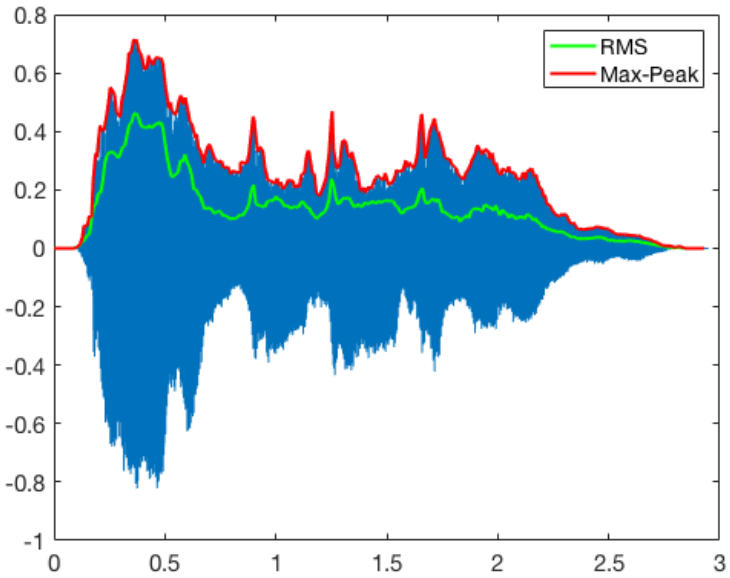
CTP431- Music and Audio Computing
Spectral Analysis

Graduate School of Culture Technology
KAIST
Juhan Nam

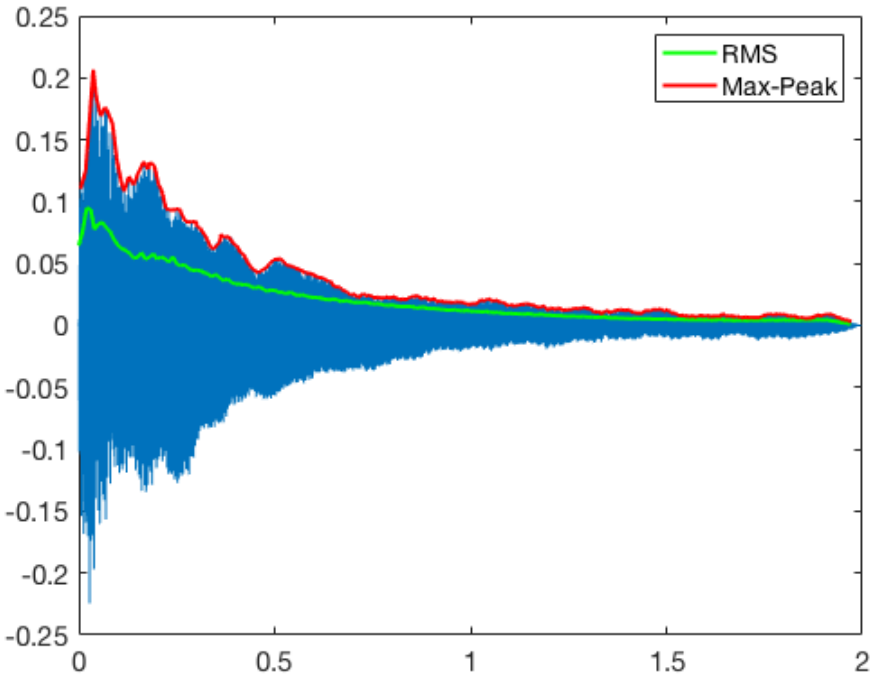
Waveform

- Time-domain representation of sound
  - Show the amplitude over time

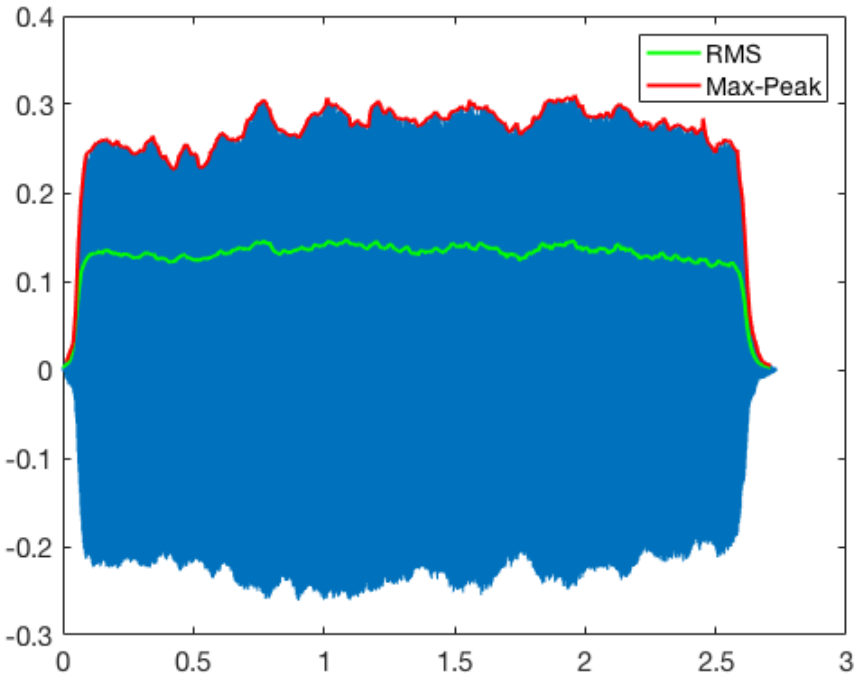


- Amplitude envelope
  - Short-term loudness: e.g. sound level meter
  - Computed by various methods
    - max-peak picking
    - root-mean-square (RMS)
    - Hilbert transform
  - ADSR
    - The amplitude envelope of musical sounds are often described with attack, decay, sustain and release.
  - Also used for dynamic range compression: e.g. compressor, expander

