Digital Signal Processing (Lecture Note 9)

Jin Ho Chang

Professor

Dept. of Electrical Engineering and Computer Science

Filter Design Techniques (Chap. 7)

Digital Filter Types

- Two types of Digital Filters
  - FIR (Finite Impulse Response) filter: All-zero filter
  - IIR (Infinite Impulse Response) filter: Pole-zero filter
- FIR
  - Ideal response, i.e., linear phase and stable
- IIR
  - Better magnitude response
    - ✓ Sharper transition and/or lower stopband attenuation than FIR with the same number of parameters ➔ HW efficient
    - ✓ Established filter types and design methods

Filter Specifications

- Frequency Response $H(\omega) = |H(\omega)| e^{j\Psi(\omega)}$

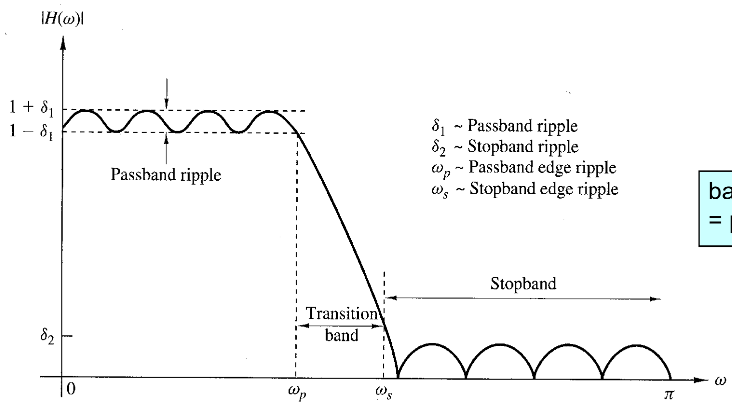
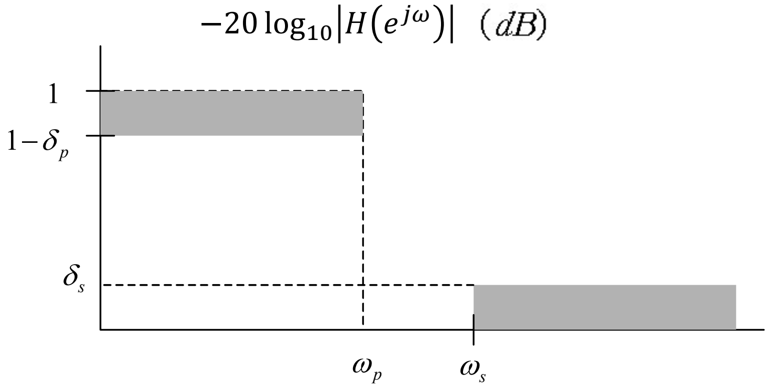


Figure 8.2 Magnitude characteristics of physically realizable filters.

bandwidth of the filter
= passband width

- Other definition (generally used for IIR filters)



Passband ripple: $A_p = -20\log(1-\delta_p) \approx 8.6859\delta_p$

Stopband ripple: $A_s = -20\log\delta_s$

(-3dB) cutoff frequency: Θ_{3db}

5

Finite Impulse Response (FIR) Filter Design

EE 401

FIR Filter

- All-zero filter

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

$$y(n) = b_0 x(n) + b_1 x(n-1) + \cdots + b_{M-1} x(n-M+1)$$

$$= \sum_{k=0}^{M-1} b_k x(n-k)$$

$$y(n) = \sum_{k=0}^{M-1} h(k) x(n-k)$$

- Finite impulse response
- System function

$$H(z) = \sum_{k=0}^{M-1} h(k) z^{-k}$$: a polynomial of degree M-1 in z^{-1}

- Stable
- Realizable
- Noncausal $\xrightarrow{\text{delay}}$ causal
- Linear phase: constant time or group delay
  - h(n) = h(M-1-n): symmetric
  - h(n) = -h(M-1-n): anti-symmetric
- Used in many applications:
  - Biomedical signal processing, Speech-processing, etc.
- Long filter (or many taps) for sharp transition
- Methods for designing FIR filter
  - Window Method
  - Frequency-sampling Method
  - Optimal or Minimax Method

Four Types of Linear Phase FIR Filters

- It is typically impossible to design a linear phase IIR filter, however, designing FIR filters with precise linear phase is easy.
- Consider a causal FIR filter of length M+1 (order M)

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

  – This transfer function has linear phase, if its impulse response h[n] is either

  symmetric
$$h[n]=h[M-n],\qquad 0\le n\le M$$

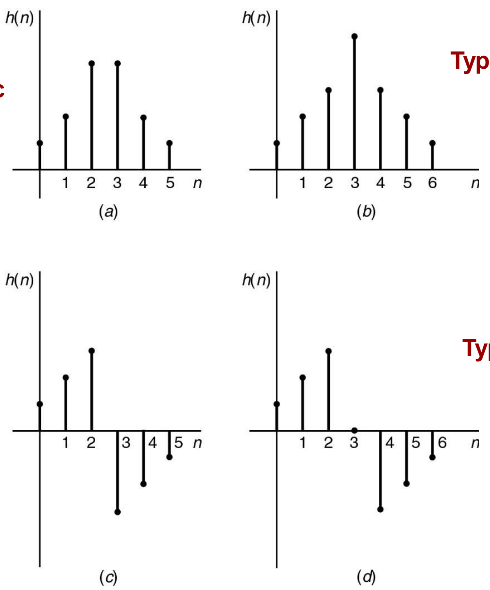
  or anti-symmetric
$$h[n]=-h[M-n],\qquad 0\le n\le M$$

Type II: Even length Even symmetric

Type I: Odd length Even symmetric

Type IV: Even length Odd symmetric

Type III: Odd length Odd symmetric



- Frequency Response of Type I
  - M is even, the sequence is symmetric and of odd length

$$H(\omega) = \underbrace{e^{-j\frac{M}{2}\omega}}_{\alpha} \underbrace{\left\{ h\left[\frac{M}{2}\right] + 2\sum_{i=1}^{M/2} h[i]\cos\left[\left(\frac{M}{2} - i\right)\omega\right] \right\}}_{G(\omega)}$$

✓The system has linear phase
✓Output is delayed by M/2 samples.

- Frequency Response of Type II
  - M is odd, the sequence is symmetric and of even length

$$H(\omega) = 2e^{-j\frac{M}{2}\omega} \left\{ \sum_{i=0}^{(M-1)/2} h[i]\cos\left[\left(\frac{M}{2} - i\right)\omega\right] \right\}$$

✓The system has linear phase
✓Output is delayed by M/2 samples

- Frequency Response of Type III
  - M is even, the sequence is anti-symmetric and of odd length

$$H(\omega) = 2e^{j\left[-\frac{M}{2}\omega + \frac{\pi}{2}\right]} \left\{ \sum_{i=1}^{M/2-1} h[i]\sin\left[\left(\frac{M}{2} - i\right)\omega\right] \right\}$$

✓The system has generalized linear phase.
✓Output is delayed by M/2 samples.

- Frequency Response of Type IV
  - M is odd, the sequence is anti-symmetric and of even length

$$H(\omega) = 2e^{j\left[-\frac{M}{2}\omega + \frac{\pi}{2}\right]} \left\{ \sum_{i=0}^{(M-1)/2} h[i]\sin\left[\left(\frac{M}{2} - i\right)\omega\right] \right\}$$

✓The system has generalized linear phase.
✓Output is delayed by M/2 samples.

- If h[n] is real, complex-valued zeros of H(z) must occur in complex-conjugate pairs
  - ➔ if z_i is a zero,
    then z_i^*, $1/z$, $1/z_i^*$ are also zeros.

- In the symmetric cases, types I and II

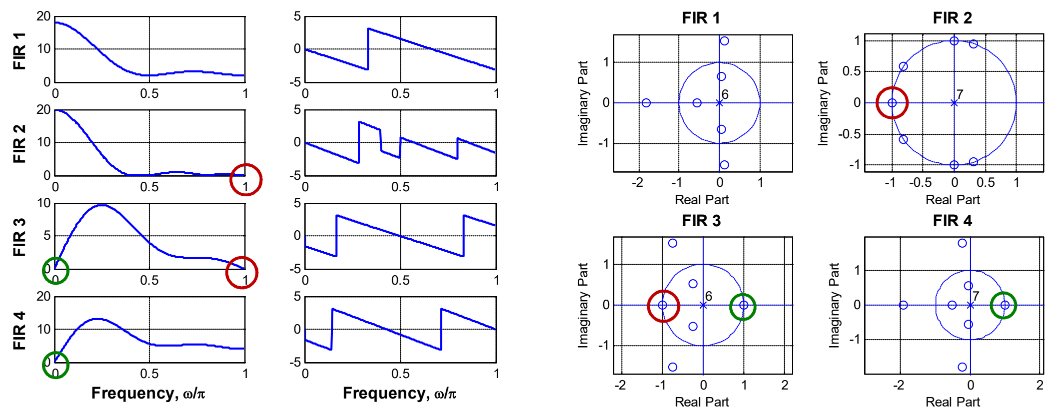
$$H(z)=\sum_{n=0}^{M}h[M-n]z^{-n}=\sum_{k=M}^{0}h[k]z^{k}z^{-M}=z^{-M}H(z^{-1})$$

  - If z_0 is a zero of H(z), then $H(z_0) = {z_0}^{-M}H({z_0}^{-1}) = 0$
  - If $z_0 = re^{j\theta}$ is a zero of H(z), then $z_0^{-1} = r^{-1}e^{-j\theta}$ is also a zero of H(z)
  - When h[n] is real and z_0 is a zero of H(z), then $z_0^* = re^{-j}$ will also be a zero of H(z), and by the proceeding argument, so will
    $$(z_0^*)^{-1} = r^{-1}e^{j\theta}$$
  - When h[n] is real, each complex zeros not on the unit circle will be part of a set of four conjugate reciprocal zeros of the form
    $$\left(1-re^{j\theta}z^{-1}\right)\left(1-re^{-j\theta}z^{-1}\right)\left(1-r^{-1}e^{j\theta}z^{-1}\right)\left(1-r^{-1}e^{-j\theta}z^{-1}\right)$$
  - From $H(z_0) = {z_0}^{-M}H({z_0}^{-1})$, $H(-1) = (-1)^M H(-1)$
    - ✓ If M is odd, H(-1)=-H(-1), so H(-1) must be zero. Thus, for symmetric impulse responses with M odd, the system function must have a zero at z=-1 ➔ **Type II case**.

- In the anti-symmetric cases, types III and IV

$$H(z) = -z^{-M} H(z^{-1})$$

  - This equation can be used to show that the zeros of H(z) for the anti-symmetric case are constrained in the same way as the zeros for symmetric case.
  - The special interesting cases, i.e., z = 1 and z = -1
    - ✓If z = 1, H(1) = -H(1)
      - H(z) must have a zero at z=1 for both M even and M odd.
    - ✓If z = -1, $H(-1) = (-1)^{-M+1} H(-1)$
      - If (M-1) is odd (i.e., if M is even), H(-1) = -H(-1).
      - Therefore, z = -1 must e a zero of H(z) if M is even.
    - ✓In the case of Type III, it must have zeros at both z = 1 and z = -1
    - ✓In the case of Type IV, it must have a zero at z = 1.



Credit to Dr. Robi Polikar, Rowan University

Type	I	II	III	IV
Order	Even	Odd	Even	Odd
Symmetry of h[n]	Symmetric	Symmetric	Anti-symmetric	Anti-symmetric
Symmetry of H(ω)	Symmetric	Symmetric	Anti-symmetric	Anti-symmetric
Period of H(ω)	2π	4π	2π	4π
Uses	LP, HP, BP, BS, …	LP, BP	Differentiators, Hilbert transformers	
Restrictions	No Restrictions	Cannot be used to design a HP filter because it always has a zero at z = -1	Cannot be used for either LP, or HP, or BS filters because it has zeros at both z = 1 and -1.	Not appropriate to design a LP filter due to the presence of a zero at z = 1

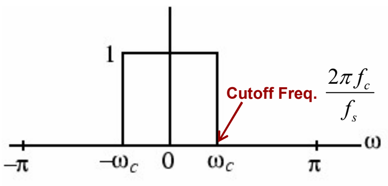
FIR Filter Design

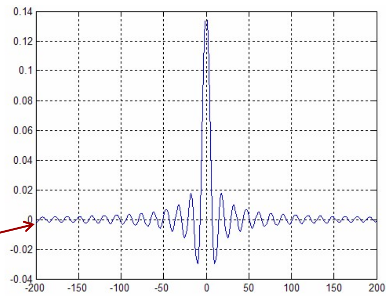
- Ideal Filter

$$H_{LP}(\omega)=\begin{cases}1, & 0\le|\omega|\le\omega_c\\ 0, & \omega_c\le|\omega|\le\pi\end{cases}$$

$$h_{LP}(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty} rect\left(\frac{\omega}{2\omega_c}\right)e^{j\omega n}d\omega$$

$$=\frac{\sin(\omega_c n)}{\pi n},\quad -\infty<n<\infty$$

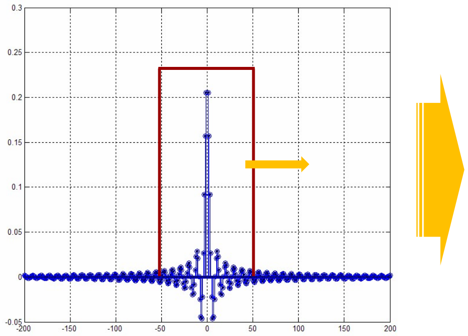




Infinite length and non-causal

Truncation of Ideal Impulse Response

- To make the length of the impulse response, use a window function to truncate it.
- To make the impulse response **causal**, shift the truncated impulse response toward the right by its half length.

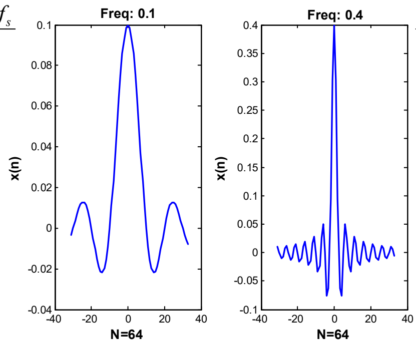


The resultant impulse response is a causal, stable FIR filter.

