Digital Signal Processing 1

안인규 (Inkyu An)

Speech And Audio Recognition
(오디오 음성인식)

https://mairlab-km.github.io/

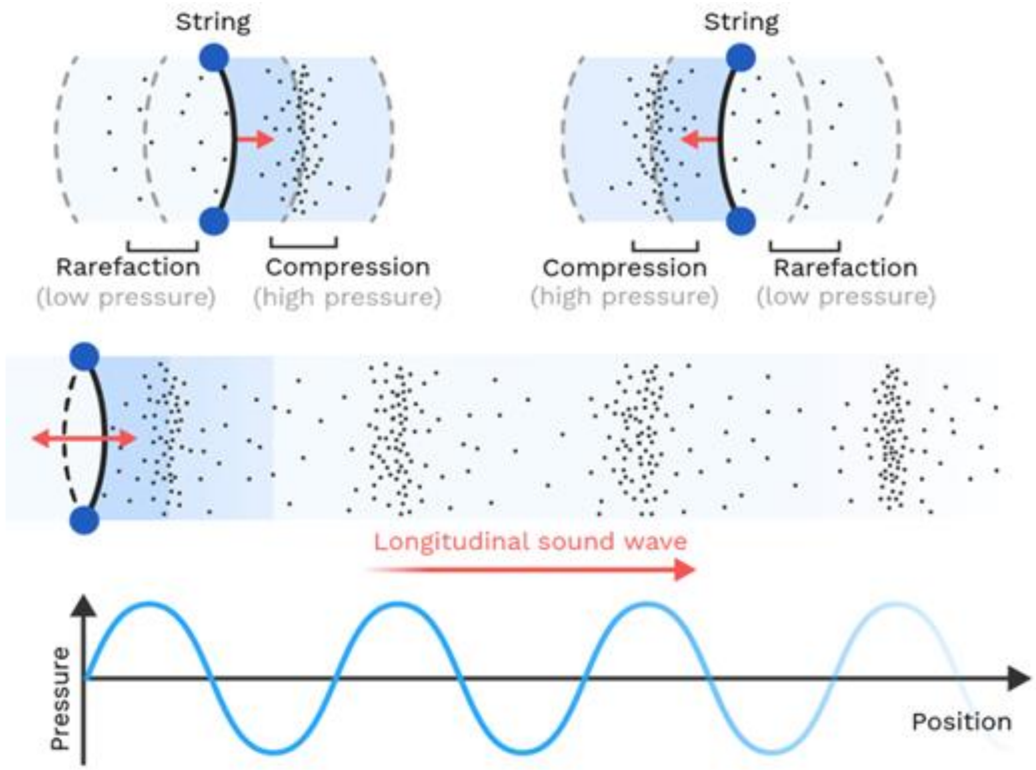
This lecture material refers to
https://github.com/yandexdataschool/speech_course?tab=readme-ov-file and
https://github.com/markovka17/dla

KMU

MAIR Lab

What is Sound?

- The air around us is filled with molecules.
- When you pull a guitar string, it creates a vibration that moves through the molecules in the air.
- The regions of high pressure are compressions and the regions of low pressure as rarefactions.

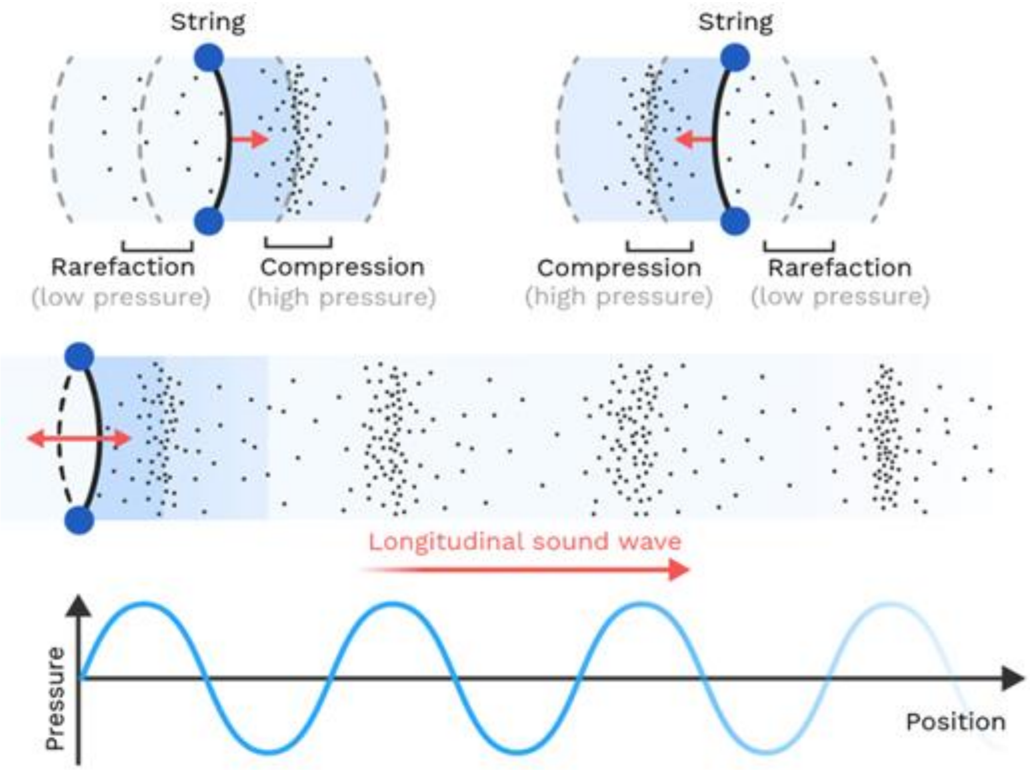


https://pudding.cool/2018/02/waveforms
https://theory.labster.com/sound-waves-dbs

What is Sound?

The microphone detects these variations in pressure.

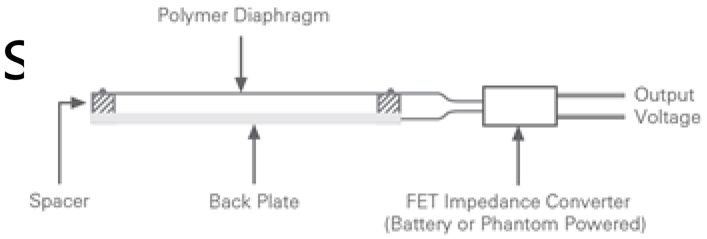
When we plot pressure, **relative to atmospheric** pressure, against the position near the string, we see the familiar sinusoidal waveform.

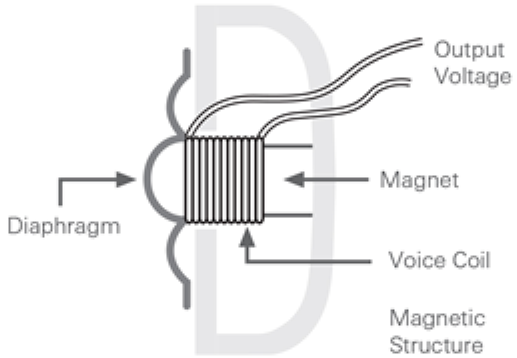


The sinusoidal waveform follows from physics equations, see this slide
https://theory.labster.com/sound-waves-dbs/

Recording

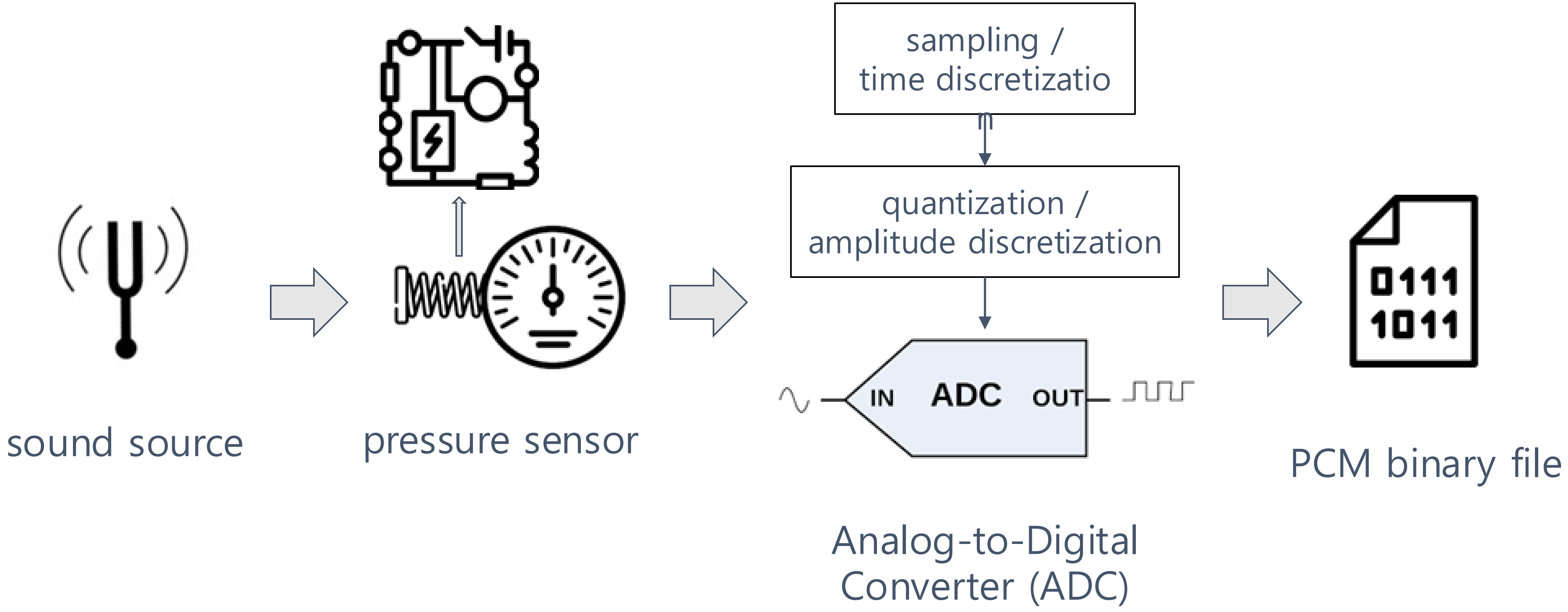
- To record people use microphones
- Microphone picks up these air oscillations in continuous form
- These oscillations are converted into an analog signal and then a dig signal





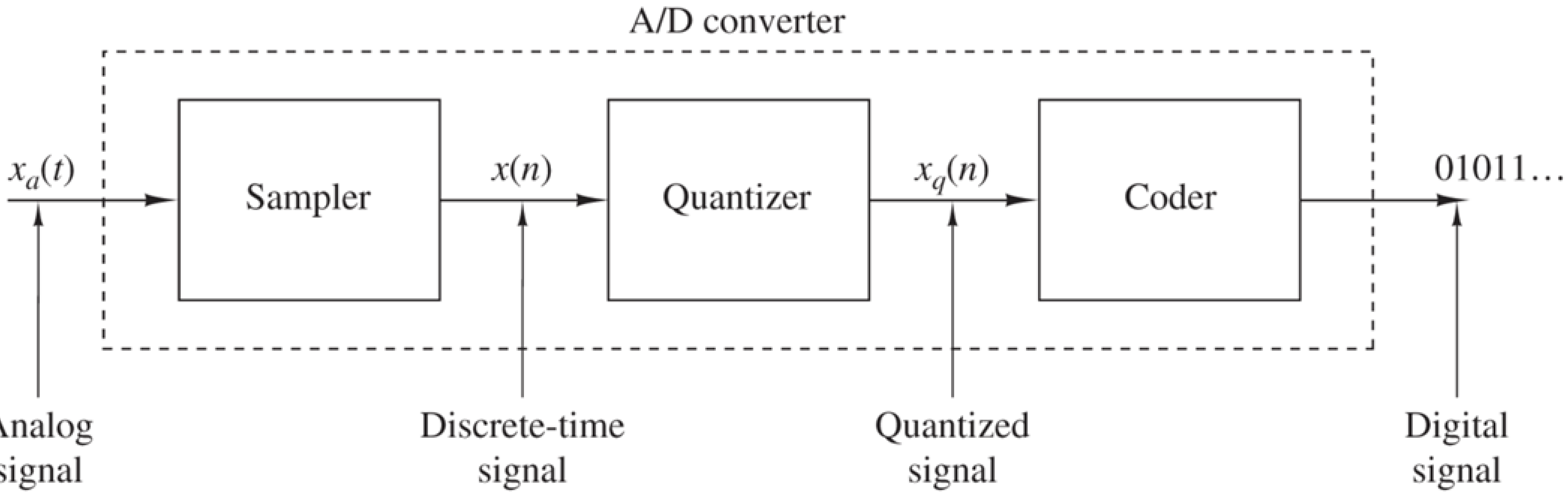
The sinusoidal waveform follows from physics equations, see this slide
https://theory.labster.com/sound-waves-dbs/

Analog and Digital Signals



The sinusoidal waveform follows from physics equations, see [this slide](#)
https://theory.labster.com/sound-waves-dbs/

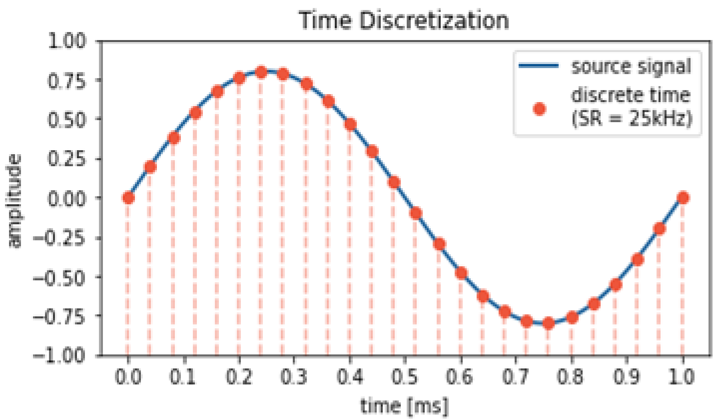
Analog and Digital Signals

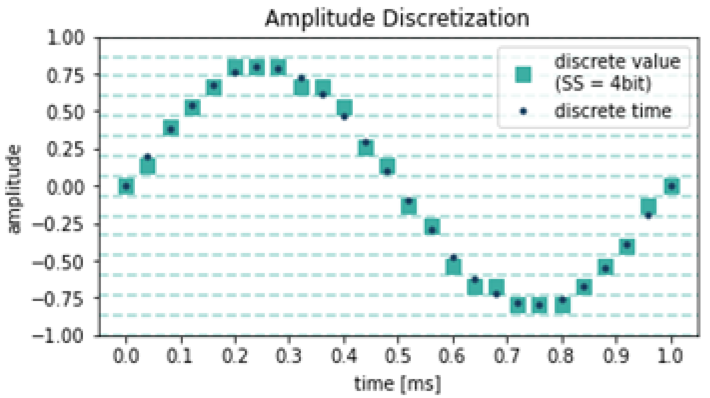


https://is.muni.cz/el/1433/jaro2012/PA190/um/Slides_02.pdf

Waveform as Pulse-Code Modulation (PCM)