Example: Waveform and Amplitude Envelopes



Piano C4 Note



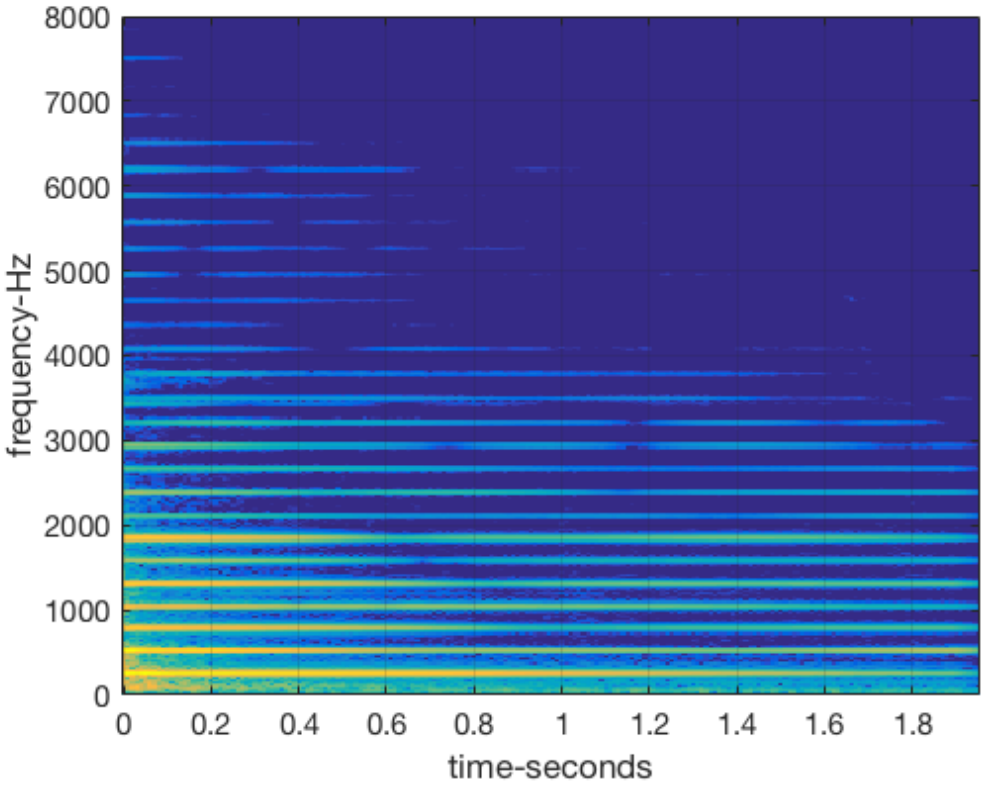
Flute A4 Note

Spectrogram

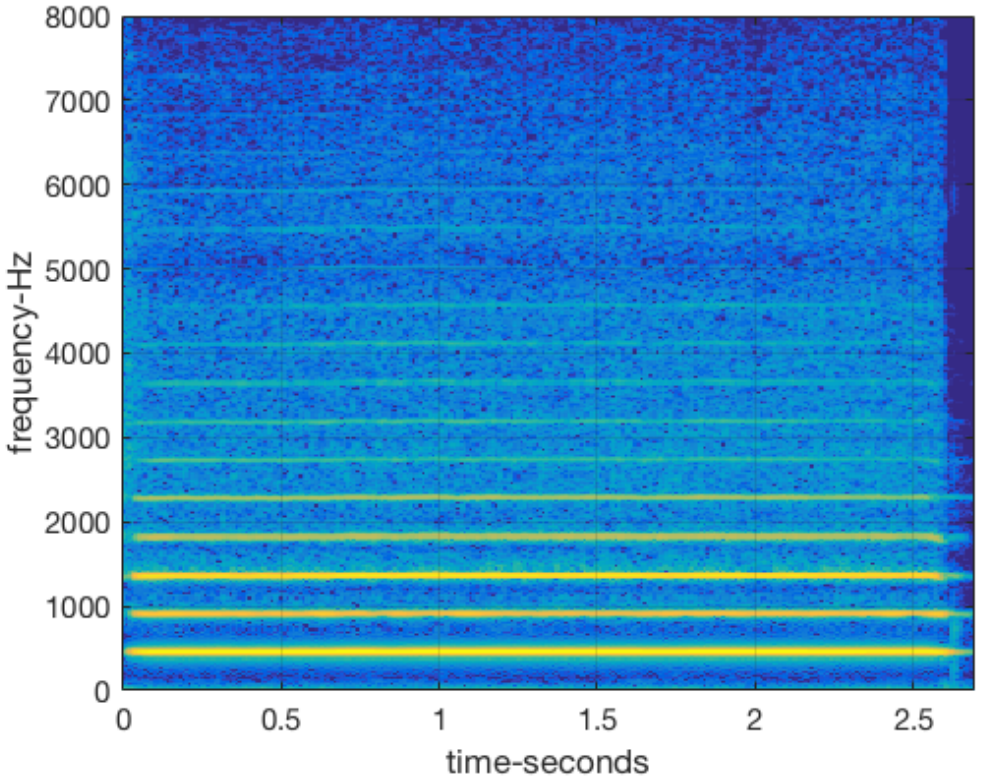
- Time/Frequency-domain representation of sound
  - Show the amplitude envelope of individual frequency components over time
  - Better representation to observe pitch and timbre characteristics
  - Often called “Sonogram”

