

Introduction to signal processing: spectral representation

E. Tikhonov and M. Sneps-Snepe

Abstract — The article introduces powerful common tool for the signals analysis – basics of a spectral analysis.

It focuses on decomposition into complex sine waves (harmonic sinusoids) called the Fourier transform.

Based on qualitative review of the natural phenomenon of visible light (that is the electromagnetic field oscillation in a working range of the eyes) decomposition into a clear colors in a prism, known as Newton's experiments, corresponding quantitative procedure of Fourier transform is announced. It translates the arbitrary signals specified in time (usually real, but not necessarily) into a set of complex sine waves in a mathematical form.

The reversibility of the transformation, similar to the second prism that collects light rays back into a single white beam, is briefly shown. For this purpose, one of the key mathematical constructions of the theory of signals (the Dirac delta function and its properties) is considered more detailed.

The phenomenon of negative frequencies is also discussed (which are not a fiction, but are characterized by standard propagating complex sinusoids twisted in different directions). It is shown that real physical signals have an excess in a spectrum (negative frequencies correspond to slightly modified positive ones), and they can be replaced with corresponding analytical ones.

All arguments are presented at a descriptive level and in mathematical form, using visualizations and in the form of MatLab scripts. This gives a more complete picture of the subject matter, allowing looking at that from various points of view.

Keywords — Fourier transform; spectrum; complex signals; complex sine waves; delta-function; negative frequencies; analytical representation; MatLab;

I. INTRODUCTION

The article continues the introduction to a signal processing tasks, actual for radio astronomy measurements and satellite data collection. The works are focused on applications that could be used in the "Ventspils International Radio Astronomy Center" of the Ventspils University of Applied Sciences.

The first paper [1] in series was dedicated to the basic concepts: complex numbers and corresponding complex signals as a change of these numbers, with a special focus on complex sine wave signal.

Based on these concepts this work offers representation of any appropriate signal as a set of complex sine waves (that

are convenient for processing and analytical analysis) in a frequency area, known as Fourier transform.

The rest of the paper is the following. In Section 2, principal and natural and best-known example of signal decomposition is given (passage of light through a prism). Section 3 suggests mathematical description of this procedure and Section 4 studies some proof details of the procedure, discussing the Dirac delta function (that will be used also in following works in a series). In Section 4 there are some reflections about negative frequencies of a spectrum and Section 5 shows redundancy of these frequencies for a real signals and a way to reduce it using corresponding analytical signal. All the Matlab codes (used as the examples and produced illustration figures to all parts) are collected in the Appendix.

II. SIGNAL AS A SET OF HARMONICS

It's a well-known phenomenon in physics that a sunlight passing through a suitable prism split into different visible color beams. More importantly, the second reversed passage through the prism collects them in white light again. Newton showed these processes in his experiments, but he analyzed these phenomena qualitatively [2].

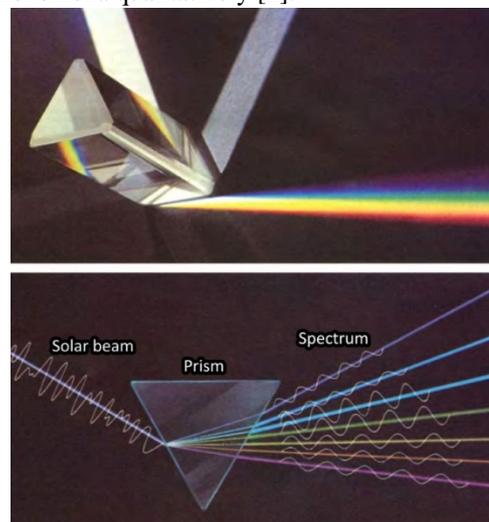


Fig 1. Decomposition of sunlight into the spectral components using the prism [3].

Clear colors are sine waves of the electromagnetic field, i.e. light coming out of the Sun during the processes occurring in it. Then they are distorted in atmosphere (each color with his own way) and are perceived by the eye in "working range" about from 380-400 nm (frequency 790-750 THz) to 760-780 nm (frequency 395-385 THz), also each with its own sensitivity by different specialized cells. But in general in this range all components are roughly at the same level.

Manuscript received March 5, 2019.

Eugene Tikhonov is with Ventspils University of Applied Sciences, 101a Inženieru Street, LV-3601, Ventspils, Latvia (e-mail: abava@abava.net).

Manfred Sneps-Snepe is with Ventspils University of Applied Sciences, 101a Inženieru Street, LV-3601, Ventspils, Latvia (e-mail: amanfredss@venta.lv).

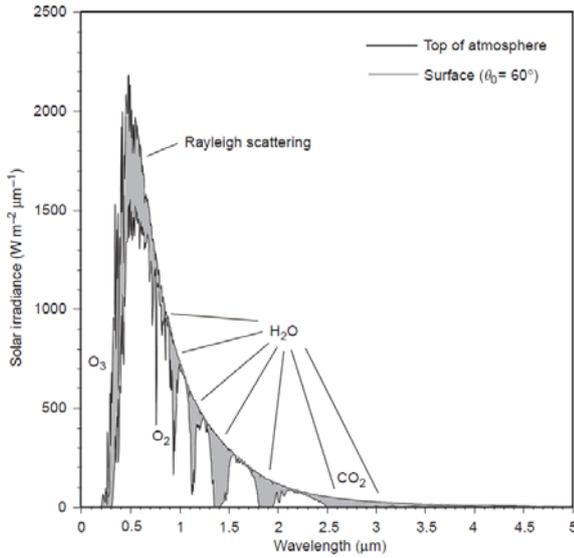


Fig 2. Solar spectrum at the top of the atmosphere and at the surface, for a solar zenith angle of 60° in a clear atmosphere. Absorption and scattering regions are indicated [4]. Visible light is from 0.38 to $0.7 \mu\text{m}$ that corresponds to the most active part of the Sun spectrum.

