

2025 edition

Deep Learning for Music Analysis and Generation

# Fundamentals for Musical Audio



**Yi-Hsuan Yang** Ph.D.  
[yhyangtw@ntu.edu.tw](mailto: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	<a href="#">Basics</a>	Basic information on Python, Jupyter notebooks, Anaconda package management system, Python environments, visualizations, and other topics	<a href="#">[html]</a>	<a href="#">[ipynb]</a>
0	<a href="#">Overview</a>	Overview of the notebooks ( <a href="https://www.audiolabs-erlangen.de/FMP">https://www.audiolabs-erlangen.de/FMP</a> )	<a href="#">[html]</a>	<a href="#">[ipynb]</a>
1	<a href="#">Music Representations</a>	Music notation, MIDI, audio signal, waveform, pitch, loudness, timbre	<a href="#">[html]</a>	<a href="#">[ipynb]</a>
2	<a href="#">Fourier Analysis of Signals</a>	Discrete/analog signal, sinusoid, exponential, Fourier transform, Fourier representation, DFT, FFT, STFT	<a href="#">[html]</a>	<a href="#">[ipynb]</a>
3	<a href="#">Music Synchronization</a>	Chroma feature, dynamic programming, dynamic time warping (DTW), alignment, user interface	<a href="#">[html]</a>	<a href="#">[ipynb]</a>

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

# Outline

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

# Audio Waveforms

[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3\\_Waveform.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3_Waveform.html)

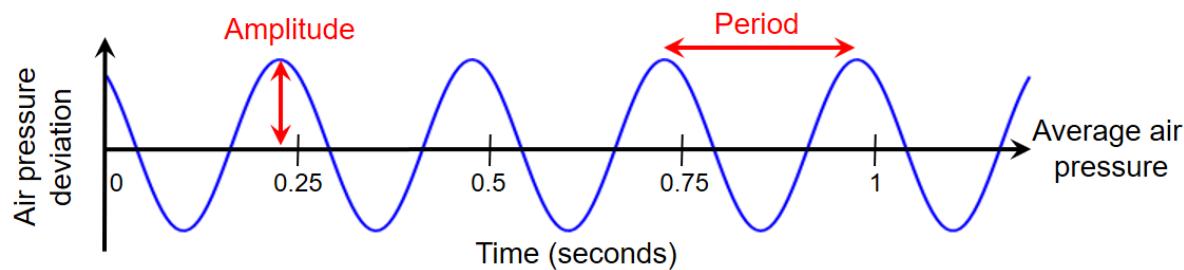
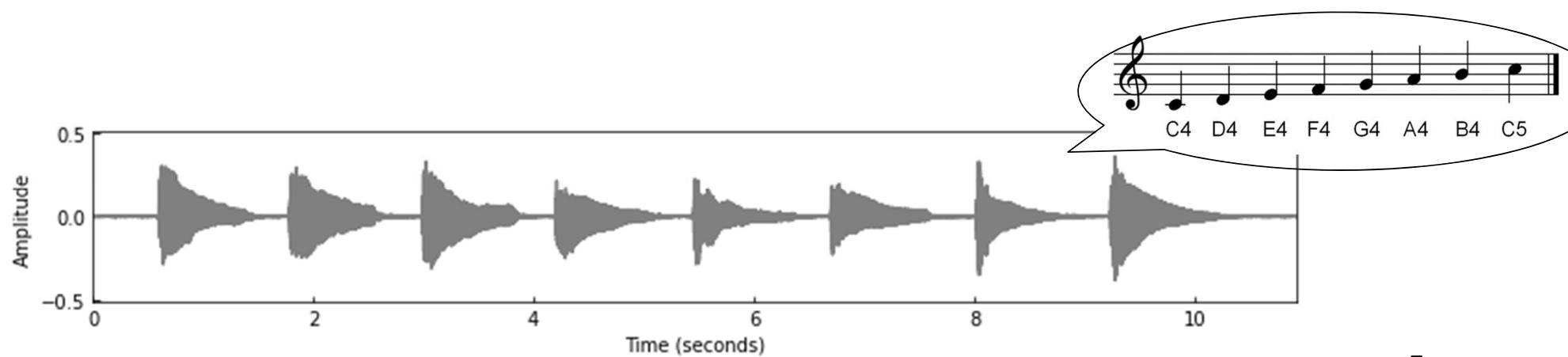


Figure 1.19 from [Müller, FMP, Springer 2015]



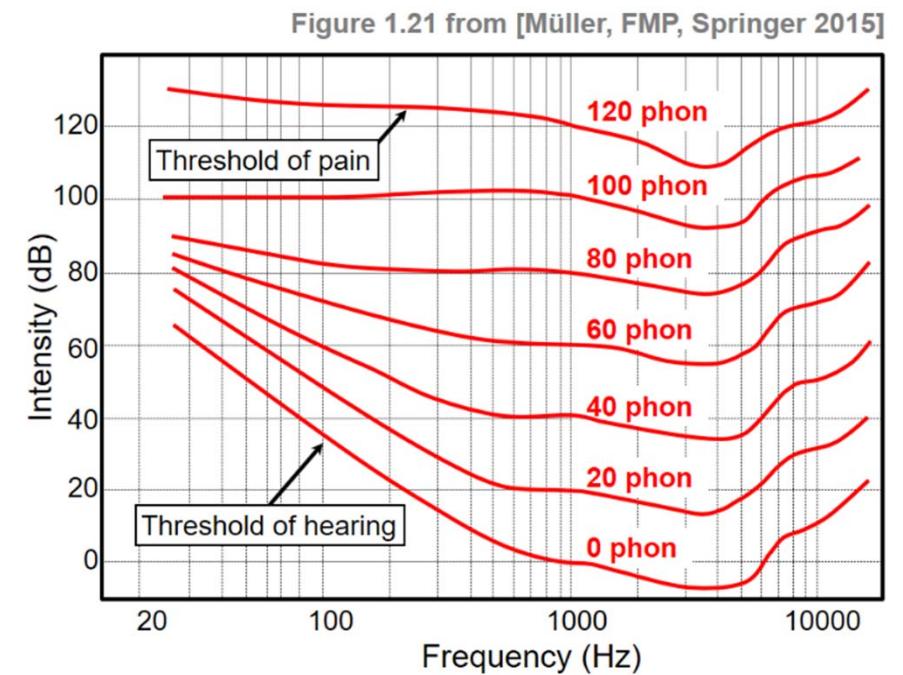
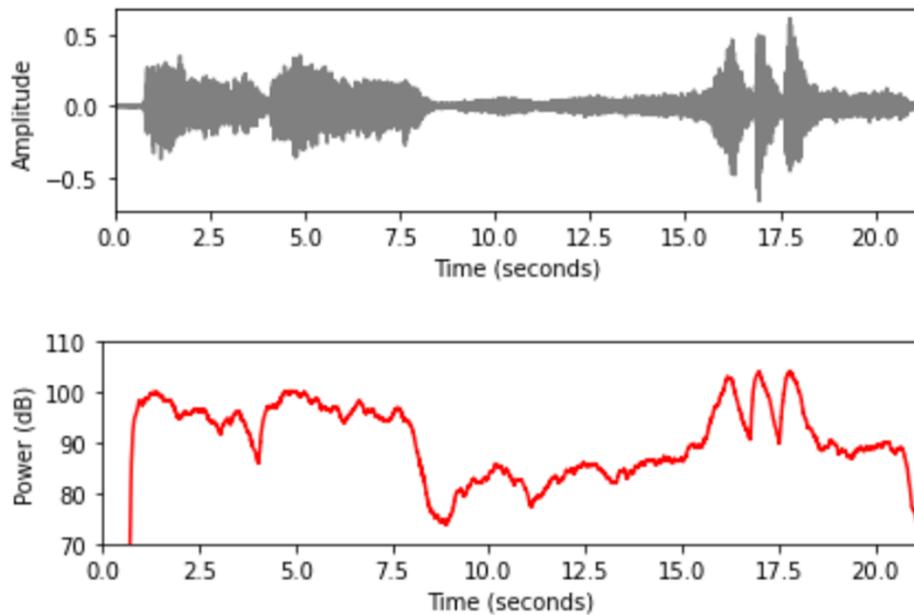
# What Do We Perceive in Musical Audio?

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

# Dynamics, Intensity, and Loudness

[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3\\_Dynamics.html](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](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3_Timbre.html)

- **Timbre** (tone color; tone quality)
  - Allows a listener to distinguish the musical tone of a violin, an oboe, or a trumpet 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*

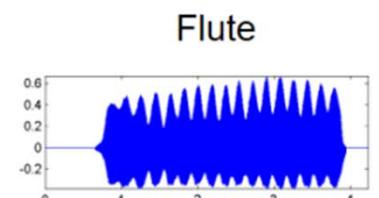
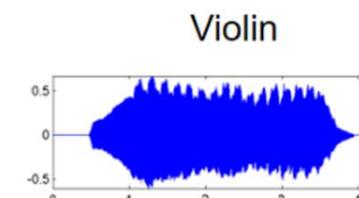
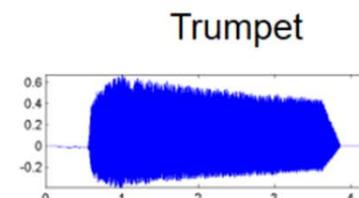
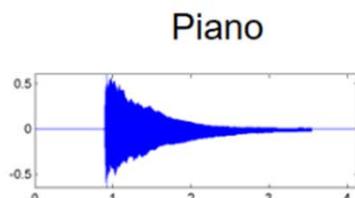


Figure 1.23 from [Müller, FMP, Springer 2015]

▶ 0:04 / 0:04 - 🔊

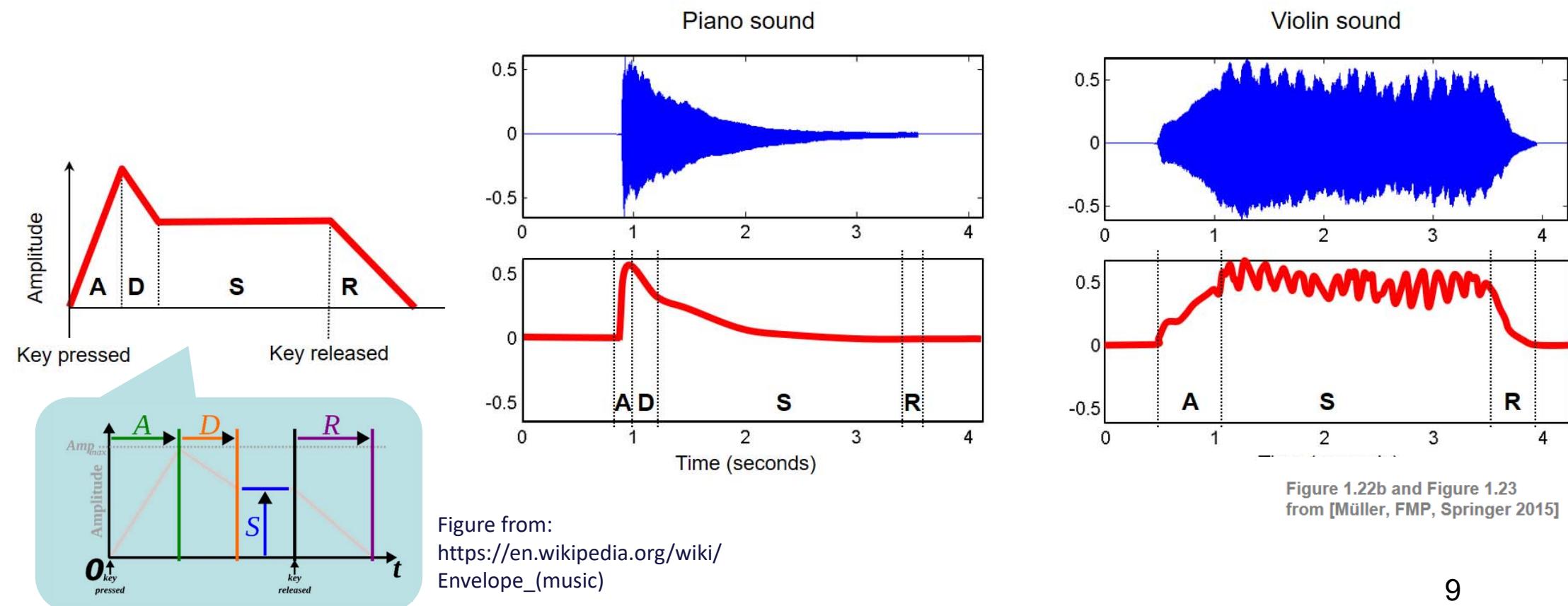
▶ 0:00 / 0:04 - 🔊

▶ 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



# Notes and Pitches

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

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

Note name	Keyboard	Frequency Hz	
A0		27.500	
B0		30.868	29.135
C1		32.703	
D1		36.708	34.648
E1		41.203	38.891
F1		43.654	
G1		48.999	46.249
A1		55.000	51.913
B1		61.735	58.270
C2		65.406	
D2		73.416	69.296
E2		82.407	77.782
F2		87.307	
G2		97.999	92.499
A2		110.00	103.83
B2		123.47	116.54
C3		130.81	
D3		146.83	138.59
E3		164.81	155.56
F3		174.61	
G3		196.00	185.00
A3		220.00	207.65
B3		246.94	233.08
C4		261.63	
D4		293.67	277.18
E4		329.63	311.13
F4		349.23	
G4		392.00	369.99
A4		440.00	415.30
B4		493.88	466.16
C5		523.25	
D5		587.33	554.37
E5		659.26	622.25
F5		698.46	
G5		783.99	739.99
A5		880.00	830.61
B5		987.77	932.33
C6		1046.5	
			1100.2
			J. Wolfe, UNSW
			4186.0

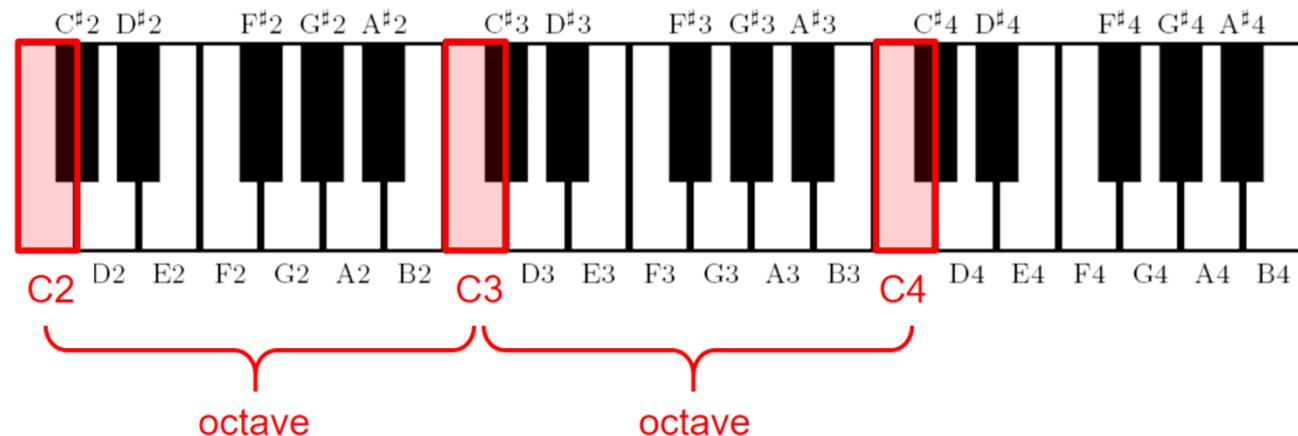
# Notes and Pitches

[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S1\\_MusicalNotesPitches.html](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*

Pitch class C = {..., C2, C3, C4, ...}



# MIDI Note Numbers

[https://en.wikipedia.org/wiki/MIDI\\_tuning\\_standard](https://en.wikipedia.org/wiki/MIDI_tuning_standard)

