Signal Processing

Lecture 9: Noise Models & Linear Filters I

Konstantinos Chatzilygeroudis - costashatz@upatras.gr

Department of Electrical and Computer Engineering
University of Patras

Template made by Panagiotis Papagiannopoulos

Noise and Its Models

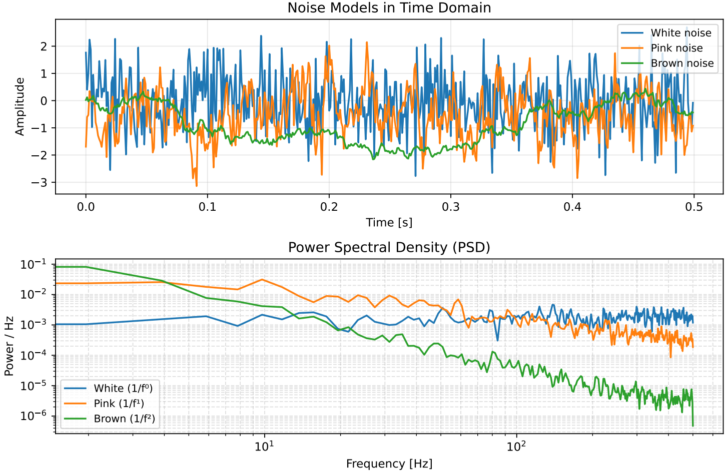
What is Noise?

- **Noise** refers to any unwanted or unpredictable component added to a signal.
- Typically modeled as a **random process** with specific statistical properties.

Why Model Noise?

- To **predict**, **filter**, or **estimate** signals corrupted by randomness.
- Accurate noise models are essential for:
  - Filter design (e.g., Wiener/Kalman filters)
  - System identification
  - Sensor fusion and denoising

Noise Models Example



Noise added to a clean signal; white, pink, brown noise differ in spectral content.

White Noise

Definition

- A discrete-time signal $x[n]$ is called **white noise** if its samples are:
  - **Uncorrelated:** $\mathbb{E}[x[n]x[m]] = 0$ for $n \neq m$
  - **Zero-mean:** $\mathbb{E}[x[n]] = 0$
  - **Constant variance:** $\mathbb{E}[x^2[n]] = \sigma^2$
- Autocorrelation function:

$$R_{xx}[m] = \sigma^2 \, \delta[m]$$

- Power Spectral Density (PSD):

$$S_{xx}(e^{j\omega}) = \sigma^2$$

  (flat spectrum; equal power at all frequencies)

Interpretation

- Represents a **completely unpredictable** process.
- Serves as a building block for more complex noise models (colored noise, AR models).
- Often modeled as Gaussian: $x[n] \sim \mathcal{N}(0, \sigma^2)$.

Colored Noise

Definition. A random process $x[n]$ is called *colored noise* when its power spectral density (PSD) is not flat:

$$S_{xx}(e^{j\omega}) \neq \text{constant}.$$

Equivalently, its samples are correlated in time:

$$R_{xx}[m] \neq 0 \quad \text{for some } m \neq 0.$$

Model. Colored noise can be obtained by passing white noise $w[n]$ through a stable linear time-invariant (LTI) filter $h[n]$:

$$x[n] = h[n] * w[n].$$

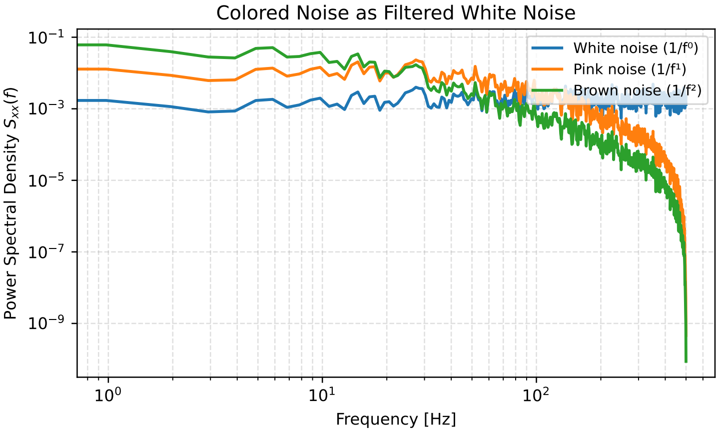
Spectral relationship:

$$S_{xx}(e^{j\omega}) = \left|H(e^{j\omega})\right|^2 S_{ww}(e^{j\omega}),$$

where:

- $S_{ww}(e^{j\omega})$: PSD of the input (white noise, flat)
- $H(e^{j\omega})$: frequency response of the shaping filter
- $S_{xx}(e^{j\omega})$: PSD of the resulting colored noise

Colored Noise: Example



Example: Pink and brown noise obtained by filtering white noise through low-pass filters.