- Visualization
  - 2D color map or waterfall

Example: Spectrogram - 2D color map

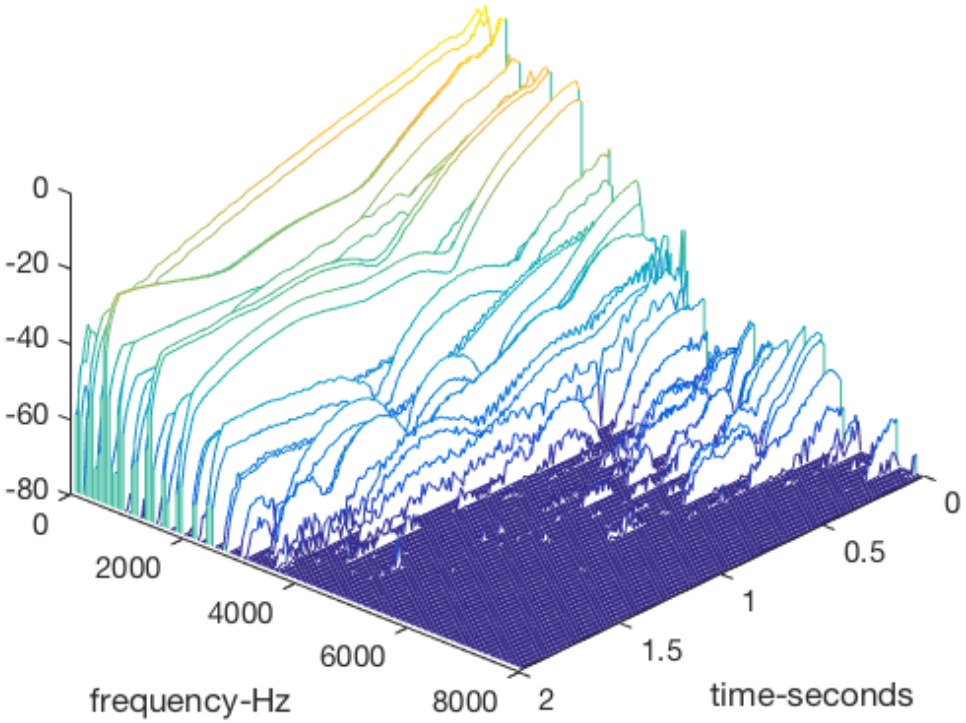


Piano C4 Note

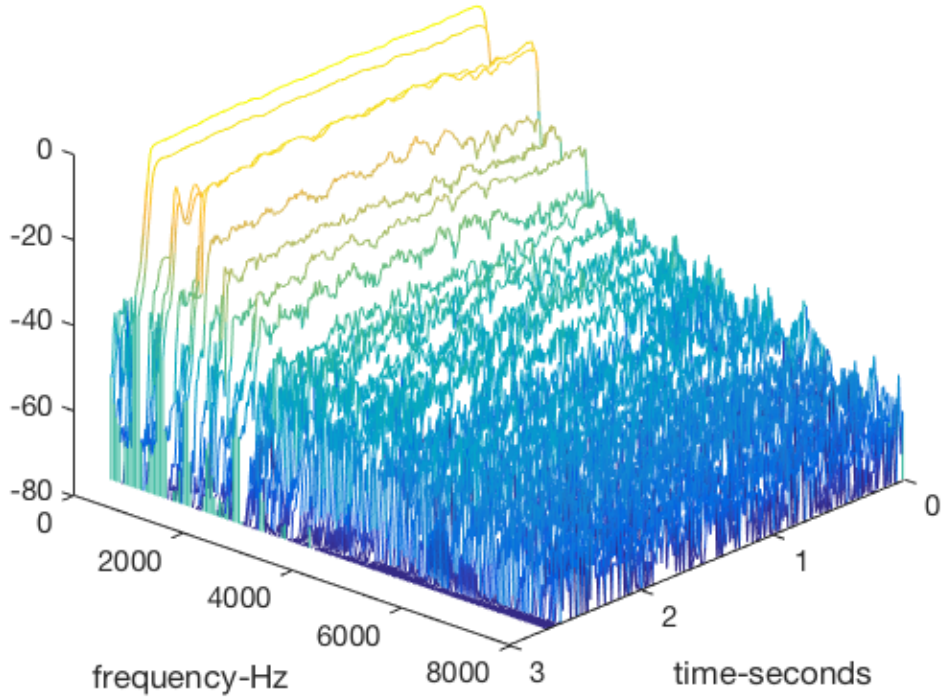


Flute A4 Note

Example: Spectrogram - 3D waterfall



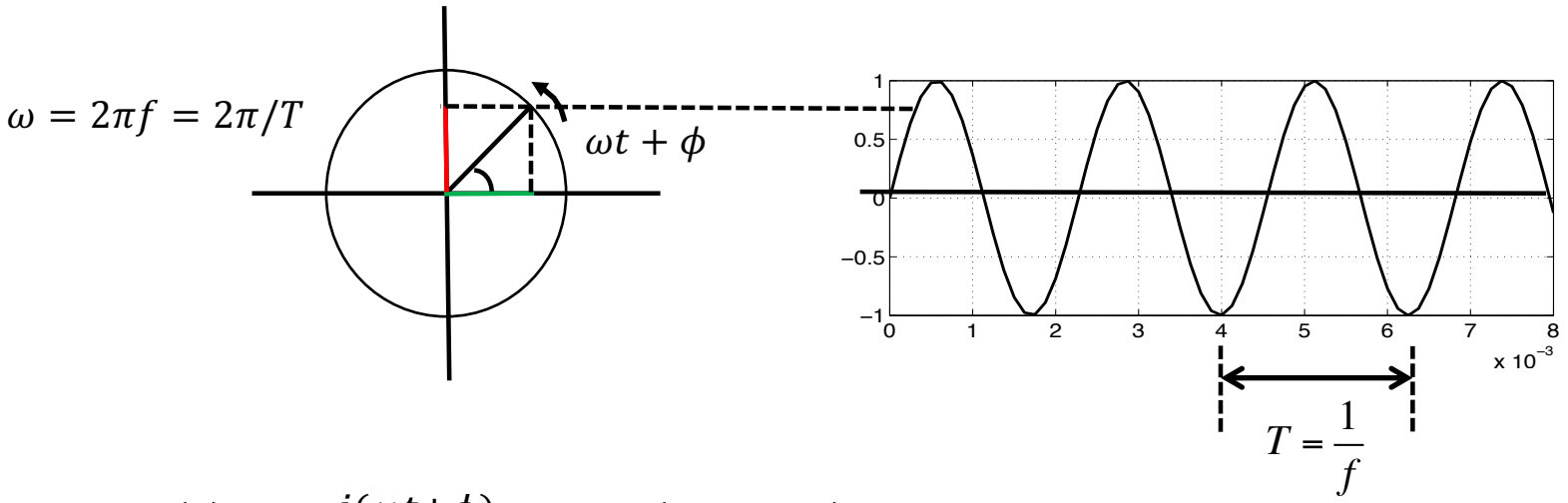
Piano C4 Note



Flute A4 Note

Phasor

- A complex number representing a sinusoidal function with
  - Amplitude, angular frequency, initial phase



$$x(t) = e^{j(\omega t+\phi)} = \cos(\omega t + \phi) + \mathrm{j}\sin(\omega t + \phi)$$

Euler's Identity

Fourier Series

- Any signal x(t) with period T can be represented as a sum of phasors
  - The periods of phasors are T, T/2, T/3, …, T/n, …

$$x(t) = \text{real}\left\{\frac{1}{T}\sum_{k=0}^{\infty} r_k e^{j\left(\frac{2\pi kt}{T}+\phi_k\right)}\right\} = \text{real}\left\{\frac{1}{T}\sum_{k=0}^{\infty} c_k^* e^{j\left(\frac{2\pi kt}{T}\right)}\right\} \qquad c_k^* = r_k e^{j\phi_k}$$

- Web Audio Examples
  - http://codepen.io/anon/pen/jPGJMK

- How can you get the coefficients?

$$a_k = r_k \cos(\phi_k)$$

$$b_k = r_k \sin(\phi_k)$$

$$r_k = \sqrt{{a_k}^2 + {b_k}^2}$$

$$\Phi_k = \arctan\left(\frac{b_k}{a_k}\right)$$