- The phase factor exp(-jLω/2) is linear phase.

$$h_{LP}(t)=\frac{\sin(\omega_c n)}{\pi n},\quad -\infty<n<\infty \qquad \Rightarrow \qquad h_{LP}(t)=\frac{\sin\left(\omega_c\left(n-\frac{L}{2}\right)\right)}{\pi\left(n-\frac{L}{2}\right)}\quad 0\le n\le L$$

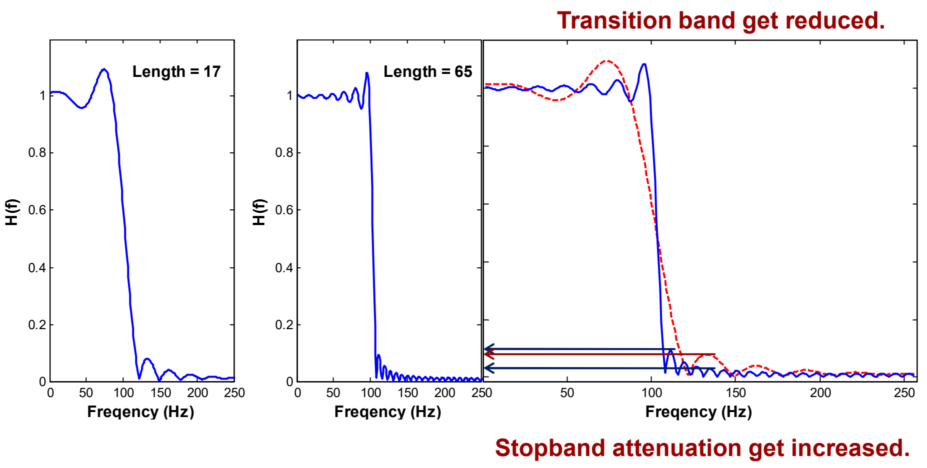
$\frac{0.1 f_s}{2}$ $\frac{0.4 f_s}{2}$



Different Cutoff Frequencies

- The effect of the truncation
  - Truncating the impulse response of an ideal filter to obtain a realizable filter, create oscillatory behavior in the frequency domain ➔ the **Gibbs Phenomenon**
  - As L increases, the number of ripples also increases with decrease of ripple widths.
  - The height of the largest ripples remain constant, regardless of the filter length.
  - As L increases, the height of all other ripples decreases.
  - The mainlobe becomes narrower as L increases, i.e., the drop-off becomes sharper.
  - Similar oscillatory behavior can be seen in all types of truncated filters: LPF, BPF, HPF, and BSP.

- Example



- Desired features of FIR filters
  - Quick drop off ➔ Narrow transition band
    - ✓Narrow mainlobe width
    - ✓increased stopband attenuation
  - Rapidly diminishing sidelobe levels, which causes the ripples.
  - Reducing Gibb's phenomenon
  - Minimize the filter order
- To reduce Gibb's phenomenon, the various window functions can be used in the truncation.
  - Mainlobe width of a window function
    - ✓Influence the sharpness of the transition band
  - Sidelobe energy of a window function
    - ✓Influence the oscillations, i.e., ripples.

1. Rectangular

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

2. Bartlett

$$w(n) = 1 - \frac{2\left|n - \frac{M}{2}\right|}{M}$$

3. Blackman

$$w(n) = 0.42 - 0.5 \cos \frac{2\pi n}{M} + 0.08 \cos \frac{4\pi n}{M}$$

4. Hamming

$$w(n) = 0.54 - 0.46 \cos \frac{2\pi n}{M}$$

5. Hanning

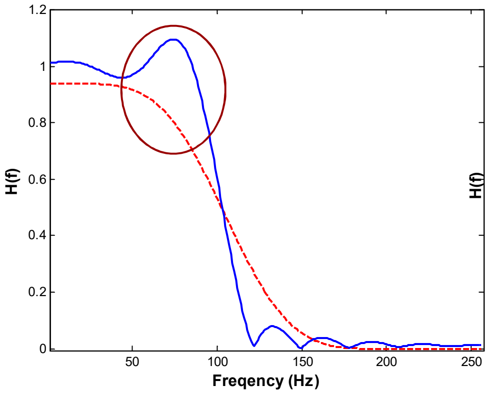
$$w(n) = 0.5 - 0.5 \cos \frac{2\pi n}{M}$$

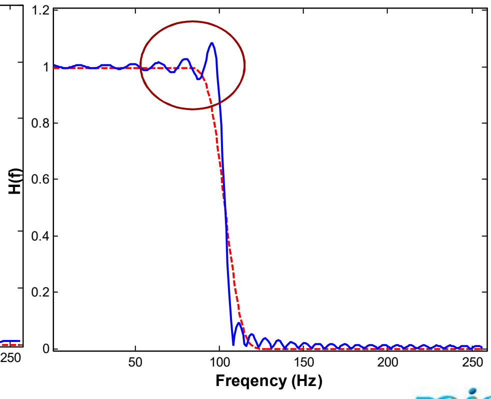
6. Kaiser

$$I_0\left\{\beta\sqrt{1 - \left(\frac{n - M/2}{M/2}\right)^2}\right\}$$

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

- Example
  - Hamming window function is applied to the filter coefficients.
    - ✓The overshoot in the passband disappears.
    - ✓The oscillations are barely visible
    - ✓Significant improvement in the sharpness of the transition band for the filter that uses more coefficients.





Window Method

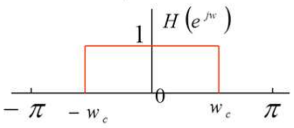
- An ideal desired frequency response

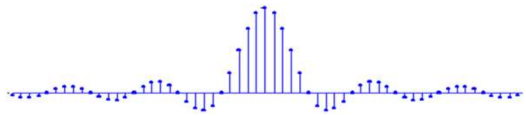
$$H_d(e^{jw}) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-jwn}$$

$$h_d[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{jw}) e^{jwn} dw$$

$$H_{lp}(e^{jw}) = \begin{cases} 1, & |w| < w_c \\ 0, & w_c < |w| < \pi \end{cases}$$

$$h_{lp}(n) = \frac{\sin w_c n}{\pi n}$$

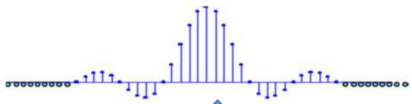


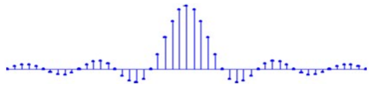


Many idealized systems are defined by piecewise-constant frequency response with dis continuities at the boundaries. As a result, these systems have impulse responses that are non causal and infinitely long.

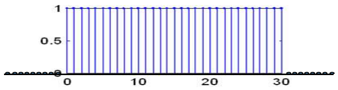
- The most straightforward approach for obtaining a causal FIR approximation is to truncate the ideal impulse response.

$$h[n]=\begin{cases} h_d[n], & 0\le n\le M \\ 0, & otherwise\end{cases}$$





X



$$h[n]=h_d[n]w[n]$$

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

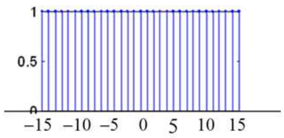
$$H\left(e^{jw}\right)=\frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\left(e^{jw}\right)W\left(e^{j(w-\theta)}\right)d\theta$$

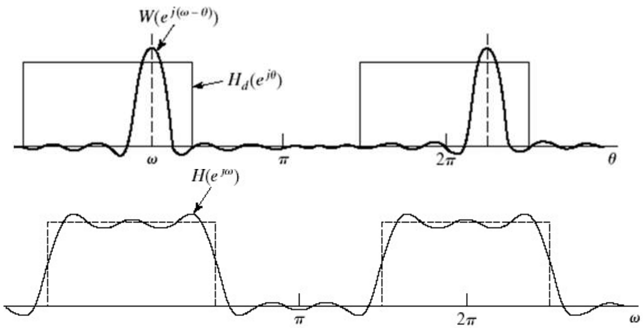
Frequency domain

- Windowing in Frequency Domain

$$H\left(e^{j\omega}\right)=\frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\left(e^{j\omega}\right)W\left(e^{j(\omega-\theta)}\right)d\theta$$

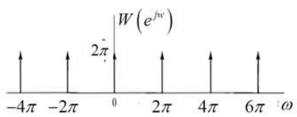
$$w[n]=1 \quad -\infty\le n\le\infty$$

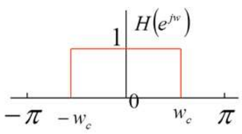




$$W\left(e^{jw}\right)=\sum_{n=-\infty}^{\infty} w[n]e^{-jwn}=2\pi\sum_{k=-\infty}^{\infty}\delta(w-2k\pi)$$

$$H\left(e^{jw}\right)=\frac{1}{2\pi}\int_{-\pi}^{\pi} H_d\left(e^{j\theta}\right)W\left(e^{j(w-\theta)}\right)d\theta=H_d\left(e^{jw}\right)$$

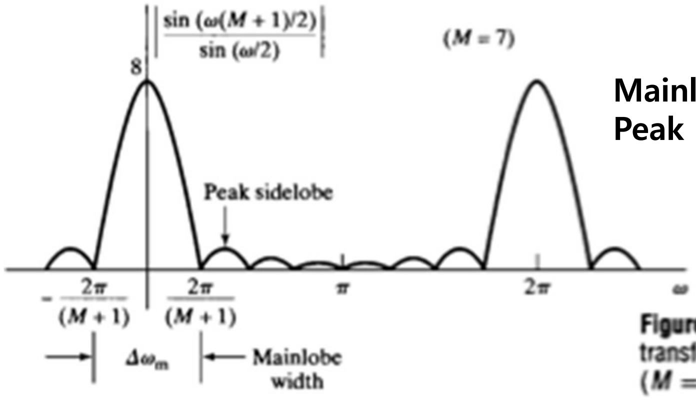




- The windowed version is smeared version of desired response.

- Properties of Window Functions

$$W(e^{j\omega}) = \sum_{n=0}^{M} e^{-j\omega n} = \frac{1 - e^{-j\omega(M+1)}}{1 - e^{-j\omega}} = e^{-j\omega M/2}\frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)}$$



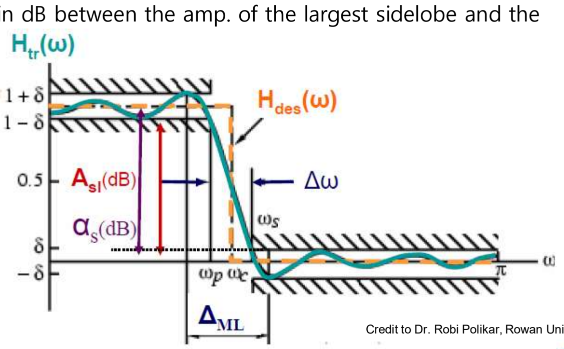
Mainlobe Width: decrease with M
Peak Sidelobe: fixed

Figure 7.20 Magnitude of the Fourier transform of a rectangular window ($M = 7$).

Window-Based FIR Filter Design

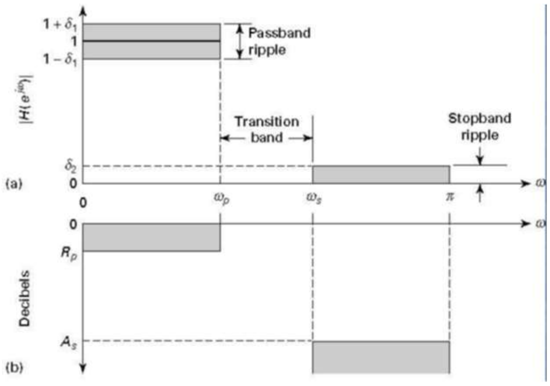
- All windows except a Kaiser window are fixed window functions
  - Magnitude spectrum of each window characterized by a mainlobe centered at ω=0 followed by a series of sidelobes with decreasing amplitudes.
  - Parameters predicting the performance of a window in filter design are:
    - ✓ Mainlobe width (Δ_{ML}): the distance b/w nearest zero-crossings on both sides or transition bandwidth ($\Delta\omega = \omega_s - \omega_p$)
    - ✓ Relative sidelobe level (A_{sl}): difference in dB between the amp. of the largest sidelobe and the mainlobe (or sidelobe attenuation α_s)

For a given window, both parameters all completely determined once the filter order L is set.



Credit to Dr. Robi Polikar, Rowan University

- Design Specs (LPF)



FIR LPF filter specifications: (a) Absolute (b) Relative

$$R_p = -20\log_{10}\frac{1-\delta_1}{1+\delta_1} \qquad A_{sl} = -20\log_{10}\frac{\delta_2}{1+\delta_1}$$

- Band $[0,\omega_p]$: pass band
- Band $[\omega_s, \pi]$: stop band
- Band $[\omega_p, \omega_s]$: transition band

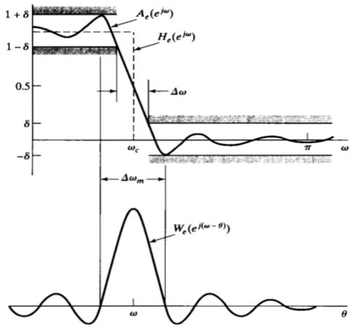
- δ_1: Absolute ripple in pass band
- δ_2: Absolute ripple in stop band

- R_p: Relative ripple in pass band (in dB)
- A_{sl}: Relative ripple in stop band (in dB)

- How to design:
  - Set $\omega_c = (\omega_p+\omega_s)/2$
  - Choose window type based on the specified sidelobe attenuation (A_{sl}) or minimum stopband attenuation (α_s)
  - Choose L according to the transition bandwidth ($\Delta\omega$) and/or mainlobe width (ΔM_L). Note that this is the only parameter that can be adjusted for fixed window functions. Once a window type and M is selected, so are all other parameters.
  - Ripple amplitude cannot be custom designed.
  - Adjustable window have a parameter that can be varied to trade-off between mainlobe width and sidelobe attenuation.

