DL in Audio: Sound and Sound Representations

NATIONAL RESEARCH UNIVERSITY

2024
Deep Learning in Audio Processing, Invited Talks

Maksim Kaledin

About

- **PhD in Applied Mathematics:** With the topic about RL and optimal control
- **Currently:** Associate Professor, Department of Big Data and Information Retrieval
- **Sound Research for Some Time:** Key topic is audio source separation
- **Btw, there is a course:** DLA
- ... **and YouTube channel:** Sound, DL and Various Statistics

A short history of Speech Recognition

In **1962**, *IBM* introduced "**Shoebox**" which understood and responded to **16 words** in English.

50's

60's

In **1952**, *Bell Laboratories* designed the "**Audrey**" system which could recognize a single voice speaking **digits** aloud

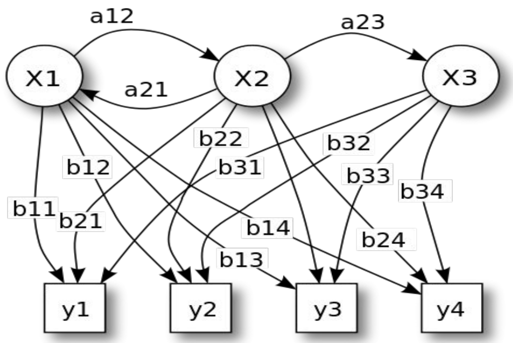
A short history of Speech Recognition

70's

DARPA's system was capable of understanding over **1,000** words. **Siri** was a spin-out of DARPA development :)

80's

The '80s saw speech recognition vocabulary go from a few hundred words to **several thousand words** thanks to **HMM**



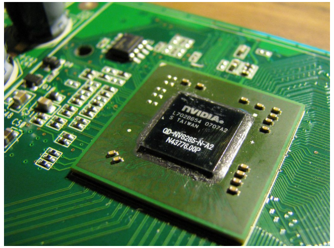
A short history of Speech Recognition

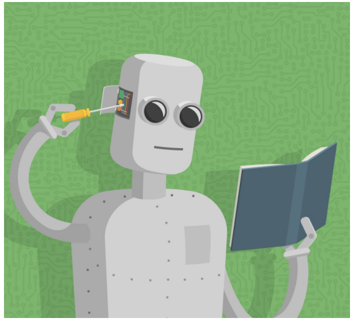
90's

Speech recognition was propelled forward in the 90s in large part because of **faster processors**

00-10's

And then came the era of big data, machine learning and GPUs





A short history of Speech Synthesis

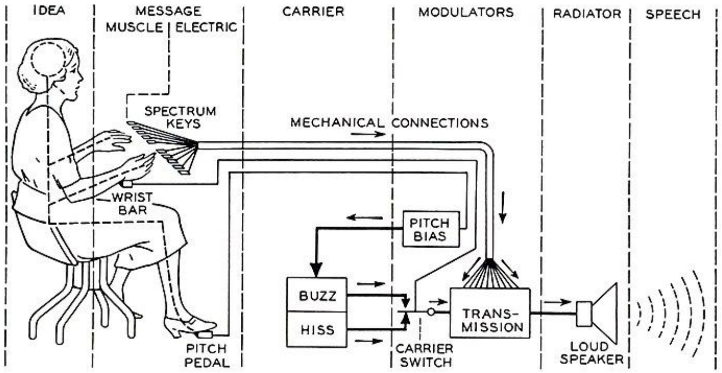
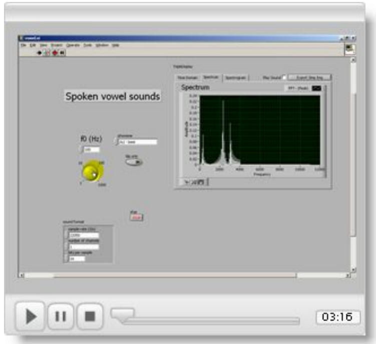


Fig. 8—Schematic circuit of the voder.

30's

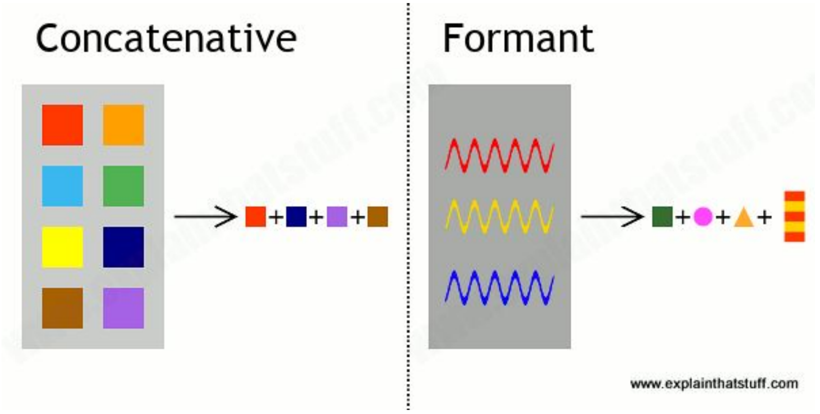
In **1939**, *The Bell Laboratory's* **Voder** was the first attempt to electronically synthesize human speech by breaking it down into its **acoustic components**



until 80's

Formant-based on rules. You may listen examples in Atari&Sega games :)

A short history of Speech Synthesis

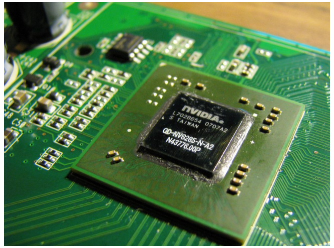


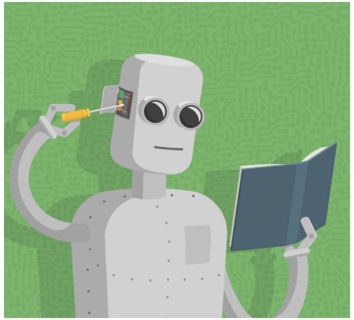
90-00's

Concatenative synthesis is a technique for synthesising sounds by concatenating short samples of recorded sound (called *units*).

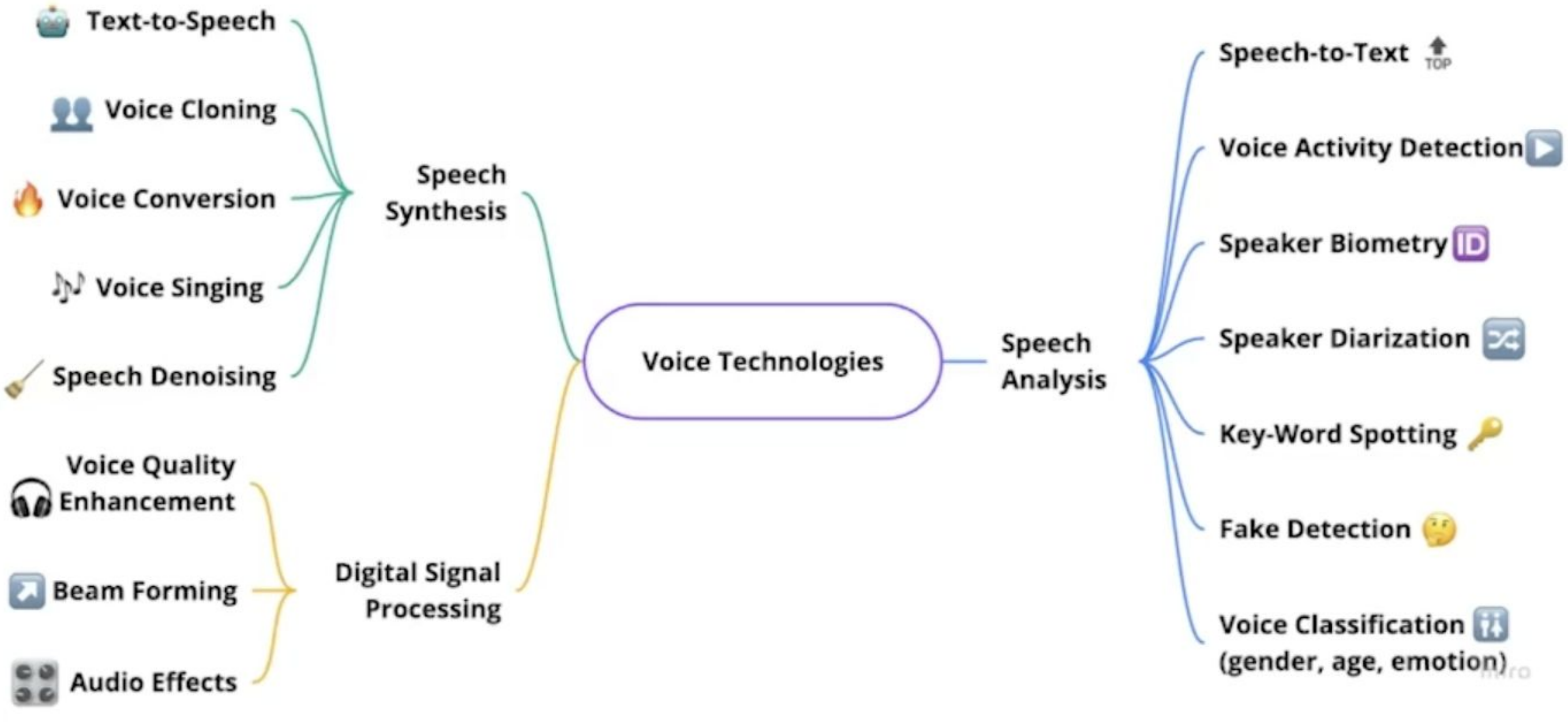
And then came the era of big data, machine learning and GPUs

