Fouriertransform, Fouriercoefficents, Discrete Fourier Transform, Fast Fourier Transform

Lukas Thalhammer, Patrick Walter

Music Processing

Dec 10, 2020

The Fourier Transform

The wave Equation describes the performance of waves (e.g. sound waves).

$$\frac{\partial^2 u(x,t)}{\partial t^2} = c^2 \frac{\partial^2 u(x,t)}{\partial x^2} \tag{1}$$

To solve this equation in this form analytically might cause difficulties.

However, the equation can easily be solved be defining a Transformation $F : f(x) \to f(k)$ with the property

$$f'(k) = ikf(k) \tag{2}$$

The Equation then simplifies to

$$\frac{\partial^2 \hat{u}(k,t)}{\partial t^2} = -c^2 k^2 \hat{u}(k,t) \tag{3}$$

A transformation with such properties is the Fourier Transfomation:

$$\tilde{f}(k) = \int_{-\infty}^{\infty} e^{-ikx} f(x) dx \tag{4}$$

With backwards transformation

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} \tilde{f}(k) dk \tag{5}$$

Solving the transformed equation 3 gives

$$\hat{u}(k,t) = \hat{F}(k)e^{-ikct} + \hat{G}(k)e^{ikct} \tag{6}$$

The solution for the original equation 1 can then be found by applying the backwards transform

$$u(x,t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ikx}\hat{u}(k,t)dk \tag{7}$$

which gives

$$u(x,t) = F(x-ct) + G(x+ct) \tag{8}$$

Physical meaning

- Quantum Mechanics: position vs momentum
- Solid-state physics: direct lattice vs reciprocal lattice (xray diffrection on Cristals)
- acoustics: position/time vs frequency

Fourier Transform

Let's take a closer look at this - somewhat ugly - formula

$$\tilde{f}(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ikx} f(x) dx$$

by starting at the back:

$$\tilde{f}(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ikx} f(x) dx$$

$f(x)$ is our original signal, in dependence of x.

Euler Equation

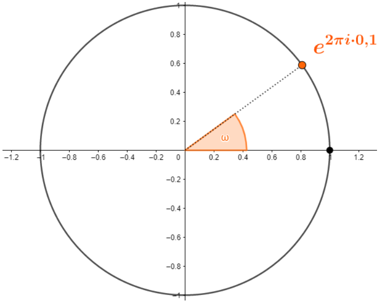
$$e^{i\pi} + 1 = 0 \tag{9}$$

We will further use the relation between exponential functions and trigonometrical functions:

$$e^{ix} = cos(x) + isin(x)$$

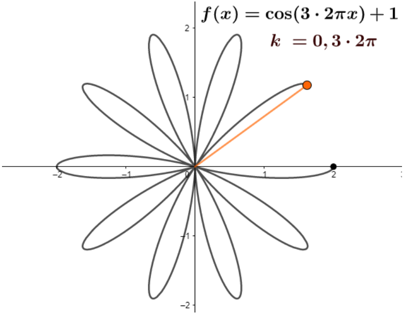
$$sin(x) = \frac{e^{-ix} - e^{ix}}{2i}$$

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



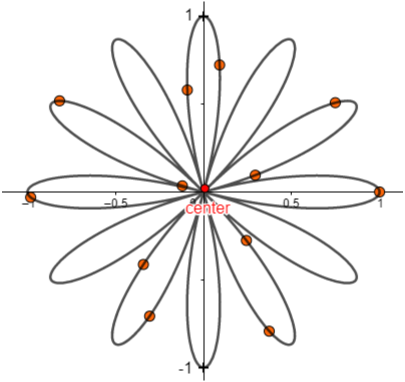
$$\tilde{f}(k) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ikx} f(x) dx$$

By multiplying these two parts, we "wrap" our signal around one point with cycling-frequency k.



We now want to find the center of mass of this wrapped signal, i.e. average it.
Intuitively, one would therefore take some values, sum them up and devide by the quantity of values. Something like this:

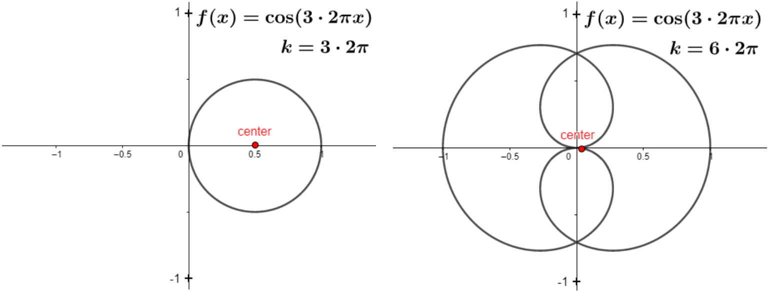
$$\frac{1}{N}\sum_{x=1}^{N} e^{-ikx} f(x)$$



As we are dealing with an infinity amount of point (a continuous signal), we take the integral and devide it by 2π to average our signal.

$$\tilde{f}(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-ikx} f(x) dx$$

If we now play around with k, we can see, why this transformation works.