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

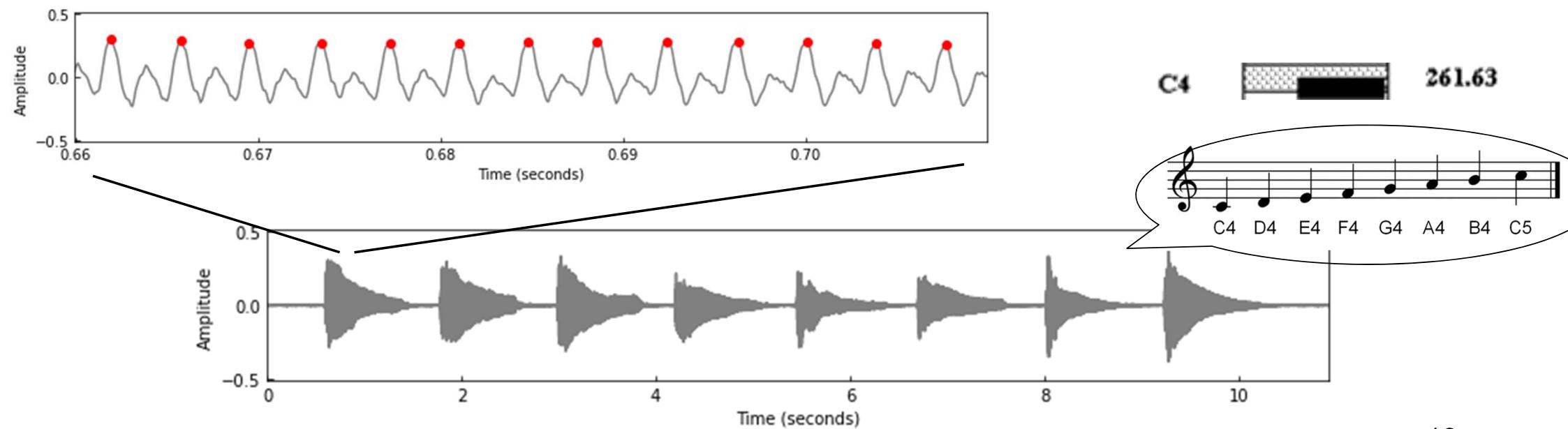
MIDI number	Note name	Keyboard	Frequency Hz							
21 22	A0		27.500		55	56	G2		220.00	207.65
23	B0		30.868	29.135	57	58	A3		246.94	233.08
24 25	C1		32.703		59		B3			
26 27	D1		36.708	34.648	60	61	C4		261.63	
28	E1		41.203	38.891	62	63	D4		293.67	277.18
29 30	F1		43.654		64		E4		329.63	311.13
31 32	G1		48.999	46.249	65	66	F4		349.23	
33 34	A1		55.000	51.913	67	68	G4		392.00	369.99
35	B1		61.735	58.270	69	70	A4		440.00	415.30
36 37	C2		65.406		71		B4		493.68	466.16
38 39	D2		73.416	69.296	72	73	C5			
40	E2		82.407	77.782	74	75	D5		523.25	
41 42	F2		87.307		76		E5		587.33	554.37
43 44	G2		97.999	92.499	77	78	F5		659.26	622.25
45 46	A2		110.00	103.83	79	80	G5		698.46	
			123.47	116.54	81	82	A5		783.99	739.99
					83		B5		880.00	830.61
					84	85	C6		987.77	932.33
									1046.5	
									1108.7	1108.7

# Audio Waveforms (Cont')

[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3\\_Waveform.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C1/C1S3_Waveform.html)

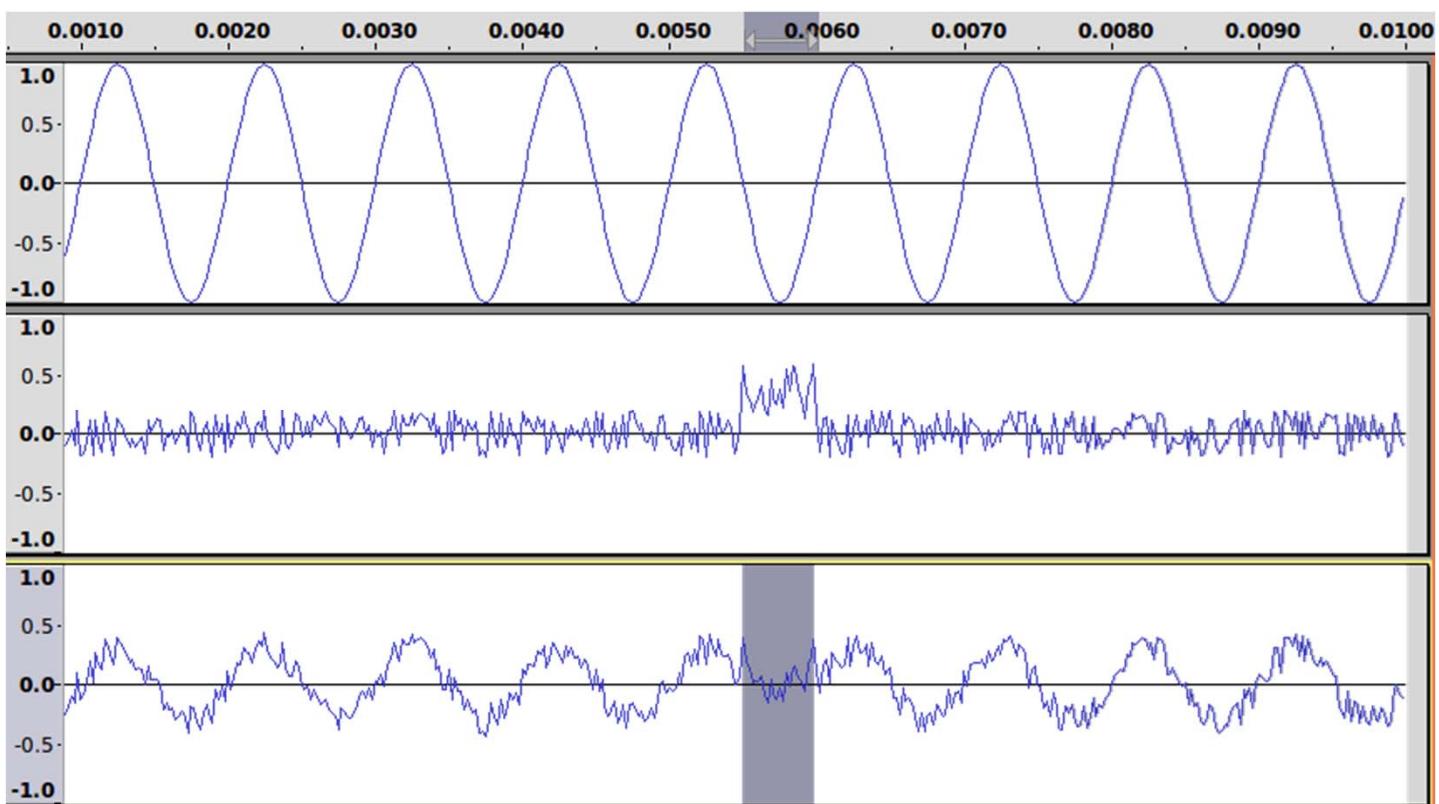
Zoom-in of the section between 0.66 and 0.71 seconds

- Highly repetitive
- 13 high-pressure (red) points  $\rightarrow 20 * 13 = 260$  Hz



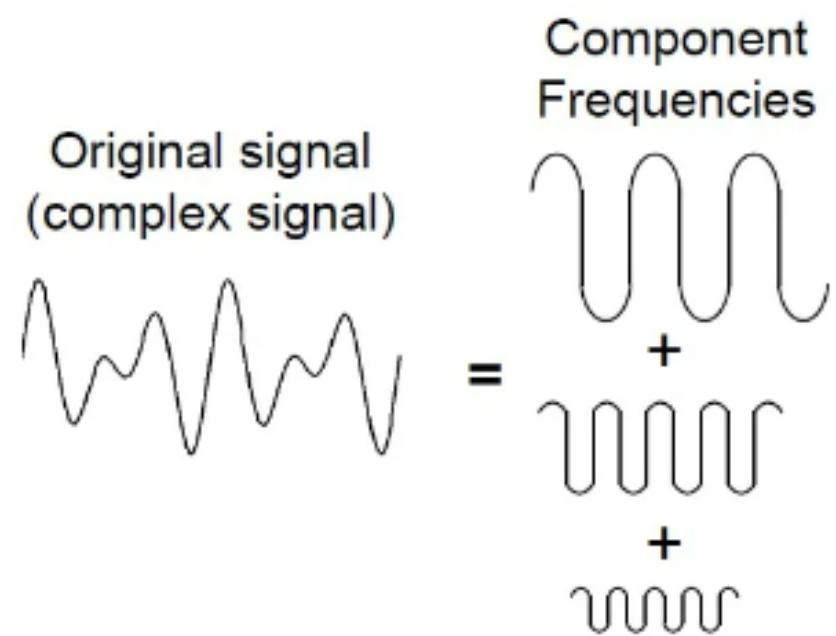
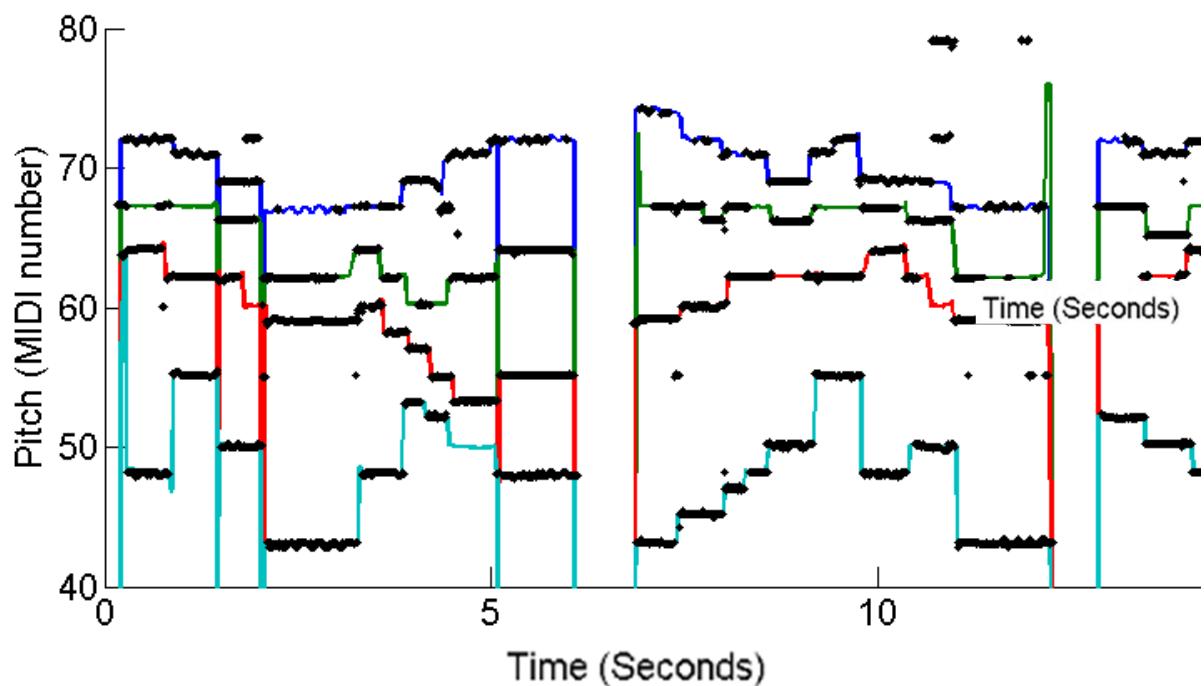
# Temporal Characteristics

- **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/>

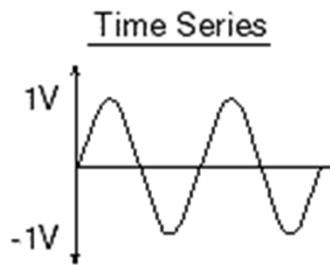
# Frequency-domain Analysis

- Main subject of “Signals and Systems”
- Fourier Transform

Description

A pure 5kHz sine wave measuring 1 volt peak

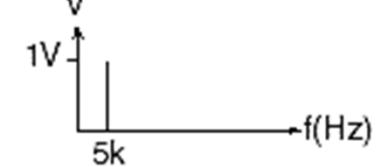
Time Series



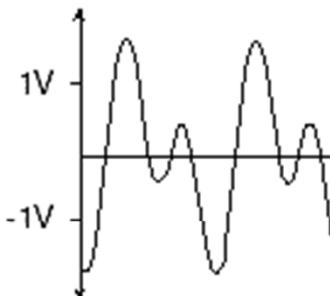
Fourier Expansion

$$v(t) = 1\sin(\omega_1)t$$
$$\omega_1 = 2\pi(5\text{kHz})$$

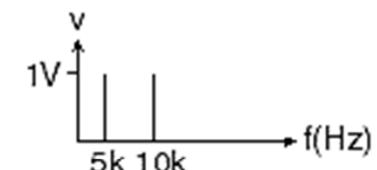
Power Spectrum



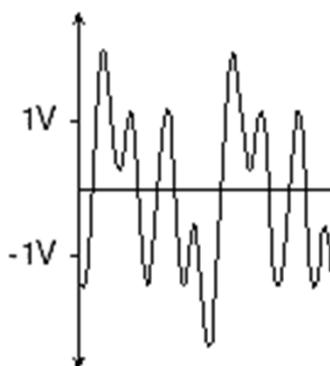
A pure 5kHz and 10kHz sine wave, each measuring 1 volt peak, added together



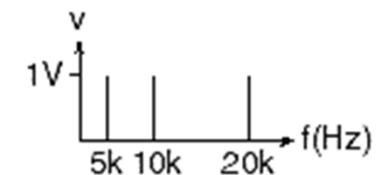
$$v(t) = 1\sin(\omega_1)t + 1\sin(\omega_2)t$$
$$\omega_1 = 2\pi(5\text{kHz})$$
$$\omega_2 = 2\pi(10\text{kHz})$$



A pure 5kHz, 10kHz, and 20kHz sine wave, each measuring 1 volt peak, added together



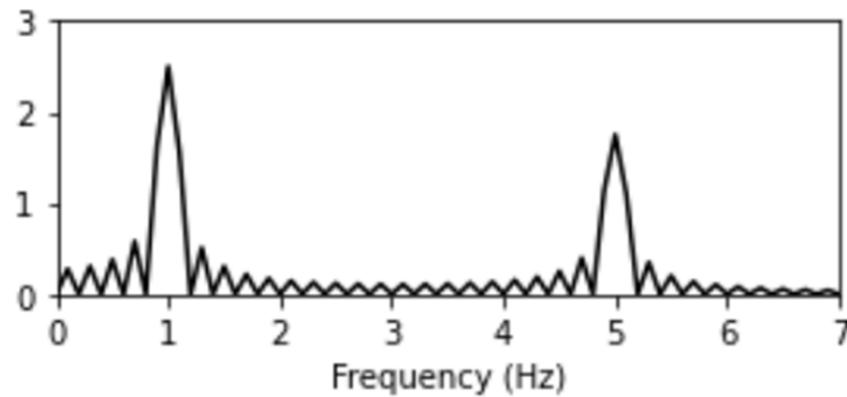
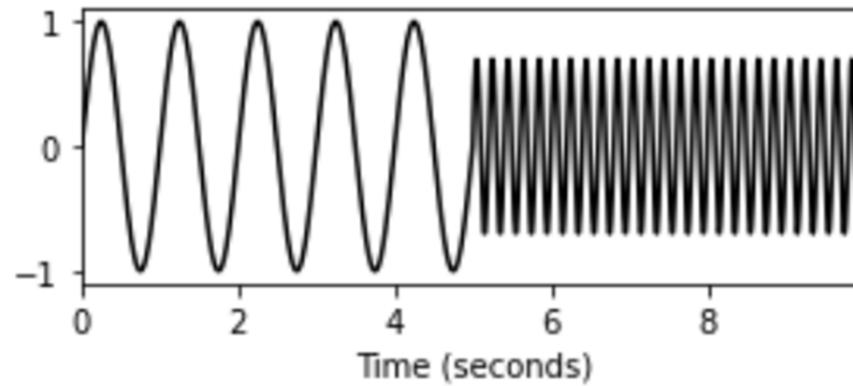
$$v(t) = 1\sin(\omega_1)t + 1\sin(\omega_2)t + 1\sin(\omega_3)t$$
$$\omega_1 = 2\pi(5\text{kHz})$$
$$\omega_2 = 2\pi(10\text{kHz})$$
$$\omega_3 = 2\pi(20\text{kHz})$$



