AUDIO ANALYSIS AND VISUALIZATION

AIST2010 Lecture 3

Fourier Analysis

Spectral Visualization

MATLAB Programming

OUTLINE

SUMMATION OF WAVES

Any continuous function, e.g. audio signal, can be expressed as a sum of (infinite many) sinusoidal waves

- Proved by French scientist and mathematician Jean Baptiste Fourier (1768–1830)
- Each sinusoidal wave has their own amplitude and frequency

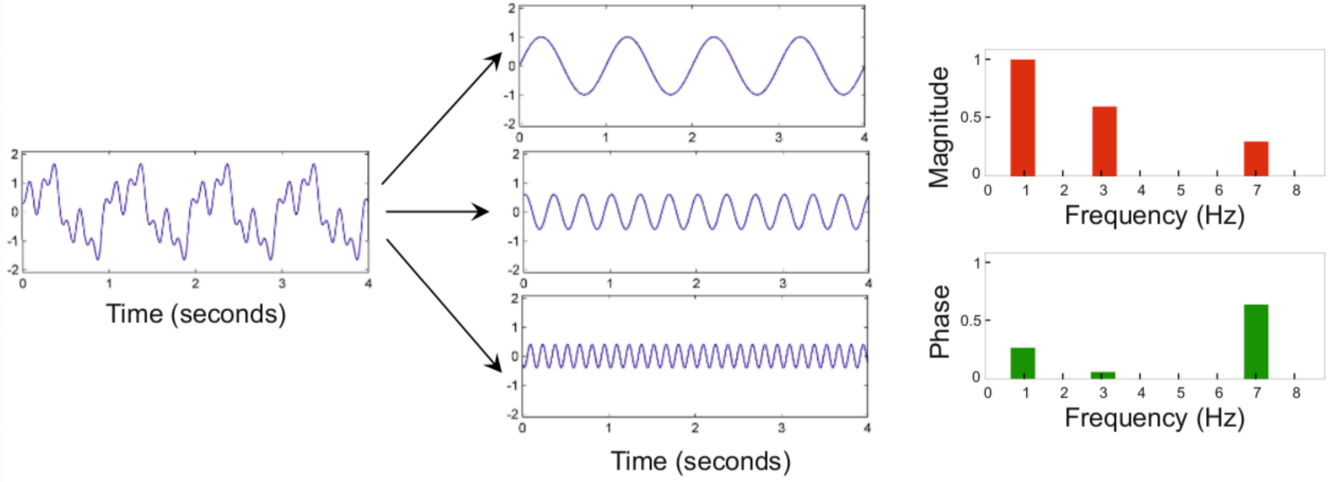


Image from: ***Fund. of Music Processing***, *p.70*

FUNDAMENTAL FREQUENCY AND HARMONICS

In particular, some waveforms sound "better"

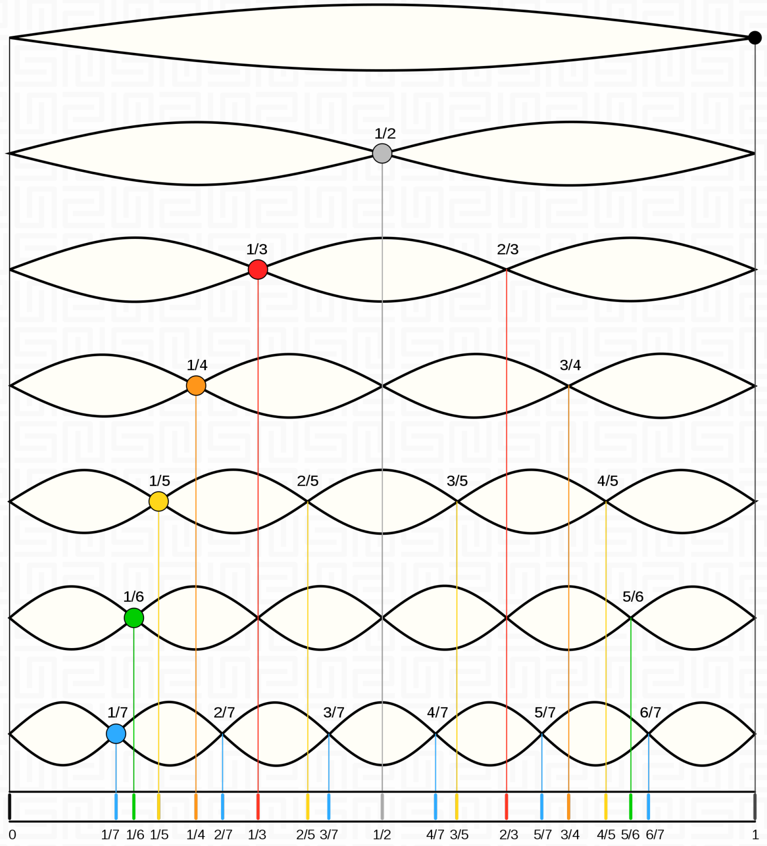
- Only/mainly frequency components in relationship of integer multiples
- The GCD is often called the ***fundamental frequency*** f_0, and the others are ***harmonics*** f_k

$$f_k = k f_0$$

- Harmonics are sometimes called **partials** and **overtones** too, but may be numbered *differently!*

