Digital Signal Processing

Traitement numérique du signal

Digitale Signalverarbeitung

Procesamiento digital de señales

1

Israel Cayetano, ing. microtechn. dipl. EPF, MSc.

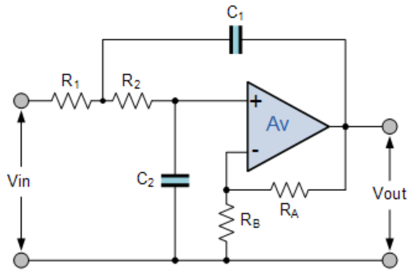
Some basic concepts

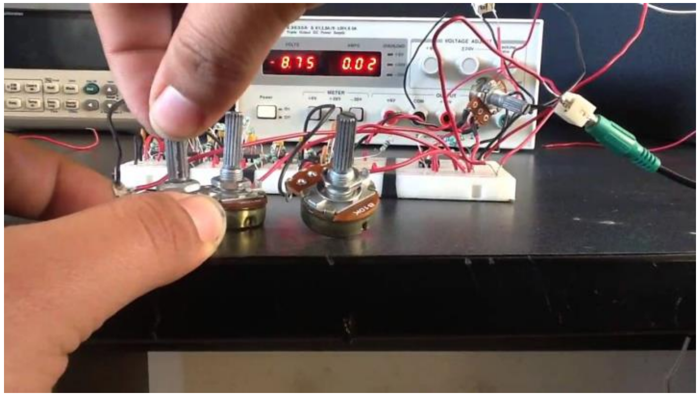
- **Signal**
  - Measurement of a phenomenon that evolves over time
  - Examples?
- **Digital signal**
  - Sequence of regular measurements of a phenomenon → **VECTOR**
  - "Double discrete signals"
- **Processing**
  - Apply operations (functions, algorithms) to a signal
    - Analysis
    - Synthesis

System

Time series

Why Digital Signal Processing (DSP)?



