Digitálne spracovanie zvuku,obrazu a biosignálov.

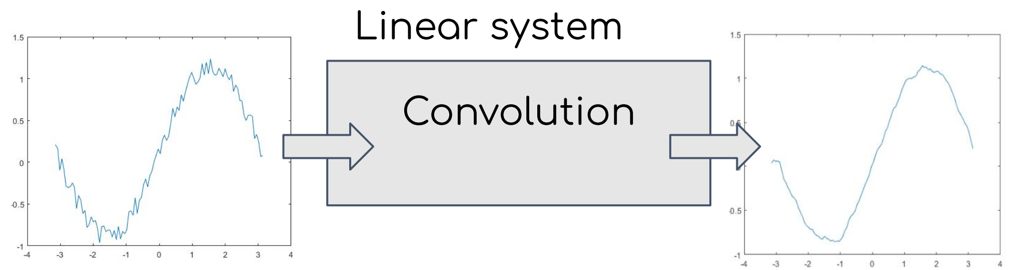
Signal filtration

Signal filtration

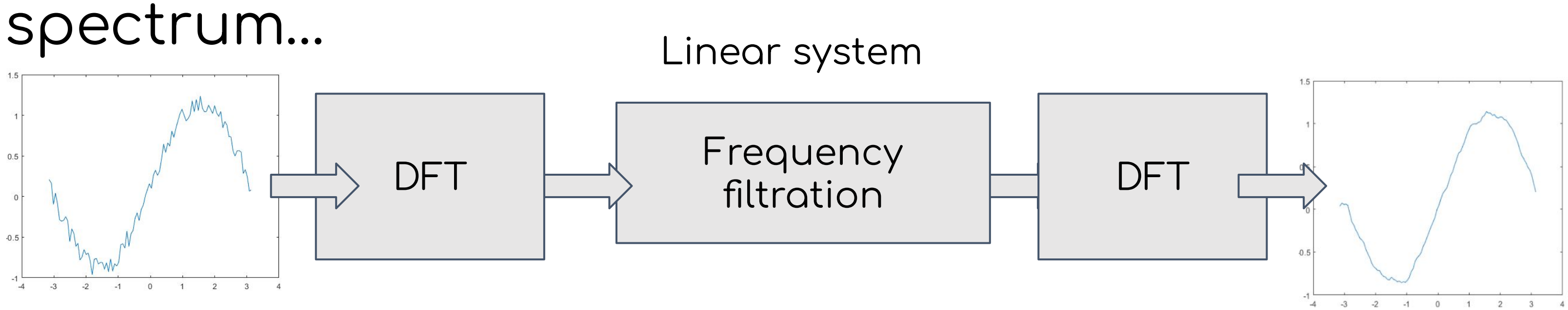
Linear signal filtration

1. in time domain
   -> direct modifying of the signal samples
   - operation convolution

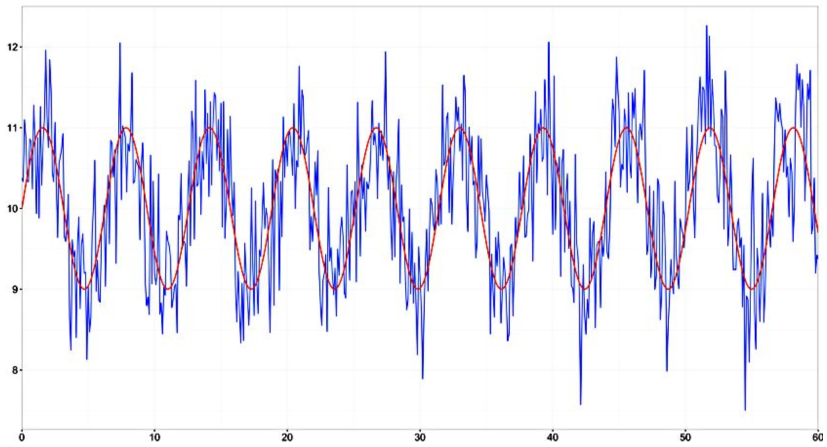
   Example:



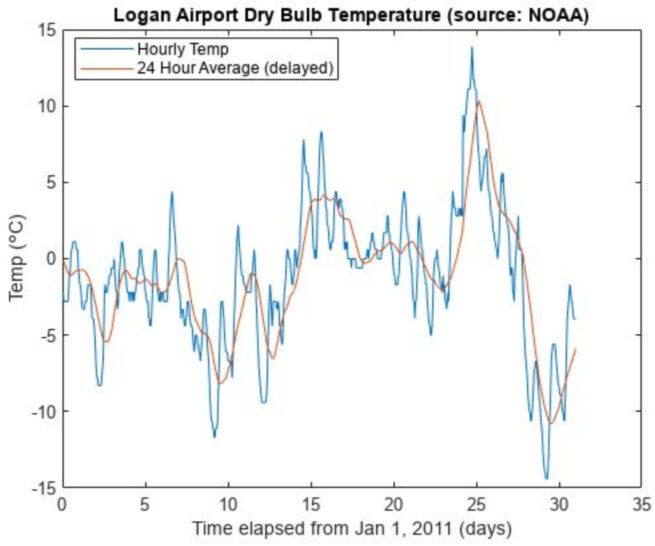
2. in frequency domain
   -> modifying the frequency spectrum
   - using DFT Fourier spectrum, DCT cosine transform spectrum...



Filtration examples
Signal smoothing - denoising

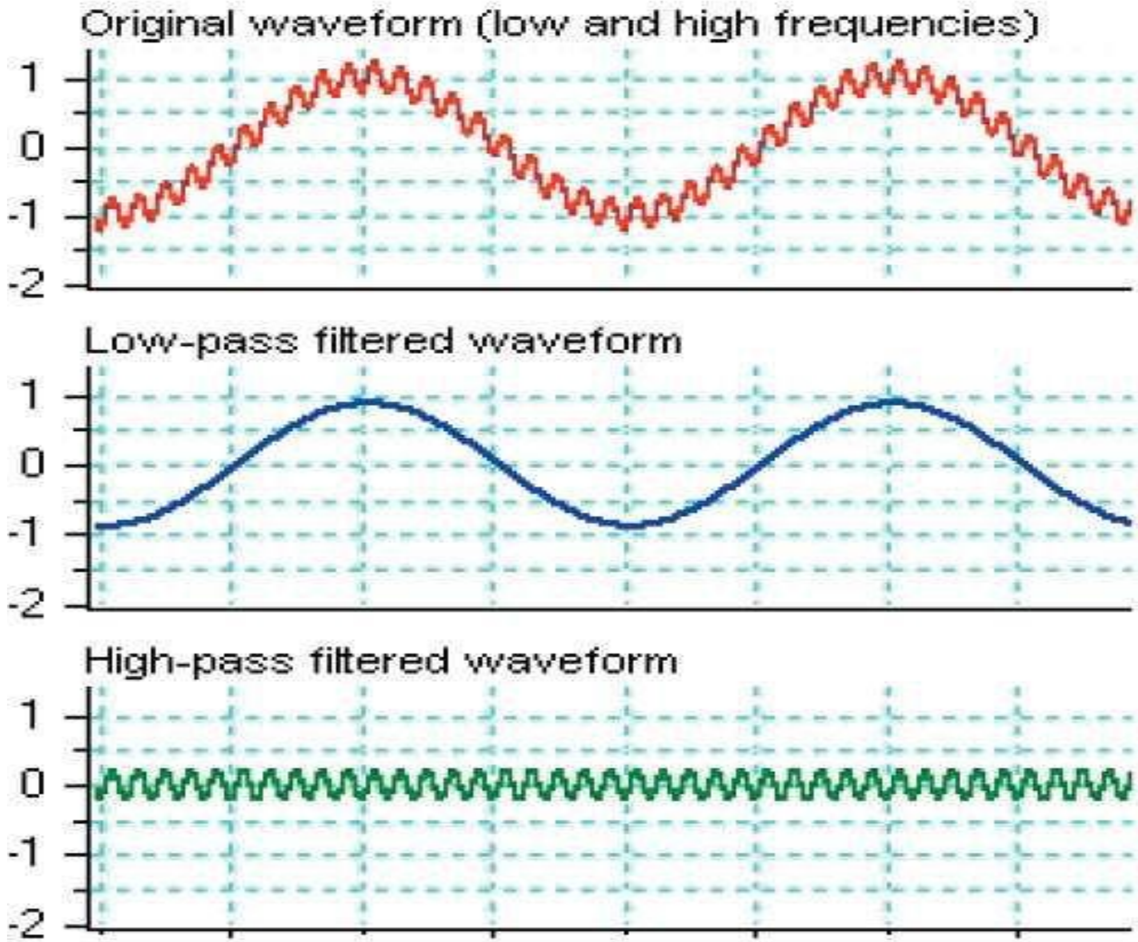


Blue: original signal
Red: filtered signal after denoising

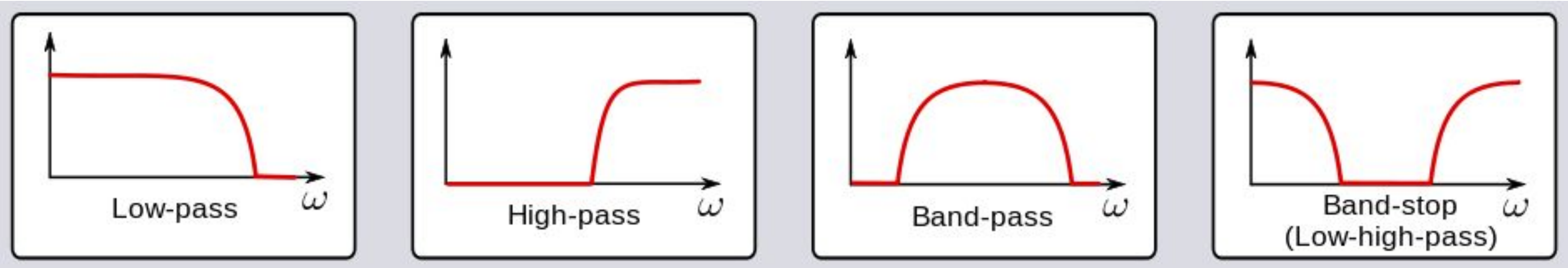


Signal filtration in spectral domain

Frequency Filter -example



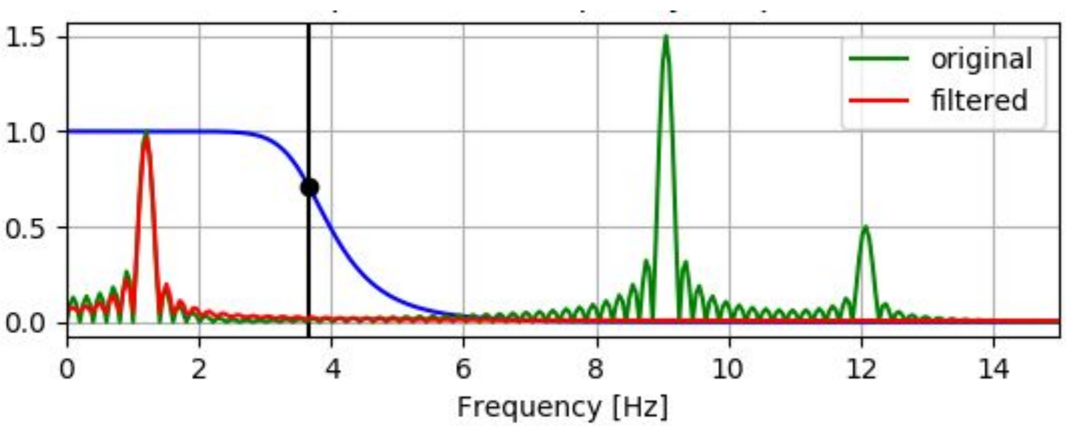
Basic types of frequency filters: LP, HP, BP, BS