- **Time discretization**: we represent the analog signal as a sequence of samples measured at discrete points in time
  - **Sample rate** – number of audio samples per second (8kHz, 22.05kHz, 44.1kHz)
- **Amplitude discretization:** round continuous amplitude to the nearest discrete value
  - **Bit depth** – number of bits per sample (eg. 8, 16, 24, 32 bits)
  - **Bit rate** = bit-depth * sample-rate * audio-channels
- **Number of channels:** number of signals recorded in parallel (e.g., mono vs. stereo)





https://github.com/yandexdataschool/speech_course/tree/2022/week_02

Properties of Waveforms: Intensity

Intensity is defined to be the *energy* (E) per unit *area* (A) and time unit (t) carried by a wave:

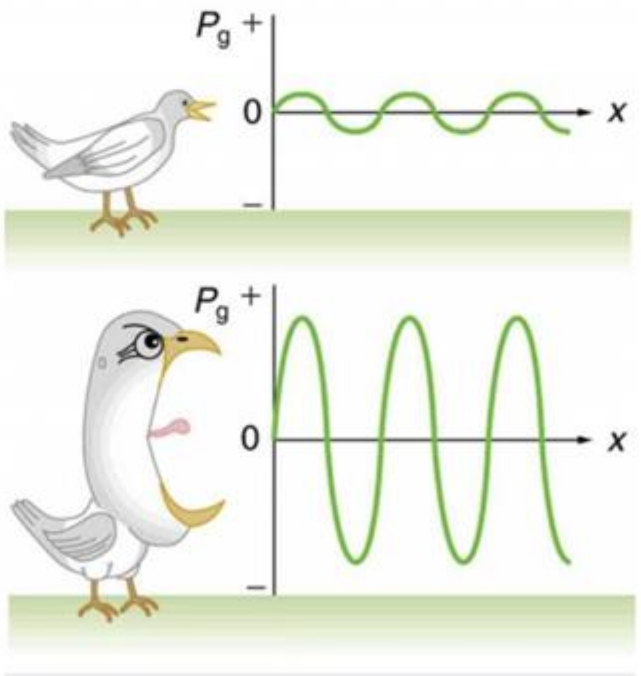
$$I = \frac{E}{tA} = \frac{P}{A}$$ ← *power*

The intensity of a sound wave is proportional to its amplitude squared:

$$I \sim (\Delta p)^2$$

For a discrete-time signal $\{p_n\}_{1:T}$ of length T :

$$I_n \sim p_n^2$$



https://github.com/yandexdataschool/speech_course/tree/2022/week_02

Properties of Waveforms: Loudness

- **Loudness** is intensity measured in decibel
- **bel** reports log10 of ratio between measuring signal and reference signal
- In physics, **decibels** are used instead of bels because 1 bel is too large

$$I_{dB} = 10\log\left(\frac{I}{I_0}\right) = 10\log\left(\frac{p^2}{p_0^2}\right) =$$
$$= 20\log\left(\frac{p}{p_0}\right)$$

https://decibelpro.app/blog/how-many-decibels-does-a-human-speak-normally/

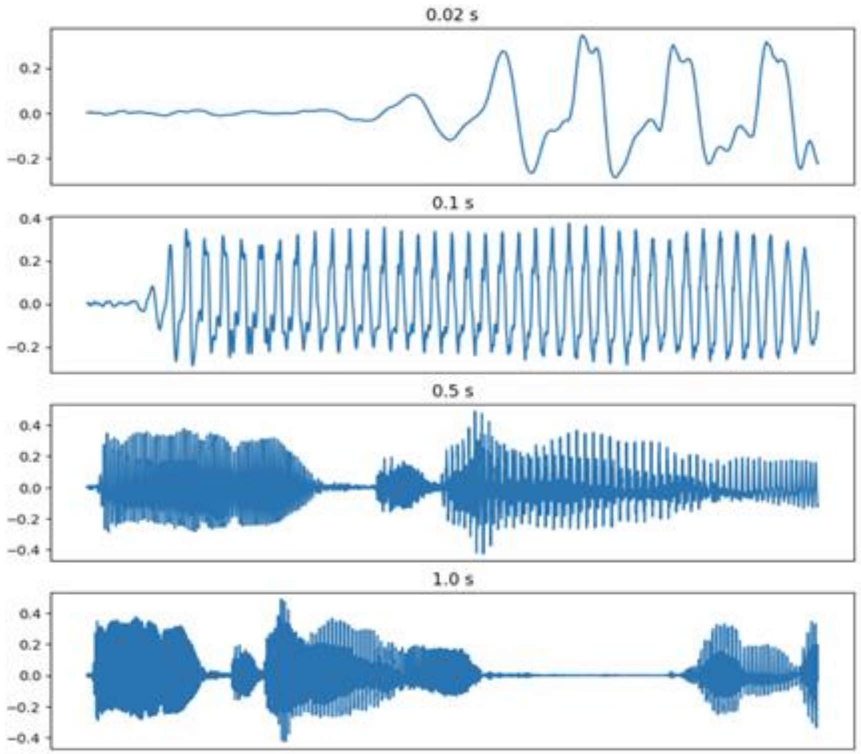
What about audio formats?

- Uncompressed: WAV, AIFF
- Lossless compression: FLAC, ALAC
- Lossy compression: MP3, Opus

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

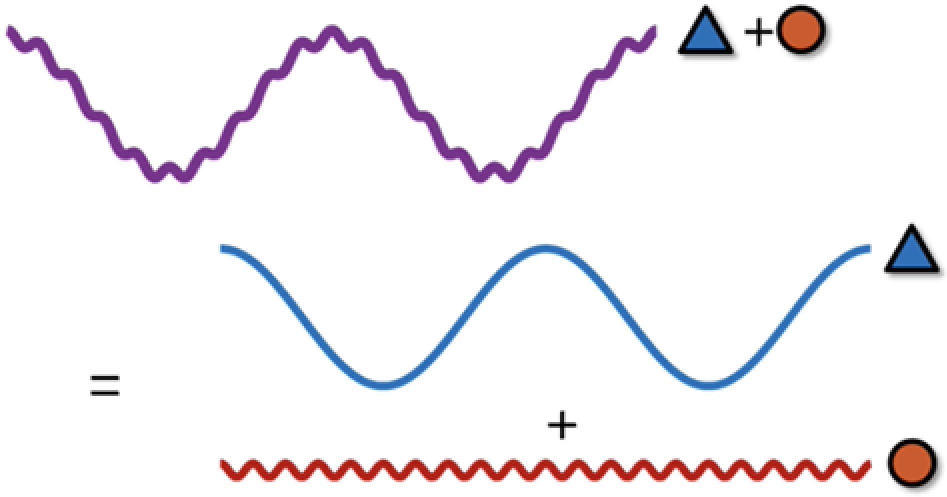
Difficulties in using waveforms

- Waveform is really long (e.g. 44K samples / sec)
- Waveforms provide limited insight into a recording's pitch and speech content
- Can we get a more compact and informative sound representation?



Fourier Analysis

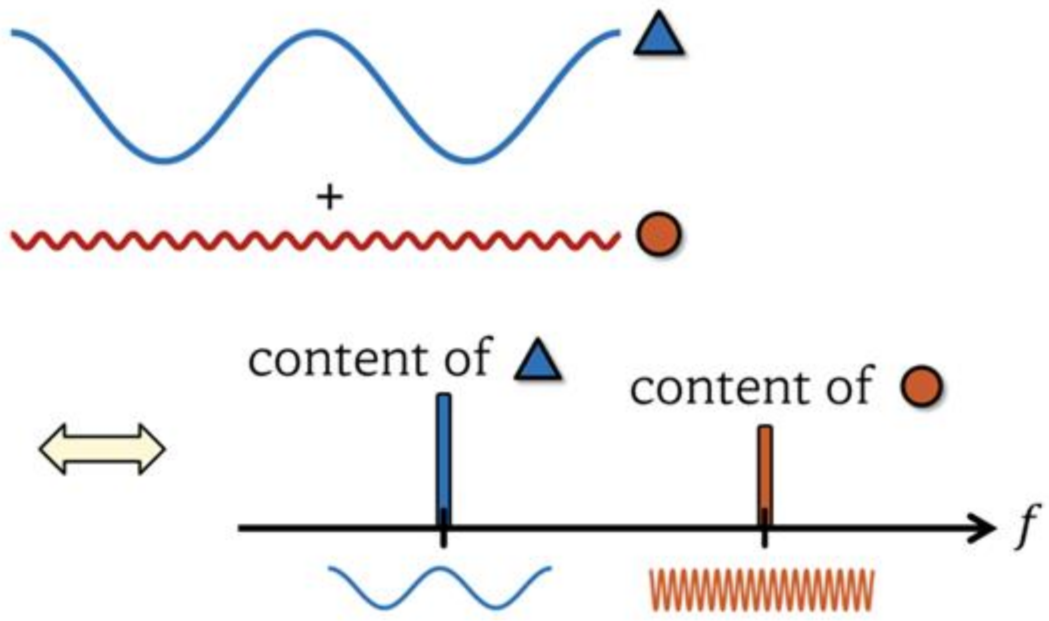
- At its core, Fourier analysis breaks down complex signals into their frequency components.
- It's similar to breaking down a song into its individual musical notes.



https://angeloyeo.github.io/2019/06/23/Fourier_Series_en.html

Fourier Analysis

- The amplitudes of these frequency components appear on the frequency axis rather than the time axis, forming what's called the (frequency) spectrum

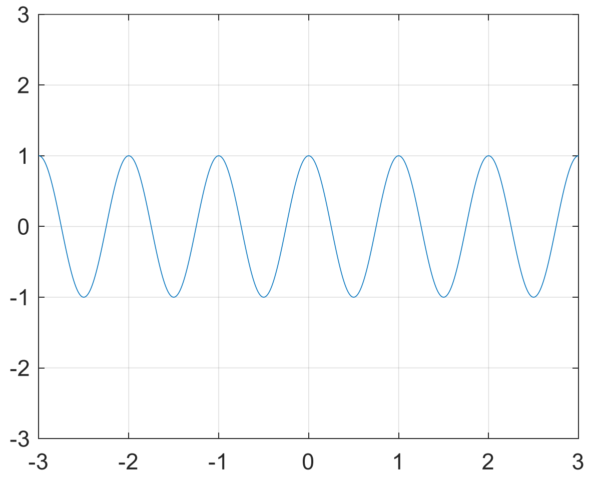


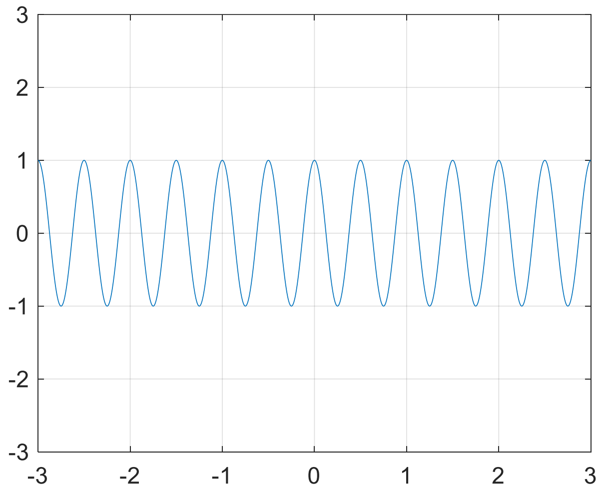
https://angeloyeo.github.io/2019/06/23/Fourier_Series_en.html

Fourier Analysis

- What is Frequency?
  - Period: T
  - Frequency: f=1/T

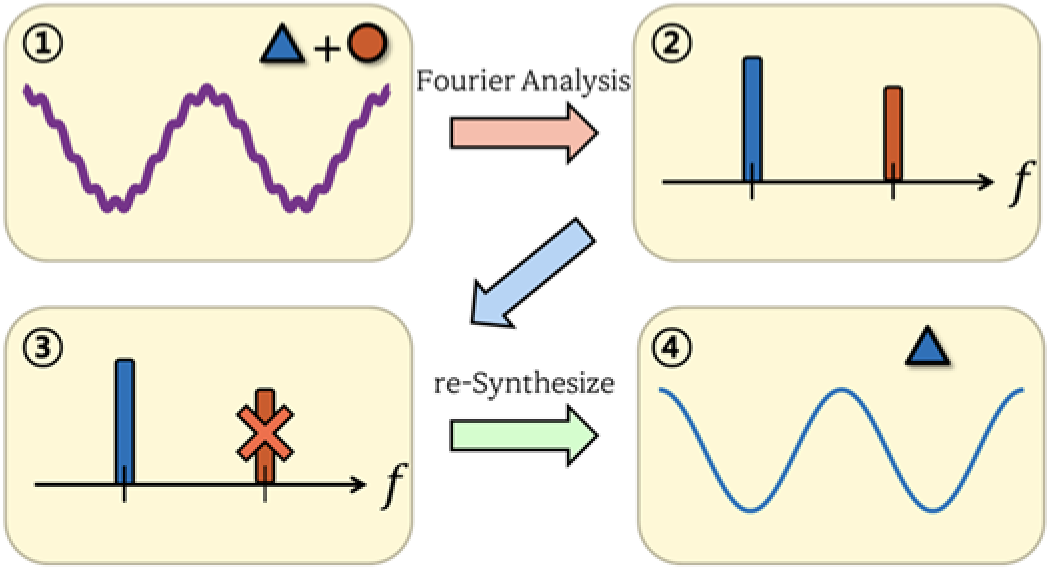
  - Question:
    - What is the frequency of the below graph?





Fourier Analysis

- The usefulness of Fourier analysis:
  - It converts a long sequence of time samples into a compact frequency representation.
  - It enables analysis of frequency components and filtering of unwanted frequencies.

