February 2, 2024

Audio Filter Design

ELEC-E5620 - Audio Signal Processing, Lecture #3

Vesa Välimäki

Acoustics Lab, Dept. Information and Communications Engineering

Aalto University
School of Electrical
Engineering

Sound check

1

Course Schedule for 2024 (Periods III-IV)

0.	General issues (Vesa, Gloria & Eloi)	12.1.2024
1.	History and future of audio DSP (Vesa)	19.1.2024
2.	Digital filters in audio (Vesa)	26.1.2024
3.	**Audio filter design (Vesa)**	**2.2.2024**
4.	Analysis of audio signals (Eloi)	9.2.2024
5.	Audio effects processing (Vesa)	16.2.2024
	** No lecture (Evaluation week for Period III)*	**23.2.2024*
6.	Sound synthesis (Dr. Fabian Esqueda, Korg Germany, Berlin)	1.3.2024
7.	Artificial reverberation (Dr. Karolina Prawda, Aalto Acoustics Lab)	8.3.2024
8.	Physics-based sound synthesis (Dr. Max Schäfer, Univ. Erlangen)	15.3.2024
9.	Sampling rate conversion (Vesa)	22.3.2024
	** No lecture (Spring Break/Easter)*	**29.3.2024*
10.	Audio coding (Vesa)	5.4.2024
11.	BONUS: Machine learning in audio (Eloi & Gloria)	12.4.2024

Aalto University
School of Electrical
Engineering

2

Demo: Saturating Resonant Filter

Ivan Benc

Xiaojie Pi

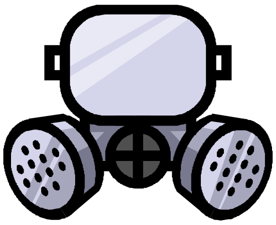
Ruijie Wang

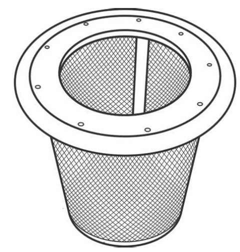
Aalto University
School of Electrical
Engineering

3

Outline

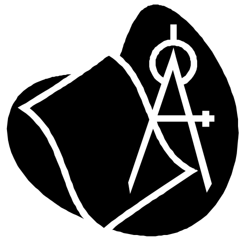
- Introduction and motivation
- Properties of digital filters
- FIR or IIR?
- Error norms for filter design
- Auditory resolution
- FIR filter design
- IIR filter design
- Graphic equalizers





Aalto University
School of Electrical
Engineering

Introduction & Motivation



- Basic audio filters
  - Usually, straightforward low-order design in closed form (see lecture #2 on Digital Filters)
- More advanced filters needed in many real-life applications
  - Equalization of loudspeakers/headphones
  - Room correction (loudspeaker-room EQ)
  - Graphic equalization (for music production, concerts etc.)
  - HRTF (head-related transfer function) filters for 3-D sound
  - Digital musical instruments, e.g., piano soundboard, guitar or violin body
  - Reverberation algorithms (to adjust the T60 curve)
  - Modeling of signal frames, e.g. in audio coding

Example, HRTF Modeling

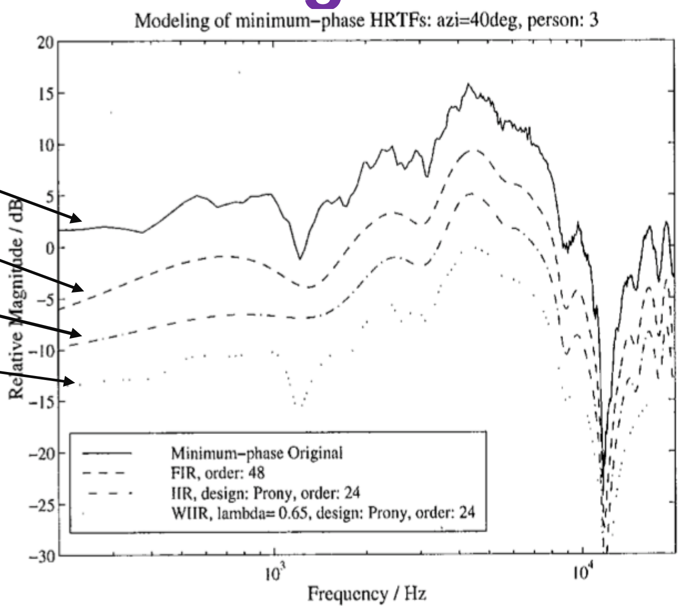
- A simple filter is insufficient

Target response

FIR, order 48

IIR, order 24

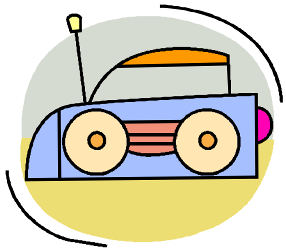
Warped IIR, order 24



• Source: Huopaniemi, J., Zacharov, N., and Karjalainen, M., Objective and Subjective Evaluation of Head-Related-Transfer Function Filter Design, 105th AES Convention

Introduction & Motivation 2

- Which application, what purpose?
  - Requirements help in choosing the design method
- General principle
  - **Quick** methods for real-time applications (aim at designing a low-order filter)
  - **Accurate** methods for off-line processing (high-order filters, when necessary)



- Things to remember in audio filter design
  - Frequency resolution of the hearing system is nonlinear (logarithmic): good at low, poor at high frequencies
  - Conventional ”textbook” filter design techniques (lowpass, highpass, bandpass…) are usually unsuitable

Properties of Filters

- Convolution for linear and time-invariant (LTI) systems:

$$y[n] = h[n] * x[n]$$

Impulse response of the filter

Input signal

- The same in Z-domain

$$Y(z) = H(z)X(z) \Leftrightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)}$$

FIR part

$$\sum_{k=0}^{M} b_k z^{-k}$$

All-pole part

$$\sum_{k=0}^{N} a_k z^{-k}$$

Properties of Filters: Magnitude

- Frequency response
  - Replace z with $e^{j\omega}$

$$H(e^{j\omega}) = \frac{B(e^{j\omega})}{A(e^{j\omega})}$$

- Magnitude response
  - Take the absolute value

$$|H(e^{j\omega})|$$

- In Matlab
  - `freqz.m`

Aalto University School of Electrical Engineering

Properties of Filters: Phase

- Phase response
  - Radian phase shift

$$\theta(\omega) = \arg\{H(e^{j\omega})\}$$

- Phase delay
  - Time delay of each sinusoidal component (in samples)

$$P(\omega) = -\frac{\theta(\omega)}{\omega}$$

- Group delay
  - Time delay (in samples) of the amplitude envelope of a sinusoid at frequency ω

$$D(\omega) = -\frac{d}{d\omega}\theta(\omega)$$

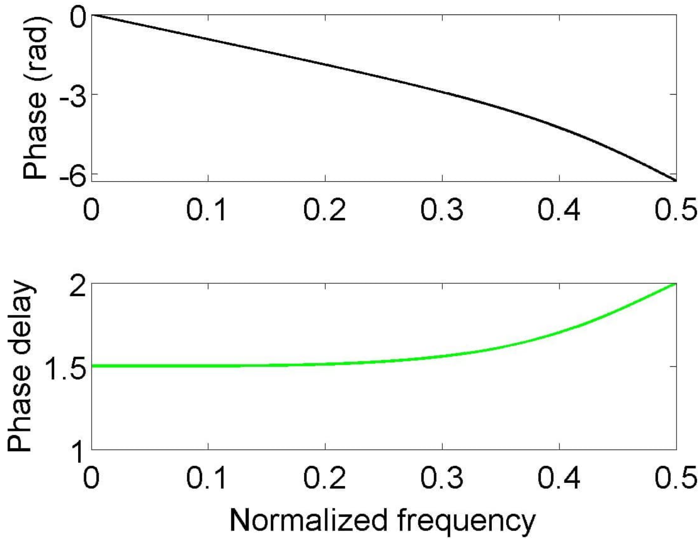
- In Matlab
  - `phasez.m`
  - `phasedelay.m`
  - `grpdelay.m`

Aalto University School of Electrical Engineering

Phase Delay of a Digital Filter

- Phase response and phase delay
  - o 2nd-order allpass filter

$$P(\omega) = -\frac{\theta(\omega)}{\omega}$$



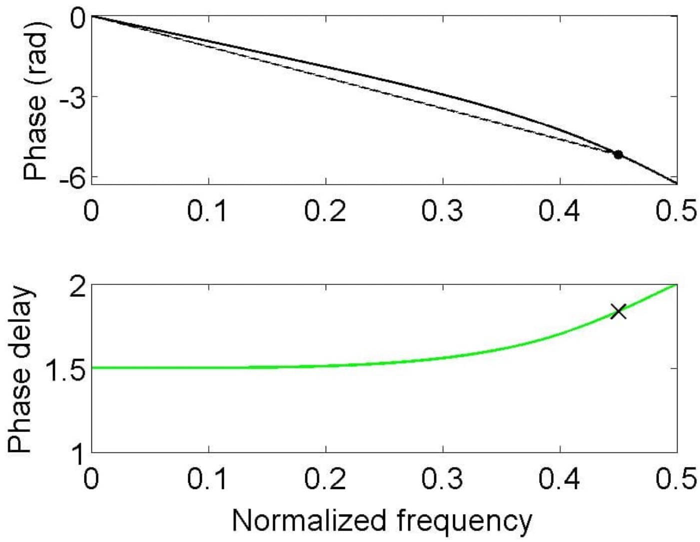
Aalto University School of Electrical Engineering

Phase Delay of a Digital Filter (2)

- Phase delay is negative of the slope of a straight line fitted through 0 and $\theta(\omega)$:

$$P(\omega) = -\frac{\theta(\omega)}{\omega}$$

$$= \frac{0-\theta(\omega)}{0-\omega}$$

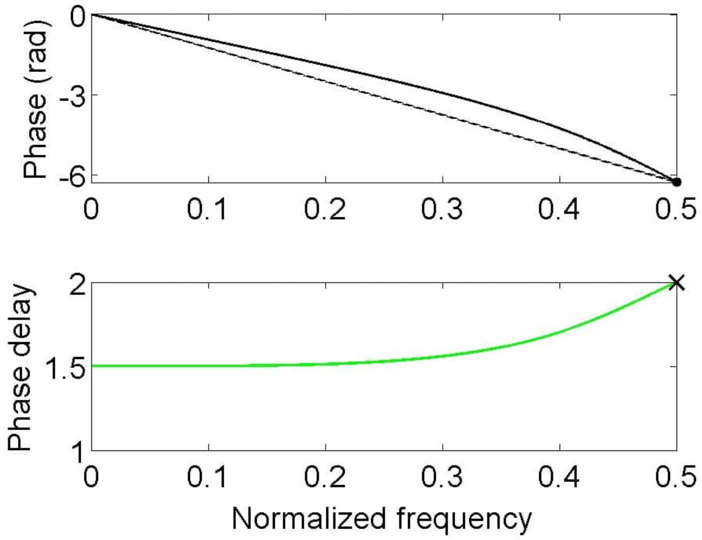


Aalto University School of Electrical Engineering

Phase Delay of a Digital Filter (3)

- Phase delay is negative of the slope of a straight line fitted through 0 and $\theta(\omega)$:

$$P(\omega) = -\frac{\theta(\omega)}{\omega}$$
$$= \frac{0 - \theta(\omega)}{0 - \omega}$$