- Properties of Commonly Used Windows

Windows	Mainlobe Width	Sidelobe Attenuation (dB)	Min. Stopband Attenuation, A_{sl} (dB)	Transition BW, ω_s-ω_p
Rectangular	4π/L	-13	21	1.8π/L
Bartlett	8π/L	-27	25	6.1π/L
Hanning	8π/L	-32	44	6.2π/L
Hamming	8π/L	-43	53	6.6π/L
Blackman	12π/L	-58	74	11π/L



DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

Incorporation of Generalized Linear Phase

- In designing FIR filters, it is desirable to obtain causal systems with a generalized linear phase response.
  - The above five windows are all symmetric about the point M/2

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

  - Their Fourier transforms are of the form $W(e^{j\omega}) = \sum_{n=0}^{M} e^{-j\omega n} = \frac{1-e^{-j\omega(M+1)}}{1-e^{-j\omega}} = e^{-j\omega M/2}\frac{\sin[\omega(M+1)/2]}{\sin(\omega/2)}$

$$W(e^{jw}) = W_e(e^{jw})e^{-jwM/2} \qquad W_e(e^{jw})\ is\ a\ real\ and\ even\ function\ of\ w$$

$$h[n] = h_d[n]w[n] \quad : causal \quad if\ h_d[M-n] = h_d[n] \quad \because h[n] = h_d[n]w[n]$$

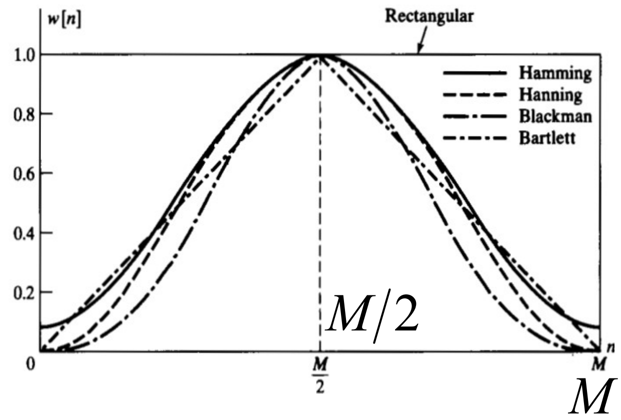
$$\Rightarrow \quad h[M-n] = h[n] \quad : generalized\ linear\ phase \quad H(e^{jw}) = A_e(e^{jw})e^{-jwM/2}$$

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

$$if\ h_d[M-n]=-h_d[n] \quad \because h[n]=h_d[n]w[n]$$

$$\Rightarrow \quad h[M-n]=-h[n] \quad : generalized\ linear\ phase$$

$$H(e^{jw})=jA_o(e^{jw})e^{-jwM/2}$$



DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

- Frequency Domain Representation

$$if\ h_d[M-n]=h_d[n] \qquad H_d(e^{jw})=H_e(e^{jw})e^{-jwM/2}$$

$$w[n]=w[M-n] \qquad W(e^{jw})=W_e(e^{jw})e^{-jwM/2}$$

$$H(e^{jw})=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_d(e^{j\theta})W\left(e^{j(w-\theta)}\right)d\theta \quad \Leftarrow h[n]=h_d[n]w[n]$$

$$=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_e(e^{j\theta})e^{-j\theta M/2}\,W_e\left(e^{j(w-\theta)}\right)e^{-j(w-\theta)M/2}d\theta$$

$$=A_e(e^{jw})e^{-jwM/2}$$

$$where \quad A_e(e^{jw})=\frac{1}{2\pi}\int_{-\pi}^{\pi}H_e(e^{jw})W_e\left(e^{j(w-\theta)}\right)d\theta$$

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

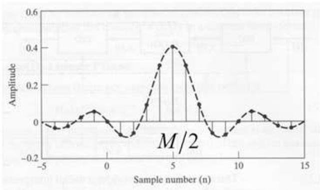
Example: Linear-Phase Lowpass Filter

- The desired frequency response is

$$H_{lp}\left(e^{jw}\right)=\begin{cases} e^{-jwM/2}, & |w|<w_c \\ 0, & w_c<|w|\le\pi \end{cases}$$

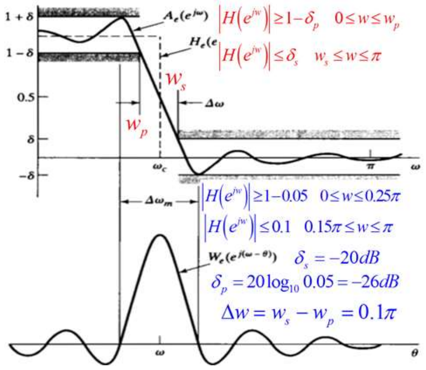
$$h_{lp}[n]=\frac{1}{2\pi}\int_{-w_c}^{w_c} e^{-jwM/2}e^{jwn}dw$$
$$=\frac{\sin\left[w_c\left(n-M/2\right)\right]}{\pi\left(n-M/2\right)} \quad for\ -\infty<n<\infty$$
$$=h_{lp}\left[M-n\right]$$

$$h[n]=\frac{\sin\left[w_c\left(n-M/2\right)\right]}{\pi\left(n-M/2\right)}w[n]$$



$$\left|H\left(e^{jw}\right)\right|\ge 1-\delta \quad 0\le w\le w_p$$
$$\left|H\left(e^{jw}\right)\right|\le \delta \quad w_s\le w\le \pi$$

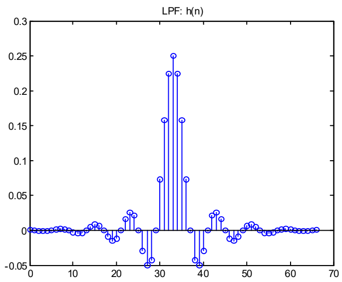
$$\delta_p=20\log_{10}\delta_p$$
$$\delta_s=20\log_{10}\delta_s$$
$$\Delta w=w_s-w_p$$

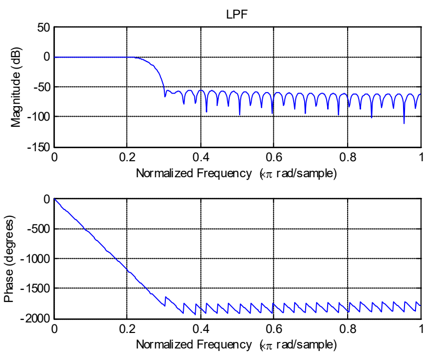


Example – LPF design

$$\omega_p=0.2\pi$$
$$\omega_s=0.3\pi$$
$$A_s=50dB$$

```
wp=0.2*pi; ws=0.3*pi; tr_width=ws-wp;
M=ceil(6.6*pi/tr_width)+1;
n=[0:M-1];
wc=(ws+wp)/2; %ideal cutoff frequency
hd=ideal_lp(wc,M);
w_hamming=(hamming(M))';
h=hd.*w_hamming;

figure(1);stem(n,h); title('h(n)')
figure(2);freqz(h,[1])

%%%%%%%%%%%%%%%%%%%%%%%%%
function hd=ideal_lp(wc,M)
alpha=(M-1)/2;
n=[0:M-1];
m=n-alpha;
fc=wc/pi;
hd=fc*sinc(fc*m);
```





Kaiser Window-Based Filter Design

- The most popular adjustable window

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

  - β is an adjustable parameter to trade-off between the mainlobe width and sidelobe attenuation, and $I_0(x)$ is the modified zero-order Bessel function of the first kind: $$I_0(x) = 1 + \sum_{k=1}^{\infty}\left[\frac{(x/2)^k}{k!}\right]^2$$
  - In practice, this infinite series can be computed for a finite number of terms for a desired accuracy. In general, 20 terms is adequate.

$$I_0(x) \cong 1 + \sum_{k=1}^{20}\left[\frac{(x/2)^k}{k!}\right]^2$$

Given the following:

- ω_p - passband edge frequency and ω_s - stopband edge frequency
- δ_p - peak ripple value in the passband and δ_s - peak ripple value in the stopband

Calculate:

1. Minimum ripple in dB: $\alpha_s = -20\log_{10}(\delta_s)$ *or* $-20\log_{10}\left(\min\{\delta_s, \delta_p\}\right)$
2. Normalized transition bandwidth: $\Delta\omega = \omega_s - \omega_p$
3. Window parameters: $$\beta = \begin{cases} 0.1102(\alpha_s - 8.7), & \alpha_s > 50\,\text{dB} \\ 0.5842(\alpha_s - 21)^{0.4} + 0.07886(\alpha_s - 21), & 21 \le \alpha_s \le 50\,\text{dB} \\ 0, & \alpha_s \le 21\,\text{dB} \end{cases}$$
4. Filter length, M+1: $$M+1 = \begin{cases} \frac{\alpha_s - 7.95}{2.285\Delta\omega} + 1, & \alpha_s > 21 \\ \frac{5.79}{\Delta\omega}, & \alpha_s < 21 \end{cases}$$

Design specs for Kaiser window in your book is different. This one, while may seem more complicated is actually easier to follow.

5. Determine the corresponding Kaiser window
6. Obtain the filter by multipling the ideal filter $h_I[n]$ with $w[n]$

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

Credit to Dr. Robi Polikar, Rowan University

Example 1

Design an FIR filter with the following characteristics:

ω_p=0.3π ω_s= 0.5π, δ_s= δ_p=0.01 → α=40dB, Δω=0.2π

$$\beta = 0.5842(\alpha_s - 21)^{0.4} + 0.07886(\alpha_s - 21),\ 21 \le \alpha_s < 50$$

$$\beta = 0.5842(19)^{0.4} + 0.07886 \times 19 = 3.3953$$

$$M + 1 = \frac{32.05}{2.285(0.2\pi)} + 1 = 23.2886 \approx 24$$

$$\boxed{w[n] = \text{kaiser}(M+1, \beta)\quad (\textit{from matlab})}$$

$$h_{LP}[n] = \begin{cases} \dfrac{\sin(\omega_c(n - M/2))}{\pi(n - M/2)}, & 0 < n < M,\ n \ne \dfrac{M}{2} \\ \dfrac{\omega_c}{\pi}, & n = \dfrac{M}{2} \end{cases} \rightarrow h_{LP}[n] = \begin{cases} \dfrac{\sin(0.4\pi(n-12))}{\pi(n-12)}, & 0 < n < 23,\ n \ne 12 \\ 0.4, & n = 12 \end{cases}$$

$$h_t[n] = h_{LP}[n] \cdot w[n], \quad -12 \le n \le 12$$

Credit to Dr. Robi Polikar, Rowan University

Example 2

We wish to design an FIR lowpass filter satisfying the specifications

$$0.98 < H(e^{j\omega}) < 1.02, \qquad 0 \le |\omega| \le 0.63\pi,$$

$$-0.15 < H(e^{j\omega}) < 0.15, \qquad 0.65\pi \le |\omega| \le \pi,$$

by applying a Kaiser window to the impulse response $h_d[n]$ for the ideal discrete-time lowpass filter with cutoff $\omega_c = 0.64\pi$. Find the values of β and M required to satisfy this specification.

- Since filters designed by the window method inherently have $\delta_1 = \delta_2$ we must use the smaller value for δ.

$$\delta = 0.02$$

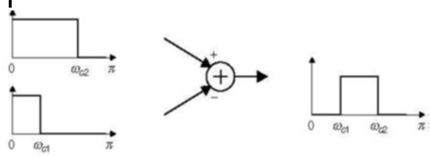
$$A = -20\log_{10}(0.02) = 33.9794$$

$$\beta = 0.5842(\underline{33.9794} - 21)^{0.4} + 0.07886(\underline{33.9794} - 21) = 2.65$$

$$M = \frac{A - 8}{2.285\Delta\omega} = \frac{33.9794 - 8}{2.285(0.65\pi - 0.63\pi)} = 180.95 \rightarrow 181$$

BPF, HPF, and BSF

- Those can be derived in the same manner from generated by applying an inverse FT to rectangular structures with the corresponding shape.



$$b(n) = \frac{\sin[\pi(n-L/2)]}{\pi(n-L/2)} - \frac{\sin[2\pi f_c T_s(n-L/2)]}{\pi(n-L/2)} \quad \text{Highpass}$$

$$b(n) = \frac{\sin[2\pi f_H T(n-L/2)]}{\pi(n-L/2)} - \frac{\sin[2\pi f_L T_s(n-L/2)]}{\pi(n-L/2)} \quad \text{Bandpass}$$

$$b(n) = \frac{\sin[2\pi f_L T(n-L/2)]}{\pi(n-L/2)} + \frac{\sin[\pi(n-L/2)]}{\pi(n-L/2)} - \frac{\sin[2\pi f_H T_s(n-L/2)]}{\pi(n-L/2)} \quad \text{Bandstop}$$

Example 1

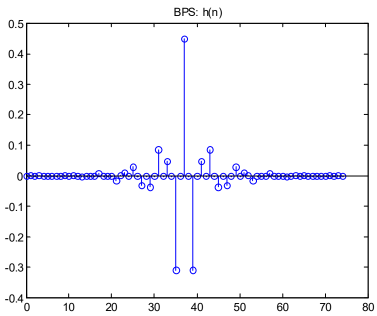
$\omega_{1s}=0.2\pi$, $\omega_{1p}=0.35\pi$,

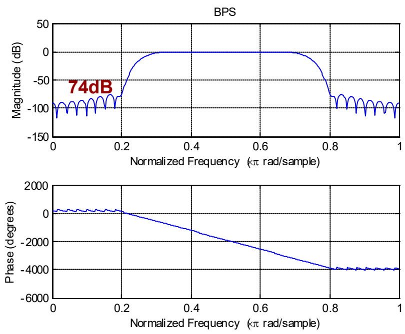
$\omega_{2p}=0.65\pi$, $\omega_{2s}=0.8\pi$,

$A_s=60$ dB

Windows	Mainlobe Width	Sidelobe Attenuation (dB)	Min. Stopband Attenuation, A_{sl} (dB)	Transition BW, ω_s-ω_p
Hamming	8π/L	-43	53	6.6π/L
Blackman	12π/L	-58	74	11π/L

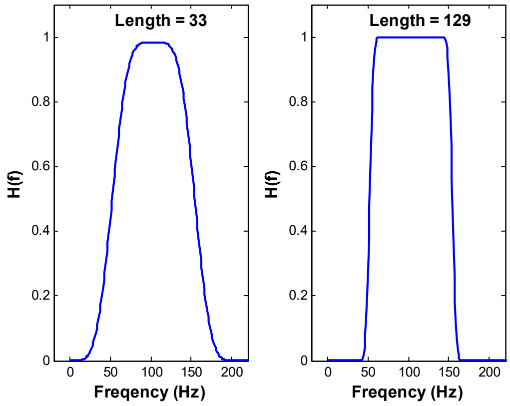
```
wp1=0.35*pi; ws1=0.2*pi;
wp2=0.65*pi; ws2=0.8*pi;

%only one transition bandwidth allowed in window design
tr_width=min(wp1-ws1,ws2-wp2);

M=ceil(11*pi/tr_width)+1;
n=[0:M-1];

wc1=(ws1+wp1)/2; %ideal cutoff frequency 1
wc2=(ws2+wp2)/2; %ideal cutoff frequency 2

hd=ideal_lp(wc2,M)-ideal_lp(wc1,M);
w_blackman=(blackman(M))';
h=hd.*w_blackman;

figure(1);stem(n,h); title('h(n)')
figure(2);freqz(h,[1])
```





Example 2

- Frequency characteristics of an FIR bandpass filter with a coefficient function in conjunction with the Blackman window function.



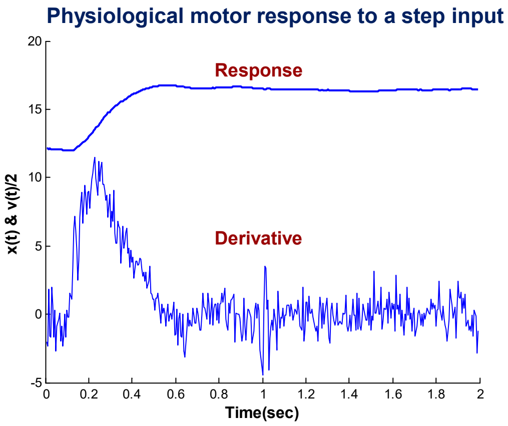
Derivative Operation

- This is a common operation in signal processing and is particularly useful in analyzing certain physiological signals.
  - Digital differentiation can be implemented by taking the difference between two adjacent points, scaling by 1/Ts, and repeating this operation along the entire waveform.
  - In the concept of FIR filter design, this is equivalent to a two coefficient filter

$$\left[\frac{+1}{T_s}, \frac{-1}{T_s}\right]$$

  - This is computed by using diff() function in MATLAB.
  - The frequency characteristic of the derivative operation is a linear increase with frequency ➔ there is considerable gain at the higher frequencies.

