

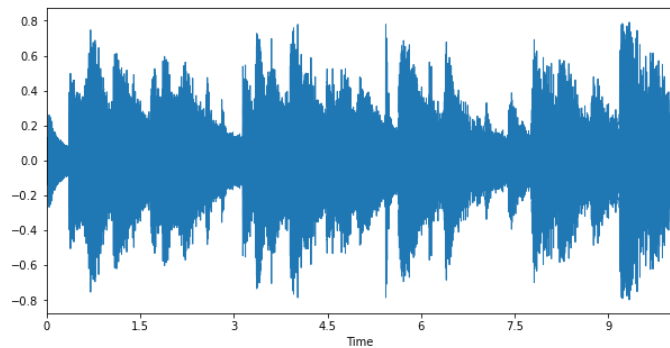
GCT634/AI613: Musical Applications of Machine Learning

Audio Data Representations

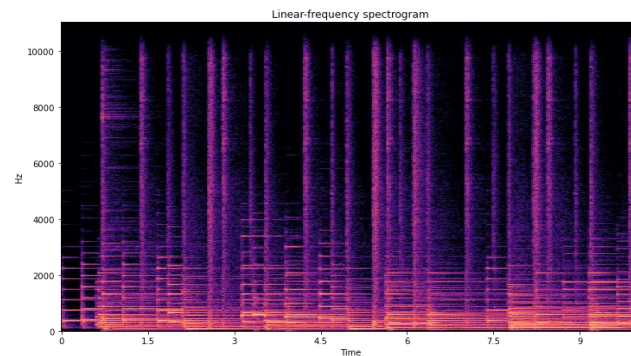


Juhan Nam

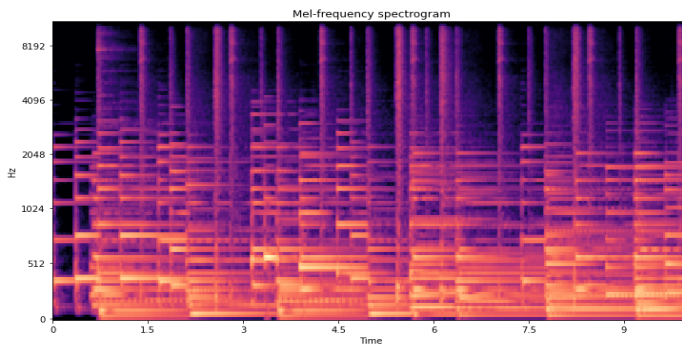
Types of Audio Data Representations



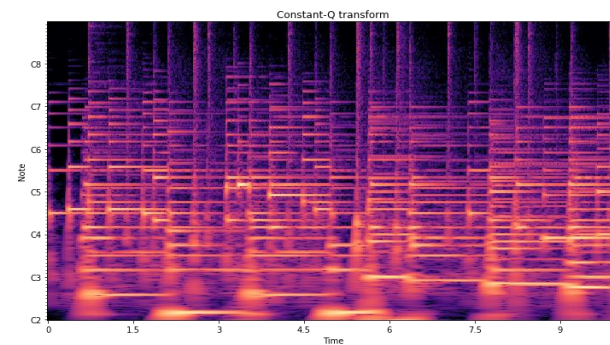
Waveform



Spectrogram



Mel-spectrogram



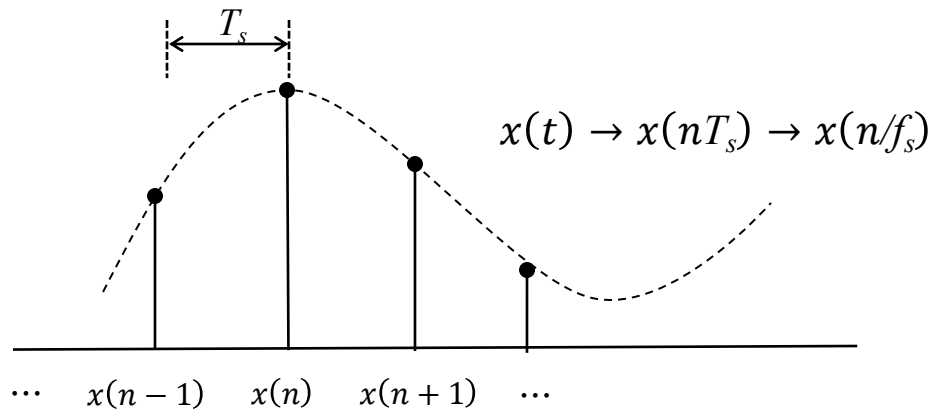
Constant-Q transform

Digital Audio

- Audio file formats: wav, mp3 or flac
- Standards: sampling rate, bit depth
 - Consumer audio (CD) : 44.1kHz, 16bits
 - Speech audio: 8/16 kHz, 8/16 bits
 - Professional audio: 48/96/192 kHz, 24/32 bits
- Why do they have the specific sampling rates and bit depths?
 - Need to understand the basic concept of **sampling** and **quantization**
 - Need to know the bandwidth (or maximum frequency) and dynamic range of the audio content

Sampling

- Convert **continuous-time** signals to **discrete-time** signals by periodically picking up the instantaneous values
 - Represented as a sequence of numbers
 - Sampling period (T_s): the amount of time between samples
 - **Sampling rate** ($f_s = 1/T_s$): the number of samples per second



Sampling Theorem

- How can we determine the sampling rate?
 - Too high: increase the number of samples
 - Too low: cannot reconstruct the original signal
- Sampling Theorem
 - The sampling rate must be greater than twice the maximum frequency in the signal in order to reconstruct the original signal

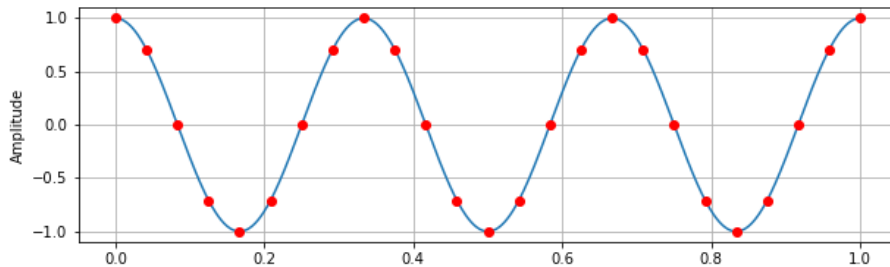
$$f_s > 2 \cdot f_m$$

f_s : sampling rate

f_m : maximum frequency of the signal

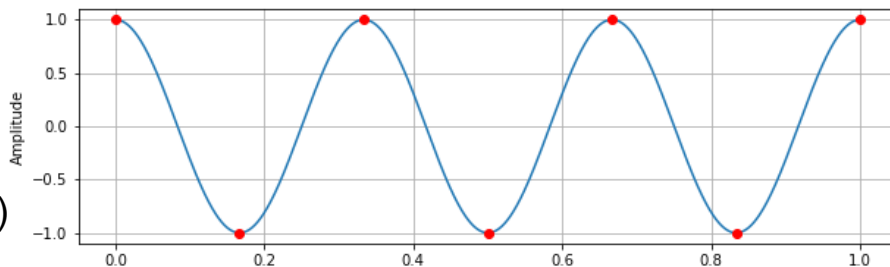
Sampling Theorem

$f_s > 2f_m$
(oversampled)



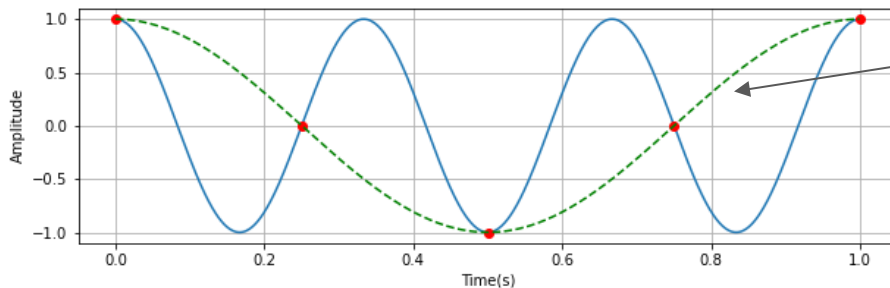
$$f_s = 24 \text{ Hz} \quad f_m = 3 \text{ Hz}$$

$f_s = 2f_m$
(critically sampled)



$$f_s = 6 \text{ Hz} \quad f_m = 3 \text{ Hz}$$

$f_s < 2f_m$
(undersampled)



The reconstructed sinusoid with 1 Hz frequency

$$f_s = 4 \text{ Hz} \quad f_m = 3 \text{ Hz}$$

Aliasing

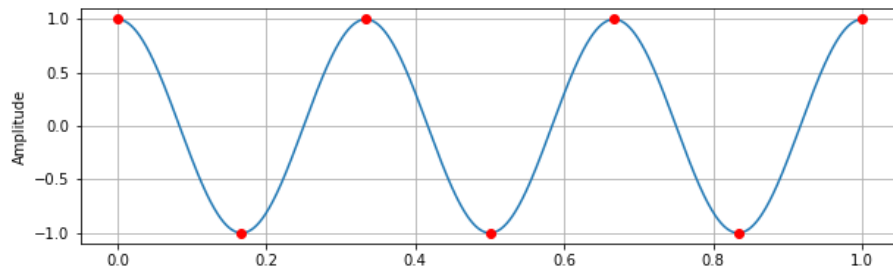
- A continuous-time sinusoid with frequency f_0 is sampled:

$$\begin{array}{lcl} x(t) = A \sin(2\pi f_0 t + \phi) & \xrightarrow{x(t) \rightarrow x(n/f_s)} & x(n) = A \sin(2\pi f_0 n / f_s + \phi) \\ x(t) = A \sin(2\pi(f_0 \pm l f_s)t + \phi) & \xrightarrow{\hspace{1.5cm}} & x(n) = A \sin(2\pi(f_0 \pm l f_s)n / f_s + \phi) \\ & (l = 1, 2, 3, \dots) & = A \sin(2\pi f_0 n / f_s \pm 2\pi l n + \phi) \\ & & = A \sin(2\pi f_0 n / f_s + \phi) \end{array}$$