Similarly, as in a prism, the dispersion (dependence of the refraction on the frequency of the component of the field oscillation) leads to the deviation at their own angle for each color (corresponding to a specific frequency), actually visualizing the following general principle: the original signal can be divided and be presented as a sum of harmonic (sine) components, and in this sense signal consist of them.

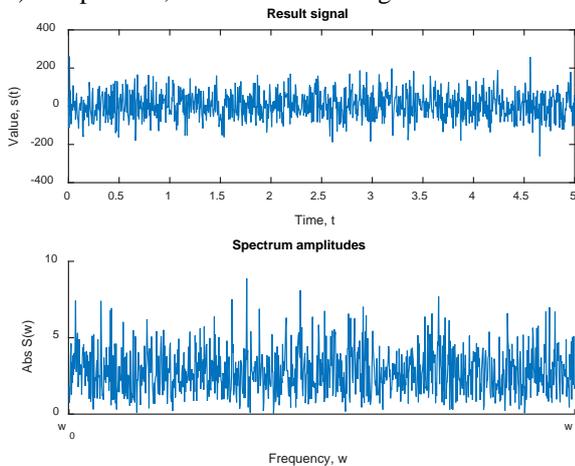


Fig 3. Example result of summing $N=10000$ single-amplitude sine waves with uniform distributed random frequencies in range $0..2000\pi$ and all possible phases. Result is almost roughly same level for all components and very noise-like fast changing signal. See Matlab A code in Appendix.

In telecommunications, the signal also is a change in the propagating electromagnetic field waves with a complex form, which determines the information — the numerical data we want to transmit. Therefore, an approach with decomposition into sinusoidal components is also applicable to it.

III. FOURIER TRANSFORM

Quantitatively, i.e. mathematically, the decomposition is written in the form of a **Fourier transform** [5], and not only

for light or electromagnetic signal, but also for an abstract function of time (or even another parameter) – the signal in general. This set of harmonic is called **spectrum** of the signal.

For the signal $s(t)$ which could be integrated on the whole time line (it is correct, for example for all real continuous signals) Fourier transform is:

$$\begin{aligned} \mathcal{F}(s) = S(\omega) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} s(t) \\ &\quad \times [\cos(\omega t) - j \sin(\omega t)] dt \end{aligned}$$

This function takes only frequency depending complex values. That is the spectrum is a set of complex sine waves, and the values of spectrum determine the amplitude and initial phase of these waves.

Initial signal $s(t)$ could be received back from the spectrum using inverse Fourier transform (it shows how to assemble back signal from complex sine waves):

$$\begin{aligned} \mathcal{F}^{-1}(S) = s(t) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} d\omega \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} S(\omega) (\cos(\omega t) + j \sin(\omega t)) d\omega \end{aligned}$$

Note: Multipliers before integrals may be different, but their multiplication must be equals to $\frac{1}{2\pi}$.

Another important remark is that Fourier transform (and inverse Fourier transform) is correct not only to a real signals, but for appropriate complex signals also.

The process of restoring the signal follows from definition of inverse transform and properties of a specific abstraction – delta-function that will be discussed further:

$$\begin{aligned} \mathcal{F}^{-1}(\mathcal{F}(s)) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{\omega=-\infty}^{\infty} \left(\int_{\tau=-\infty}^{\infty} s(\tau) e^{-j\omega\tau} d\tau \right) e^{j\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{\omega=-\infty}^{\infty} \left(\int_{\tau=-\infty}^{\infty} s(\tau) e^{j\omega(t-\tau)} d\tau \right) d\omega \\ &= \frac{1}{2\pi} \int_{\tau=-\infty}^{\infty} \left(\int_{\omega=-\infty}^{\infty} s(\tau) e^{j\omega(t-\tau)} d\omega \right) d\tau \\ &= \int_{\tau=-\infty}^{\infty} s(\tau) \left(\underbrace{\frac{1}{2\pi} \int_{\omega=-\infty}^{\infty} e^{j\omega(t-\tau)} d\omega}_{\delta(t-\tau)} \right) d\tau \\ &= \int_{\tau=-\infty}^{\infty} s(\tau) \underbrace{\delta(t-\tau)}_{\delta(x)=\delta(-x)} d\tau \\ &= \underbrace{\int_{\tau=-\infty}^{\infty} s(\tau) \delta(\tau - t) d\tau}_{\text{Sampling}} = s(t) \end{aligned}$$

In a philosophical sense, this inverse transformation is

analogous to Newton’s second prism, which gathers all the “clear” sinusoidal components back into a single signal.