Orthogonality of Sinusoids

- The phasors are orthogonal to each other unless they have the same frequency

$$\int_{-T/2}^{T/2} e^{j\left(\frac{2\pi mt}{T}\right)} e^{-j\left(\frac{2\pi nt}{T}\right)} dt = \begin{cases} 0 & (m \neq n) \\ T & (m = n) \end{cases}$$

- Using the orthogonality

$$c_k^* = \int_{-T/2}^{T/2} x(t) e^{-j\left(\frac{2\pi kt}{T}\right)} dt$$

Discrete Fourier Transform (DFT)

- Discrete-time version of Fourier series

$$x(n) = [x_0, x_1, x_2, \cdots, x_{N-1}]$$

- The number of discrete samples, N, corresponds on the period T
  - We assume that the segment x(n) is repeated every N samples

- Then, we can directly derive DFT and Inverse DFT from

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\pi kn}{N}\right)}$$

DFT

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\left(\frac{2\pi kn}{N}\right)}$$

IDFT

Discrete Fourier Transform

- Discrete Fourier Transform

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\left(\frac{2\pi kn}{N}\right)} = X_R(k) + jX_I(k)$$

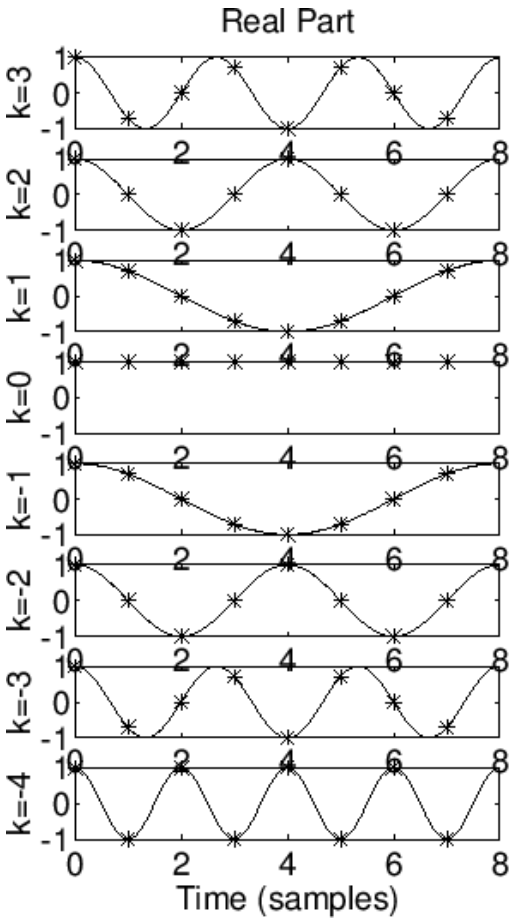
  - Magnitude spectrum: $A(k) = \sqrt{X_R(k)^2 + X_I(k)^2}$
  - Phase spectrum: $\Phi(k) = \arctan\left(\frac{X_I(k)}{X_R(k)}\right)$