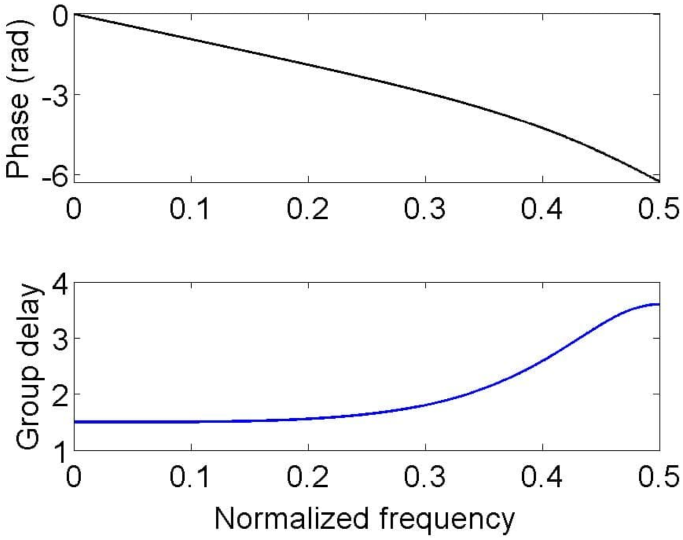


Aalto University School of Electrical Engineering

Group delay of a Digital Filter

- Phase response and group delay
  - o The same 2nd-order allpass filter as before

$$D(\omega) = -\frac{d}{d\omega}\theta(\omega)$$

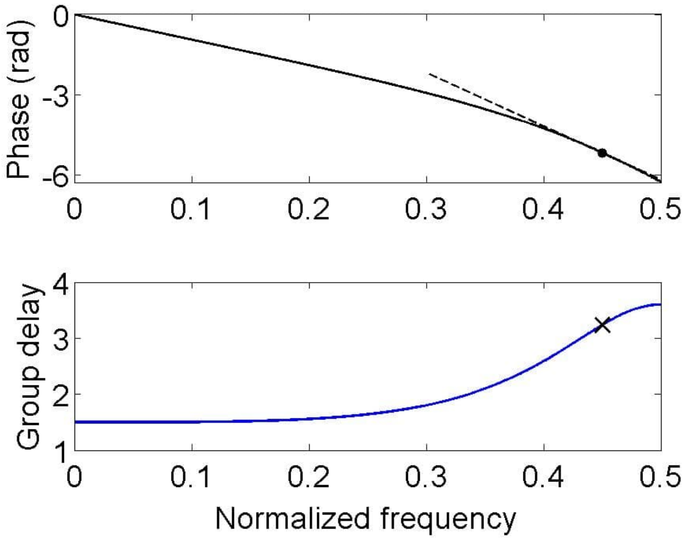


Aalto University School of Electrical Engineering

Group delay of a Digital Filter (2)

- Group delay is -1 times the local gradient of the phase function $\theta(\omega)$

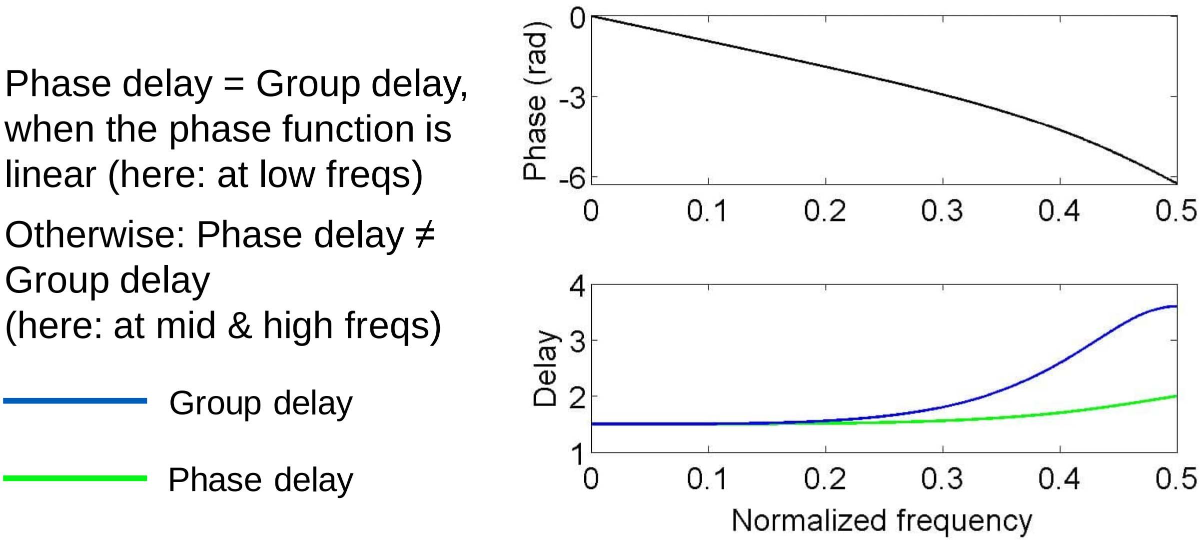
$$D(\omega) = -\frac{d}{d\omega}\theta(\omega)$$



15

Phase Delay vs. Group delay

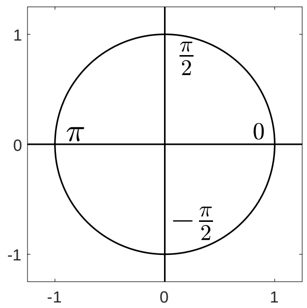
- Phase delay = Group delay, when the phase function is linear (here: at low freqs)
- Otherwise: Phase delay ≠ Group delay (here: at mid & high freqs)



16

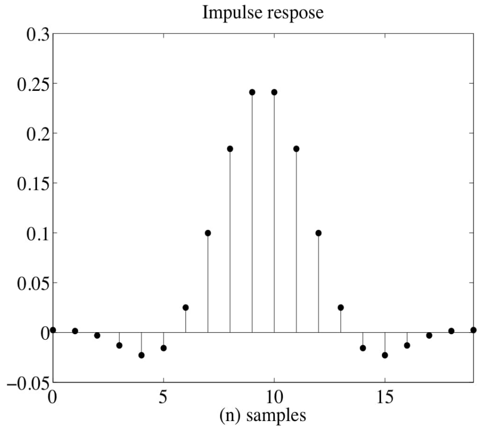
How to Present the Digital Frequency Axis?

- Radians
  - Sampling frequency = 2π → Axis $[0, \pi]$
- Normalized
  - Sampling frequency = 1 → Axis $[0, 0.5]$
  - Easy to convert to Hz by multiplying by the sampling frequency
- Matlab (default, but not a great choice)
  - Sampling frequency = 2 → Axis $[0, 1]$
- Hz
  - E.g., if the sampling frequency is 44.1 kHz → Axis $[0, 22.05]$ kHz



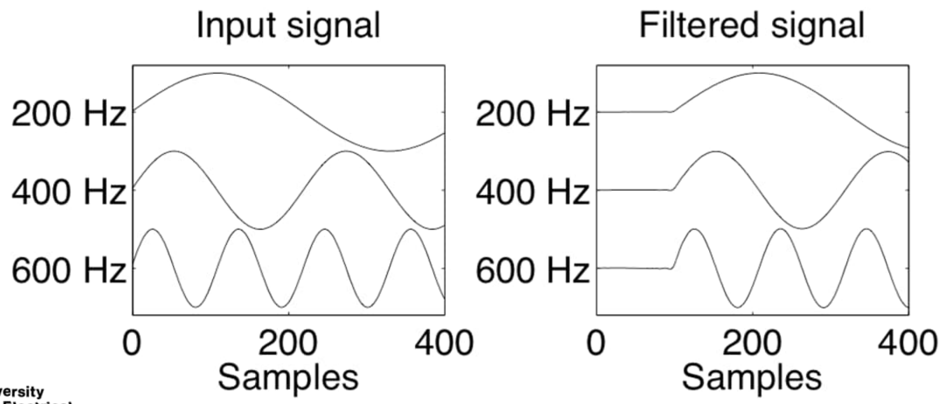
Phase Characteristics (1)

- Linear phase
  - Easy to obtain using FIR filters
  - Symmetric impulse response
  - Phase delay = group delay
  - Sometimes desirable, but why?
- Zero phase
  - Special case of a linear-phase filter
  - Impulse response symmetric w.r.t. 0 → noncausal!
  - Possible offline: `filtfilt.m`



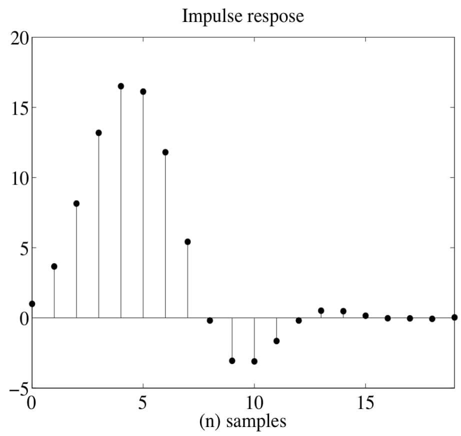
Linear Phase in Practice

- Linear phase → constant phase delay at all frequencies
  - In the example here, the filter causes 100 samples of delay



Aalto University School of Electrical Engineering

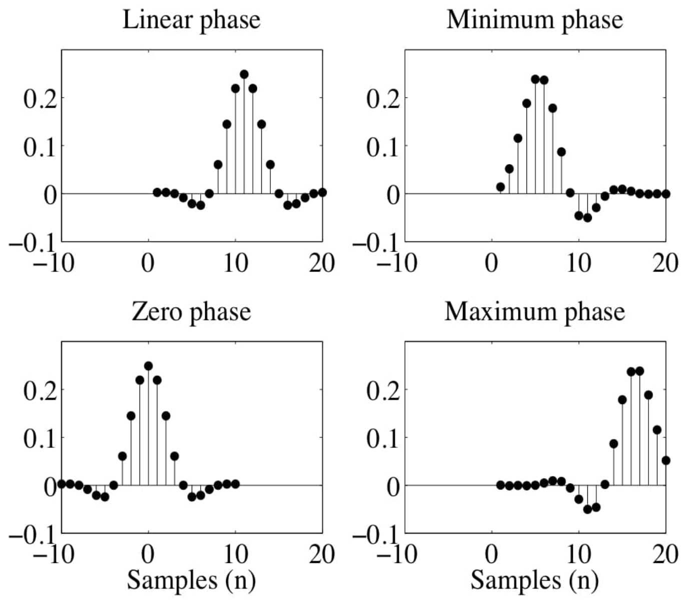
Phase Characteristics (2)



- Minimum phase
  - No zeros outside the unit circle
  - Often desirable, why?
  - Releases energy as fast as possible for the given magnitude response
  - Always stable and causal
  - Also its inverse filter is stable
- Maximum phase
  - All zeros outside the unit circle (or on the unit circle)
  - Time-reversed version of the minimum-phase system

Aalto University School of Electrical Engineering

Impulse Response with Different Phase

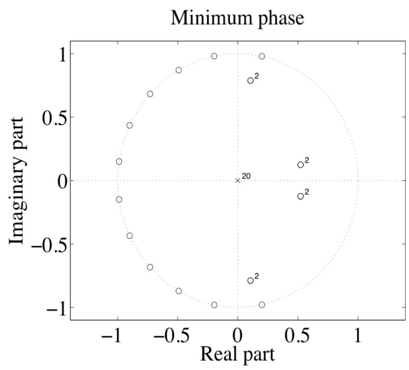


How to Construct a Minimum Phase?

- Option #1: Mirroring the Zeros
  - Zeros outside the unit circle are relocated to inside the unit circle

$$z = re^{j\theta}, r > 1 \ \rightarrow z = \frac{e^{j\theta}}{r}$$

  - Not feasible when the filter order is large, because then it's difficult or impossible to find the zeros (*i.e.*, roots of a polynomial)



How to Construct a Minimum Phase?

