Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Introduction Basic Audio Feature Extraction

(with slides by Meinard Müller)

Sound and Music Technology,

December 12, 2019

Today

- Main modules
  A. Sound and music for games
  B. Analysis, classification, and retrieval of sound and music for media
  C. Generation and manipulation of sound and music for games and media

Today

- Main modules
  A. Sound and music for games
  B. Analysis, classification, and retrieval of sound and music for media
     - Representation of musical data: symbolic vs. audio data
  C. Generation and manipulation of sound and music for games and media

Recap

- What is a melody?
  - What are different ways of representing a melody?
- We talked about the Earth Mover Distance. What did we use it for in the context of modeling melody?
- What is a chord?
- What is harmony?
- What is a “musical pattern”? In what context is it used?

Today: Course Module

Module B:

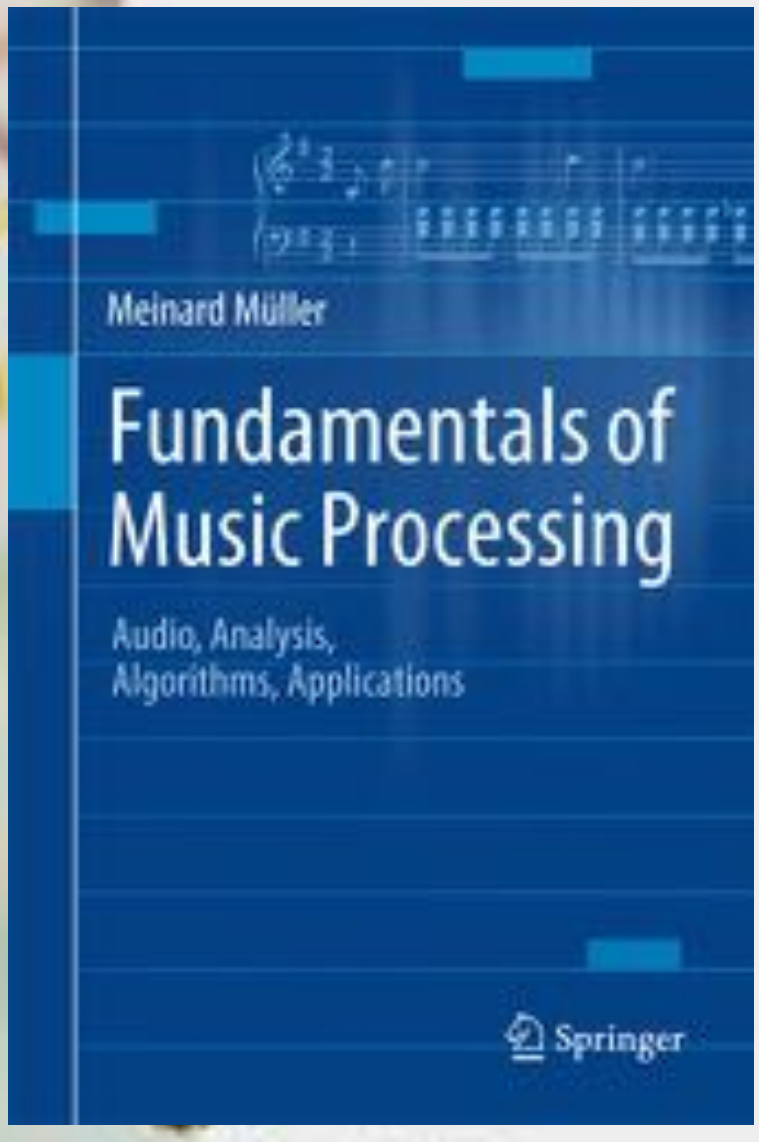
- 1. Introduction to signal processing:
  - How to analyze the content of an audio signal
- 2. Basic audio feature extraction:
  - From digital audio to features
    - Case studies: practical examples
    - Toolboxes, etc.

Introduction

Aim of this lecture:

- Show you the basics of *frequency analysis (Fourier transform)*
- Show you the basics of audio *feature* extraction
- You should be able to understand basic tasks using audio feature extraction toolboxes
- Have some fun with sound!

Book: Fundamentals of Music Processing



Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
483 p., 249 illus., hardcover
ISBN: 978-3-319-21944-8
Springer, 2015

Accompanying website:
www.music-processing.de

Recently released Jupyter notebooks:

https://www.audiolabs-erlangen.de/fau/professor/mueller/notebookFMP

What are *Features*?

- If we want to compare/find/classify things, we need a way to describe them in a **useful** way
- Visual: **colors**, **shapes**, etc
- What is the analogy for sound?

What are *Features*?

- Audio: **frequencies**, **amplitudes:**
  - Low/high sounds: ex. Tuba vs Flute instrument
  - Loud/quiet sounds: ex. Metal vs Acoustic music
- (A manipulation of) these elements is called a *feature*.
- How do we obtain this information?
  - **Fourier analysis**

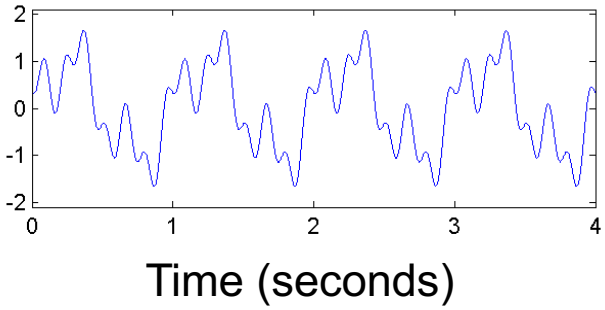
Recap: What are *Features*?

- Important feature **properties**:
  - ***Abstraction***:
    - Low: e.g. Frequencies
      - low/high pitched music instruments
    - Mid: e.g. Chords
      - harmonic differences between composers
    - High: e.g. Emotion
      - sad/happy music
  - ***Locality***:
    - *Global feature*: entire signal
      - e.g. genre
    - *Local feature*: part of a signal
      - e.g. pitch

Fourier Transform

Idea: Decompose a given signal into a superposition of sinusoids (elementary signals).

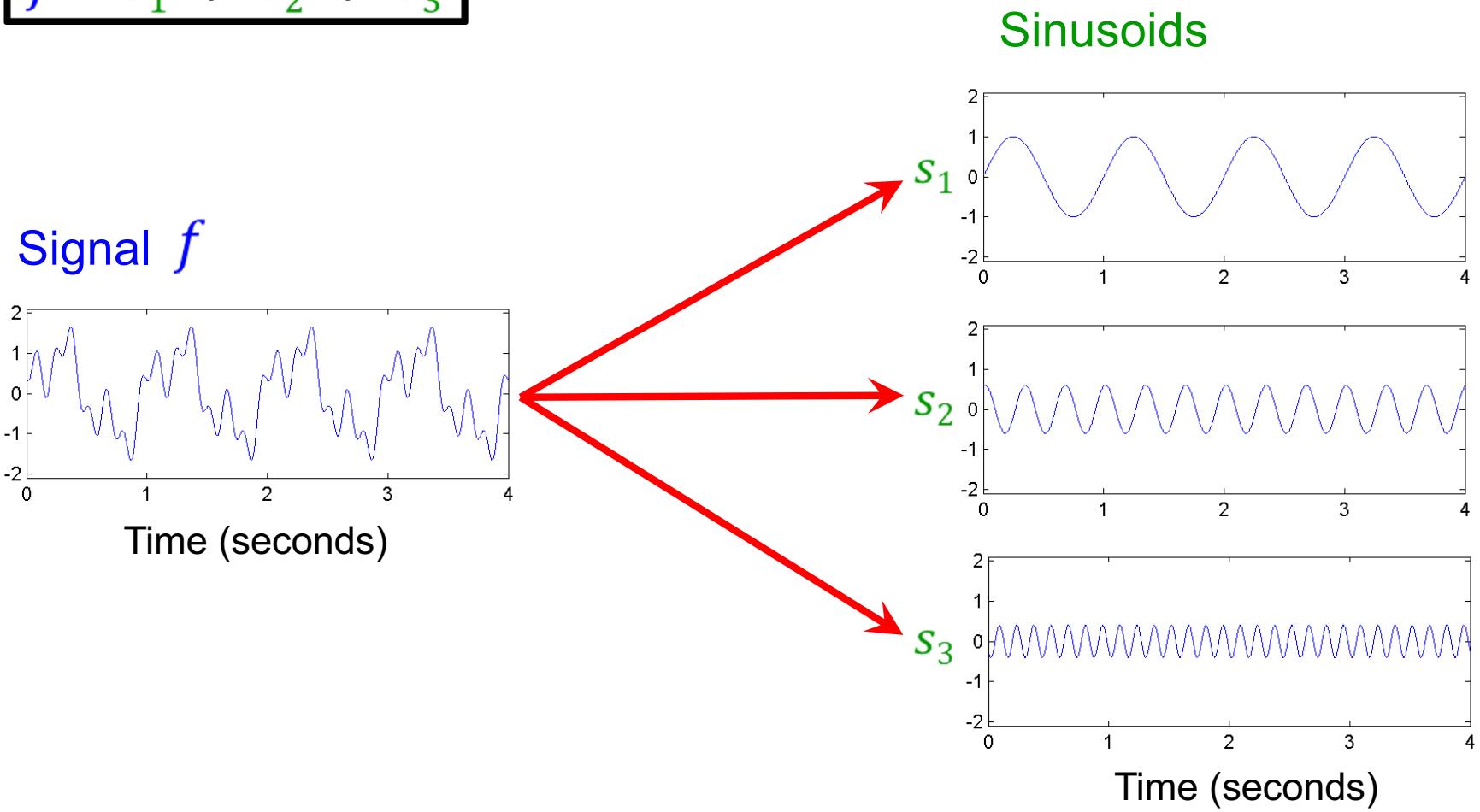
Signal f



Fourier Transform

Idea: Decompose a given signal into a superposition of sinusoids (elementary signals).

$$f = s_1 + s_2 + s_3$$



Fourier Transform

Each sinusoid has a physical meaning
and can be described by three parameters:

$$s_{(A,\omega,\varphi)}(t) = A \cdot \sin(2\pi(\omega t - \varphi))$$

ω = frequency
A = amplitue
φ = phase

Interpretation:

The amplitude A reflects the intensity at which the sinusoidal of frequency ω appears in f.

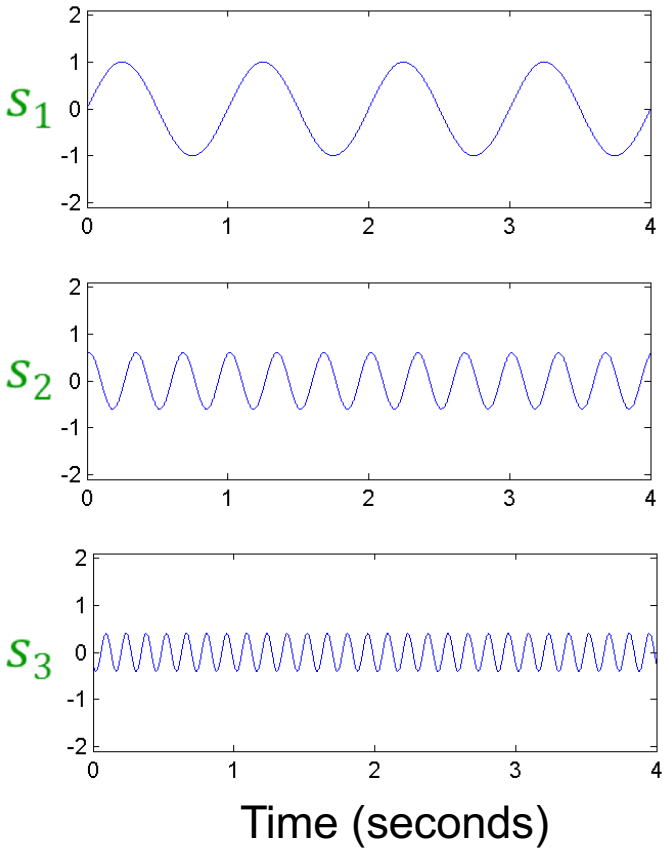
The phase φ reflects how the sinusoidal has to be shifted to best correlate with f.

$A_1 = 1$
$\omega_1 = 1$
$\varphi_1 = 0$

$A_2 = 0.6$
$\omega_2 = 3$
$\varphi_2 = -0.2$

$A_3 = 0.4$
$\omega_3 = 7$
$\varphi_3 = 0.4$

Sinusoids

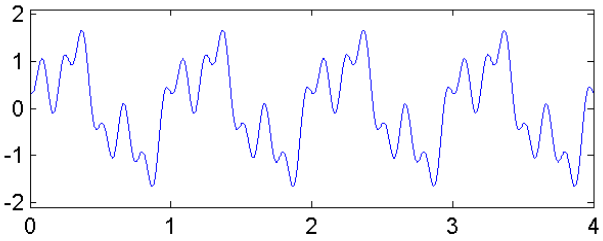


Fourier Transform

Each sinusoid has a physical meaning
and can be described by three parameters:

$$f = s_1 + s_2 + s_3$$

Signal f



Time (seconds)

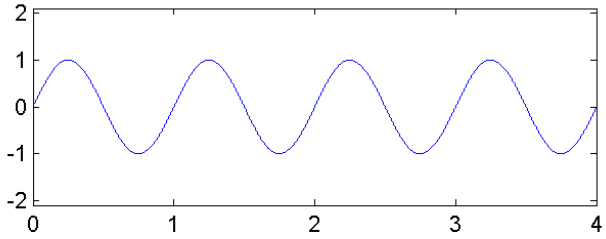
$A_1 = 1$
$\omega_1 = 1$
$\varphi_1 = 0$

$A_2 = 0.6$
$\omega_2 = 3$
$\varphi_2 = -0.2$

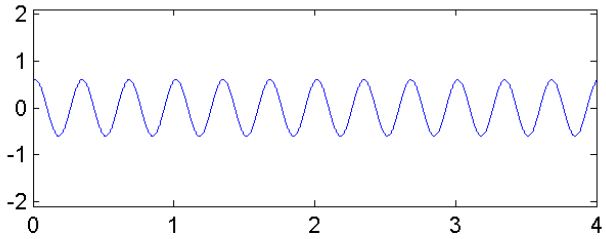
$A_3 = 0.4$
$\omega_3 = 7$
$\varphi_3 = 0.4$

Sinusoids

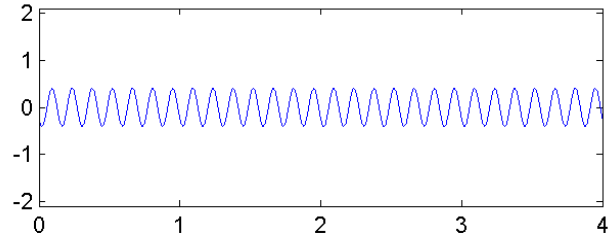
s_1



s_2



s_3



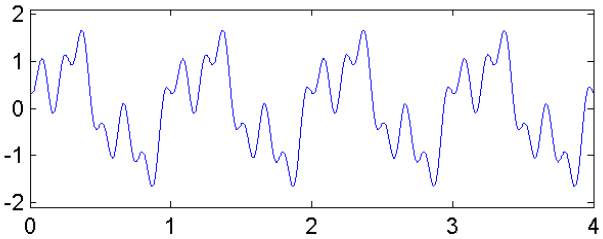
Time (seconds)

Fourier Transform

Each sinusoid has a physical meaning
and can be described by three parameters:

$$f = s_1 + s_2 + s_3$$

Signal f

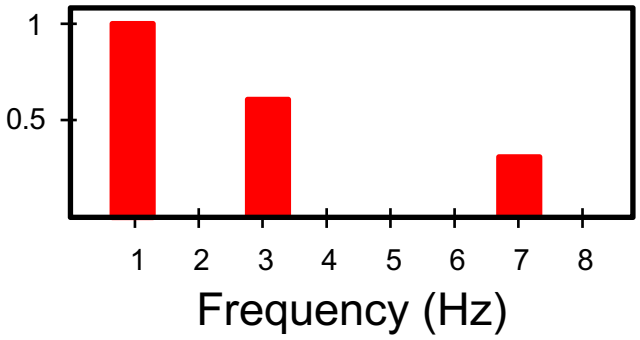


$A_1 = 1$
$\omega_1 = 1$
$\varphi_1 = 0$

$A_2 = 0.6$
$\omega_2 = 3$
$\varphi_2 = -0.2$

$A_3 = 0.4$
$\omega_3 = 7$
$\varphi_3 = 0.4$

Fourier transform $|\hat{f}|$



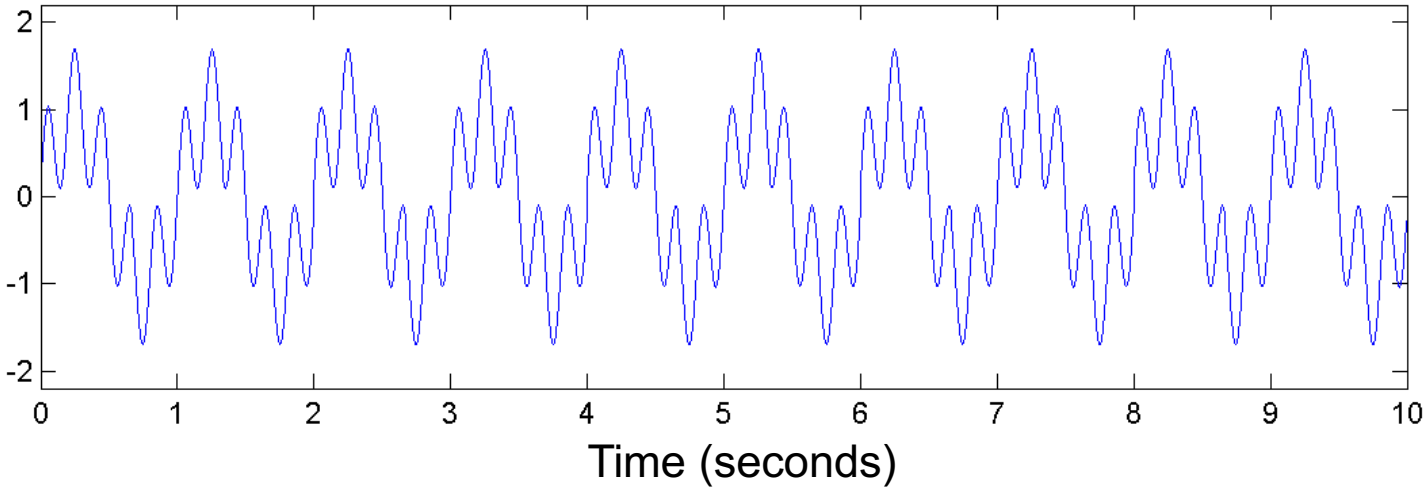
Fourier Transform

https://youtu.be/c5fotAbiAhI

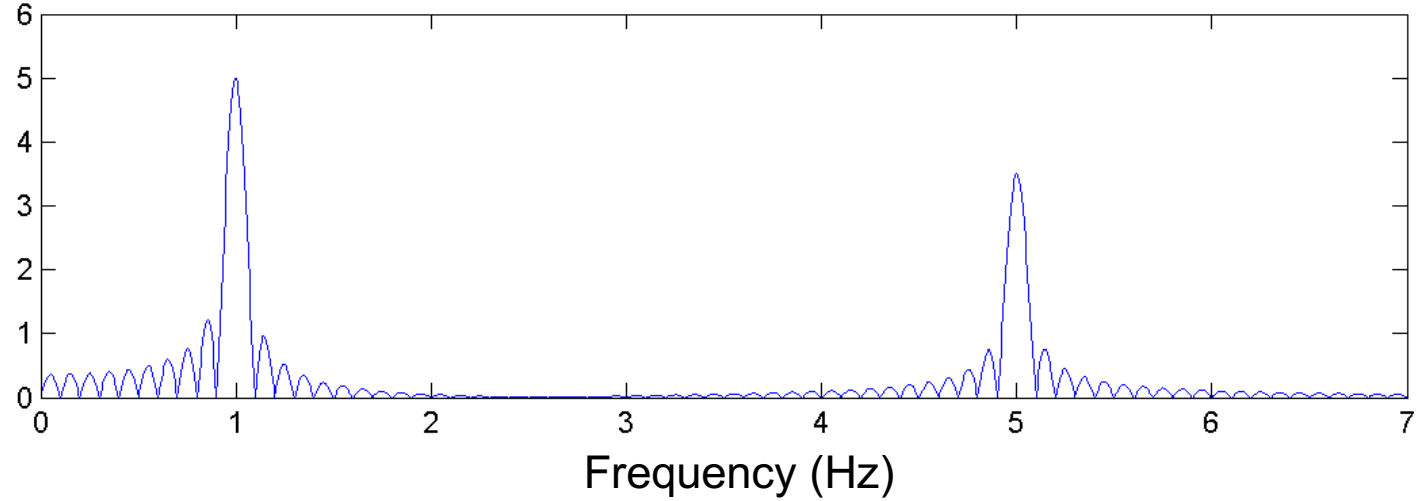
Fourier Transform

Example: Superposition of two sinusoids

f



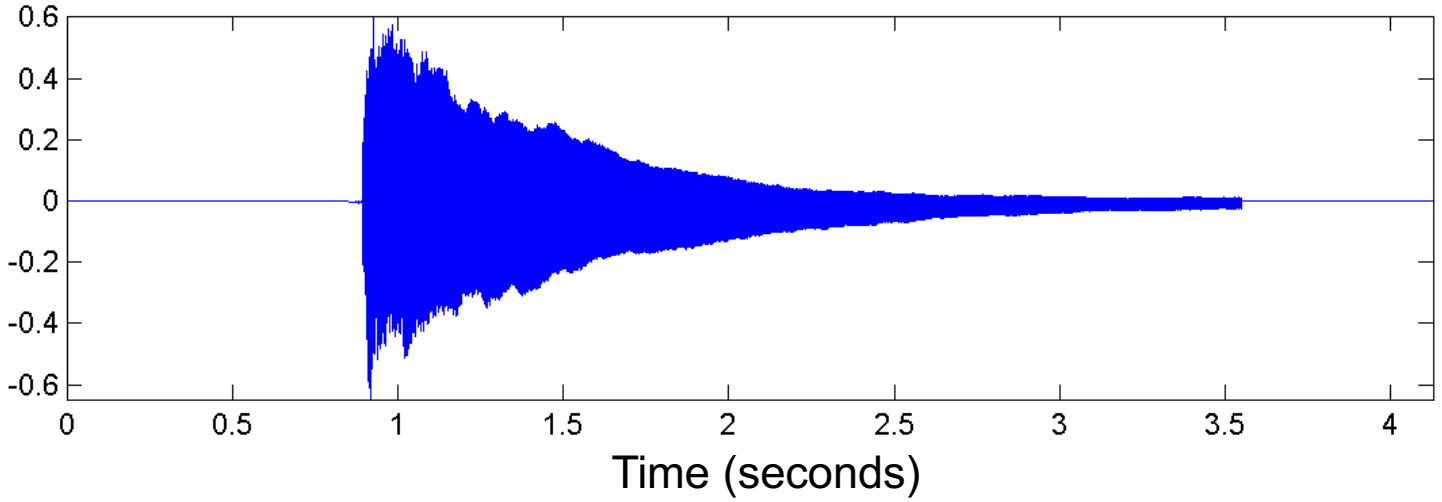
$|\hat{f}|$



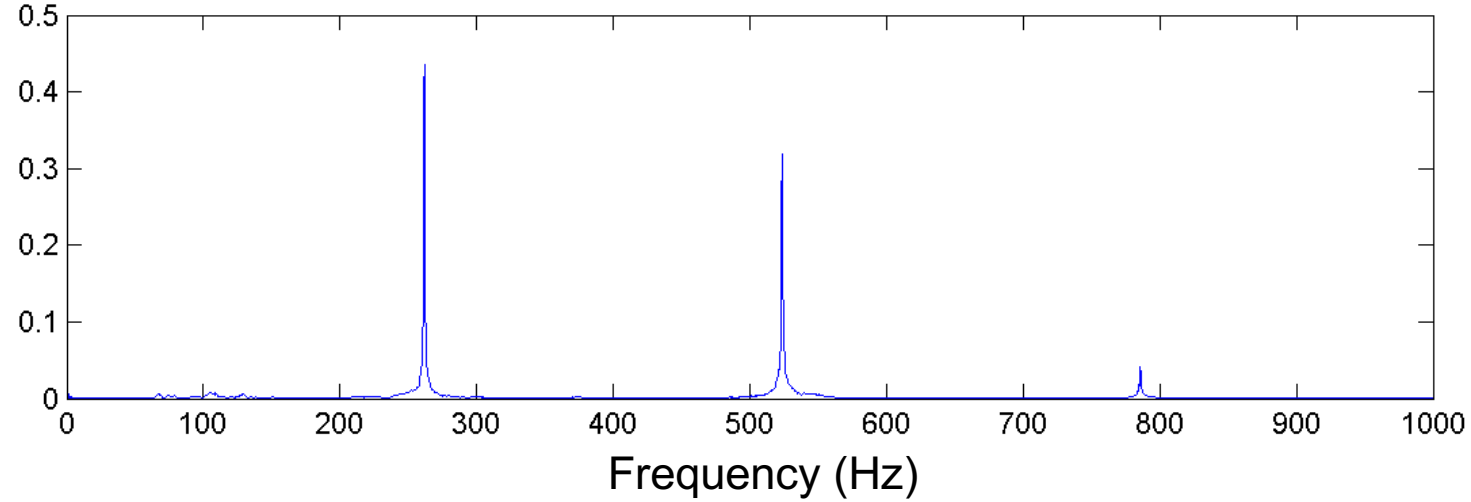
Fourier Transform

Example: C4 played by piano

f



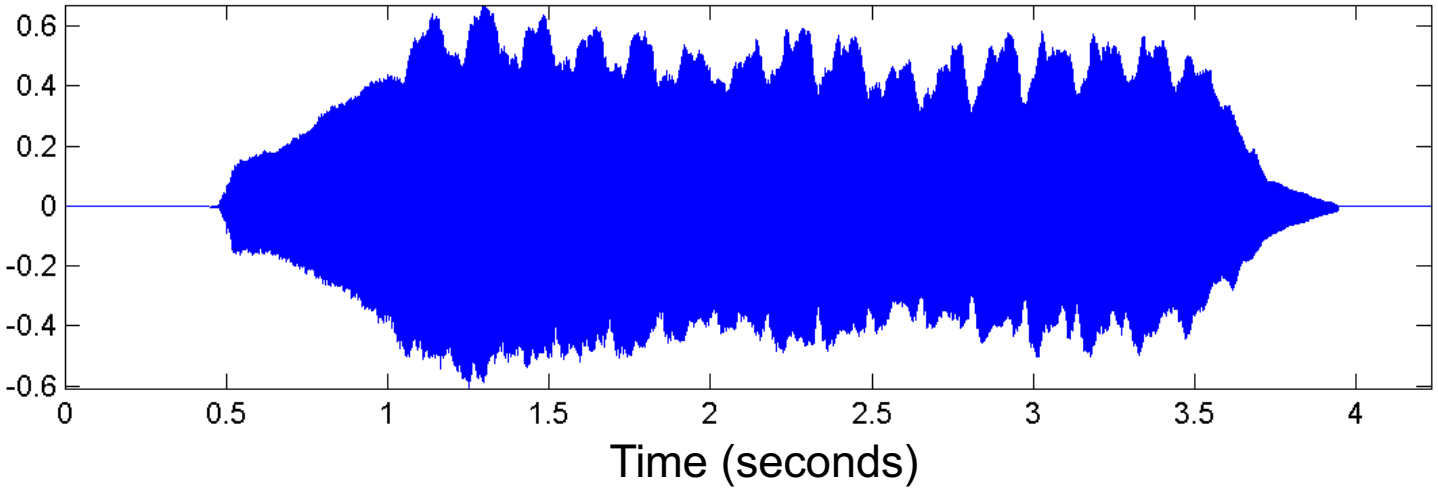
$|\hat{f}|$



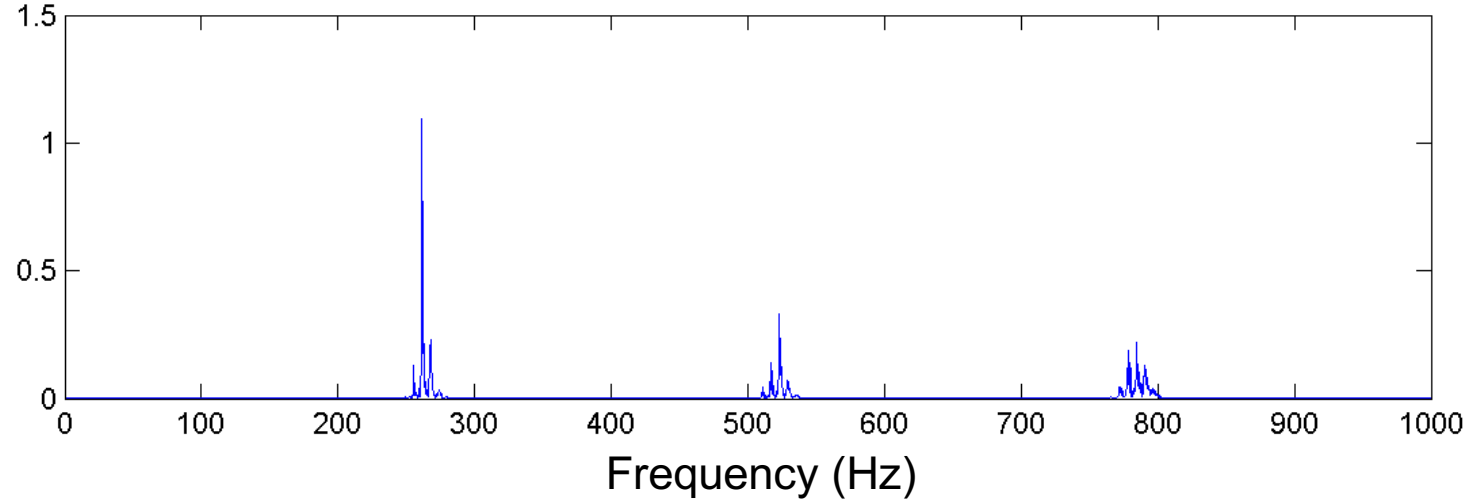
Fourier Transform

Example: C4 played by violin

f



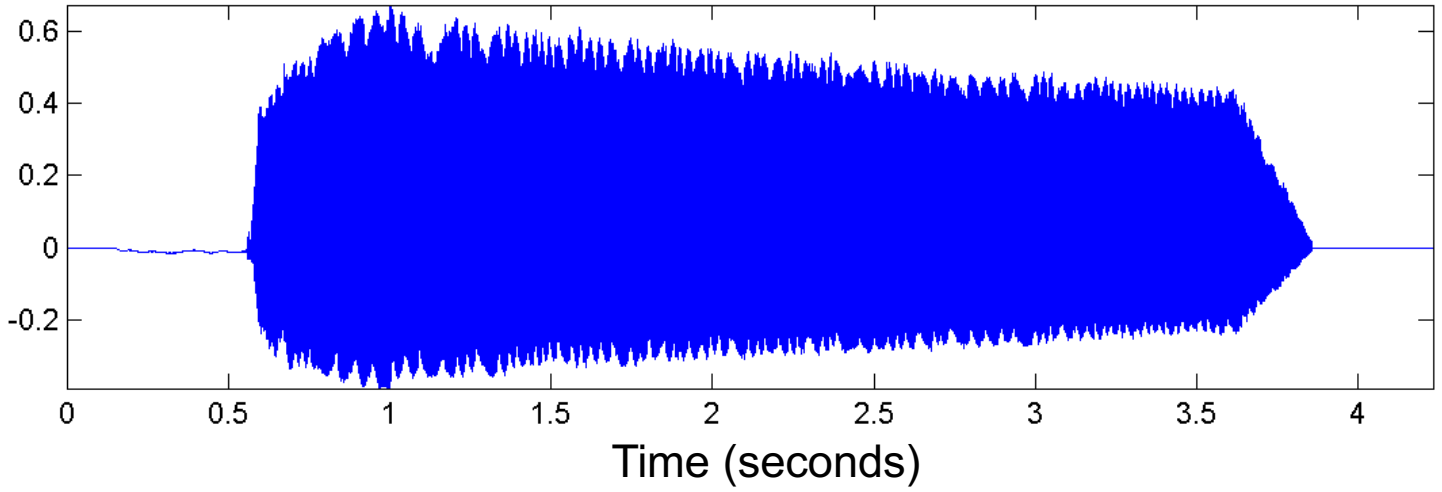
$|\hat{f}|$



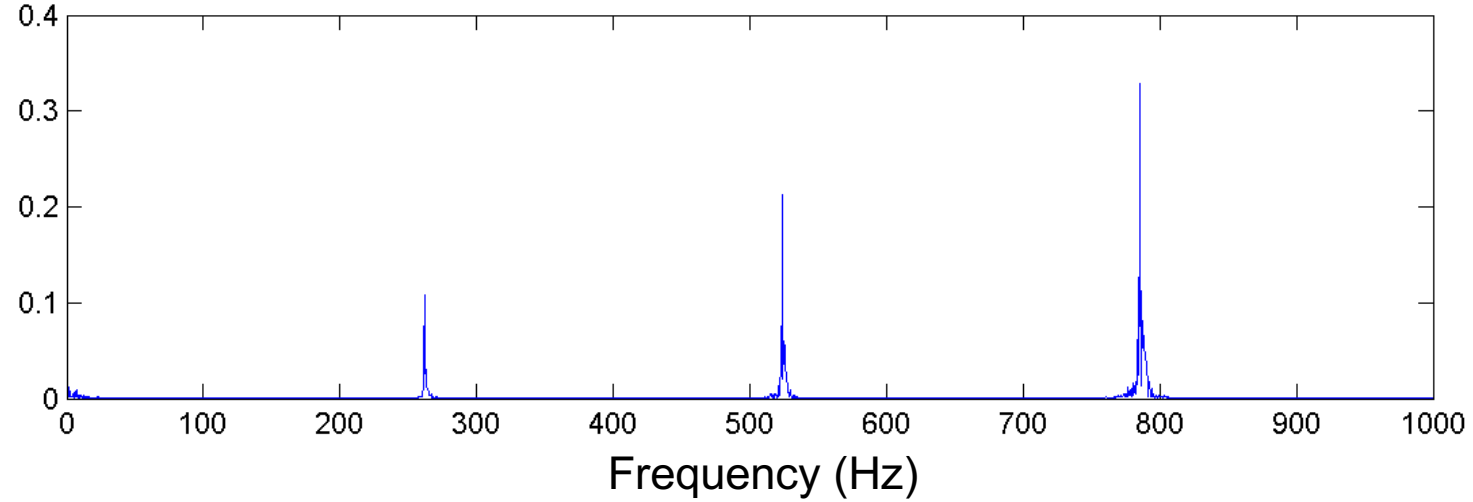
Fourier Transform

Example: C4 played by trumpet

f



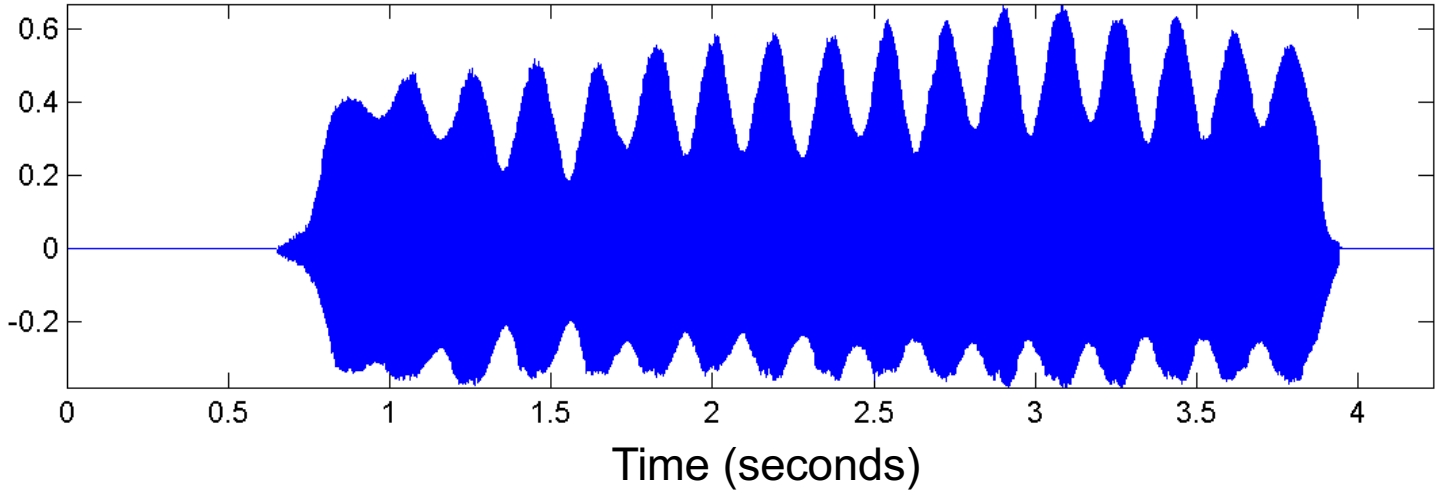
$|\hat{f}|$



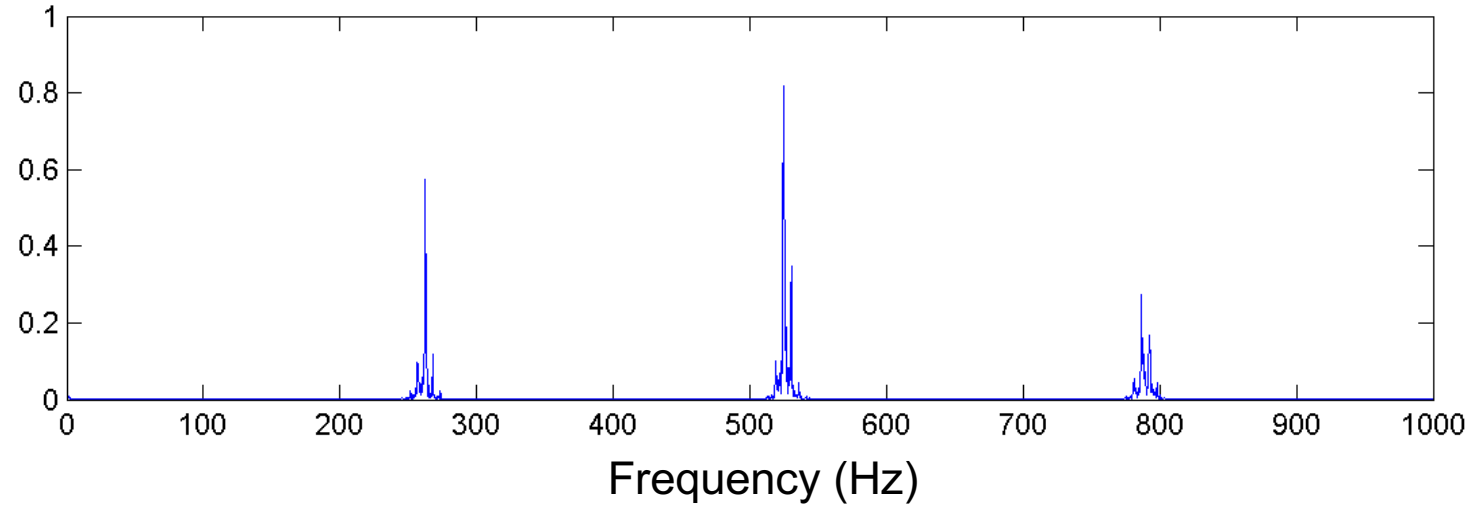
Fourier Transform

Example: C4 played by flute

f

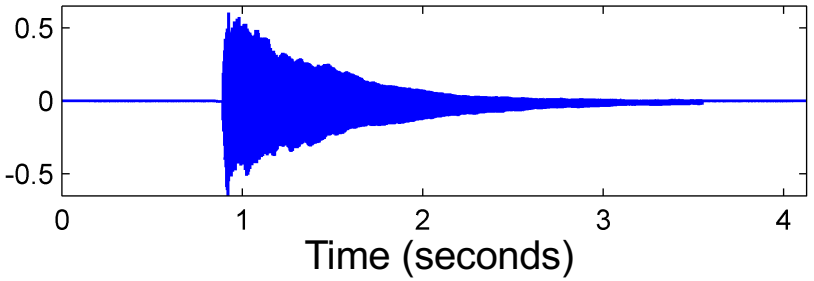


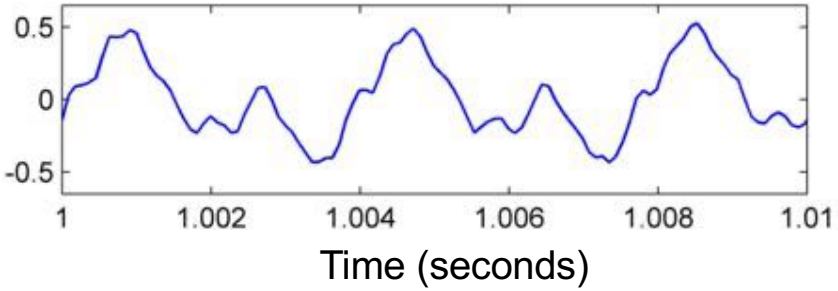
$|\hat{f}|$

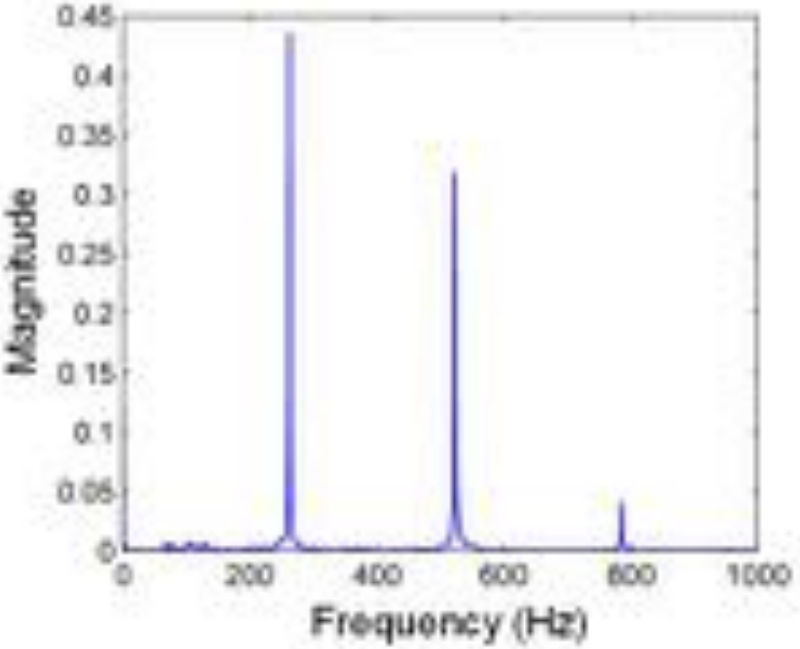


Fourier Transform

Example: Piano tone (C4, 261.6 Hz)

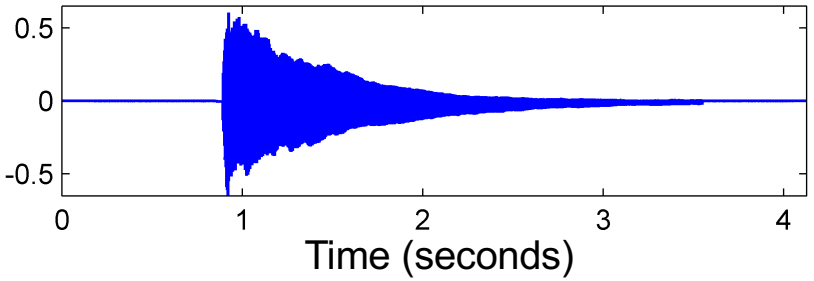


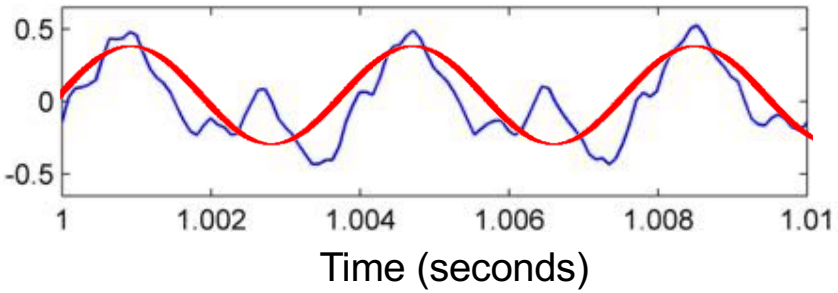


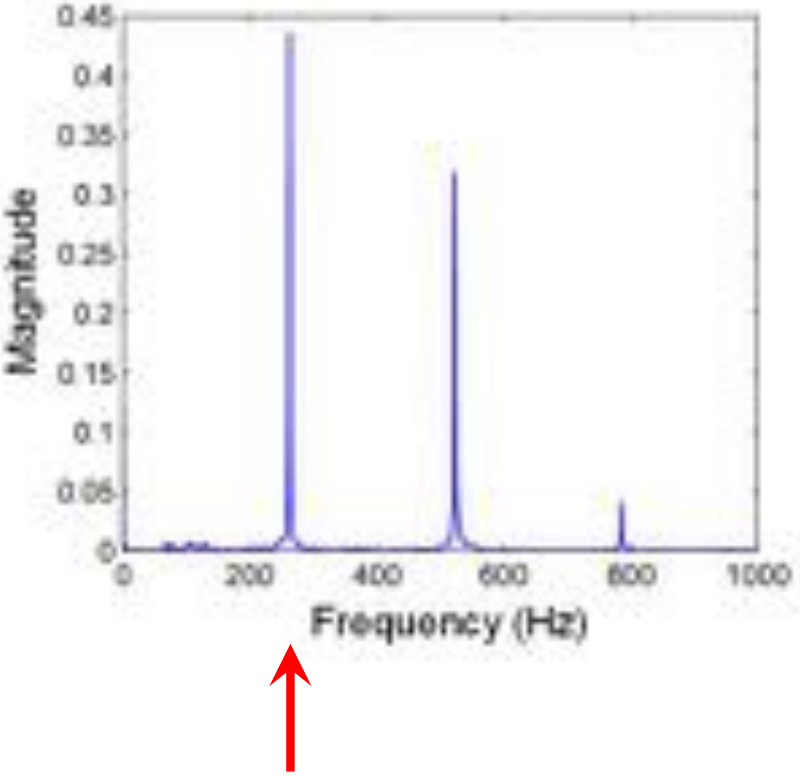


Fourier Transform

Example: Piano tone (C4, 261.6 Hz)







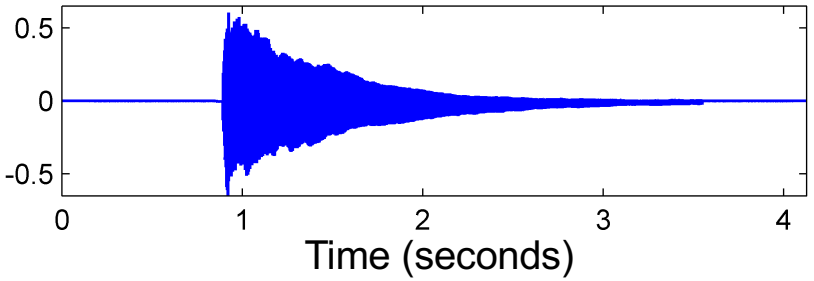
Analysis using sinusoid with **262 Hz**

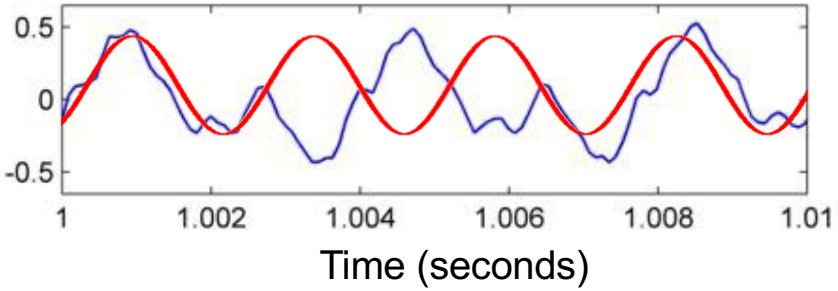
→ high correlation

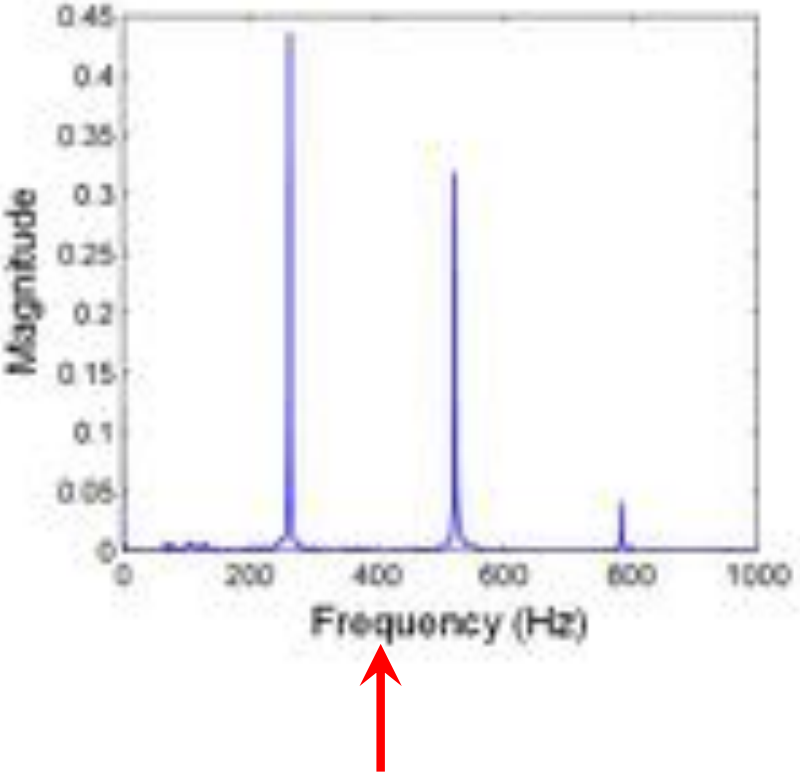
→ large Fourier coefficient

Fourier Transform

Example: Piano tone (C4, 261.6 Hz)







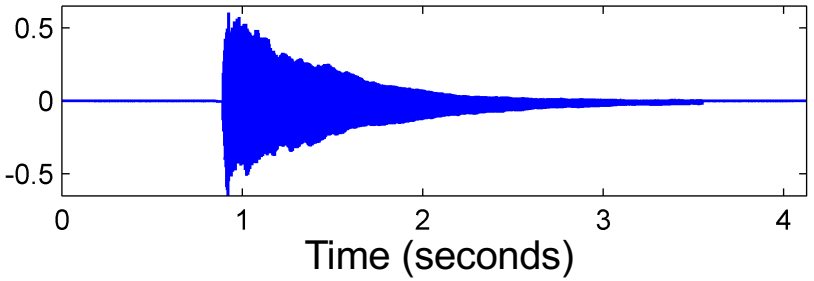
Analysis using sinusoid with **400 Hz**

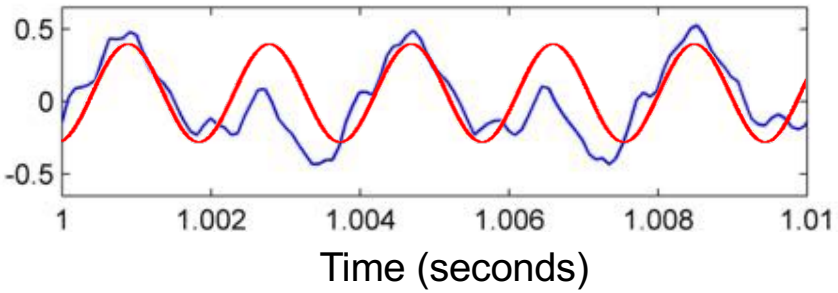
→ low correlation

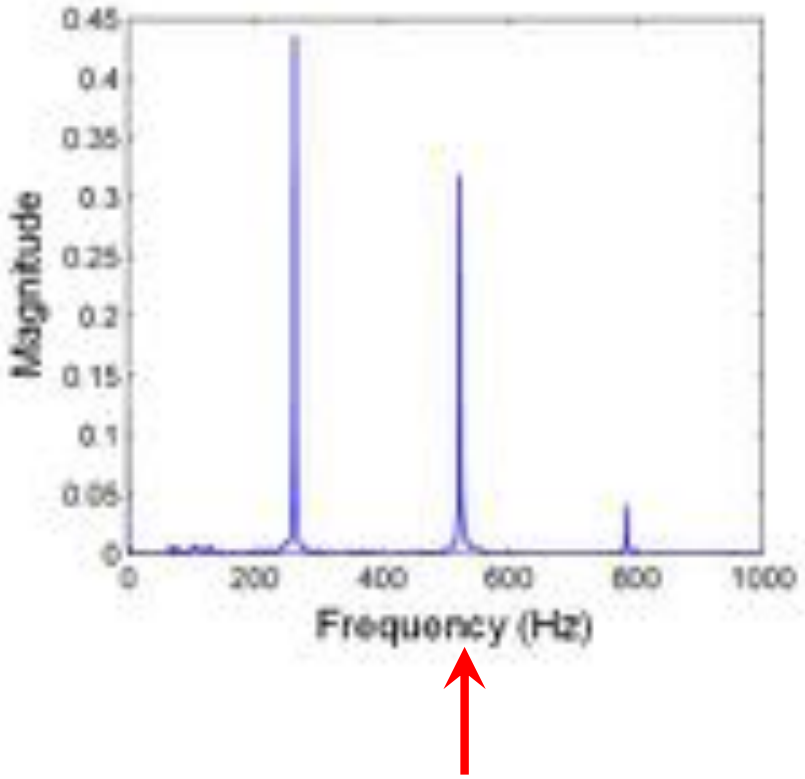
→ small Fourier coefficient

Fourier Transform

Example: Piano tone (C4, 261.6 Hz)