Pink Noise Model

Pink Noise (1/f noise)

- Empirical model with a power spectral density

$$S_{xx}(e^{j\omega}) \propto \frac{1}{|\omega|^{\alpha}}, \quad \alpha \approx 1$$

- Equal power per logarithmic frequency band (e.g., per octave).
- Can be approximated by filtering white noise through a **first-order low-pass filter**:

$$x_{\text{pink}}[n] = h_{\text{pink}}[n] * w[n]$$

with

$$H_{\text{pink}}(e^{j\omega}) = \frac{1}{\sqrt{1 + (\omega/\omega_c)^2}}$$

where ω_c controls the corner frequency: the frequency where the PSD transitions from flat to $\frac{1}{f}$ slope.

Brown Noise Model

Brown (Red) Noise

- Obtained by integrating white noise:

$$x_{\text{brown}}[n] = \sum_{k=-\infty}^{n} w[k] \quad \Rightarrow \quad x_{\text{brown}}[n] = x_{\text{brown}}[n-1] + w[n]$$

- Corresponding transfer function:

$$H_{\text{brown}}(e^{j\omega}) = \frac{1}{1 - e^{-j\omega}}$$

- Power spectral density:

$$S_{xx}(e^{j\omega}) = \frac{S_{ww}(e^{j\omega})}{|1 - e^{-j\omega}|^2} \approx \frac{\sigma_w^2}{\omega^2}$$

- Hence, Brown noise $\Rightarrow 1/f^2$ spectrum.

Colored Noise from AR Processes

Model (AR(p)):

$$x[n] = -\sum_{k=1}^{p} a_k\, x[n-k] + w[n], \qquad \mathbb{E}[w[n]] = 0,\; R_{ww}[m] = \sigma_w^2 \delta[m]$$

PSD of the AR(p) process:

$$S_{xx}(e^{j\omega}) = |H(e^{j\omega})|^2 S_{ww}(e^{j\omega}) = \frac{\sigma_w^2}{\left|A(e^{j\omega})\right|^2} = \frac{\sigma_w^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-j\omega k}\right|^2}$$

Special case (AR(1)):

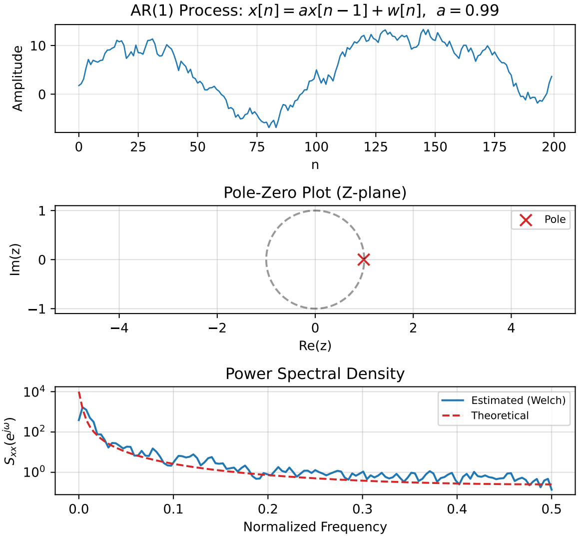
$$x[n] = a\, x[n-1] + w[n] \qquad \Rightarrow \qquad S_{xx}(e^{j\omega}) = \frac{\sigma_w^2}{\left|1 + ae^{-j\omega}\right|^2}$$

$$R_{xx}[m] = \frac{\sigma_w^2}{1-a^2}\, a^{|m|} \quad (\text{for } |a| < 1)$$

Interpretation:

- Poles are the roots of $A(z) = 0$; **stability** requires all poles inside the unit circle.
- The PSD peaks near the pole angles; bandwidth narrows as poles approach the unit circle.
- Hence, AR processes produce *colored* spectra by shaping white noise with $H(e^{j\omega})$.

Colored Noise from AR Processes: Example



Digital Linear Filters

Linear filters satisfy the superposition principle:

$$y[n] = \sum_{k} h[k]\, x[n-k],$$

where $h[k]$ is the impulse response.
They are broadly divided into two classes depending on the duration of $h[k]$ and the system structure:

1. FIR (Finite Impulse Response) Filters

$$y[n] = \sum_{k=0}^{M} b_k\, x[n-k]$$

- *Nonrecursive*: output depends only on current and past inputs.
- Always *stable* and can achieve *exact linear phase* (if $h[k]$ symmetric).
- Require higher order M for sharp transitions.
- Ideal for data smoothing, delay lines, and phase-sensitive applications.