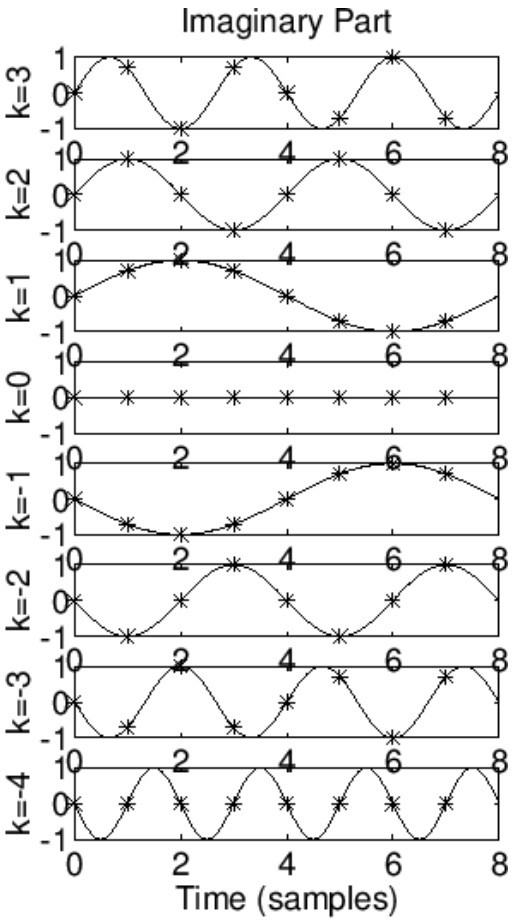
- We use the magnitude spectrum to display spectrograms

DFT Sinusoids

$$s_k^*(n) = e^{j\left(\frac{2\pi kn}{N}\right)}$$

$$N = 8$$





$s_3^*(n)$

$s_2^*(n)$

$s_1^*(n)$

$s_0^*(n)$

$s_{-1}^*(n) = s_7^*(n)$

$s_{-2}^*(n) = s_6^*(n)$

$s_{-3}^*(n) = s_5^*(n)$

$s_{-4}^*(n) = s_4^*(n)$

Source: the JOS DFT book

Fast Fourier Transform

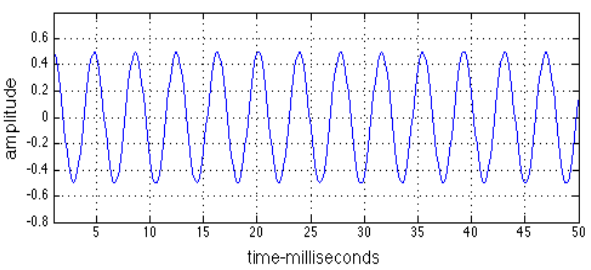
- Matrix multiplication view of DFT

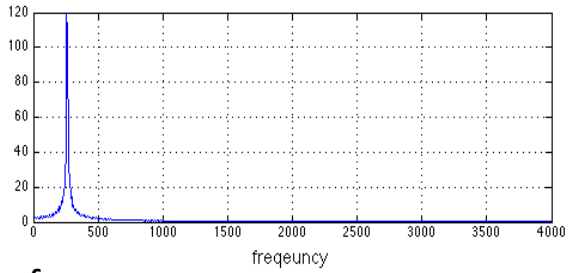
$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} s_0^*(0) & s_0^*(1) & \cdots & s_0^*(N-1) \\ s_1^*(0) & s_1^*(1) & \cdots & s_1^*(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ s_{N-1}^*(0) & s_{N-1}^*(1) & \cdots & s_{N-1}^*(N-1) \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

Source: the JOS DFT book

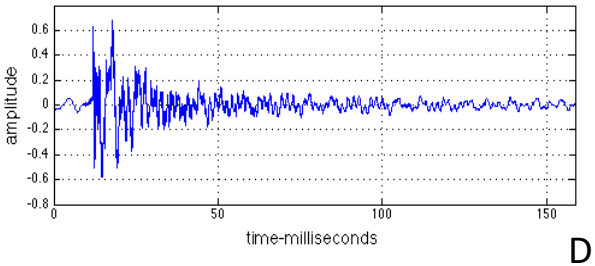
- In fact, we don't compute this directly. There is a more efficiently way, which is called "Fast Fourier Transform (FFT)"
  - Complexity reduction by FFT: $O(N^2)$ → $O(N\log_2 N)$
  - Divide and conquer

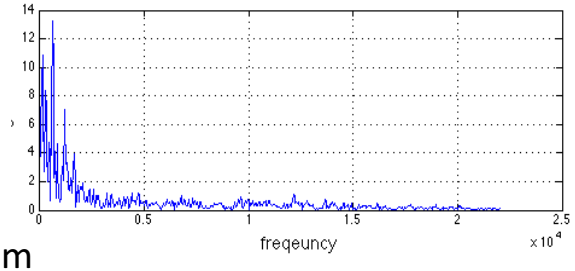
Examples of DFT



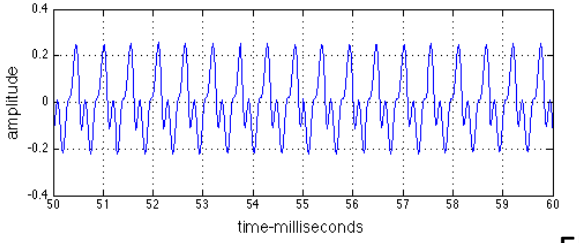


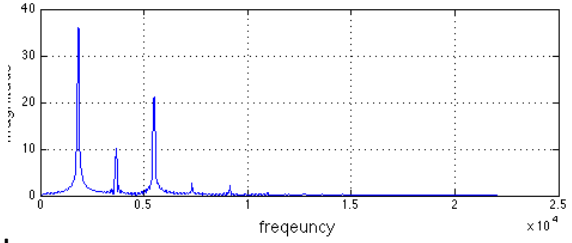
Sine waveform





Drum





Flute

Short-Time Fourier Transform (STFT)

