CTP431: Fundamentals of Computer Music

Spectrum Analysis

KAIST Graduate School of Culture Technology

Juhan Nam

Goals

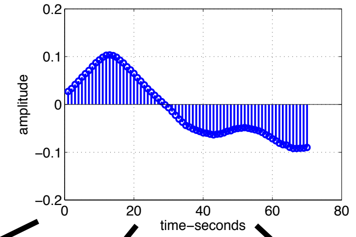
- Representing discrete-time signals with sampled sinusoids
  - Discrete Fourier Transform
  - Short-Time Fourier Transform

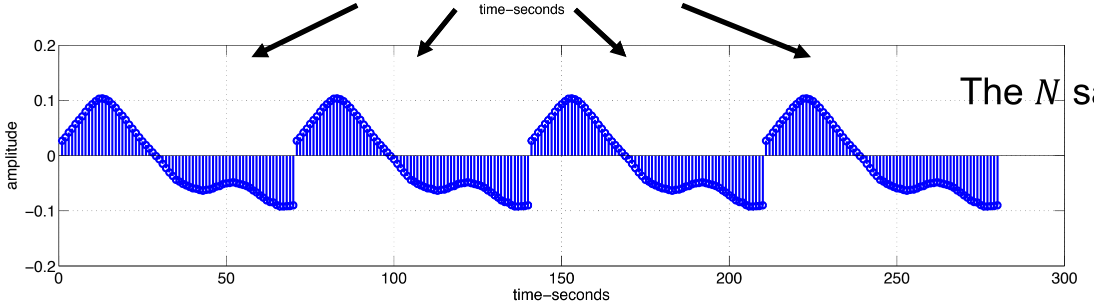
Fourier Analysis on Computers

- The signals are sampled in time
- The signals are not necessarily periodic
  - Even if they are periodic, the period is usually not given beforehand
- Computers take a “segment” of the signals
  - We usually take a 10 msec to 1 second long segment in practice
  - It often corresponds to the “audio buffer” size

From Fourier Series to Discrete Fourier Transform

- We first take a segment of “sampled” waveform
  - The length of the segment is N
- Then, we assume that the N samples are repeated with a period of N





“sampled” and “periodic” waveform

From Fourier Series to Discrete Fourier Transform

- Periodic --> Fourier Series

$$C_k = \frac{f_s}{N}\int_0^{N/f_s} x(t)e^{-j\omega_k t}\,dt$$

$$\omega_k = \frac{2\pi f_s}{N}k \quad (k = 0,1,2,\ldots)$$

$P = N$ samples = N/f_s (second)

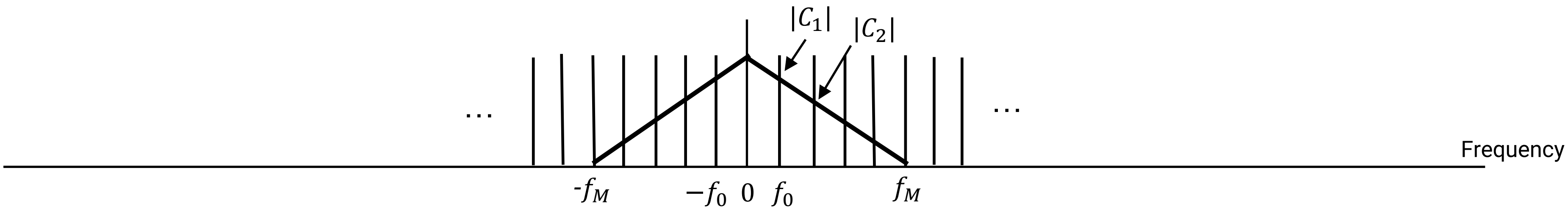
$$x(t) = \sum_{k=-M}^{M} C_k\, e^{j\omega_k t}$$

$$|C_k| = \sqrt{\mathrm{Re}\{C_k\}^2 + \mathrm{Im}\{C_k\}^2}$$

$$\phi_k = \angle C_k = \tan^{-1}(\mathrm{Im}\{C_k\}/\mathrm{Re}\{C_k\})$$

Magnitude spectrum of $x(t)$

Periodic in time → Discrete in Frequency



From Fourier Series to Discrete Fourier Transform

- Sampling: $x(t) \rightarrow x(n/f_s)$

$$C_k = \frac{f_s}{N}\int_0^{N/f_s} x(t)e^{-j\omega_k t}\,dt \longrightarrow X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}}$$

$$\omega_k = \frac{2\pi f_s}{N}k$$

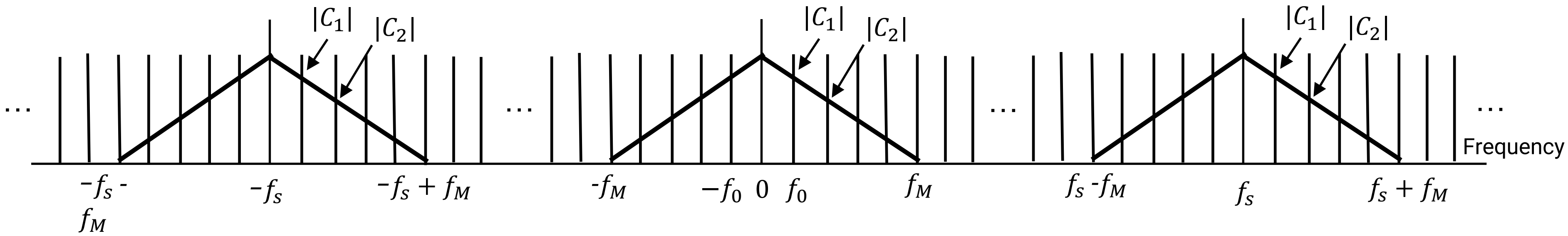
$$(k = 0,1,2,\dots,N-1)$$

$$x(t) = \sum_{k=-M}^{M} C_k\, e^{j\omega_k t} \longrightarrow x(n) = \frac{1}{N}\sum_{k=-N/2}^{N/2-1} X(k)e^{j\frac{2\pi kn}{N}} = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j\frac{2\pi kn}{N}}$$

Magnitude spectrum of $x(n)$

$(f_s > 2\cdot f_M)$

Periodic in time → Discrete in Frequency   Discrete in time → Periodic in Frequency



What's the difference?