Digital Linear Filters (2)

2. IIR (Infinite Impulse Response) Filters

$$y[n] = \sum_{k=0}^{M} b_k \, x[n-k] - \sum_{k=1}^{N} a_k \, y[n-k]$$

- *Recursive*: feedback introduces infinite impulse response.
- Can achieve sharp frequency selectivity with low order.
- May exhibit *nonlinear phase* and possible stability issues.
- Derived from analog prototypes (Butterworth, Chebyshev, Elliptic).

Summary:

- FIR: stable, linear-phase, high order
- IIR: efficient, recursive, nonlinear-phase

Finite Impulse Response (FIR) Filters

Definition. An FIR filter is a linear, time-invariant system whose impulse response $h[n]$ has finite duration: $h[n] = 0 \quad \text{for } n < 0 \text{ or } n > M$.
The input-output relation is

$$y[n] = \sum_{k=0}^{M} b_k \, x[n-k]$$

Frequency response:

$$H(e^{j\omega}) = \sum_{k=0}^{M} b_k e^{-j\omega k}$$

Key properties:

- Always **stable** (finite impulse response).
- Can be made **linear-phase** if $h[n]$ is symmetric or antisymmetric.
- Suitable for both **causal real-time** and **non-causal offline** processing.

Common uses:

- Smoothing and denoising (e.g., moving-average filters)
- Frequency-selective filtering (low-pass, high-pass, band-pass)
- Implementing digital filters from desired magnitude responses

Moving Average (MA) Filter

Definition. An M-point moving average (MA) filter computes the average of the most recent M samples:

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]$$

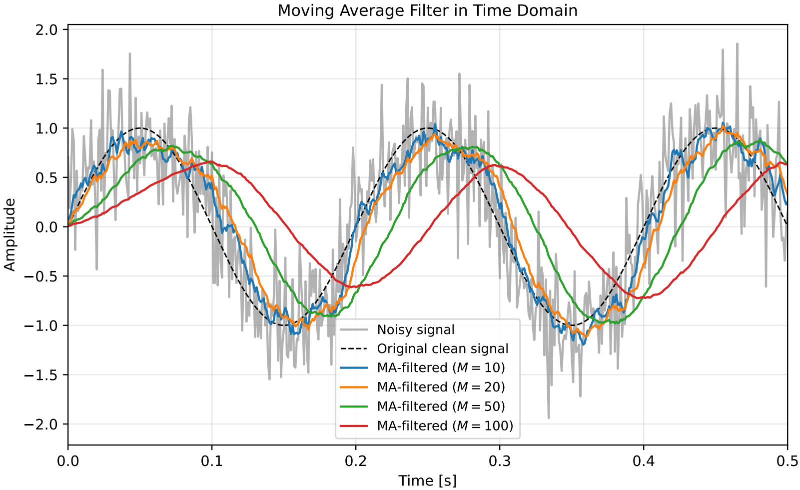
Impulse response:

$$h[n] = \begin{cases} \frac{1}{M}, & 0 \leqslant n \leqslant M-1 \\ 0, & \text{otherwise} \end{cases}$$

Properties:

- Low-pass filter that attenuates rapid fluctuations (noise).
- Linear-phase FIR filter (symmetric $h[n]$).
- Increasing M improves smoothing but increases delay.

Moving Average (MA) Filter: Example



Example: smoothing a noisy sinusoid using a moving average filter.

Band-Pass FIR Filter

Definition. A band-pass FIR filter passes frequencies within a desired range $[\omega_1, \omega_2]$ and attenuates all others.

$$H(e^{j\omega}) = \begin{cases} 1, & \omega_1 \leqslant |\omega| \leqslant \omega_2 \\ 0, & \text{otherwise} \end{cases}$$

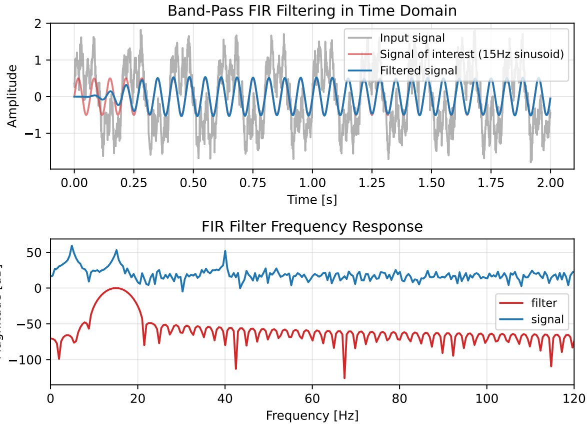
In practice, $H(e^{j\omega})$ is approximated by a finite-length impulse response $h[n]$, often designed using the *window method*:

$$h[n] = h_{\text{ideal}}[n]\, w[n]$$

where $w[n]$ is a tapering window (e.g., Hamming).