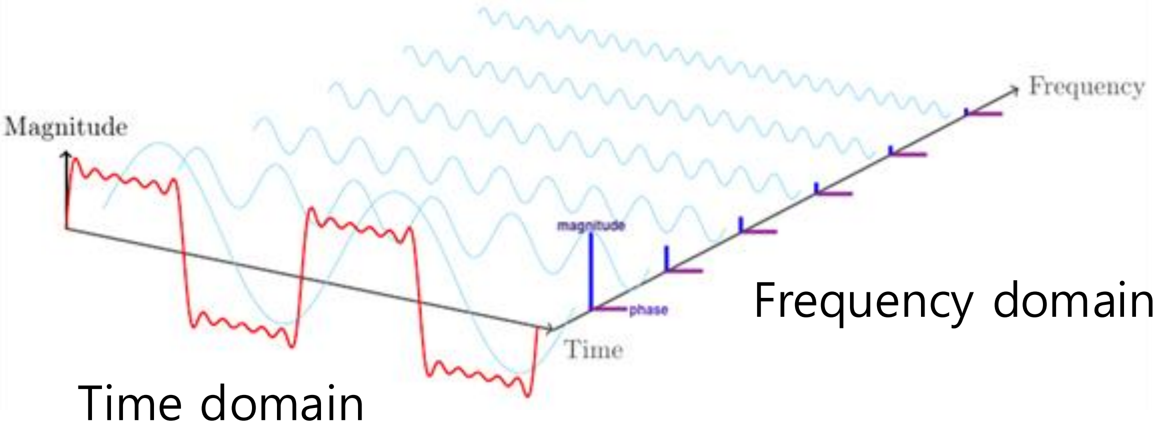


https://angeloyeo.github.io/2019/06/23/Fourier_Series_en.html

Fourier Series

- Any absolutely integrable periodic function $x(t)$ with period P can be represented as

$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cos\left(2\pi \frac{n}{P} t - \phi_n\right)$$

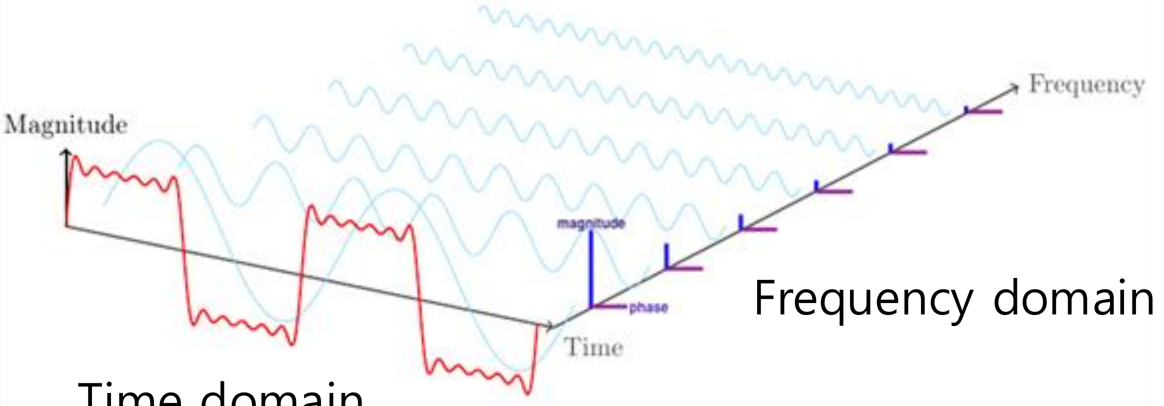


https://angeloyeo.github.io/2019/06/23/Fourier_Series_en.html

Fourier Series

- Any absolutely integrable periodic function $x(t)$ with period P can be represented as

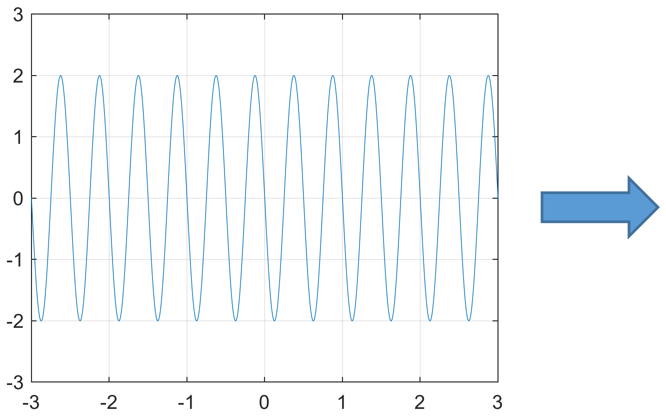
$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} \overbrace{A_n}^{\text{Amplitude}} \cos\left(2\pi \underbrace{\frac{n}{P}}_{\text{Frequency}} t - \overbrace{\phi_n}^{\text{Phase}}\right)$$



Time domain

Frequency domain

- What is the amplitude and the phase?



- Amplitude?
- Phase?

https://angeloyeo.github.io/2019/06/23/Fourier_Series_en.html

Fourier Series

- Any absolutely integrable periodic function $x(t)$ with period P can be represented as

$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cos\left(2\pi \frac{n}{P} t - \phi_n\right)$$

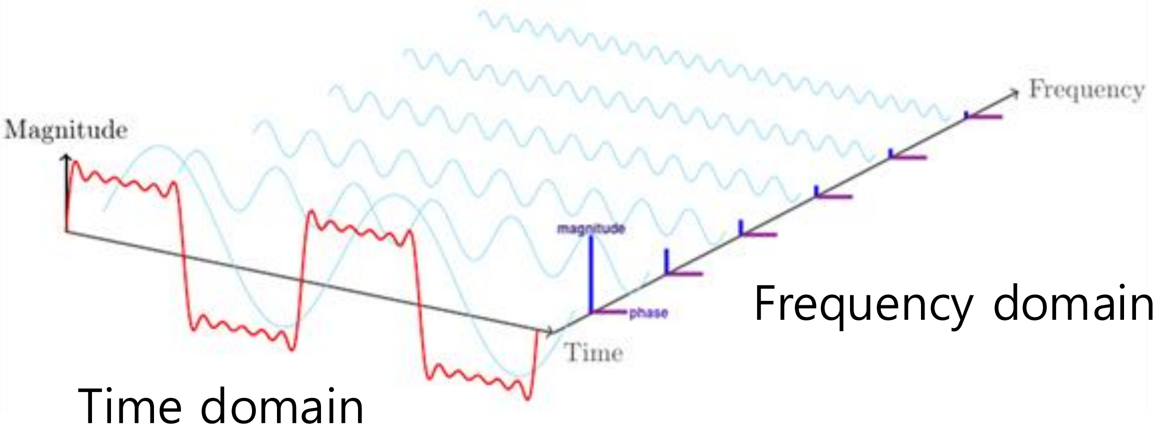
Fourier Series: Amplitude-phase form

$$= a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right)\right)$$

Fourier Series: sine-cosine form

$$= \sum_{n=-\infty}^{\infty} c_n e^{2\pi i \frac{n}{P} t}$$

Fourier Series: exponential form



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

Fourier Series

- Any absolutely integrable periodic function $x(t)$ with period P can be represented as

$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cos\left(2\pi \frac{n}{P} t - \phi_n\right)$$

Fourier Series: Amplitude-phase form

Use formular for cosine of the difference
$\cos(A - B) = \cos A \cos B + \sin A \sin B$

$$= a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right)\right)$$

Fourier Series: sine-cosine form

Use Euler's formula

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}$$

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i}$$

$$= \sum_{n=-\infty}^{\infty} c_n e^{2\pi i \frac{n}{P} t}$$

Fourier Series: exponential form

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

Fourier Series

Exponential form coefficients

$$c_n = \begin{cases} \frac{1}{2}(a_n - ib_n) & \text{if } n > 0, \\ a_n & \text{if } n = 0, \\ \frac{1}{2}(a_{-n} + ib_{-n}) & \text{if } n < 0, \end{cases}$$

- Any absolutely integrable periodic function $x(t)$ with period P can be represented as

$$x(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cos\left(2\pi \frac{n}{P} t - \phi_n\right)$$

Fourier Series: Amplitude-phase form

$$= a_0 + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2\pi n t}{P}\right) + b_n \sin\left(\frac{2\pi n t}{P}\right)\right)$$

Fourier Series: sine-cosine form

$$= \sum_{n=-\infty}^{\infty} c_n e^{2\pi i \frac{n}{P} t}$$

Fourier Series: exponential form

Use Euler's formula

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}, \qquad \sin x = \frac{e^{ix} - e^{-ix}}{2i}$$

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

Fourier Series: Exponential form

- Fourier Series & Fourier Coefficient

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i \frac{n}{P} t}$$

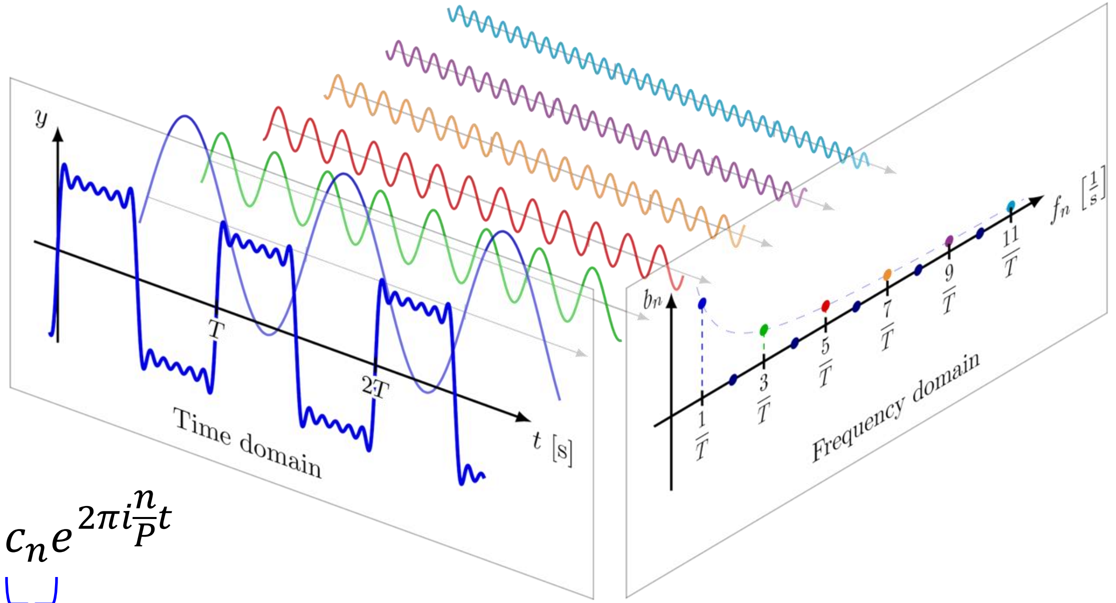
$$c_n = \frac{1}{P} \int_0^P x(t) e^{-2\pi i \frac{n}{P} t} dt$$

Fourier coefficient

- The set of Fourier coefficients is also called the **spectrum** of $\boldsymbol{x(t)}$
- $\boldsymbol{c_n}$ are **complex** numbers

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

Fourier Transform



- Fourier Series & Fourier Coefficient

P: the period

$$c_n = \frac{1}{P}\int_0^P x(t)e^{-2\pi i\frac{n}{P}t}dt$$

Fourier coefficient

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{2\pi i\frac{n}{P}t}$$

- Let's assume $P \to \infty$ and $0 \to -\infty$

Frequency

Original signal

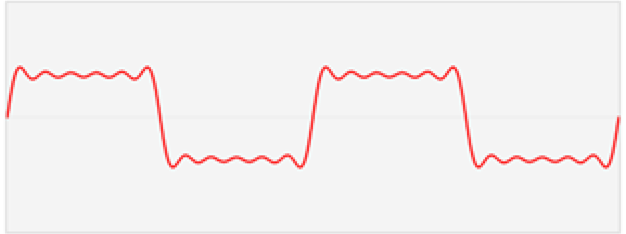
$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-2\pi i f t}dt$$

Fourier transform

- Fourier Transform!
  - A mathematical formular that allows us to decompose any signal into its individual **frequencies** and the frequency's **amplitude**

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

Fourier Transform



- Fourier Transform!
  - A mathematical formular that allows us to decompose any signal into its individual **frequencies** and the frequency's **amplitude**

Frequency — Original signal

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-2\pi i f t} dt$$

Fourier transform

- There is also the inverse Fourier Transform:

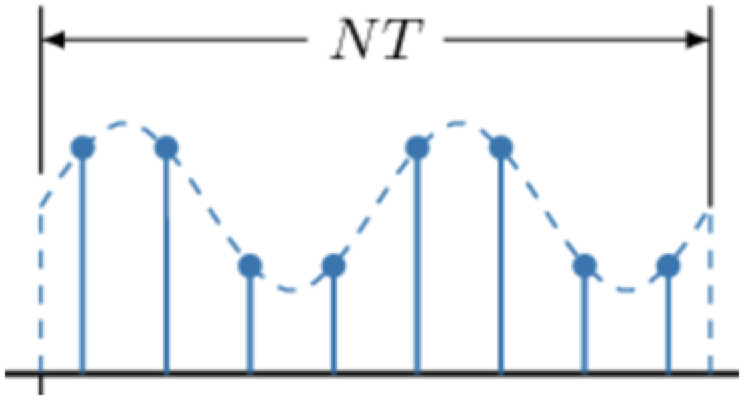
Original signal — Frequency

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{2\pi i f t} dt$$

Inverse Fourier transform

Discrete-Time Fourier Series (DTFS)

- In practice our time signal is time-limited and contains N non-zero samples taken with a sampling period T seconds.
- Let's explore what happens when we try to find its Fourier coefficients.



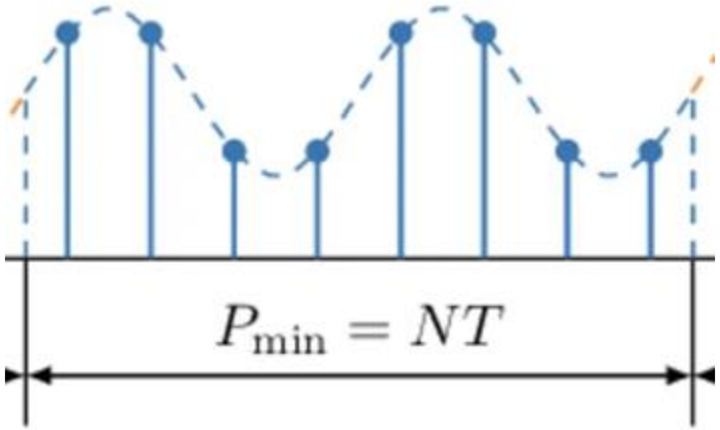
https://ru.dsplib.org/content/dft/dft.html

Discrete-Time Fourier Series (DTFS)

$$c_k = \frac{1}{P}\int_0^P x(t)e^{-2\pi i\frac{n}{P}t}dt$$

- The integral "turns" into a sum over a discrete set of values
- $P = NT$

$$X[k] = \frac{1}{NT}\sum_{n=0}^{N-1} x(nT)e^{-2\pi i\frac{k}{NT}nT}$$



https://ru.dsplib.org/content/dft/dft.html

Discrete-Time Fourier Series (DTFS)

- Disappears! Now the basis functions are defined by N:

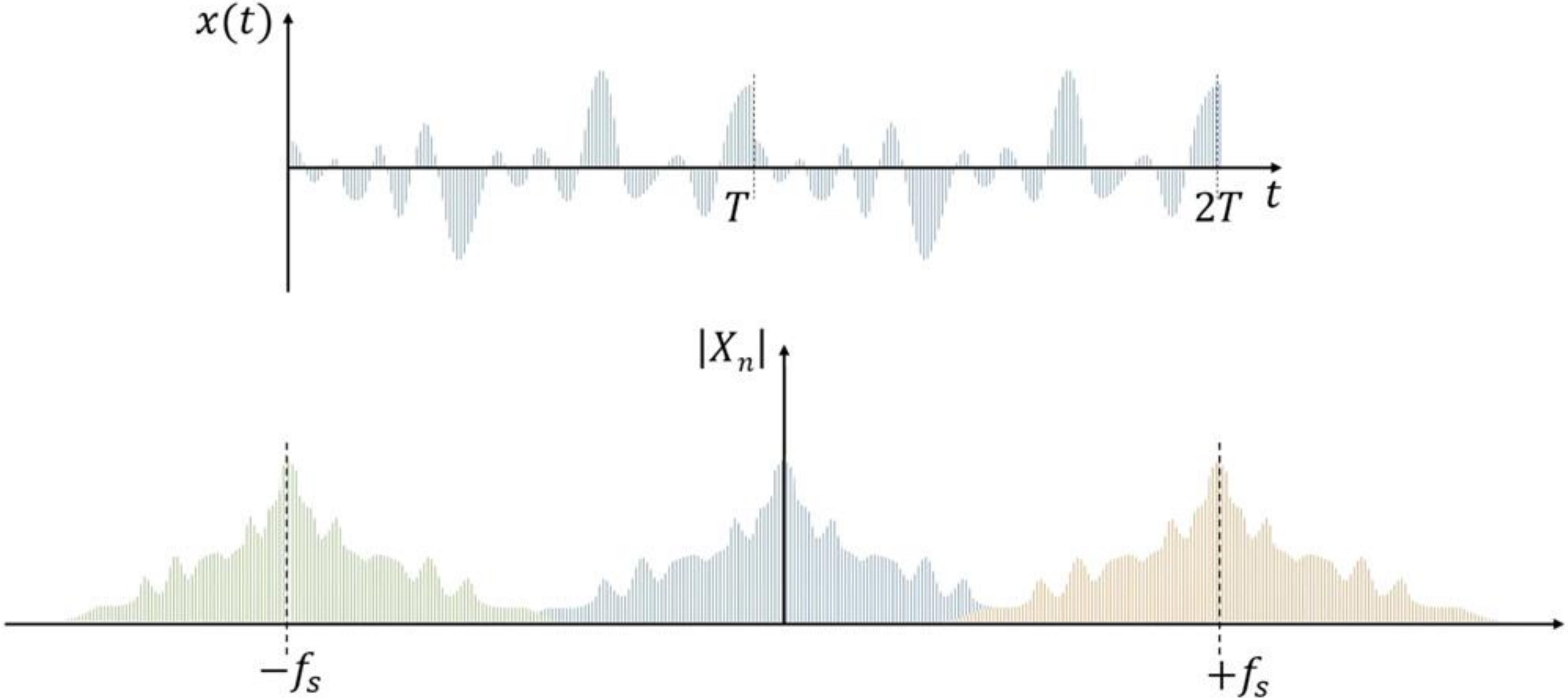
$$c_k = \frac{1}{NT}\sum_{n=0}^{N-1} x(nT)e^{-2\pi i\frac{k}{NT}nT} = \frac{1}{NT}\sum_{n=0}^{N-1} x(nT)e^{-2\pi i\frac{k}{N}n}$$

