CTP 431 Music and Audio Computing

Sound Processing and Digital Filters

Graduate School of Culture Technology (GSCT)
Juhan Nam

Outlines

- Introduction: Sound Processing
- Linear Time-Invariant Digital Filter
  - Impulse response
  - Convolution
- Digital Filters
  - FIR Filters
  - IIR Filters
- Frequency Response
- Transfer functions
  - Z-transform
  - Pole-Zero Analysis
- Bi-quad Filters

Introduction: Sound Processing

- Sounds captured on computers are processed in various ways
  - Sample editing: cut, copy, paste
  - Amplitude: gain, fade in/out, automation curve, compressor
  - Timbre: lowpass/highpass filters, EQ, distortion, modulation
  - Spatial effect: delay, reverberation
  - Pitch: pitch shifting (e.g. auto-tune)
  - Time stretching
  - Noise suppression

Sound Processing

$$\text{Input} \longrightarrow \boxed{\text{Processor}} \longrightarrow \text{Output}$$

- Linear processing
  - No sinusoidal components are introduced by the processing
    - Only the amplitude and phase of sinusoidal components of the input change
  - Filters (lowpass, highpass, bandpass, …), EQ
  - Delay-based audio effect: delay, chorus, flanger, reverberation

- Non-linear processing
  - New sinusoidal components are generated
  - Compressor, distortion, clipping
  - Pitch shifting, ring modulation, …

Linear Time-Invariant (LTI) Digital Filters

$$\underset{x(n)}{\text{Input}} \longrightarrow \boxed{\text{Filter}} \longrightarrow \underset{y(n)}{\text{Output}}$$

- What linear filters can do
  - Amplifying the input (or the past output): e.g. $y(n) = ax(n)$
  - Delaying the input (or the past output): e.g. $y(n) = x(n\text{-}1)$
  - Summing them all: $y(n) = ax(n) + x(n\text{-}1)$
- "Easy-to-understand" definition

LTI Digital Filters (Formal)

Input $\longrightarrow$ Filter $\longrightarrow$ Output

$x(n)$ $\qquad\qquad\qquad\qquad$ $y(n)$

- Linearity
  - Homogeneity: if $x(n) \rightarrow y(n)$, $ax(n) \rightarrow ay(n)$
  - Superposition: if $x_1(n) \rightarrow y_1(n)$ and $x_2(n) \rightarrow y_2(n)$, $x_1(n) + x_2(n) \rightarrow y_1(n) + y_2(n)$

- Time-Invariance
  - If $x(n) \rightarrow y(n)$, $x(n\text{-}N) \rightarrow y(n\text{-}N)$ for any N
  - The system does not change its behavior with time.
    - In practice, most systems do change over time but not quickly

Example: Simple LTI Digital Filters

- Moving-average filter
  - $y(n) = 0.5x(n) + 0.5x(n\text{-}1)$
  - Low-pass
- Differentiator
  - $y(n) = 0.5x(n) - 0.5x(n\text{-}1)$
  - High-pass
- Feed-forward comb filter
  - $y(n) = x(n) + x(n\text{-}M)$ where M is, say,100
  - Renders harmonically distributed peaks and valleys

Impulse Response

Input → Filter → Output

$$x(n) = \delta(n) \qquad h(n) \qquad y(n) = h(n)$$

- Characterize filters as a number sequence
- Obtained when *x(n)* is a unit impulse
  - $x(n) = \delta(n) = [1, 0, 0, 0, \ldots] \rightarrow y(n) = h(n)$
- Can be measured from a linear system (black-box approach)
  - If you excite the linear system with an impulse, you can record the output and use that to determine exactly what the system response would be to any arbitrary input.

Convolution

- The output of LTI systems is represented by convolution operation between the input $x(n)$ and impulse response $h(n)$

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

- Deriving convolution
  - The input is decomposed into a time-ordered set of weighted impulse
    - $x(n) = [x_0, x_1, x_2, x_3, \ldots, x_M]$
      $= x_0\delta(n) + x_1\delta(n-1) + x_2\delta(n-2) + x_3\delta(n-3) + \ldots + x_M\delta(n-M)$
  - By the linearity and time-invariance
    - $y(n) = x_0 h(n) + x_1 h(n-1) + x_2 h(n-2) + x_3 h(n-3) + \ldots + x_M h(n-M)$

Convolution in Practice

- From the commutative law

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

- $y(n) = h_0x(n) + h_1x(n\text{-}1) + h_2x(n\text{-}2) + h_3x(n\text{-}3) + \ldots + h_Mx(n\text{-}M)$
- For example: $x(n)\ (n=0,1,2,3,4,5)$, $h(n)$ = [$h_0\ h_1\ h_2$]
  - $y(0) = h_0x(0)$
  - $y(1) = h_0x(1) + h_1x(0)$

    Transient region
  - $y(2) = h_0x(2) + h_1x(1) + h_2x(0)$
  - $y(3) = h_0x(3) + h_1x(2) + h_2x\ (1)$
  - $y(4) = h_0x(4) + h_1x(3) + h_2x\ (2)$
  - $y(5) = h_0x(5) + h_1x(4) + h_2x\ (3)$

    Fully overlapped region (steady-state)
  - $y(6) = \qquad h_1x(5) + h_2x\ (4)$
  - $y(7) = \qquad h_2x\ (5)$

    Transient region

Convolution in Practice

- If the length of $x(n)$ is M and the length of $h(n)$ is N, the length of $y(n)$ is M+N-1

- Computation Complexity
  – In Big-O notation, it requires M x N multiplications
  – If N is a large number, it is quite expensive to compute
  – We can compute convolution in frequency domain, which is much cheaper than the time-domain approach when the impulse response is long (e.g. reverberation or head-related transfer functions)

Properties of LTI systems

- Commutative

$$x(n) * h_1(n) * h_2(n) = x(n) * h_2(n) * h_1(n)$$

- Associative

$$\{x(n) * h_1(n)\} * h_2(n) = x(n) * \{h_1(n) * h_2(n)\}$$

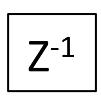
- Distributive

$$x(n) * h_1(n) + x(n) * h_2(n) = x(n) * \{h_1(n) + h_2(n)\}$$

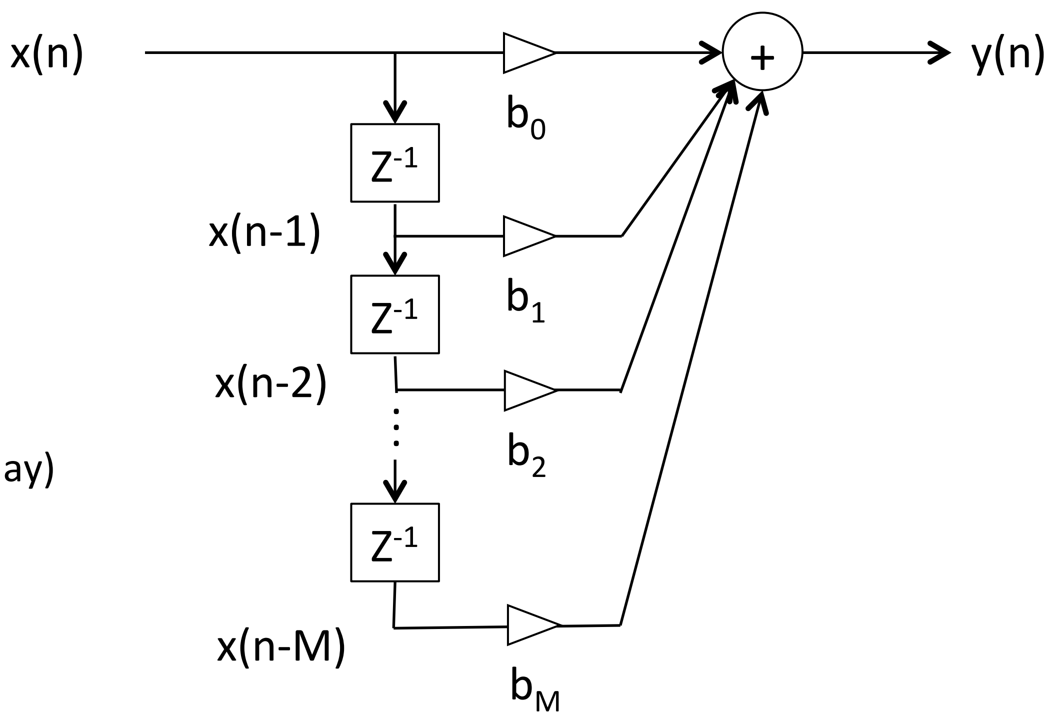
FIR Filters

- The output is formed from input and its past input
  - They have finite impulse responses (FIR)
  - Convolution with the finite impulse responses