Band-Pass FIR Filter: Example

- Input: mixture of low and high-frequency sinusoids + noise.
- FIR filter designed with desired passband $[f_1, f_2]$.
- Output: only components within $[f_1, f_2]$ remain.



Example: band-pass FIR filter isolating a 13-27 Hz component from a mixed signal.

Filter Length and Sampling Frequency

Filter length M and sampling frequency f_s are coupled: they jointly determine the effective transition bandwidth, delay, and computational cost of an FIR filter.

1. **Frequency resolution and transition width**

$$\text{Transition bandwidth } \Delta f \approx \frac{f_s}{M}$$

- For a given f_s, increasing $M \Rightarrow$ narrower $\Delta f \rightarrow$ sharper filter.
- For a fixed M, increasing $f_s \Rightarrow$ wider $\Delta f \rightarrow$ relatively smoother response.

2. **Delay and computational complexity**

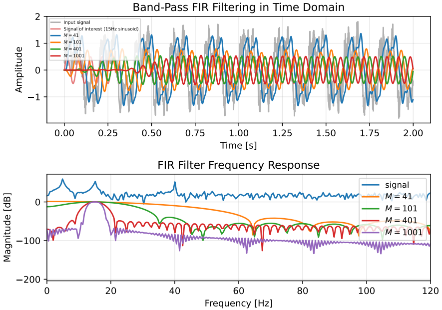
$$\text{Group delay } \tau_g \approx \frac{M-1}{2f_s}$$

- Large M or small $f_s \Rightarrow$ higher temporal delay.
- For real-time systems, delay may become unacceptable if M/f_s is large.

Filter Length and Sampling Frequency (2)

3. Summary of interactions:

Condition	Effect on Filter	Interpretation
Small M, high f_s	Wide transition band	Fast, low-selectivity filter
Large M, high f_s	Sharp transition, small normalized delay	Precise but computationally heavy
Small M, low f_s	Broad transitions, possible aliasing	Weak spectral control
Large M, low f_s	Sharp but large absolute delay	Slow response, not real-time suitable



Band-pass FIR filter with different M lengths.

Infinite Impulse Response (IIR) Filters

Definition. A causal IIR filter has a recursive difference equation

$$y[n] = \sum_{k=0}^{M} b_k \, x[n-k] \; - \; \sum_{k=1}^{N} a_k \, y[n-k],$$

with (generally) infinite-duration impulse response.

Transfer function and frequency response.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}, \qquad H(e^{j\omega}) = H(z)\big|_{z=e^{j\omega}}.$$

Causality and stability.

- Causal if $h[n] = 0$ for $n < 0$ (equivalently, a proper real-time realization exists).
- BIBO-stable iff **all poles** of $H(z)$ lie **strictly inside** the unit circle.

Infinite Impulse Response (IIR) Filters (2)

Why IIR? What can we do with them?

- Achieve **sharp frequency selectivity** with **few coefficients** (efficient).
- **Main families**: Butterworth (maximally flat), Chebyshev I/II (equiripple), Elliptic (equiripple pass/stop).
- Implement **low/high/band-pass** and notch filters; **biquad cascades** for robust designs.

Trade-offs.

- Potential **nonlinear phase** (distortion); **stability sensitivity** to coefficients.
- **Finite-precision** effects: pole drift, limit cycles; prefer biquad (SOS) implementations.

Main Families of IIR Filters (Discrete-Time Domain)

- **Butterworth (maximally flat):**

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \varepsilon^2 \left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

- Smooth, monotonic magnitude; no ripples.
- Linear-phase approximation over small ranges.

- **Chebyshev Type I (equiripple passband):**

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \varepsilon^2 C_N^2\left(\frac{\Omega}{\Omega_c}\right)}, \quad C_N(x) = \begin{cases} \cos\left(N \cos^{-1} x\right), & |x| \leqslant 1 \\ \cosh\left(N \cosh^{-1} x\right), & |x| > 1 \end{cases}$$

- Ripples in passband, monotonic stopband.
- Sharper roll-off than Butterworth for same N.

- **Chebyshev Type II (equiripple stopband):**

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \frac{1}{\varepsilon^2 C_N^2\left(\frac{\Omega_c}{\Omega}\right)}}$$

- Flat passband, rippled stopband.
- Used when stopband attenuation is critical.

Implementation of IIR Filters

Notes.

- These filters differ only in *magnitude shaping*; all have recursive realizations of the form above.
- Digital implementations use biquads (2nd-order sections) for numerical stability.