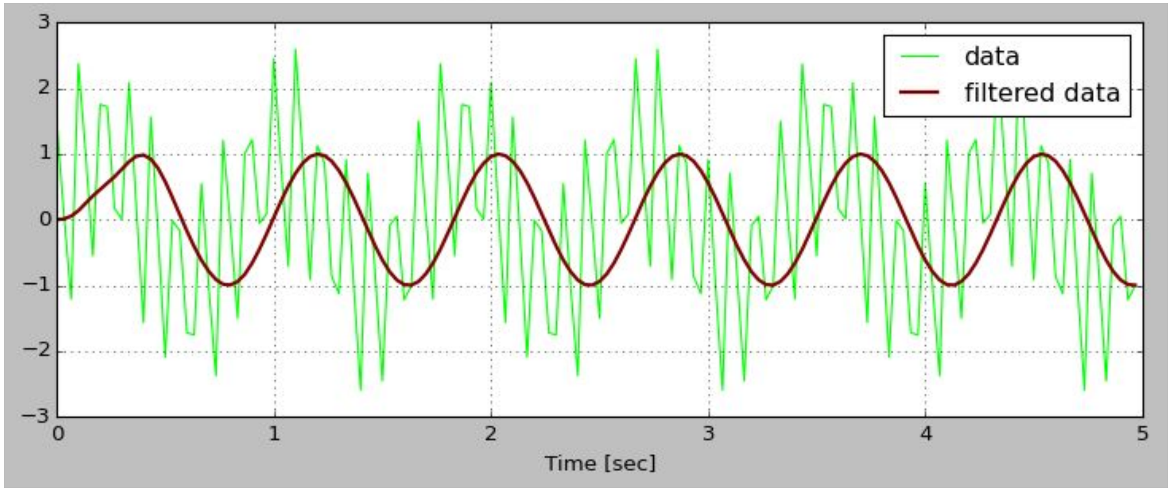


Notch filter

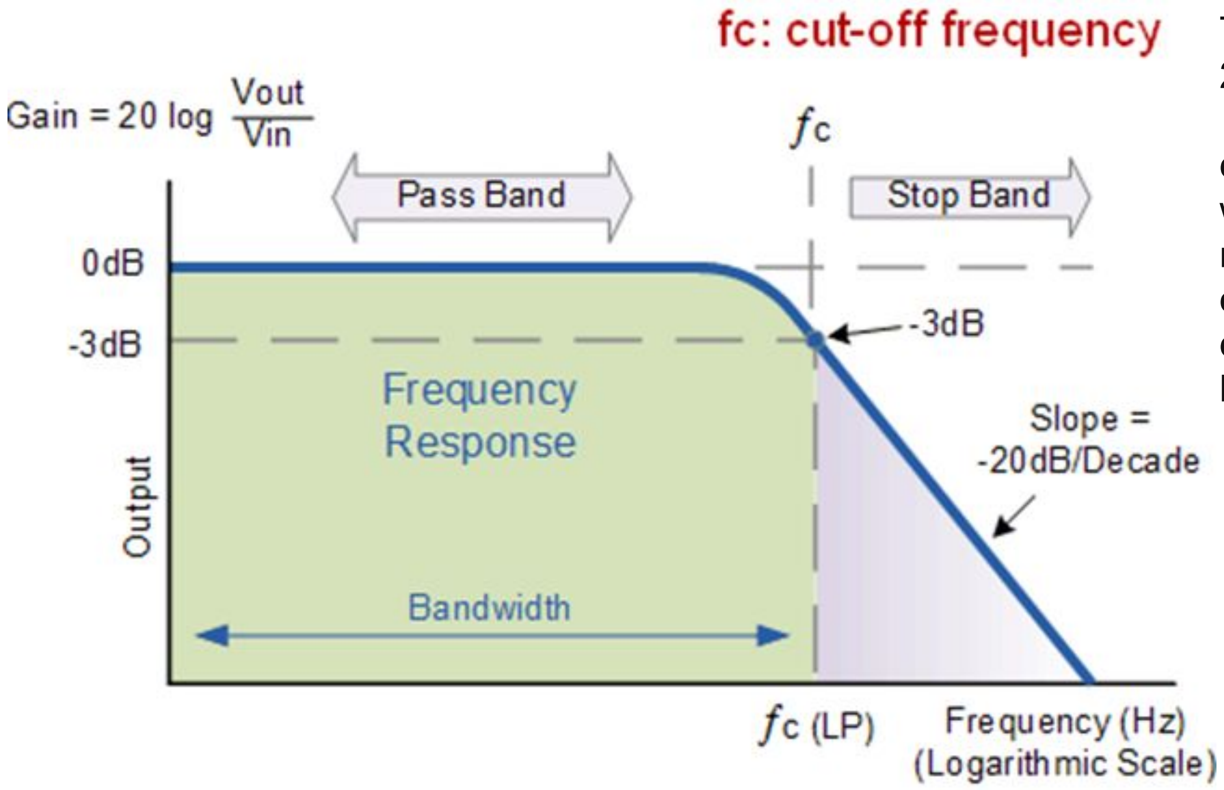
A notch filter is a type of band-stop filter, which is a filter that attenuates frequencies within a specific range while passing all other frequencies unaltered. For a notch filter, this range of frequencies is very narrow.

Low-pass filter - example





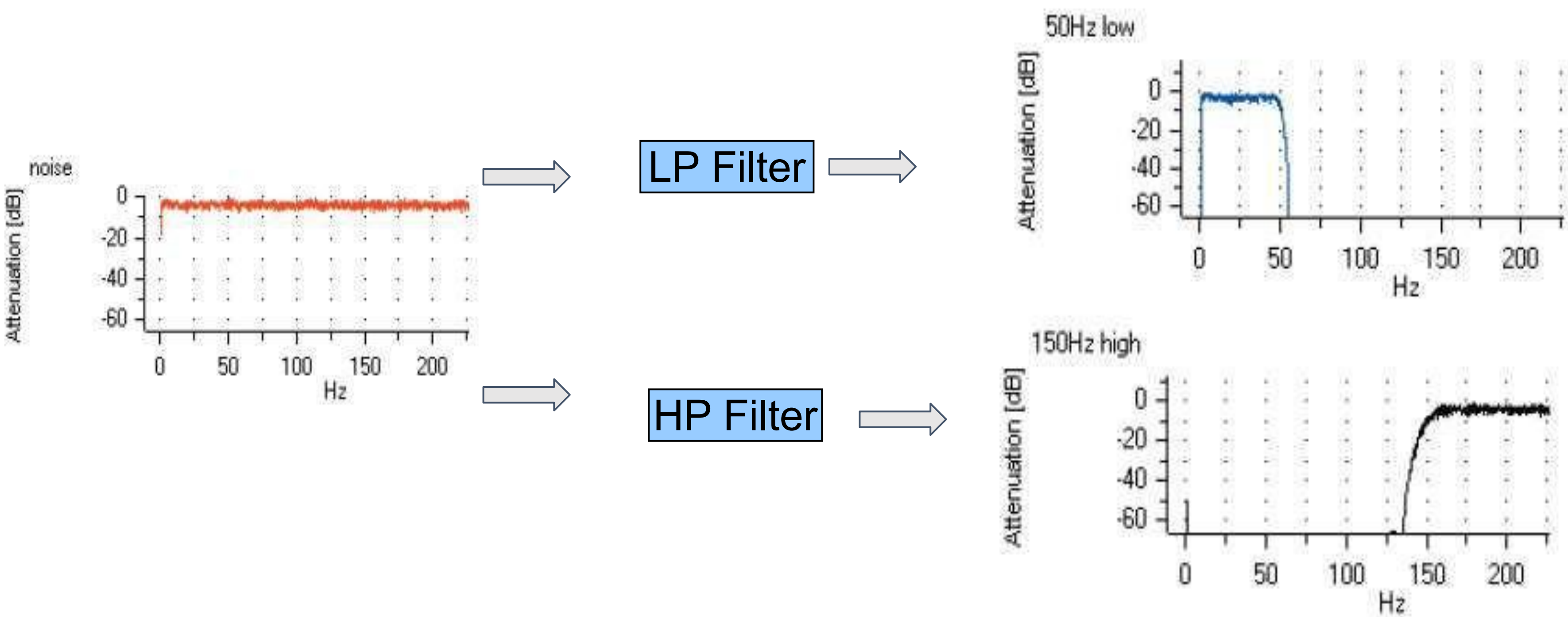
Filter parameters
Cutoff frequency



The -3dB, come from
20 Log (0.707) or 10 Log (0.5).

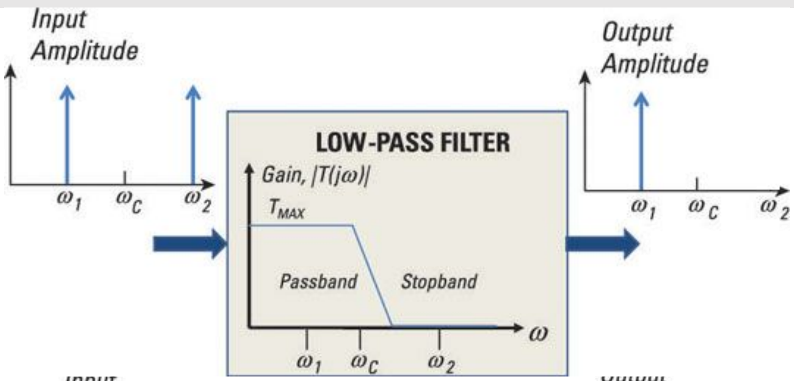
determine the bandwidth of signal, when decrease the voltage from maximum to 0.707 Max
or
decreasing the power from max to half power.