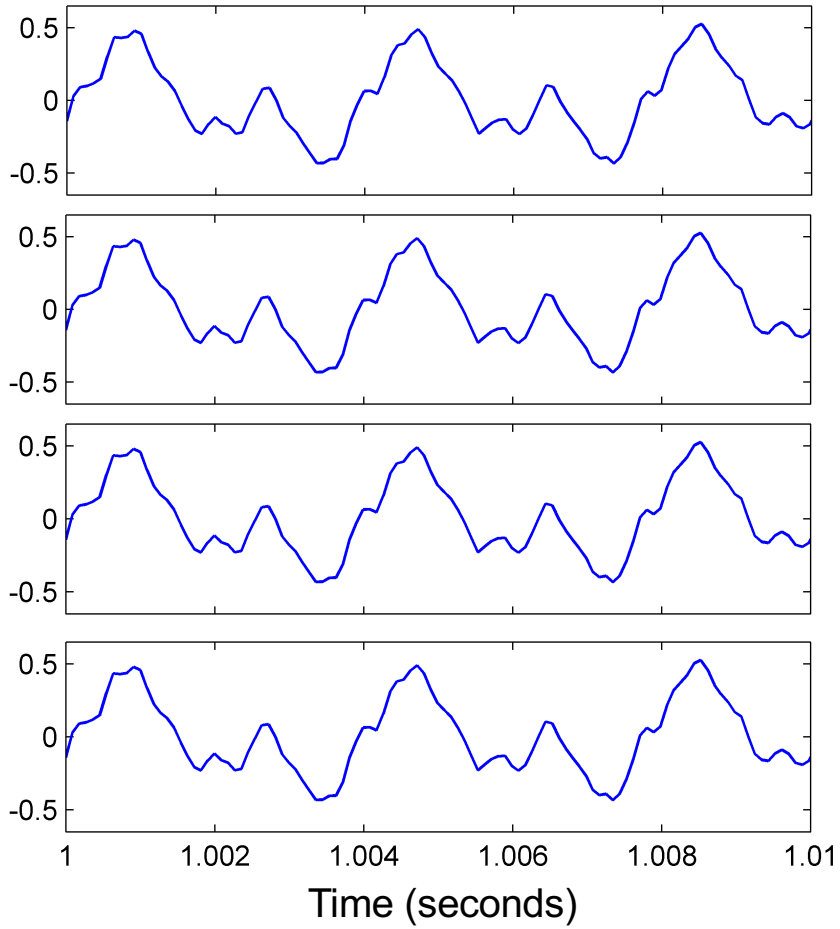
Analysis using sinusoid with **523 Hz**

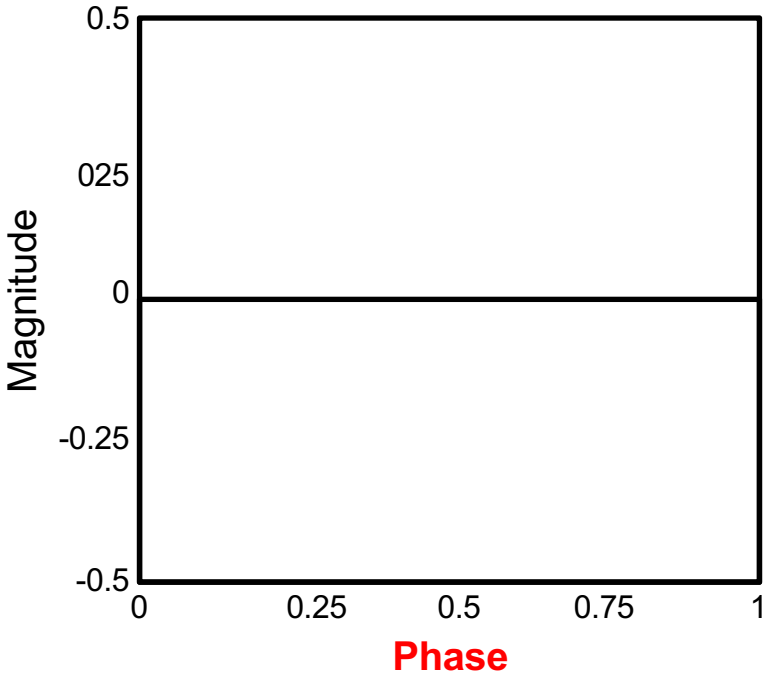
→ high correlation

→ large Fourier coefficient

Fourier Transform

Role of phase

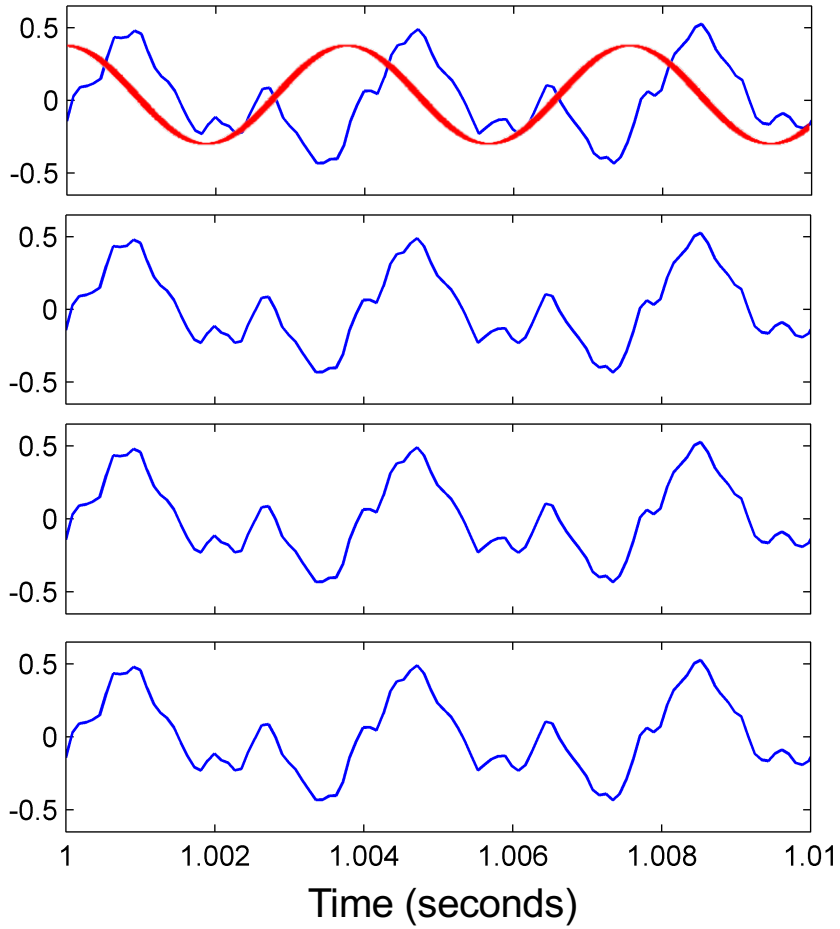


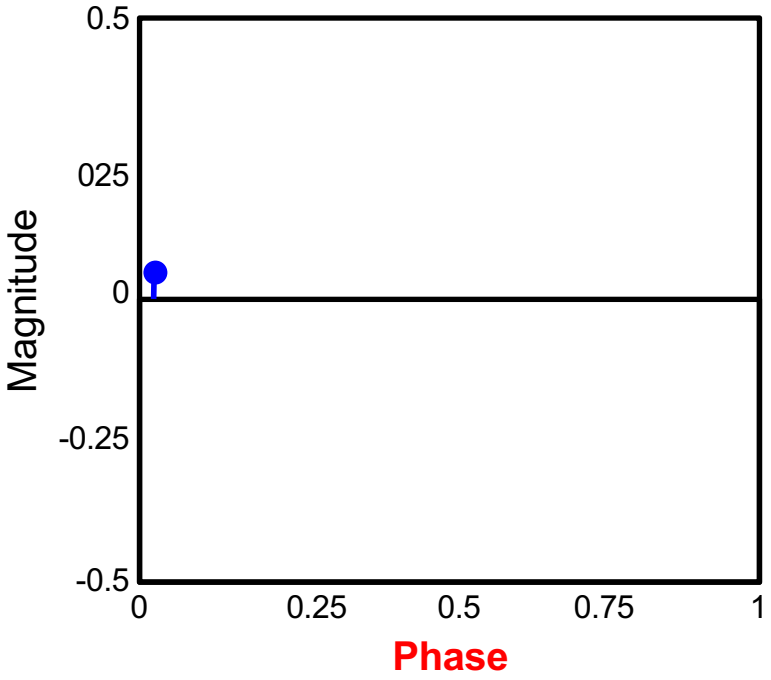


Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\varphi = 0.05$

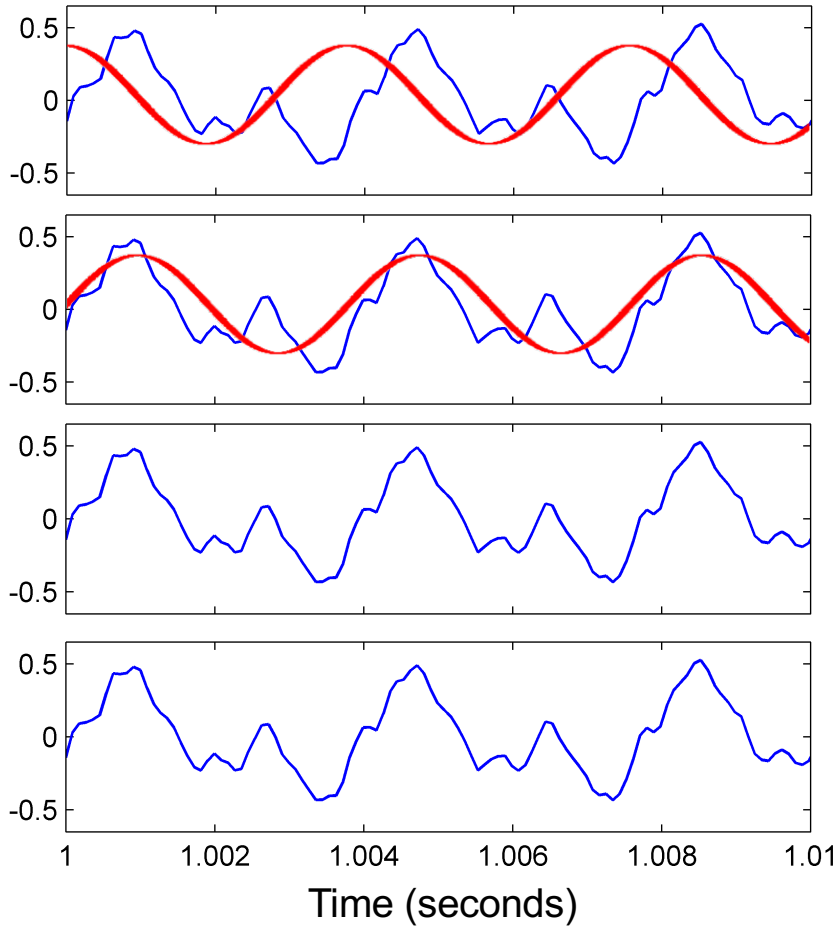


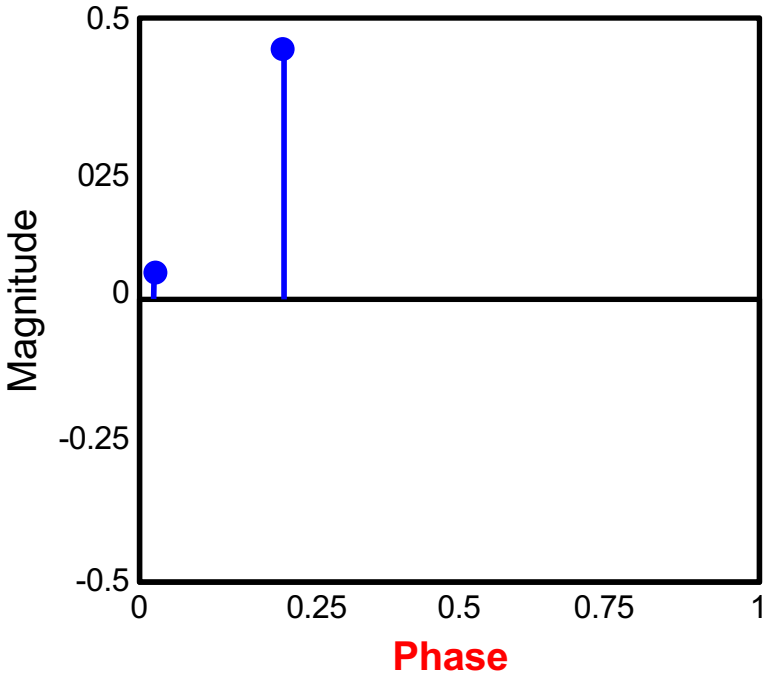


Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\varphi = 0.24$

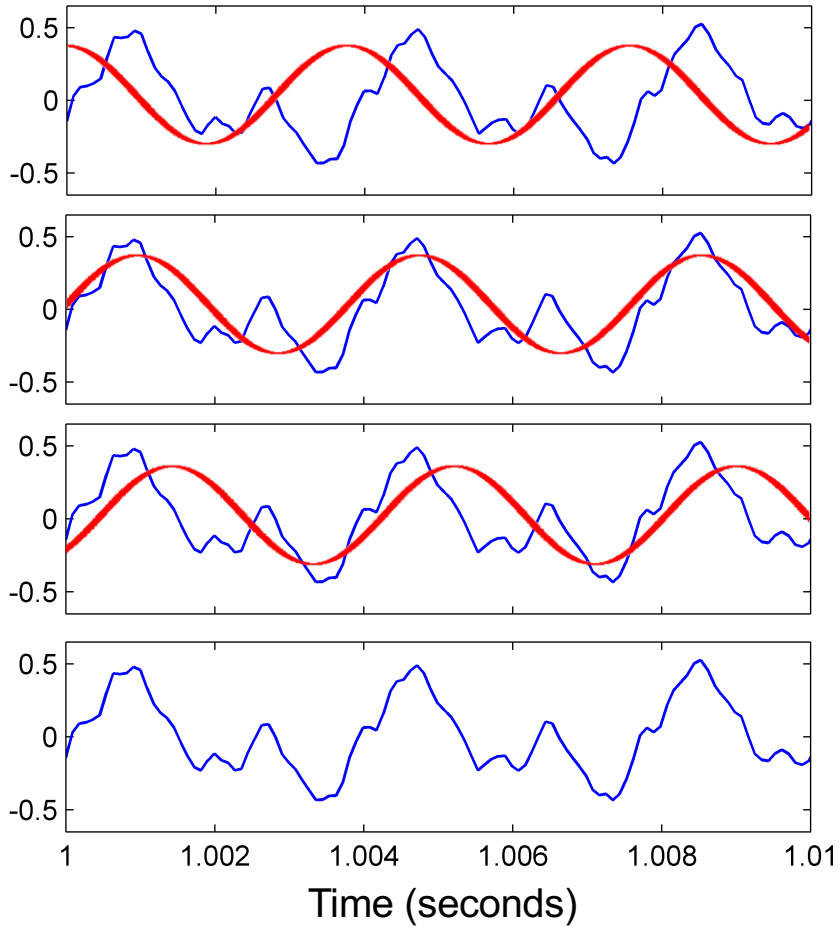


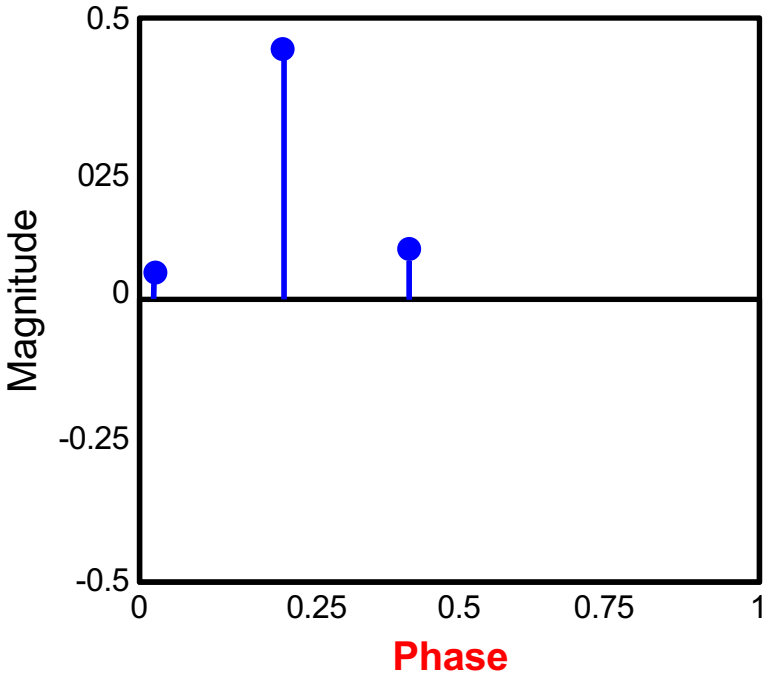


Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\varphi = 0.45$

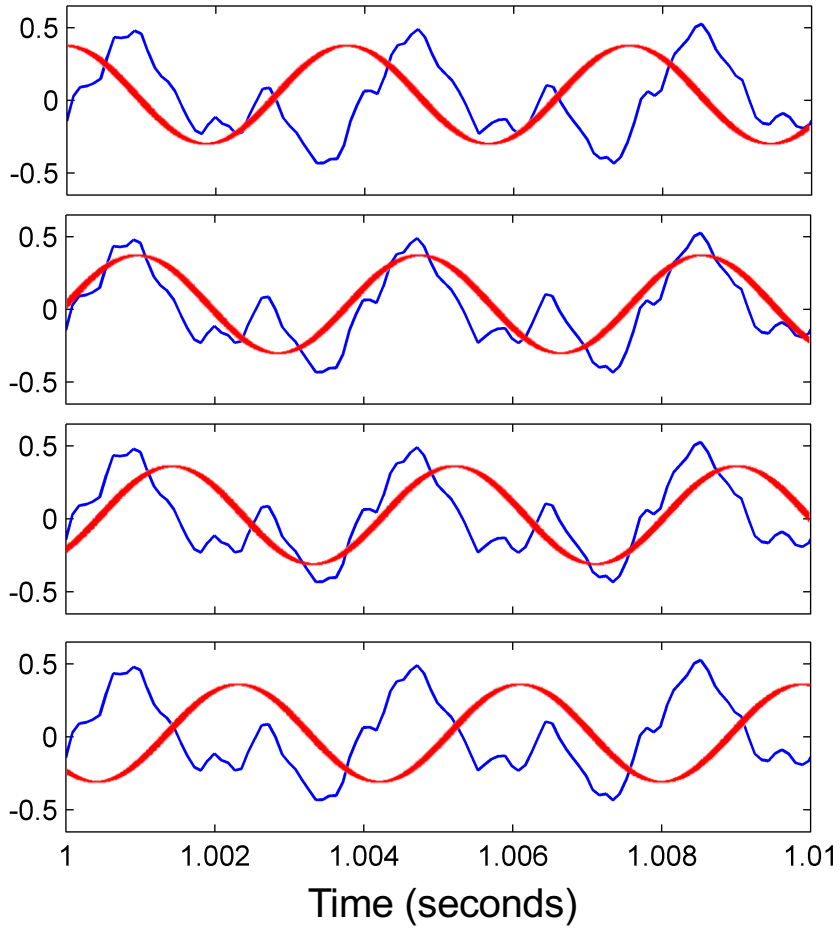


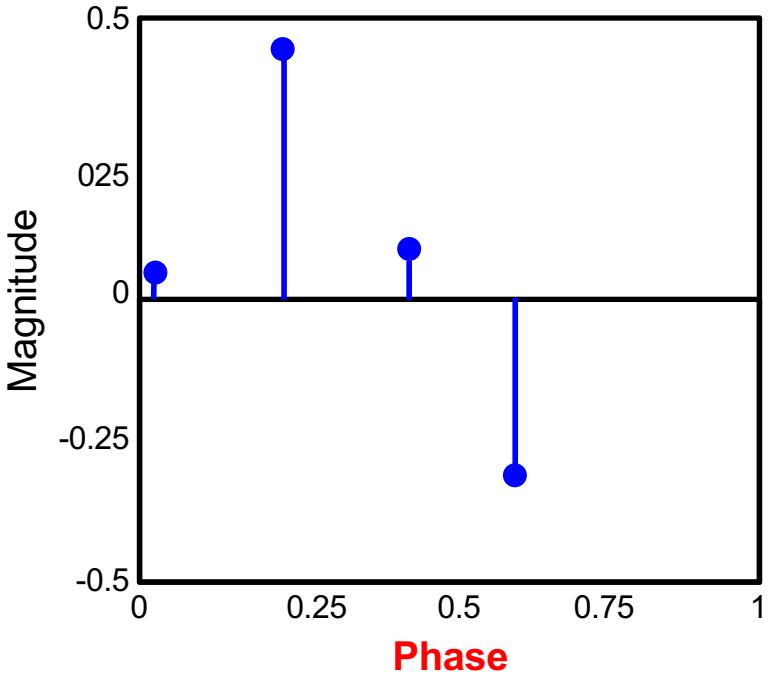


Fourier Transform

Role of phase

Analysis with sinusoid having frequency 262 Hz and phase $\varphi = 0.6$





Fourier Transform

Signal $f: \mathbb{R} \to \mathbb{R}$

Fourier representation $f(t) = \int_{\omega \in \mathbb{R}} c_\omega \exp(2\pi i \omega t) d\omega$

Fourier transform $c_\omega = \hat{f}(\omega) = \int_{t \in \mathbb{R}} f(t) \exp(-2\pi i \omega t) dt$

Animation: https://bl.ocks.org/jinroh/7524988

Fourier Transform

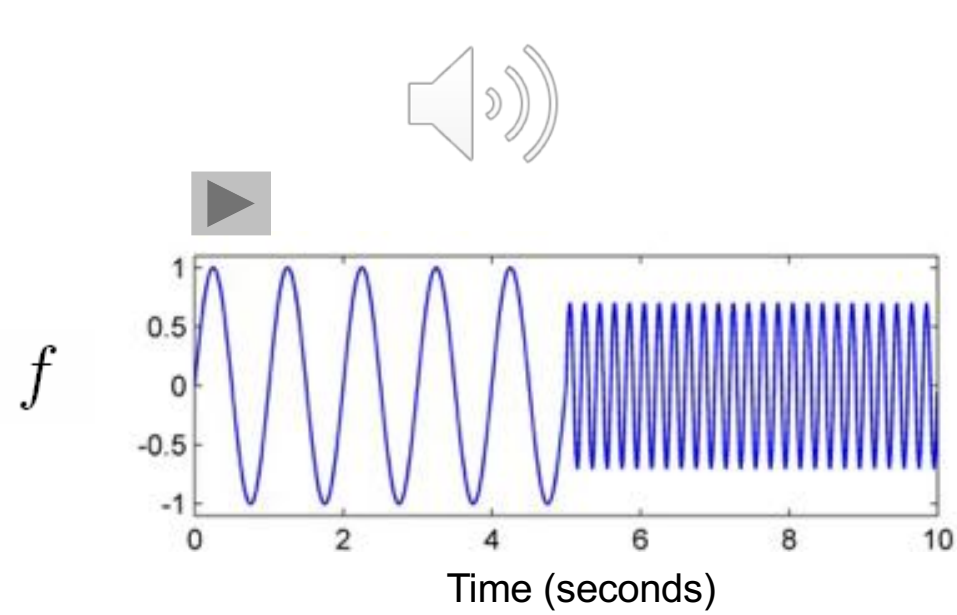
Signal $f: \mathbb{R} \to \mathbb{R}$

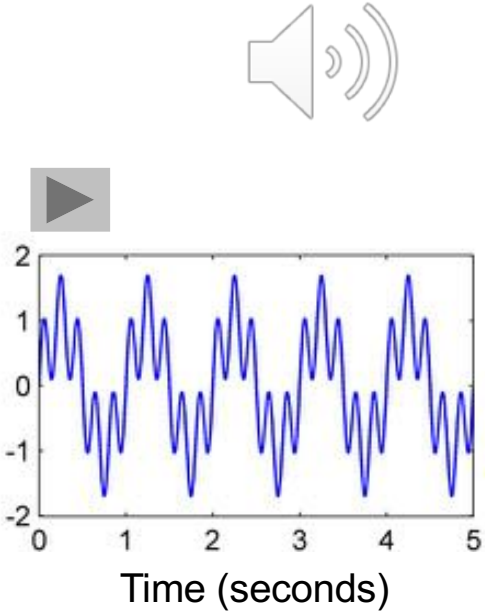
Fourier representation $f(t) = \int_{\omega \in \mathbb{R}} c_\omega \exp(2\pi i \omega t) d\omega$

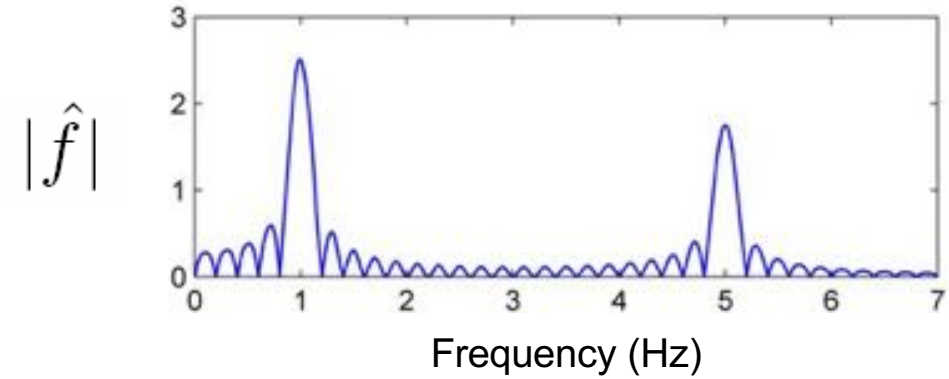
Fourier transform $c_\omega = \hat{f}(\omega) = \int_{t \in \mathbb{R}} f(t) \exp(-2\pi i \omega t) dt$

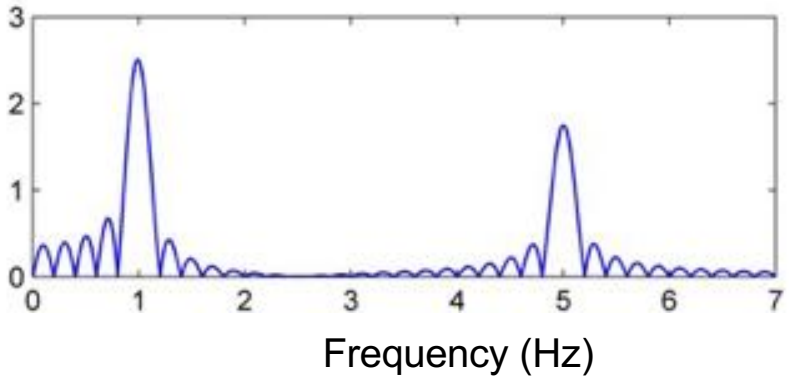
- Tells which frequencies occur, but does not tell when the frequencies occur.
- Frequency information is averaged over the entire time interval.
- Time information is hidden in the phase

Fourier Transform









Short Time Fourier Transform

Idea (Dennis Gabor, 1946):

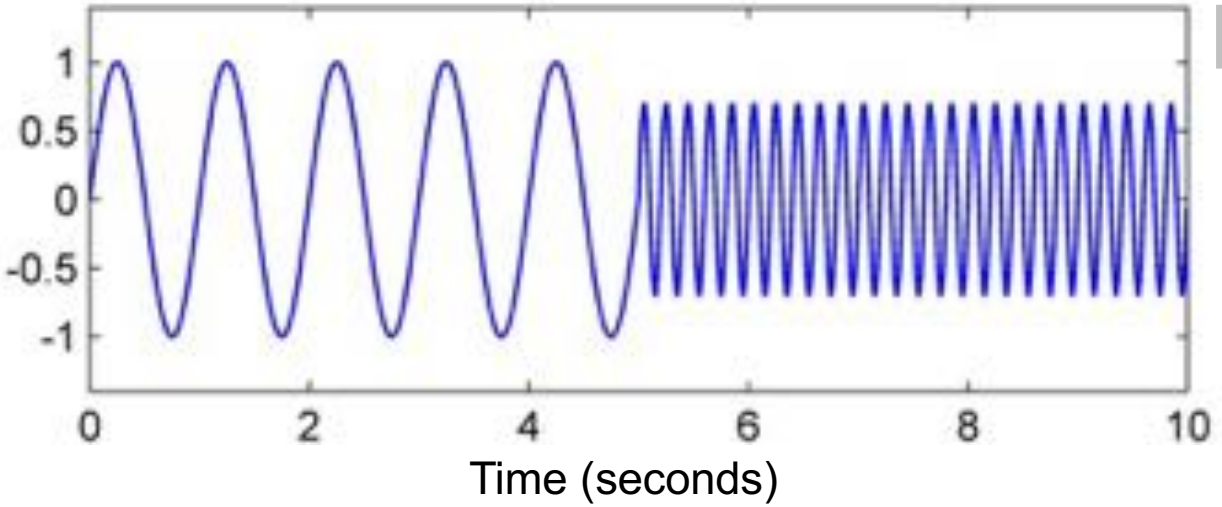
- Consider only a small section of the signal for the spectral analysis

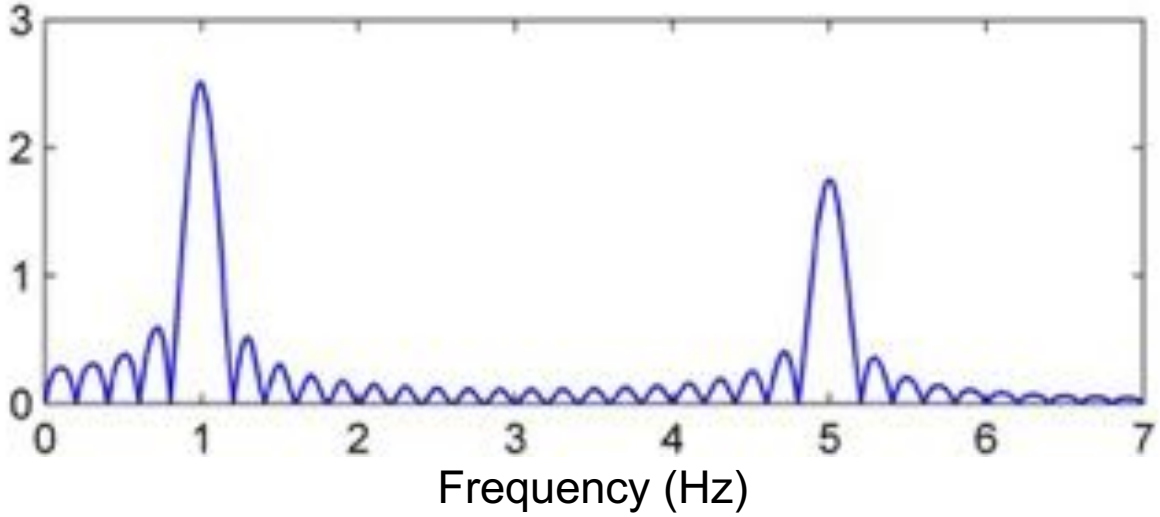
  → recovery of time information

- Short Time Fourier Transform (STFT)

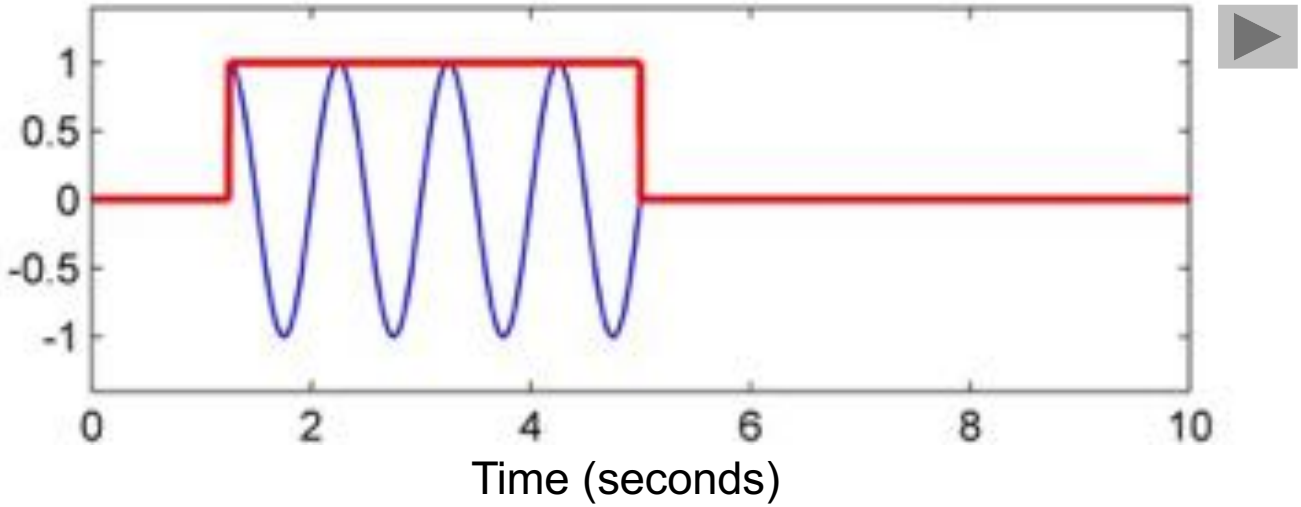
- Section is determined by pointwise multiplication of the signal with a localizing window function