Biquad (second-order section; SOS)

$$H_i(z) = \frac{b_{0,i} + b_{1,i}z^{-1} + b_{2,i}z^{-2}}{1 + a_{1,i}z^{-1} + a_{2,i}z^{-2}}$$

Why Biquads? The issue with high-order direct form

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

Large $N \Rightarrow$ wide dynamic range in coefficients, rounding/quantization $\Rightarrow$ pole drift, response distortion, potential instability.

Idea: Implement the full filter as a *product* of biquads:

$$H(z) = \prod_{i=1}^{L} H_i(z), \qquad L = \left\lceil \frac{N}{2} \right\rceil$$

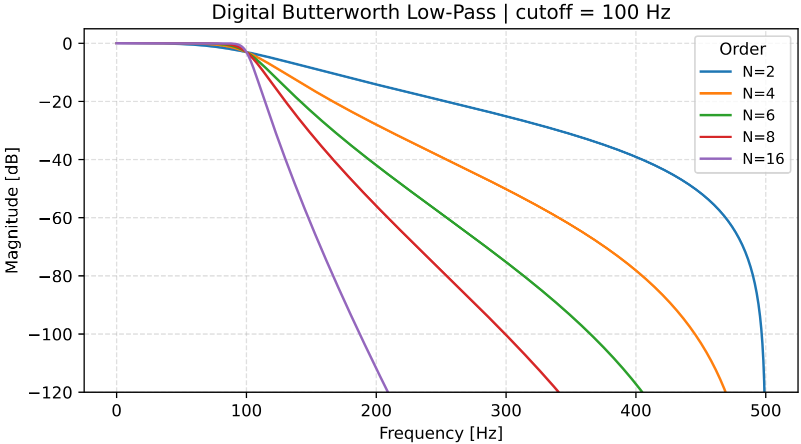
Definition (discrete time). A Butterworth prototype has maximally flat passband; its digital transfer is

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}},$$

with order N (number of poles).
Magnitude shape. For low-pass, increasing N yields steeper roll-off at the cutoff Ω_c (monotonic, no ripples). High-pass and band-pass responses inherit the same monotonic skirts.

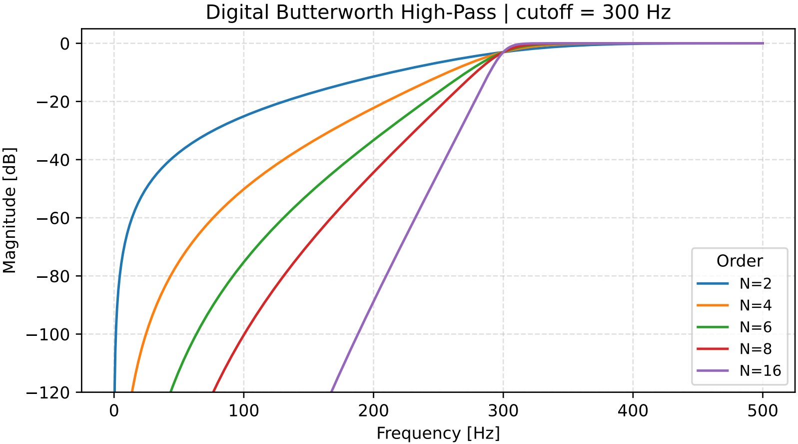
Digital Butterworth Filters: Low-pass

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \varepsilon^2 \left(\frac{\Omega}{\Omega_c}\right)^{2N}}, \quad \Omega_c = 100\,\text{Hz}$$



Digital Butterworth Filters: High-pass

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \varepsilon^2 \left(\frac{\Omega_c}{\Omega}\right)^{2N}}, \quad \Omega_c = 300\,\text{Hz}$$



Digital Chebyshev Filters

Chebyshev Type I (equiripple passband). Magnitude:

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \varepsilon^2 \, C_N^2\!\left(\frac{\Omega}{\Omega_c}\right)}.$$

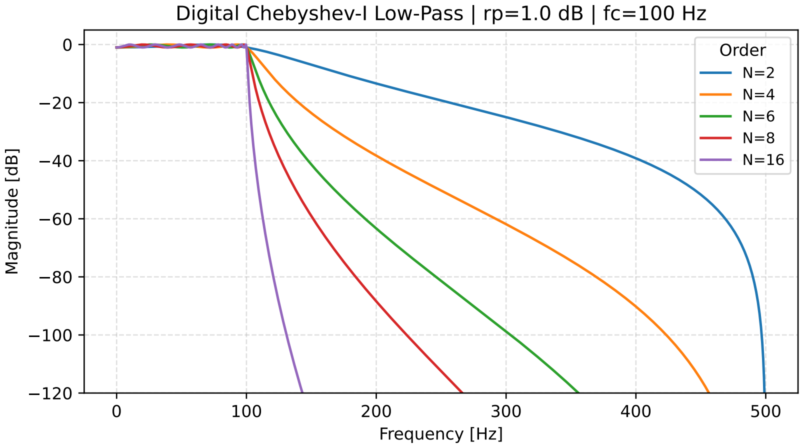
Parameters: order N, passband ripple ε (e.g., 1 dB), cutoff Ω_c.
Chebyshev Type II (inverse; equiripple stopband).

