2025 edition

Deep Learning for Music Analysis and Generation

Fundamentals for Musical Audio

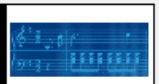
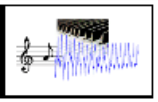
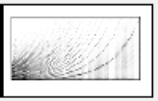
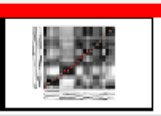
Yi-Hsuan Yang Ph.D.
yhyangtw@ntu.edu.tw

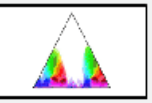
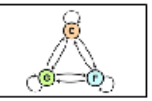
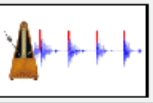
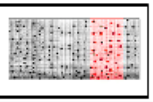
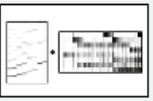
Outline

- Time and frequency representations for music
- Math in STFT

FMP Notebook

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1.html
https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2.html

Part	Title	Notions, Techniques & Algorithms	HTML	IPYNB
B 	Basics	Basic information on Python, Jupyter notebooks, Anaconda package management system, Python environments, visualizations, and other topics	[html]	[ipynb]
0 	Overview	Overview of the notebooks (https://www.audiolabs-erlangen.de/FMP)	[html]	[ipynb]
1 	Music Representations	Music notation, MIDI, audio signal, waveform, pitch, loudness, timbre	[html]	[ipynb]
2 	Fourier Analysis of Signals	Discrete/analog signal, sinusoid, exponential, Fourier transform, Fourier representation, DFT, FFT, STFT	[html]	[ipynb]
3 	Music Synchronization	Chroma feature, dynamic programming, dynamic time warping (DTW), alignment, user interface	[html]	[ipynb]

Part	Title	Notions, Techniques & Algorithms	HTML	IPYNB
4 	Music Structure Analysis	Similarity matrix, repetition, thumbnail, homogeneity, novelty, evaluation, precision, recall, F-measure, visualization, scape plot	[html]	[ipynb]
5 	Chord Recognition	Harmony, music theory, chords, scales, templates, hidden Markov model (HMM), evaluation	[html]	[ipynb]
6 	Tempo and Beat Tracking	Onset, novelty, tempo, tempogram, beat, periodicity, Fourier analysis, autocorrelation	[html]	[ipynb]
7 	Content-Based Audio Retrieval	Identification, fingerprint, indexing, inverted list, matching, version, cover song	[html]	[ipynb]
8 	Musically Informed Audio Decomposition	Harmonic/percussive separation, signal reconstruction, instantaneous frequency, fundamental frequency (F0), trajectory, nonnegative matrix factorization (NMF)	[html]	[ipynb]

Outline

- **Time and frequency representations for music**
- Math in STFT

Audio Waveforms

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3_Waveform.html

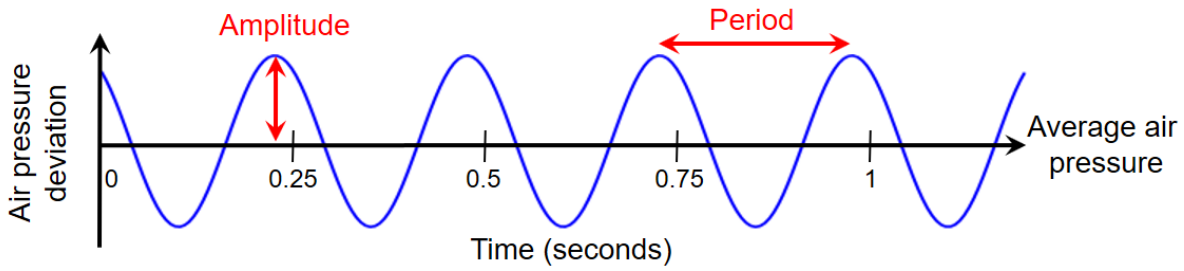
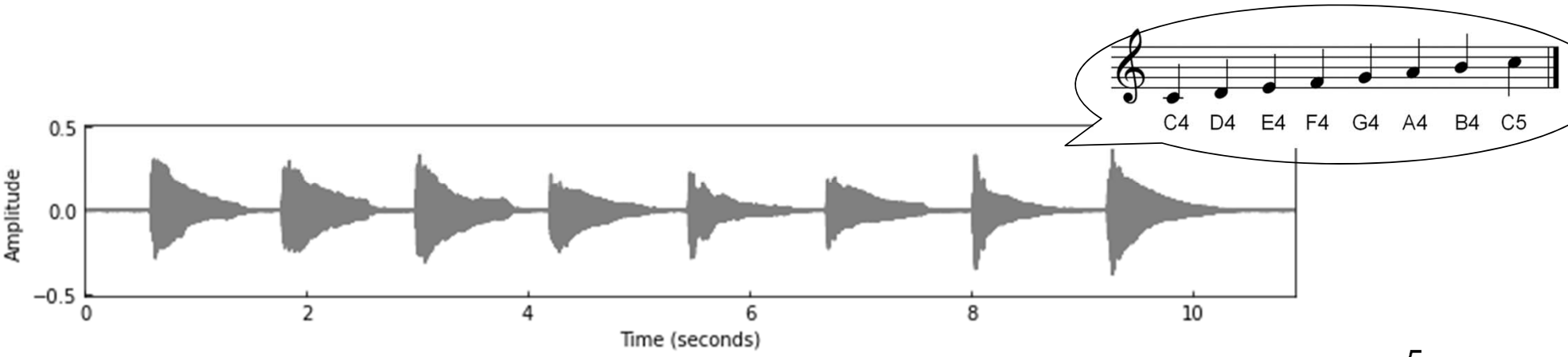


Figure 1.19 from [Müller, FMP, Springer 2015]



What Do We Perceive in Musical Audio?

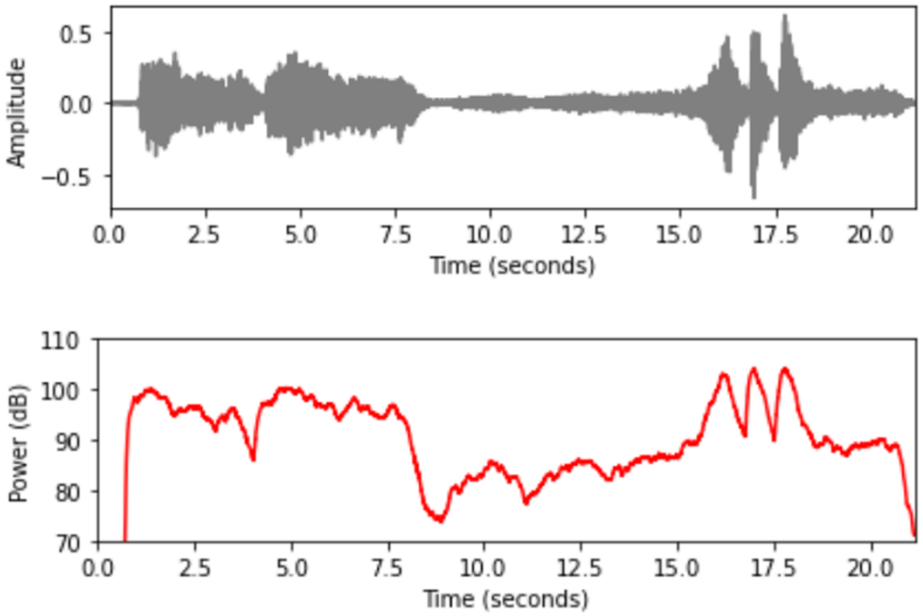
- Loudness
- Timbre / instrument / singer identity
- Pitch / melody
- Chord / harmony
- Tempo / rhythm
- Style
- Emotion
- Spatial information

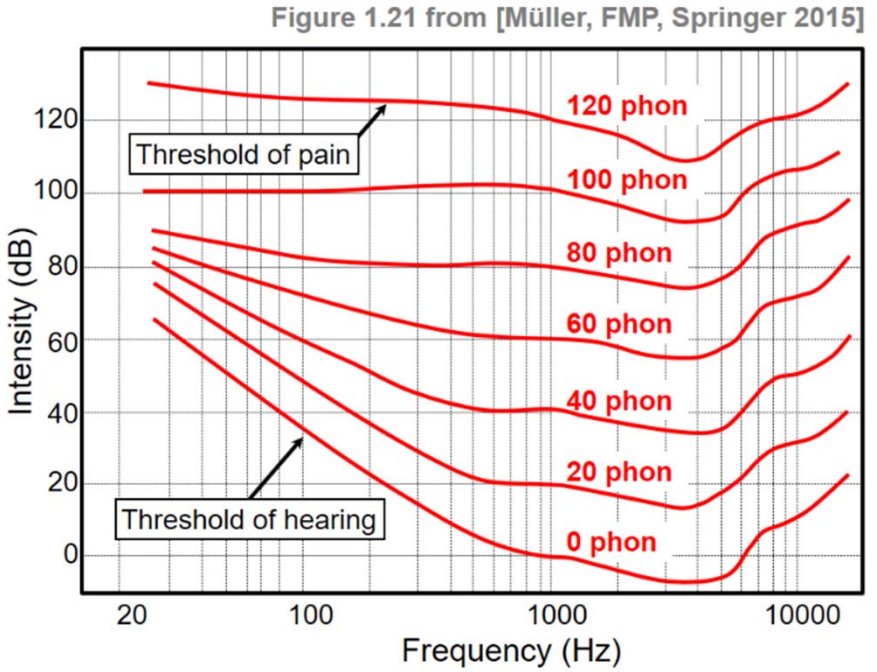
- Others (e.g., “The power of music compounds over time through repeated listening. Culture builds around shared experiences.”)

Dynamics, Intensity, and Loudness

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3_Dynamics.html

- **Intensity**: a physical property
  - Other names: power (dB); energy
- **Loudness**: a perceptual property
  - Less used in deep learning research





Timbre: Different Instruments

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3_Timbre.html

- **Timbre** (tone color; tone quality)
  - Allow usto distinguish the musical tone (i.e., a single musical note) of different instruments, even if the tone is played at the same **pitch** and with the same **loudness**
  - Hard to grasp & *subjective*
  - May be described with words such as *bright*, *dark*, *warm*, and *harsh*

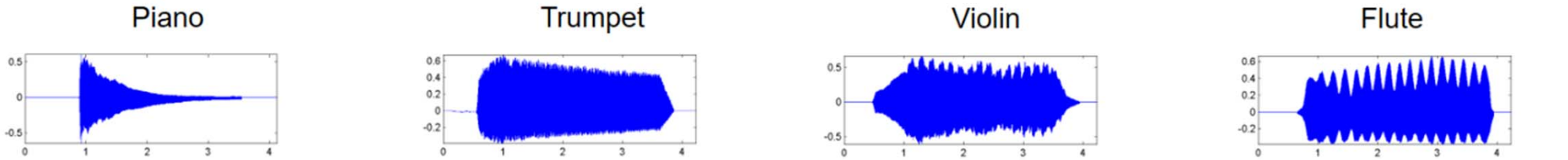
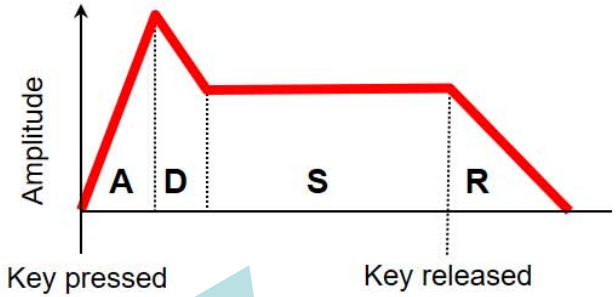


Figure 1.23 from [Müller, FMP, Springer 2015]

0:04 / 0:04 0:00 / 0:04 0:00 / 0:04 0:00 / 0:04

Temporal Characteristics

- **ADSR**: attack (A), decay (D), sustain (S), and release (R)
  - Hard to analyze when there are concurrent notes



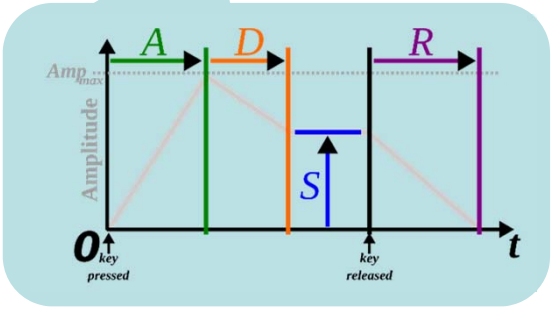
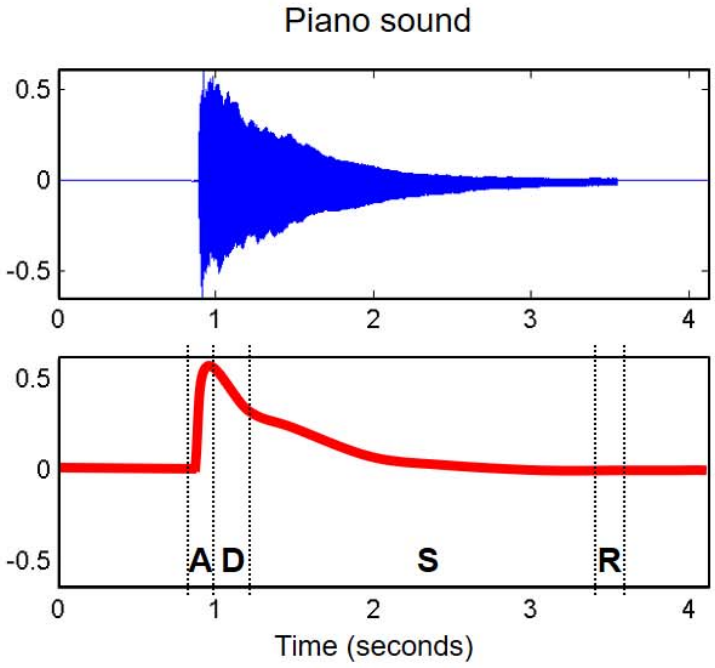


Figure from:
https://en.wikipedia.org/wiki/Envelope_(music)



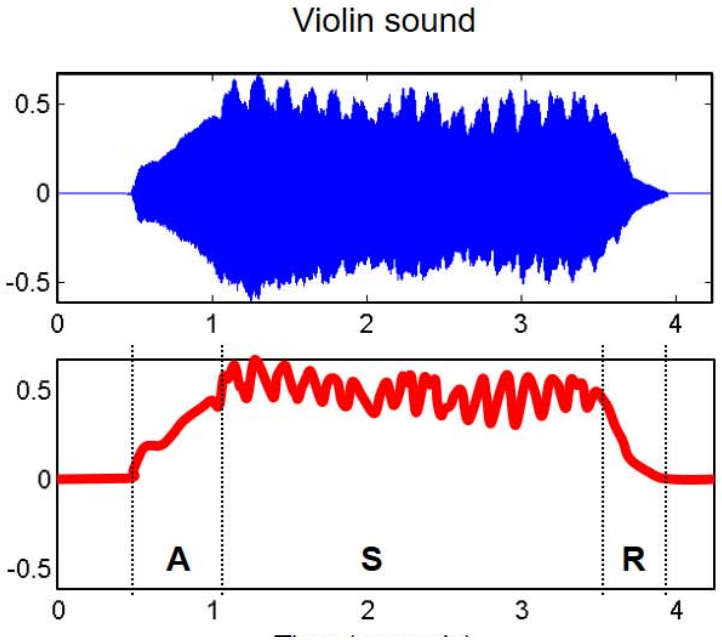
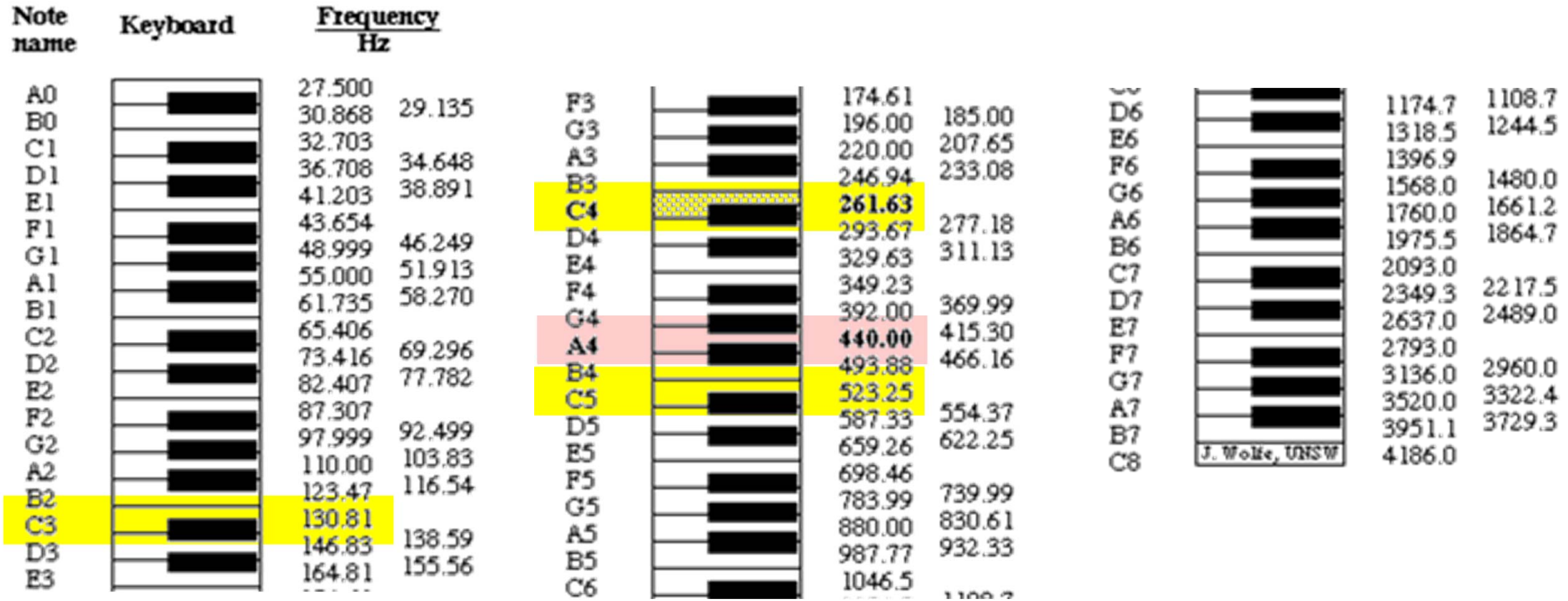


Figure 1.22b and Figure 1.23
from [Müller, FMP, Springer 2015]

Notes and Pitches

(Assuming A4=440 Hz; but there are *other tunings*)

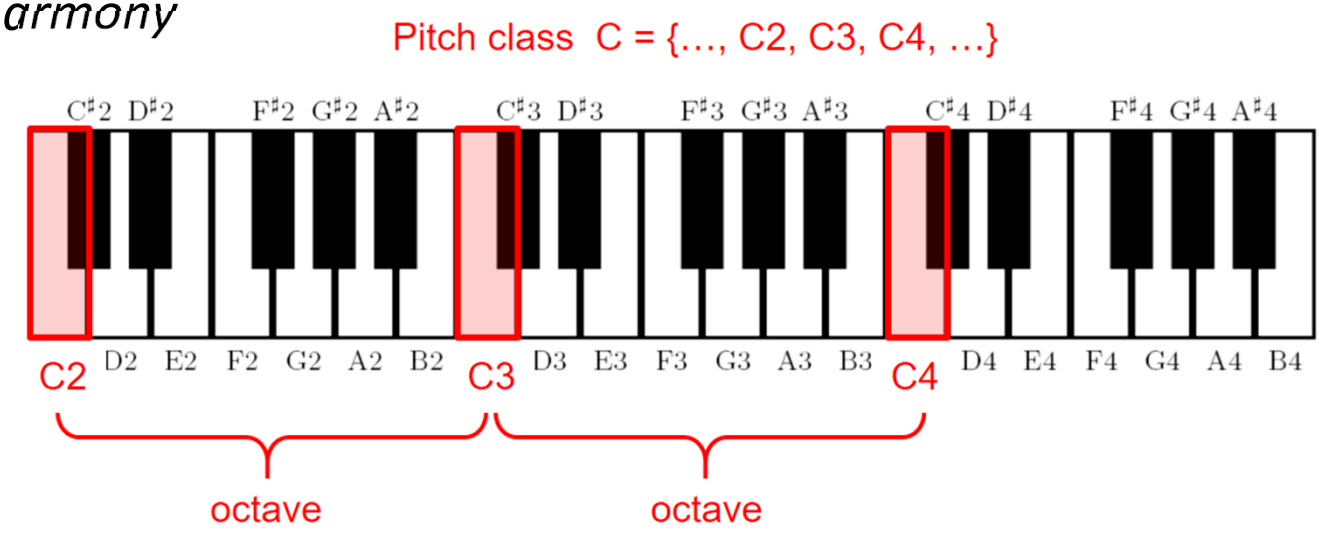
https://newt.phys.unsw.edu.au/jw/notes.html