– Since the higher frequencies openly contain a greater percentage of noise, this operation tends to produce a noisy derivative curve.



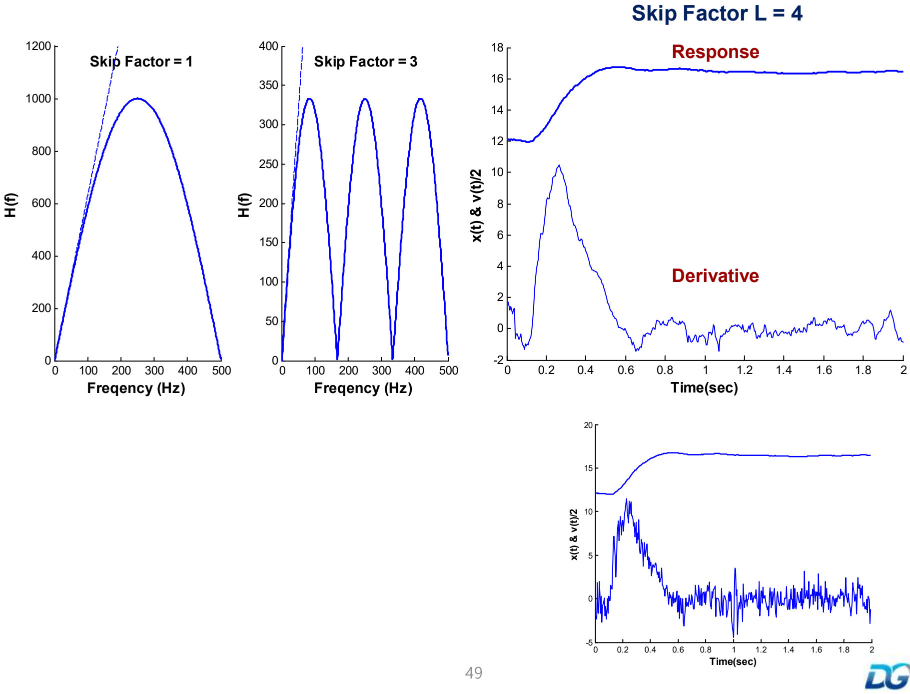
The derivative was calculated by taking the difference in adjacent points and scaling by the sample frequency

- **Two-point central difference algorithm**
- This algorithm acts as a differentiator for the lower frequencies and as an integrator (or lowpass filter) for higher frequencies.
  - This algorithm uses two coefficients of equal but opposite value spaced L points apart. It's the input-output equation:

$$y(n) = \frac{x(n+L) - x(n-L)}{2LT_s}$$

    - ✓ L is the skip factor that influences the effective bandwidth, Ts is the sample interval.
  - Filter coefficients for the two-point central difference algorithm:

$$h(n) = \begin{cases} -0.5/L \cdot T_s & n = -L \\ 0.5/L \cdot T_s & n = +L \\ 0 & n \neq L \end{cases}$$

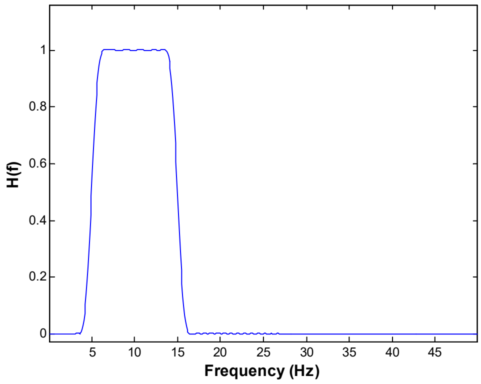


FIR Filter Design in MATLAB

- b = fir1(N, wn, 'ftype', window);
  - N is the filter order (N+1 filter length)
  - Wn is the cutoff frequency
  - ftype is the filter type
    - ✓ For a highpass filter, 'high', for a stopband filter, 'stop'.
    - ✓ If not specified, a lowpass or bandpass filter is assumed depending on the length of wn.
  - window specifies the window function (i.e., Blackman, Hamming, triangular, etc).
    - ✓ The window length should be equal to N+1.
  - The output, b, is a vector containing the filter coefficients.
  - For bandpass and bandstop filters, N must be even and is incremented if not.
  - Hamming window is default if this argument is not specified.

– Example of Filter Design with fir1()
  ✓ Design a window-based bandpass filter with cutoff frequencies of 5 and 15 Hz. Filter order is 128. Assume a sampling frequency of 100 Hz.

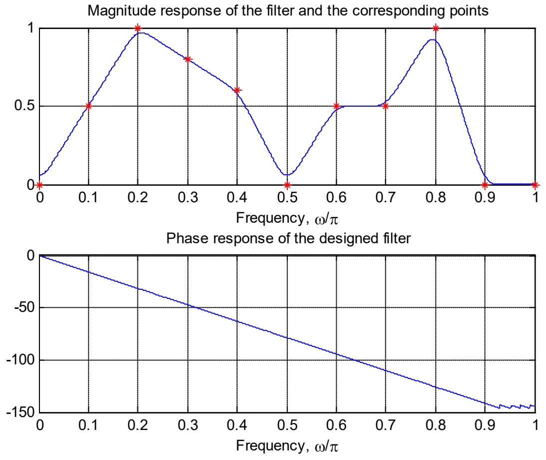
```
fs = 100;                    % Sampling frequency
order = 128;                 % Filter order
wn = [5 15]/(fs/2);          % Cutoff frequencies
b = fir1(128,wn);            % Use default Hamming window
[h ,freq] = freqz(b,1,512,100);
plot(freq,abs(h),'b');
xlabel('Frequency (Hz)');
ylabel('H(f)');
```



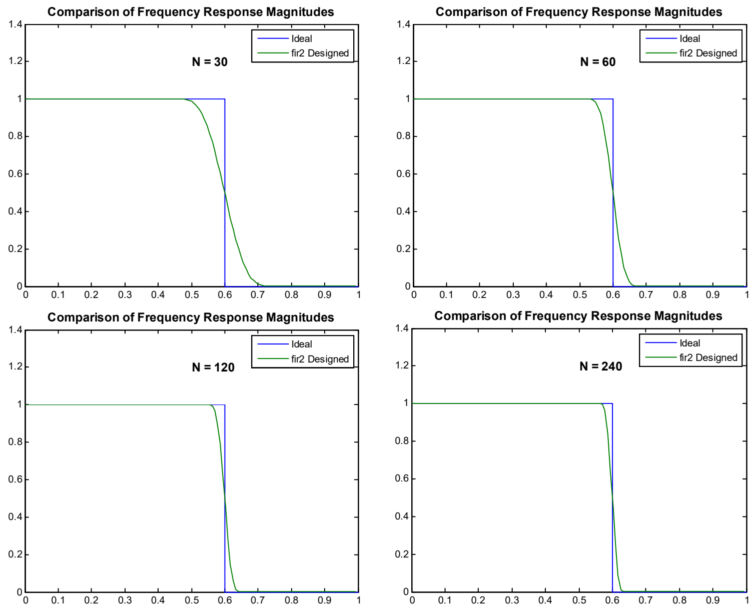
- b = fir2(N, f, A, window);
  - This uses the frequency sampling-based finite impulse response filter design when a more general, or arbitrary frequency response curve is desired.
  - N is the filter order
  - f is a vector of normalized frequencies in ascending order
  - A is the desired gain of the filter at the corresponding frequency in vector f.
    - ✓ plot(f, A) would show the desired magnitude frequency curve.
    - ✓ f and A must be the same length.
    - ✓ Frequency ranges between 0 and 1, normalized to fs/2.
  - The argument, window, is the same as in fir1().
  - The output, b, is the filter coefficients
  - You will need to determine the filter order by trial and error.
    - ✓ You may need higher orders if your specified points requires a sharp transition.

– Example of Filter Design with fir2()

```
f=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1];
A=[0 0.5 1 0.8 0.6 0 0.5 0.5 1 0 0];
b=fir2(100, f, A, 1024);
subplot(211)
[H w]=freqz(b, 1, 1024);
plot(w/pi, abs(H)); hold on
plot(f, A, 'r*')
grid
xlabel('Frequency, \omega/\pi')
title(' Magnitude response of the filter and the
corresponding points')
subplot(212)
plot(w/pi, unwrap(angle(H)));
xlabel('Frequency, \omega/\pi')
title(' Phase response of the designed filter')
grid
```



f = [0 0.6 0.6 1]; A = [1 1 0 0];



- Two-Stage FIR filter design
  - 1st Stage: determining the filter order and cutoff frequencies to best approximate a desired frequency response curve.
    - ✓Inputs to these routines specify an ideal frequency response
    - ✓Outputs include the number of stages required, cutoff frequencies, and other information, which are the inputs to the second stage
  - 2nd Stage: generating the filter coefficient function based on the input arguments which include the filter order, the cutoff frequencies, the filter type (generally optional)

- Example 1: Parks-McClellan Optimal FIR filters (Chap. 7.4.3)
  - An iterative algorithm for finding the optimal Chebyshev FIR filter
  - The goal of the algorithm is to minimize the error in the pass and stop bands by using the Chebyshev approximation.
  - For 1st Stage, [n, f0, a0, w] = remezord(f, a, dev, Fs);
    - ✓f: specifying frequency ranges between 0 and Fs/2 as a pair of frequencies
    - ✓a: specifying the desired gains within each of these ranges
      - f has a length of 2n-2, where n is the length of a.
    - ✓dev: a vector specifying the maximum allowable deviation or ripple within each of these ranges, which has the same length as a.
    - ✓n: required filter order
    - ✓f0: normalized frequency ranges
    - ✓a0: the frequency amplitudes for those ranges
    - ✓w: a set of weights that tell the second stage how to assess the accuracy of the fit in each of frequency ranges.
  - F=[100 300 400 500]; a=[1 0 1]; dev=[.01 .1 .01];

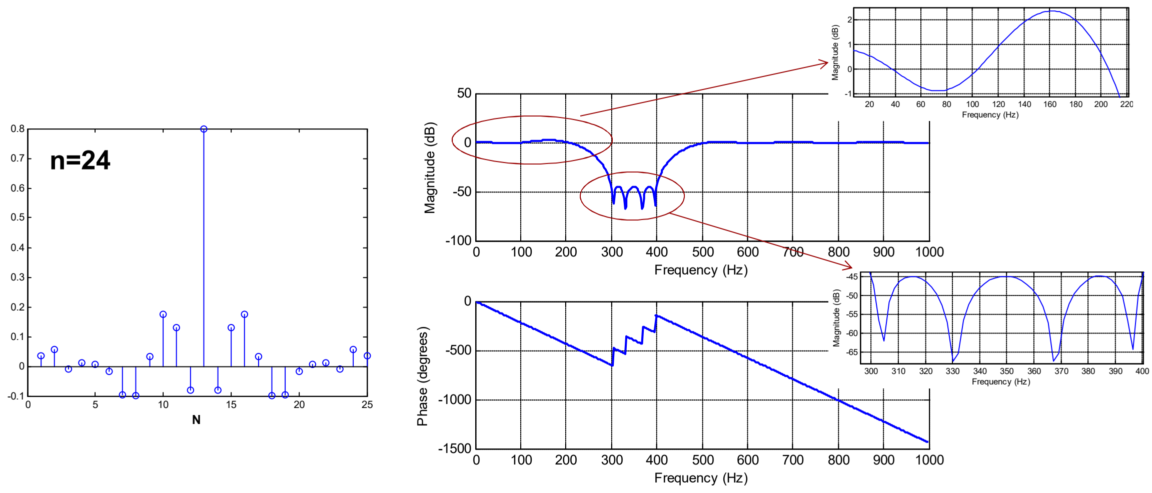
- For 2nd Stage, b = remez (n, f, a, w, 'ftype');
  - ✓ The four outputs of the first stage become the input to the second stage filter design routine.
  - ✓ 'ftypte': specifying a filter type, which is
    - 'hilbert', for linear-phase filters with odd symmetry (type III and IV)
      - This class of filters includes the Hilbert transformer, which has a desired amplitude of 1 across the entire band.
    - 'differentiator', for type III and IV filters, using a special weighting technique
      - For nonzero amplitude bands, it weights the error by a factor of 1/f so that the error at low frequencies is much smaller than at high frequencies.
- Alternative functions: firpmord() for remezord() and firpm() for remez()

• Example 2: Least Square Linear-phase FIR Filters
- Minimizing the weighted, integrated squared error between an ideal piecewise linear function and the magnitude response of the filter over a set of desired frequency bands.
- b = firls(n, f, a, w, 'ftype'); ➔ same input structure as remez()

• Practice I
- Design a bandstop filter with a passband gain of 1 (0dB) between 0 and 100, a stop-band gain of -40 dB between 300 and 400 Hz, and an upper passband gain of 1 between 500 and 1000 Hz (fs/2, fs=2 kHz). Maximum ripple for the pass band should be ±1.5 dB (i.e., 3 dB).