- DFT assumes that the signal is stationary
  - It is not a good idea to apply DFT to a long and dynamically changing signal like music
  - Instead, we segment the signal and apply DFT separately

- Short-Time Fourier Transform

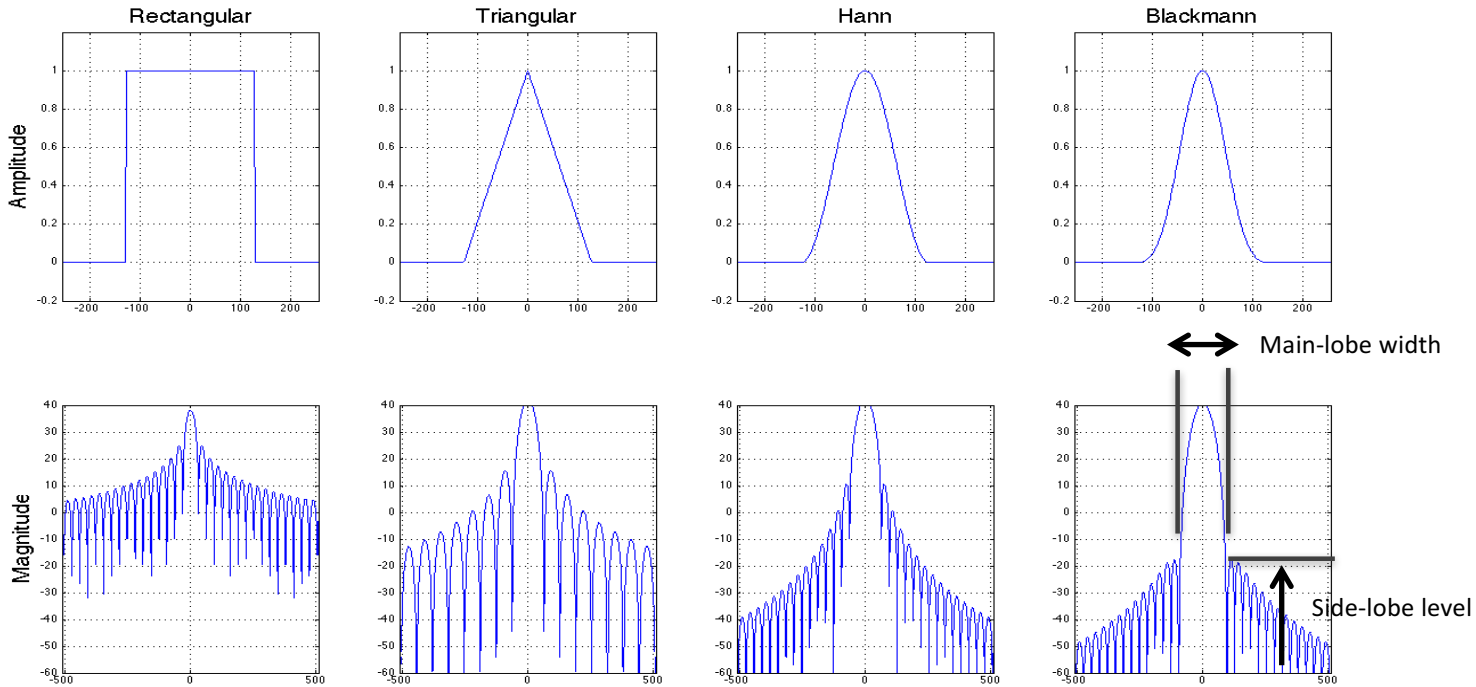
$$X(k,l) = \sum_{n=0}^{N-1} w(n)x(n + l \cdot h)e^{-j\left(\frac{2\pi kn}{N}\right)}$$

h : hop size
$w(n)$: window
N : FFT size

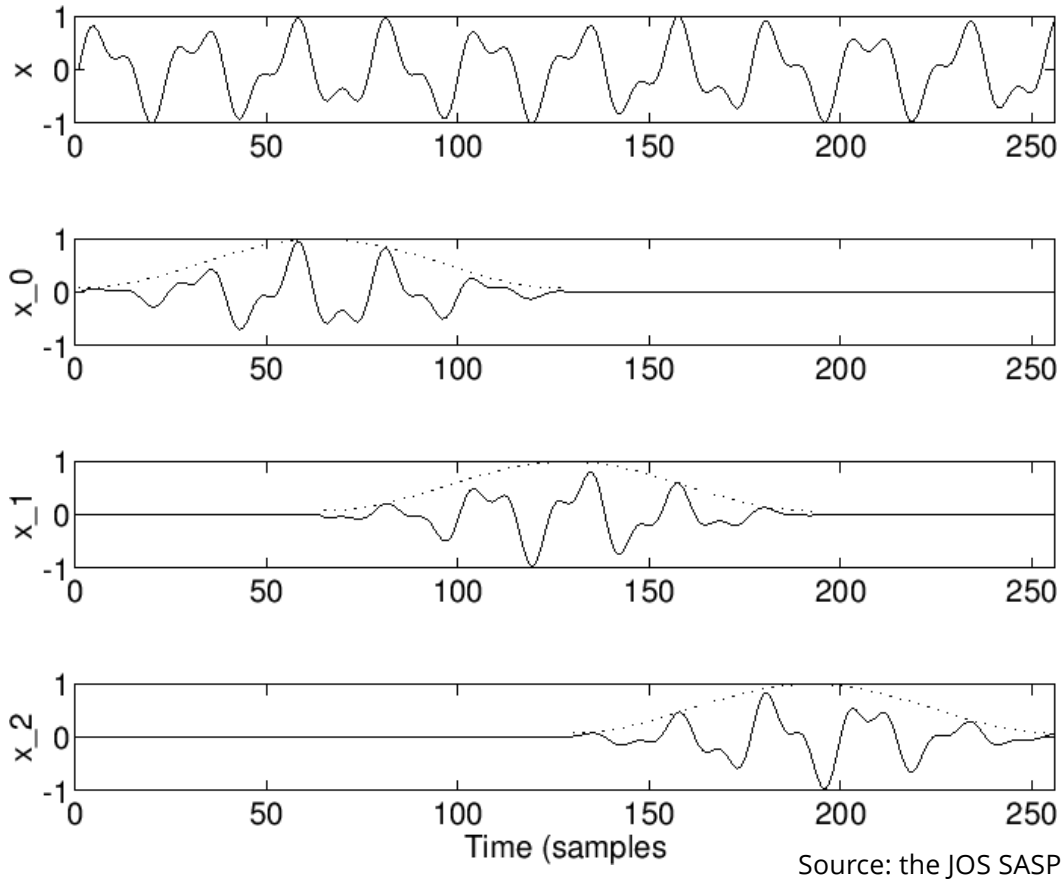
- This produces 2-D time-frequency representations
  - Get “spectrogram” from the magnitude
  - Parameters: window size, window type, FFT size, hop size