# Frequency-domain Analysis

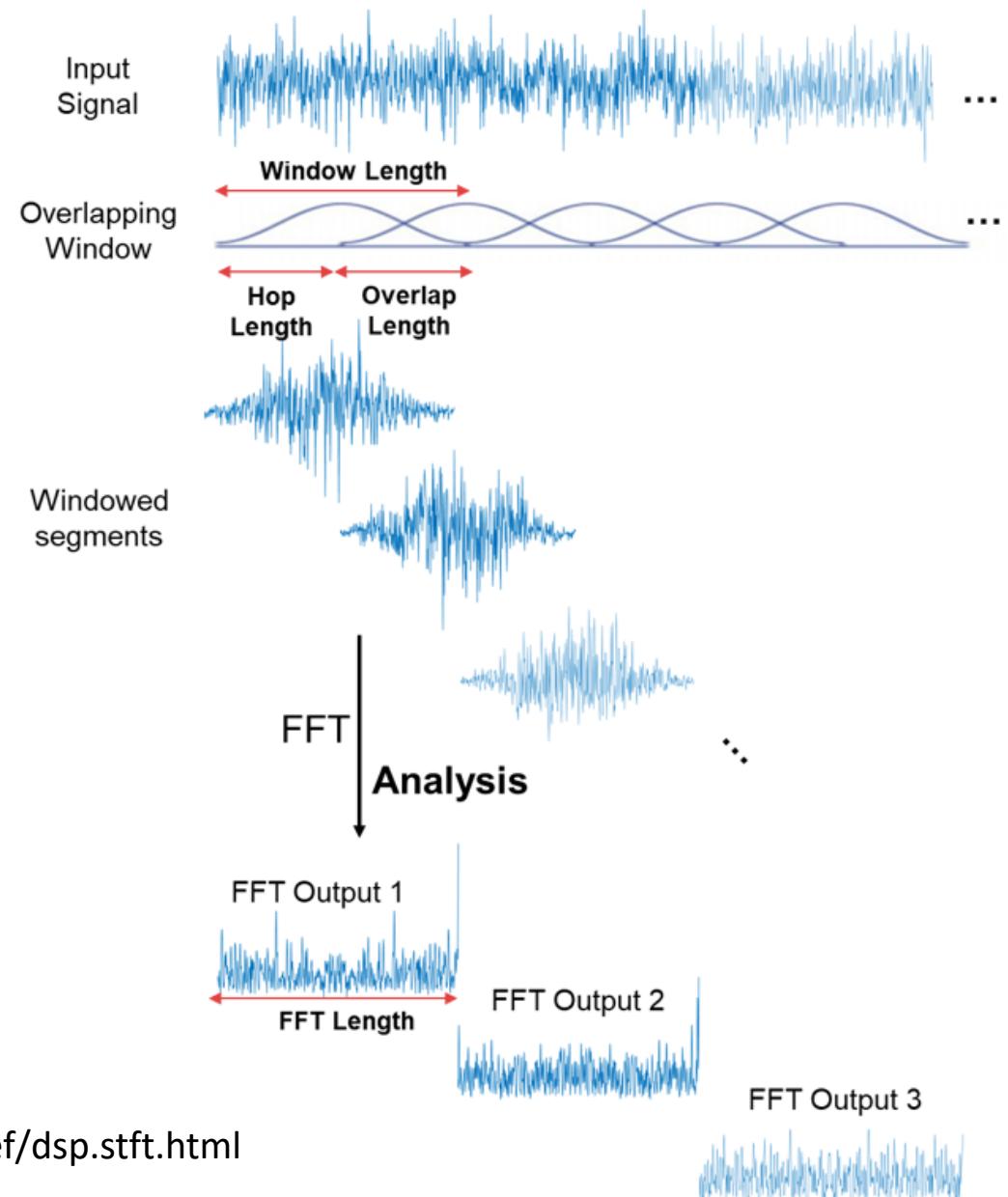
[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2\\_STFT-Basic.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html)

- Fourier Transform cannot localize temporal events



# Frequency-domain Analysis

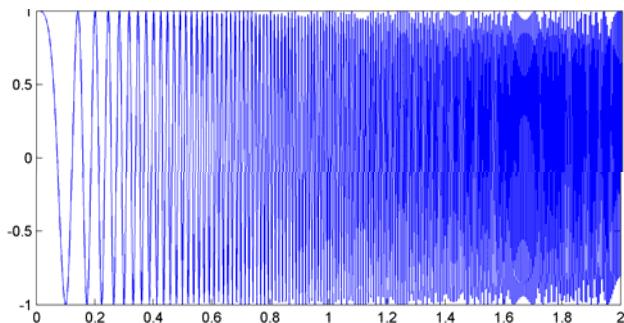
- Fourier Transform (FFT) cannot localize temporal events
- **Short-Time Fourier Transform (STFT)**
  - *Time-frequency representation of signals*
  - Slide **short windows** over the input signal and then compute the FFT locally for each windowed signal



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

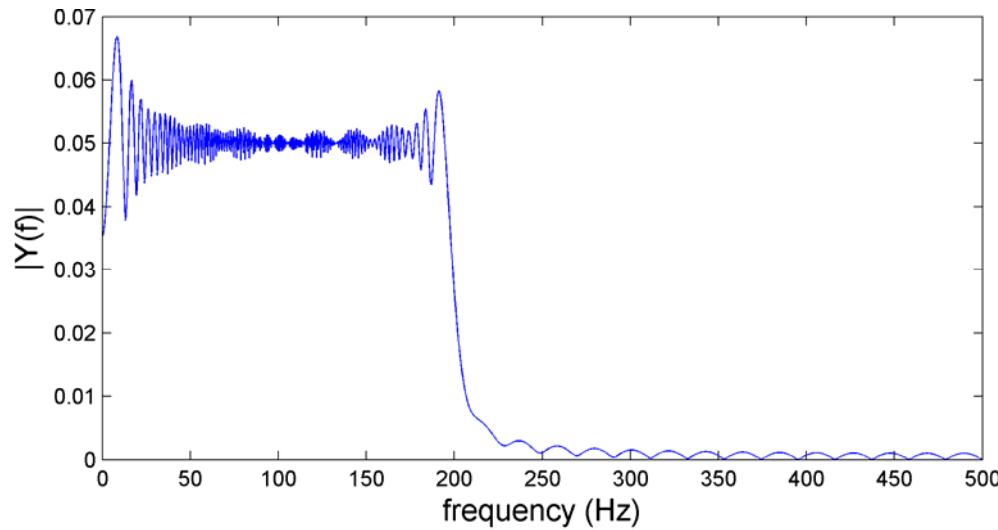
# FFT vs STFT

Time-domain waveform

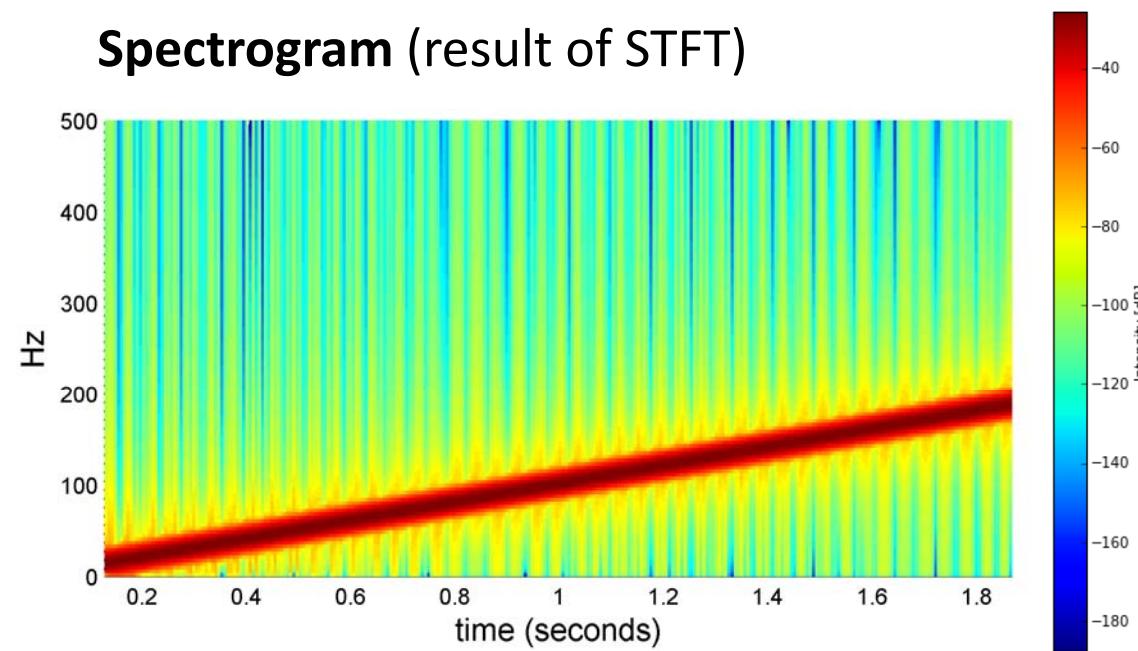


- **Waveform:** amplitude vs time
- **Spectrum:** amplitude vs frequency
- **Spectrogram:** time vs frequency; use *color* for amplitude

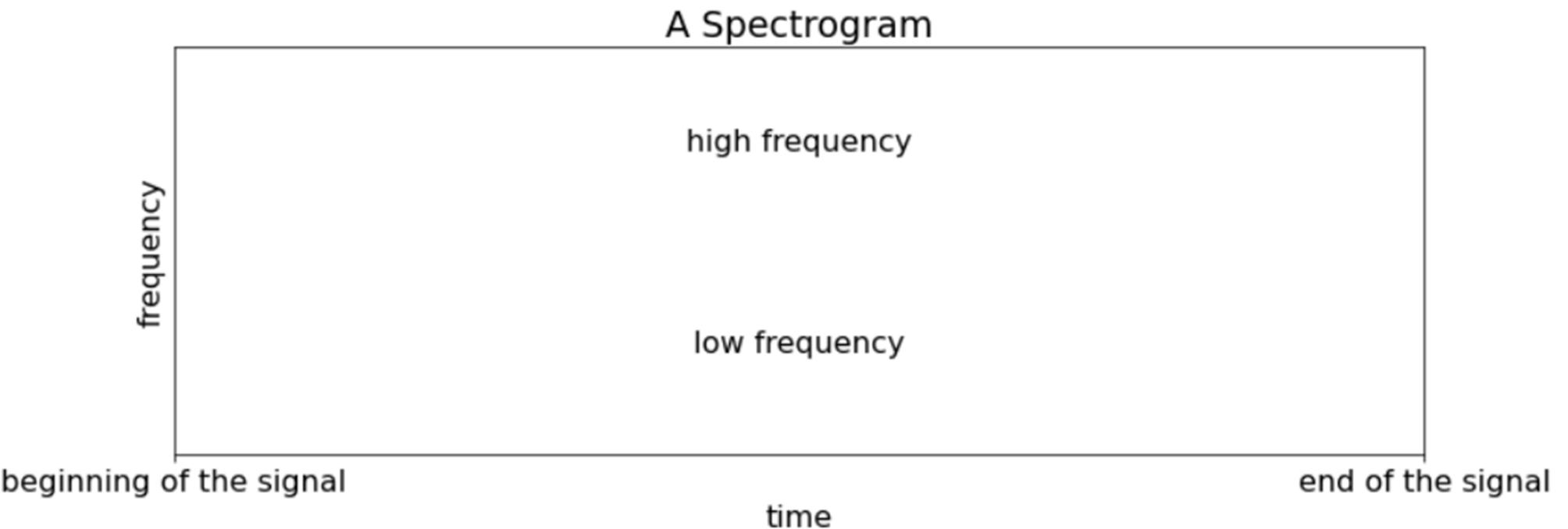
Spectrum (result of FFT)



Spectrogram (result of STFT)



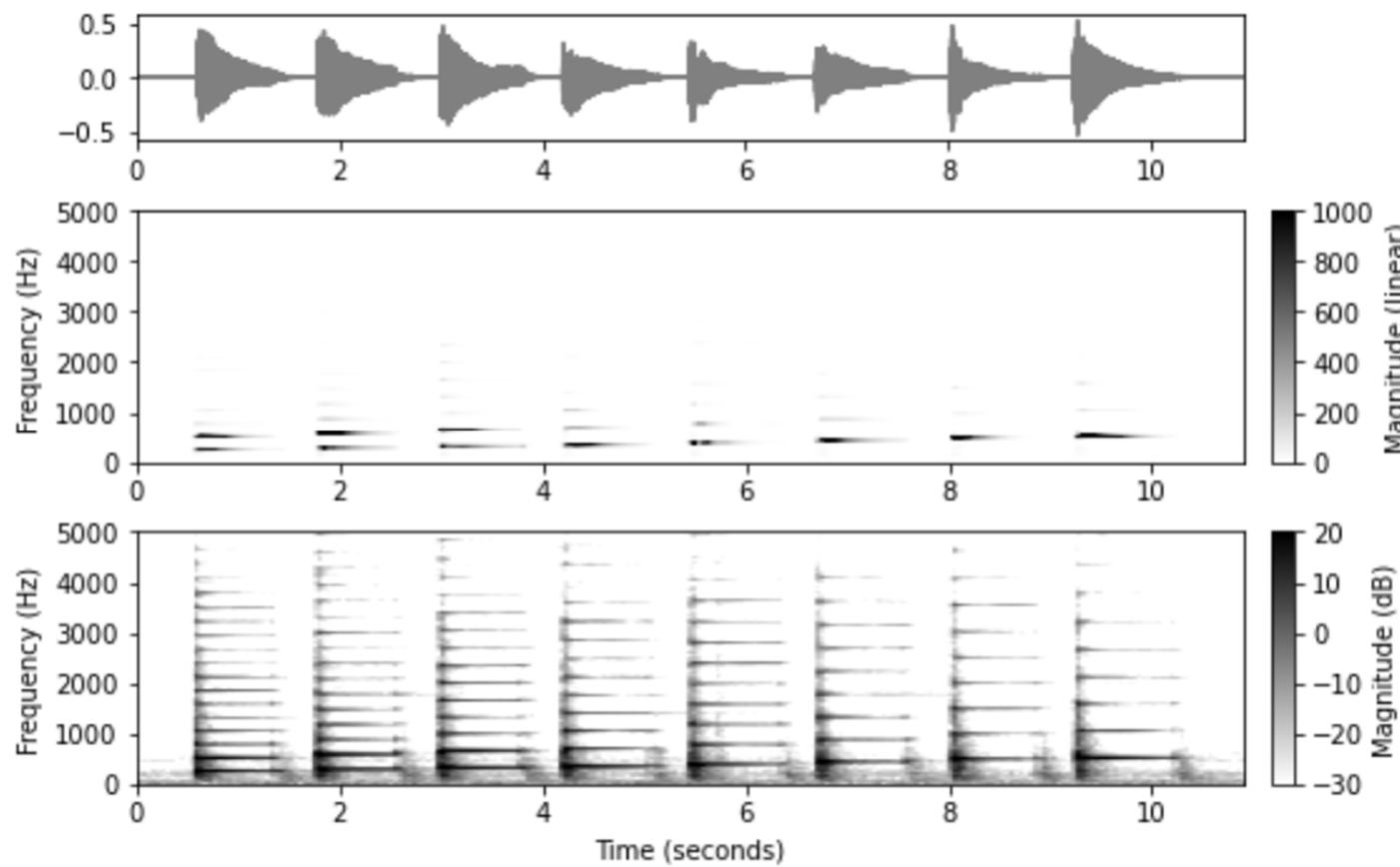
# Time-Frequency Analysis via STFT



[https://music-classification.github.io/tutorial/part2\\_basics/input-representations.html](https://music-classification.github.io/tutorial/part2_basics/input-representations.html)

# Time-Frequency Analysis via STFT

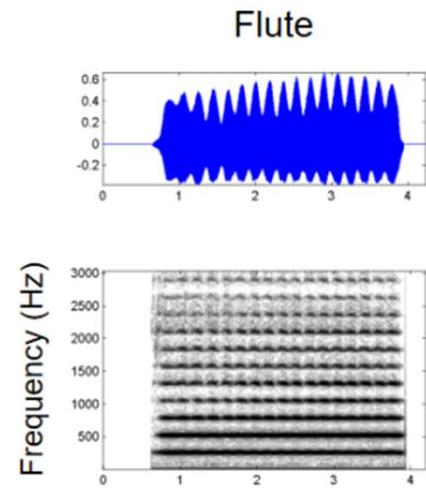
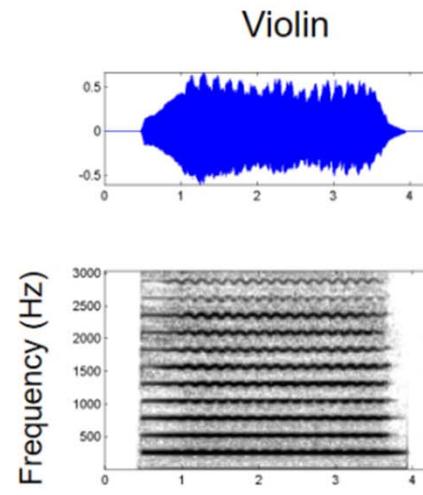
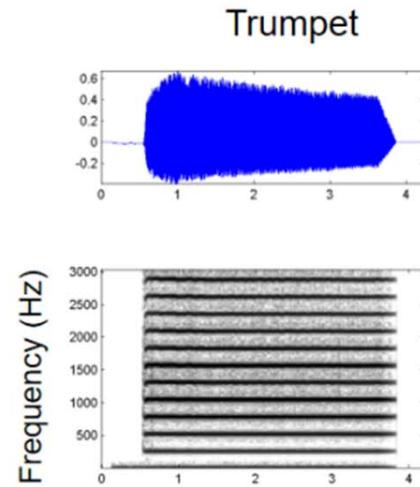
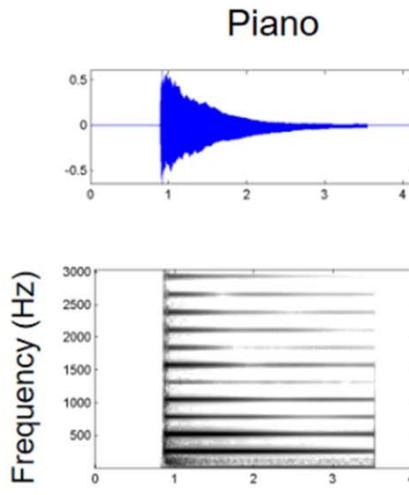
[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2\\_STFT-Basic.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html)