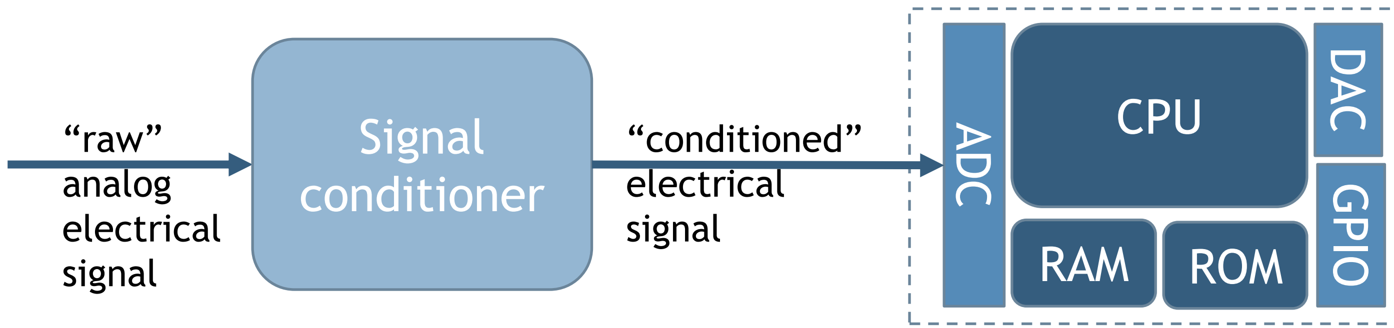


Code

Flexibility

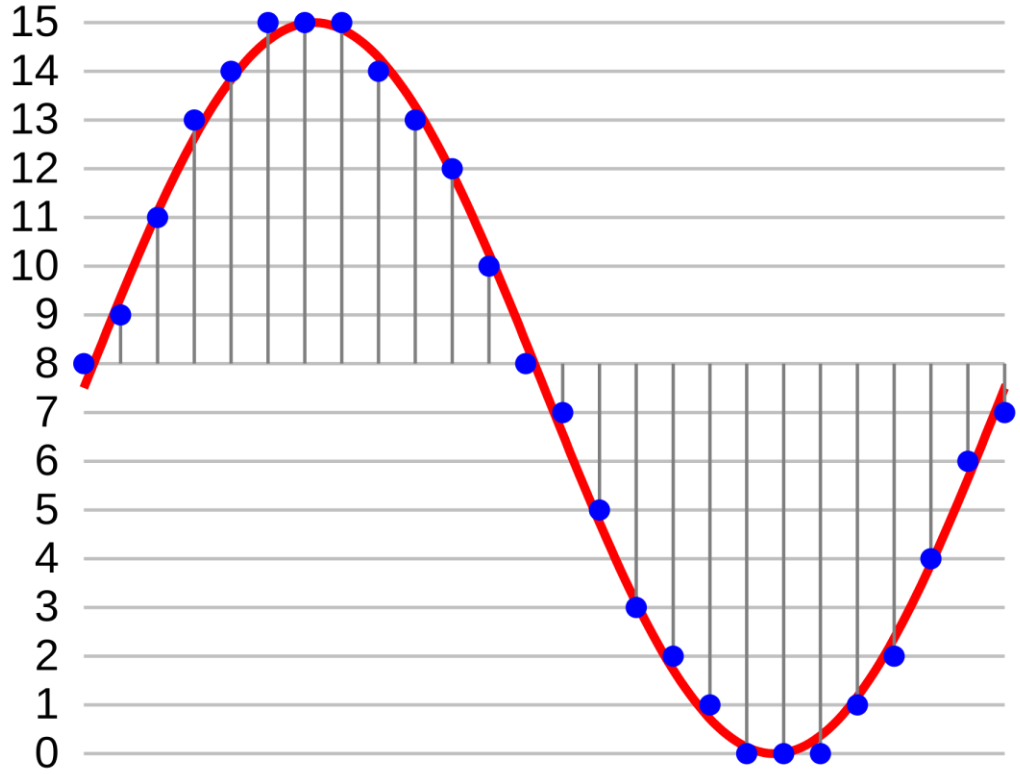
Some caveats...

- Both analog and digital signal processing interact
  - Example: Signal conditioning



- Not everything in analog can be implemented in digital
- Not everything in digital can be implemented in analog
- **But a lot more can be done in digital!!!**

How to convert an analog signal into a digital one?