- Option #2: The Hilbert Transform Method
  - Logarithm of the magnitude response and the phase function of a minimum-phase system are Hilbert transform pairs
  - To convert a filter to minimum-phase: Replace the filter's phase response with the Hilbert transform of its log magnitude response!
  - The Hilbert transform shifts the phase of the input signal by 90 degrees (π radians)
  - See: https://ccrma.stanford.edu/~jos/sasp/Minimum_Phase_Filter_Design.html

How to Construct a Minimum Phase?

- Option #3: Cepstral Windowing
  - Cepstrum is the inverse Fourier transform of the log of the spectrum
    - F^{-1} is the inverse Discrete Fourier Transform (DFT)
  - Deleting the left side of the cepstrum modifies the phase so that the signal becomes minimum-phase
    - The window $w(n)$ is a unit-step function (selecting the right half)
  - The minimum-phase impulse response is obtained with DFT.
  - Download the codes (`minphase_rceps.m`, `minphase_hilbert.m`) from MyCources and try!
  - See also: https://ccrma.stanford.edu/~jos/fp/Creating_Minimum_Phase_Filters.html

$$c(n) = F^{-1}\{\log|H(z)|\}$$

$$\hat{c}(n) = c(n)w(n)$$

$$h_{\mathrm{mp}}(n) = F^{-1}\{e^{F\{\hat{c}(n)\}}\}$$

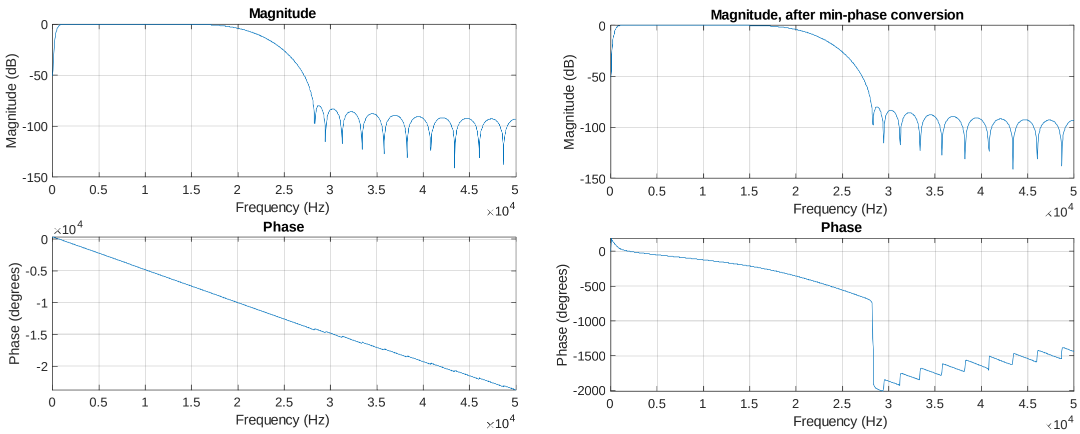
Linear-Phase/Minimum-Phase FIR Filter

- Loudspeaker model
  - o Low corner frequency (highpass): ≈500 Hz order 256
  - o High corner frequency (lowpass): ≈20 kHz order 32
- For plotting: `freqz.m, impz.m, zplane.m`
  - o Pay attention to the frequency normalization!
- Download the code (`LS_FIR_demo.m`) from MyCources!

Aalto University School of Electrical Engineering

Linear-Phase/Minimum-Phase FIR Filter

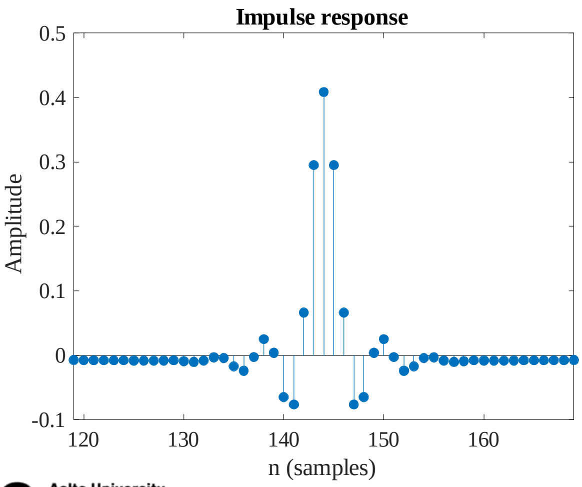
Linear phase Minimum phase



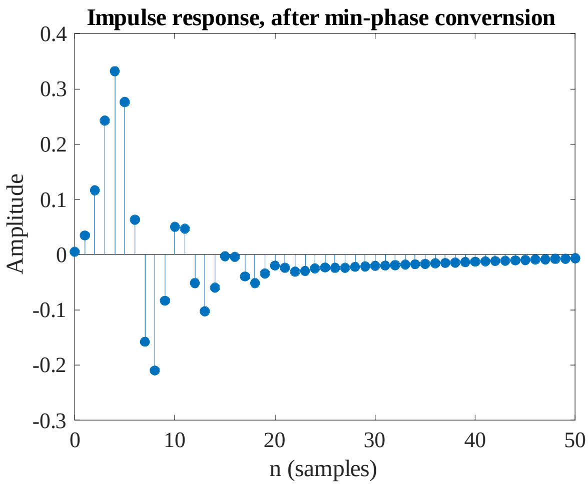
Aalto University School of Electrical Engineering

Linear-Phase/Minimum-Phase FIR Filter

Linear phase



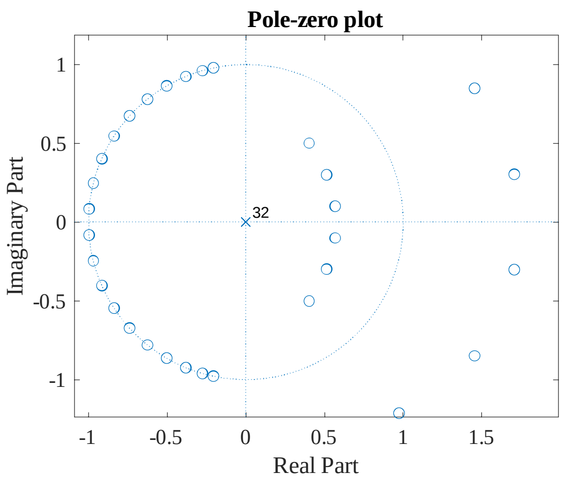
Minimum phase



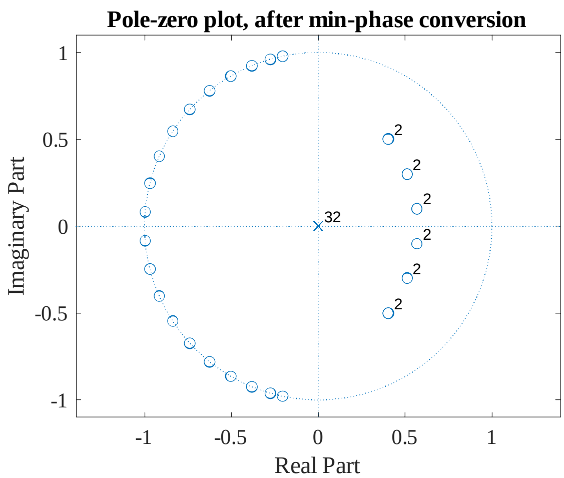
Aalto University
School of Electrical
Engineering

Linear-Phase/Minimum-Phase FIR Filter

Linear phase



Minimum phase



Aalto University
School of Electrical
Engineering

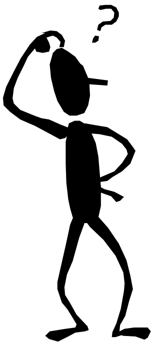
Filter Design Problem

- How to approximate the target response with a digital filter
- Minimize the frequency-response error:

$$E(e^{j\theta}) = H(e^{j\theta}) - \hat{H}(e^{j\theta})$$

Error = target – approximation

- In digital filter design, L_p norms are often used
  - o L_2 leads to the least-squares design
  - o L_∞ leads to the Chebyshev (minimax) design



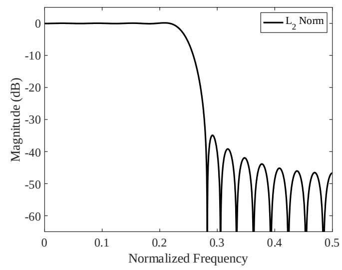
Least-squares (L_2) Norm

$$E_2(e^{j\omega}) = \left\| H(e^{j\omega}) - \hat{H}(e^{j\omega}) \right\|_2^2 \mathrel{\widehat{=}} \sum_{n=0}^{\infty} \left| h(n) - \hat{h}(n) \right|^2$$

- Many popular filter design techniques, such as Prony's method, use this method (`prony.m` in Matlab)
- An easy LS FIR filter design method is to truncate the impulse response of the ideal filter
  - o By including the largest samples, you minimize the squared error!

Solving FIR Filter Coefficients in LS Design

- If $\boldsymbol{h}$ is the desired response vector, $\boldsymbol{\varepsilon}$ is the frequency response error vector, and $\mathbf{F}$ is the discrete Fourier transform (DFT) matrix
- Minimize the squared error to design coefficients $\hat{h}$:



$$\mathbf{F}\hat{\boldsymbol{h}} = \boldsymbol{h} + \boldsymbol{\varepsilon}$$

$$\min\|\boldsymbol{\varepsilon}\|^2 = \min\left\|\mathbf{F}\hat{\boldsymbol{h}} - \boldsymbol{h}\right\|^2$$

$$\frac{\partial}{\partial h}\left(\mathbf{F}\hat{\boldsymbol{h}} - \boldsymbol{h}\right)\left(\mathbf{F}\hat{\boldsymbol{h}} - \boldsymbol{h}\right)^T = 0$$

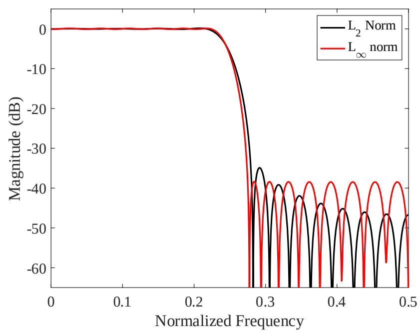
$$\frac{\partial}{\partial h}\left(\hat{\boldsymbol{h}}^T\mathbf{F}^T\mathbf{F}\hat{\boldsymbol{h}} - 2\boldsymbol{h}^T\mathbf{F}\hat{\boldsymbol{h}} + \boldsymbol{h}^T\boldsymbol{h}\right) = \mathbf{F}^T\mathbf{F}\hat{\boldsymbol{h}} - \mathbf{F}^T\boldsymbol{h} = 0$$

$$\Leftrightarrow \hat{\boldsymbol{h}} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T\boldsymbol{h}^T$$ “pseudoinverse” Matlab: hhat = F\h

Aalto University School of Electrical Engineering

Chebyshev (L_∞) Norm

- Chebyshev (minimax) norm:
  - o Minimize the maximum component of the error norm
  - o Requires an iteration loop and an end rule (for selected accuracy)
  - o The error function will be “equiripple”

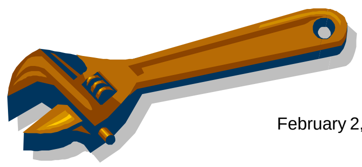


$$E_\infty(e^{j\omega}) = \left\|H(e^{j\omega}) - \hat{H}(e^{j\omega})\right\|_\infty \triangleq \max_{-\pi<\omega<\pi}\left|H(e^{j\omega}) - \hat{H}(e^{j\omega})\right|$$