- The sampled sinusoid with frequency f_0 has **aliases** at $f_0 \pm l f_s$

Aliasing

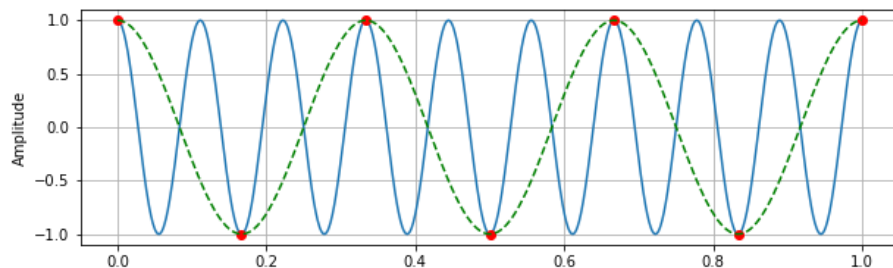
$$f = f_0$$



$$f_s = 6 \text{ Hz} \quad f = 3 \text{ Hz}$$

$$f = f_0 + f_s$$

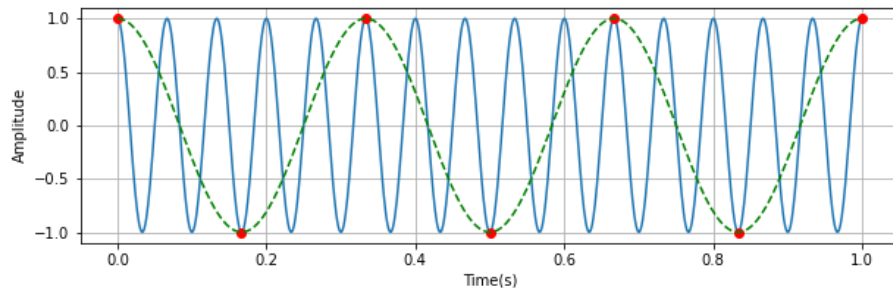
(Alias)



$$f_s = 6 \text{ Hz} \quad f = 9 \text{ Hz}$$

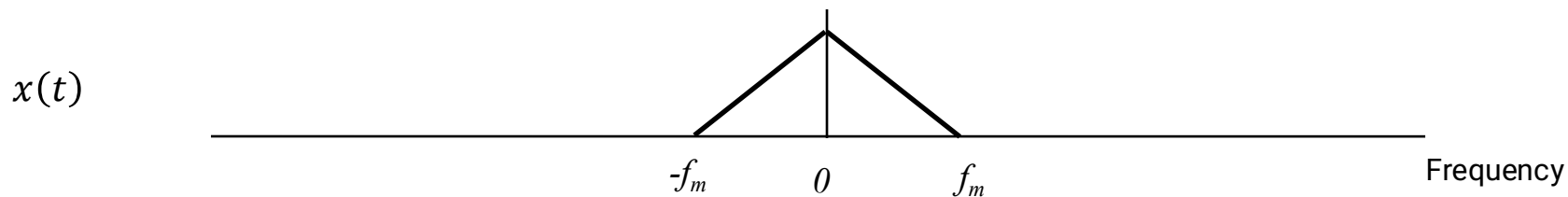
$$f = f_0 + 2f_s$$

(Alias)

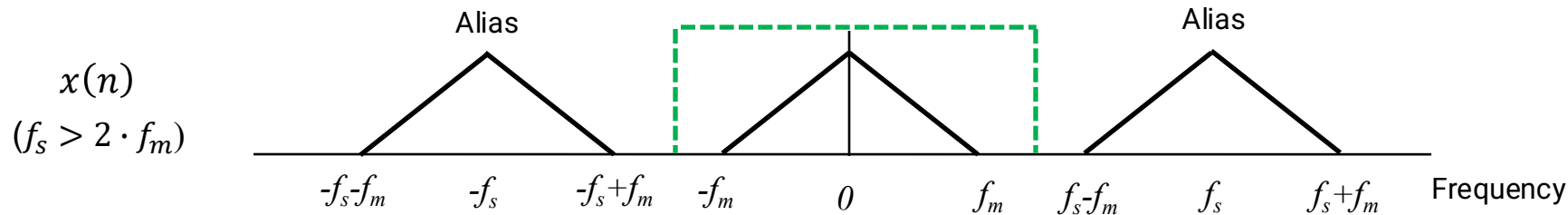


$$f_s = 6 \text{ Hz} \quad f = 15 \text{ Hz}$$

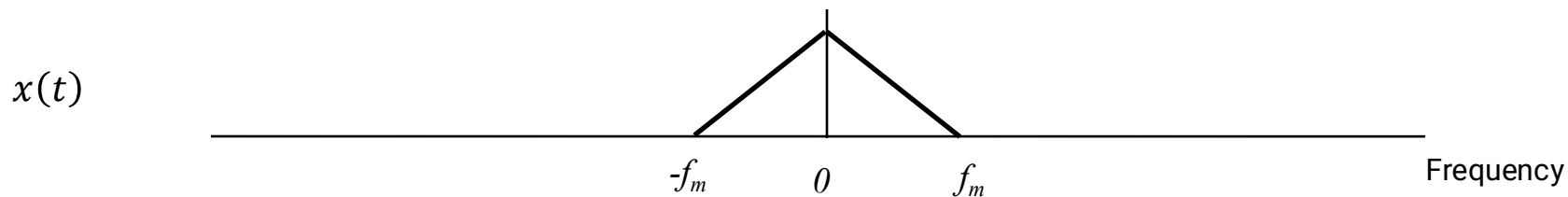
Sampling in the Frequency Domain



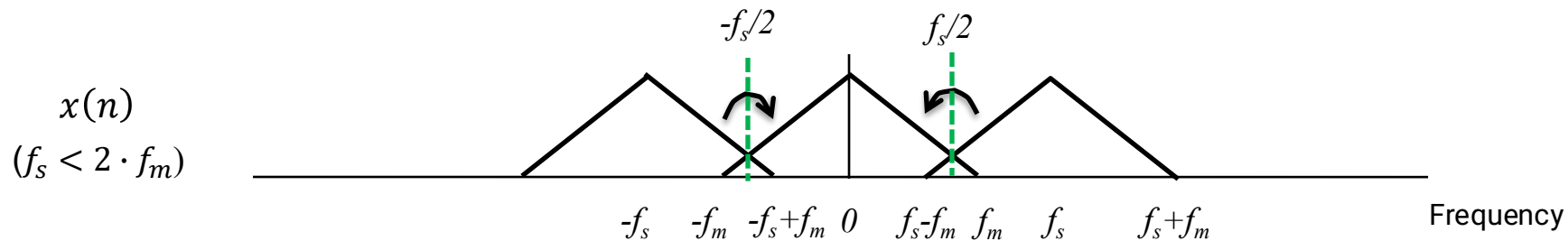
The original signal can be reconstructed using a lowpass filter



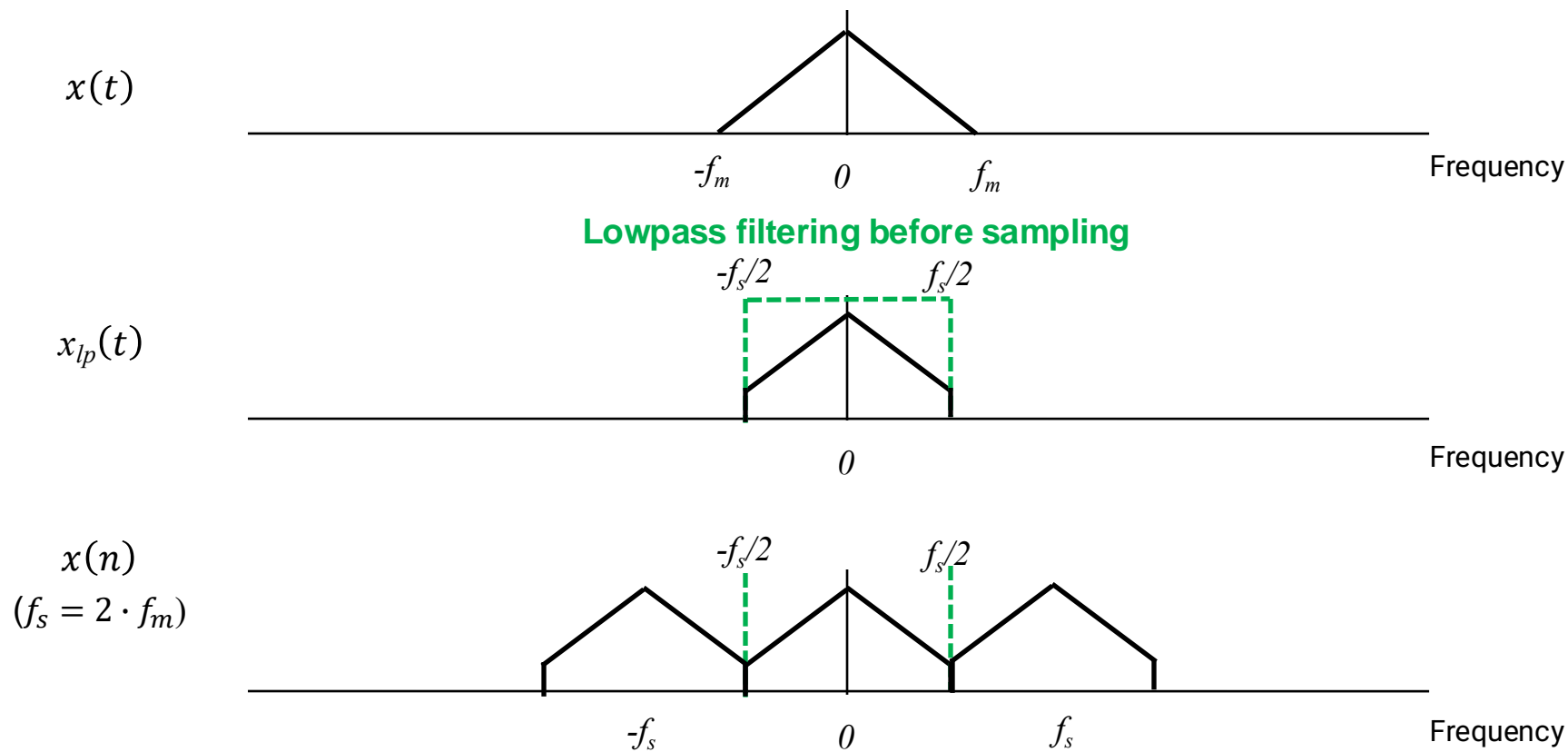
Sampling in the Frequency Domain: Aliasing