- The expression above is valid for any integer k but it is *periodic* and repeats every N samples:

$$c_{k+N} = \frac{1}{NT}\sum_{n=0}^{N-1} x(nT)e^{-2\pi i\frac{k+N}{N}n} = \frac{1}{NT}\sum_{n=0}^{N-1} x(nT)e^{-2\pi i\frac{k}{N}n} \cdot \underbrace{e^{-2\pi i n}}_{=1} = c_k$$

https://ru.dsplib.org/content/dft/dft.html

Discrete-Time Fourier Series (DTFS)



https://www.youtube.com/watch?v=nreiTseFZQ0

Discrete Fourier Transform (DFT)

- If we operate only with indices of the input signal and spectral samples (setting $T = 1$), we obtain the Discrete Fourier Transform expression:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-2\pi i\frac{k}{N}n}$$

- We can also write out the expression for the Inverse Discrete Fourier Transform (we'll skip the complete derivation):

$$x[n] = \sum_{k=0}^{N-1} X[k]\, e^{2\pi i\frac{n}{N}k}$$

Digital Signal Processing 2

안인규 (Inkyu An)

Speech And Audio Recognition
(오디오 음성인식)

https://mairlab-km.github.io/

This lecture material refers to
https://github.com/yandexdataschool/speech_course?tab=readme-ov-file and
https://github.com/markovka17/dla

Discrete Fourier Transform (DFT)

- If we operate only with indices of the input signal and spectral samples (setting $T = 1$), we obtain the Discrete Fourier Transform expression:

$$X[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-2\pi i \frac{k}{N} n}$$

- We can also write out the expression for the Inverse Discrete Fourier Transform (we'll skip the complete derivation):

$$x[n] = \sum_{k=0}^{N-1} X[k]\, e^{2\pi i \frac{n}{N} k}$$

frequency, phase, amplitude

$$X[k] = a_k e^{-i\phi_k} = a_k(\cos(\phi_k) - i\sin(\phi_k))$$

Discrete Fourier Transform (DFT)

- $X = M \cdot x$, where $x = \{x[0], \cdots, x[N-1]\}$ and $X = \{X[0], \cdots, X[N-1]\}$

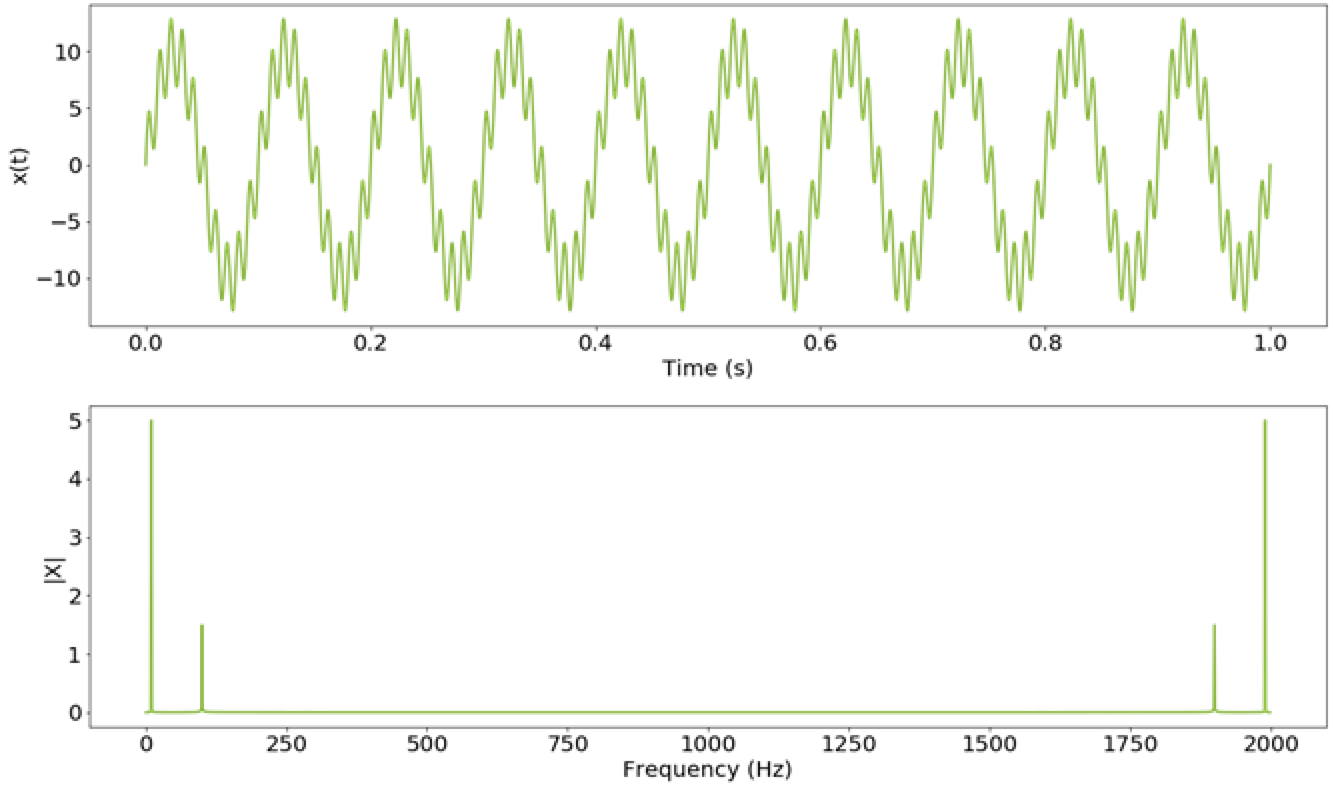
$$M_{mn} = \exp\left(-2\pi i \frac{(m-1)(n-1)}{N}\right)$$

$$\mathbf{M} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-\frac{2\pi i}{N}} & e^{-\frac{4\pi i}{N}} & e^{-\frac{6\pi i}{N}} & \cdots & e^{-\frac{2\pi i}{N}(N-1)} \\ 1 & e^{-\frac{4\pi i}{N}} & e^{-\frac{8\pi i}{N}} & e^{-\frac{12\pi i}{N}} & \cdots & e^{-\frac{2\pi i}{N}2(N-1)} \\ 1 & e^{-\frac{6\pi i}{N}} & e^{-\frac{12\pi i}{N}} & e^{-\frac{18\pi i}{N}} & \cdots & e^{-\frac{2\pi i}{N}3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{2\pi i}{N}(N-1)} & e^{-\frac{2\pi i}{N}2(N-1)} & e^{-\frac{2\pi i}{N}3(N-1)} & \cdots & e^{-\frac{2\pi i}{N}(N-1)^2} \end{pmatrix}$$

We can compute it with FFT, which is extremely fast (theoretically O(NlogN) for signal of size N)

Discrete Fourier Transform (DFT)

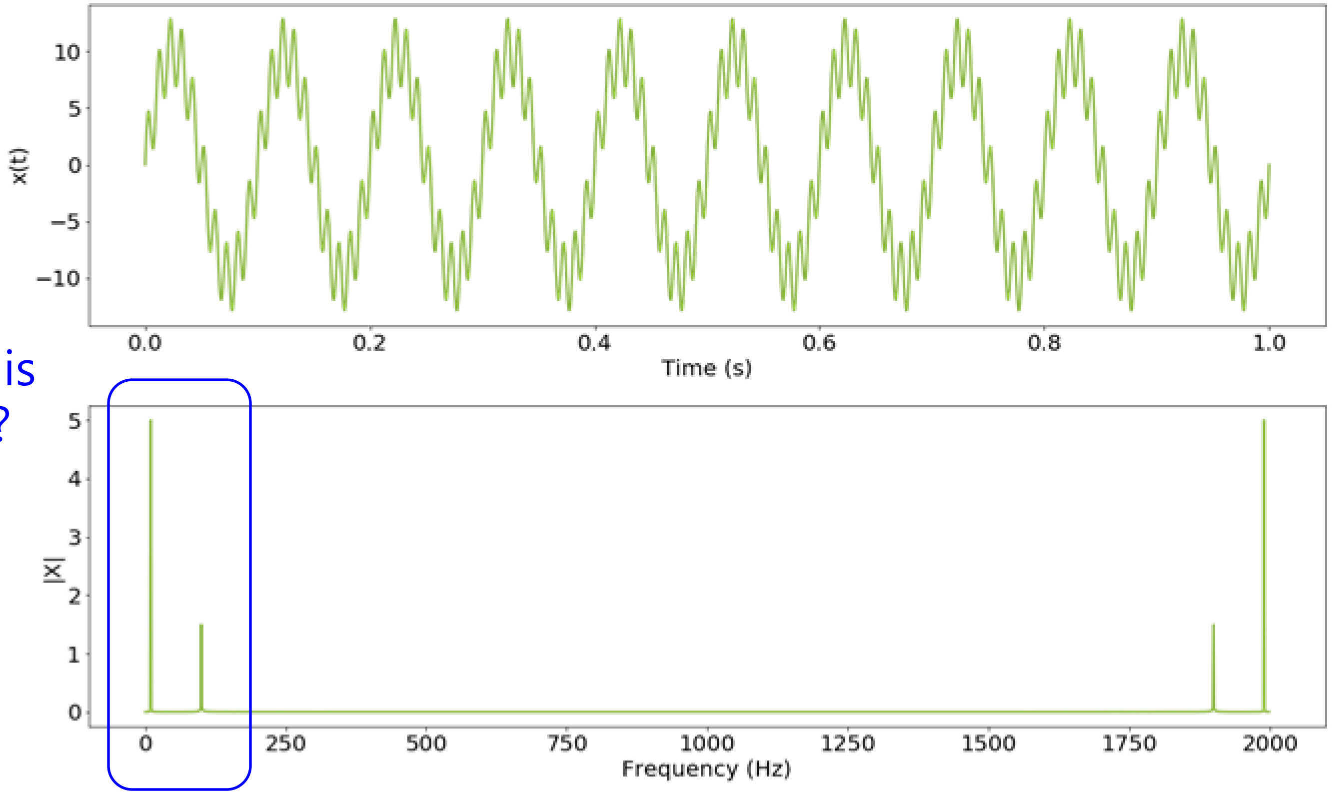
- Example of DFT:
  - $f(t) = 10\sin(2\pi 10t) + 3\sin(2\pi 100t)$



https://www.gaussianwaves.com/2015/11/interpreting-fft-results-obtaining-magnitude-and-phase-information/

Discrete Fourier Transform (DFT)

- Example of DFT:
  - $f(t) = 10\sin(2\pi 10t) + 3\sin(2\pi 100t)$

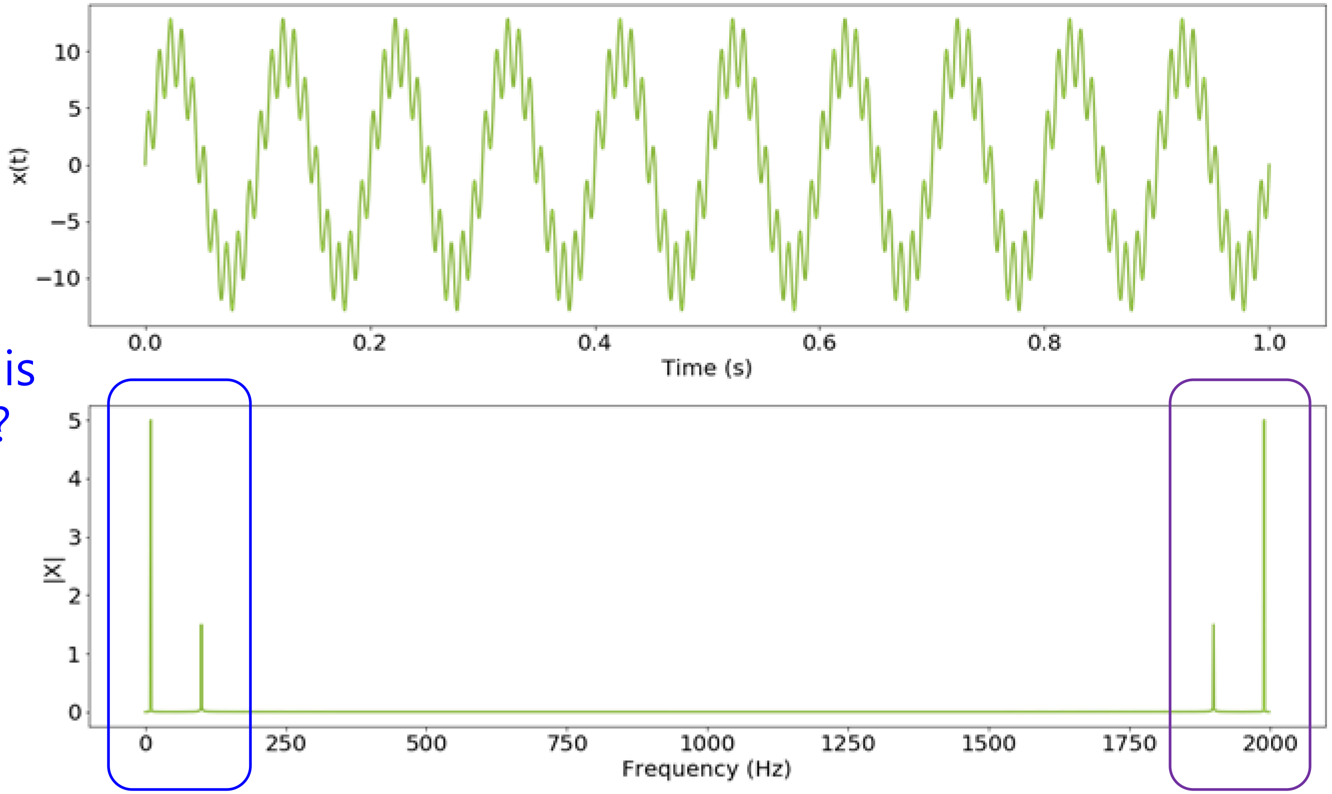


1. What does this graph indicate?

https://www.gaussianwaves.com/2015/11/interpreting-fft-results-obtaining-magnitude-and-phase-information/