Stem plot

Die Digitalisierung
La numérisation

- Two step process:
  1. Sampling
  2. Quantization

A digital signal is a vector

Example:
CD Quality Audio
fs
Bits

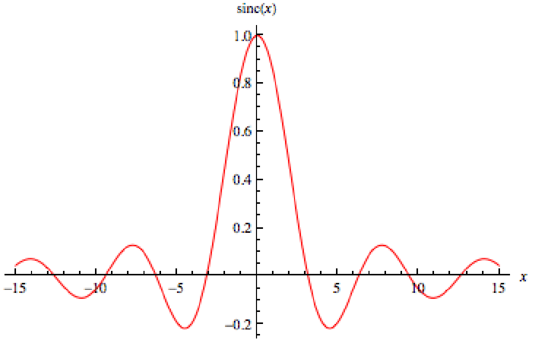
Information loss

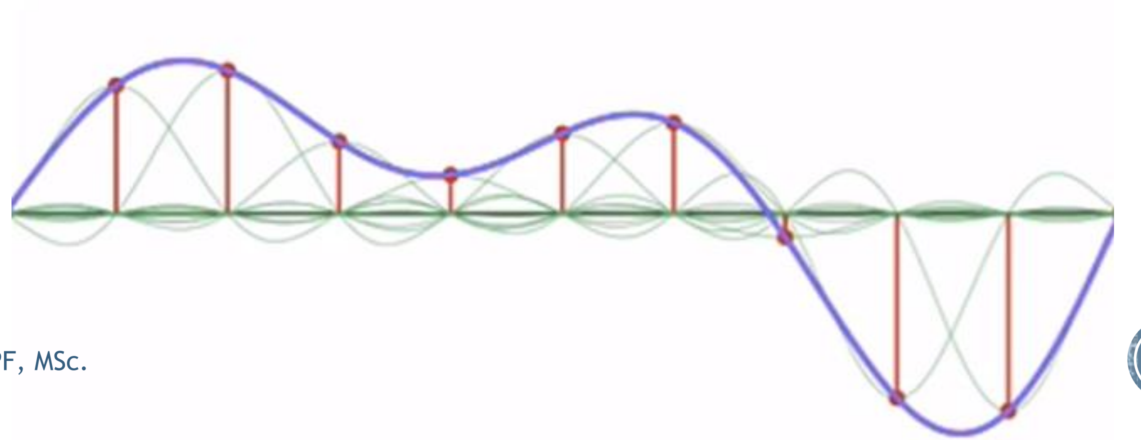
Less storage
Less processing power

Sampling

Die Abtastung
L'échantillonnage

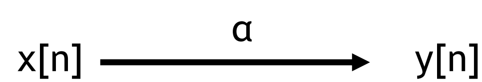
- Sampling Theorem
  - Harry Nyquist and Claude Shannon
  - From continuous to discrete
    - Sample with at least twice the bandwidth of the signal
      - Avoid **aliasing**
  - From discrete to continuous
    - A continuous signal can be recovered from discrete samples as a linear combination of the *sinc* function shifted and scaled by the values of the discrete signal



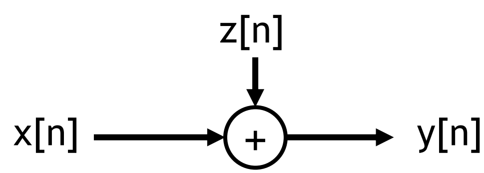


Basic notation, operators and properties

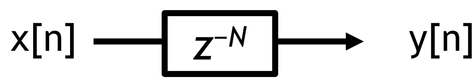
- Sequence: $x[n]$ where $n \in \boldsymbol{Z}$
- Scaling: $y[n] = \alpha x[n]$



- Sum: $y[n] = x[n] + z[n]$



- Shift (delay) by N samples : $y[n] = x[n - N]$

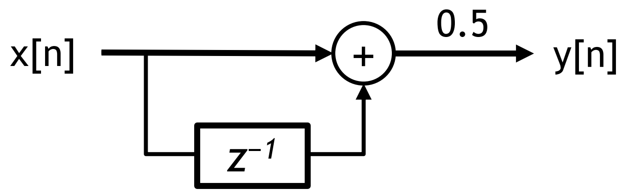


If online processing:
Causal system

LTI System → Linear Time Invariant
BIBO stability → Bounded Input Bounded Output

Exercise (1)

- Transform from equation to blocks or viceversa and show the plot of the effect of applying them $x[n] = \delta[\mathrm{n}]$ and $x[n] = \mathrm{u}[\mathrm{n}]$



$$y[n] = 1.06y[n-1] + x[n]$$

Moving Average

Exercise (2)

- Get a "short" equation and draw the blocks of an average that considers both the current and all of the previous datapoints.

Cumulative Moving Average

Exponential Smoothing

- "Weighted Cummulative Moving Average"
- Also known as "Leaky Integrator"

$$y[n] = \lambda x[n] + (1 - \lambda) y[n-1]$$
$$\text{where } 0 < \lambda < 1$$

Filtering

- Remove modifying and interfering signals while preserving the desired signal
- Modulate amplitude, frequencies and phase
- "Traditional Filters"
  - Types: LPF, HPF, BPF, BSF, Notch Filter, etc.
- Characteristics
  - Typology
  - Order
  - Cut-off frequency(ies)
  - Sampling frequency