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \ldots + b_M x(n-M)$$



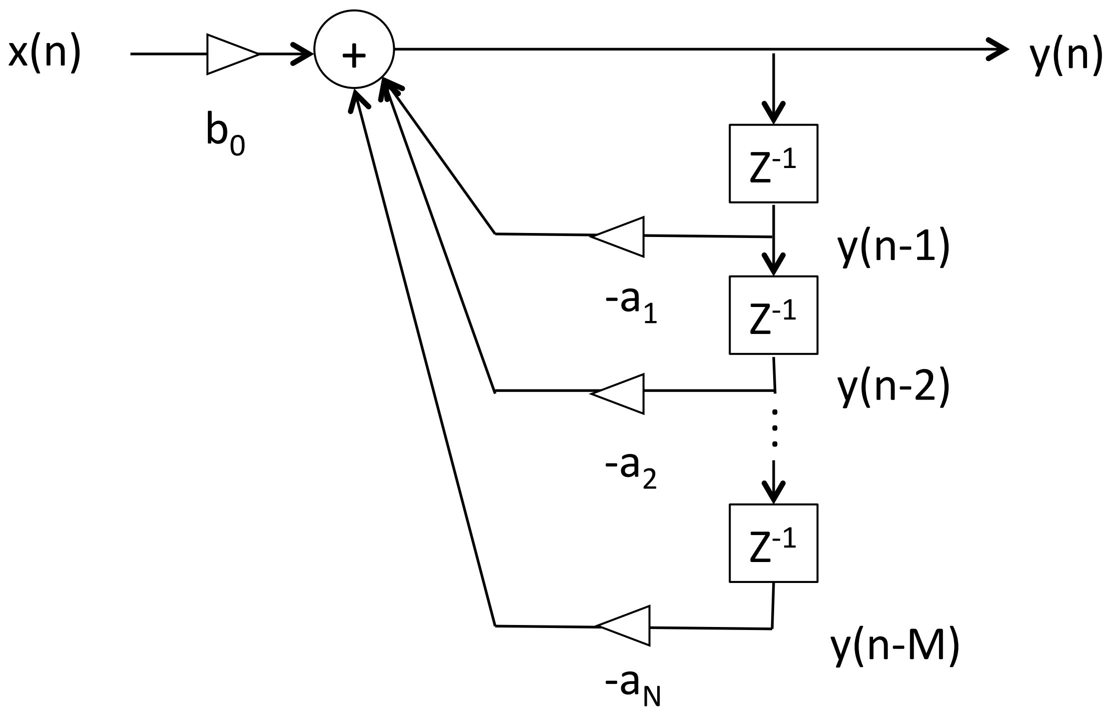
“Delay operator”
(a unit sample of delay)



IIR Filters

- The output can be also formed by input and past outputs
  - The feedback creates an infinite impulse response!
  - Convolution with the infinite impulse responses

$$y(n) = b_0 x(n) - a_1 y(n-1) - a_2 y(n-2) - \ldots - b_N y(n-N)$$



IIR Filters

- The infinite impulse response
  - For example: $y(n) = x(n) + ry(n\text{-}1)$
    - $y(0) = x(0)$
    - $y(1) = x(1) + ry(0) = x(1) + rx(0)$
    - $y(2) = x(2) + ry(1) = x(2) + rx(1) + r^2x(0)$
    - $y(3) = x(3) + ry(2) = x(3) + rx(2) + r^2x(1) + r^3x(0)$
    - $y(4) = x(4) + ry(3) = x(4) + rx(3) + r^2x(2) + r^3x(1) + r^4x(0)$

$$\rightarrow h(n) = [1\ r\ r^2\ r^3\ r^4\ r^5\ r^6\ \ldots]$$

  - Stability issue!
    - If $r < 1$, the filter becomes stable
    - If $r = 1$, the filter oscillates
    - If $r > 1$, the filter becomes unstable
  - The impulse response is long but, in practice, it is finite (for $r < 1$) because the level goes below the the quantization noise floor

Example: IIR Filters

- Leaky Integrator
  - $y(n) = x(n) + ry(n-1)$
  - r is a slightly less than 1. (1-r) is the "leak"
  - Lowpass filtering

- "Reson" filter
  - $y(n) = x(n) + 2r\cos\theta\, y(n-1) - r^2 y(n-2)$
  - r controls the resonance and θ controls its frequency
    - Resonance: 0 (low resonance) $< r <$ 1 (low resonance)
    - Cut-off frequency (f_c): $\theta = 2\pi f_c / f_s$ (f_s: sampling rate)
  - Low-pass/band-pass/high-pass filtering depending on additional zeros

- Feed-back comb filter
  - $y(n) = x(n) + ry(n-M)$
  - Renders a harmonic tone

KAIST

CT

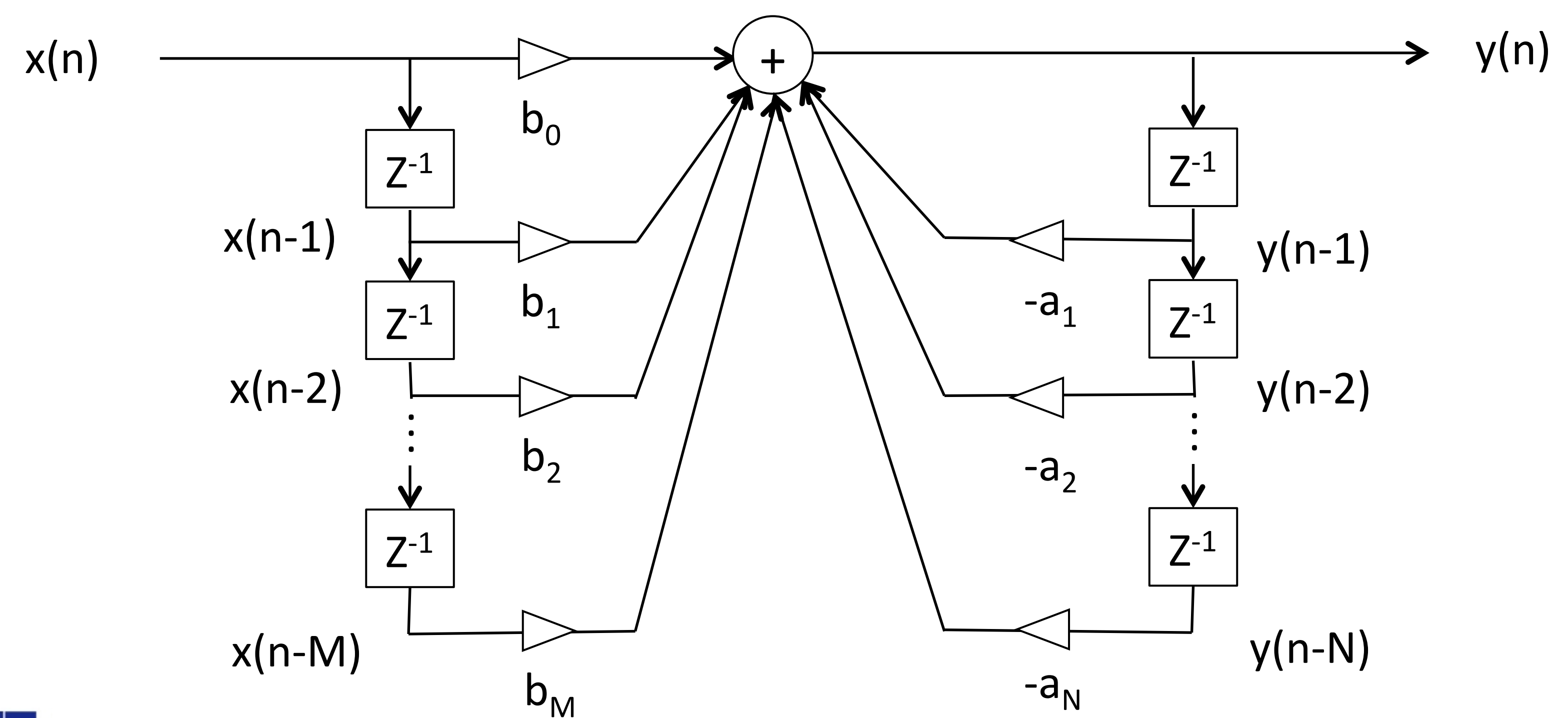
General Filter Form

- The general form of digital Filters

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) + \ldots + b_M x(n-M)$$
$$- a_1 y(n-1) - a_2 y(n-2) - \ldots - a_N y(n-N)$$