Discrete Fourier Transform (DFT)

- Example of DFT:
  - $f(t) = 10\sin(2\pi 10t) + 3\sin(2\pi 100t)$



1. What does this graph indicate?

2. Why does this part keep repeating?

https://www.gaussianwaves.com/2015/11/interpreting-fft-results-obtaining-magnitude-and-phase-information/

Discrete Fourier Transform (DFT)

- Example of DFT:
  - $f(t) = 10\sin(2\pi 10t) + 3\sin(2\pi 100t)$

$$X_m = \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi\frac{m}{N}n\right)$$

$$X_{N-m} = \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi\frac{N-m}{N}n\right)$$

$$= \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi n + j2\pi\frac{m}{N}n\right)$$

$$= \sum_{n=0}^{N-1} x_n \exp\left(j2\pi\frac{m}{N}n\right)$$

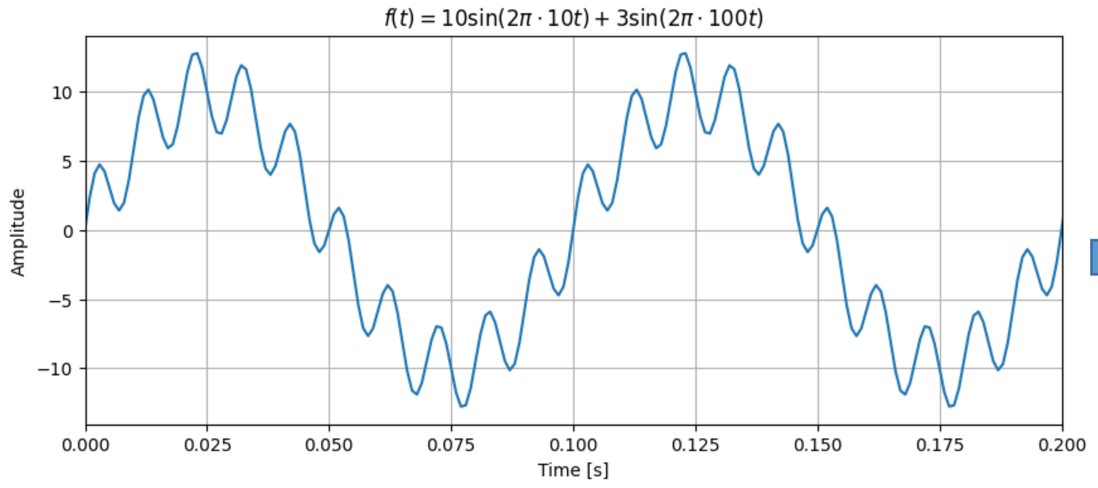
$$= (X_m)^*$$

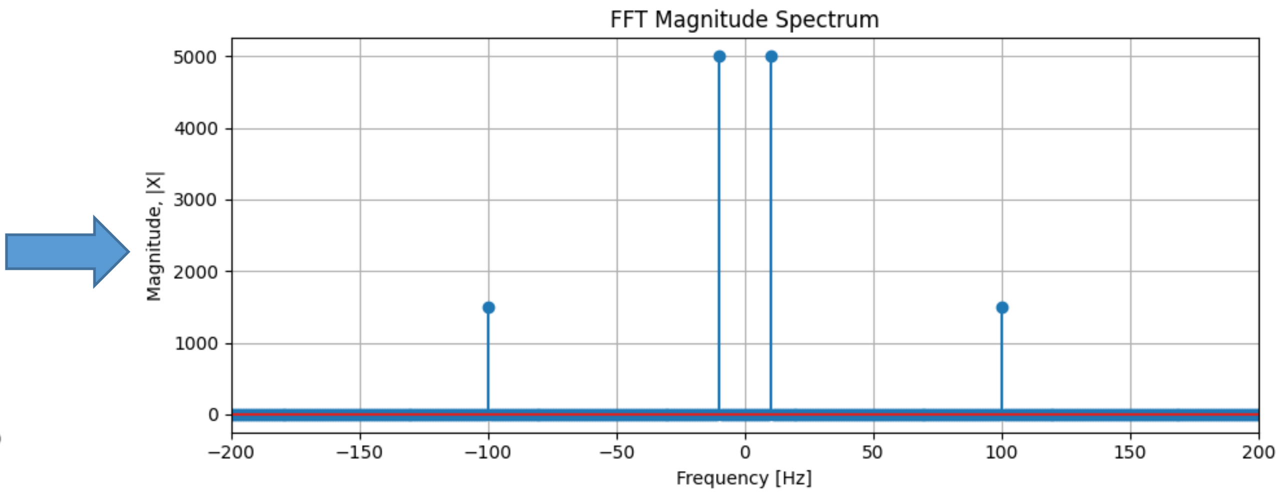
$$X_m = a_k(\cos(\phi_k) - i\sin(\phi_k))$$

$$(X_m)^* = a_k(\cos(\phi_k) + i\sin(\phi_k))$$

Practice: Discrete Fourier Transform (DFT)

- Colab Practice!
- Example of DFT:
  - $f(t) = 10\sin(2\pi 10t) + 3\sin(2\pi 100t)$



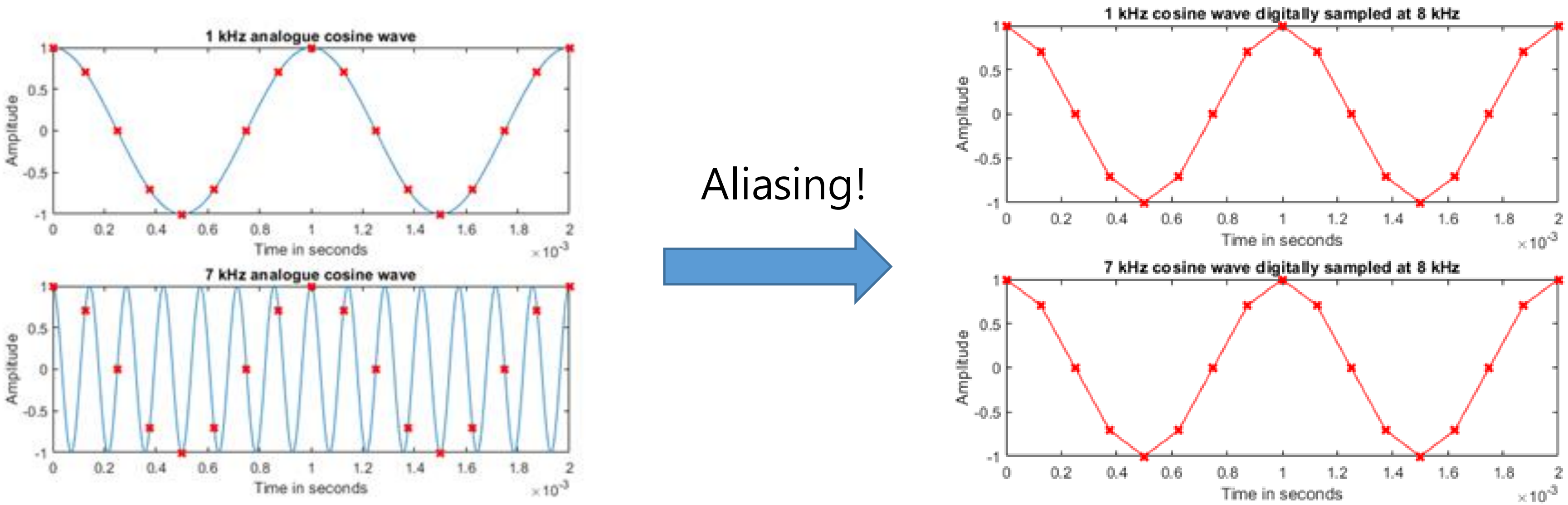


Discrete Fourier Transform (DFT)

- Nyquist(나이퀴스트) Theorem:
  - If a function *f(t)* contains no frequencies higher than *B* hertz, it is completely determined by giving its ordinates at series of points spaced 1/2B (**Nyquist frequency**) seconds apart (함수 $f(t)$가 B 헤르츠보다 높은 주파수를 포함하지 않는다면, 1/2B초 간격으로 주어진 함수값 만으로도 완전히 결정할 수 있다.)
  - E.g., If signal contains frequency *100 Hz*, you need to sample at 200 Hz at least to observe this frequency component
  - DFT of a segment of a signal with sample rate B, will produce amplitudes for nfft evenly spread frequencies in range [-B/2; B/2]

Discrete Fourier Transform (DFT)

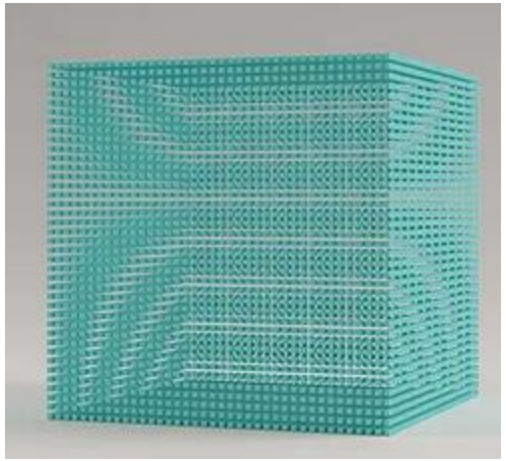
- Nyquist(나이퀴스트) Theorem:
  - If a function *f(t)* contains no frequencies higher than *B* hertz, it is completely determined by giving its ordinates at series of points spaced 1/2B (**Nyquist frequency**) seconds apart (함수 $f(t)$가 B 헤르츠보다 높은 주파수를 포함하지 않는다면, 1/2B초 간격으로 주어진 함수값 만으로도 완전히 결정할 수 있다.)
  - E.g., <u>If signal contains frequency *100 Hz*, you need to sample at 200 Hz at least to observe this frequency component</u>
  - DFT of a segment of a signal with sample rate B, will produce amplitudes for nfft evenly spread frequencies in range [-B/2; B/2]



Discrete Fourier Transform (DFT)

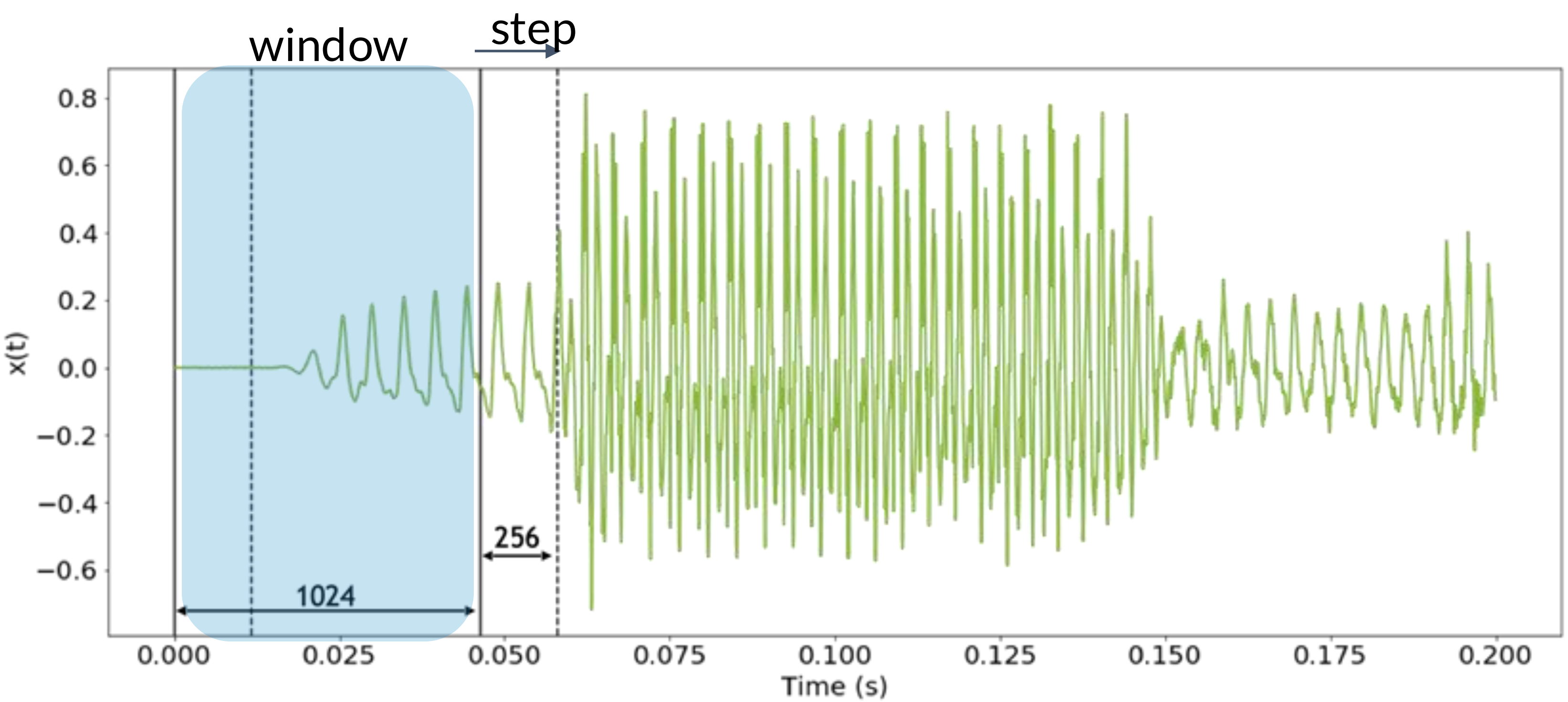
https://en.wikipedia.org/wiki/Moir%C3%A9_pattern#
https://www.adobe.com/creativecloud/photography/discover/anti-aliasing.html

- Nyquist(나이퀴스트) Theorem:
  - If a function *f(t)* contains no frequencies higher than *B* hertz, it is completely determined by giving its ordinates at series of points spaced 1/2B (**Nyquist frequency**) seconds apart (함수 $f(t)$가 B 헤르츠보다 높은 주파수를 포함하지 않는다면, 1/2B초 간격으로 주어진 함수값 만으로도 완전히 결정할 수 있다.)
  - E.g., <u>If signal contains frequency *100 Hz*, you need to sample at 200 Hz at least to observe this frequency component</u>
  - DFT of a segment of a signal with sample rate B, will produce amplitudes for nfft evenly spread frequencies in range [-B/2; B/2]