Discritization

The Fourier Transform discribes the change of display of two continuos spaces.
Sometimes you might have systems, in wich one or both of the domains allow only discrete values.
This comes with limitations to the completeness of the transformation.

Fourier Series

Discritization of backwards transform:

$$f(x) = \int_{-\infty}^{\infty} e^{ikx}\tilde{f}(k)dk \rightarrow f(x) = \sum_{n=-\infty}^{\infty} e^{-ik_n x}\tilde{f}(k_n) \qquad (10)$$

$n \in \mathbb{Z}$

Due to the limitation of resolution in this domain results a limitation of functions, that can be discribed. These have to be periodic functions in a given period L. The calculation of the coefficents $\tilde{f}(k_n)$ then modifies:

$$\tilde{f}(k) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ikx}f(x)dx \rightarrow \tilde{f}(k_n) = \frac{1}{L}\int_{-L/2}^{L/2} e^{-ik_n x}f(x)dx \qquad (11)$$

Fourier Series in trigonometrical functions

When using $\sin$ and $\cos$ instead of e, we have to compute 2 coefficients per frequency, a_n and b_n. This simple works through interpreting the complex plane as a 2-dimensional real plane.

$$a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)dx$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} \cos(2\pi \cdot n \cdot x) \cdot f(x)dx$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} \sin(2\pi \cdot n \cdot x) \cdot f(x)dx$$

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos(2\pi \cdot n \cdot x) + b_n \sin(2\pi \cdot n \cdot x)$$

Discrete Fourier Transform

If both domains are discretized one obtains the Discrete Fourier Transform (DFT) given by:

$$\tilde{f}_n = \frac{1}{N} \sum_{j=-\frac{N}{2}}^{\frac{N}{2}-1} e^{-\frac{2\pi inj}{N}} f_j \tag{12}$$

$$f_j = \sum_{n=-\frac{N}{2}}^{\frac{N}{2}-1} e^{\frac{2\pi inj}{N}} \tilde{f}_n \tag{13}$$

Where N is the number of discrete values, you're given.

Implementing the DFT

The computation of the Fourier Transformation using numerical procedures in a naive attempt comes down to implementing DFT. This means that for a number of N values f_n you would need to calculate N transorfmed values $\tilde{f}_j$ each by calculating a sum of N values. This means, that the computing time is at least

$$T_{DFT} \propto N^2 \tag{14}$$

Recursion Formula

Let's try to make this better. Due to the periodicity in N of the Forurier coefficients, one might aswell wright the indices like this:

$$f_{-\frac{N}{2}}, \ldots, f_{\frac{N}{2}-1} \rightarrow f_0, \ldots, f_{N-1} \tag{15}$$

The sum resulting of this indexation can be split up in two sums, if N can be divided by 2.

$$\sum_{j=0}^{N-1} e^{\frac{2i\pi nj}{N}} f_j = \sum_{j'=0}^{\frac{N}{2}-1} e^{\frac{2i\pi n(2j')}{N/2}} f_{2j} + \sum_{j'=0}^{\frac{N}{2}-1} e^{\frac{2i\pi n(2j'+1)}{N/2}} f_{2j+1} \tag{16}$$

Fast Fourier Transform (FFT)

If $N = 2^m$ $(m \in \mathbb{N})$ one can do this splitting of the sum $m = \log_2 N$ times. These m "sums" only have one value which makes the calculation as trivial as $f_0 = \tilde{f}_0$.
So the computing time for each of the N values is then reduces to $\log_2 N$ operations, wich gives:

$$T_{FFT} \propto N \log_2 N \tag{17}$$

So we can now fourier transform any audio signal! Cool! But what does that mean?

Time domain to Frequency domain

We can change the representation of an audio signal:

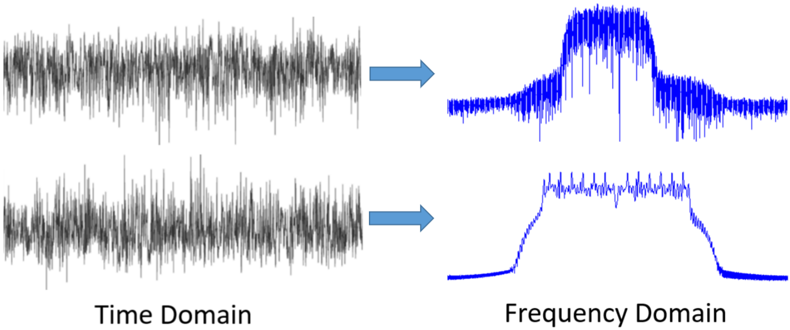


Figure: https://pysdr.org/_images/time_and_freq_domain_example_signals.png

How can we represent frequency AND time?