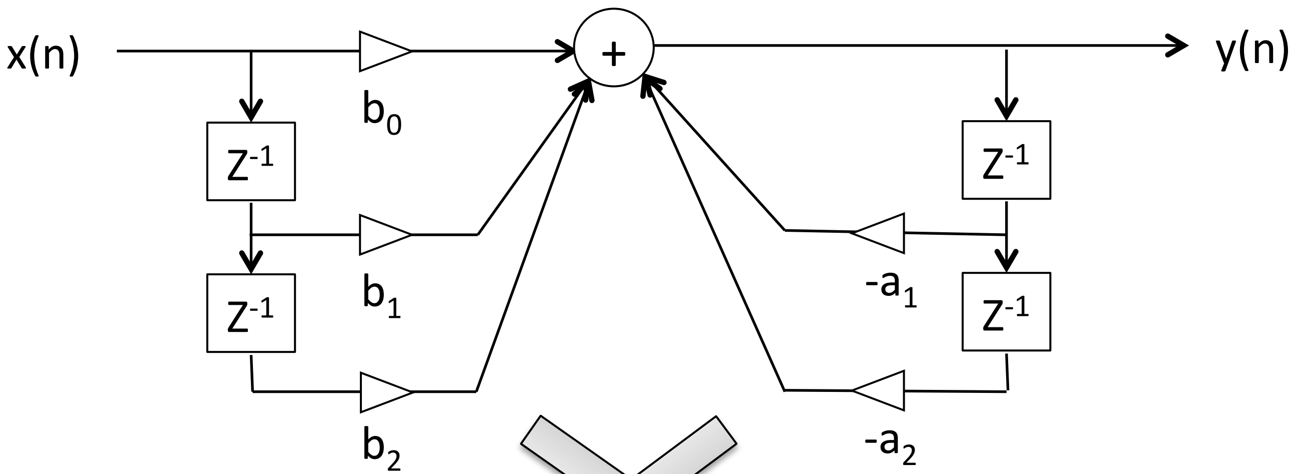
"Difference Equation"

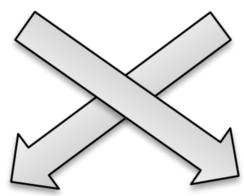
  – The order of the filter is the greater of M or N



Filter Implementation Forms

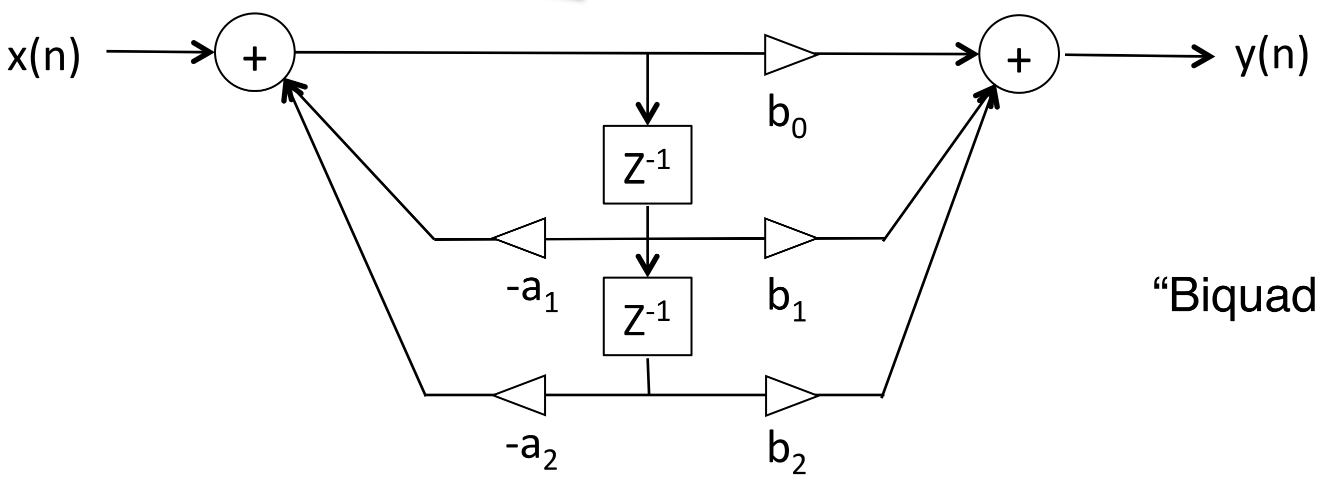
- Direct Form I





(By the commutative rule)

- Direct Form II



"Biquad Filter"

Example of Filter Implementation

- Typically implemented in time-domain

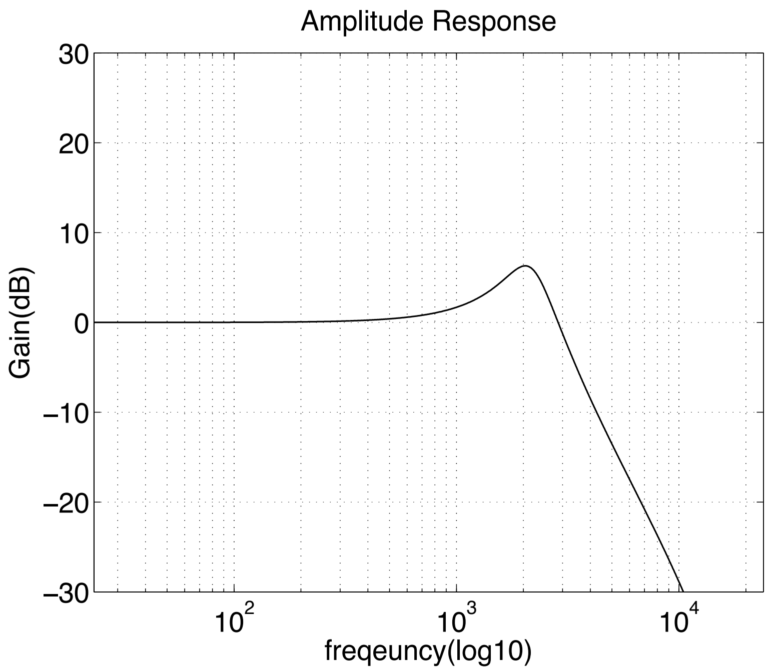
```
x = audioread('my_sound.wav');

% delay elements
xz1 =0; xz2 =0;
yz1 =0; yz2 =0;

% output
y = zeros(length(x),1);

% Direct Form I
for i=1:length(x)
    y(i) = (b0*x(i) + b1*xz1 + b2*xz2 - a1*yz1 - a2*yz2)/a0;
    xz2 = xz1;
    xz1 = x(i);
    yz2 = yz1;
    yz1 = y(i);
end
```

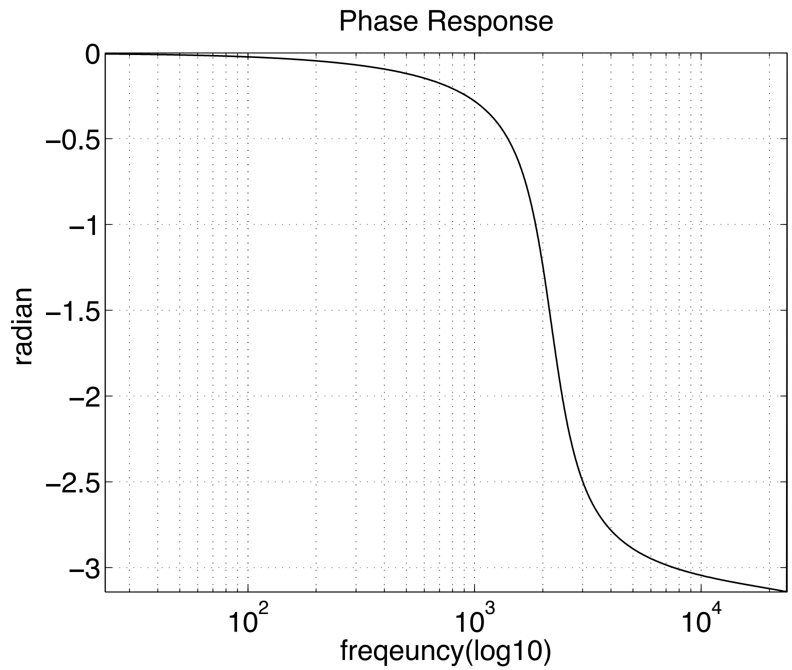
```
B = [b0 b1 b2];
A = [a0 a1 a2];

y = filter(B,A,x);
```

A short version
using “filter” function in Matlab

Frequency Responses

- Describe the characteristics of filters in frequency domain
  - Amplitude response: the amount of amplitude change (often in dB)
  - Phase response: the amount of delay (-2pi ~ 0)