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

# Spectral Characteristics

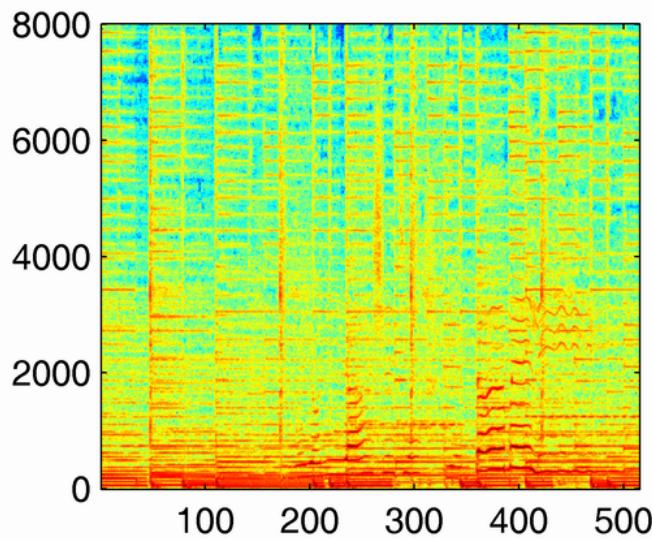
- Fundamental frequency (**F0**)
- **Partials** (harmonics): usually at frequencies which are **integer multiples** of F0
  - The partials in different instruments may exhibit **relative strengths**
  - The frequency may *deviate* from the ideal harmonics (this is called **inharmonicity**)



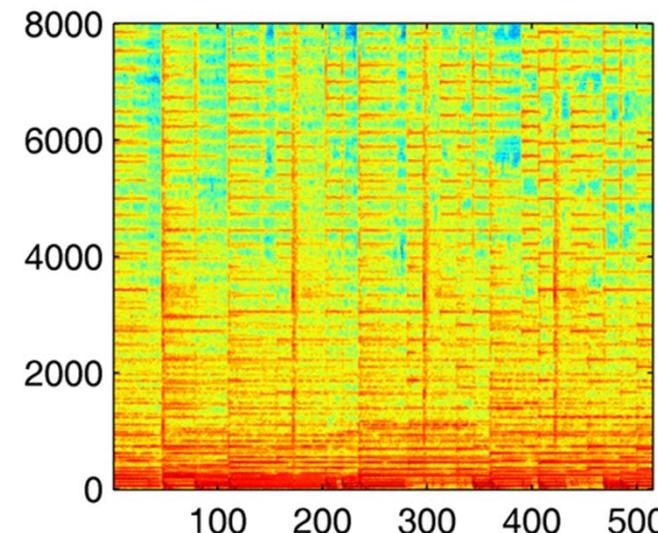
# Time-Frequency Analysis via STFT

- For tasks such as classification/detection and source separation

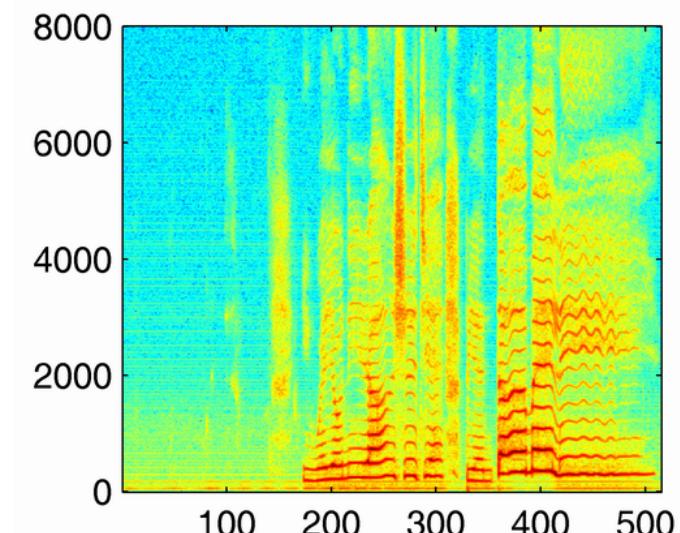
mixture



instrumental (clean)

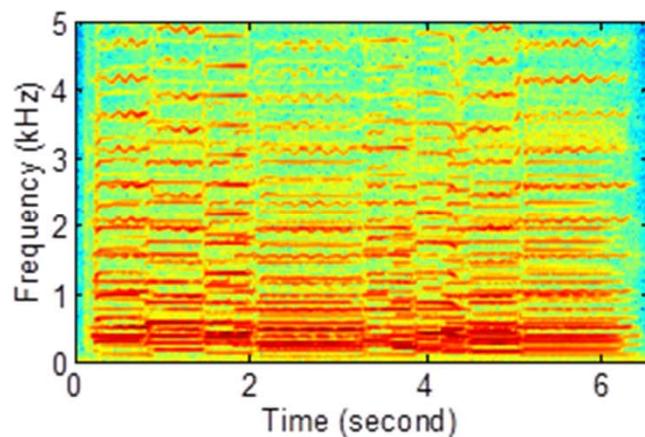


vocal (clean)

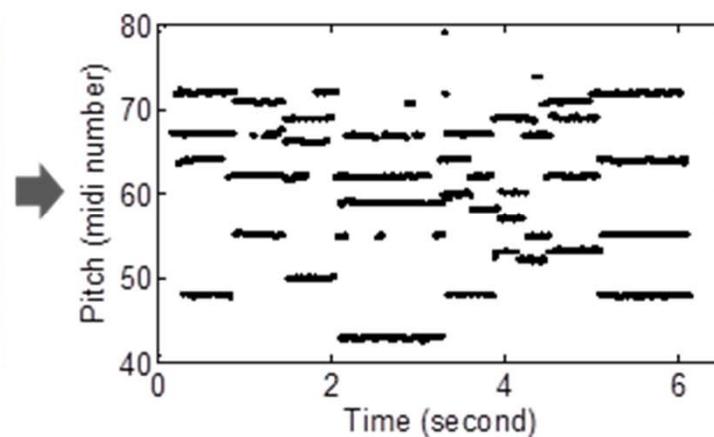


# Time-Frequency Analysis via STFT

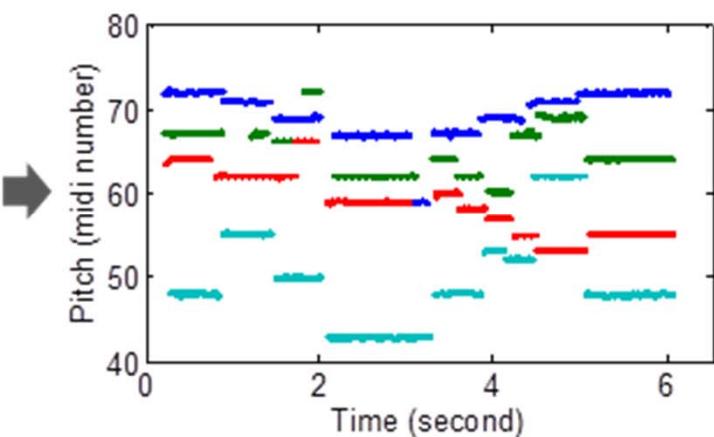
- For tasks such as multi-pitch estimation (MPE) and transcription



Spectrogram



Multi-pitch estimation results



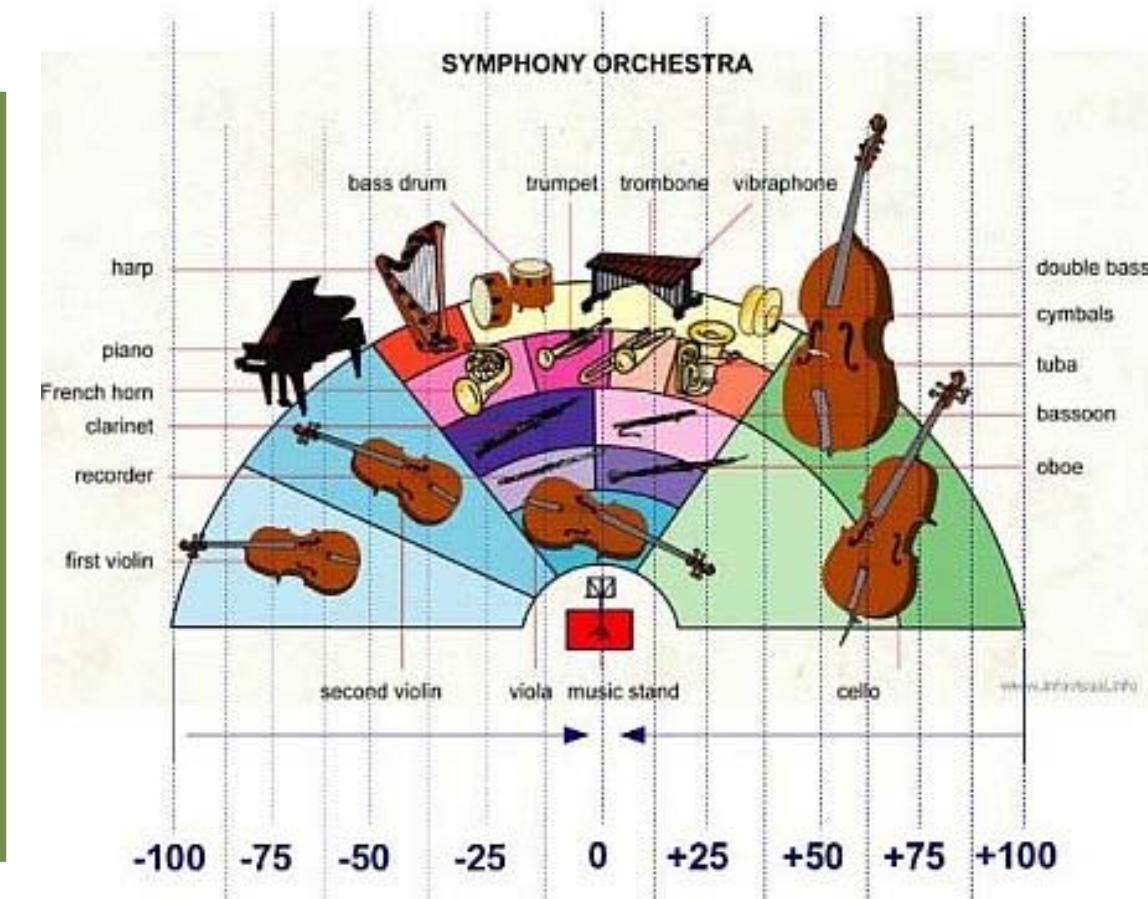
MPE + instrument detection

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

# Spatial Information



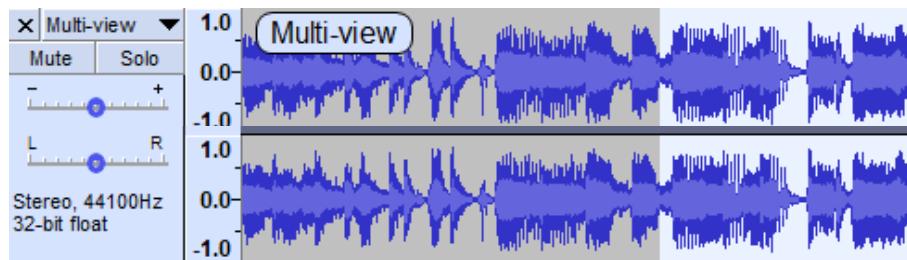
<https://www.renegadeproducer.com/audio-panning.html>



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

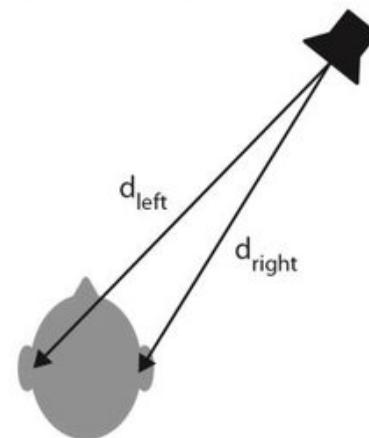
# Spatial Information: Mono vs Stereo

- Monaural
- Binaural

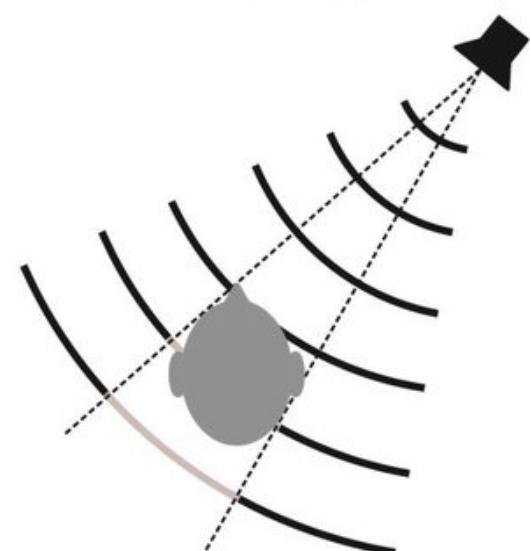


- Stereo-to-mono conversion
  - By taking the average
- Mono-to-stereo conversion
  - Need research  
[https://www.youtube.com/watch?v=aWxmQKm\\_s8Q](https://www.youtube.com/watch?v=aWxmQKm_s8Q)

a) Binaural localization cue:  
interaural time difference (ITD)



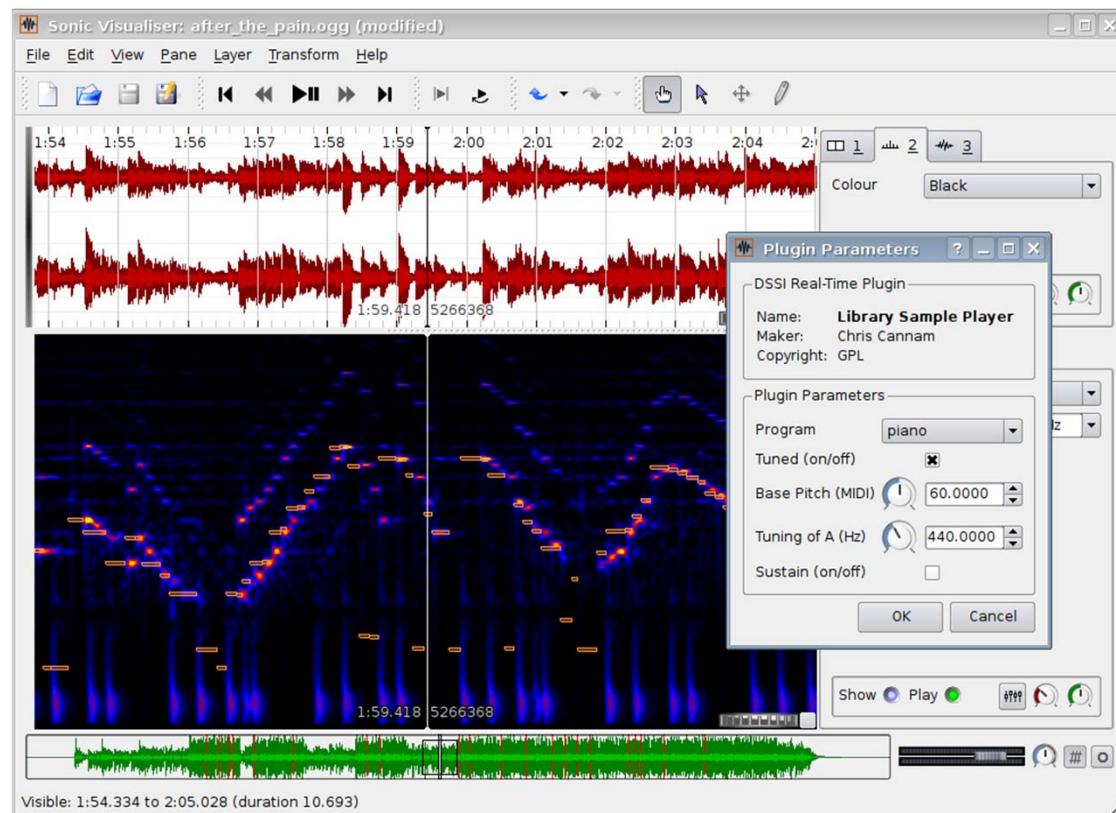
b) Binaural localization cue:  
interaural intensity difference (IID)