Frequency Filter

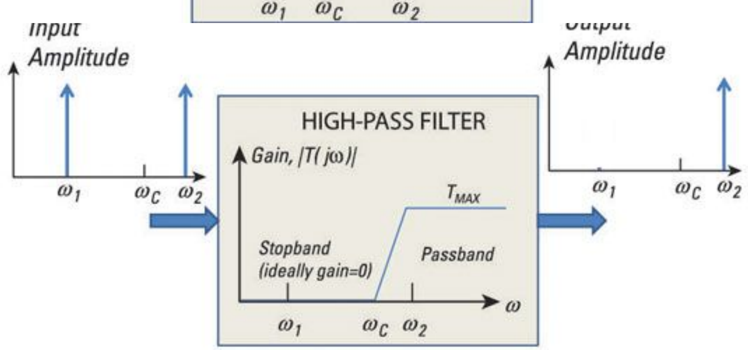


Lp, HP, BP Frequency Filter

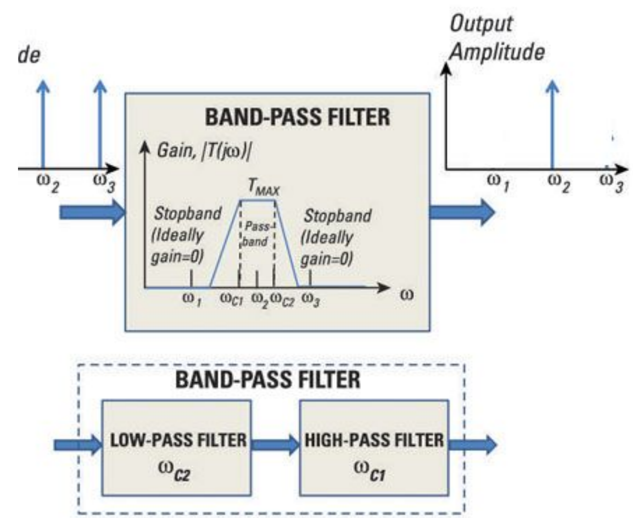
- Low-pass



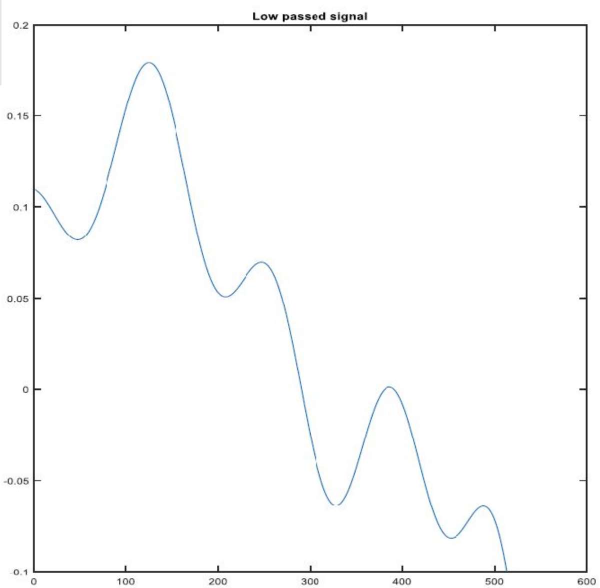
- High-pass

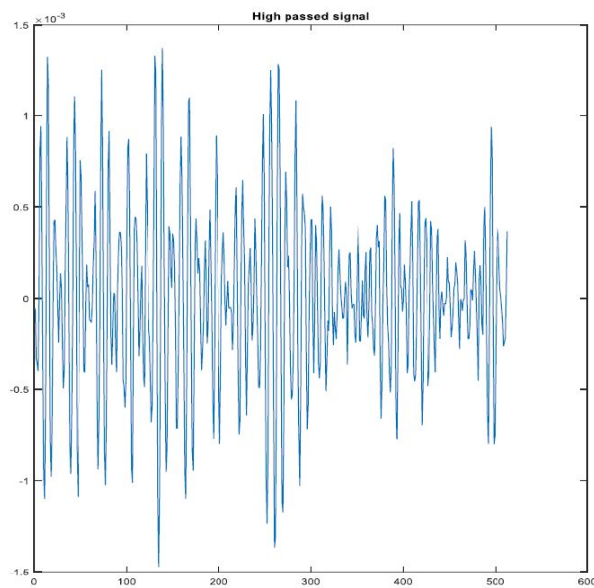


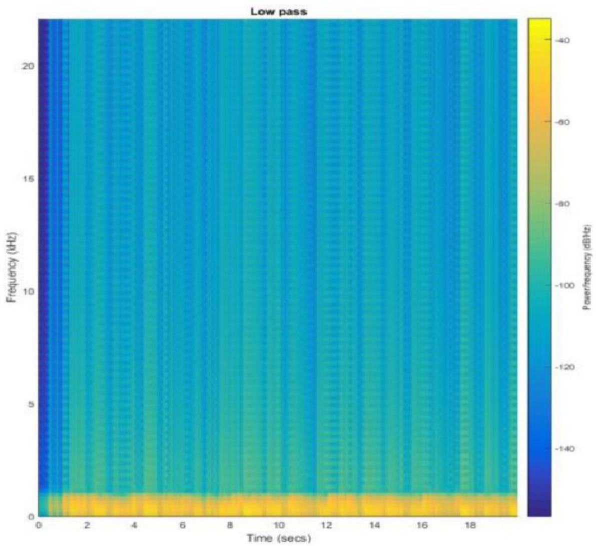
- Band-pass
- Band-stop

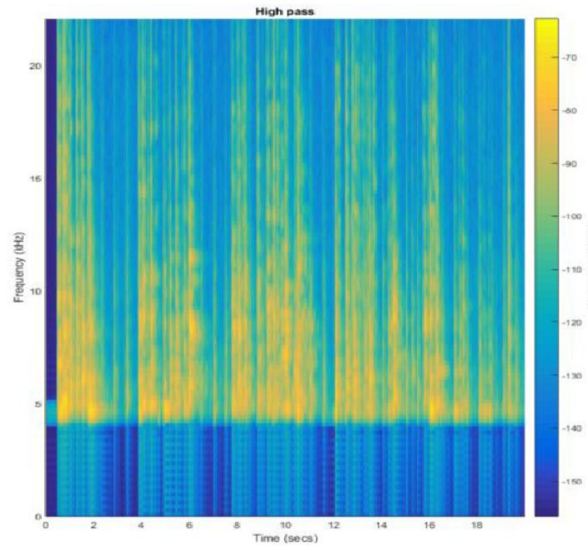


Example - LP a HP sound filtration



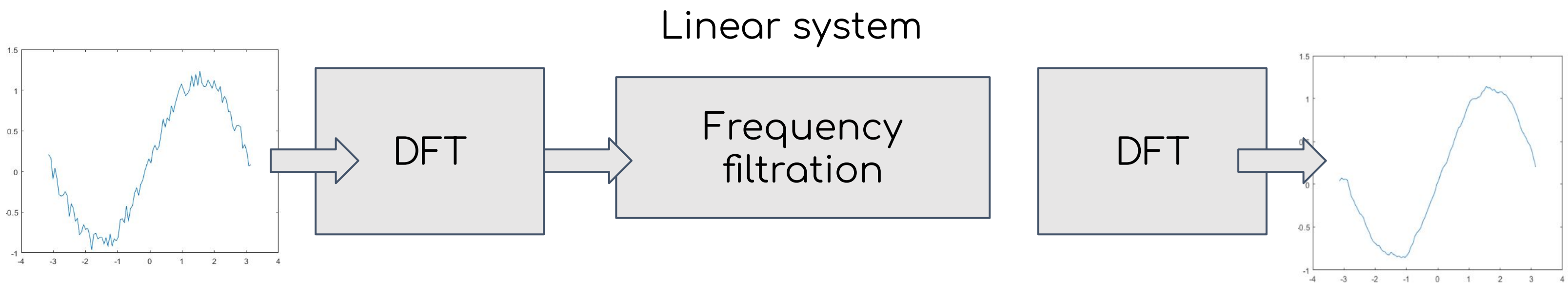






Spectrograms of
low passed signal LP
filter (left)
and
high passed signal HP
filter (right)