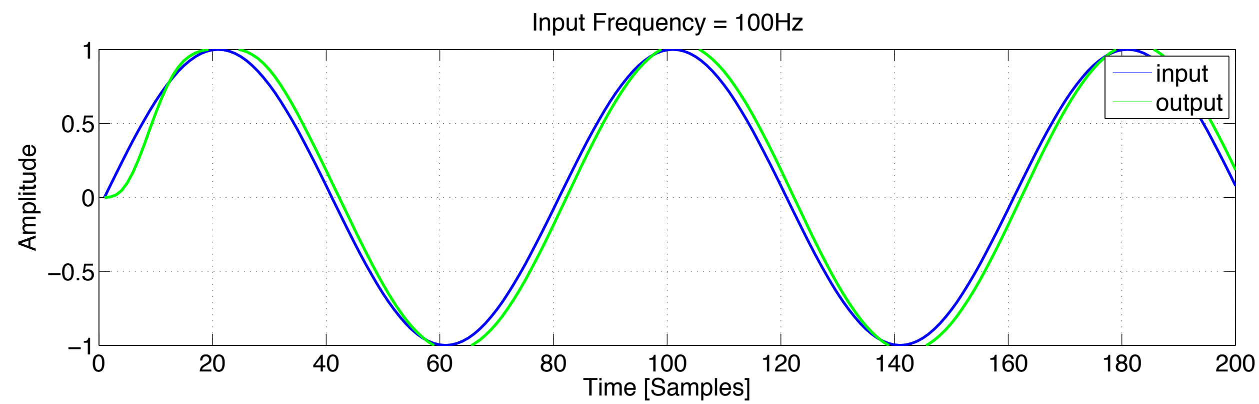


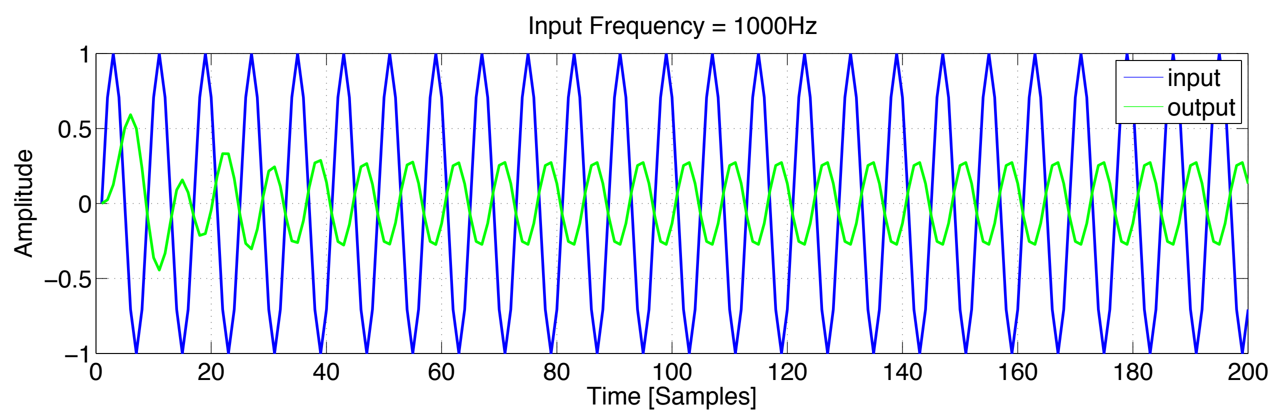
Frequency Responses

Input → Filter → Output

$$x(n) = e^{j\omega n}$$

$$y(n) = G(\omega)e^{j(\omega + \Theta(\omega))}$$





Frequency Response

- For the sinusoidal input and outputs
  - $x(n) = e^{j\omega n} \rightarrow x(n\text{-}m) = e^{j\omega(n\text{-}m)} = e^{\text{-}j\omega m}\, x(n)$ for any m
  - $y(n) = G(\omega)e^{j(\omega n+\Theta(\omega))} \rightarrow y(n\text{-}m) = G(\omega)e^{j(\omega(n\text{-}m)+\Theta(\omega))} = e^{\text{-}j\omega m}\, y(n)$ for any m

- Putting this property into the general form of difference equation

$$y(n) = b_0 x(n) + b_1 e^{-j\omega} x(n) + b_2 e^{-j2\omega} x(n) + \ldots + b_M e^{-jM\omega} x(n)$$
$$- a_1 e^{-j\omega} y(n) - a_2 e^{-j2\omega} y(n) - \ldots - a_N e^{-jN\omega} y(n)$$

$$y(n) = \frac{b_0 + b_1 e^{-j\omega} + b_2 e^{-j2\omega} + \ldots + b_M e^{-jM\omega}}{1 + a_1 e^{-j\omega} + a_2 e^{-j2\omega} + \ldots + a_N e^{-jN\omega}} x(n)$$

$$H(\omega) = \frac{b_0 + b_1 e^{-j\omega} + b_2 e^{-j2\omega} + \ldots + b_M e^{-jM\omega}}{1 + a_1 e^{-j\omega} + a_2 e^{-j2\omega} + \ldots + a_N e^{-jN\omega}}$$

$H(\omega)$: frequency response

$|H(\omega)| = G(\omega)$: amplitude response

$\angle H(\omega) = \Theta(\omega)$:phase response

Examples of Frequency Response

- Moving-average filter (lowpass)
  - $y(n) = 0.5(x(n) + x(n\text{-}1))$
  - $H(\omega) = 0.5(1+ e^{-j\omega}) = 0.5(e^{j\omega/2}+ e^{-j\omega/2})e^{-j\omega/2} = \cos(\omega/2)\ e^{-j\omega/2}$
  - $G(\omega) = \cos(\omega/2),\ \ \Theta(\omega) = -\omega/2$

- Differentiator (highpass)
  - $y(n) = 0.5(x(n) + x(n\text{-}1))$
  - $H(\omega) = 0.5(1- e^{-j\omega}) = 0.5(e^{j\omega/2}- e^{-j\omega/2})e^{-j\omega/2} = \sin(\omega/2)\ e^{-j\omega/2+j\pi/2}$
  - $G(\omega) = \sin(\omega/2),\ \ \Theta(\omega) = -\omega/2+\pi/2$

Z-Transform

- Z-transform
  - Define z to be a variable in complex plane: we call it z-plane
  - When $z = e^{j\omega}$ (on unit circle), the frequency response is a particular case of the following

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

  - We call this Z-transform of h(n) or transfer function
  - z^{-1} corresponds to one sample delay:
    - Called delay operator or delay element
  - Filters are often expressed as Z-transform

Poles and Zeros

- $H(z)$ can be factorized and we can find roots for each of polynomials

$$H(z)=\frac{B(z)}{A(z)}=\frac{(1-q_1z^{-1})(1-q_2z^{-1})(1-q_3z^{-1})\ldots(1-q_Mz^{-1})}{(1-p_1z^{-1})(1-p_2z^{-1})(1-p_3z^{-1})\ldots(1-p_Nz^{-1})}$$

  - Zeros: the numerator roots
  - Poles: the denominator roots

- We can analyze frequency response of filters more easily with poles and zeros than numerator or denominator coefficient!!!

Pole-Zero Analysis: Amplitude Response

- The amplitude response is represented as

$$G(\omega) = \left|H(z = e^{j\omega})\right| = \left|\frac{(1-q_1e^{-j\omega})(1-q_2e^{-j\omega})(1-q_3e^{-j\omega})\ldots(1-q_Me^{-j\omega})}{(1-p_1e^{-j\omega})(1-p_2e^{-j\omega})(1-p_3e^{-j\omega})\ldots(1-p_Ne^{-j\omega})}\right|$$

$$= \left|\frac{(e^{j\omega}-q_1)(e^{j\omega}-q_2)(e^{j\omega}-q_3)\ldots(e^{j\omega}-q_M)}{(e^{j\omega}-p_1)(e^{j\omega}-p_2)(e^{j\omega}-p_3)\ldots(e^{j\omega}-p_N)}\right|$$