- Fourier Series: period P is from the period of signal
  - We assume that we know the period of the signal
- Discrete Fourier Transform: Period N is not from the signal
  - Period N is determined by the system (i.e., ”we” select the period)
  - It is called **DFT size** or **FFT size**: typically, 512, 1024, 2048 samples
- **The repetition of N samples are implicitly assumed in DFT although they are not physically repeated**

Duality Between Time and Frequency Domains

- **Periodic (or Finite)** in time ↔ **Sampled (or Discrete)** in frequency
- **Sampled (or Discrete)** in time ↔ **Periodic (or Finite)** in frequency

Comparison of Fourier Transforms

Time Duration		
Finite	Infinite	
Discrete FT (DFT) $X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\omega_k n}$ $k = 0, 1, \ldots, N-1$	Discrete Time FT (DTFT) $X(\omega) = \sum_{n=-\infty}^{+\infty} x(n) e^{-j\omega n}$ $\omega \in (-\pi, +\pi)$	discr. time n
Fourier Series (FS) $X(k) = \int_0^P x(t) e^{-j\omega_k t} dt$ $k = -\infty, \ldots, +\infty$	Fourier Transform (FT) $X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$ $\omega \in (-\infty, +\infty)$	cont. time t
discrete freq. k	continuous freq. ω	

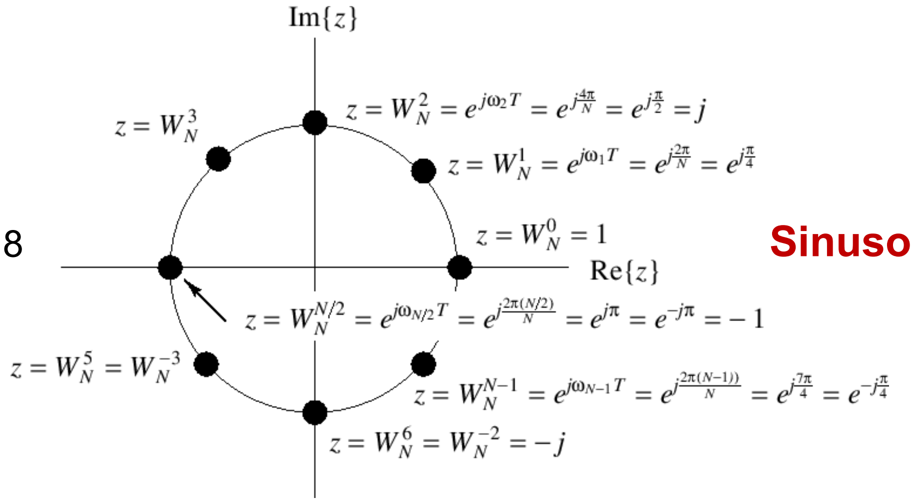
Complex Sinusoids in DFT

- The frequencies of complex sinusoid in DFT is determined by N

$$\boxed{s_k(n) = e^{j\frac{2\pi kn}{N}} = \cos\frac{2\pi kn}{N} + j\sin\frac{2\pi kn}{N}} \quad (n = 0,1,2,\dots,N-1)$$

  - Frequencies: $\frac{2\pi k}{N}$ radian or $\frac{k}{N} f_s$ Hz $(K = 0, 1, 2, \dots, N-1)$

Example: N = 8

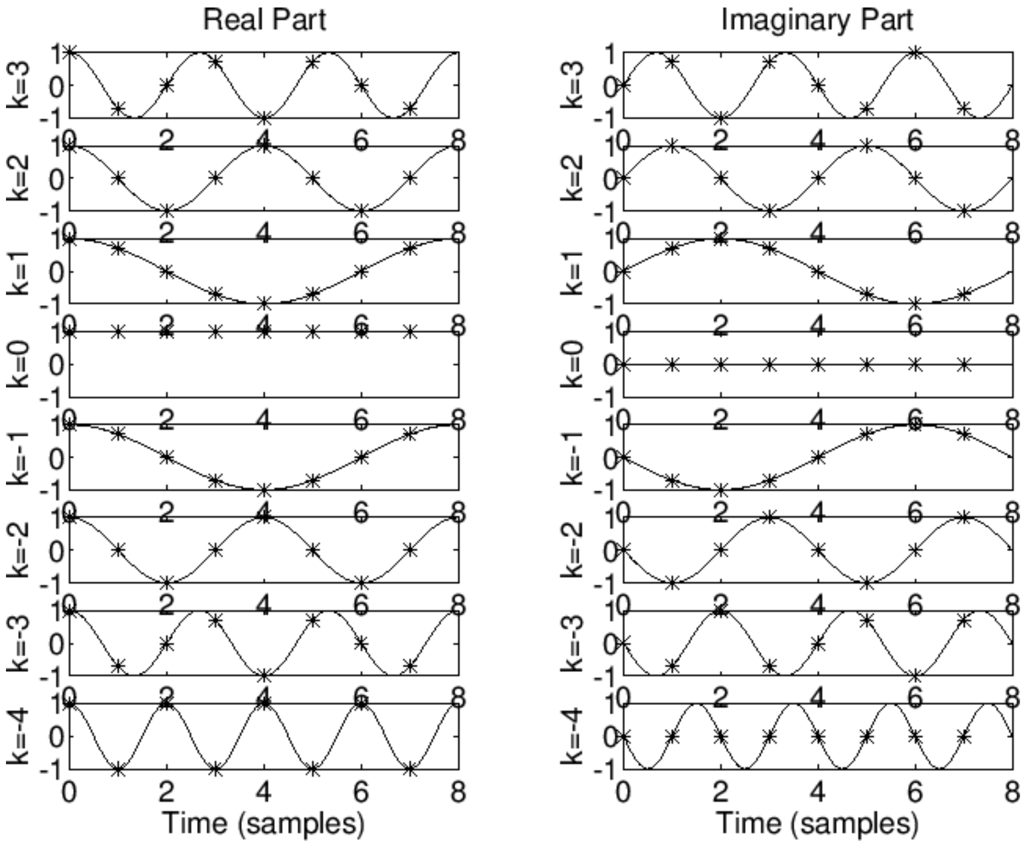


Sinusoids are *N*-dimensional vectors!

Source: https://ccrma.stanford.edu/~jos/dft/

Complex Sinusoids in DFT

Example: $N = 8$



$$\text{Re}\{s_k(k)\} = \text{Re}\{s_k(-k)\} = \text{Re}\{s_k(N-k)\}$$

Even-Symmetric

$$\text{Im}\{s_k(k)\} = -\text{Im}\{s_k(-k)\} = -\text{Im}\{s_k(N-k)\}$$

Odd-Symmetric

Source: https://ccrma.stanford.edu/~jos/dft/

Orthogonality of Sinusoids

- Inner product between two complex sinusoids

$$s_p(n) \cdot s_q^*(n) = \sum_{n=0}^{N-1} e^{j\frac{2\pi pn}{N}} \cdot e^{-j\frac{2\pi qn}{N}} = \begin{cases} N & \text{if } p = q \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{n=0}^{N-1} \sin(2\pi pn/N)\sin(2\pi qn/N)) = \begin{cases} 0 & \text{otherwise} \\ N/2 & \text{if } p = q \\ -N/2 & \text{if } p = N - q \end{cases}$$

$$\sum_{n=0}^{N-1} \cos(2\pi pn/N)\sin(2\pi qn/N)) = 0$$

$$\sum_{n=0}^{N-1} \cos(2\pi pn/N)\cos(2\pi qn/N)) = \begin{cases} N/2 & \text{if } p = q \text{ or } p = N - q \\ 0 & \text{otherwise} \end{cases}$$