Filtration in the frequency domain – basic concept



Filtration in the frequency domain – basic concept

1. Transform signal g into the frequency domain :

   $F[g] = G$

2. Transform filter kernel h into the frequency domain (or direct define the filter H in the frequency domain):

   $F[h] = H$

3. Multiply (per elements):

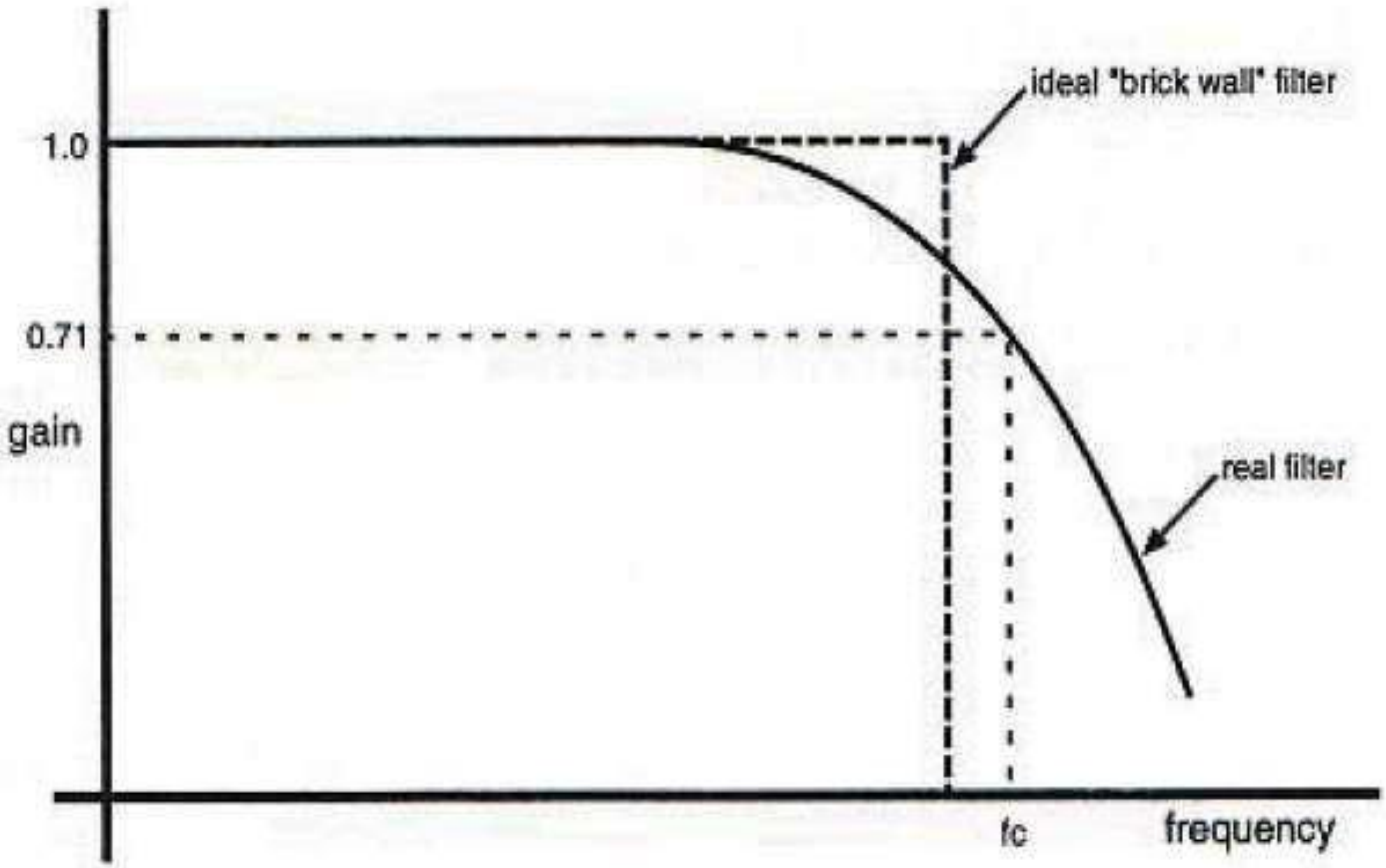
   $G.H$

4. Calculate inverse transform from frequency to time domain:

   $F^{-1}[G.H]$

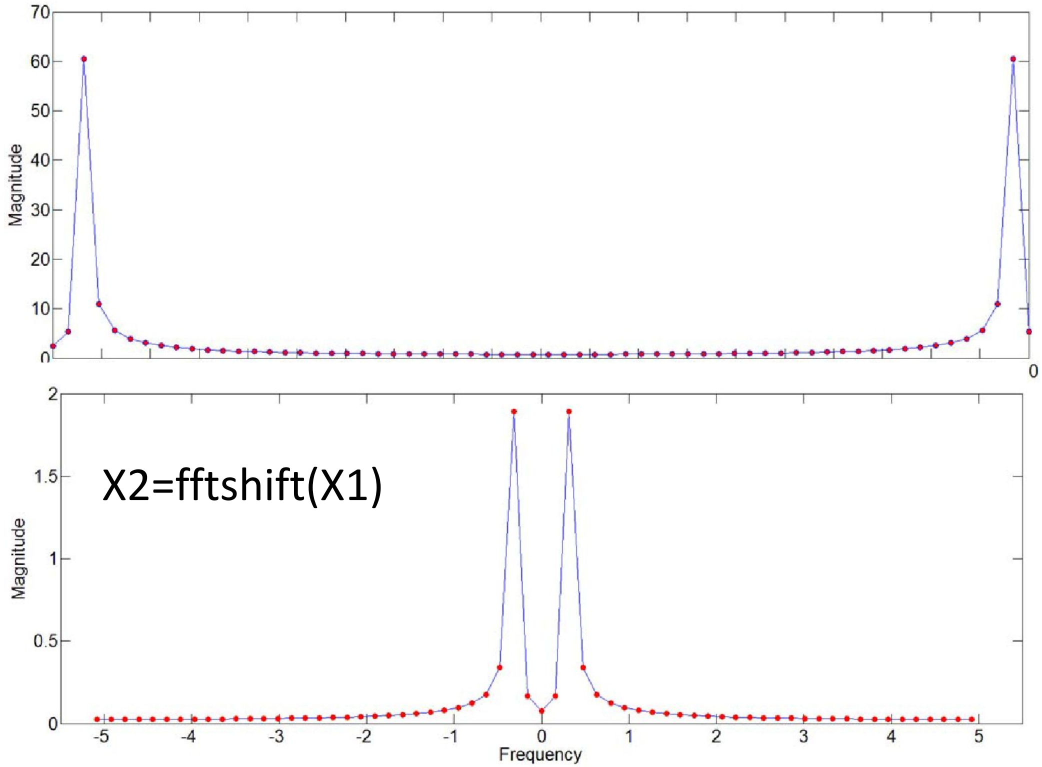
-> filtered signal in the time domain : $F^{-1}[G.H]$

Ideal filter



Rem.- Implementation of FFT (in MATLAB)

- The highest frequency that can be resolved is 1/(2Δt)
- Use *fftshift ()* to convert it into a two-sided spectrum centred around f=0



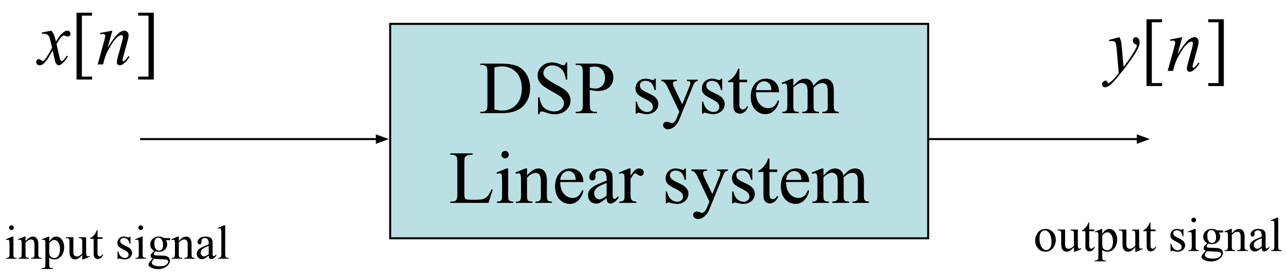
Frequency Filter -demo

Demo filter applet

http://www.falstad.com/dfilter/

Filtration in the time domain
Convolution

Digital signal processing in the time domain



Discrete 1D convolution

Definition of Discrete 1D convolution:

Given are 2 causal discrete functions:

$$f = (f_0, \ldots, f_{N-1})^T \in \mathbb{C}^N, \; h = (h_0, \ldots, h_{M-1})^T \in \mathbb{C}^N$$

The Discrete 1D convolution is vector: $g \in \mathbb{C}^{M+N-1}$

where

$$g(x) = \sum_{m=0}^{N-1} f(m)h(x-m),$$

pre $x = 0, \ldots, M+N-2.$

Similarity
Convolution <-> Correlation

Correlation:

$$r(x) = \sum_{m=-\infty}^{\infty} f(m)h(m-x) \qquad\qquad x = 0, \pm 1, \pm 2 \ldots$$

Convolution:

$$g(x) = \sum_{m=0}^{N-1} f(m)h(x-m), \qquad\qquad \text{pre } x = 0, \ldots, M+N-2.$$

Similarity
Convolution <-> Correlation

Similar definition as correlation except that the mask is flipped - no folding (time-reversal)

(in the image : both horizontally and vertically)

Note that if the convolution mask (convolution kernel) w(x,y) is symmetric,

that is w(x,y)=w(-x,-y),

-> then convolution is equivalent to correlation!

In Matlab: Conv(x,fliplr(y))

Signal filtration in time domain

Discrete 1D convolution
The running average filter -LP digital filter

Example (filter length = 3)

Filter kernel: 1/3 [1, 1, 1]:

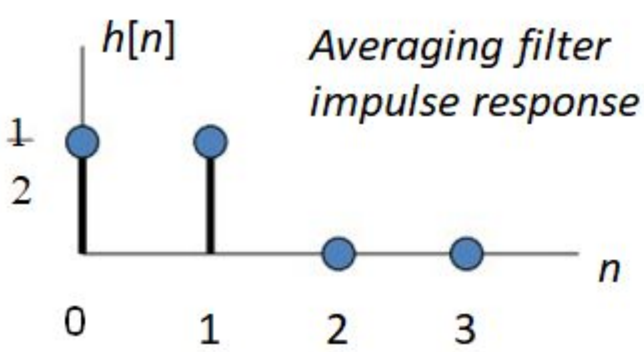
$$y[n] = \sum_{k=0}^{M} x[n-k]$$

$$y[n] = \frac{1}{3}(x[n] + x[n-1] + x[n-2])$$

LP digital filter
The running average filter - example

Example of finite impulse response (FIR) filter

Finite Impulse Response

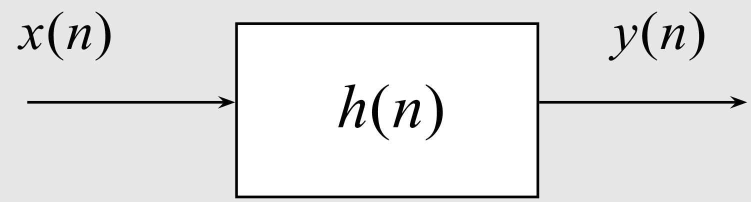


$h=[1,1]$ example avaraging filter

$y[n] = h[0]\, x[n] + h[1]\, x[n-1]$

$= (\, x[n] + x[n-1]\,) / 2$

Impulse response

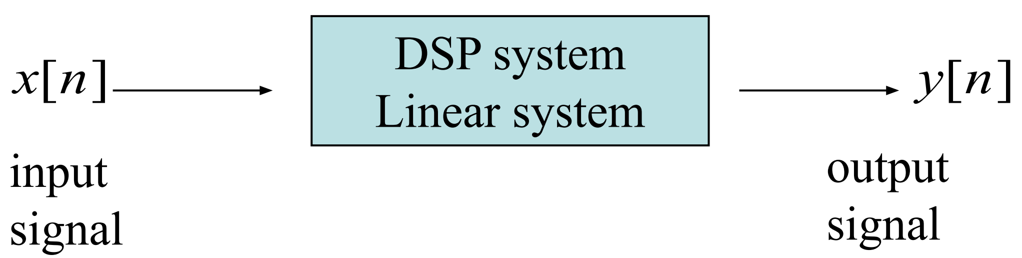


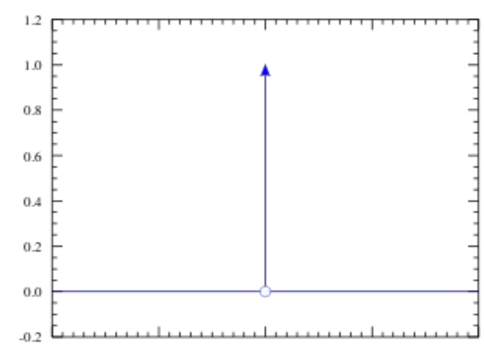
impulse response function (IRF), of a dynamic system is its output y(n) when presented with a brief input signal x(n), called an **impulse** (a short-duration time-domain signal).

for continuous-time systems, this is the Dirac delta function δ(t), while for discrete-time systems, the Kronecker delta function δ[n] is typically used.

$\delta_{ij}=0$ if $i\neq j$,

$\delta_{ij}=1$ if $i=j$





LP digital filter - Finite impulse response (FIR) filter - example

Example (filter length M = 5):

Filter kernel: [3, -1, 2, 1, 1]

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

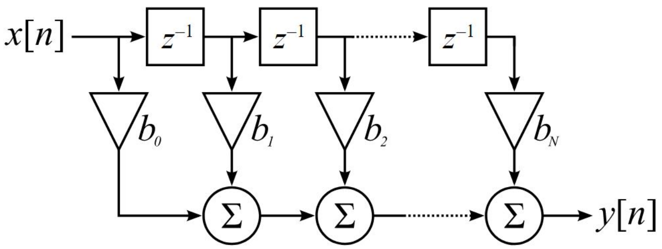
$$= 3x[n] - x[n-1] + 2x[n-2] + x[n-3] + x[n-4]$$

Finite Impulse Response (FIR) Filters

a filter whose impulse response has finite duration.

For a causal discrete time FIR filter of order N, each value of the output sequence is a weighted sum of the most recent input values:

$$y(n) = b_1x(n) + b_2x(n-1) + \ldots + b_Nx(n-N)$$



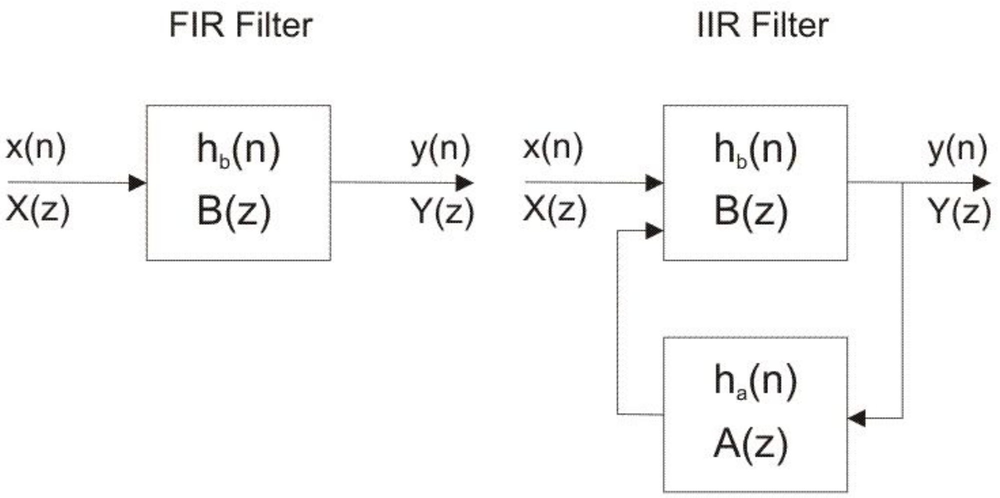
- x(n) is the input signal,
- y(n) is the output signal,
- N is the filter order
- b_i is i-th coefficient of the filter

This computation is also known as discrete convolution.