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \frac{1}{\varepsilon^2 \, C_N^2\left(\frac{\Omega_c}{\Omega}\right)}}.$$

Parameters: order N, stopband attenuation ε (e.g., 60 dB), cutoff Ω_c.
Behavior. Increasing N steepens the transition. Type I allows passband ripple (sharper than Butterworth); Type II keeps passband flat and ripples the stopband.

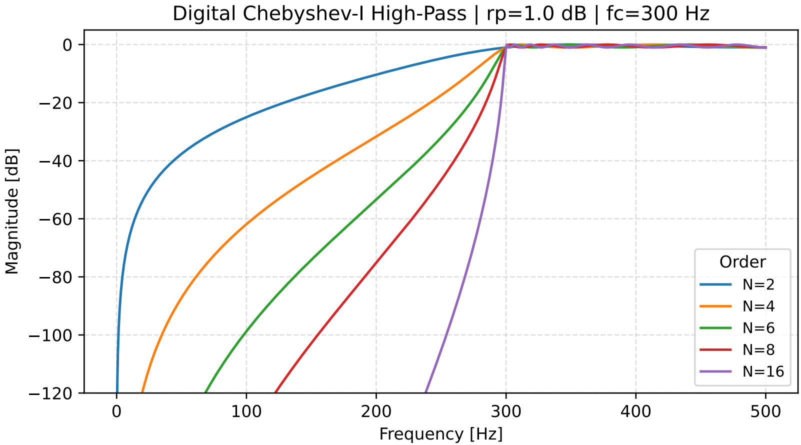
Digital Chebyshev (Type I) Filters: Low-pass

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \varepsilon^2 \, C_N^2\!\left(\frac{\Omega}{\Omega_c}\right)}, \quad \Omega_c = 100\,\text{Hz}$$



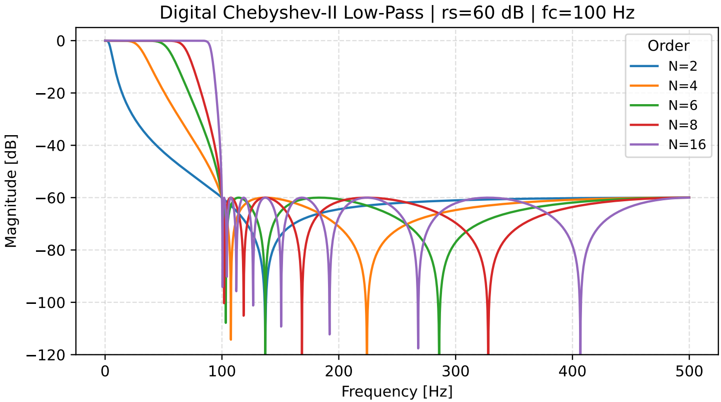
Digital Chebyshev (Type I) Filters: High-pass

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \varepsilon^2 \, C_N^2\!\left(\frac{\Omega_c}{\Omega}\right)}, \quad \Omega_c = 300\,\text{Hz}$$



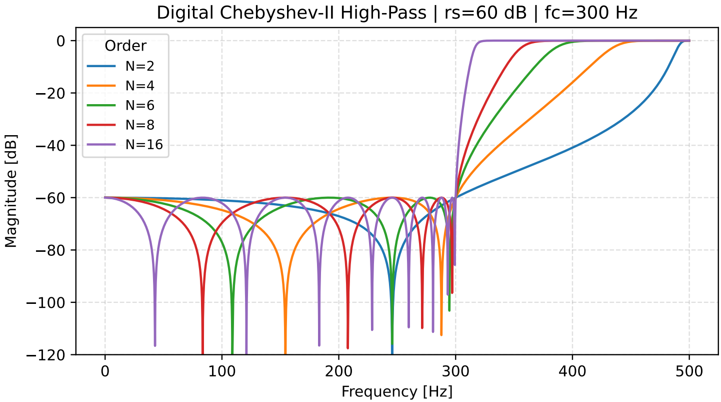
Digital Chebyshev (Type II) Filters: Low-pass

