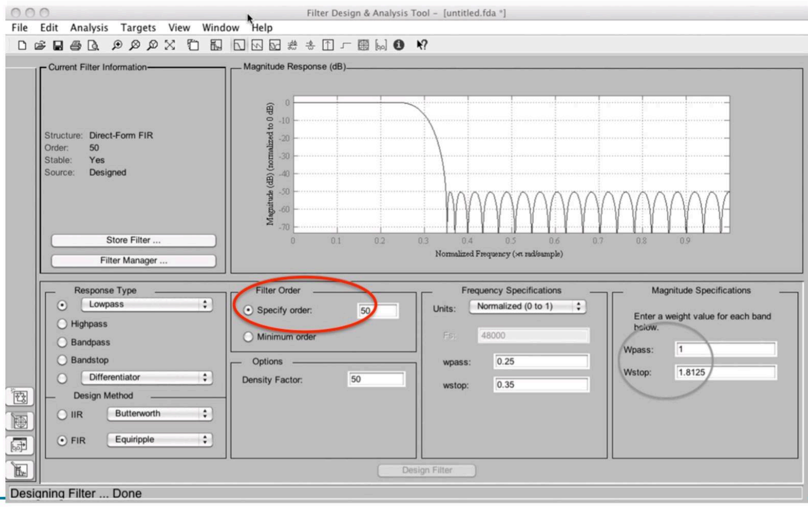


EECE-5666: DSP FIR Filter Design - PM Algorithm 24/30

24

`filterdesigner` Tool for Optimum FIR Filter Design

- Setup for obtaining optimum filter order M



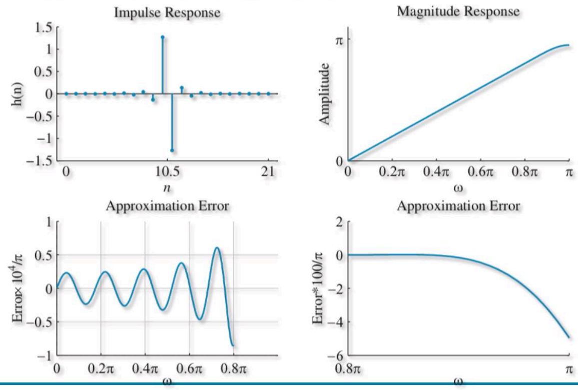
EECE-5666: DSP FIR Filter Design - PM Algorithm 25/30

25

Example: Digital Differentiator Design using Kaiser WIndow

- Type-IV differentiator of order $M = 21$ using Kaiser window ($\beta = 4.5$):

```
>> M = 21; L = M+1; n = 0:M;
>> alpha = M/2; na = (n-alpha);
>> hd = cos(pi*na)./na - sin(pi*na)./(pi*na.^2);
>> % Differentiator design using Kaiser window
>> beta = 4.5; h = hd.*kaiser(L,beta)';
```



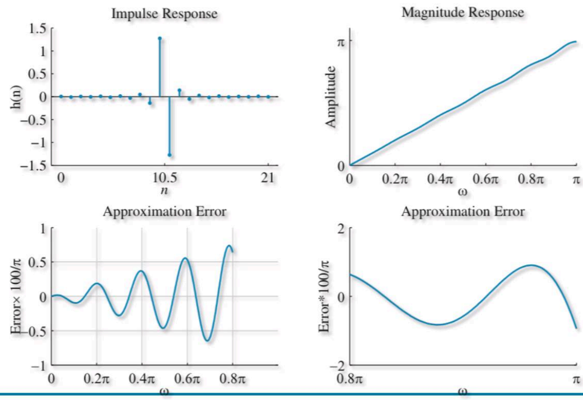
EECE-5666: DSP FIR Filter Design - PM Algorithm 26/30

26

Example: Digital Differentiator Design using PM Algorithm

- Type-IV differentiator of order $M = 21$ using Parks-McClellan algorithm:

```
>> % Differentiator Design using FIRPM function
>> fo = [0,1]; % Band-edge array
>> ao = [0,pi]; % Band-edge desired slope
>> [h,delta] = firpm(M,fo,ao,'differentiator');
```



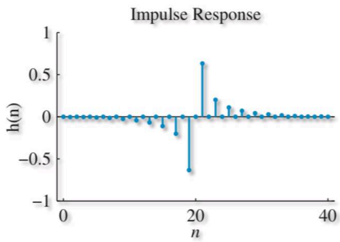
EECE-5666: DSP FIR Filter Design - PM Algorithm 27/30