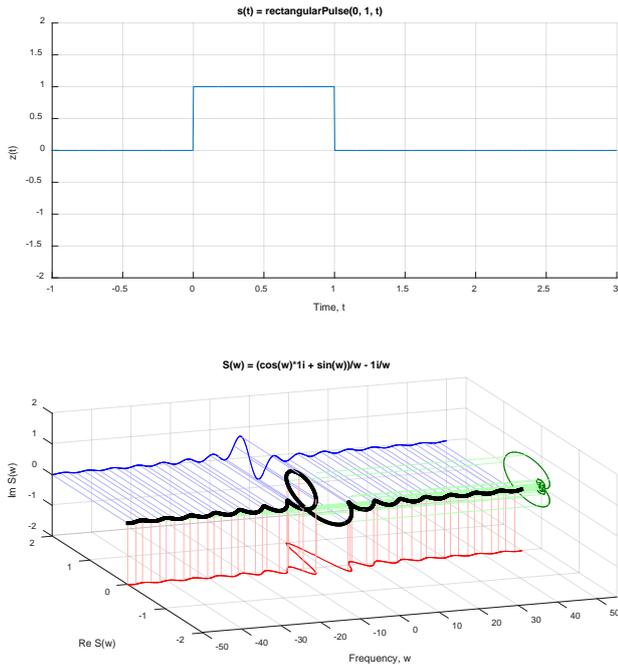


Fig. 4. Fourier transform example for the signal $s(t) = \text{rectangular}(0, 1, t)$. See Matlab B code in Appendix.

IV. DIRAC DELTA-FUNCTION

The key point in signal restoring using inverse Fourier transform is using the property of delta-function, suggested by the physicist Paul Dirac [6]. This function also is used to build the theoretical basis of discrete signal processing theory, and therefore is often is referred to as a **sampling function**.

Dirac delta function (δ -function) is a generalized function. Frivolous it could be presented like a infinite high and zero-thickness point:

$$\delta(x) = \begin{cases} \infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

More exactly delta function is defined as a limit of functions sequence that provides indicated properties (height, thickness, area):

$$\delta(x) = \lim_{\varepsilon \rightarrow 0} f(t, \varepsilon) = \begin{cases} \infty, & x = 0 \\ 0, & x \neq 0 \end{cases}$$

$$\lim_{\varepsilon \rightarrow 0} \int_{-\infty}^{\infty} f(t, \varepsilon) dt = 1$$

So, the function can be represented as infinitely narrow and infinitely high rectangle (or any other signal – that’s important), whose area is finite and equals to 1. In a sense, delta function is a set of interchangeable sets of all sorts of suitable signal sequences.

For example function sequence $f(n, x) = \frac{\sin nx}{\pi x}$ (area under that is always 1) limit is delta-function could be visualized using Matlab.

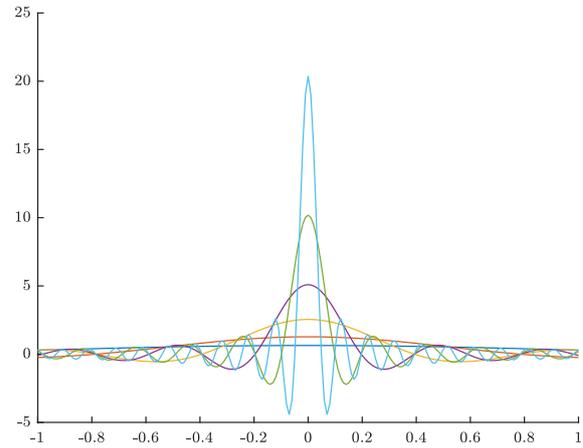


Fig. 5. Function $f(n, x) = \frac{\sin nx}{\pi x}$ iterations ($n = 2, 4, 8, 16, 32, 64$). Form of a curve approaches to a delta-function. See Matlab C code in Appendix.

A. Important example of delta-function realization

For example, following form was used for Fourier’s transform reversible proof:

$$\delta(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega t} d\omega = f(t)$$

Exploring specified function we could try to modify it the way to represent it as a limit of integral sequence, which could be found exactly [7].

Define the function:

$$I(\varepsilon) = \int_{-\infty}^{\infty} e^{j\omega t} \times e^{-\frac{\varepsilon\omega^2}{2}} d\omega$$

Obviously it tends to $f(t)$ as a limit:

$$f(t) = \lim_{\varepsilon \rightarrow 0} I(t, \varepsilon) = \lim_{\varepsilon \rightarrow 0} \int_{-\infty}^{\infty} e^{j\omega t} \times e^{-\frac{\varepsilon\omega^2}{2}} d\omega = \int_{-\infty}^{\infty} e^{j\omega t} d\omega$$

Let’s do some mathematical tricks.

Modifying exponent rate of $I(\varepsilon)$:

$$j\omega t - \frac{\varepsilon\omega^2}{2} = -\frac{\varepsilon}{2} \left(\omega^2 - 2\frac{jt}{\varepsilon}\omega \right)$$

$$= \frac{-\varepsilon}{2} \left(\frac{\omega^2 - 2\frac{jt}{\varepsilon}\omega + \left(\frac{jt}{\varepsilon}\right)^2}{\left(\omega - \frac{jt}{\varepsilon}\right)^2} - \left(\frac{jt}{\varepsilon}\right)^2 \right)$$

$$= -\frac{\varepsilon}{2} \left(\omega - \frac{jt}{\varepsilon} \right)^2 - \frac{t^2}{2\varepsilon}$$