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \frac{1}{\varepsilon^2 \, C_N^2\left(\frac{\Omega_c}{\Omega}\right)}}, \quad \Omega_c = 100\,\text{Hz}$$



Digital Chebyshev (Type II) Filters: High-pass

$$|H(e^{j\Omega})|^2 = \frac{1}{1 + \frac{1}{\varepsilon^2\, C_N^2\left(\frac{\Omega}{\Omega_c}\right)}}, \quad \Omega_c = 300\,\text{Hz}$$



IIR Filters: Magnitude, Phase, and Group Delay

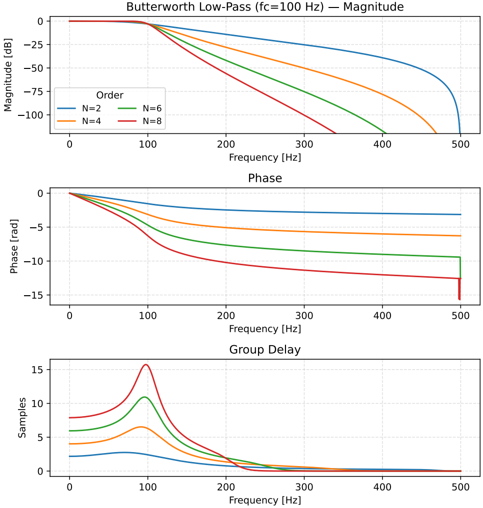
Setup. IIR (Butterworth) filters (orders $N = \{2, 4, 6, 8\}$), designed in discrete time with cutoff(s) specified in Hz. We examine:

$$|H(e^{j\Omega})|, \quad \phi(\Omega) = \arg H(e^{j\Omega}), \quad \tau_g(\Omega) = -\frac{d\phi(\Omega)}{d\Omega}.$$

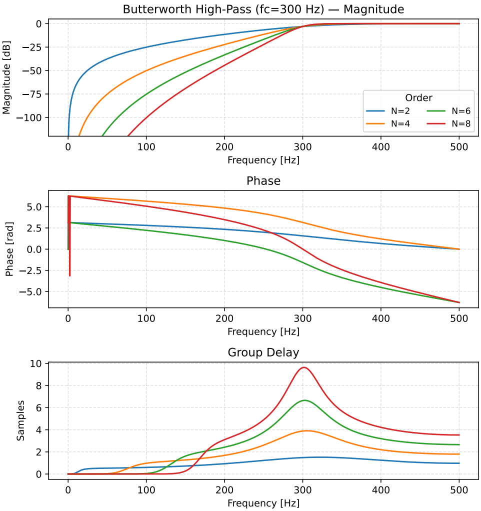
Key observations.

- *Magnitude:* higher order $\Rightarrow$ steeper transition (monotonic, no ripples).
- *Phase:* nonlinear (curved) near the cutoff; stronger with higher order.
- *Group delay:* frequency-dependent; peaks around the transition band.

IIR LP Filters: Magnitude, Phase, and Group Delay



IIR HP Filters: Magnitude, Phase, and Group Delay



IIR Band-Pass (Digital) Filters

Purpose. An IIR band-pass filter passes a range of frequencies $[f_1, f_2]$ and attenuates all others. It is obtained by transforming a low-pass prototype to a band-pass form.
General form.

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

with zeros near $z = \pm 1$ (to reject DC and π) and poles clustered around the center frequency.
Key parameters.

$$f_0 = \sqrt{f_1 f_2}, \qquad B = f_2 - f_1, \qquad Q = \frac{f_0}{B}$$

- f_0; center (resonant) frequency.
- B; bandwidth, controls width of passband.
- Q; quality factor, narrow vs. wide band.

Characteristics.

- Derived from low-pass prototypes (Butterworth, Chebyshev, Elliptic).
- Recursive $\Rightarrow$ efficient but *nonlinear phase*.
- Implemented as cascaded biquads for stability.
- Useful for resonance extraction, tone isolation, communications bands.

Frequency response: smooth passband around f_0, steeper skirts with higher order N.

IIR Frequency Transformations

Goal: obtain new filters from a low-pass prototype $H_{LP}(z)$ by *frequency transformations* that remap the unit circle.

$$H_{LP}(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}$$

1. Low-Pass $\Rightarrow$ High-Pass Transformation

$$z^{-1} \;\rightarrow\; -z^{-1}$$

or, more precisely,

$$A_{HP}(z^{-1}) = -z^{-1}$$

This maps low frequencies ($\Omega \approx 0$) to high frequencies ($\Omega \approx \pi$).

- Inverts the frequency axis: $\Omega' = \pi - \Omega$.
- Zeros at $z = 1$ move to $z = -1$ (reject DC, pass high frequencies).