Read: https://en.wikipedia.org/wiki/Harmonic#Partials,_overtones,_and_harmonics

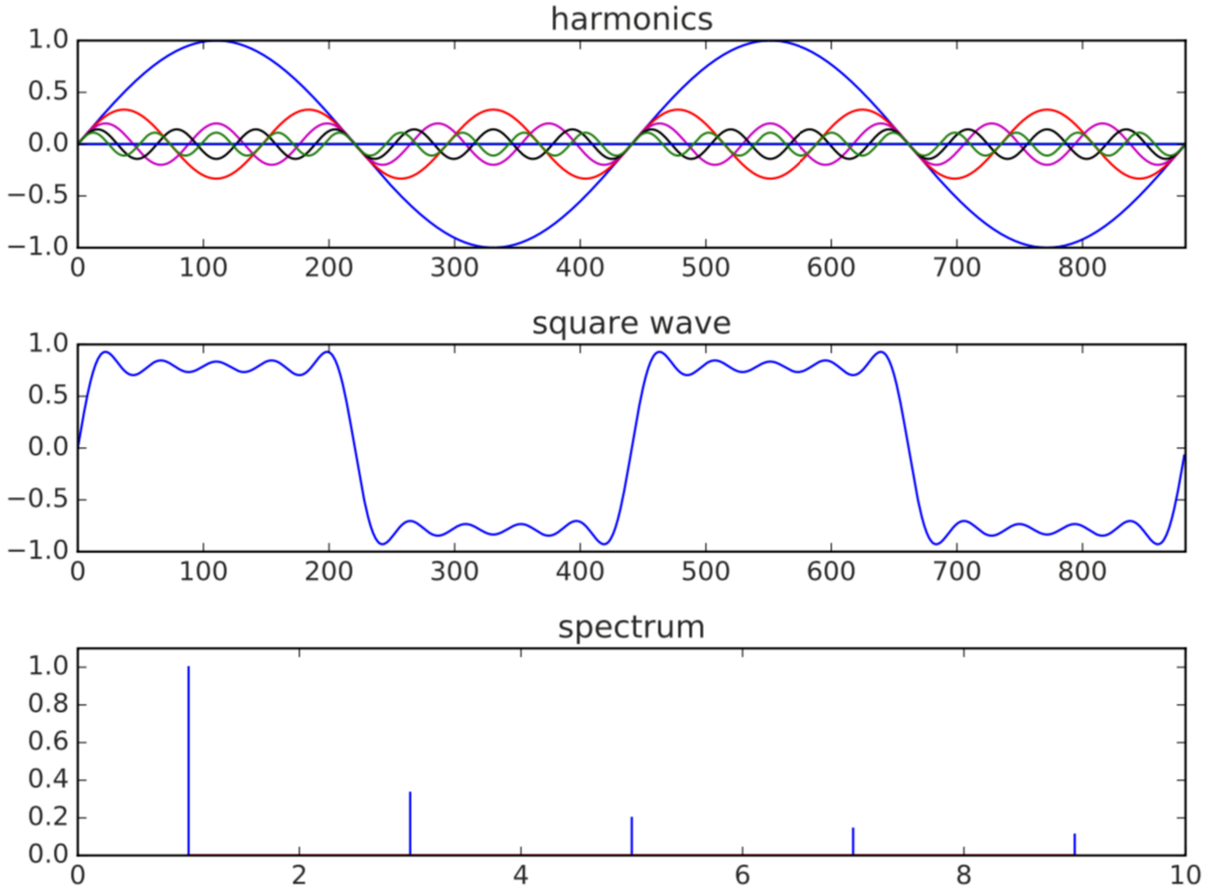
Image from: *https://en.wikipedia.org/wiki/Harmonic*



Harmonics of multiple relationship

Image from: ***Comp. Music Instruments***, *p.63*

OTHER WAVEFORMS



Decomposing a square wave

Sawtooth wave

- Sum of all harmonics, with each decreasing in amplitude

Square wave

- Sum of odd harmonics

Triangle wave

- Sum of odd harmonics, with a negative sign for alternating odd harmonics, and each decreasing in amplitude

Some more animations of the square wave decomposition here: http://bilimneguzellan.net/fuyye-serisi/

MATHEMATICAL REPRESENTATION

Sometimes you may see this as well:
$g(t) \coloneqq A\cos(2\pi f t + \varphi)$

A sinusoidal wave can be represented as…

$$g(t) \coloneqq A\sin(2\pi f t + \varphi)$$

Sometimes you may see this: $\omega = 2\pi f$

where

- A = ***amplitude***, i.e. loudness of the sound
- f = ***frequency*** (in Hz), i.e. pitch of the sound
  - Note: ***period*** T = 1 / f, in seconds
- φ = ***phase*** (in radians, where 2π rad=360°), i.e. relative position of an oscillation within its cycle
  - Note: A phase shift by φ+2π has the same effect as a phase shift by φ

FOURIER ANALYSIS

Transformation from time domain (amplitude vs. time) into frequency domain (magnitude vs. frequency)