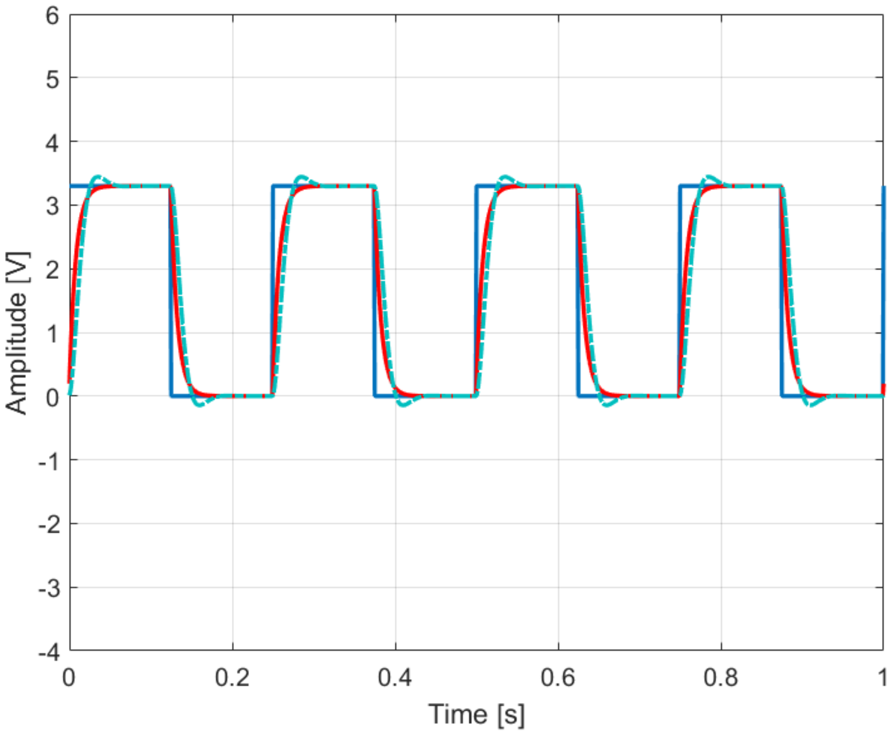
Normalized frequency

MATLAB

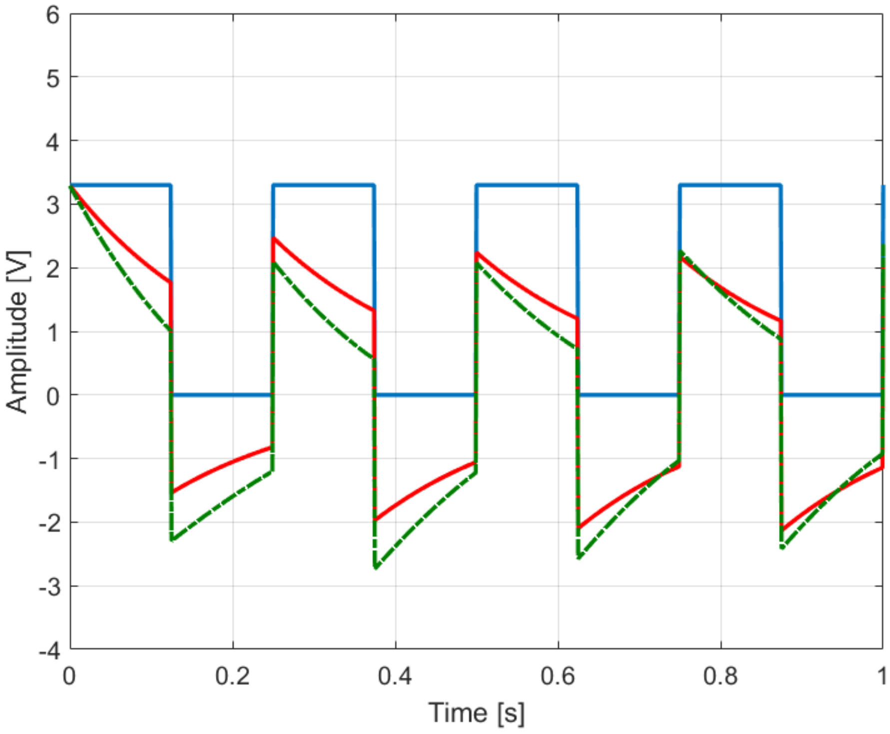
$$f_d = \frac{f}{f_s/2}$$

Exercise (3)

- What are the parameters of the following signal?
- Qualitatively, plot the effect of applying a “soft” LPF and HPF to the following signal



LPF



HPF

Time domain

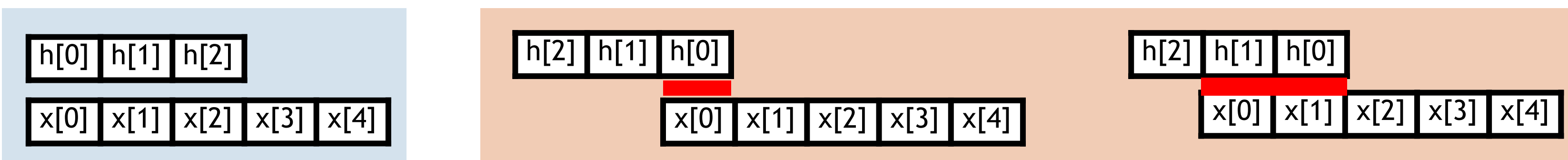
- **Difference equations**
- Impulse response h[n]:
  - Output of a filter when the input is an impulse δ[n]
  - Can fully describe the filter
  - A filter is applied to signal by convolving the signal with the impulse response of the filter
  - Infinite and Finite Impulse Response
- Convolution

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] * h[n]$$

Exercise (4)

- Apply the Moving Average Filter with a window of two samples to u[n] using convolution.