The high-frequency content above half the sampling rate (Nyquist rate) is folded over

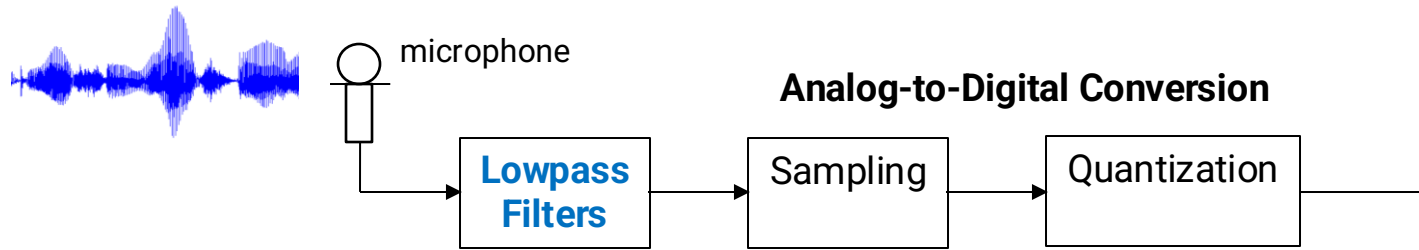


Sampling in the Frequency Domain: Lowpass filtering



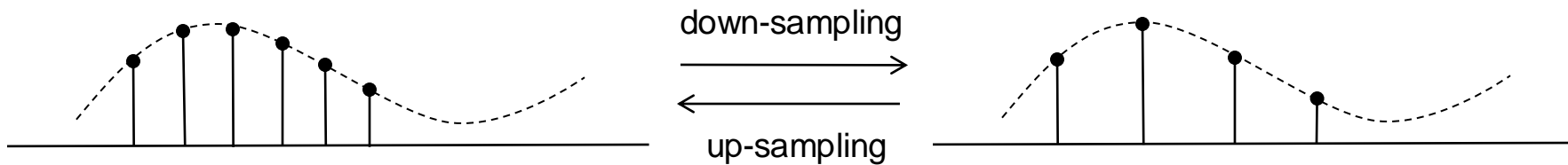
Lowpass filter in Sampling

- The lowpass filter is implemented as an analog circuit when it is used before the audio-to-digital conversion in the digital audio system



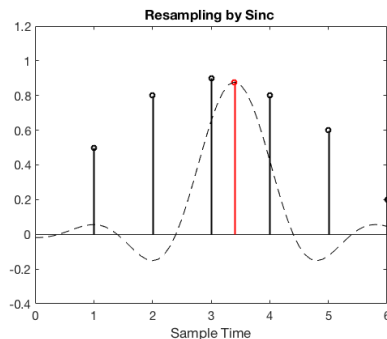
Resampling

- Sampling in the digital domain
 - Increasing or decreasing the sampling rate



Resampling

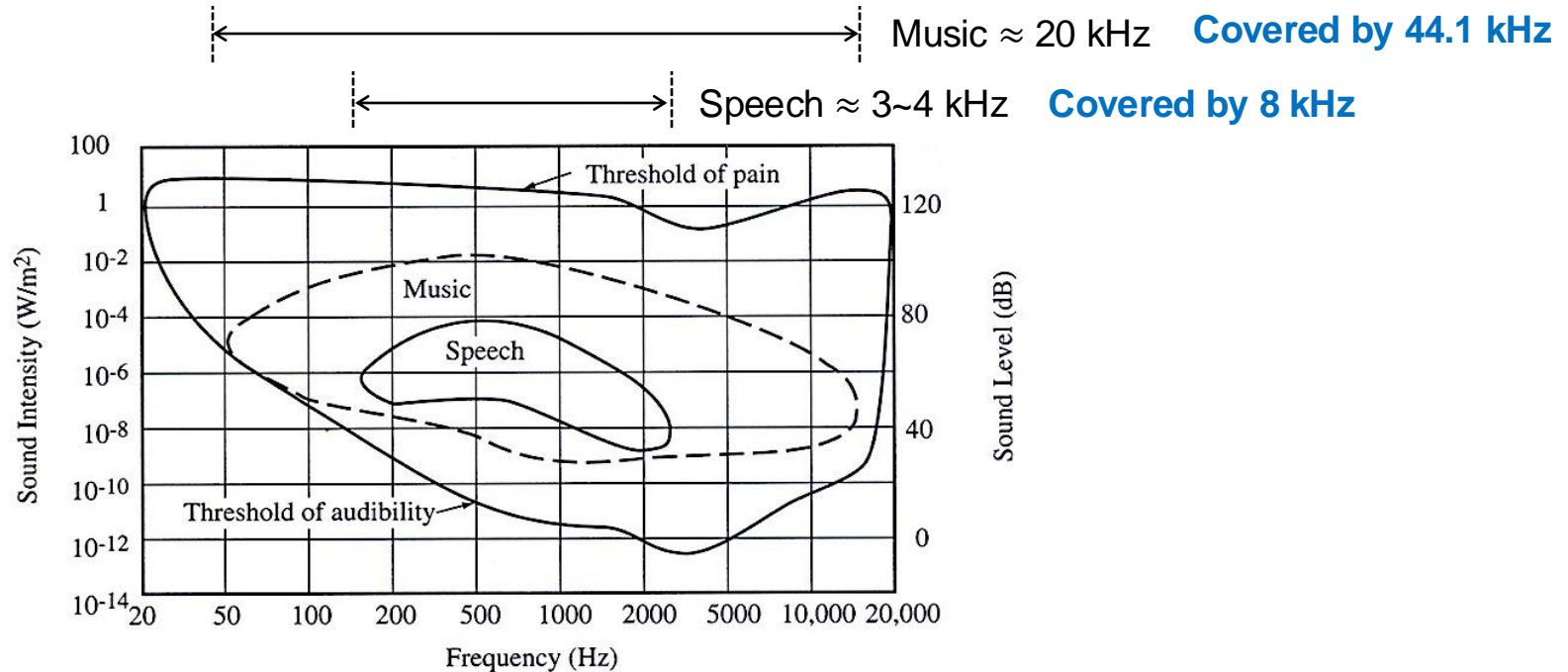
- It is common to down-sample audio files in music and audio research
 - Speeds up the processing and save the disk space
 - For example, 44.1kHz \rightarrow 22.05 kHz
 - High frequency content above 10 kHz has relatively less information
- A “digital” lowpass filter is applied to avoid aliasing in down-sampling
 - The lowpass filter is typically implemented with a finite impulse response filter such as a windowed sinc function



See more at <https://ccrma.stanford.edu/~jos/resample/>

Sampling Rate in Audio Standard

- Determined by the maximum frequency of signals

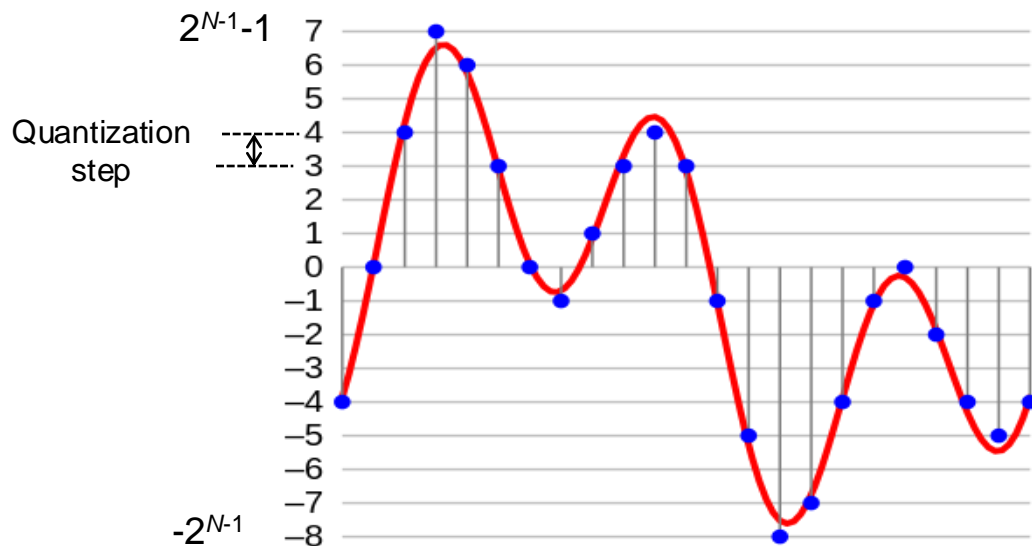


Review

- Keep in mind: sampling limits the available frequency range
- Sampling rate f_s → the audio content is available between 0 and $f_s/2$
 - Sampling rate at 44.1 kHz → 0 to 22.05 kHz are available
 - Sampling rate at 22.05 kHz → 0 to 11.025 kHz are available
- When set up the frequency range in Spectrogram

Quantization

- Round the amplitude to the nearest discrete steps
 - The number of discrete steps are determined by the bit depth
 - N bits range from -2^{N-1} to $2^{N-1}-1$: 8 bit (-128 to 127), 16 bit (-32767 to 32766)