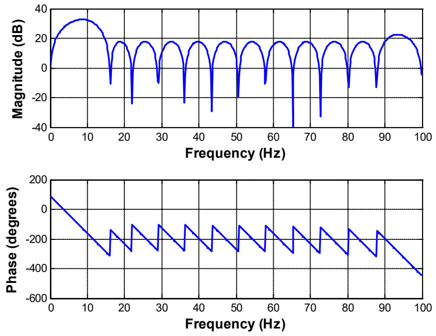
```
rp_pass = 3;                    % Specify ripple tolerance in passband
rp_stop = 40;                   % Specify error tolerance in stopband
fs = 2000;                      % Sample frequency: 2 kHz
f = [100 300 400 500];          % Define frequency ranges
a= [1 0 1];                     % Specify gain in those regions
dev = [(10^(rp_pass/20)-1)/(10^(rp_pass/20)+1) 10^(-rp_stop/20) …
        (10^(rp_pass/20)-1)/(10^(rp_pass/20)+1)]
% Design filter - determine filter order
[n,fo,ao,w] = remezord(f,a,dev,fs)   % Determine filter order, Stage 1
b = remez(n, fo, ao, w);             % Determine filter weights, Stage 2
freqz(b,1,512,fs);                   % Plot filter frequency
```

$$\boxed{R_p = -20\log_{10}\frac{1-\delta_1}{1+\delta_1}}$$

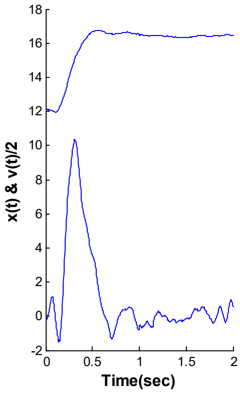


- Practice II
  - Design a FIR derivative filter and compare it to the two point central difference algorithm

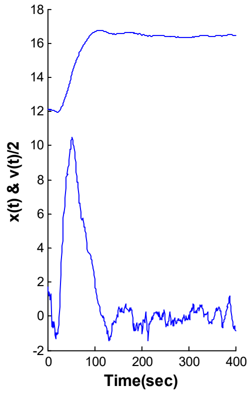
```
Ts = 1/200;                          % Assume a Ts of 5 msec.
fs = 1/Ts;                           % Sampling frequency
L = 4;                               % Use skip factor of 4
fc = .05;                            % Derivative cutoff frequency
t = (1:length(data))*Ts;
%----------------- Design filter -----------------%
f = [0  fc  fc+.1  .9];              % Construct desired frequency characteristic
a = [0  (fc*fs*pi)  0  0];           % Upward slope until .05 fs then lowpass
b = remez(28, f, a, 'differentiator');
freqz(b, 1, 512, fs);                % Plot filter frequency response
d_dt1 = filter(b, 1, data);          % Apply FIR differentiator
figure, subplot(1, 2, 1);
hold on;
plot(t,data(1:400)+12, 'k');         % Plot data offset
plot(t,d_dt1(1:400)/2, 'k');         % Scale velocity by 1/2
xlabel('Time(sec)'); ylabel('x(t) & v(t)/2');
%----------------- Now apply two point central difference algorithm -----------------%
hn = zeros((2*L)+1,1);               % Set up h(n)
hn(1,1) = 1/(2*L* Ts);
hn((2*L)+1,1) = -1/(2*L*Ts);         %Note filter weight reversed if using convolution
d_dt2 = conv(data,hn);
subplot(1,2,2);
hold on;
plot(data(1:400)+12, 'k');
plot(d_dt2(1:400)/2, 'k');
xlabel('Time(sec)'); ylabel('x(t) & v(t)/2');
```



Derivative produced by an FIR filter



Derivative produced by the two-point central difference differentiator



- ❖ The FIR filter does produce a cleaner derivative without reducing the value of the peak velocity.
- ❖ The FIR filter order (n=28) and derivative cutoff frequency (fc=0.05fs/2) were chosen empirically to produce a clean derivative with a maximal velocity peak.

Infinite Impulse Response (IIR) Filter Design

IIR Filter

$$y(k)=\sum_{n=1}^{L_N} b(n)x(k-n)-\sum_{n=1}^{L_D} a(n)y(k-n)$$

- a(n): the feedback coefficients
  - They act on past outputs y(n)
- b(n): the feedforward coefficients
  - They only act on the input signal x(n)

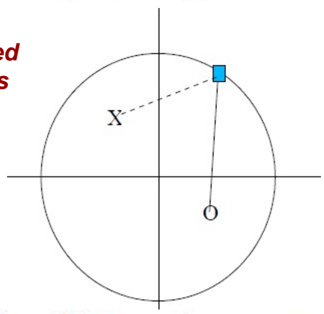
$$H(z)=\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_Mz^{-M}}{1+a_1z^{-1}+a_2z^{-2}+\cdots+a_Nz^{-N}}$$

$$=z^{(N-M)}\frac{b_0\prod_{l=1}^{M}\underbrace{(z-\alpha_l)}_{\text{Zeros}}}{a_0\prod_{l=1}^{N}\underbrace{(z-\beta_l)}_{\text{Poles}}}$$

- Frequency response in terms of poles and zeros
  - It can be determined from the transfer functions by substituting z=exp(j2πF).

$$H(F)=\frac{b_0+b_1e^{-j2\pi F}+\cdots+b_Me^{-j2\pi FM}}{1-a_1e^{-j2\pi F}-\cdots-a_Ne^{-j2\pi FN}}=\frac{b_0e^{-j2\pi FM}(e^{j2piF}-z_1)(e^{j2piF}-z_2)\cdots(e^{j2piF}-z_M)}{e^{-j2\pi FN}(e^{j2\pi F}-p_1)\cdots(e^{j2\pi F}-p_N)}$$

The magnitude response at any frequency (indicated by the point on the unit circle) is the ratio of lengths to the zero (solid) and to the pole (dashed).



$$H(F)=\frac{e^{j2\pi F}-z_1}{e^{j2\pi F}-p_1}\Rightarrow|H(F)|=\frac{|e^{j2\pi F}-z_1|}{|e^{j2\pi F}-p_1|}$$

Consequently, when $z=e^{j2\pi F}$ approaches the location of the zero, the numerator decreases, and therefore so too does $|H(F)|$. Conversely, when $z=e^{j2\pi F}$ approaches a pole, the denominator decreases, and therefore, the magnitude *increases*.

	FIR	IIR
Advantages	• Exact linear phase (or constant group delay) can be achieved. • Filter structure always stable. • It is less sensitive to finite implementation with quantized coefficients. • Initial transients are of finite duration • Its extension to 2-D applications is straightforward.	• It can achieve a desired sharpness of response with much fewer coefficients. • It can implement unusual characteristics such as being an allpass. • Established filter types and design methods.
Disadvantages	• Order of an FIR filter is usually much higher than the order of an equivalent IIR filter meeting the same specifications. • It requires higher computational complexity.	• The design of IIR filters is not as straightforward as FIR filters. ➔ MATLAB provides a number of advantaged routines to assist in this process. • It is difficult to achieve exact linear phase (or constant group delay), i.e. nonlinear phase characteristics. ➔ Noncausal techniques can be used to produce zero phase filters.

IIR Filter Design Procedure

1) Set up digital filter specifications
2) Determine the corresponding analog filter specifications
   ➔ Frequency translation involved
3) Design the analog filter
4) Transform the analog filter to digital filter using various transformation methods
   ➔ Transformation Goal : $H(s)$ → $H(z)$
   - Impulse Invariant Method
   - Bilinear Transformation

Analog Filter Specifications

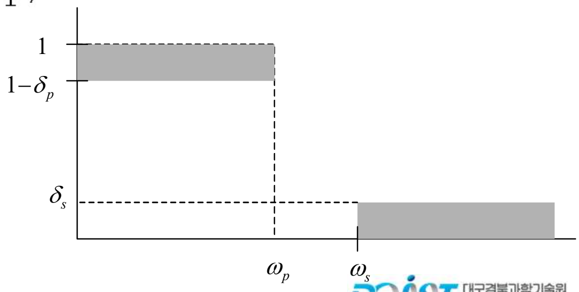
- Important parameters
  - Passband ripple : $A_p = -20\log(1-\delta_p) \approx 8.6859\delta_p$
  - Stopband attenuation : $A_s = -20\log\delta_s$
  - Discrimination factor :

$$d = \left[\frac{(1-\delta_p)^{-2}-1}{\delta_s^{-2}-1}\right]^{1/2} = \left(\frac{10^{0.1A_p}-1}{10^{0.1A_s}-1}\right)^{1/2}$$

  - Selectivity factor :

$$k = \frac{\omega_p}{\omega_s}$$

  - (-3dB) cutoff frequency : ω_{3db}



67

- Frequency response
  - Transfer function : Rational

$$H(s) = \frac{b_0 s^q + b_1 s^{q-1} + \cdots + b_q}{s^p + a_1 s^{p-1} + \cdots + a_p}$$

  - Asymptotic attenuation at high frequency

$$20\log|H(\omega)| \approx 20\log|b_0| - 20(p-q)\log\omega$$

$$20(p-q)\ dB/decade$$

  - Attenuation function: $\Lambda(\cdot)$ (rational or polynomial function)

$$|H(\omega)|^2 = \frac{1}{1+\Lambda(\frac{\omega}{\omega_0})}$$: Square magnitude frequency response

ω_0: reference frequency

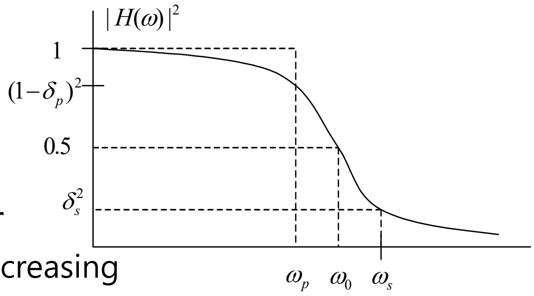
✓If $\Lambda(\cdot)$ is monotone, so is $|H(\omega)|^2$
✓If $\Lambda(\cdot)$ is oscillatory, $|H(\omega)|^2$ exhibits ripple.

68

Popular Analog Filter: Butterworth Filters

- Magnitude Squared Response

$$|H(\omega)|^2 = \frac{1}{1+(\frac{\omega}{\omega_0})^{2N}}$$



- Properties of a LP Butterworth filter
  - Magnitude response : monotonically decreasing
  - Maximum gain : 0 at $\omega = 0$
  - $|H(\omega_0)| = \sqrt{0.5}$ ➔ ω_0: -3 dB point
  - Asymptotic attenuation at high frequency : 20N dB/decade
  - Maximally flat at DC (maximally flat filter)

$$\left.\frac{\partial |H(\omega)|^2}{\partial \omega^k}\right|_{\omega=0} = 0, \qquad 1 \le k \le 2N-1$$

- LP Butterworth filter design procedure
  1. Set up filter spec : $\delta_p \ \delta_s \ \omega_p \ \omega_s \ => \ d, k$
  2. Compute N, using $N \geq \frac{\log_e(1/d)}{\log_e(1/k)}$
  3. Choose ω_0 using

$$\omega_p[(1-\delta_p)^{-2}-1]^{-1/2N} \le \omega_0 \le \omega_s[\delta_s^{-2}-1]^{-1/2N}$$

  4. Compute the poles s_k, using

$$s_k = \omega_0 \cos\left[\frac{\pi}{2} + \frac{(2k+1)\pi}{2N}\right] + j\omega_0 \sin\left[\frac{\pi}{2} \ \frac{(2k+1)\pi}{2N}\right], \quad 0 \le k \le N-1$$

  5. Compute $H(s)$, using

$$H(s) = \prod_{k=0}^{N-1} \frac{-s_k}{s-s_k}$$

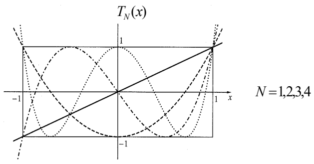
Popular Analog Filter: Chebyshev Filters

- Chebyshev polynomial of degree

$$T_N(x) = \begin{cases} \cos(N\cos^{-1}x), & |x| \le 1 \\ \cosh(N\cosh^{-1}x), & |x| > 1 \end{cases}$$

Recursive formula: $T_N(x) = 2T_{N-1}(x) - T_{N-2}(x), \quad T_0(x) = 1, \; T_1(x) = x$

If N is even(odd), so is $T_N(x)$



- Monotone only in one band
  - Chebyshev Type I : equiripple in the passband
  - Chebyshev Type II : equiripple in the stopband
- Sharper than Butterworth due to the ripples ! Why ?
  - Sharpest if equiripple in both bands, pass- and stop-bands.
  - Phase response : Better for maximally flat or monotonic mag response filters

- **Chebyshev-I**: Chebyshev filter of the first kind

$$|H(\omega)|^2 = \frac{1}{1+\varepsilon^2 T_N^2\left(\frac{\omega}{\omega_0}\right)}$$

$$\omega_0 = \omega_p$$

$$\epsilon = \left[(1-\delta_p)^{-2} - 1\right]^{\frac{1}{2}}$$

$$N \ge \frac{\cosh^{-1}\left(\frac{1}{d}\right)}{\cosh^{-1}\left(\frac{1}{k}\right)}$$

- Properties
  - All-pole filter
  - For $0 \le \omega \le \omega_0$

$$\frac{1}{1+\varepsilon^2} \le |F(\omega)|^2 \le 1 \qquad |H(\omega)|^2 = \begin{cases} 1/(1+\varepsilon^2), & N \text{ even} \\ 1, & N \text{ odd} \end{cases}$$

  - For $\omega > \omega_0$
    - ✓ Monotonically decreasing because $T_N(x)$: *monotonic* for $|x| > 1$
    - ✓ asymptotic attenuation : $20N \; dB/decade$

- **Chebyshev-II**: Chebyshev filter of the second kind
- Inverse Chebyshev filter or Chebyshev-II

$$|H(\omega)|^2 = 1 - \frac{1}{1+\varepsilon^2 T_N^2(\frac{\omega_0}{\omega})} = \frac{\varepsilon^2 T_N^2(\frac{\omega_0}{\omega})}{1+\varepsilon^2 T_N^2(\frac{\omega_0}{\omega})}$$

$$\omega_0 = \omega_s$$

$$\epsilon = [\delta_s^{-2} - 1]^{-\frac{1}{2}}$$

$$N \geq \frac{\cosh^{-1}\left(\frac{1}{d}\right)}{\cosh^{-1}\left(\frac{1}{k}\right)}$$

- Properties
  - Passband : monotonic Stopband : equi-ripple
  - Contains both the poles and zeros

$\boxed{\omega \geq \omega_0}$ $\quad 0 \leq |H(\omega)|^2 \leq \frac{\varepsilon^2}{1+\varepsilon^2}$ $\quad |H(0)|^2 = 1$ for all $N, \omega_0, \varepsilon > 0$

$\boxed{0 \leq \omega \leq \omega_0}$: monotonically decreasing $\quad v_k = \frac{\omega_0^2}{s_k}, \quad 0 \leq k \leq N-1$

$$H(s) = \prod_{k=0}^{N-1} \frac{v_k(s-u_k)}{u_k(s-v_k)} \qquad u_k = \frac{j\omega_0}{\cos[\frac{(2k+1)\pi}{2N}]}, \quad 0 \leq k \leq N-1$$