10's





Briefly on Speech Technologies



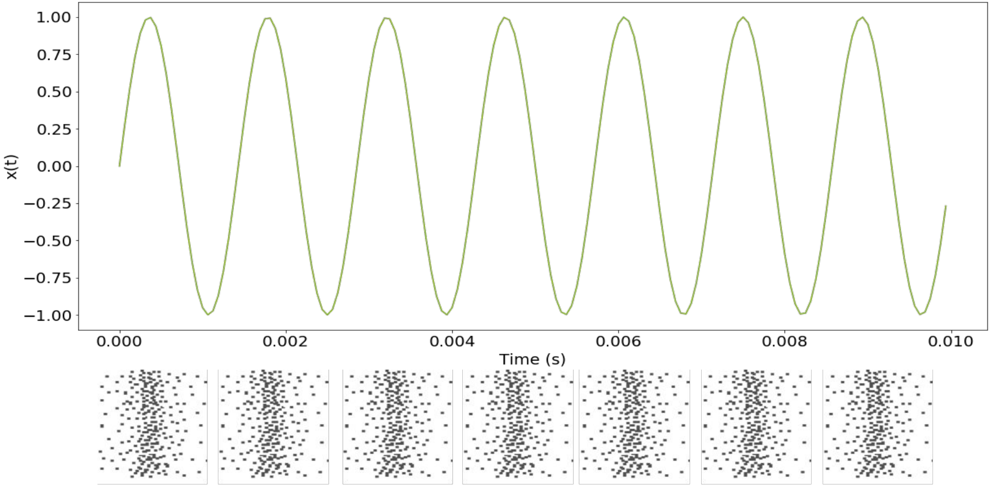
https://github.com/yandexdataschool/speech_course

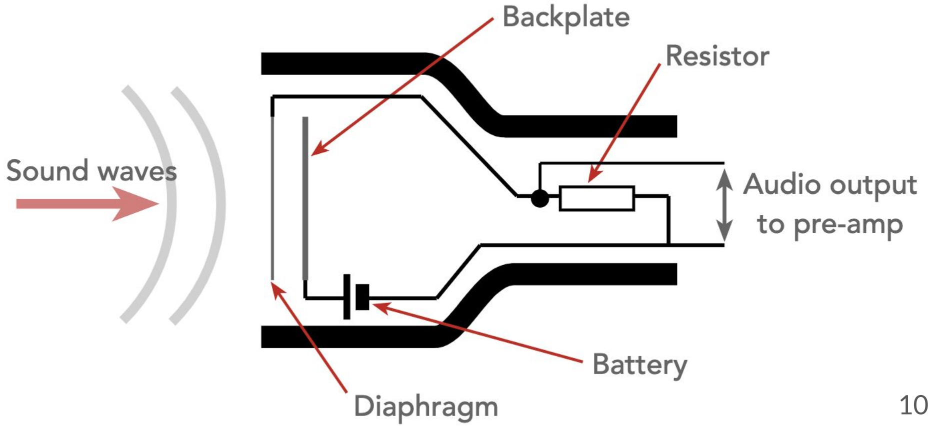
Sound representation

What is sound and how to store it in memory?

What is sound?

- **Sound wave** is the pattern of **oscillations** caused by the movement of energy traveling through the air
- **Microphone** picks up these air **oscillations** and converts them into electrical vibrations
- These **oscillations** are converted into an **analog** signal and then a **digital** signal

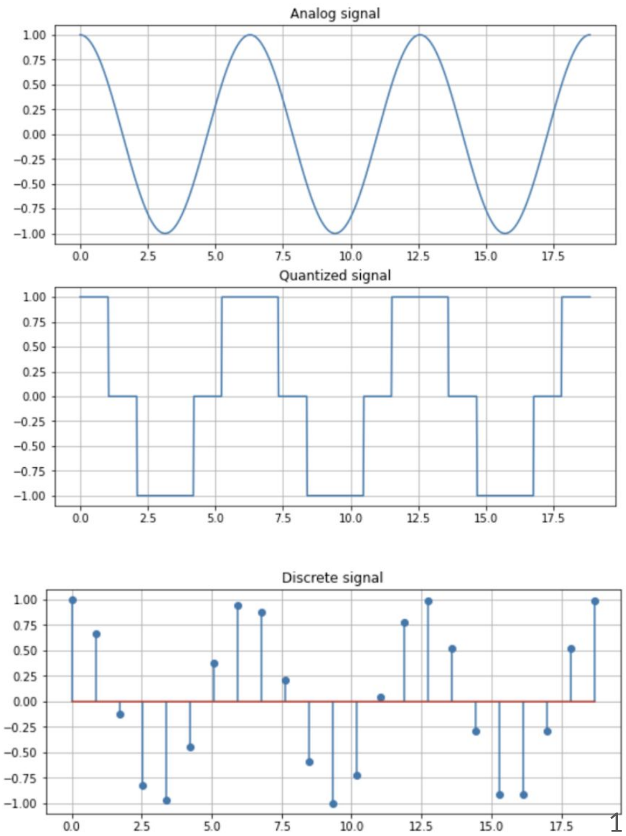




https://pudding.cool/2018/02/waveforms/
https://blog.accusonus.com/science-of-sound/how-microphones-work

How is sound stored in the computer?

- The **analog** signal is discretized, quantized and encoded
- An analog signal is **discretized** in that the signal is represented as a sequence of values taken at discrete points in time **t** with step **d**
- **Quantisation** of a signal consists in splitting the range of signal values into **N** levels in increments of **d** and selecting for each reference the level that corresponds to it
- Signal **encoding** is just a way of presenting the signal in a more compact form

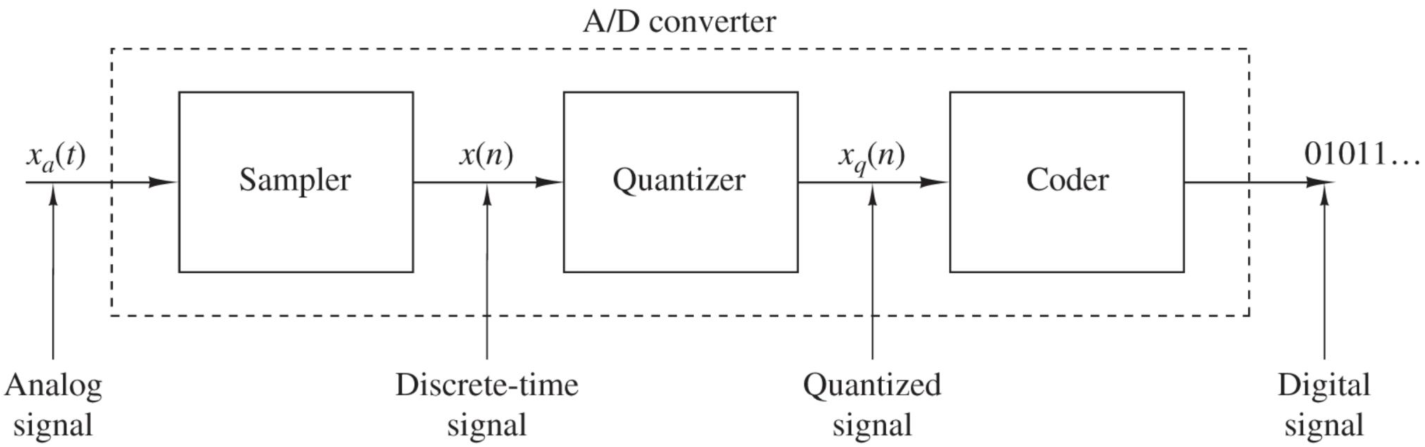


https://github.com/markovka17/dla/blob/master/week01/dsp.ipynb

Analog-to-Digital Conversion

- Converting analog signals to a sequence of numbers having finite precision

- Corresponding devices are called A/D converters (ADCs)