Infinite impulse response (IIR)

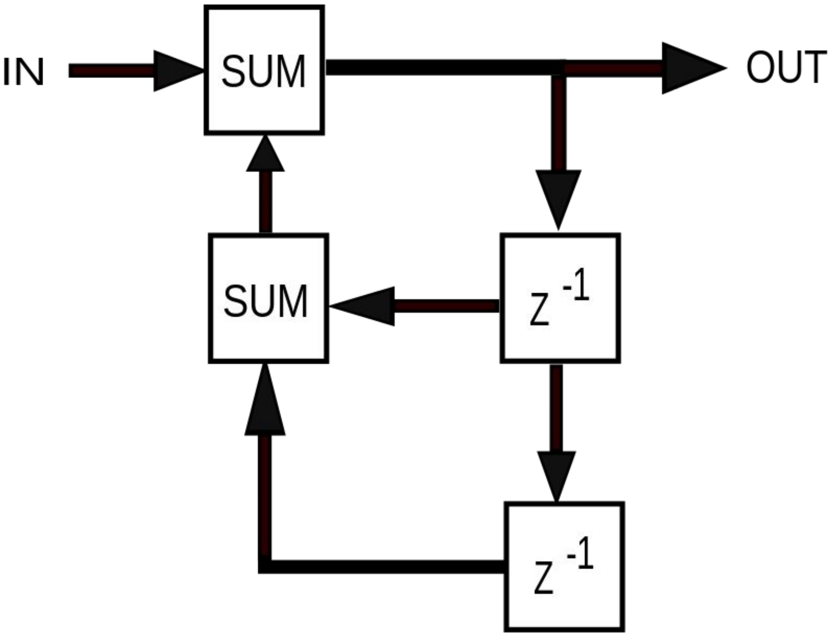
Systems with this property are known as *IIR systems* or *IIR filters*, and are distinguished by having impulse response which does not become exactly zero past a certain point, but continues indefinitely.

$$H(z) = \frac{\sum_{i=0}^{P} b_i z^{-i}}{1 + \sum_{j=1}^{Q} a_j z^{-j}}$$



HP digital filter – Infinite impulse response filter (IIR)

$$H(z) = \frac{\sum_{i=0}^{P} b_i z^{-i}}{1 + \sum_{j=1}^{Q} a_j z^{-j}}$$



Matlab demo

filterDesigner

Convolution Theorem

Convolution Theorem

Fourier transform of g: F[g] = G convolution: *

Fourier transform of g: F[h] = H multiplication: .

The Fourier transform of the convolution of two functions is the product of their Fourier transforms

$$\mathrm{F}[g*h]=\mathrm{F}[g].\mathrm{F}[h]=G.H$$

The inverse Fourier transform of the product of two Fourier transforms (= the convolution of the two inverse Fourier transforms) = convolution of the signals in the time domain

$$\mathrm{F}^{-1}[G.H]=\mathrm{F}^{-1}[G]*\mathrm{F}^{-1}[H]=g*h$$

Convolution in time domain is equivalent to multiplication in frequency domain and vice-versa.

Convolution Theorem

Convolution in time domain (spatial domain for images) is equivalent to multiplication in frequency domain and vice-versa.

time domain (spatial domain for images) | frequency domain

$$g = f * h \quad \longleftrightarrow \quad G = FH$$

$$g = fh \quad \longleftrightarrow \quad G = F * H$$

Theory of windowing - explanation

Spectral leakage explanation by Convolution theorem

Convolution Theorem

$$\mathrm{F}[g.h] = \mathrm{F}[g] * \mathrm{F}[h] = G * H$$

spectrum of g.h = G convolved by H

In the case of windowing:

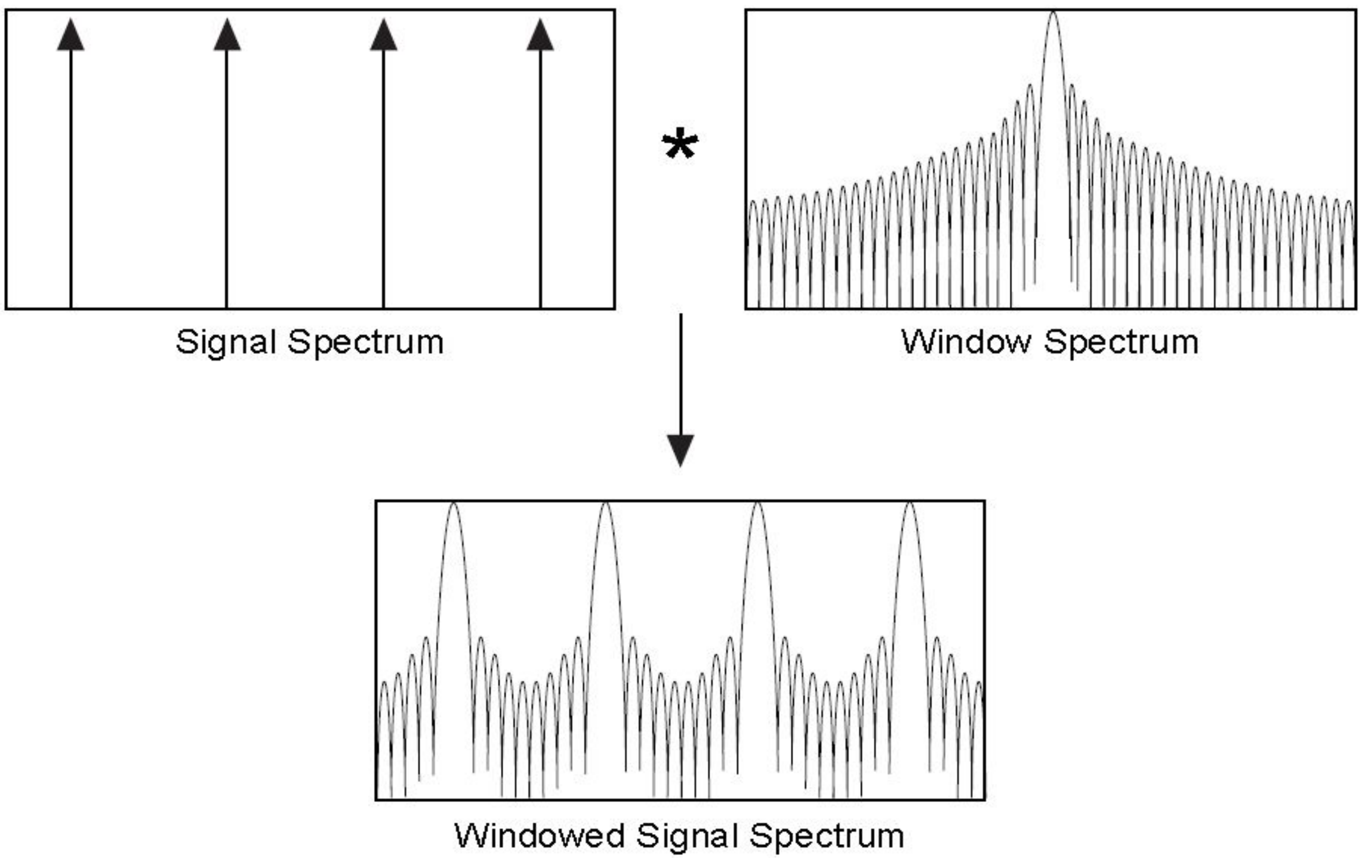
g: signal

h: window (also w)

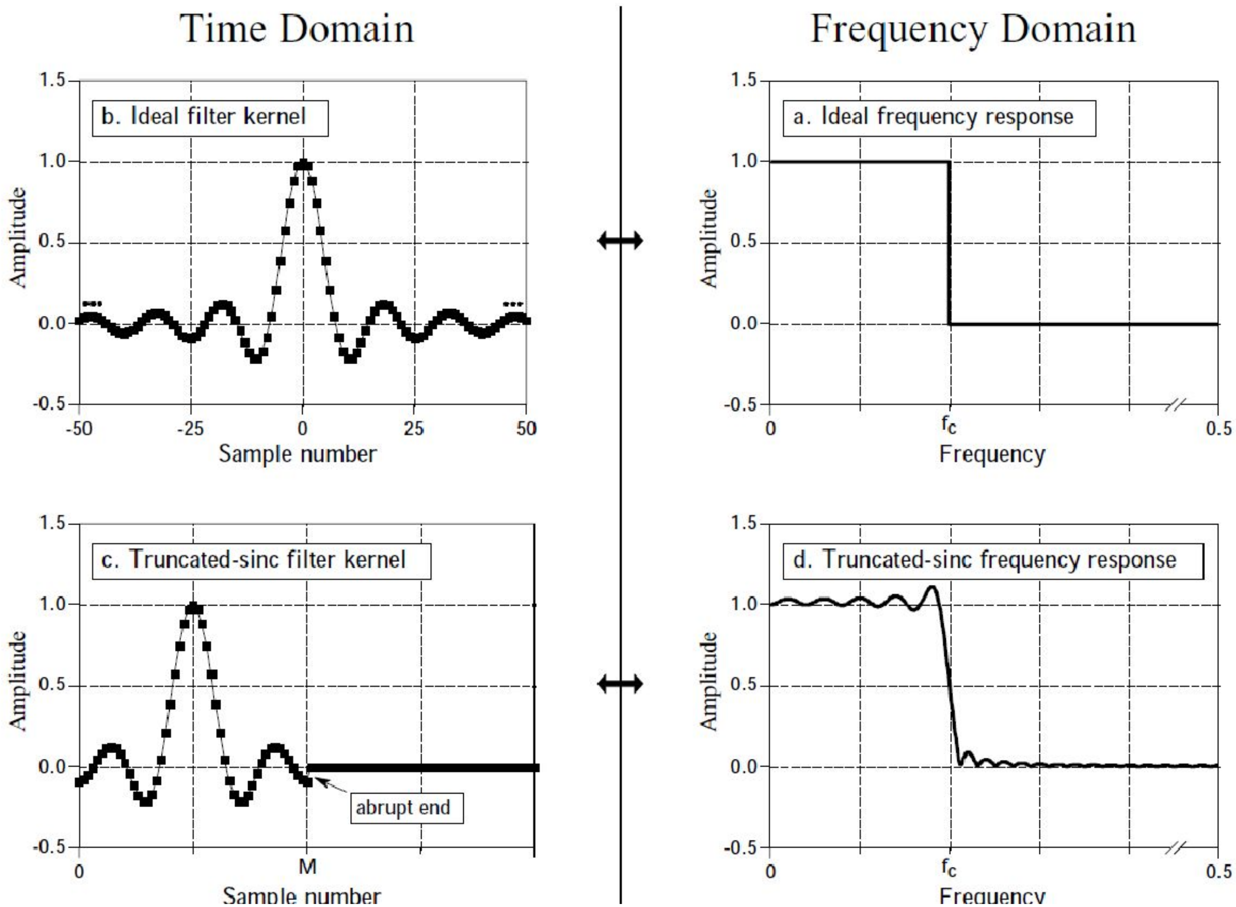
Frequency Characteristics of a Windowed Spectrum