Orthogonal Projection on Complex Sinusoids

- Do the inner product with the signal and sinusoids

$$x(n) \cdot s_k(n) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi kn}{N}} = \sum_{n=0}^{N-1} (\frac{1}{N} \sum_{p=0}^{N-1} X(k) e^{j\frac{2\pi pn}{N}}) e^{-j\frac{2\pi kn}{N}}$$

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

Discrete Fourier Transform

- Discrete Fourier Transform

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

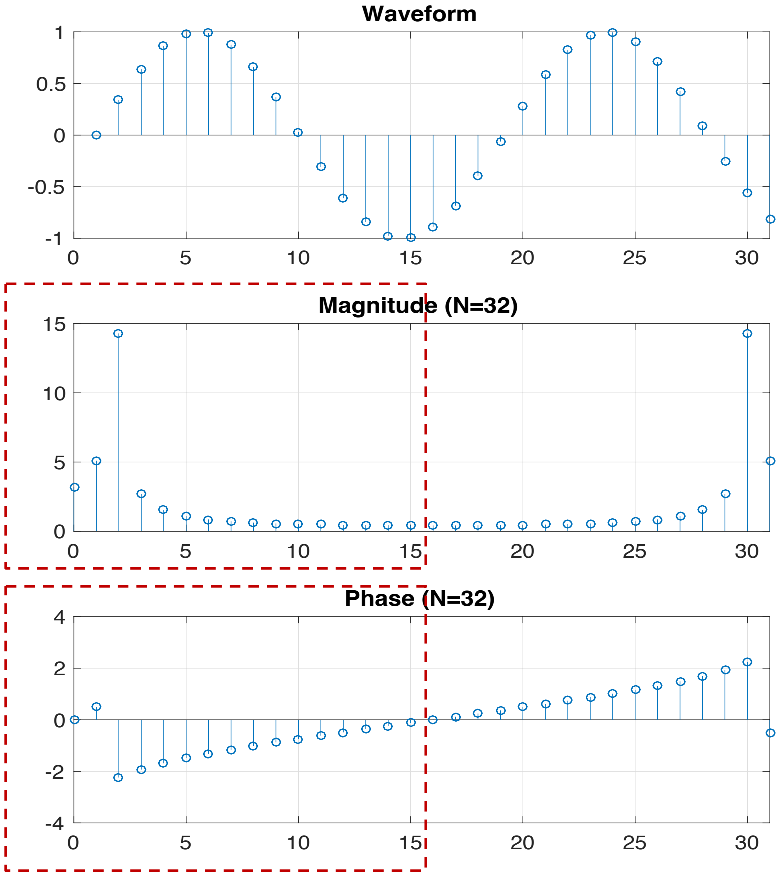
  - Magnitude spectrum: $|X(k)| = A(k) = \sqrt{X_R^2(k) + X_I^2(k)}$
  - Phase spectrum: $\angle X(k) = \phi(k) = \tan^{-1}(\frac{X_I(k)}{X_R(k)})$

- Inverse Discrete Fourier Transform

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

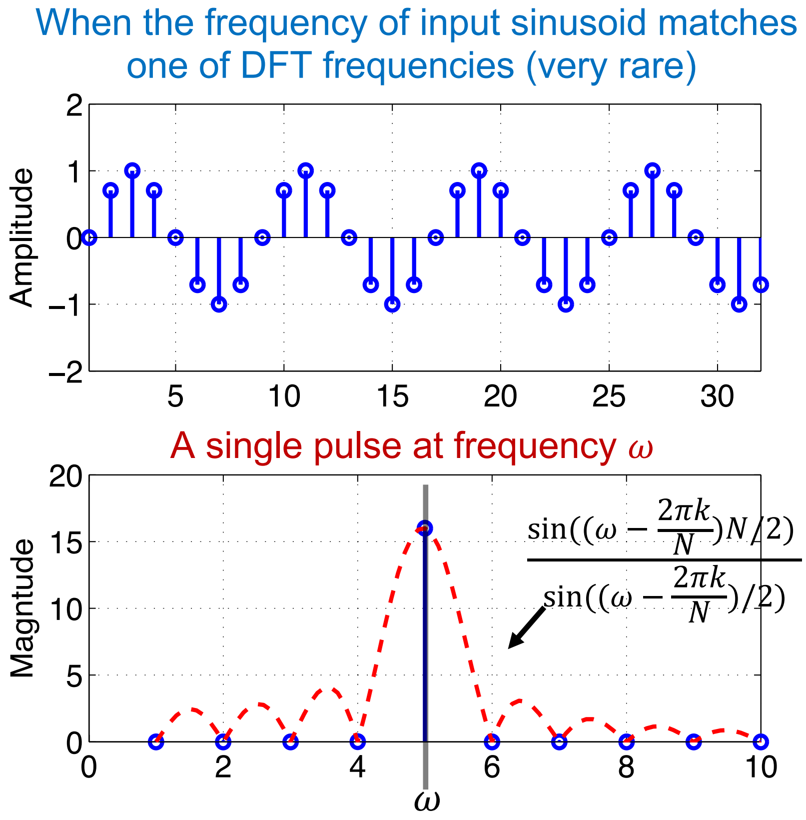
Properties of DFT

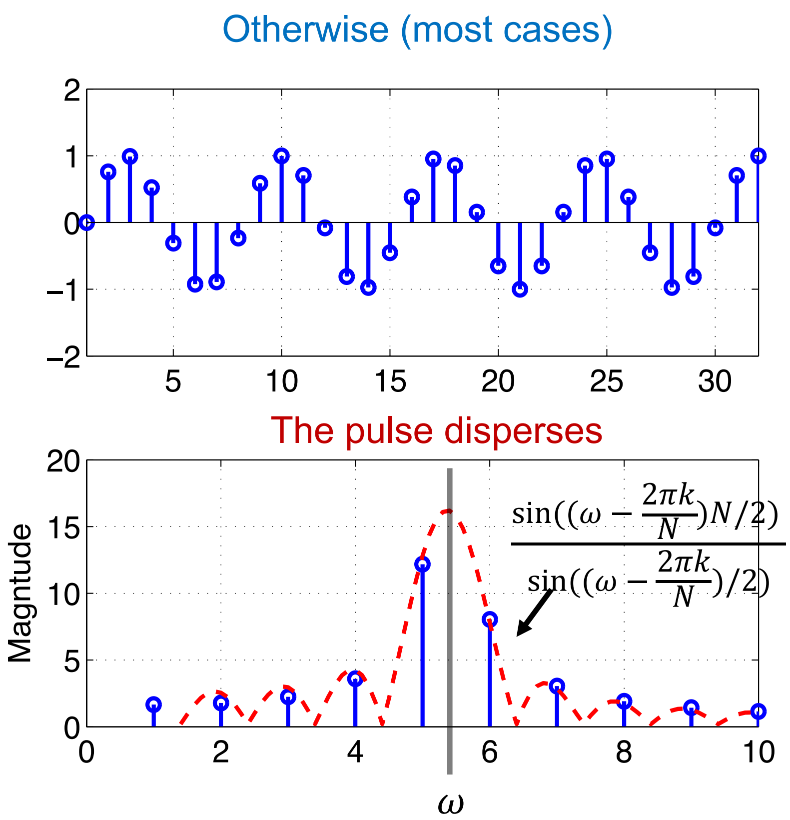
- Symmetry
  - $|X(k)| = |X(-k)| = |X(N-k)|$
  - $\angle X(k) = -\angle X(-k) = -\angle X(N-k)$
- Periodicity
  - $X(k) = X(k \pm N) = X(k \pm 2N) = \ ...$