So, in respect that classical exponential integral:

$$\int_{-\infty}^{\infty} e^{-ax^2} dx = \sqrt{\pi/a}$$

get another representation of I :

$$\begin{aligned}
 I(t, \varepsilon) &= \int_{-\infty}^{\infty} e^{-\frac{\varepsilon}{2}(\omega - \frac{jt}{\varepsilon})^2} \times e^{-\frac{t^2}{2\varepsilon}} d\omega \\
 &= e^{-\frac{t^2}{2\varepsilon}} \int_{-\infty}^{\infty} e^{-\frac{\varepsilon}{2}(\omega - \frac{jt}{\varepsilon})^2} d\omega \\
 &= \left\{ \begin{array}{l} v = \sqrt{\frac{\varepsilon}{2}} \left(\omega - \frac{jt}{\varepsilon} \right) \\ d\omega = \sqrt{\frac{2}{\varepsilon}} dv \end{array} \right\} \\
 &= \sqrt{\frac{2}{\varepsilon}} e^{-\frac{t^2}{2\varepsilon}} \int_{-\infty}^{\infty} e^{-v^2} dv = \sqrt{\frac{2\pi}{\varepsilon}} e^{-\frac{t^2}{2\varepsilon}}
 \end{aligned}$$

When $\varepsilon \rightarrow 0$ this function tends to $2\pi\delta(t)$:

1. $\lim_{\varepsilon \rightarrow 0} I(0, \varepsilon) = \lim_{\varepsilon \rightarrow \infty} \sqrt{\frac{2\pi}{\varepsilon}} = \infty$
2. $\lim_{\substack{\varepsilon \rightarrow 0 \\ t \neq 0}} I(t, \varepsilon) = 0$
3. $\lim_{\varepsilon \rightarrow 0} \int_{-\infty}^{\infty} I(t, \varepsilon) dt = \lim_{\varepsilon \rightarrow 0} \sqrt{\frac{2\pi}{\varepsilon}} \times \sqrt{\frac{\pi}{1/2\varepsilon}} = 2\pi$

As a limit of $\lim_{\varepsilon \rightarrow 0} I(t, \varepsilon)$:

$$\frac{f(t)}{2\pi} = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{j\omega t} d\omega = \delta(t)$$

So, the initial assumption of delta-function representation form permissibility was correct and that was really delta-function in restoring the signal in inverse Fourier transform. But what was the property of delta-function used then?

B. Sampling property of delta-function

The key property of delta-function is **sampling** or **filtering**:

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$$

Because integral expression is equals to 0 if $x \neq x_0$, so for any $\varepsilon > 0$:

$$I(x) = \int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = \int_{x_0 - \varepsilon}^{x_0 + \varepsilon} f(x)\delta(x - x_0)dx$$

Generalized mean theorem (Cauchy's theorem) states for integrable $f(x)$ and $g(x)$ (in our case $g(x) = \delta(x)$) and continuous $f(x)$:

$$\begin{aligned}
 \int_{x_0 - \varepsilon}^{x_0 + \varepsilon} f(x)\delta(x - x_0)dx &= f(\theta) \int_{x_0 - \varepsilon}^{x_0 + \varepsilon} \delta(x - x_0)dx \\
 &= f(\theta), \text{ where } \theta \in [x_0 - \varepsilon; x_0 + \varepsilon]
 \end{aligned}$$

So limit value of $I(x)$ is:

$$\lim_{\varepsilon \rightarrow 0} I(x) = f(x_0)$$

Finally it proves that inverse Fourier transform allows to restore initial signal from the spectrum, that indicates the parameters of its sine waves components.

V. NEGATIVE FREQUENCIES

In spectrum according to Fourier transform, the frequencies could be positive either negative: we should take into account all of them to restore the signal.

Negative frequency value is not something “virtual” or “imaginary”: it only informs that full phase of a signal decreases in time (not increases) with speed $|\omega| > 0$. It's not even very important: increasing angle/phase direction is arbitrary. For example, we could accept clockwise rotating as the increasing direction and counter-clockwise rotating as the decreasing or vice versa if needed.

Mathematical explanation is almost obviously: if $\omega = \frac{d\varphi(t)}{dt} < 0$, that means $d\varphi(t) = \omega dt < 0$ while $dt > 0$ (time moving forward). Therefore, when the time for the negative frequency increases as usual, the phase will decrease.

Essential is that fact that in spectrum both complex sine waves with phase increasing (positive frequency) and phase decreasing (negative frequency) always exist.

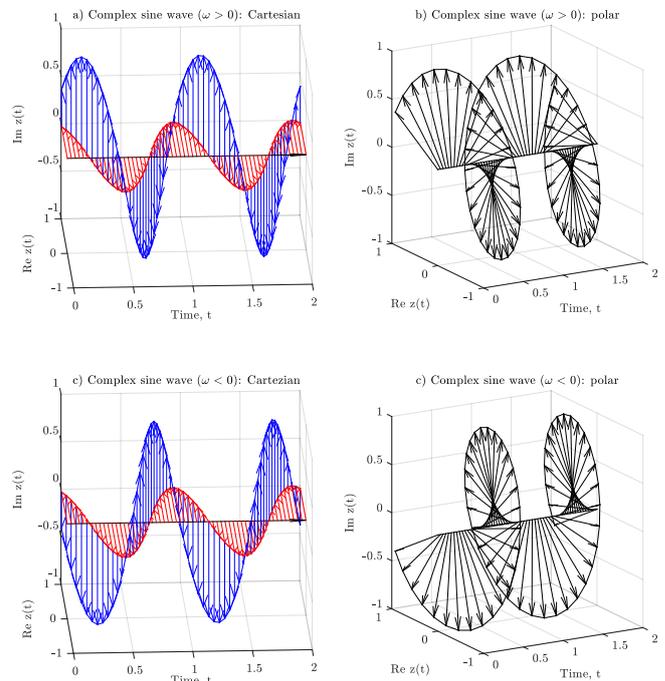


Fig. 6. Complex sine waves with positive frequency $z^+(t) = e^{j\frac{\pi}{8}} \times e^{j\omega t}$ and negative frequency $z^-(t) = e^{j\frac{\pi}{8}} \times e^{-j\omega t}$ ($\omega = 2\pi$) are left-handed and right-handed spirals respectively. See Matlab D code in Appendix.