Result of a windowed spectrum:

Signal spectrum convolved by the Window spectrum



Ideal filter - explained



Parseval's Theorem

Parseval's Theorem

Parseval's Theorem states that a digital signal must have the same power in both the time (sample) domain and the frequency domain:

$$\boxed{\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2 .}$$

Deconvolution

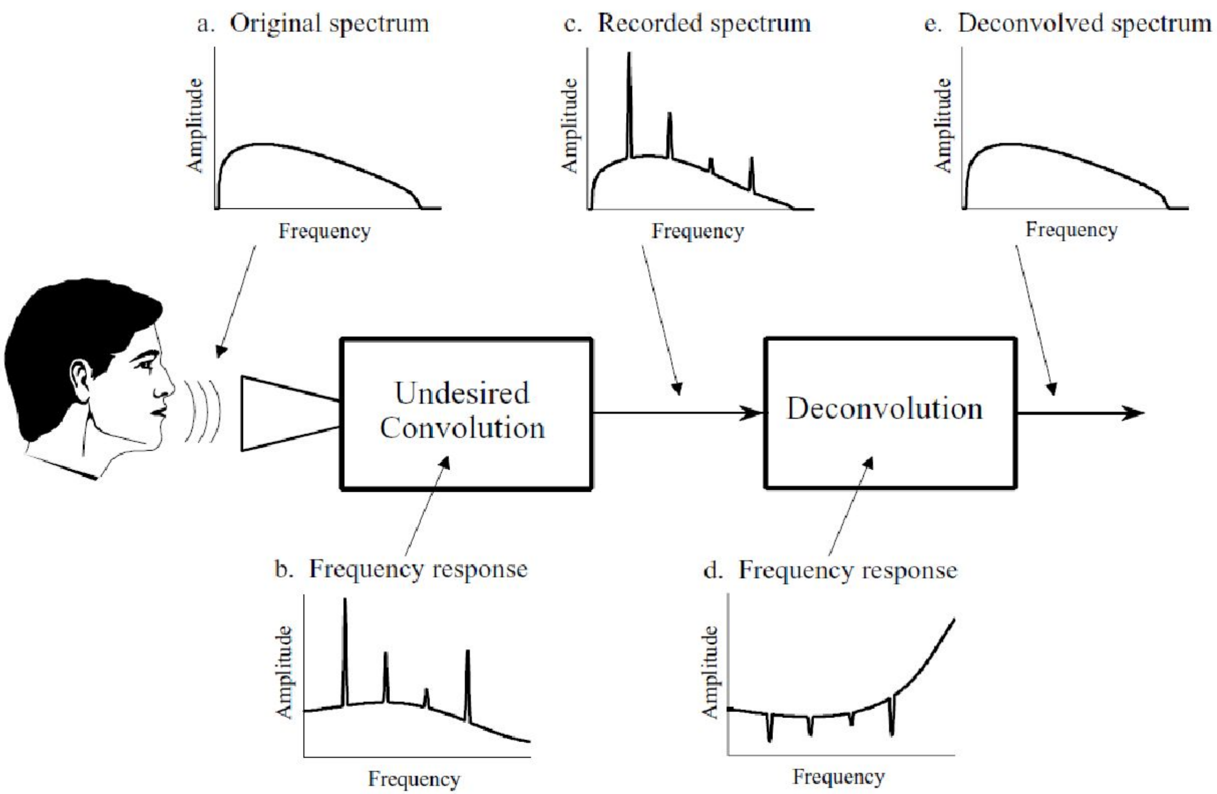
Deconvolution

The goal of deconvolution is to recreate the signal as it existed before the convolution took place.

This usually requires the characteristics of the convolution (i.e., the impulse or frequency response) to be known.

This can be distinguished from blind deconvolution, where the characteristics of the parasitic convolution are not known. Blind deconvolution is a much more difficult problem that has no general solution, and the approach must be tailored to the particular application.

Deconvolution - example



The frequency response of the deconvolution filter

Music audio signal

Audio signal of music
Terminology and Concepts

There are four subjective qualities that are particularly useful in characterizing sound events:

- pitch,
- loudness,
- duration,
- and timbre (farba zvuku)

Pitch and Fundamental frequency (F0)

Pitch is a perceptual attribute which allows the ordering of sounds on a frequency-related scale extending from low to high. More exactly, pitch is defined as the frequency of a sine wave that is matched to the target sound by human listeners.

Fundamental frequency is the corresponding physical term and is defined for periodic or nearly periodic sounds only.

$T[s]=1/f[Hz]$

Loudness

The perceived loudness of an acoustic signal has a non-trivial connection to its physical properties, and computational models of loudness perception constitute a fundamental part of psychoacoustics.

In music processing - express the level of sounds with their mean-square power and to apply a logarithmic (decibel) scale to deal with the wide dynamic range involved.

Duration of a sound

one-to-one mapping to its physical duration in cases where this can be unambiguously determined

Timbre

Timbre is sometimes referred to as sound 'colour' and is closely related to the recognition of sound sources.

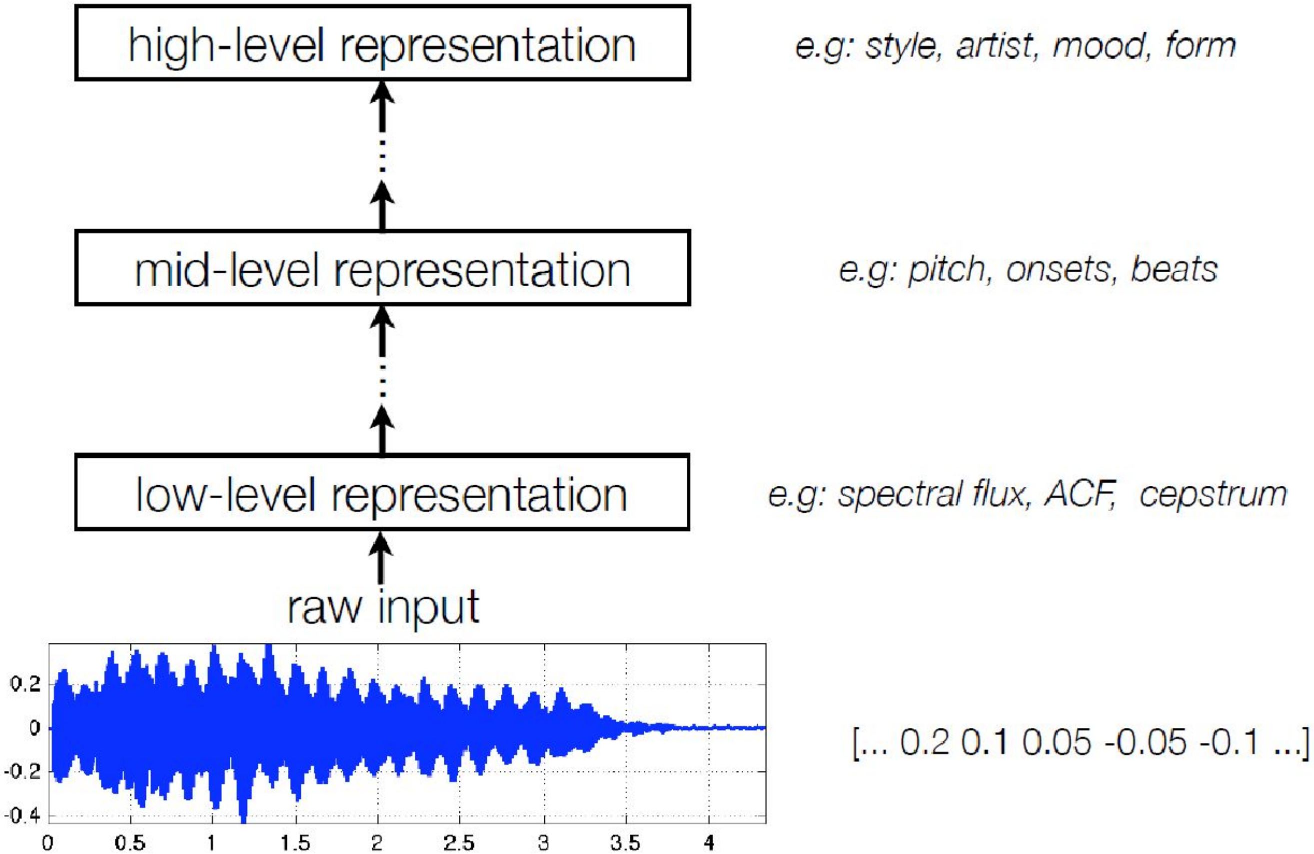
For example, the sounds of the violin and the flute may be identical in their pitch, loudness, and duration, but are still easily distinguished by their timbre.

Recomended readimgs

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1.html

Musical information retrieval (MIR)
Music transcription

Music signal analysis



Musical information retrieval (MIR) Music transcription

Aims at extending the understanding and usefulness of music data, through the research, development and application of computational approaches and tools

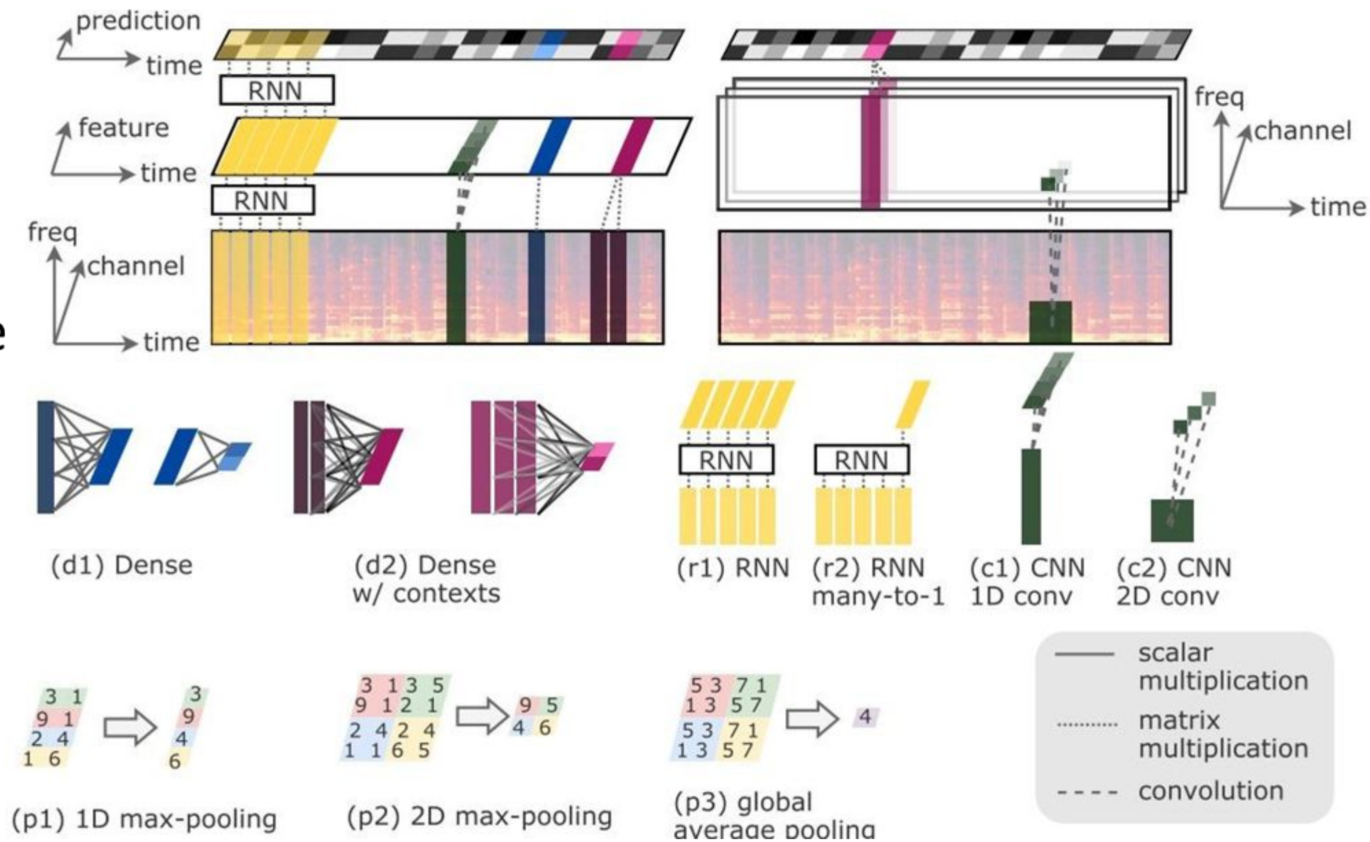
Music information: bibliographical, tags, scores, MIDI, audio, etc.