$$\boxed{e^{it} = \cos t + i \sin t}$$

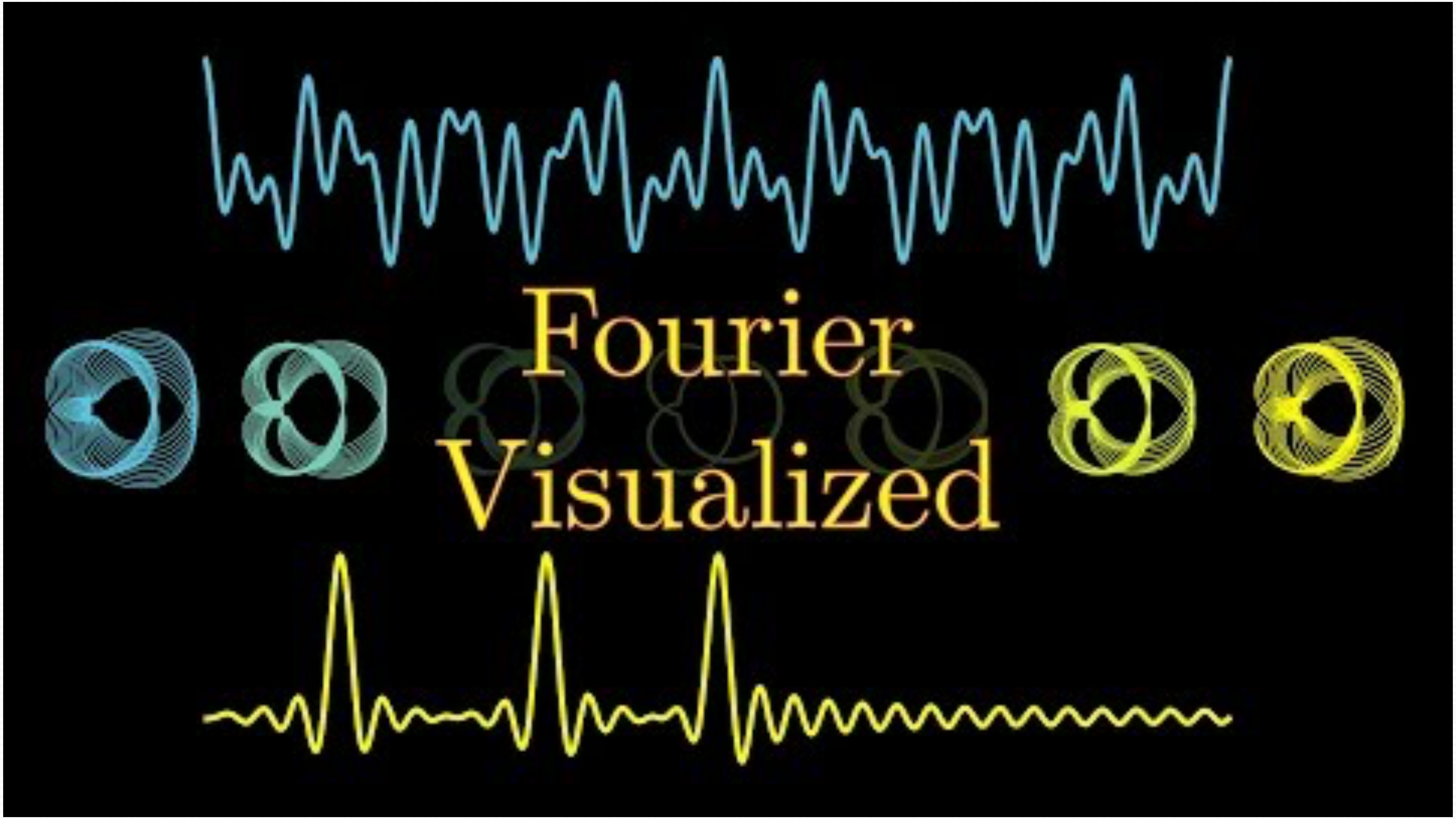
$$\mathcal{F}\{g(t)\} = \hat{g}(f) = \int_{t\in\mathbb{R}} g(t)e^{-2\pi i f t}\, dt$$

- You may view it as counting the occurrence of frequencies in the waveform

Yet, this function for continuous f and t cannot be applied to digital signals!

Read: https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/

Video from: https://youtu.be/spUNpyF58BY



DISCRETE FOURIER TRANSFORM (DFT)

Since the input values (samples) are equally spaced, the Fourier Transform for sound samples is discrete

- Sum of finite series of sinusoidal waves

$$e^{it} = \cos t + i \sin t$$

From a sequence of N (complex) samples

$$\{x_n\} := x_0, x_1, \dots, x_{N-1}$$

into a sequence of N complex numbers

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{i 2\pi k \frac{n}{N}}$$

To find the energy at a particular frequency, spin your signal around a circle at that frequency, and average a bunch of points along that path.

Image from: https://blog.revolutionanalytics.com/2014/01/the-fourier-transform-explained-in-one-sentence.html

DISCRETE FOURIER TRANSFORM (DFT)

$$X_k := \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i k n}{N}}$$

- The X_k series is called the DFT coefficients (of N ***frequency bins***)
  - Magnitude $|X_k| = \sqrt{Re(X_k)^2 + Im(X_k)^2}$
  - Phase $arg(X_k) = \arctan(Im(X_k)/Re(X_k))$
  - Bin frequency $f_k = f_s \cdot \frac{k}{N}$

$$\boxed{e^{it} = \cos t + i \sin t}$$

*DFT is a very popular tool for **digital** signal processing*

- Usually implemented as Fast Fourier Transform (**FFT**)
  - Ordinary DFT is $O(N^2)$ while FFT is $O(N \log N)$
- Luckily, you can often use FFT simply as a black box in programming libraries, without understanding the math behind!

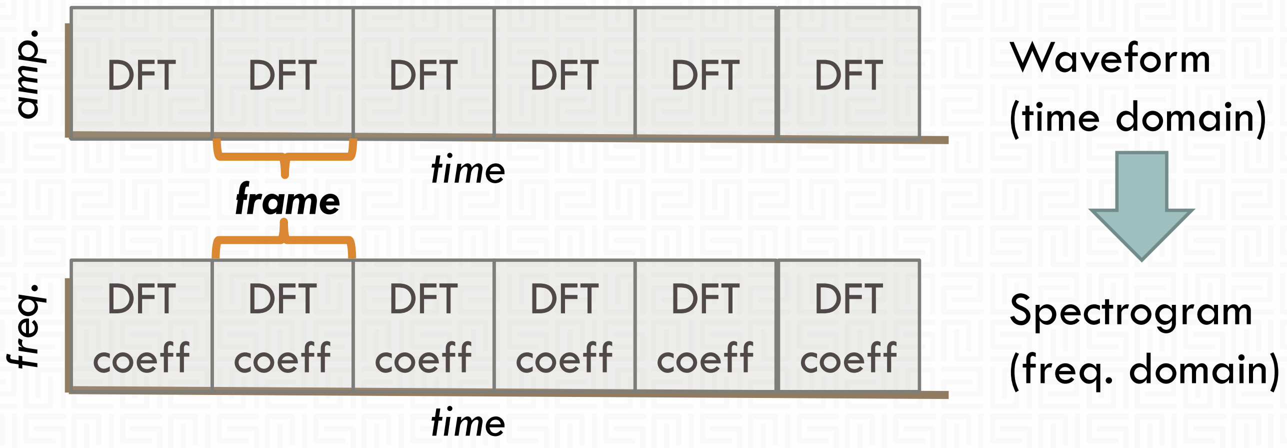
SHORT-TIME FOURIER TRANSFORM (STFT)

DFT can only show the general "histogram" of frequencies

- The appearance of frequencies in the whole analyzed sound

STFT breaks the process into multiple DFT/FFT in time segments

- Analysis ***frames***
- The result is a ***spectrogram***
  - Magnitude vs. frequency vs. time
  - Magnitude often represented in the colour dimension