Popular Analog Filter: Elliptic Filters

- Overview
  - Equiripple in both the passband and the stop band
  - Minimum possible order for a given spec : Sharpest (optimum)
- Magnitude Squared Response: LP elliptic filter

$$|H(\omega)|^2 = \frac{1}{1+\varepsilon^2 R_N^2(\frac{\omega}{\omega_0})}$$

$$\epsilon = \left[(1-\delta_p)^{-2} - 1\right]^{\frac{1}{2}}$$

$$N \geq \frac{K(k^2)K(1-d^2)}{K(1-k^2)K(d^2)}$$

$$K(m) = \int_0^{0.5\pi} (1 - m \cdot sin^2 x)^{-\frac{1}{2}} dx$$

❖ $R_N(x)$: **Jacobian elliptic function of degree N**

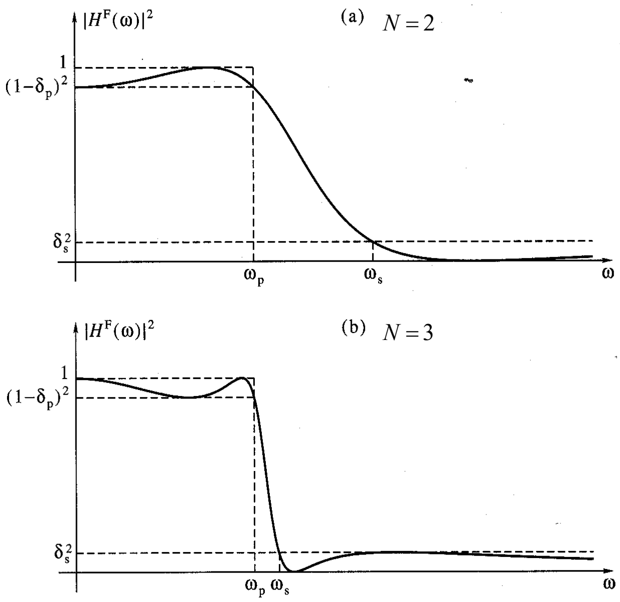
- Even(odd) function of x for even(odd) N
- For $-1 \leq x \leq 1$, $R_N(x)$ oscillates between -1 and +1

  => $|H(\omega)|^2$ *oscillates between 1 and* $1/(1+\varepsilon^2)$ *for* $0 \leq |\omega| \leq \omega_0$

- For $1 < |x| < \infty$

  $|R_N(x)|$ *oscillates between* $1/d$ *and* ∞ *for* $\omega_0 < |\omega| < \infty$

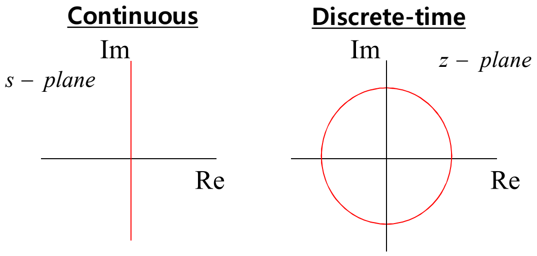
- Example

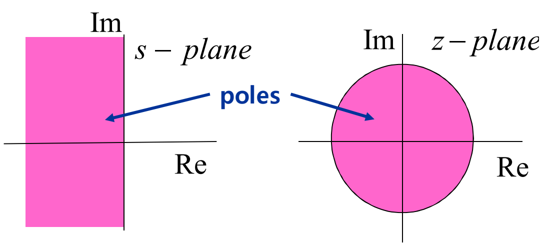


Constrains of Transformation

- To preserve the essential properties of the frequency response, the imaginary axis of the s-plane is mapped onto the unit circle of the z-plane.

$$s = j\Omega \quad \Leftrightarrow \quad z = e^{jw}$$





If the continuous system has poles only in the left half of the s-plane, then the discrete-time filter must have poles only inside the unit circle.

-> preserve the property of stability

- Relation between Laplace Transform and Z-transform

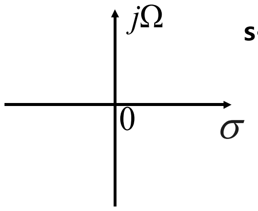
Laplace transform

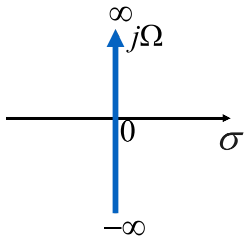
Time domain: $x(t)$ ⇨ **Complex frequency domain:** $X(s) \overset{\Delta}{=} \int_{-\infty}^{\infty} x(t) e^{-st} dt$

$s = \sigma + j\Omega \quad \Omega = 2\pi f$

$\sigma = 0$ ⇨ $s = j\Omega$

s-plane





Fourier Transform

Laplace transform ⇔ continuous time signal

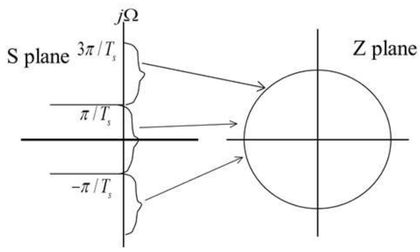
z-transform ⇔ discrete-time signal

$$z = e^{sT} = e^{(\sigma + j\Omega)T} = e^{\sigma T} \cdot e^{j\Omega T} = re^{j\omega}$$

$r = e^{\sigma T}$

$\omega = \Omega T$ ⇔ Relationship between s and z

$$\omega = \Omega T = 2\pi f / f_s \qquad z = re^{j\omega}\big|_{r=1} = e^{j\omega}$$



DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

Impulse Invariance Transformation

- We know

$$h[n] = h_c(nT) \Longrightarrow H(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_c\left(j\frac{\omega}{T} + j\frac{2\pi}{T}k\right)$$

- If $H_c(j\Omega) = 0, \ |\Omega| \geq \pi/T$ (i.e., bandlimited),

  then $H(e^{j\omega}) = \frac{1}{T} H_c\left(j\frac{\omega}{T}\right), \quad |\omega| \leq \pi$

- If $h[n] = Th_c(nT)$, then $H(e^{j\omega}) = H_c\left(j\frac{\omega}{T}\right), \quad |\omega| \leq \pi$
- The discrete-time and continuous-time frequency responses are related by a linear scaling of the frequency axis
  - ✓ $\omega = \Omega T$ for $|\omega| < \pi$
- Any practical continuous-time filter cannot be exactly bandlimited

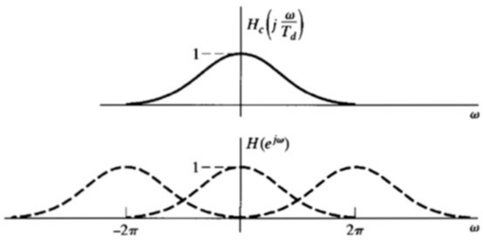
  ➔ Aliasing occurs.

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

- Aliasing in the impulse invariance transformation

$$if\ H_c(j\Omega)=0,\quad |\Omega|\geq \pi/T_d \qquad then\ H(e^{jw})=H_c\left(j\frac{w}{T_d}\right),\quad |w|\leq\pi$$

$$H(e^{jw})=\sum_{k=-\infty}^{\infty}H_c\left(j\frac{w}{T_d}+j\frac{2\pi}{T_d}k\right)$$



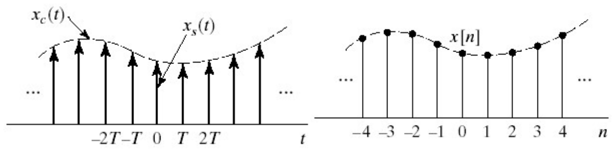
Periodic sampling

$$s(t)=\sum_{n=-\infty}^{\infty}\delta(t-nT)$$

T: sample period; fs=1/T: sampling rate
Ωs=2π/T: sample frequency

$$x_s(t)=x_c(t)s(t)=x_c(t)\sum_{n=-\infty}^{\infty}\delta(t-nT)=\sum_{n=-\infty}^{\infty}x_c(nT)\delta(t-nT)$$

$$x[n]=x_c(t)|_{t=nT}=x_c(nT)$$



- If the continuous-time filter approaches zero at high frequencies, the aliasing may be negligibly small. ➔ $\Omega=\omega/T_d$

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

- Transformation from continuous to discrete after obtaining the CW-time filter specifications from the specifications on $H(e^{j\omega})$.

$$H(s)=\sum_{k=1}^{N}\frac{A_k}{s-s_k} \qquad h_c(t)=\begin{cases}\sum_{k=1}^{N}A_k e^{s_k t}, & t\geq 0\\ 0, & t<0\end{cases}$$

$$h[n]=T_d h(t)\Big|_{t=nT_d}=T_d h_c(nT_d)=\sum_{k=1}^{N}T_d A_k e^{s_k\cdot nT_d}\cdot u[n]=\sum_{k=1}^{N}T_d A_k(e^{s_k T_d})^n\cdot u[n]$$

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

$$pole:\quad s=s_k\Rightarrow z=e^{s_k T_d}$$

two requirements for transformation

The real part of s_k less than zero, then the magnitude $z=\exp(s_k T_d)$ will be less than unity, the corresponding pole in the discrete-time filter is inside the unit circle.
➔ **Stable**

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

• Summary of Impulse Invariant Transformation

– Definition

$$h(n) = Th(nT)$$

$$\Rightarrow \quad H(s)|_{s=0} = \int_0^{\infty} h(t)dt, \quad H(z=1) = \sum_{n=0}^{\infty} h(n) = \sum_{n=0}^{\infty} Th(nT)$$

– Procedure

1. $H(s) \rightarrow h(t)$
2. $h[n] = T_d h(nT_d)$
3. $h[n] \rightarrow H(z)$

– High-pass filter cannot be transformed!!!

✓It is not a band-limited transfer function.

– Filter orders are not changed after transformation

Example) $H(s) = \alpha/(s+\alpha)$

$$\Rightarrow \quad h(t) = \alpha e^{-\alpha t} u(t)$$

$$\Rightarrow \quad h(n) = Th(nT) = \alpha T e^{-n\alpha T} u(n)$$

$$\therefore H(z) = \frac{\alpha T}{1 - e^{-\alpha T} z^{-1}}$$

$$\alpha > 0 \;\Rightarrow\; 0 < e^{-\alpha T} < 1$$

$$H(z=1) \approx 1 \text{ if } \alpha T \ll 1$$

• Example of Butterworth Filters

- **Specifications for the discrete-time filter:**

$$0.89125 \le |H(e^{jw})| \le 1, \quad 0 \le |w| \le 0.2\pi$$

$$|H(e^{jw})| \le 0.17783, \qquad 0.3\pi \le |w| \le \pi$$

$$let\ T_d = 1 \quad \Rightarrow \quad w = \Omega T_d = \Omega$$

Corresponding continuous time system function

$$0.89125 \le |H_c(j\Omega)| \le 1, \quad 0 \le |\Omega| \le 0.2\pi$$

$$|H_c(j\Omega)| \le 0.17783, \qquad 0.3\pi \le |\Omega|$$

Assume the effect of aliasing is negligible

$$|H_c(j0.2\pi)| \ge 0.89125$$

$$|H_c(j0.3\pi)| \le 0.17783$$

The magnitude-squared function of a Butterworth filter

$$|H_c(j\Omega)|^2 = \frac{1}{1 + (j\Omega/j\Omega_c)^{2N}}$$

-> the filter design process consists of determining the parameters N and Ω_c

$$1 + (j\Omega/j\Omega_c)^{2N} = \frac{1}{|H_c(j\Omega)|^2}$$

$$1 + \left(\frac{0.2\pi}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.89125}\right)^2$$

$$\left(\frac{0.2\pi}{\Omega_c}\right)^{2N} = 0.25893$$

$$1 + \left(\frac{0.3\pi}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.17783}\right)^2$$

$$\left(\frac{0.3\pi}{\Omega_c}\right)^{2N} = 30.62204$$

$$\frac{\left(\frac{0.3\pi}{\Omega_c}\right)^{2N} = 30.62204}{\left(\frac{0.2\pi}{\Omega_c}\right)^{2N} = 0.25893} = \left(\frac{3}{2}\right)^{2N} = 118.26378$$

$$\boxed{N = 5.8858, \quad \Omega_c = 0.70470}$$

$$\boxed{N = 6, \quad \Omega_c = 0.7032}$$

$$|H_c(j\Omega)|^2 = \frac{1}{1+(j\Omega/j\Omega_c)^{2N}} \Rightarrow |H_c(s)|^2 = H_c(s)H_c(-s) = \frac{1}{\underline{1+(s/j\Omega_c)^{2N}}} = 0$$

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

$$s_k = j\Omega_c(-1)^{1/2N} = \Omega_c e^{(j\pi/2N)(2k+N-1)}, \quad k = 0,1,\ldots,2N-1$$

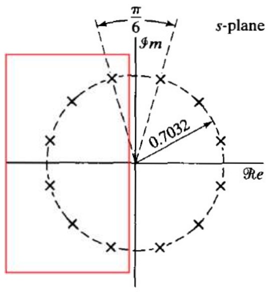
$$N = 6, \quad \Omega_c = 0.7032$$

$H_c(s)$ **Pole pairs:**

$-0.182 \pm j0.679,$

$-0.497 \pm j0.497,$

$-0.679 \pm j0.182$



12 poles of the magnitude-squared function $H_c(s)H_c(-s)$ are uniformly distributed in angle on a circle of radius $\Omega_c = 0.7032$

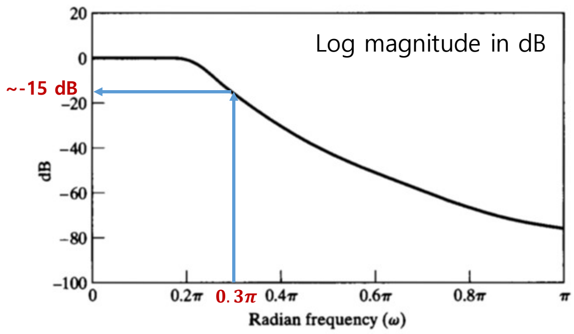
$$\Omega_c^N = 0.7032^6$$

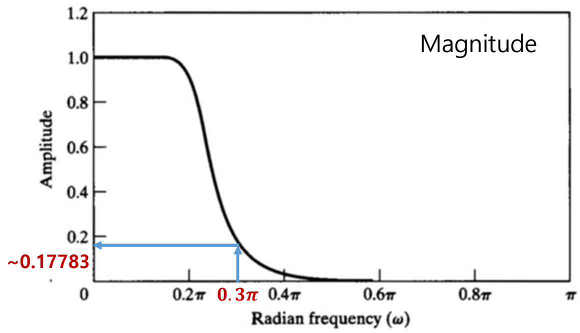
$$H_c(s) = \frac{0.12093}{(s+0.182-j0.679)(s+0.182+j0.679)(s+0.497-j0.497)} \times \frac{1}{(s+0.497+j0.497)(s+0.679-j0.182)(s+0.679+j0.182)}$$

$$H_c(s) = \frac{0.12093}{(s^2+0.3640s+0.4945)(s^2+0.9945s+0.4945)(s^2+1.3585s+0.4945)}$$

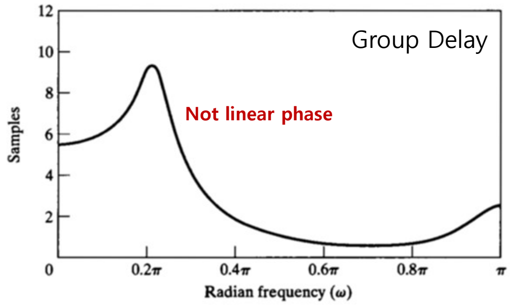
$$H(z) = \sum_{k=1}^{N} \frac{T_d A_k}{1-e^{s_k T_d} z^{-1}} = \sum_{k=1}^{N} \frac{A_k}{1-e^{s_k} z^{-1}} \qquad = \sum_{k=1}^{N} \frac{A_k}{s-s_k}$$

$$= \frac{0.2871-0.4466z^{-1}}{1-1.2971z^{-1}+0.6949z^{-2}} + \frac{-2.1428+1.1455z^{-1}}{1-1.0691z^{-1}+0.3699z^{-2}} + \frac{1.8557-0.6303z^{-1}}{1-0.9972z^{-1}+0.2570z^{-2}} \qquad T_d = 1$$





Frequency response of sixth-order Butterworth filter transformed by impulse invariance.