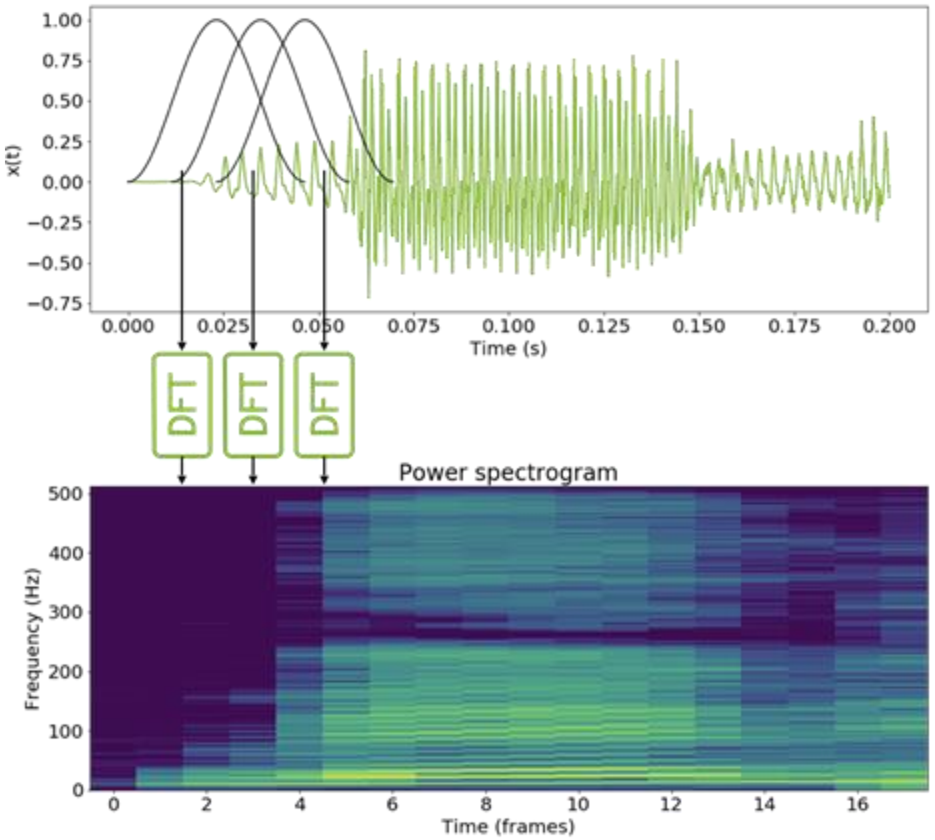
Aliasing effect in images

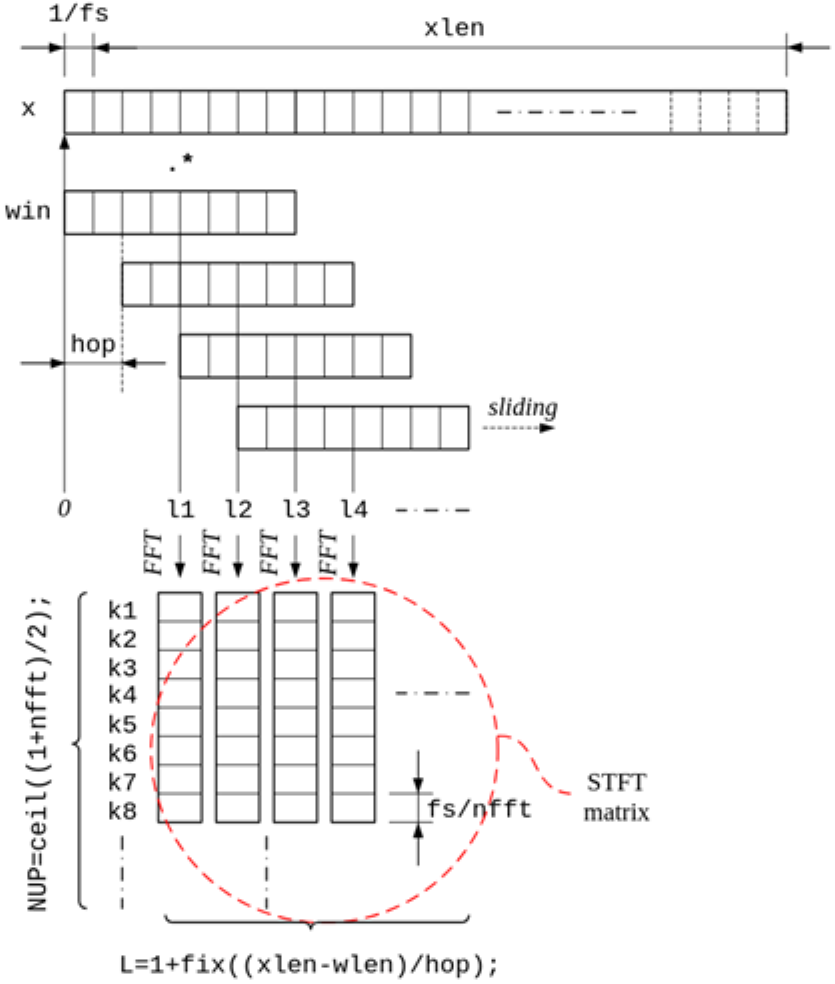
Shor-Time Fourier Transform



Shor-Time Fourier Transform

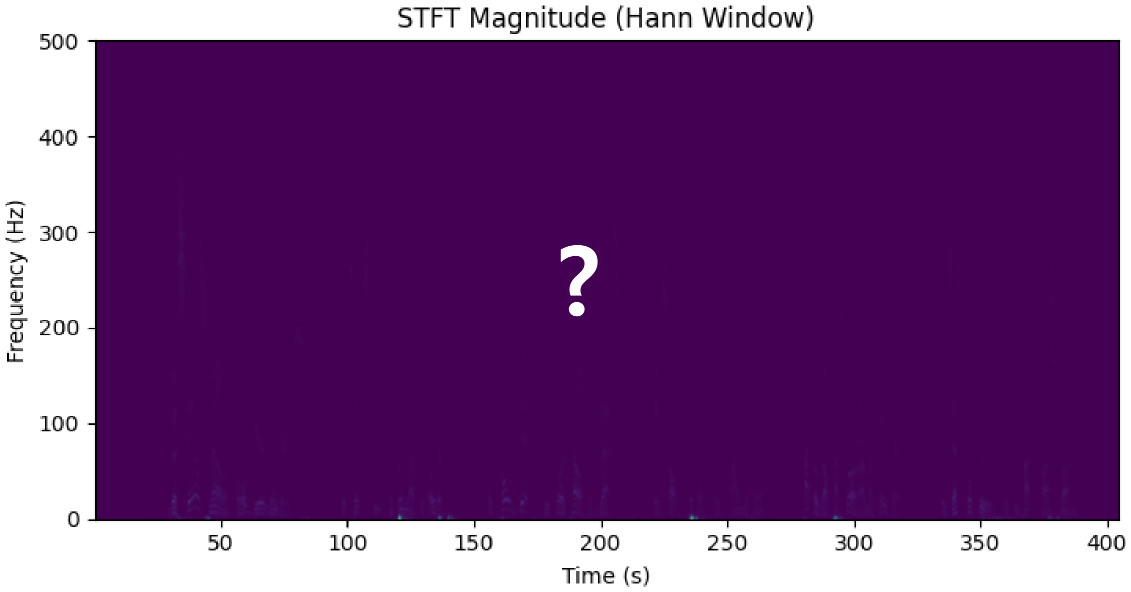
- STFT + Window function

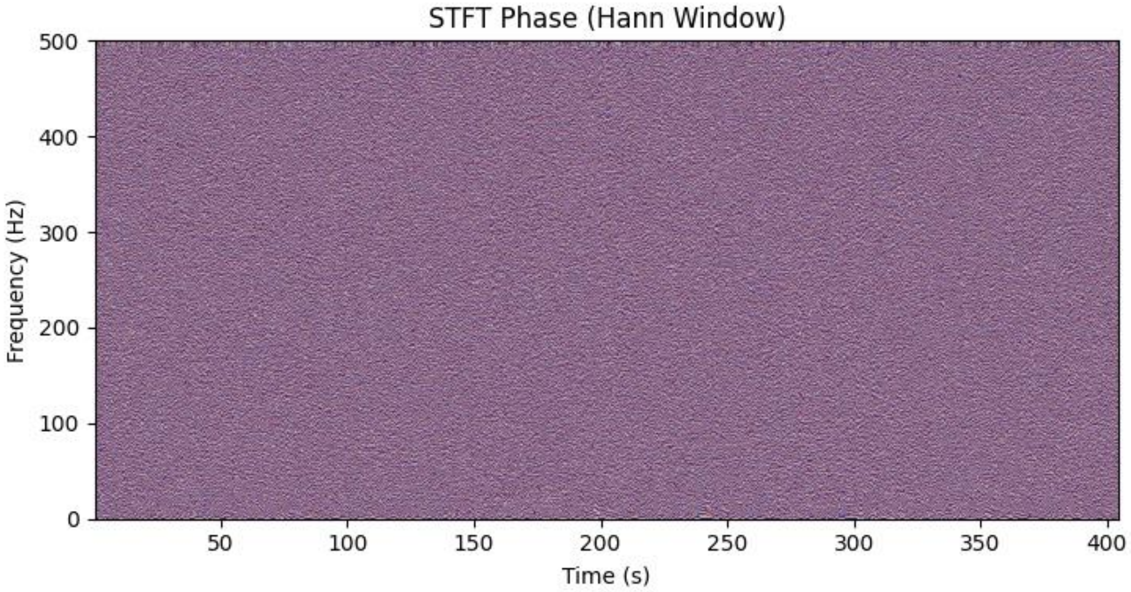




Shor-Time Fourier Transform

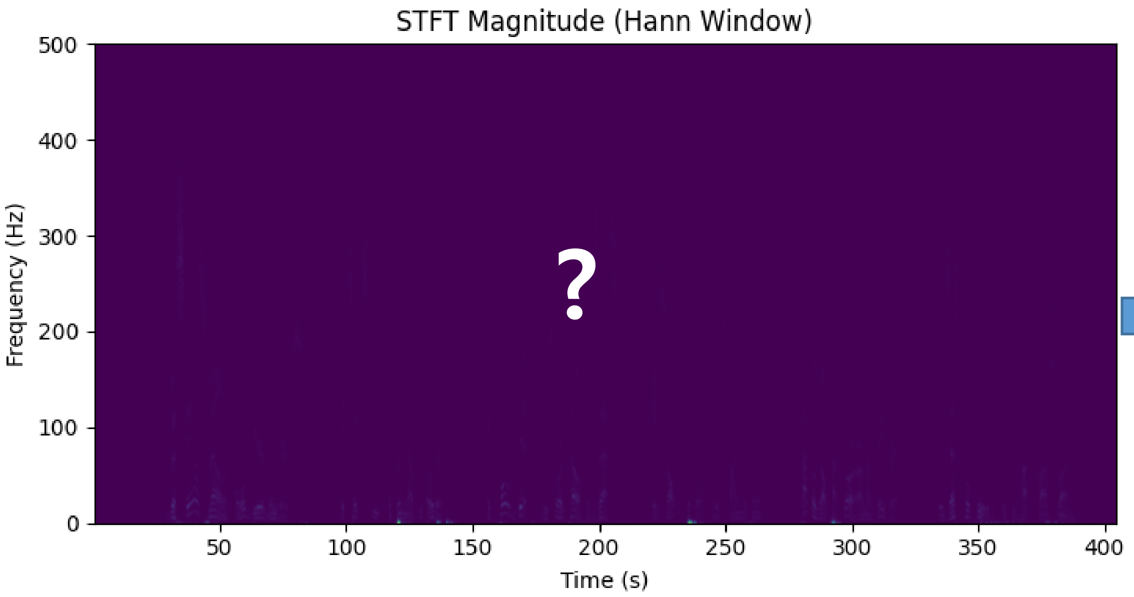
- Spectrogram:
  - The output of STFT is a complex matrix of size nFreq * nWindows
  - Useually, we look at magnitude and phase of the output

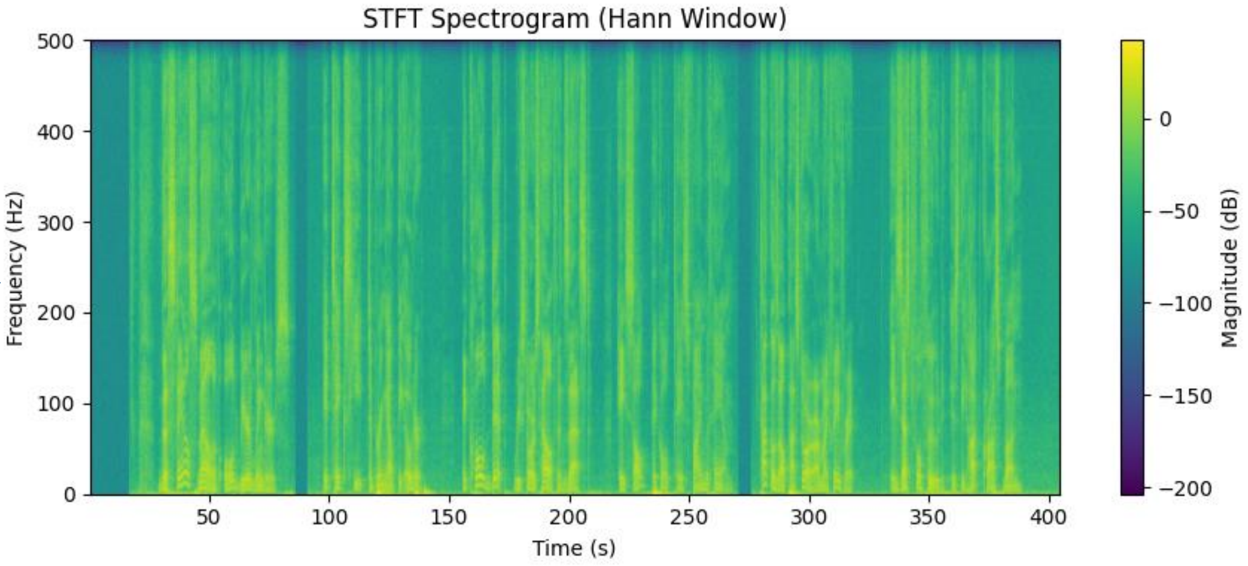




Shor-Time Fourier Transform

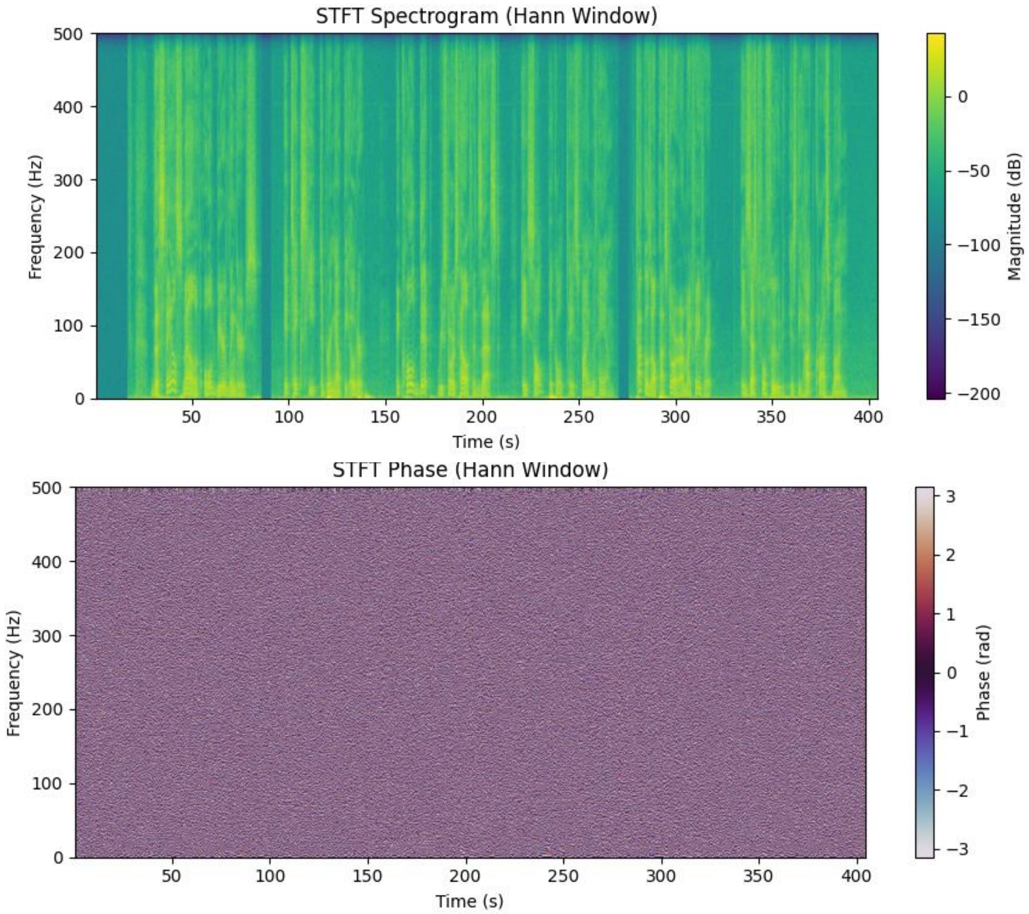
- Spectrogram:
  - The logarithm of the magnitude spectrogram is much easier visually to interpret