WINDOW FUNCTION

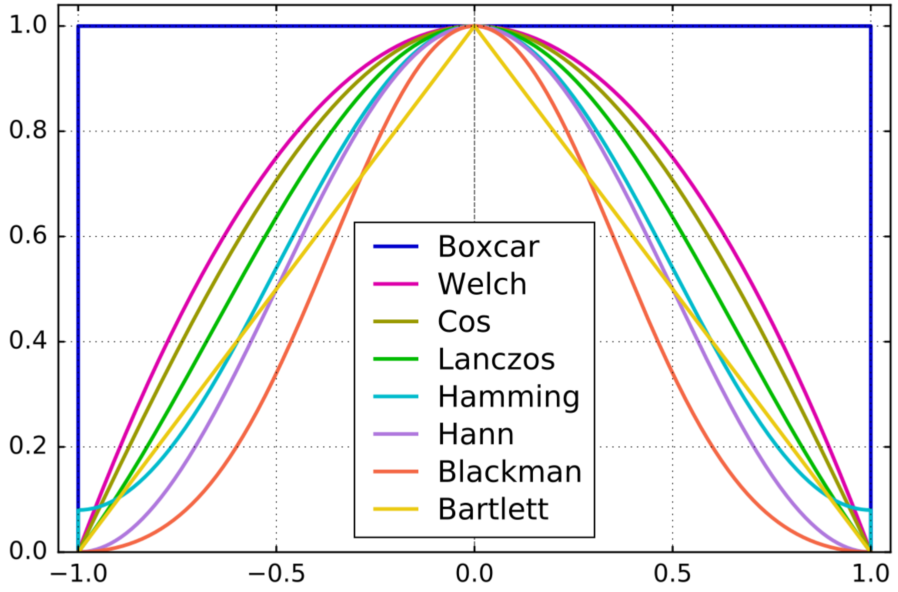
For a step-by-step Fourier analysis, a window function is needed

- The value is 1 only for a short time, and 0 otherwise

The shape of the window function will affect frequency responses

- E.g. The sharp edges of a rectangular window will result in high frequency components
- Usual choices to avoid ***spectral leakage***: Hamming Window, Hann Window, …

Image from: https://commons.wikimedia.org/wiki/File:Mplwp_window-functions-symmetric.svg