Notes and Pitches

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S1_MusicalNotesPitches.html

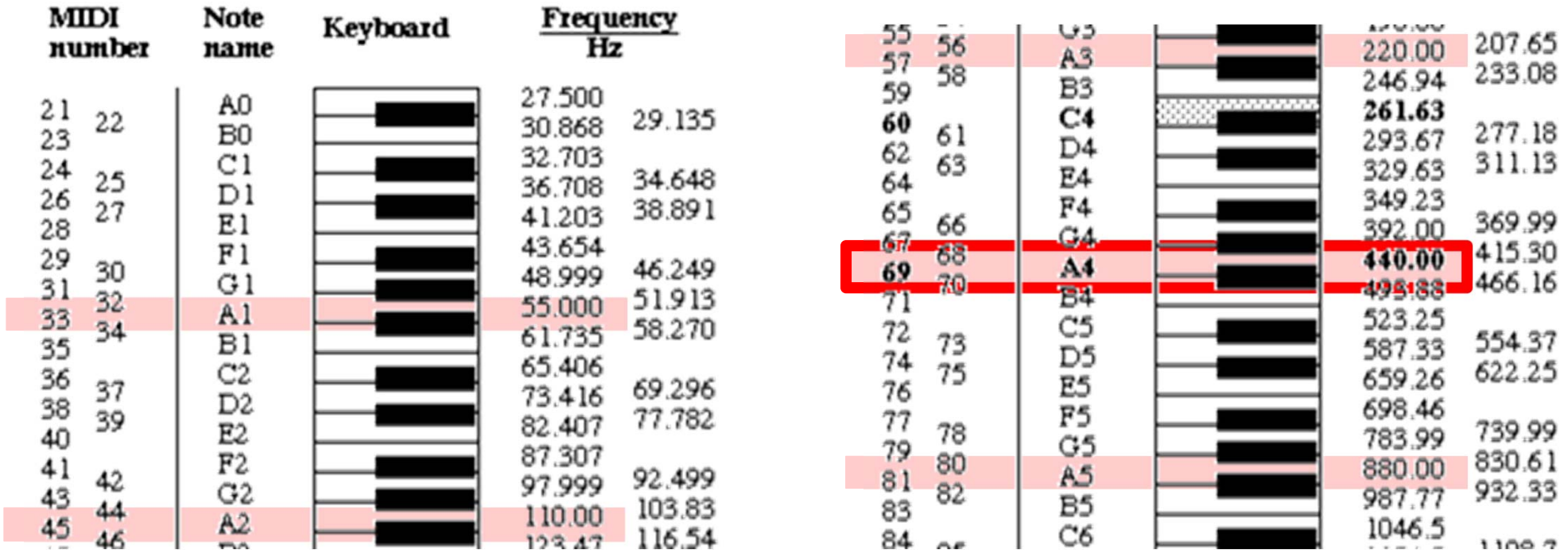
- **Pitch Class**
  - Two notes with *fundamental frequencies* in a ratio equal to *any power of* ***two*** (e.g., half, twice, or four times) are perceived as very **similar**
  - All notes with this kind of relation can be grouped under the same *pitch class*
  - Related to *chords/harmony*



MIDI Note Numbers

https://en.wikipedia.org/wiki/MIDI_tuning_standard

$$f = 2^{(d-69)/12} \cdot 440\ \text{Hz} \qquad d = 69 + 12\log_2\left(\frac{f}{440\ \text{Hz}}\right)$$

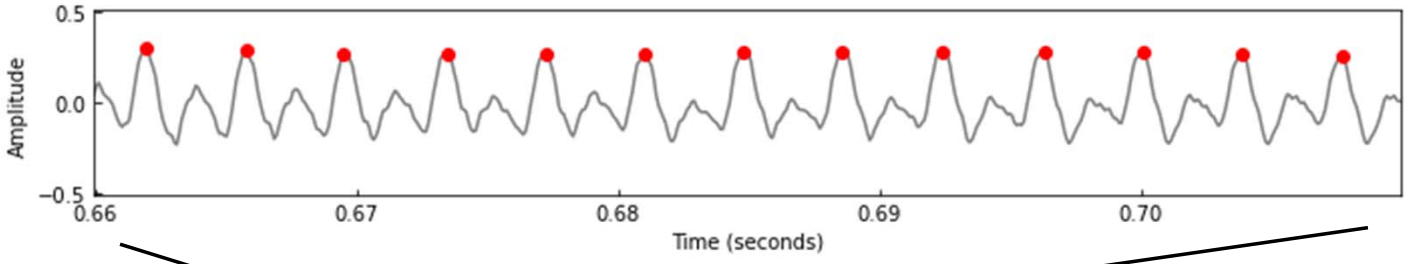


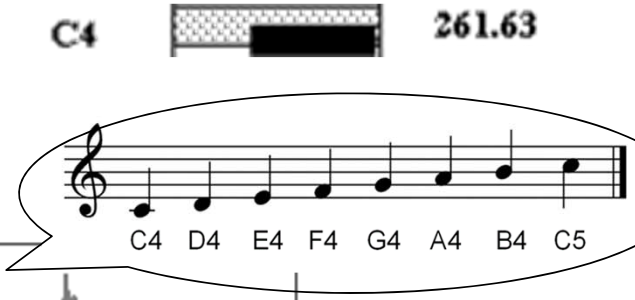
Audio Waveforms (Cont')

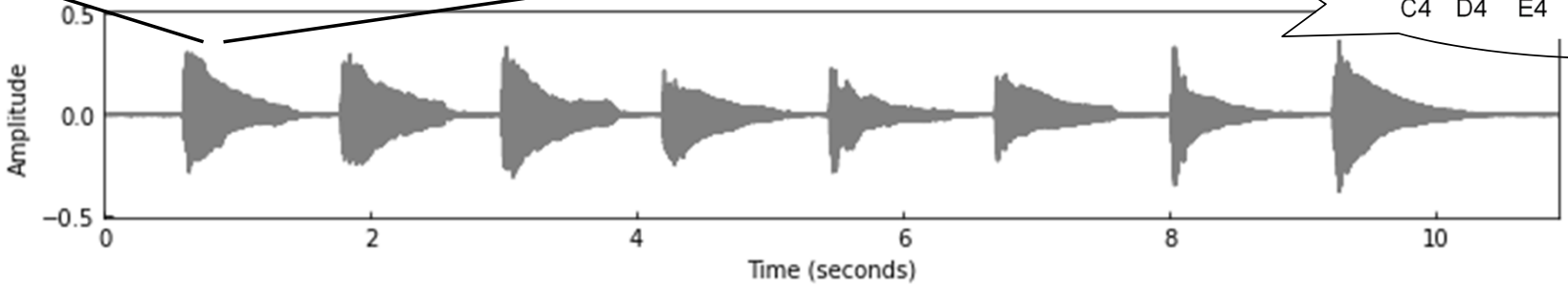
https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3_Waveform.html

Zoom-in of the section between 0.66 and 0.71 seconds

- Highly repetitive
- 13 high-pressure (red) points → 20*13=260 Hz

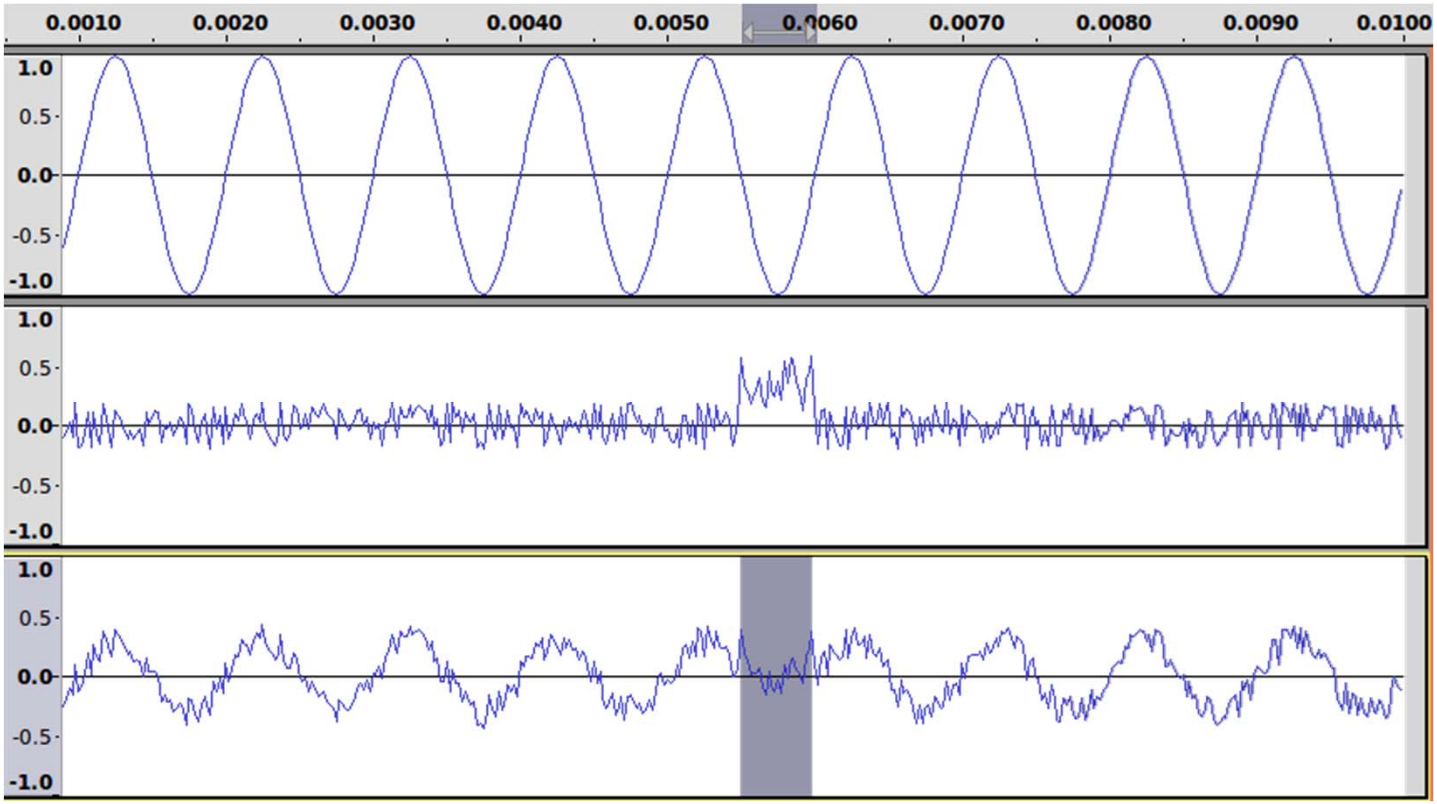






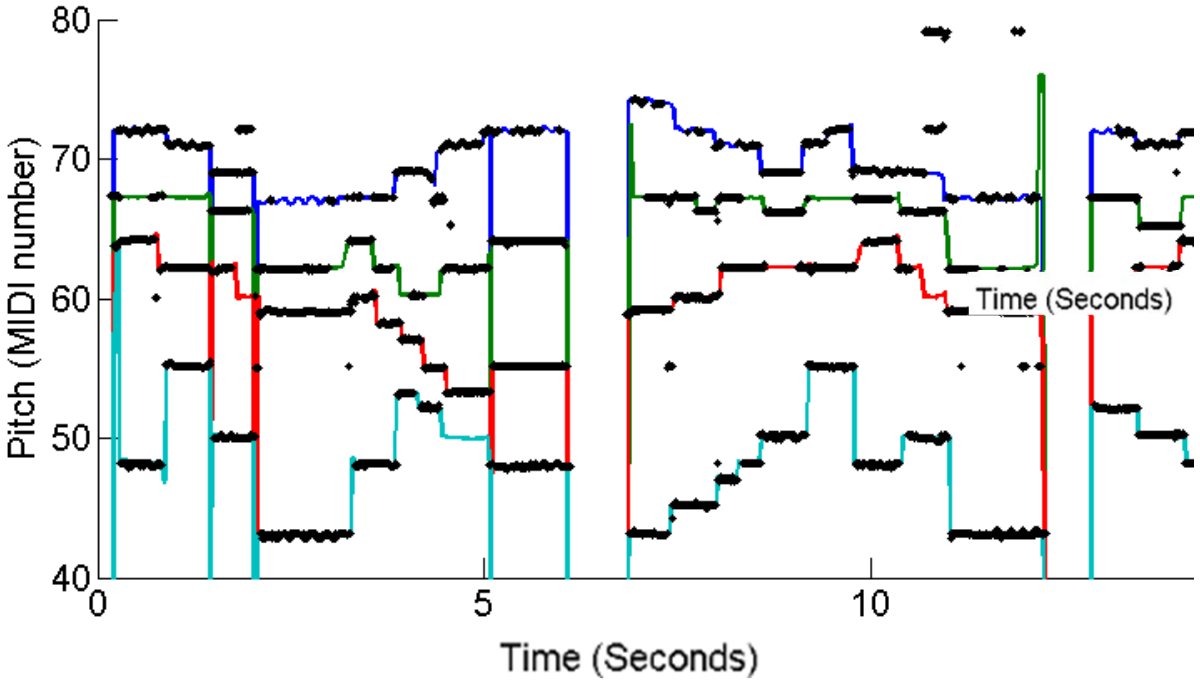
Temporal Characteristics

- **Zero-crossing rate**
  - How often a signal changes its sign
  - **Nosie**-like or not
  - Can be used in detecting the presence of vocals in speech signals
  - Cannot be used for estimating pitch due to noises and concurrent notes

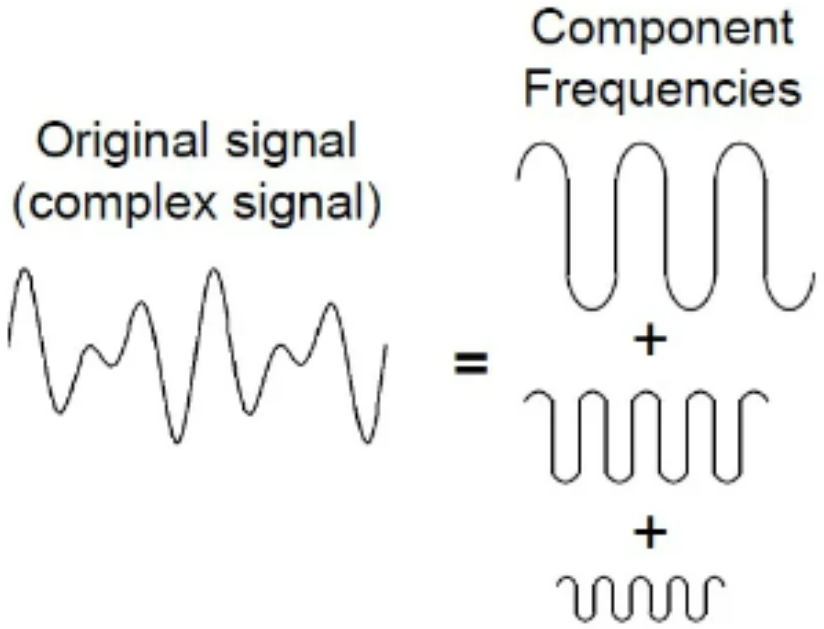


https://sandilands.info/sgordon/impact-of-noise-on-sine-wave-compared-to-square-wave

Multiple Concurrent Pitches



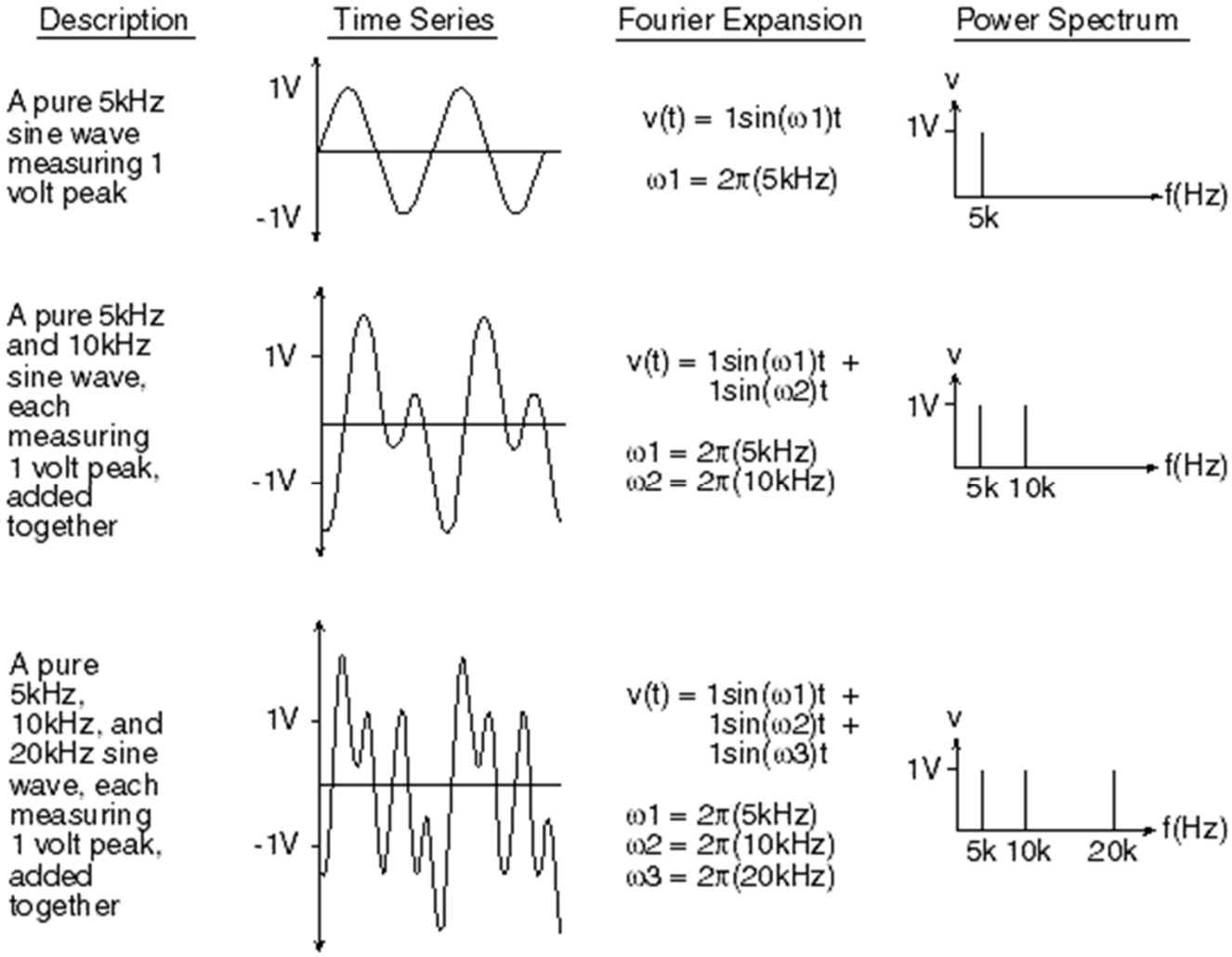
https://labsites.rochester.edu/air/projects/multipitch/multipitch.html



https://www.allaboutcircuits.com/technical-articles/an-introduction-to-filters/

Frequency-domain Analysis

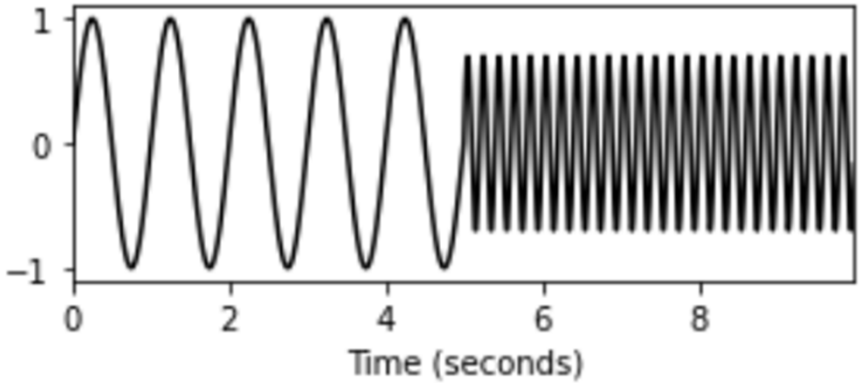
- Main subject of "Signals and Systems"
- **Fourier Transform**