Aalto University School of Electrical Engineering

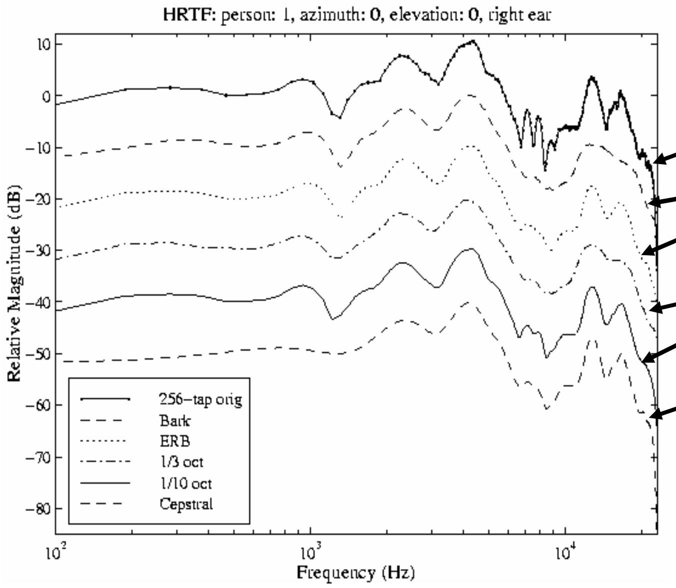
Auditory Resolution

- Account for the properties of the human hearing
  - Linear or non-uniform frequency scale, *e.g.*, critical bands (in Bark)
  - The domain in which the error is matched (*e.g.*, loudness or T60 domain)
- Tools
  - Auditory smoothing: Average of the spectrum with a sliding window (the window size should increase with frequency, as in octaves or Barks
  - Weighting functions: Suppress less important frequencies in the error (e.g., weighting proportional to Bark bandwidths)
  - Frequency warping: Expand the important frequency range; shrink the less important range



Aalto University
School of Electrical
Engineering

Auditory Smoothing



Example target response

- ERB & Bark smoothing
  (Equivalent Rectangular Bandwidth)
- 1/3 and 1/10-octave smoothing
- Cepstral smoothing

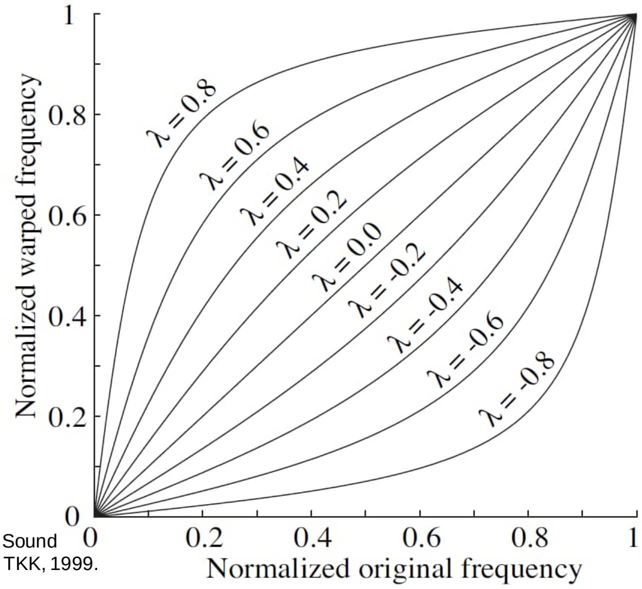
Source: Huopaniemi, J.
Virtual Acoustics and 3-D Sound
in Multimedia Signal Processing.
Doctoral Thesis, TKK, 1999.

Aalto University
School of Electrical
Engineering

Frequency Warping

- Frequency scale warping before designing the filter
  - o Replace the unit delay with a first-order allpass filter
  - o λ is the warping parameter

$$z^{-1} \rightarrow \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}}$$

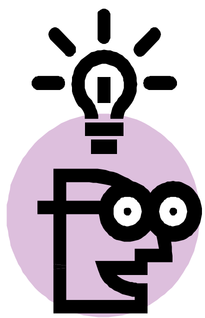


Source: Huopaniemi, J. Virtual Acoustics and 3-D Sound in Multimedia Signal Processing. Doctoral Thesis, TKK, 1999.

Aalto University
School of Electrical
Engineering

Properties of Warped Filters

- Allpass filter → flat magnitude response with frequency-dependent group delay
- If $\lambda = 0.7661$ (f_s = 48 kHz), the warping closely matches the Bark scale (Smith & Abel 1999)
- Design the filter in warped frequency domain → stretch the important frequencies



Aalto University
School of Electrical
Engineering

How to Warp?

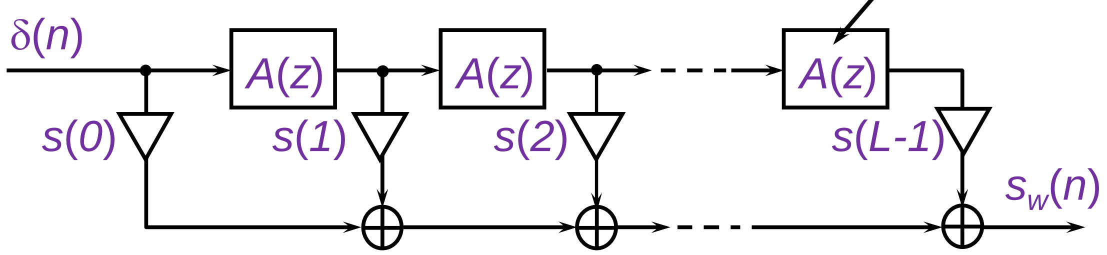
- Implement the filter in the warped frequency domain
  - WFIR or WIIR structures
  - More accuracy, but more computation as well
  - In WIIR structure the delay-free loops must be avoided → remap coefficients
- Unwarp the warped transfer function to a standard filter

Original $\lambda = 0.4$ $\lambda = 0.8$

Warped FIR Filter

- Chain of first-order allpass filters in an FIR structure

$$A(z) = \frac{z^{-1} + \lambda}{1 + \lambda z^{-1}}$$



- $s(n)$ is the signal to be warped, $\delta(n)$ is an impulse, $s_w(n)$ is the warped signal
- The extent of warping is determined by λ
  - With $\lambda = 0$, the warped FIR filter becomes a standard FIR filter

Windowing Methods for FIR Filters

- Truncate the impulse response using a window function
- Many choices for windowing functions!
- Suitable for modeling measured/estimated impulse responses in audio
- In Matlab: fir1.m

Frequency Sampling Method

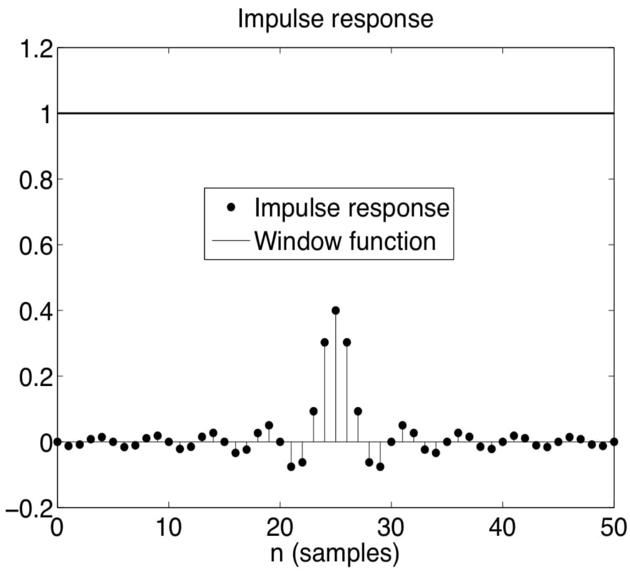
- Sample the target frequency response uniformly (magnitude & phase)
- Reflect it to the negative frequencies: same magnitude, opposite phase
- Use Inverse DFT (IDFT) to obtain the filter coefficients
- The length of the IDFT specifies the filter length
- Check the frequency response with zero-padded DFT
- Note: there will be “don’t care” regions between the sample points
- In Matlab: fir2.m

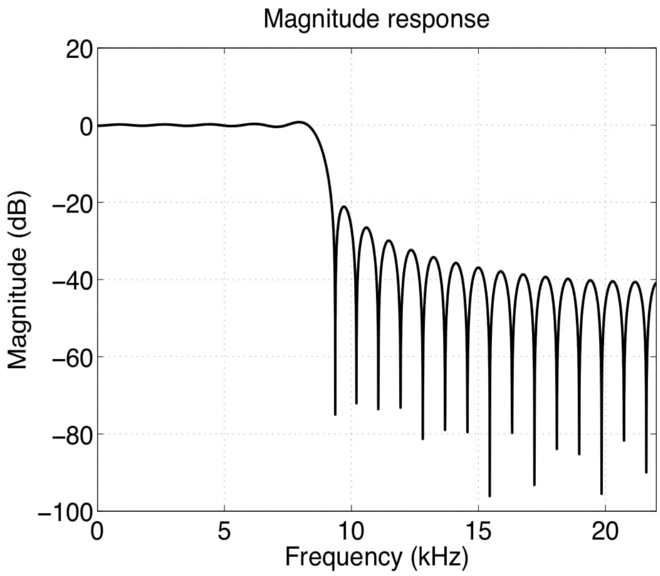
Properties of Window Functions

- Rectangular window
  - o Minimizes the truncated time-domain L_2 norm
  - o Gibb's phenomenon (passband ripple)
  - o Low sidelobe rejection
  - o Steep transition band
- Other window functions
  - o Hann
  - o Hamming
  - o Blackman
  - o …