Windowing

- Types of window functions
  - Trade-off between the width of main-lobe and the level of side-lobe



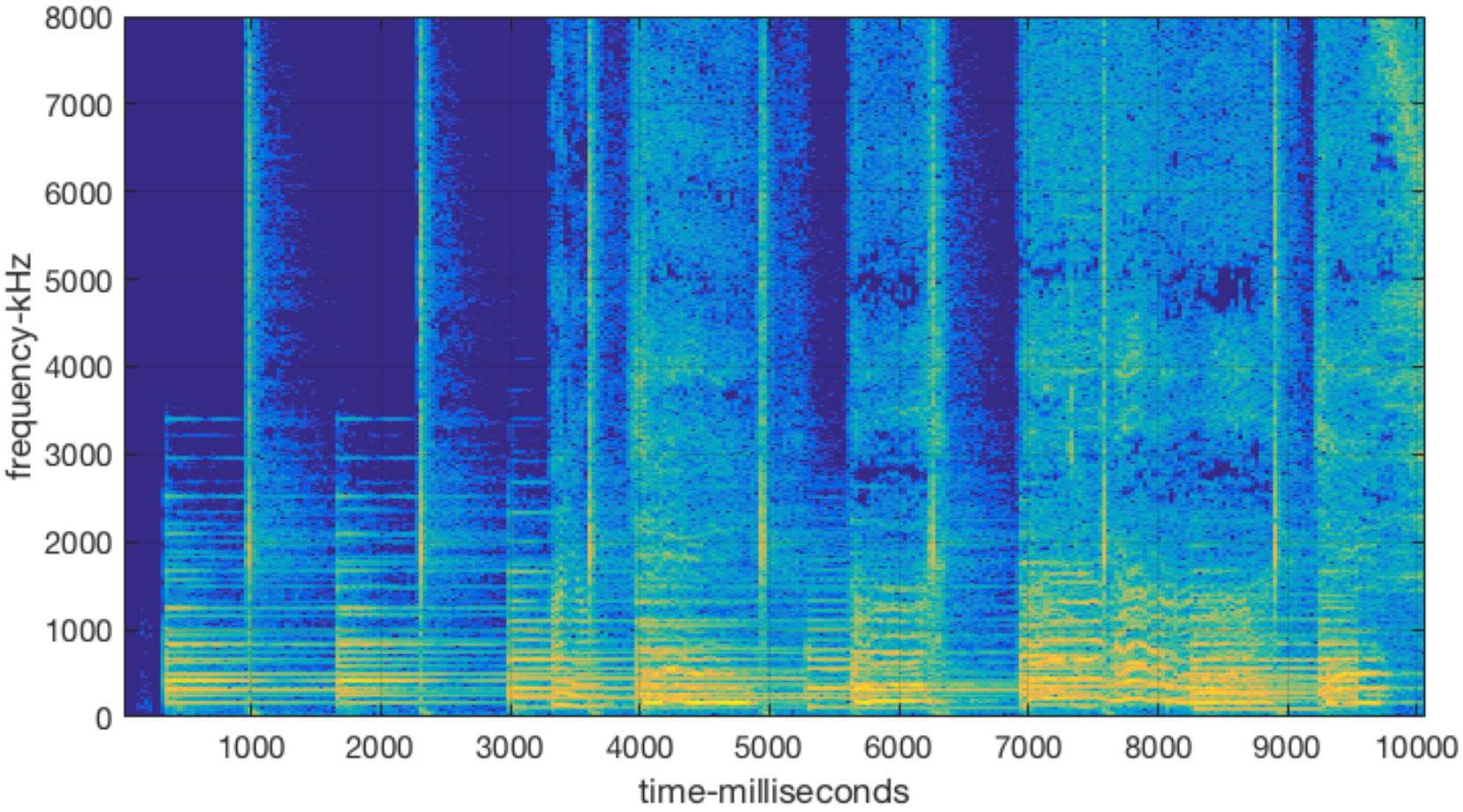
Short-Time Fourier Transform (STFT)