VI. ANALYTICAL SIGNAL

Fourier transform allows find a spectrum not only for real signals, but also for complex ones.

But for the real (not complex) signal $s(t)$ spectrum is superfluous:

$$\begin{aligned}
 S(-\omega) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} s(t)e^{-j(-\omega)t} dt \\
 &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} s(t) \\
 &\quad \times (\cos(\omega t) + j \sin(\omega t)) dt = \\
 &= \text{Re } S(\omega) - \text{Im } S(\omega) = S^*(\omega)
 \end{aligned}$$

That means: the part of spectrum of a real signal with negative frequencies is exactly the same as complex

conjugate for positive frequencies and have no new information. When we sum them in reverse Fourier transform, these parts will destroy the opposite imaginary parts of each other so that only the real part remains – the result will be the real signal as expected.

Could we ignore this redundant information given the knowledge that the signal is real?

The signal, that have the same information as initial real (t) , could be calculated using only positive frequencies:

$$\sigma(t) = \frac{1}{\sqrt{2\pi}} \int_0^{\infty} S(\omega) e^{j\omega t} d\omega$$

Complex conjugation of $\sigma(t)$ corresponding to a signal with only negative frequencies:

$$\begin{aligned} \sigma^*(t) &= \frac{1}{\sqrt{2\pi}} \int_0^{\infty} S^*(\omega) e^{j\omega t} d\omega = \frac{1}{\sqrt{2\pi}} \int_0^{\infty} S(-\omega) e^{j\omega t} d\omega \\ &= \left\{ \begin{array}{l} \alpha = -\omega \Rightarrow d\omega = -d\alpha \\ \tau = -t \Rightarrow \omega t = \alpha \tau \\ \infty \rightarrow -\infty \\ 0 \rightarrow 0 \end{array} \right\} = \\ &= -\frac{1}{\sqrt{2\pi}} \int_0^{\infty} S(\alpha) e^{j\alpha \tau} d\alpha \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^0 S(\alpha) e^{j\alpha \tau} d\alpha \end{aligned}$$

That means $s(t)$:

$$\begin{aligned} s(t) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} d\omega = \sigma(t) + \sigma^*(t) \\ &= \text{Re } \sigma(t) + j \times \text{Im } \sigma(t) + \text{Re } \sigma^*(t) + j \\ &\quad \times \text{Im } \sigma^*(t) \\ &= \text{Re } \sigma(t) + j \times \text{Im } \sigma(t) + \text{Re } \sigma(t) - j \\ &\quad \times \text{Im } \sigma(t) = 2 \times \text{Re } \sigma(t) \end{aligned}$$

is a real part of $z(t)$ with half complex amplitudes only positive frequencies in spectrum, $s(t) = \text{Re } z(t)$, where:

$$z(t) = \frac{\sigma(t)}{2} = \frac{1}{\sqrt{2\pi}} \int_0^{\infty} \frac{S(\omega)}{2} e^{j\omega t} d\omega$$

In other words, we use $z(t)$ instead of $s(t)$ and always could return to initial real $s(t)$ getting the real part of it. This function is called **analytical representation** of a real-valued function facilitates many mathematical manipulations.

Note: more exactly for frequency $\omega = 0$ complex amplitude value is $Z(0) = \frac{S(0)}{4}$, not $\frac{S(0)}{2}$ (because it is included in Z and Z^* at the same time):

$$Z(\omega) \stackrel{\text{def}}{=} \begin{cases} \frac{S(\omega)}{2}, & \omega > 0 \\ \frac{S(\omega)}{4}, & \omega = 0 \\ 0, & \omega < 0 \end{cases}$$

Can be shown that $z(t) = s(t) + j \left(\frac{1}{\pi t} * s(t) \right)$, where $*$ is the symbol of convolution:

$$(f * g)(\tau) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(t) g(\tau - t) dt$$

For example, analytical representation for sine wave $s(t) = A \cos(\omega t + \varphi_0)$ will be $z(t) = A e^{j(\omega t + \varphi_0)} = A \times$

$e^{j\varphi_0} \times e^{j\omega t} = Z \times e^{j\omega t}$, what corresponds to Euler's formula and previous conclusions.

For another example for rectangular pulse signal $s(t) = \text{rect}(a, b, t) = \begin{cases} 1, & a \leq t \leq b \\ 0, & \text{else} \end{cases}$:

$$\begin{aligned} z(t) &= s(t) + j \left(\frac{1}{\pi t} * s(t) \right) = s(t) + \frac{j}{\pi} \int_{-\infty}^{\infty} s(\tau) \frac{1}{(t - \tau)} d\tau \\ &= s(t) + \frac{j}{\pi} \int_a^b \frac{d\tau}{(t - \tau)} \\ &= s(t) + \frac{j}{\pi} \ln \left(\left| \frac{a - t}{b - t} \right| \right) \end{aligned}$$

So, analytical representation is not the same as the signal, but we could easily restore initial real signal from it with an advantage of only positive frequencies without redundancy. For that we could get real part of signal $z(t)$, or reconstruct $S(\omega)$ spectrum (with the rule: $S(\omega) = \begin{cases} Z(\omega), & \omega \geq 0 \\ Z^*(-\omega), & \omega < 0 \end{cases}$) and then find $s(t)$ from $S(\omega)$.