Organized into the International Society for Music Information Retrieval and its conference (ISMIR)

Music Analysis in Spectral Domain

Common setup:

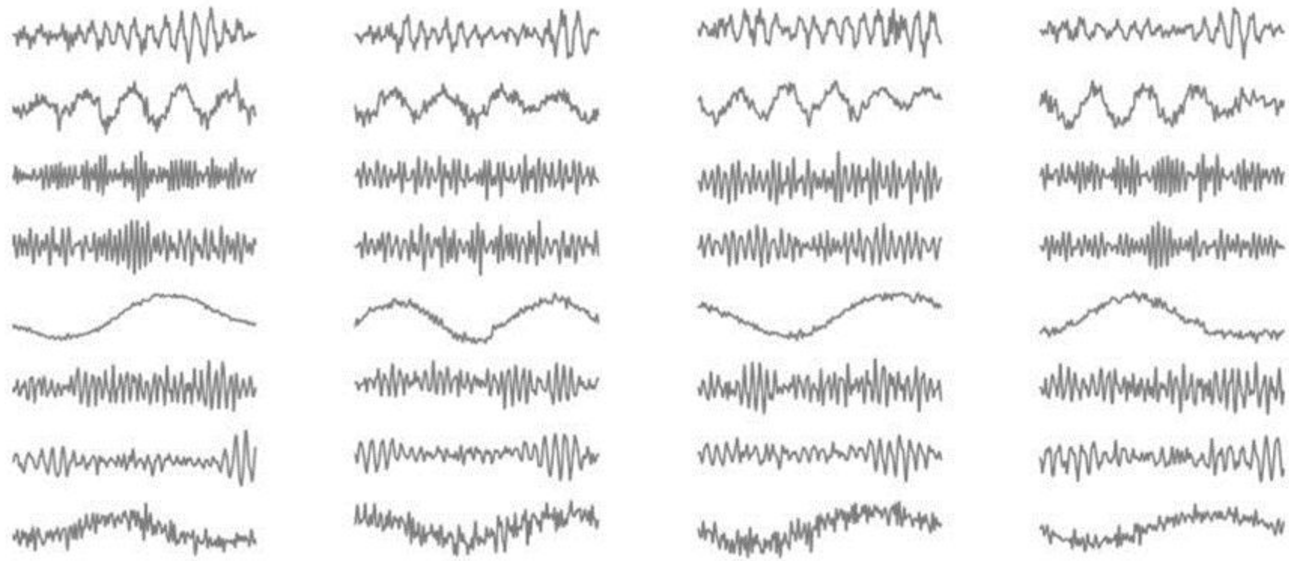
- frame-based processing
- convolution over magnitude spectrogram
- RNN on top



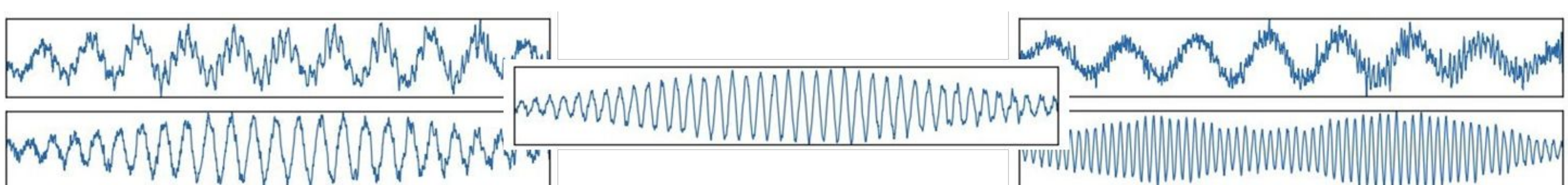
[Choi et al. 2017]

Music Analysis inTime Domain

Common setup:
1D convolution
dilated, strided
stacked
FC layer / 1x1
conv on top



[Thickstun et al. 2017]



[Thickstun et al. 2017]

Synthetic Sounds

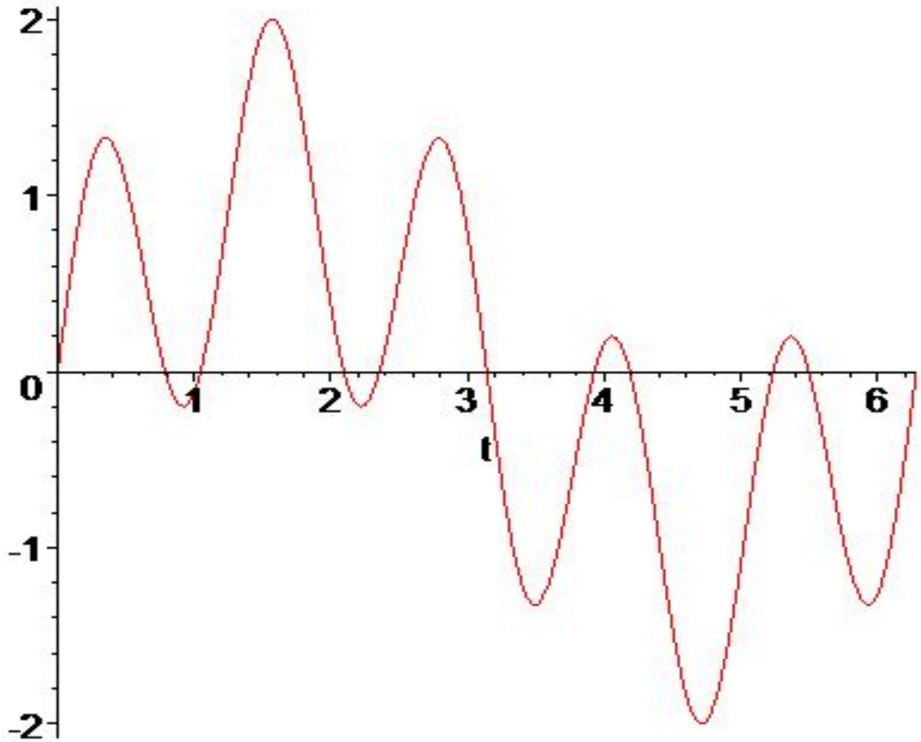
Additive Synthesis

We add together different soundwaves sample-by-sample to create a new sound

In additive synthesis, we start with simple sounds and add them together to form more complex ones.

Example

sin(5*x) + sin(x),
x=0.....2*pi



Changing Sound Parameters

The quality of a synthesized sound can be improved by varying its parameters over time

- partial frequencies (harmonics),
- amplitudes,
- envelope .

Subtractive Synthesis: Filters

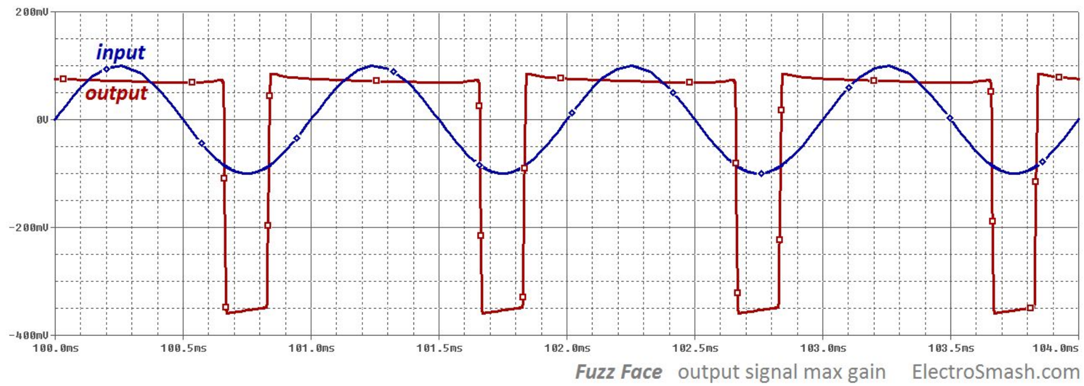
In subtractive synthesis, we start with a complex sound (like noise) and subtract, or filter out, parts of it.

Waveshaping Synthesis (guitar effect...)

Waveshaping technique turns simple sounds into complex sounds. We can take a pure tone, like a sine wave, and transform it into a harmonically rich sound by changing its shape.

A guitar fuzz box is an example of a waveshaper. An unamplified electric guitar sound is fairly close to a sine wave. But the fuzz box amplifies it and gives it sharp corners.

Waveshaper generally have much more energy in their higher-frequency harmonics, which gives them a "richer" sound.



Wave Table synthesis

A more accurate way of generating sounds from digital signals.

In this technique, the actual digital samples of sounds from real instruments are stored.

Since wave tables are stored in memory on the sound card, they can be manipulated by software so that sounds can be combined, edited, and enhanced.

Generative Neural Networks for Music generation

A separate lecture will be devoted to this topic....

Symbolic Format of Music

[Symbolic Format: CSV]

[Symbolic Format: MusicXML]

[Symbolic Format: MIDI]

MIDI

Musical Instrument Digital Interface

MIDI history

Musical Instrument Digital Interface

end of the 1970s:

electronic musical devices were becoming increasingly common

not compatible

-Dave Smith was working on a polyphonic analogue synthesizer (Prophet 5). Dave's innovative idea: instrument with multiple identical sound-producing engines ("voices") , all the parameters of the voices digitally controllable.

1991- the MIDI Show Control (MSC) protocolwas ratified by the MIDI Manufacturers Association.

MIDI history

Hammond Novachord, The First Synthesizer,

MIDI: Musical Instrument Digital Interface

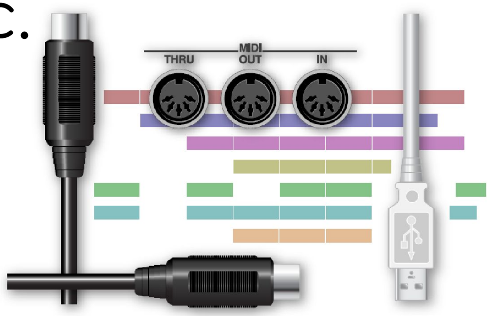
MIDI codes “events” that stand for the production of sounds. E.g., a MIDI event might include

values for the pitch of a single note,

its duration,

its volume.