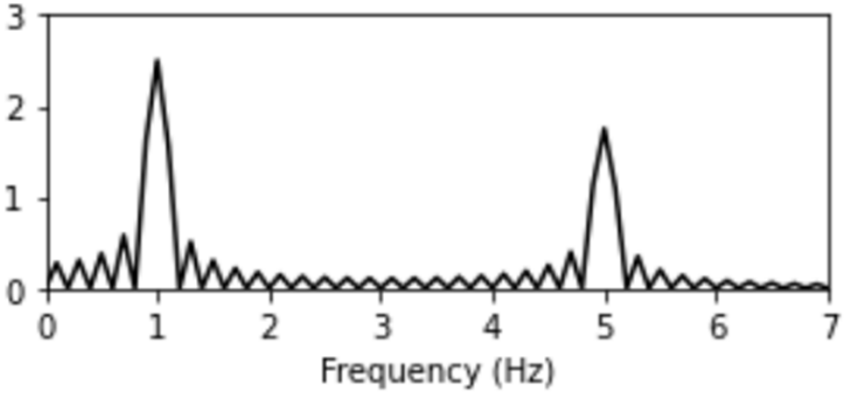


Frequency-domain Analysis

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html

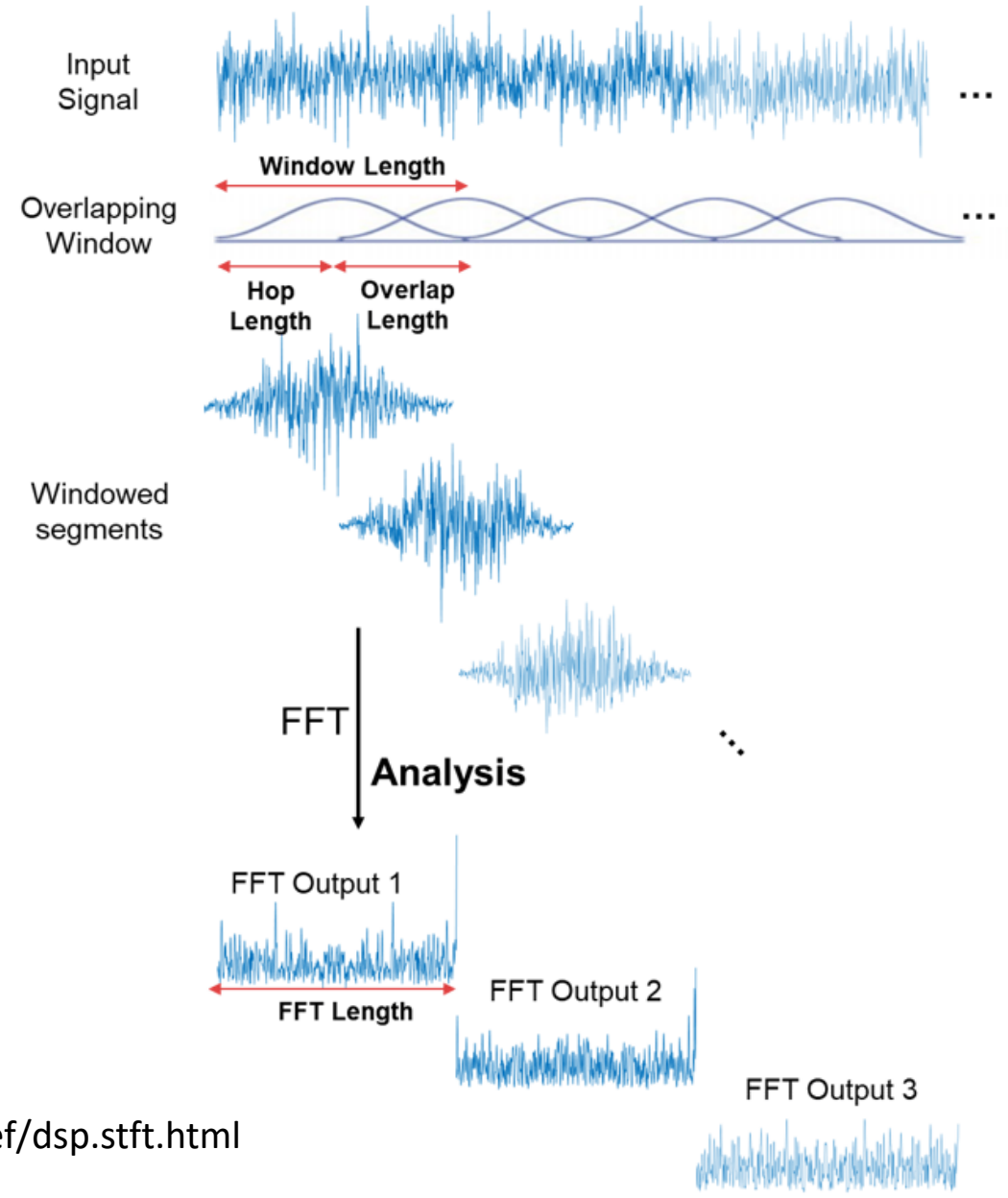
- **Fourier Transform cannot localize temporal events**





Frequency-domain Analysis

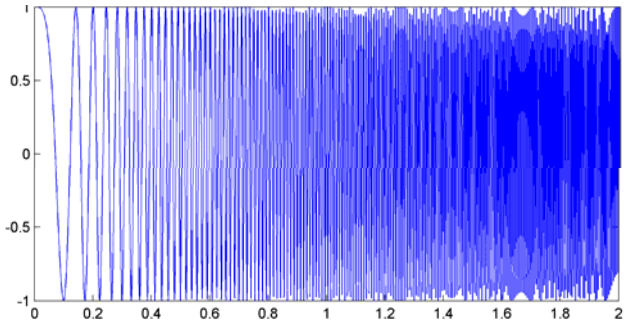
- Fourier Transform (FFT) cannot localize temporal events
- **Short-Time Fourier Transform** (STFT)
  - *Time-frequency representation* of signals
  - Slide **short windows** over the input signal and then compute the FFT locally for each windowed signal
- "**Window**": a short chunk of time



https://ww2.mathworks.cn/help/dsp/ref/dsp.stft.html

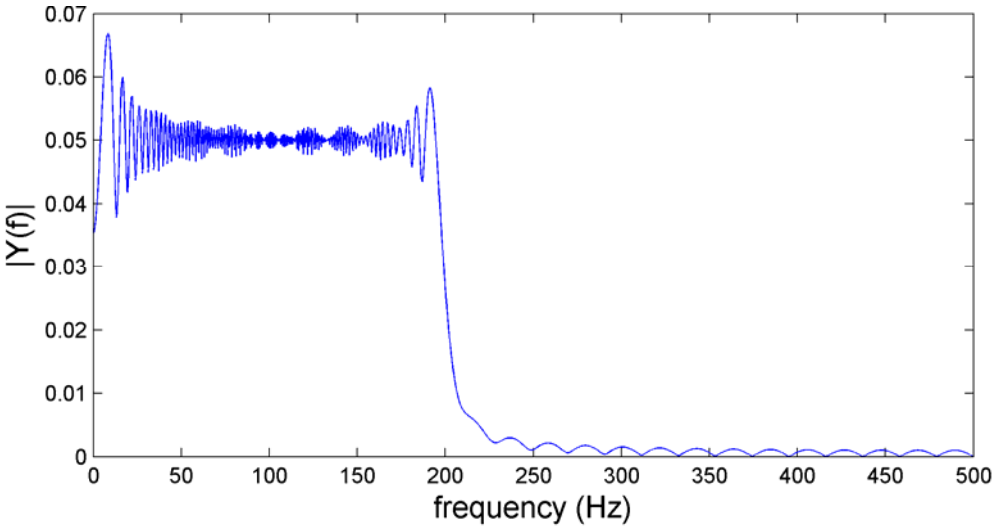
FFT vs STFT

Time-domain **waveform**

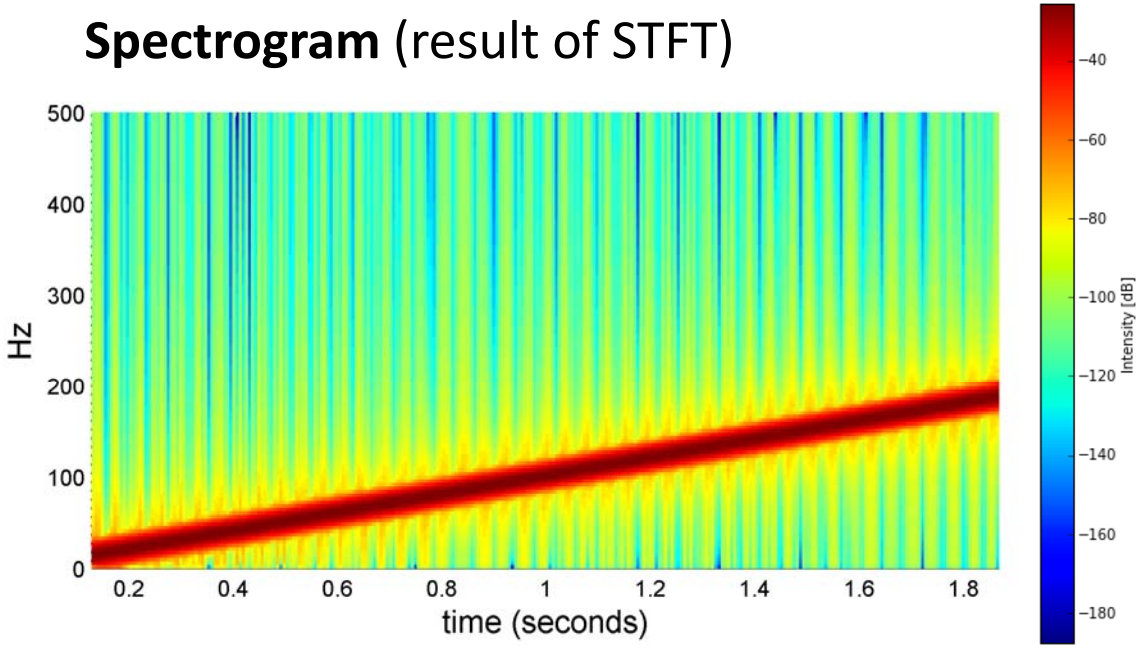


- **Waveform**: amplitude vs time
- **Spectrum**: amplitude vs frequency
- **Spectrogram**: time vs frequency; use *color* for amplitude

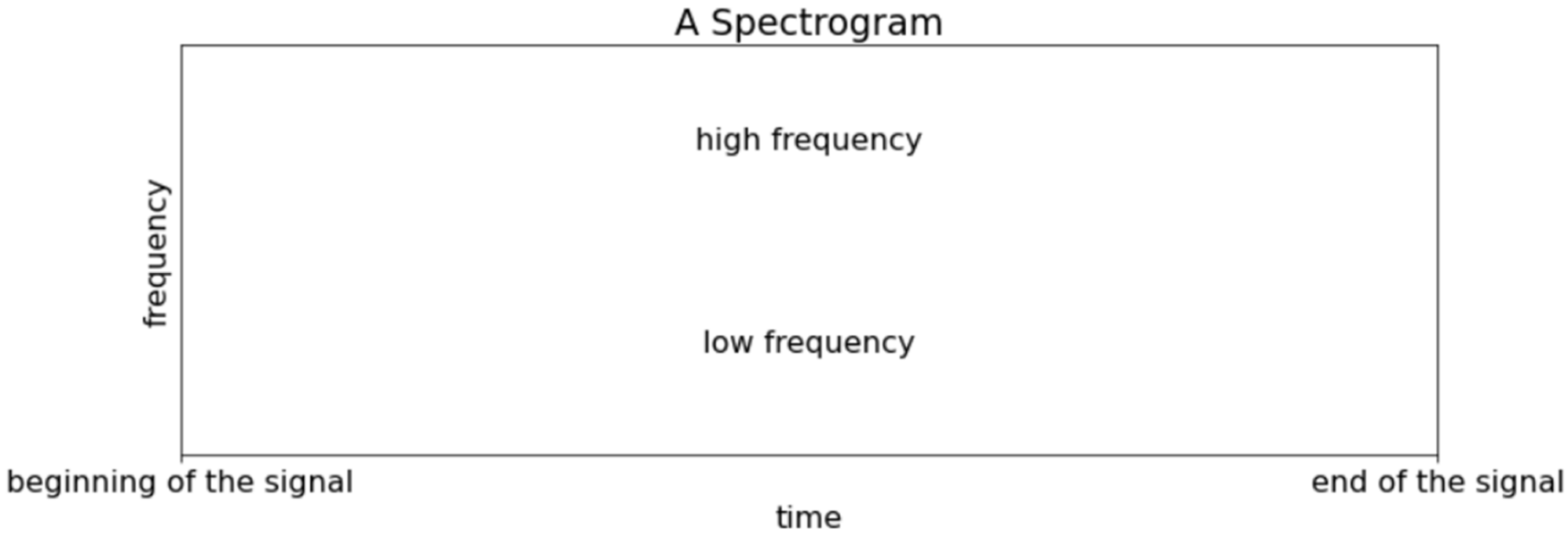
Spectrum (result of FFT)



Spectrogram (result of STFT)



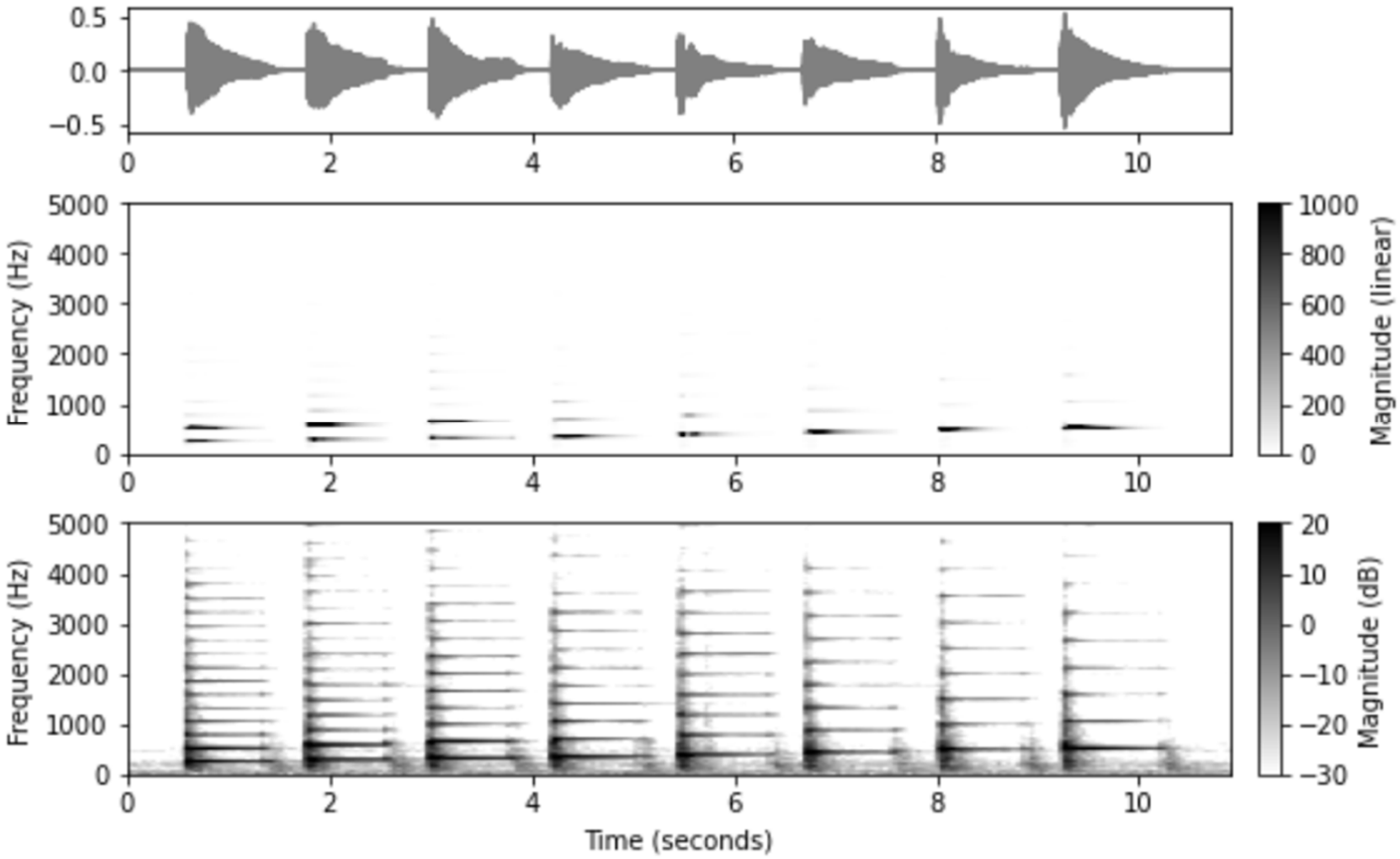
Time-Frequency Analysis via STFT



https://music-classification.github.io/tutorial/part2_basics/input-representations.html

Time-Frequency Analysis via STFT

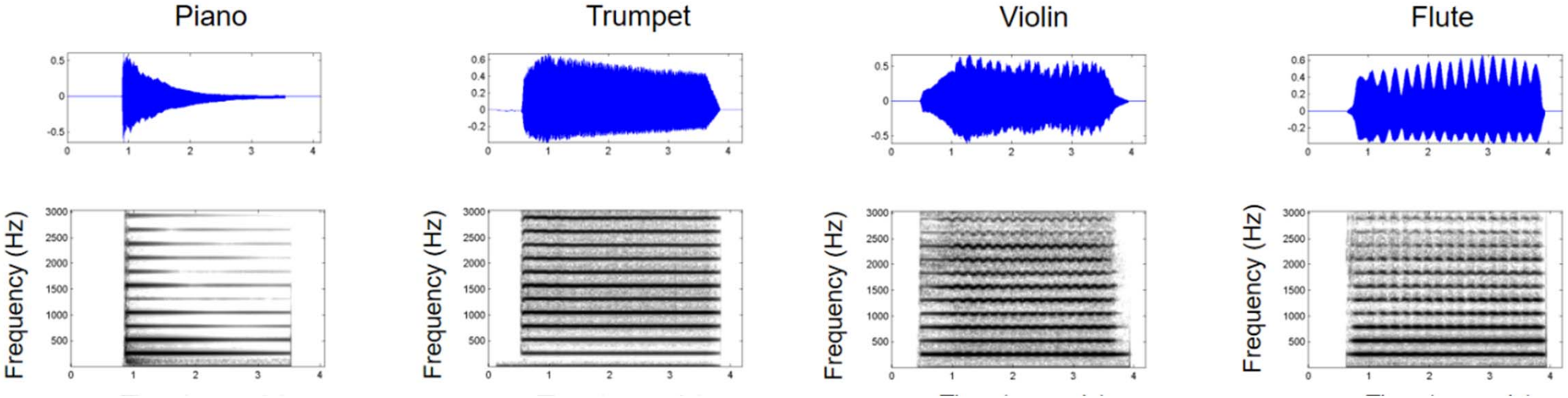
https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html



- **Time-frequency representation** of audio
  - Linear frequency, **linear** magnitude
  - Linear frequency, **logarithmic** magnitude (dB)

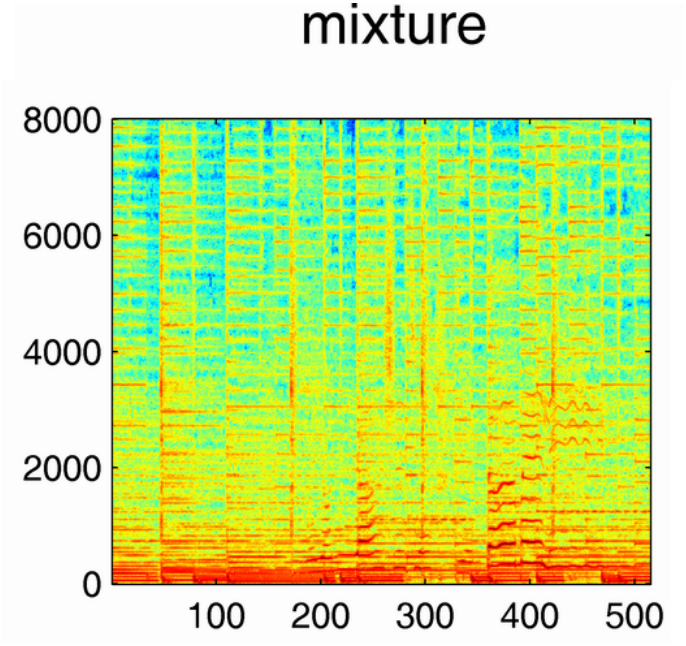
Spectral Characteristics

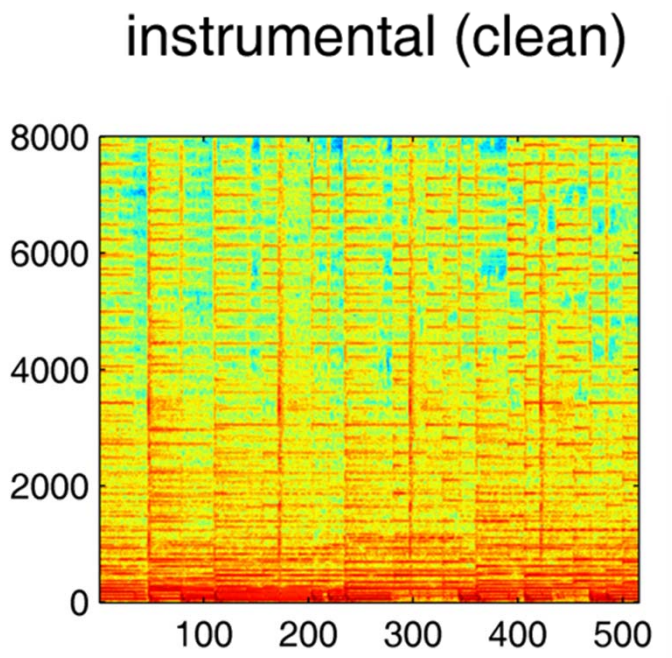
- Fundamental frequency (**F0**)
- **Partials** (harmonics): usually at frequencies which are **integer multiples** of F0
  - The partials in different instruments may exhibit **relative strengths**
  - The frequency may *deviate* from the ideal harmonics (this is called **inharmonicity**)