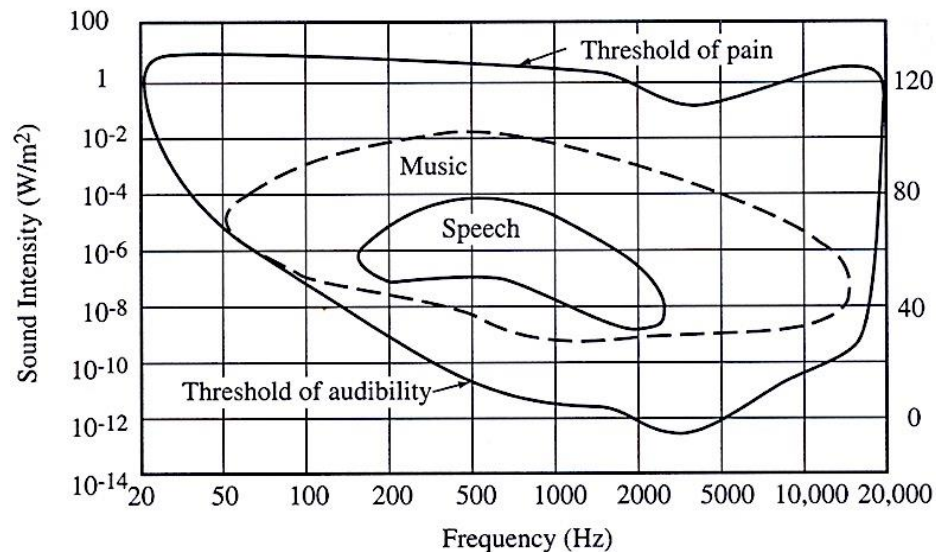


Bit Depth and Dynamic Range

- Bit depth determines dynamic range of digital signals
- Dynamic range = $20\log_{10}(\text{maximum value} / \text{minimum value})$
 - 8 bits = $20\log_{10}(127/1) \approx 48\text{dB}$
 - 16 bit = $20\log_{10}(32766/1) \approx 96\text{dB}$
 - Adding one bit (x2) increases 6dB : N bits $\approx 6N$ dB

Bit Depth in Audio Standard

- Determine bit depth to cover the dynamic range of audio content



Sound Level (dB)

Music ≈ 80 dB

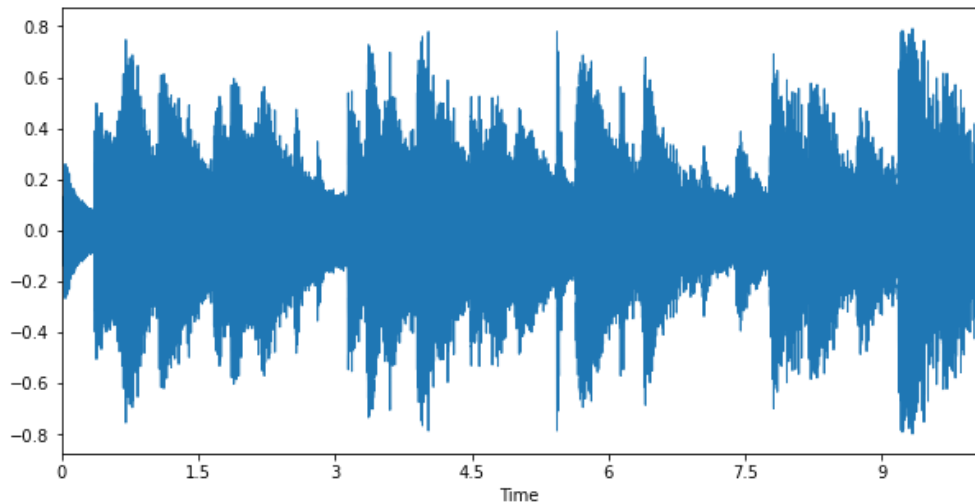
Covered by 16 bits (96dB)

Speech ≈ 48 dB

Covered by 8 bits (48dB)

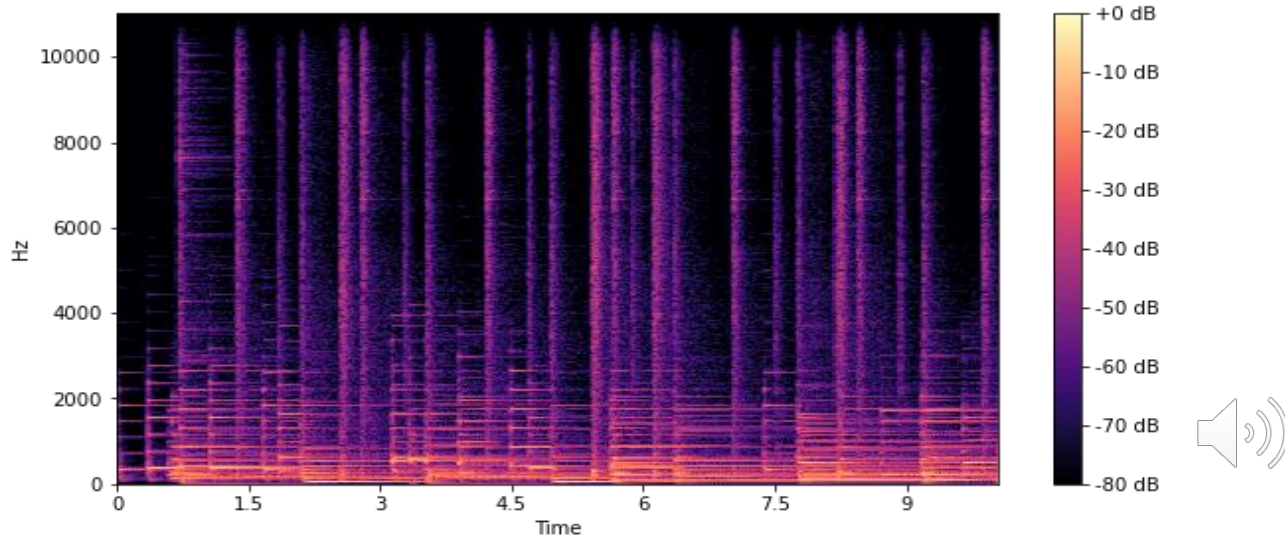
Waveform

- Waveform is a natural representation of audio but limited in analyzing the content
 - Mainly show the temporal energy



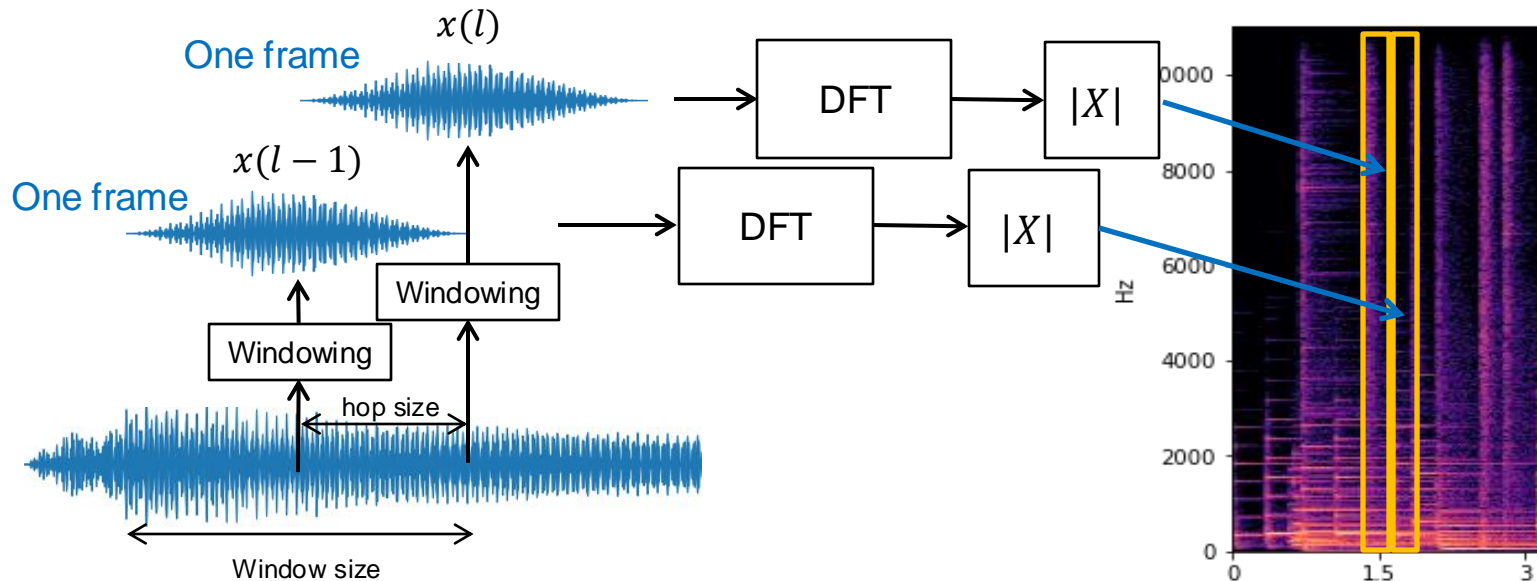
Spectrogram

- 2D-image representation of audio using Short-Time Fourier Transform
 - x-axis: time, y-axis: frequency, color: magnitude response
 - It is common to use dB scale (a log scale) for the magnitude
 - Easy to match what you hear to what you see



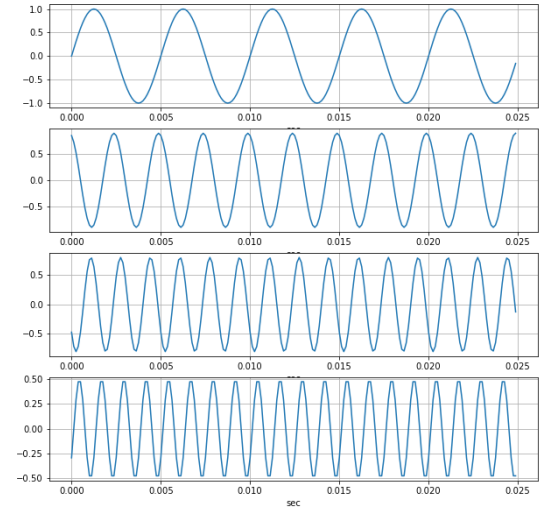
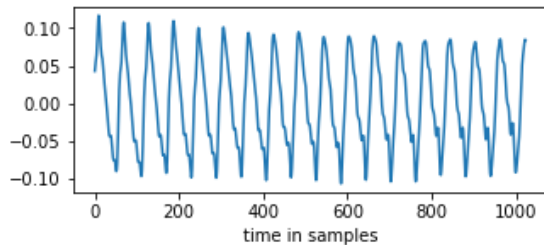
Short-Time Fourier Transform (STFT)

- STFT is a series of discrete Fourier transform (DFT) computed from windowed waveform segments



Discrete Fourier Transform (DFT)