MIDI is a standard adopted by the electronic music industry for controlling devices, such as synthesizers and sound cards, that produce music.



The current MIDI specification includes:

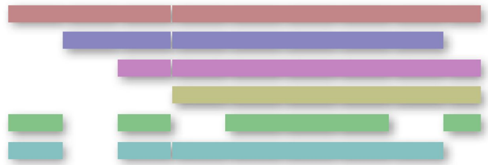
- a hardware scheme for physically connecting electronic musical instruments (MIDI Interface, MIDI Adapter, MIDI Cable).
- a data encoding scheme for storage and transmission of musical performance and control event data as messages. Typical message types include:
  - musical notation, pitch, velocity, control signals for parameters
  - (MIDI messages, MIDI file)
- communication protocols for transmitting and synchronizing musical performance and control event data.
- schemes for categorizing instrument and percussive sounds or timbres, also referred to as patches or

MIDI Controllers

A MIDI device that controls another is referred to as a MIDI "controller."

- Keyboard Controllers
- Guitar Controllers
- Drum Controllers
- …

MIDI Sequencers



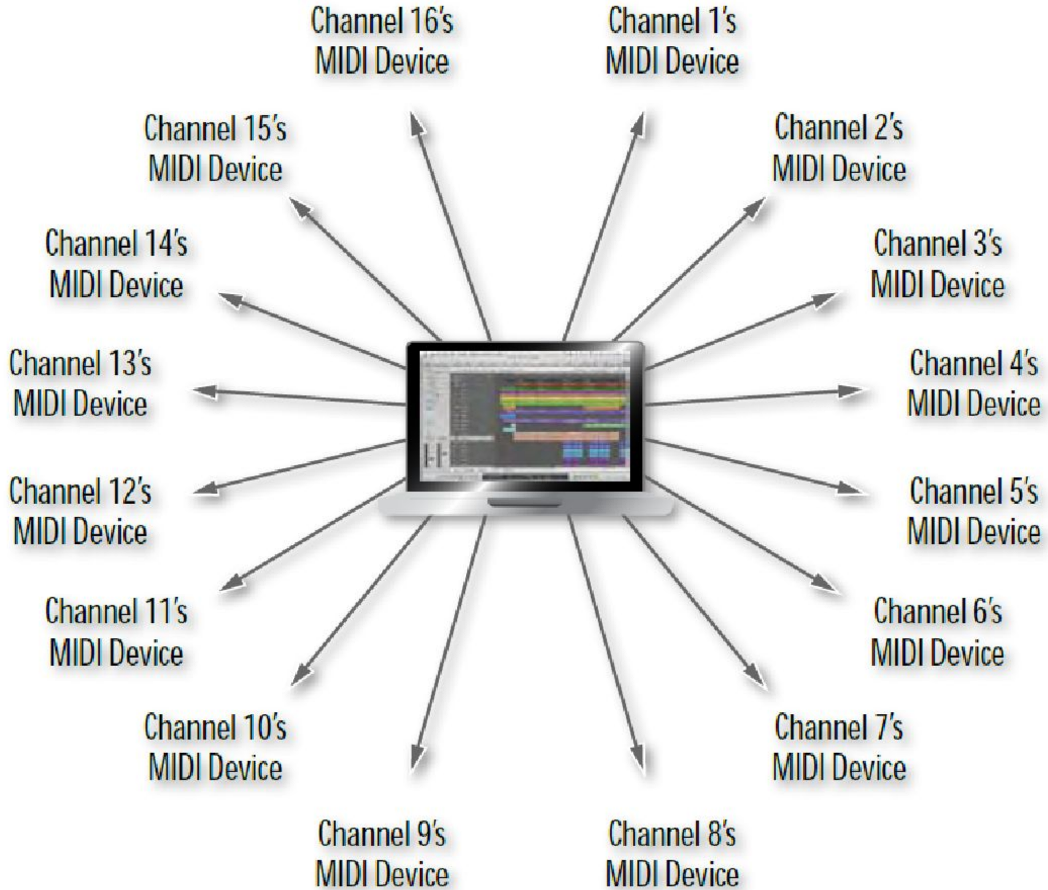
During recording, the sequencer captures and plays back live MIDI performances.

Performances can also be constructed slowly, note-by-note, using a variety of methods that may include “step sequencing” and onscreen pencil tools that let you “draw” the notes you want.

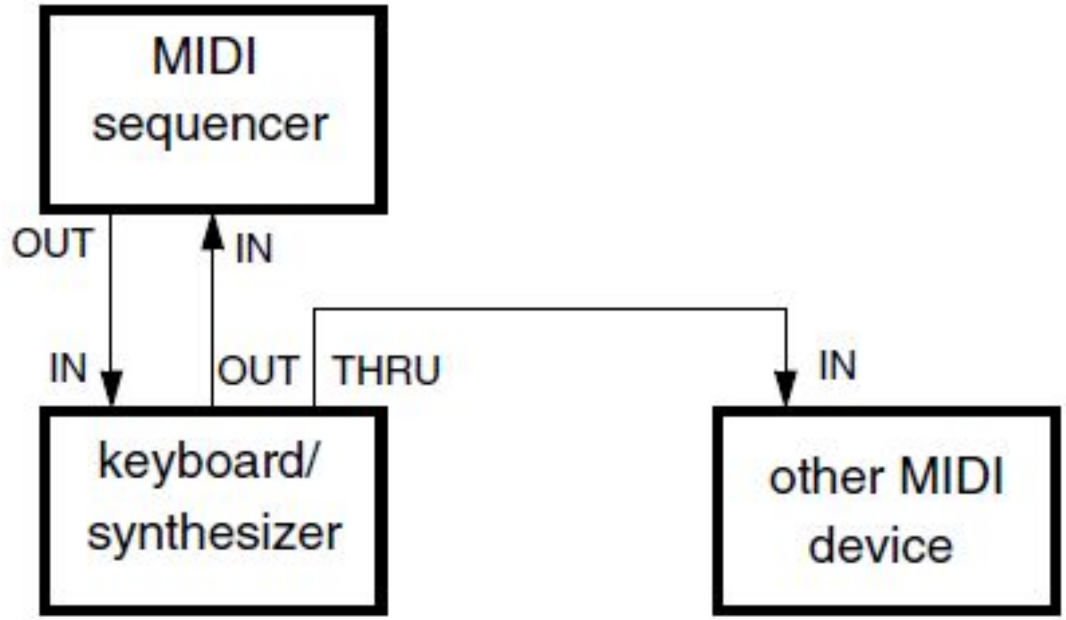
Each recorded performance is typically stored on its own track, and a sequencer may offer anywhere from 16 tracks to over a hundred.

MIDI Channels

A single MIDI connection can carry 16 independent streams, or "channels," of MIDI information.



Example of a MIDI system configuration



Part off Instrument Patch Map

- **A1. General MIDI Instrument Patch Map:**

Prog#	Instrument	Prog#	Instrument
-----	----------	-----	----------
(1-8	PIANO)	(9-16	CHROM PERCUSSION)
1	Acoustic Grand	9	Celesta
2	Bright Acoustic	10	Glockenspiel
3	Electric Grand	11	Music Box
4	Honky-Tonk	12	Vibraphone
5	Electric Piano 1	13	Marimba
6	Electric Piano 2	14	Xylophone
7	Harpsichord	15	Tubular Bells
8	Clav	16	Dulcimer
(17-24	ORGAN)	(25-32	GUITAR)
17	Drawbar Organ	25	Acoustic Guitar(nylon)
18	Percussive Organ	26	Acoustic Guitar(steel)
19	Rock Organ	27	Electric Guitar(jazz)
20	Church Organ	28	Electric Guitar(clean)
21	Reed Organ	29	Electric Guitar(muted)
22	Accoridan	30	Overdriven Guitar
23	Harmonica	31	Distortion Guitar
24	Tango Accordian	32	Guitar Harmonics
...			
….			
….			
(113-120	PERCUSSIVE)	(121-128	SOUND EFFECTS)
113	Tinkle Bell	121	Guitar Fret Noise
114	Agogo	122	Breath Noise
115	Steel Drums	123	Seashore
116	Woodblock	124	Bird Tweet
117	Taiko Drum	125	Telephone Ring
118	Melodic Tom	126	Helicopter
119	Synth Drum	127	Applause
120	Reverse Cymbal	128	Gunshot

Part off Percussion Key Map

- **A2. General MIDI Percussion Key Map:**

MIDI Key	Drum Sound	MIDI Key	Drum Sound
35	Acoustic Bass Drum	59	Ride Cymbal 2
36	Bass Drum 1	60	Hi Bongo
37	Side Stick	61	Low Bongo
38	Acoustic Snare	62	Mute Hi Conga
39	Hand Clap	63	Open Hi Conga
40	Electric Snare	64	Low Conga
41	Low Floor Tom	65	High Timbale
42	Closed Hi-Hat	66	Low Timbale
43	High Floor Tom	67	High Agogo
44	Pedal Hi-Hat	68	Low Agogo
45	Low Tom	69	Cabasa
46	Open Hi-Hat	70	Maracas
47	Low-Mid Tom	71	Short Whistle
48	Hi-Mid Tom	72	Long Whistle
49	Crash Cymbal 1	73	Short Guiro
50	High Tom	74	Long Guiro
51	Ride Cymbal 1	75	Claves

…. .
…. .
….

The data in a MIDI

The data in a MIDI status byte are in the range: 128 - 255;

Data bytes are in the range: 0 and 127.

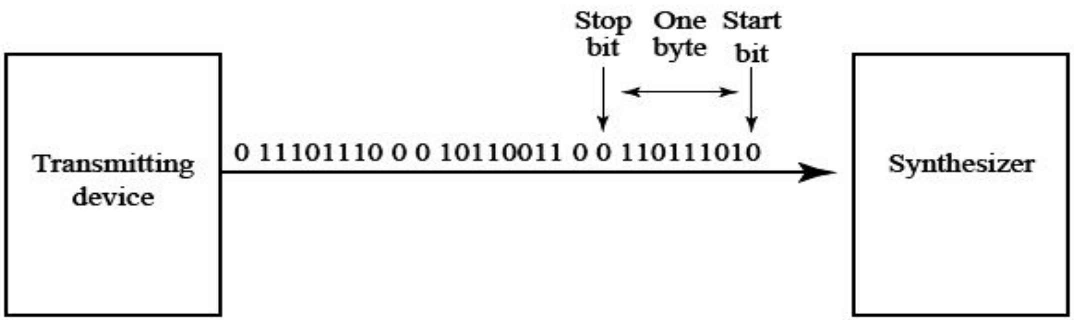
Actual MIDI words are 10-bit, including a 0 start and 0 stop bit.

Stream for typical MIDI messages:

10-bit bytes (words) - these consist of

{Status byte, Data Byte, Data Byte} =

{Note On, Note Number, Note Velocity}



MIDI and Video

Some MIDI devices allow performers

—called “VJs”—

to manipulate video images onstage, creating exciting visuals.

General MIDI (GM 1)
General MIDI 2

https://www.midi.org

https://www.midi.org/specifications-old/item/general-midi

General MIDI 2 includes everything in General MIDI, adding more sounds, standards for sound editing, and other features...

Wave2Midi2Wave