Display the spectrum up to $N/2$

Cracks in Sinusoids

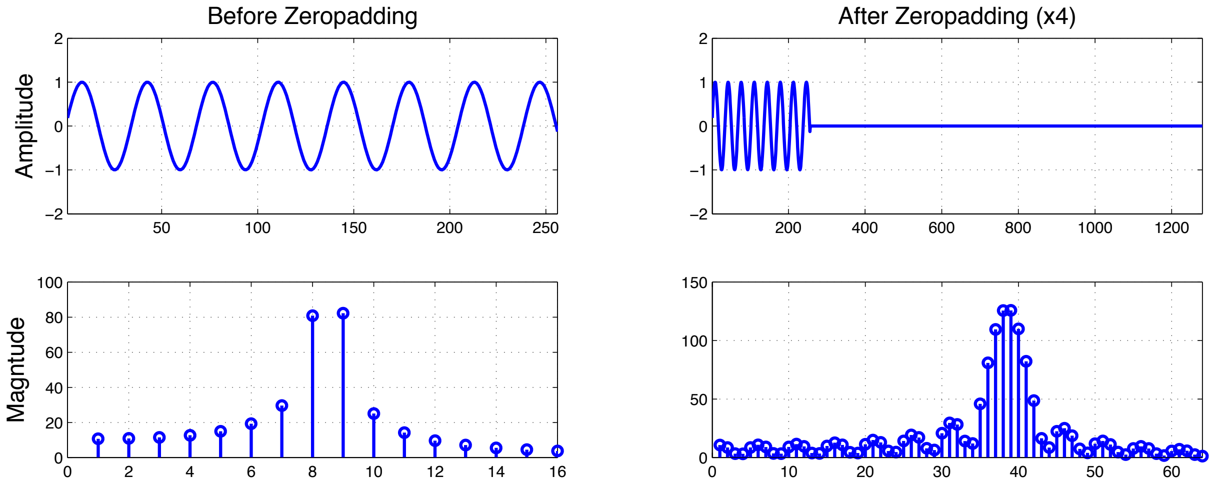




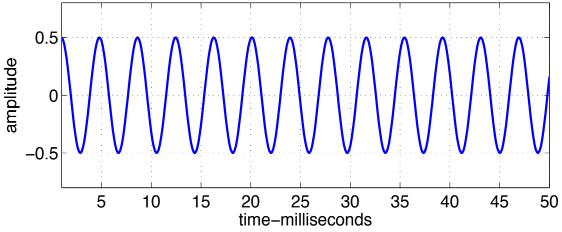
We usually use a large number of N to have a better frequency resolution

Zero-padding

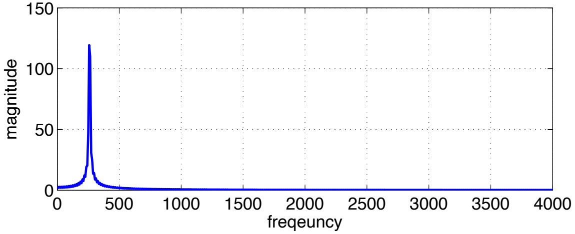
- Adding zeros to a windowed frame in time domain
  - Corresponds to “ideal interpolation” in the frequency domain
    - Convolution with the sinc function $(\sin(x)\,/x)$
  - In practice, FFT size increases by the size of zero-padding



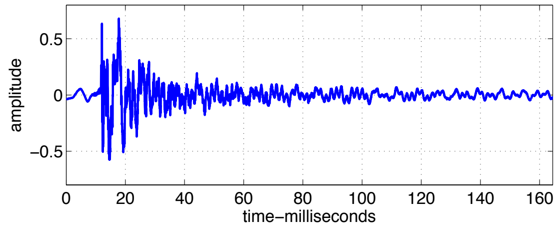
Examples of DFT



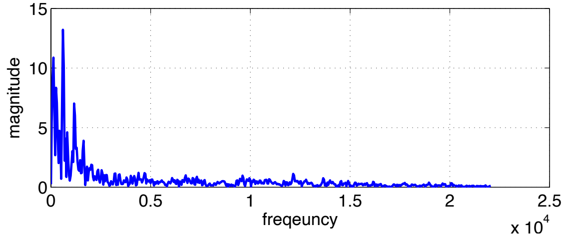
Sine: waveform



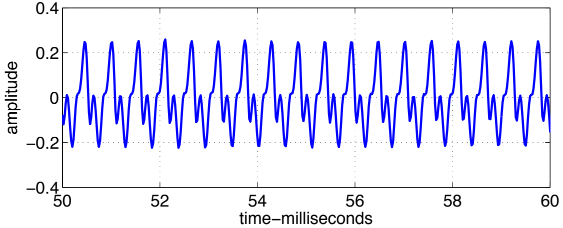
Sine: spectrum



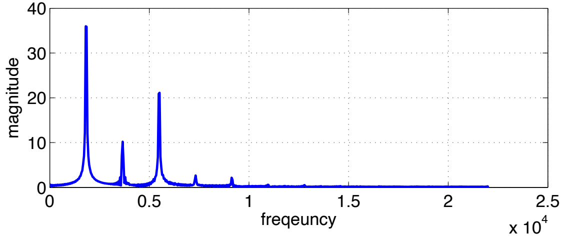
Drum: waveform



Drum: spectrum



Flute:waveform



Flute: spectrum

Fast Fourier Transform (FFT)

- Matrix multiplication view of DFT

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

- 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: $O(N^2) \rightarrow O(N\log_2 N)$
    - Divide and conquer

Short-Time Fourier Transform (STFT)

- DFT assumes that the signal is stationary within N samples
  - 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