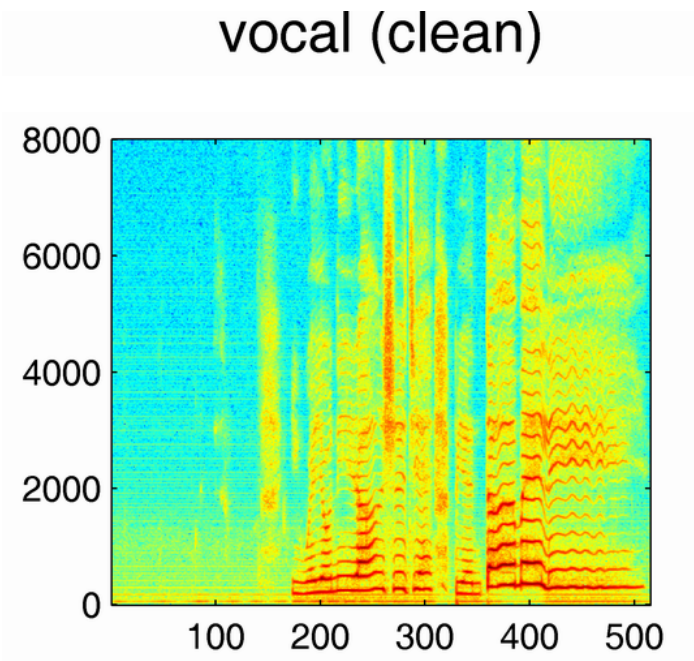


Time-Frequency Analysis via STFT

- For tasks such as classification/detection and source separation

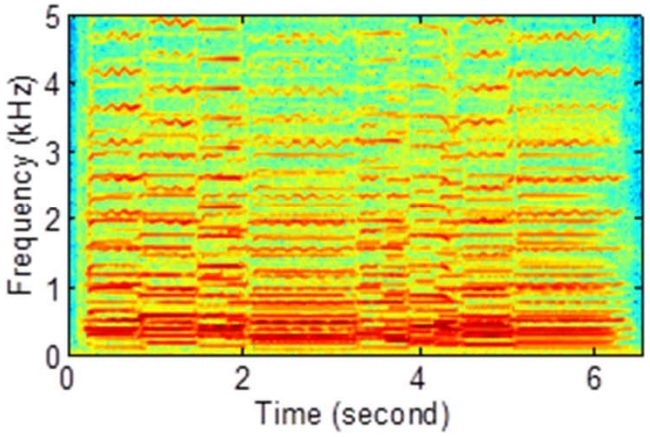




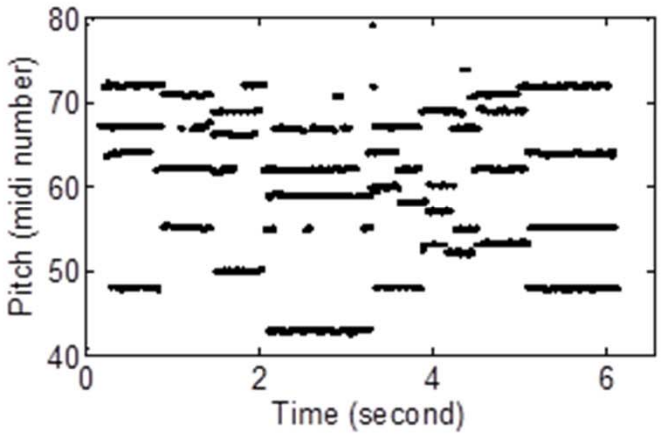


Time-Frequency Analysis via STFT

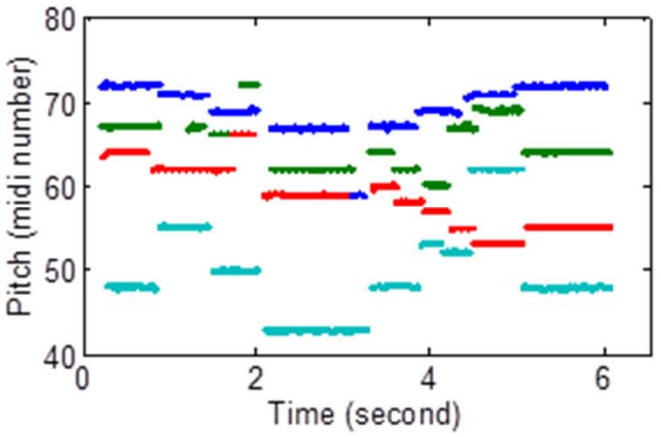
- For tasks such as multi-pitch estimation (MPE) and transcription



Spectrogram



Multi-pitch estimation results

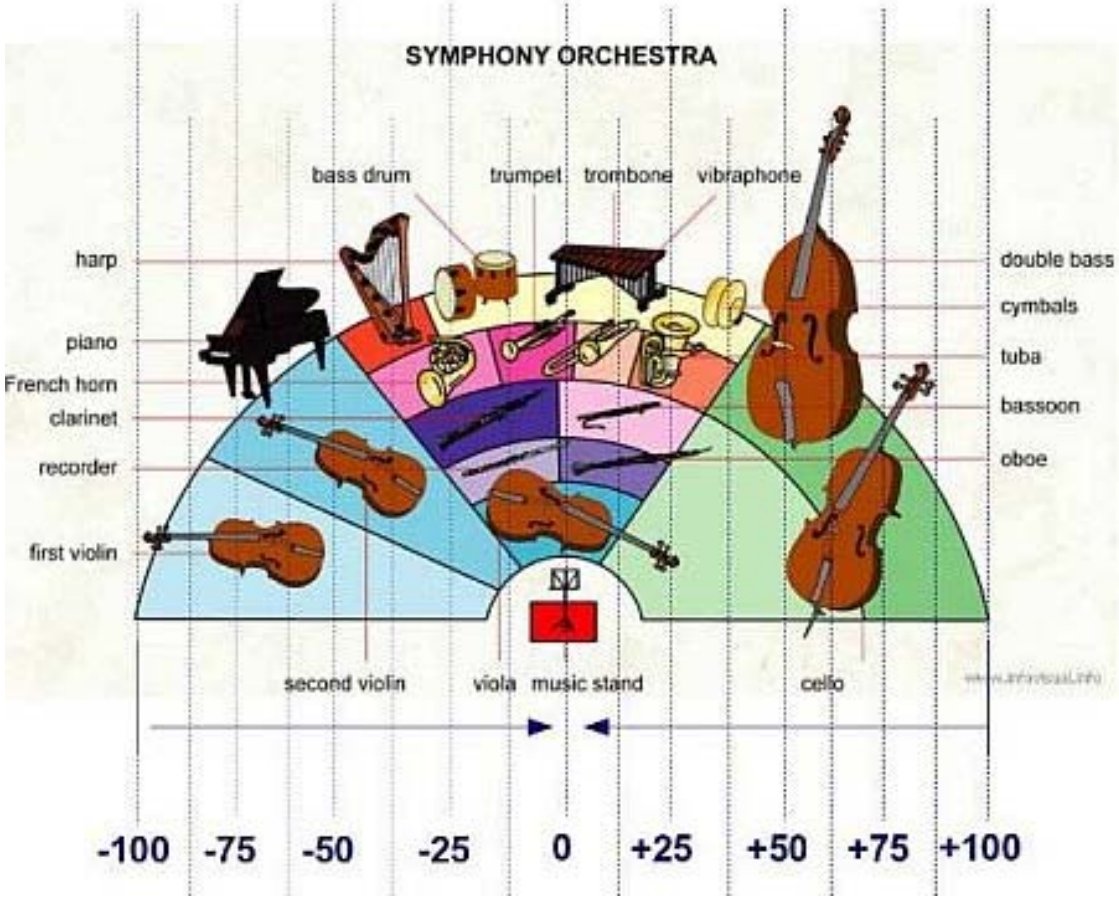


MPE + instrument detection

https://labsites.rochester.edu/air/projects/multipitch/multipitch.html

Spatial Information

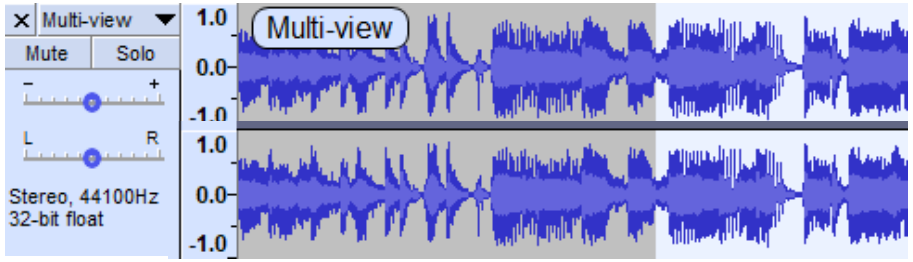
https://www.renegadeproducer.com/audio-panning.html

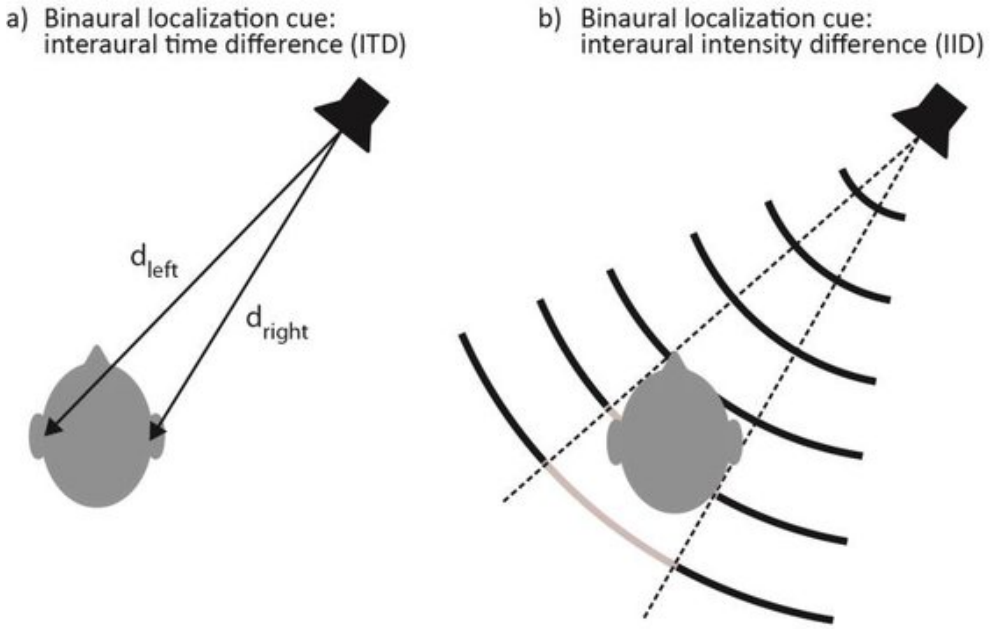


https://www.audiorecording.me/symphony-orchestra-panning-and-reverb-settings.html

Spatial Information: Mono vs Stereo

- Monaural
- Binaural



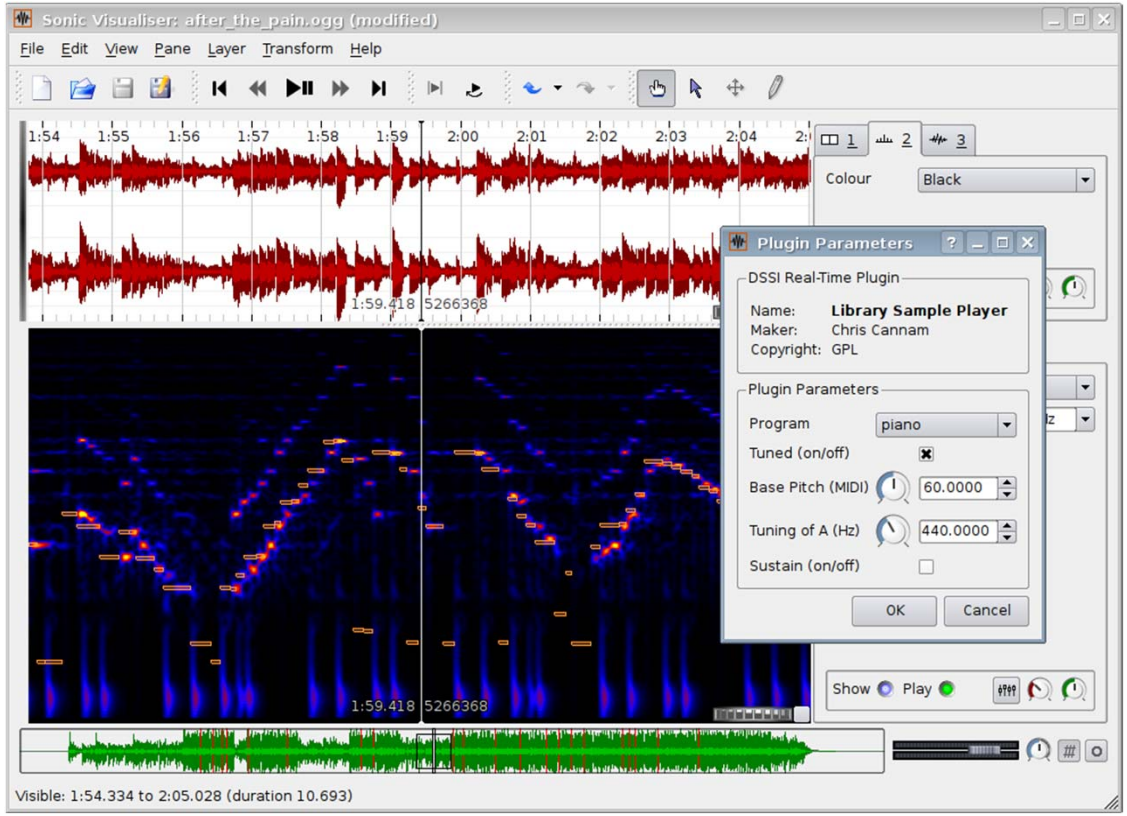


Source: https://www.researchgate.net/figure/Binaural-and-monaural-cues-used-for-sound-localization-a-For-sound-sources-off-the_fig1_299281975

- Stereo-to-mono conversion
  - By taking the average
- Mono-to-stereo conversion
  - Need research
    https://www.youtube.com/watch?v=aWxmQKm_s8Q

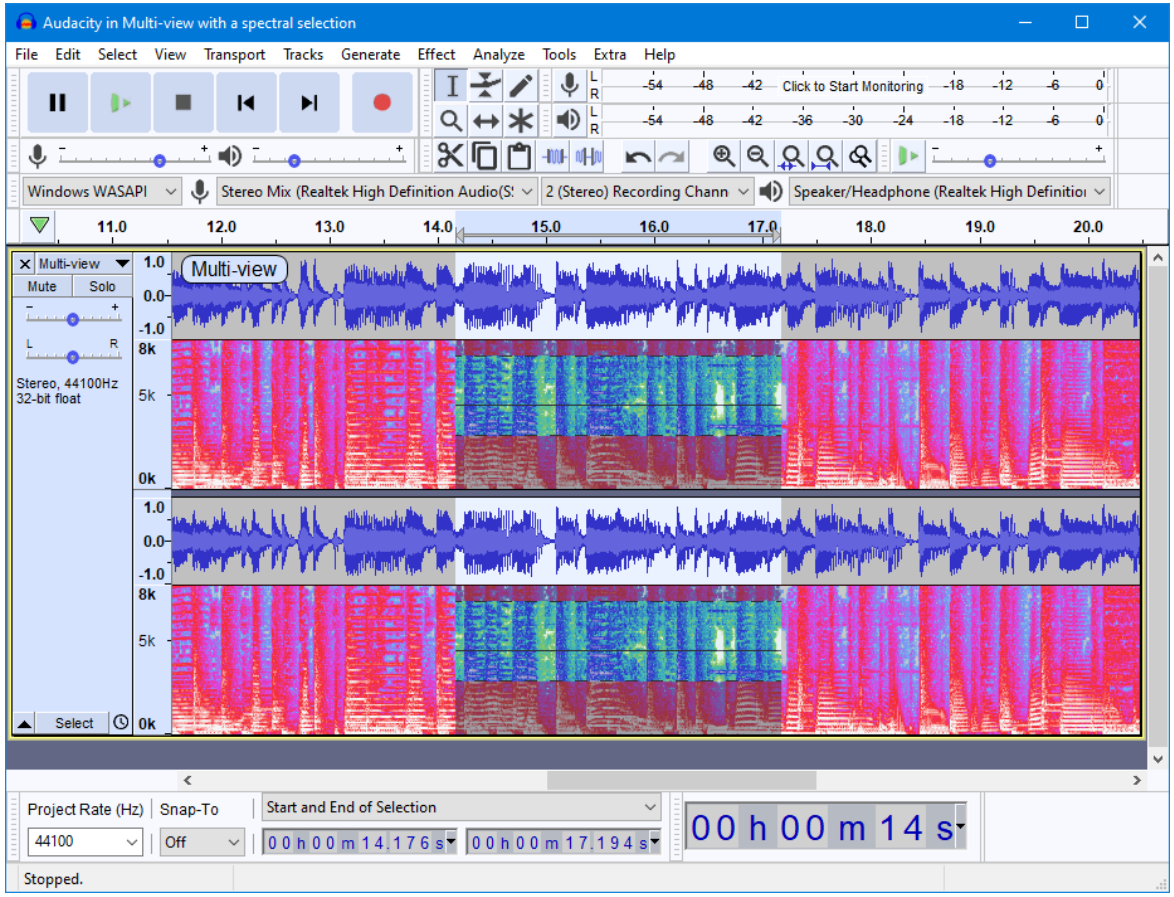
Software: Sonic Visualiser

http://www.sonicvisualiser.org/



Software: Audacity

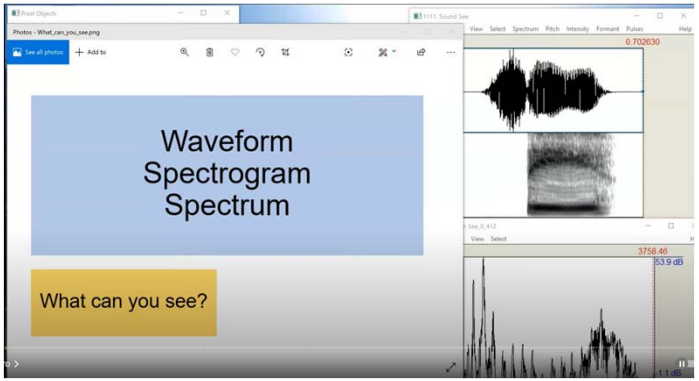
https://www.audacityteam.org/



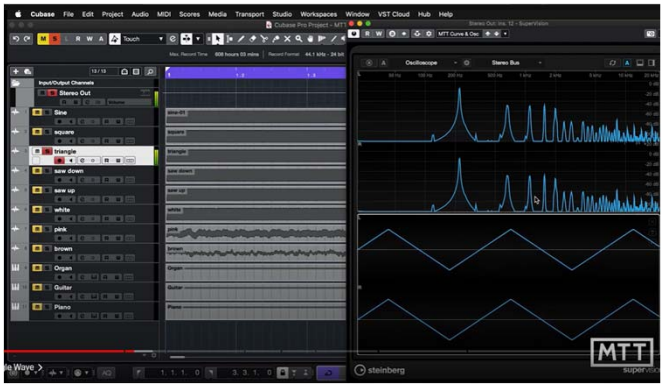
Let's Find Some Audio Recordings

- https://www.freesound.org/
  - https://www.freesound.org/people/acclivity/sounds/22347/
  - https://www.freesound.org/people/Rudmer_Rotteveel/sounds/316915/
  - https://www.freesound.org/people/Jaylew1987/sounds/321112/
  - https://www.freesound.org/people/mickel11/sounds/90803/