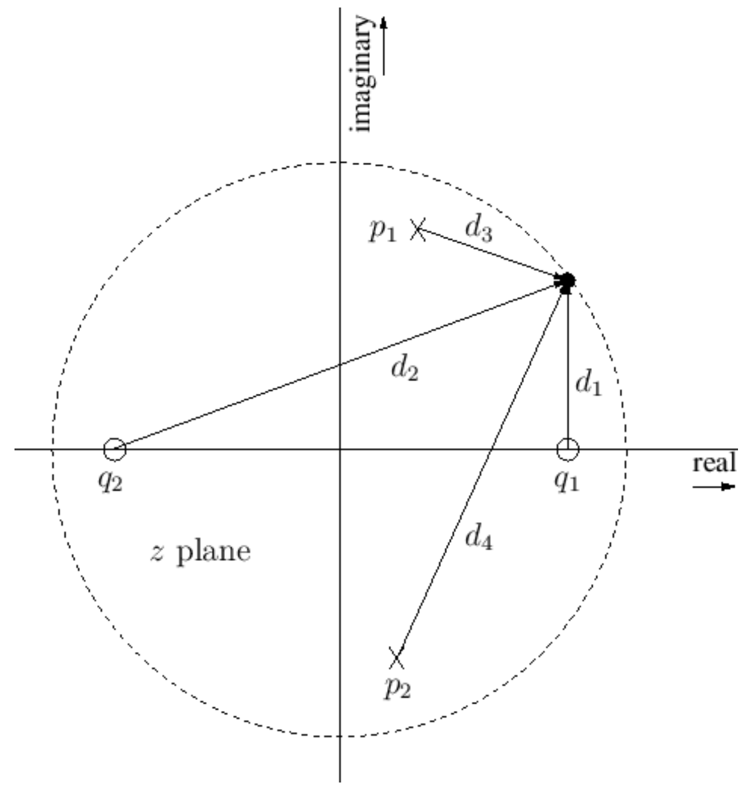
$$= \frac{\left|(e^{j\omega}-q_1)\right|\left|(e^{j\omega}-q_2)\right|\left|(e^{j\omega}-q_3)\right|\ldots\left|(e^{j\omega}-q_M)\right|}{\left|(e^{j\omega}-p_1)\right|\left|(e^{j\omega}-p_2)\right|\left|(e^{j\omega}-p_3)\right|\ldots\left|(e^{j\omega}-p_N)\right|}$$

  - Numerator: factors of distance between zero and unit circle
  - Denominator: factors of distance between pole and unit circle

Pole-Zero Analysis: Amplitude Response

- Bi-quad case
  - The amplitude response is given as

$$G(\omega) = \frac{d_1(\omega) d_2(\omega)}{d_3(\omega) d_4(\omega)}$$



  - As poles are close to the unit circle, the amplitude response is boosted
  - As zeros are close to the unit circles, the amplitude response is damped
- If poles are outside the unit circle, the filter becomes unstable!
  - If poles are on the unit circles, the filter oscillates.

Examples

- Moving-average filter (lowpass)
  - $y(n) = 0.5(x(n) + x(n\text{-}1))$
  - zeros: $z = -1$ (no poles)

- Leaky Integrator
  - $y(n) = x(n) + \mathrm{r}y(n\text{-}1)$
  - poles: $z = -\mathrm{r}$ (no zeros)

- Reson filter
  - $y(n) = x(n) + 2\mathrm{r}\cos\theta\, y(n\text{-}1) - \mathrm{r}^2 y(n\text{-}2)$
  - poles: $z = \mathrm{r}(\cos\theta + \mathrm{j}\sin\theta), \mathrm{r}(\cos\theta - \mathrm{j}\sin\theta)$ (no zeros)

- Feed-back comb filter
  - $y(n) = x(n) - ry(n\text{-}\mathrm{M})$ (for convenience, the sign of r has changed)
  - poles: $z = r^{1/M}\,(\cos(2\pi/Mn) + \mathrm{j}\sin(2\pi/Mn))$ $(n=0, 1, 2, \ldots M\text{-}1)$ (no zeros)

Pole-Zero Analysis: Phase Response

- The phase response is represented as

$$\Theta(\omega) = \angle H(z = e^{j\omega}) = \angle \frac{(1 - q_1 e^{-j\omega})(1 - q_2 e^{-j\omega})(1 - q_3 e^{-j\omega})\ldots(1 - q_M e^{-j\omega})}{(1 - p_1 e^{-j\omega})(1 - p_2 e^{-j\omega})(1 - p_3 e^{-j\omega})\ldots(1 - p_N e^{-j\omega})}$$

$$= \angle(e^{j\omega} - q_1) + \angle(e^{j\omega} - q_2) + \angle(e^{j\omega} - q_3)\ldots\angle(e^{j\omega} - q_M)$$

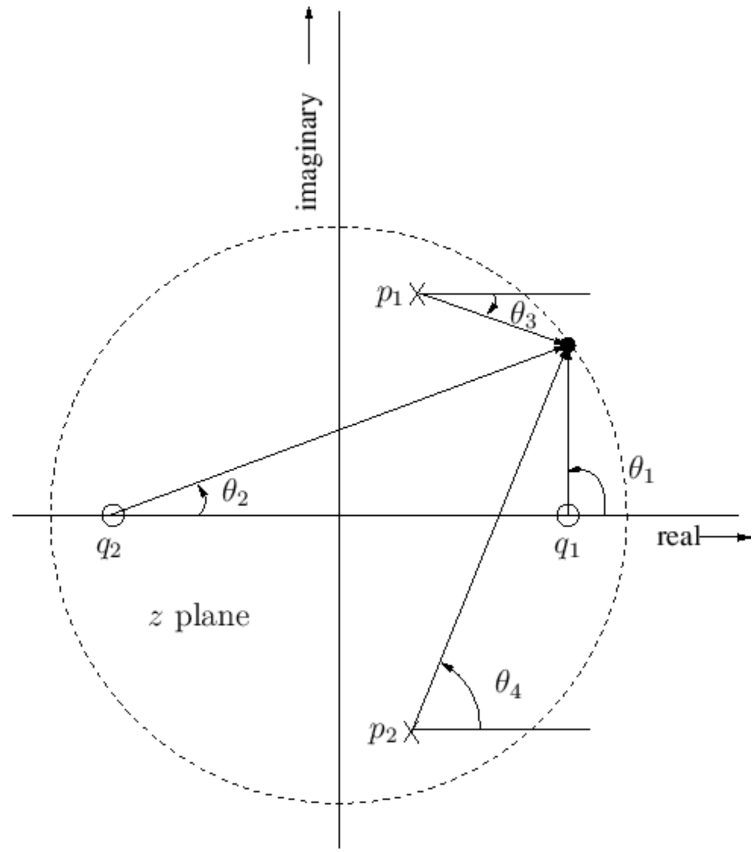
$$- \angle(e^{j\omega} - p_1) - \angle(e^{j\omega} - p_2) - \angle(e^{j\omega} - p_3)\ldots - \angle(e^{j\omega} - p_N)$$

Pole-Zero Analysis: Phase Response

- In the following examples, the phase response is given as

$$\Theta(\omega) = \theta_1 + \theta_2 - \theta_3 - \theta_4$$

  - Positive angles for zeros
  - Negative angle for poles



Formal Definition of Z-transform

- Z-transform

$$X(z) = \sum_{n=0}^{\infty} x(n) z^{-n}$$

- Properties
  - Shift theorem: $x(n-\Delta) \leftrightarrow z^{-\Delta} X(z)$
  - Convolution theorem: $x(n) * h(n) \leftrightarrow X(z) H(z)$
    - Therefore, the transfer function is represented as

$$y(n) = x(n) * h(n) \leftrightarrow H(z) = \frac{Y(z)}{X(z)}$$

- Decomposing z-transforms
  - Series combination: $H(z) = H_1(z) H_2(z) \leftrightarrow h(n) = h_1(n) * h_2(n)$
  - Parallel combination: $H(z) = H_1(z) + H_2(z) \leftrightarrow h(n) = h_1(n) + h_2(n)$

KAIST

Frequency Response by DTFT

- Discrete-Time Fourier Transform
  - Putting $z^{-1} = e^{-j\omega}$ back to the Z-transform

$$X(e^{-j\omega}) = X(\omega) = \sum_{n=0}^{\infty} x(n)e^{-jn\omega}$$

- The properties work for DTFT in the same manner
  - Shift theorem: $x(n-\Delta) \leftrightarrow e^{-j\Delta\omega}X(\omega)$
  - Convolution theorem: $x(n) * h(n) \leftrightarrow X(\omega)H(\omega)$

- The frequency response is represented as

$$y(n) = x(n) * h(n) \leftrightarrow H(\omega) = \frac{Y(\omega)}{H(\omega)}$$

Practical Filters

- One-pole one-zero filters
  - Moving average filters: low-pass filter
  - Leaky integrator: low-pass filter
  - DC-removal filters: high-pass filter
  - Bass / treble shelving filter

$$H(z) = \frac{b_0 + b_1 z^{-1}}{a_0 + a_1 z^{-1}}$$

- Bi-quad filters
  - Low-pass / high-pass filters
  - Band-pass / botch filters
  - EQ
  - The two-poles are real-numbers or complex conjugate

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

- Note that any high-order filter can be factored into one-pole one-zero filters and bi-quad filters!