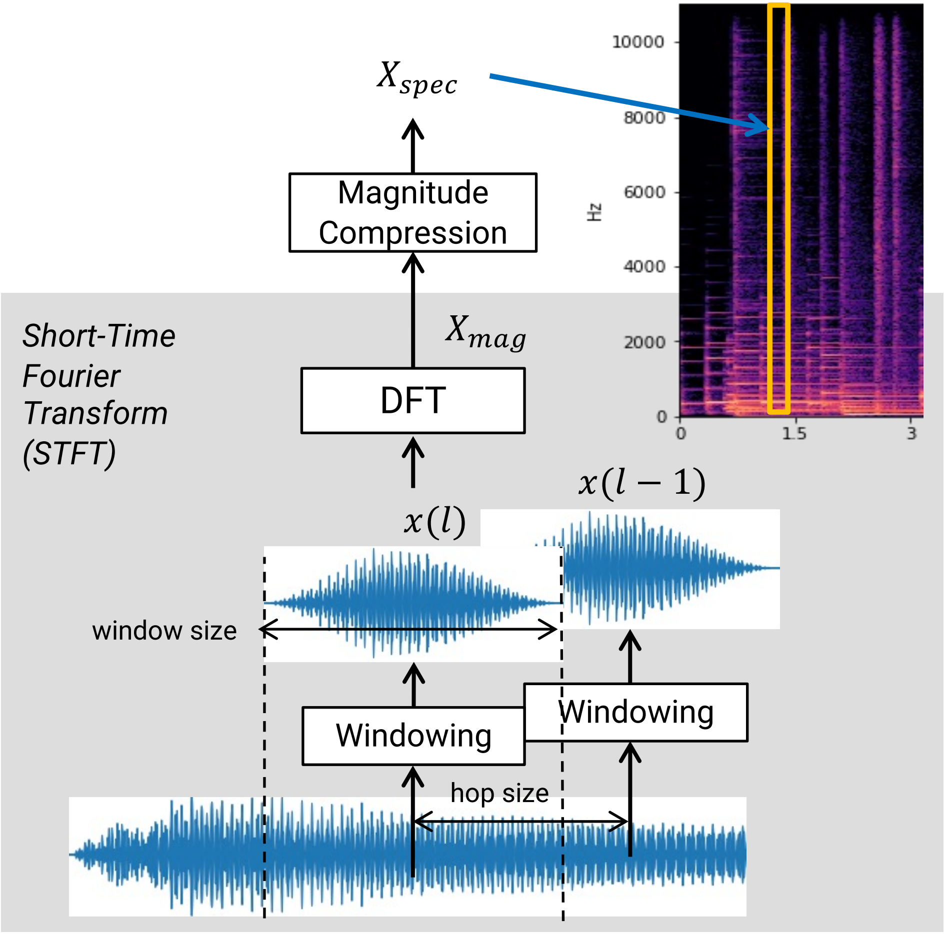
$w(n)$: window

N : FFT size

- This produces 2-D time-frequency representations
  - Get "spectrogram" from a series of the magnitude response

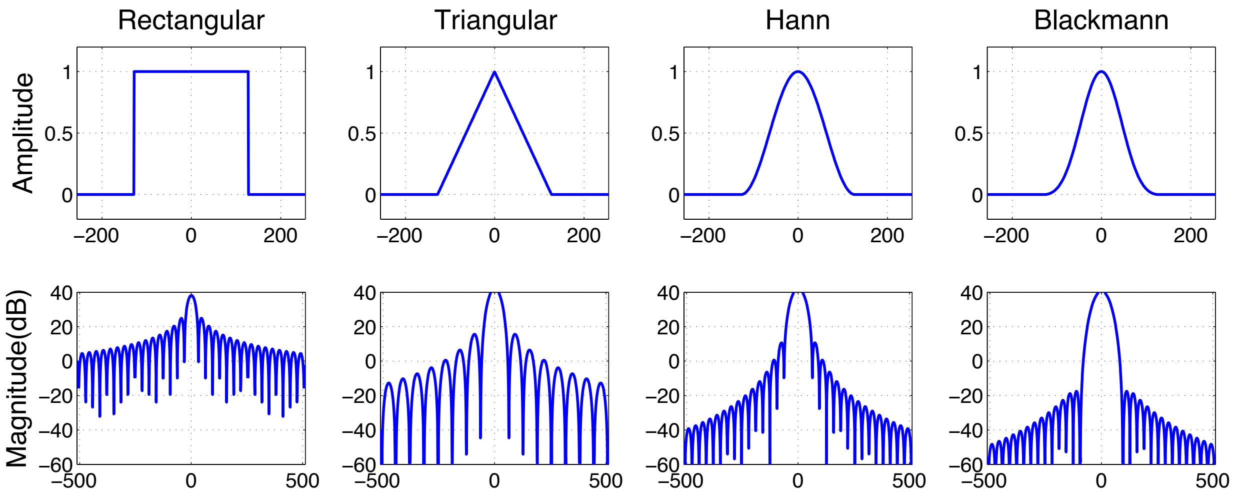
Computing Spectrogram

- For each short segment (frame)
  - Take a window (one frame)
  - Compute DFT (FFT)
  - Convert them to polar coordinate
    - Magnitude and Phase
  - Compress the magnitude
    - $20\log_{10} X_{mag}$: decibel
  - Shifting by a hop size
- Spectrogram parameters
  - Window size (FFT size)
  - Hop size
  - Window type