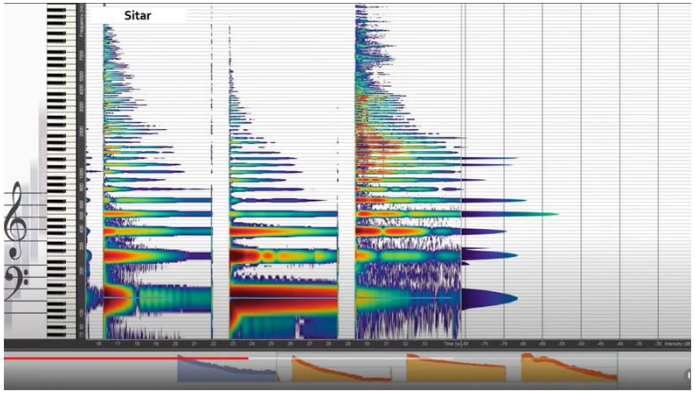
Related Videos



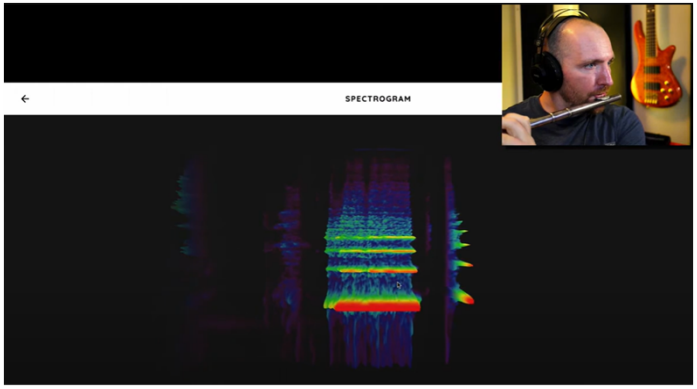
https://www.youtube.com/watch?v=2Hj1kAWVjLo



https://www.youtube.com/watch?v=fZ18C5RXDcc



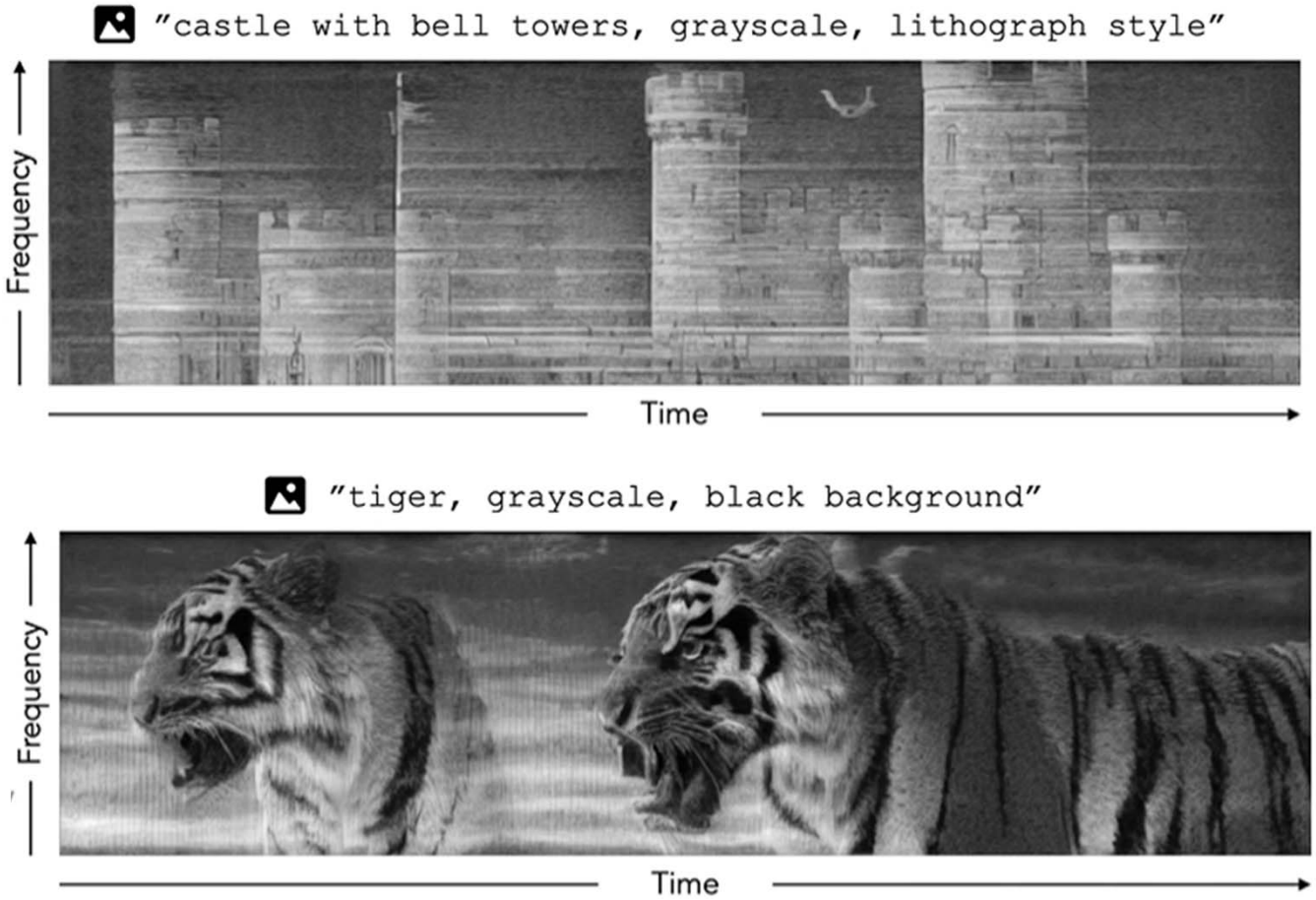
https://www.youtube.com/watch?v=VRAXK4QKJ1Q

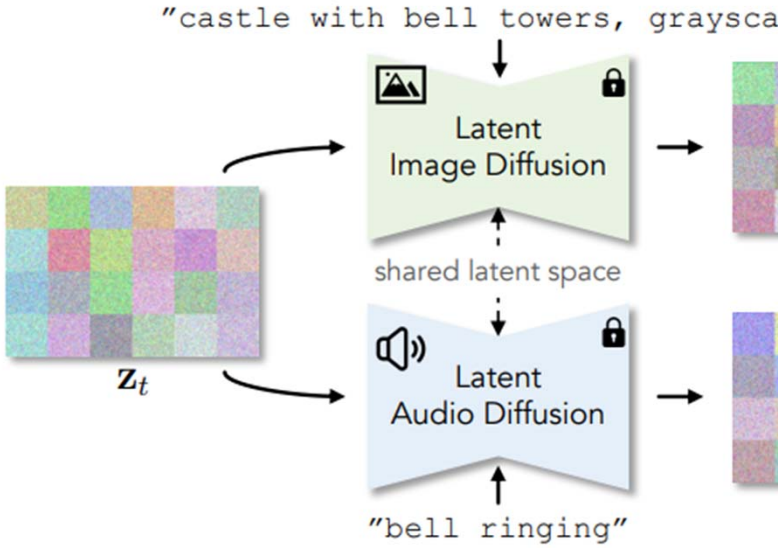


https://www.youtube.com/watch?v=jEO0GHU3xsU

Fun Stuff: “Images that Sound”

https://ificl.github.io/images-that-sound/



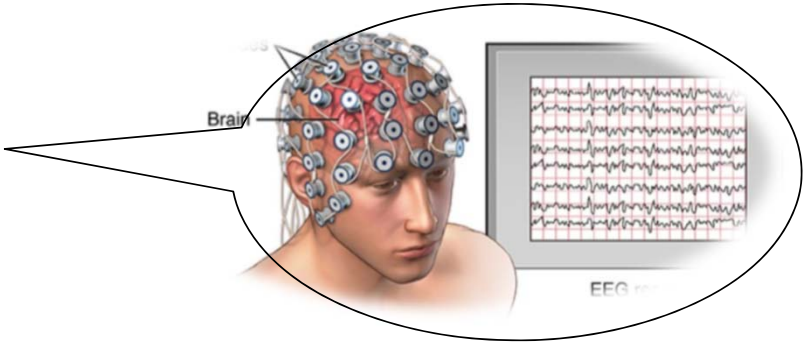


Outline

- Time and frequency representations for music
- **Math in STFT**

Sampling Rate

- Definition: number of samples per second
- Why: analog to digital
- Examples
  - EEG signal: **128** Hz
  - Telephone audio: **8k** Hz
  - Music audio: **44k** HZ

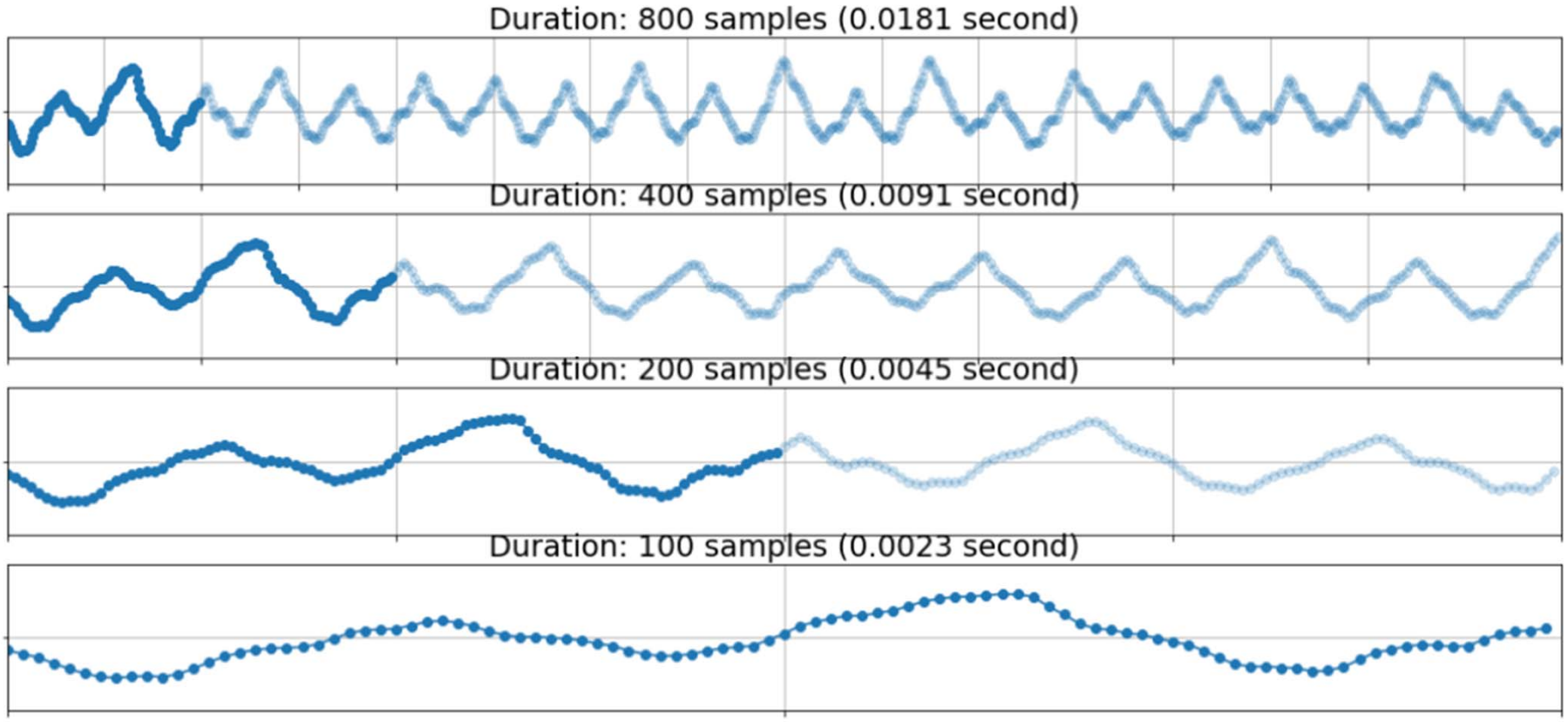


https://www.brightbraincentre.co.uk/electroencephalogram-eeg-brainwaves/

```
[ ]  import  librosa
     x,  sr  =  librosa.load('audio/simple_loop.wav')
```

```
waveform, sample_rate = get_speech_sample()
```

Sampling Rate



https://music-classification.github.io/tutorial/part2_basics/input-representations.html

Sampling Rate

- Usually, a model considered signals with a **certain and fixed** sampling rate
  - The “x”s with different sampling rates are not on the same time basis

```
x, sr = librosa.load('audio/simple_loop.wav')
```

  - Make sure your data are under the same sampling rate
  - Double check the sampling rate assumed by the models you sourced from GitHub
    - Speech: 16k Hz
    - Music: 22k, 44k, or 48k Hz
  - If we want to ML/DL model trained on signals with one sampling rate to process a signal under a different sampling rate, “**resampling**” of that signal is needed

Resampling

Library	Time on CPU (ms)
soxr (HQ)	7.2
scipy.signal.resample	13.4
soxr (VHQ)	15.8
torchaudio	19.2

https://github.com/dofuuz/python-soxr

```
y = soxr.resample(
    x,          # 1D(mono) or 2D(frames, channels) array input
    48000,      # input samplerate
    16000       # target samplerate
)
```

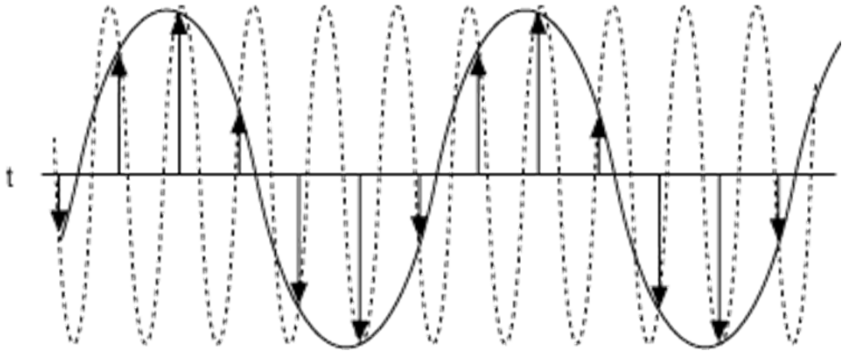
Sample Rate independent Models

- Usually, a model considered signals with a certain and fixed sampling rate
- Building a **sampling-rate agnostic** model that can take signals sampled at arbitrary sampling rates is an advanced topic

Ref 1: Saito et al, “Sampling-frequency-independent audio source separation using convolution layer based on impulse invariant method,” EUSIPCO 2021
Ref 2: Carson et al, “Sample rate independent recurrent neural networks for audio effects processing,” DAFx 2024

Nyquist–Shannon Sampling Theorem

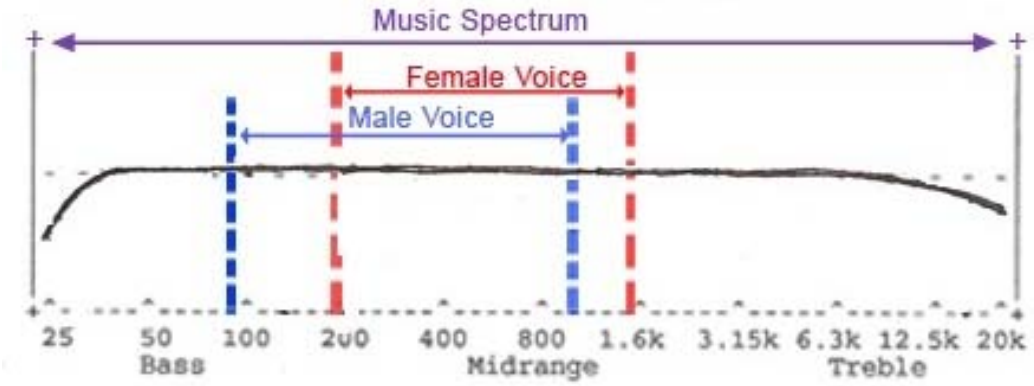
- A signal must be sampled at least **twice** as fast as the bandwidth of the signal to accurately reconstruct the waveform; otherwise, the high-frequency content will alias at a frequency inside the spectrum of interest
- **Sampling freq (Fs) > 2* the highest freq in the signal**



http://zone.ni.com/reference/en-XX/help/370524T-01/siggenhelp/fund_nyquist_and_shannon_theorems/

Nyquist–Shannon Sampling Theorem

- Telephone audio: 8k Hz
  - Via phone, we cannot hear frequency higher than 4k Hz



https://www.quora.com/How-do-HRT-sex-reassignment-and-other-such-procedures-affect-vocal-production-particularly-the-singing-voice

- **Question**: With Fs=128 Hz, we assume that we don't need to care about frequency higher than __ Hz in brain waves

Nyquist–Shannon Sampling Theorem

Brainwaves, Frequencies and Functions

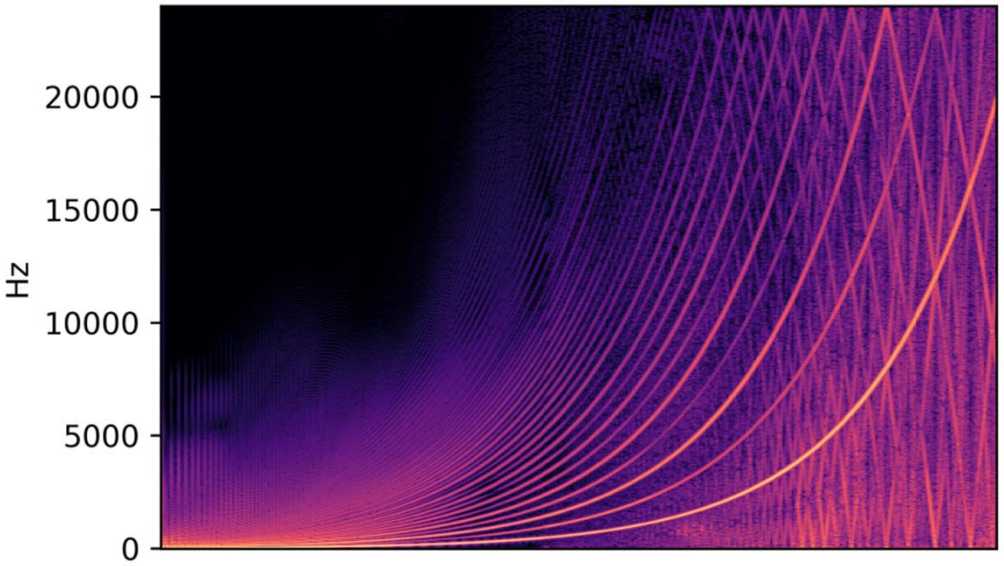
Unconscious		Conscious		
Delta	**Theta**	**Alpha**	**Beta**	**Gamma**
0,5 – 4 Hz	4 – 8 Hz	8 – 13 Hz	13 – 30 Hz	30-42 Hz
Instinct	**Emotion**	**Consciousness**	**Thought**	**Will**
Survival Deep sleep Coma	Drives Feelings Trance Dreams	Awareness of the body Integration of feelings	Perception Concentration Mental activity	Extreme focus Energy Ecstasy

http://altered-states.net/barry/update236/

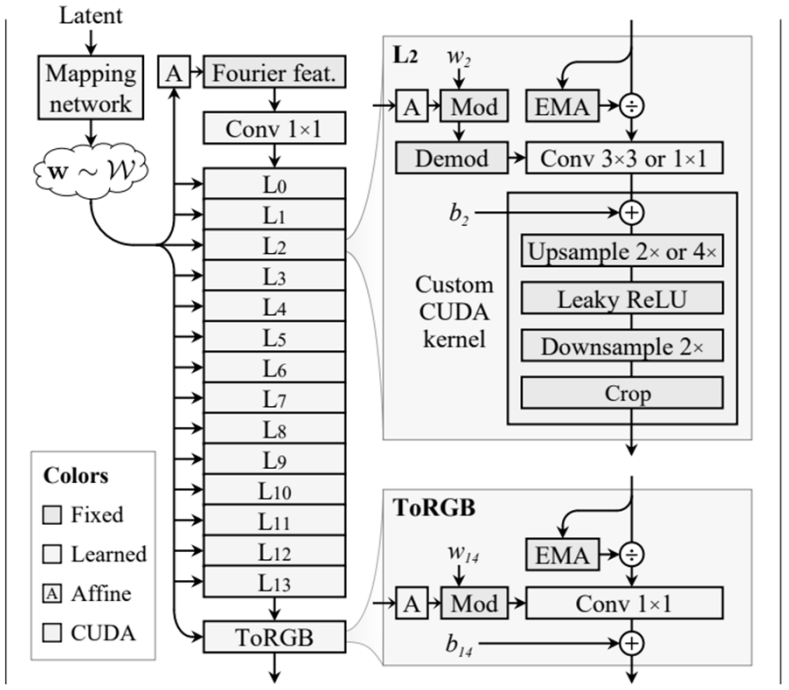
- **Question**: With Fs=128 Hz, we assume that we don't need to care about frequency higher than 64 Hz in brain waves