Or in other words, analytical signal is a way to turn real signal to a complex one without redundancy.

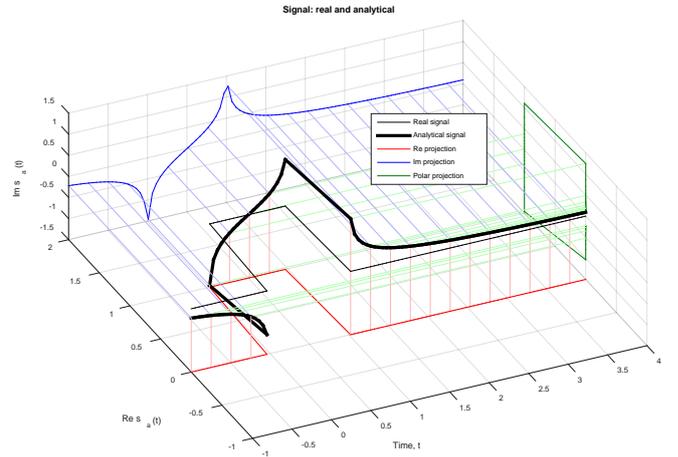


Fig. 7. Real signal (gray thin) and corresponding analytical (black thick) with real, imaginary, polar projections. See Matlab E code in Appendix.

VII. OTHER SPECTRUM TRANSFORMS

Fourier transform is not the only one way to obtain the spectrum of a signal.

Different generalizations and alternatives, that extend the applicability of the method are also available, for example Fourier–Stieltjes transform (for multiple-dimensional signals), locally compact abelian group (used other forms of spectrum components signals), Gelfand transform, wavelets and so on.

Also there are multiple analogous integral transforms, that are more convenient for specific cases, for example, Hartley transform (that transforms real-valued functions to real-valued functions), Laplace transform (that used complex sine waves with changing amplitudes), Mellin transform (that used not the “e” as a base of transform core) and many others.

But the Fourier transform is the most popular one not only because was presented historically first but also

because it's quite simple and close to naturally (and so, easy to understand), direct and reverse procedure are very similar, signal convolution correspond to spectrum multiplication (and wise versa), basic signals (sine waves) moves to basic signals (sine waves) and many other convenient properties.

That's why this classical Fourier spectrum is most popular and is used as a "default" spectrum.

VIII. CONCLUSION

The basic concept tool of spectrum representation as a part of signal processing analysis was discussed, namely, the complex mathematical procedures, details and their meaning in general.

In the following parts in series some technologies based on spectrum analysis are planned to discuss in detail: information (digital) signal, modulations, spectrum modifications (like Direct Sequence Spread Spectrum, Frequency Hopping, Time Hopping, Chirp Spread), etc.

APPENDIX

A. Matlab code: sine wave summation (see Figure 3)

```
%% Sine summations
N = 10000;           % number of waves
t0 = 0;             % start plotting time
t1 = 5;             % end plotting time
w0 = 0;             % min frequency
w1 = 2*pi*1000;     % max frequency, 1KHz

accuracy = 1000;

%% Processing
s = @(t, w, p) sin(w*t+p); % sine wave function
o = @(t) zeros(size(t));   % zero-vector

dw = w1-w0;
t = linspace(t0, t1, accuracy); % t in [t0..t1]
spectrum = zeros(1, accuracy);

s_sum = o(t);
for i = 1:N           % sine waves cycle
% uniformly distributed random in range [w0..w1]
    w = w0+dw*rand();
% uniformly distributed random in range [0..2 pi]
    p = 2*pi*rand();
    wi = 1+round((w-w0)/dw*(accuracy-1));
    spectrum(wi) = spectrum(wi)+exp(j*p);
    signal = s(t, w, p);
    s_sum = s_sum + signal;
end

%% Result signal
subplot(2,1,1);
plot(t, s_sum);
title('Result signal');
xlabel('Time, t');
ylabel('Value, s(t)');

%% Components amplitude summation with phases
subplot(2,1,2); hold on;
title('Spectrum amplitudes');
xlabel('Frequency, w');
ylabel('Abs S(w)');
set(gca, 'XTick', [1; accuracy]);
```

```
set(gca, 'XTickLabel', ['w_0'; 'w_1']);
plot(abs(spectrum));
```