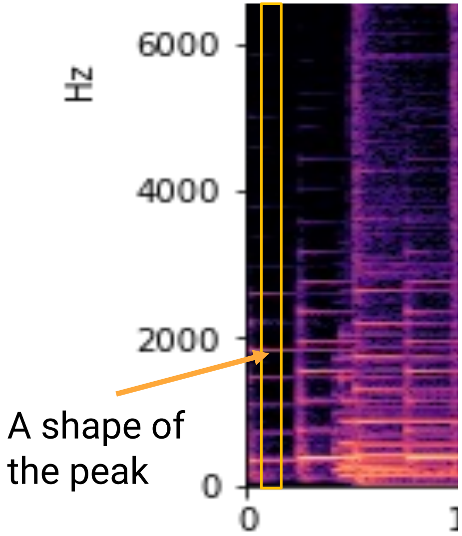


Window Function

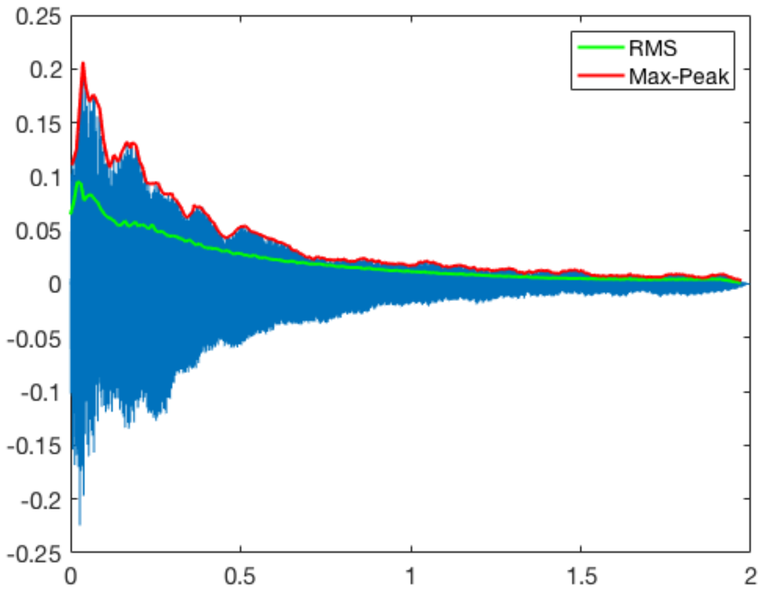
- Types of window functions
  - Trade-off between the width of main-lobe and the level of side-lobe
  - Hann window is the most widely used in music signal procenssing