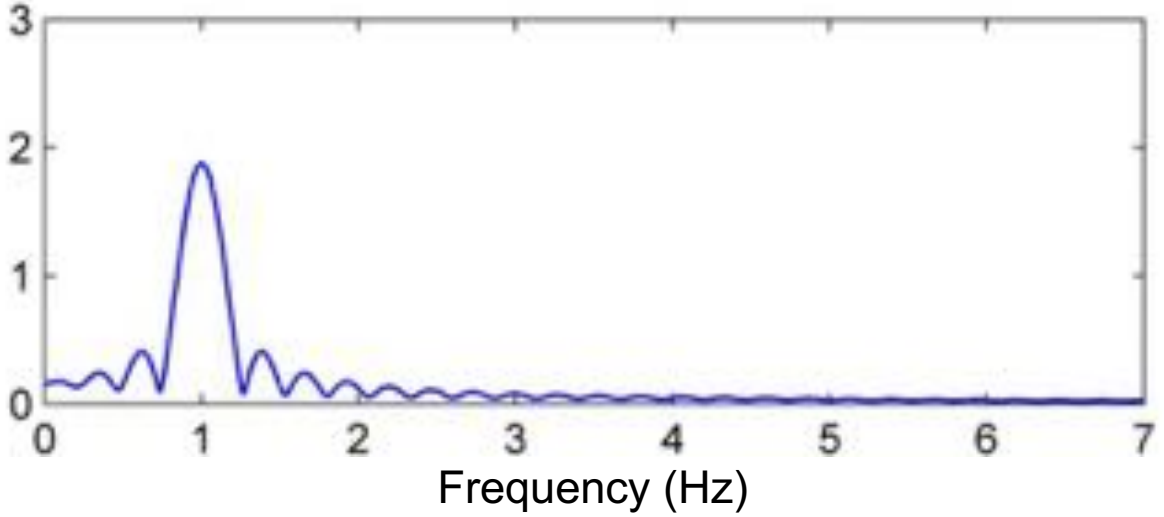
Short Time Fourier Transform



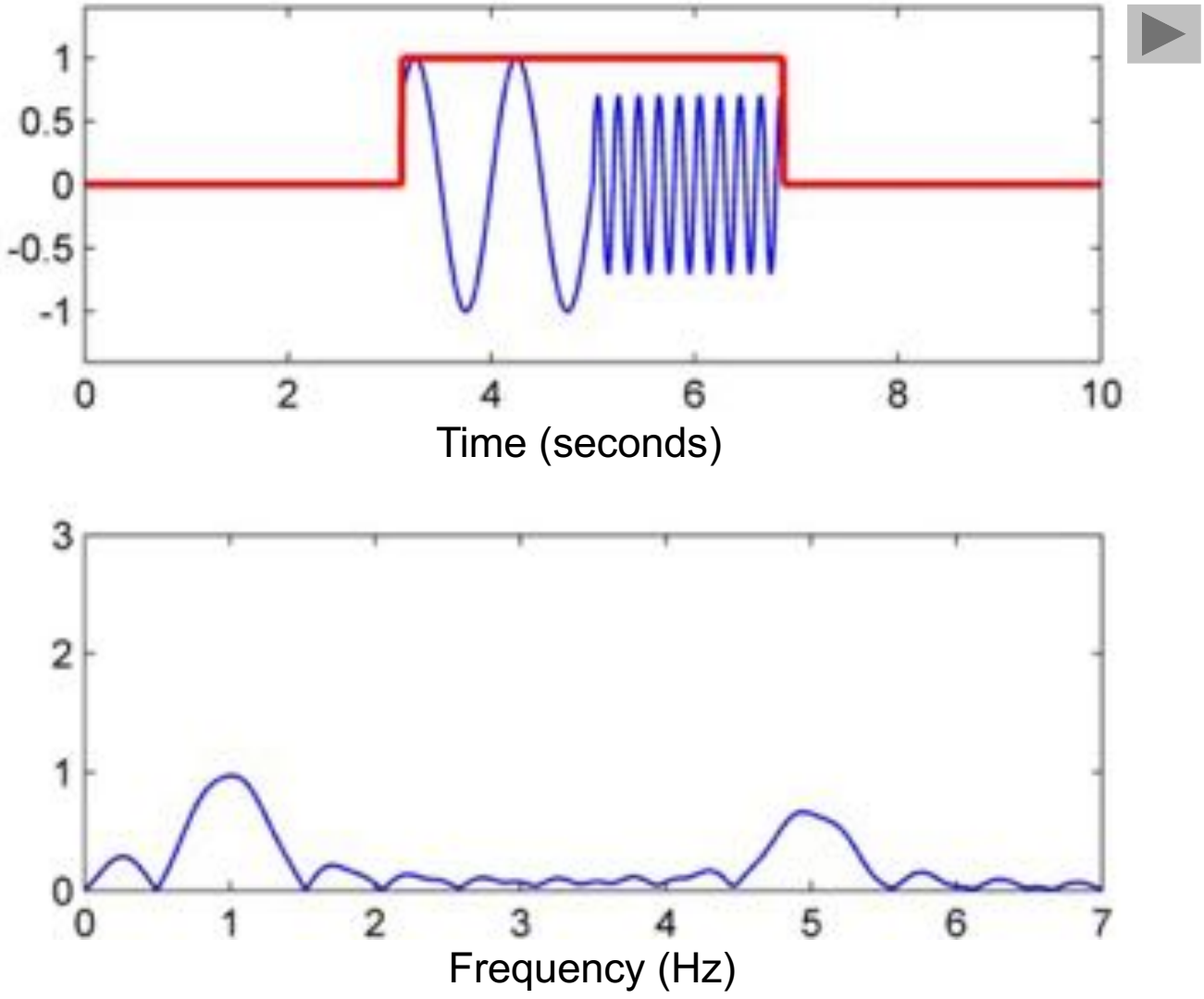


Short Time Fourier Transform

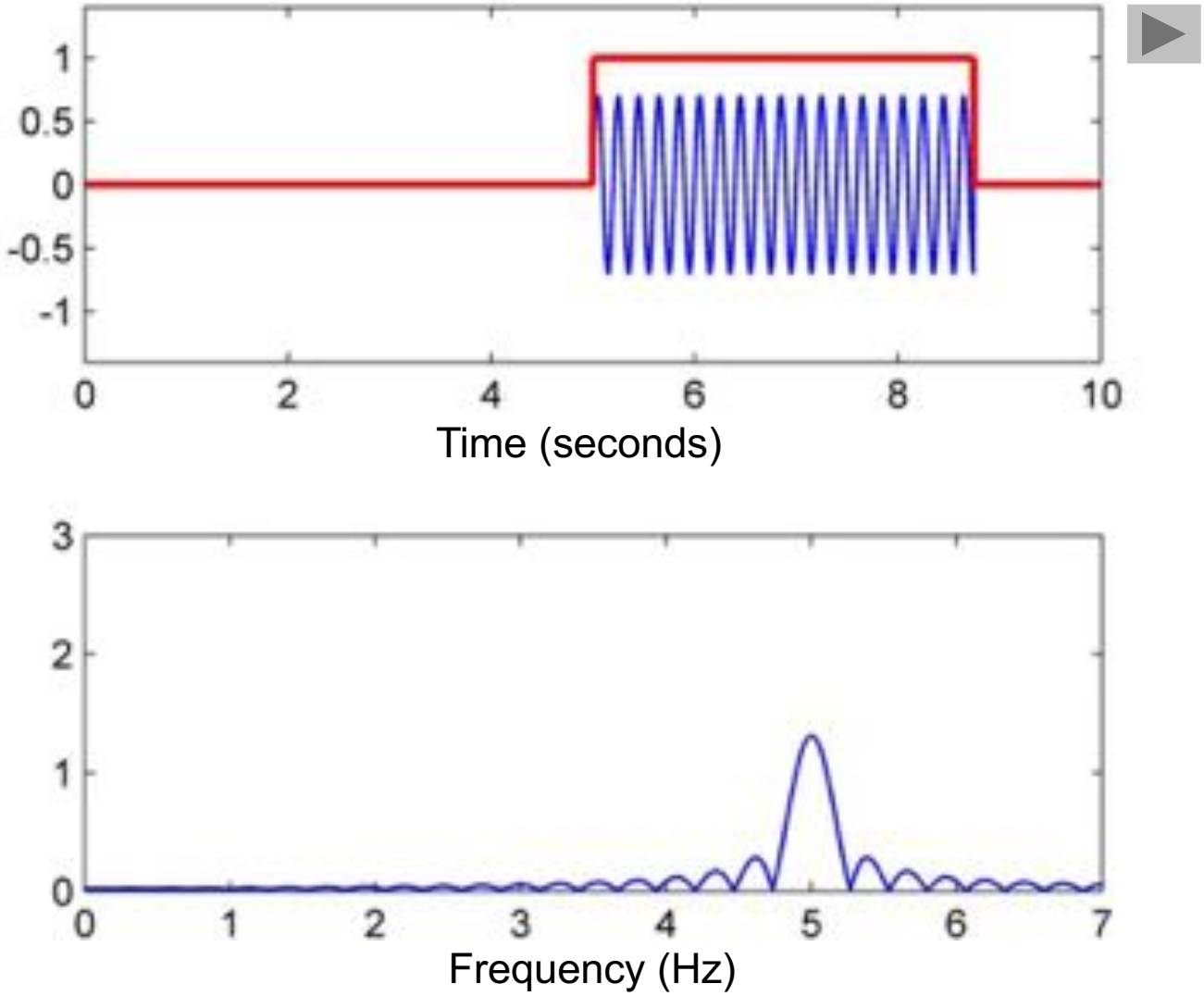




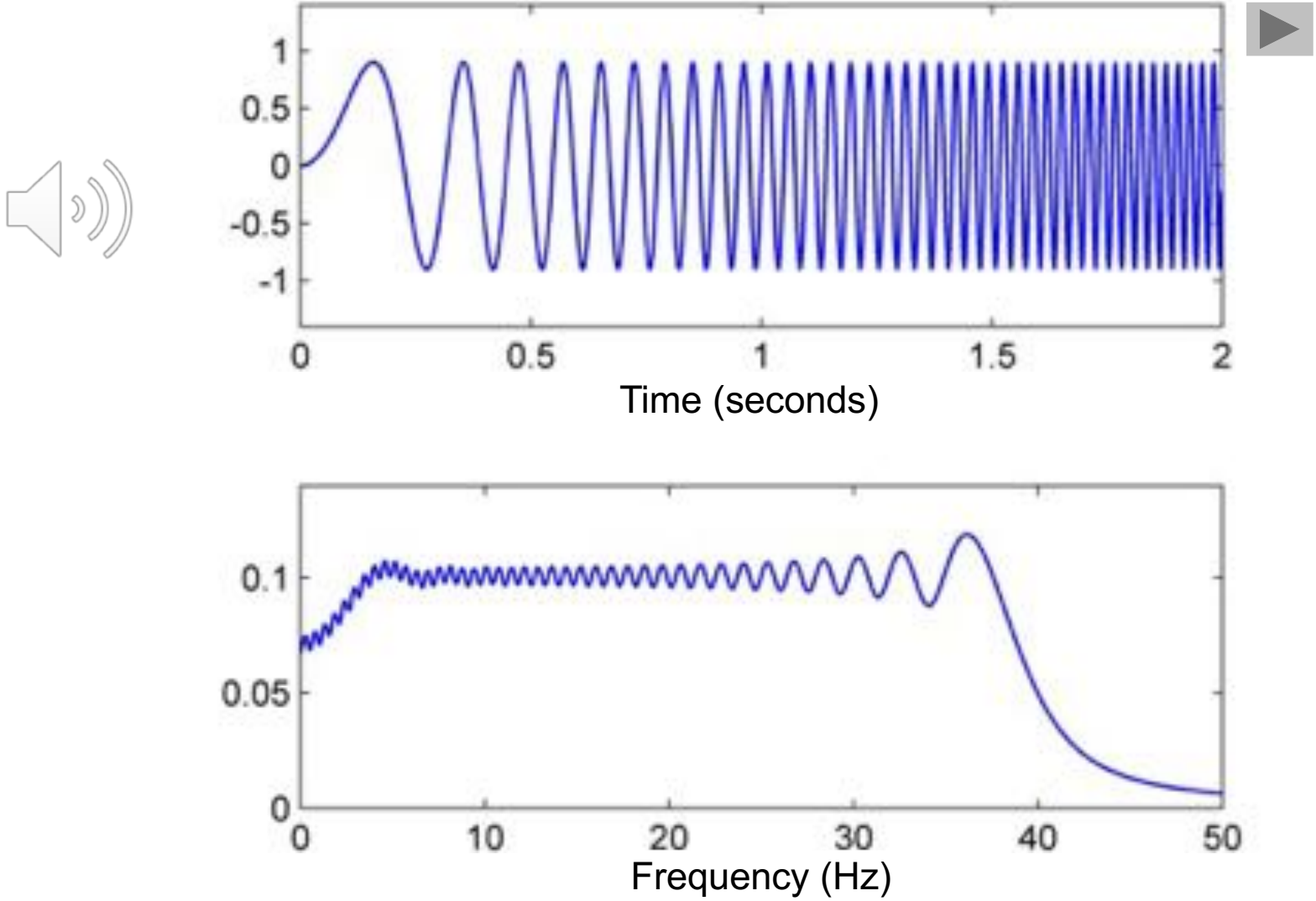
Short Time Fourier Transform



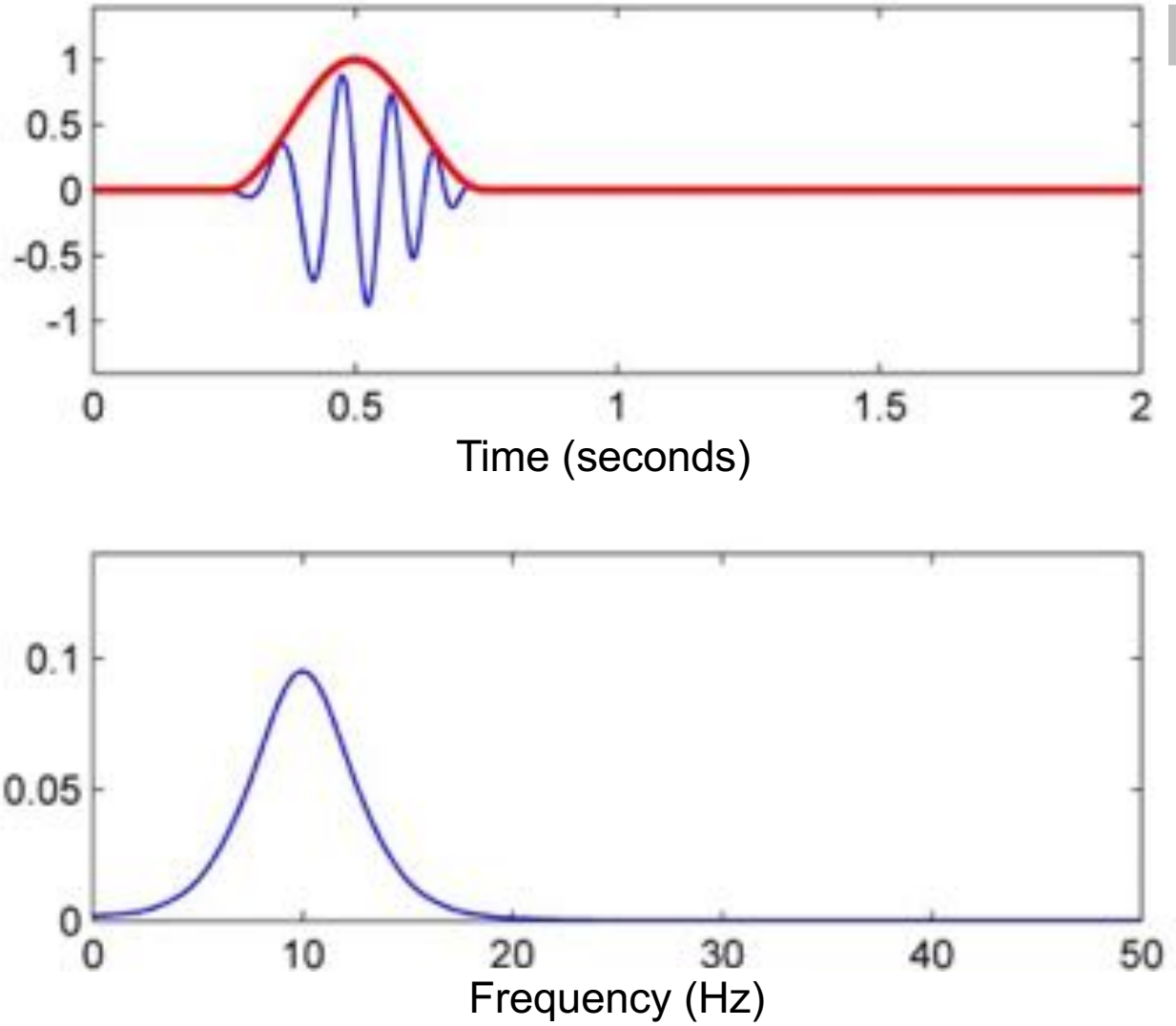
Short Time Fourier Transform



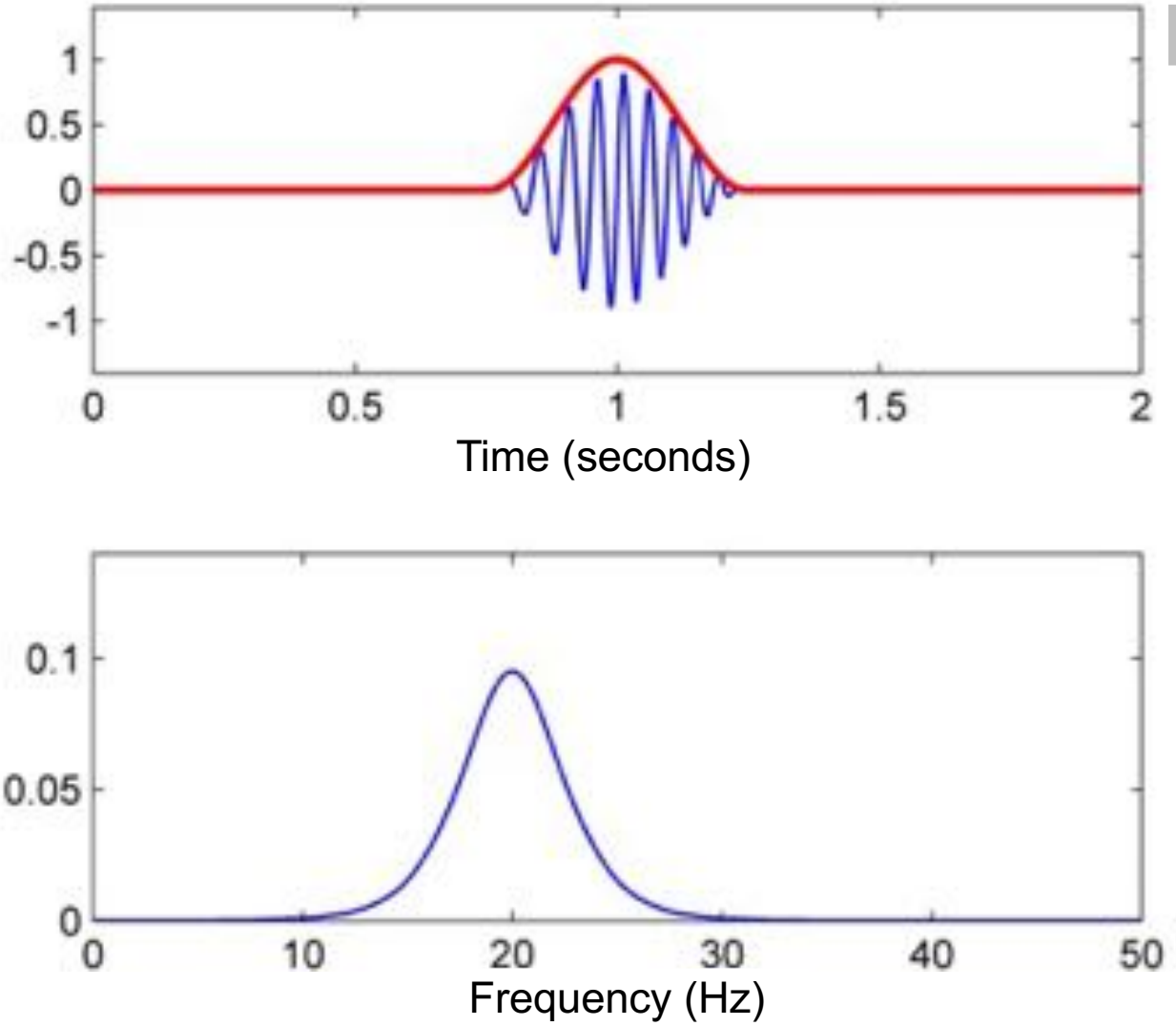
Short Time Fourier Transform



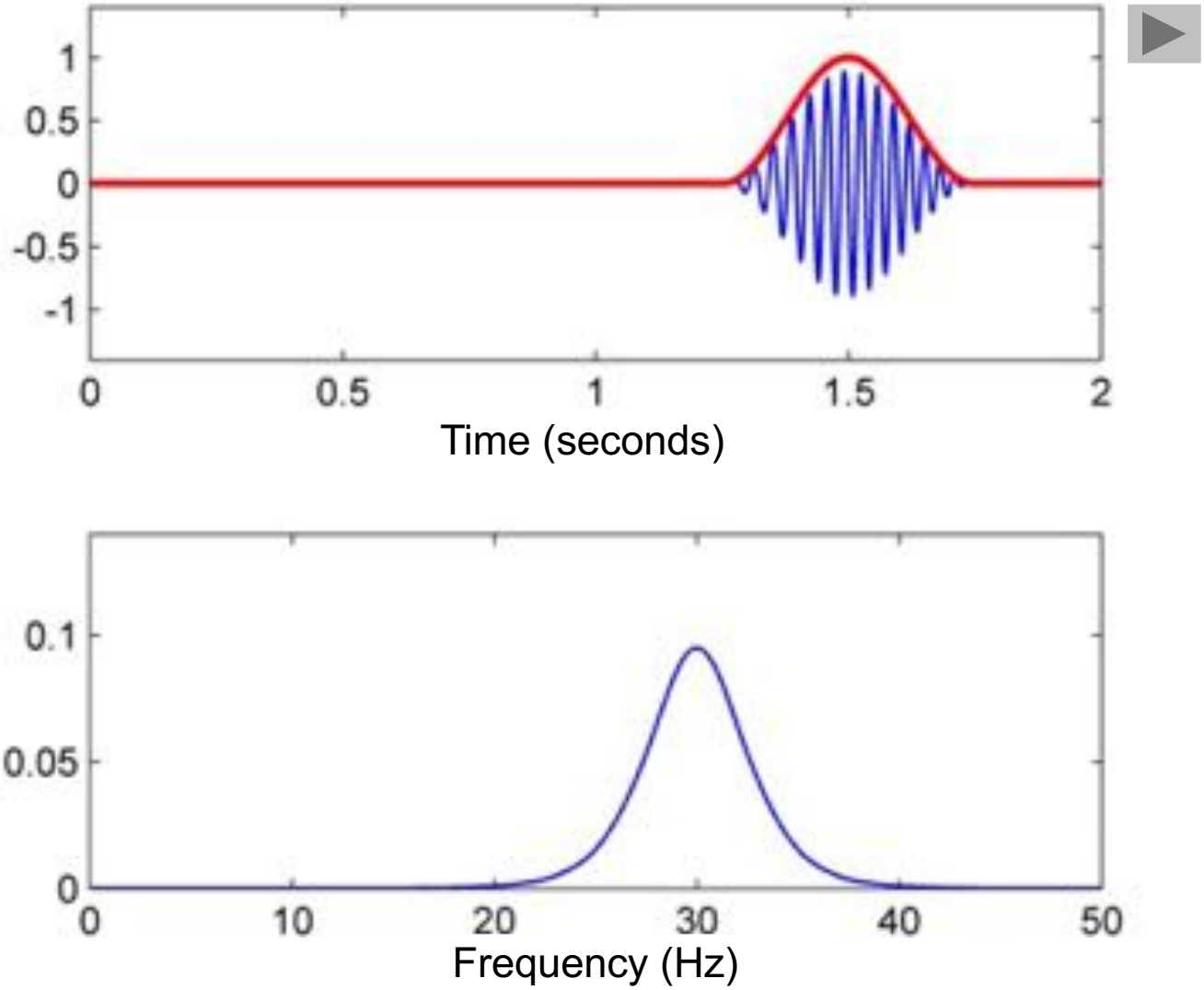
Short Time Fourier Transform



Short Time Fourier Transform



Short Time Fourier Transform



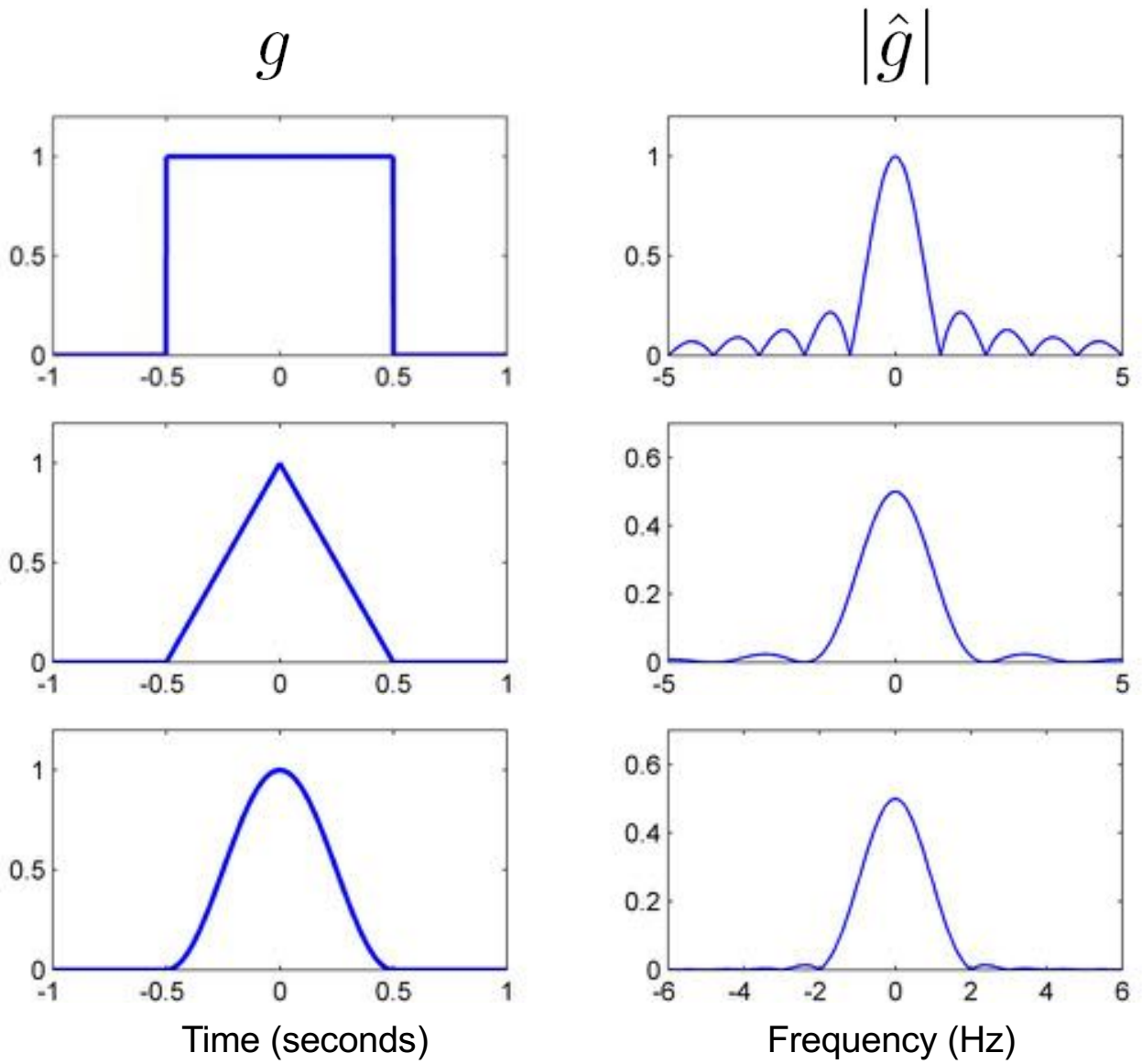
Short Time Fourier Transform

Window functions

Rectangular window

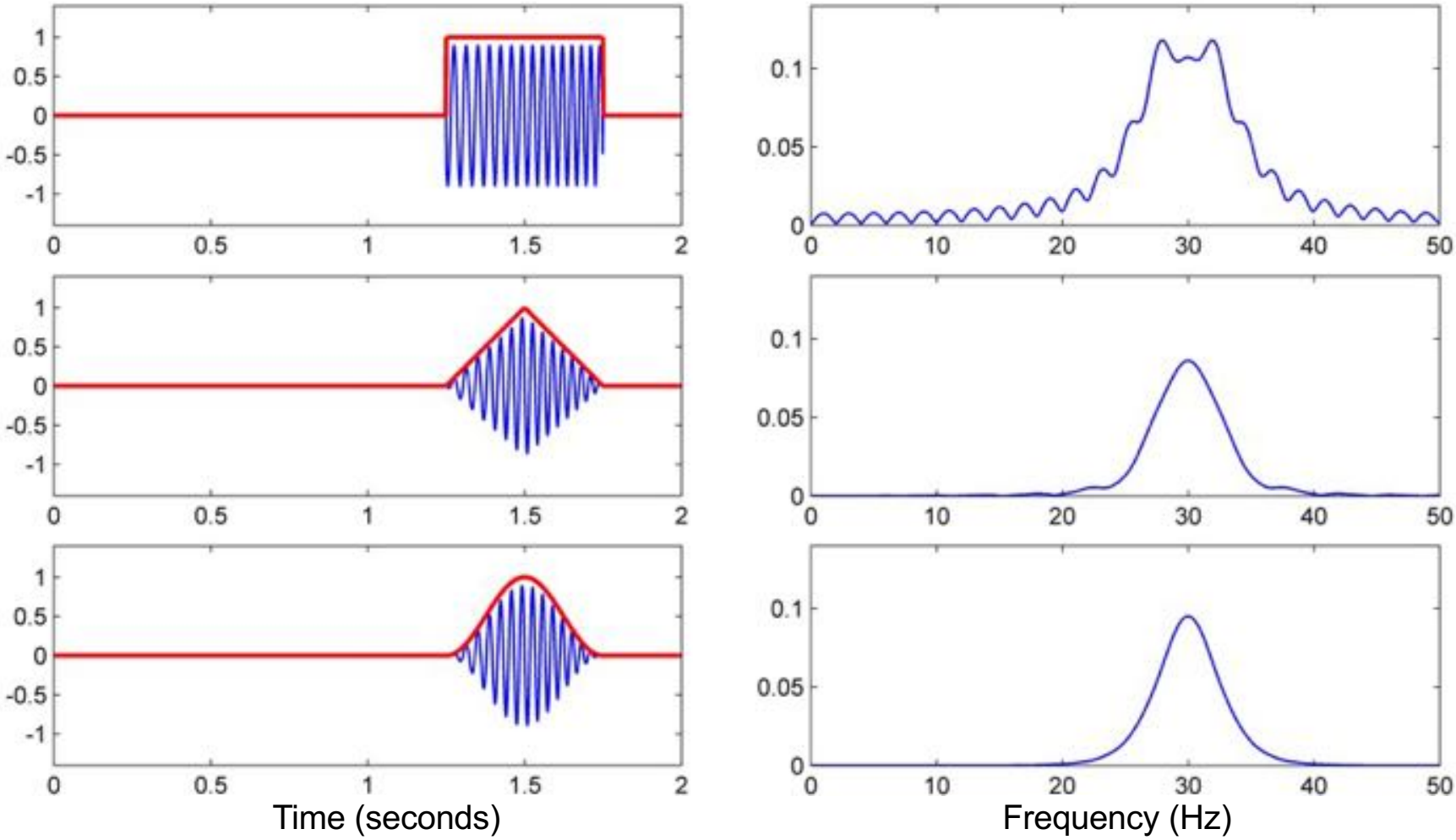
Triangular window

Hann window



Short Time Fourier Transform

Window functions



→ Trade off between smoothing and “rippling”