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = x[n] * h[n]$$



Frequency domain: Fourier Transform

- Jean Baptiste Joseph Fourier
  - Théorie analytique de la chaleur
  - A signal can be expressed as a sum of sinusoids
- DFT
  - Discrete Fourier Transform of a discrete signal (finite length)
- DTFT
  - Continuous Fourier Transform of a discrete signal (infinite length)
- STFT
  - DFT over a period of time (time window)
  - Spectrogram
- FFT
  - "Most important algorithm of all times" (Prof. Gilbert Strang)
  - Computationally efficient algorithm of the DFT → Gauss, Cooley-Tuckey
  - PSD (Power Spectral Density)

Frequency domain: z-transform

- Frequency domain representation of a discrete signal
- Form to solve difference equations

$$X[z] = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \mathrm{z} = e^{j\omega}$$

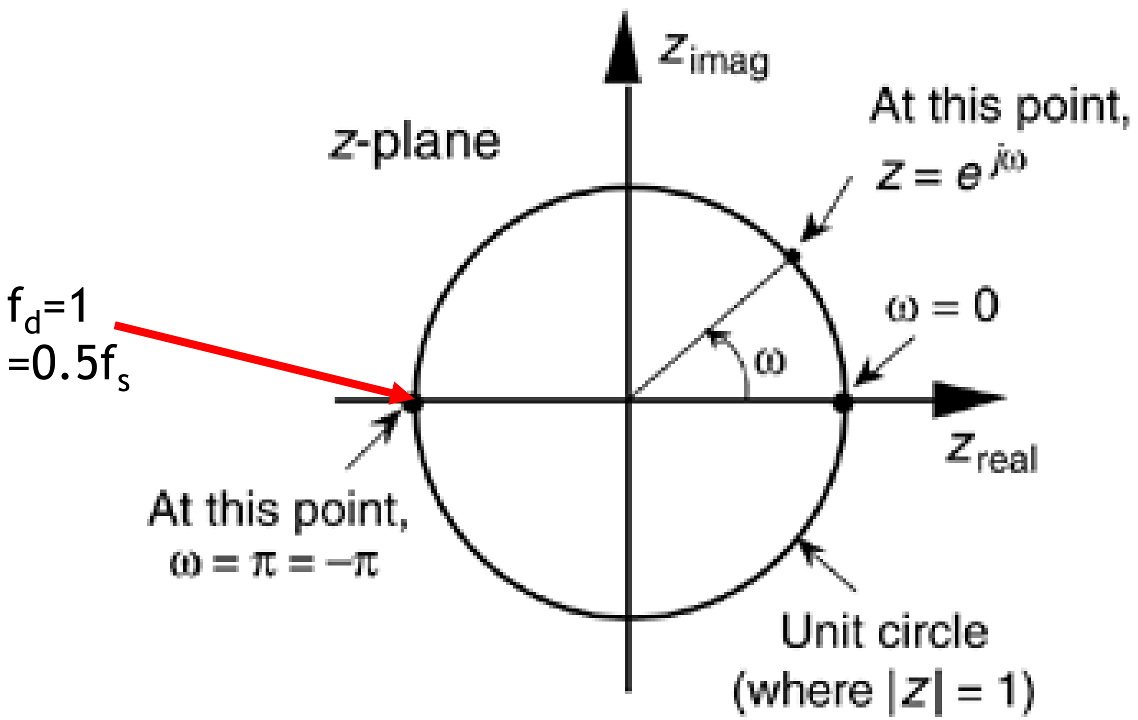
- Transfer function of a filter:

$$H[z] = \frac{Y[z]}{X[z]} = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{\sum_{k=0}^{N-1} a_k z^{-k}}$$

- A filter is stable if its poles are inside the unit circle
- In frequency domain, convolution is replaced by pointwise multiplication

z-plane

- For a digital filter, we only care about how “up” or “down” is the border of the unit circle at each frequency



Exercise (5)

- Transform the difference equations from exercise 1 into their z-transform Transfer Function equivalents. Plot their poles and zeros in the z-plane and obtain the frequency response plot (just the gain)

Exercise (6)

- Generate the difference equation, z-transform, z-plane and frequency response of a system that:

a) Obtains the "derivative" of the input signal with a time step T
b) Integrates the input signal (using rectangles and using trapezoids)

Exercise (7)

- Qualitatively, how do you relate the s-plane with the z-plane?