Basic for Impulse Invariance

- To chose an impulse response for the discrete-time filter that is similar to the impulse response of the continuous-time filter.
- If the continuous-time filter is bandlimited, the discrete-time filter frequency response will closely approximate the continuous-time frequency response.
- The relationship between continuous-time and discrete-time frequency is linear; consequently, except for aliasing, the shape of the frequency response is preserved.

Bilinear Transformation

- Bilinear transformation can avoid the problem of aliasing.
- Bilinear transformation maps $-\infty \le \Omega \le \infty$ onto $-\pi \le \Omega \le \pi$
- Restricted to situations in which the corresponding warping of the frequency axis is acceptable.

Bilinear transformation: $s = \frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$

$$H(z) = H_c\left[\frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right] = H_c[s]\Big|_{s=\frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

Impulse invariance: $z = e^{s_k T_d}$

$$s = \frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \qquad z = \frac{1+(T_d/2)s}{1-(T_d/2)s}$$

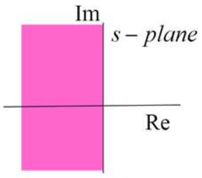
$$(T_d/2)s(1+z^{-1}) = 1 - z^{-1}$$

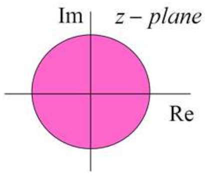
$$\{1+(T_d/2)s\}]z^{-1} = 1-(T_d/2)s$$

if $s = \sigma + j\Omega \qquad z = \dfrac{1+\sigma T_d/2 + j\Omega T_d/2}{1-\sigma T_d/2 - j\Omega T_d/2}$ $\qquad \sigma < 0 \;\Rightarrow\; |z| < 1 \;\; for \;\; any \;\; \Omega$, $\qquad \sigma > 0 \;\Rightarrow\; |z| > 1 \;\; for \;\; any \;\; \Omega$

if $j\Omega - axis \quad s = j\Omega \qquad z = \dfrac{1+j\Omega T_d/2}{1-j\Omega T_d/2} \;\Rightarrow\; |z| = 1 \qquad e^{jw} = \dfrac{1+j\Omega T_d/2}{1-j\Omega T_d/2}$

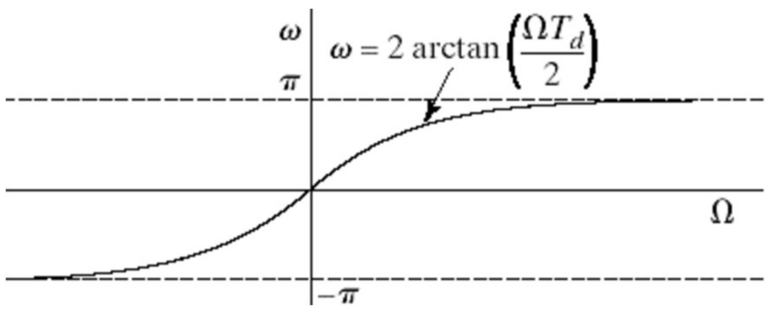
The axis maps onto the unit circle.



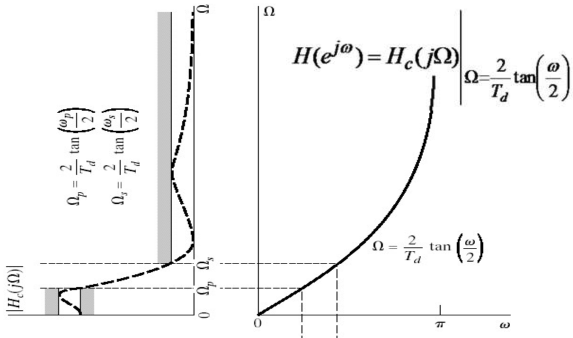


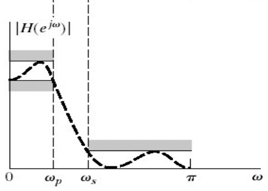
$$s = \frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \qquad j\Omega = \frac{2}{T_d}\left(\frac{1-e^{-jw}}{1+e^{-jw}}\right) = \frac{2}{T_d}\left(\frac{e^{-jw/2}(e^{jw/2}-e^{-jw/2})}{e^{-jw/2}(e^{jw/2}+e^{-jw/2})}\right) = \frac{2}{T_d}\left(\frac{2j\sin w/2}{2\cos w/2}\right) = j\frac{2}{T_d}\tan(w/2)$$

$$\Omega = \frac{2}{T_d}\tan(w/2) \qquad w = 2\tan^{-1}(\Omega T_d/2)$$



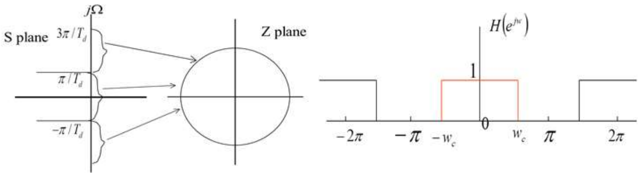
- Relationship between frequency response of $H_C(s)$ and $\mathrm{H}(z)$





$$\begin{cases} \Omega_p = \dfrac{2}{T_d}\tan\left(\dfrac{\omega_p}{2}\right) \\ \Omega_s = \dfrac{2}{T_d}\tan\left(\dfrac{\omega_s}{2}\right) \end{cases}$$

- It avoids the problem of aliasing encountered with the use of impulse invariance.
- It is nonlinear compression of frequency axis.
- The design of discrete-time filters using bilinear transformation is useful only when this compression can be tolerated or compensated for, as in the case of filters that approximate ideal piecewise-constant magnitude-response characteristics.



89

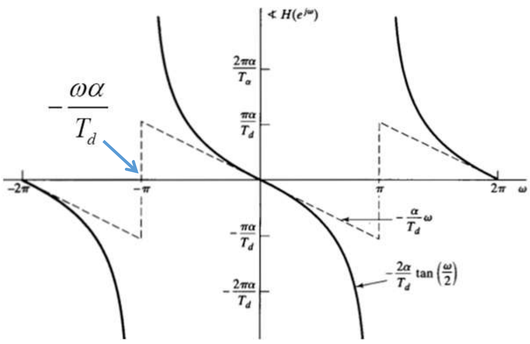
- Bilinear Transformation of an ideal linear-phase factor $e^{-s\alpha}$

$$s = \frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \qquad \Omega = \frac{2}{T_d}\tan(w/2) \qquad e^{-j\Omega\alpha}$$

$$-\Omega\alpha = -\frac{2\alpha}{T_d}\tan(w/2)$$

Effect of the bilinear transformation on a linear-phase characteristic

We cannot obtain a discrete-time lowpass filter with a linear-phase characteristic by applying the bilinear transformation to a continuous-time lowpass filter with a linear-phase characteristic



90

Comparisons of Impulse Invariance and Bilinear Transformation

- The use of bilinear transformation is restricted to the design of approximations to filters with piecewise-constant frequency magnitude characteristics, such as highpass, lowpass and bandpass filters.
- Impulse invariance can also design lowpass filters. However, it cannot be used to design highpass filters because they are not bandlimited.
- Bilinear transformation cannot design a filter whose magnitude response isn't piecewise constant, such as differentiator. However, Impulse invariance can design an bandlimited differentiator.

- Example of Butterworth Filter

Specifications for a DT filter

$$0.89125 \le \left|H\left(e^{jw}\right)\right| \le 1, \quad 0 \le w \le 0.2\pi$$
$$\left|H\left(e^{jw}\right)\right| \le 0.17783, \quad 0.3\pi \le w \le \pi$$

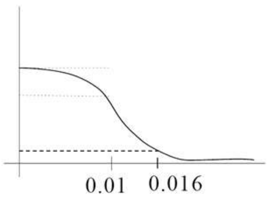
$$\Omega = \frac{2}{T_d}\tan(w/2)$$

Specifications for a CT filter

$$0.89125 \le \left|H_c(j\Omega)\right| \le 1, \quad 0 \le \Omega \le \frac{2}{T_d}\tan\left(\frac{0.2\pi}{2}\right)$$
$$\left|H_c(j\Omega)\right| \le 0.7783, \quad \frac{2}{T_d}\tan\left(\frac{0.3\pi}{2}\right) \le \Omega \le \infty$$

$$\textit{For convenience, we choose } T_d = 1$$

$$\Omega = \frac{\frac{2}{T_d}\tan\left(\frac{0.2\pi}{2}\right)}{\frac{2}{T_d}\tan\left(\frac{0.3\pi}{2}\right)} \qquad H(e^{j\omega}) = H_c(j\Omega)\Big|_{\Omega = \frac{2}{T_d}\tan\left(\frac{\omega}{2}\right)}$$



$$\left|H_c(j2\tan(0.1\pi))\right| \ge 0.89125,$$
$$\left|H_c(j2\tan(0.15\pi))\right| \le 0.17783,$$
$$\left|H_c(j\Omega)\right|^2 = \frac{1}{1+(j\Omega/j\Omega_c)^{2N}}$$

$$1+\left(\frac{2\tan(0.1\pi)}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.89125}\right)^2$$
$$1+\left(\frac{2\tan(0.15\pi)}{\Omega_c}\right)^{2N} = \left(\frac{1}{0.17783}\right)^2$$

$$N = 5.305$$
$$N = 6,$$
$$\Omega_c = 0.766$$

Locations of Poles

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

$$s_k = j\Omega_c(-1)^{1/2N} = \Omega_c e^{(j\pi/2N)(2k+N-1)}, \quad k = 0,1,\ldots,2N-1$$

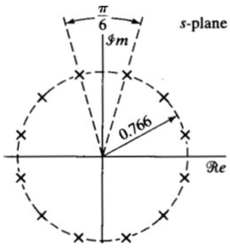
$$N = 6, \Omega_C = 0.7662$$

$H_c(s)$ **Plole pairs:**

$-0.1983 \pm j0.7401$

$-0.5418 \pm j0.5418$

$-0.7401 \pm j0.1983$



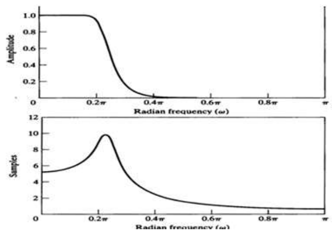
$$|H_c(s)|^2 = H_c(s)H_c(-s) = \frac{1}{1+(s/j\Omega_c)^{2N}} = \frac{\Omega_c^{2N}}{s^{2N}+\Omega_c^{2N}}$$

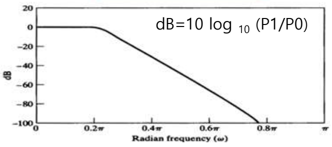
$$H_c(s) = \frac{0.20238}{(s^2+\cancel{0.3996s}+0.5871)(s^2+1.0836s+0.5871)(s^2+1.4802s+0.5871)}$$

0.3966s

$$H(z) = \frac{0.0007378(1+z^{-1})^6}{(1-1.2686z^{-1}+0.7051z^{-2})(1-1.0106z^{-1}+0.3583z^{-2})} \times \frac{1}{(1-0.9904z^{-1}+0.2155z^{-2})}$$

$$\boxed{s = \frac{2}{T_d}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$





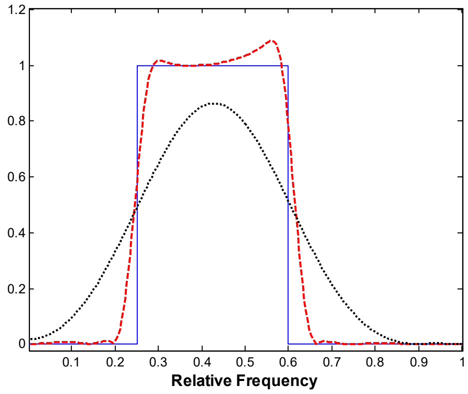
The magnitude response falls off much rapidly than that of the original continuous-time filter due to the frequency warping.

IIR Filter Design in MATLAB

- The Yule-Walker recursive filter
  - Using a least-squares fit into a specified frequency response.
  - The only IIR filter that is not supported by an order-selection routine.
  - This is the IIR counterpart of the remez() and fir2() functions used for FIR filter design.
  - [b, a] = yulewalk (n, f, m)
  - n is the filter order
  - m and f specify the desired frequency characteristic in a fairly straightforward way.
    - ✓ m is a vector of the desired filter gains at the frequencies specified in f.
    - ✓ The frequencies in f are relative to fs/2: the first point in f must be zero and the last point 1.