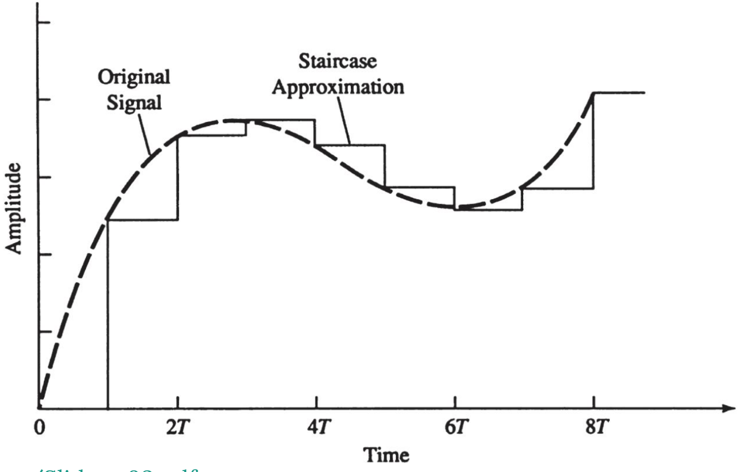


https://is.muni.cz/el/1433/jaro2012/PA190/um/Slides_02.pdf

Digital-to-Analog Conversion

- Process of converting a digital signal into an analog signal

- Interpolation
  - Connecting dots in a digital signal
  - Approximations: zero-order hold (staircase), linear, quadratic, and so on



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

What other characteristics are there?

- **Sample rate (SR)** - number of audio samples per one second (e.g. 8 kHz, 22.05 kHz, 44.1 kHz)

- **Sample size** - number of bits per one sample (e.g. 8, 16, 25, 32 bits)

- **Number of channels** -- how many signals we record in parallel (e.g. mono(1), stereo(2))

8000 Hz

The international G.711 standard for audio used in telephony uses a sample rate of 8000 Hz (8 kHz). This is enough for human speech to be comprehensible.

44100 Hz

The 44.1 kHz sample rate is used for compact disc (CD) audio. CDs provide uncompressed 16-bit stereo sound at 44.1 kHz. Computer audio also frequently uses this frequency by default.

48000 Hz

The audio on DVD is recorded at 48 kHz. This is also often used for computer audio.

96000 Hz

High-resolution audio.

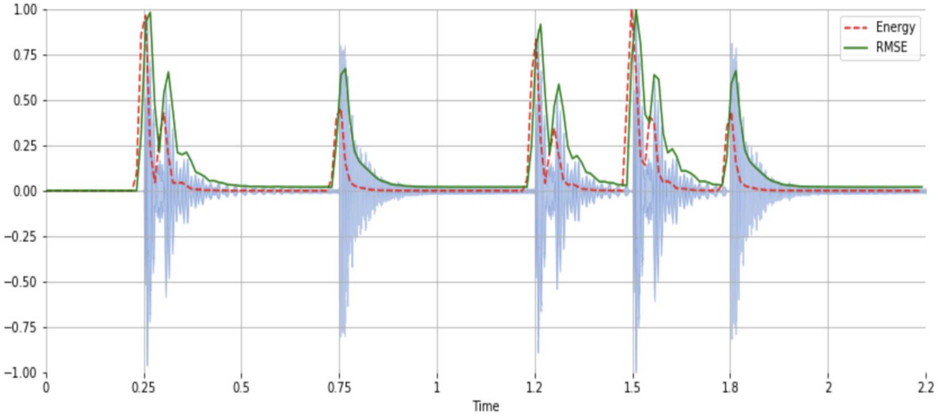
192000 Hz

Ultra-high resolution audio. Not commonly used yet, but this will change over time.

https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Audio_concepts

What other characteristics are there?

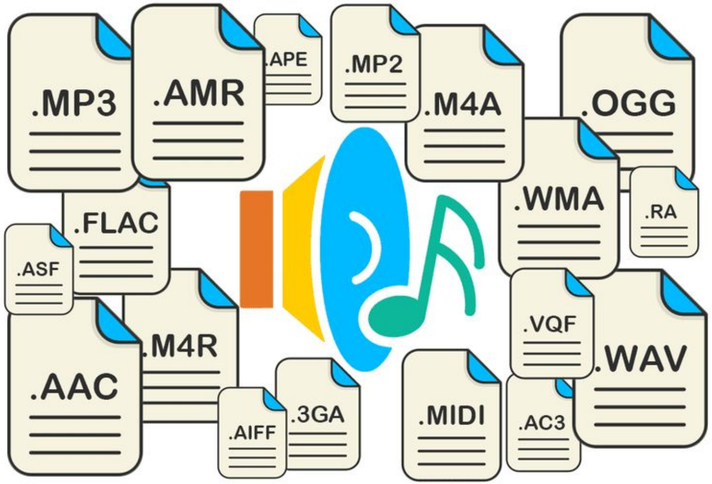
- Assume **f(n)** is our signal where **n** is time
- Power of signal is $f^2(n)$
- Energy of signal (**E**) is $\sum f^2(n)$
- In practice estimated by some window
- Energy in **decibels:** $10 \log_{10} E$
- $\mathrm{SNR}_{dB} = 10 \log_{10} \frac{E_{\text{signal}}}{E_{\text{noise}}}$



https://musicinformationretrieval.com/energy.html

What about audio formats?

- Non-compressed formats: **WAV, AIFF, etc.**
- Lossless compression(2:1) : **FLAC, ALAC, etc.**
- Lossy compression(10:1) : **MP3, Opus, etc**
- **Bit rate** measure a degree of compression. Number of bit that are conveyed or processed per **unit of time.**



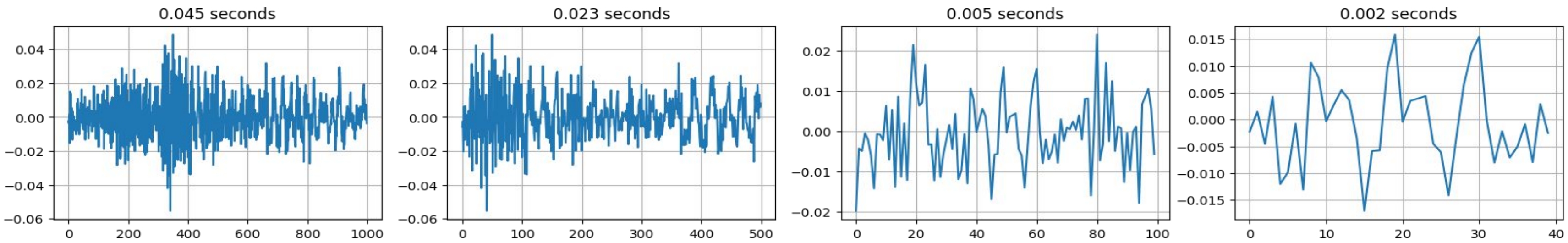
https://en.wikipedia.org/wiki/Audio_file_format

Frequencies and Spectrograms

Why not just use wave representation for ML?

Problems with the waveform

- One letter/sound consists of 2000-4000 amplitudes, so they are expensive to process and store



- No "invariant" regarding noise and transformations
- Periodical nature of audio signals

Voice

Voice + Noise

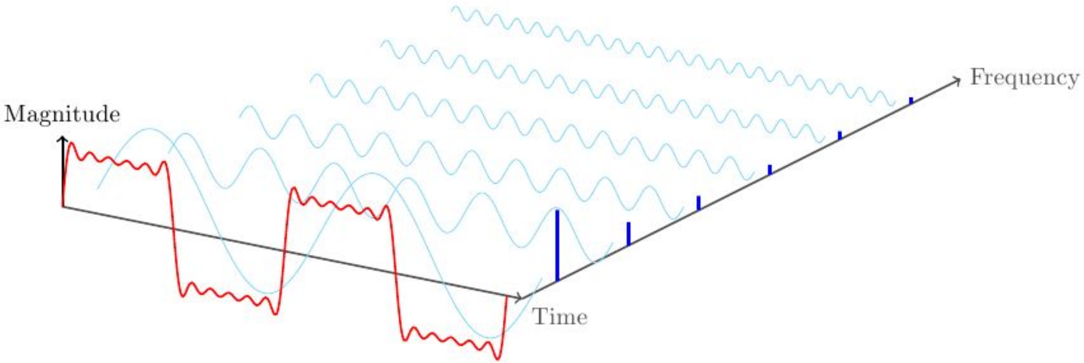
Complex waves as a sum of sigmoids

We want to represent a periodic function as a sum of sigmoids with different periods (frequencies), shifts and amplitudes.