B. Matlab code: rectangle signal spectrum (see Fig. 4)

```
figure('units', 'normalized', 'outerposition', [0
0 0.5 1]);

%% Parameters of rectangle function
a = 0;           % start time
b = 1;           % end time

N_counts = 1000; % accuracy
scale=2;         % plot scaling

w_range = linspace(-50, 50, N_counts);
t_range = linspace(a-1, a+b*3, N_counts);

syms t;         % t - symbolic variable
% f - symbolic rectangular function of t
f = rectangularPulse(a, b, t);

%% Processing
o = @(t) zeros(size(t)); % zero-vector
e = @(t) ones(size(t));  % one-vector

% F = symbolic expression, Fourier function
F = fourier(f);

% Evaluate symbolics
for i=1:length(w_range); % spectrum
    w=w_range(i);
    S(i)=eval(F);
end
Re = real(S);
Im = imag(S);

for i=1:length(t_range); % function
    t=t_range(i);
    s(i)=eval(f);
end

%% Display results function
subplot(2,1,1); grid on; hold on;
xlim([a-1 a+b*3]); ylim([-scale scale]);
plot(t_range, s);

xlabel('Time, t'); ylabel('z(t)');
title(['s(t) = ' char(f)]); % convert to string

%% Display spectrum results
subplot(2,1,2);
xlim([-50 55]);
ylim([-scale scale]);
zlim([-scale scale]);
grid on; hold on; view(-20, 40);

% S(w)
plot3(w_range, Re, Im, 'LineWidth', 3, 'Color',
'k');

% Re S(w) projection
plot3(w_range, Re, -scale*e(w_range), 'Color',
'r');

% Im S(w) projection
plot3(w_range, scale*e(w_range), Im, 'Color',
'b');
```

```

% Polar S(w) projection
plot3(55*e(w_range), Re, Im, 'Color', [0 0.5 0]);

xlabel('Frequency, w');
ylabel('Re S(w)');
zlabel('Im S(w)');
title(['S(w) = ' char(F)]); % convert to string

%% Design features
% decimate w_range points 4 projection
p = w_range(1:20:end);
Re = real(S(1:20:end));
Im = imag(S(1:20:end));

%% Re, Im projection lines

% Real projection lines
quiver3(p, Re, -e(p)*scale, o(p), o(p),
Im+e(p)*scale, 0, 'Color', [1 0.7 0.7],
'ShowArrowHead', 'off');

% Imag projection lines
quiver3(p, e(p)*scale, Im, o(p), Re-e(p)*scale,
o(p), 0, 'Color', [0.7 0.7 1], 'ShowArrowHead',
'off');

% Polar projection lines
p = w_range(1:20:end); % decimate 4 projection
Re = real(S(1:20:end));
Im = imag(S(1:20:end));

quiver3(p, Re, -Im, 55*e(p)-p, o(p),o(p), 0,
'Color', [0.7 1 0.7], 'ShowArrowHead', 'off');

```

C. Matlab code: delta-function sequence (Figure 5)

```

%% Delta-limit for fn = sin(nx)/(pi*x)
hold on;

fn = @(t, n) sin(n*t)./(pi*t); % function sequeuse

t=-1:0.01:1; % timeline
t(find(t==0))=0.0000001; % zero-division fix

for n = 1:6
    plot(t, fn(t, 2^n));
end

```

D. Matlab code: negative frequencies (Figure 6)

```

% Negative frequencies
figure('units', 'normalized', 'outerposition', [0
0 0.5 1]) % fullscreen/2
plotArrow3 = @(x, y, z, varargin)
quiver3(x(1),y(1),z(1), x(2)-x(1), y(2)-y(1),
z(2)-z(1), 0, varargin{:}); % arrow x1->x2

%% Constant definitions
w = 2*pi; % angle velociiry
ph0 = pi/8; % initial phase
A = 1; % amplitude

t_max = 2; % time to plot
N_counts = 60; % accuracy

%% Signal
t = linspace(0, t_max, N_counts); % timeline
o = zeros(size(t)); % zero-vector

sine_c = A*exp(j*(w*t+ph0)); % complex s-wave
sine_r = real(sine_c);
sine_i = imag(sine_c);

```

```

%% Figure a: real + image
subplot(2, 2, 1); hold on; grid on;
axis([0 t_max -1 1 -1 1]); view(-3, 20);
title('a) Complex sine wave ( $\omega > 0$ ): Cartesian');
xlabel('Time, t');
ylabel('Re z(t)');
zlabel('Im z(t)');

% real part
plot3(t, sine_r, o, 'r');

% real + imaginary part
plot3(t, sine_r, sine_i, 'b');

% arrow field: real part
quiver3(t, o, o, o, sine_r, o, 0, 'r');

% arrow field: real + imaginary part
quiver3(t, sine_r, o, o, o, sine_i, 0, 'b');

% axe line
plotArrow3([0 t_max], [0 0], [0 0], 'Color', 'k');

%% Figure b: real+image polar
subplot(2, 2, 2); hold on; grid on;
axis([0 t_max -1 1 -1 1]); view(-30, 15);
title('b) Complex sine wave ( $\omega > 0$ ): polar');
xlabel('Time, t');
ylabel('Re z(t)');
zlabel('Im z(t)');

% sine curve
plot3(t, sine_r, sine_i, 'k');

% arrow fields
quiver3(t, o, o, o, sine_r, sine_i, 0, 'k');

% axe line
line([0 t_max], [0 0], [0 0], 'Color', 'k');

%% Figure c: real - image
subplot(2, 2, 3); hold on; grid on;
axis([0 t_max -1 1 -1 1]); view(-3, 20);
title('c) Complex sine wave ( $\omega < 0$ ): Cartezian');
xlabel('Time, t');
ylabel('Re z(t)');
zlabel('Im z(t)');

% real part
plot3(t, sine_r, o, 'r');
% real - imaginary part
plot3(t, sine_r, -sine_i, 'b');

% arrow field: real part
quiver3(t, o, o, o, sine_r, o, 0, 'r');

% arrow field: real - imaginary part
quiver3(t, sine_r, o, o, o, -sine_i, 0, 'b');

% axe line
plotArrow3([0 t_max], [0 0], [0 0], 'Color', 'k');

%% Figure d: real - image polar
subplot(2, 2, 4); hold on; grid on;
axis([0 t_max -1 1 -1 1]); view(-30, 15);
title('c) Complex sine wave ( $\omega < 0$ ): polar');
xlabel('Time, t');