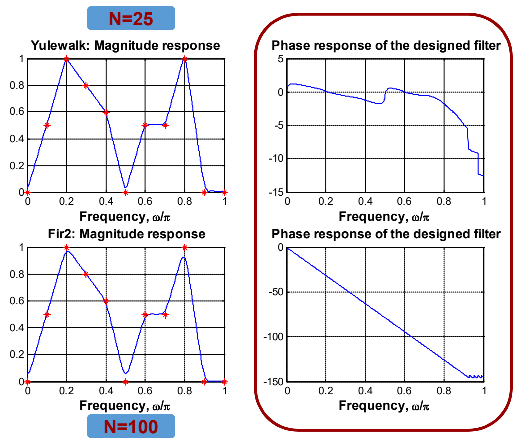
- Example 1
  - Design a 12th-order Yule-Walker bandpass filter with cutoff frequencies of 0.25 and 0.5.

```
n = 12;                    % Filter order
f = [0 .25 .25 .6 .6 1];   % Specify desired frequency response
m = [0 0 1 1 0 0];
[b,a] = yulewalk(n,f,m);
h = freqz(b,a,256);
b1 = fir2(n,f,m);
h1 = freqz(b1,1,256);
plot(f,m,'b');
hold on;
w = (1:256)/256;
plot(w,abs(h),'--r');
plot(w,abs(h1),':k');
xlabel('Relative Frequency');
```



- Example 2

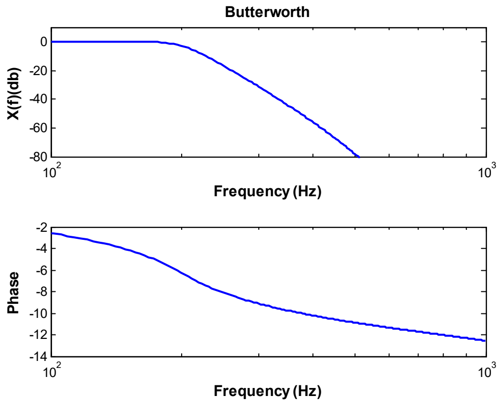
```
freq=[0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]
amp=[0 0.5 1 0.8 0.6 0 0.5 0.5 1 0 0]
[b1 a1]=yulewalk(25, freq, amp);
b2=fir2(100, freq, amp, 1024);
subplot(221)
[H1 w]=freqz(b1, a1, 1024);
[H2 w]=freqz(b2, 1, 1024);
plot(w/pi, abs(H1)); hold on
plot(freq, amp, 'r*'); grid
xlabel('Frequency, \omega/\pi')
title(' Yulewalk: Magnitude response ')
subplot(222)
plot(w/pi, unwrap(angle(H1))); grid
xlabel('Frequency, \omega/\pi')
title(' Phase response of the designed filter')
subplot(223)
plot(w/pi, abs(H2)); hold on
plot(freq, amp, 'r*'); grid
xlabel('Frequency, \omega/\pi')
title(' Fir2: Magnitude response')
subplot(224)
plot(w/pi, unwrap(angle(H2))); grid
xlabel('Frequency, \omega/\pi')
title(' Phase response of the designed filter')
```



- Butterworth Filter
  - **[n, wn] = buttord(wp, ws, rp, rs);**
  - Butterworth filter order selection
  - wp is the passband frequency relative to fs/2
  - ws is the stopband frequency in the same units
    - ✓For HPF, wp is greater than ws. For BPF and BSF, wp and ws are two-element vector that specify the corner frequencies at both edges of filter, the lower frequency edge first.
  - rp is the passband ripple in dB
  - rs is the stopband ripple in dB
  - Since the Butterworth filter does not have ripple in either the passband or stopband, rp is the maximum attenuation in the passband and rs is the minimum attenuation in the stopband.
  - n is the required filter order
  - wn is the actual -3dB cutoff frequency.

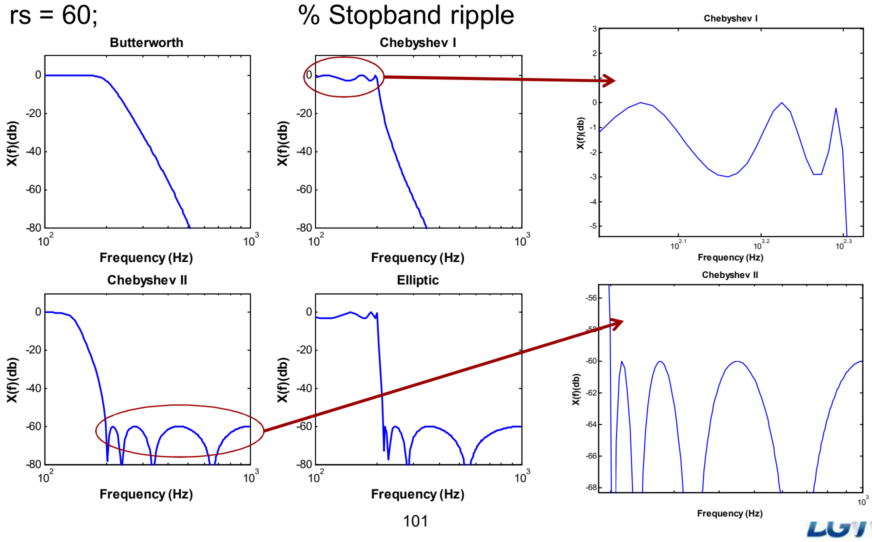
- **[b, a] = butter (n, wn, 'ftype);**
- n and wn are the order and cutoff frequencies, respectively.
  - ✓The cutoff frequencies wn must be 0<wn<1 with 1 corresponding to half the sample rate.
- The argument 'ftype' should be 'high' if a highpass filter is desired and 'stop' for a bandstop filter with wn=[w1 w2]. w1<w2
- To specify a BPF, use a two-element vector without the 'ftype' argument.
- b (numerator) and a (denominator) coefficients are used in routines 'filter' or 'filtfilt', or in 'freqz' for plotting the frequency response. The coefficients are listed in descending powers of z.
- butter(n, wn, 's'), u=butter(n, wn, 'high', 's'), and butter(n, wn, 'stop', 's') design analog Butterworth filters. In this case, wn is in [rad/s] and it can be greater than 1.

```
fs = 2000;              % Sampling filter
n = 8;                  % Filter order
wn = 200/1000;          % Filter cutoff frequency
rp = 3;                 % Maximum passband ripple
rs = 60;                % Stopband ripple

% Determine filter coefficients
[b,a] = butter(n,wn);
[h,f] = freqz(b,a,256,fs);
h_dB = 20*log10(abs(h));
subplot(2,1,1)
semilogx(f,h_dB);
axis([100 1000 -80 10]);
title('Butterworth');
ylabel('X(f)(db)');
xlabel('Frequency (Hz)');
subplot(2,1,2)
semilogx(f, unwrap(angle(h)));
axis([100 1000 -14 -2]);
ylabel('Phase');
xlabel('Frequency (Hz)');
```



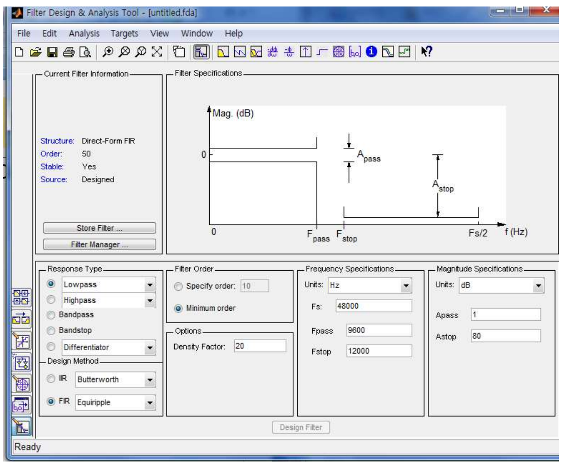
- **[n, wn] = chel1ord(wp, ws, rp, rs);**
  - Chebyshev Type I filter order selection
- **[n, wn] = chel2ord(wp, ws, rp, rs);**
  - Chebyshev Type II filter order selection
- The arguments are the same as in buttord()
- **[b, a] = cheb1 (n, rp, wn, ‘ftype’);**
  - The arguments are the same as in butter for the additional argument, rp, which specifies the maximum desired passband ripple in dB.
- **[b, a] = cheb2 (n, rs, wn, ‘ftype’);**
  - It has an argument, rs, which specifies the stopband ripple in dB.
- **[b, a] = ellip (n, rp, rs, wn, ‘ftype’);**
  - It includes both rp and rs.

```
fs = 2000;          % Sampling filter
n = 8;              % Filter order
wn = 200/1000;      % Filter cutoff frequency
rp = 3;             % Maximum passband ripple
rs = 60;            % Stopband ripple
```

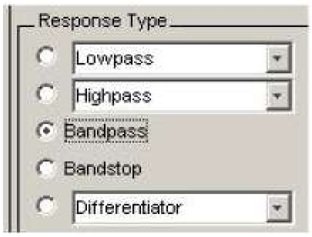


FDA Tool in MATLAB

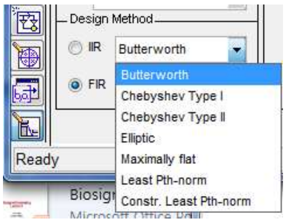
- Filter Design and Analysis (FDA) Tool supporting easy design of filters in MATLAB.
  - To open the FDA tool
    - ✓ fdatool

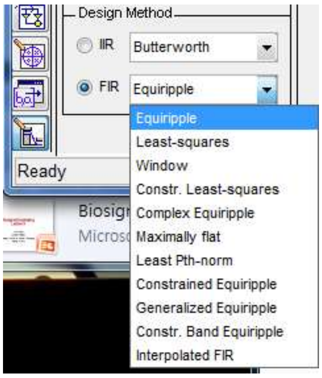


- Choose a filter type
  - LPF, HPF, BPF, BSF
  - Differentiator
  - Hilbert transformer
  - Multiband
  - Arbitrary magnitude

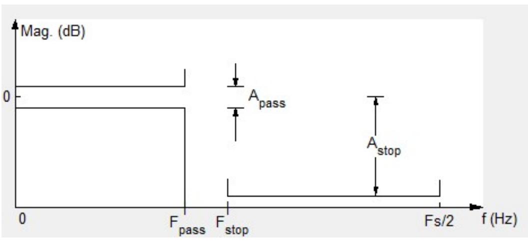


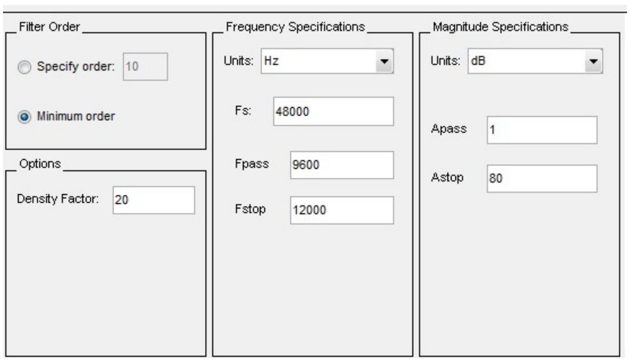
- Choose a filter design method



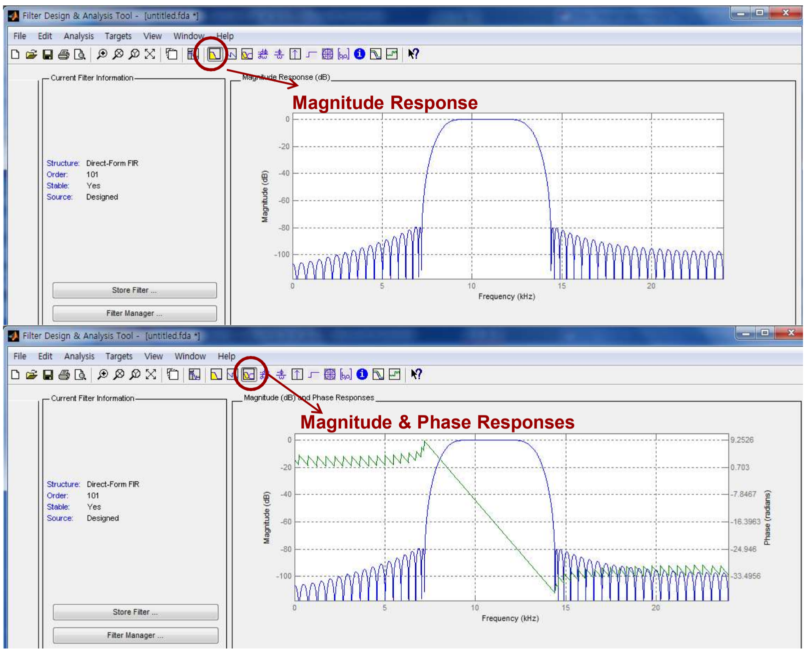


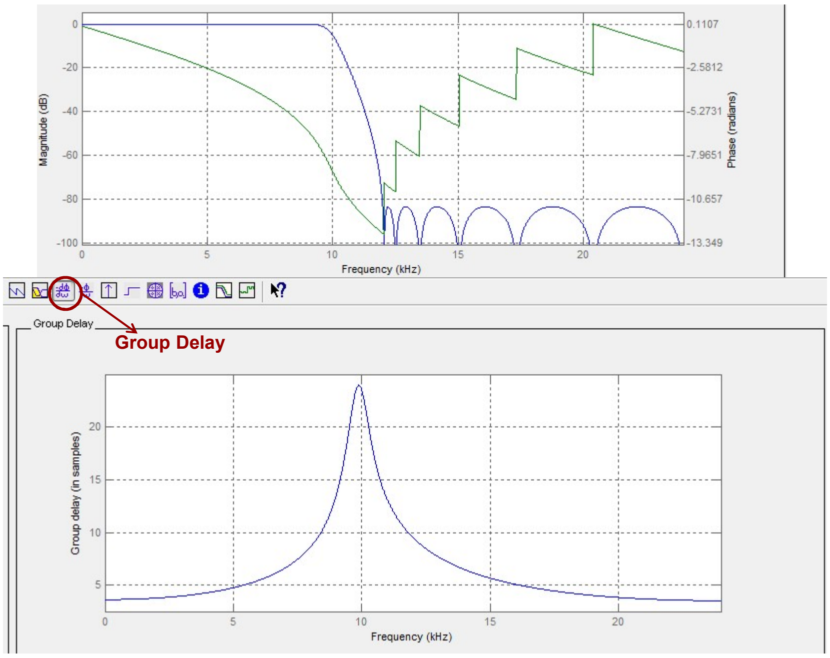
- Setting the filter design specification





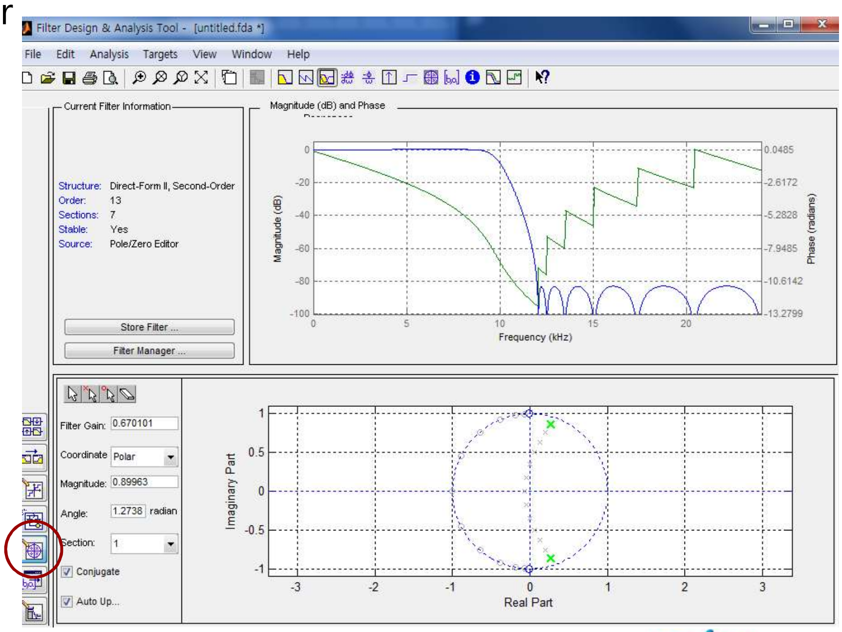
- Analyzing the filter
  - Magnitude response
  - Phase response
  - Magnitude and Phase responses
  - Group delay response
  - Phase delay response
  - Impulse response
  - Step response
  - Pole-zero plot
  - Zero-phase response





DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

- Pole-Zero Editor



DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

- Exporting a filter design
  - Export coefficients or objects to the workspace
    - ✓File>Export