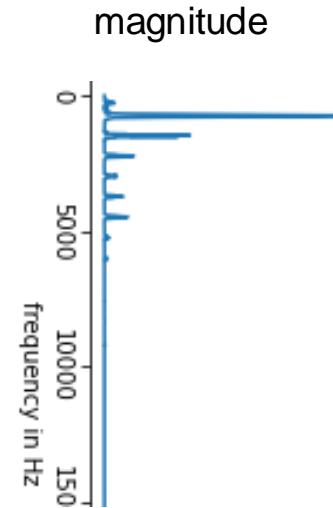
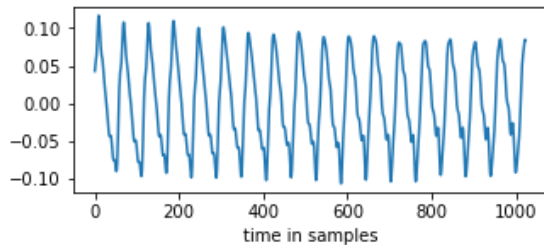
- Decompose the signal into sinusoidal components
 - Compute the magnitude and phase of the sinusoidal components



This image is from Pink Floyd's "The Dark Side of the Moon"

Discrete Fourier Transform (DFT)

- The signal is represented in the frequency domain



This image is from Pink Floyd's "The Dark Side of the Moon"

Discrete Fourier Transform (DFT): Definition

- Inner product between $x(n)$ with a length N and a complex sinusoid $e^{j\frac{2\pi kn}{N}}$ with a length N at frequency k ($k = 0, 1, \dots, N - 1$)

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}} = \sum_{n=0}^{N-1} x(n)\left(\cos\frac{2\pi kn}{N} - j\sin\frac{2\pi kn}{N}\right) \\ &= \sum_{n=0}^{N-1} x(n)\cos\frac{2\pi kn}{N} - j\sum_{n=0}^{N-1} x(n)\sin\frac{2\pi kn}{N} = X_{re}(k) + jX_{im}(k) \end{aligned}$$

$e^{j\frac{2\pi kn}{N}} = \cos\frac{2\pi kn}{N} + j\sin\frac{2\pi kn}{N}$

Euler's identity

- The magnitude and phase

$$X_{mag}(k) = \sqrt{X_{re}(k)^2 + X_{im}(k)^2} \quad X_{phase}(k) = \tan^{-1}\left(\frac{X_{im}(k)}{X_{re}(k)}\right)$$

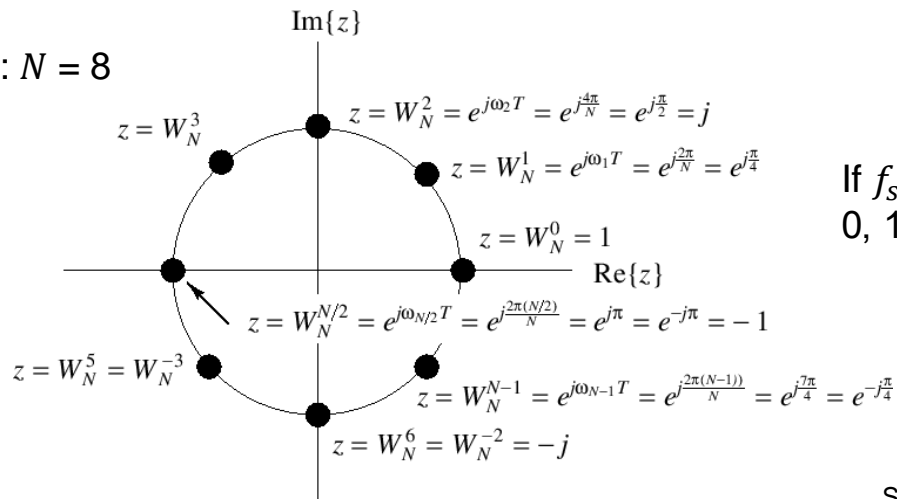
Complex Sinusoids in DFT

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

$$s_k(n) = e^{j\frac{2\pi kn}{N}} = \cos\frac{2\pi kn}{N} + j\sin\frac{2\pi kn}{N}$$

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

Example: $N = 8$



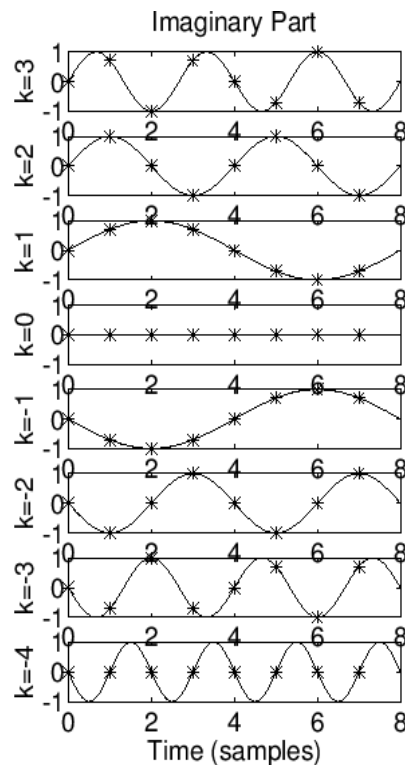
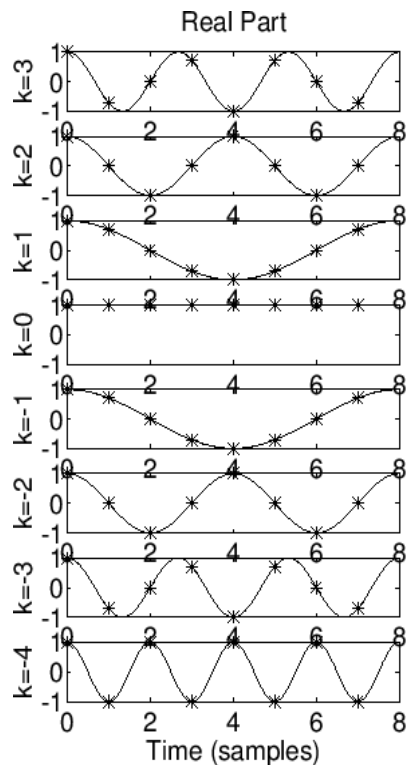
If $f_s=8000$ Hz, the DFT frequencies are 0, 1000, 2000, 3000, ..., 7000 Hz

Complex Sinusoids in DFT

Example: $N = 8$

Even-symmetric
with period N

$$\begin{aligned}\operatorname{Re}\{s_k(k)\} &= \operatorname{Re}\{s_k(-k)\} \\ \operatorname{Re}\{s_k(k)\} &= \operatorname{Re}\{s_k(N - k)\} \\ \operatorname{Re}\{s_k(k)\} &= \operatorname{Re}\{s_k(l \cdot N + k)\}\end{aligned}$$



Odd-symmetric
with period N

$$\begin{aligned}\operatorname{Im}\{s_k(k)\} &= -\operatorname{Im}\{s_k(-k)\} \\ \operatorname{Im}\{s_k(k)\} &= -\operatorname{Im}\{s_k(N - k)\} \\ \operatorname{Im}\{s_k(k)\} &= \operatorname{Im}\{s_k(l \cdot N + k)\}\end{aligned}$$

Orthogonality of Complex Sinusoids

- Inner product between two complex sinusoids
 - Two complex sinusoids with different frequencies are **orthogonal**

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

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

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

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

Discrete Fourier Transform (DFT)

- Matrix multiplication view: $X = S^* \cdot x$

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

- S is a unitary matrix (an orthogonal matrix for complex number) from the previous slide

$$S \cdot S^* = S^* \cdot S = N \cdot I$$

N : input length

I : identity matrix

S^* : conjugate transpose

- **DFT is the orthogonal projection on S^***

Inversed DFT

- Obtain **Inverse DFT** from the orthogonality: $x = S^{-1} \cdot X = \frac{1}{N} S^* \cdot X$

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

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

- The inverse DFT indicates that the original signal is the sum of sinusoidal components over frequency k , and the magnitude and phase of the sinusoidal components are determined by DFT

Properties of DFT

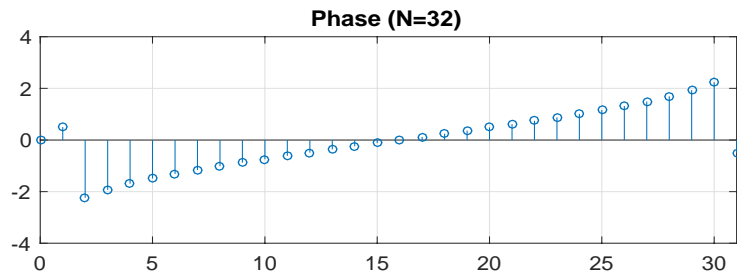
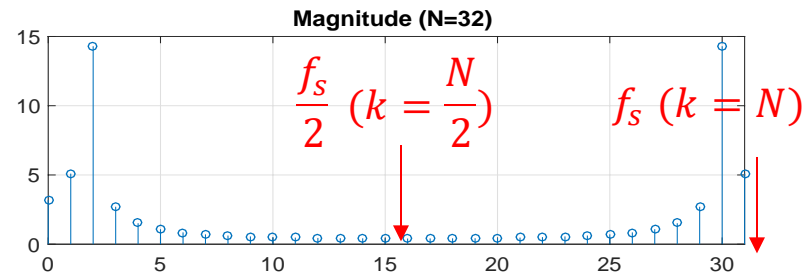
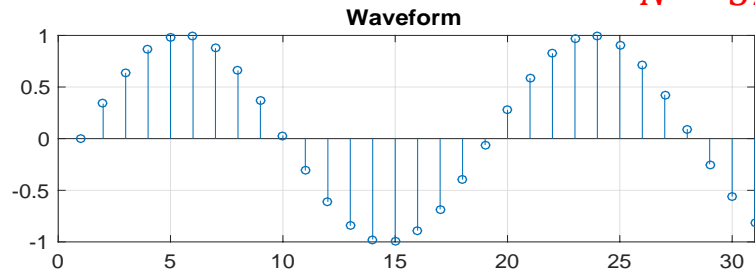
- Symmetry

- $|X(k)| = |X(-k)| = |X(N - k)|$
- $\angle X(k) = -\angle X(-k) = -\angle X(N - k)$