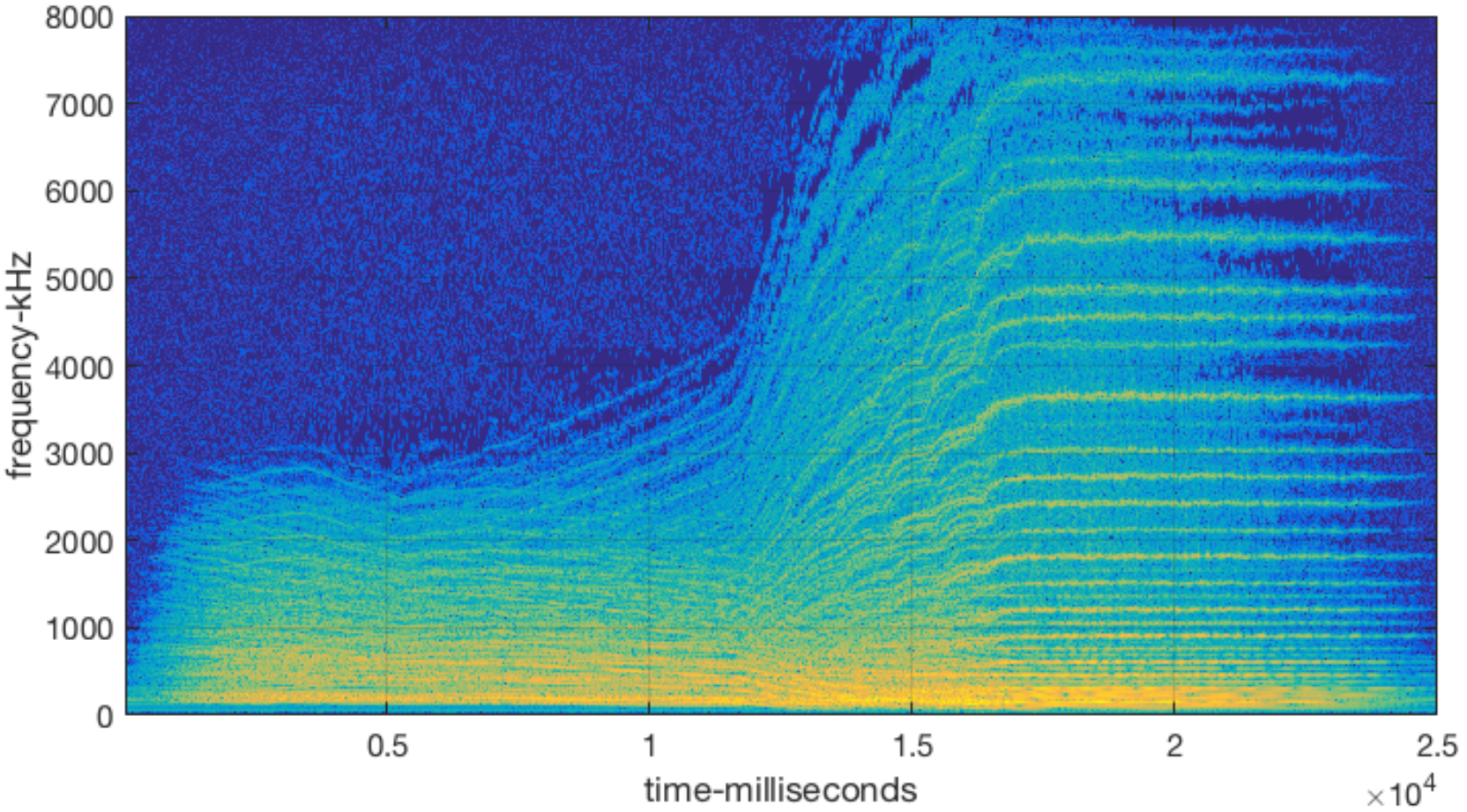
Source: the JOS SASP book

50% overlap

Example: Pop Music



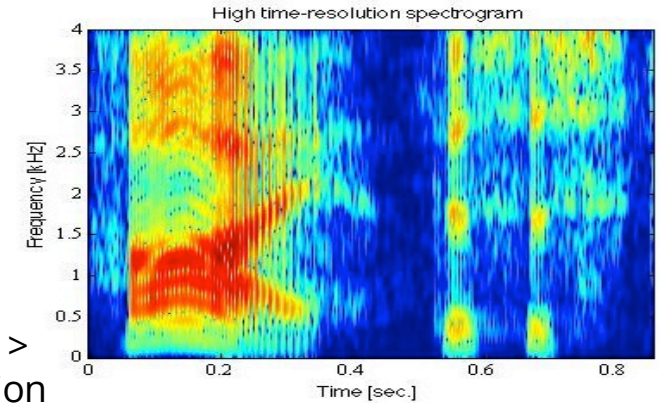
Example: Deep Note

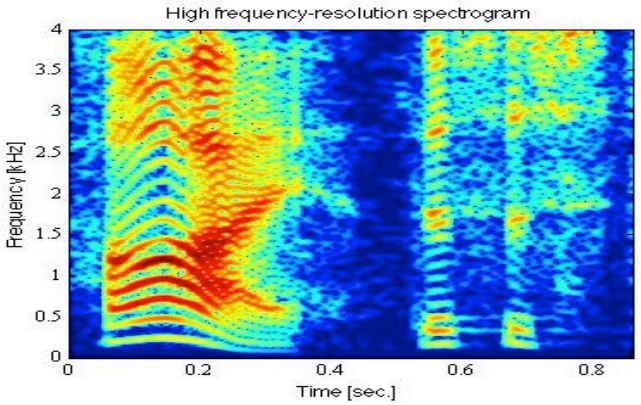


Time-Frequency Resolutions in STFT

- Trade-off between time- and frequency-resolution by window size

< Short window >
low freq.-resolution
high time-resolution





< Long window >
high freq.-resolution
low time-resolution