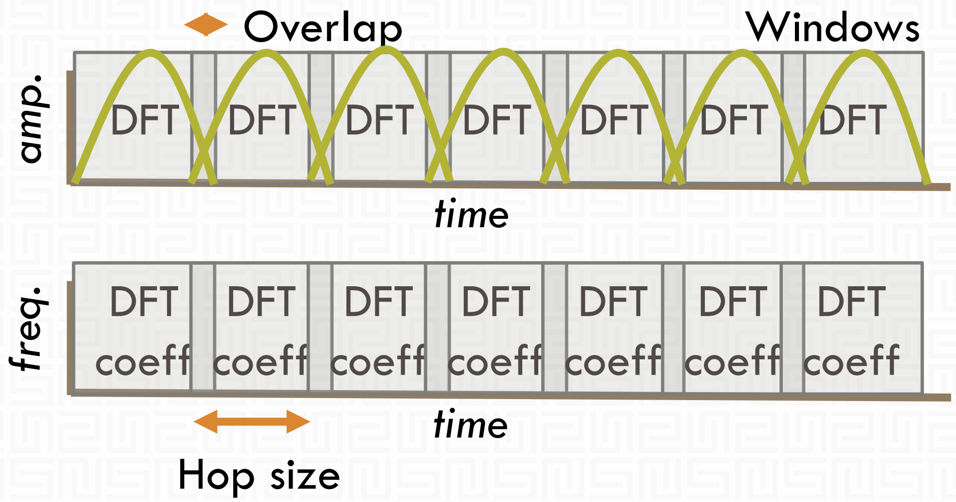


Comparison of windows

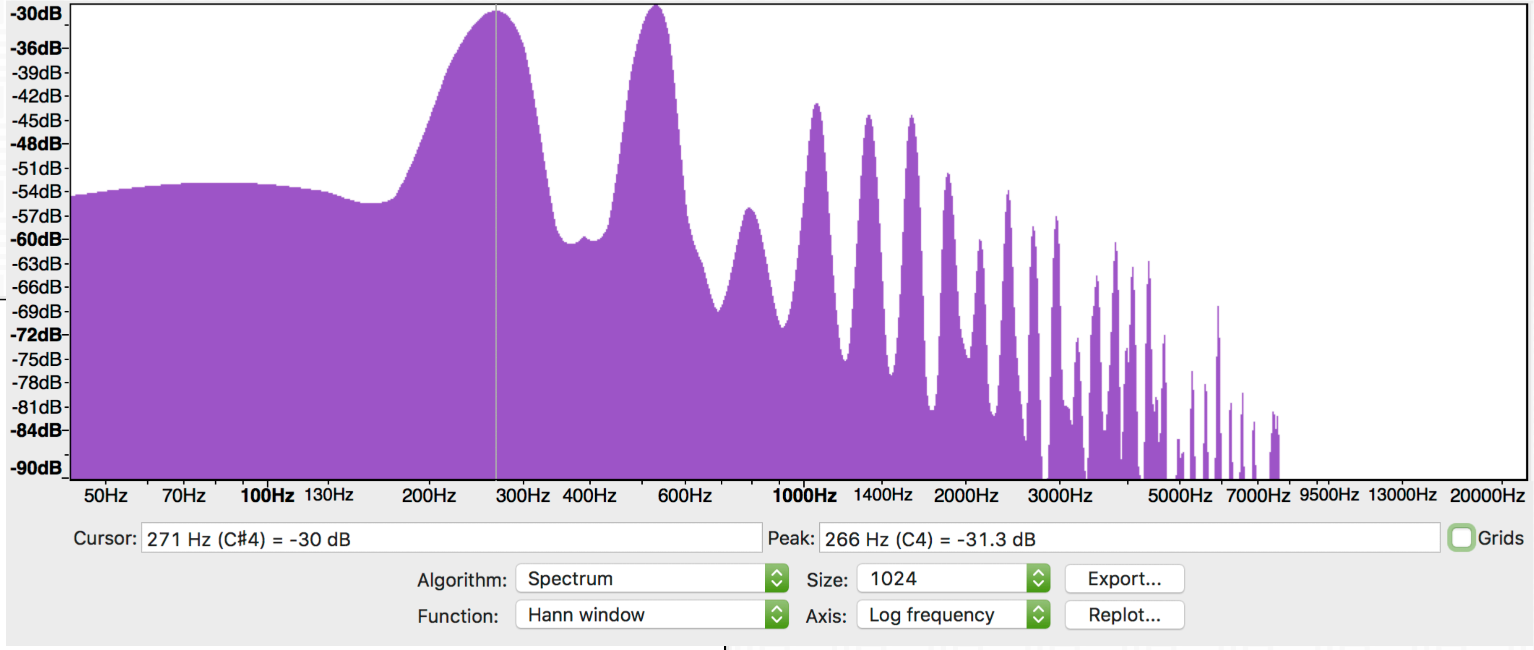
FREQUENCY BINS

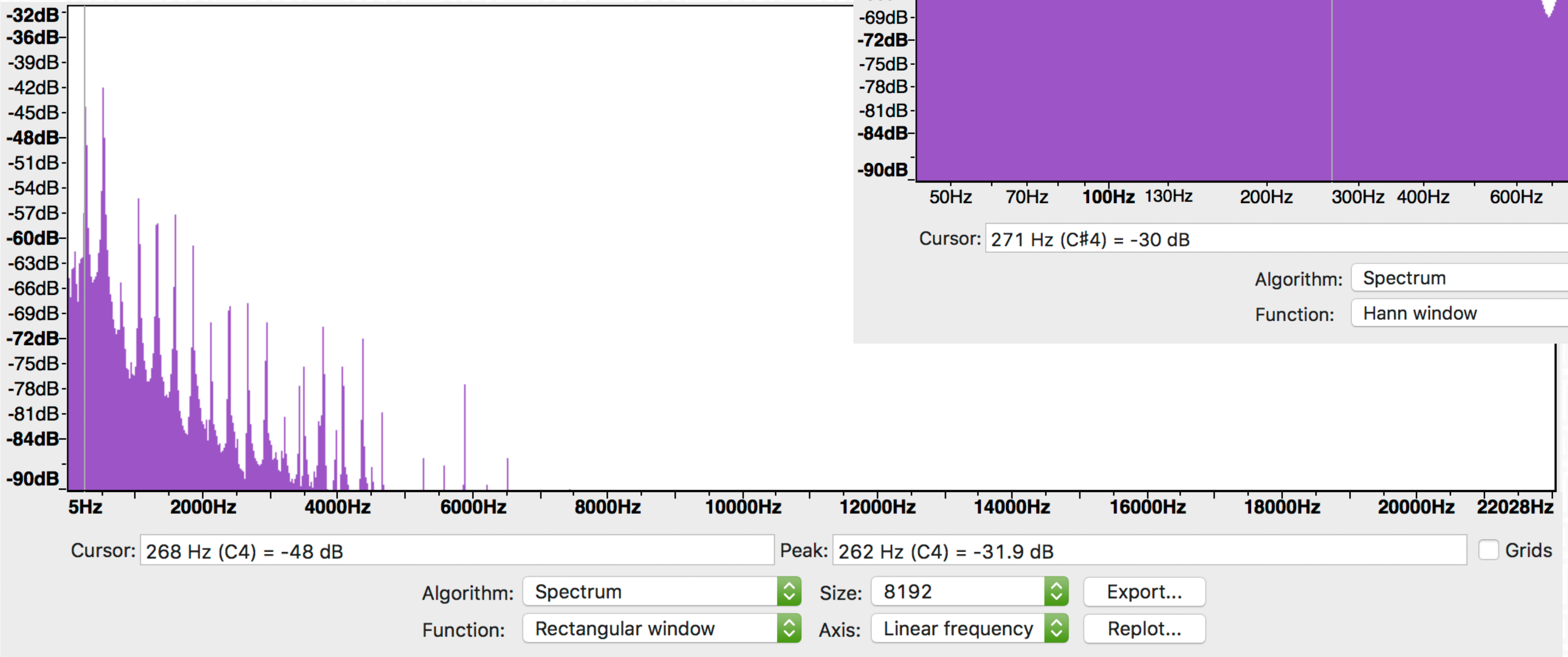
Usual window size: ***powers of two*** *to facilitate FFT, e.g. 1024, …*

- Often with an overlap of 50% to compensate loss of data by windowing
- 1024-point FFT = 1024 time samples = 1024 frequency bins
- The more samples in the window, the higher the frequency resolution
  - The results fall into frequency bins of smaller range
- The higher the frequency resolution, the lower the time resolution
  - Basically it is a trade-off between time and frequency