Is it possible to have oscillatory behavior with a digital 1^{st} order system?

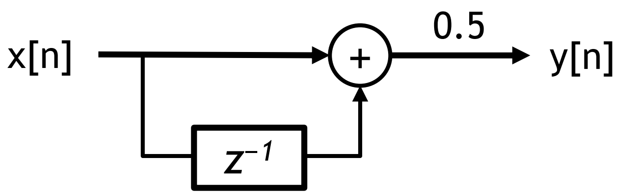
Exercise (8)

- Using the z-transform, find a general formula to solve:

$$y[n] = ay[n-1] + x[n] \quad \text{where} \quad x[n] = \delta[\mathrm{n}] \ \text{ and } \ x[n] = r\,\mathrm{u}[\mathrm{n}]$$

Exercise (9)

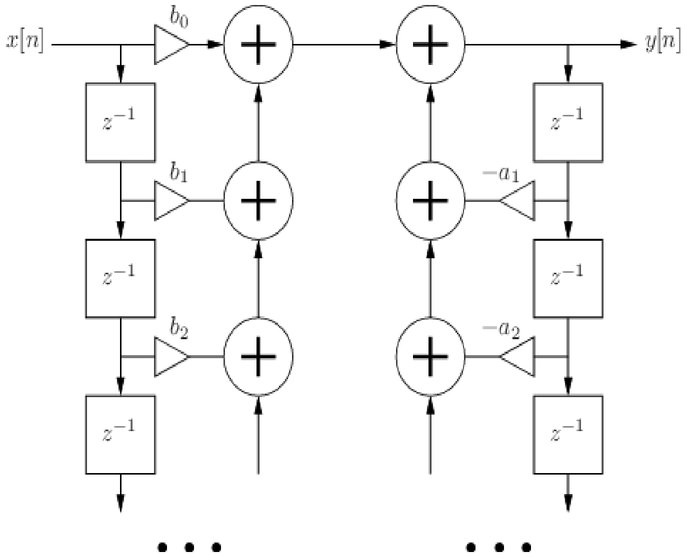
- Using the z-transform, solve:



where $x[n] = \mathrm{u[n]}$

IIR Filters

- Infinite impulse response but with finite length(short) implementation
- Recursive
  - Consider cumulative effect with feedback terms
  - Order given by the maximum delay of the feedback terms
- Have poles and zeros
- Can be implemented in analog filters
- Non-linear phase
- Can or cannot be BIBO stable



Israel Cayetano, ing. microtechn. dipl. EPF, MSc.

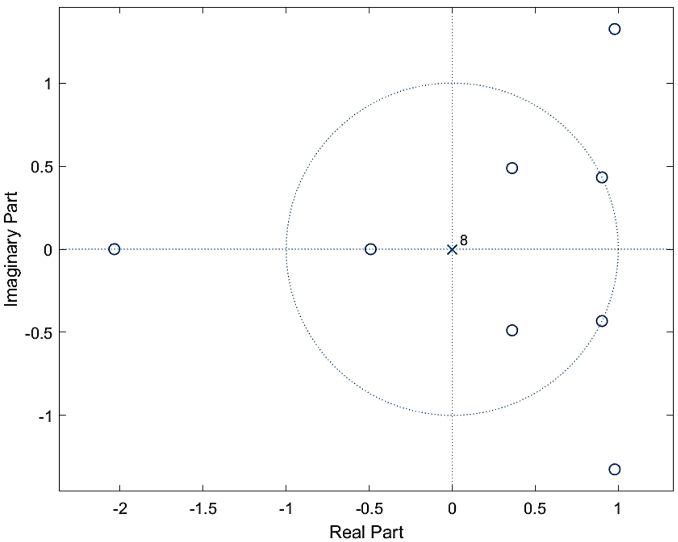
FIR Filters

- Non recursive
  - "Only" zeros → only feedforward terms
- Cannot be implemented as analog filters, only as digital
- **Linear phase**
- Always BIBO
- Require more terms to have a behavior similar to that of an IIR filter