Aliasing

- High-frequency components fold back



Ref 1: Yeh et al, "PyNeuralFx: A Python package for neural audio effect modeling," ISMIR-LBD 2024



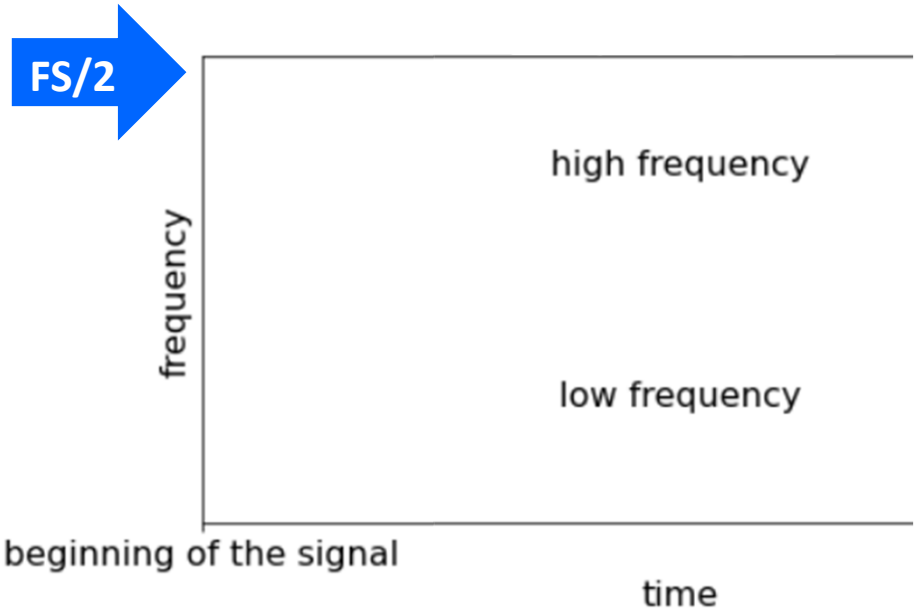
(b) Our alias-free StyleGAN3 generator architecture

https://nvlabs.github.io/stylegan3/

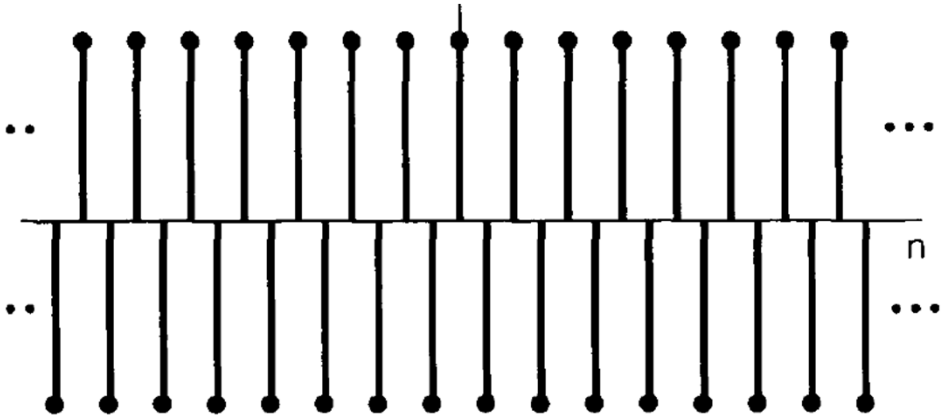
Ref 2: Karras et al, "Alias-free generative adversarial networks," NeurIPS 2021

Nyquist–Shannon Sampling Theorem

- Sampling freq (Fs) > 2* the highest freq in the signal
- **Highest freq bin in STFT = Fs/2**



Sinusoid with the highest freq given a fixed FS



DTFS: Fourier Series Representation for Discrete-time Periodic Signals

- For **periodic** signals with fundamental **period N**, i.e., $x[n] = x[n+N]$
- The Fourier series coefficients a_k can be interpreted as the inner product of the signal $x[n]$ and a basis function $e^{jk\frac{2\pi}{N}n}$, which is complex-valued
- a_k are **complex-valued**
- a_k are also **periodic with period N**, i.e., $a_k = a_{k+N}$

Synthesis equation: $x[n] = \sum_{k=\langle N \rangle} a_k e^{jk\frac{2\pi}{N}n}$

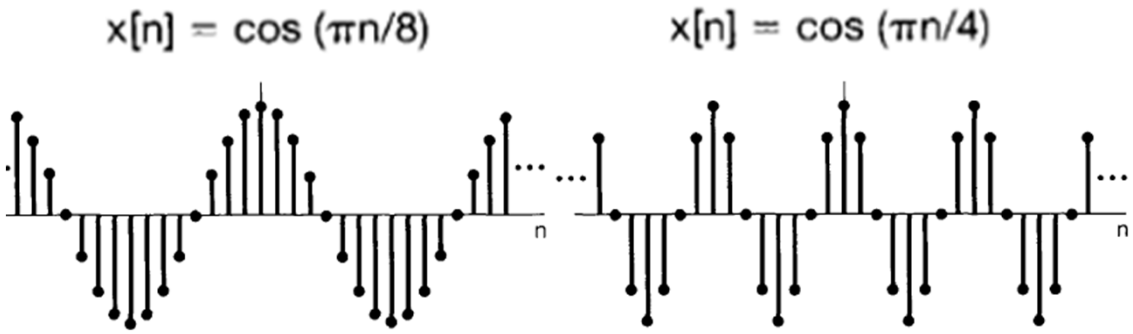
Analysis equation: $a_k = \frac{1}{N}\sum_{n=\langle N \rangle} x[n] e^{-jk\frac{2\pi}{N}n}$

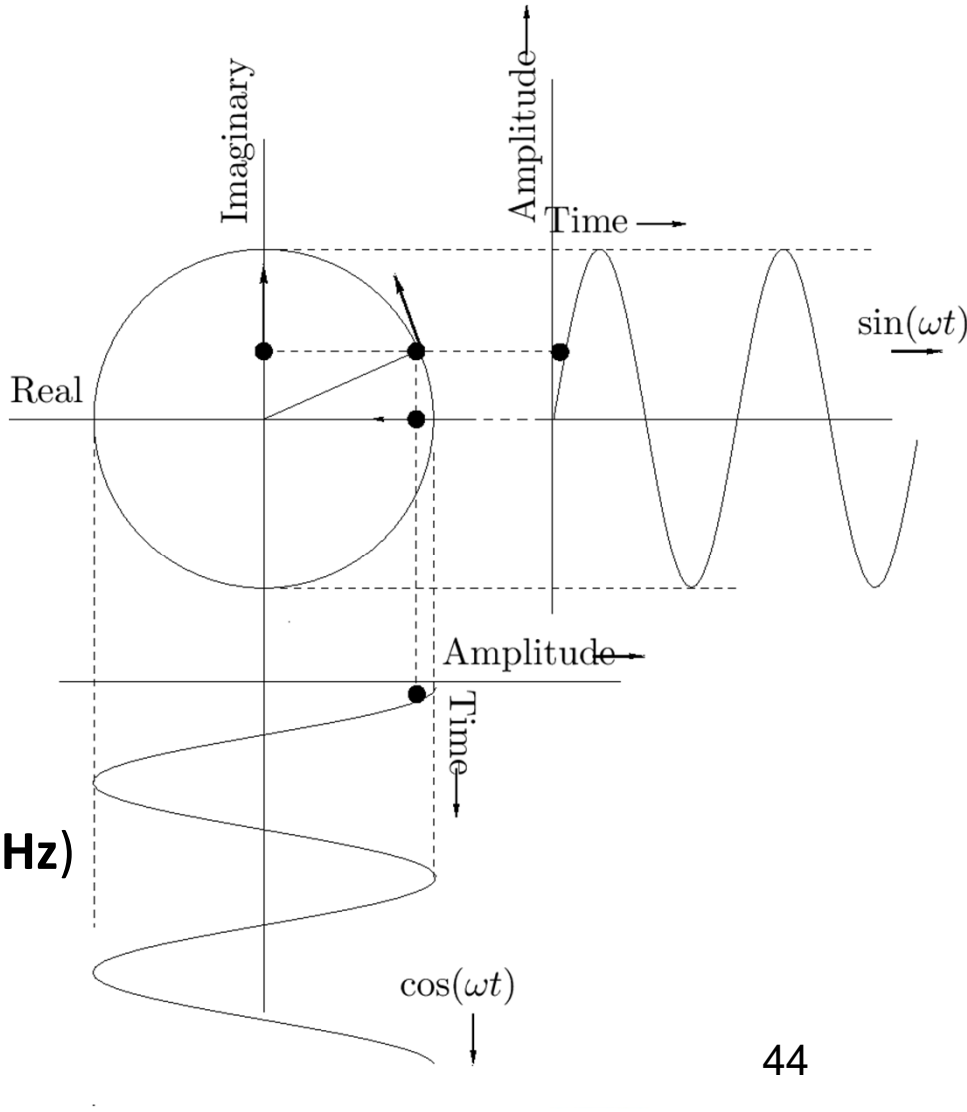
In discrete-time, $e^{jk\omega n} = e^{j(k+N)\omega n}$; so there are **only N unique basis vectors**

Sum over a set of N successive integers

The Basis Functions in DTFS

- $e^{j\omega n} = \cos(\omega n) + j\sin(\omega n)$





- angular or ordinary frequency
  - ω: **angular frequency** (radians per second)
  - f: **ordinary frequency** (circles per second; in **Hz**)
  - $e^{j2\pi fn} = \cos(2\pi fn) + j\sin(2\pi fn)$

Discrete-time Fourier Transform (DFT): Extending DTFS to Deal with Aperiodic, Finite-Duration DT Signals

- For **periodic DT** signal $x[n] \overset{\mathcal{FS}}{\leftrightarrow} a_k$

Synthesis: $x[n] = \sum_{k=\langle N \rangle} a_k e^{jk\omega n}$

Analysis: $a_k = \frac{1}{N} \sum_{n=\langle N \rangle} x[n] e^{-jk\omega n}$

- For **aperiodic, finite-duration DT** signal $x[n] \overset{\mathcal{DFT}}{\longleftrightarrow} X_k$

Synthesis: $x[n] = \sum_{k=0}^{N-1} X_k e^{jk\omega n}$

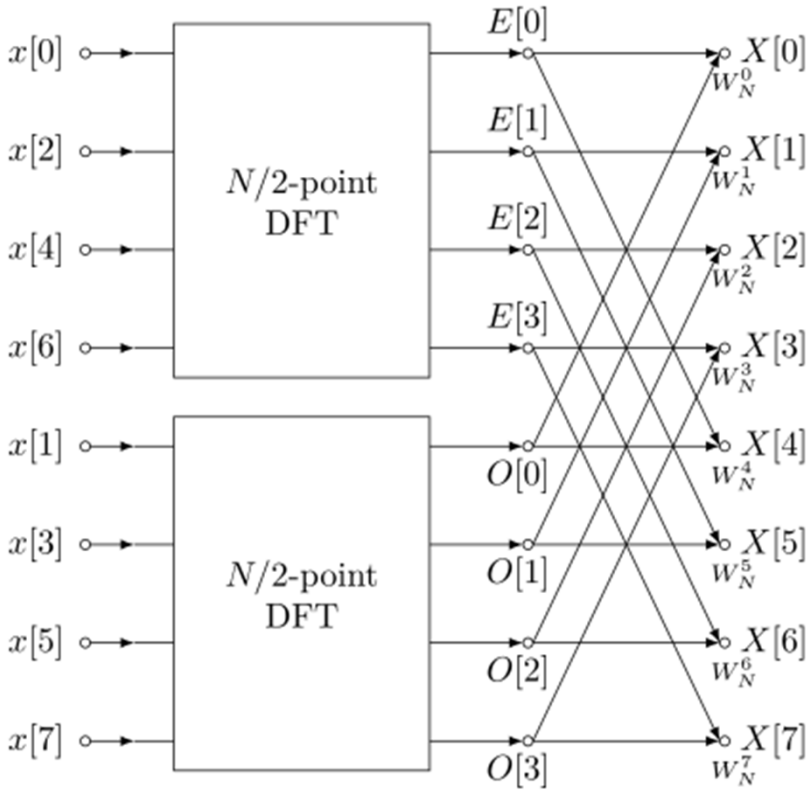
Analysis: $X_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk\omega n}$

NOTE: While the length of $x[n]$ is N, the length of X_k is also N

DFT and FFT (Fast Fourier Transform)

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_DFT-FFT.html

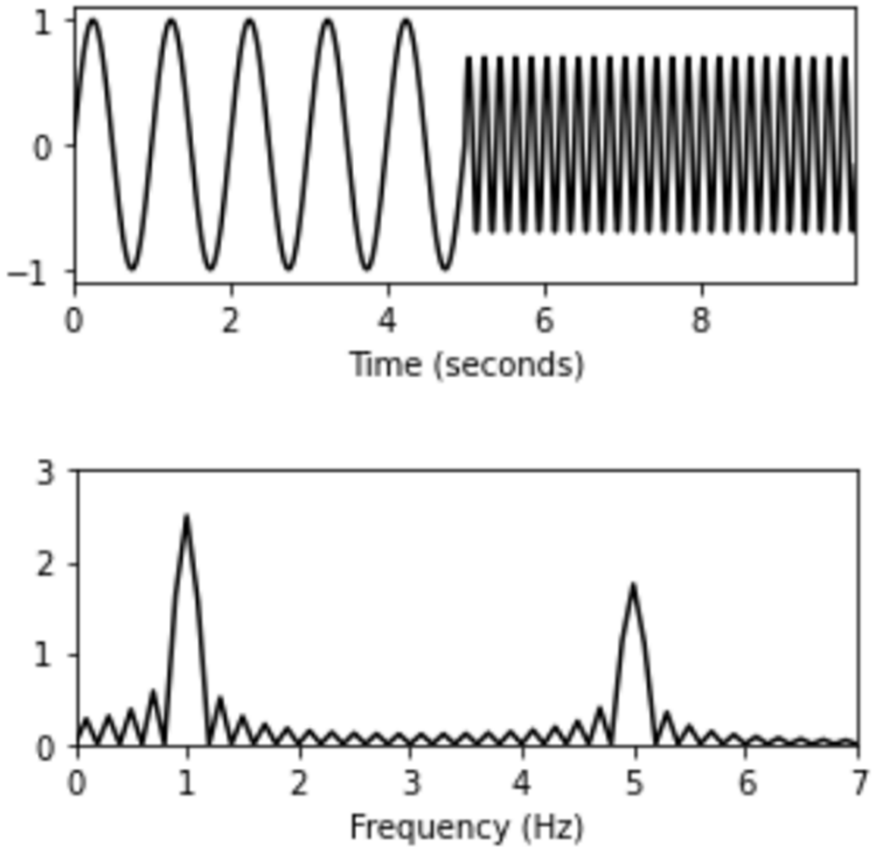
- FFT is a fast algorithm to compute the DFT



https://en.wikipedia.org/wiki/Fast_Fourier_transform

FFT cannot Localize Temporal Events

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html



STFT: Short-time Fourier Transform

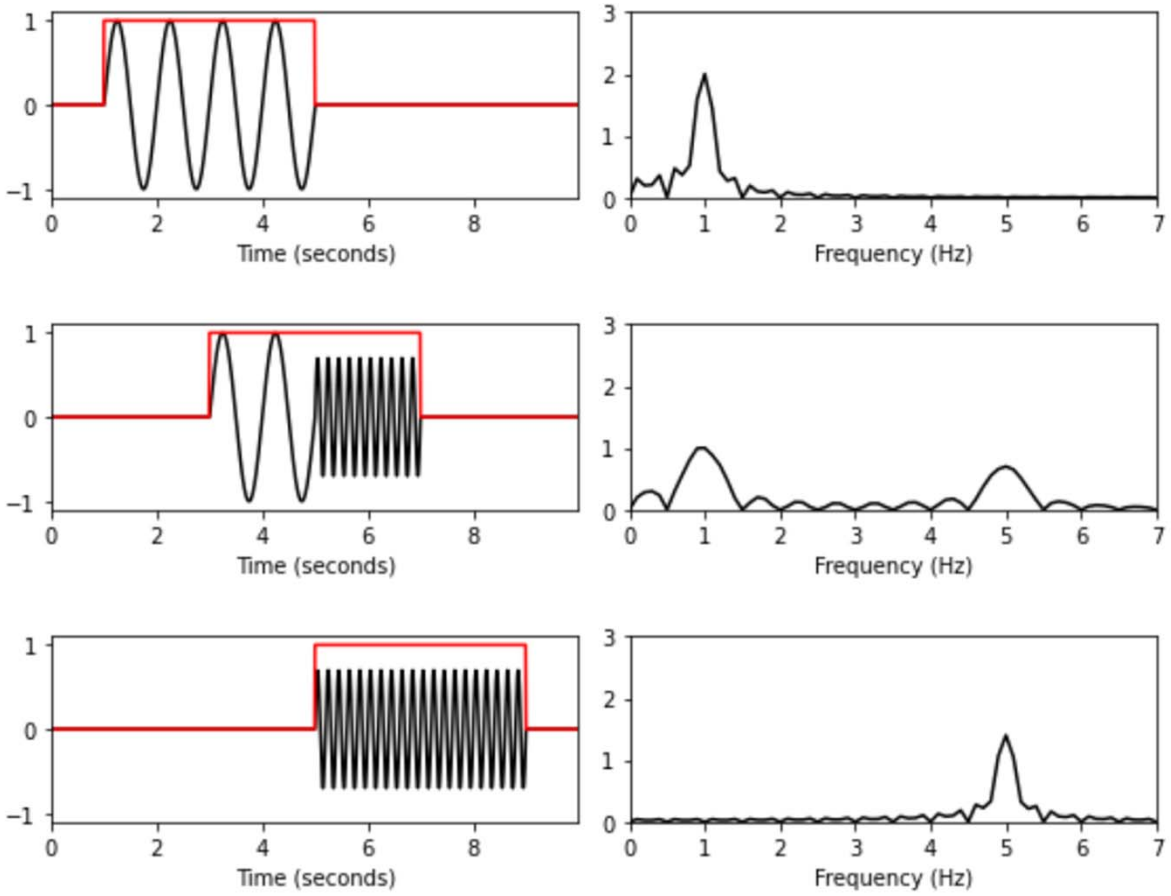
https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html

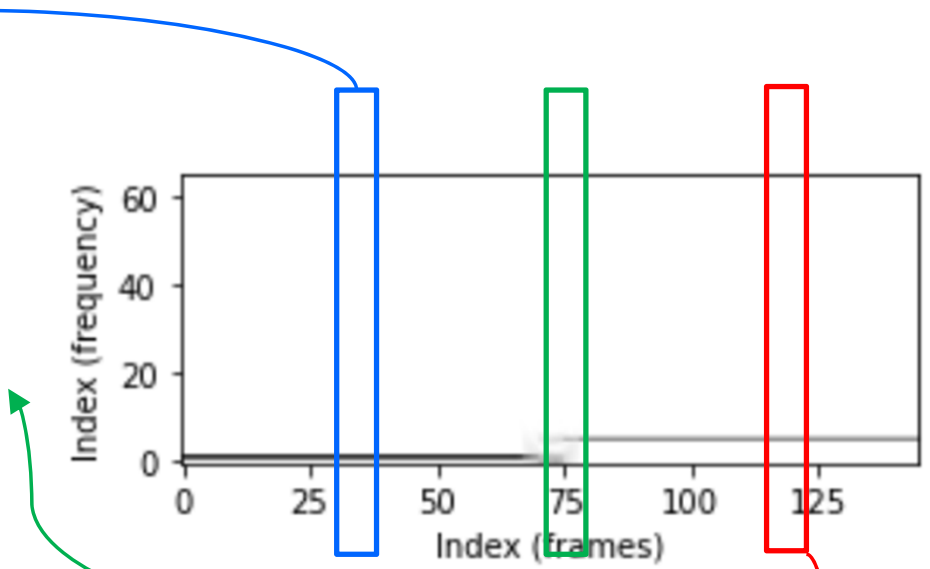
$$\mathcal{X}(m,k) := \sum_{n=0}^{N-1} x(n+mH)w(n)\exp(-2\pi ikn/N)$$

- $x[n]$: real-valued discrete-time signal of length *L* obtained by equidistant sampling with respect to a fixed sampling rate
- $w[n]$: window function of length $N < L$
- H: "hop size"