Source: [https://www.researchgate.net/figure/Binaural-and-monaural-cues-used-for-sound-localization-a-For-sound-sources-off-the\\_fig1\\_299281975](https://www.researchgate.net/figure/Binaural-and-monaural-cues-used-for-sound-localization-a-For-sound-sources-off-the_fig1_299281975)

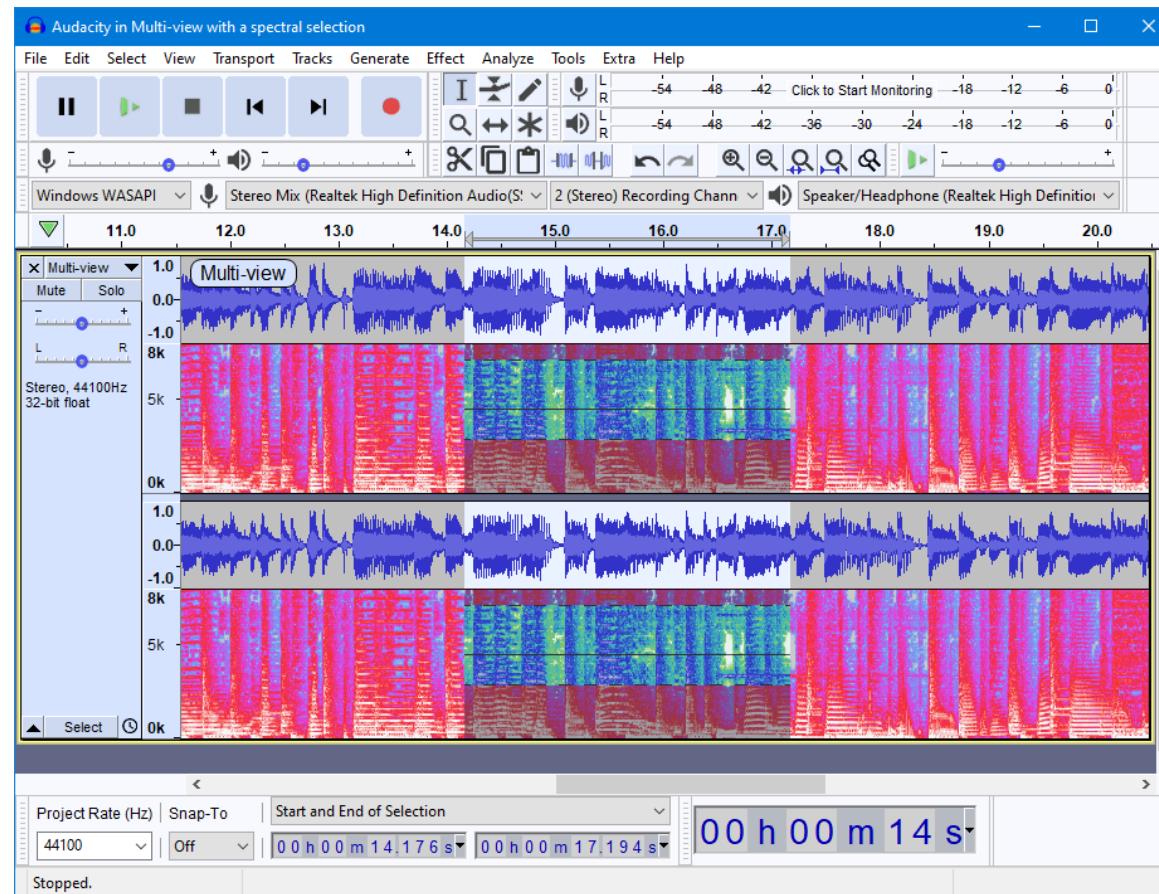
# Software: Sonic Visualiser

<http://www.sonicvisualiser.org/>



# Software: Audacity

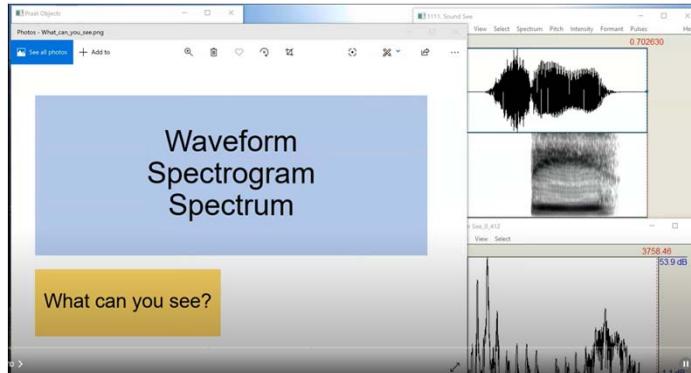
<https://www.audacityteam.org/>



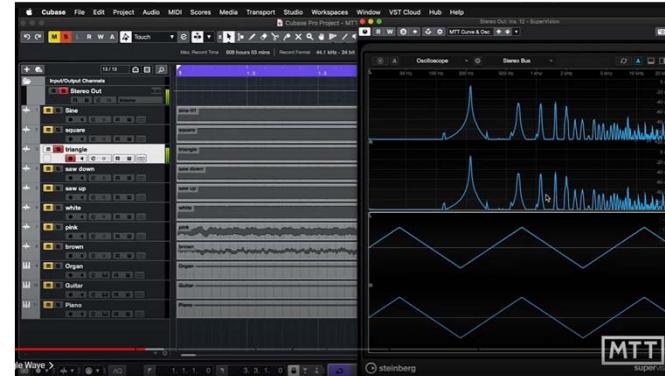
# Let's Find Some Audio Recordings

- <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/Rudmer_Rotteveel/sounds/316915/)
  - <https://www.freesound.org/people/Jaylew1987/sounds/321112/>
  - <https://www.freesound.org/people/mickel11/sounds/90803/>

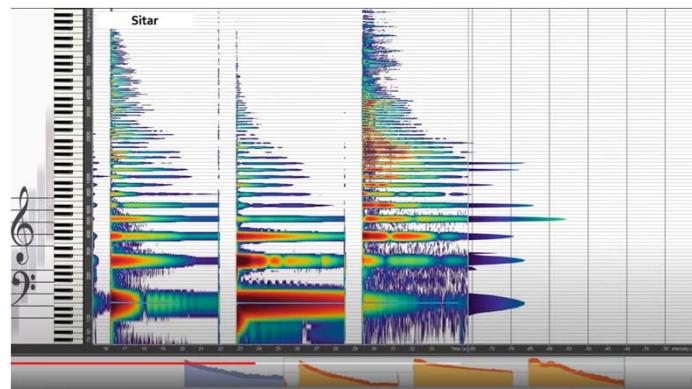
# Related Videos



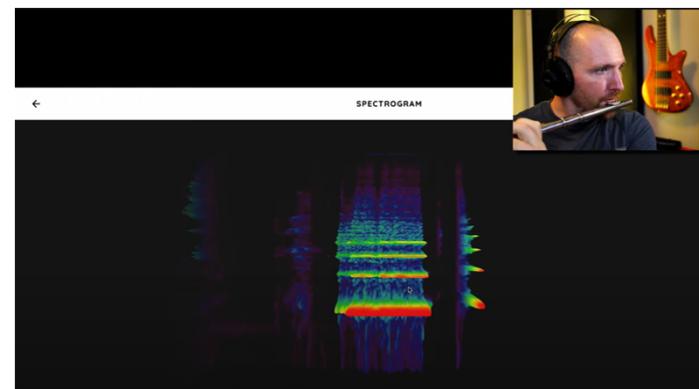
<https://www.youtube.com/watch?v=2Hj1kAWVjLo>



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



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

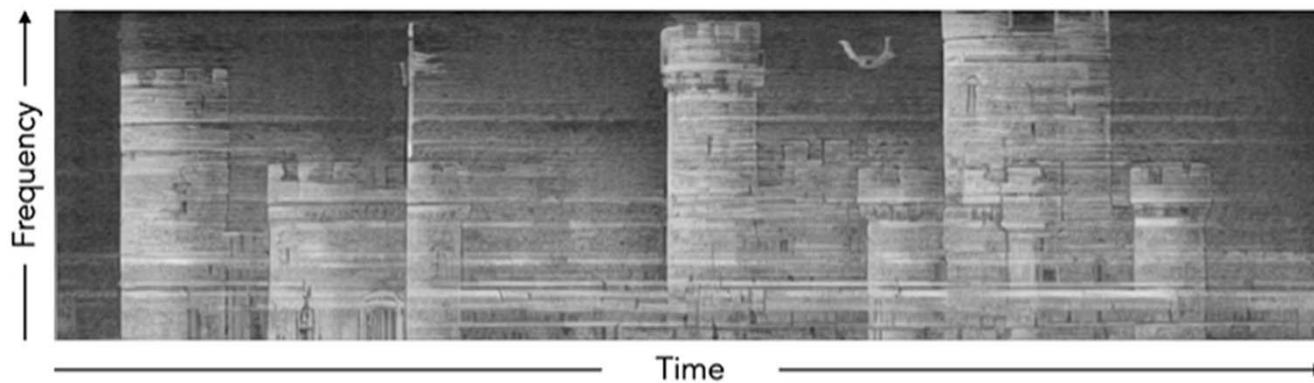


<https://www.youtube.com/watch?v=jEO0GHU3xsU>

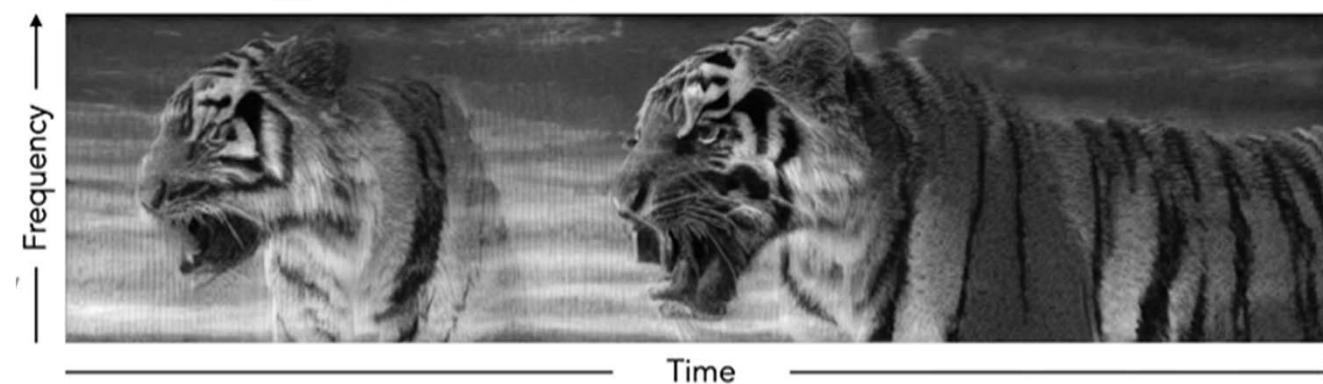
# Fun Stuff: “Images that Sound”

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

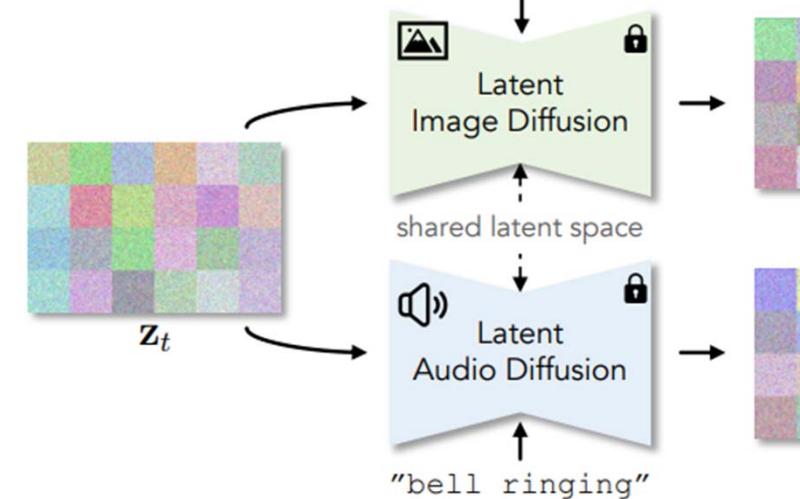
“castle with bell towers, grayscale, lithograph style”



“tiger, grayscale, black background”



“castle with bell towers, grayscale”