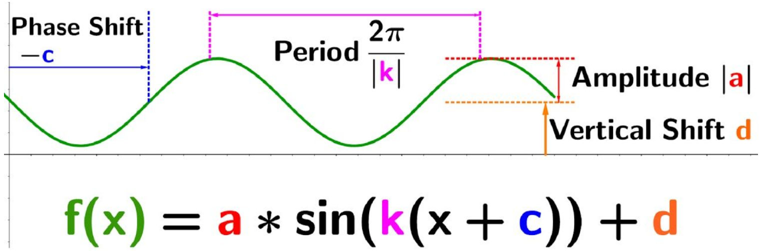
$$f(x) = A_1 * sin(freq_1 x + \phi_1) + \ldots$$

$$\ldots$$

$$\ldots + A_n * sin(freq_n x + \phi_n)$$



Parameters of a sine wave



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

Complex waves as a sum of sigmoids

We want to represent a periodic function as a sum of sigmoids with different periods (frequencies), shifts and amplitudes.

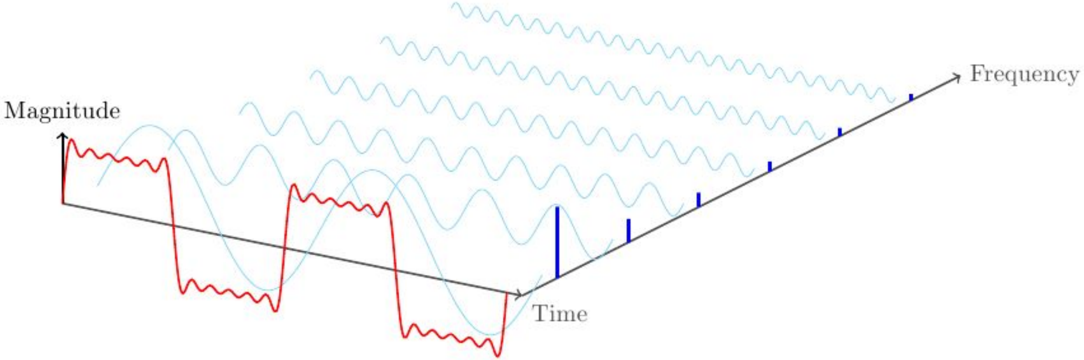
$$f(x) = A_1 * sin(freq_1 x + \phi_1) + \ldots$$

$$\ldots$$

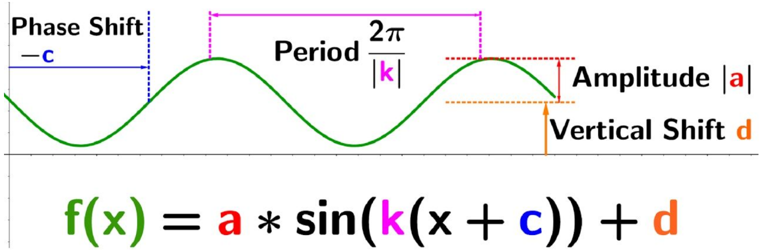
$$\ldots + A_n * sin(freq_n x + \phi_n)$$

And for audio processing we are only interested in:

- **Frequencies**
- **Amplitudes**



Parameters of a sine wave



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

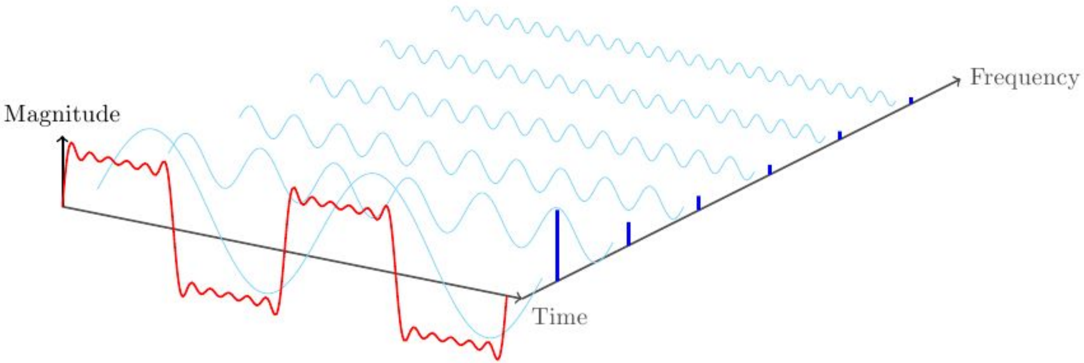
Complex waves as a sum of sigmoids

We want to represent a periodic function as a sum of sigmoids with different periods (frequencies), shifts and amplitudes.

$$f(x) = A_1 * sin(freq_1 x + \phi_1) + ...$$

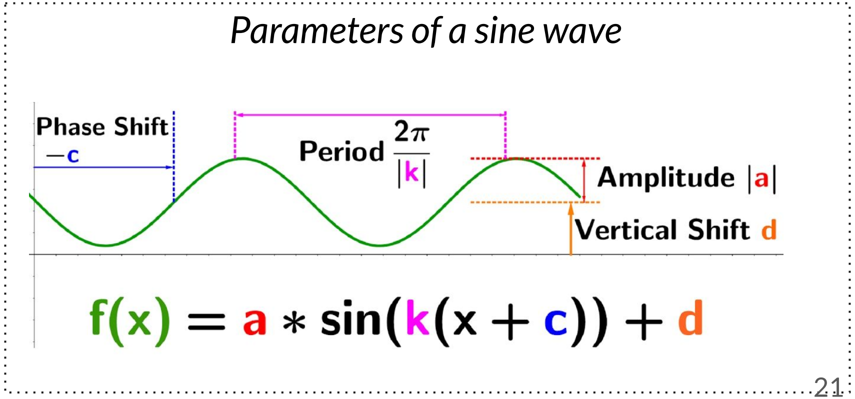
$$...$$

$$... + A_n * sin(freq_n x + \phi_n)$$



And for audio processing we are only interested in:

- **Frequencies**
- **Amplitudes**
- **Phases(?)**



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

Fourier Transform

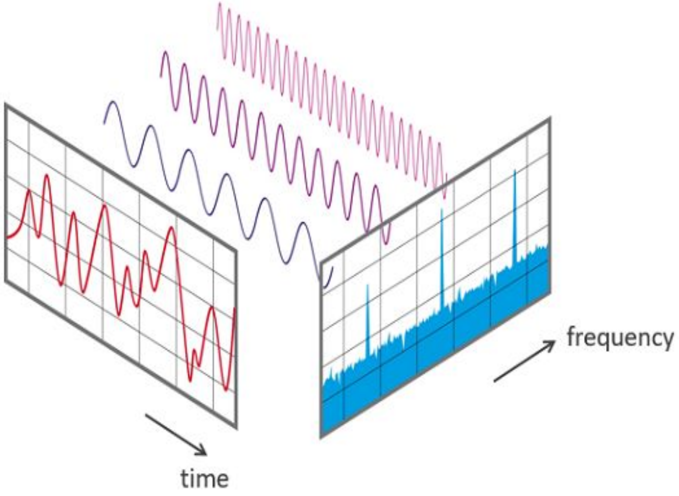
- The **Fourier transform(FT)** is a mathematical formula that allows us to decompose a signal into its individual **frequencies** and the frequency's **amplitude**

- FT transfer a signal from real-valued function of the **time domain** to a complex-valued function of **frequency domain**

Fourier transform integral

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx, \quad \forall\, \xi \in \mathbb{R}.$$

$$f : \mathbb{R} \to \mathbb{R}$$
$$\hat{f} : \mathbb{R} \to \mathbb{C}$$



- The function must meet the following conditions:
  - to be **bounded**
  - to be **absolutely integrable**
  - to have a **finite number** of minimas, maximas and discontinuities

https://youtu.be/spUNpyF58BY

Fourier Transform

- The **Fourier transform(FT)** is a mathematical formula that allows us to decompose a signal into its individual **frequencies** and the frequency's **amplitude**