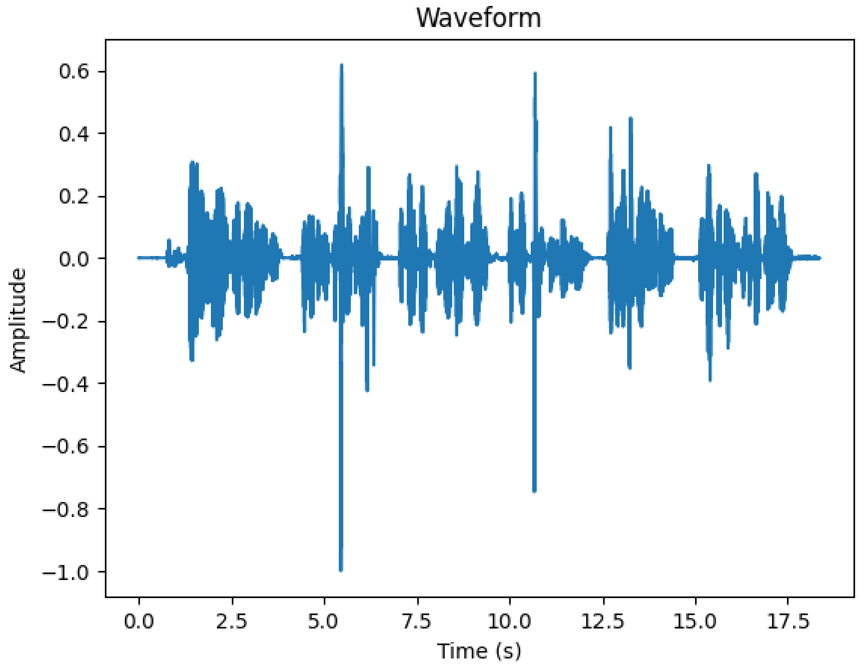




Shor-Time Fourier Transform

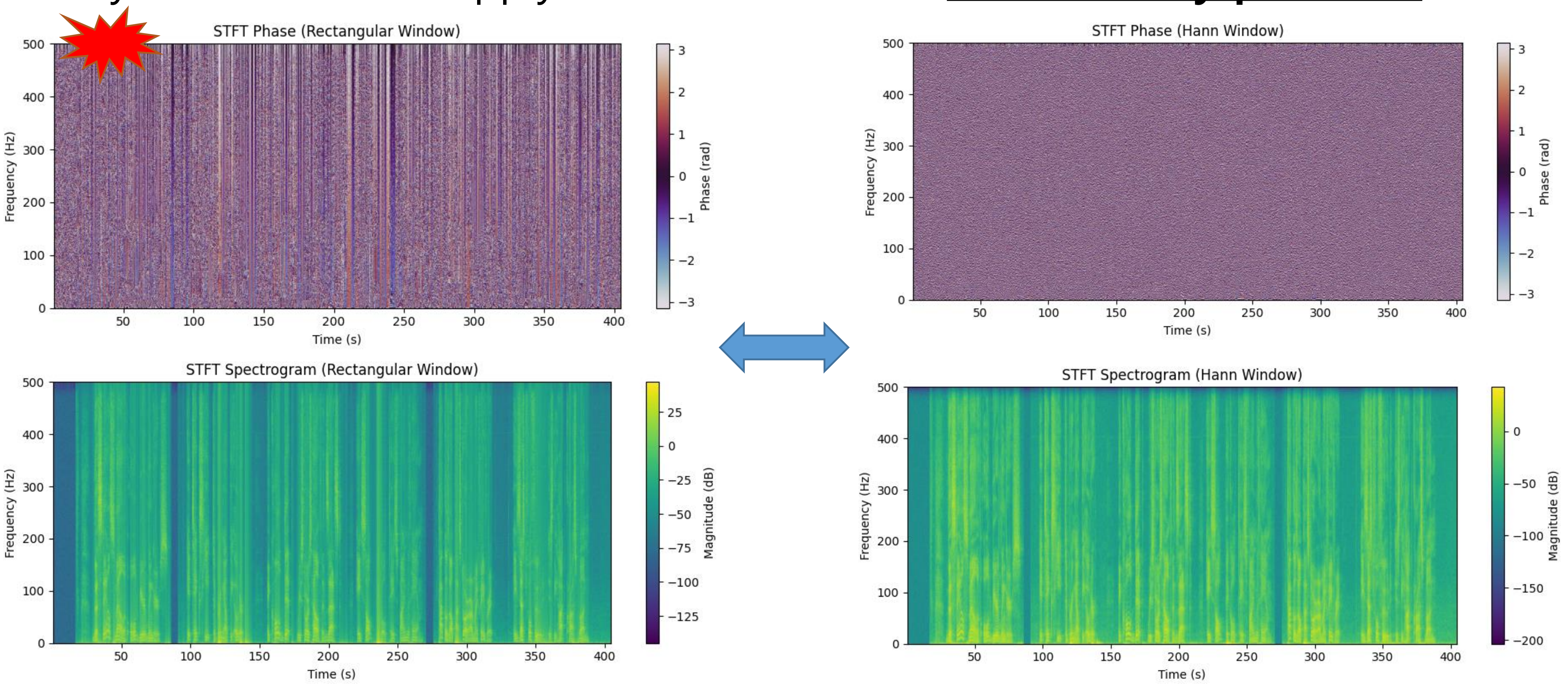
- Inver STFT (iSTFT)
  - STFT result can be inverted back given the parameters are known (window, hop and step sizes)





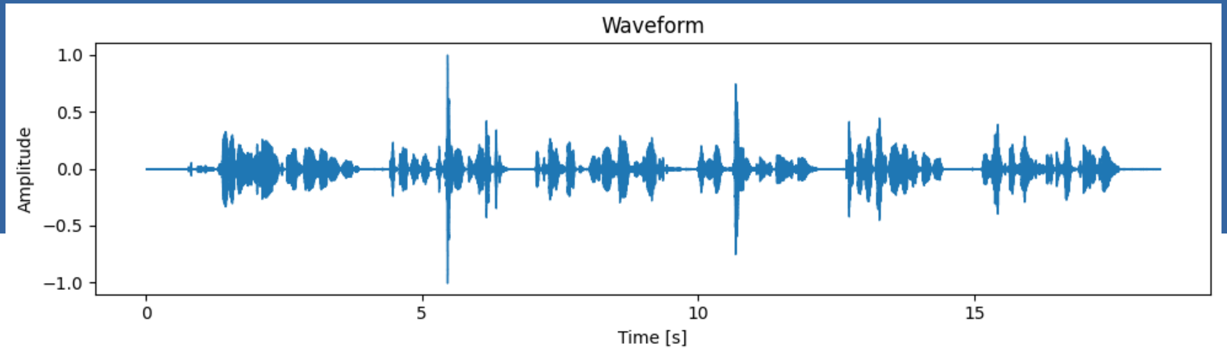
Shor-Time Fourier Transform

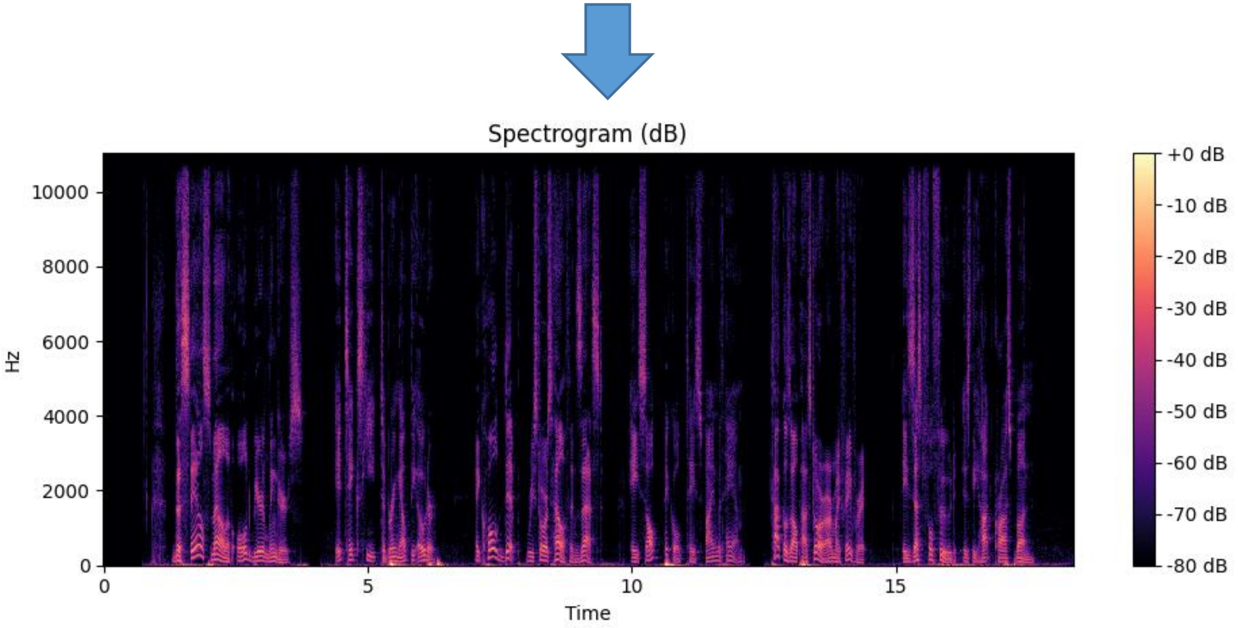
- Why do we have to apply window functions: **discontinuity problem!**



Practice!

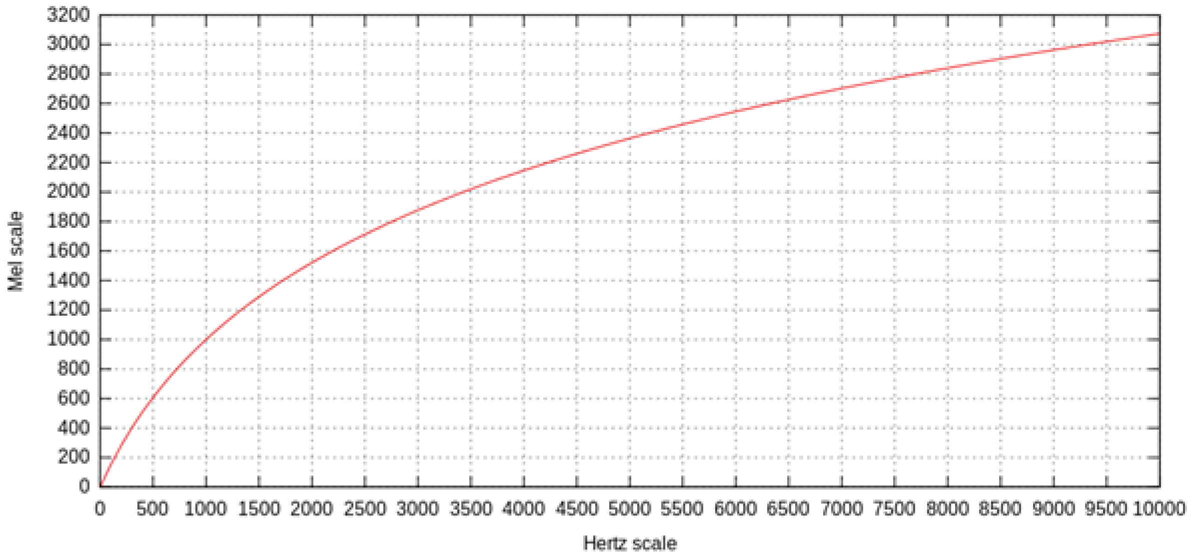
- Colab practice!





Mel Spectrogram

- Humans perceive sound on a log-scale
- For human ear:
  - 500 Hz << 600 Hz
  - but 5000 Hz ~= 5100 Hz



There is no single mel-scale formula.[3] The popular formula from O'Shaughnessy's book can be expressed with different logarithmic bases:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) = 1127 \ln\left(1 + \frac{f}{700}\right)$$

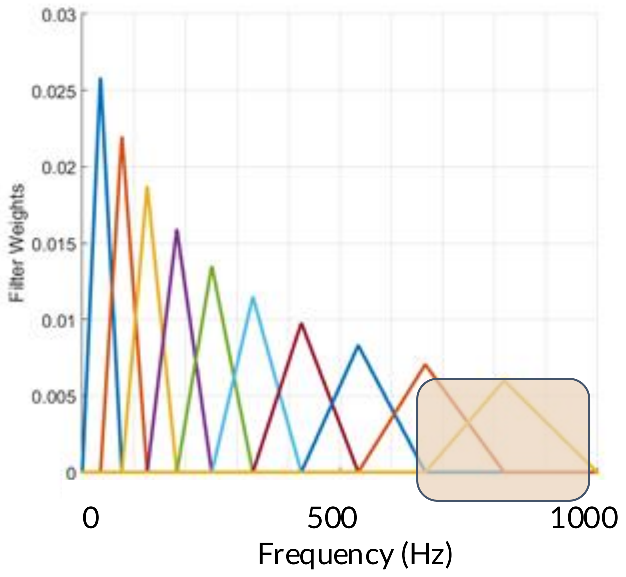
The corresponding inverse expressions are:

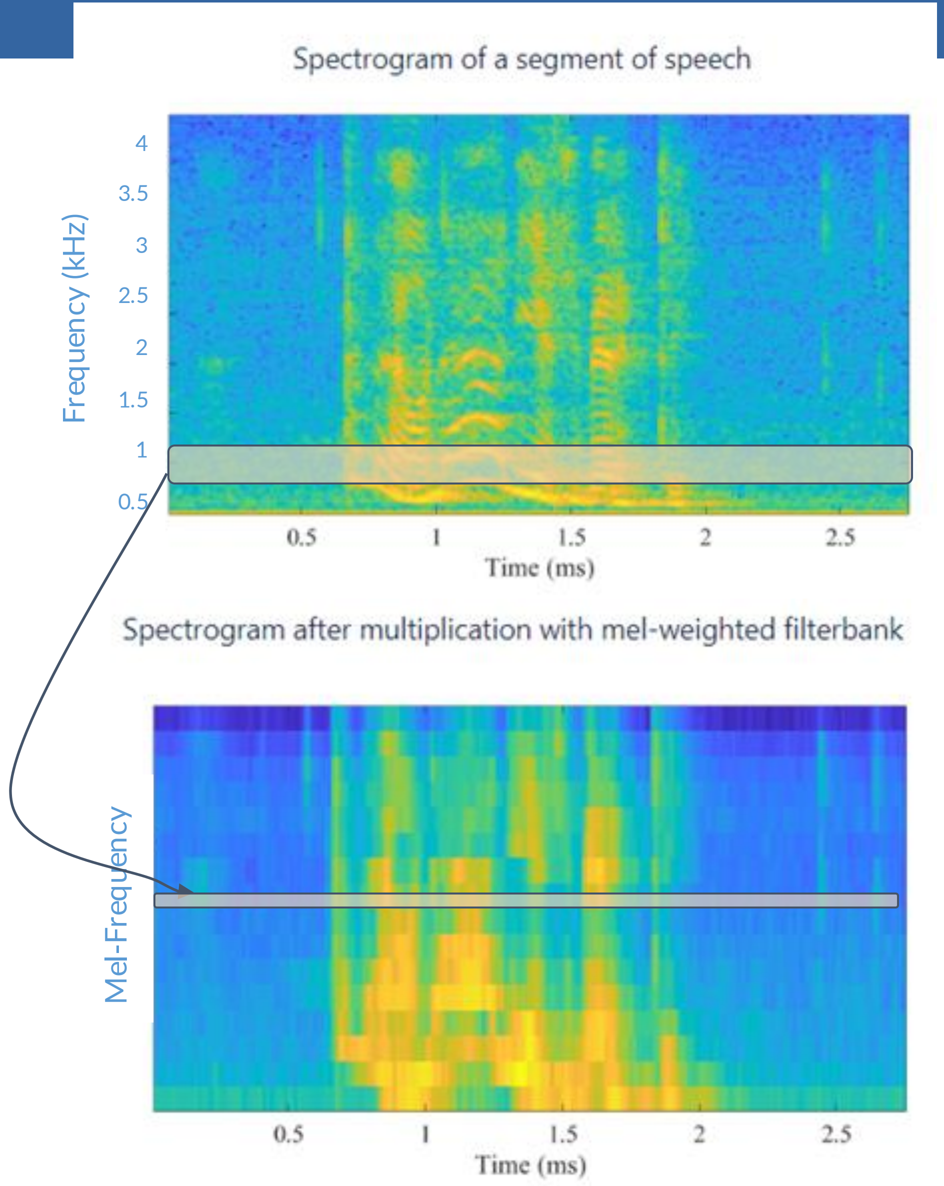
$$f = 700\left(10^{\frac{m}{2595}} - 1\right) = 700\left(e^{\frac{m}{1127}} - 1\right)$$

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

Mel Spectrogram

- Mel Spectrogram

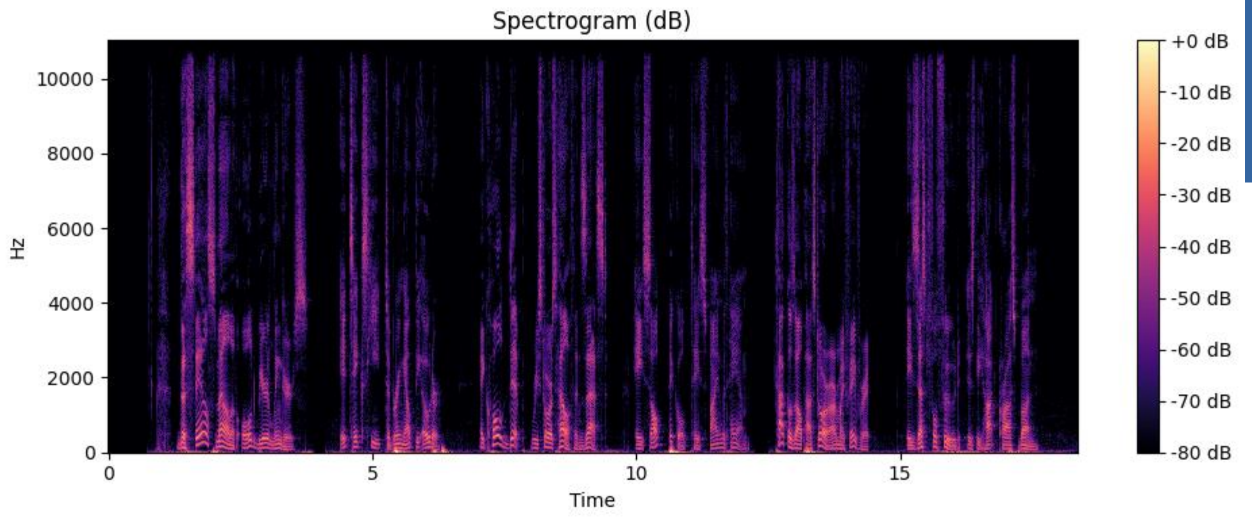


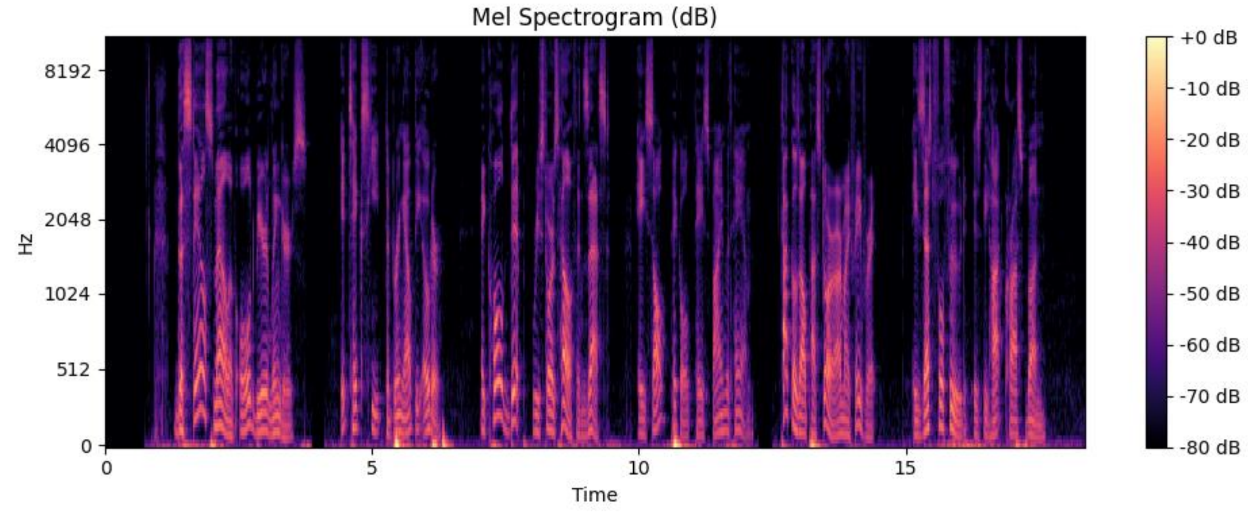


https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

Practice!

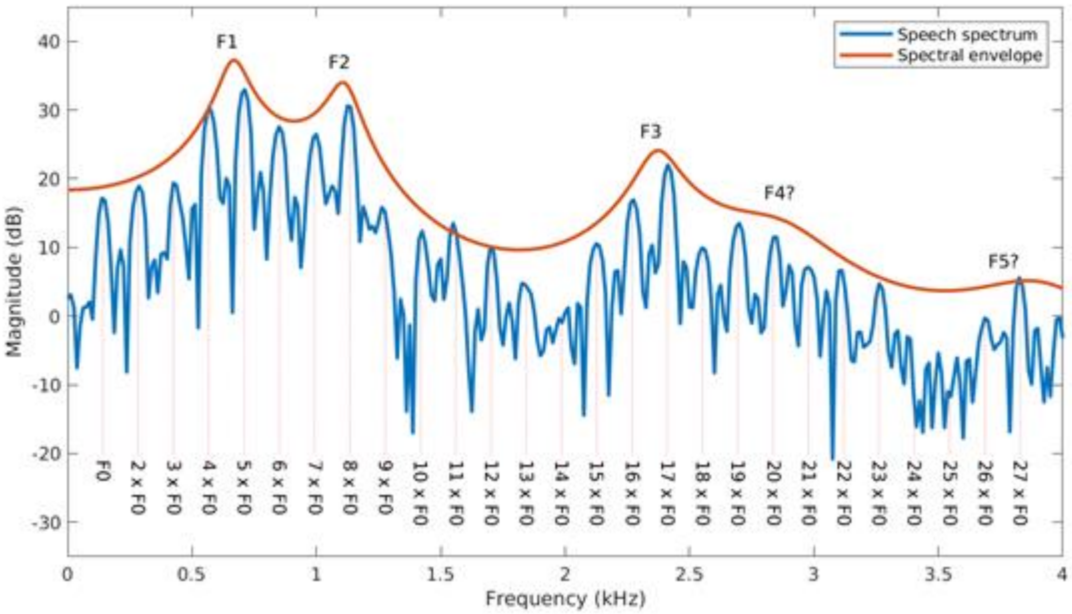
- Colab practice!





MFCC

- Fundamental frequency is the physical source frequency, but there are **resonances** (공진) and **harmonics** (배음)
- Peaks on envelope curve are **formants**
- Pitch is perceptual value, **F0 is physical**, **harmonics are k*F0**
- For speech F0 lie roughly in the range 80 to 450 Hz, typically males have lower voices than females and children



https://speechprocessingbook.aalto.fi/Representations/Fundamental_frequency_F0.html

MFCC

- Fundamental frequency is the physical source frequency, but there are **resonances** (공진) and **harmonics** (배음)

- Peaks on envelope curve are **formants**

- Pitch is perceptual value, **F0 is physical**, **harmonics are k*F0**

- For speech F0 lie roughly in the range 80 to 450 Hz, typically males have lower voices than females and children

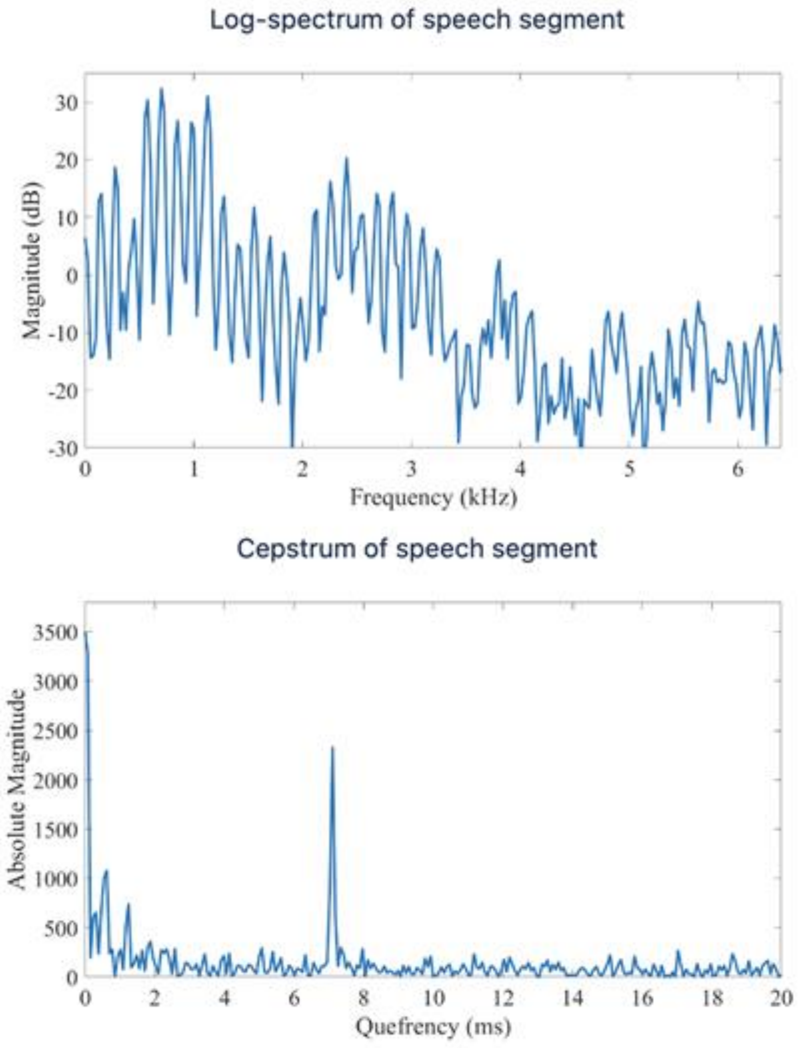
Age [years]	Female Frequency [Hz]	Male Frequency [Hz]
Infant	440-590	440-590
3	255-360	255-360
8	215-300	210-295
12	200-280	195-275
15	185-260	135-205
Adults	175-245	105-160

https://www.researchgate.net/figure/Normal-ranges-of-fundamental-frequency-parameter_tbl1_226656636

MFCC

- Cepstrum
  - Fourier spectrum of voice has periodic structure
  - Apply Inverse DFT to log-spectrum ($\log|X(\omega)|$) and obtain Cepstrum
  - Peak in Cepstrum should be located at $\frac{1}{F_0}$

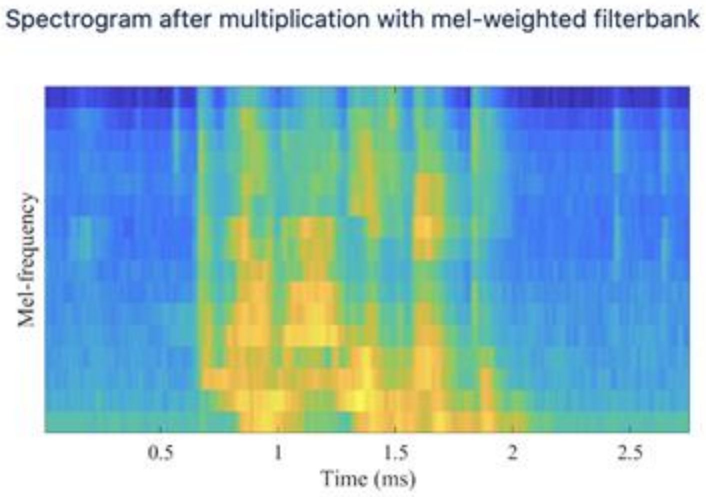
왜 이럴까?

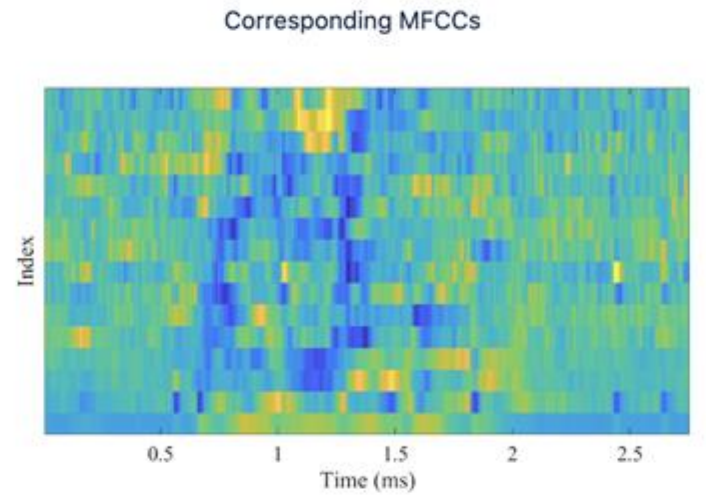


https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

MFCC

- Mel-Frequency Cepstral Coefficient (MFCC)
    - Apply STFT to the signal
    - Apply mel filters
    - Take the log value
    - Apply Discrete Cosine Transform





https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

Practice!

- Colab practice!