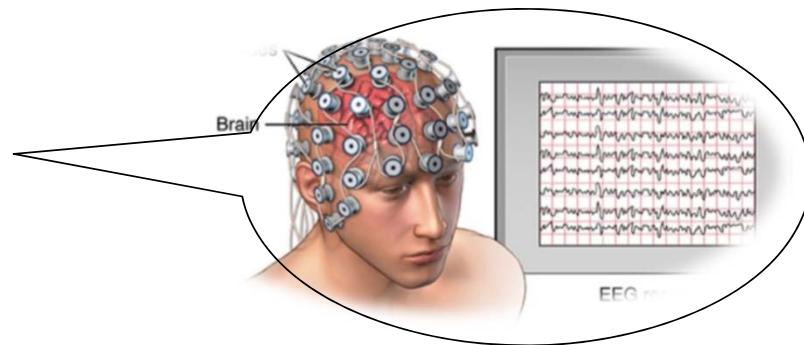


# Outline

- 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**

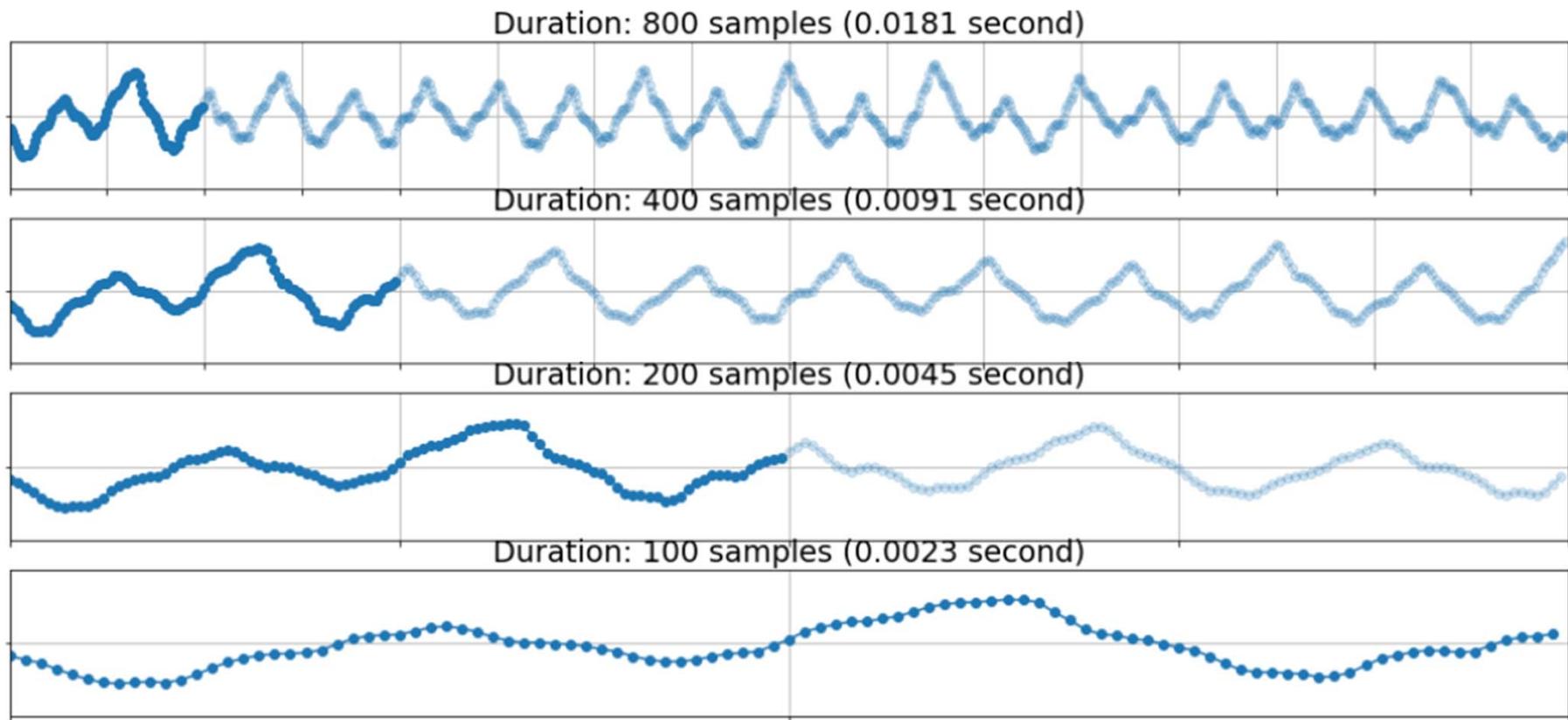


<https://www.brightbraincentre.co.uk/electroencephalogram-eeg-brainwaves/>

```
[ ] 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](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

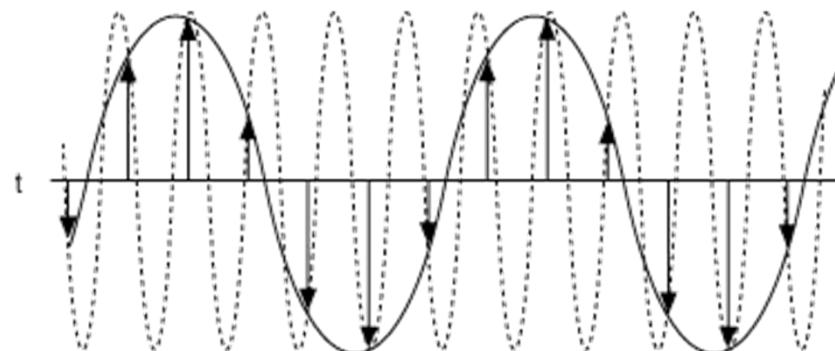
- Usually, a model considered signals with a certain and fixed sampling rate
- Building a **sampling-rate agnostic** model that can take signals sampled at arbitrary sampling rates is an advanced topic

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

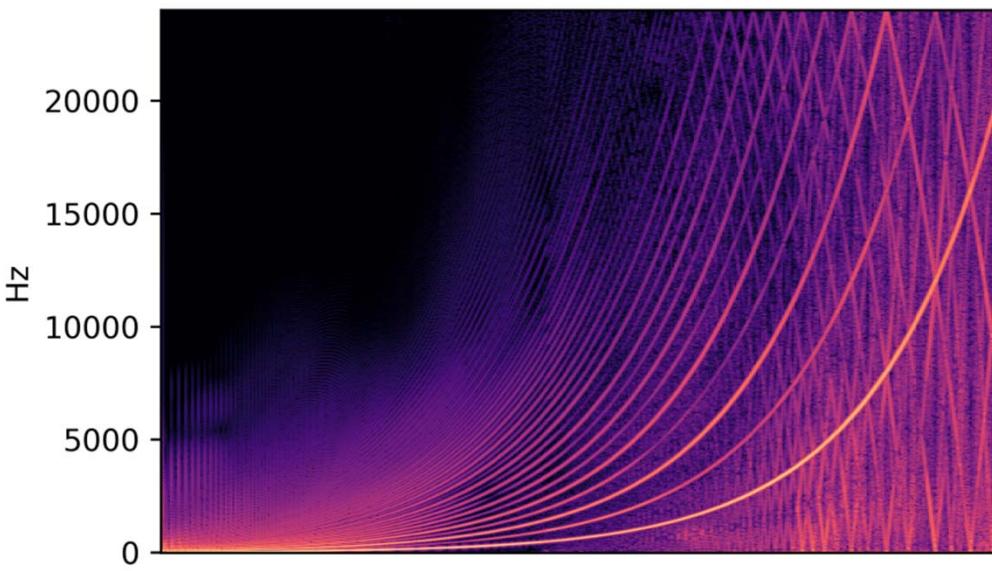
- A signal must be sampled at least **twice** as fast as the bandwidth of the signal to accurately reconstruct the waveform; otherwise, the high-frequency content will alias at a frequency inside the spectrum of interest
- **Sampling freq (Fs) > 2\* the highest freq in the signal**



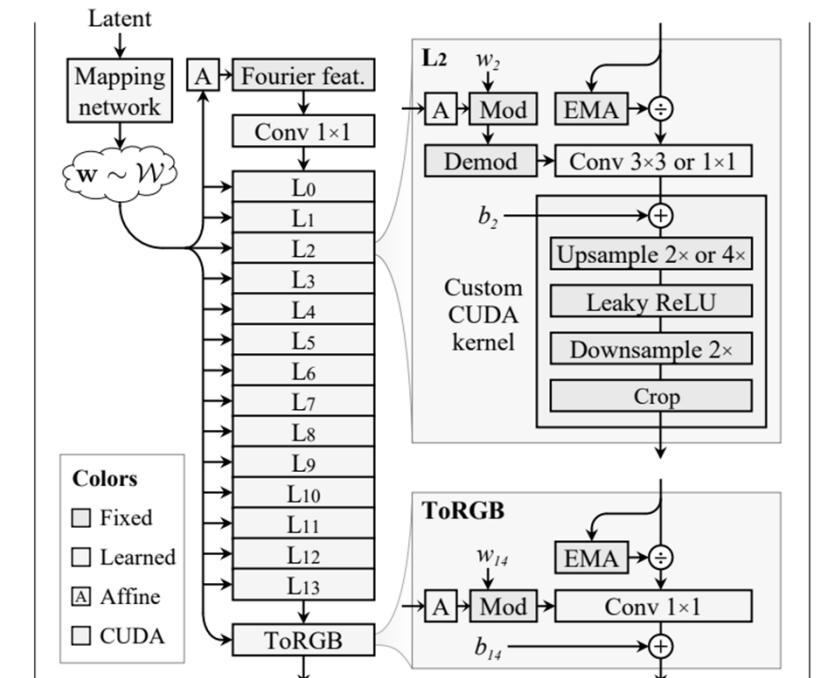
[http://zone.ni.com/reference/en-XX/help/370524T-01/siggenhelp/fund\\_nyquist\\_and\\_shannon\\_theorems/](http://zone.ni.com/reference/en-XX/help/370524T-01/siggenhelp/fund_nyquist_and_shannon_theorems/)

# Aliasing

- High-frequency components fold back



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



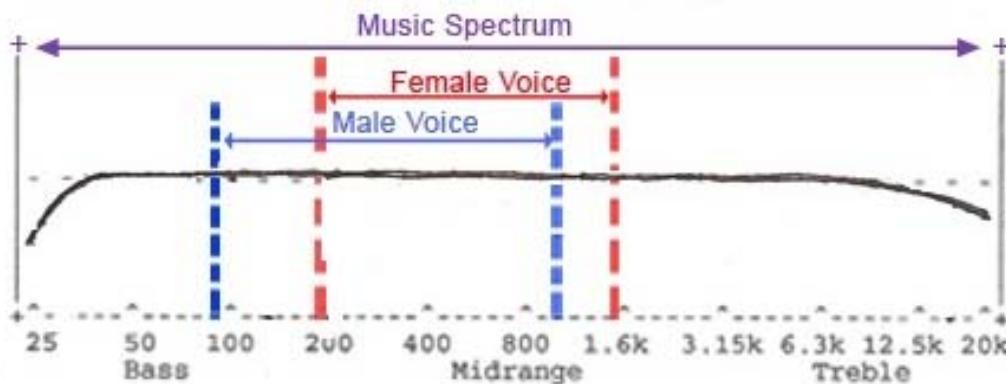
(b) Our alias-free StyleGAN3 generator architecture

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

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

# Nyquist–Shannon Sampling Theorem

- 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  $F_s=128$  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

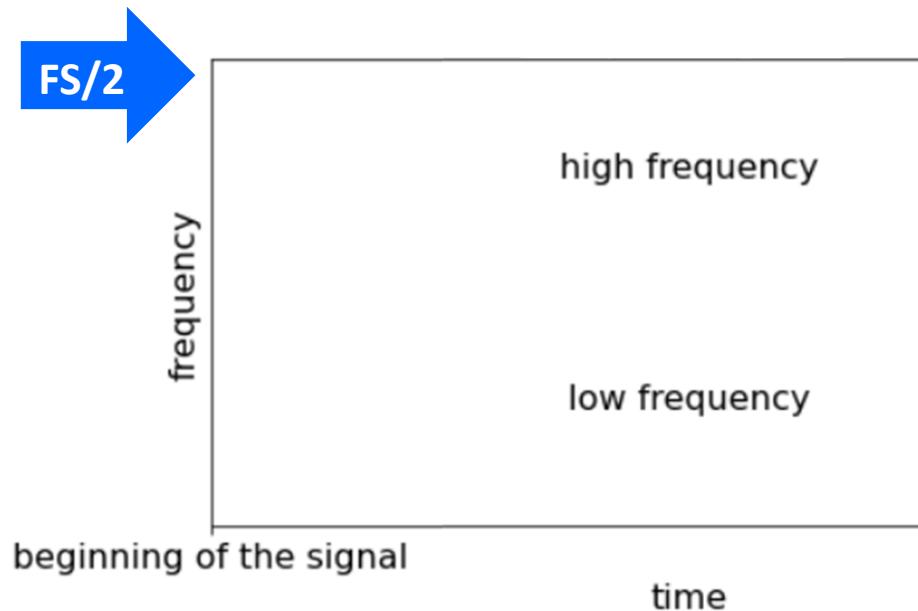
Unconscious		Conscious		
Delta	Theta	Alpha	Beta	Gamma
0,5 – 4 Hz	4 – 8 Hz	8 – 13 Hz	13 – 30 Hz	30-42 Hz
Instinct	Emotion	Consciousness	Thought	Will
Survival Deep sleep Coma	Drives Feelings Trance Dreams	Awareness of the body Integration of feelings	Perception Concentration Mental activity	Extreme focus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

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

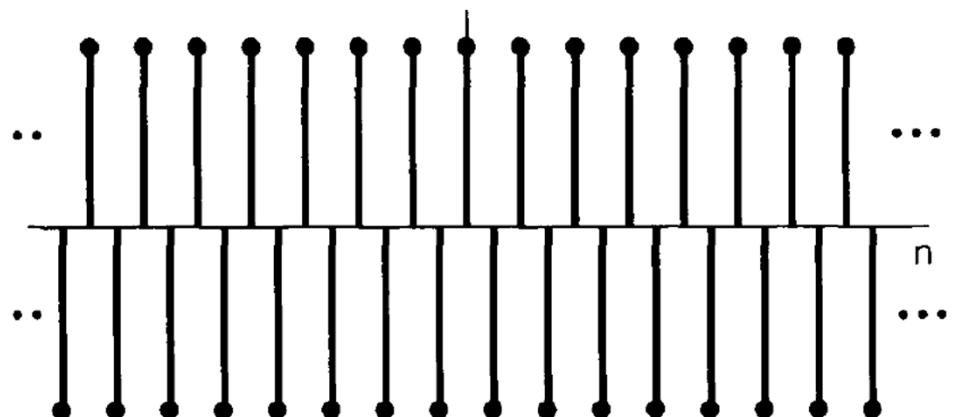
- **Question:** With  $F_s=128$  Hz, we assume that we don't need to care about frequency higher than 64 Hz in brain waves

# Nyquist–Shannon Sampling Theorem

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



Sinusoid with the highest freq given a fixed FS



# DTFS: Fourier Series Representation for Discrete-time Periodic Signals

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

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

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

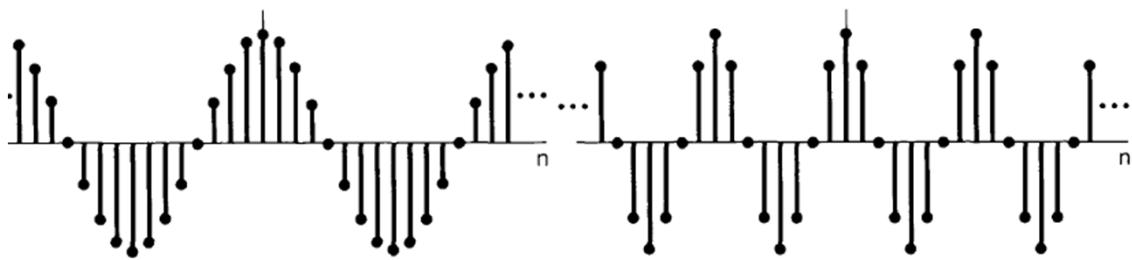
Analysis equation:  $a_k = \frac{1}{L} \sum_{n=\langle L \rangle} x[n] e^{-jk\frac{2\pi}{L}n}$