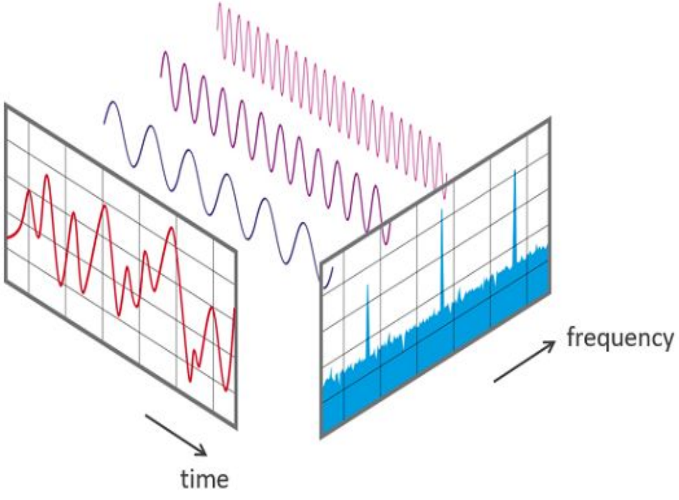
- FT transfer a signal from real-valued function of the **time domain** to a complex-valued function of **frequency domain**

- Fourier transform integral

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx, \quad \forall\, \xi \in \mathbb{R}.$$

Frequency | Original signal

$$f : \mathbb{R} \to \mathbb{R}$$
$$\hat{f} : \mathbb{R} \to \mathbb{C}$$



Inverse Fourier Transform

Fourier transform integral

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx, \quad \forall\, \xi \in \mathbb{R}.$$

Fourier inversion integral

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{i2\pi\xi x}\, d\xi, \quad \forall\, x \in \mathbb{R},$$

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

Inverse Fourier Transform

Fourier transform integral

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx, \quad \forall\, \xi \in \mathbb{R}.$$

Fourier inversion integral

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{i2\pi\xi x}\, d\xi, \quad \forall\, x \in \mathbb{R},$$

$$= 2\int_{0}^{\infty} \mathrm{Re}\left(\hat{f}(\xi) \cdot e^{i2\pi\xi x}\right) d\xi$$

Property of FT

$$\hat{f}(\xi) = \begin{cases} \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx, & \xi \geq 0 \\ \hat{f}^{*}(|\xi|) & \xi < 0, \end{cases}$$

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

Inverse Fourier Transform

Property of FT

$$\hat{f}(\xi) = \begin{cases} \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx, & \xi \geq 0 \\ \hat{f}^{*}(|\xi|) & \xi < 0, \end{cases}$$

Euler's formula

$$e^{jx} = \cos x + j \sin x$$

Fourier transform integral

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx, \quad \forall\, \xi \in \mathbb{R}.$$

Fourier inversion integral

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{i2\pi\xi x}\, d\xi, \quad \forall\, x \in \mathbb{R},$$

$$= 2\int_{0}^{\infty} \mathrm{Re}\left(\hat{f}(\xi) \cdot e^{i2\pi\xi x}\right) d\xi$$

$$= 2\int_{0}^{\infty} \left(\mathrm{Re}(\hat{f}(\xi)) \cdot \cos(2\pi\xi x) - \mathrm{Im}(\hat{f}(\xi)) \cdot \sin(2\pi\xi x)\right) d\xi.$$

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

Discrete Fourier Transform (DFT)

How to calculate Fourier Transform in practice?

Discrete Fourier transform

- Operates on signal **X** consisting of **N uniformly sampled across [0,T] points**
- **Discrete analogue of FT:** at frequency number k (which is 2πk/N*sampleRate Hz) it gives a complex number

$$\hat{X}_k = \sum_{t=0}^{N-1} X_t e^{-\frac{2\pi k i}{N} t}$$

Discrete Fourier transform

- Operates on signal **X** consisting of **N uniformly sampled across [0,T] points**
- **Discrete analogue of FT:** at frequency number k (which is 2πk/N*sampleRate Hz) it gives a complex number

$$\hat{X}_k = \sum_{t=0}^{N-1} X_t e^{-\frac{2\pi k i}{N} t}$$

phase

$$\hat{X}_{\boxed{k}} = \boxed{a_k} e^{-i\boxed{\phi_k}} = a_k(\cos(\phi_k) - i\sin(\phi_k))$$

frequency amplitude

Discrete Fourier transform

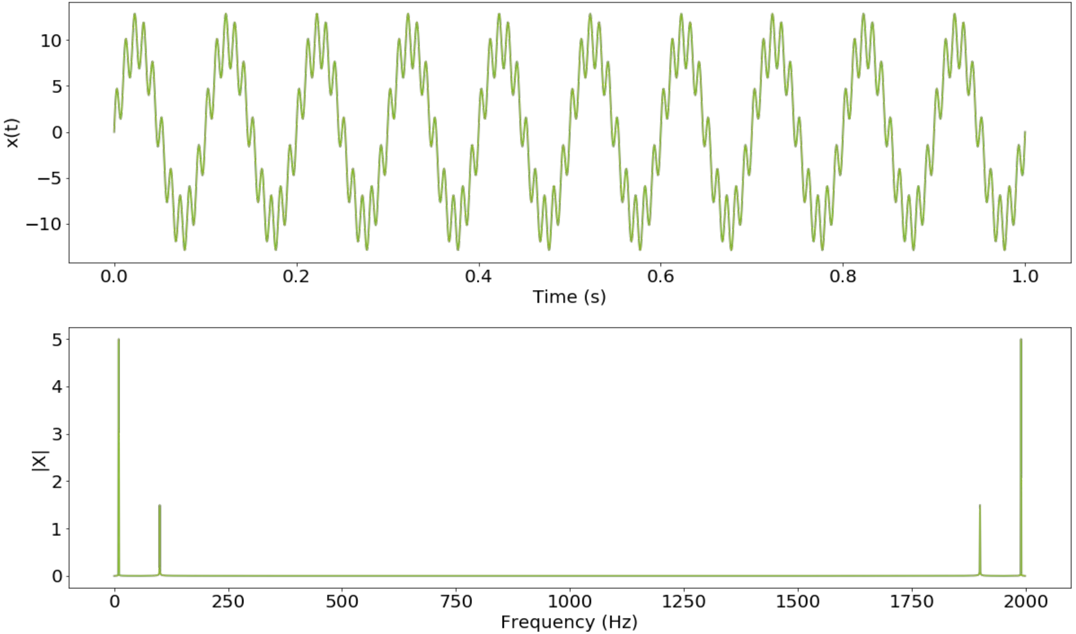
$$\hat{X} = MX$$

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

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

Example of DFT

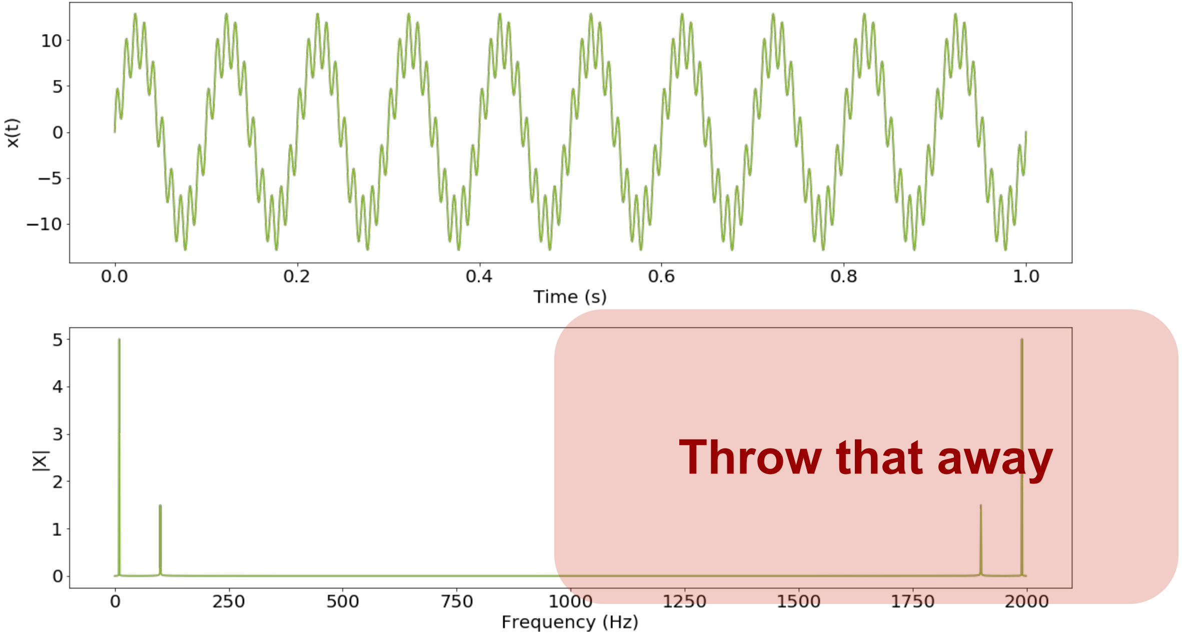
$$f(t) = 10\sin(2\pi 10t) + 3\sin(2\pi 100t)$$