- Periodicity

- $X(k) = X(k \pm N) = X(k \pm 2N) = \dots$

$N = 32$

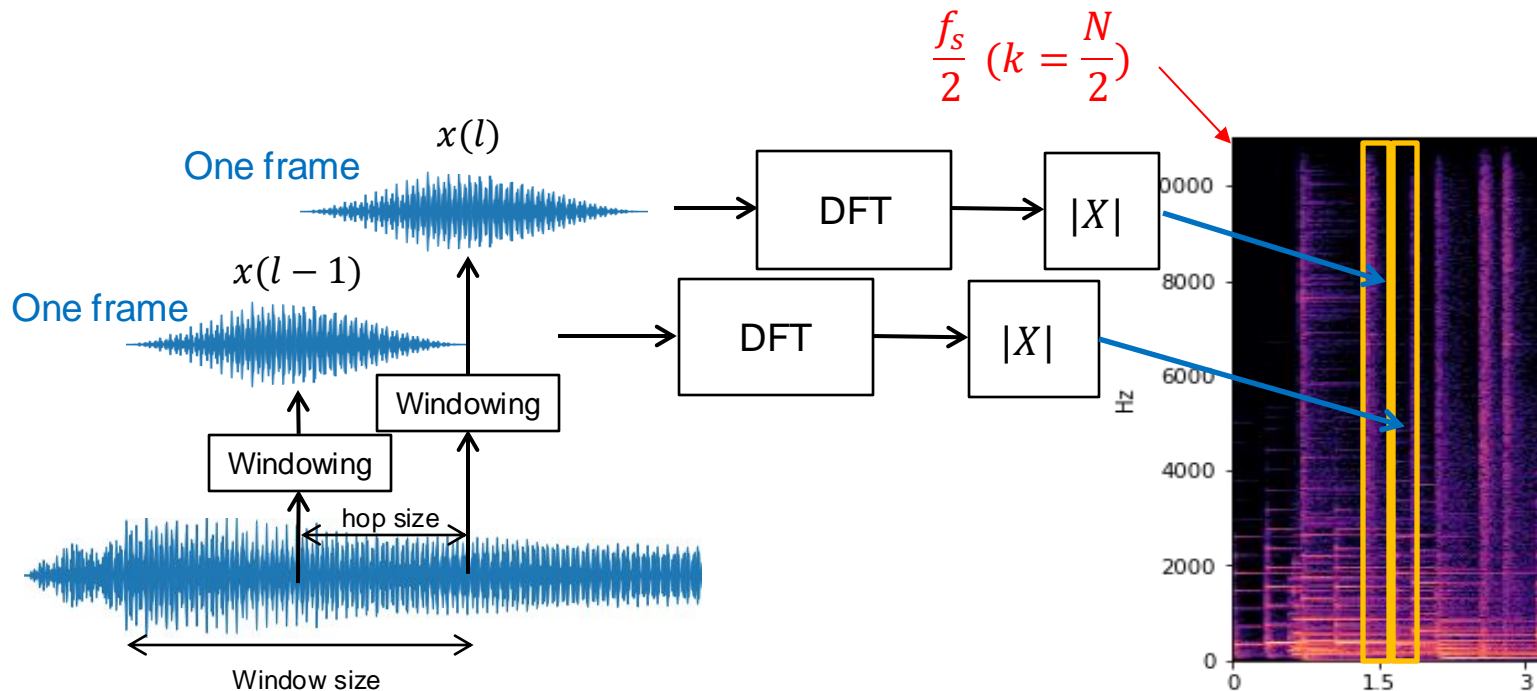


Fast Fourier Transform (FFT)

- In practice, we use an FFT algorithm instead of direct matrix multiplication
 - Divide the matrix into small matrices recursively
 - Complexity reduction: $O(N^2) \rightarrow O(N \log_2 N)$:
 - See more about the complexity reduction in Gilbert Strang's lecture on FFT (<https://www.youtube.com/watch?v=M0Sa8fLOajA>)

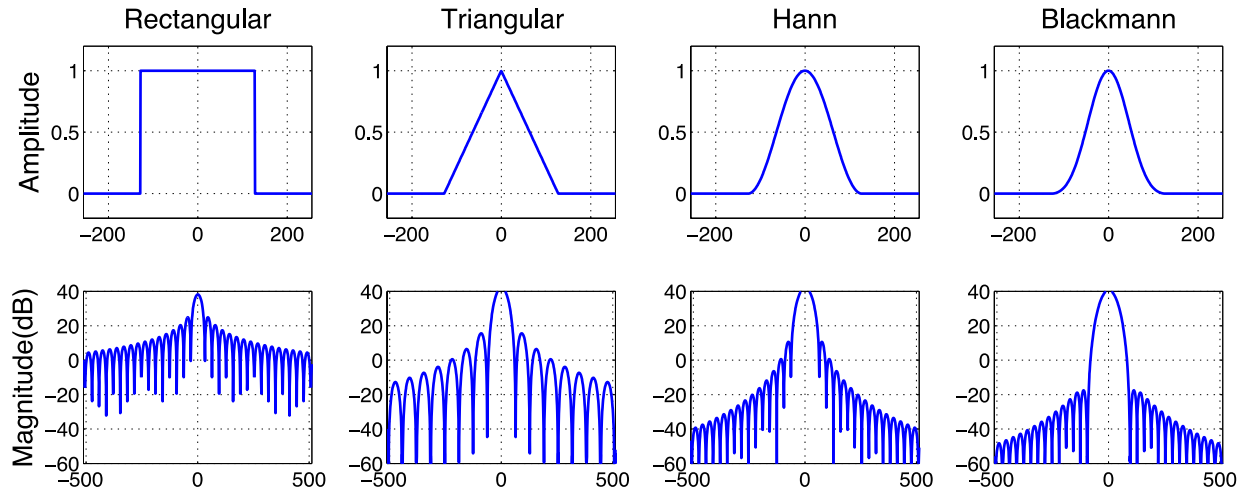
Review of STFT

- STFT is a series of DFT computed from windowed waveform segments
 - Spectrogram changes according to window type, window size, and hop size

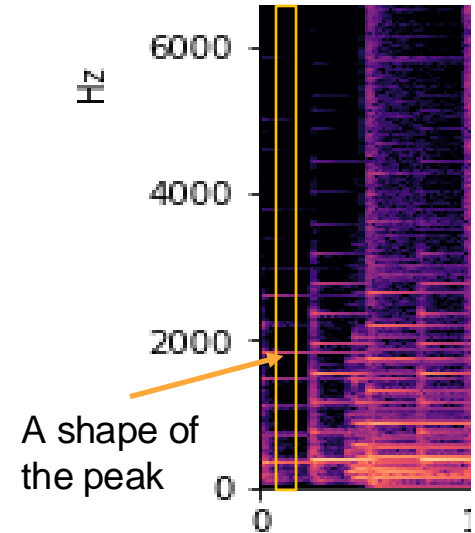


Window Types

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

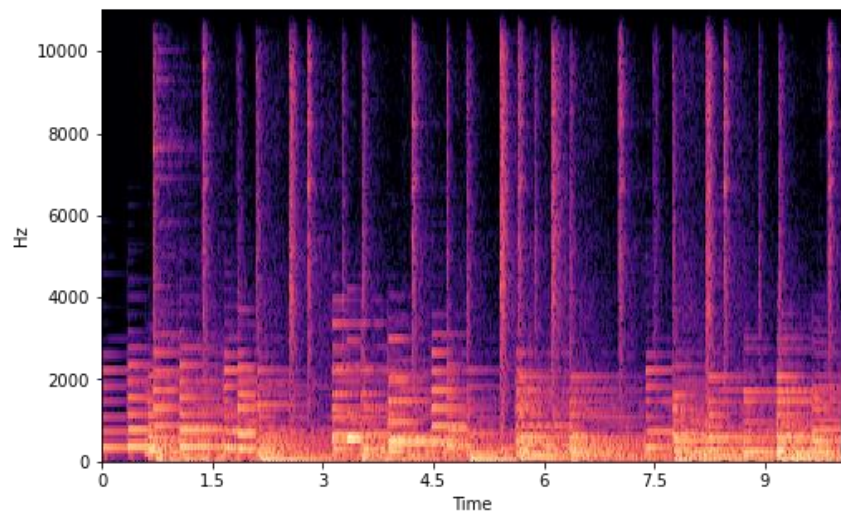


Spectra of windowed sinusoids

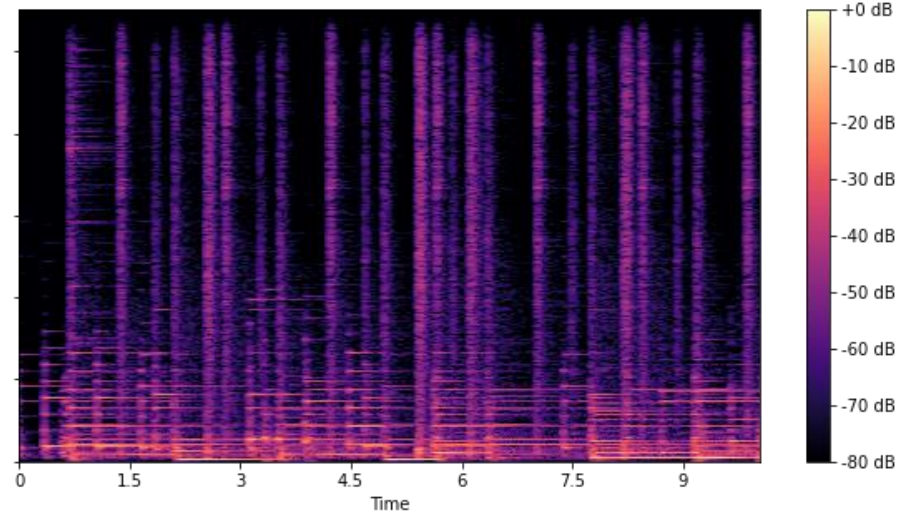


Effect of Window Size

- Trade-off between time and frequency resolutions
 - Short window: low frequency-resolution and high time-resolution
 - Long window: high frequency-resolution and low time-resolution