**Sum over a set of  $L$  successive integers**

# The Basis Functions in DTFS

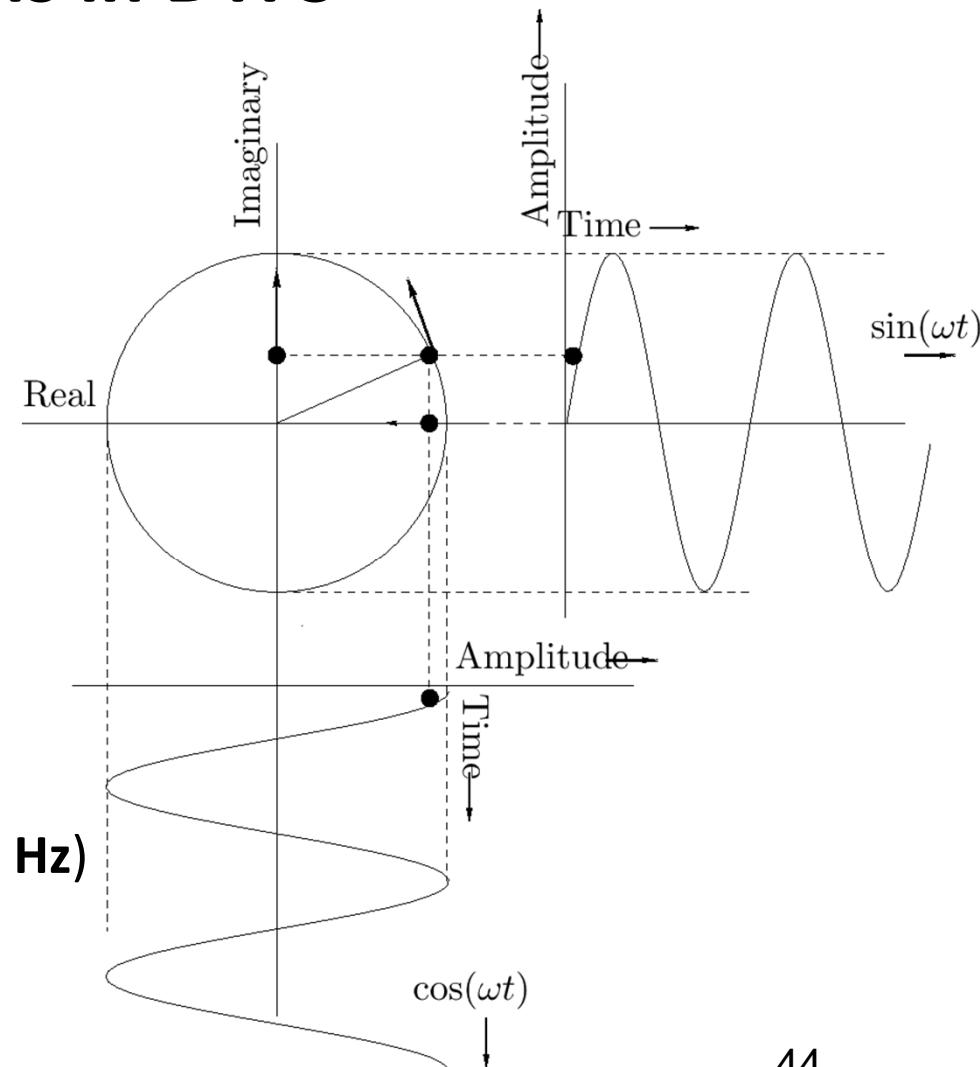
- $e^{j\omega n} = \cos(\omega n) + j\sin(\omega n)$

$$x[n] = \cos(\pi n/8)$$



$$x[n] = \cos(\pi n/4)$$

- angular or ordinary frequency
  - $\omega$ : 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  $x[n] \xrightarrow{\mathcal{FS}} a_k$

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

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

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

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

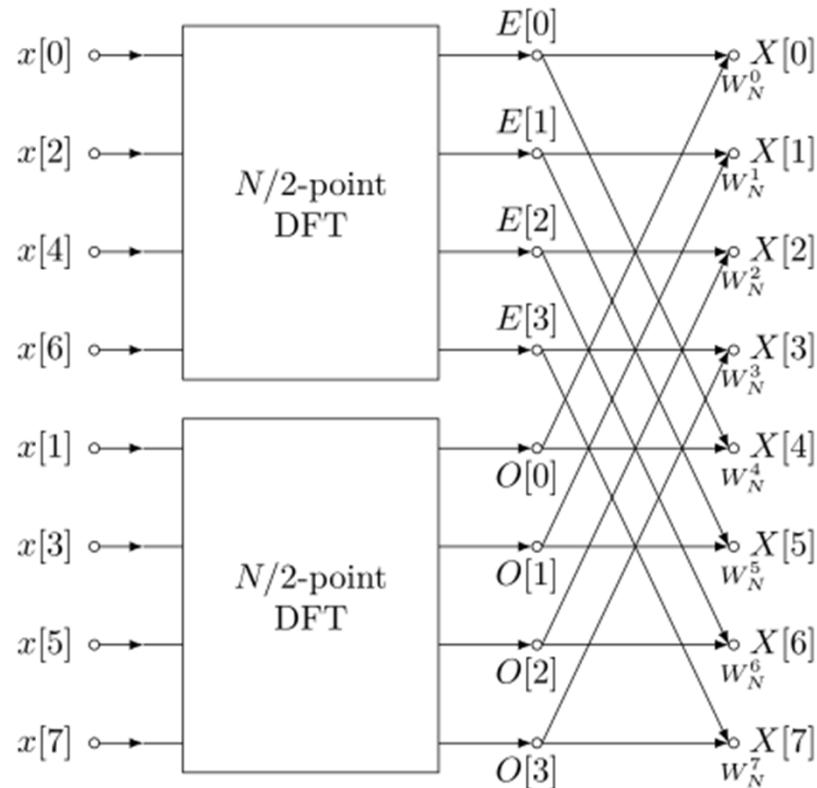
**Analysis:**  $X_k = \frac{1}{L} \sum_{n=0}^{L-1} x[n] e^{-jk\omega n}$

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

# DFT and FFT (Fast Fourier Transform)

[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2\\_DFT-FFT.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_DFT-FFT.html)

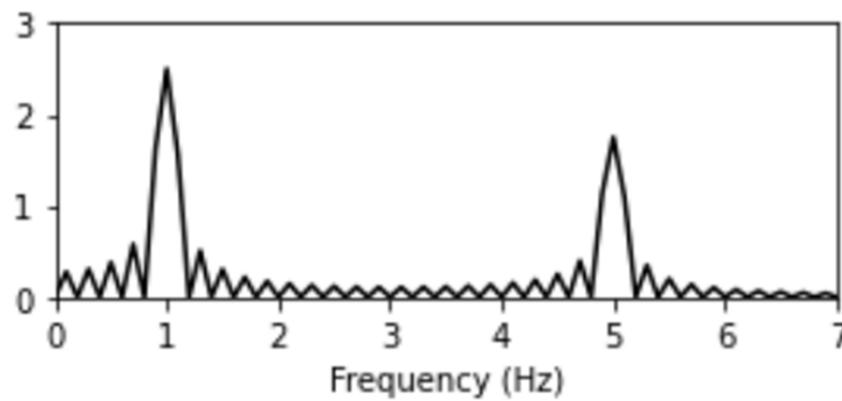
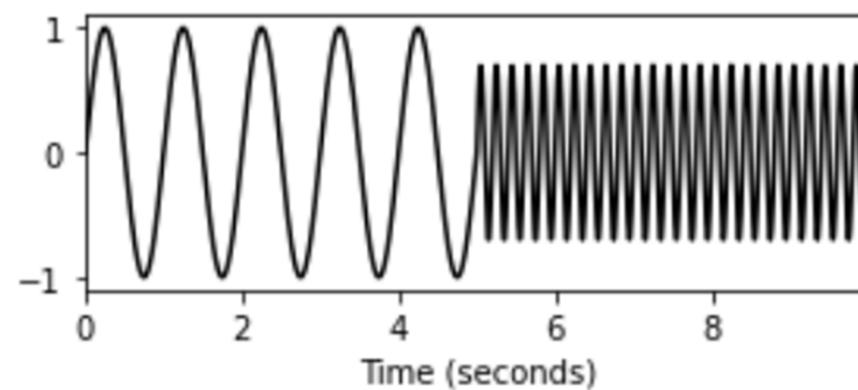
- FFT is a fast algorithm to compute the DFT



[https://en.wikipedia.org/wiki/  
Fast\\_Fourier\\_transform](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](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html)



# STFT: Short-time Fourier Transform

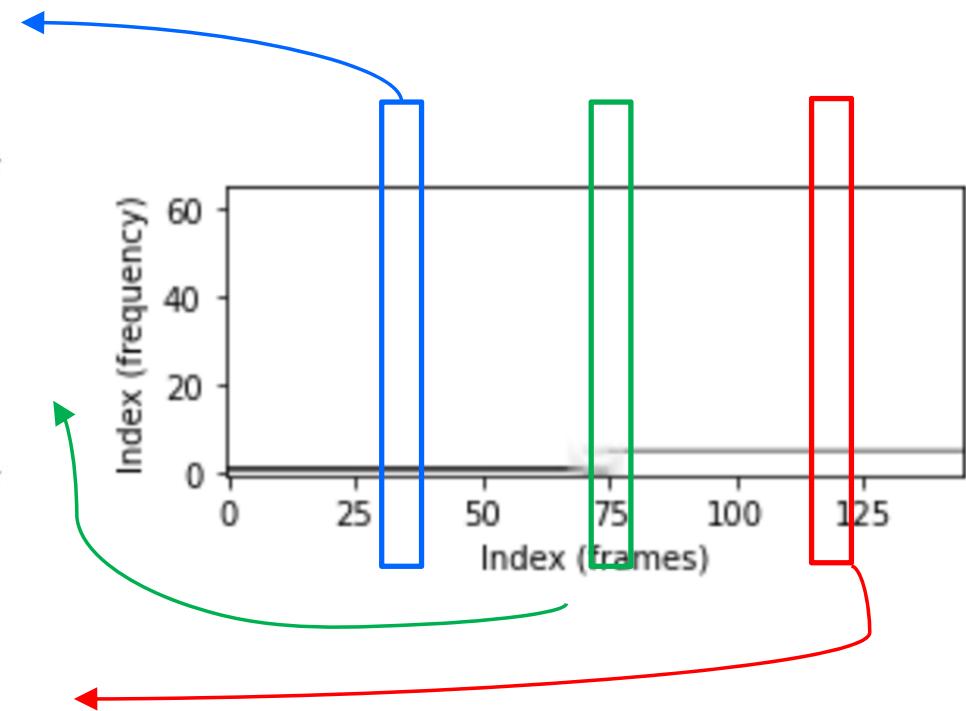
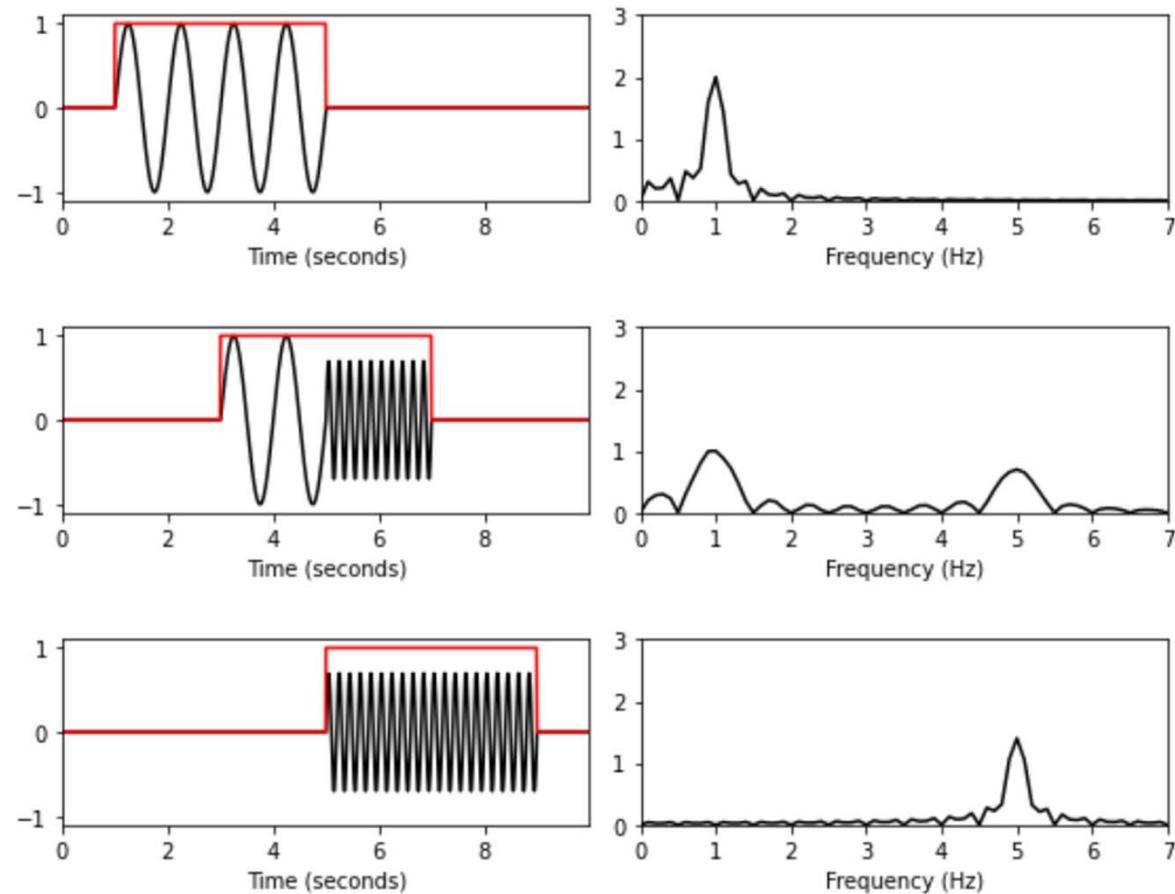
[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2\\_STFT-Basic.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html)

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

- $x[n]$ : 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”

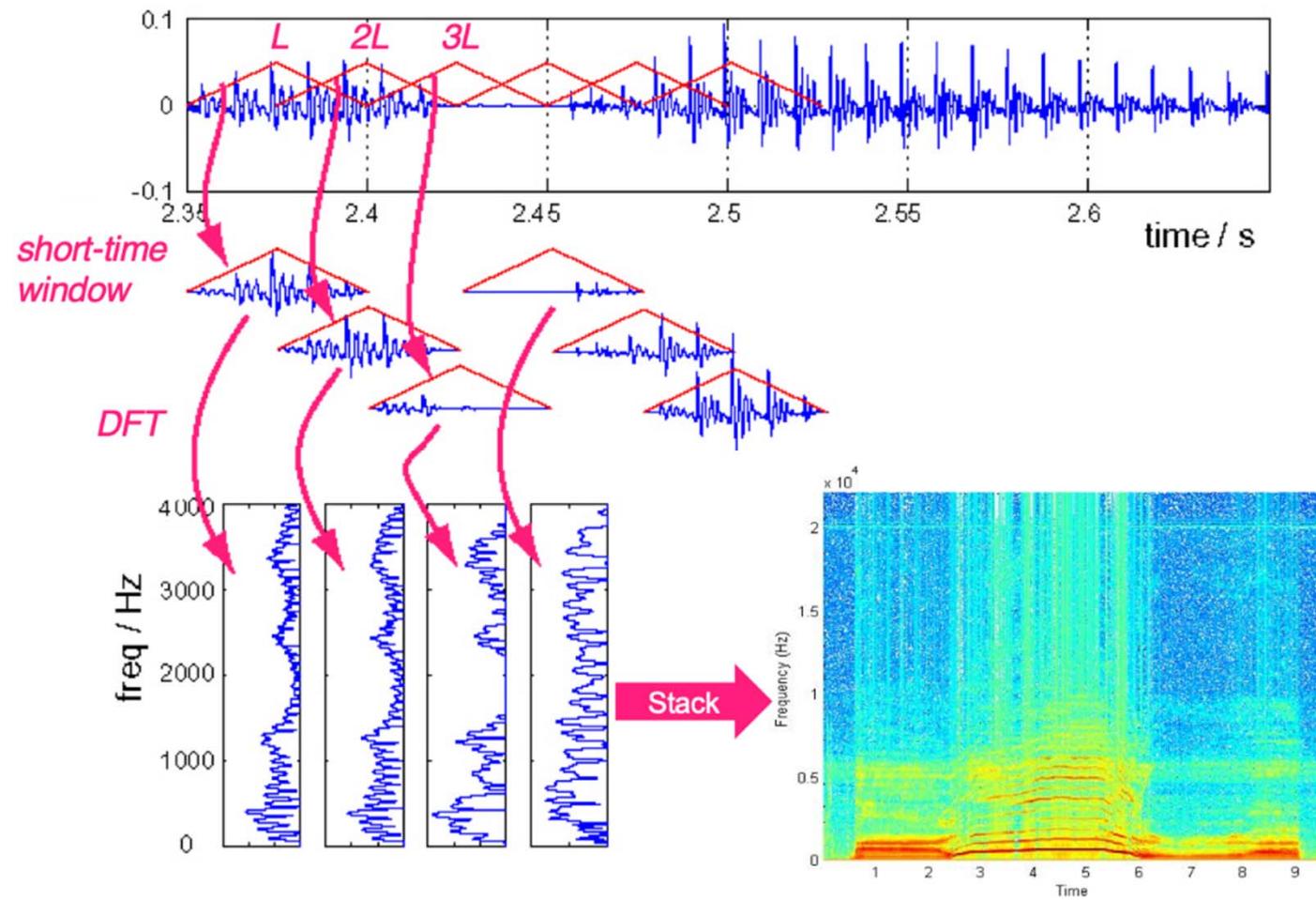
# Windowed FFT

[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2\\_STFT-Basic.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html)



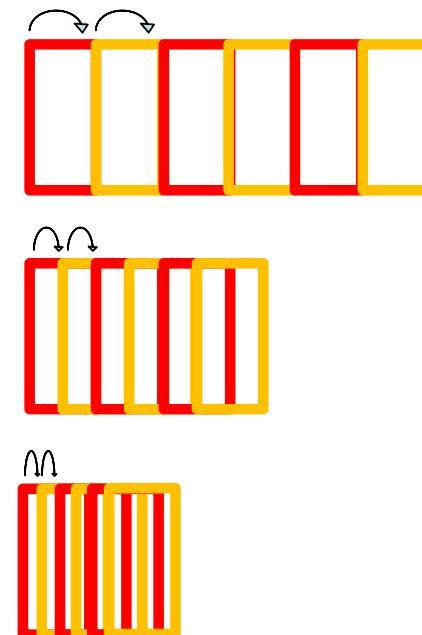
# Windowed FFT

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



# Hop Size in STFT

- Hop size
  - Can be proportional to the window size
    - $\text{hop\_size} = \text{win\_size}$
    - $\text{hop\_size} = 0.5 * \text{win\_size}$
    - $\text{hop\_size} = 0.1 * \text{win\_size}$
  - Can also be not proportional to the window size

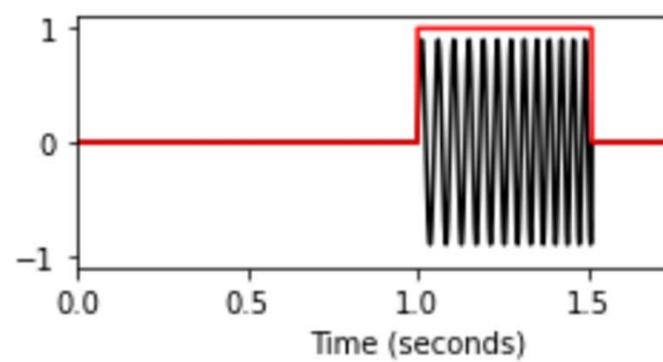


# Window Functions in STFT

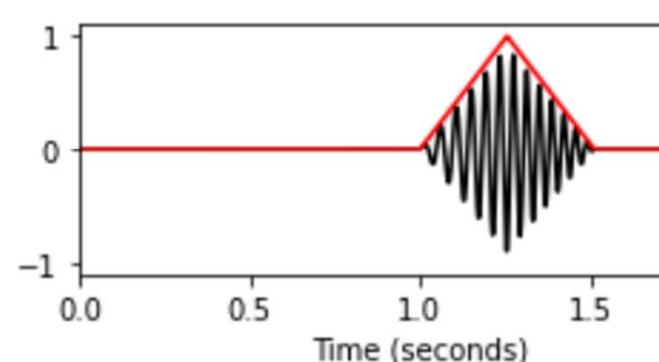
[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2\\_STFT-Window.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Window.html)

- Hann window is often used

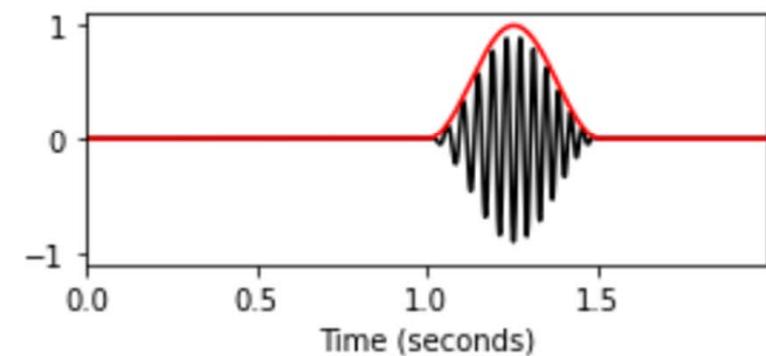
Rectangular window:



Triangular window:



Hann window:

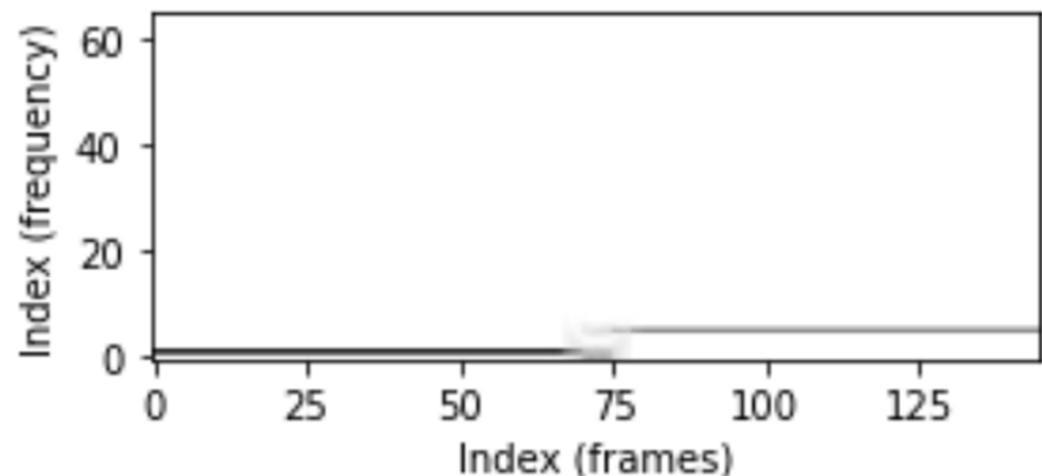


# Interpretation of Time and Frequency Indices in STFT

[https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2\\_STFT-Basic.html](https://www.audiolabs-erlangen.de/resources/MIR/FMP/C2/C2_STFT-Basic.html)

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

- **Frequency resolution:**  $\frac{F_s}{N}$
