FFT and Sound processing

1 марта 2023 г. 22:11

FFT (DFT)

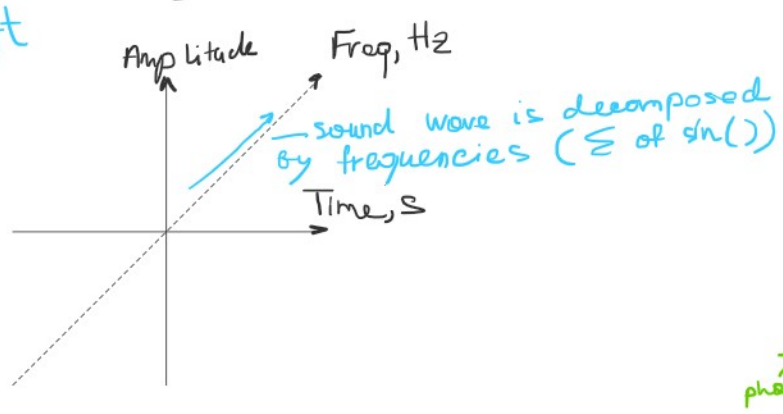
Signal → Wave (discrete)

Wave → (fft) → spectrum

Wave → (stft) → spectrogram

Two domains

1) Time domains
2) Frequency domain

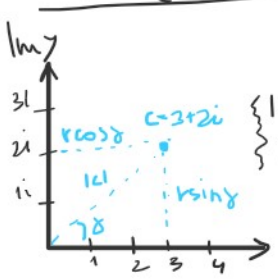


$\sin(2\pi(\underset{1}{f}\cdot t - \underset{2}{\varphi}))$ - formula of sin-wave

1 - frequency ($2\pi \cdot f = w$)

2 - phase

FT (with complex numbers and math)



$\{|c| = \sqrt{a^2+b^2} = r,\ \gamma = \operatorname{arctg}\frac{a}{b}\} \Rightarrow c = r(\cos\gamma + i\sin\gamma) = r\cdot e^{i\varphi}$ (3)

(dist from origin = magnitude); polar coordinates representation; direction

$e^{i\pi} + 1 = 0$ - Euler Identity

complex numbers recap.-

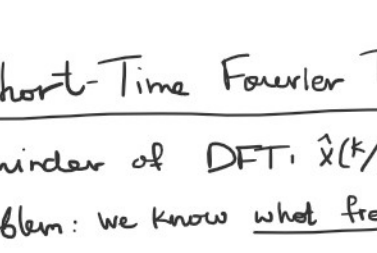
now for FFT.

how to go from (1),(2) to (3)?

Intuition: use magnitude and phase as polar coordinates.

$c_f = \frac{d_f}{\sqrt{2}} \cdot e^{-i2\pi\varphi_f}$ - complex FFT coefs.

($\frac{d_f}{\sqrt{2}} = |c|$; $\gamma = 2\pi\varphi_f$)



* - added for 2) rotation

Idea of FFT:

- compare signal with different sinusoids
- FFT(freq) → magnitude + phase
- high magnitude => sin ≈ signal

Choose freq.
⇓
Optimize phase
⇓
Calculate magnitude

$\varphi_f = \underset{p\in[0,1)}{\arg\max} \int s(t)\cdot\sin(2\pi(ft-\varphi))dt$ (1)

(φ_f - phase; $s(t)$ - signal; $\sin(2\pi(ft-\varphi))$ - chosen sinusoid)

$d_f = \max_{\varphi\in[0,1)} \int s(t)\cdot\sin(2\pi(ft-\varphi))dt$ (2)

(d_f - magnitude)

Fourier representation (ift)

$g(t) = \int c_f \cdot e^{i2\pi ft} df$

add up all weighted sinusoids and get original sound.