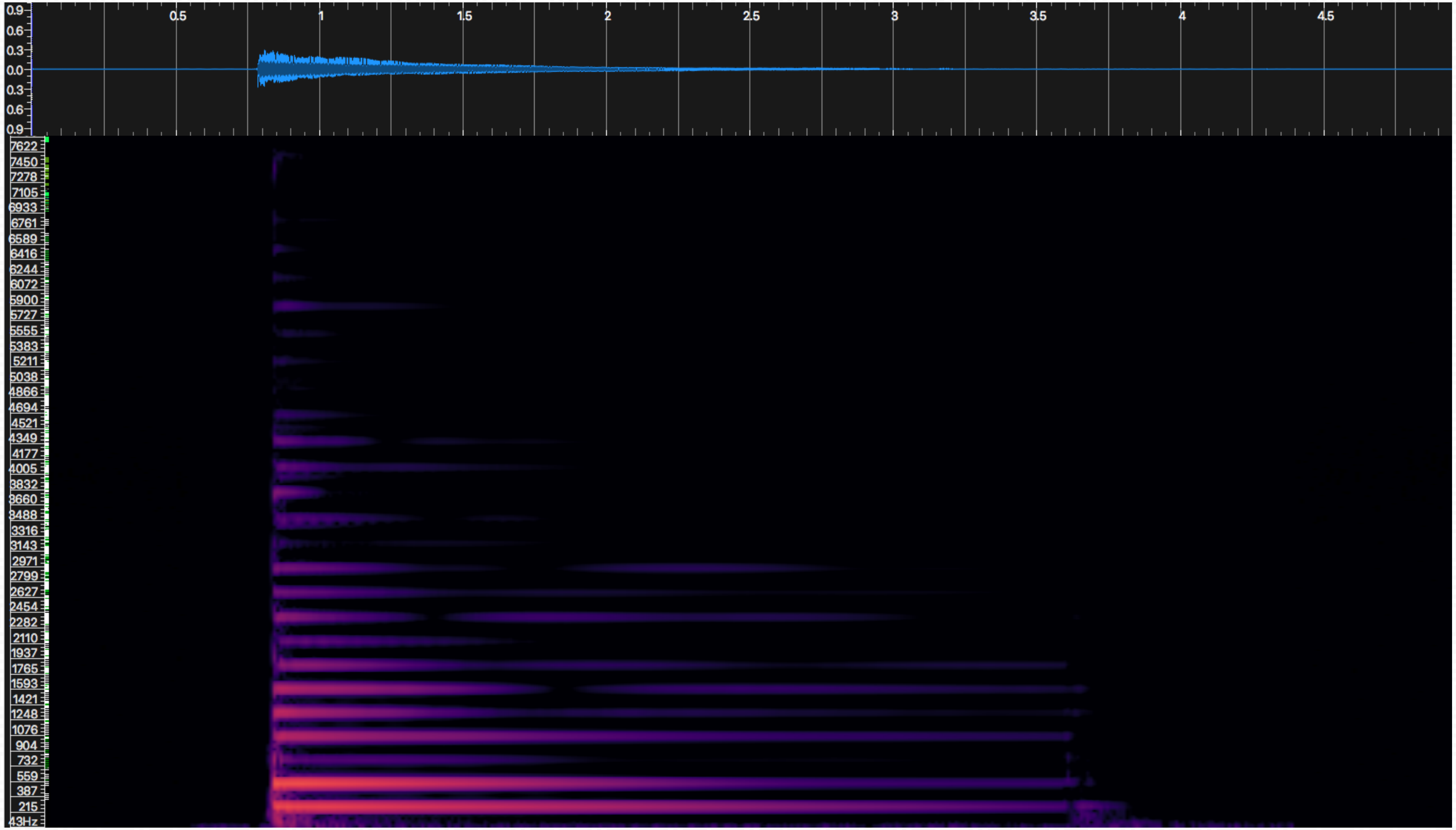


HOW TO READ THE PLOTS?





HOW TO READ THE PLOTS?



INVERSE OF THE FOURIER TRANSFORM

Rebuilding audio signal from the Fourier analysis data

Inverse DFT

- From frequency domain back to time domain
- Can easily be expressed in terms of the DFT

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i k n}{N}} \qquad x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i k n}{N}}$$

Inverse STFT

- Overlap-add (OLA) method

CEPSTRUM

What would happen for a Fourier transform in the frequency domain?

- **Cepstrum**: the patterns found in the spectrum
- **Quefrency**: a measure of time related to the sampling rate in time domain
- ***Lifter***: a filter in the cepstrum (quefrency) domain