Aalto University
School of Electrical
Engineering

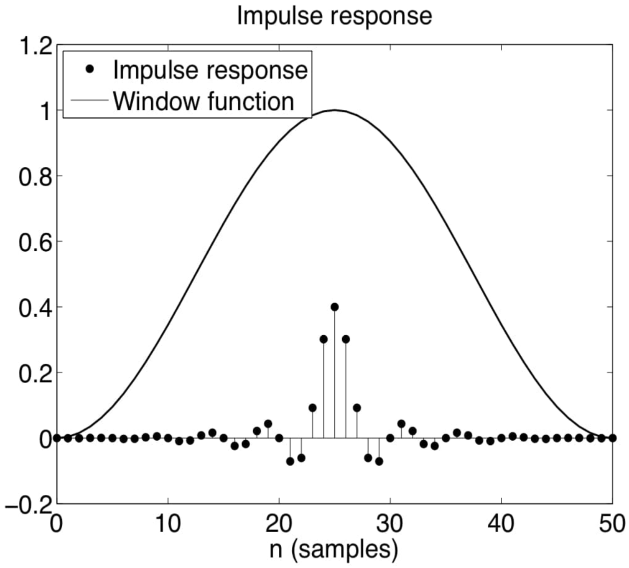
Rectangular window

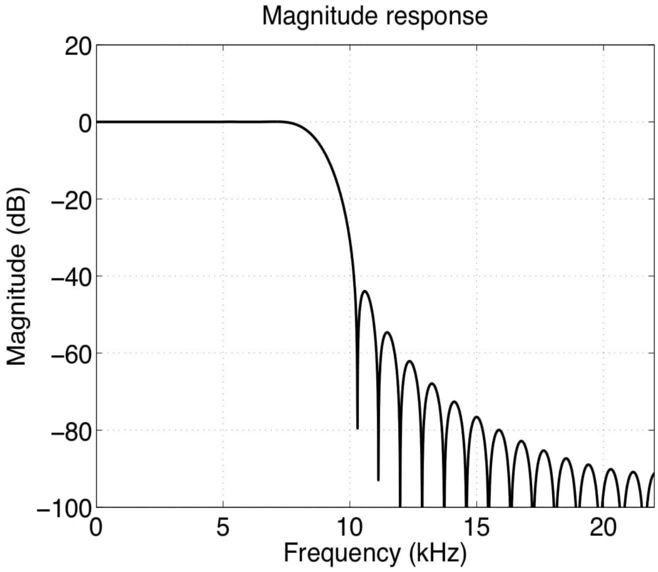




Aalto University
School of Electrical
Engineering

Hann window



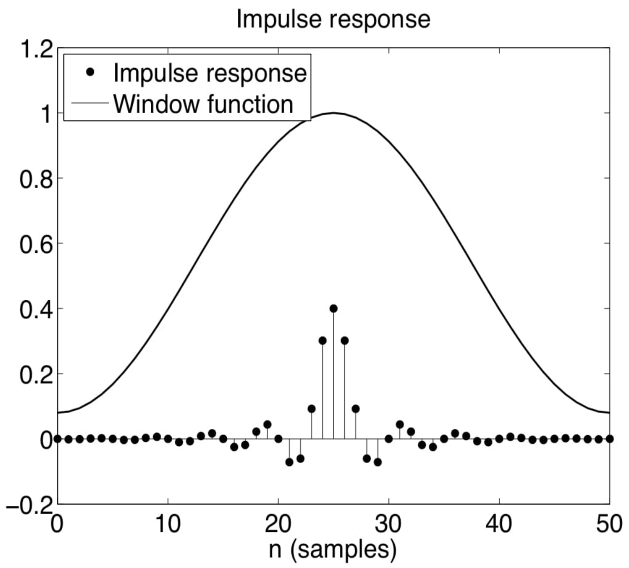


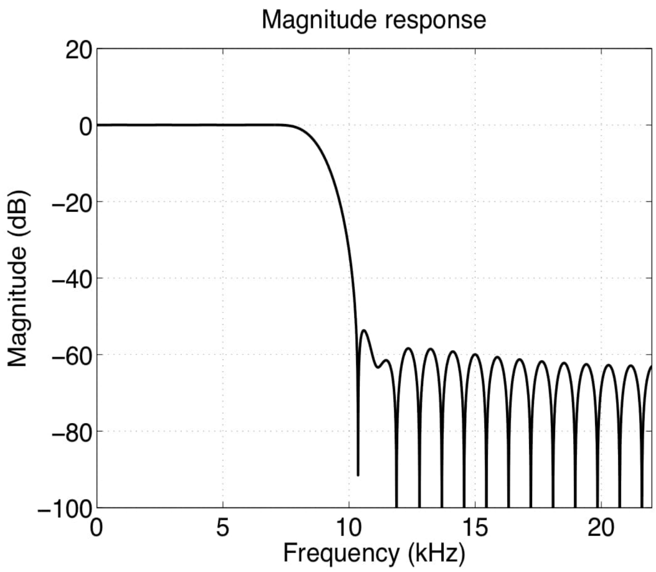
Aalto University
School of Electrical
Engineering

February 2, 2024
43

43

Hamming window



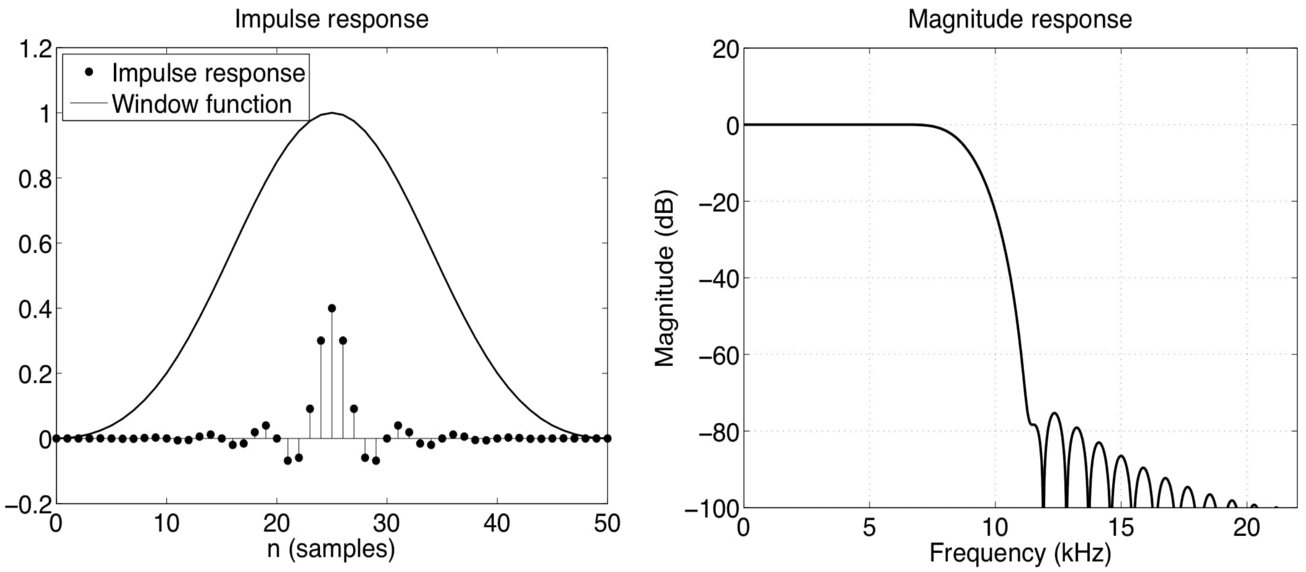


Aalto University
School of Electrical
Engineering

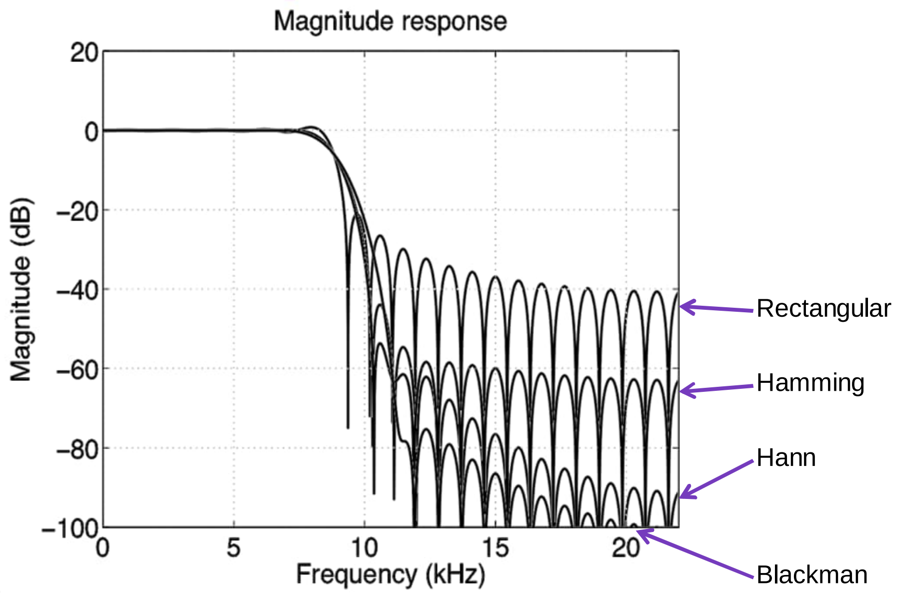
February 2, 2024
44

44

Blackman window



Window Comparison



Weighted LS FIR Filter Design

- Using a non-negative weight function W, minimize

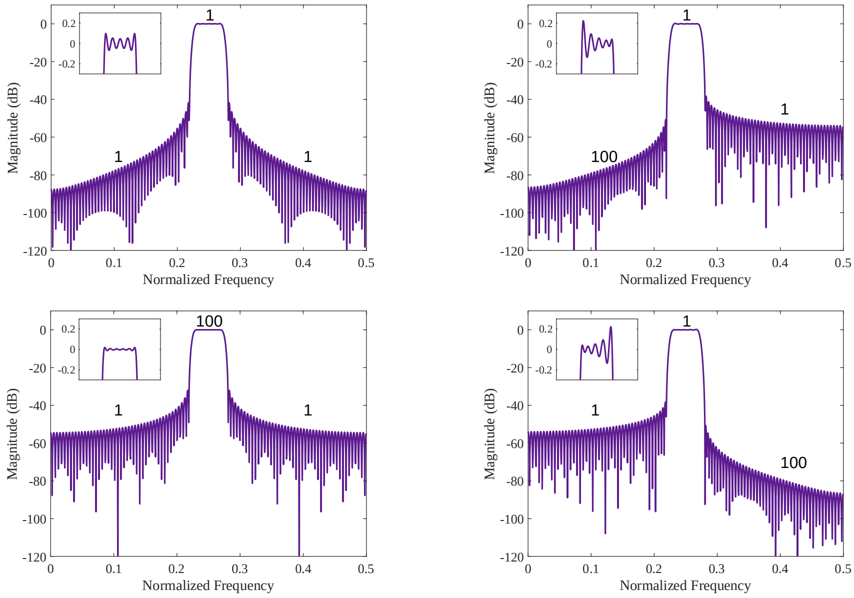
$$E = \sum_{k=0}^{L-1} W(\omega_k)\ \left|H(\omega_k) - \widehat{H}(\omega_k)\right|^2$$

- Again, minimize the squared error:

$$\widehat{\boldsymbol{h}} = (\mathbf{F}^T\mathbf{W}\mathbf{F})^{-1}\mathbf{F}^T\mathbf{W}\boldsymbol{h}^T$$

- Give more weight to important points
  - o Frequency ranges of interest

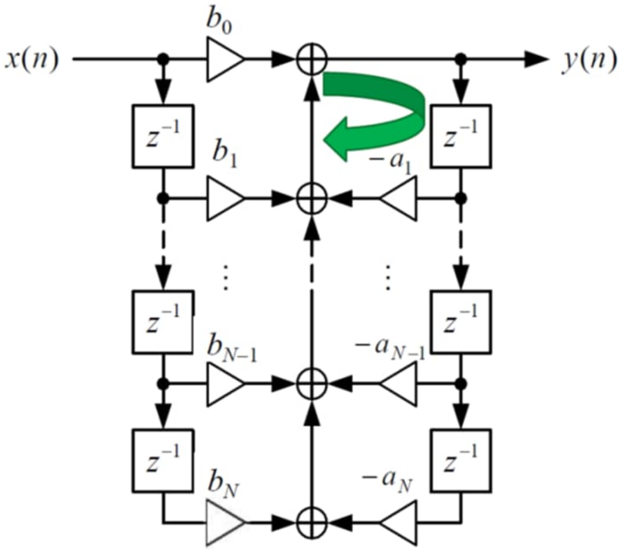
Weighted LS FIR Filter Design



`firls.m`
Least-squares linear-phase FIR filter design

IIR Filter Design Methods

- Linear prediction
- Prony's method
- Fixed-pole IIR filters
  - o Balazs Bank 2007, 2008
- Graphic equalization



Linear Prediction

- Also used in Linear Predictive Coding (LPC)
- Powerful analysis technique, especially in speech processing
- Predict the next sample as a linear combination of previous samples
- Many applications
  - o Spectral envelope estimation
  - o Pitch prediction
  - o Signal modeling for sound synthesis
  - o Detection of clicks in audio restoration