Windowed FFT

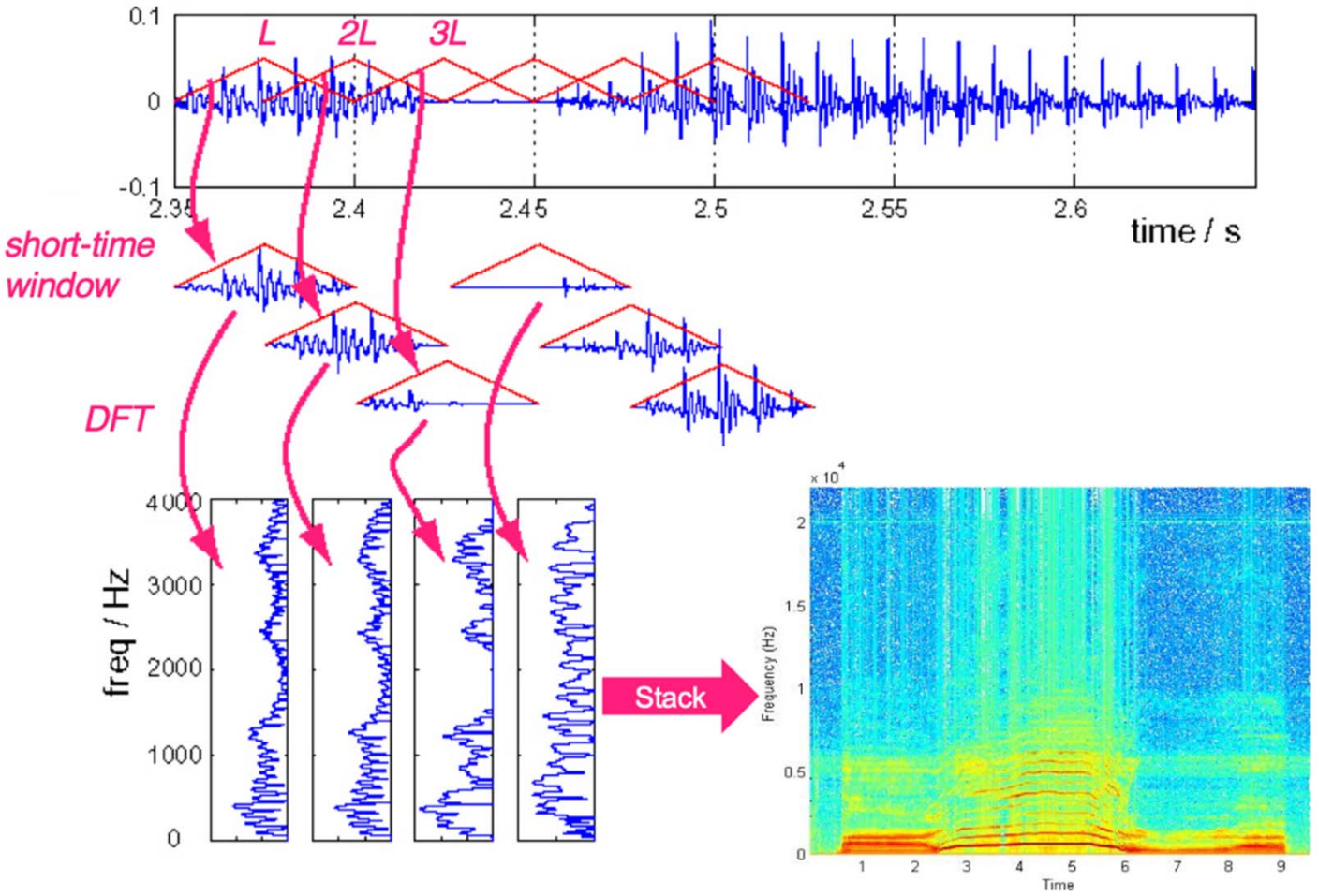
https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html





Windowed FFT

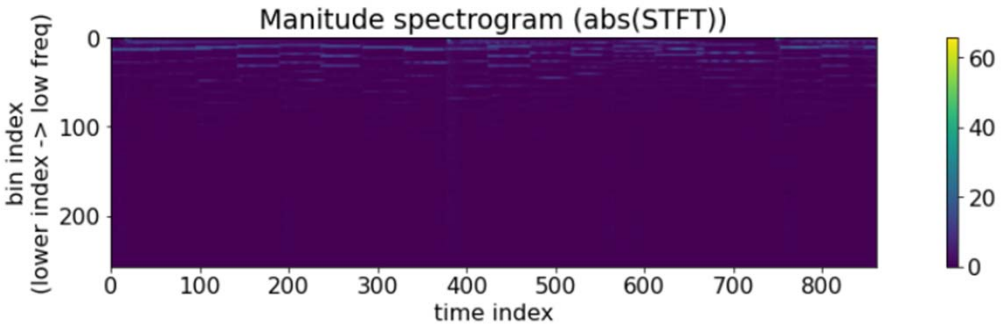
https://source-separation.github.io/tutorial/basics/representations.html

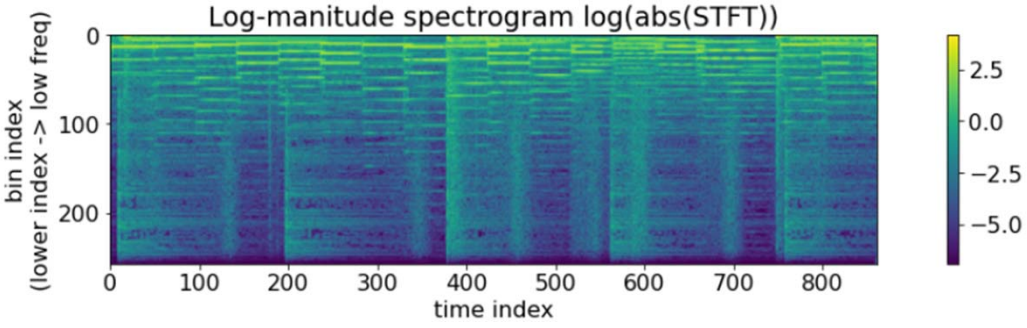


Magnitude Spectrogram

https://music-classification.github.io/tutorial/part2_basics/input-representations.html?highlight=phase

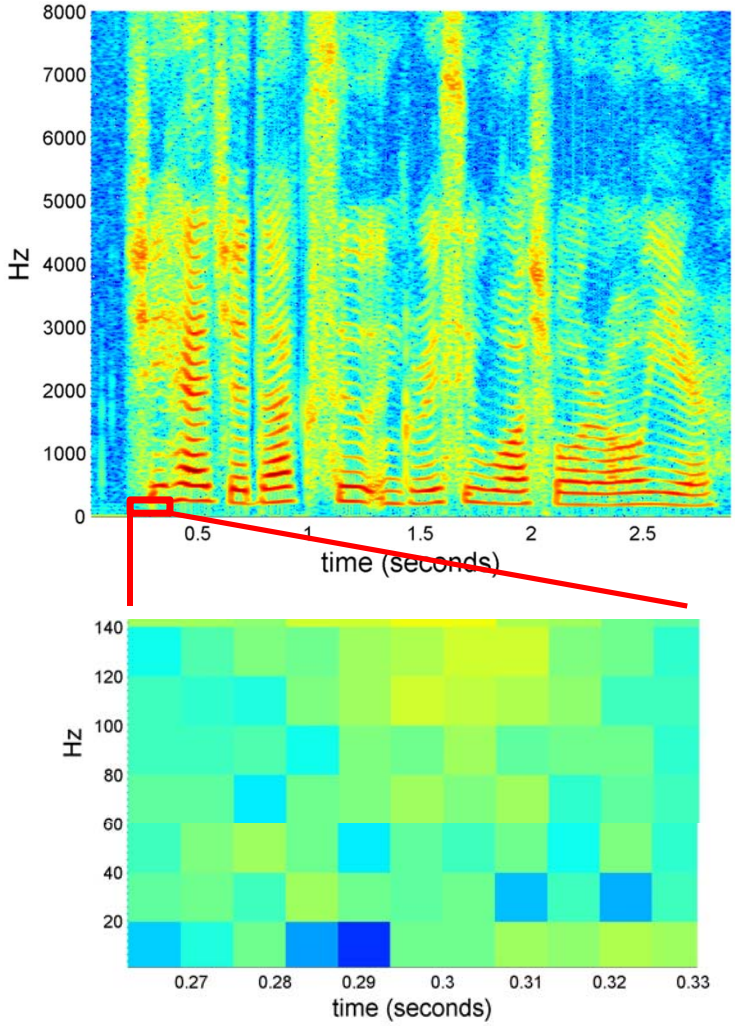
- **STFT**: complex-valued, linear frequency scale
- **Spectrogram**: a time-frequency representation, usually computed via STFT
- **Magnitude spectrogram**: abs(STFT); real-valued, non-negative
- **Log-magnitude spectrogram**: log(abs(STFT)); real-valued, non-negative
- The phase information of STFT is often considered less important





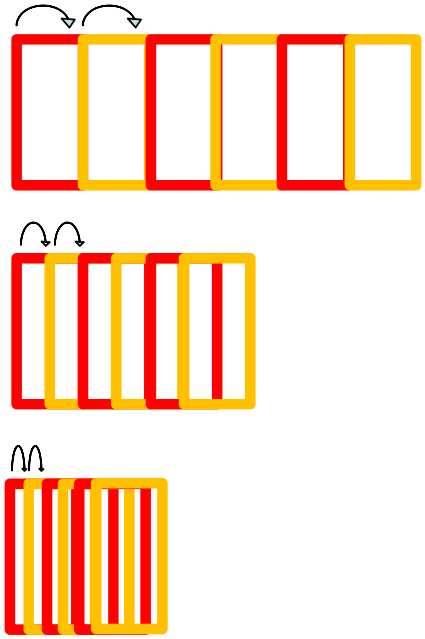
Frame, Frequency Bin, and Time-Frequency Point

- The magnitude spectrogram can be represented as a matrix
- **Frame**: along time
- **Frequency bin**: along frequency
- **Time-frequency point**



Hop Size in STFT

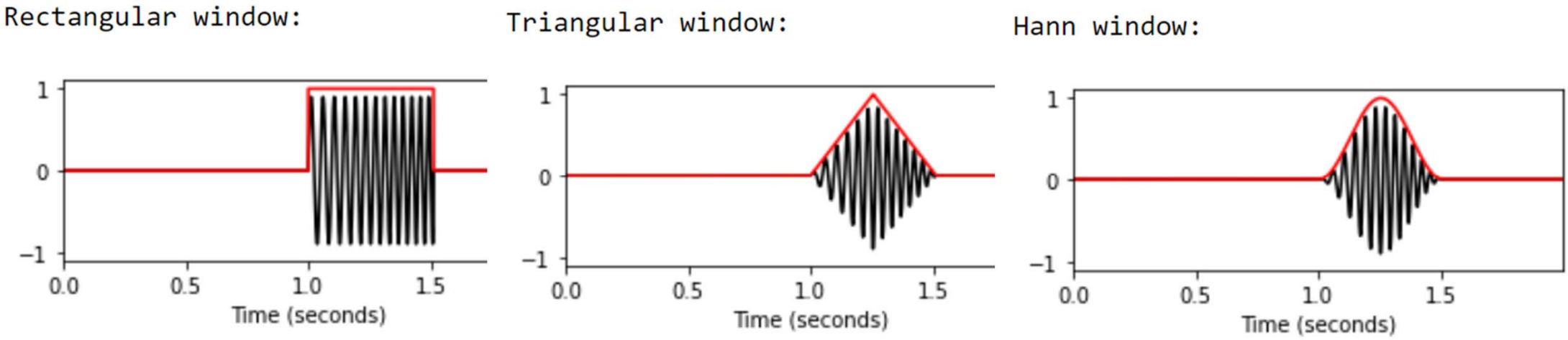
- Hop size
  - Can be proportional to the window size
    - hop_size = win_size
    - hop_size = 0.5*win_size
    - hop_size = 0.1*win_size
  - Can also be not proportional to the window size



Window Functions in STFT

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Window.html

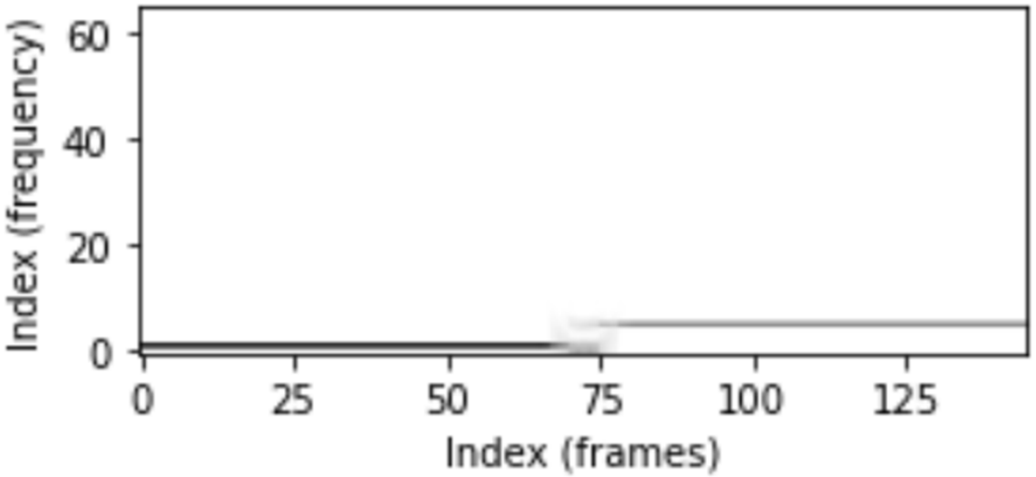
- Hann window is often used



Interpretation of Time and Frequency Indices in STFT

https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html

$$F_{\mathrm{coef}}(k) := \frac{k \cdot F_{\mathrm{s}}}{N}$$



- **Frequency spacing** (Δfreq): $\frac{F_s}{N}$
- **Temporal spacing** (Δtime): $\frac{H}{F_s}$

$$T_{\mathrm{coef}}(m) := \frac{m \cdot H}{F_{\mathrm{s}}}$$

Trade-off between Frequency/Temporal Resolution in STFT

- **Frequency resolution**: $\frac{F_S}{N}$
  - Longer window size (N) → finer frequency resolution (but larger resulting STFT)
    → can localize events along the frequency axis
- **Temporal resolution**: $\frac{H}{F_S}$
  - Smaller hop size (H) → finer temporal resolution (but larger resulting STFT)
    → can localize events along the time axis

- If $H = N/R$ (e.g., $R = 4$)
  - Temporal resolution: $\frac{N}{RF_S}$ (no good freq/time resolution at the same time)

Size of the Magnitude Spectrogram

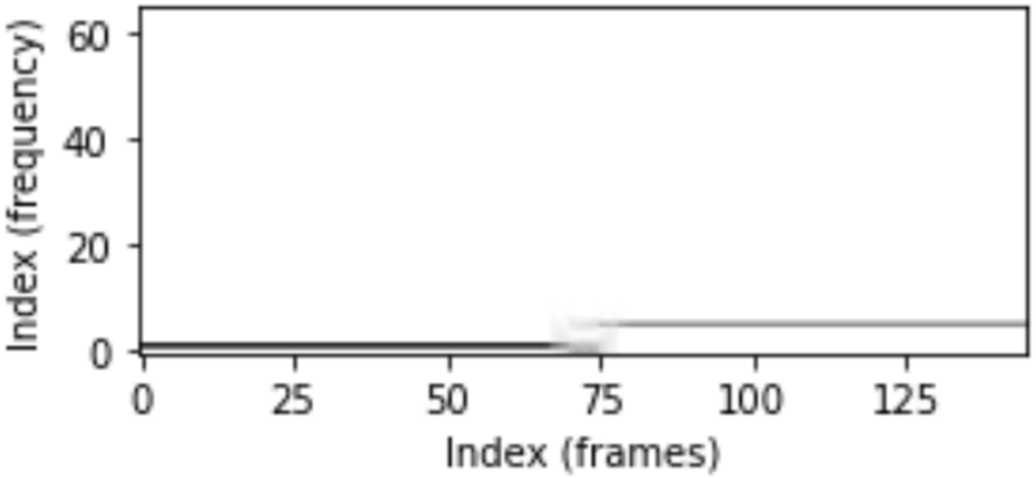
- Number of frequency bins (rows in the matrix): $\left\lceil \frac{N}{2} \right\rceil + 1$
  - If N is even, this is $\frac{N}{2} + 1$
  - If N is odd, this is $\frac{N+1}{2}$
  - The DFT output is symmetric, since the input signal is real-valued
- Number of time frames (columns in the matrix): $\left\lfloor \frac{L-N}{H} \right\rfloor + 1$
  - L is the total number of samples in the signal
  - The $+1$ accounts for the first window starting at the beginning of the signal

How about Frequencies in Between?

$$F_{\text{coef}}(k) := \frac{k \cdot F_s}{N}$$

- First frequency bin ($k = 0$): DC
- Second bin ($k = 1$): $\frac{F_S}{N}$ Hz
- Frequencies that are not integer multiples of the $\frac{F_S}{N}$: their energy is distributed across the bins, and we approximate it by the nearest bin:

$$k = \text{round}\left(\frac{f \cdot N}{F_S}\right)$$



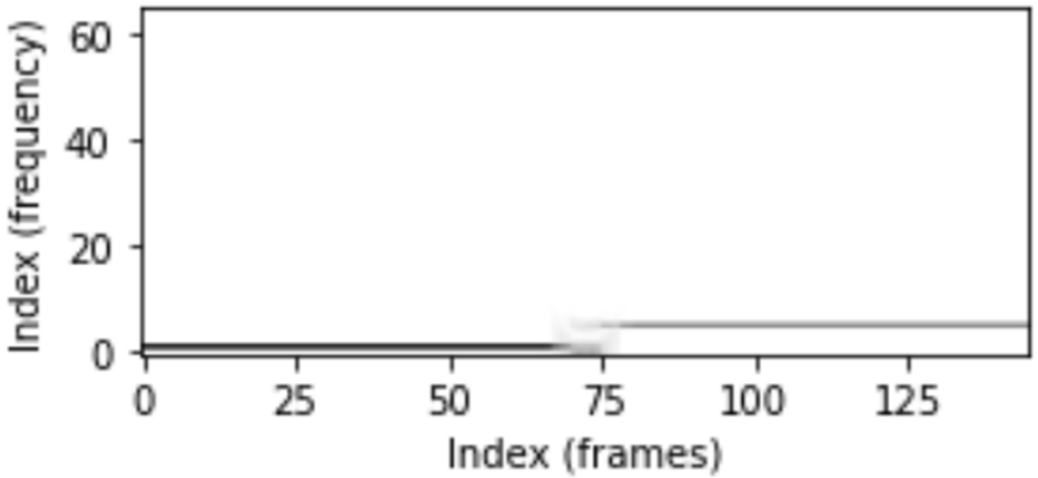
How about Frequencies in Between?