2. Low-Pass ⇒ Band-Pass Transformation

$$z^{-1} \;\rightarrow\; A_{BP}(z^{-1}) = \frac{z^{-2} - 2\cos\Omega_0\, z^{-1} + 1}{1 - 2r\cos\Omega_0\, z^{-1} + r^2 z^{-2}}$$

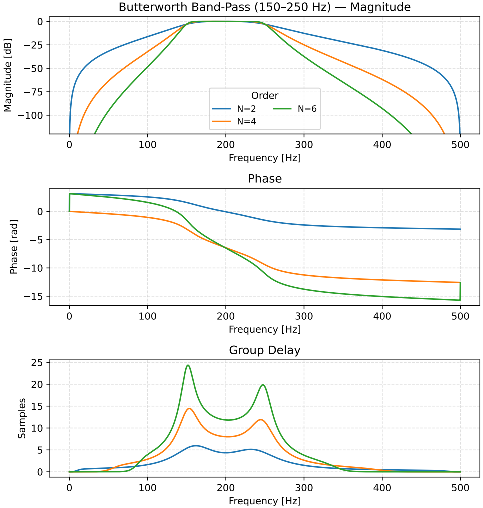
where

$$\Omega_0 = 2\pi\frac{f_0}{f_s}, \quad r = 1 - \pi\frac{B}{f_s}, \quad B = f_2 - f_1, \quad f_0 = \sqrt{f_1 f_2}$$

- Ω_0: center (resonant) frequency.
- r: determines bandwidth ($r \to 1 \to$ narrow band).
- Substituting A_{BP} into $H_{LP}(z)$ yields $H_{BP}(z) = H_{LP}(A_{BP}(z^{-1}))$.

Note. Both mappings preserve stability: poles of $H_{LP}(z)$ inside the unit circle remain inside after transformation.

IIR BP Filters: Magnitude, Phase, and Group Delay



FIR and IIR Notch (Band-Stop) Filters

Goal: Attenuate one frequency f_0 (or narrow band) while passing others.

1. FIR Notch Filter (zeros on unit circle):

$$H_{FIR}(z) = 1 - 2\cos(\Omega_0)\, z^{-1} + z^{-2}, \qquad \Omega_0 = 2\pi \frac{f_0}{f_s}$$

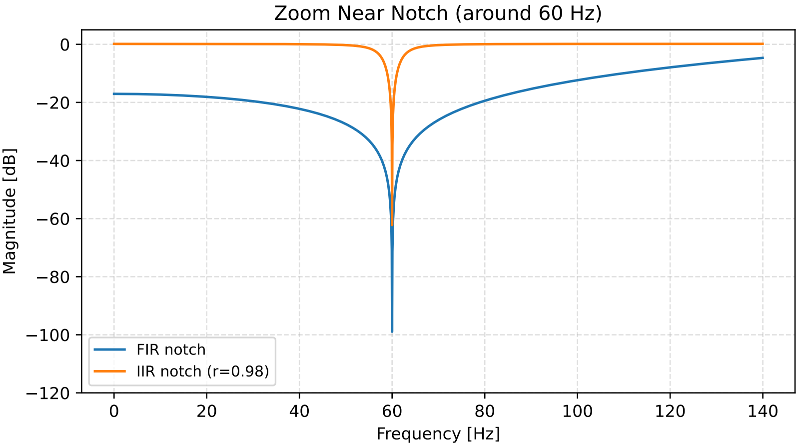
- Zeros at $z = e^{\pm j\Omega_0}$ remove f_0.
- Linear phase, always stable.
- For narrower notch, we increase order.

2. IIR Notch Filter (with poles near zeros):

$$H_{IIR}(z) = \frac{1 - 2\cos(\Omega_0)\, z^{-1} + z^{-2}}{1 - 2r\cos(\Omega_0)\, z^{-1} + r^2 z^{-2}}, \quad 0 < r < 1$$

- Poles at $re^{\pm j\Omega_0}$ sharpen the notch.
- Bandwidth $\approx \dfrac{(1-r)f_s}{\pi}$.
- Narrower notch when $r \to 1$ (but slower transient).

FIR vs IIR Notch (Band-Stop) at $f_0 = 60$ Hz



Example: $f_s = 1000$ Hz, $f_0 = 60$ Hz, IIR radius $r = 0.98$.

Thank you

- **Any Questions?**
- **Office Hours**:
  - **Tue & Thu** *(09:00-11:00)*
  - 24/7 by email (`costashatz@upatras.gr`, subject: *ECE_SP_AM*)
- **Material and Announcements**

Laboratory of Automation & Robotics