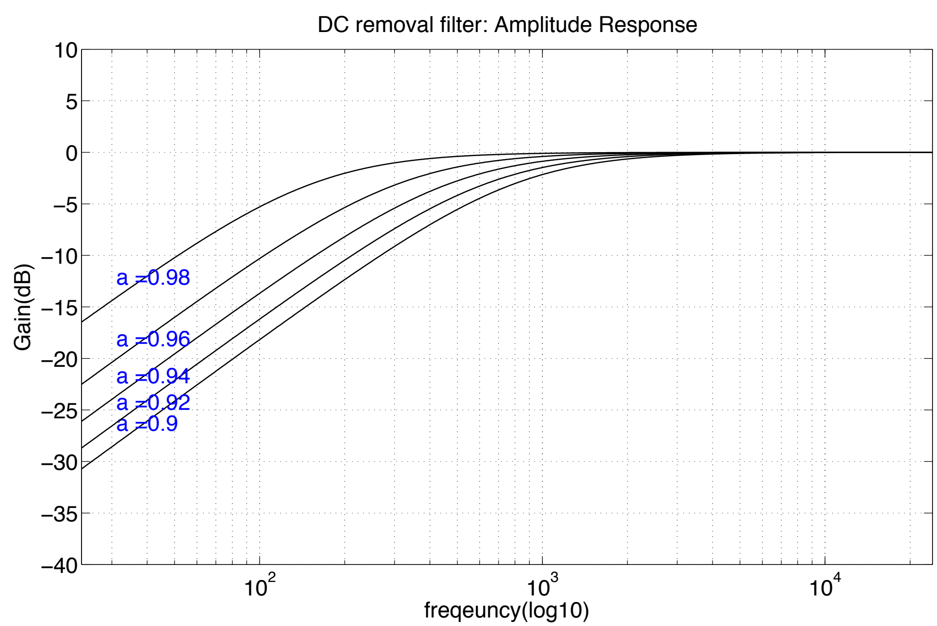
KAIST

CT

One-pole one-zero filters

- DC-removal filters: high-pass filter

$$H(z) = \frac{1 - z^{-1}}{1 - az^{-1}} \left(\frac{1 + a}{2}\right)$$



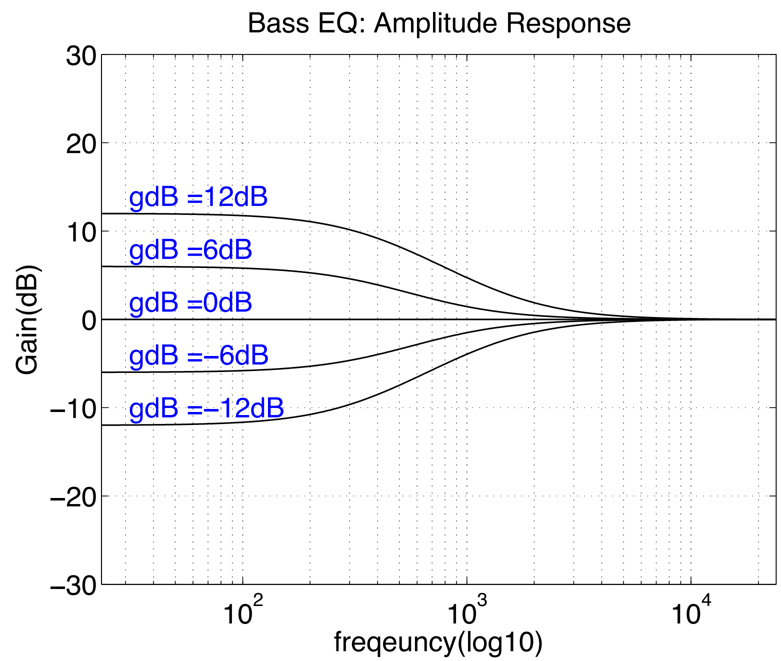
KAIST

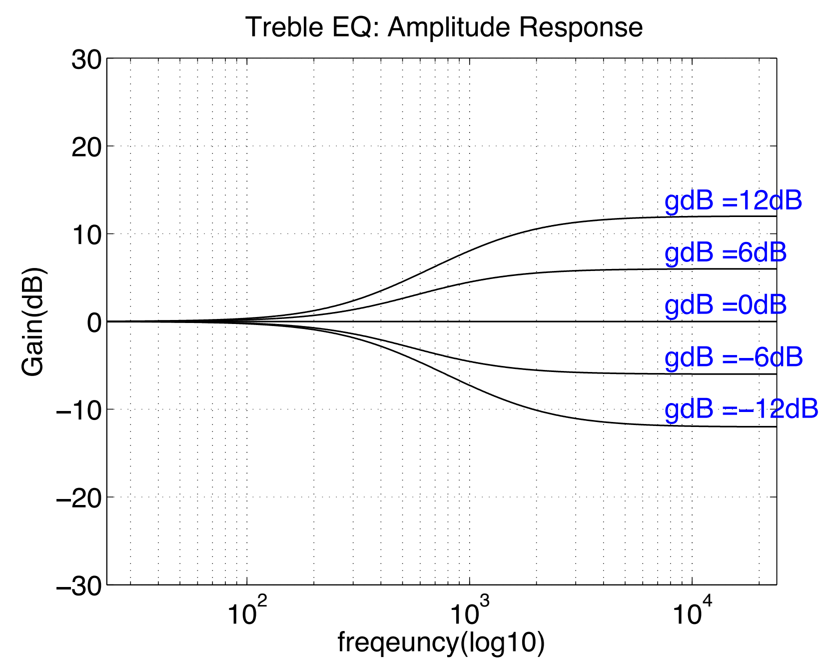
One-pole one-zero filters

- Bass / Treble shelving

$$H(z) = 1 + g\frac{1+z^{-1}}{1-az^{-1}}\frac{1-a}{2} \qquad H(z) = 1 + g\frac{1-z^{-1}}{1-az^{-1}}\frac{1+a}{2}$$

$$g = 10^{\frac{\mathrm{GdB}}{20}} - 1$$



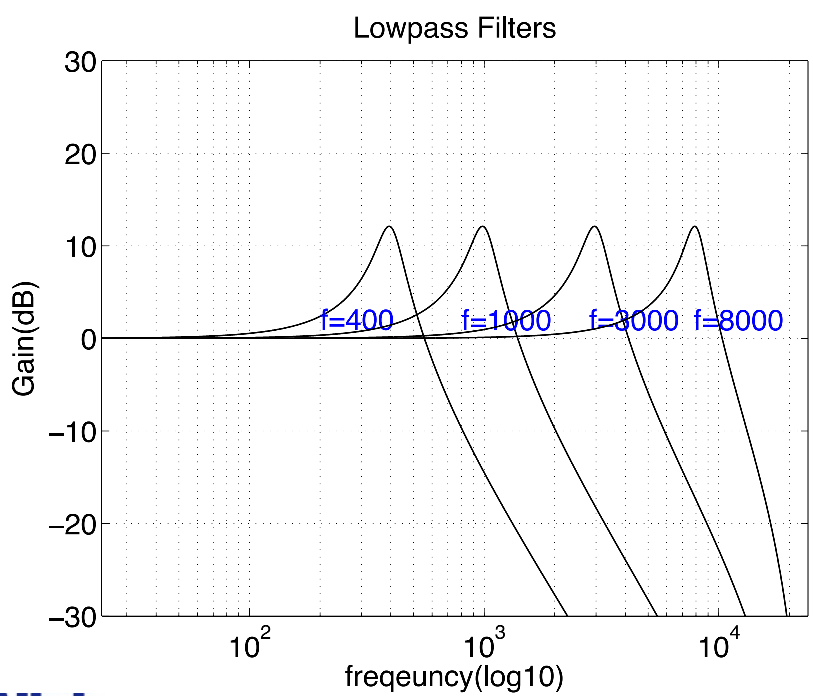


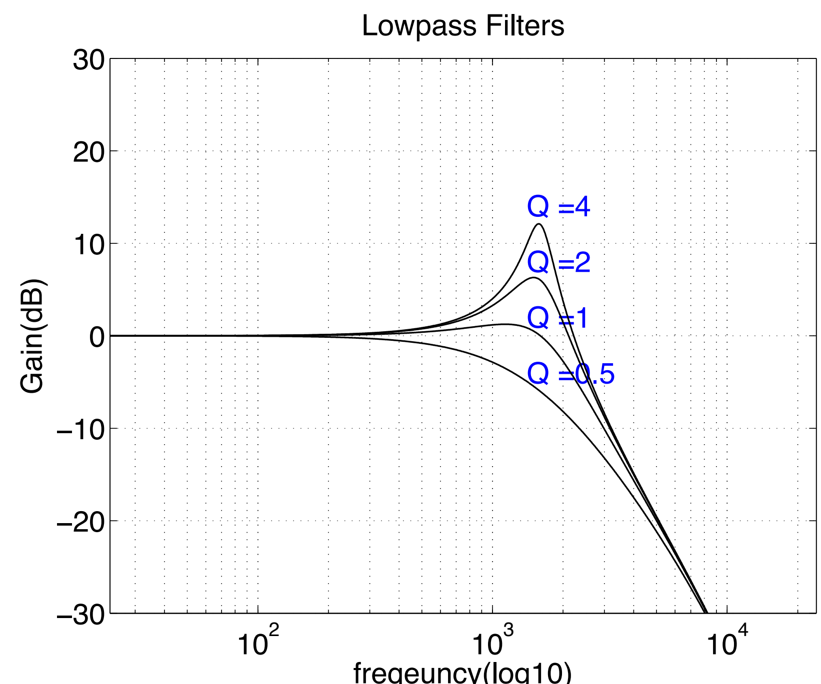
Bi-quad Filters

- Low-pass filter

$$H(z) = \left(\frac{1-\cos\Theta}{2}\right)\frac{1+2z^{-1}+1z^{-2}}{(1+\alpha)-2\cos\Theta z^{-1}+(1-\alpha)z^{-2}} \qquad \alpha = \frac{\sin\Theta}{2Q} \qquad \Theta = 2\pi f_c / f_s$$