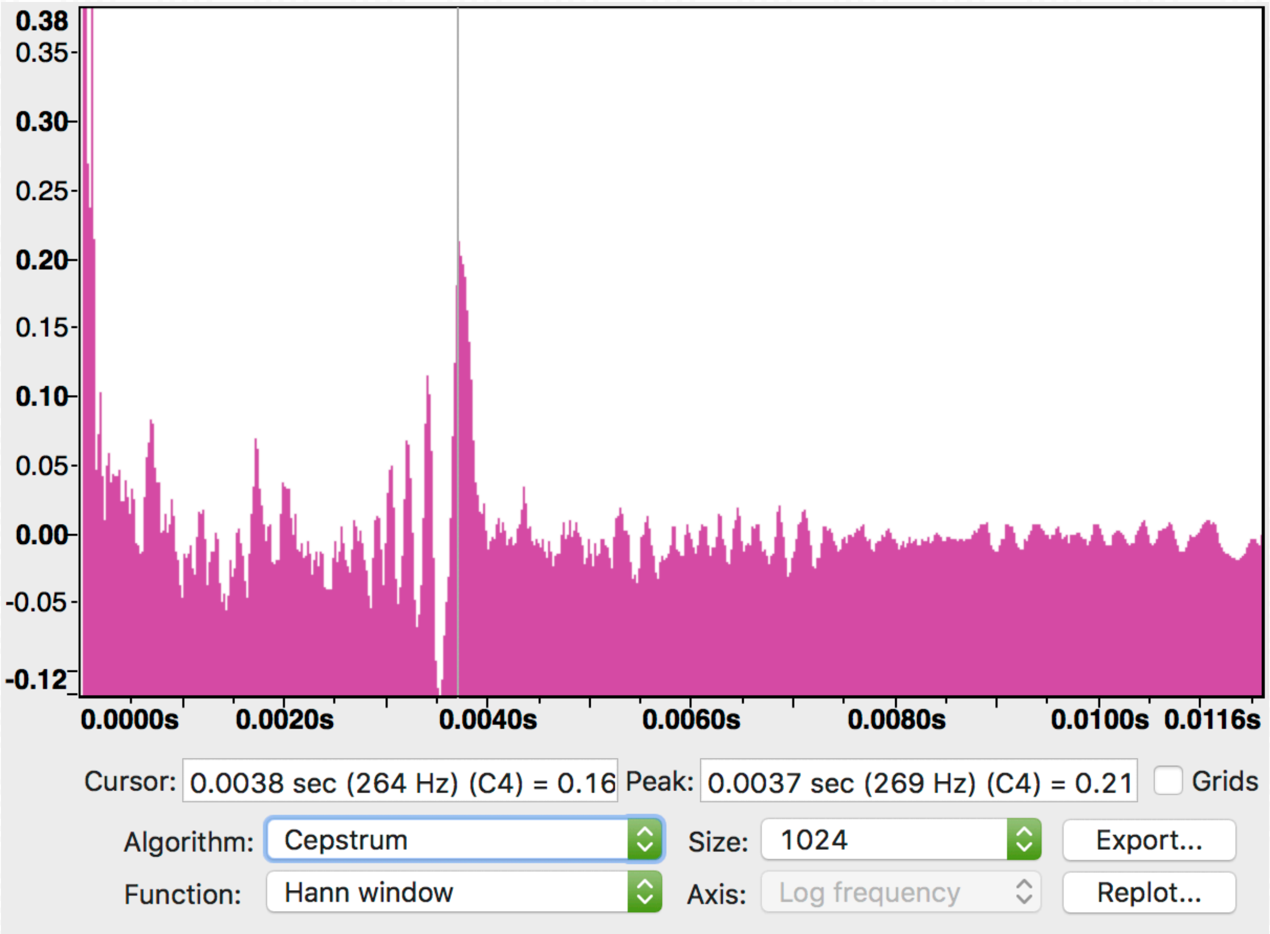


Image from: *https://sethares.engr.wisc.edu/vocoders/phasevocoder.html*

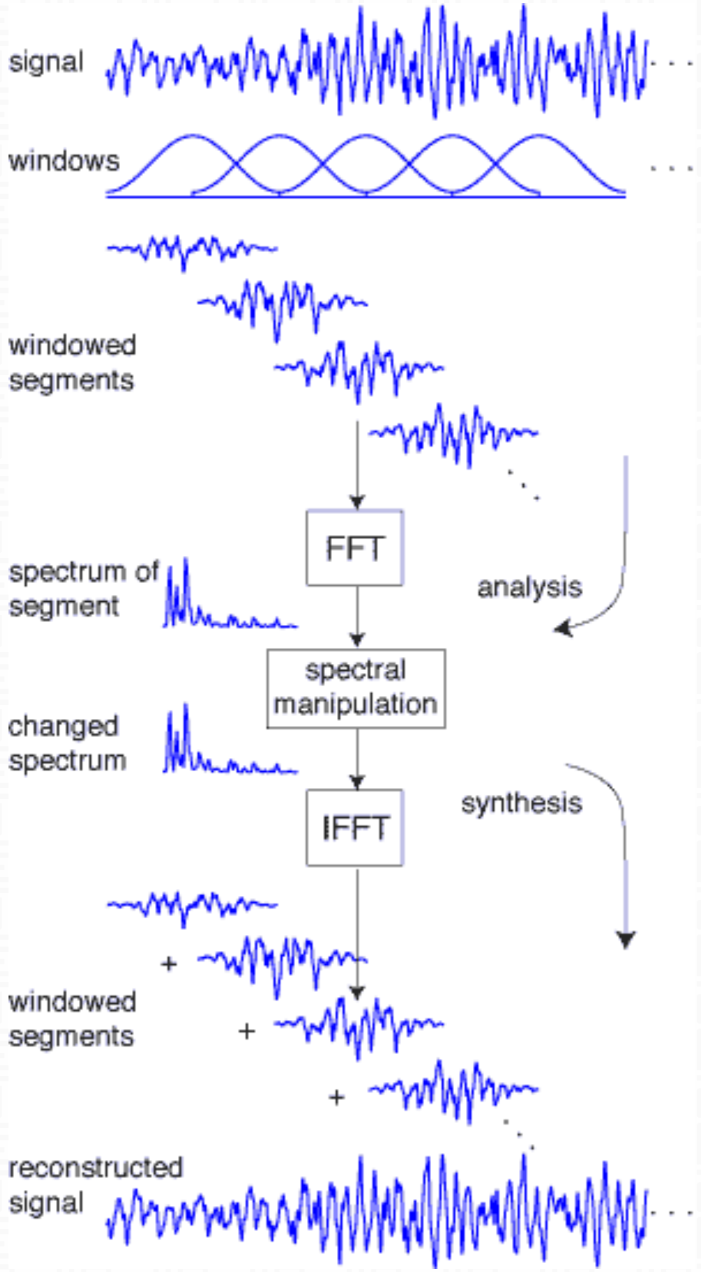
PHASE VOCODER METHOD

A special kind of FFT analysis is the Phase Vocoder method

- Phase information is used to compensate the inadequate frequency resolution
- Mimicking the analog method using "filter banks"
- Possible for spectral edits and resynthesis

Especially good for analysis of harmonic sounds

- Using an appropriate frequency bin size to fit harmonics

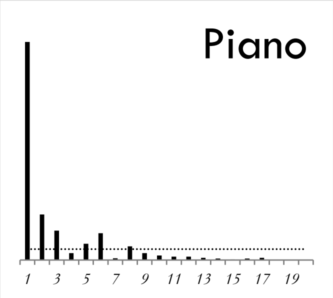


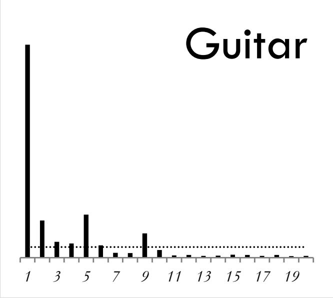
ANALYSIS ON THE HARMONIC SERIES

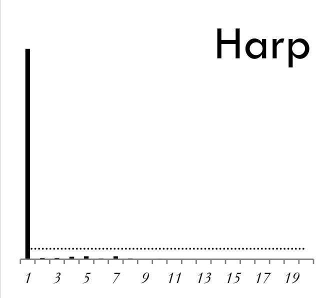
For harmonic sounds (e.g. musical instruments), a series of peaks of integer multiples can be found in the spectrum $f_k = kf_0$