Window size: 256 samples



Window size: 1024 samples

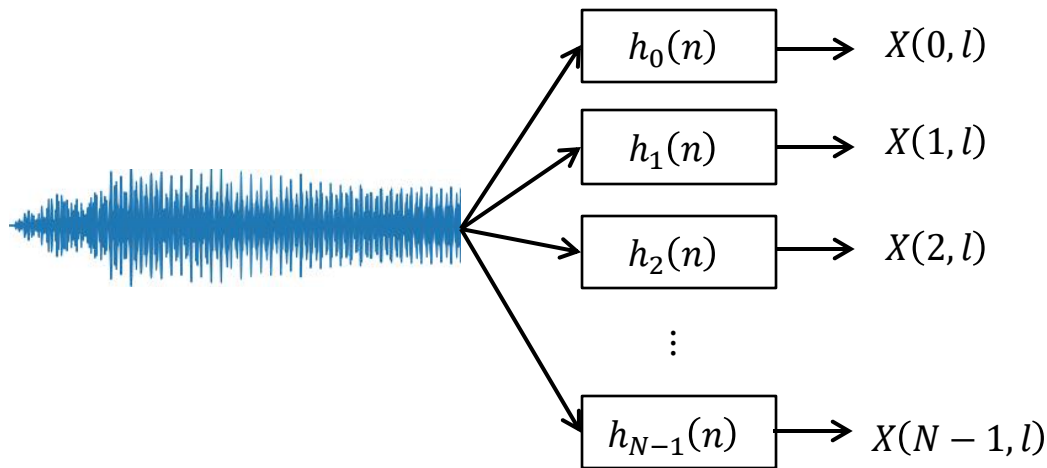
Filter Bank View of STFT

- STFT can be viewed as bandpass filter banks

$$X(k, l) = \sum_{n=-\infty}^{\infty} x(n) \boxed{w(n - l \cdot R) e^{-j\left(\frac{2\pi k(n-l \cdot R)}{N}\right)}} = \sum_{n=-\infty}^{\infty} x(n) \boxed{h_k(n - l \cdot R)} = x * h_k(n)$$

R : hop size

This is the convolution of $x(n)$ and $h(n)$ but the output is down-sampled by R .



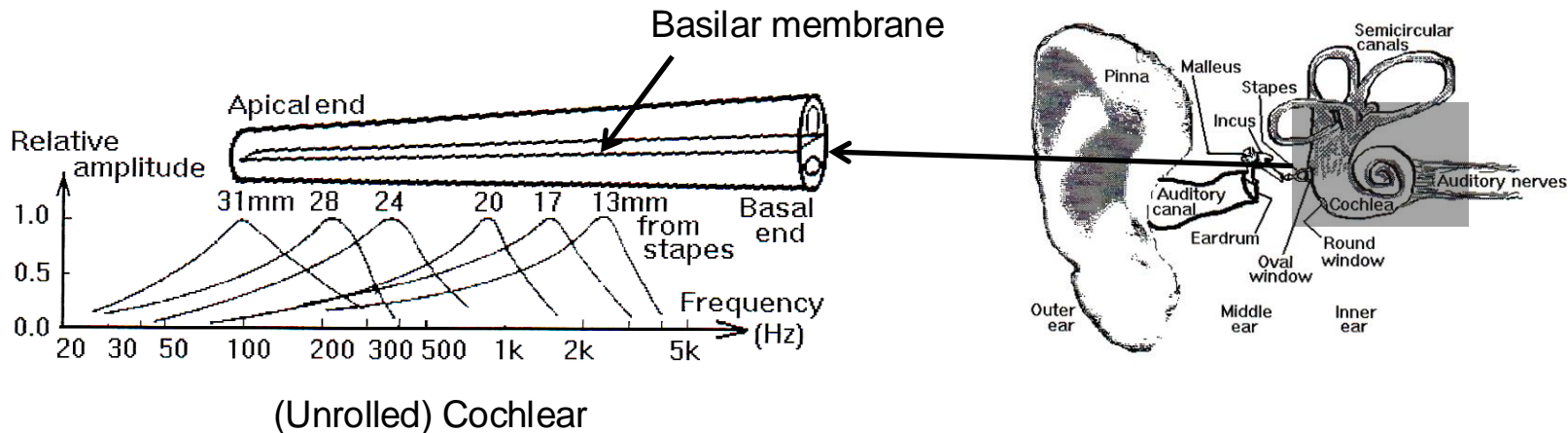
$$\boxed{h_k(n) = w(n) e^{-j\left(\frac{2\pi kn}{N}\right)}}$$

The impulse response of bandpass filters is a windowed complex sinusoid (They all have the same length and the center frequency is equally spaced)

bandpass filter banks

Human Ears

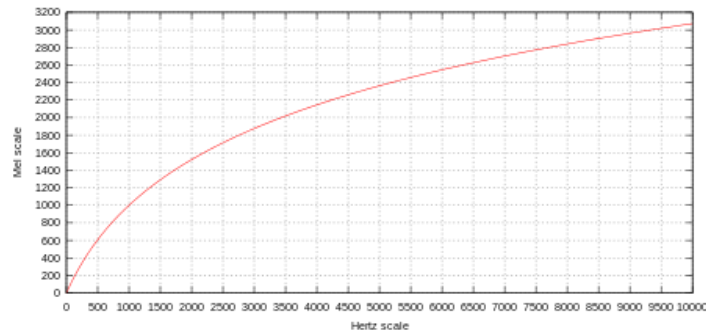
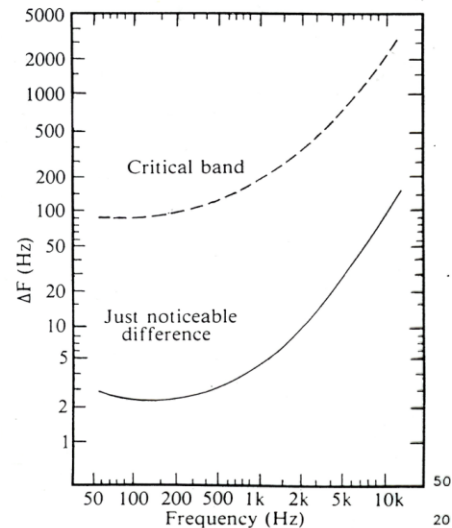
- Human ear is a spectrum analyzer?
 - Cochlea in the inner ear is a bandpass-filter bank
 - Basilar membrane resonates at a different position for a different frequency of the input.
 - The resonance frequency increases in a log scale along the membrane



Human Pitch Perception

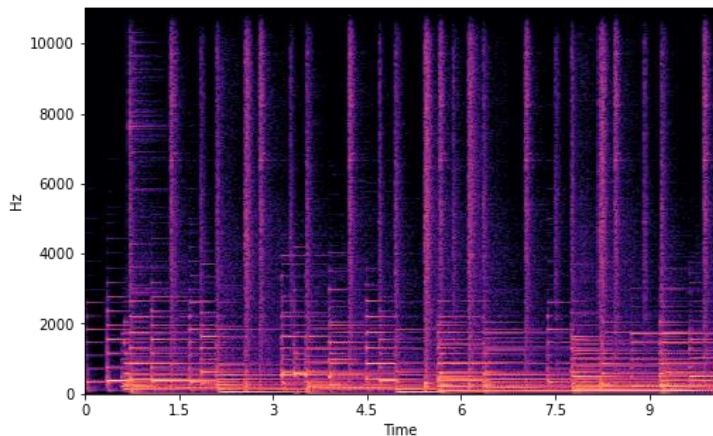
- Pitch Resolution
 - Just noticeable difference (JND) increases as frequency goes up
- Mel scale
 - Most widely used for speech and music analysis
 - Approximate the human pitch resolution based on pitch ratio of tones
 - A log frequency scale:

$$m = 2595 \log_{10}(1 + f / 700) \quad \longrightarrow$$

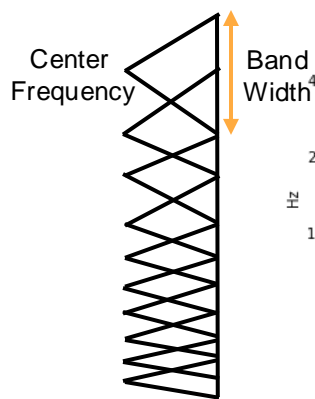


Computing Mel-Spectrogram

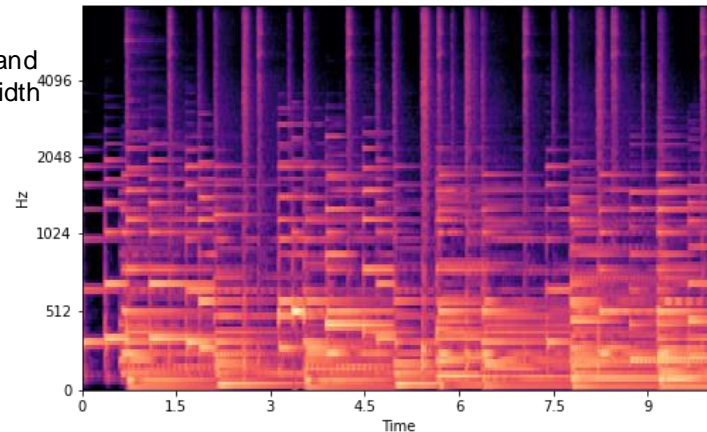
- Mapping linear frequency to mel scale
 - All frequency content within the band of each filter is summarized into a single mel bin
 - High-frequency range content is relatively more squeezed than low-frequency range content



Spectrogram (1024 freq. bins)



Mel-scaled
filter bank



Mel-spectrogram (128 mel bins)

Musical Pitch Scale

- Musical tuning system

- Equal temperament: $1: 2^{1/12}$ ratio for semi-note
- Music note (m) and frequency (f) in Hz

$$m = 12 \log_2 \left(\frac{f}{440} \right) + 69, \quad f = 440 \times 2^{(m-69)/12}$$