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

Example of DFT

$$f(t) = 10\sin(2\pi 10t) + 3\sin(2\pi 100t)$$

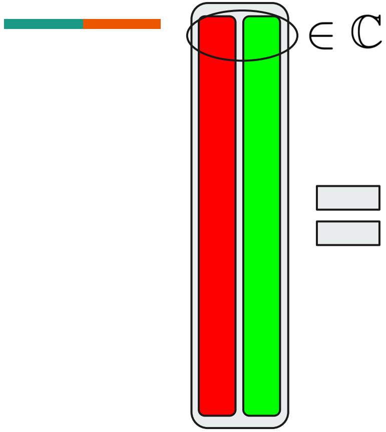


https://www.gaussianwaves.com/2015/11/interpreting-fft-results-obtaining-magnitude-and-phase-information/

Why spectrum is mirroring?

$$
\begin{aligned}
X_m &= \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi \frac{m}{N} n\right) \\
X_{N-m} &= \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi \frac{N-m}{N} n\right) \\
&= \sum_{n=0}^{N-1} x_n \exp\left(-j2\pi n + j2\pi \frac{m}{N} n\right) \\
&= \sum_{n=0}^{N-1} x_n \exp\left(j2\pi \frac{m}{N} n\right) \\
&= (X_m)^*
\end{aligned}
$$

Discrete Fourier transform



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

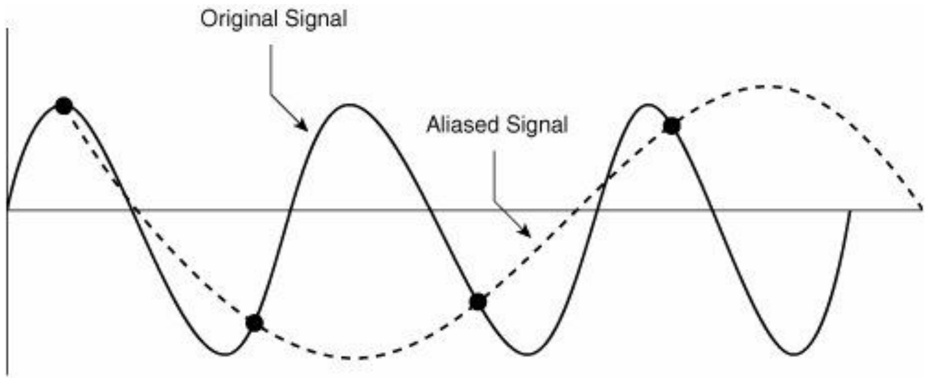
phase

$$\hat{X}_{\boxed{k}} = \boxed{a_k} e^{-i\boxed{\phi_k}} = a_k(cos(\phi_k) - isin(\phi_k))$$

frequency amplitude

Kotelnikov Theorem

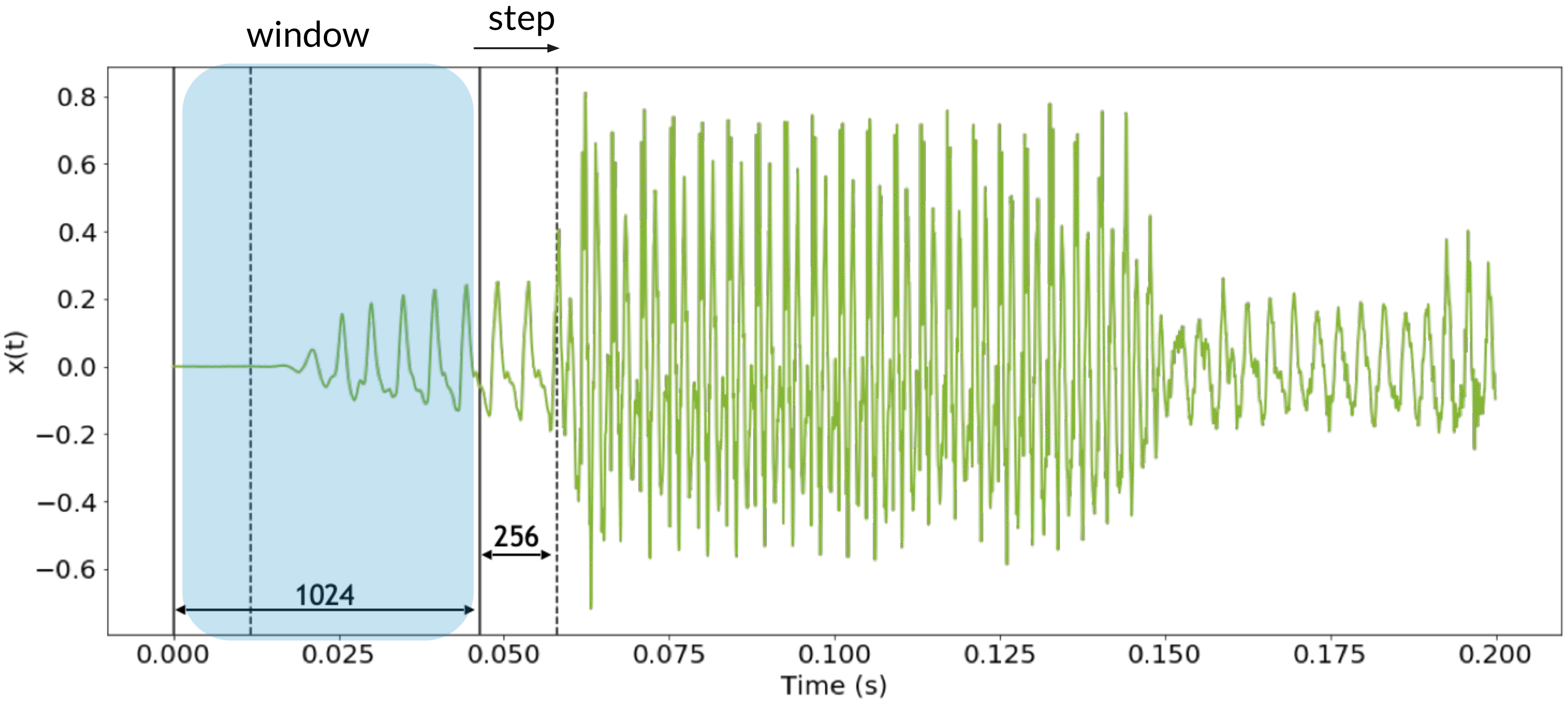
- If a function **f(t)** contain no frequencies higher than **B hertz**, it is completely determined by giving its ordinates at series of points spaced **1/2B** seconds apart
- **Example:** If signal contains frequency 100 Hz, the sampling rate for this signal needs to be 200 Hz at least
- DFT of a segment of a signal with sample rate N, will produce amplitudes for nfft evenly spread frequencies in range [-sampleRate / 2; sampleRate /2]



Short Time Fourier Transform (STFT)

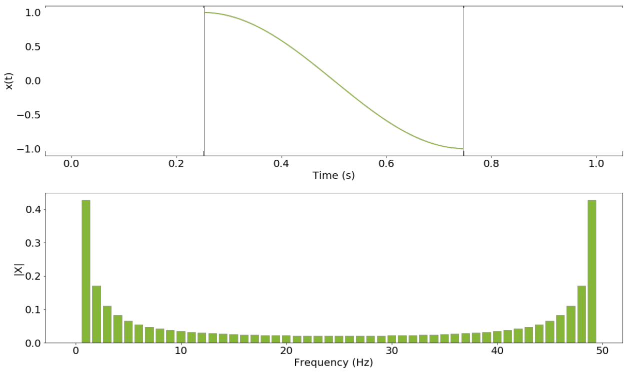
How to apply FT to a long non-periodic signal?