```

```
ylabel('Re z(t)');
xlabel('Im z(t)');

% sine curve
plot3(t, sine_r, -sine_i, 'k');

% arrow fields
quiver3(t, o, o, o, sine_r, -sine_i, 0, 'k');

% axe line
line([0 t_max], [0 0], [0 0], 'Color', 'k');
```

E. Matlab code: analytical signal (Figure 7)

```
% Negative frequencies
figure('units', 'normalized', 'outerposition', [0
0 0.5 1]) % fullscreen/2
plotArrow3 = @(x, y, z, varargin)
quiver3(x(1),y(1),z(1), x(2)-x(1), y(2)-y(1),
z(2)-z(1), 0, varargin{:}); % arrow x1->x2

%% Constant definitions
w = 2*pi; % angle
velocityr
ph0 = pi/8; % initial
phase
A = 1; % amplitude

t_max = 2; % time to plot
N_counts = 60; % accuracy

%% Signal
t = linspace(0, t_max, N_counts); % timeline
o = zeros(size(t)); % zero-vector

sine_c = A*exp(j*(w*t+ph0)); % complex sin
wave
sine_r = real(sine_c);
sine_i = imag(sine_c);

%% Figure a: real + image
subplot(2, 2, 1); hold on; grid on; axis([0 t_max
-1 1 -1 1]); view(-3, 20);
title('a) Complex sine wave ( $\omega > 0$ ):
Cartesian');
xlabel('Time, t'); ylabel('Re z(t)'); zlabel('Im
z(t)');

plot3(t, sine_r, o, 'r');
% real part
plot3(t, sine_r, sine_i, 'b');
% real + imaginary part

quiver3(t, o, o, o, sine_r, o, 0, 'r');
% arrow field: real part
quiver3(t, sine_r, o, o, o, sine_i, 0, 'b');
% arrow field: real + imaginary part
plotArrow3([0 t_max], [0 0], [0 0], 'Color',
'k');% axe line

%% Figure b: real+image polar
subplot(2, 2, 2); hold on; grid on; axis([0 t_max
-1 1 -1 1]); view(-30, 15);
title('b) Complex sine wave ( $\omega > 0$ ): polar');
xlabel('Time, t'); ylabel('Re z(t)'); zlabel('Im
z(t)');

plot3(t, sine_r, sine_i, 'k');
% sine curve

quiver3(t, o, o, o, sine_r, sine_i, 0, 'k');
```

```
% arrow fields
line([0 t_max], [0 0], [0 0], 'Color', 'k');
% axe line

%% Figure c: real - image
subplot(2, 2, 3); hold on; grid on; axis([0 t_max
-1 1 -1 1]); view(-3, 20);
title('c) Complex sine wave ( $\omega < 0$ ):
Cartezian');
xlabel('Time, t'); ylabel('Re z(t)'); zlabel('Im
z(t)');

plot3(t, sine_r, o, 'r');
% real part
plot3(t, sine_r, -sine_i, 'b');
% real - imaginary part

quiver3(t, o, o, o, sine_r, o, 0, 'r');
% arrow field: real part
quiver3(t, sine_r, o, o, o, -sine_i, 0, 'b');
% arrow field: real - imaginary part
plotArrow3([0 t_max], [0 0], [0 0], 'Color',
'k');% axe line

%% Figure d: real - image polar
subplot(2, 2, 4); hold on; grid on; axis([0 t_max
-1 1 -1 1]); view(-30, 15);
title('c) Complex sine wave ( $\omega < 0$ ): polar');
xlabel('Time, t'); ylabel('Re z(t)'); zlabel('Im
z(t)');

plot3(t, sine_r, -sine_i, 'k');
% sine curve

quiver3(t, o, o, o, sine_r, -sine_i, 0, 'k');
% arrow fields
line([0 t_max], [0 0], [0 0], 'Color', 'k');
% axe line

%% Text design
set(groot, 'defaulttextinterpreter', 'latex');
set(groot, 'defaultAxesTickLabelInterpreter',
'latex');
set(groot, 'defaultLegendInterpreter', 'latex');
set(findall(gcf, '-property', 'FontSize'),
'FontSize', 10);
```

REFERENCES

- [1] E. Tikhonov, M. Snep-Sneppe. "Introduction to signal processing: sine wave and complex signals", International Journal of Open Information Technologies, Vol. 7, No 3, March 2019, ISBN 2307-8162.
- [2] Isaac Newton (1704). "Optics", London: Royal Society, ISBN 0-486-60205-2.
- [3] Ronald N. Bracewell. "The Fourier Transform", Scientific American. Journal "V mire nauki", No 8, 1989, pages 48–56.
- [4] Qiang, Fu (2003). "Radiation (Solar)", In Holton, James R. Encyclopedia of atmospheric sciences. Amsterdam: Academic Press. pp. 1859–1863. ISBN 978-0-12-227095-6. OCLC 249246073.
- [5] Emmanuel C. Ifeakor (2011) "Digital Signal Processing A practical approach download". Prentice Hall; 2 edition (October 7, 2001), ISBN-10: 9780201596199
- [6] A.B Sergienko, Digital signal processing (Original title: Tsifrovaja obrabotka signalov, Цифровая обработка сигналов). // Piter, 2002, ISBN 5-318-00666-3
- [7] User's "md2perpe" comment (August 25, 2017) on math.stackovernet website. Available: <https://math.stackovernet.com/ru/q/692456#2406250>. Retrieved: Mar, 2019.