DFT case (what is actually going on)

$g(t) \mapsto x(n),\ n\in\mathbb{N}$

$t = n\cdot T$ (T - sampling rate)

$\hat{x}(f) = \sum_{n=0}^{N-1} x(n)\cdot e^{-i2\pi fn}$ - discrete version of $\int_{-\infty}^{+\infty}$

- consider n where freq ≠ 0
- compute for finite set of $\{f_1,\dots,f_n\}$, M = N - number of samples (allows ift and efficient)

$f = \frac{k}{N},\ k\in[0,M-1]=[0,N-1]$

$F(k) = \frac{k}{NT} = \frac{k\cdot sr}{N}$ - actual freq. value

BUT: redundancy - plot will be symmetrical around Nyquist fr. = $\frac{N}{2}$ > $F(N/2) = sr/2$. So we can throw everything after it.

Note on FT to FFT (very fast computing)
- $O(N^2) \to O(N\log N)$
- $N = 2^x$ ($x\in\mathbb{N}$)

$g(t): \mathbb{R}\to\mathbb{R}$ - original audio signal (time domain)

$\hat{g}(f): \mathbb{R}\to\mathbb{C}$ - FT, outputs a complex number (c_f)

$\hat{g}(f) = \int_{-\infty}^{+\infty} g(t)\cdot e^{-i2\pi ft} dt$

($g(t)$ - sound signal (amplitude at time t); $e^{-i2\pi ft}$ - phase term; $2\pi f$ ~ speed of unit circle rotation)

Interpretation of f - center of mass. The greater $|c_m|$ - the greater the alignment with true sinusoid.

* Since sound is discrete, instead of $\int$ we do Σ over all timestamps.

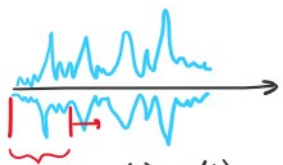
$d_f = \sqrt{2}\cdot|\hat{g}(f)|,\ \varphi_f = -\frac{\gamma_f}{2\pi}$

Short-Time Fourier Transform (signal → spectrogram)

reminder of DFT: $\hat{x}(k/N) = \sum_{n=0}^{N-1} x(n)\cdot e^{-i2\pi n\cdot\frac{k}{N}}$

Problem: we know what freq. are present, but don't know when. (cause they are averaged over time)

Solution: use windowing



window size - number of samples W() is applied to

frame size - number of samples in chunk

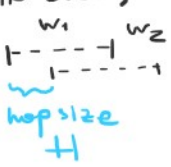
ws can be < then fs (then samples outside of ws will be 0-padded)

1) $x_w(t) = x(k)\cdot W(k)$ (W(k) - window function)

2) apply fft on window

1) windows are overlapping. Each window position is defined by hop size parameter.

However, windows are overlapping. Each window position is defined by hop size parameter.



$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH)\cdot w(n)\cdot e^{-i2\pi n\frac{k}{N}}$$

- phase thingy
- proxy time value, frame number
- N = frame size we are Σ over frame
- all samples from m frame.
- window function

DFT → 1-d vector, N complex FT

STFT → 2-d matrix, freq-bins x frames

$\#\text{freq-bins} = \frac{\text{framesize}}{2} + 1$ (related to Nyq. freq.)

$\#\text{frames} = \frac{\text{samples} - \text{framesize}}{\text{hopsize}} + 1$

Example:

signal – 10k samples

fs = ws = 1000

hs = 500

number of fr. bins = $\frac{1000}{2} + 1 = 501 \rightarrow (0, \frac{fs}{2})$

num. of frames = (10k − 1k)/500 + 1 = 19

STFT → (501, 19)

Parameters study

- fs – (512, 1024, .., 2^n)
  - fs↓ ⇒ freq.res↑ time res.↓ (time/freq trade off)
  - fs↑ ⇒ freq.res.↓ time res.↑
- hs (256, 512, .., 2^m), $m = (n-1, n-2, \dots) \sim hs = \frac{fs}{4}$
- w(n) – - butter, like Hann window

How to get to spectrogram?

$$Y(m,k) = |S(m,k)|^2$$

↖ real-valued matrix