- Example

  If $F_s = 44100$ Hz and $N = 1024$

  - First bin: 0 Hz
  - Second bin: 43.066 Hz
  - Third bin: 86.132 Hz

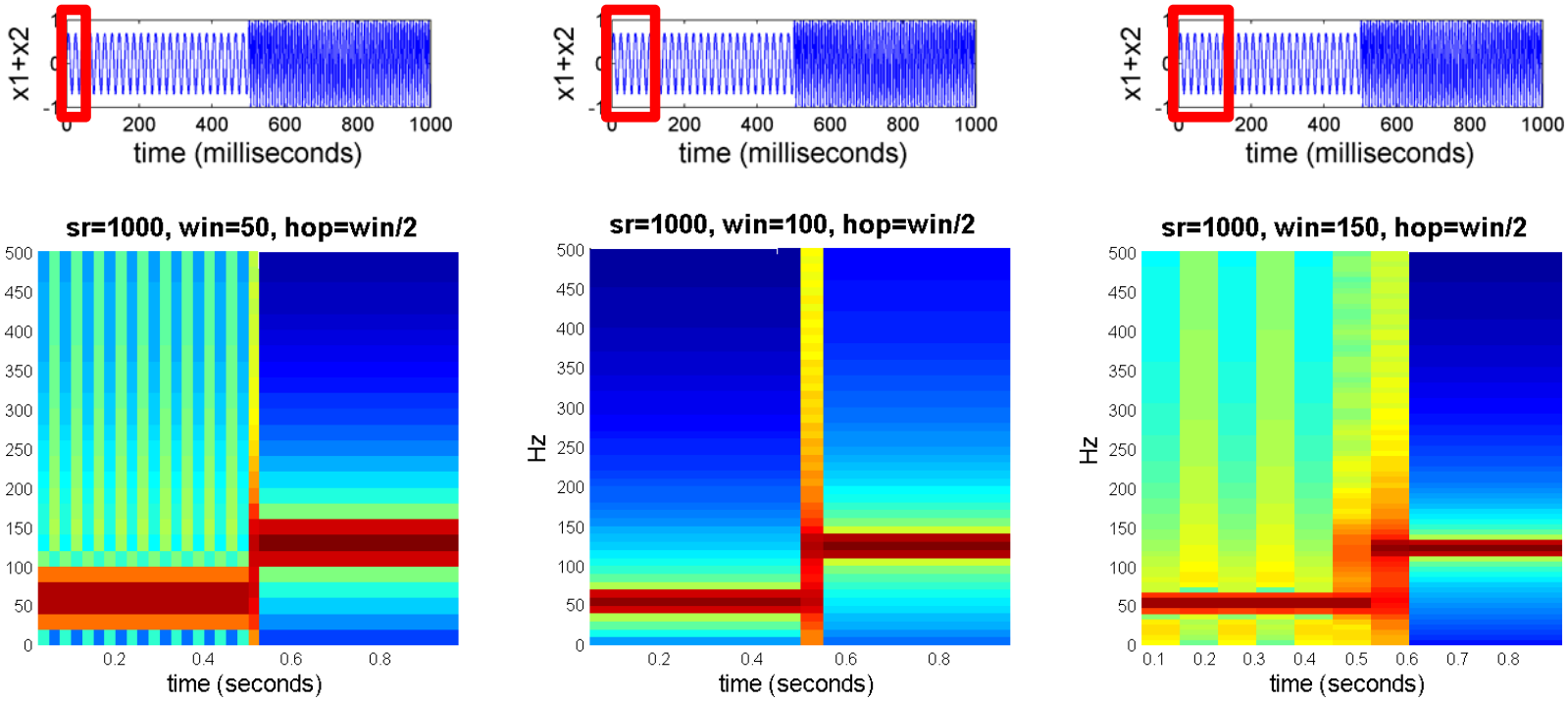
  - 20Hz → go to the first bin
  - 30Hz → go to the second bin



  - First bin: $[0, 21.533)$ Hz
  - Second bin: $[43.066 - 21.533, 43.066 + 21.533)$ Hz

Trade-off between Frequency/Temporal Resolution in STFT

- Different window size (win_size = 50, 100, 150)
- hop_size = win_size/2



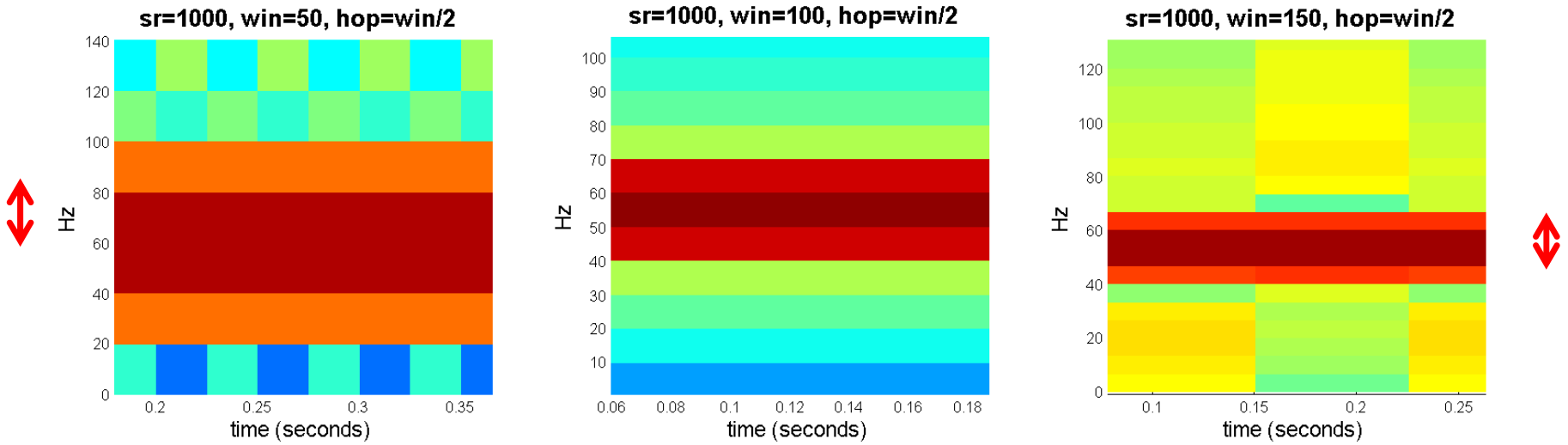
size: 26 x 39

size: 51 x 19

size: 76 x 12

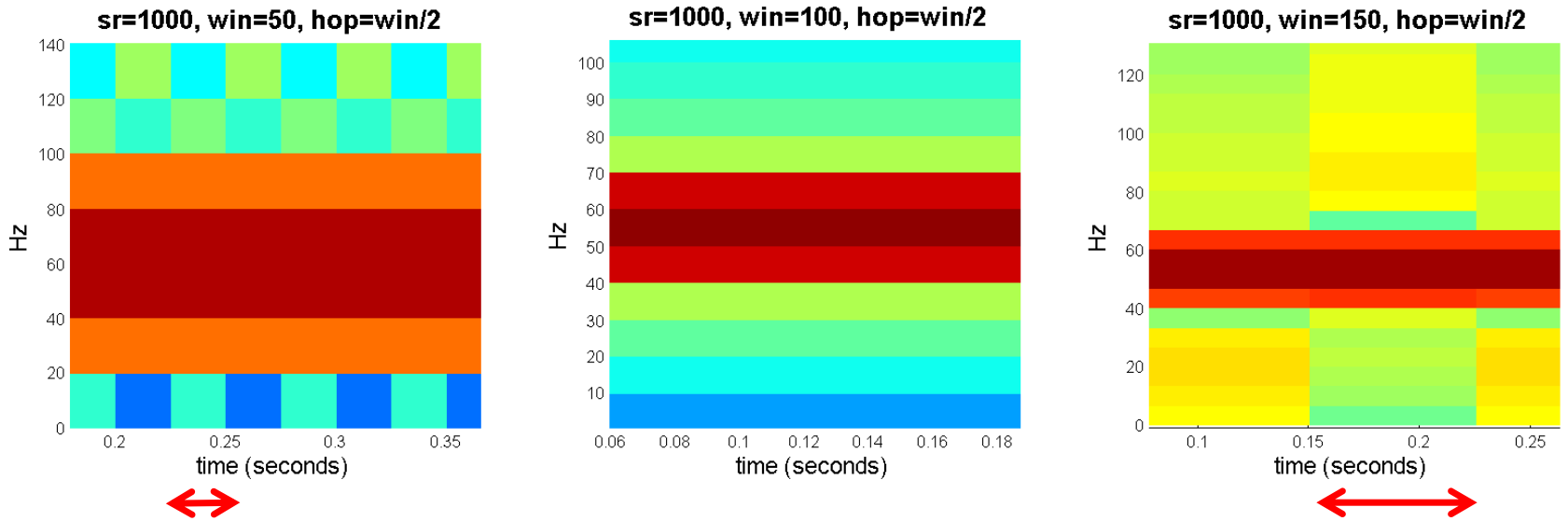
Frequency Resolution in STFT

- **freq_resolution = Fs/win_size**
  - longer window → better frequency resolution
  - freq_resolution = 20, 10, 6.6667 (Hz), respectively



Understanding STFT

- **temporal_resolution: hop_size/Fs**
  - longer hop → worse temporal resolution
  - temp_resolution = 25, 50, 75 (ms), respectively

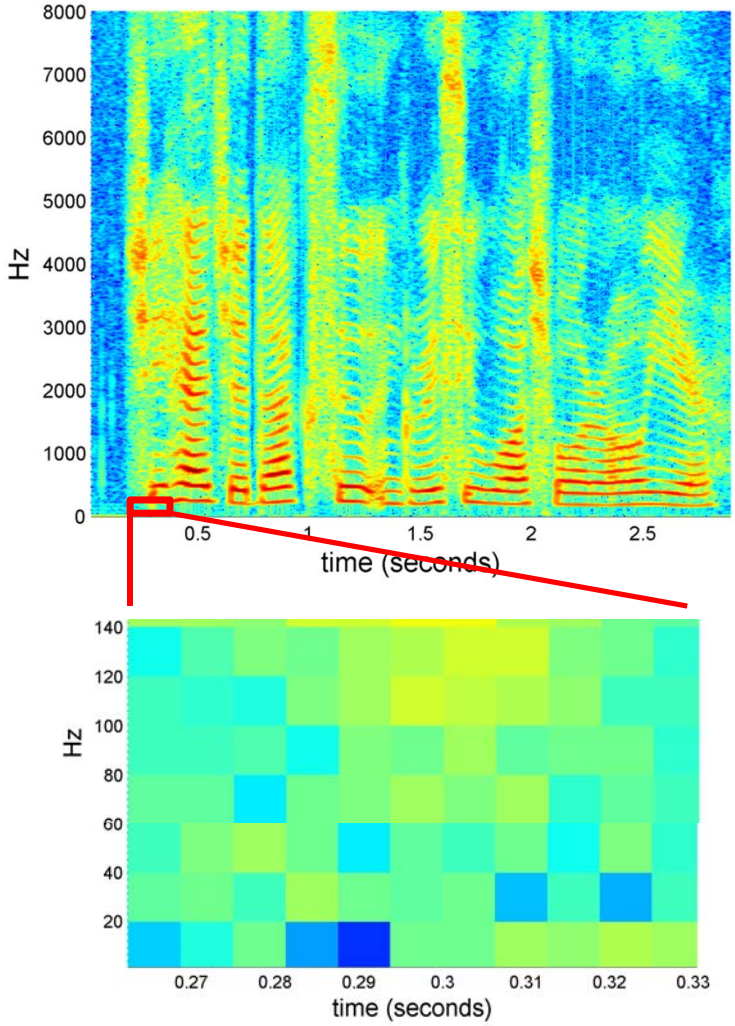


Quiz

1. The figure on the top-right is the spectrogram of a signal. What is the underlying sampling rate?

2. The figure on the bottom-right is a zoom-in of the above figure. We can see that the frequency resolution is 20 Hz. What is the window size (in samples)?

3. The temporal resolution is close to 6.6 ms. What's the hop size (in samples), approximately?



Quiz

4. Given an EEG headset that samples signals at 128 Hz, if we want to be able to discriminate frequency components that differ by 0.5 Hz in frequency, what is the minimal window size (in samples) we need to use? What is the length of such a window in seconds then?
5. Following the previous question, if we further want to discriminate events that differ in time by 0.5 second, what is the maximal hop size (in samples) we need to use?

Brainwaves, Frequencies and Functions

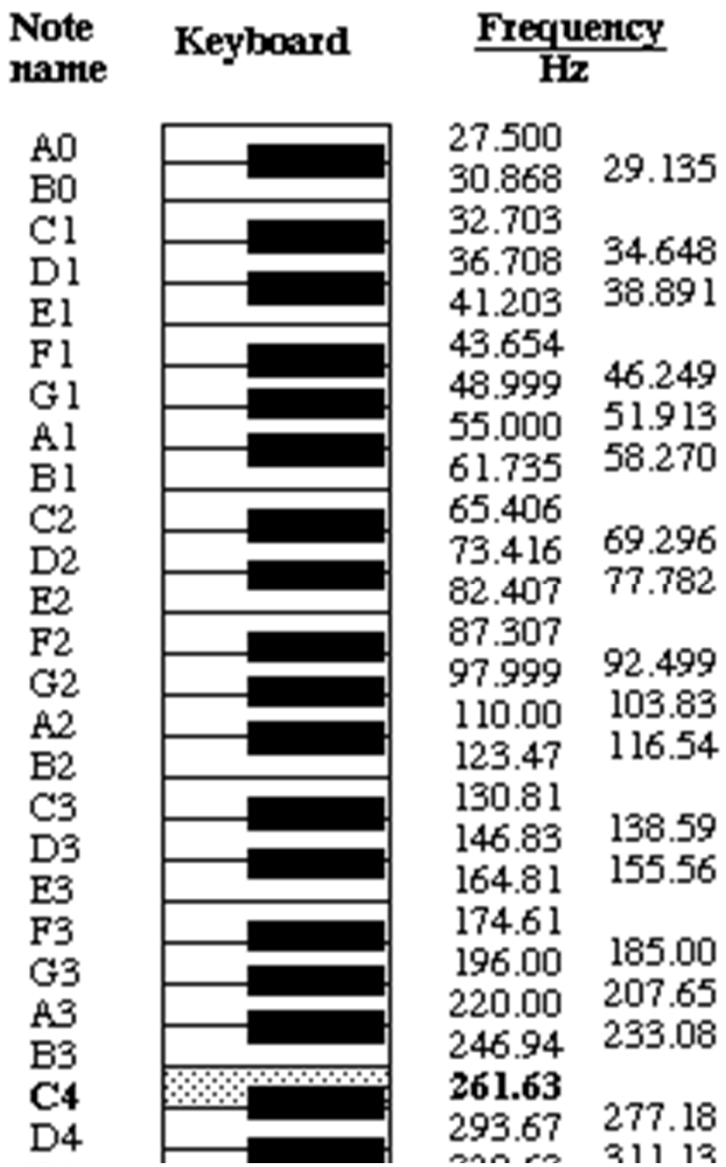
Unconscious		Conscious		
Delta	**Theta**	**Alpha**	**Beta**	**Gamma**
0,5 – 4 Hz	4 – 8 Hz	8 – 13 Hz	13 – 30 Hz	30-42 Hz
Instinct	**Emotion**	**Consciousness**	**Thought**	**Will**

Quiz

6. Given a music signal with sr = 44,100 Hz, when we use a window size of 1,024 samples, what would be the frequency resolution?

7. According to the figure on the right, we know that the fundamental frequency (f0) of A1 is 55 Hz, that of A♯1 is 58.27 Hz, etc. Following the previous question, which notes does the first frequency bin of the STFT would cover?

https://newt.phys.unsw.edu.au/jw/notes.html

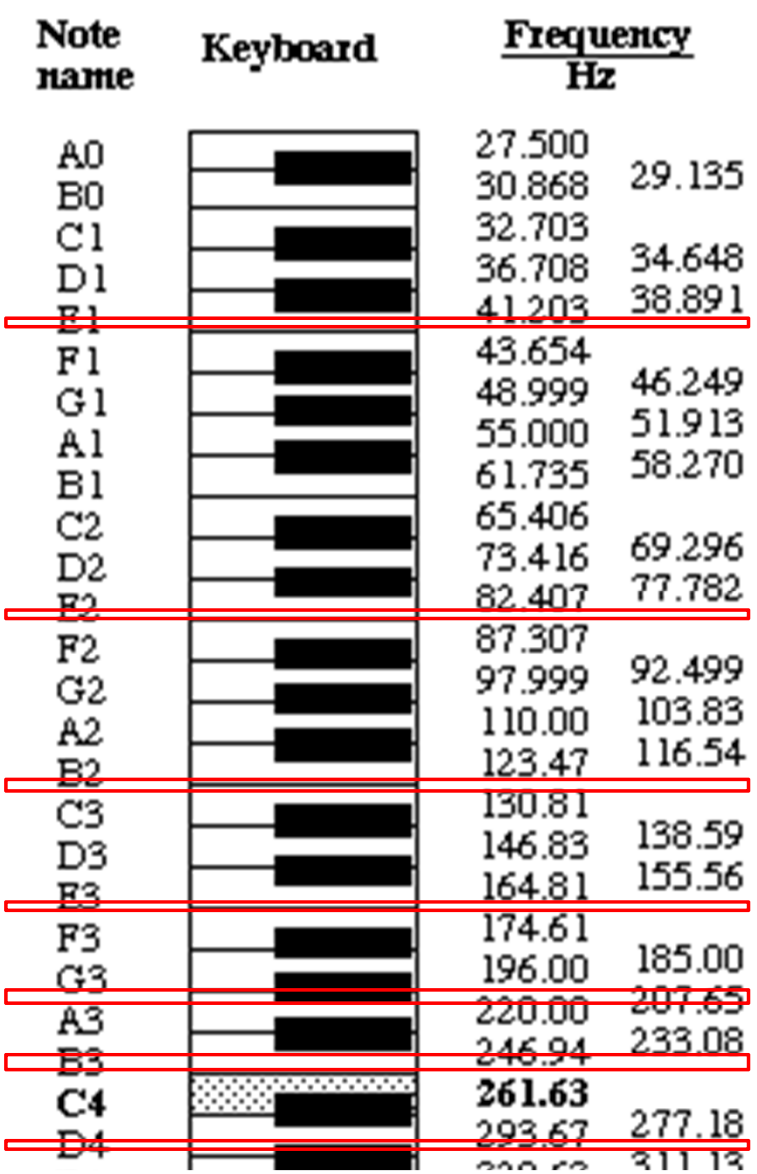


Quiz

6. Given a music signal with sr = 44,100 Hz, when we use a window size of 1,024 samples, what would be the frequency resolution?
   Sol: 43.1 Hz

7. According to the figure on the right, we know that the fundamental frequency (f0) of A1 is 55 Hz, that of A♯1 is 58.27 Hz, etc. Following the previous question, which notes does the first frequency bin of the STFT would cover?

https://newt.phys.unsw.edu.au/jw/notes.html

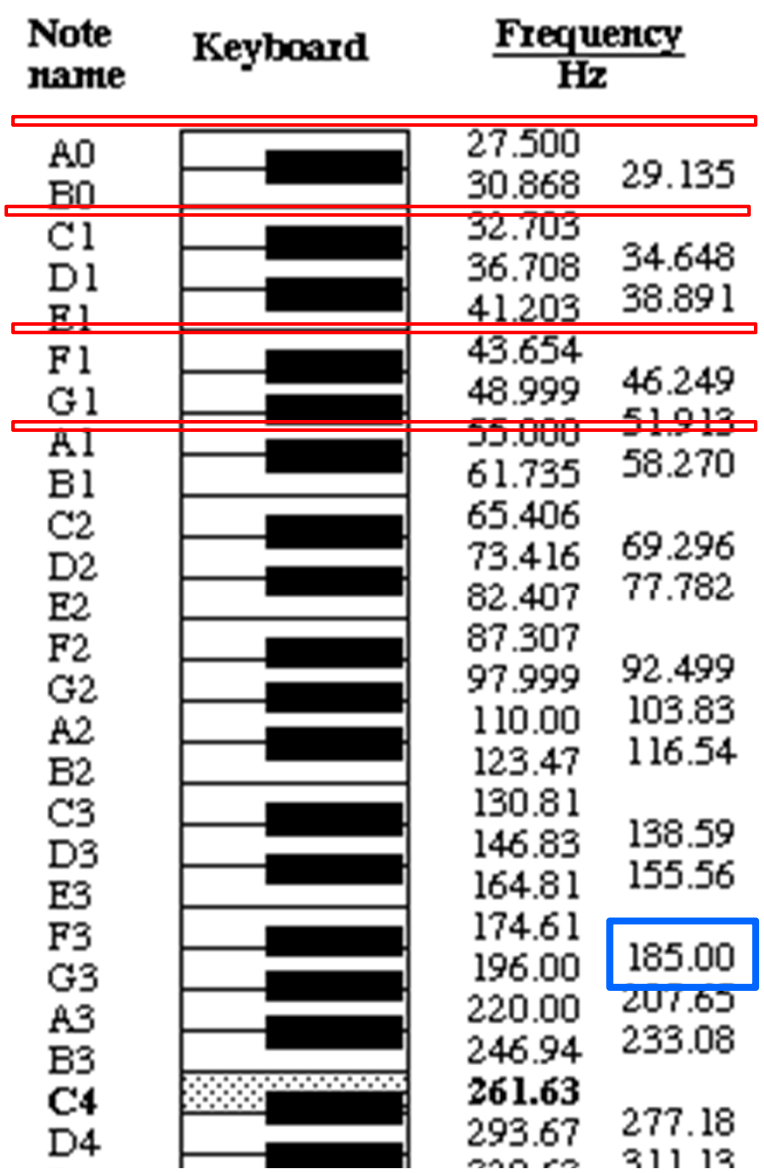


Quiz

8. Following the `6` and `7`;
   how if we use a window size of 4,096 samples?

(Note: Musical notes after F♯3 would be covered by only one frequency bin now)

https://newt.phys.unsw.edu.au/jw/notes.html



Summary

- **Sampling rate**, **window size**, **hop size** all matter
  - Frequency resolution
  - Temporal resolution
  - Number of samples, number of frames, and physical time (in milliseconds)

- Make sure the sampling rate is “right”
  - Do “resample” if the sampling rate of your audio files is different from what is assumed by an open-source model
  - Speech: 16k Hz
  - Music: 22k, 44k, or 48k Hz
  - Similarly for window size, hop size, window function, etc