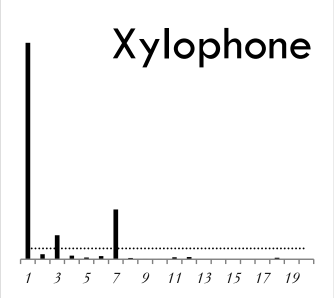
***Timbre**: tone colour*

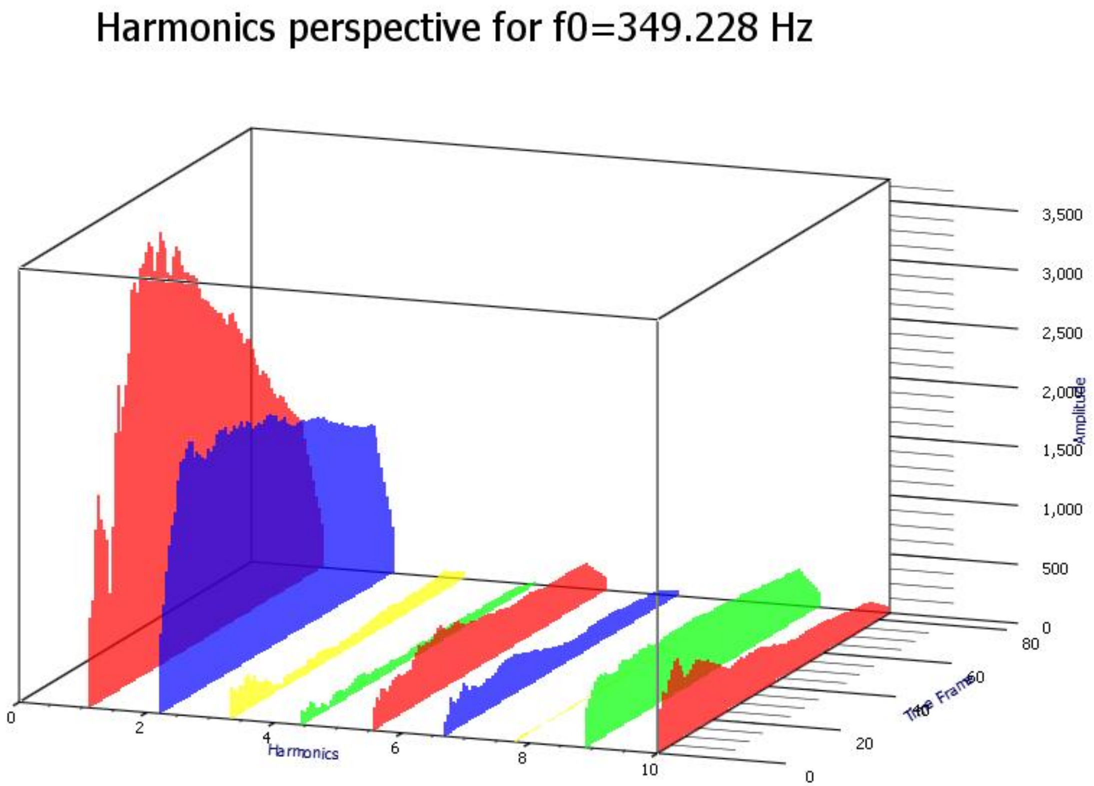
- The difference between musical instruments, or human voice











Integer multiples of f0

ALTERNATIVES TO STFT

STFT has drawbacks such as the resolution constraints of time vs. frequency

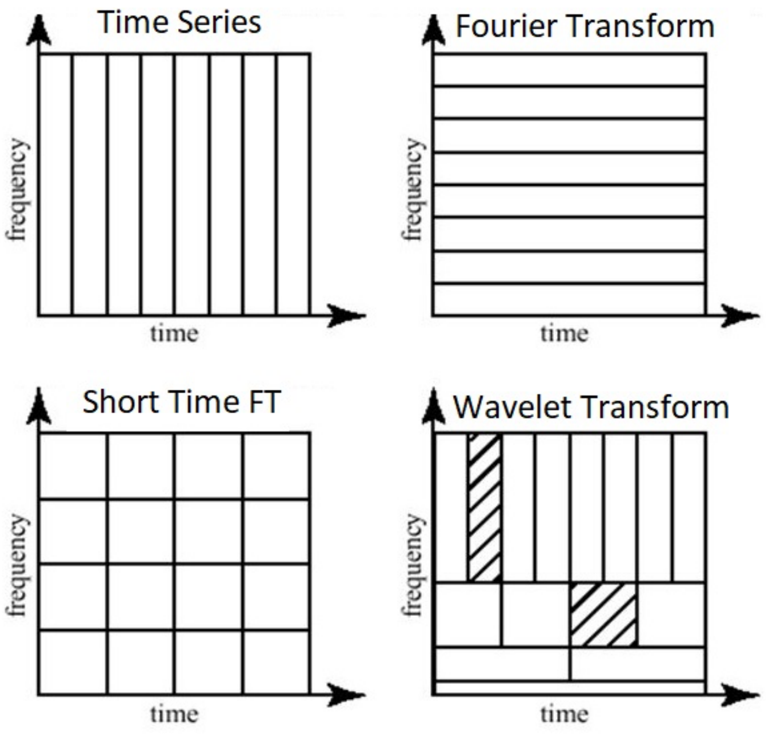
There are alternatives, such as

- Wavelet Transform (WT)
- **Constant-Q** Transform (CQT)

The main aim is to reduce frequency resolution at higher frequencies

- Frequency bins gets larger in the high end

Image from: *http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/*



Time series and various transforms

LECTURE REVIEW

The lecture is half-way done… with these discussed:

- How can sounds be represented mathematically
- The transform between time and frequency domain
  - Continuous vs. Discrete transforms
- Different settings of FFT
- Further possible analysis based on FFT

In the next half of this lecture, we will learn basic MATLAB programming!

READ FURTHER

Chapter 7, “Frequency-Domain Techniques”, Computer Music Instruments

Chapter 2, “Fourier Analysis of Signals”, Fundamentals of Music Processing