LPC Coefficients

- Compute the next sample as a linear combination of the previous samples

$$\hat{x}(n) = \sum_{k=1}^{P} a_k x(n-k)$$

- Minimize the expected value of the mean-squared prediction error

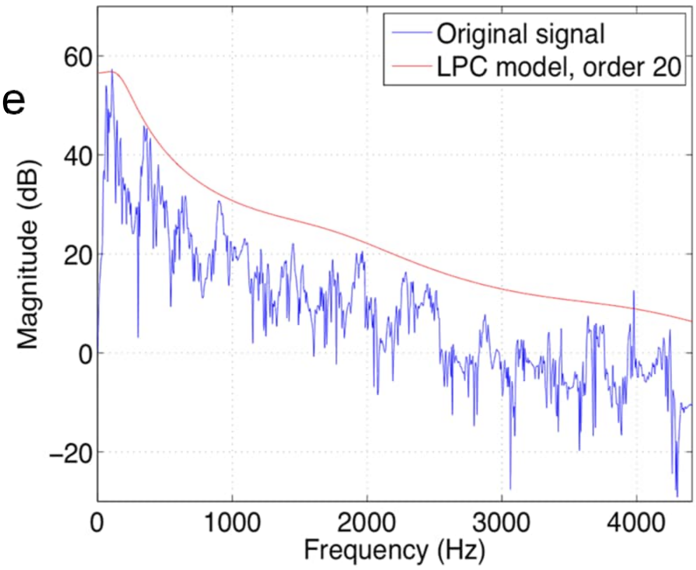
$$E\{e_n^2\} = E\{(x(n) - \hat{x}(n))^2\}$$

LPC Model

- Analysis
  - Use the autocorrelation or covariance method to obtain prediction coefficients
  - FIR filter with prediction coefficients is an inverse filter, which whitens the input signal (decorrelation)
  - Autocorrelation method → minimum phase
- Synthesis
  - Prediction coefficients used in an all-pole IIR filter yield a filter with the spectral characteristics of the input signal
- In Matlab: `lpc.m`

Piano Soundboard Modeling with LPC, $N = 20$

- (Recorded) target response

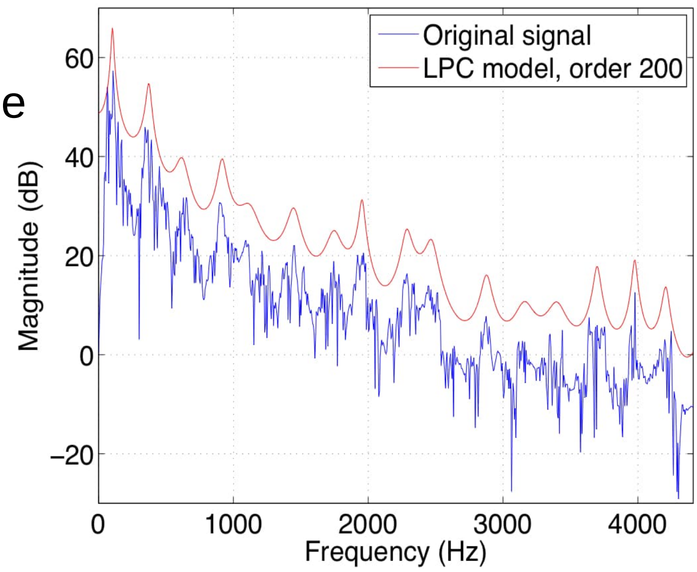


Aalto University School of Electrical Engineering

53

Piano Soundboard Modeling with LPC, $N = 200$

- (Recorded) target response
- Modeled

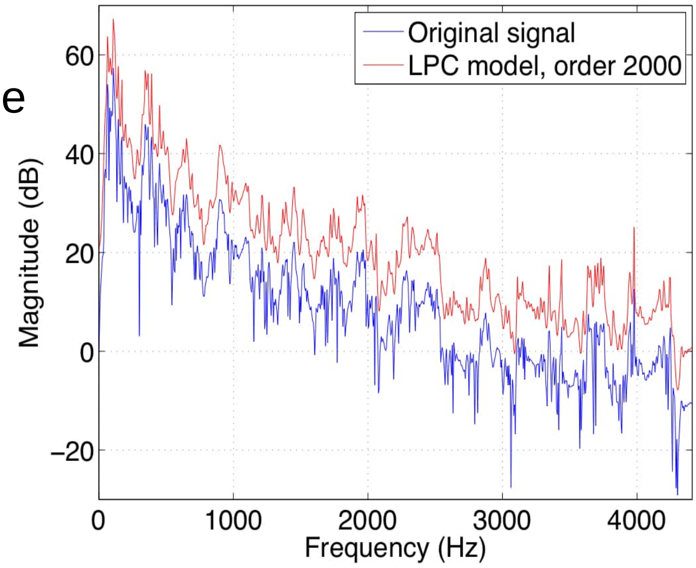


Aalto University School of Electrical Engineering

54

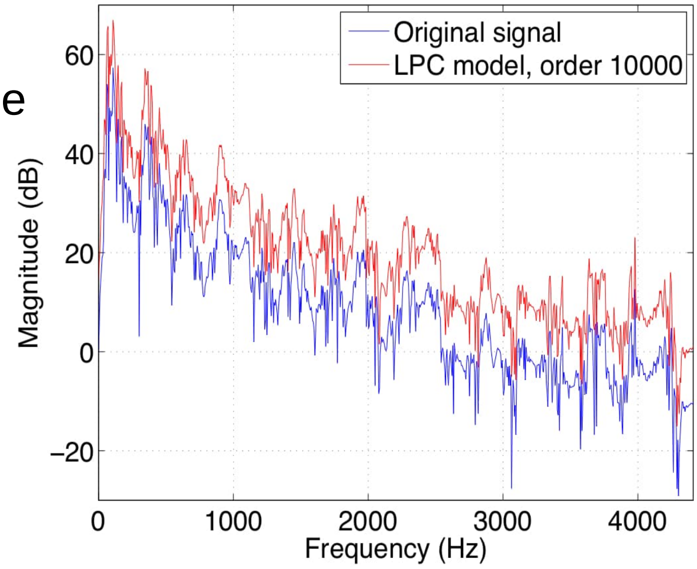
Piano Soundboard Modeling with LPC, $N = 2000$

- (Recorded) target response
- Modeled



Piano Soundboard Modeling with LPC, $N = 10000$

- (Recorded) target response
- Modeled



Prony's Method

$$H(z) = \frac{B(z)}{A(z)}$$

- Design an IIR filter from a given impulse response
- Solve denominator (all-pole IIR) coefficients using linear prediction of order *N*

$$\sum_{k=0}^{N} a_k z^{-k}$$

- Solve numerator (FIR) coefficients by matching the *M* + 1 first samples of the impulse response

$$\sum_{k=0}^{M} b_k z^{-k}$$

- In Matlab: `prony.m`

"Duel of Digital Filters"

- Why is a FIR filter better?
- Why is an IIR filter better?
- …in audio applications
- Brainstorm in small groups:
  - Some groups collect advantages of **FIR** filters
  - Other groups collect advantages of **IIR** filters
  - Please list your ideas, so that we can write them down

"Duel of Digital Filters"

FIR Advantages

- Always stable
- Linear phase

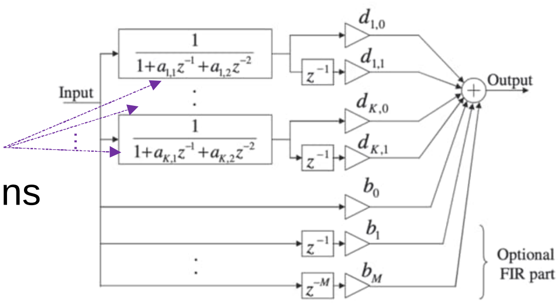
IIR Advantages

- More efficient computationally
- Smaller order
- Parametric control possible
- Analog-like phase response
- Good for real-time applications

2 - 5

Fixed-Pole IIR Filters

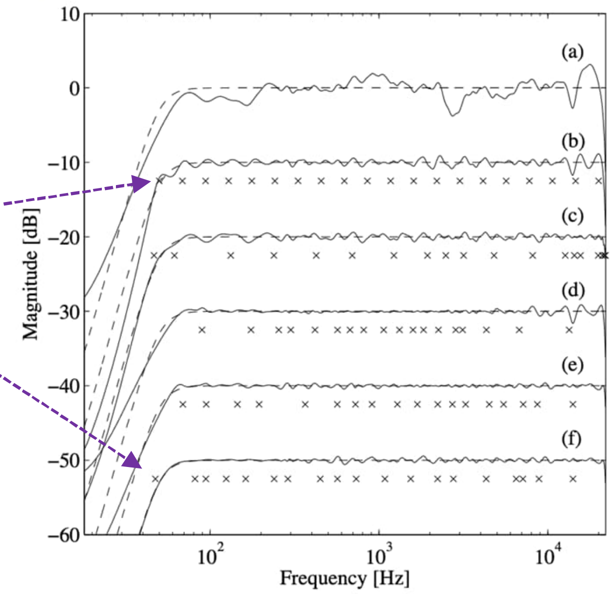
- Perceptually-motivated parallel IIR structure proposed by Bank (2007)
- First, select pole frequencies, e.g. logarithmically: they form basis functions
- Then, solve zeros with LS method, as weights for the basis functions (d_0, d_1)
- That's the final implementation directly; no conversions needed
- Still, the efficiency and resolution can be as good as with warped filters



Source: B. Bank, Audio Equalization with Fixed-Pole Parallel Filters: An Efficient Alternative to Complex Smoothing, *Journal of AES*, 2013

Fixed-Pole IIR Filters

- Frequency resolution can be controlled by the pole positions (x)
  - o Logarithmic pole positions gives a logarithmic frequency resolution
  - o Alternatively, the resolution can be tweaked manually (arbitrarily) to fit the target response



B. Bank, "Loudspeaker and room response equalization using parallel filters: Comparison of pole positioning strategies," in *Proc. AES 51st Int. Conf.*, 2013.

Fixed-Pole IIR Filters

- When the poles are set → linear optimization problem
- Least-squares solution can be used to solve zeros!

$$\mathbf{h}_{\mathrm{t}} = \mathbf{M}\mathbf{p},$$

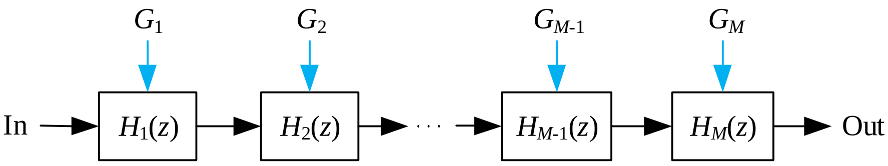
where $\mathbf{h}_{\mathrm{t}}$ is the target impulse response, $\mathbf{M}$ is the modeling signal matrix, and $\mathbf{p}$ contains the zeros (and FIR part coeffs.)

$$\mathbf{p}_{\mathrm{opt}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{h}_{\mathrm{t}}$$

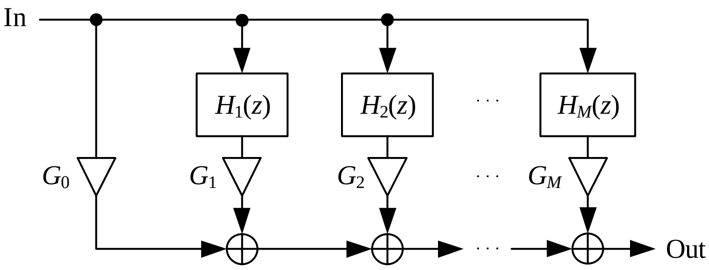
- Design can be done either in the time or frequency domain
  - o Matrix $\mathbf{M}$ contains either impulse responses or the transfer functions of the all-pole second-order sections (basis functions)
  - o In frequency domain: magnitude-only design & frequency-dependent weighting