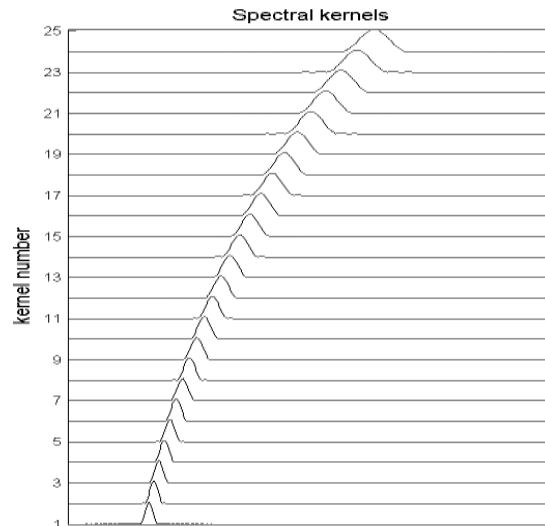
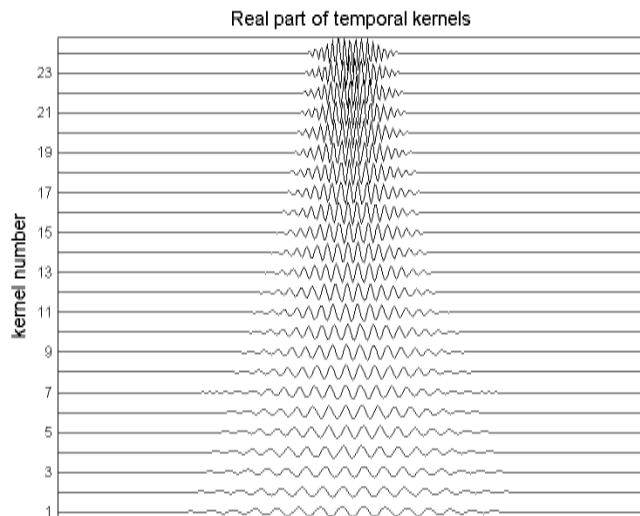
- We can get “musical spectrogram” using filter banks from the musical scale

- However, the resolution in low frequency is not sufficient (e.g., one frequency bin covers several notes)

Frequency	Keyboard	Note name	MIDI number
4186.0			
3951.1		C8	108
3729.3		B7	107
3520.0		A7	106
3322.4		G7	105
3136.0		F7	104
2960.0		E7	103
2793.8		D7	102
2637.0		C7	101
2489.0		B6	99
2349.3		A6	98
2217.5		G6	97
1975.5		F6	96
1864.7		E6	95
1760.0		D6	94
1661.2		C6	93
1480.0		B5	92
1318.5		A5	91
1244.5		G5	90
1174.7		F5	89
1108.7		E5	88
987.77		D5	87
932.33		C5	86
880.00		B4	85
830.61		A4	84
783.99		G4	83
739.99		F4	82
698.46		E4	81
659.26		D4	80
622.25		C4	79
587.33		B3	78
554.37		A3	77
493.88		G3	76
466.16		F3	75
440.00		E3	74
415.30		D3	73
392.00		C3	72
369.99		B2	71
349.23		A2	70
329.63		G2	69
311.13		F2	68
293.67		E2	67
277.18		D2	66
246.94		C2	65
233.08		B1	64
220.00		A1	63
207.65		G1	62
196.00		F1	61
185.00		E1	60
164.81		D1	59
155.56		C1	58
146.83		B0	57
138.59		A0	56
130.81		G0	55
123.47		F0	54
116.54		E0	53
110.00		D0	52
103.83		C0	51
97.999		B-1	50
92.499		A-1	49
87.307		G-1	48
82.407		F-1	47
77.782		E-1	46
73.416		D-1	45
69.296		C-1	44
65.406		B-2	43
61.735		A-2	42
58.270		G-2	41
55.000		F-2	40
51.913		E-2	39
48.999		D-2	38
46.249		C-2	37
43.654		B-1	36
41.203		A-1	35
38.891		G-1	34
36.708		F-1	33
34.648		E-1	32
32.703		D-1	31
30.868		C-1	30
29.135		B0	29
27.500		A0	28

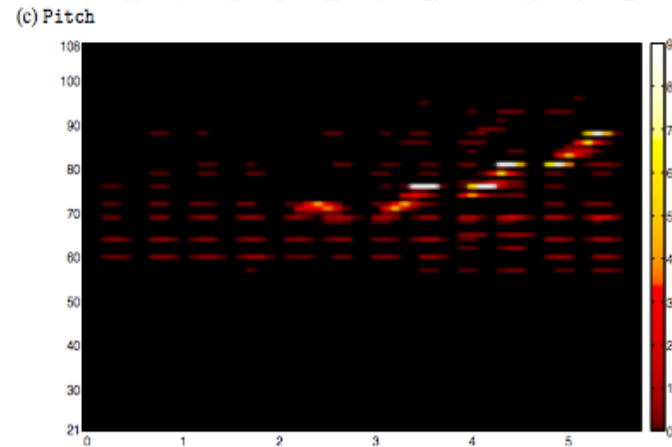
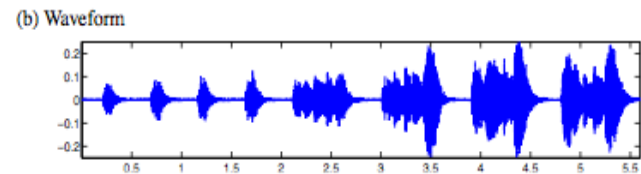
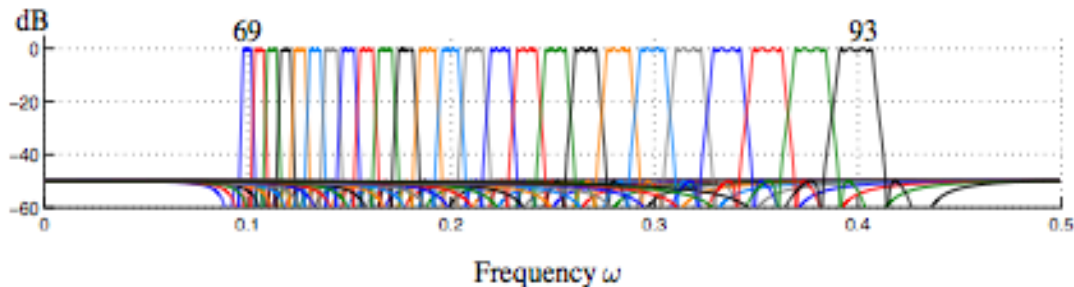
Constant-Q Transform

- Time-frequency representation which uses a set of sinusoidal kernels with log-spaced frequencies
 - As the frequency increases, the length of sinusoidal kernels becomes shorter (bandwidth becomes wider) to have constant Q ($= \text{frequency}/\text{bandwidth}$)



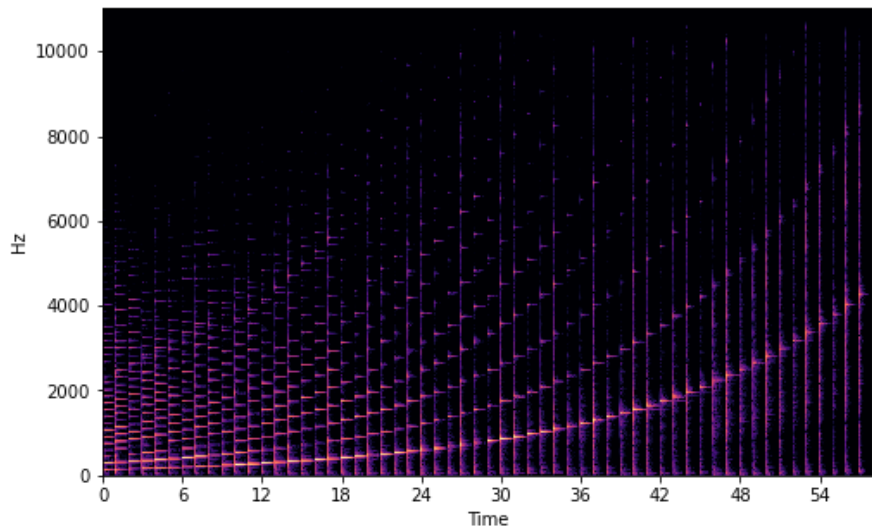
Constant-Q IIR Filter Bank

- Musically designed constant-Q transform
 - 88 IIR bandpass filters
 - The center frequency corresponds to the pitch of each piano note
 - The bandwidth is set to have constant-Q with +/- 25 cent around the center (100 cents = 1 semi-tone)

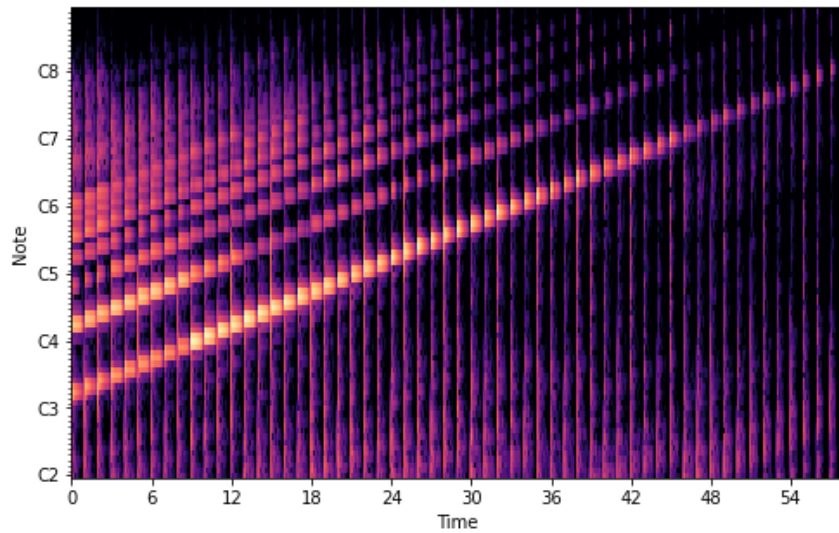


Example: Constant-Q Transform

- Chromatic music scale
 - The harmonics of notes increase linearly in the constant-Q transform



Spectrogram



Constant-Q transform

Neural Network View of Time-frequency Representations

- Convolutional layers followed by a non-linear layer
 - Filter banks \rightarrow filter weights in the convolutional layer
 - Window size \rightarrow filter size in the convolutional layer
 - Hop size \rightarrow stride size in the convolutional layer