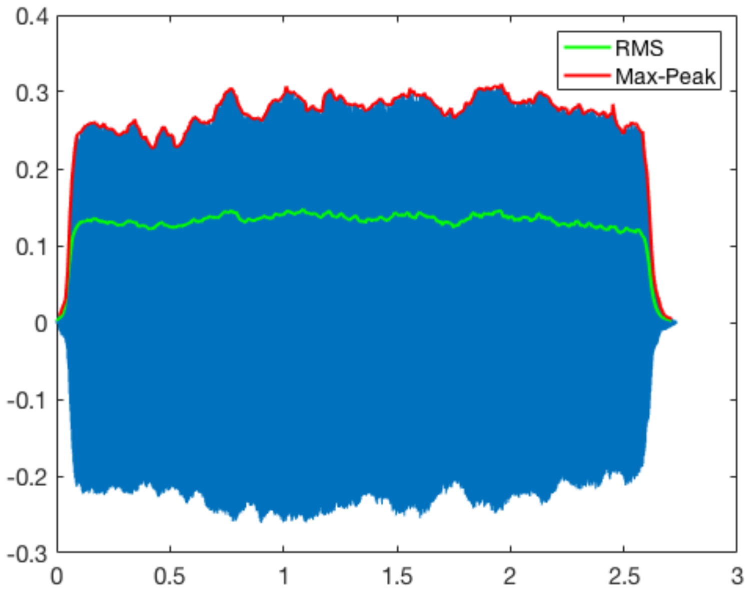
Spectra of windowed sinusoids



Example: Waveform

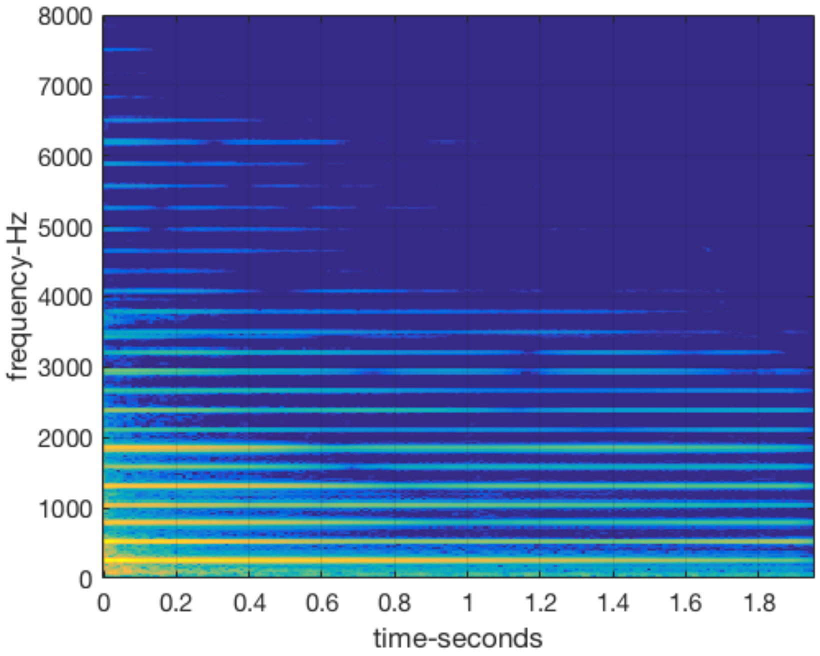


Piano C4 Note

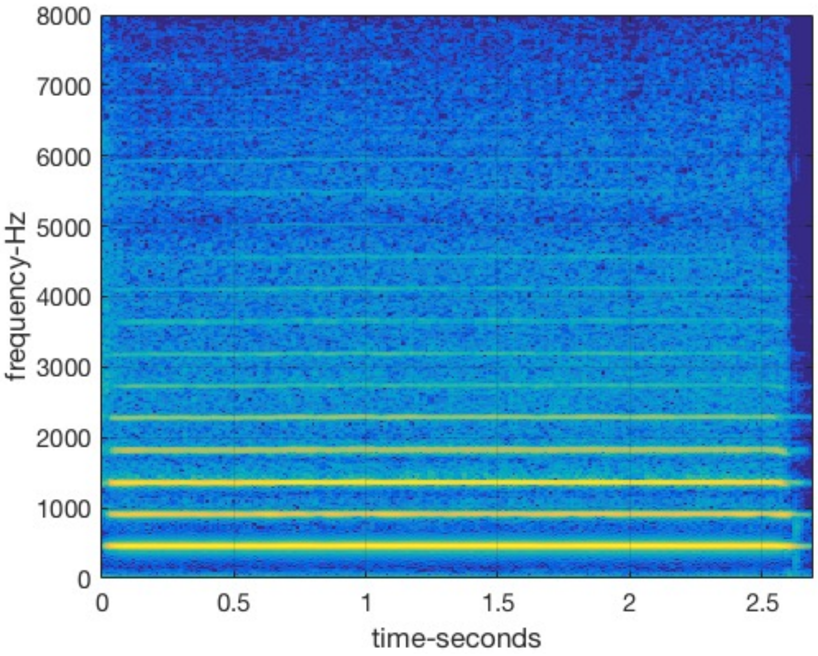


Flute A4 Note

Example: Spectrogram - 2D color map

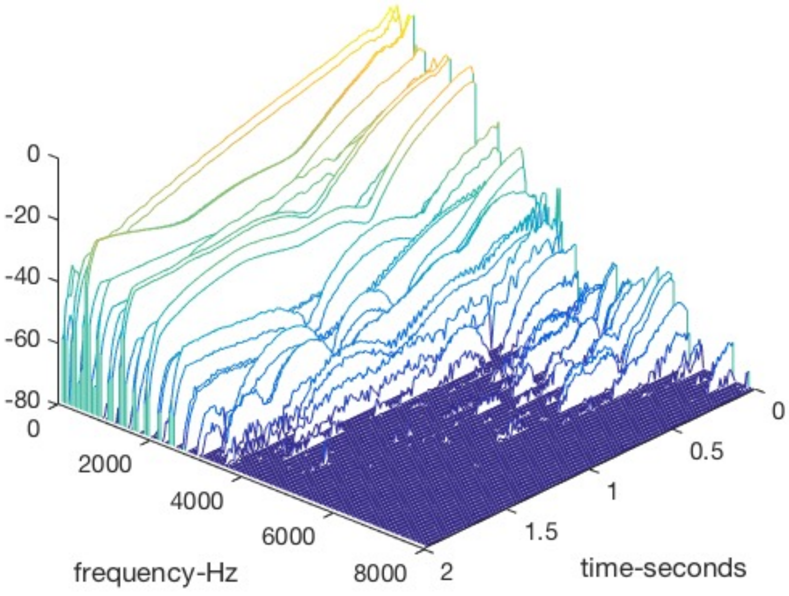


Piano C4 Note

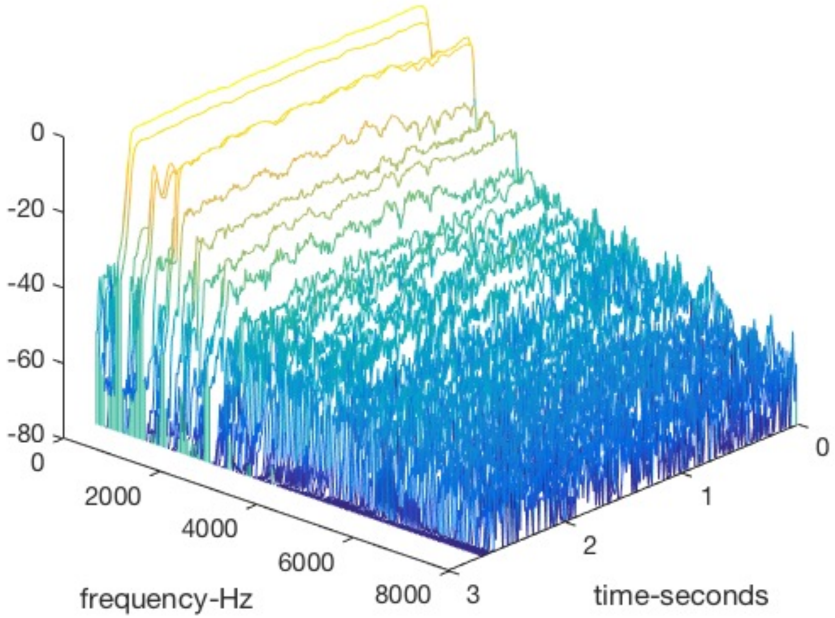


Flute A4 Note

Example: Spectrogram - 3D waterfall

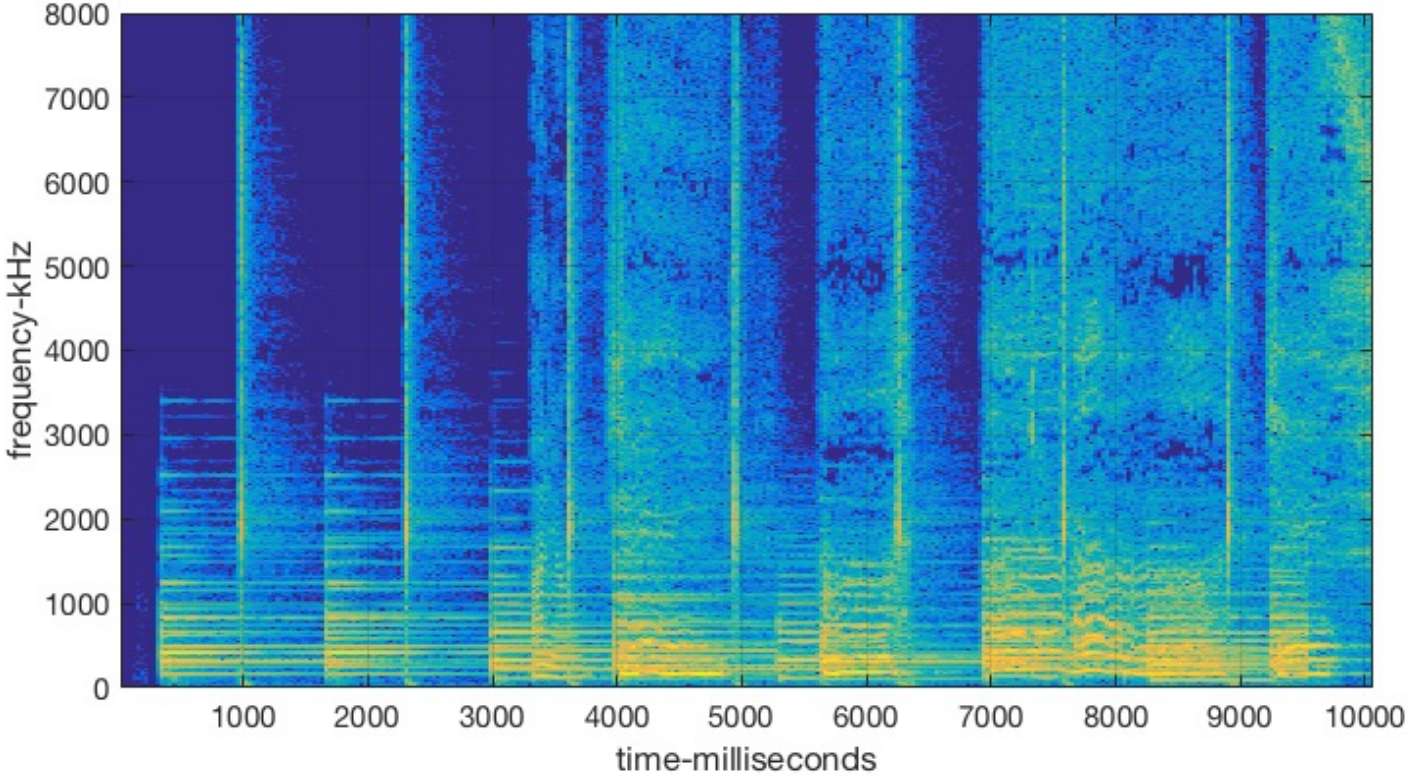


Piano C4 Note

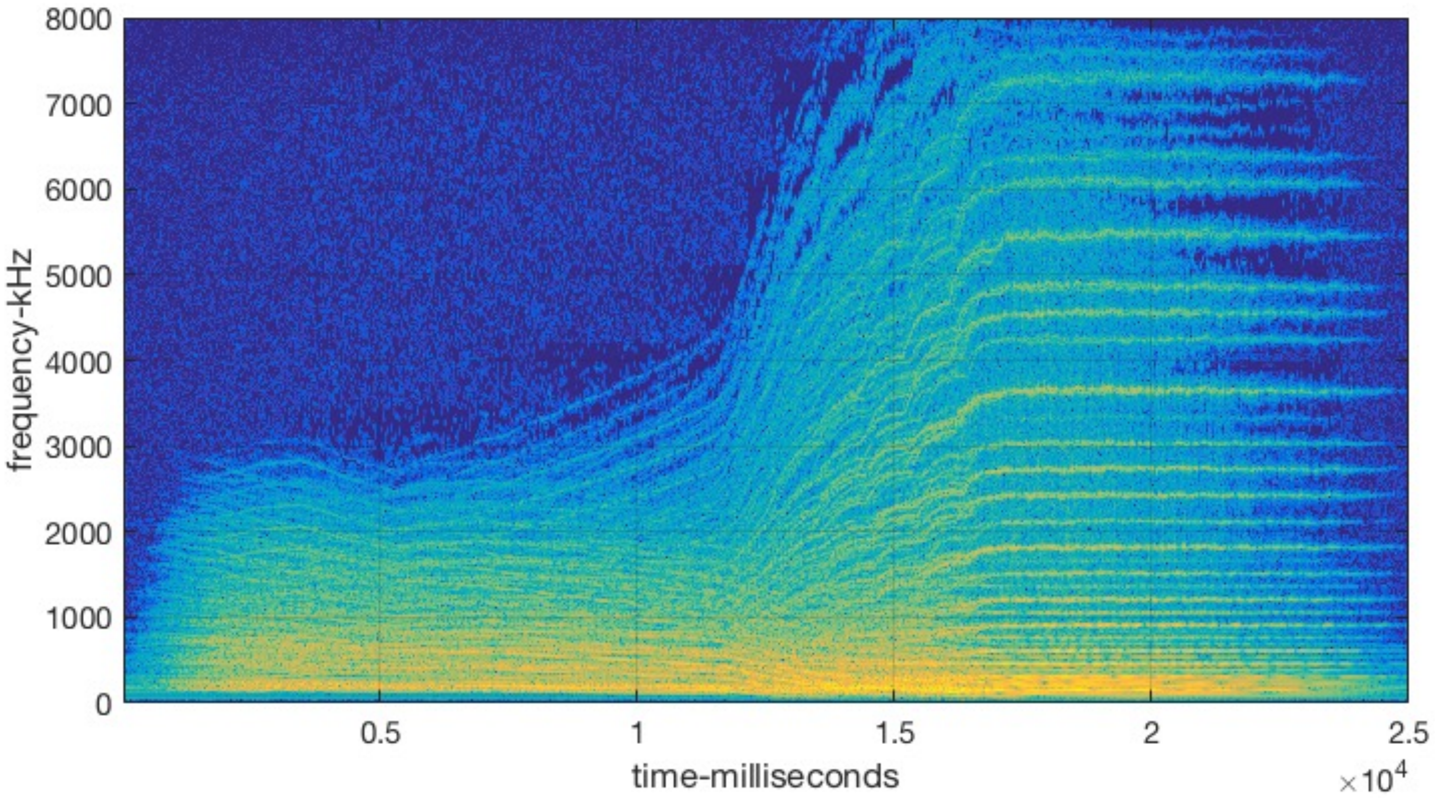