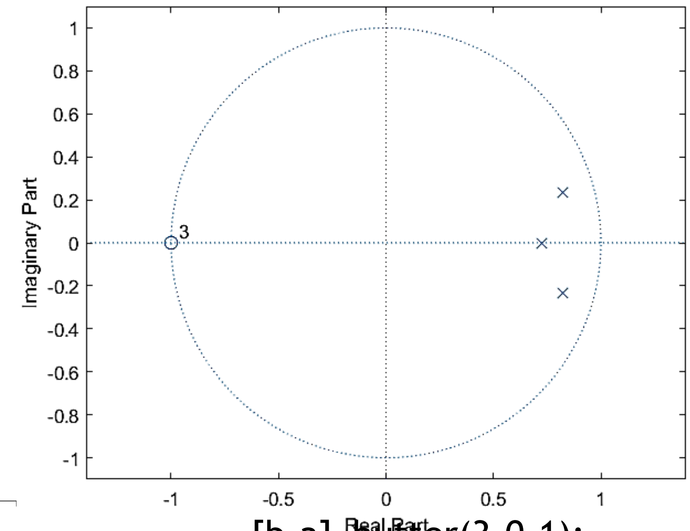
Taps

Exercise (10)

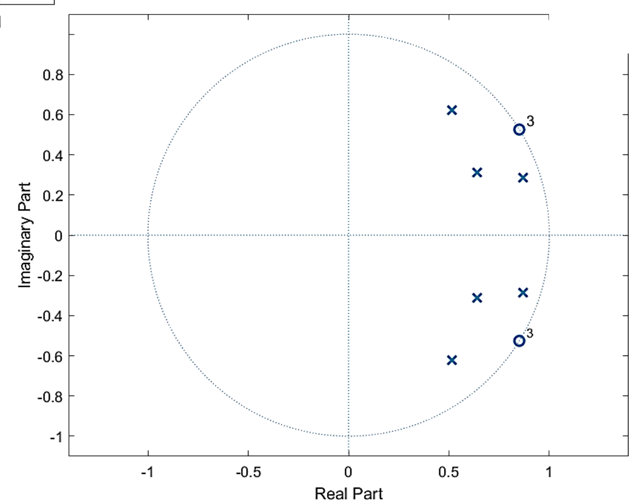
- Identify if the filter is IIR or FIR, its type (LPF, HPF, etc.) and order



b=fir1(8,[0.7 0.8],'bandpass');



[b,a]=butter(3,0.1);



[b,a]=butter(3,[0.1 0.3],'stop');

Autoregressive (AR) models

- Main assumption: The new output of a signal can be thought as a linear combination of its past outputs plus some white noise.

$$y[n] = \sum_{i=1}^{p} a_i y[n-i] + \varepsilon[n]$$

- All pole filter
- In time domain: used to make predictions when the new actual datapoint of the time series from which the model was obtained is still not available.
- In frequency domain: *"Filter identification"* → filter a signal to have the same frequency content as the one from which the model was obtained.
- AR(p)→ p is the number of poles to fit the filter
- Coefficients a_i can be obtained using the Yule-Walker equations.

Additional important concepts

- Forward-Backward Filtering
- Signal-to-Noise Ratio (SNR)
- Signal compression
- ARMA/ARIMA model
- Adaptive filters
- Digital Image Processing
- **Wavelets**
- Singular Value Decomposition

Mexican Hat

Useful MATLAB functions

- impz(b,a) → Impulse response
- freqz(b,a) → Frequency response (Bode plot)
- zplane(b,a)
- butter(n,fd,type) → Obtain coefficients for a Butterworth filter
- pwelch(signal,[],[],[],fs) → Frequency content of a signal (FFT)

Default values

MATLAB

$$f_d = \frac{f}{f_s/2}$$

$$a_0 y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] \ldots - a_1 y[n-1] - a_2 y[n-2] \ldots, \qquad n = 0, 1, 2, \ldots$$

References

- Vetterli, M., Prandoni, P. (2017). *Digital Signal Processing*. EPFL
- Vesin, J. (2016). *Biomedical Signal Processing Course Notes*. EPFL

- Siemens (2018). *Introduction to Filters*. Information available on the URL <https://community.plm.automation.siemens.com/t5/Testing-Knowledge-Base/Introduction-to-Filters-FIR-versus-IIR/ta-p/520959>