Graphic Equalizer Design

- Cascade structure: use equalizing filters; well understood today

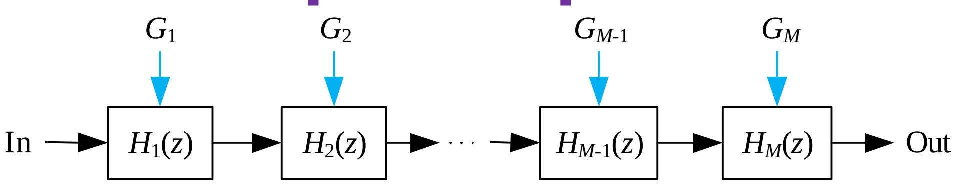


- Parallel structure: band of resonator (bandpass) filters; difficult to design



63

Cascade Graphic Equalizer

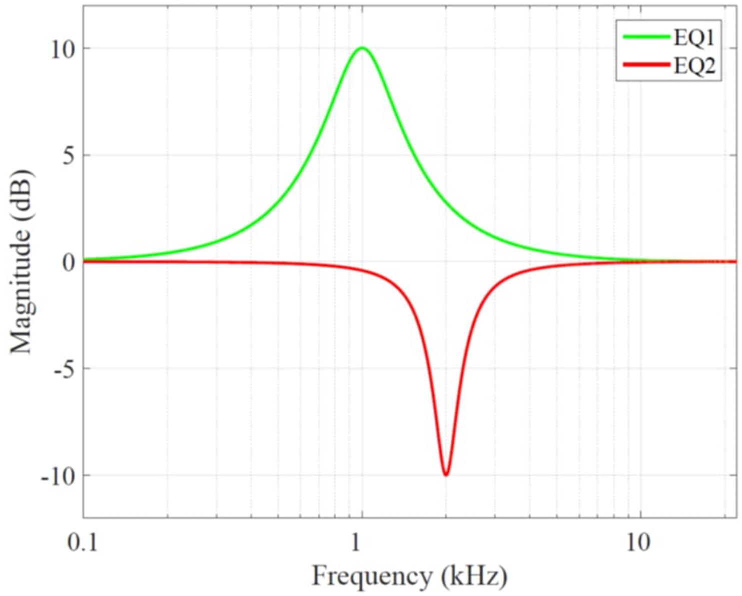


- Each subfilter is an equalizing filter with fixed f_c and Q
- Gain G_k at each peak can be adjusted (elsewhere ≈ 0 dB)
- Logarithmic frequency division is used, which is closely related to the auditory frequency resolution
- Problem: **interaction** between bands…

64

Cascade Graphic Equalizer, Ex. #1

- Two EQ filters
  - EQ1: f_c = 1 kHz & $G = 10$ dB
  - EQ2: f_c = 2 kHz & $G = -10$ dB

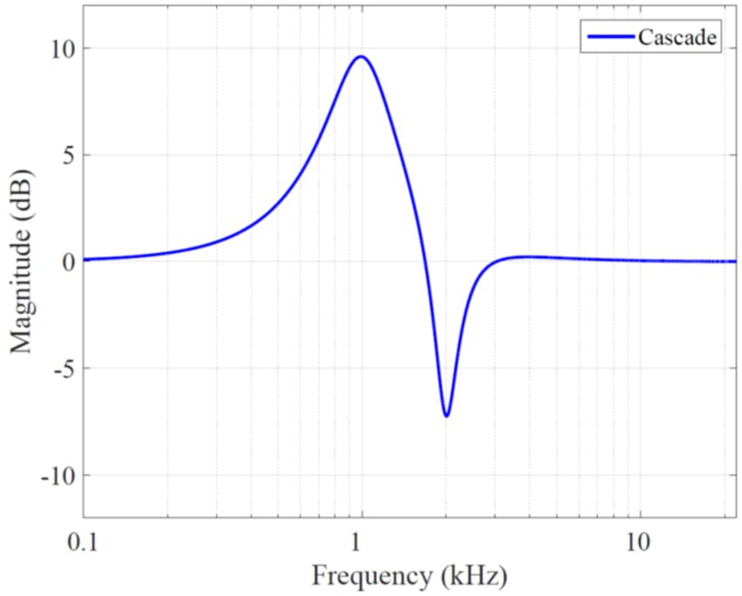


Cascade Graphic Equalizer, Ex. #1

- Responses interact so the overall response is biased:

~~10 dB and –10 dB~~

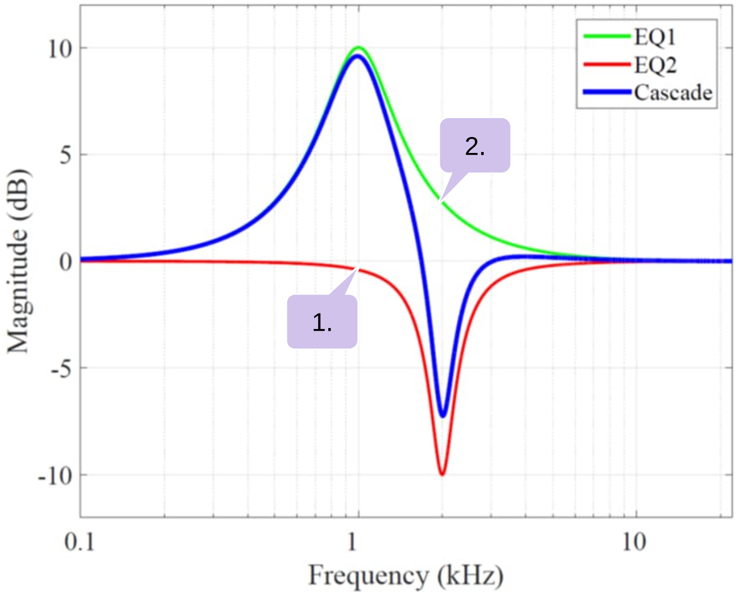
9.6 dB and -7.1 dB



Cascade Graphic Equalizer, Ex. #1

- Interaction problem:
1. Gain of EQ2 at 1 kHz is < 0 dB
2. Gain of EQ1 at 2 kHz is > 0 dB

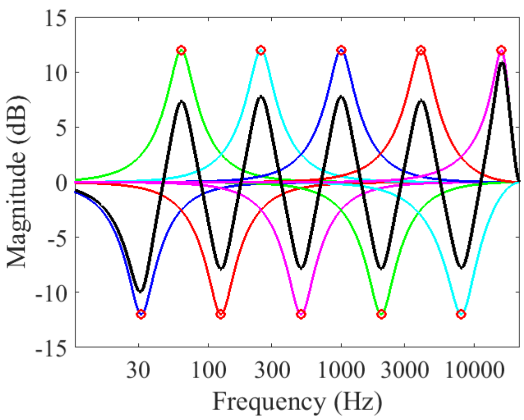
- Both filters affect the gain at all frequencies!!! (not only around their center frequency)



Aalto University School of Electrical Engineering

Cascade Graphic EQ for Octave Bands

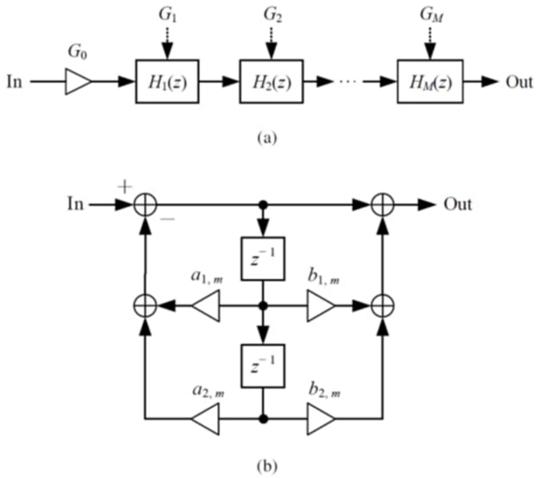
- Naïve design leads to severe approximation problems
  - o Filter gains = command gains



Aalto University School of Electrical Engineering

Novel Cascade Graphic EQ Design

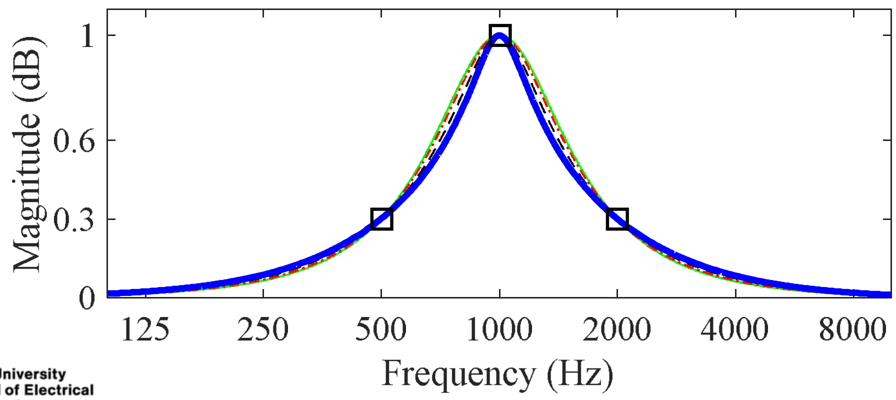
- It is desirable to use 2nd-order band filters
- The most accurate cascade design uses different peak filter shapes than previously (Välimäki & Liski 2017, Liski *et al*. 2019)
- The magnitude response is optimized at center frequencies and at intermediate frequencies



Second-order cell $H_k(z)$

Cascade GEQ with Variable-Q Filters

- Force the magnitude response to be "constant" at its f_c and the 2 neighboring f_c's (Välimäki & Liski 2017)
- Approximately constant basis functions are obtained



Normalized magnitude responses of filters with different gain (-12 dB…12 dB)

Interaction Matrix

- Interaction matrix **B** shows how much each band filter leaks to neighboring bands (Oliver & Jot 2015; Välimäki & Liski 2017)
- Computed with discrete-time Fourier transform, when the band filters are adjusted to a prototype gain
- Matrix **B** is normalized so that it shows how much 1 dB of gain leaks from each band filter to other frequency points

Filter

$$\mathbf{B}_0 = \begin{pmatrix} 1.0 & 0.30 & 0.081 & 0.021 & 0.0052 & 0.0013 & 3.2\times10^{-4} & 7.8\times10^{-5} & 1.7\times10^{-5} & 1.9\times10^{-6} \\ 0.30 & 1.0 & 0.30 & 0.081 & 0.021 & 0.0052 & 0.0013 & 3.1\times10^{-4} & 6.8\times10^{-5} & 7.5\times10^{-6} \\ 0.081 & 0.30 & 1.0 & 0.30 & 0.081 & 0.021 & 0.0052 & 0.0013 & 2.7\times10^{-4} & 3.0\times10^{-5} \\ 0.021 & 0.081 & 0.30 & 1.0 & 0.30 & 0.081 & 0.021 & 0.0050 & 0.0011 & 1.2\times10^{-4} \\ 0.0053 & 0.021 & 0.081 & 0.30 & 1.0 & 0.30 & 0.080 & 0.020 & 0.0043 & 4.8\times10^{-4} \\ 0.0013 & 0.0053 & 0.021 & 0.081 & 0.30 & 1.0 & 0.30 & 0.078 & 0.017 & 0.0019 \\ 3.4\times10^{-4} & 0.0014 & 0.0054 & 0.022 & 0.083 & 0.31 & 1.0 & 0.30 & 0.071 & 0.0081 \\ 8.2\times10^{-5} & 3.3\times10^{-4} & 0.0013 & 0.0052 & 0.021 & 0.08 & 0.30 & 1.0 & 0.27 & 0.033 \\ 2.0\times10^{-5} & 7.9\times10^{-5} & 3.2\times10^{-4} & 0.0013 & 0.0050 & 0.020 & 0.078 & 0.29 & 1.0 & 0.16 \\ 4.6\times10^{-6} & 1.9\times10^{-5} & 7.4\times10^{-5} & 3.0\times10^{-4} & 0.0012 & 0.0047 & 0.019 & 0.072 & 0.26 & 1.0 \end{pmatrix}$$