Short-Time Fourier Transform

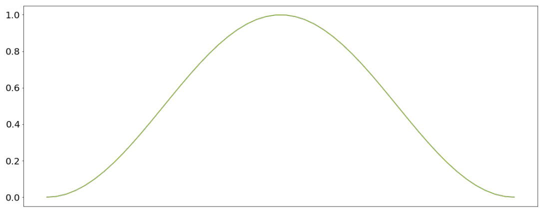


Window functions

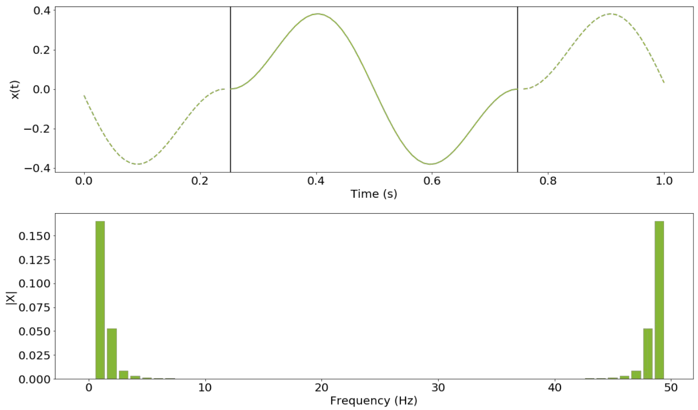
Sliced signal



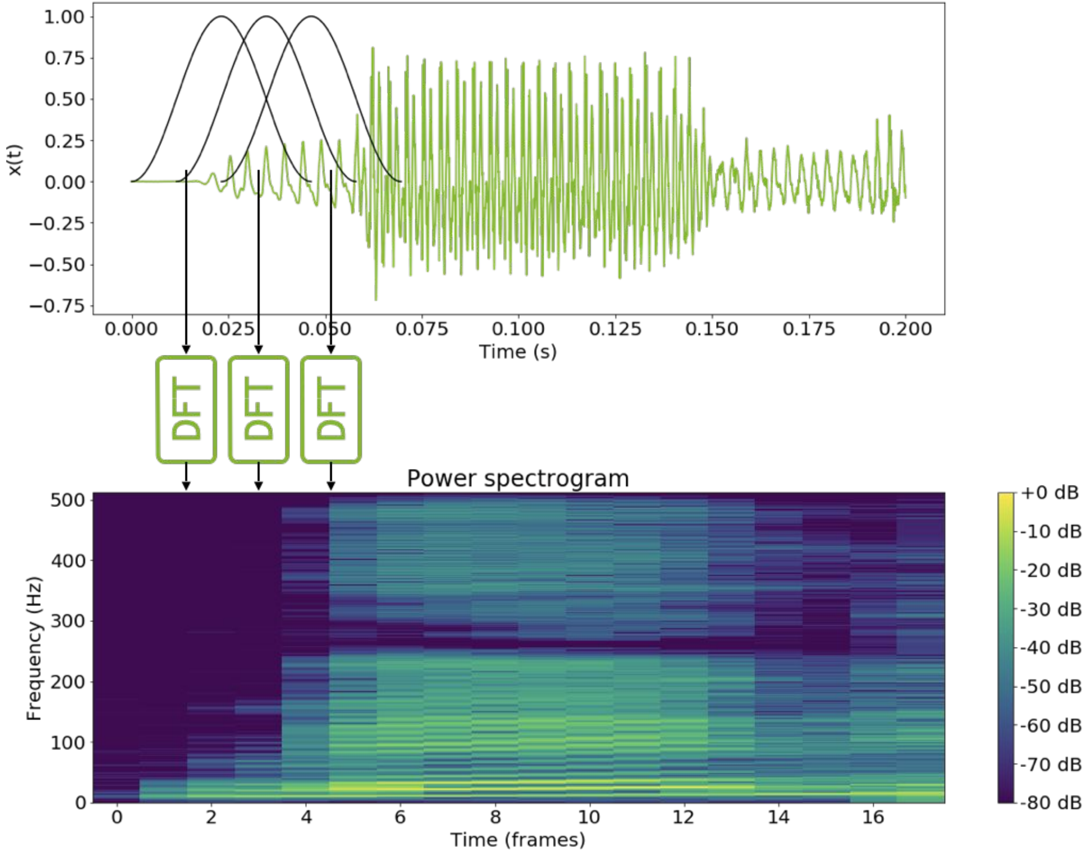
Window



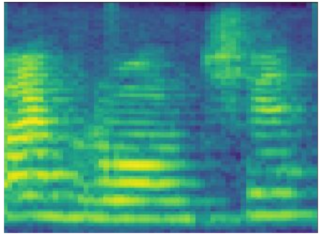
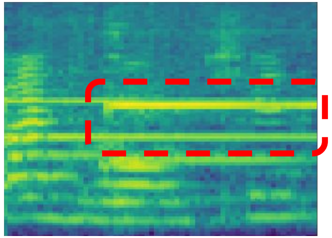
Windowed signal



Short Time Fourier Transform + window function



Spectrogram

	Wave	Spectrogram
Voice		
Voice + Noise		

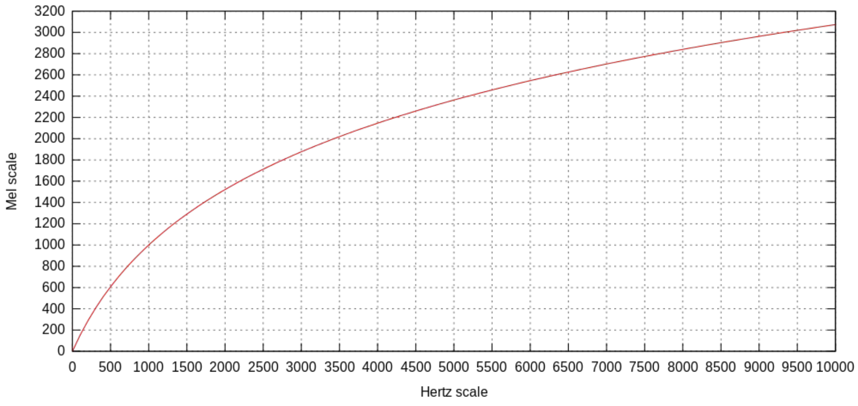
Practical use: values of the spectrogram are very small, so typically the log-spectrogram is used instead

Mel Scale

Compressing the spectrogram

Mel Scale

- 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:

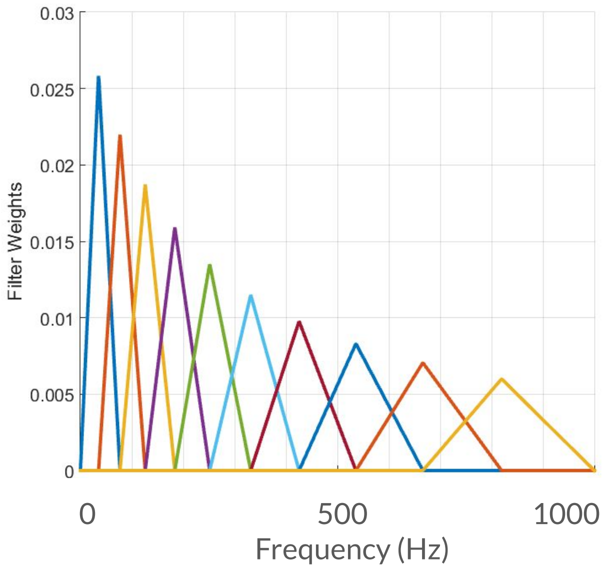
$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) = 1127 \ln\left(1 + \frac{f}{700}\right)$$

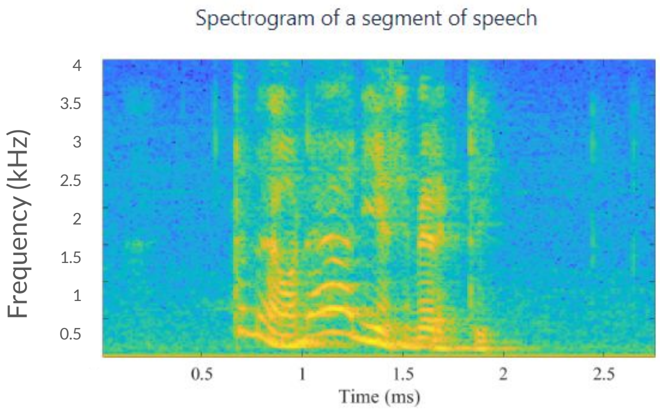
The corresponding inverse expressions are:

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

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

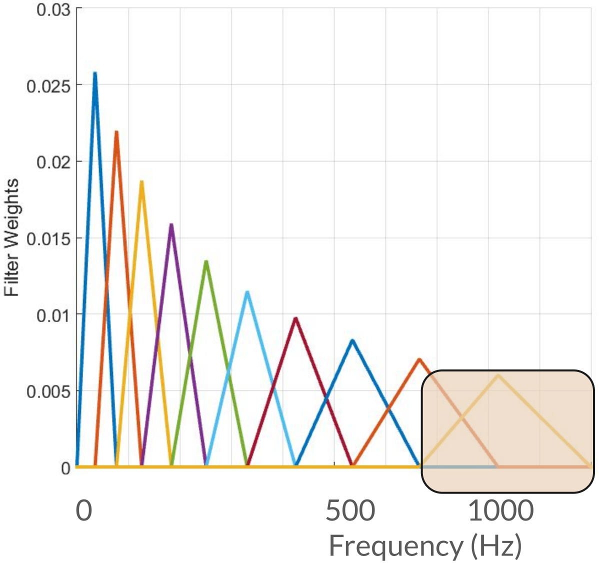
Mel Spectrogram

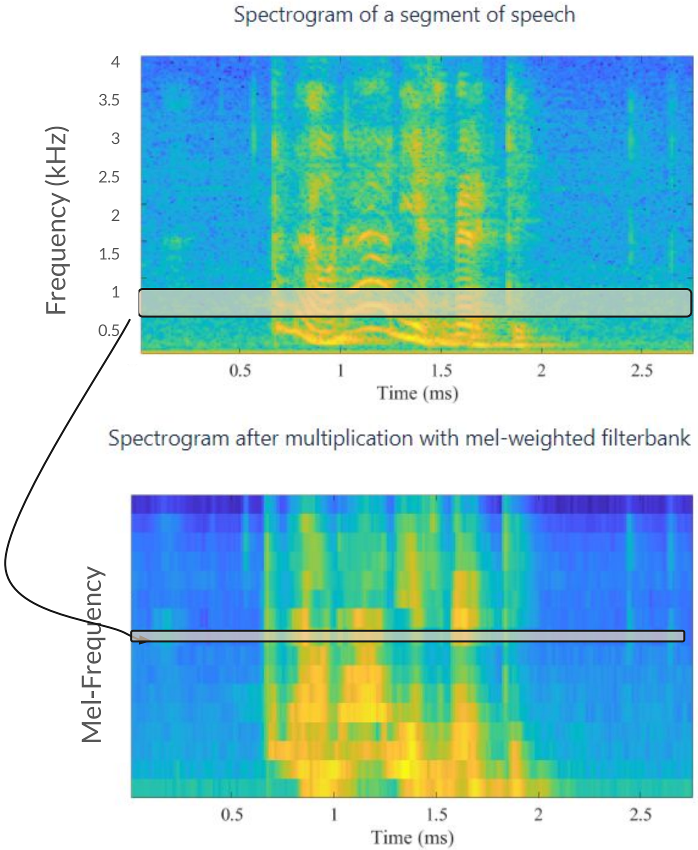




https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

Mel Spectrogram





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

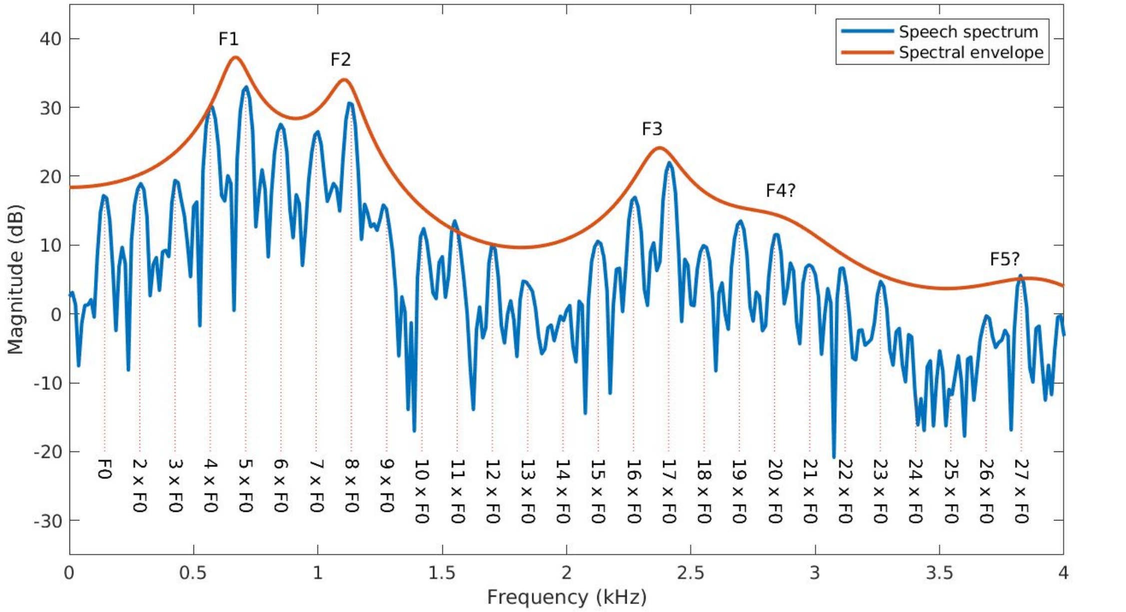
MFCC

Decorrelating the spectrogram

- Sound representation
- Motivation for spectrograms
- Fourier Transform
- Discrete Fourier Transform
- Short Time Fourier Transform
- Spectrogram
- Mel scale
- **MFCC**

Fundamental Frequency

- **Fundamental frequency** refers to the approximate frequency of the (quasi-)periodic structure of voiced speech signals
- Peaks on envelope curve are **formants**
- **Pitch** is perceptual value, F0 is physical
- F0 lie roughly in the **range 80 to 450 Hz**, where males have lower voices than females and children

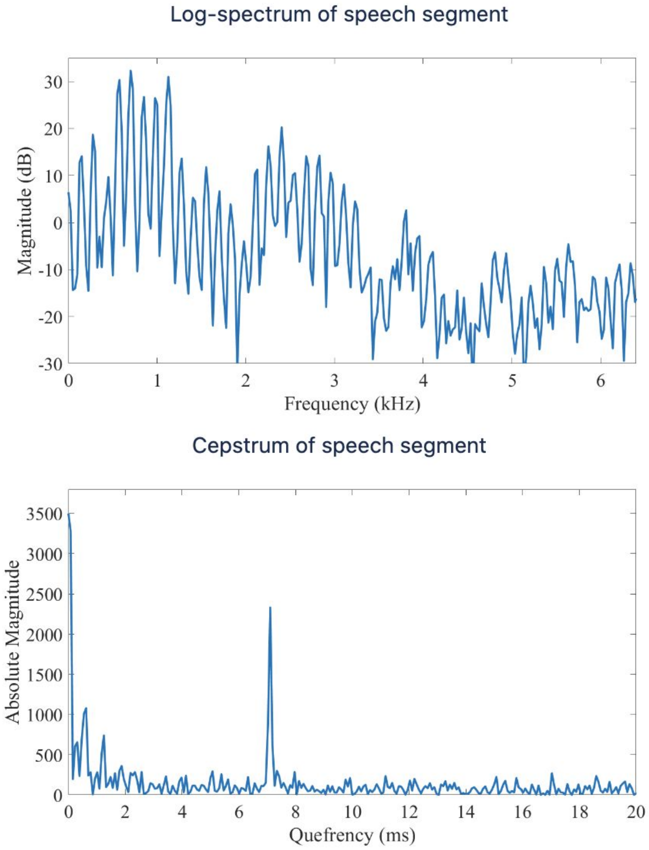


Cepstrum

- Fourier spectrum of voice has **periodic** structure

- Apply **DCT** (Discrete Cosine Transform) to spectrum and obtain **Cepstrum**

- **Peak** in Cepstrum should be located at $\dfrac{1}{F_0}$

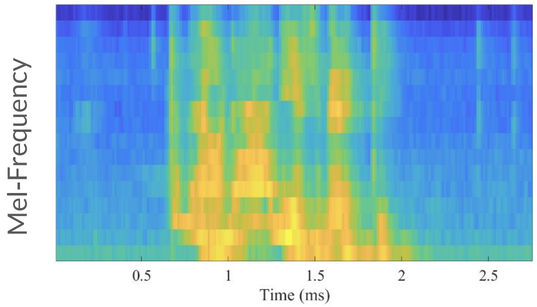


https://wiki.aalto.fi/display/ITSP/Cepstrum+and+MFCC

Mel-Frequency Cepstral Coefficients (MFCCs)

- Algorithm of acquiring MFCC:
  - Apply STFT to the signal
  - Apply mel filters
  - Take the log value
  - Apply DCT

Spectrogram after multiplication with mel-weighted filterbank



Corresponding MFCCs