Flute A4 Note

Example: Pop Music



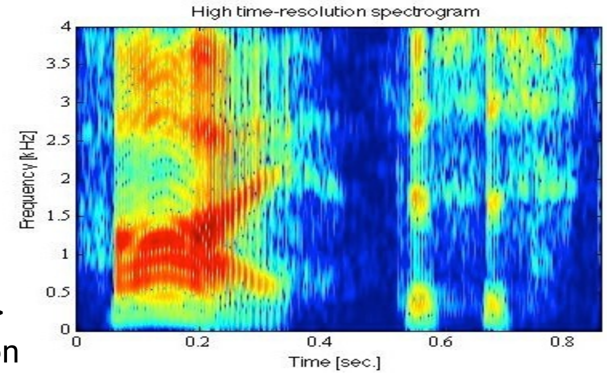
Example: Deep Note

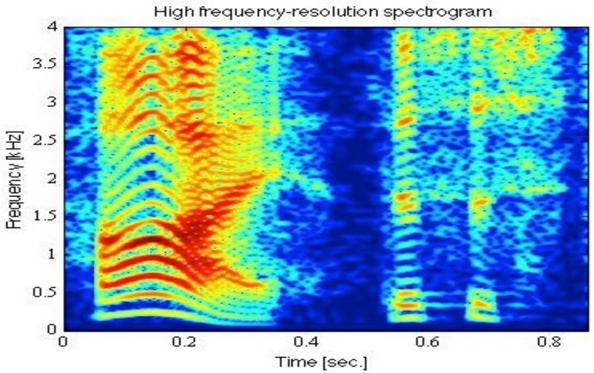


Effect of Window Size

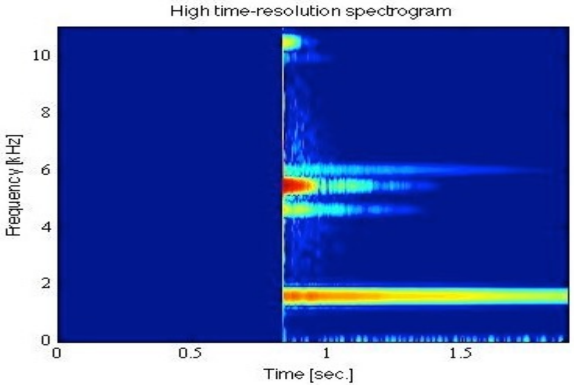
- Trade-off between time- and frequency-resolution by window size

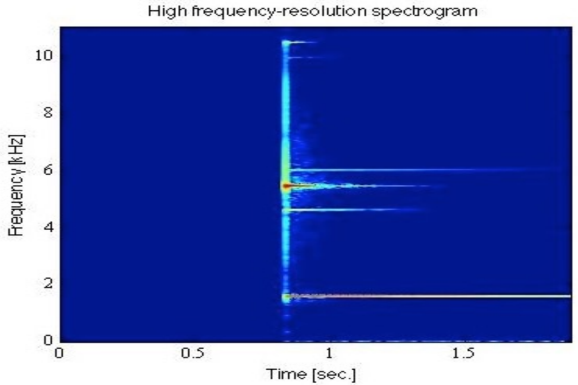
< Short window >
low freq.-resolution
high time-resolution





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





Real-Time Spectrograms

- https://musiclab.chromeexperiments.com/Spectrogram/