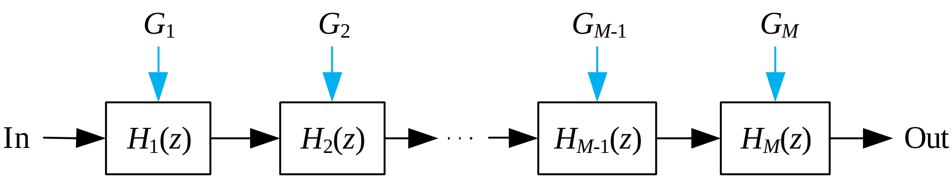
Frequency

Cascade GEQ Based on 2nd-Order Filters

- Use inverse matrix of **B** to solve the optimal dB-gains in the least-squares (LS) sense

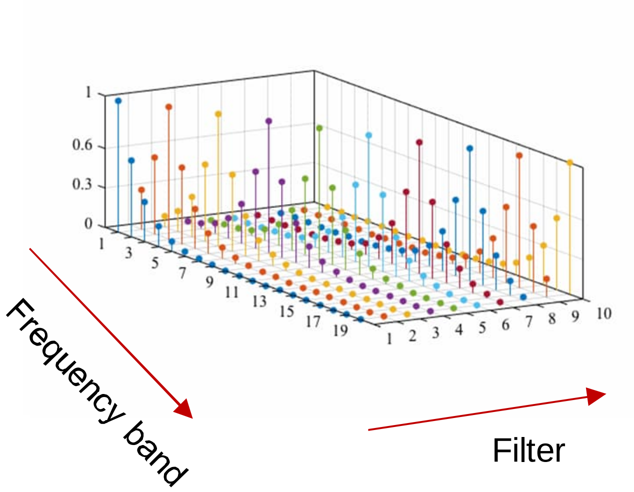
$$\mathbf{g}_{\mathrm{opt}} = \mathbf{B}^{-1}\mathbf{g}$$

- After optimization: filter gains ≠ command gains



Graphic Equalizer Design Using Interaction Matrix

Interaction matrix $\mathbf{B}$



Optimize dB gains in LS sense using pseudoinverse:

$$\mathbf{g} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{t}_1,$$

where

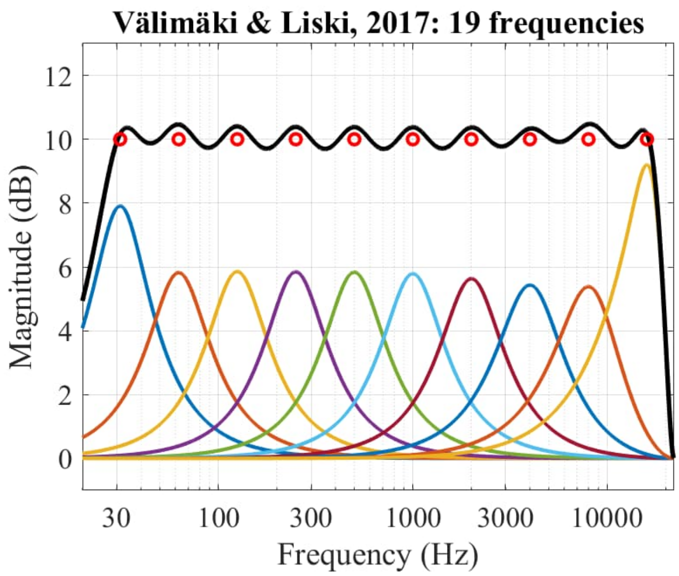
$\mathbf{t}_1$ = target gains (dB)

$\mathbf{B}$ = interaction matrix

$\mathbf{g}$ = optimized filter gains (dB)

Aalto University School of Electrical Engineering

Octave Equalizer: Välimäki & Liski 2017

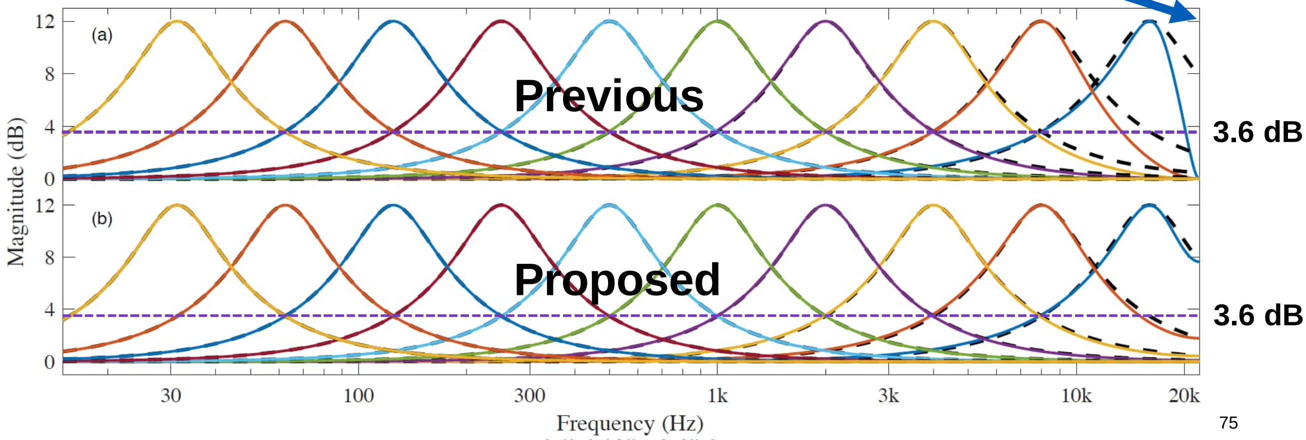


Peak gains of band filters are now very different from target gains o

Aalto University School of Electrical Engineering

Design with Symmetric Band Filters

- The latest version of the GEQ design (Liski *et al.*, IEEE WASPAA-19) uses symmetric filters having a prescribed Nyquist gain: the ripple is reduced to about 0.8 dB
- The ideal target curve (---) is obtained with oversampling



75

75

Conclusions

- Properties of filters: magnitude and phase
  - Phase delay and group delay are identical when the phase is linear; otherwise, they are different
- Various audio filter design methods are available
  - For FIR filters: Least squares, weighted least squares, frequency sampling, impulse response truncation/windowing
  - The magnitude and phase responses of the FIR filter can be tuned independently
  - For IIR filters: linear prediction, Prony's method, fixed-pole design, graphic EQ
- Auditory resolution can be applied
  - With frequency weighting or warping or with a log frequency scale
- The most suitable method depends on the applications
- Typical design problems are equalizers, HRTF filters, reverberation algorithms, and musical-instrument responses

Aalto University School of Electrical Engineering

February 2, 2024
76

76

Literature

- Easily accessible online book (check this out!)
  - Smith, J. O., "Introduction to Digital Filters with Audio Applications," http://ccrma.stanford.edu/~jos/filters/
- Filter design
  - Parks, T. W. and Burrus, C. S. Digital Filter Design. John Wiley & Sons, 1987.
  - Mitra, S. K. Digital Signal Processing. McGraw-Hill, 2002.
- Frequency warping
  - A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Frequency-warped signal processing for audio applications," Journal of the Audio Engineering Society, vol. 48, no. 11, pp. 1011–1031, Nov. 2000.
  - Smith, J. O. III and Abel, J. S. "The Bark and ERB bilinear transforms," IEEE Trans. Speech and Audio Process., Vol. 7, No, 6, pp. 697-708, Nov. 1999.
- Graphic equalizers
  - J. Liski, "Equalizer Design for Sound Reproduction," DSc thesis, Aalto University, 2020.

Aalto University
School of Electrical
Engineering

References

- J. S. Abel and D. P. Berners, "Filter design using second-order peaking and shelving sections," in *Proc. Int. Computer Music Conf.*, Miami, FL, USA, Nov. 2004.
- B. Bank, "Direct design of parallel second-order filters for instrument body modeling," in *Proc. Int. Computer Music Conf.*, Copenhagen, Denmark, Aug. 2007.
- B. Bank, "Perceptually motivated audio equalization using fixed-pole parallel second-order filters," *IEEE Signal Processing Letters*, vol. 15, pp. 477-480, 2008.
- B. Bank, "Audio equalization with fixed-pole parallel filters: An efficient alternative to complex smoothing," *Journal of Audio Engineering Society*, vol. 61, pp. 39-49, 2013.
- B. Bank, "Loudspeaker and room response equalization using parallel filters: Comparison of pole positioning strategies," in *Proc. AES 51st International Conf.*, Helsinki, Finland, Aug. 2013.
- B. Bank, "Warped, Kautz, and Fixed-Pole Parallel Filters: A Review," *J. Audio Eng. Soc.*, vol. 70, no. 6, pp. 414–434, (2022 June). https://doi.org/10.17743/jaes.2022.0016
- Huopaniemi, J., Zacharov, N., and Karjalainen, M., " Objective and subjective evaluation of head-related transfer function filter design," in *Proc. 105th AES Convention*, preprint 4805, San Francisco, CA, USA, September 26-29, 1998. Invited paper.
- J. Huopaniemi, *Virtual Acoustics and 3-D Sound in Multimedia Signal Processing*. Doctoral thesis. Report no. 53 / Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing. Espoo, Finland, 1999.
- R. J. Oliver and J.-M. Jot, "Efficient multi-band digital audio graphic equalizer with accurate frequency response control," in *Proc. AES 139th Conv*., New York, NY, USA, Oct. 2015.
- Lehtonen, H.-M., Rauhala, J., and Välimäki, V. "Sparse multi-stage loss filter design for waveguide piano synthesis," In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 331-334, New Paltz, NY, USA, October 16-19, 2005.
- J. Liski, J. Rämö, and V. Välimäki and, "Graphic equalizer design with symmetric biquad filters," in *Proc. IEEE WASPAA*, Oct. 2019.
- V. Välimäki and J. D. Reiss, "All about audio equalization: Solutions and frontiers," *Appl. Sci.*, vol. 6, no. 5, p. 129, May 2016.
- V. Välimäki and J. Liski, "Accurate cascade graphic equalizer," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 176–180, Feb. 2017.

Aalto University
School of Electrical
Engineering

Learning Diary by Next Thursday

- Here are some tips what you could do this time (one of them, not all)
- Convert a linear-phase FIR filter to minimum phase using the Hilbert transform and/or cepstral windowing method and create plots to compare
- Auditory smoothing: compare what a few smoothing techniques can do to a “random” magnitude response, such as 1/3-octave smoothing
- Compare the effect of different window functions on the DFT spectrum (fft.m) of a pure tone or another signal (to truncate it)
- Compare two or more digital filters: the impulse response, the magnitude & phase responses (freqz.m), and/or the group delay (grpdelay.m), and the poles and zeros (zplane.m) could be plotted and studied

Learning Diaries – Submitting Code

- Ensure that all files, including audio and functions, are included in the submitted material
- Verify that the code runs successfully before submission
- When submitting functions, include a file demonstrating their usage
- When renaming files, declare the file extension accurately
- Utilize file compression in .zip format whenever possible
- Avoid using external packages.