Short Time Fourier Transform

Definition

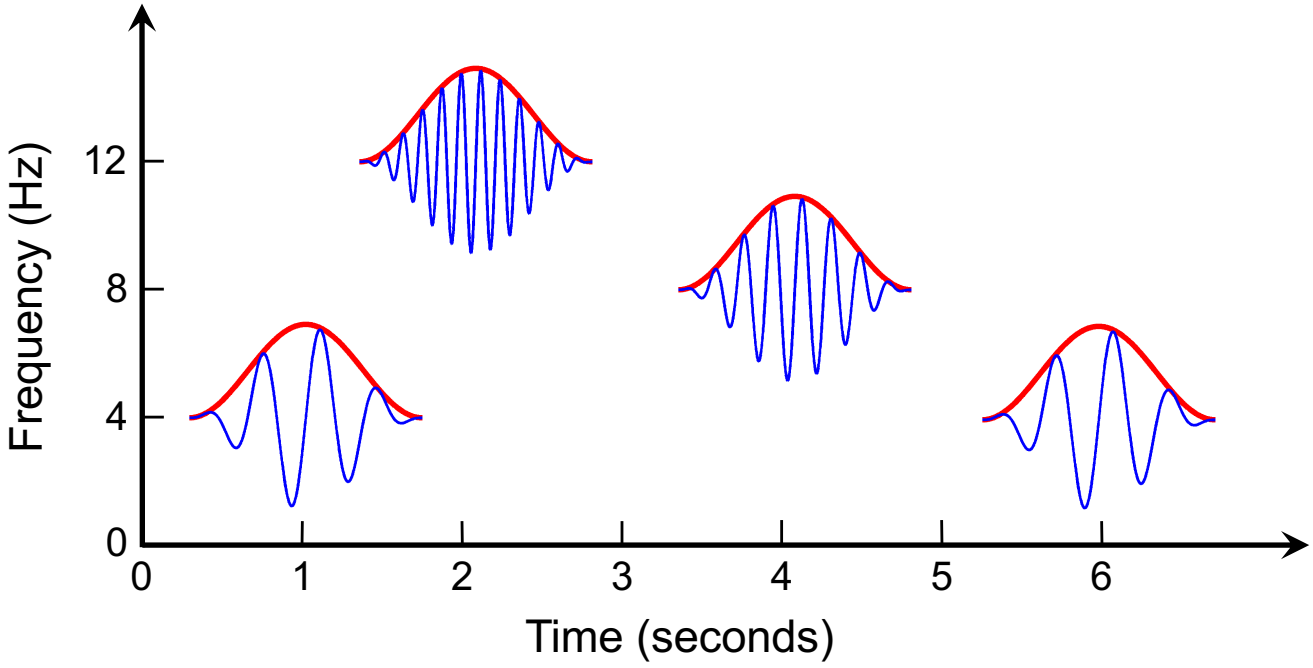
- Signal $f: \mathbb{R} \to \mathbb{R}$
- Window function $g : \mathbb{R} \to \mathbb{R} \quad (g \in L^2(\mathbb{R}), \|g\|_2 \neq 0)$
- STFT $\widetilde{f}_g(t,\omega) = \int_{u\in\mathbb{R}} f(u)\overline{g}(u-t)\exp(-2\pi i\omega u)du = \langle f | g_{t,\omega}\rangle$

with $g_{t,\omega}(u) = \exp(2\pi i\omega(u-t))g(u-t)$ for $u \in \mathbb{R}$

Short Time Fourier Transform

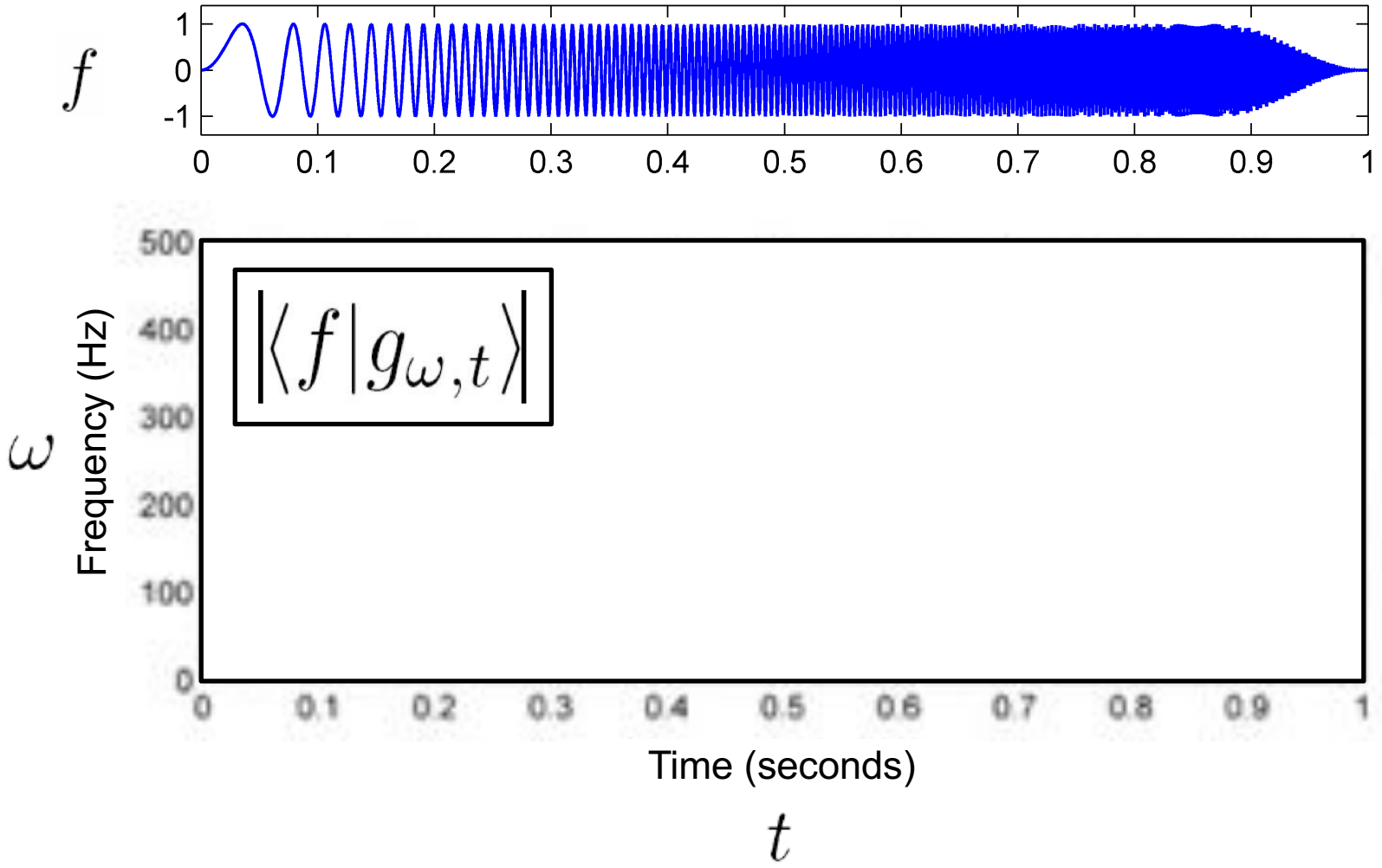
Intuition:

- $g_{t,\omega}$ is "musical note" of frequency ω centered at time t
- Inner product $\langle f | g_{t,\omega} \rangle$ measures the correlation between the musical note $g_{t,\omega}$ and the signal f



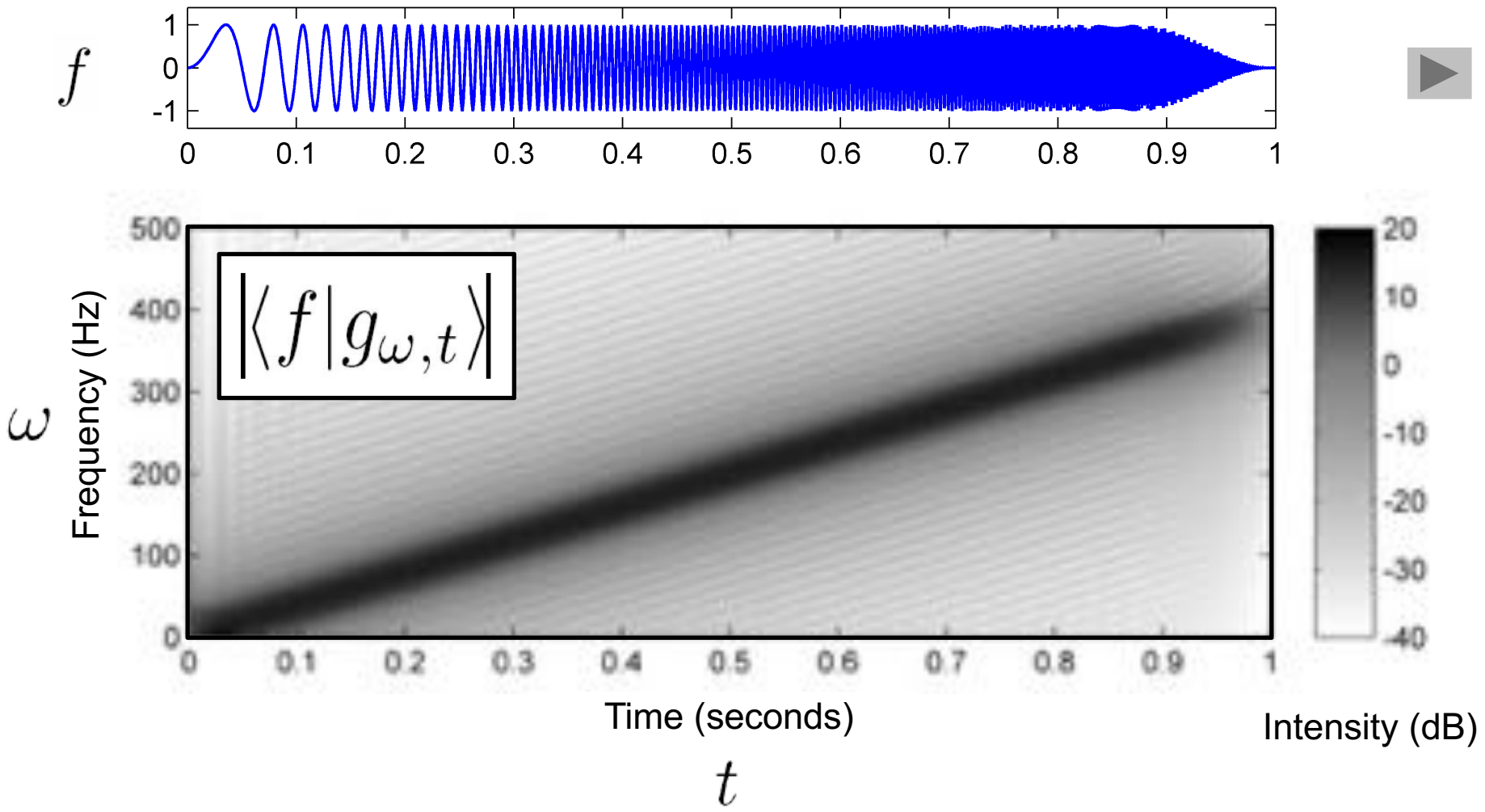
Time-Frequency Representation

Spectrogram



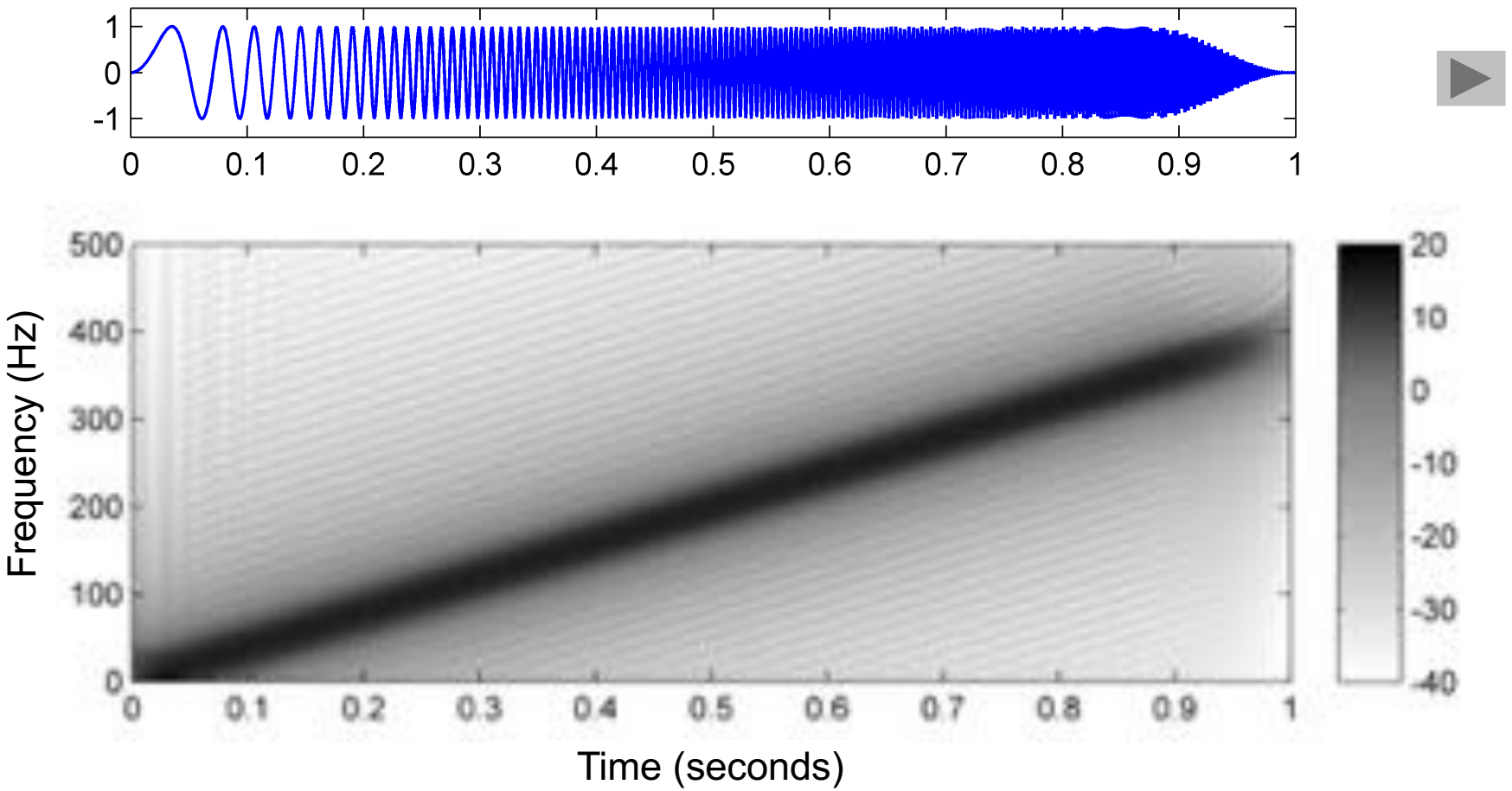
Time-Frequency Representation

Spectrogram



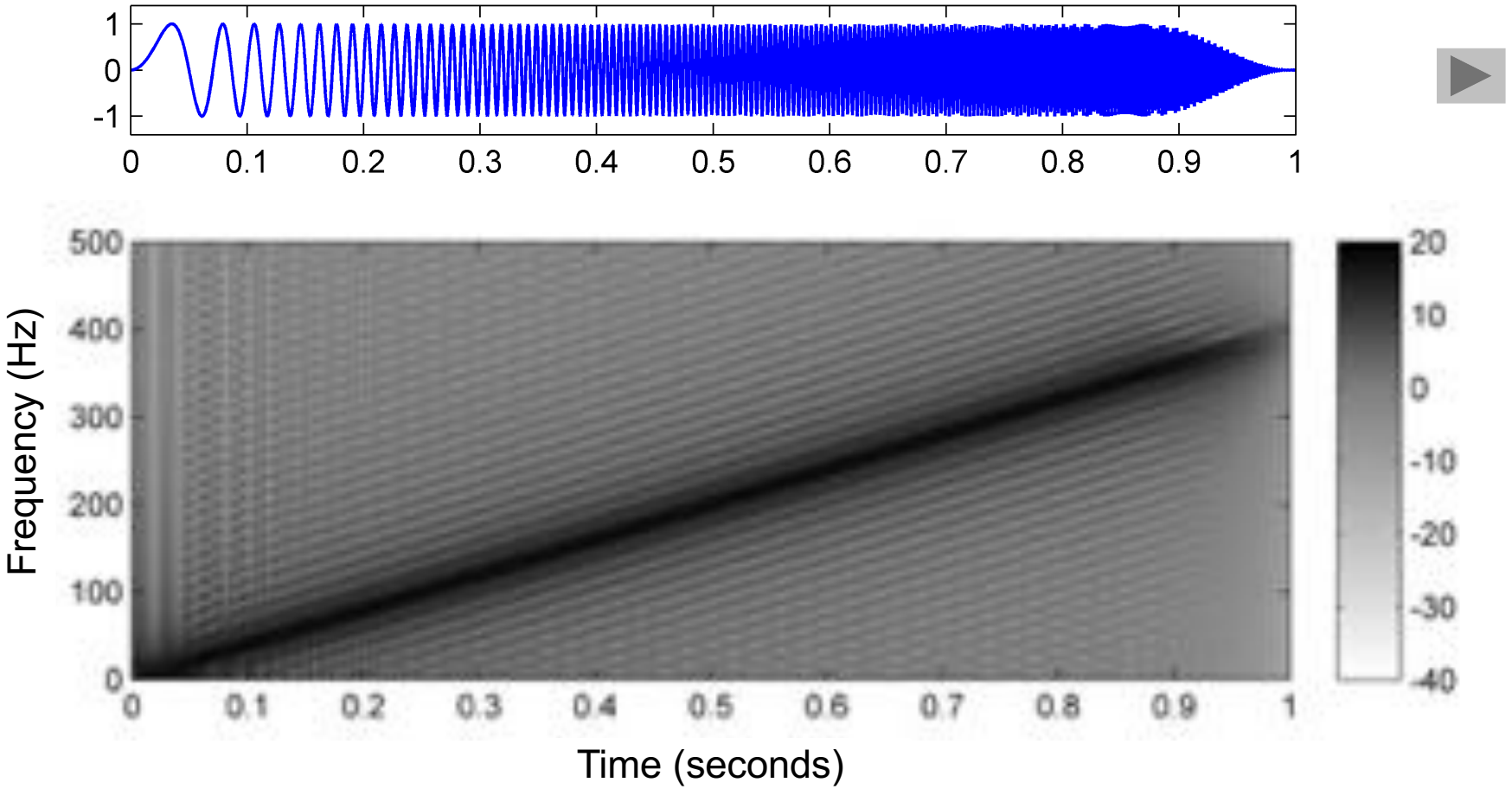
Time-Frequency Representation

Chirp signal and STFT with Hann window of length 50 ms



Time-Frequency Representation

Chirp signal and STFT with box window of length 50 ms



Time-Frequency Representation

Time-Frequency Localization

- Size of window constitutes a trade-off between time resolution and frequency resolution:

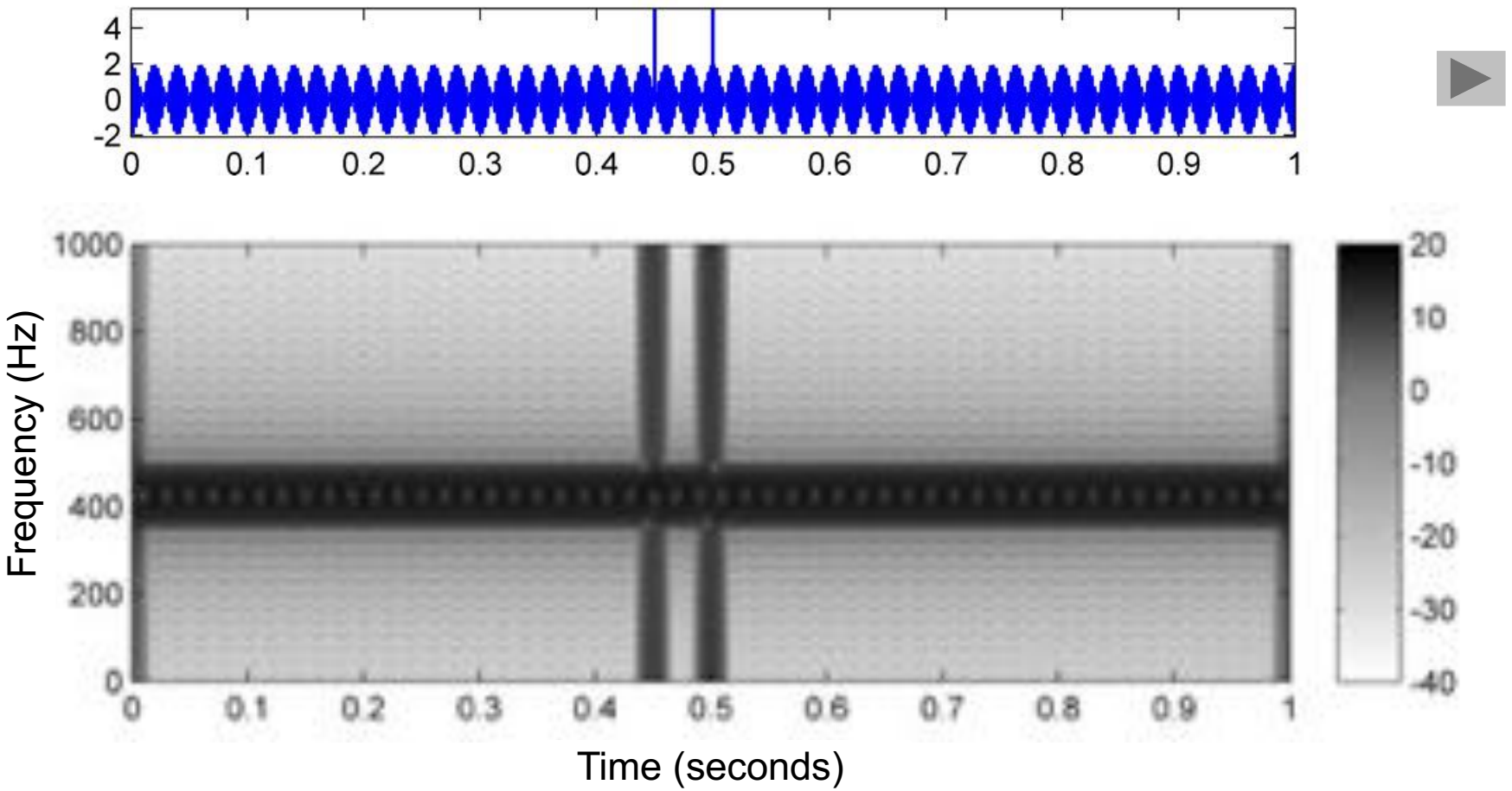
  Large window : poor time resolution
  good frequency resolution

  Small window : good time resolution
  poor frequency resolution

- Heisenberg Uncertainty Principle: there is no window function that localizes in time and frequency with arbitrary position.