  – fc : cut-off frequency, Q: resonance

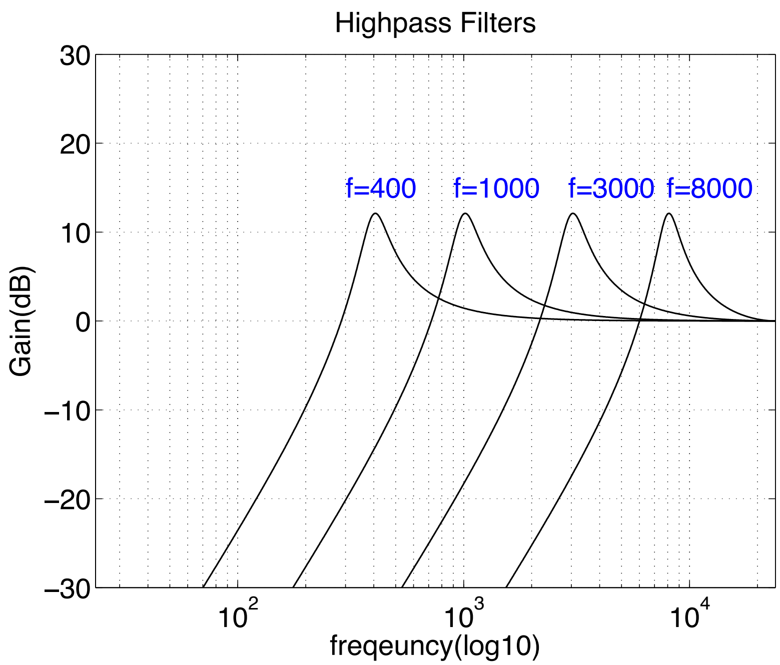


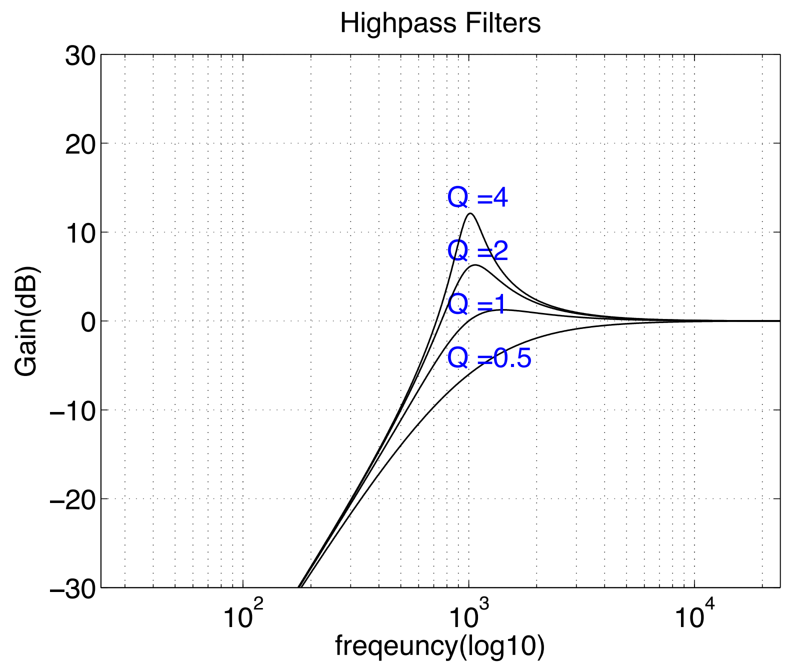


Bi-quad Filters

- High-pass filter

$$H(z) = \left(\frac{1+\cos\Theta}{2}\right)\frac{1-2z^{-1}+1z^{-2}}{(1+\alpha)-2\cos\Theta z^{-1}+(1-\alpha)z^{-2}} \qquad \alpha = \frac{\sin\Theta}{2Q} \qquad \Theta = 2\pi f_c / f_s$$

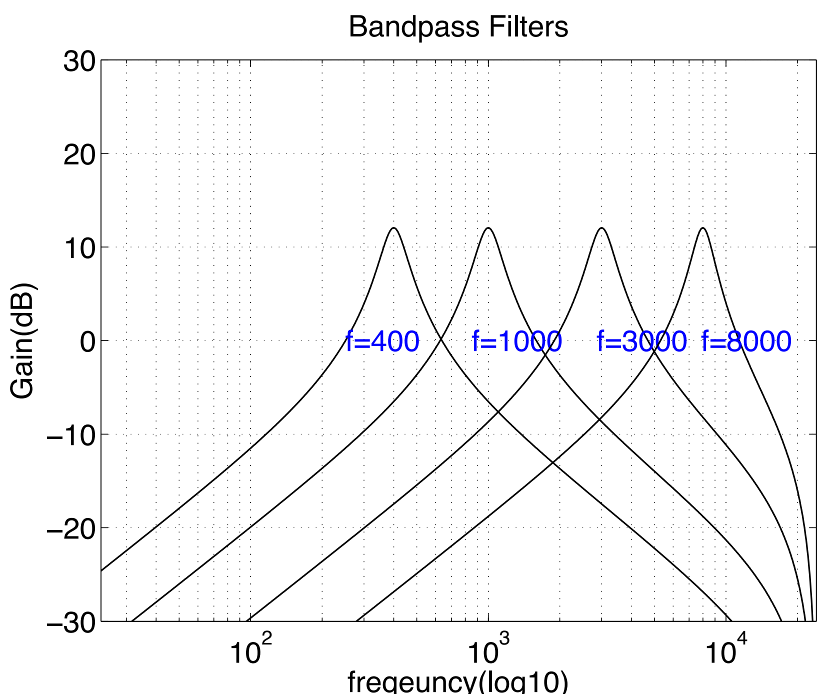


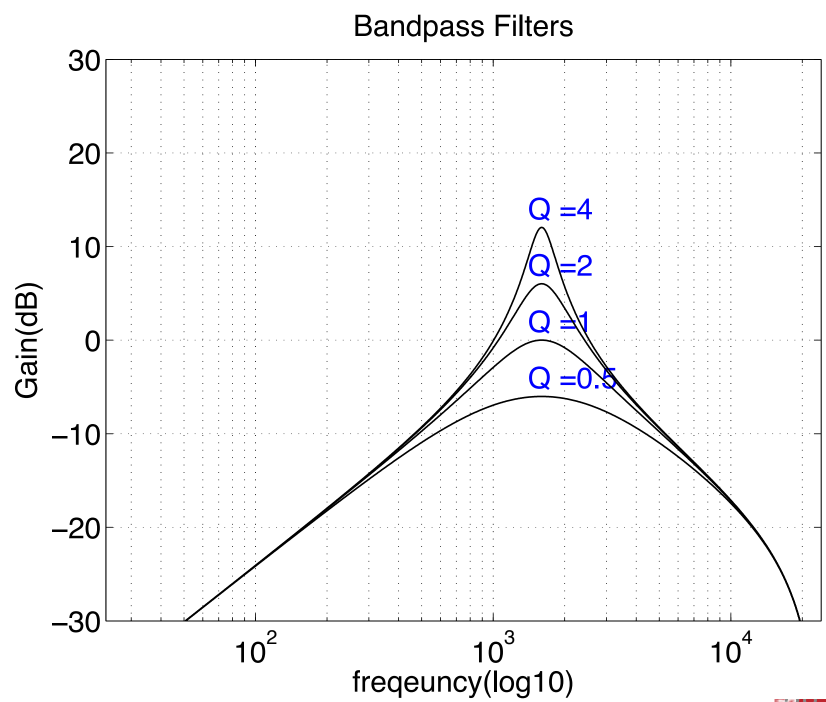


Bi-quad Filters

- Band-pass filter

$$H(z) = \left(\frac{\sin\Theta}{2}\right)\frac{1 - z^{-2}}{(1+\alpha) - 2\cos\Theta z^{-1} + (1-\alpha)z^{-2}} \qquad \alpha = \frac{\sin\Theta}{2Q} \qquad \Theta = 2\pi f_c / f_s$$