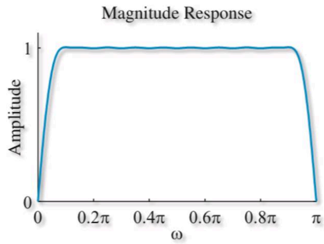
27

Example: Digital Hilbert Transformer Design using Window

- Type-III Hilbert transformer of order $M = 40$ using Hamming window:

```
>> % Hilbert Transformer Specifications
>> M = 40; L = M+1; n = 0:M;
>> alpha = M/2; na = (n-alpha);
>> Dw = 2*pi/L; % Width between frequency samples
>> w1 = 0*Dw; w2 = L/2*Dw;
>> hd = (cos(w1*na)-cos(w2*na))./(pi*na); hd(alpha+1) = 0;
>> % Transformer design using Hamming window
>> h = hd.*hamming(L)';
```

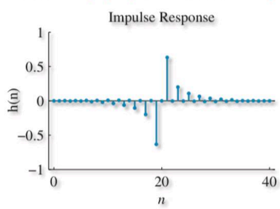


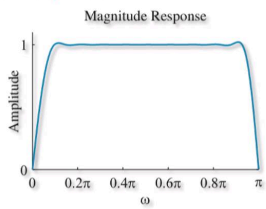


Example: Digital Hilbert Transformer Design using FS Method

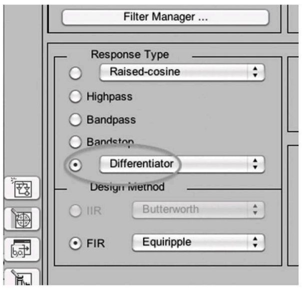
- Type-III Hilbert Transformer of order $M = 40$ using freq. samp. design

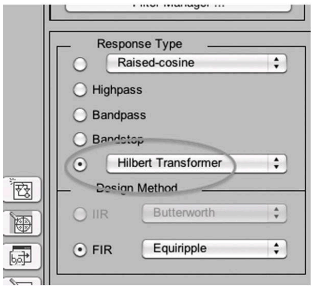
```
>> % Frequency Sampling Design: Assumed Parameters
>> M = 40; n = 0:M; L = M+1; % Impulse response length
>> alpha = M/2; Q = floor(alpha); % phase delay parameters
>> k = 0:M; % Frequency sample index
>> Dw = 2*pi/L; % Width between frequency samples
>> % Transformer Design using Frequency Sampling (Smooth transition)
>> Ad = [0,sin(pi/4),ones(1,18),0.5,-0.5,-ones(1,18),-sin(pi/4)];
>> psid = -alpha*2*pi/L*[(0:Q),-(L-(Q+1:M))]; % Desired Phase
>> Hd = -1j*Ad.*exp(1j*psid); % Desired Freq Resp Samples
>> hd = real(ifft(Hd)); % Desired Impulse response
>> h = hd.*rectwin(L)'; % Actual Impulse response
```





FilterDesigner App for Special Filters



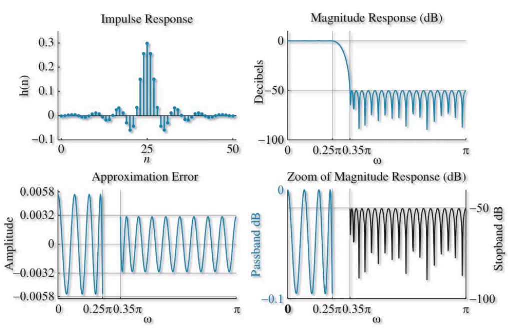


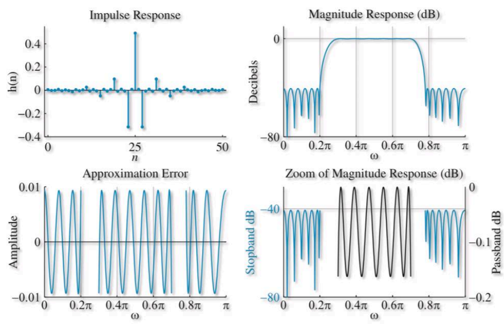
EECE-5666: DSP FIR Filter Design - PM Algorithm 30/30

30

FIR Filter Design

Parks-McClellan Algorithm





31