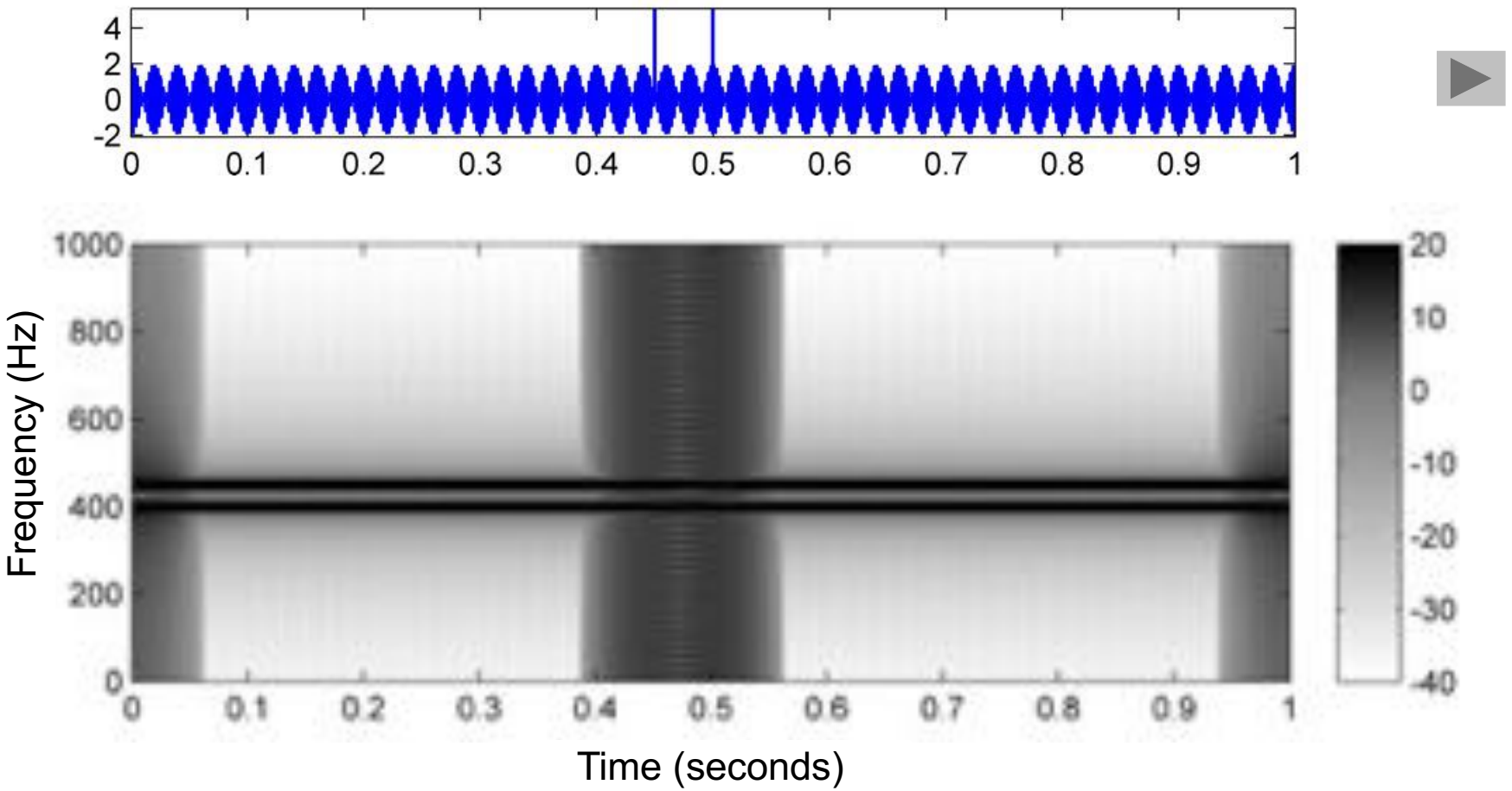
Time-Frequency Representation

Signal and STFT with Hann window of length 20 ms



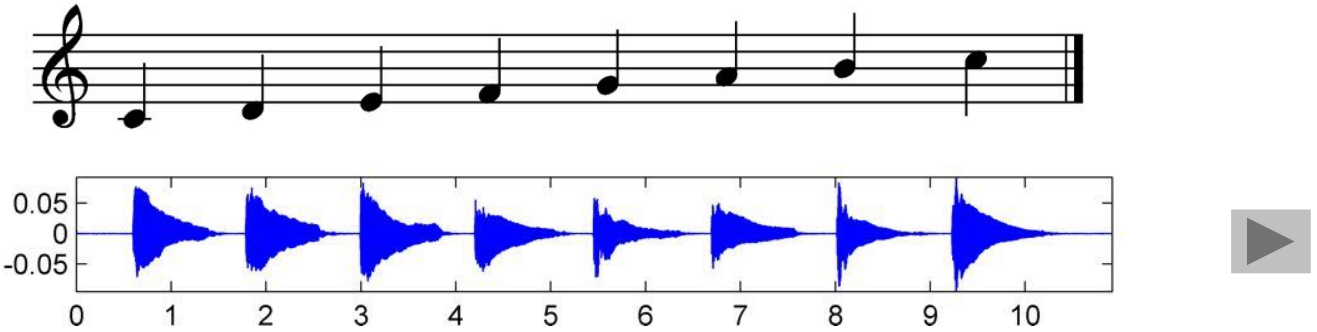
Time-Frequency Representation

Signal and STFT with Hann window of length 100 ms

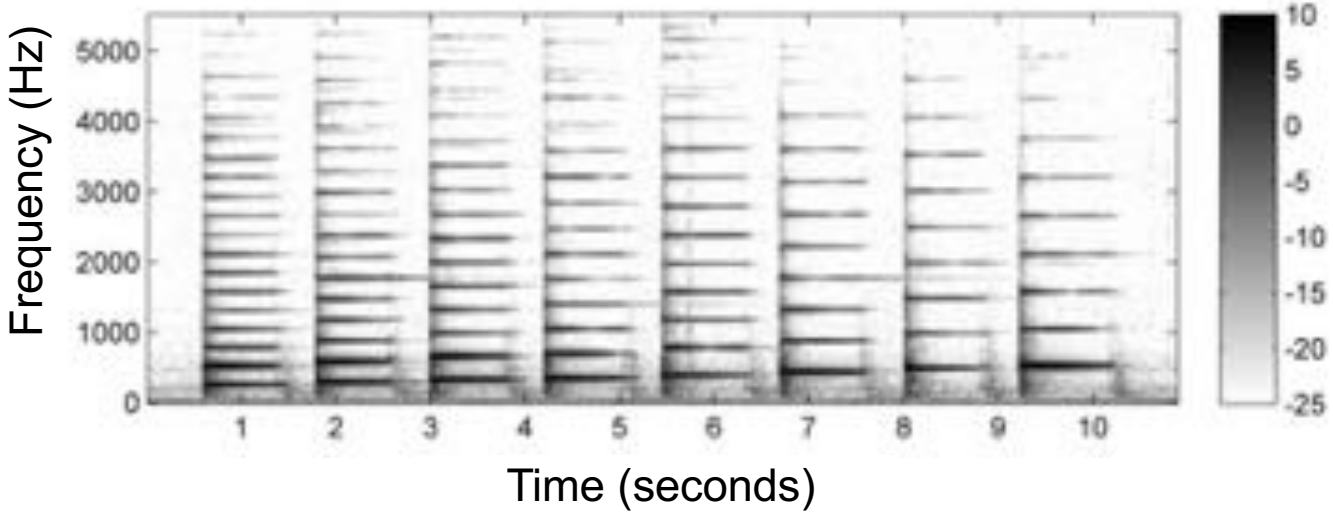


Audio Features

Example: C-major scale (piano)

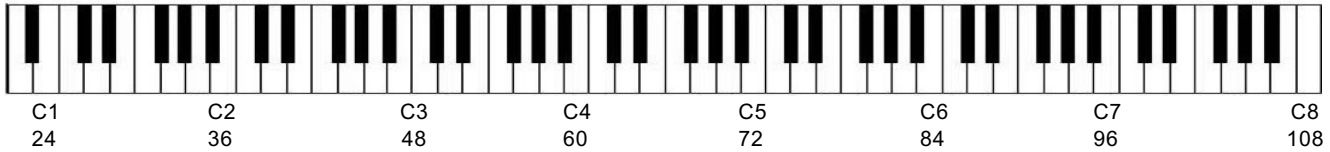


Spectrogram

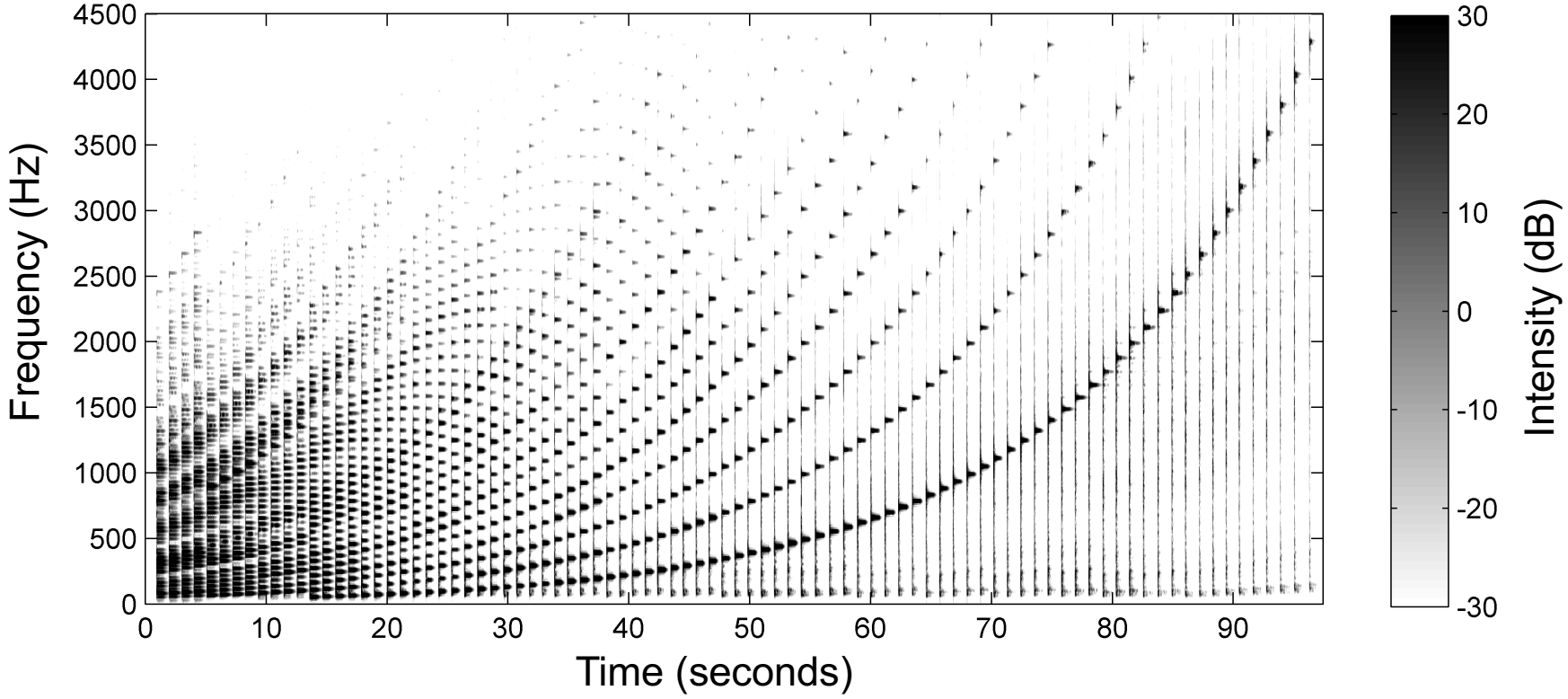


Audio Features

Example: Chromatic scale



Spectrogram



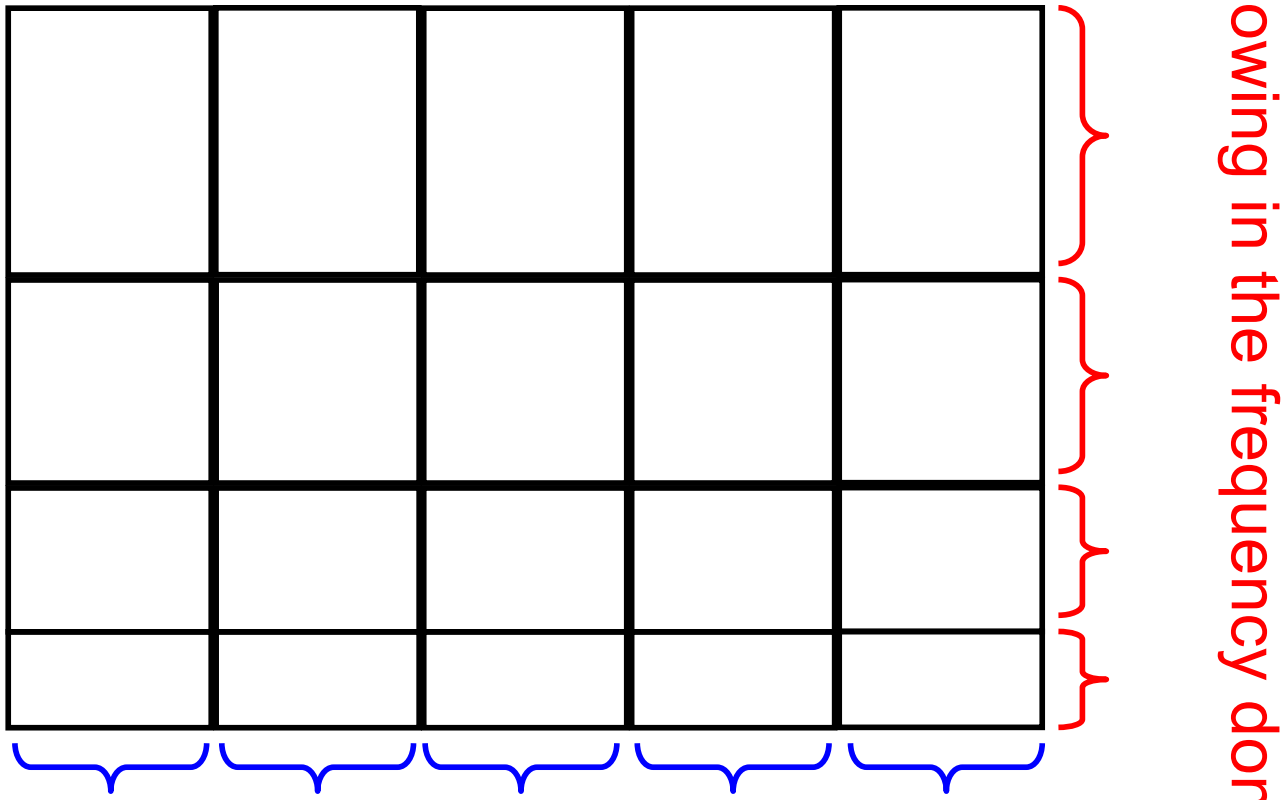
Audio Features

Idea: Binning of Fourier coefficients

Divide up the fequency axis into
logarithmically spaced “pitch regions”
and combine spectral coefficients
of each region to a single pitch coefficient.

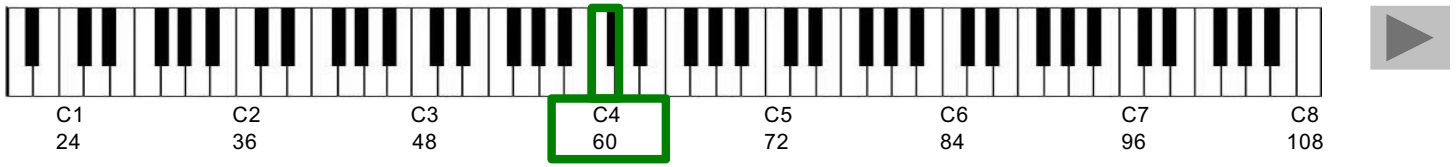
Audio Features

Time-frequency representation

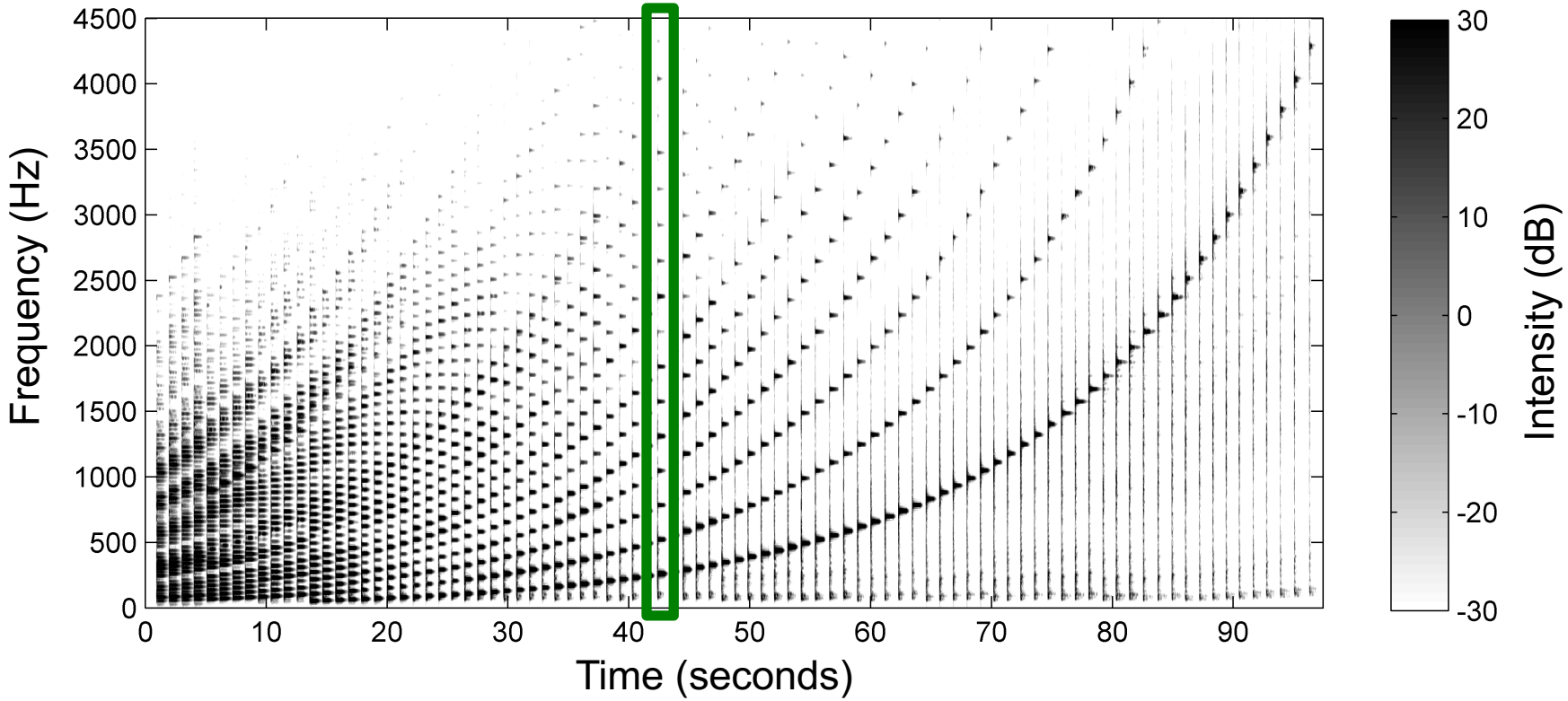


Audio Features

Example: Chromatic scale

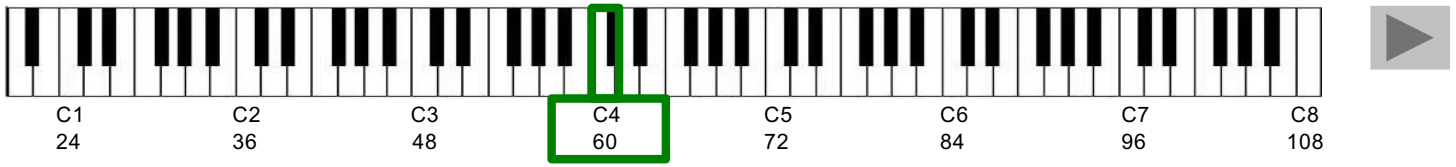


Spectrogram

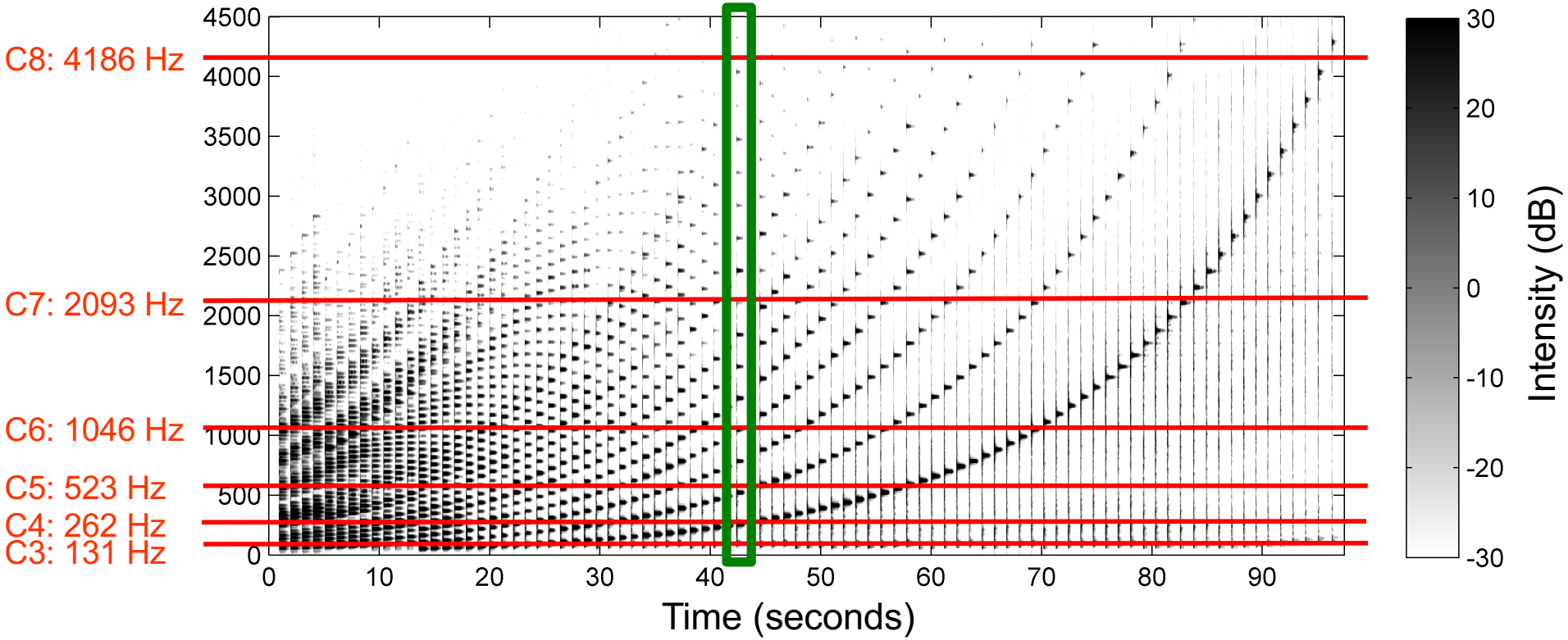


Audio Features

Example: Chromatic scale

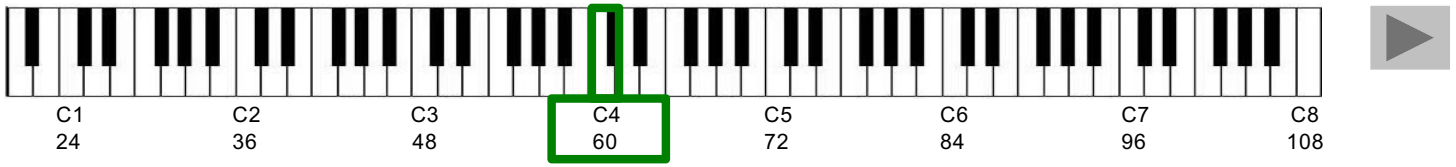


Spectrogram

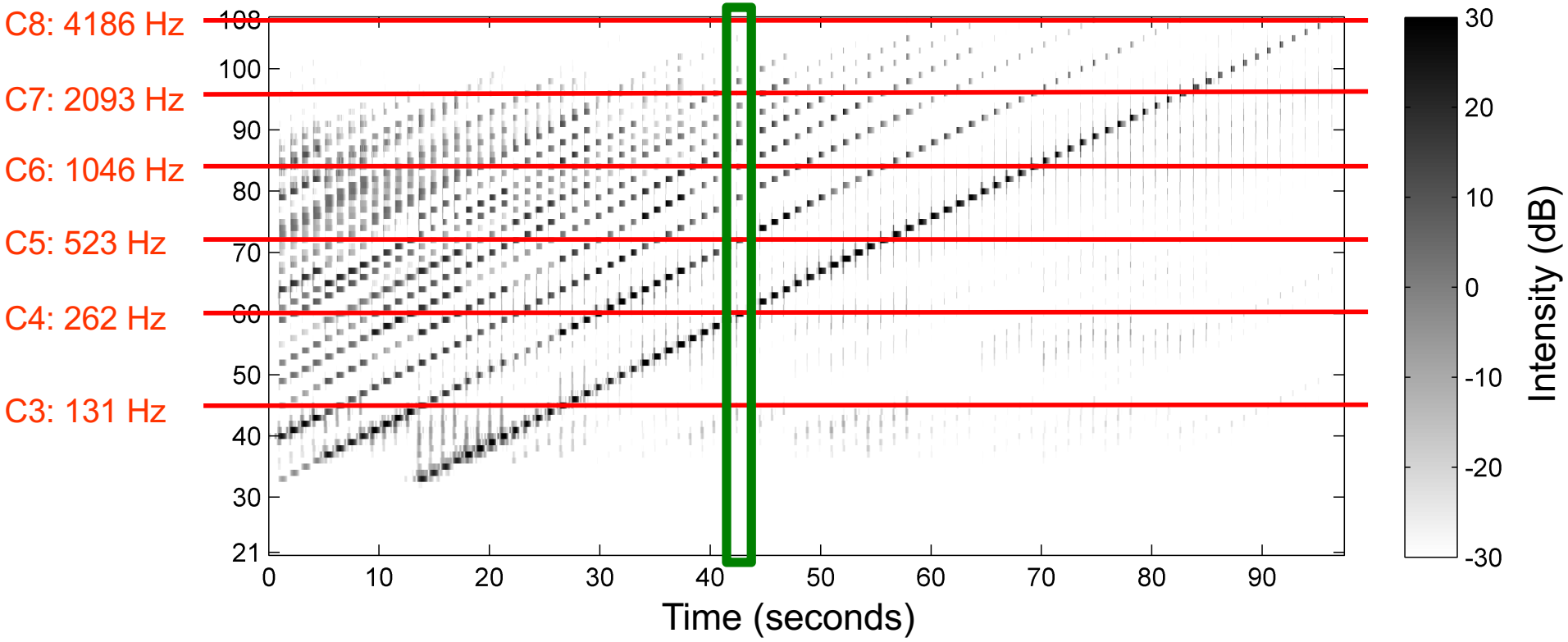


Audio Features

Example: Chromatic scale



Log-frequency spectrogram



Audio Features

Chroma features

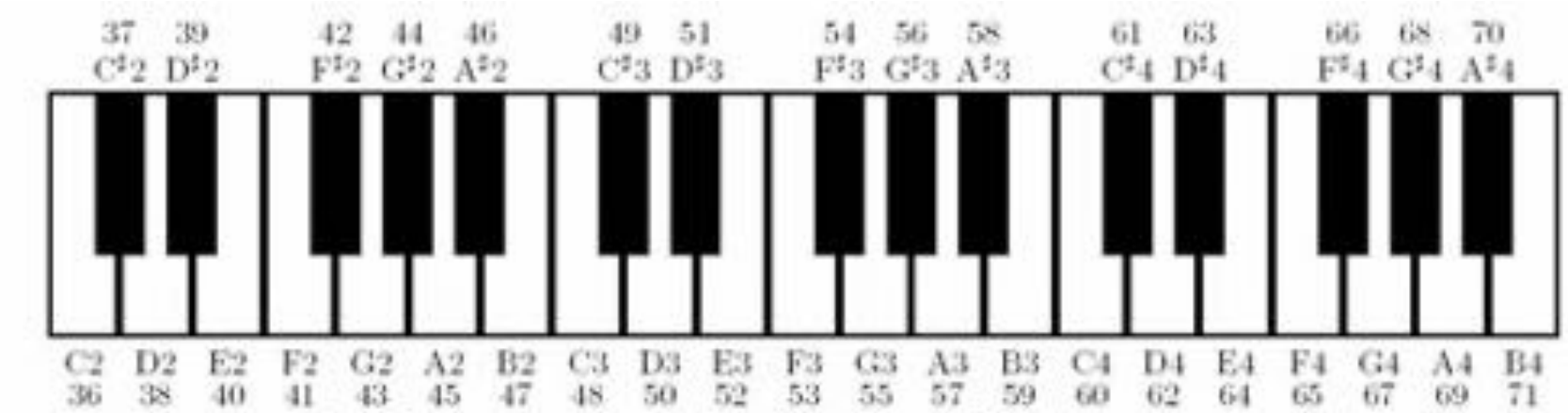
- Human perception of pitch is periodic in the sense that two pitches are perceived as similar in color if they differ by an octave.
- Separation of pitch into two components: tone height (octave number) and chroma.
- Chroma : 12 traditional pitch classes of the equal-tempered scale. For example:

  Chroma C $\widehat{=} \{\ldots, \mathrm{C0}, \mathrm{C1}, \mathrm{C2}, \mathrm{C3}, \ldots\}$
- Computation: pitch features → chroma features

  Add up all pitches belonging to the same class
- Result: 12-dimensional chroma vector.