if we cut the audio signal in small "portions" in time

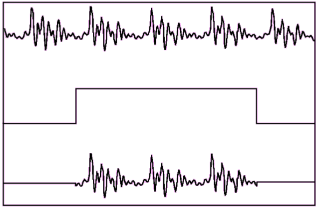


Figure: Balazs, Peter. Talk: Frame Theory and its Applications. Acoustics Research Institute, October 18, 2017

and fourier transform each of these protions, one can create a representation, that might look something like this:

Spectogram

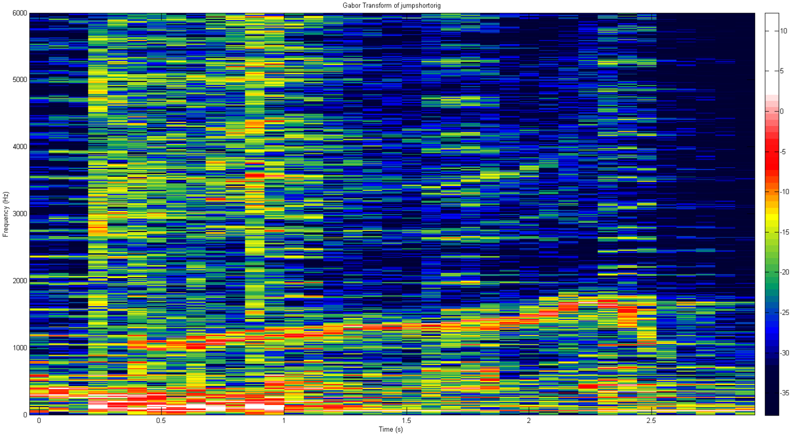
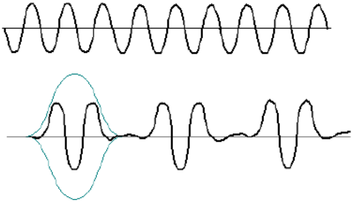


Figure: Balazs, 2017

Let's improve this!

Changing the window one might obtain better results.



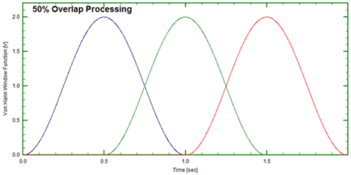


Figure: Balazs, 2017

better spectogramms

50% overlap:

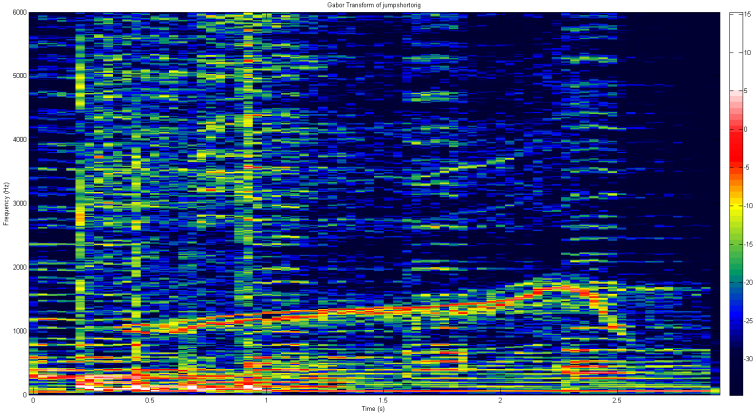


Figure: Balazs, 2017

better spectogramms

8 x overlap

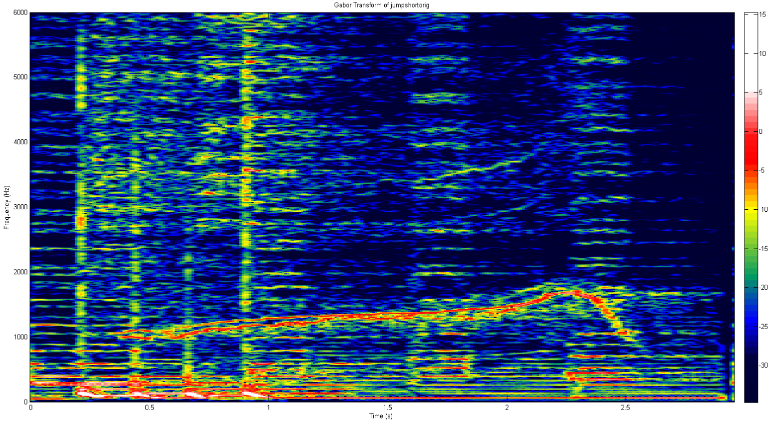


Figure: Balazs, 2017

Problems?

Thank you for your attention!