- **Temporal resolution:**  $\frac{H}{F_s}$



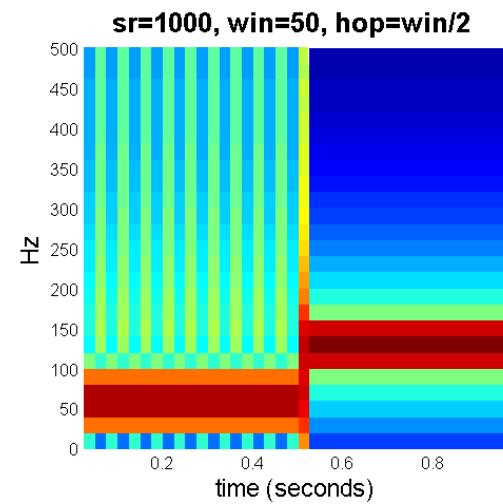
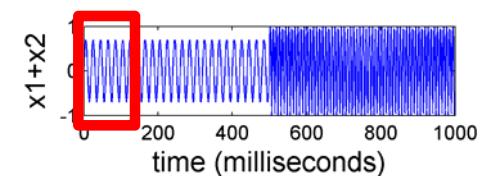
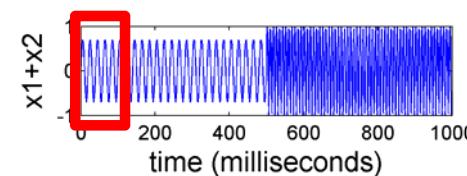
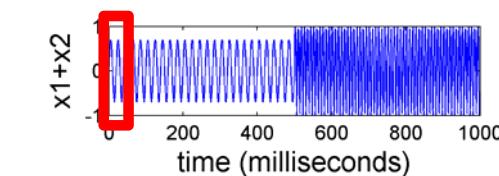
$$T_{\text{coef}}(m) := \frac{m \cdot H}{F_s}$$

# Trade-off between Frequency/Temporal Resolution in STFT

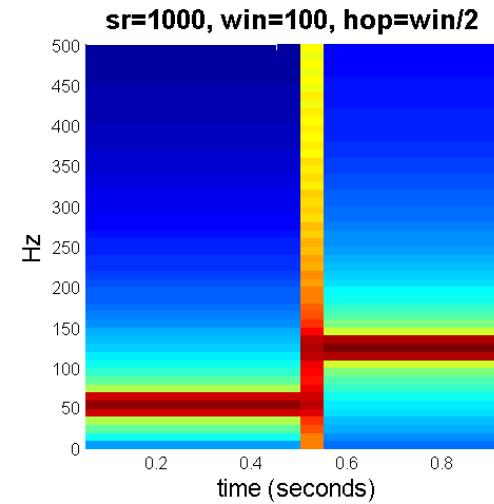
- **Frequency resolution:**  $\frac{F_s}{N}$ 
  - Longer window size ( $N$ ) → finer frequency resolution (but larger resulting STFT)  
→ can localize events along the frequency axis
- **Temporal resolution:**  $\frac{H}{F_s}$ 
  - Smaller hop size ( $H$ ) → finer temporal resolution (but larger resulting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

# Trade-off between Frequency/Temporal Resolution in STFT

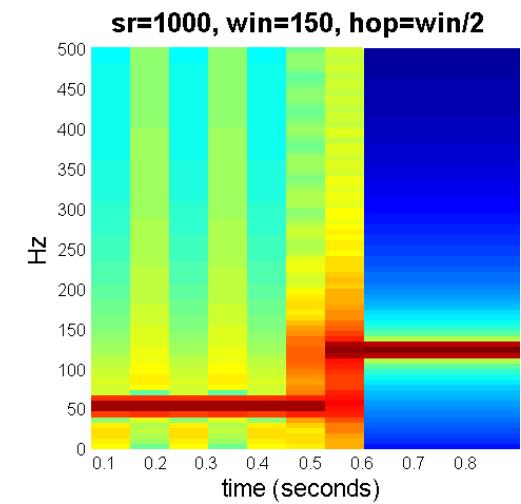
- Different window size (`win_size = 50, 100, 150`)
- `hop_size = win_size/2`



size:  $26 \times 39$



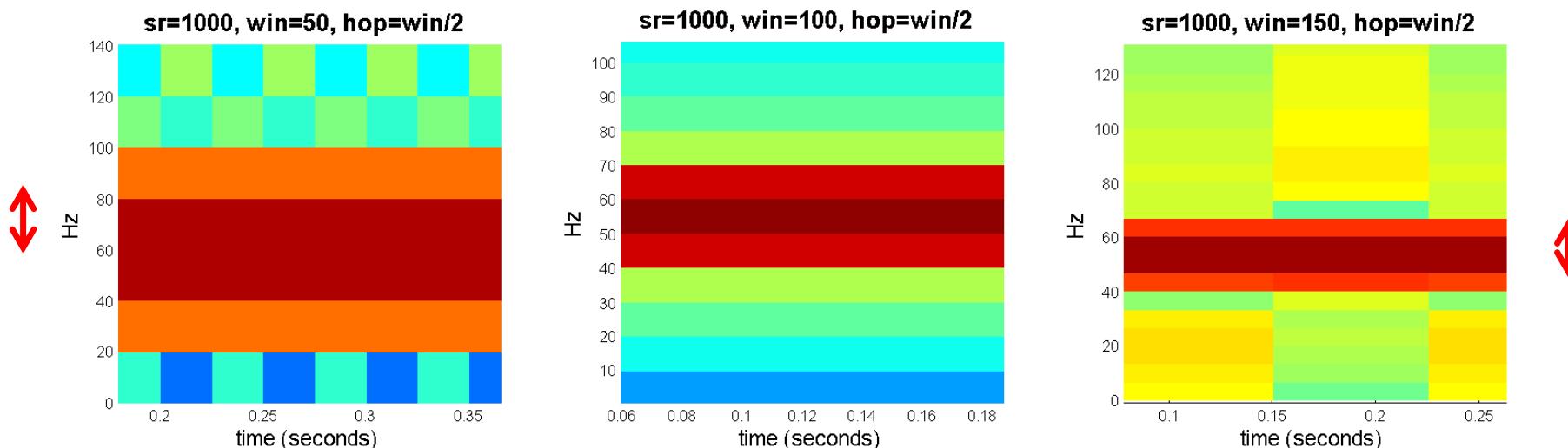
size:  $51 \times 19$



size:  $76 \times 12$

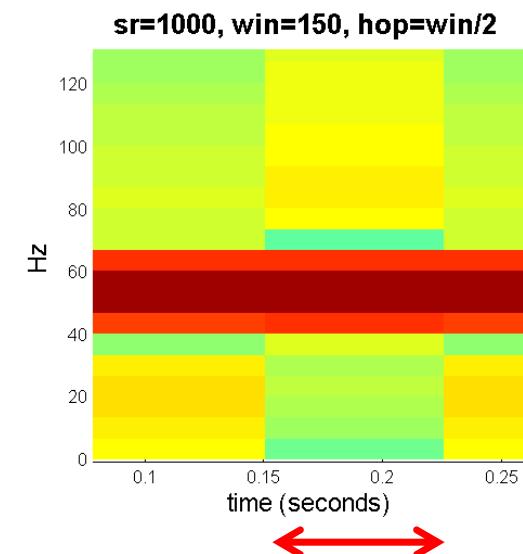
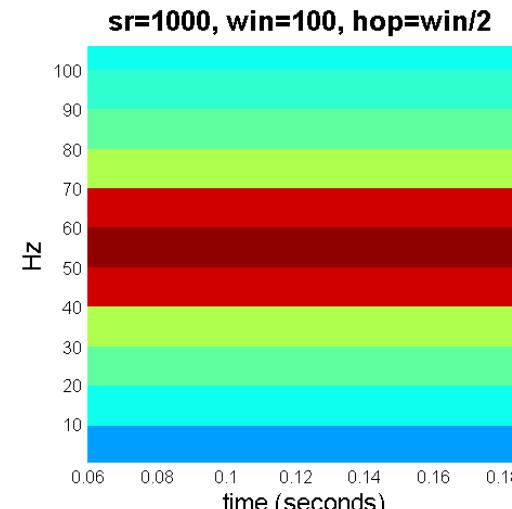
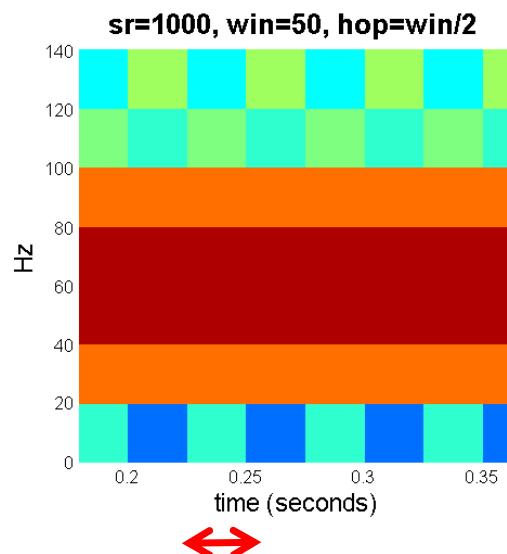
# Frequency Resolution in STFT

- **freq\_resolution = Fs/win\_size**
  - longer window → better frequency resolution
  - freq\_resolution = 20, 10, 6.6667 (Hz), respectively



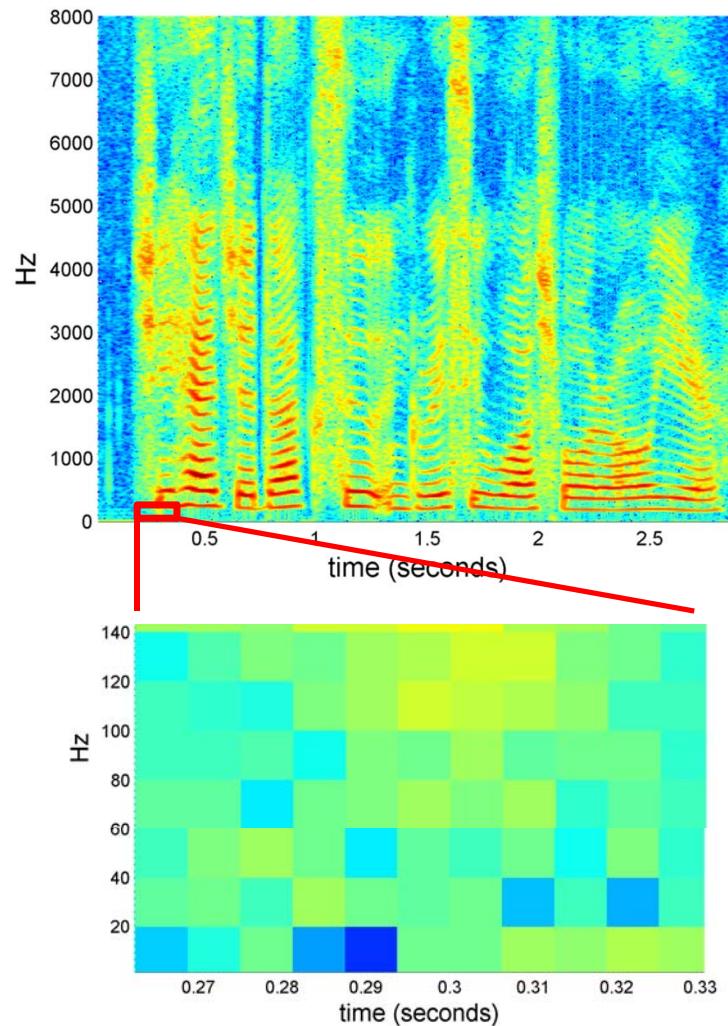
# Understanding STFT

- **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 ( $f_0$ ) 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?

Note name	Keyboard	Frequency Hz
A0		27.500
B0		30.868
C1		32.703
D1		36.708
E1		41.203
F1		43.654
G1		48.999
A1		55.000
B1		61.735
C2		65.406
D2		73.416
E2		82.407
F2		87.307
G2		97.999
A2		110.00
B2		123.47
C3		130.81
D3		146.83
E3		164.81
F3		174.61
G3		196.00
A3		220.00
B3		246.94
C4		<b>261.63</b>
D4		293.67
		311.13

# 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 ( $f_0$ ) 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?

Note name	Keyboard	Frequency Hz
[0, 43.1)		27.500
B0		30.868
C1		32.703
D1		36.708
E1		38.891
[43.1, 86.2)		43.654
G1		48.999
A1		55.000
B1		61.735
C2		65.406
D2		73.416
E2		82.407
[86.2, 129.3)		87.307
G2		97.999
A2		110.00
B2		123.47
[129.3, 172.4)		130.81
C3		146.83
E3		164.81
[172.4, 215.5)		174.61
G3		196.00
A3		220.00
B3		246.94
C4		261.63
D4		293.67
		211.12

# Quiz

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

[0, 10.8)  
[10.8, 21.5)  
[21.5, 32.3)  
[32.3, 43.1)  
[43.1, 53.8)  
...

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

Note name	Keyboard	Frequency Hz
[21.5, 32.3)		27.500 30.868
[32.3, 43.1)		32.703 36.708
D1		34.648
E1		41.203
[43.1, 53.8)		43.654 48.999
G1		46.249 51.913
A1		55.000
B1		61.735
C2		65.406
D2		73.416
E2		82.407
F2		87.307
G2		97.999
A2		110.00
B2		123.47
C3		130.81
D3		146.83
E3		164.81
F3		174.61
G3		196.00
A3		220.00
B3		246.94
C4	.....	261.63
D4		293.67
		311.13