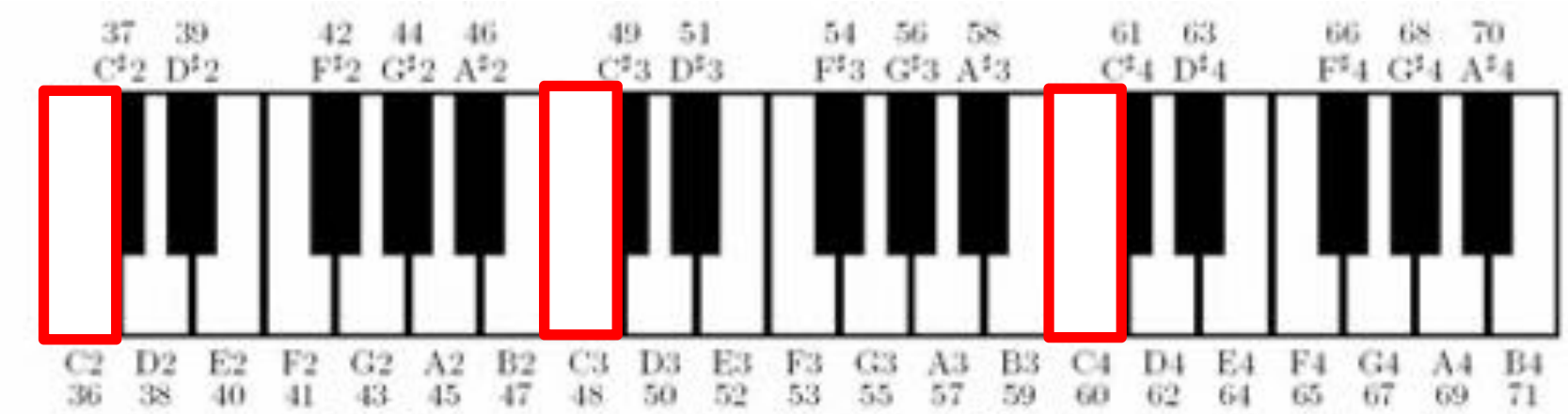
Audio Features

Chroma features



Audio Features

Chroma features

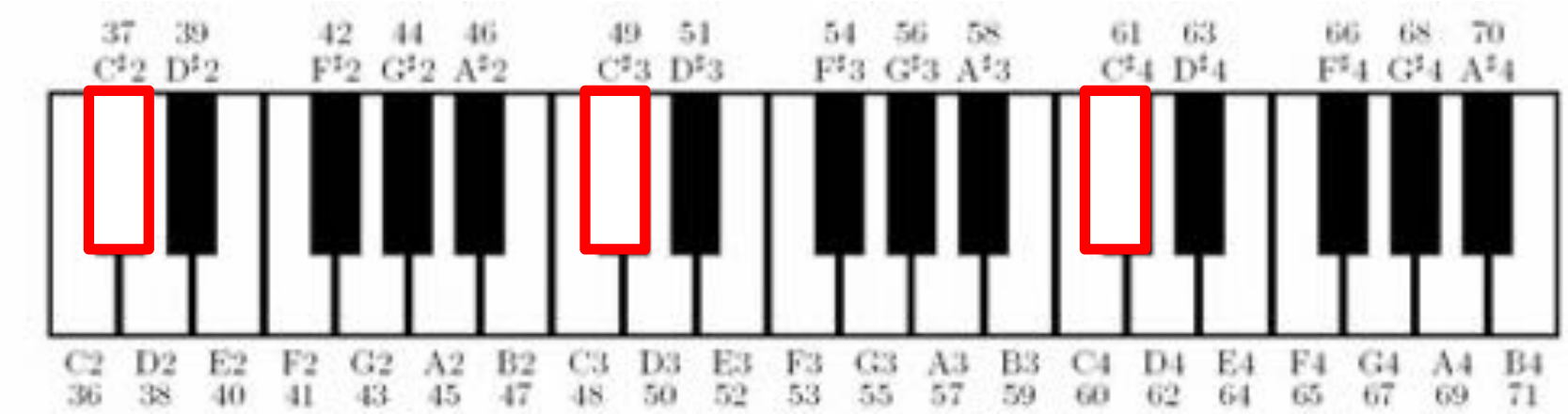


C2 C3 C4

Chroma C

Audio Features

Chroma features

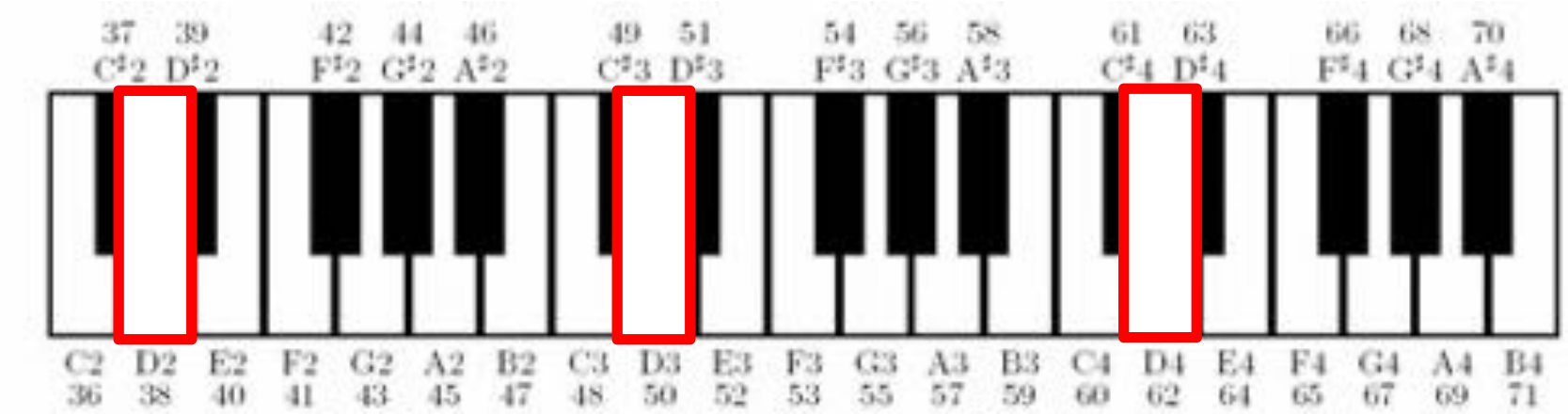


$C^{\#}2$ $C^{\#}3$ $C^{\#}4$

Chroma $C^{\#}$

Audio Features

Chroma features

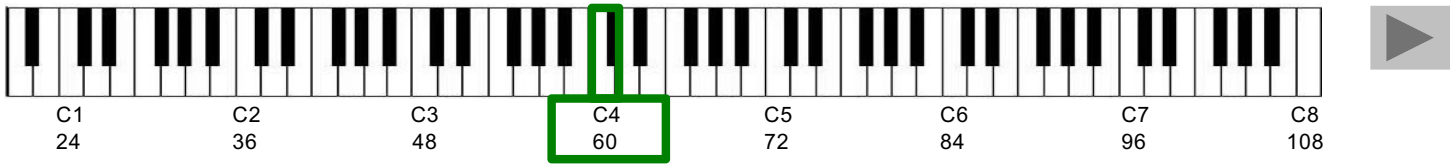


D2 D3 D4

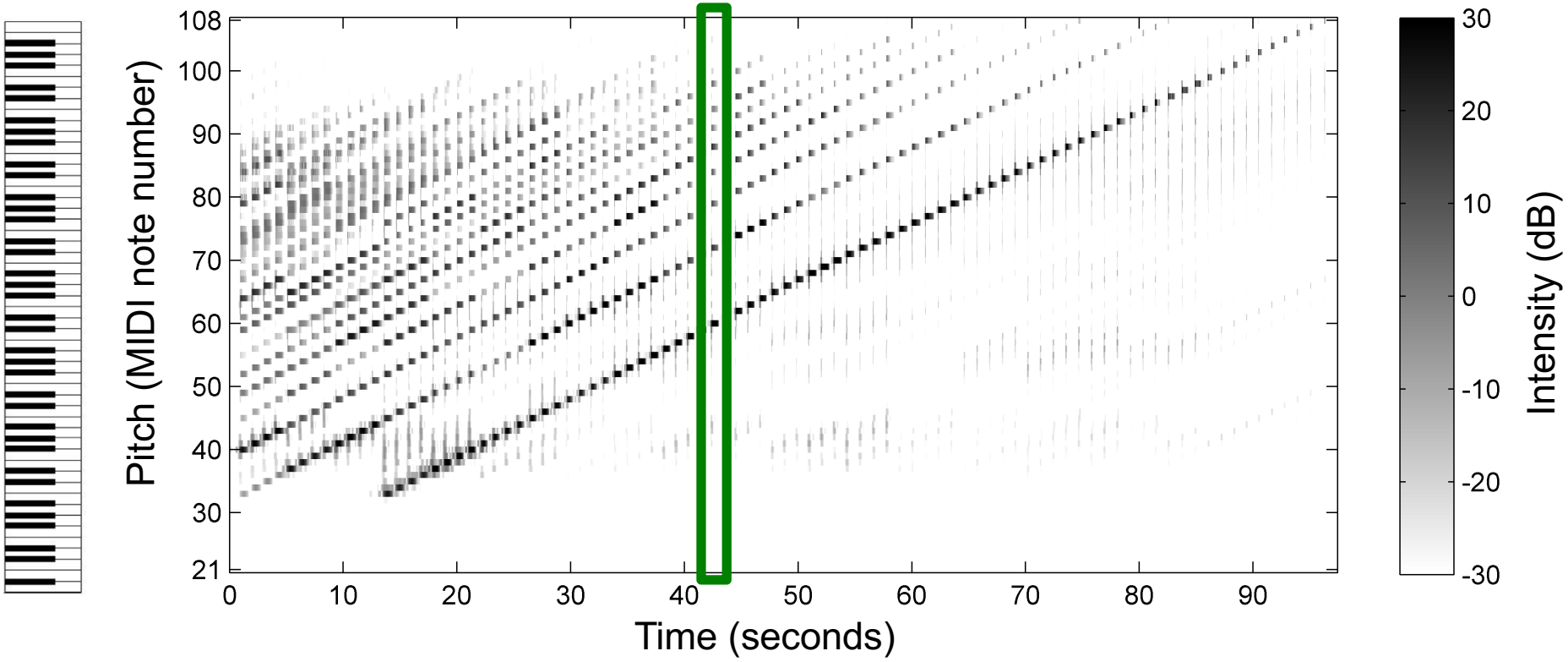
Chroma D

Audio Features

Example: Chromatic scale

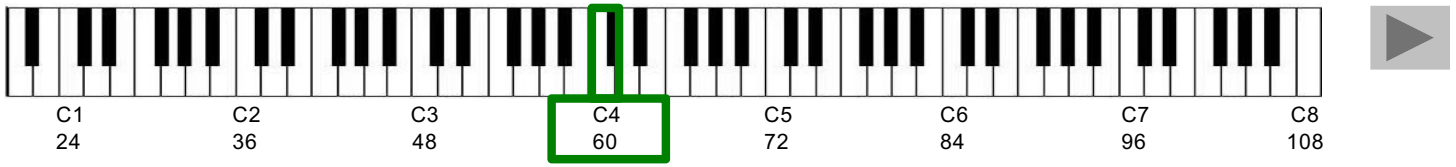


Log-frequency spectrogram

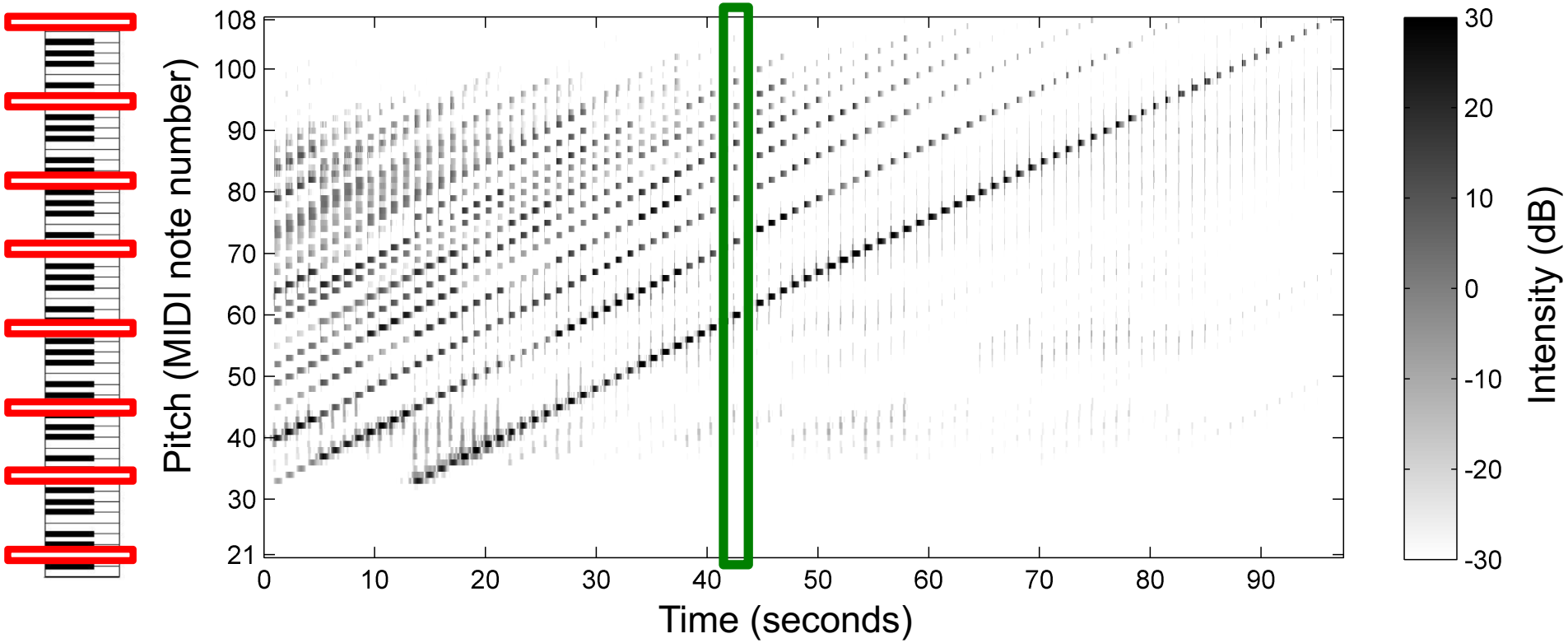


Audio Features

Example: Chromatic scale



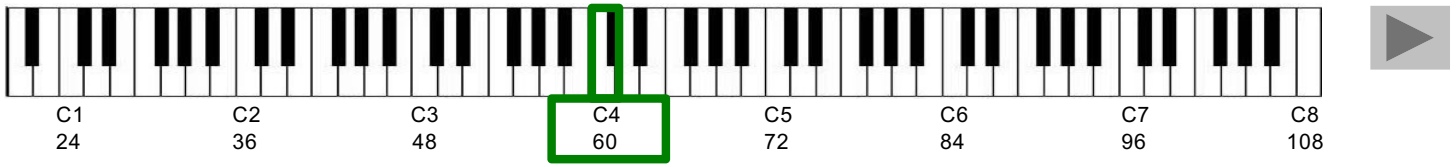
Log-frequency spectrogram



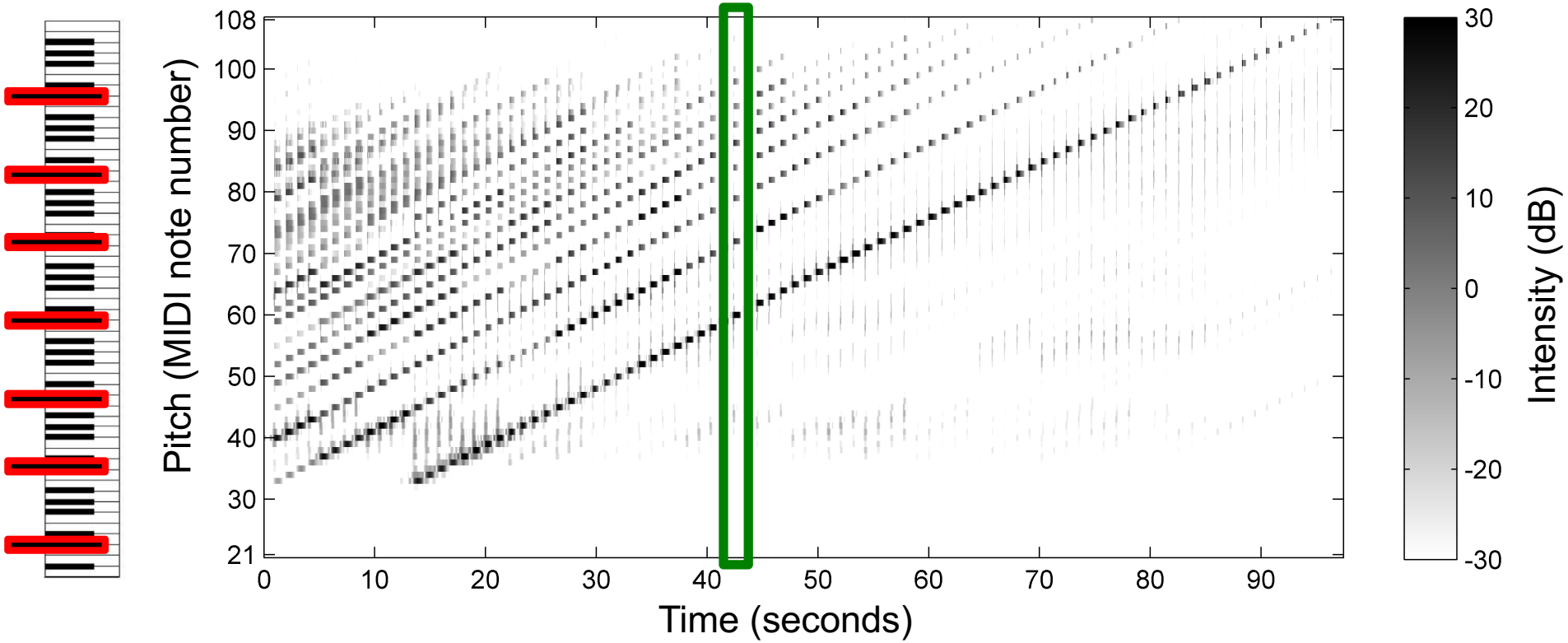
Chroma C

Audio Features

Example: Chromatic scale



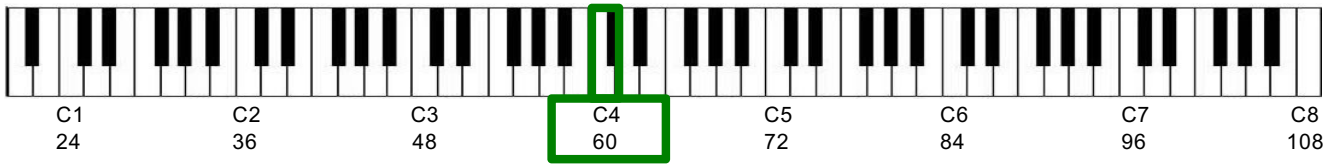
Log-frequency spectrogram



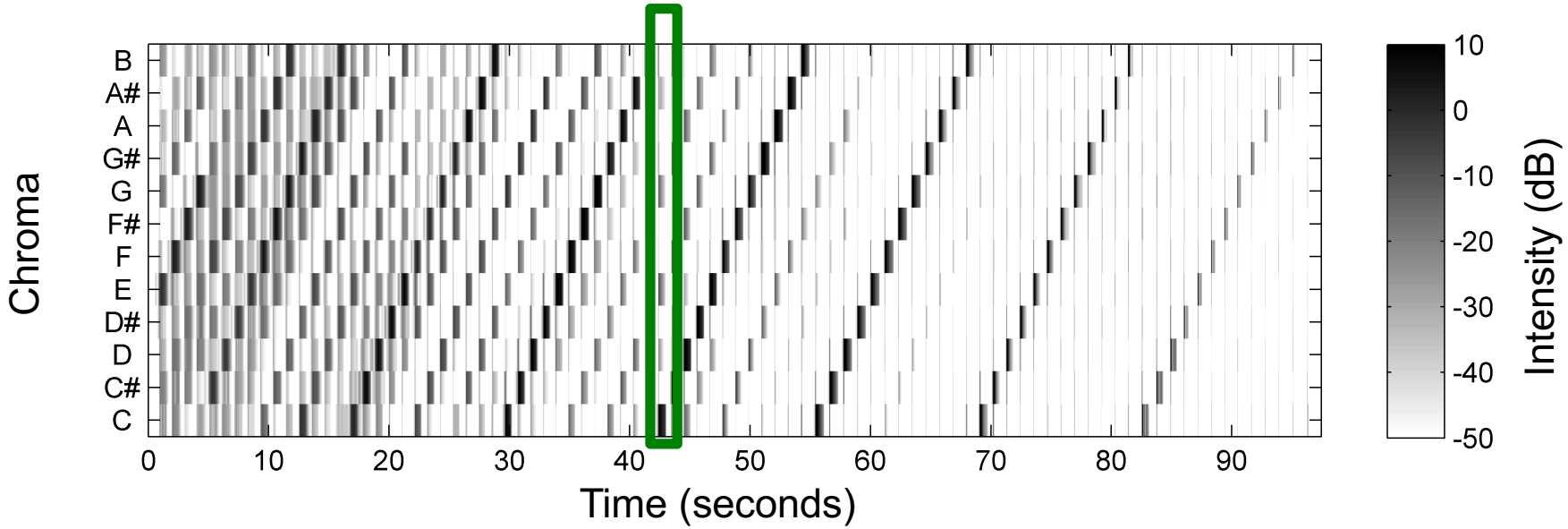
Chroma $C^{\#}$

Audio Features

Example: Chromatic scale

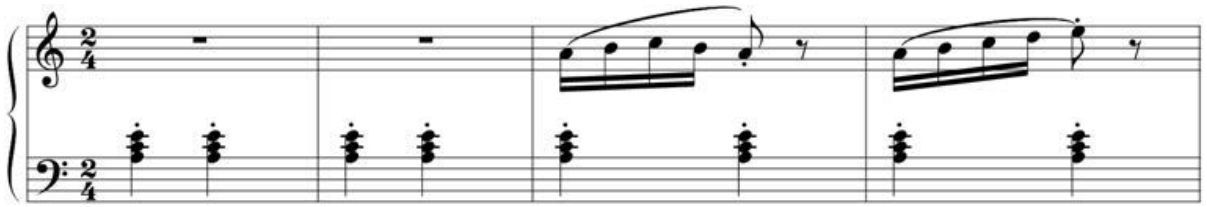


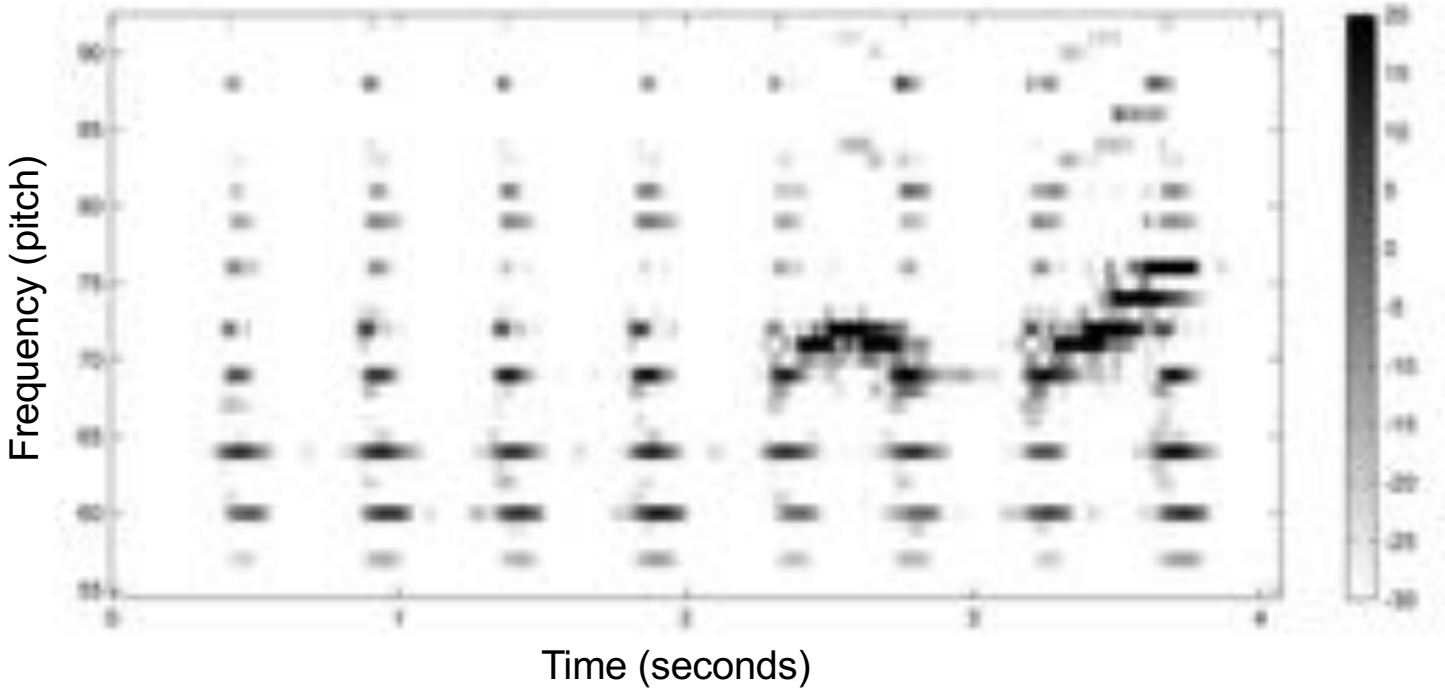
Chromagram



Audio Features

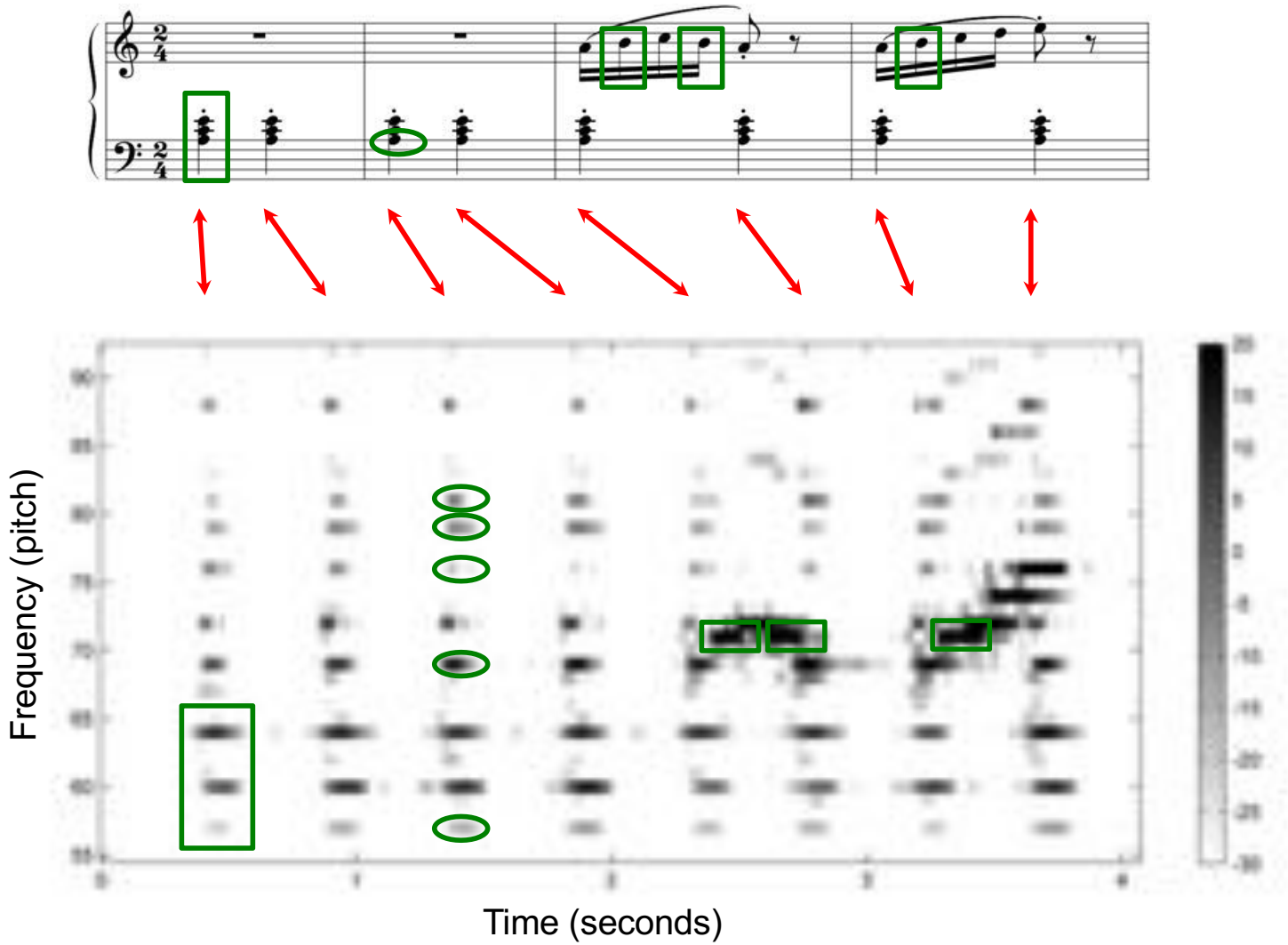
Chroma features





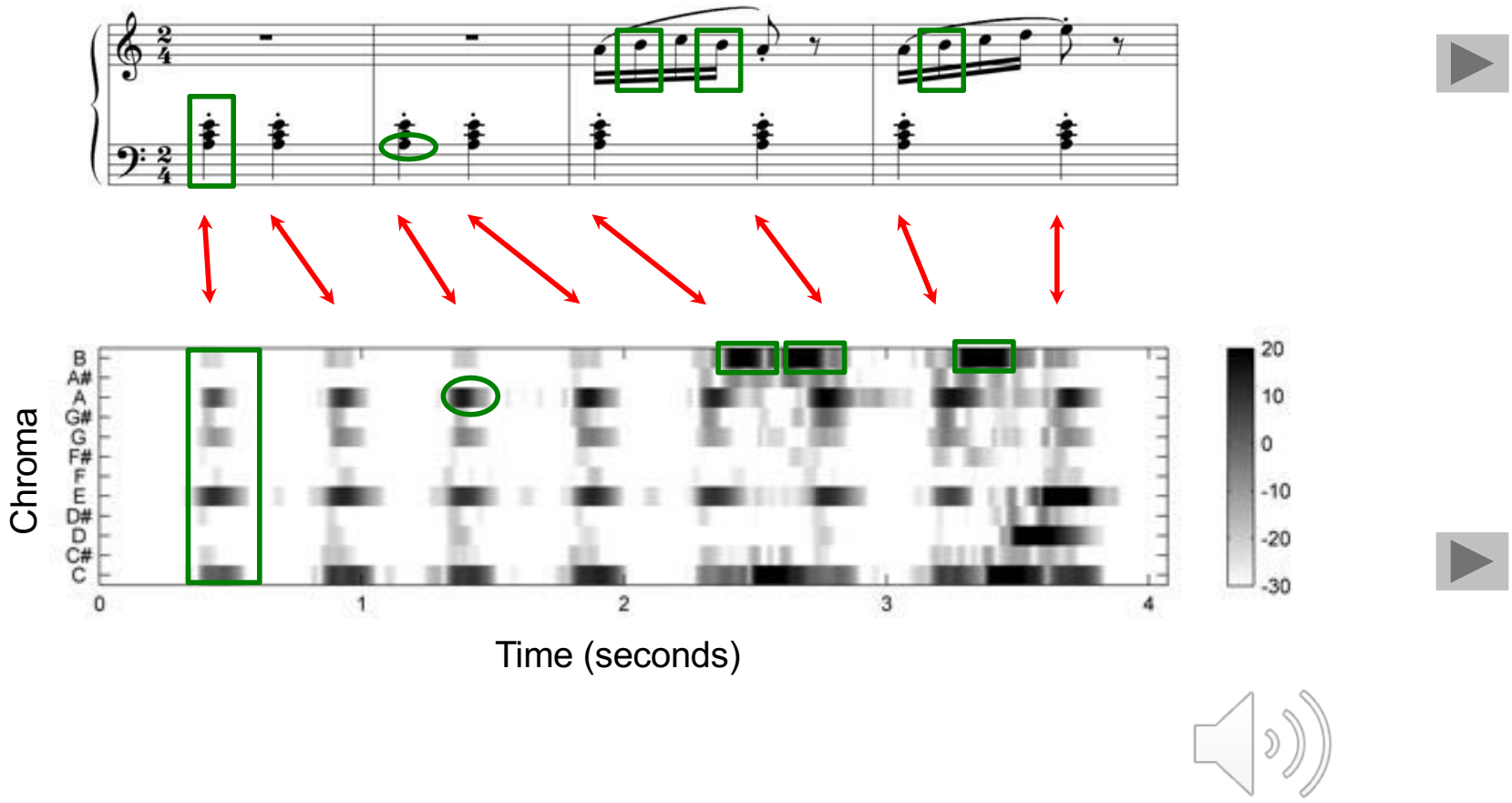
Audio Features

Chroma features



Audio Features

Chroma features



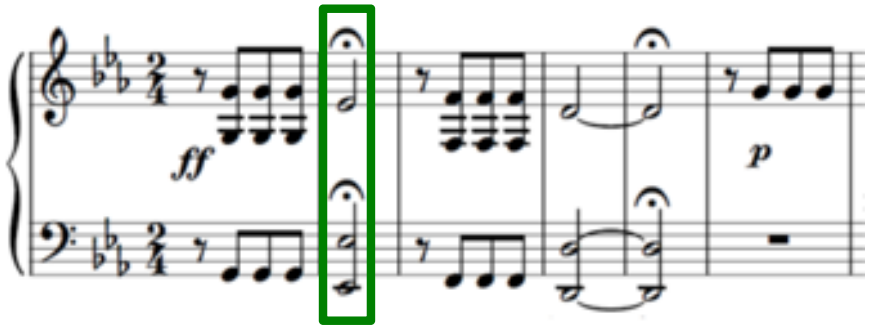
Audio Features

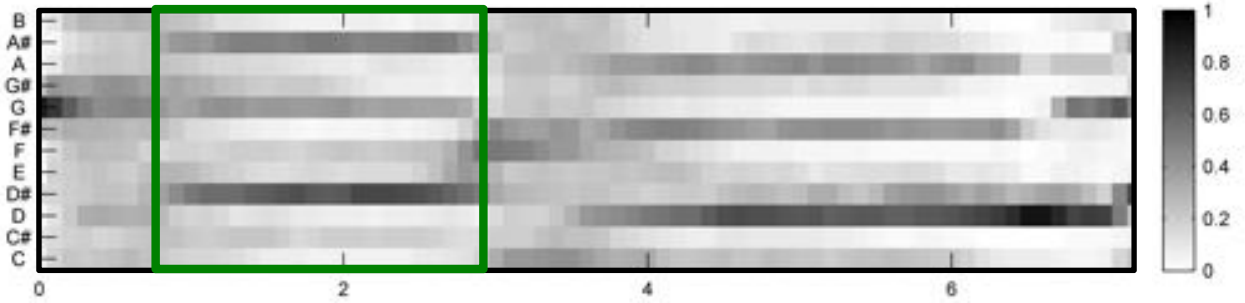
Chroma features

- Sequence of chroma vectors correlates to the harmonic progression
- Normalization $x \rightarrow x/\|x\|$ makes features invariant to changes in dynamics
- Further denoising and smoothing
- Taking logarithm before adding up pitch coefficients accounts for logarithmic sensation of intensity

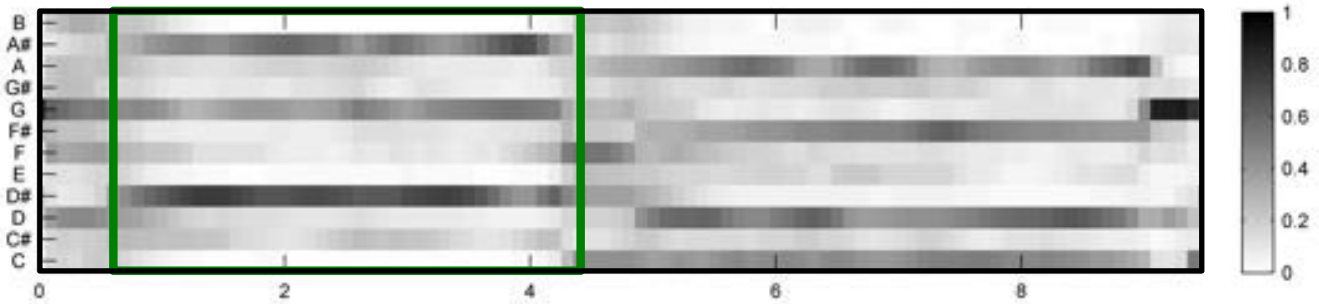
Audio Features

Chroma features (normalized)





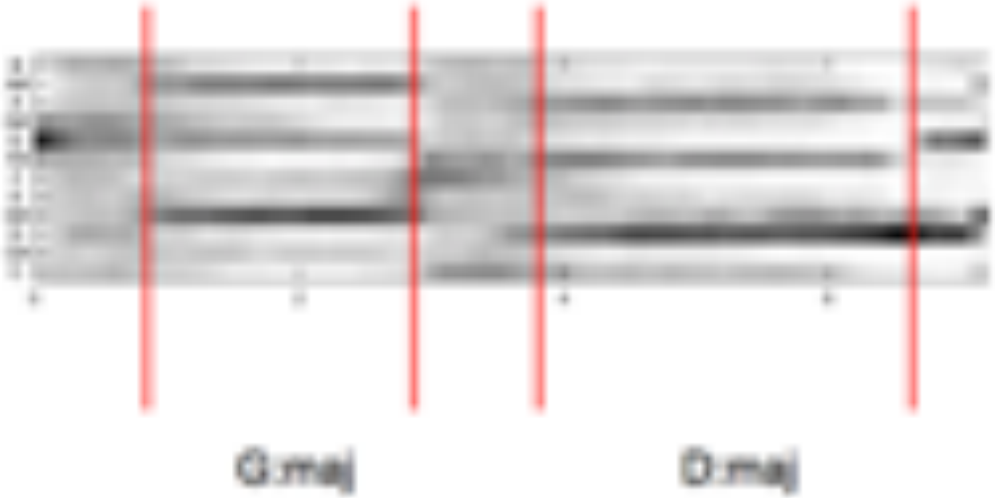
Karajan



Scherbakov

Audio Features

Automatic Chord Estimation



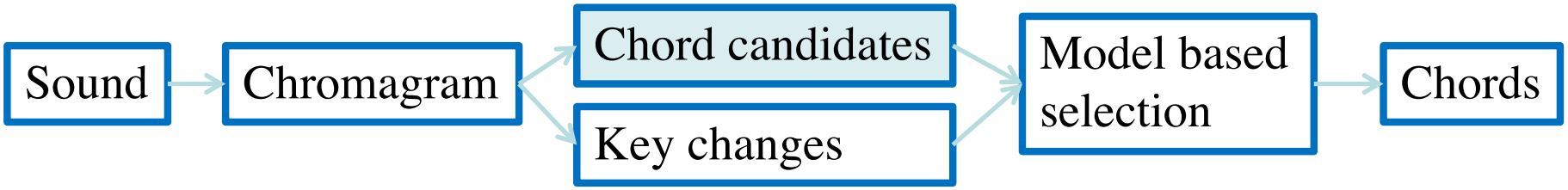
See: http://chordify.net

Audio Features

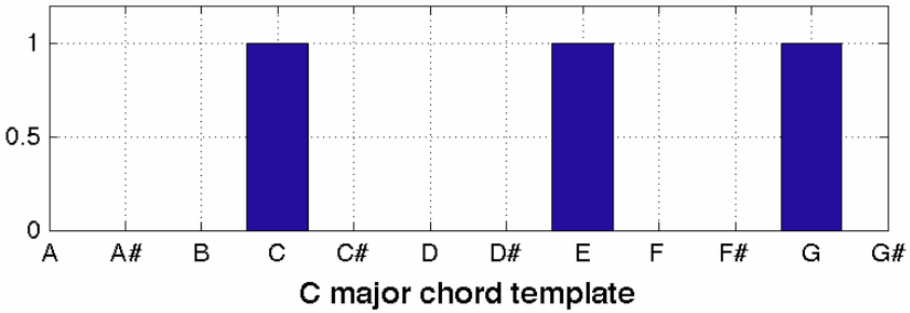
- There are many ways to implement chroma features
- Properties may differ significantly
- Appropriateness depends on respective application

- http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/
- MATLAB implementations for various chroma variants

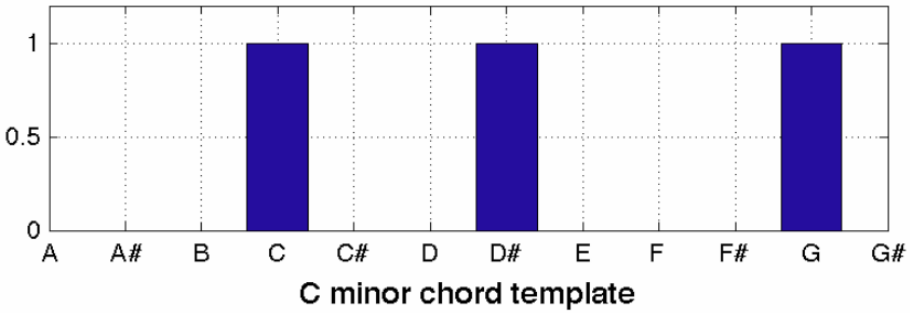
Case study: chord recognition



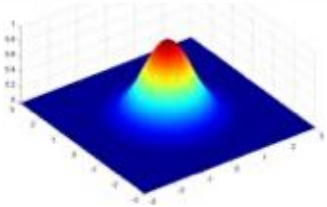
- Match the beat synchronised chroma features with chord templates
- Different approaches:
  - Use knowledge-based templates
  - Match by Euclidean distance
  - Learn an average profile from the data
  - Learn 12 dimensional Gaussian distribution for all chords



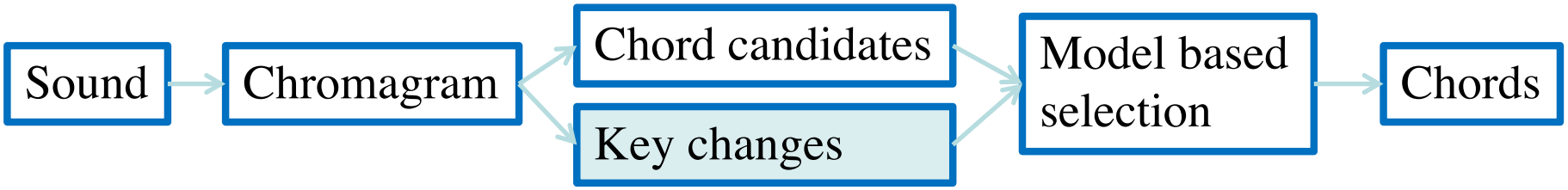
C major chord template



C minor chord template

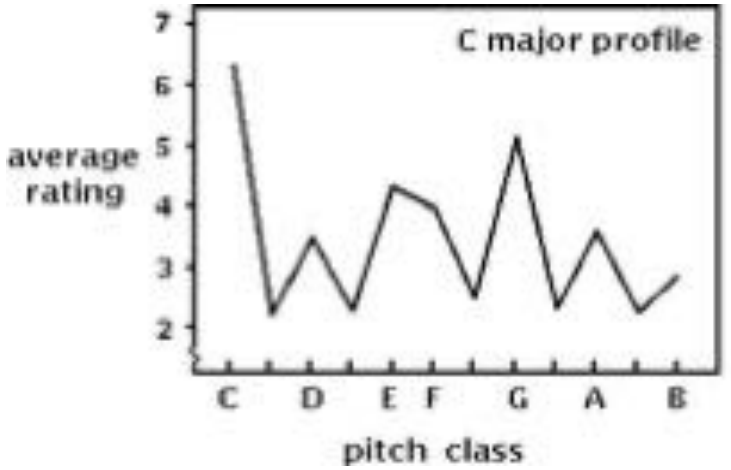


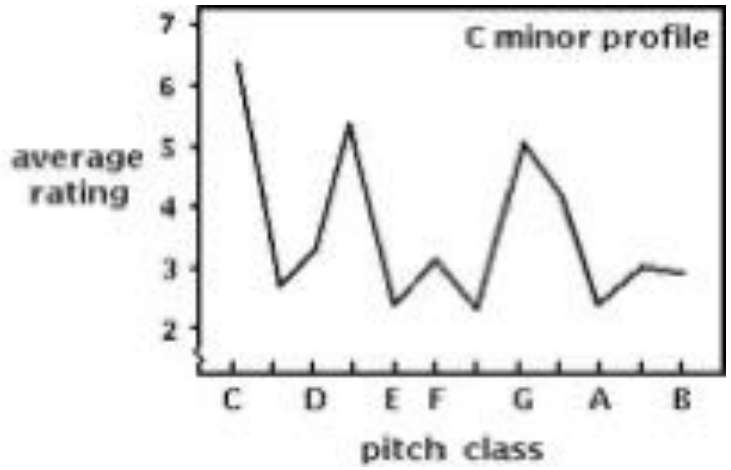
Case study: chord recognition



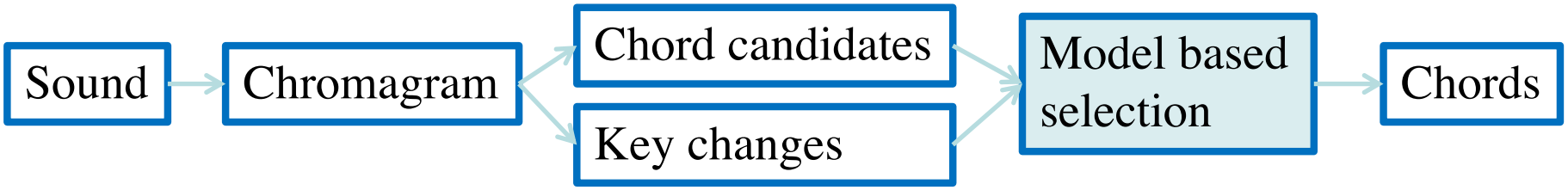
- Some models require information about the Key

- General approach
  - Krumhansl-Kessler profiles
  - Match by Pearson correlation with chroma feature
  - Variations exist
  - Approach is similar to chord candidate selection





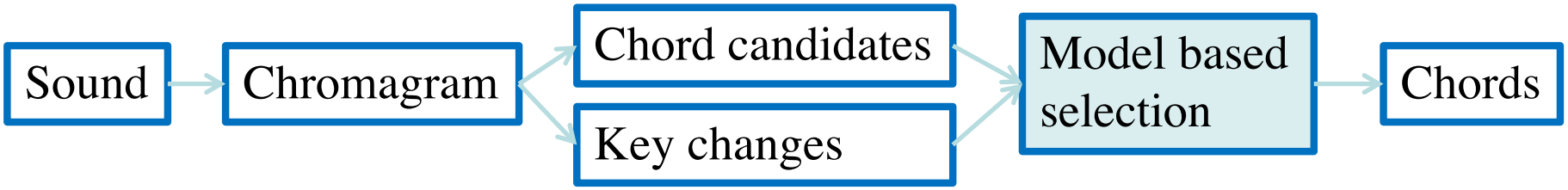
Case study: chord recognition



Two kind of approaches:

- Knowledge driven:
  - Musical knowledge is modelled and used to select a plausible chord sequence
- Data driven:
  - The Transition probabilities between chords are learned from a large corpus of chords

Case study: chord recognition

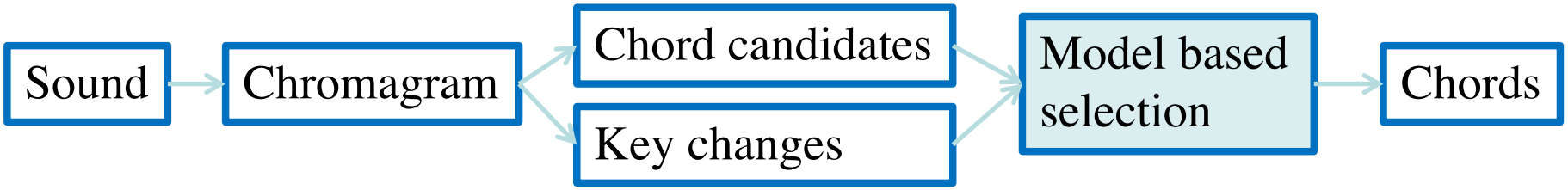


Knowledge driven:

HarmTrace (de Haas) model:

- Analyses the function of chord
- Needs Key information
- Robust against noisy data
- Flexible
- Based on functional programming

Case study: chord recognition

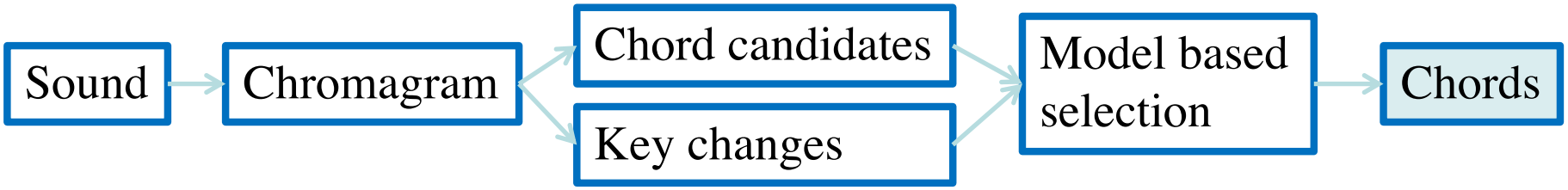


Data Driven:

Hidden Markov models:

- Estimate the probability of a chord-transition by counting the chord transitions in a corpus
- Use chroma features to estimate the probability of a chord candidate
- Goal: to find the most likely sequence of chords that results on the current chromagram:
  - Viterbi algorithm
- Many variants exist

Case study: chord recognition



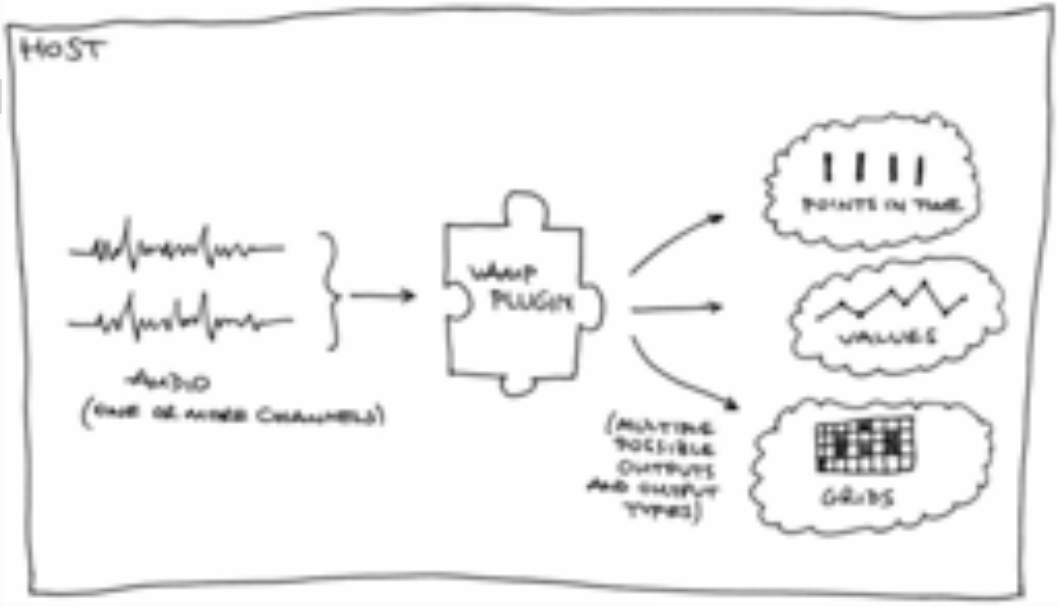
chordify®

Tool boxes

- Many tool boxes exist that can help you develop your own MIR application
- Toolboxes have their pro's and con's depending on:
  - Language of choice: Matlab, Python, C, C++, etc.
- I will only list the large frameworks, a lot of other code is available:
  - Use you favourite search engine!
- See also:
  - http://www.music-ir.org/evaluation/tools.html (rather outdated)
  - https://wiki.python.org/moin/PythonInMusic

VAMP plugins

- http://www.vamp-plugins.org/
- You need a *host* and a *plugin:*
  - Sonic visualizer
  - Sonic annotator
  - A lot of plugins for high and low-l extraction are available
  - Outputs text files or XML (RDF)
  - Has Python bindings (VamPy)

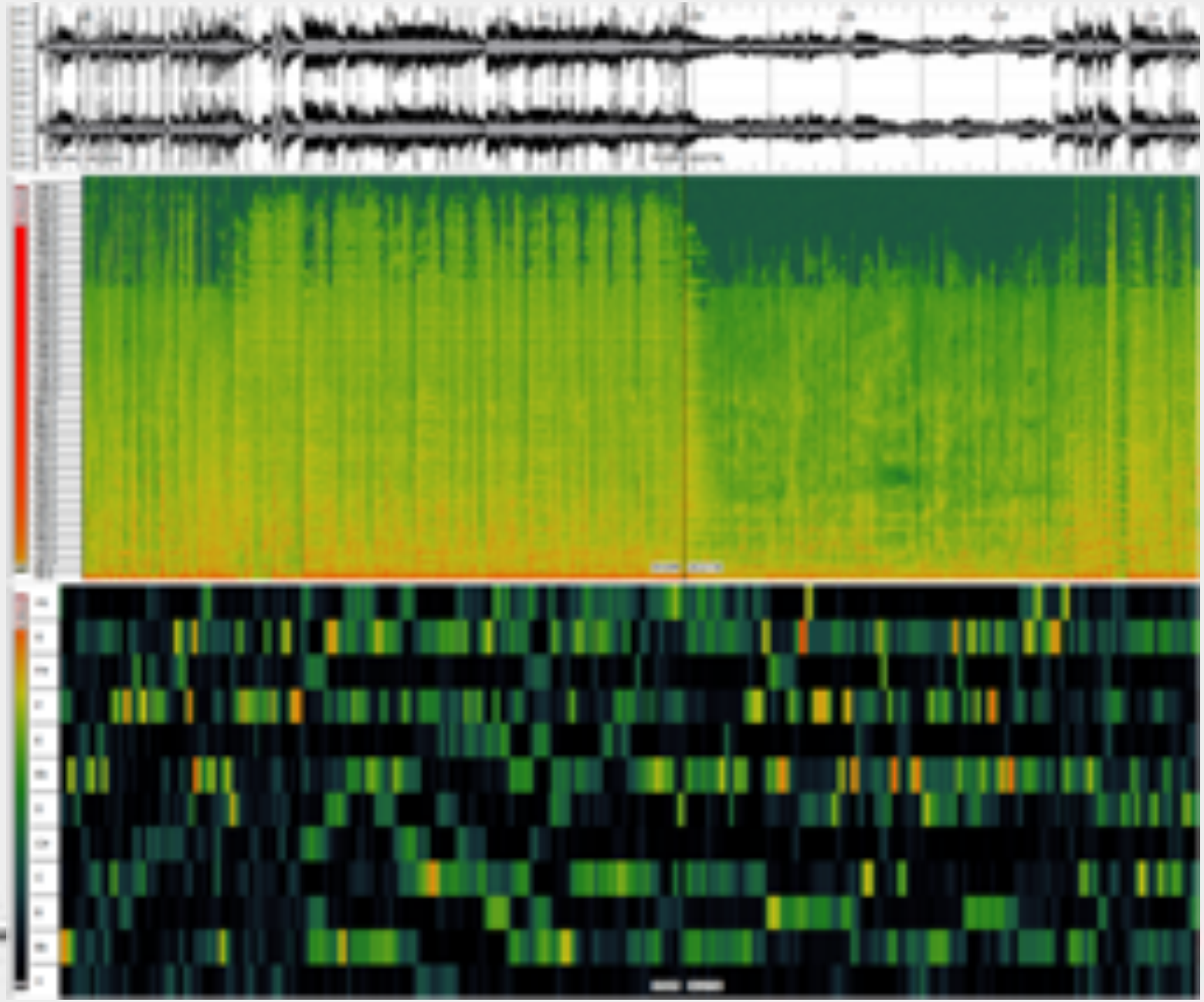


Sonic Visualizer

Waveform

Spectrogram

Chromagram



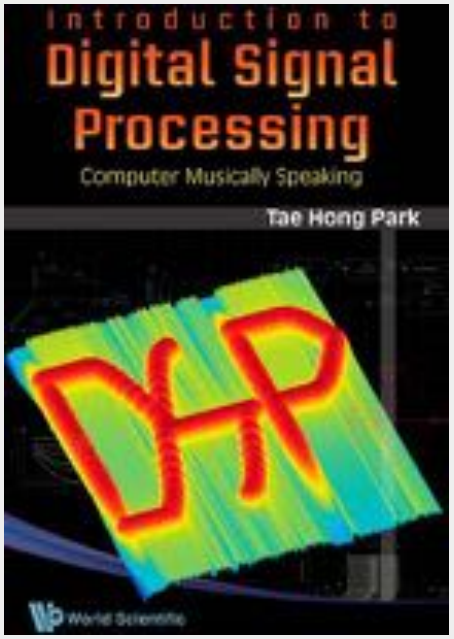
Matlab toolboxes

- MIR toolbox
  - https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox
- Similarity Matrix Toolbox:
  - http://www.audiolabs-erlangen.de/resources/MIR/SMtoolbox/
- Dan Ellis' Matlab resources:
  - http://www.ee.columbia.edu/~dpwe/resources/matlab/
- Chroma toolbox
  - http://www.mpi-inf.mpg.de/resources/MIR/chromatoolbox/
  - All sorts of different chroma variants
- Constant Q toolbox:
  - http://www.eecs.qmul.ac.uk/~anssik/cqt/

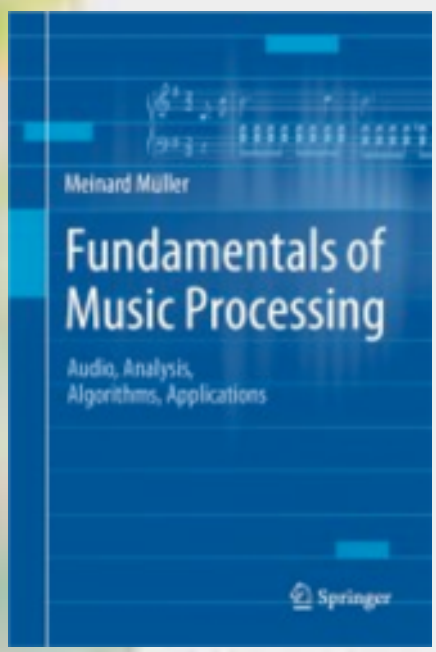
Other Frameworks

- Essentia:
  - http://essentia.upf.edu/
  - C++ library
- Marsyas:
  - http://marsyas.info/
  - C++ (and Java)
- Aubio:
  - http://aubio.org/
  - Written in C
  - Has Python bindings
- Clam
  - http://clam-project.org/
  - Written in C++
  - Has Python bindings
- Yaafe
  - http://yaafe.sourceforge.net/
  - Python

Additional Literature



Park, Tae Hong (2010). *Introduction to digital signal processing: Computer musically speaking*. World Scientific



Meinard Müller
Fundamentals of Music Processing
Audio, Analysis, Algorithms, Applications
ISBN: 978-3-319-21944-8
Springer, 2015

Summary

- Fourier analysis: analyses the frequency content of a signal
  - tells us which frequencies but not where
- STFT gives us an analysis over time
  - Trade-off between frequency and time resolution
- Different windowing functions give different frequency information

- For analysing music we prefer a log-frequency representation
- Chroma feature: Sum the spectral energy per octave
- Chroma features are suitable for chord and key detection
- For all analysis/features holds: not one single best solution, depends on what you want