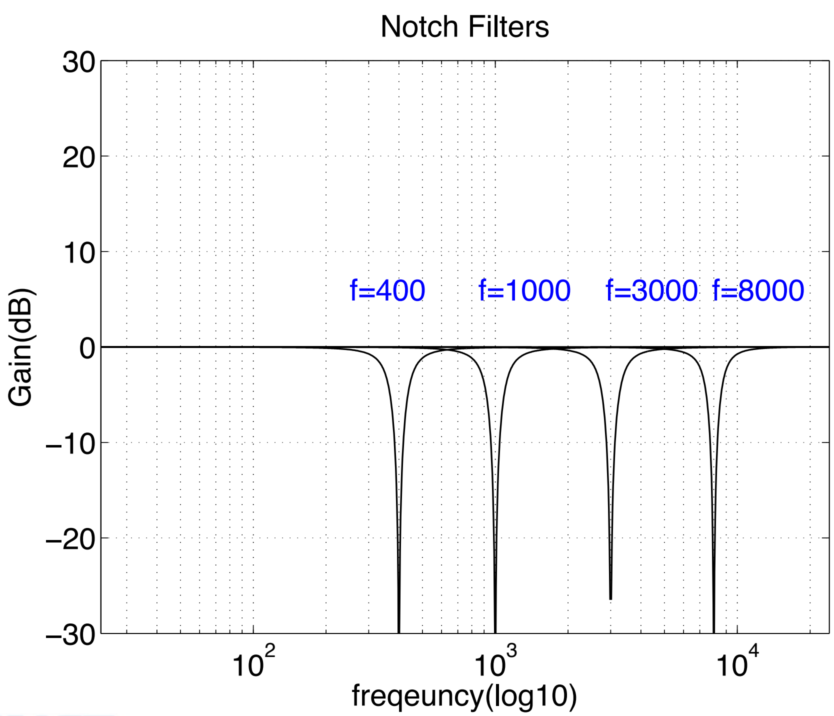


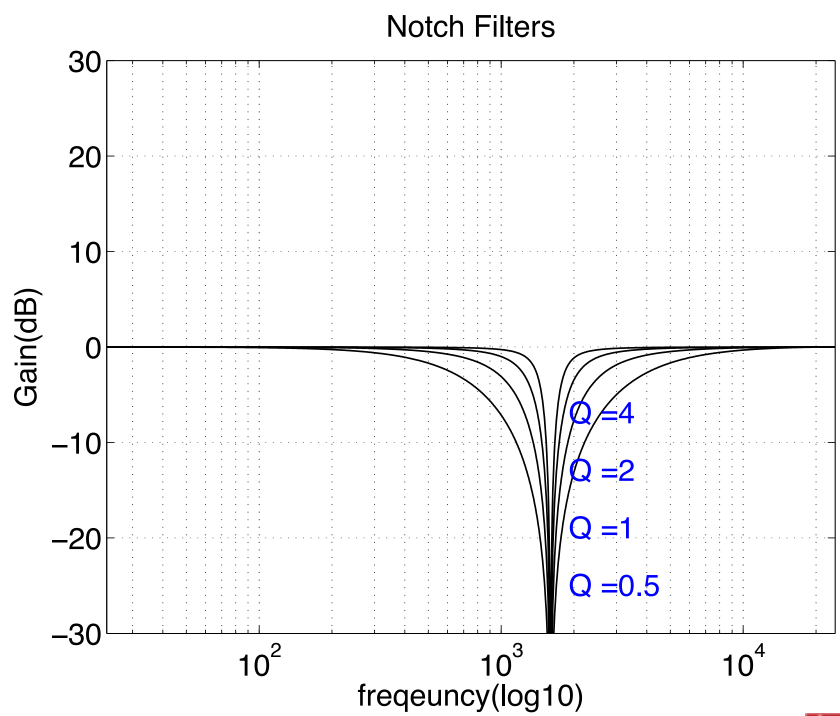


Bi-quad Filters

- Notch filter

$$H(z) = \frac{1 - 2\cos\Theta z^{-1} + z^{-2}}{(1+\alpha) - 2\cos\Theta z^{-1} + (1-\alpha) z^{-2}} \qquad \alpha = \frac{\sin\Theta}{2Q} \qquad \Theta = 2\pi f_c / f_s$$

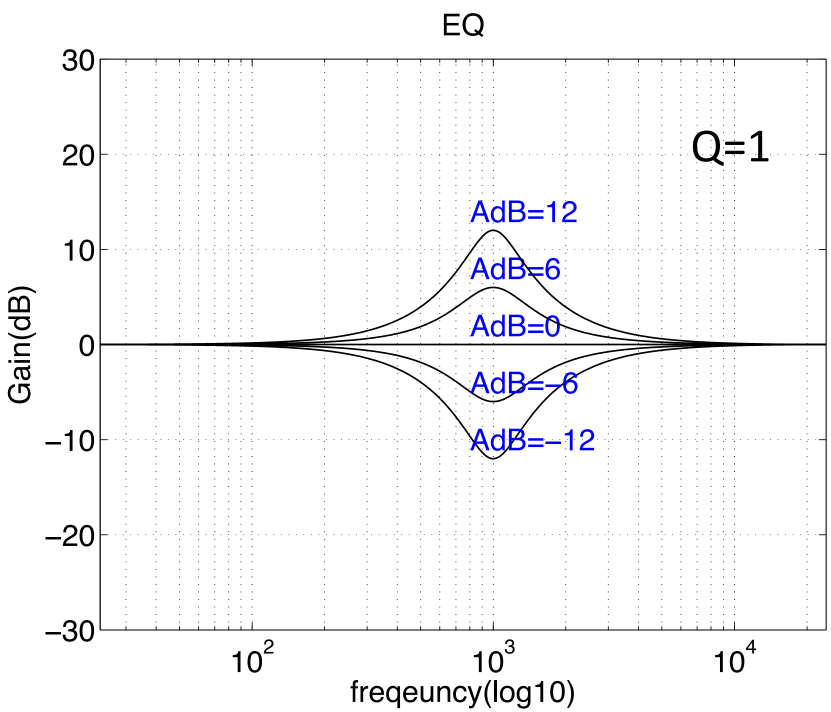


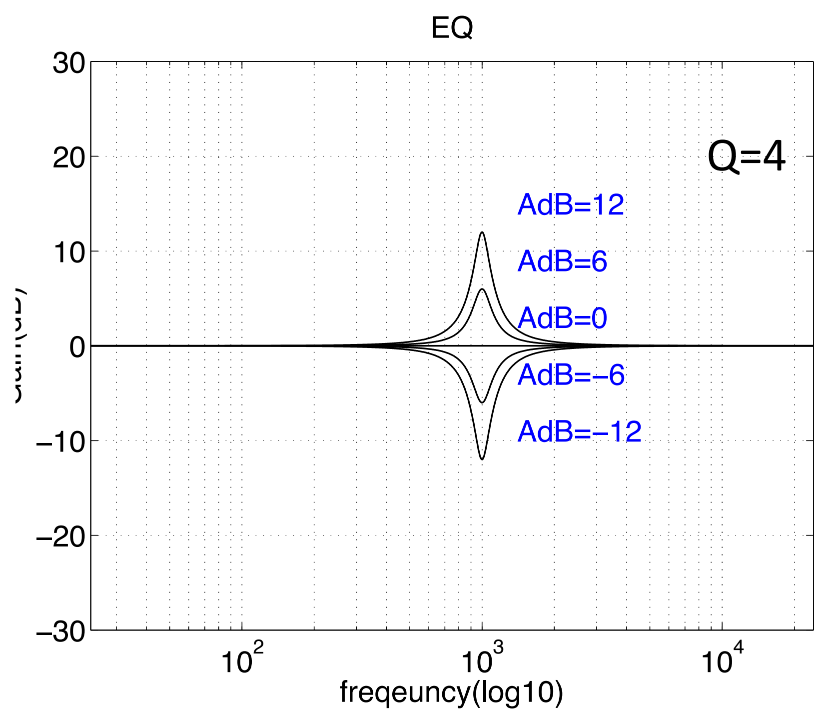


Bi-quad Filters

- Equalizer

$$H(z) = \frac{(1+\alpha\cdot A) - 2\cos\Theta z^{-1} + (1+\alpha\cdot A)z^{-2}}{(1+\alpha / A) - 2\cos\Theta z^{-1} + (1-\alpha / A)z^{-2}} \qquad \alpha = \frac{\sin\Theta}{2Q} \qquad \Theta = 2\pi f_c / f_s$$





References

- Cookbook formulae for audio EQ biquad filter coefficient, R. Bristow-Johnson
  - http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt