

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE  
NATIONAL TECHNICAL UNIVERSITY OF UKRAINE  
«IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE»

**LABORATORY PRACTICE  
OF DISCIPLINE  
«REGISTRATION AND PROCESSING OF  
BIOSIGNALS AND MEDICAL IMAGES»**

**Tutorial**

Recommended by the Methodical Council of Igor Sikorsky KPI  
as a tutorial for bachelor's degree students  
educational program "Medical Engineering"  
specialty 163 "Biomedical engineering"

Compilers: A. Popov, K. Ivanko, A.Porieva

Electronic online educational publication

Kyiv  
Igor Sikorsky KPI  
2022

Reviewer *Popovych Pavlo*, PhD, Associate professor of AMES department,  
Igor Sikorsky KPI

Editor-in-Chief *Timofeev Volodymyr*, Doctor of Science, Prof.

*The stamp is provided by the Methodical Council of the Igor Sikorsky KPI  
(protocol № dated \_\_\_\_\_ 2022)  
at the request of the Academic Council of the Faculty of Electronics  
(protocol № dated \_\_\_\_ 2022)*

The tutorial is intended for bachelor's degree students of educational program "Medical Engineering", specialty 163 "Biomedical engineering". It is dedicated to helping students to complete the laboratory practice of the discipline "Registration and processing of biosignals and medical images". The educational component "Registration and processing of biosignals and medical images" contributes to the development of professional expertise in knowledge of methods for processing and analysis of biomedical signals of different nature. The main purpose of the discipline is to form a holistic view of signals and methods of their study, as well as the acquisition of knowledge, skills, abilities and experience in using methods of registration, processing and analysis of biosignals and images in practice. Laboratory works are performed on the following topics: types of signals, their visualization, linear systems, spectral analysis, filtration, correlation analysis, wavelet analysis, and image processing.

Register. № **III XX/XX-XXX**. Scope **XX** author's sheets  
National Technical University of Ukraine  
Igor Sikorsky Kyiv Polytechnic Institute  
Peremohy Ave., 37, Kyiv, 03056  
<https://kpi.ua>

Certificate of inclusion in the state register of publishers, manufacturers  
and distributors of publishing products ДК № 5354 dated 25.05.2017 p.



## CONTENT

Introduction.....	2
Laboratory work № 1. Basics of Python programming.....	4
Laboratory work № 1. Signals visualization using Python.....	13
Laboratory work № 3. Modeling of linear systems in time and frequency domain...	27
Laboratory work № 4. Fourier spectral analysis of signals.....	36
Laboratory work № 5. Time-Spectral Fourier analysis of signals.....	56
Laboratory work № 6. Signal filtering.....	72
Laboratory work № 7. Correlation analysis of deterministic signals.....	80
Laboratory work № 8. Wavelet signal analysis.....	83
Laboratory work № 9. Processing of medical images.....	86
References.....	93

## INTRODUCTION

The tutorial is intended for bachelor's degree students of educational program "Medical Engineering", specialty 163 "Biomedical engineering". It is dedicated to helping students to complete the laboratory practice of the discipline "Registration and processing of biosignals and medical images". The educational component "Registration and processing of biosignals and medical images" contributes to the development of professional expertise in knowledge of methods for processing and analysis of biomedical signals of different nature. The main purpose of the discipline is to form a holistic view of signals and methods of their study, as well as the acquisition of knowledge, skills, abilities and experience in using methods of registration, processing and analysis of biosignals and images in practice.

Expected learning outcomes of the discipline "Registration and processing of biosignals and medical images" include knowledge of:

1. Types and parameters of signals and images of different nature, including biomedical signals and images;
2. Methods of mathematical description of linear stationary discrete systems;
3. Methods of spectral, spectral-temporal, wavelet and correlation analysis, conditions and limitations in their application;
4. The essence of frequency-dependent signal processing using filters;
5. Basic approaches to stochastic, nonlinear and multivariate signal analysis and pattern recognition;
6. Trends in signal theory and application of signal research methods in the specialty.

Upon successful completion of the course students are expected to be able to have:

- Ability to apply knowledge of biomedical signal and image processing in practical situations;

- Knowledge and understanding of the subject area of signal processing and applying it in professional activity;
- Skills in the use of information and communication technologies;
- Ability to perform research at the appropriate level;
- Ability to search, process and analyze information from various sources;
- Ability to generate new ideas (creativity);
- Ability to make well-grounded decisions.

An integral part of the training of students in the discipline "Registration and processing of biosignals and medical images" is the implementation of the laboratory works. Tasks in this tutorial are divided into mandatory tasks for all students and additional tasks, designed for in-depth study of the subject.

# LABORATORY WORK № 1

## BASICS OF PYTHON PROGRAMMING

**Purpose:** To get acquainted with the basics of Python programming on the examples of using standard functions, building script files and creating user functions.

### Basic information

Python is a high-level general-purpose programming language focused on improving developer productivity and code reading with minimalist kernel syntax and a wide range of useful features. Python supports structured, generalized, object-oriented, functional, and aspect-oriented programming. The main architectural features are dynamic typing, automatic memory management, full introspection, exception handling mechanism, support for multithreaded computing, high-level data structures. It supports the division of programs into modules, which, in turn, can be combined into a package.

Python is one of the most common programming languages, and is used for web applications development, machine learning, data analysis and visualization, graphical interface development, audio and video data development, embedded system programming, etc.

In the first laboratory work, you must master the basics of Python at a level sufficient to perform subsequent laboratory works. To do this, students must choose an easy-to-learn online resource with which to learn Python.

It is suggested to pay attention to the following resources and online courses:

- *The Python Tutorial*

<https://docs.python.org/3/tutorial/>

- *Python for Data Science and AI (IBM by Coursera)*

<https://www.coursera.org/learn/python-for-applied-data-science-ai#syllabus>

- *Python for Everybody Specialization (University of Michigan by Coursera)*

## 1. Programming for Everybody (Getting Started with Python)

<https://www.coursera.org/learn/python?specialization=python#syllabus>

## 2. Python Data Structures

<https://www.coursera.org/learn/python-data?specialization=python#syllabus>

- *How to Learn Python in 21 Days?*

<https://www.geeksforgeeks.org/how-to-learn-python-in-21-days/?ref=leftbar-rightbar>

- *Learn Python 2*

<https://www.codecademy.com/learn/learn-python>

Using the development environment is useful for creating programs in Python. It is suggested to pay attention and use the following free products:

- **PyCharm Community Edition** <https://www.jetbrains.com/pycharm/> – development environment. Getting acquainted with the work in it can be combined with learning the Python language in this course: *PyCharm Edu -- Learning Python*

<https://www.jetbrains.com/help/education/learner-start-guide.html?section=Introduction%20to%20Python>

- **Jupyter** <https://jupyter.org/> – interactive web development environment. To get acquainted with the work in Jupyter, you can use these resources:

- Jupyter Notebook Tutorial: The Definitive Guide

<https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook>

- Jupyter Notebook for Beginners: A Tutorial

<https://www.dataquest.io/blog/jupyter-notebook-tutorial/>

- Jupyter Tutorial

<https://www.tutorialspoint.com/jupyter/index.htm>

- **Colaboratory** <https://colab.research.google.com/> – interactive web development environment.

## Tasks

1. Get acquainted with the types of data and representation of variables in Python.

- with the rules of entering variables and naming variables;

- with operations on numbers and matrices. To study matrix and element-by-element operations on matrices.

Provide some examples of code and results of code execution in the laboratory protocol.

*Example:*

```
A = np.matrix("1 2 3; 4 5 6")
```

```
B = np.matrix("3 3 3; 2 2 2")
```

```
# matrix multiplication by number:
```

```
C = 3 * A
```

```
print("\n",C)
```

```
# matrix addition:
```

```
C = A + B
```

```
print("\n",C)
```

```
# subtraction of matrices:
```

```
C = A - B
```

```
print("\n",C)
```

```
# matrix multiplication:
```

```
A = np.matrix("2 2 2; 1 1 1")
```

```
B = np.matrix("5 5; 4 4; 3 3")
```

```
C = A * B
```

```
print("\n",C)
```

2. Complete tasks, provide code and results of its execution in the laboratory protocol:

2.1. Specify an array, the elements of which are an arithmetic progression;

2.2. Get acquainted with the function of constructing histograms, construct histograms of random numbers with different probability density distributions. Use random number generation functions with given probability distribution densities (Fig.1.1, 1.2).

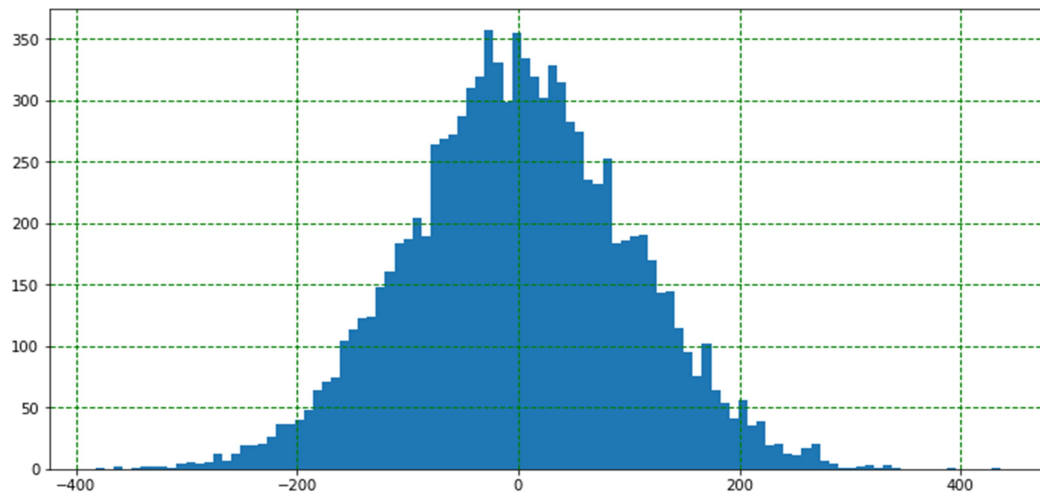


Fig.1.1. Example of histogram of random numbers with normal probability density distribution

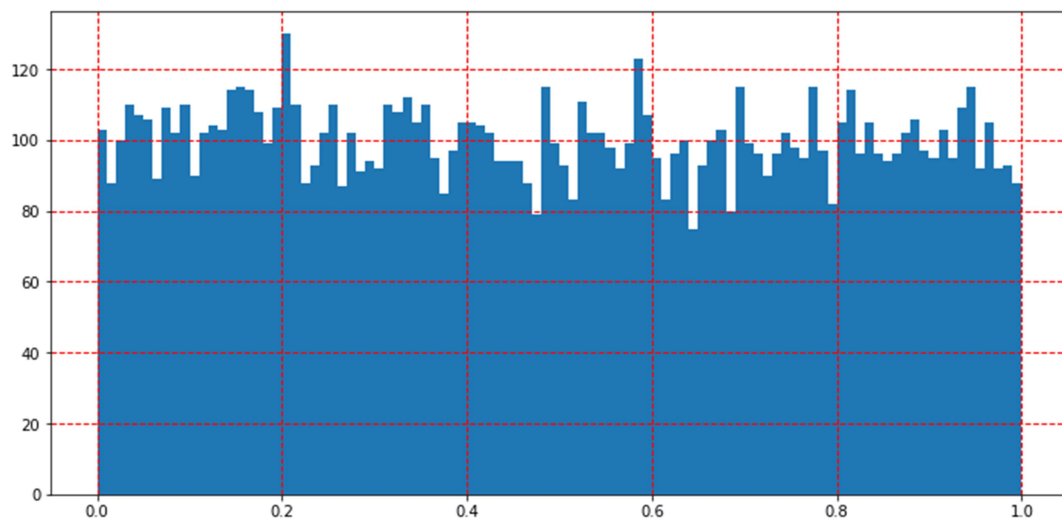


Fig.1.2. Example of histogram of random numbers with uniform probability density distribution

*Useful libraries and functions:*

numpy (numpy.random.normal, numpy.random.uniform),

random (random.triangular, random.expovariate, random.gauss,  
random.normalvariate),

matplotlib.pyplot (hist, grid, subplots, plot)

3. Write your own script file, in which plot a function of one variable. Mark the axis and the title of the graph, draw a grid (Fig.1.3).

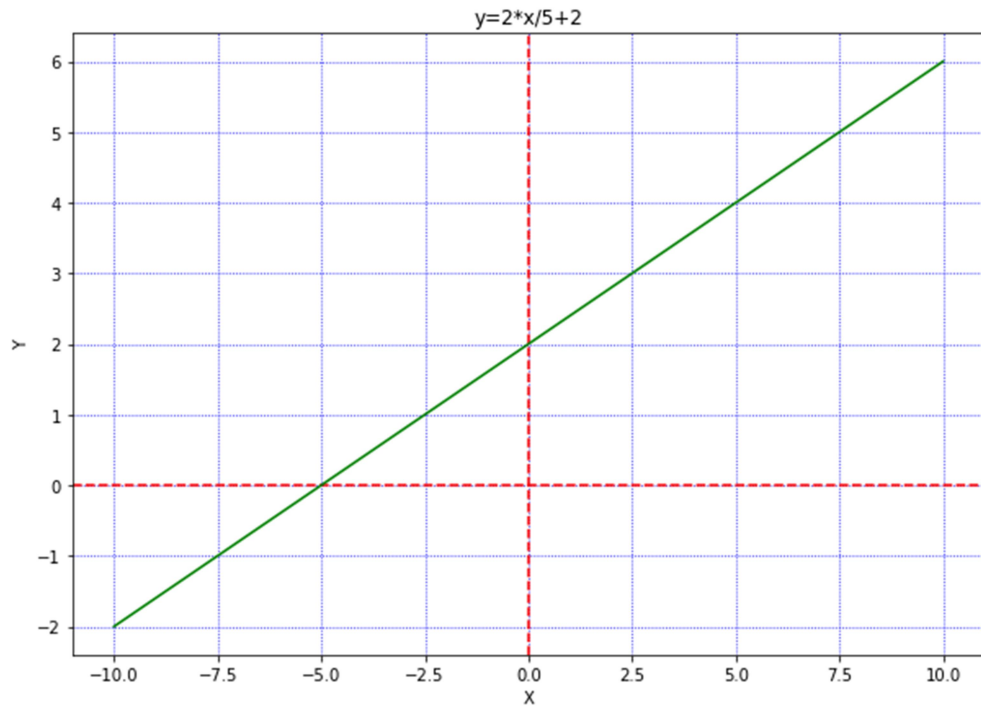


Fig.1.3. Example of plotting a function of one variable

*Useful libraries and functions:* numpy, matplotlib.pyplot (hist, grid, subplots, plot)

4. Write a script file in which:

4.1 Plot sine the waves with frequencies of 1, 10, 50 Hz. Signals duration is 1 sec, sampling frequency is 256 Hz. Build graphs in one window, but in different axes. The amplitudes of each sine wave must be random numbers (Fig.1.4).

4.2 Do the same task as in 4.1, but set the amplitude of each sine wave from the keyboard;

*Useful function:* **input**

4.3 Sign the title of each graph with the text that contains the values of the frequency and amplitude of the corresponding sine wave (Fig.1.5).



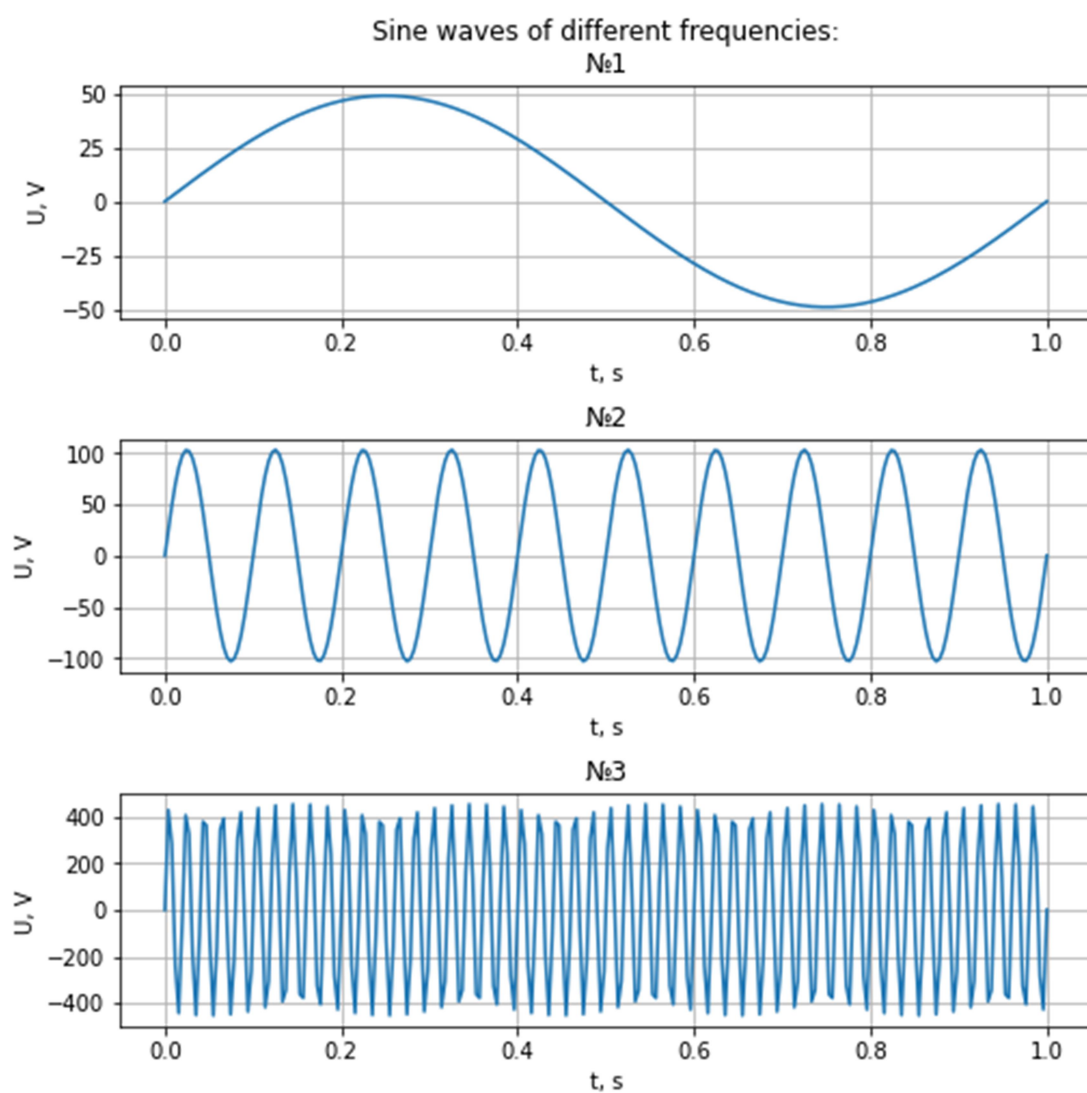


Fig.1.4. Example of plotting sine wave with frequencies of 1, 10, 50 Hz  
(graphs in one window, but in different axes)

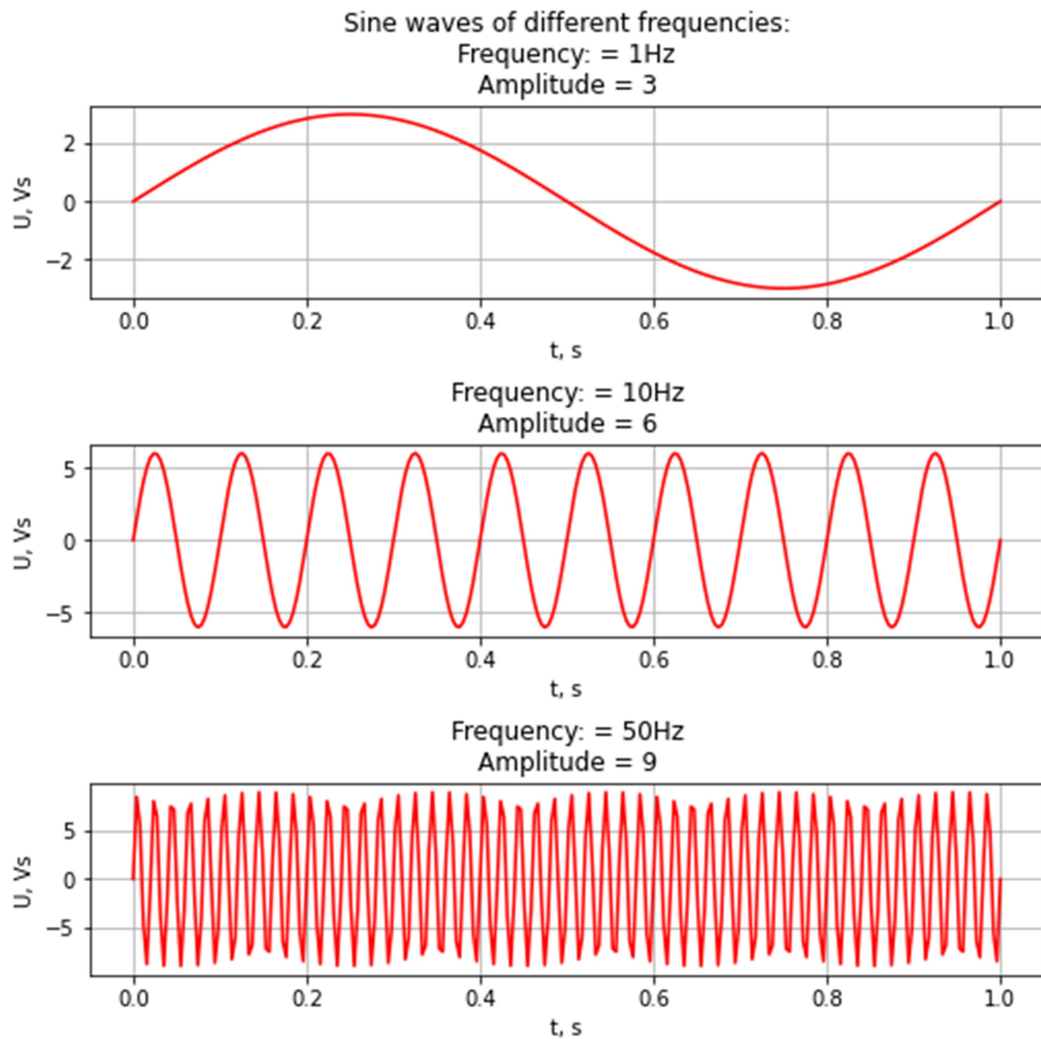


Fig.1.5. Example of signing the title of each graph with the text that contains the values of the frequency and amplitude of the corresponding sine wave

5. Provide examples of using functions, which generate pulses (<https://docs.scipy.org/doc/scipy/reference/signal.html>, waveforms): Gaussian pulses, triangular pulses and rectangular (Fig.1.6).

Useful functions: `gausspulse`, `sawtooth`, `square`

5.1 Construct a single rectangular pulse. Set the time interval to 10 seconds, sampling frequency 256 Hz. Plot a graph of a single rectangular pulse with a width of 300 ms, with the center at time 4s (Fig.1.7).

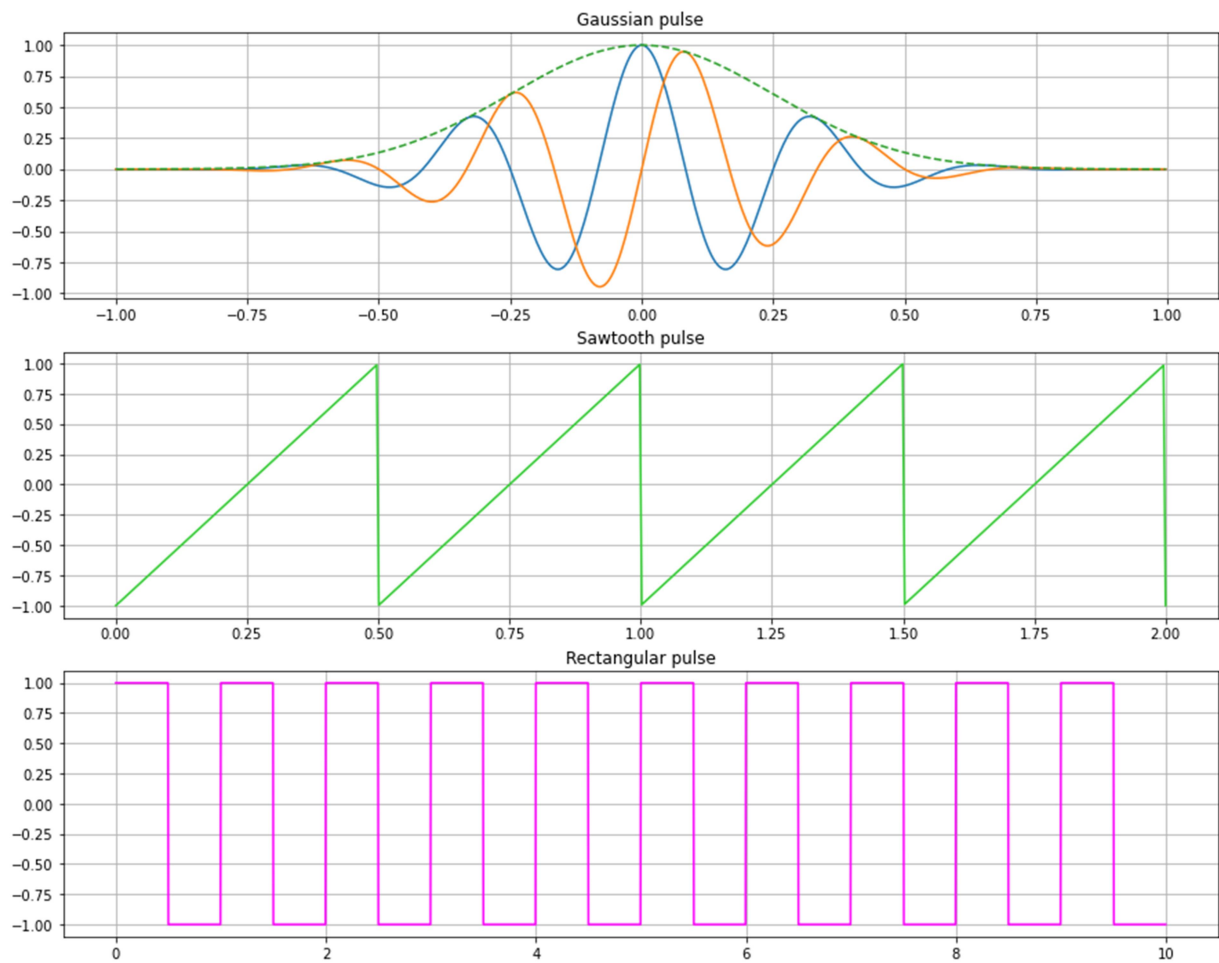


Fig.1.6. Example of using functions, which generate Gaussian pulses, triangular pulses and rectangular pulses

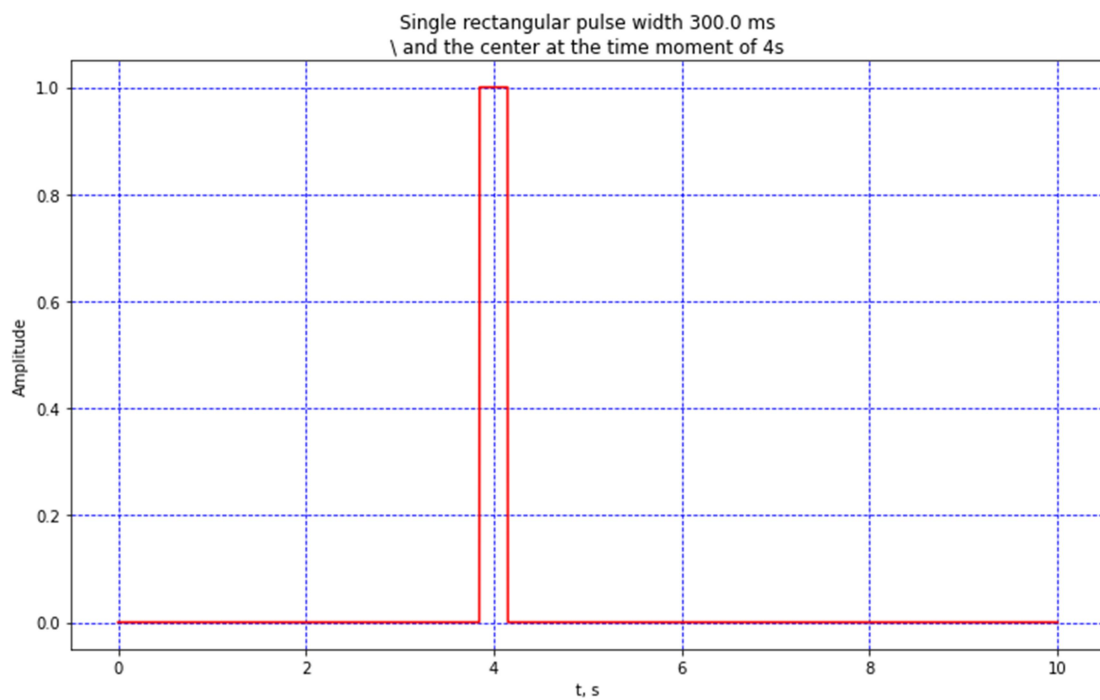


Fig.1.7. Plotting a graph of a single rectangular pulse

5.2 Write a script file to plot a rectangular pulse, the duration and amplitude of which will be set from the keyboard. Set the location of the pulse with a random number, but implement checking whether the pulse is outside the graph.

6. Save the calculation data to a file. Read them from a file in another script, plot the figures.

*Useful functions:* np.savez, np.load

7. Build your own function (in a separate or in the same file) to plot a sine wave with a given frequency, amplitude and duration for a sampling frequency of 256 Hz. As the output parameter of the function, calculate the average value of the sine wave.

*def is the keyword for defining a function. The function name is followed by parameter(s) in (). The colon : defines the start of the function body, which is marked by indentation. Inside the function body, the return statement determines the value to be returned.*

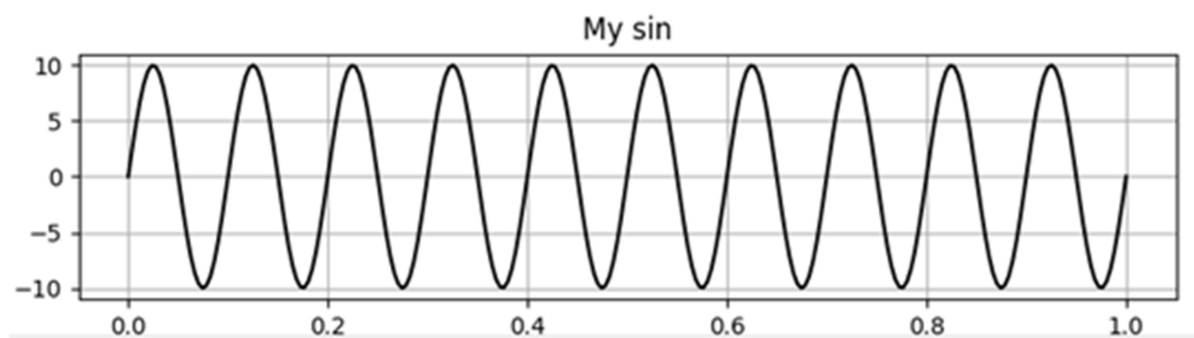


Fig.1.8. Plotting a sine wave with a given frequency, amplitude and duration

### Control questions

1. How to specify an array using Python functions, the elements of which are an arithmetic progression.
2. List examples of Python functions, which generate random numbers with different probability density distributions.
3. List examples of Python functions, which generate Gaussian pulses, triangular pulses and rectangular pulses.

## LABORATORY WORK № 2

### SIGNALS VISUALIZATION USING PYTHON

**Purpose:** acquire skills in reading and visualizing biomedical signals of different nature in Python.

#### Tasks

**All signal recordings are presented on the website**

**<https://ee.kpi.ua/~popov/teaching.html#6>.**

All graphs must have properly signed axes (based on sampling rate and physical quantities being measured). The axes must be properly scaled.

Create a **function that calculates the duration of signals**. Use it to determine the durations of all signals used in the work.

1. Record a signal from **internal sensors (accelerometer, gyroscope, etc.)** using a smartphone. To do this, pre-install the **AndroSensor application** on the smartphone:

**<https://play.google.com/store/apps/details?id=com.fivasim.androsensor&hl=en>**

Launch the app and master it. Set in the settings "Data update" - "very fast", "Recording interval" - set the minimum available.

Perform signal registration from available smartphone sensors when performing the following exercises:

- record signals while standing, hold the device in your hand, which is lowered along the torso for 1 minute;
- start recording, put the device in your pocket, stand for 10 seconds, walk down the hall calmly, stand for 10 seconds, stop recording;
- start recording, put the device in your pocket, stand for 10 seconds, run down the hall, stand for 10 seconds, stop recording.

Save the data to a csv file, open it with Python. Save the signals to disk for later use. Plot graphs of accelerometer and gyroscope signals, sign axes (Fig.2.1).

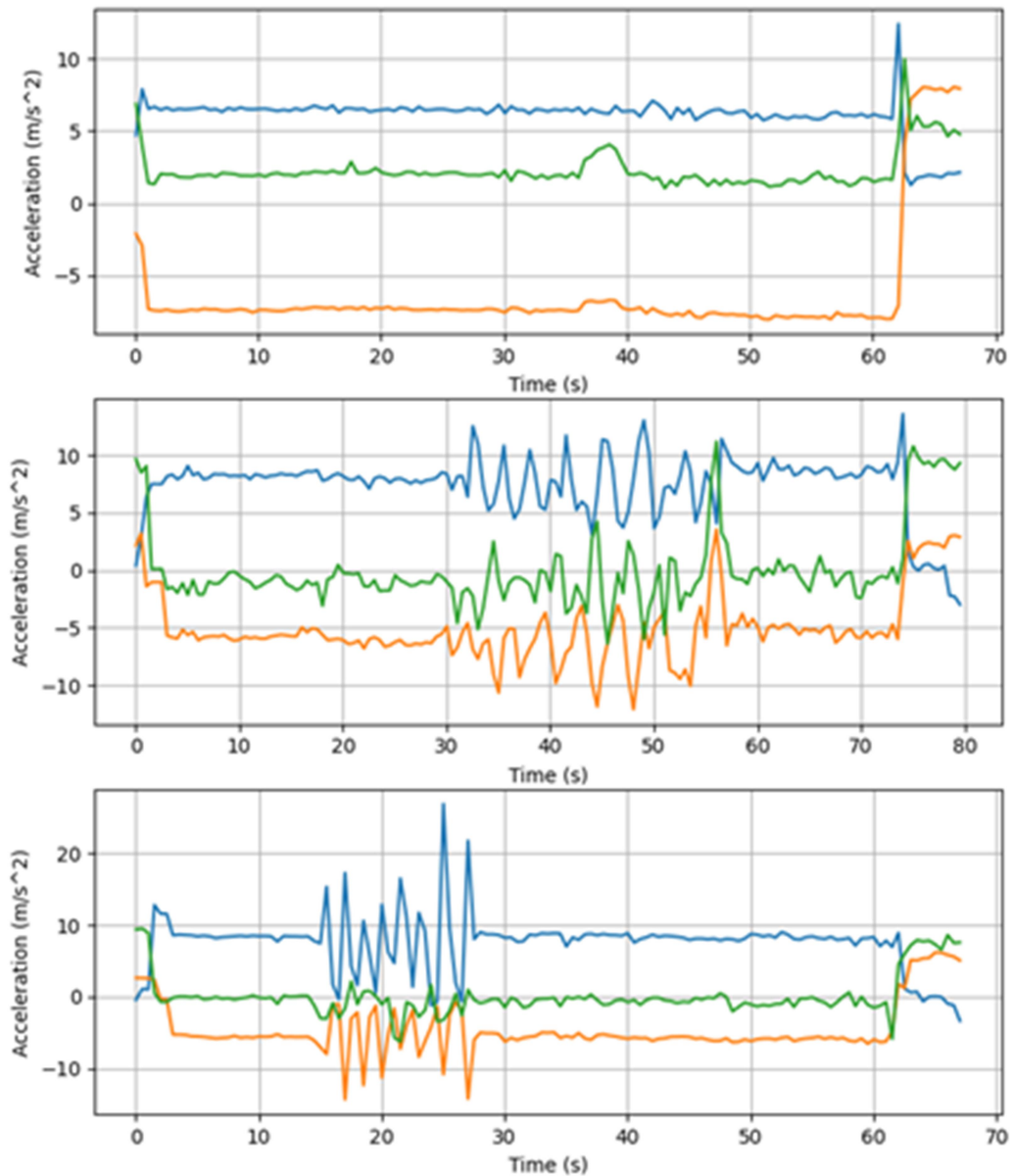


Fig. 2.1. Signal registration from smartphone sensors (standing, walking, running)

2. Record 5 sec of sound from an external device using Python. Record the same phrases or music with a sampling rate of **8 kHz and 44.1 kHz**. Read from the file and listen to the obtained recordings. Print a graph, mark the axis (Fig.2.2, 2.3).

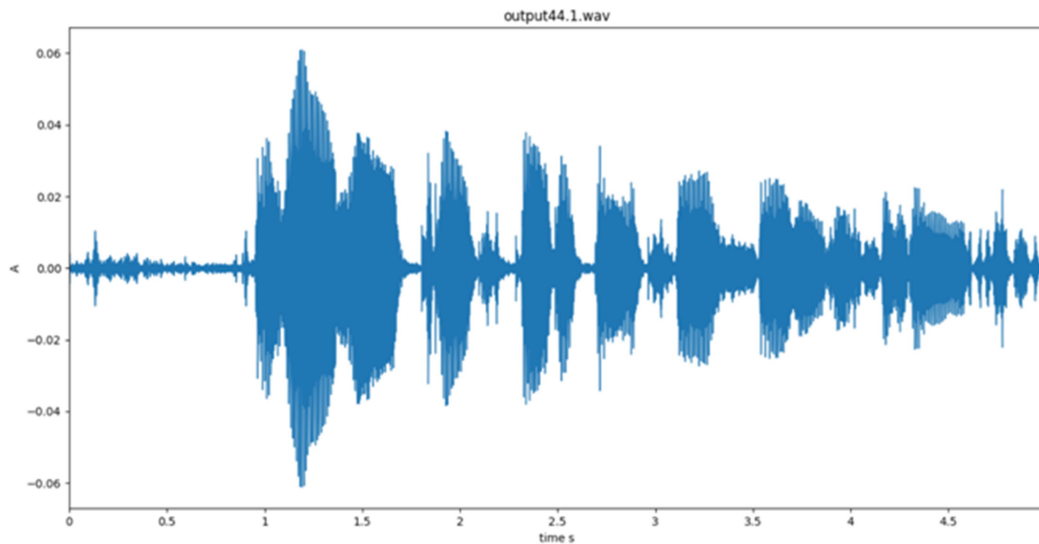


Fig. 2.3. Example of audio signal with sampling rate of 8 kHz

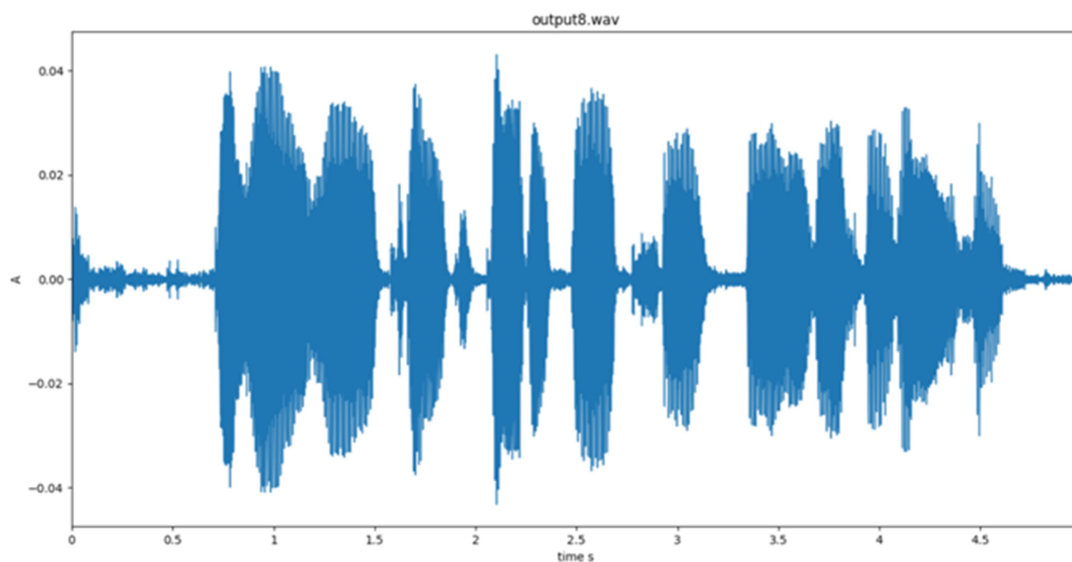


Fig. 2.3. Example of audio signal with sampling rate of 44.1 kHz

3. Read the EEG signals of a healthy and sick person, obtained using a computer electroencephalograph and stored in a mat-file (MatLAB). Print a graph, mark the axis (Fig.2.4, 2.5). Save the received signal for use in future work.

EEG\_healthy.rar and EEG\_sick.rar archive files; select the signal according to your number on the list of students; The EEG is sampled with a frequency of 256 Hz, the voltage value is given in microvolts.



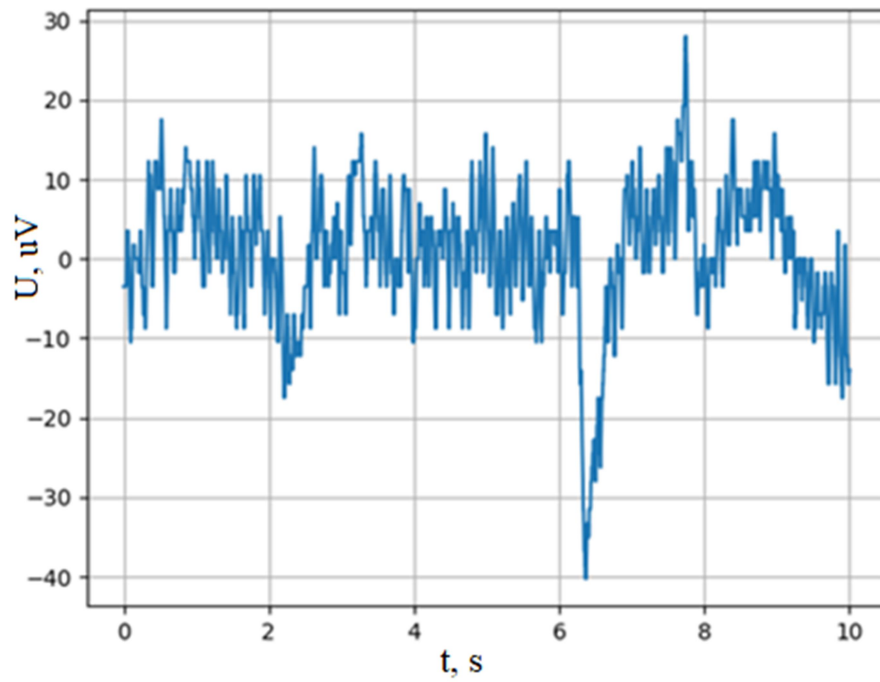


Fig. 2.4. Example of EEG signal in norm

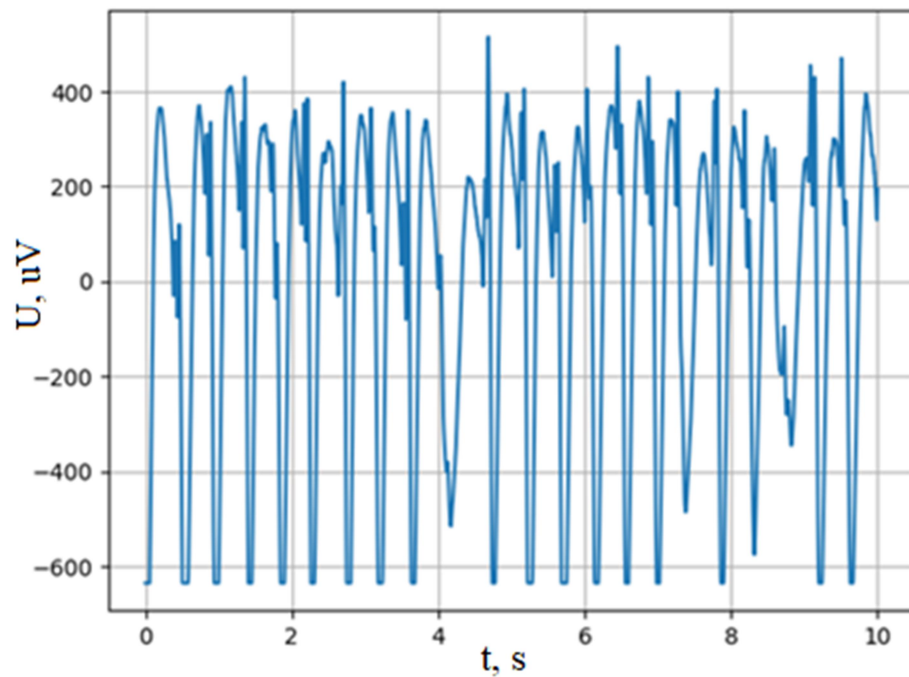


Fig. 2.5. Example of EEG signal in epilepsy

4. Read the ECG signals of a healthy (Norm) and sick (anomaly) person. Each file has the same structure:

description - signal description, may be empty;

source - the line in which the source of the file is specified, usually "mitdb / 200";

fs - sampling frequency;



units - signal units;

signal - the signal itself in the form of a numpy array;

labels - text labels to the signal;

labels\_indexes - indices of the corresponding labels;

source\_start - additional field indicating the beginning of the period in the original signal specified in source;

source\_end - an additional field indicating the end of the period in the original signal specified in source.

Plot graphs of signals (\*) and labels for the corresponding heartbeats (Fig.2.6, 2.7). You can read more about these records at <https://archive.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm>

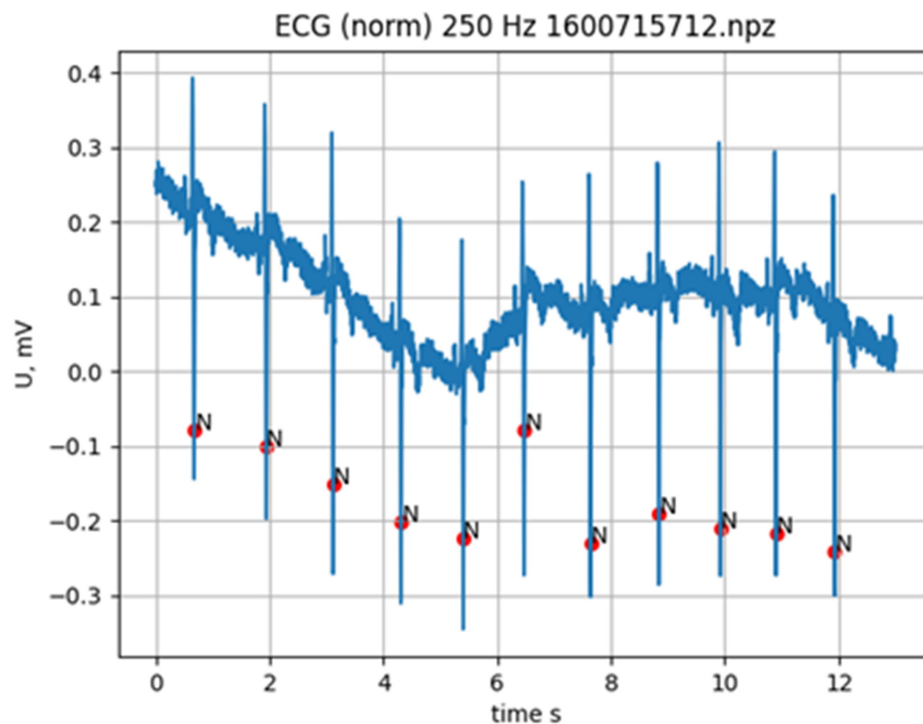


Fig. 2.6. Example of ECG signal in norm

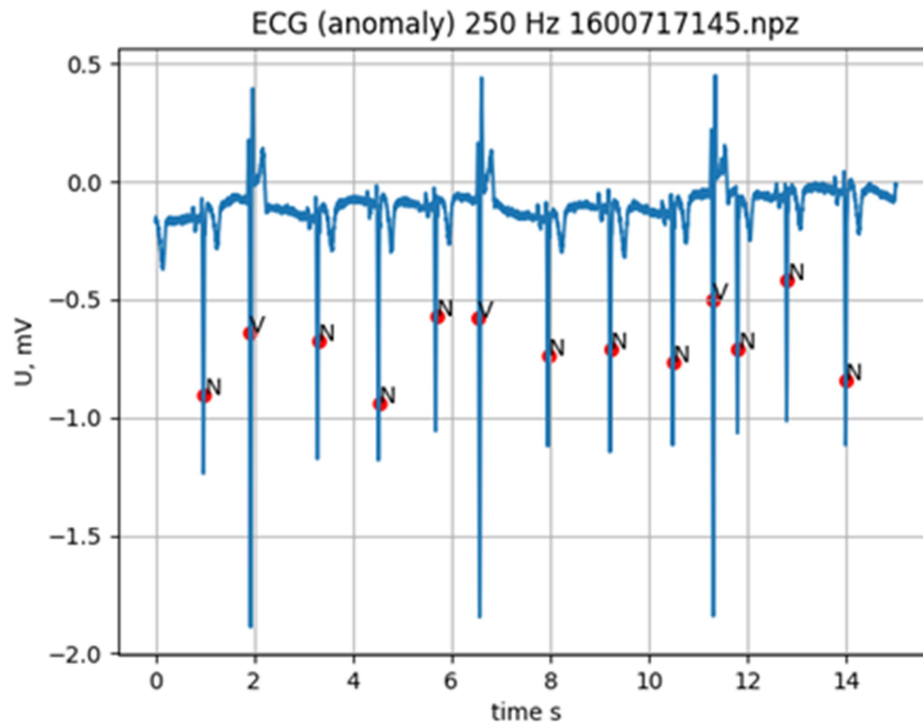


Fig. 2.7. Example of ECG signal in pathology

5. Read the signals of cardiogram of a healthy and sick person, saved in a mat-file. Print a graph, mark the axis (Fig.2.8, 2.9). Signal values are recorded in milliseconds, each count corresponds to the interval between heartbeats.

Determine using the custom function (create your own) duration of the recorded signal, taking into account the non-uniform sampling of the signal. Interpolate the signal to determine its values on a uniform grid of nodes for a sampling frequency of 1 Hz.

Save the received signal for use in future work.

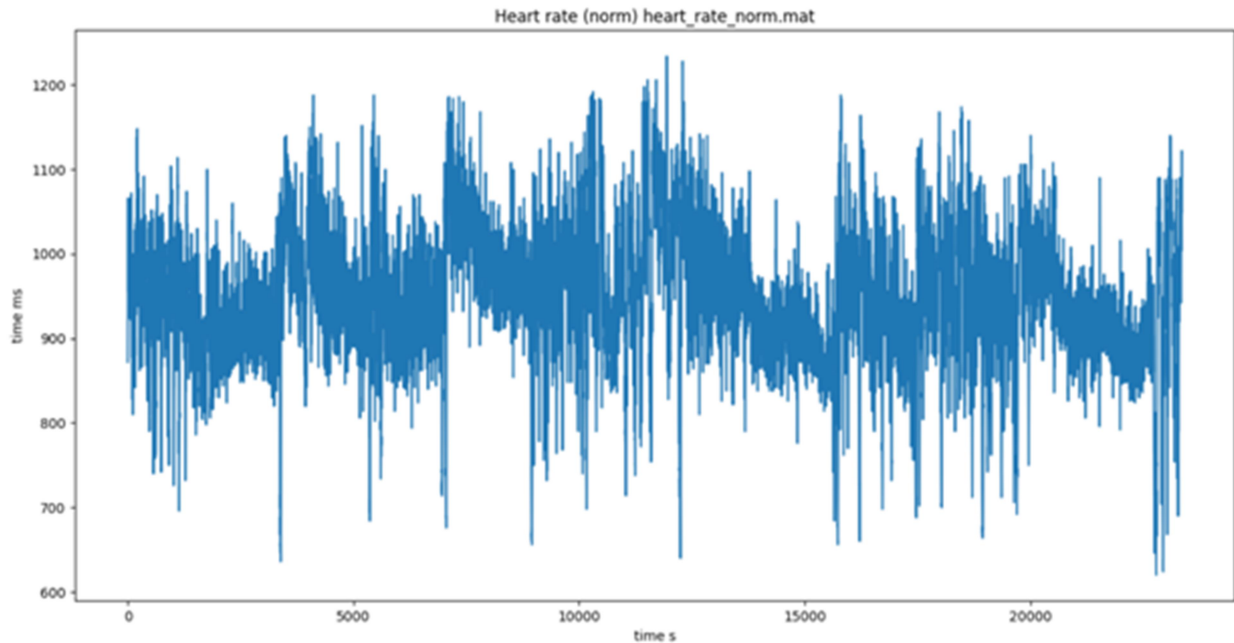


Fig. 2.8. Example of cardiorythmogram in norm

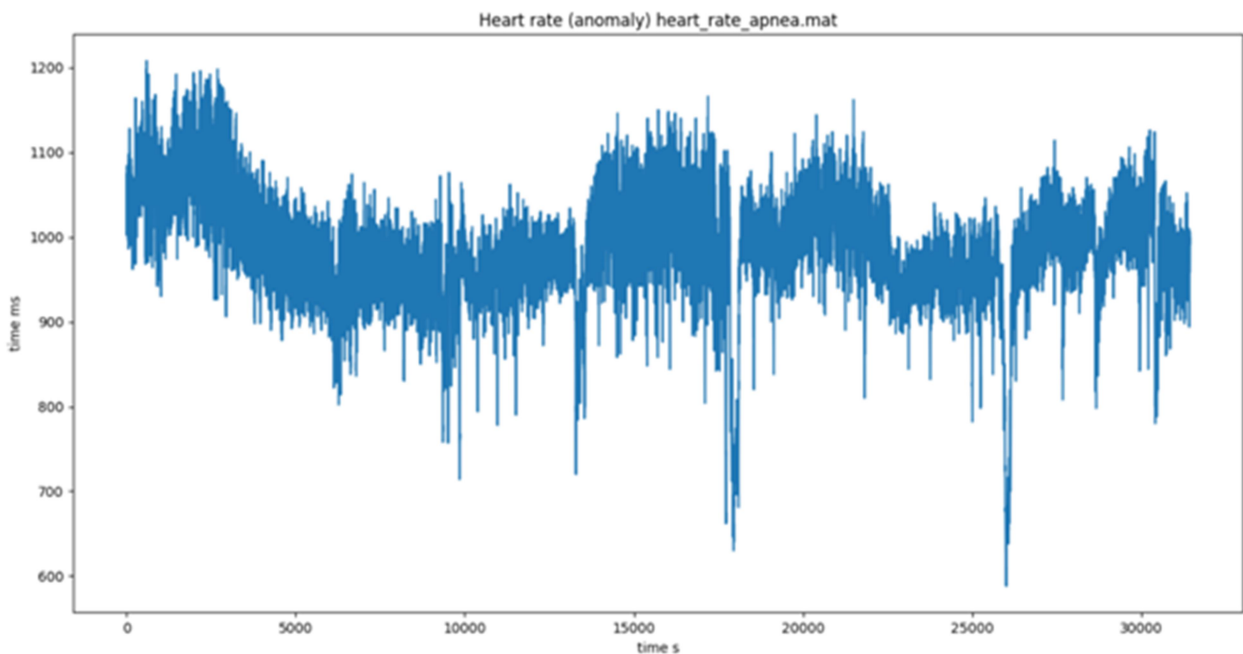


Fig. 2.9. Example of cardiorythmogram in pathology

6. Read human stabilogram signals that have been registered using the Wii Balance Board platform ([https://en.wikipedia.org/wiki/Wii\\_Balance\\_Board](https://en.wikipedia.org/wiki/Wii_Balance_Board)). The data folder contains signals for two groups of athletes - handball players and acrobats, (handball and acrobats, respectively). Each type of athlete performed certain experiments:

base\_close - stand straight, legs together, eyes closed;

base\_open - stand straight, legs together, eyes open;  
sway\_front-back\_30 - rocking back and forth at an angle of 30 degrees, legs together, eyes open;  
sway\_front\_back\_60 - rocking back and forth at an angle of 60 degrees, legs together, eyes open;  
sway\_left-right\_30 - swing left and right at an angle of 30 degrees, legs together, eyes open;  
sway\_left-right\_60 - swing left and right at an angle of 60 degrees, legs together, eyes open.

Perform the following tasks:

6.1 Read CoP signals (Center-of-Pressure) stabilograms of athletes specializing in different sports. For this purpose we suggest to use function **read\_csv of Pandas** (library for working with tabular data). Build graphs (Fig.2.10, 2.11)

The order of the columns in the file:

time\_ms - time in ms [column0]  
top\_left\_f\_kg - raw top\_left sensor data [column1]  
top\_right\_f\_kg = raw top\_right sensor data [column2]  
bottom\_left\_f\_kg = raw bottom\_left sensor data [column3]  
bottom\_right\_f\_kg = raw bottom\_right sensor data [column4]  
cop\_x = CoP X axis [column5]  
cop\_y = CoP Y axis [column6]  
total\_f = Total force [column7]

6.2 Select any two types of signals (e.g. base-open and sway\_left-right\_60) for the two types of athletes.

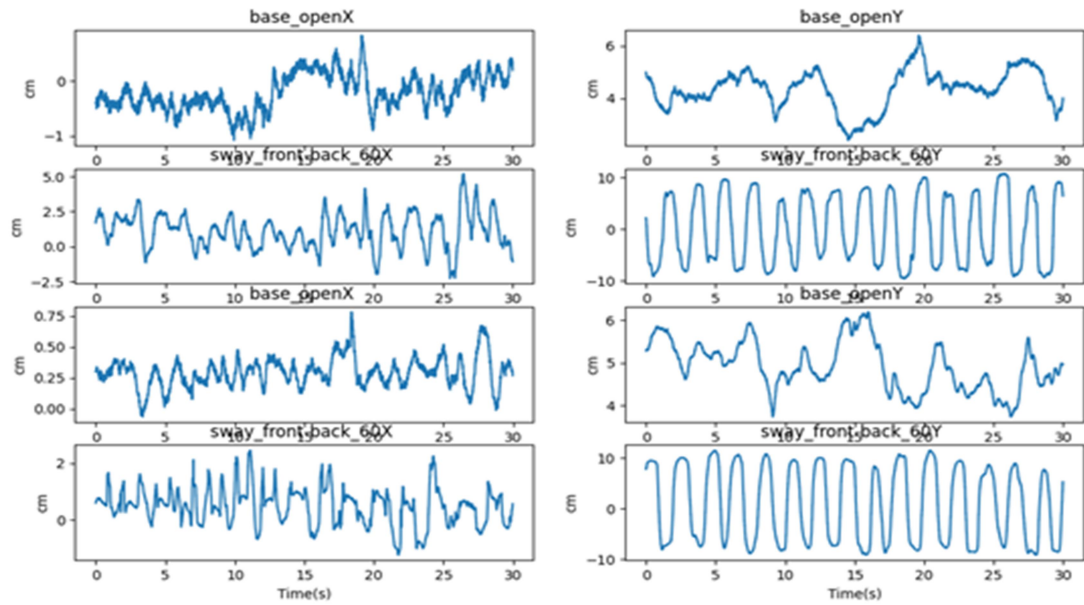


Fig. 2.10. Examples of stabilograms

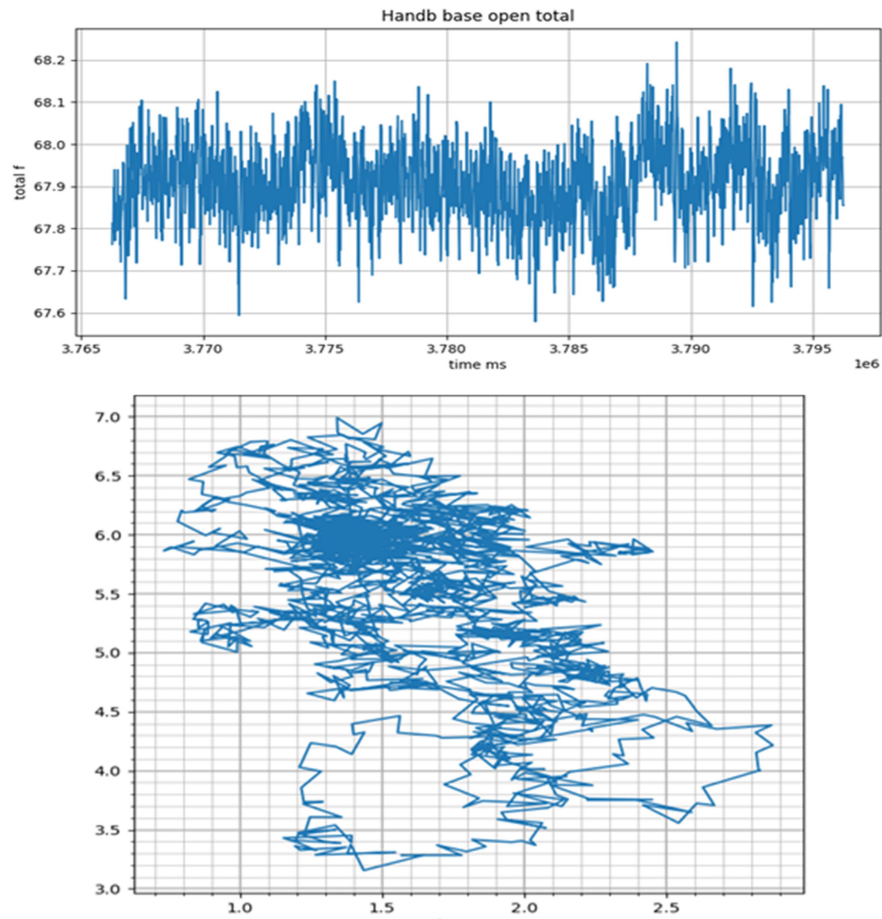


Fig. 2.11. Examples of CoP signals

6.3 For each signal in the folder of the appropriate type, calculate the statistical parameters (mean value, median value, standard deviation), separately on the X and Y axes, summarize in a table and compare for athletes of different types.

Signal	Operation	Acrobat	Handball
cop_x	Mean	-2.86	0.86
cop_x	Median	-2.833	0.865
cop_x	Standard deviation	0.185	0.088
Signal	Operation	Acrobat	Handball
cop_y	Mean	4.7	4.54
cop_y	Median	4.702	4.551
cop_y	Standard deviation	0.453	0.244
Signal	Operation	Acrobat	Handball
total_f	Mean	68.588	67.911
total_f	Median	68.591	67.913
total_f	Standard deviation	0.129	0.093

Fig. 2.12. Example of table with statistical parameters for CoP signals of athletes (mean value, median value, standard deviation)

#### 6.4 Draw conclusions.

7. Read the signals of heart rate (HR) and arterial blood oxygen saturation (SpO2). Plot graphs of both signals. Plot graphs of average values of signals, which are calculated in windows lasting 30 seconds, the windows do not overlap (Fig.2.13-2.16).

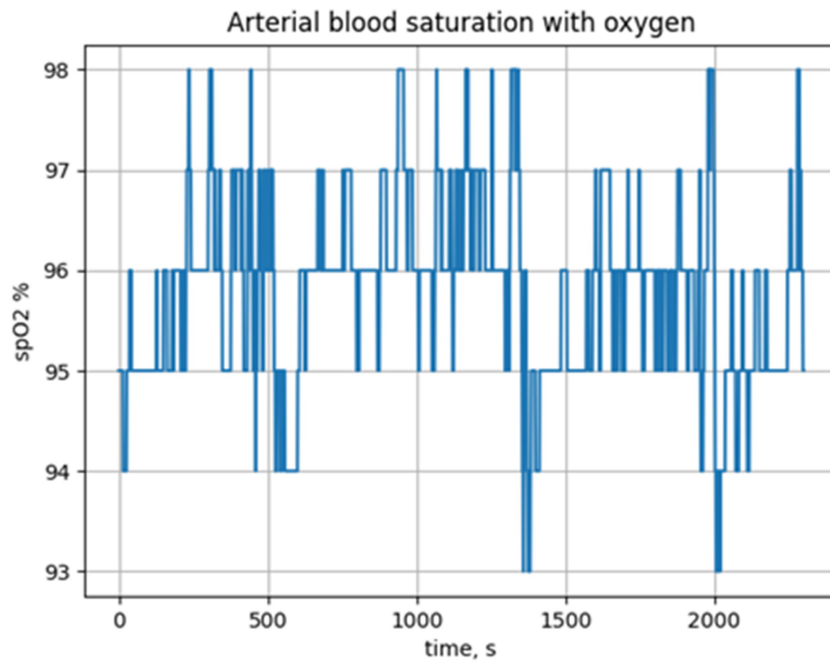


Fig.2.13. Example of arterial blood oxygen saturation signal

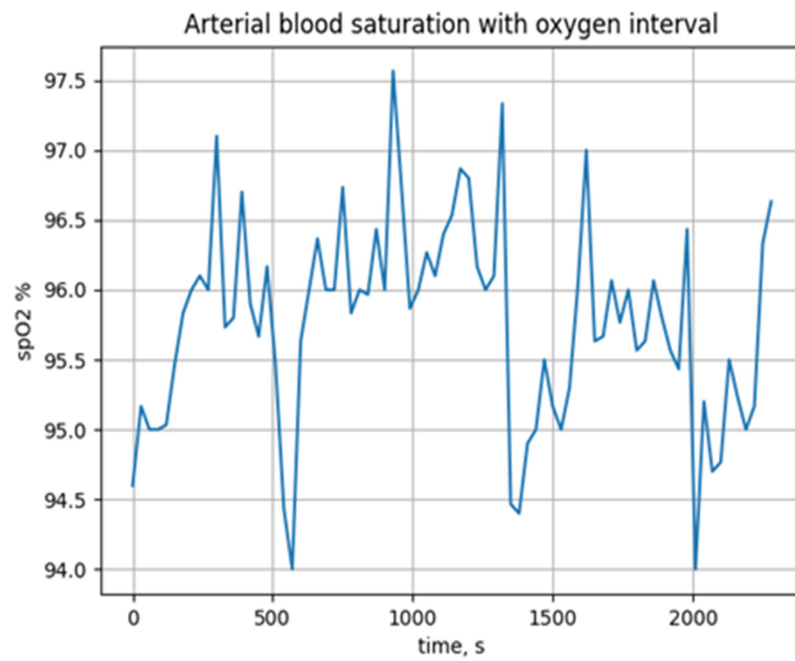


Fig.2.14. Average values of arterial blood oxygen saturation, calculated in windows lasting 30 seconds

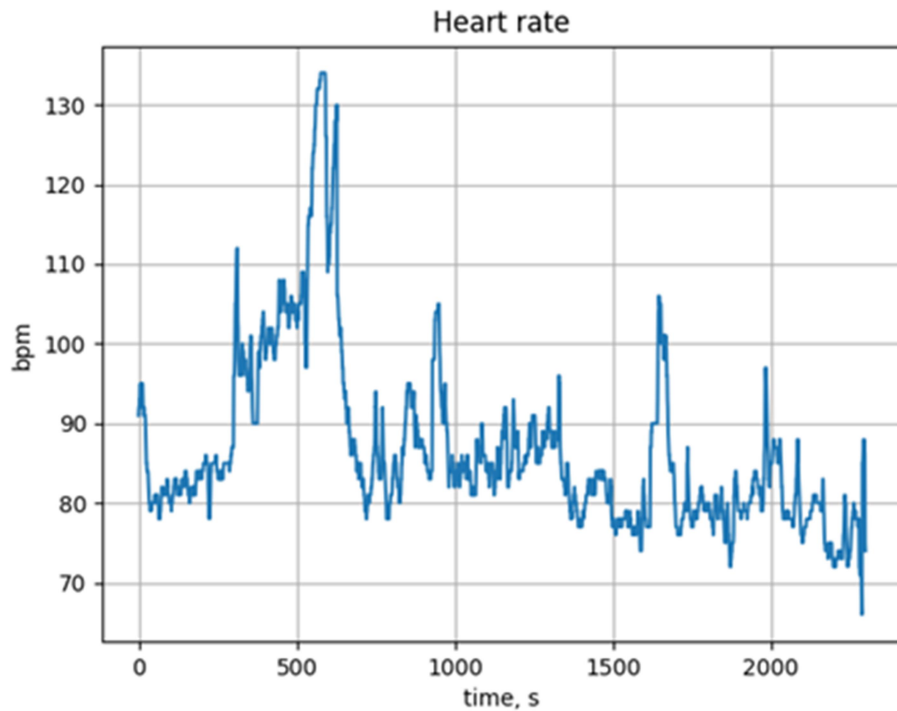


Fig.2.15. Example of heart rate signal

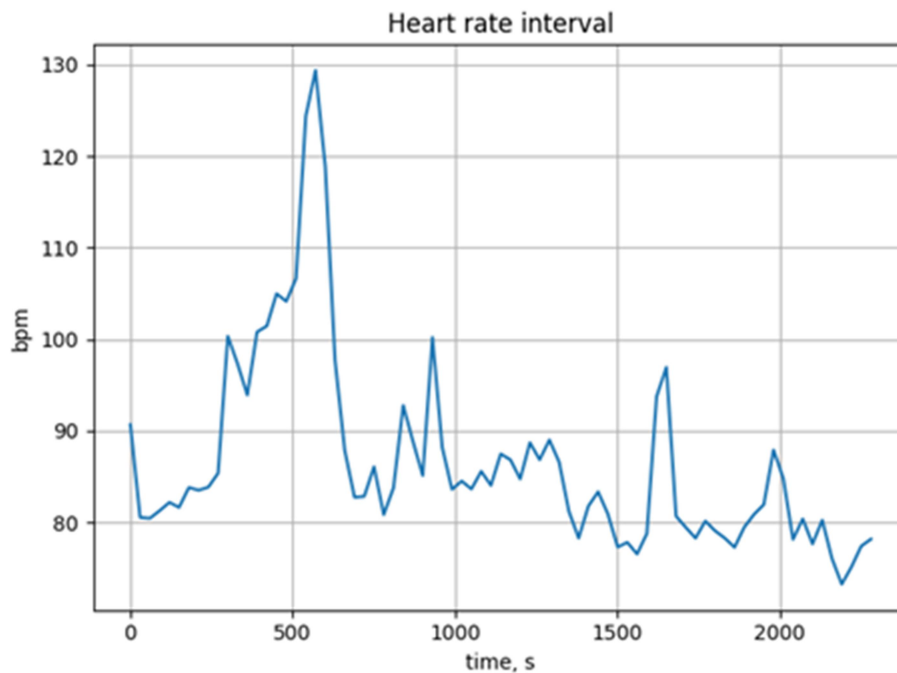


Fig.2.14. Average values of heart rate, calculated in windows lasting 30 seconds

8. Read the intracranial pressure signal in the work area. Save the received signal for use in future work. File TBI\_ICP.txt, single-channel signal, recorded with a sampling frequency of 125 Hz, units of measurement - mmHg), display the graph on the screen, mark the axis (Fig.2.15).



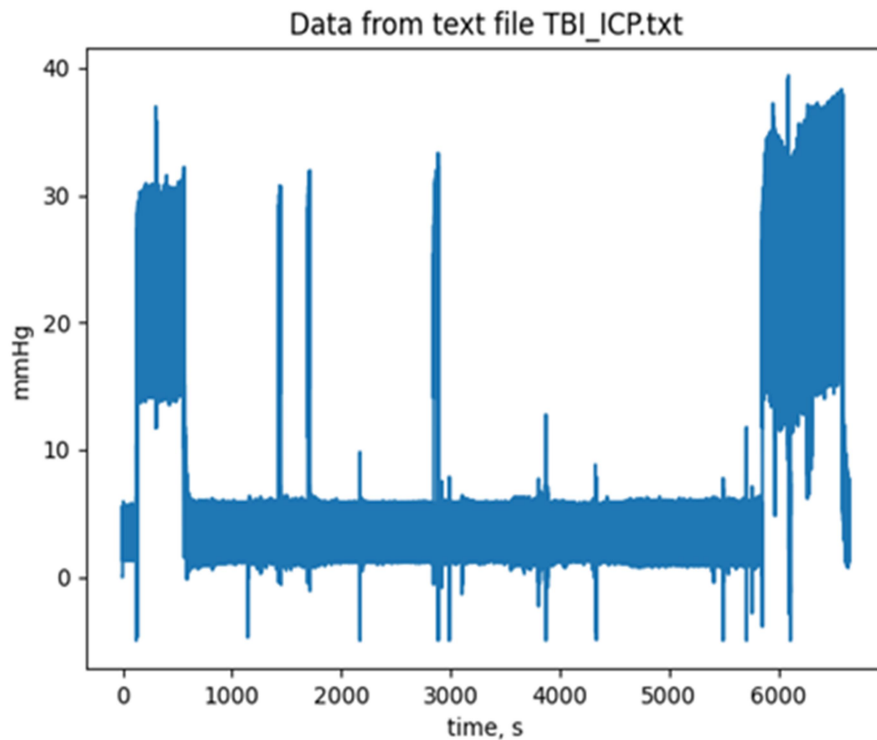


Fig.2.15. Intracranial pressure signal

9. Construct a **function for plotting the signal part** (Fig.2.16, 2.17). Parameters to pass to the function: time of the beginning and the end of a signal part (in seconds), a vector with samples of a signal, sampling frequency. Implement checking the correctness of the input of time points. Implement the possibility of the vector with samples of the signal area and time samples as an output of the function.

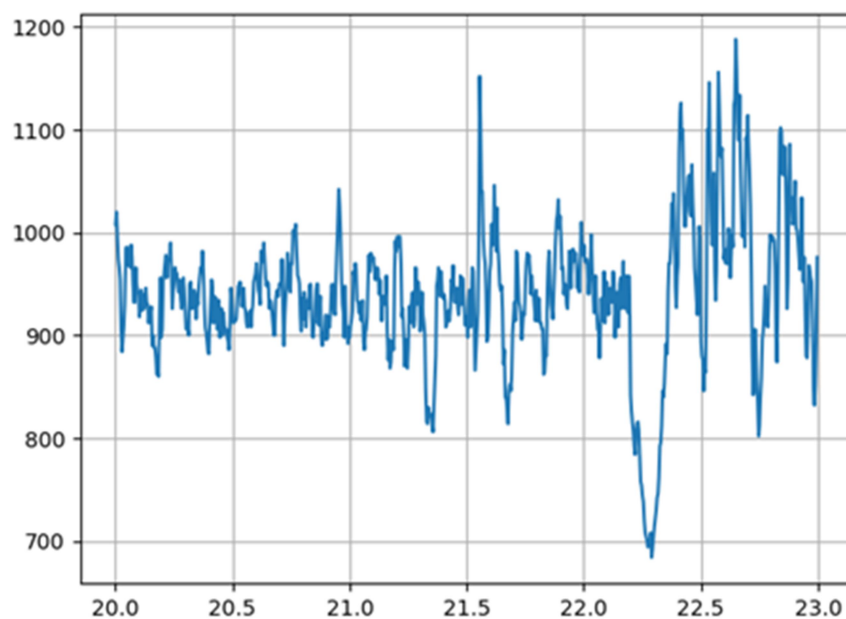


Fig.2.16. Plotting cardiorythmogram signal part (starting from 20 s and till 23 s)

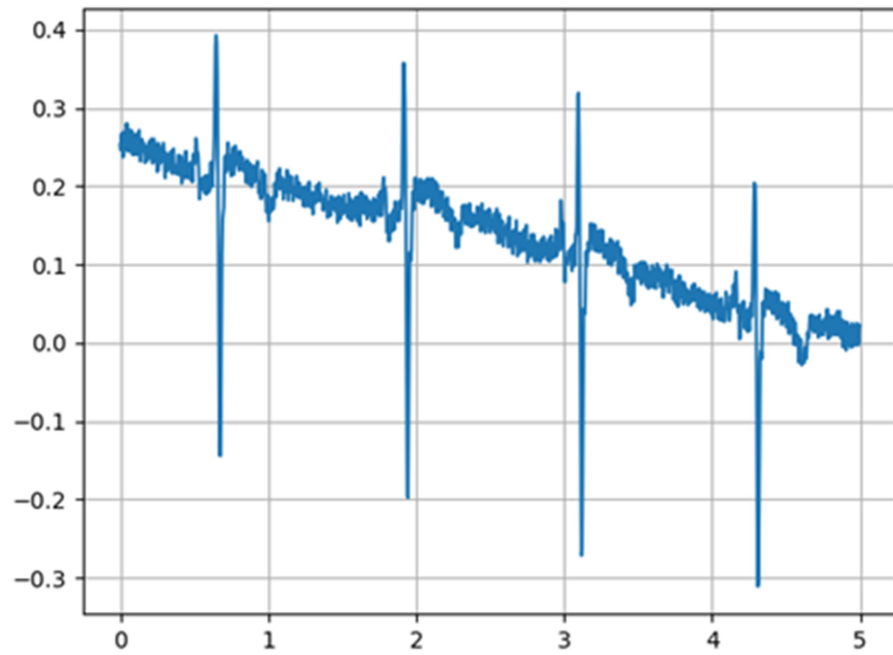


Fig. 2.16. Plotting ECG signal part (starting from 0 and till 5 s)

### Control questions

1. What are the purpose and objectives of signal processing?
2. What is the difference between continuous and discrete signal?
3. List the examples of biomedical signals.
4. List the examples of biomedical systems.

### LABORATORY WORK № 3

## MODELING OF LINEAR SYSTEMS IN TIME AND FREQUENCY DOMAIN

**Purposes:** to study the operation of linear systems for processing discrete signals; to acquire skills of modeling of linear stationary discrete systems in Python.

### Tasks

Simulation of linear discrete systems (LDS) using the difference equation.

1. Based on your own date of birth, write the difference equation:

$$y[n] + \frac{(D_1 + D_2)}{140} y[n-1] + \frac{(P_2 - D_2)}{130} y[n-2] - \frac{D_1}{150} y[n-4] - \frac{(M_1 - D_1)}{150} y[n-5] = \\ = \frac{M_1}{10} x[n] + \frac{(P_3 - D_2)}{20} x[n-1] - \frac{(M_2 - M_1)}{20} x[n-2] - \frac{P_4}{30} x[n-3] + \frac{D_2}{20} x[n-4] - \frac{M_2}{20} x[n-5]$$

$D_1 D_2$  date of your birth (for example 23 -  $D_1=2$ ,  $D_2=3$ );

$M_1 M_2$  month of your birth (for example 05 -  $M_1=0$ ,  $M_2=5$ );

$P_1 P_2 P_3 P_4$  year of your birth (for example 1998 -  $P_1=1$ ,  $P_2=9$ ,  $P_3=9$ ,  $P_4=8$ )

In the program set the vectors  $a$  and  $b$  of the coefficients of the recursive and non-recursive part of the LDS.

$$a = [1, (D_1 + D_2)/140, (P_2 - D_2)/130, 0, -D_1/150, -(M_1 - D_1)/150]$$

$$b = [M_1/10, (P_3 - D_2)/20, -(M_2 - M_1)/20, -P_4/30, D_2/20, -M_2/20]$$

2. Form samples of a sine wave of frequency 10 Hz lasting 1 sec with amplitude 1 V, sampled with a frequency of 256 Hz. Calculate the system response to the received signal (**filter** function) for two cases (Fig.3.1):

2.1. zero initial conditions;

2.2. random initial conditions (use the **rand** function).

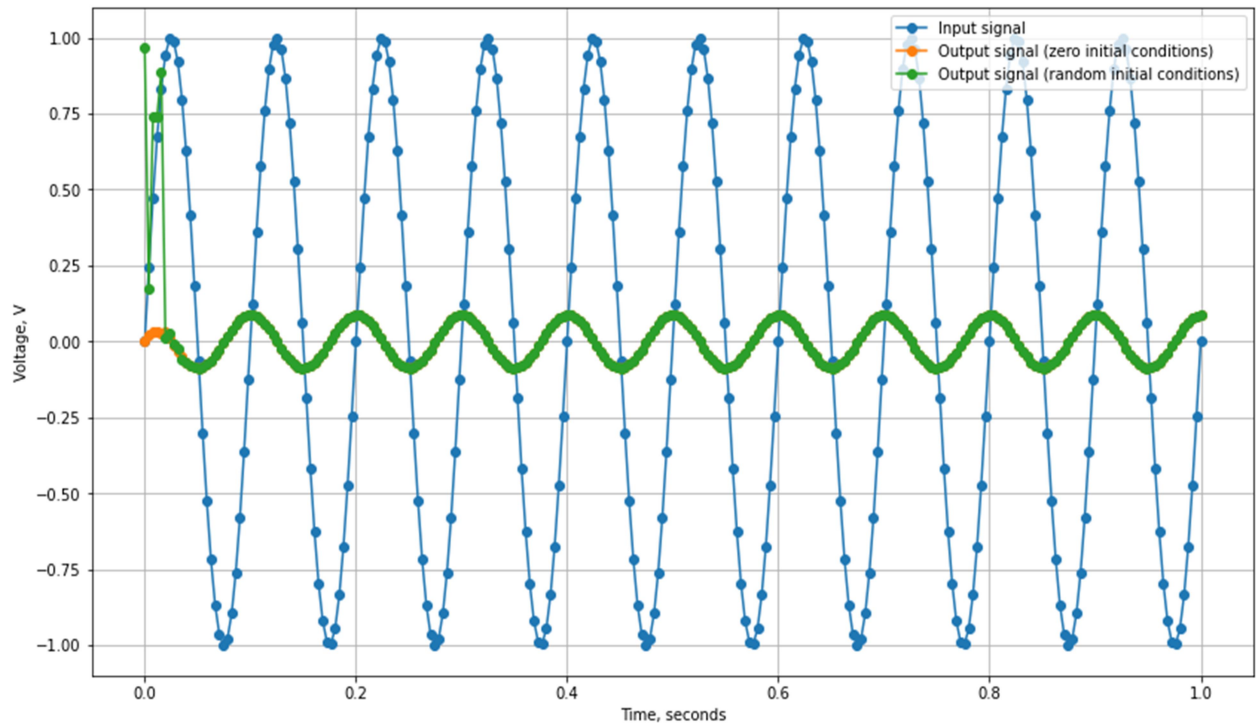


Fig. 3.1. The system response to sine wave of frequency 10 Hz lasting 1 sec with amplitude 1 V

For tasks 2.1 and 2.2, plot the input and output signals in one window, marking the points of the graph that correspond to the samples and the envelopes of the graphs. Plot in a separate window the first 100 ms of an input and output signal (Fig.3.2). Draw conclusions about the type of output signal relative to the input (shape, amplitude, distortion, gain).

3. Write a program to determine the LDS voltage transfer coefficient at a frequency of 10 Hz, (\*) and the phase difference between the output and input signal.

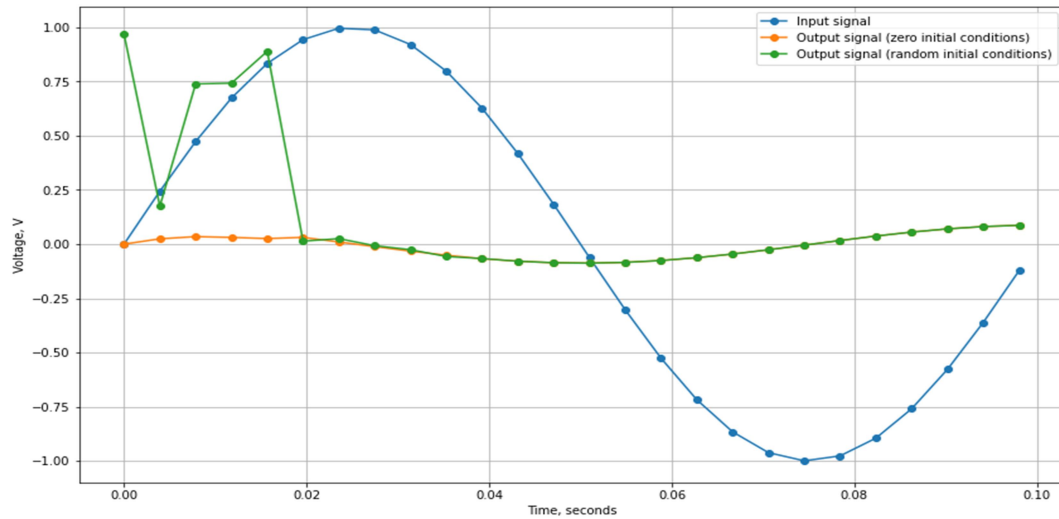


Fig.3.2. The first 100 ms of an input and output signal

4. Generate two sine waves of frequency 3 and 20 Hz lasting 1 s. Illustrate the additivity property of the system (Fig.3.3) by determining the response of the system first to each of the signals separately, and then to the sum of these signals. Illustrate also the property of system homogeneity (Fig.3.4). Provide the necessary graphs.

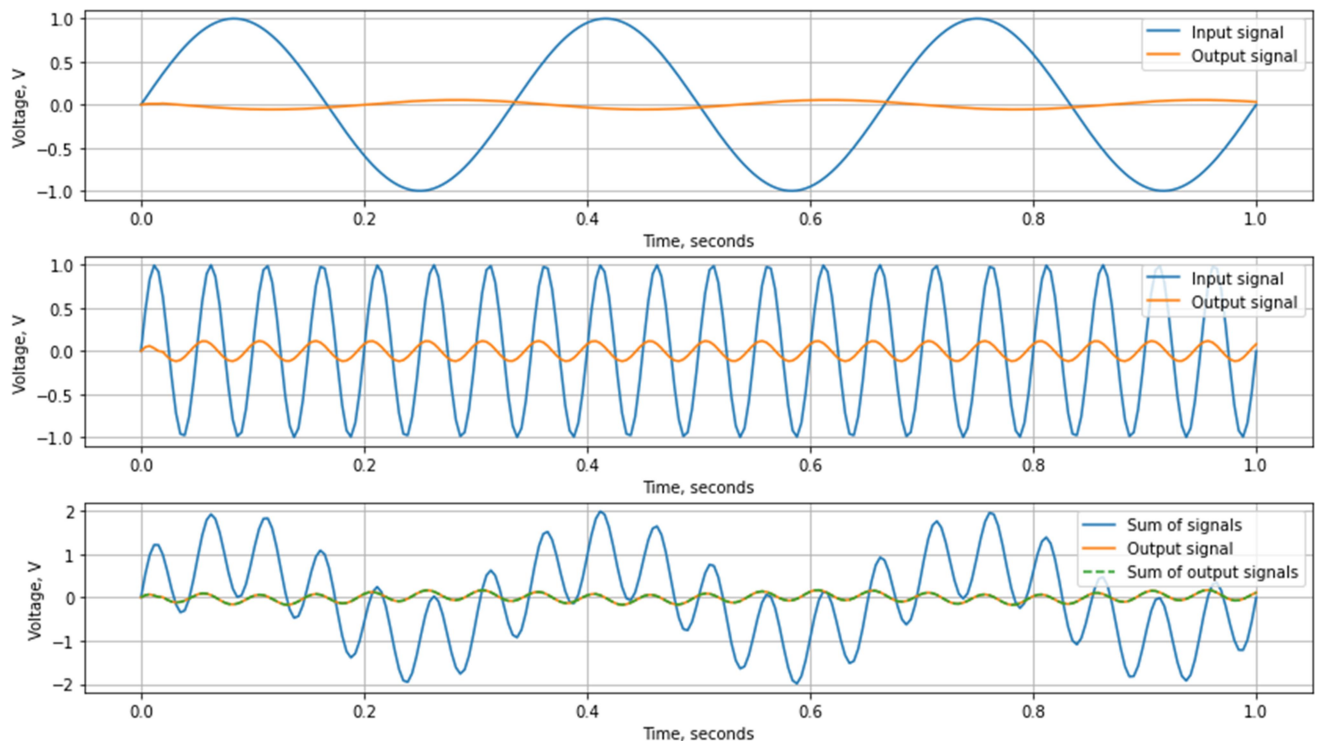


Fig.3.3. Illustration of the additivity property of the system

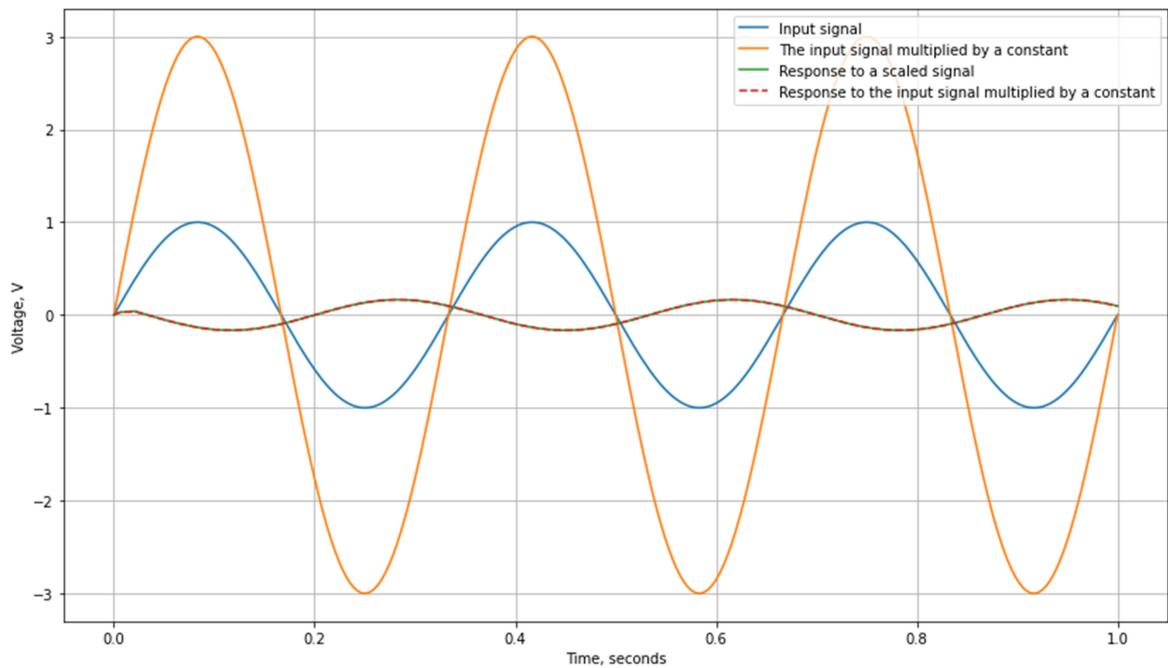


Fig.3.4. Illustration of the homogeneity property of the system

5. Calculate with the help of the **lfilter** function the first 30 samples of the impulse response of the system, applying a single pulse to the input of the system (at zero initial conditions), using the function **unit\_impulse**. Plot graphs of input and output signal (**stem** function) (Fig.3.5) .

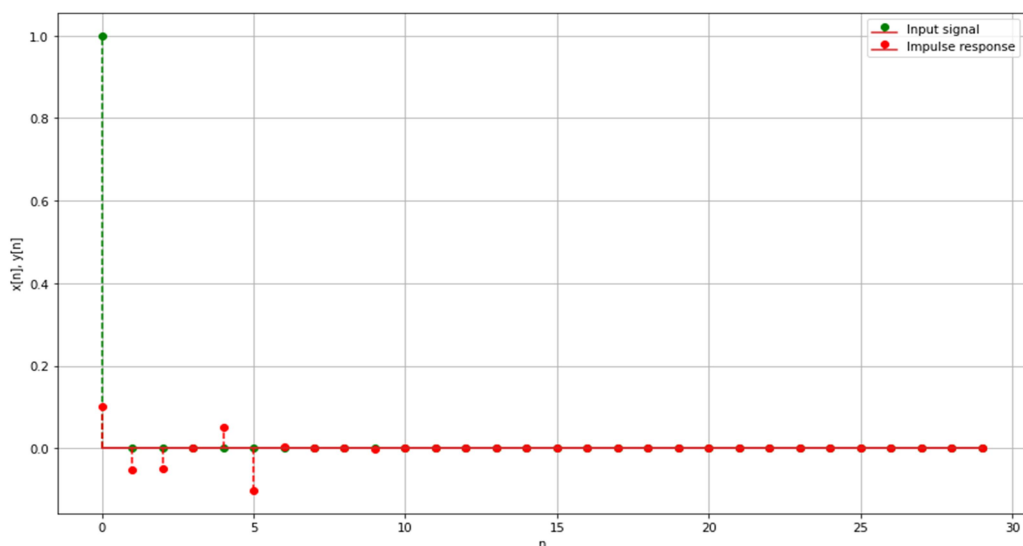


Fig.3.5. The first 30 samples of the impulse response of the system (obtained by **lfilter** function)

6. Create a system using the coefficients of the numerator and denominator of the transfer function using the function **TransferFunction** (Fig.3.6). Calculate 30

samples of the impulse response of the obtained system using the **dimpulse** function. Compare the results with the results of task 5, draw graphs, draw conclusions.

\* Calculate 100 samples of the impulse response, compare with previously obtained, draw conclusions.

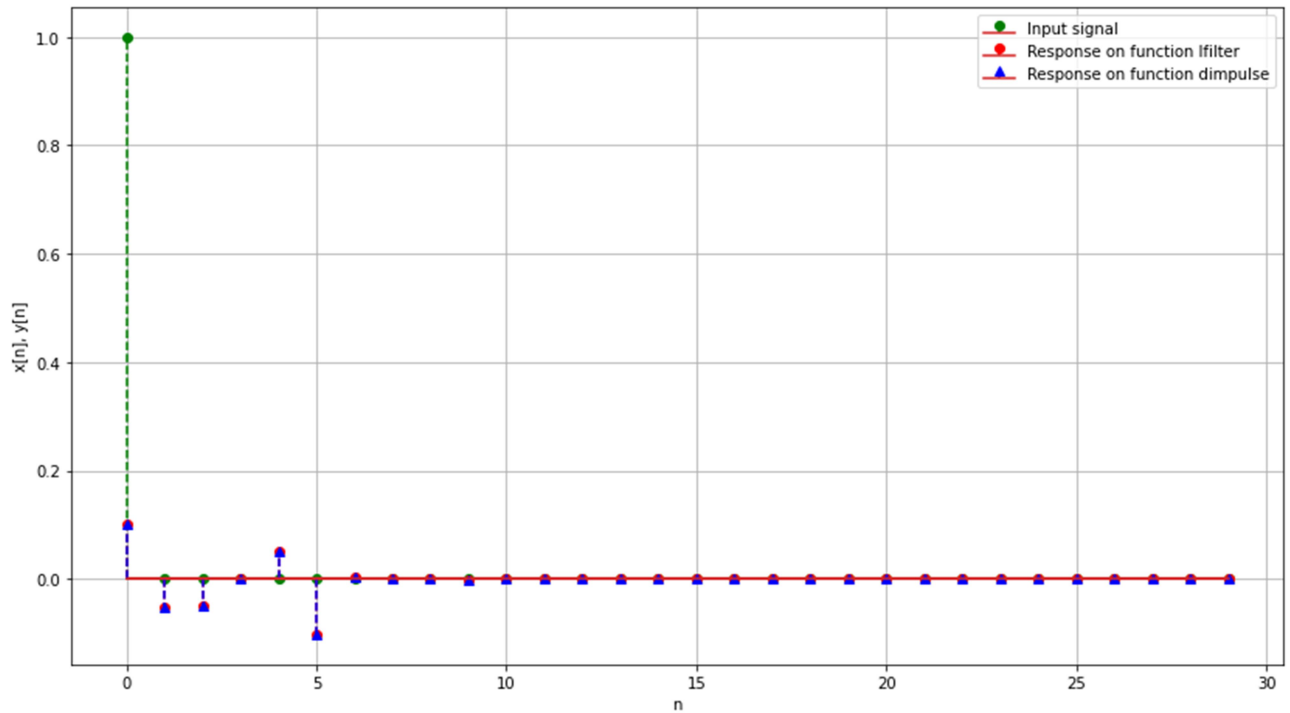


Fig.3.6. The first 30 samples of the impulse response of the system (obtained by **dimpulse** function)

### Simulation of LDS operation using the convolution equation.

7. Calculate the system response to the signal according to task 2.1 using the **convolve** convolution calculation function. Plot graphs of the input and output signal, similar to paragraph 2.1 (with zero initial conditions) (Fig.3.7). Compare all results with task 2.1. Draw conclusions.

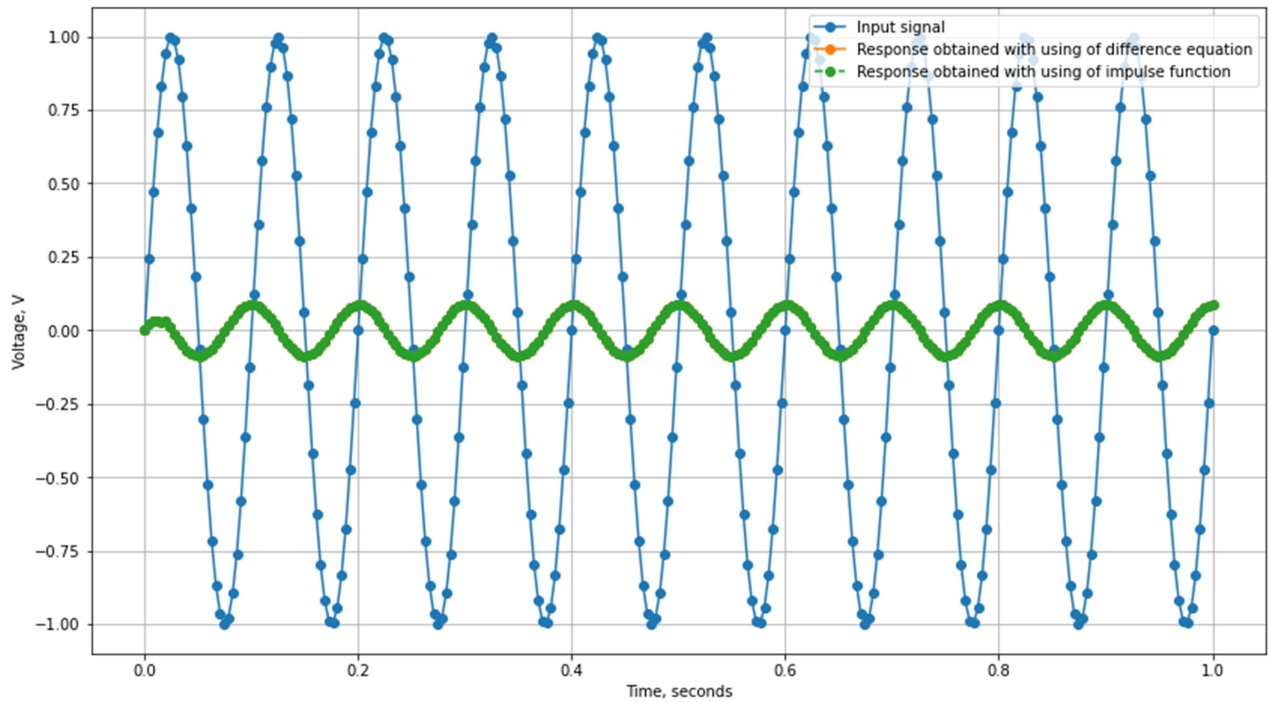


Fig.3.7. The system response obtained using convolution calculation function

### Simulation of LDS operation in the frequency domain.

8. Calculate the complex frequency response of the system using the function **freqz**. Calculate 100 FR values for a sampling frequency of 256 Hz.

9. Calculate and plot the frequency response and phase response using the complex transfer function from task 8 (Fig.3.8). Draw conclusions about the nature of the change in the modulus of the transfer coefficient of the system with frequency. Draw conclusions about the value of the transfer coefficient of the system at different frequencies. Calculate the frequency ranges at which the system amplifies the signal (write in the conclusions). Visually verify the correctness of the transfer coefficient and phase shift, which are calculated in task 3.



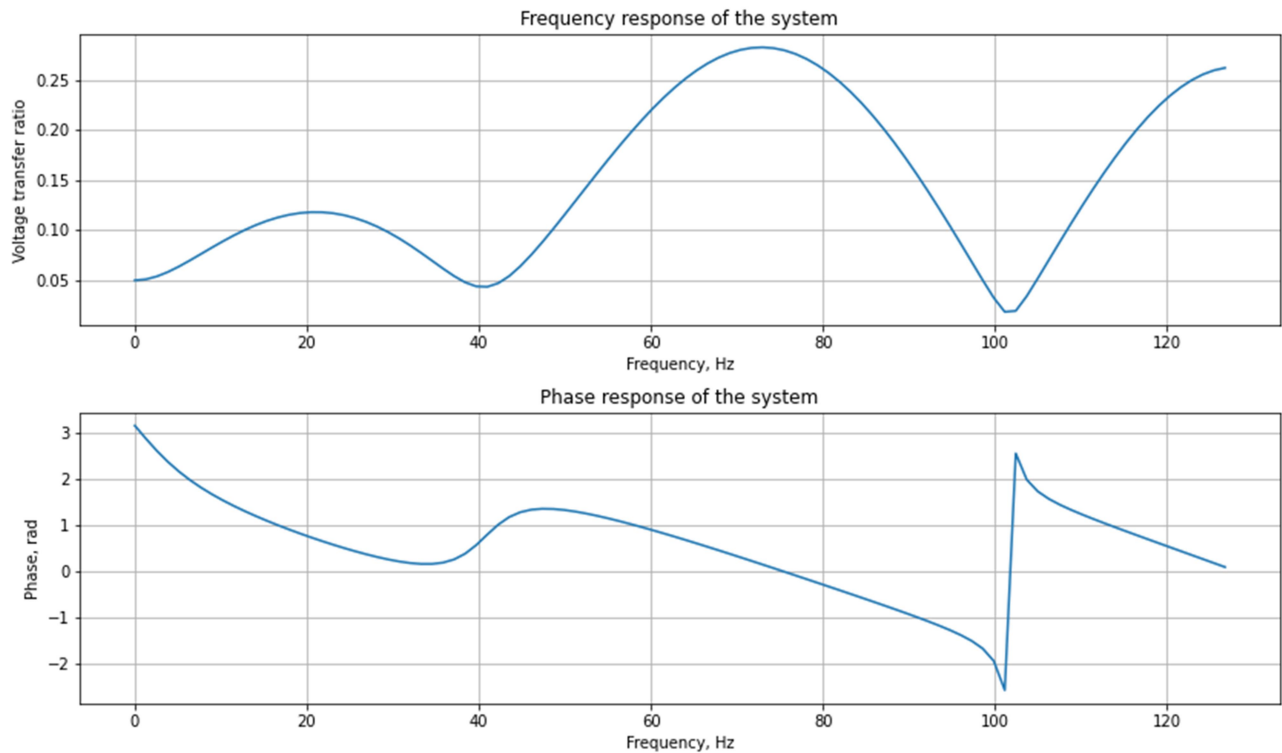


Fig.3.8. Frequency response and phase response of the system

10. Construct a function that determines the value of the frequency response and phase response of the system at an arbitrary frequency. Check the correctness of calculations according to task 2.1, taking into account that in task 2.1 the sine wave with a frequency of 10 Hz was investigated. Illustrate the operation of the function for this signal.

11. Calculate the response of LDS to a sequence of rectangular pulses with a duty cycle of 30% (Fig.3.9). Plot graphs of input and output signals, draw conclusions about the distortion of the output signal relative to the input.

12 \*. Calculate the response of the LDS to the input ECG signal lasting 5 seconds. Plot graphs of input and output signals (Fig.3.10).

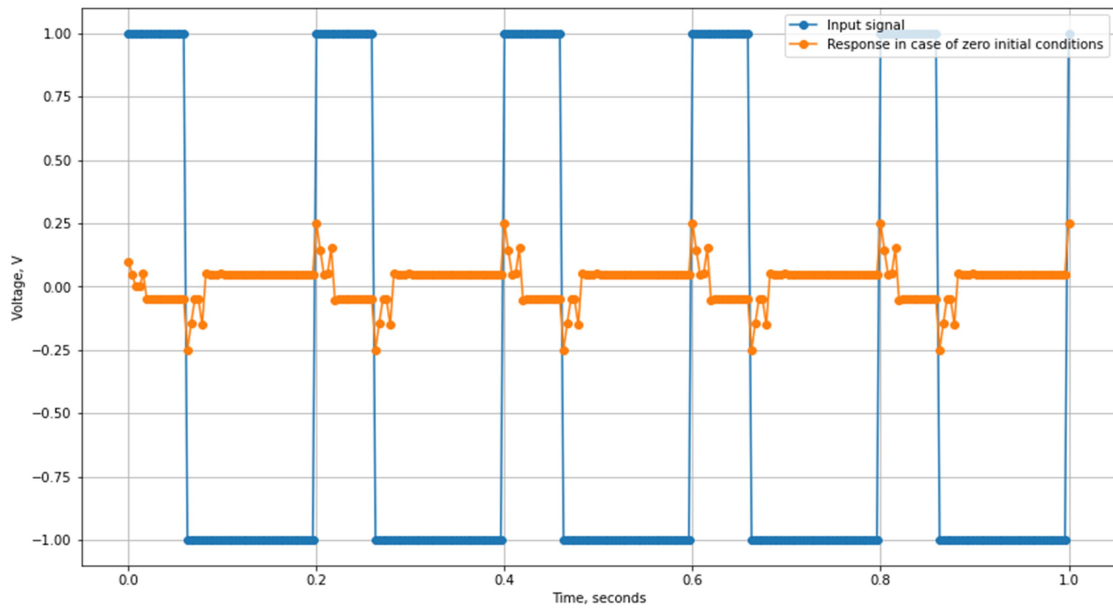


Fig.3.9. The system response to a sequence of rectangular pulses with a duty cycle of 30%

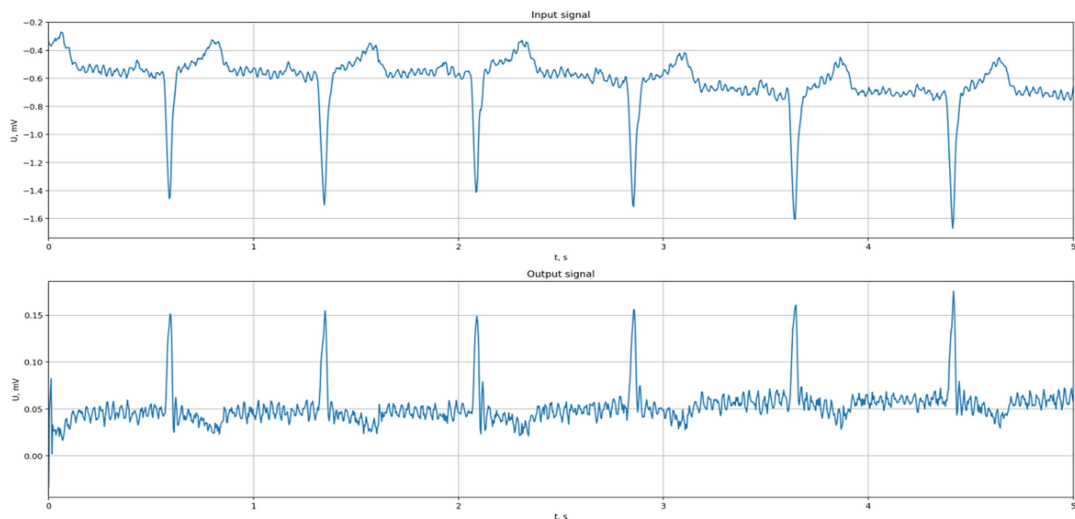


Fig.3.10. The system response to the input ECG signal lasting 5 seconds

13 \*. Calculate the response of the LDS to a previously recorded sound signal (Fig.3.11). Listen to the output signal. Draw conclusions about the distortions that the system introduces to the signals.

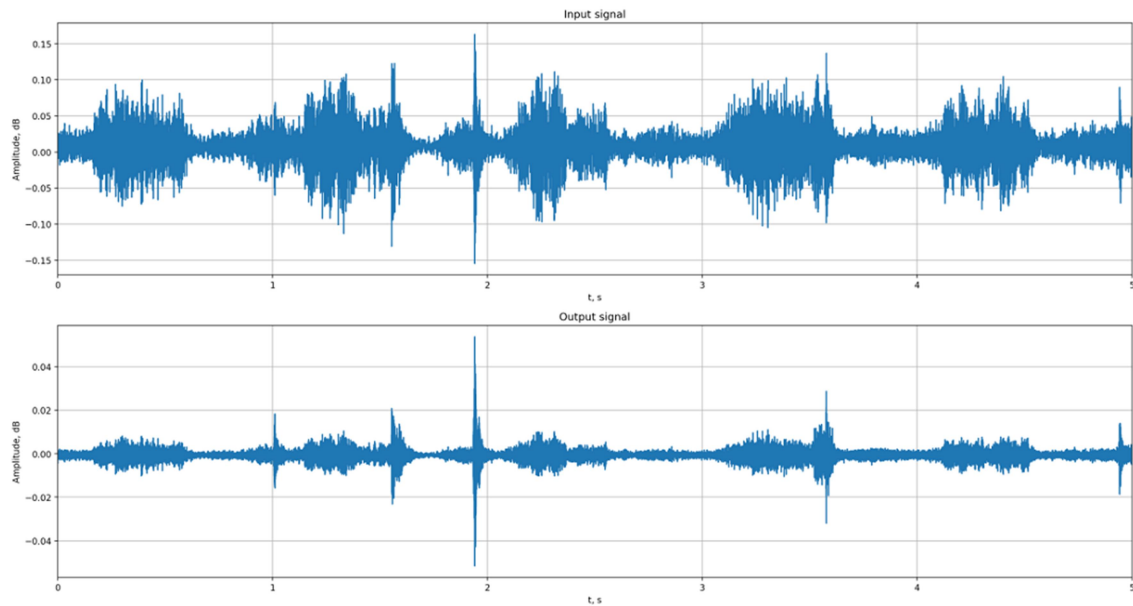


Fig. 3.11. The system response to the sound signal

14\*. Construct the LDS response to the signal from the accelerometer (Fig.3.12). Draw conclusions about the distortions that the system introduces to the signal.

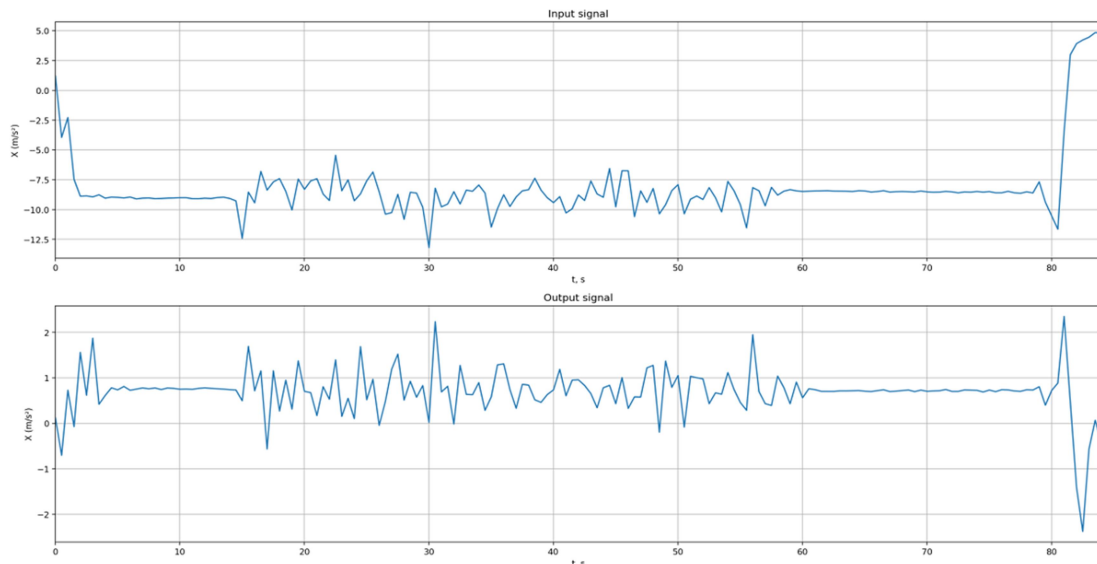


Fig.3.12. The system response to the accelerometer signal

### Control questions

1. Give definition of linear stationary discrete systems, their types and properties.
2. What is frequency and phase response of linear stationary discrete system?

## LABORATORY WORK № 4

### FOURIER SPECTRAL ANALYSIS OF SIGNALS

**Purpose:** explore the relationship between temporal and spectral characteristics of signals; to acquire skills of using the basic tools of spectral and spectral-time analysis of Fourier signals.

#### Tasks

As it is known, the spectra of all discrete signals are periodic, and the amplitude spectra are even functions of frequency. By means of Python (fft function) it is possible to calculate two halves of one period of a spectrum which are mirror copies of each other relative to the Nyquist frequency.

Therefore, in order to have an idea of the features of the signal spectrum, on all graphs of the amplitude spectrum is sufficient and it is necessary to display **only half the period of the spectrum**, because it fully describes the amplitude spectrum. In order to have an idea of the actual values of the harmonic amplitudes, the values of the samples of half the spectrum **must be divided by half the number of samples**.

To correctly present the results, on the graphs that will be included in the protocol, at the top of the figure one needs to build a graph of the signal over time, below - a graph of the amplitude spectrum, on the bottom (if necessary) - a graph of the phase spectrum. Be sure to sign all axes and indicate the units of measurement.

1. Generate a vector of time samples lasting 1 s for a sampling frequency of 128 Hz. Generate signals of a sine wave with frequency 2, 2.5, 40, 100, 600 Hz. Take into account the need to comply with the frequency of the discrete signal to obtain an adequate spectrum. Construct graphs of signals using the **plot** function and graphs of their amplitude spectra using the **stem** function (Fig.4.1-4.5). Draw conclusions about the correspondence of the obtained spectra to those that should be obtained according to theoretical considerations.

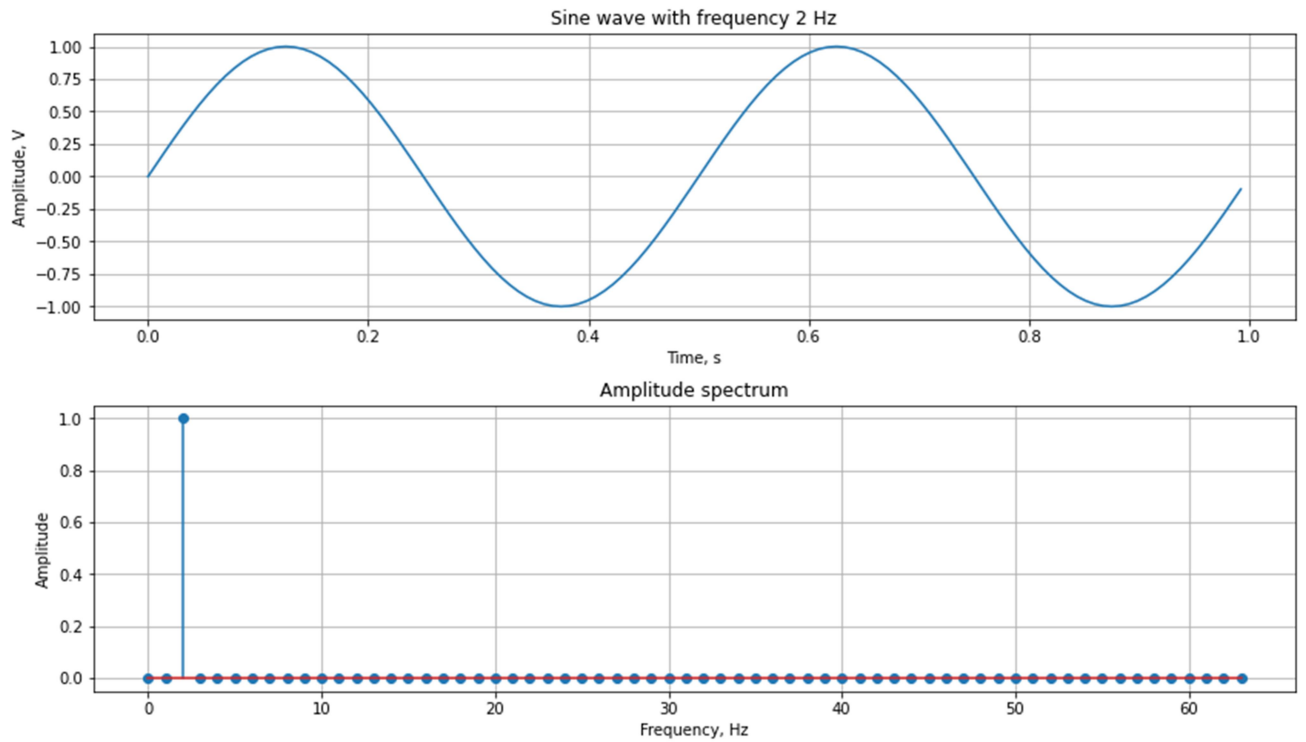


Fig.4.1. Signal of the sine wave with frequency 2 Hz and its amplitude spectrum

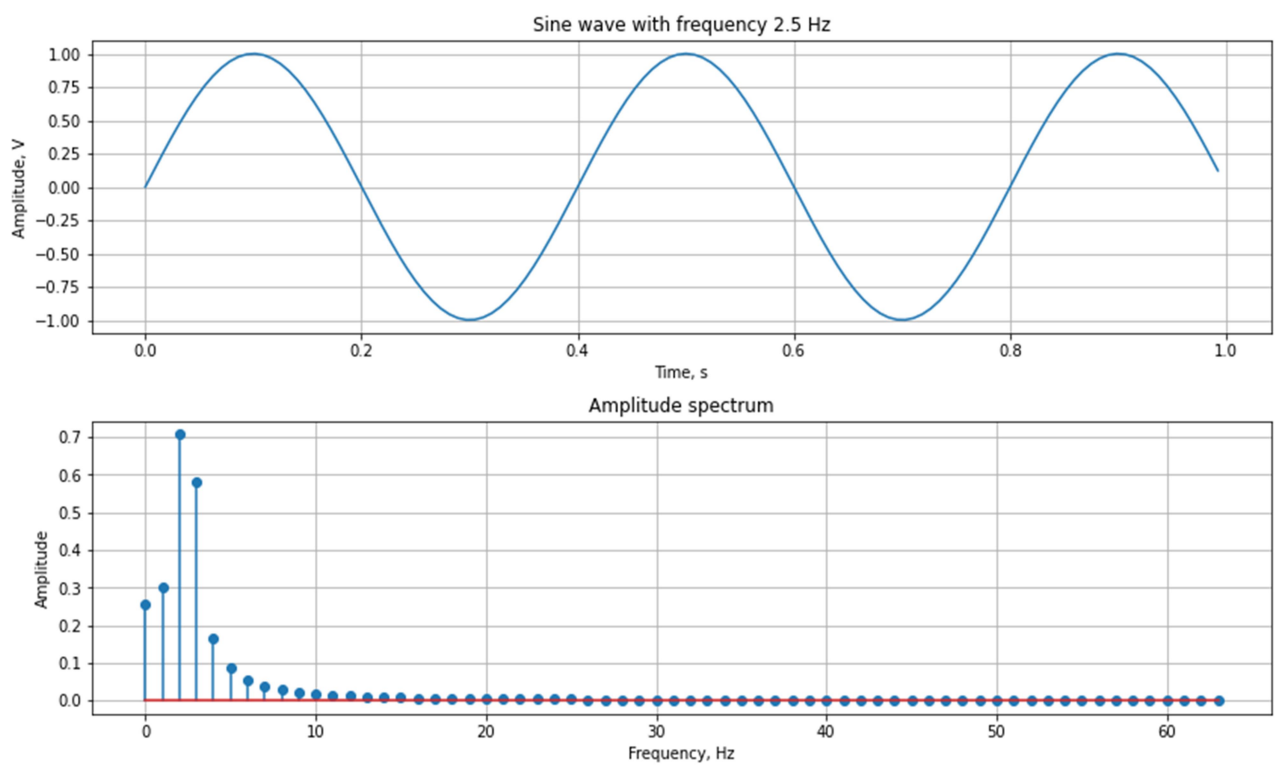


Fig.4.2. Signal of the sine wave with frequency 2.5 Hz and its amplitude spectrum

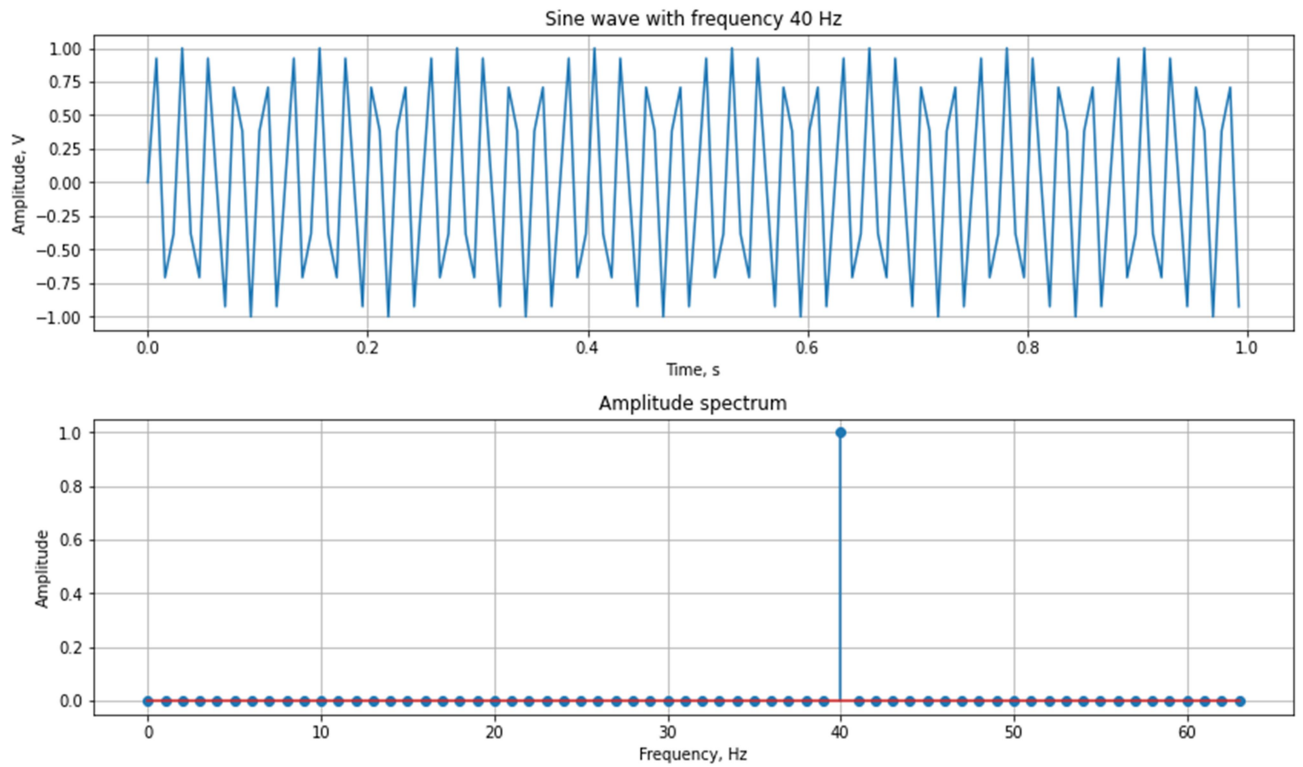


Fig.4.3. Signal of the sine wave with frequency 40 Hz and its amplitude spectrum

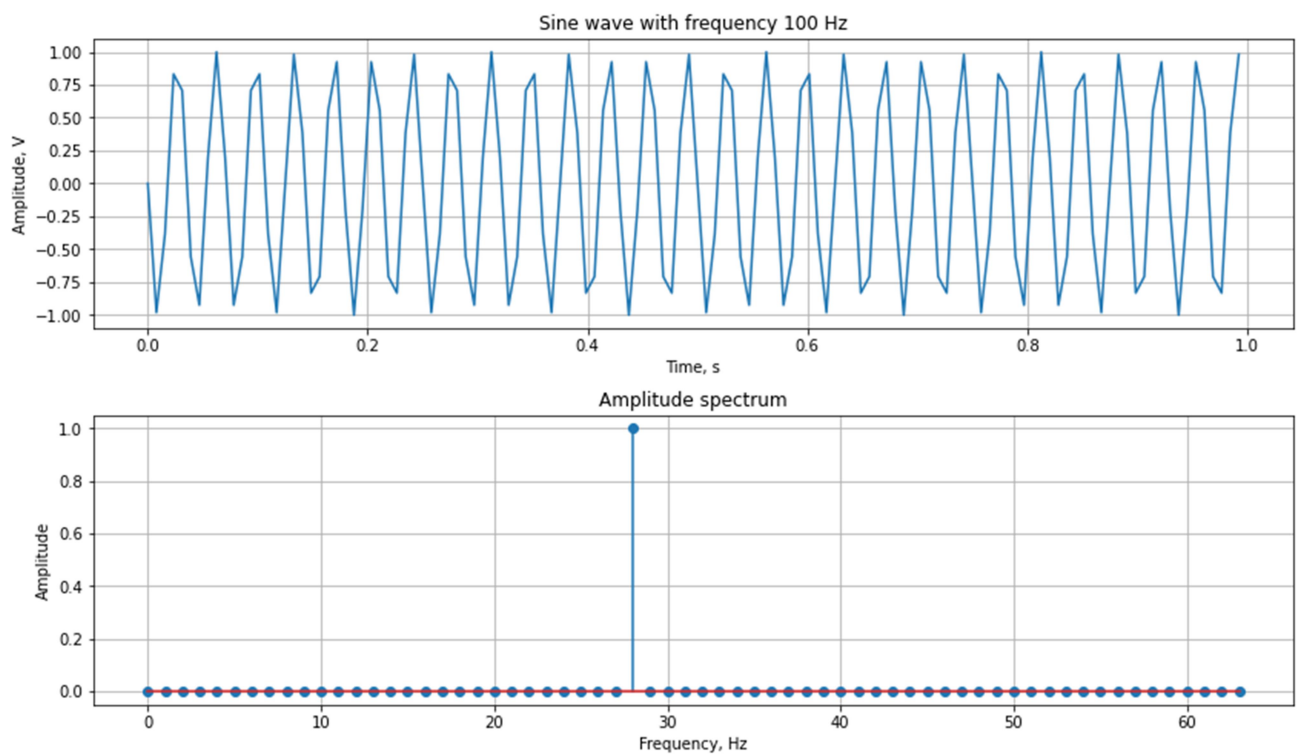


Fig.4.4. Signal of the sine wave with frequency 100 Hz and its amplitude spectrum

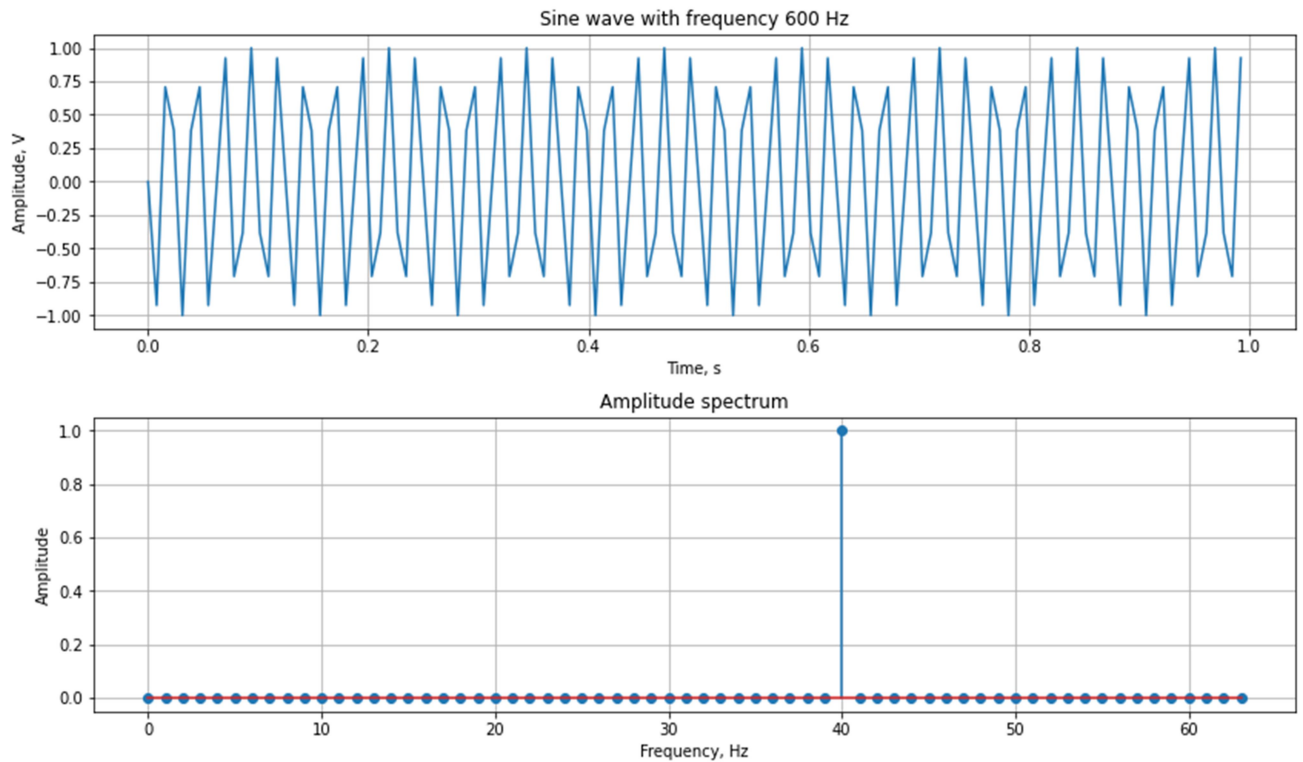


Fig.4.5. Signal of the sine wave with frequency 600 Hz and its amplitude spectrum

2. Generate a vector of time samples lasting 10 s for a sampling frequency of 256 Hz. Generate signals of the sine wave section with a frequency of 10 Hz (S1) and 100 Hz (S2). Generate three signals based on them:

- 2.1. a signal (lasting 10 s) equal to the sum of these two signals;
- 2.2. a signal that first contains a signal  $2 * S1$ , and then a signal  $2 * S2$  (will last 20 seconds);
- 2.3. a signal that first contains a  $2 * S2$  signal and then a  $2 * S1$  signal (will last 20 seconds).

Plot graphs of signals 2.1-2.3 (**plot** function) and their amplitude spectra (**stem** function) (Fig.4.6-4.8). Draw conclusions about the possibility of distinguishing the oscillations present in the signal by their spectral composition, as well as about the correspondence of the properties of signals in time domain and their spectra.

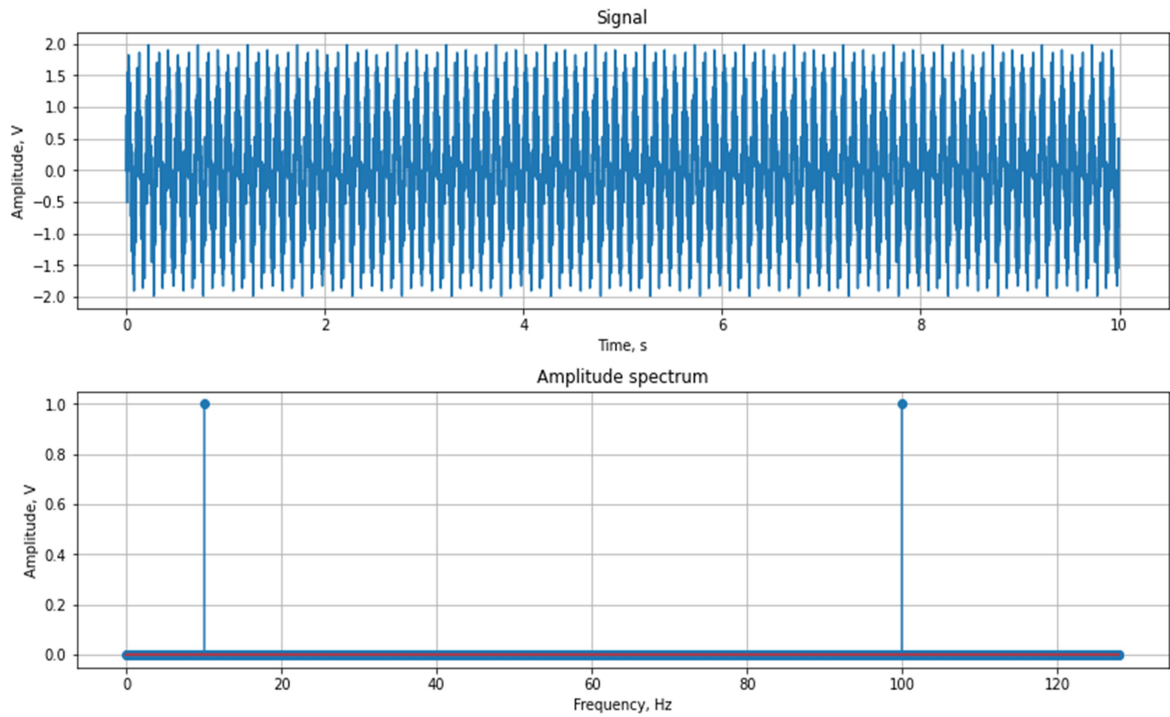


Fig.4.6. Sum of two sine waves of 10 and 100 Hz frequency and its amplitude spectrum

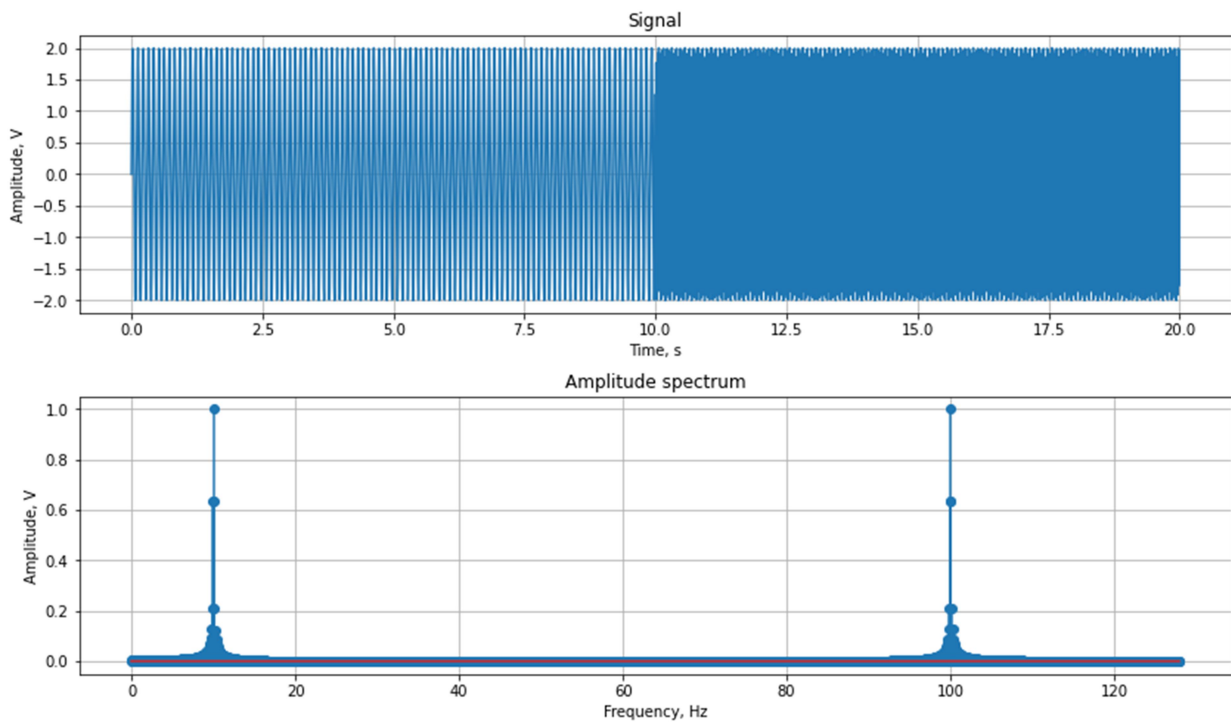


Fig.4.7. Concatenation of two sine waves of 10 and 100 Hz frequency and its amplitude spectrum



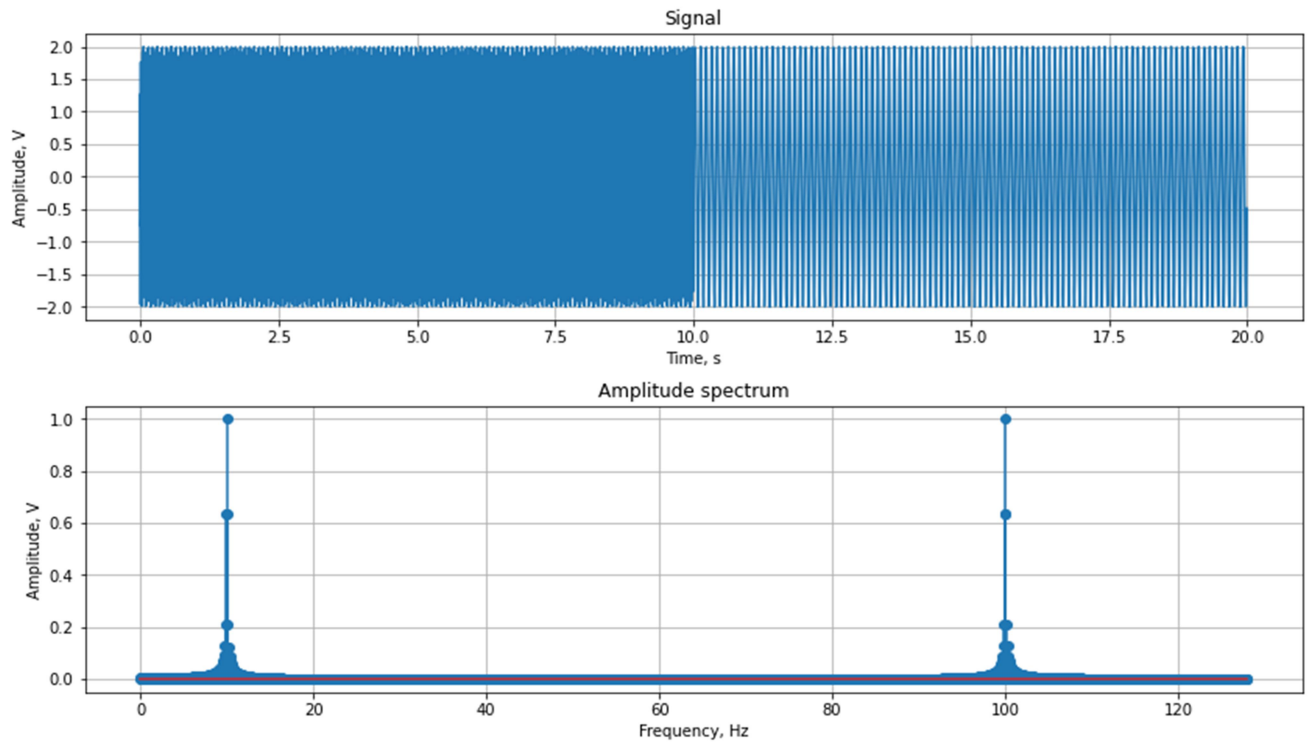


Fig.4.8. Concatenation of two sine waves of 100 and 10 Hz frequency and its amplitude spectrum

3. Generate a vector of time samples lasting 3 s for a sampling frequency of 128 Hz. Generate a signal of a sine wave with a frequency of 20 Hz. Create a gap (insert 10 zero samples instead of signal samples) in the signal at a time of 1.05 s. Get the signal spectrum. Move the gap at time 2 s, calculate the spectrum. Plot graphs of two signals with gaps and their amplitude spectra (Fig.4.9, 4.10). Draw conclusions about whether it is possible to determine the presence and exact location of the gap in the signal by analyzing the signal spectrum.

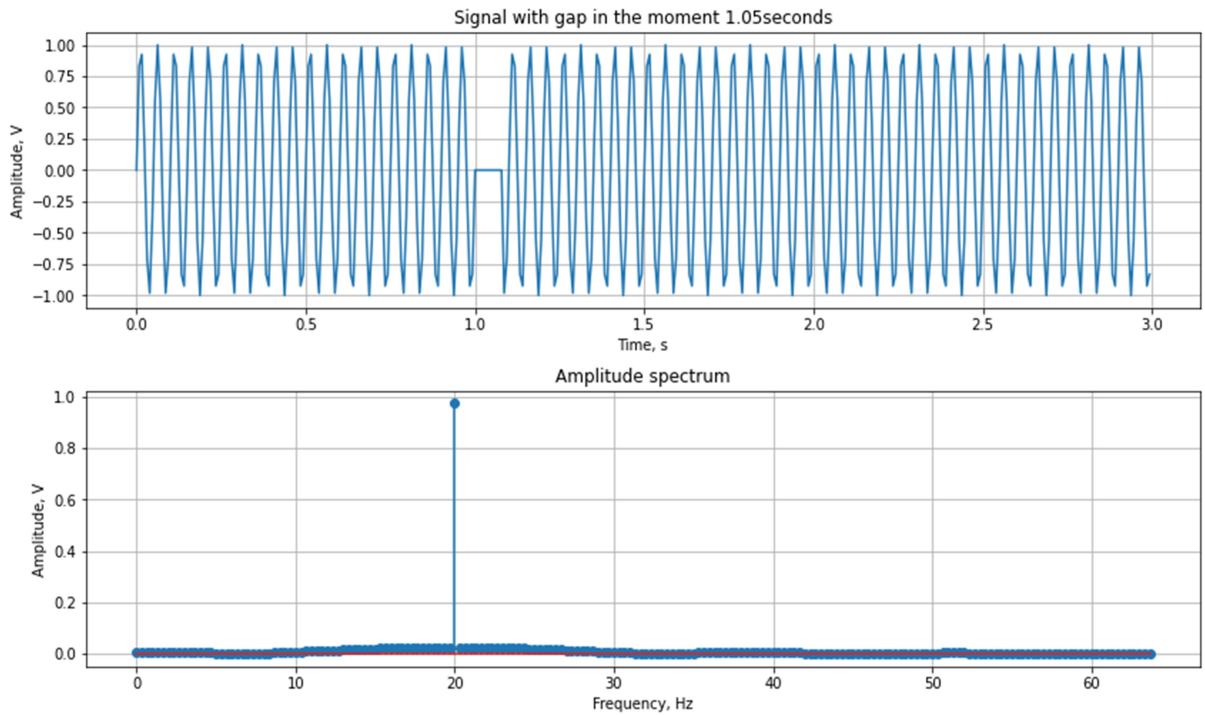


Fig.4.9. Signal of the sine wave with gap in the signal at time moment of 1.05 s and its amplitude spectrum

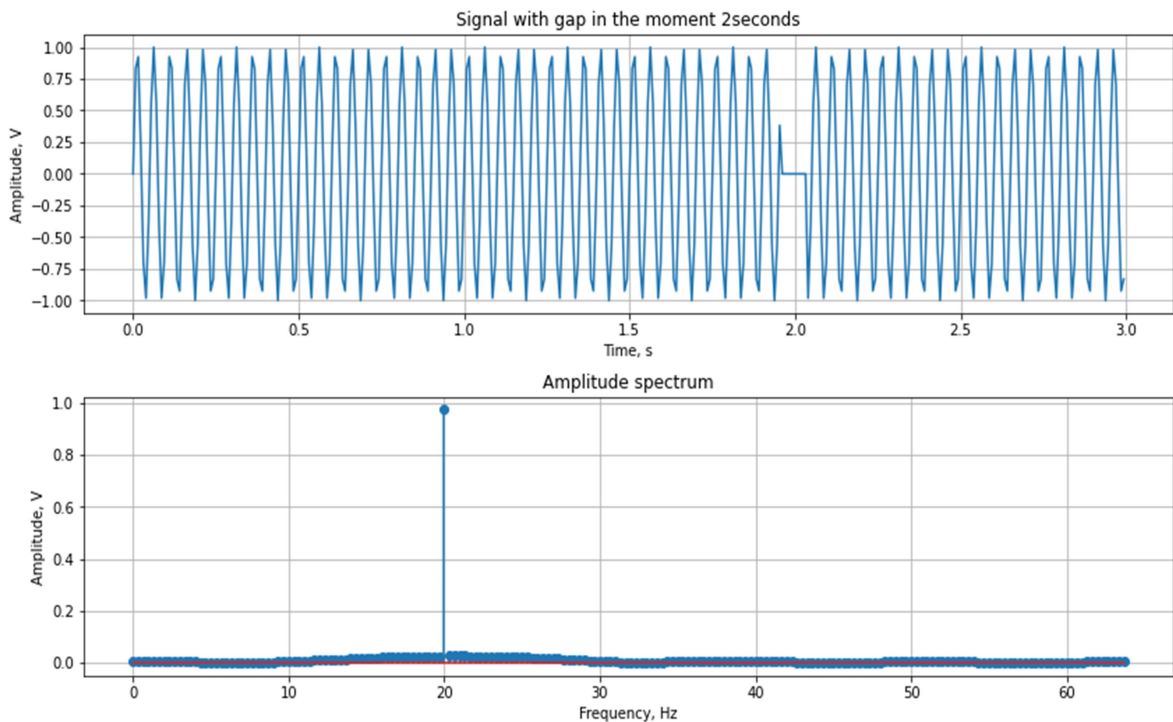


Fig.4.10. Signal of the sine wave with gap in the signal at time moment of 2 s and its amplitude spectrum

4. Generate a vector of time samples with a duration of 3 s for a sampling frequency of 512 Hz. Generate a signal of a sequence of rectangular pulses with a

frequency of 10 and 100 Hz. Construct with the help of **plot** function graphs of signals and their amplitude spectra, draw conclusions (Fig.4.11, 4.12). Build graphs for such frequencies that the features of the spectrum can be seen (plot part of the spectrum at lower frequencies).

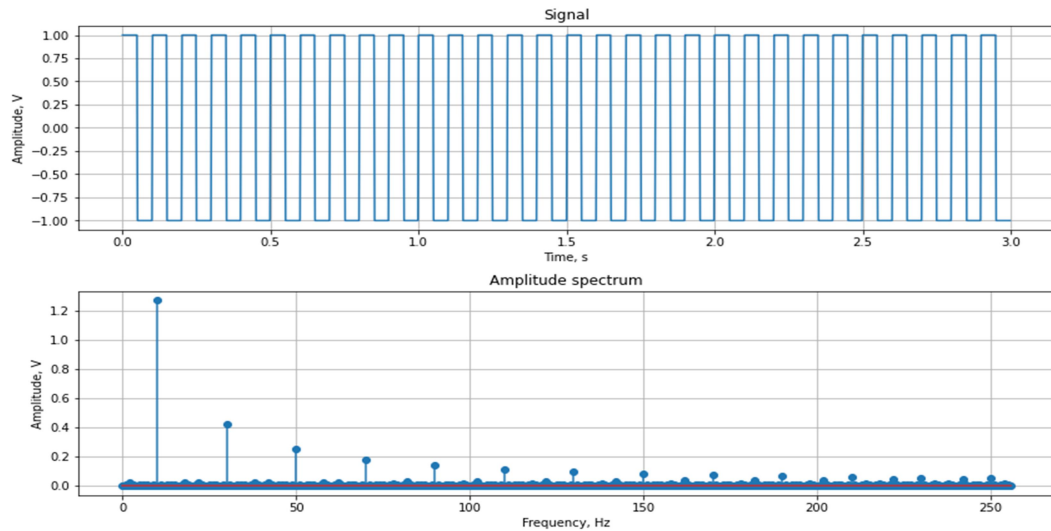


Fig.4.11. Sequence of rectangular pulses with a frequency of 10 Hz and its amplitude spectrum

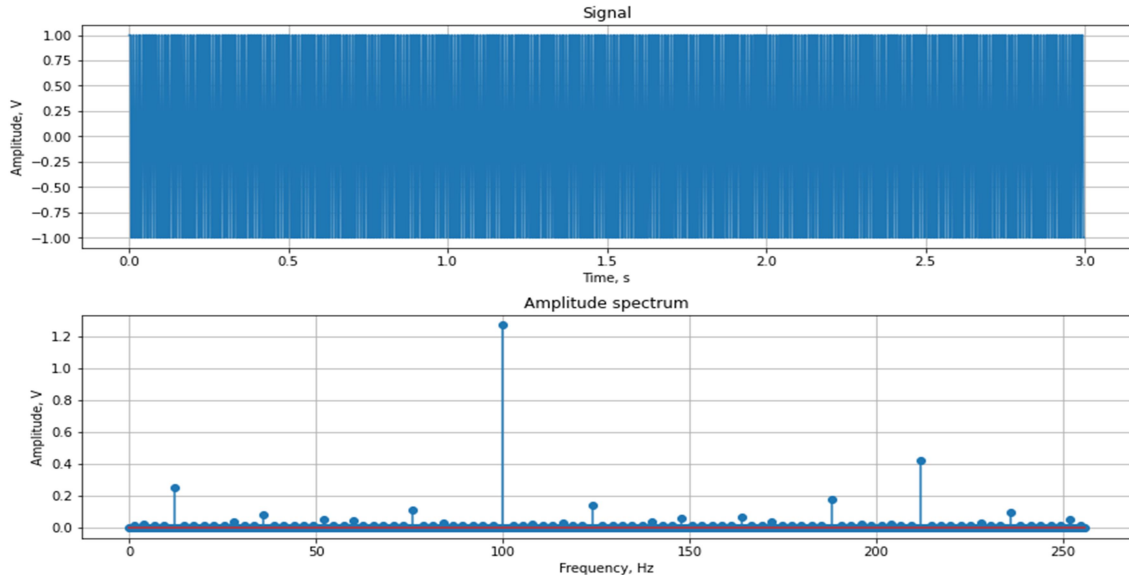


Fig.4.12. Sequence of rectangular pulses with a frequency of 100 Hz and its amplitude spectrum

5. Generate a time sample vector lasting 30 s for a sampling frequency of 512 Hz. Generate a single rectangular pulse signal for a pulse duration of 0.1, 1, 10 sec (for offset values relative to the start of time 0 and 5 s). Construct graphs of these 6 signals and their amplitude and phase spectra using the **plot** function (**angle**

function), draw conclusions (Fig.4.13, 4.14). Build graphs for such frequencies that the features of the spectrum can be seen.

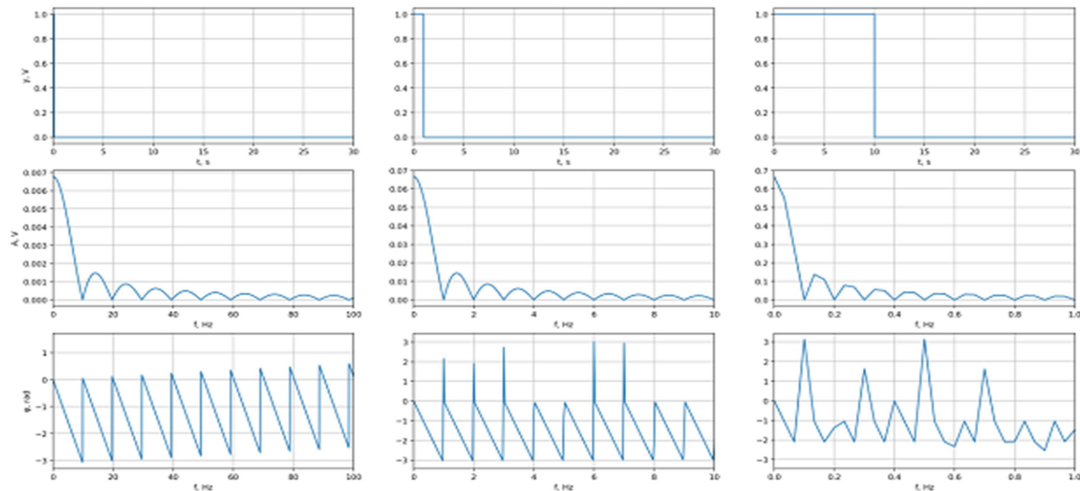


Fig. 4.13. Single rectangular pulses for a pulse duration of 0.1, 1, 10 sec and their spectra

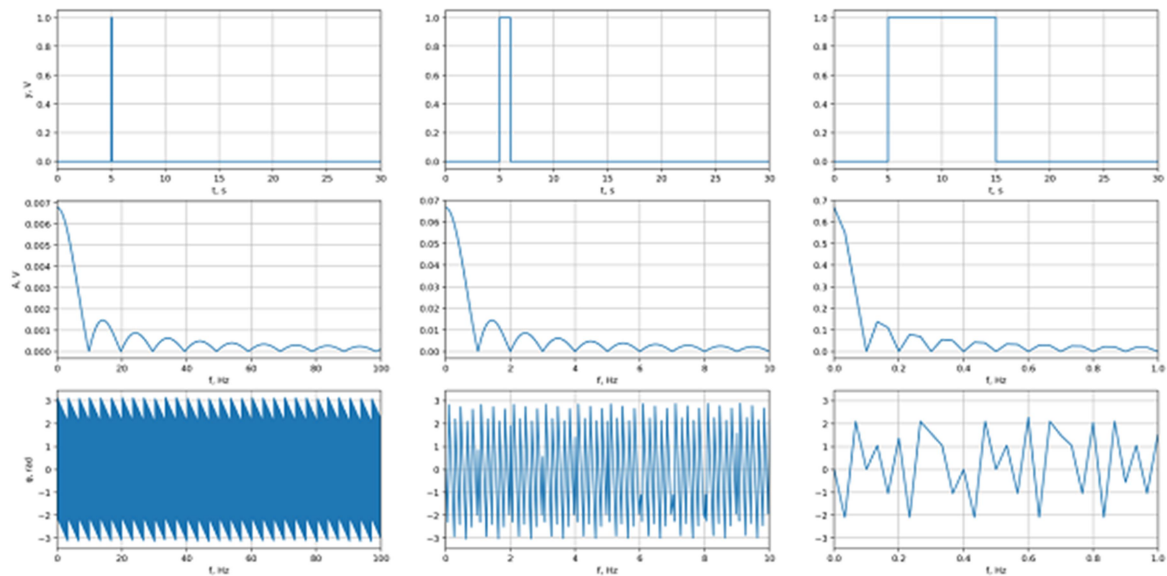


Fig. 4.14. Single rectangular pulses for a pulse duration of 0.1, 1, 10 s and offset time 5 s and their spectra

6. Generate a random signal with a duration of 10 s for a sampling frequency of 1000 Hz. Build a graph of the signal and its amplitude spectrum using the plot function, draw conclusions (Fig.4.15).

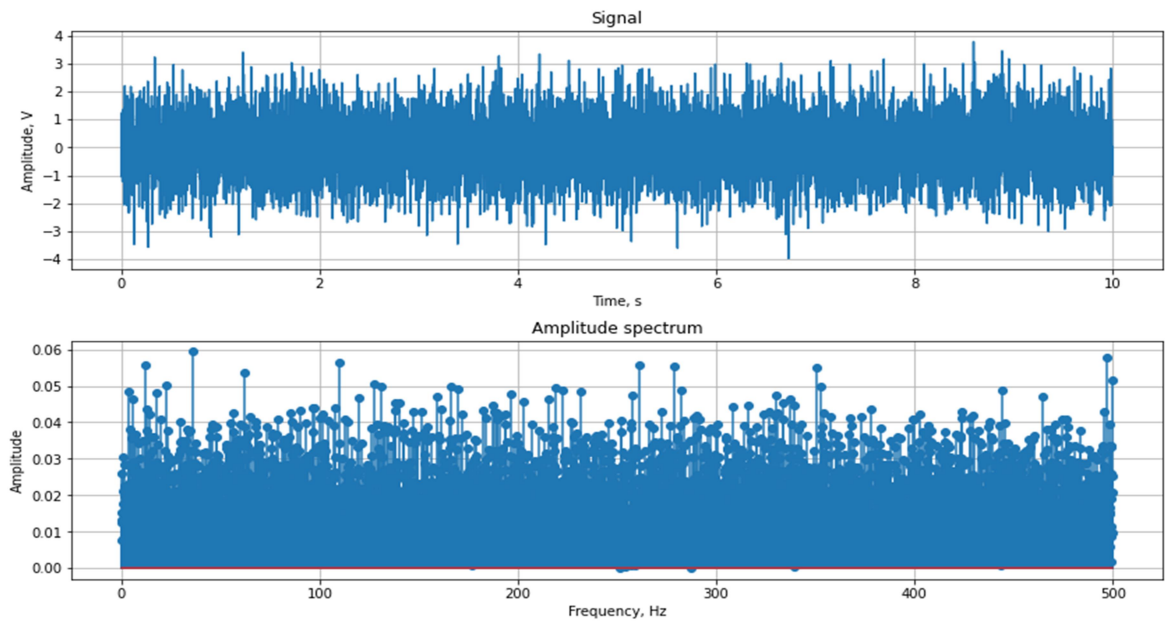


Fig.4.15. Random signal with a duration of 10 s for a sampling frequency of 1000 Hz and its amplitude spectrum

7. Generate a sine wave with a frequency of 20 Hz, 1 V magnitude and a duration of 0.5 s for a sampling frequency of 128 Hz. Add to this signal a noise component of amplitude 10 V with zero mean value (**randn** function). Get the amplitude spectrum of a given signal (Fig.4.16-4.20).

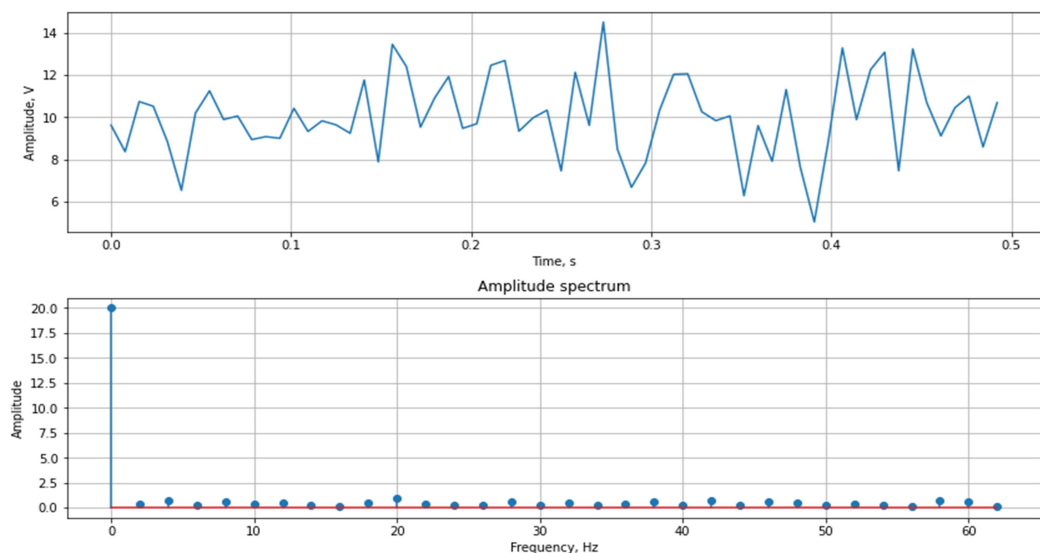


Fig.4.16. Sine wave with noise component (duration 0.5 s) and its amplitude spectrum

7.1. Increase the duration of the noisy signal (1, 10, 100, 1000 sec.), Obtaining for each signal its amplitude spectrum. Using the **plot** function, construct graphs of signals and their amplitude spectra, draw conclusions about the influence of signal duration on the resolution in the frequency domain and on the quality of determining the presence of sinusoidal oscillations in noise by spectral characteristics.

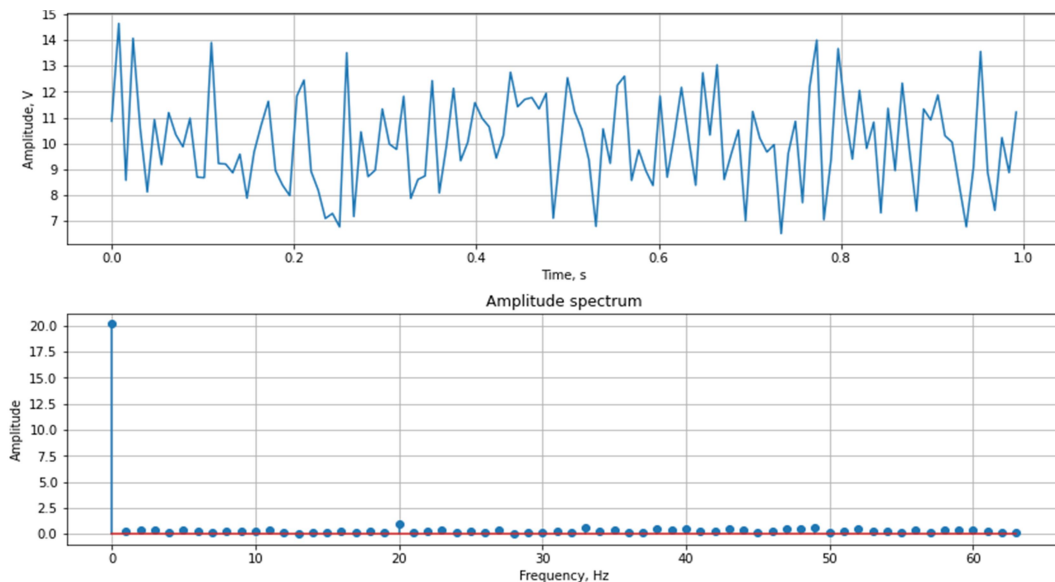


Fig.4.17. Sine wave with noise component (duration 1 s) and its amplitude spectrum

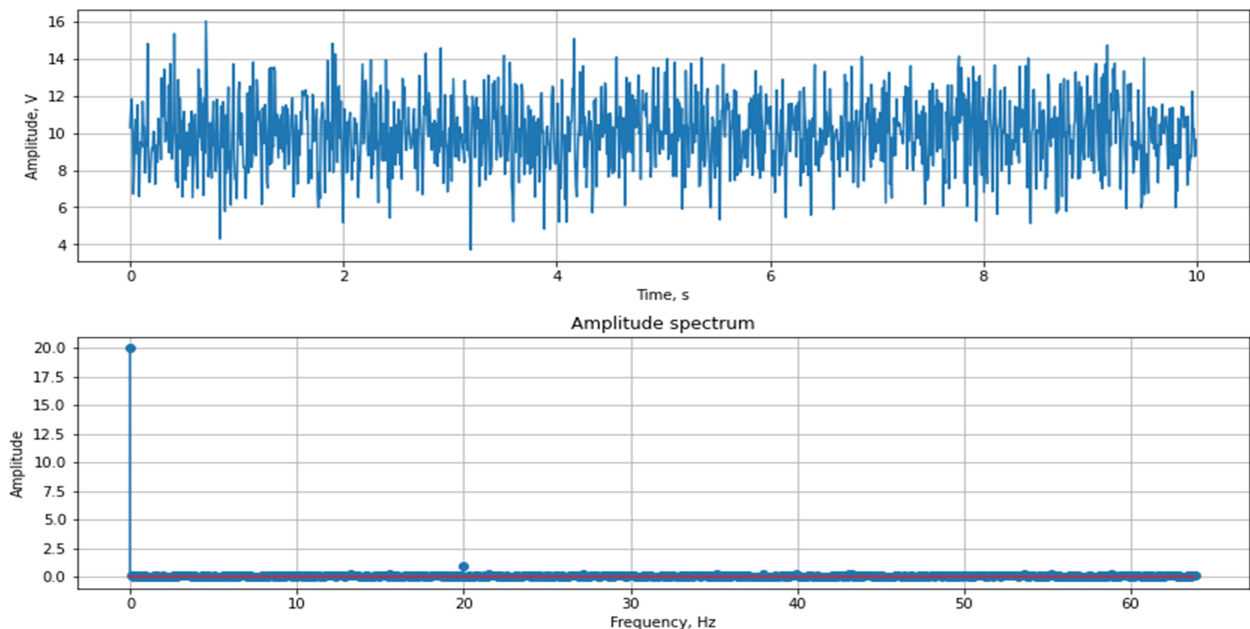


Fig.4.18. Sine wave with noise component (duration 10 s) and its amplitude spectrum

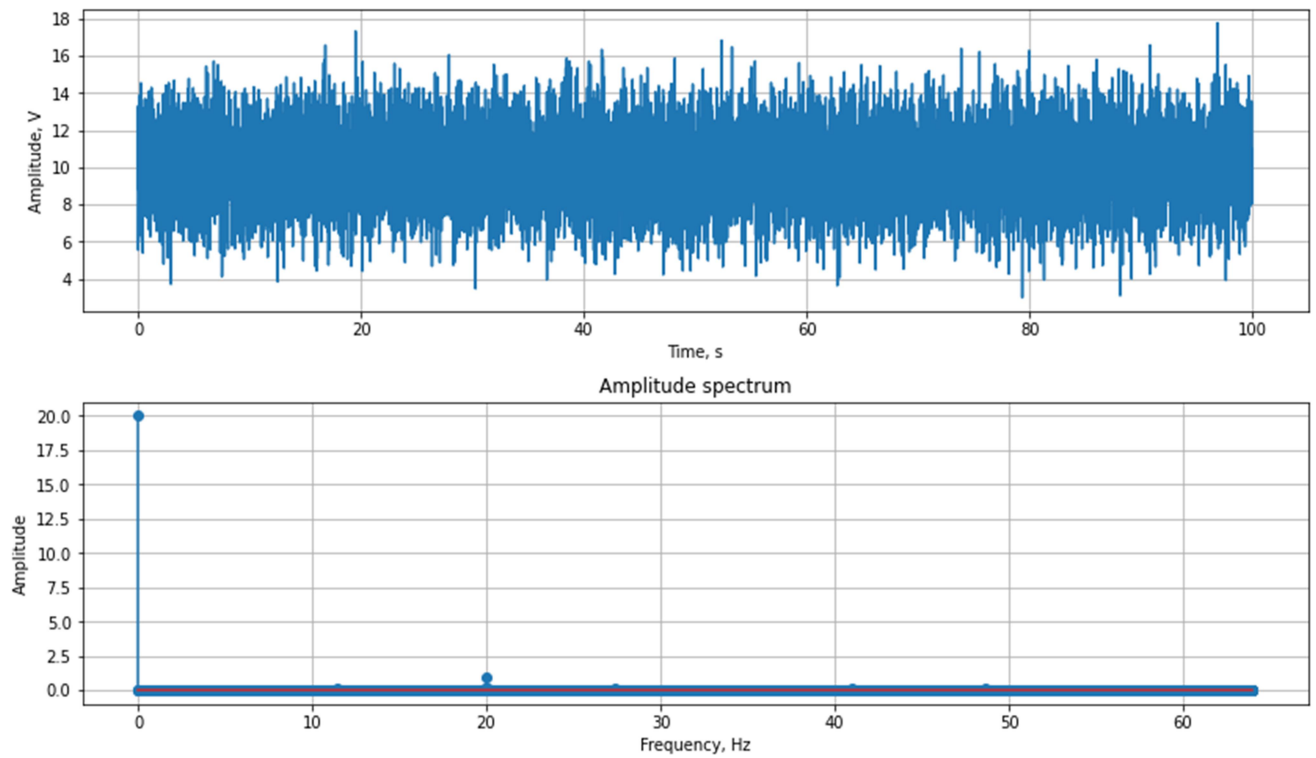


Fig.4.19. Sine wave with noise component (duration 100 s) and its amplitude spectrum

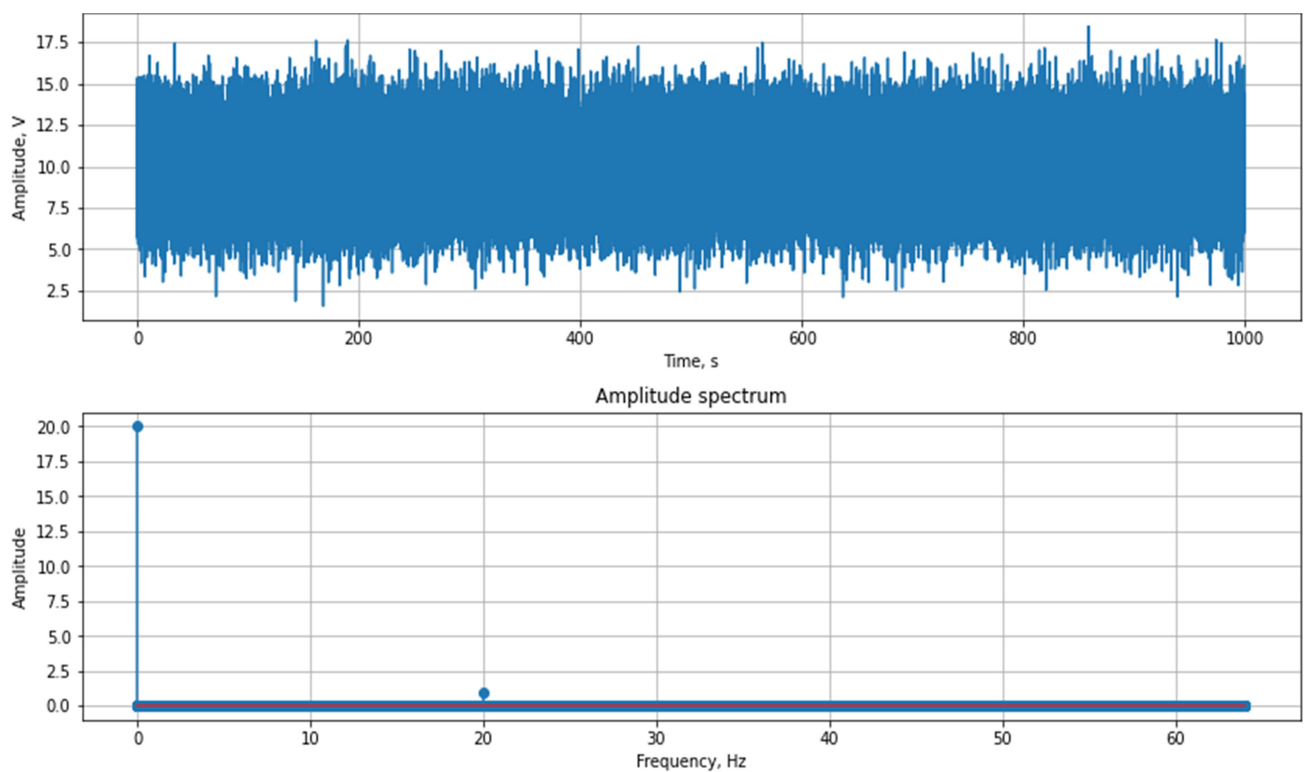


Fig.4.20. Sine wave with noise component (duration 1000 s) and its amplitude spectrum



7.2. Increase the sampling frequency of the noisy signal with a duration of 0.5 sec (1280, 12800, 128000 Hz), obtaining for each signal its amplitude spectrum. On the graph plot spectrum in the range from 0 to 100 Hz.

Using the **plot** function, construct graphs of signals and their amplitude spectra, draw conclusions about the influence of signal sampling frequency on the resolution in the frequency domain and on the quality of determining the presence of sinusoidal oscillations in noise by spectral characteristics (Fig.4.21-4.23).

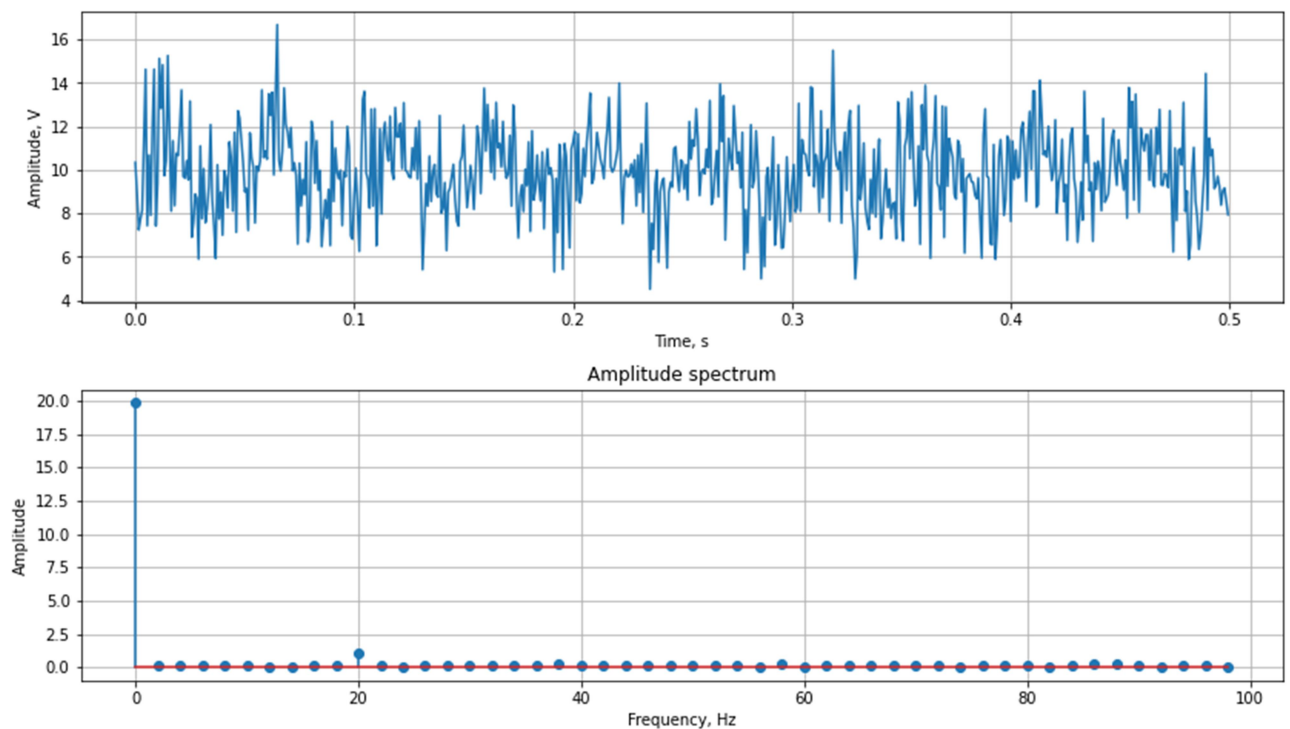


Fig.4.21. Sine wave with noise component (duration 0,5 s and sampling rate 1280 Hz) and its amplitude spectrum



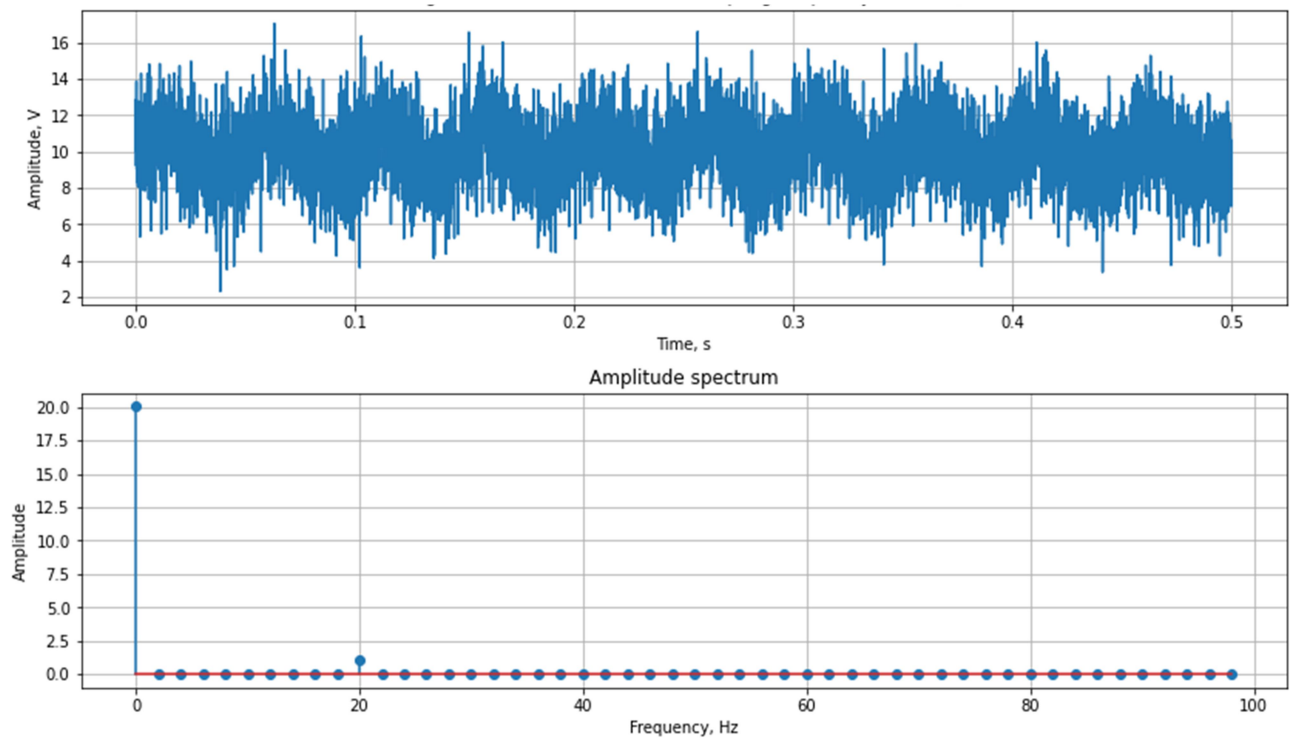


Fig.4.22. Sine wave with noise component (duration 0,5 s and sampling rate 12800 Hz) and its amplitude spectrum

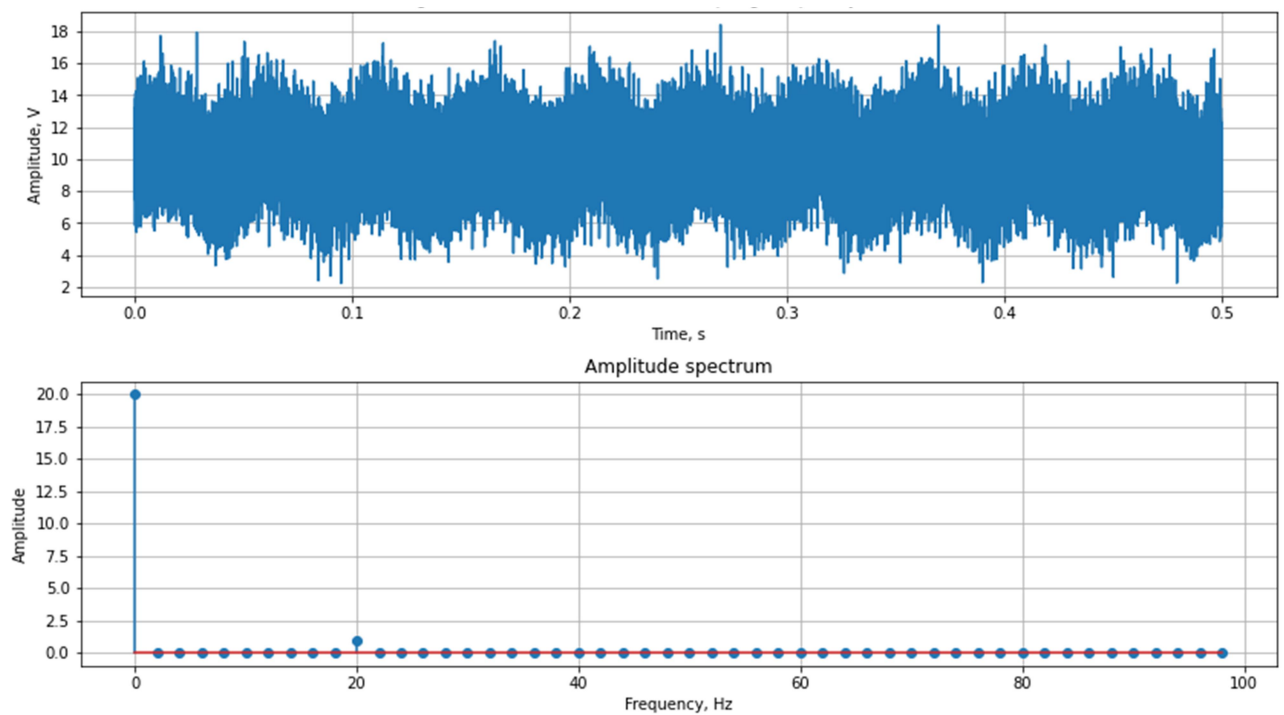


Fig.4.23. Sine wave with noise component (duration 0,5 s and sampling rate 12800 Hz) and its amplitude spectrum

8. Generate a signal of a sinusoid with a frequency of 20.5 Hz, amplitude 1 V and duration 1 sec. for a sampling frequency of 1000 Hz. Get the amplitude spectrum of a given signal.

Add zero samples (10, 100, 1000 and 10000 samples) at the end of the signal, obtaining its amplitude spectrum for each case (Fig.4.24-4.28).

Plot the spectrum on the graph each time using the **stem** function in the range from 19 to 22 Hz. Draw conclusions about the influence of the complement of the signal with zeros on the resolution in the frequency domain and on the quality of determining the presence of sinusoidal oscillations by spectral characteristics.

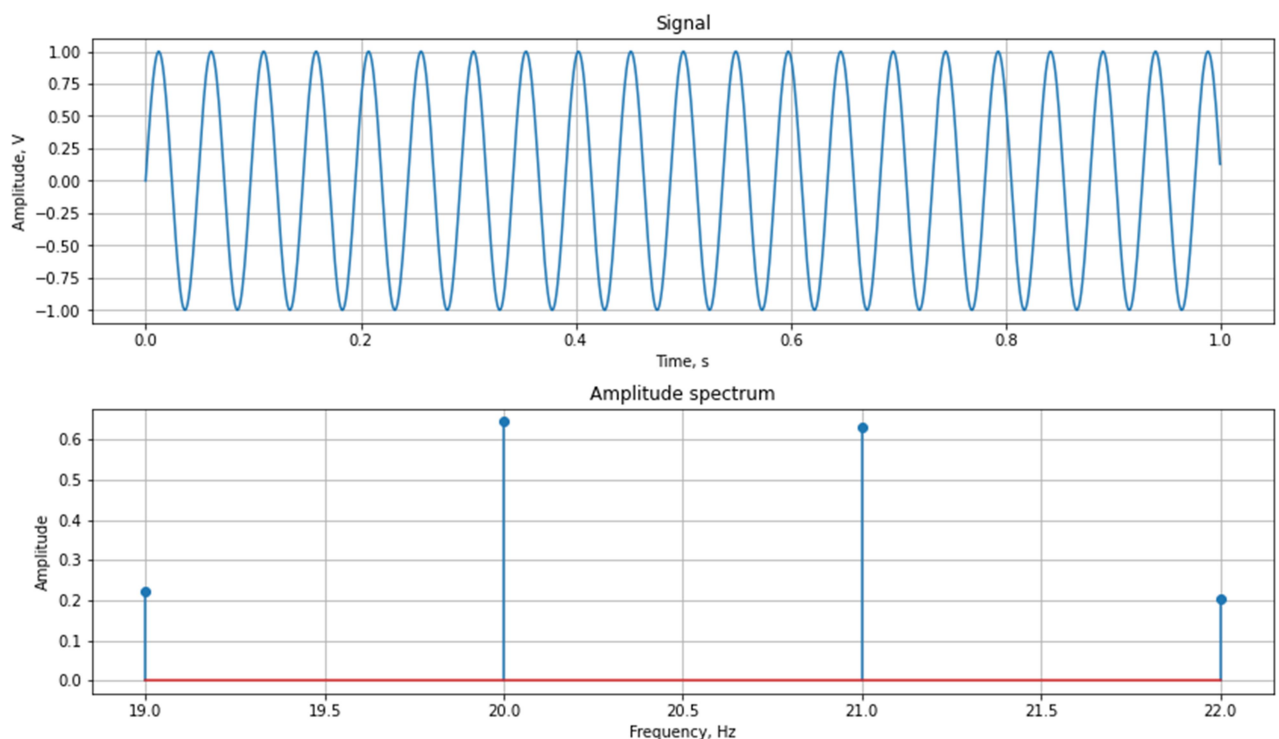


Fig.4.24. Sine wave with frequency of 20.5 Hz and its amplitude spectrum

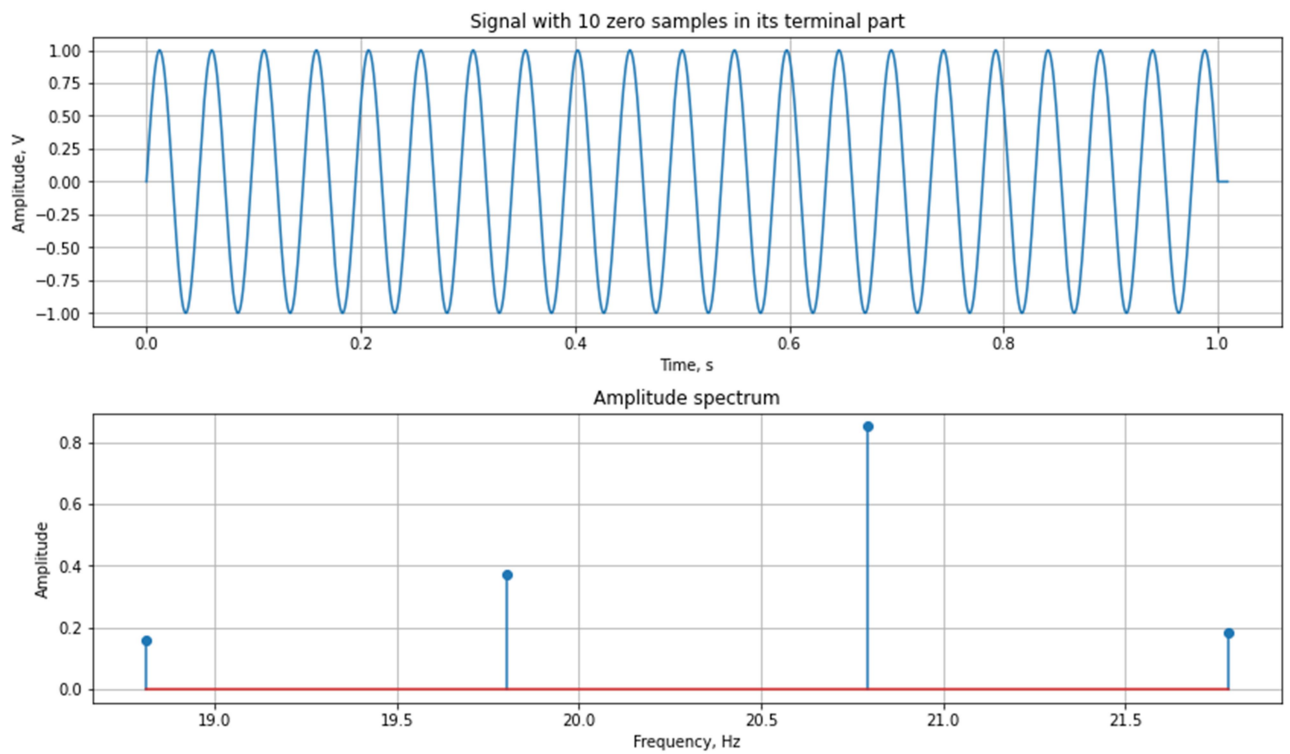


Fig.4.25. Sine wave with frequency of 20.5 Hz and added 10 zero samples and its amplitude spectrum

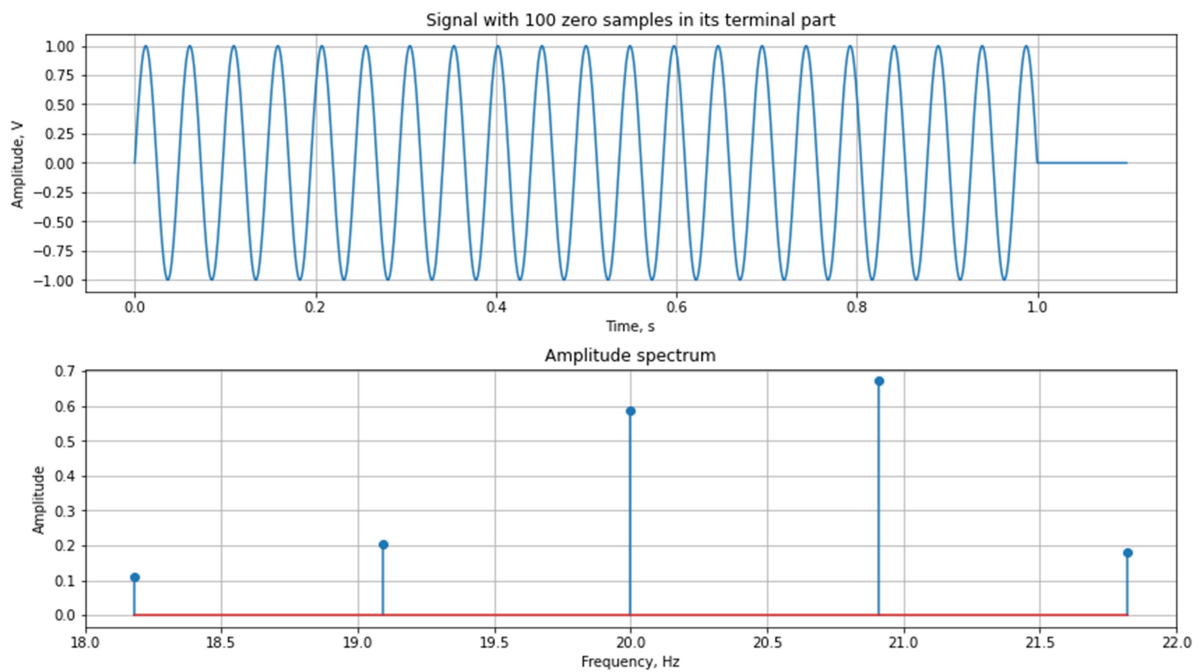


Fig.4.26. Sine wave with frequency of 20.5 Hz and added 100 zero samples and its amplitude spectrum

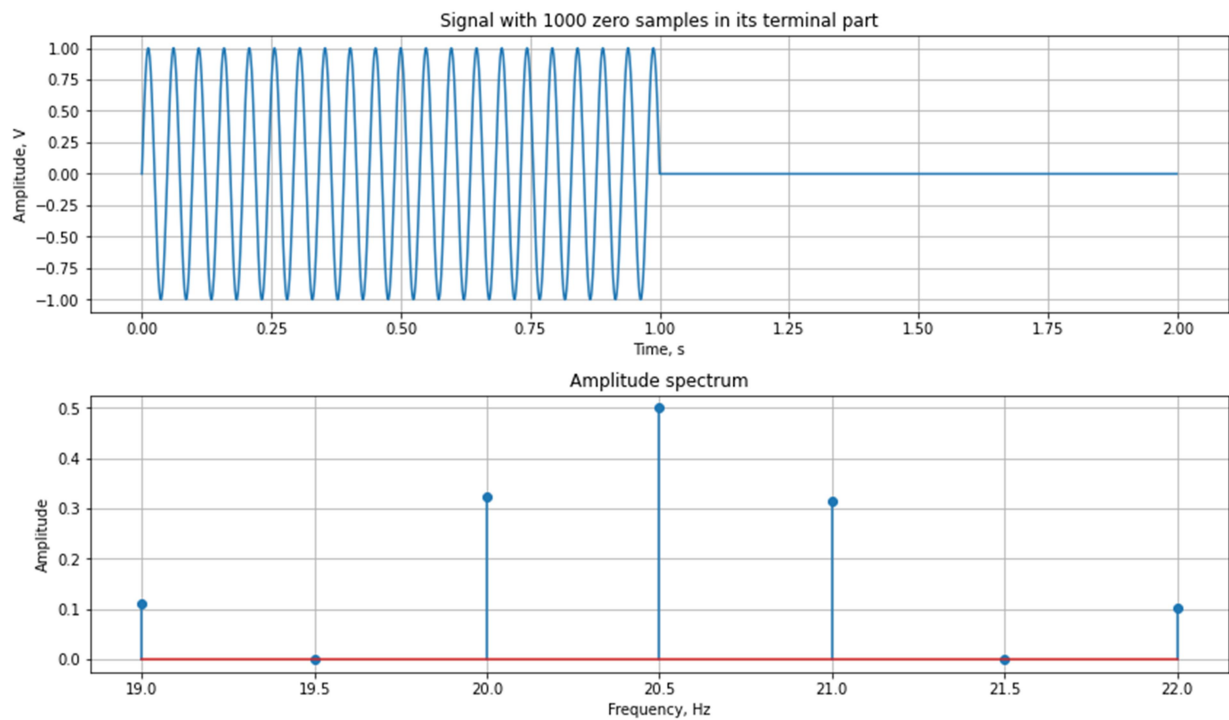


Fig.4.27. Sine wave with frequency of 20.5 Hz and added 1000 zero samples and its amplitude spectrum

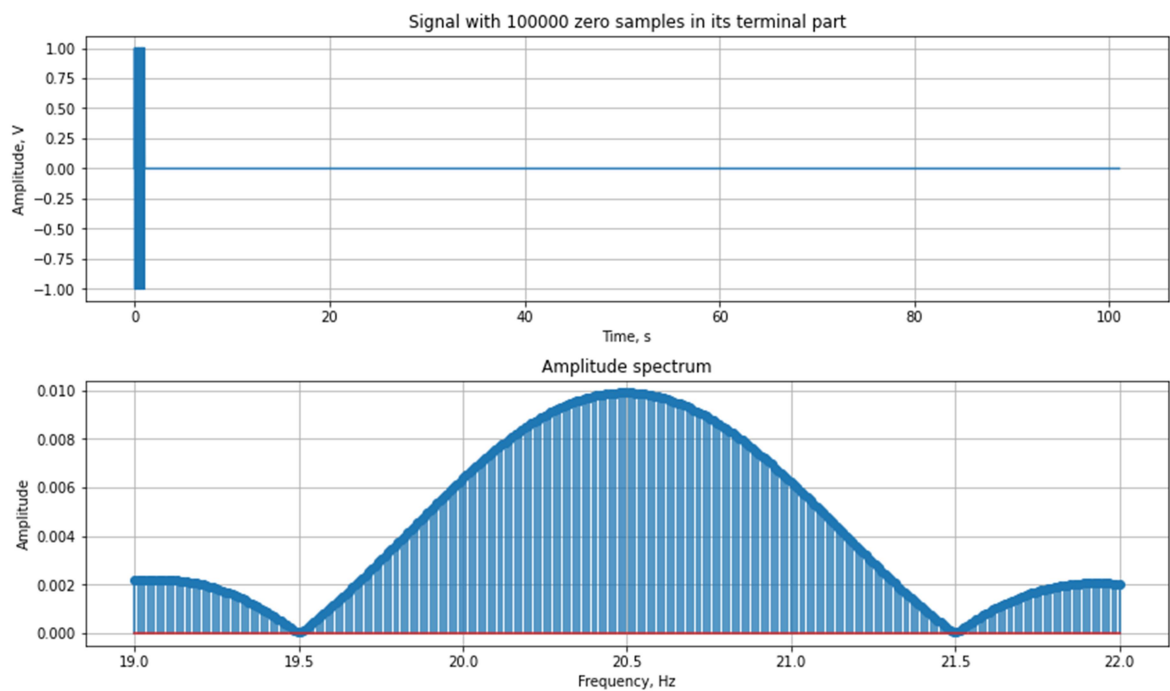


Fig.4.28. Sine wave with frequency of 20.5 Hz and added 1000 zero samples and its amplitude spectrum

9. For an arbitrary signal, perform a direct and then an inverse Fourier transform, compare the initial signal and the recovered signal (Fig.4.29). Find the root mean square recovery error (**std** function). Draw conclusions.

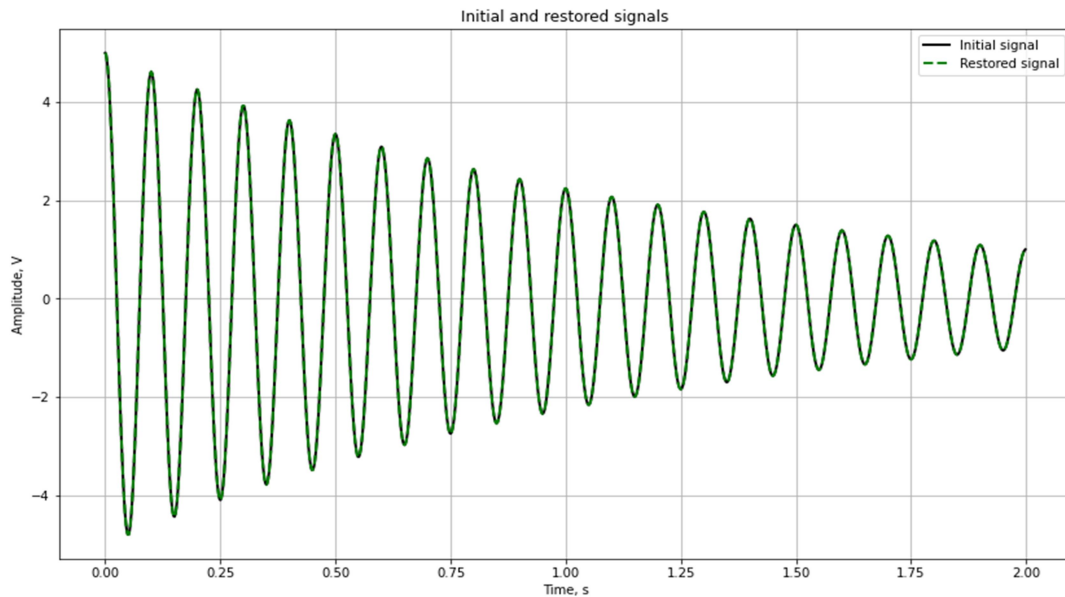


Fig.4.29. Initial signal and the recovered signal after direct and inverse Fourier transform

10. Construct a function for calculating and plotting the spectral power density of the signal loaded from the file (Fig.4.30). Transfer the file name with the signal and other necessary data to the function.

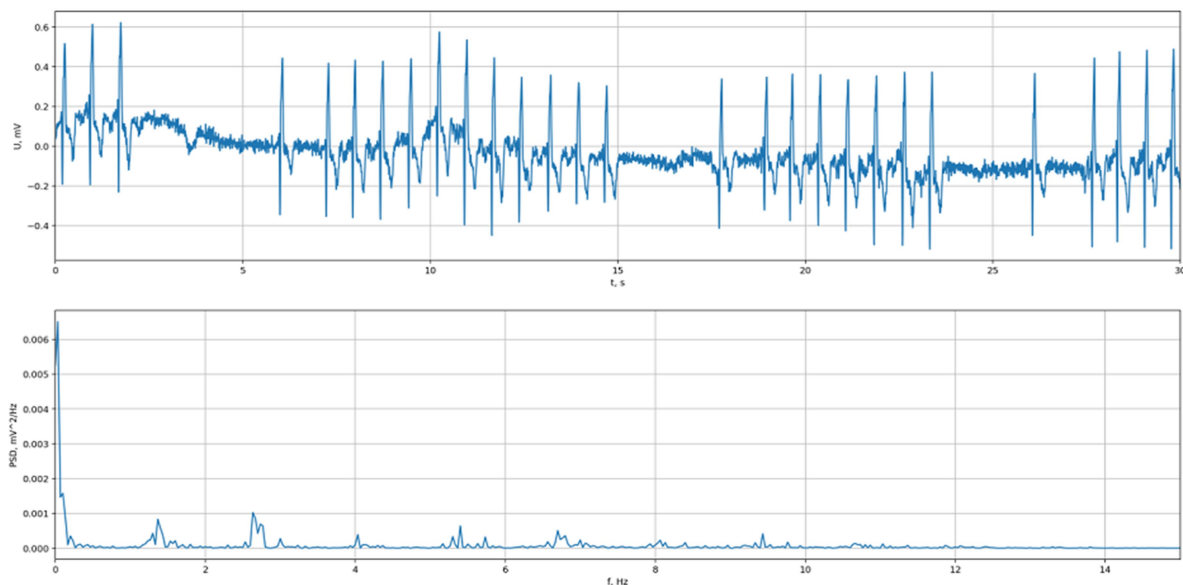


Fig.4.30. Spectral power density of ECG signal

11. Construct a function that allows you to calculate the amplitude and phase spectrum of a fragment of an arbitrary signal (Fig.4.31).

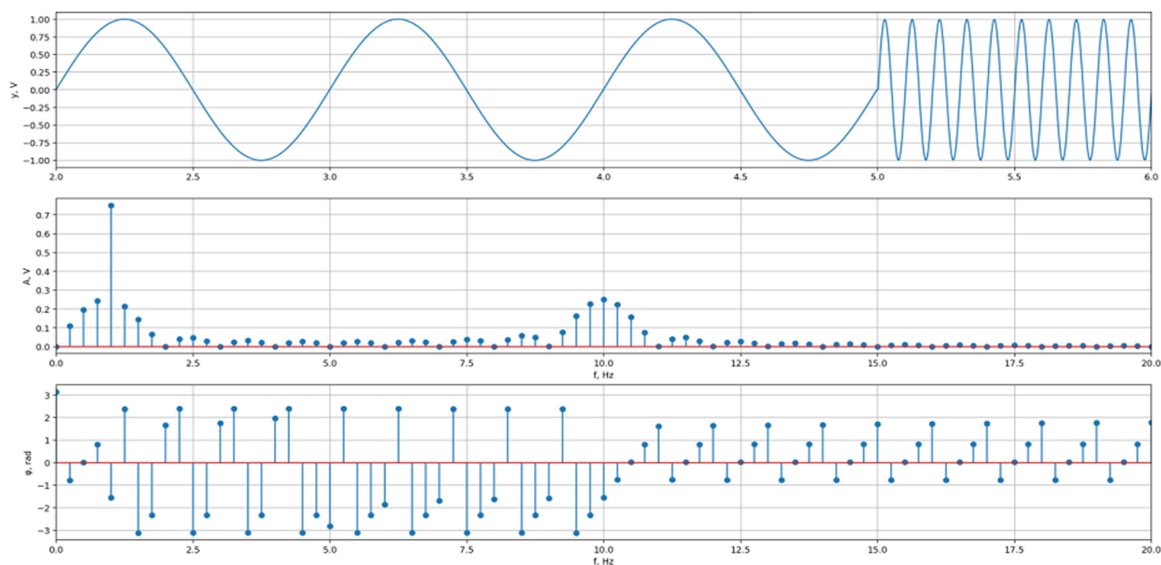


Fig.4.31. Amplitude and phase spectrum of a fragment of an arbitrary signal

12 \*. Plot a graph of the amplitude spectrum of a rectangular pulse with a duration of 0.1 s for sampling frequencies of 128 Hz, 256 Hz, 44100 Hz. Compare the results, draw conclusions.

13 \*. Generate a vector of time samples from -20 s to 20 s for a sampling frequency of 512 Hz. Generate the signals of the Kotelnikov functions  $\frac{\sin(10x)}{10x}$ ,  $\frac{\sin(x)}{x}$ ,  $\frac{\sin(0.1x)}{0.1x}$  (**sinc** function). Construct with the help of plot function graphs of signals and their amplitude spectra, draw conclusions. Build graphs for such frequencies that the features of the spectrum can be seen.

14 \*. Construct amplitude and phase spectra for digitized signals from Laboratory work №2. Draw conclusions about the range of frequencies at which the spectral components are, about the shape of the spectra.

15 \*. Perform a Fourier transform of sound signals that are obtained with different sampling rates. Construct amplitude and phase spectra, power spectrum, draw conclusions.

16 \*. Get acquainted with the functionality of spectral analysis by means of matplotlib.

## **Control questions**

1. What is spectral analysis of signals and why is it performed?
2. Write the mathematical expression of the signal decomposition into the Fourier series and the expression of the discrete Fourier transform.
3. What is a harmonica and what is it characterized by?
4. What is the amplitude and phase spectrum of the signal, what information does it carry?
5. Why use window functions in spectral analysis of signals?

## LABORATORY WORK № 5

### TIME-SPECTRAL FOURIER ANALYSIS OF SIGNALS

**Purpose:** explore the relationship between temporal and spectral characteristics of signals; to acquire skills of using the basic means of spectral and spectral-temporal Fourier analysis of signals.

#### Tasks

1. Using the **windows** function, get acquainted with the time characteristics and spectra of all window functions. According to the options listed in table 1 (according to the list, subtract your number and take two rows from the table; after the last number start counting from the first), form windows of a given type and duration of 256 counts (**get\_window** function). Plot graphs of windows and their amplitude spectra (Fig.5.1, 5.2).

Table 1.

#	Window name
1	modified Bartlett-Hann
2	Bartlett
3	Blackman
4	Blackman-Harris
5	Bohman
6	rectangular
7	simple cosine
8	Poisson
9	flat top
10	Hamming
11	Hann
12	Parzen
13	triangular
14	Tukey



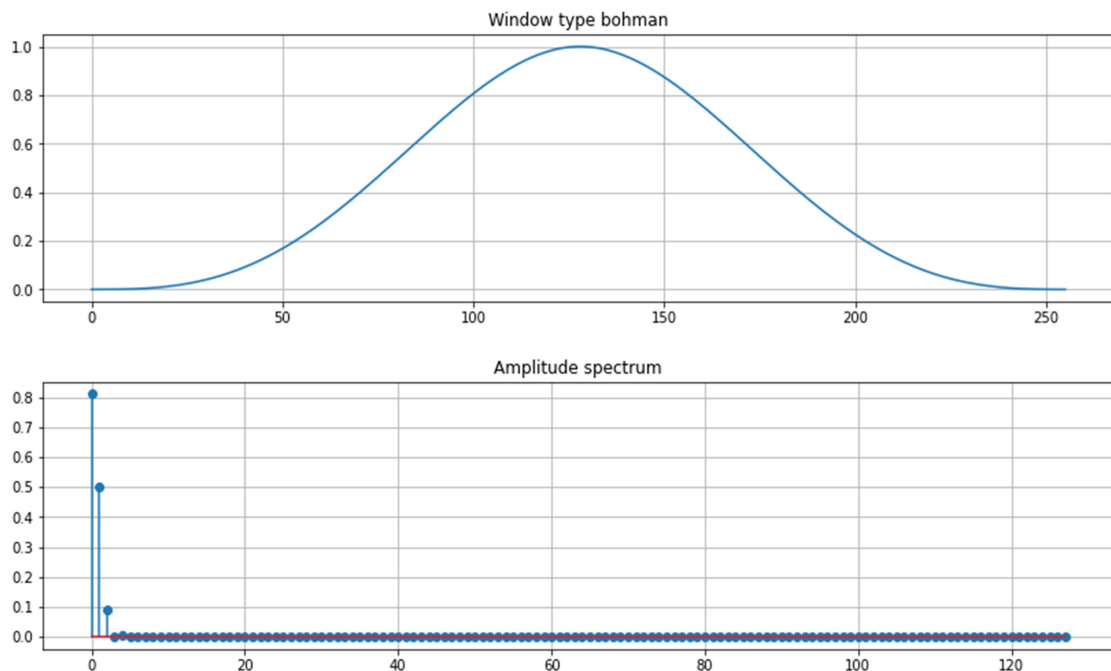


Fig.5.1. Bohman window function and its spectrum

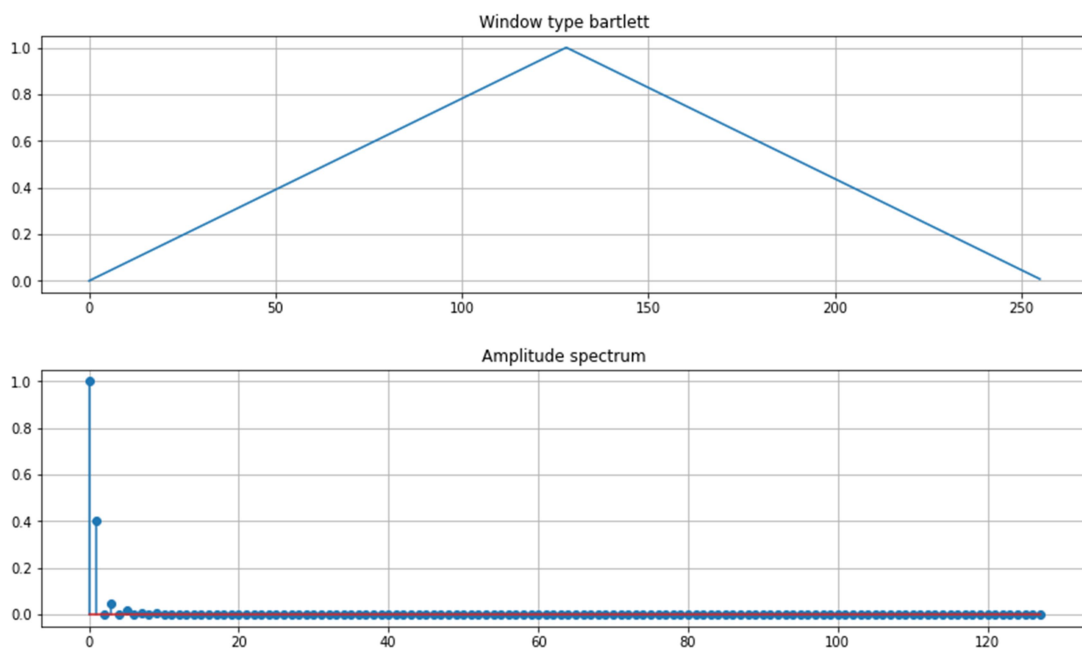


Fig.5.2. Barlett window function and its spectrum

2. Form a vector of time samples with a duration of 1 s for a sampling frequency of 128 Hz. Generate signals of the sine wave with a frequency of 2 and 2.5 Hz. Construct the amplitude spectrum of signals without using the window function and using the first window according to your variant. Select the duration of the window equal to the duration of the signals. Compare with the results of Task 1 of

the laboratory work on spectral analysis. Draw conclusions about the distortion of the spectra and the feasibility of using window decoration (Fig.5.3-5.4).

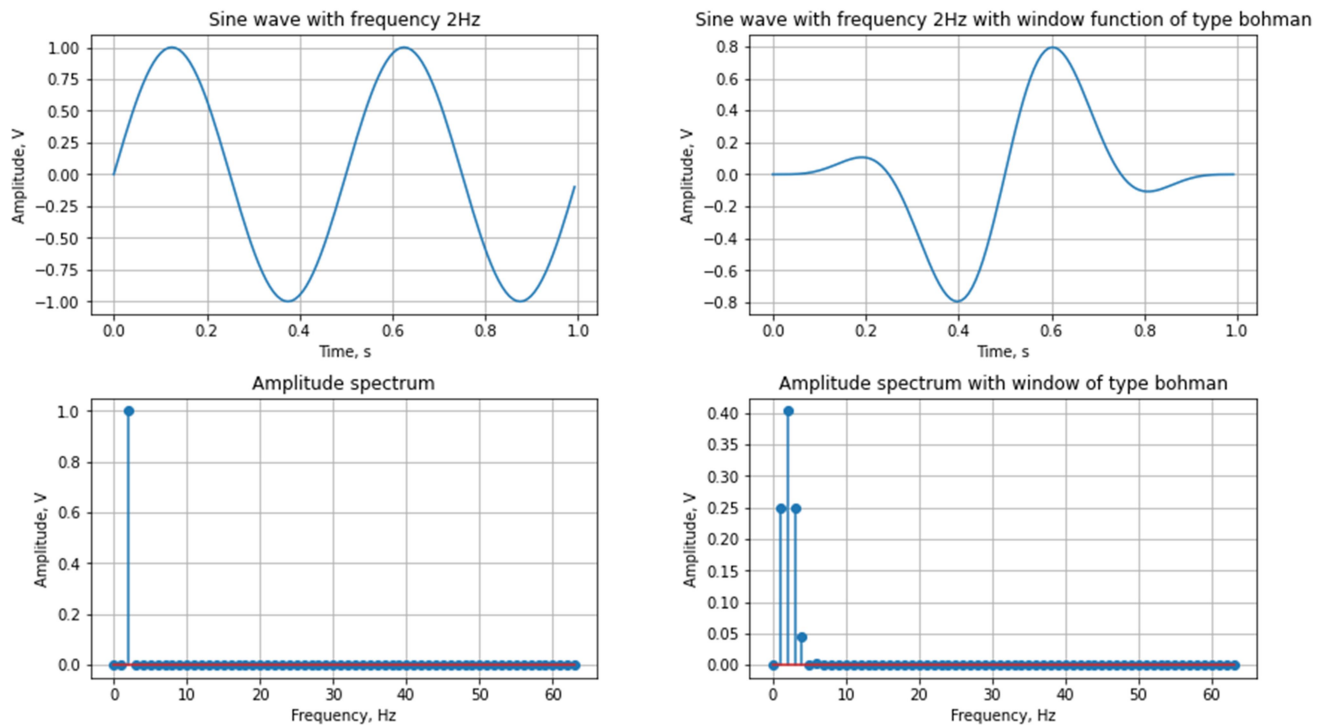


Fig.5.3. Signal of the sine wave with a frequency of 2 Hz and its amplitude spectrum (without and with using Bohman window function)

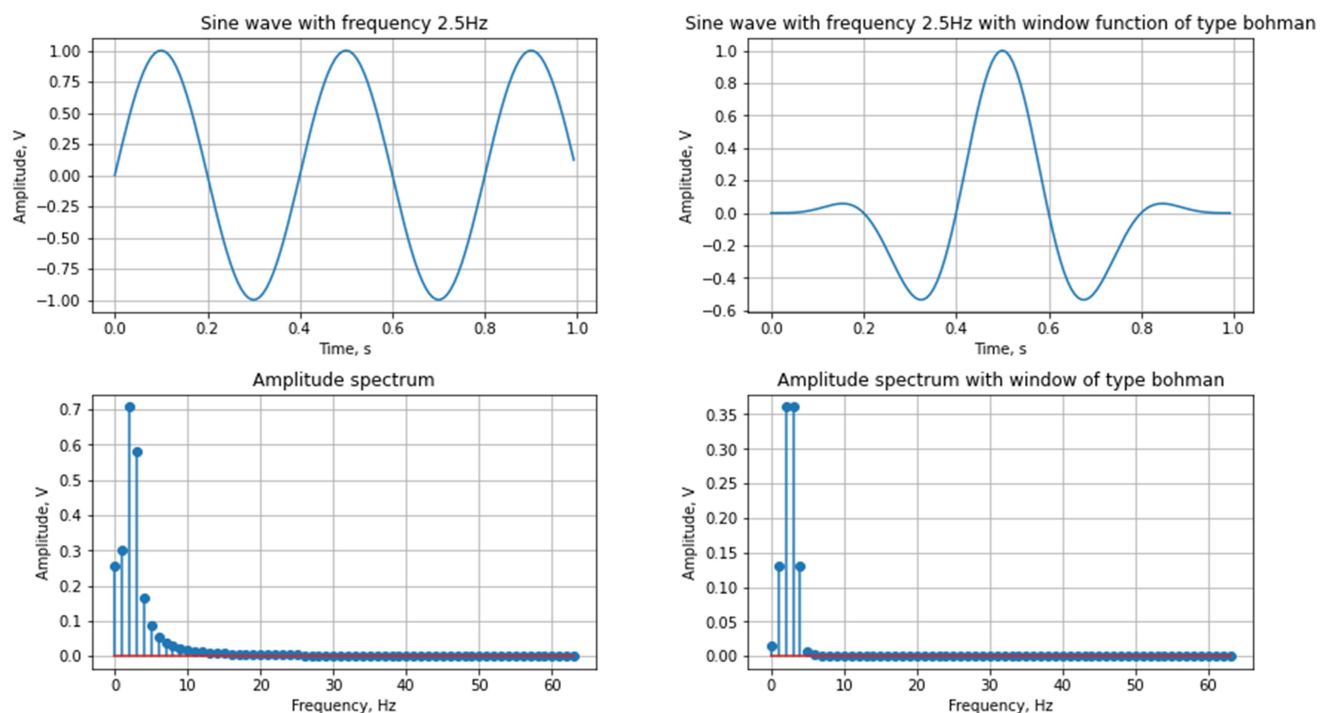


Fig.5.4. Signal of the sine wave with a frequency of 2.5 Hz and its amplitude spectrum (without and with using Bohman window function)

3. Form a vector of time samples with a duration of 15 s for a sampling frequency of 128 Hz. Generate signals:

- 3.1. sinusoid with a frequency of 40 Hz;
- 3.2. rectangular pulse of width 1 s at a time of 10 s;
- 3.3. random signal;
- 3.4. the sum of the signals 3.1 - 3.3.

Construct **spectrograms** of signals using the first window according to the option with duration of 0.2 s without the use of window overlap (Fig.5.5-5.8). Draw conclusions about the appearance of spectrograms and the correspondence of time, spectral and spectral-time properties of signals.

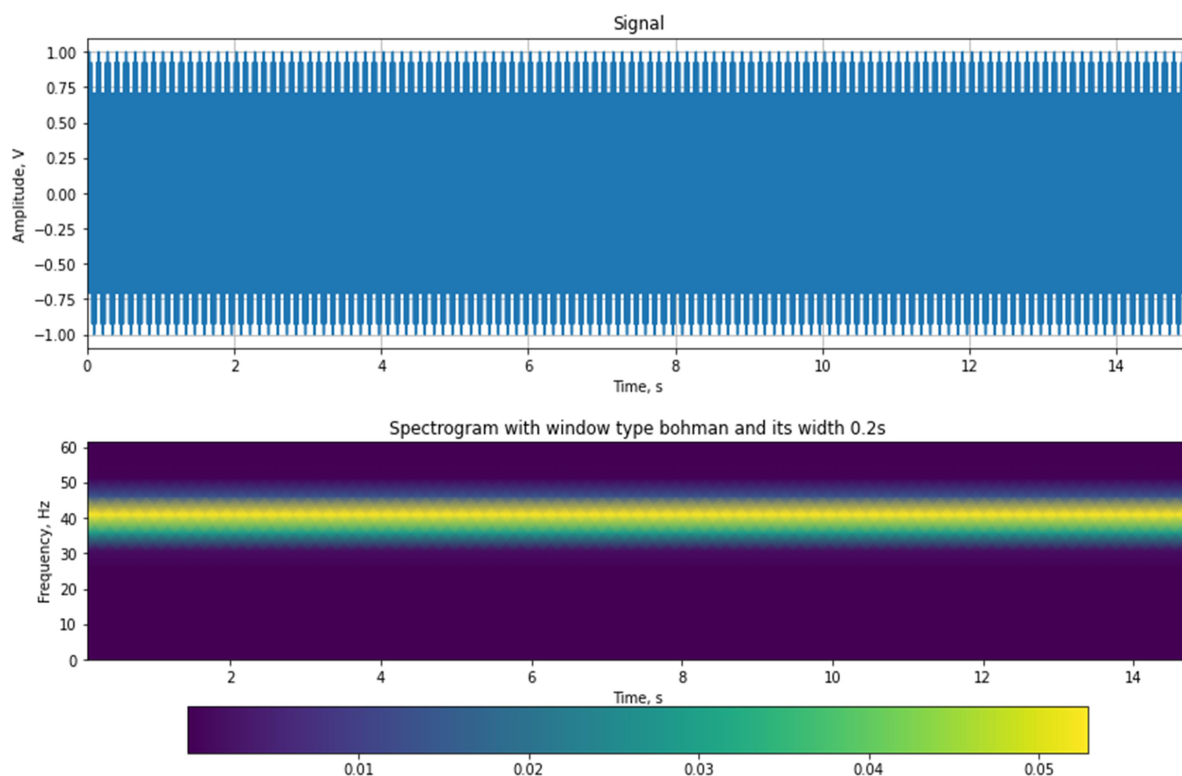


Fig.5.5. Sinusoid with a frequency of 40 Hz and its spectrogram

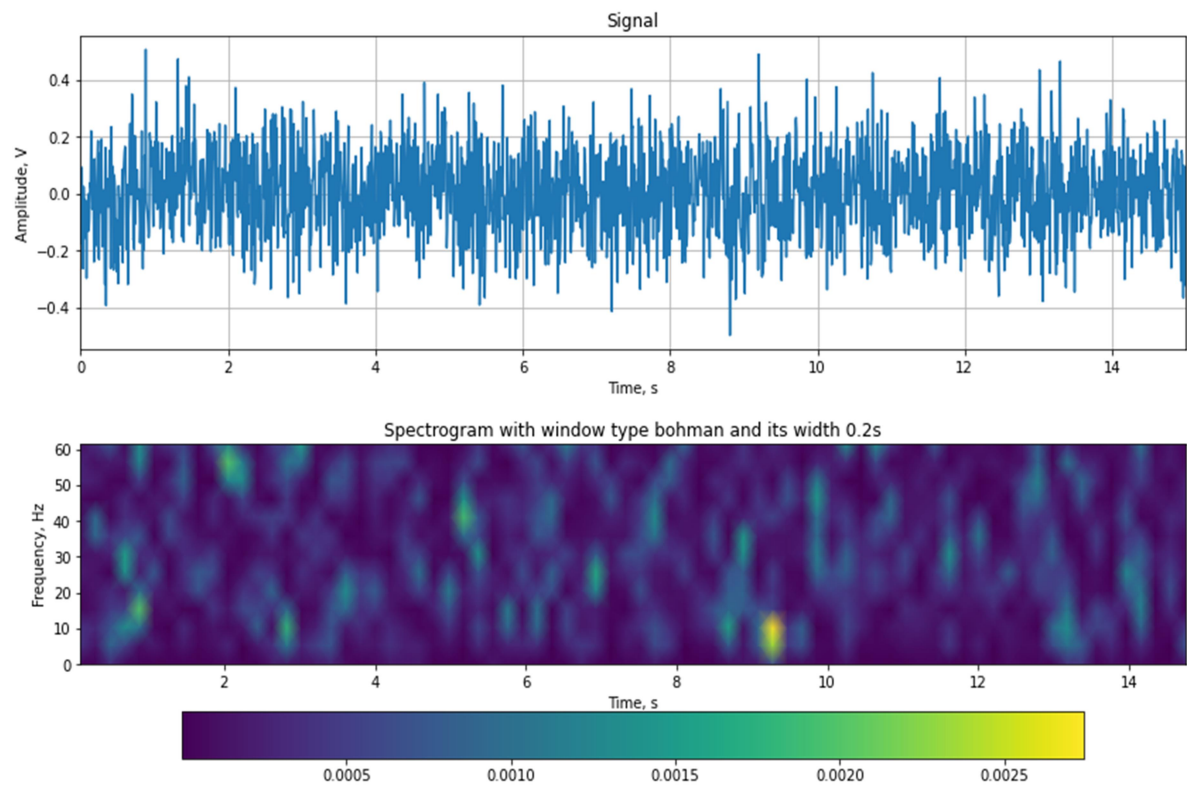


Fig.5.6. Random signal and its spectrogram

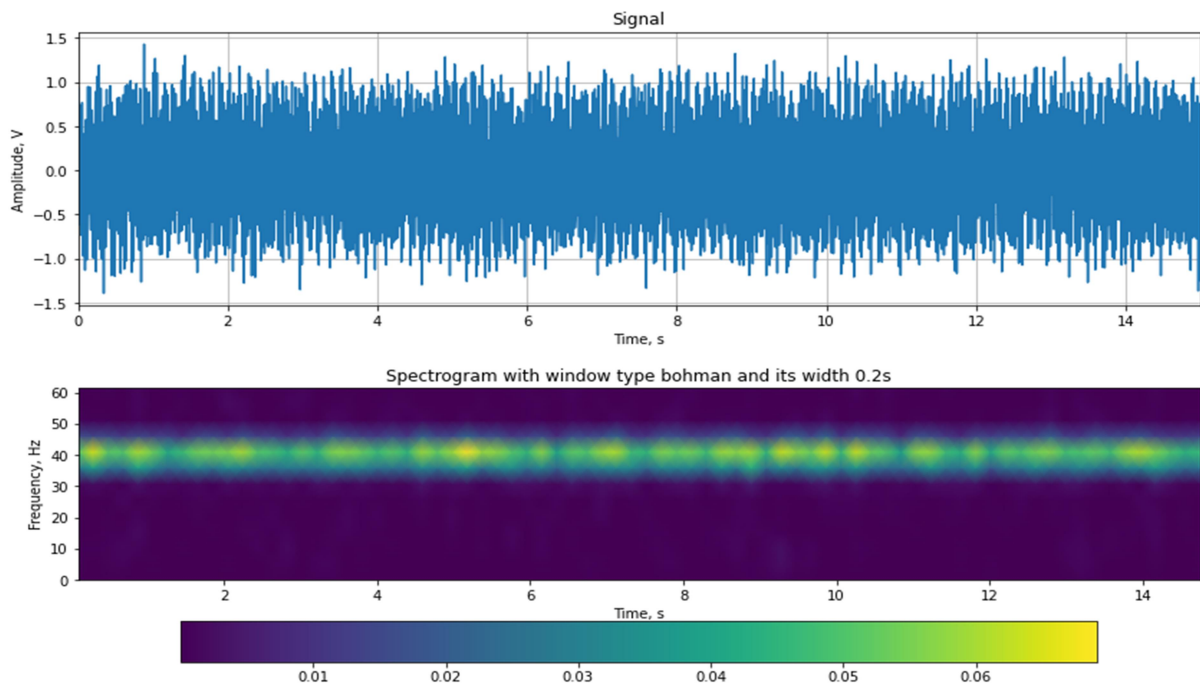


Fig.5.7. Sinusoid with a frequency of 40 Hz with noise component and its spectrogram

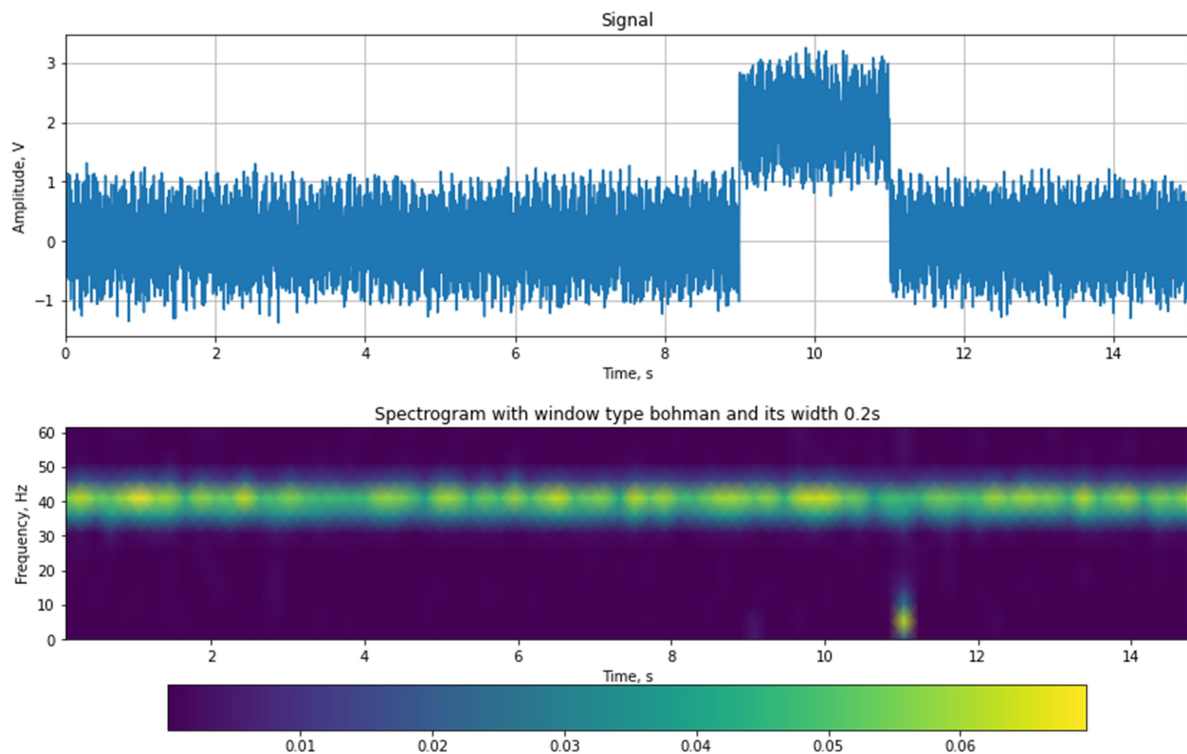


Fig.5.8. Sum of the rectangular pulse, sinusoid with a frequency of 40 Hz and noise component and spectrogram of the considered signal

4. For signals from Task 2 of laboratory work on spectral analysis, construct spectrograms (Fig.5.9-5.17):

- with a window lasting 0.1 s and 2 s (without overlapping);
- with a window lasting 1 s with an overlap of 50%.

Use the **colorbar** function to visualize spectrogram values. Draw conclusions about the display of time properties of signals in the spectrogram. Compare the informativeness of spectrograms and Fourier spectra. Construct three-dimensional graphs of two different spectrograms from the obtained ones.

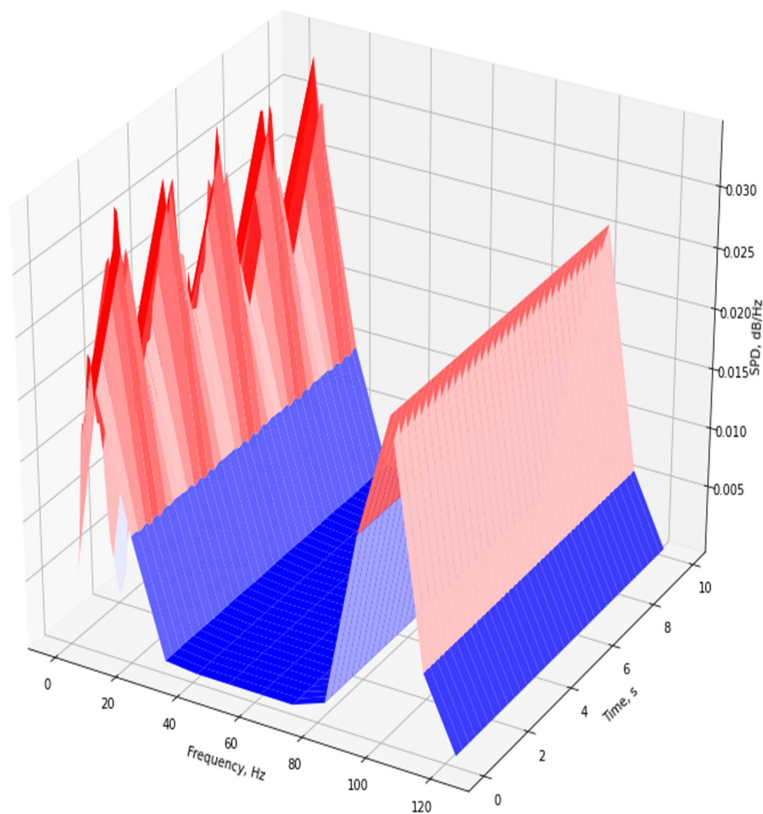
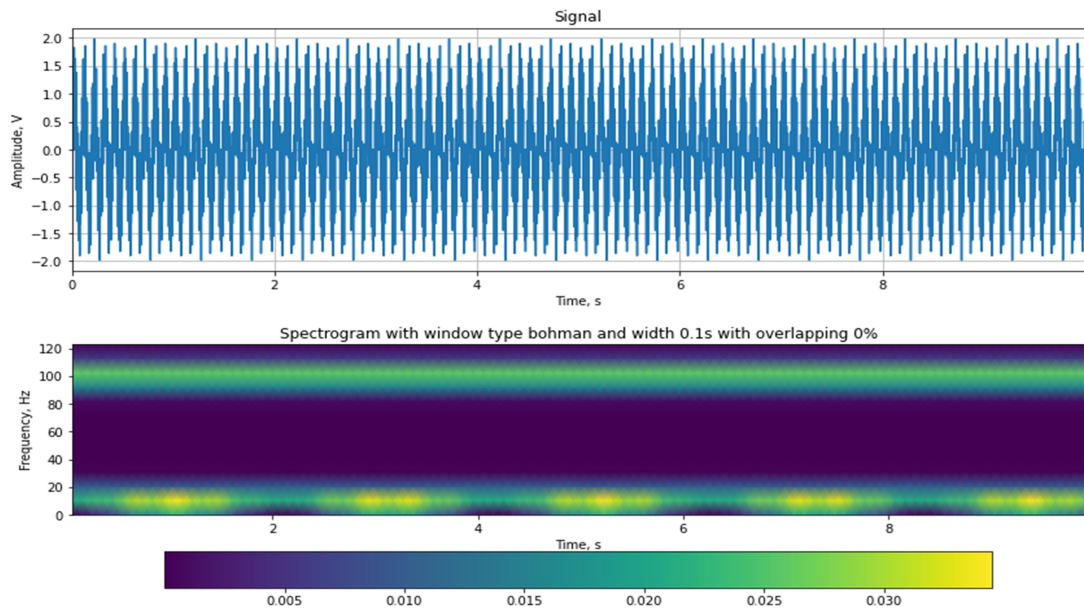


Fig.5.9. Sum of two sine waves of 10 and 100 Hz frequency and its spectrogram with a window lasting 0.1 s (without overlapping)



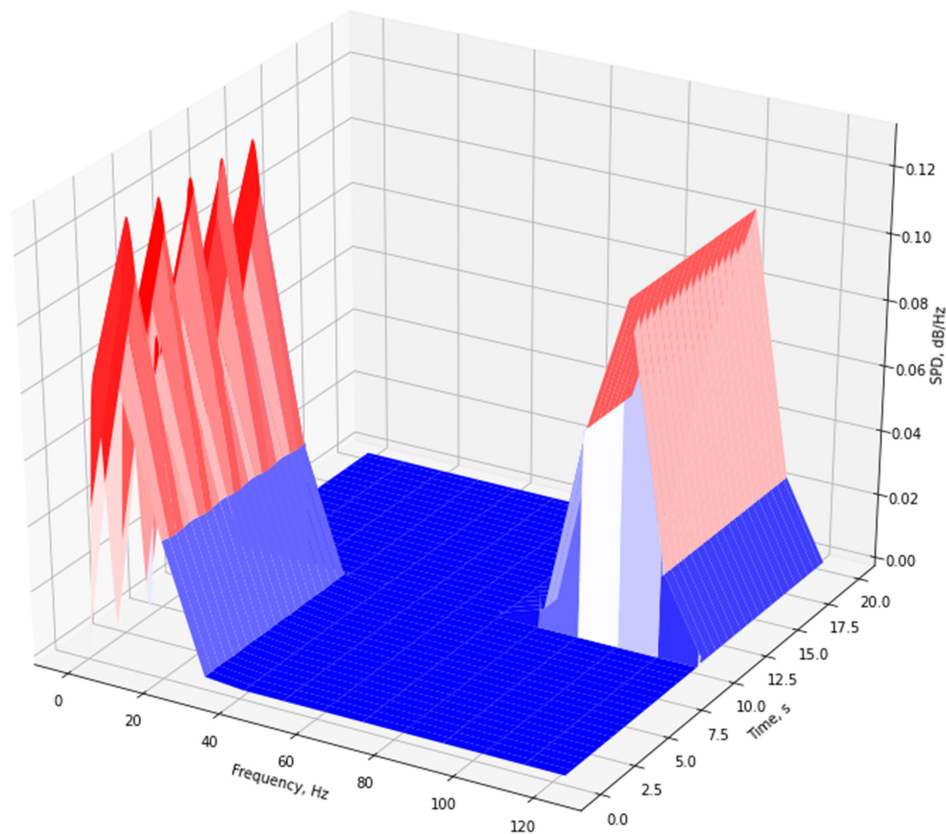
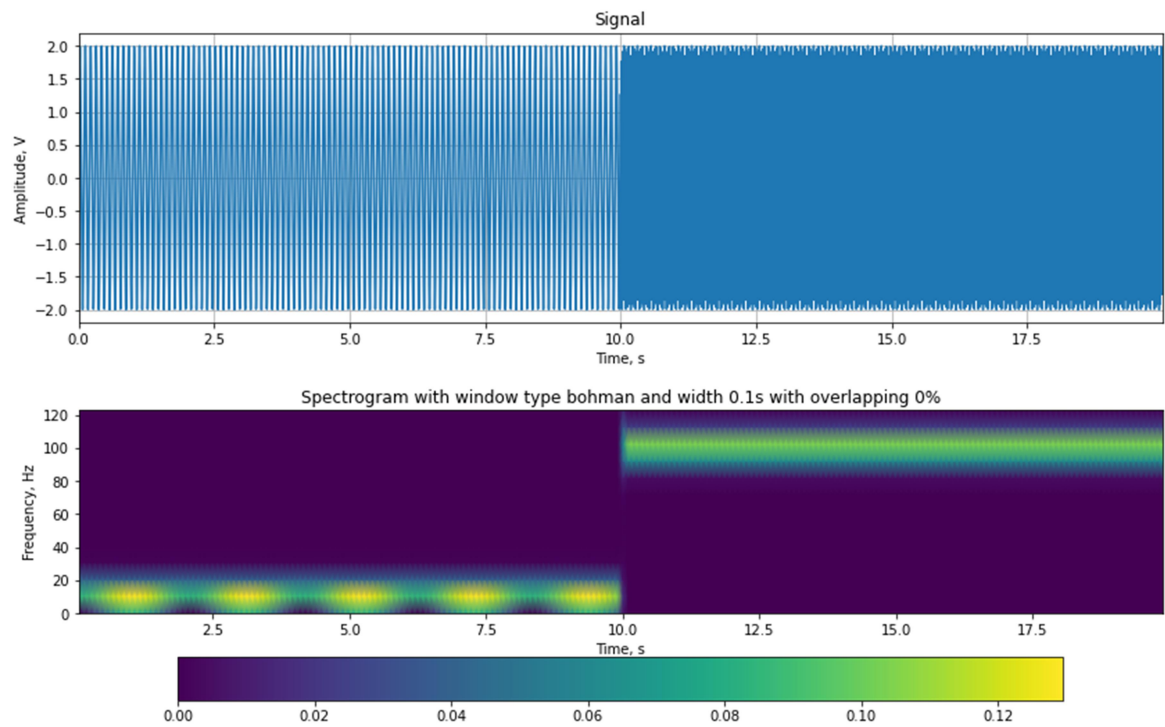


Fig.5.10. Concatenation of two sine waves of 10 and 100 Hz frequency and its spectrogram with a window lasting 0.1 s (without overlapping)

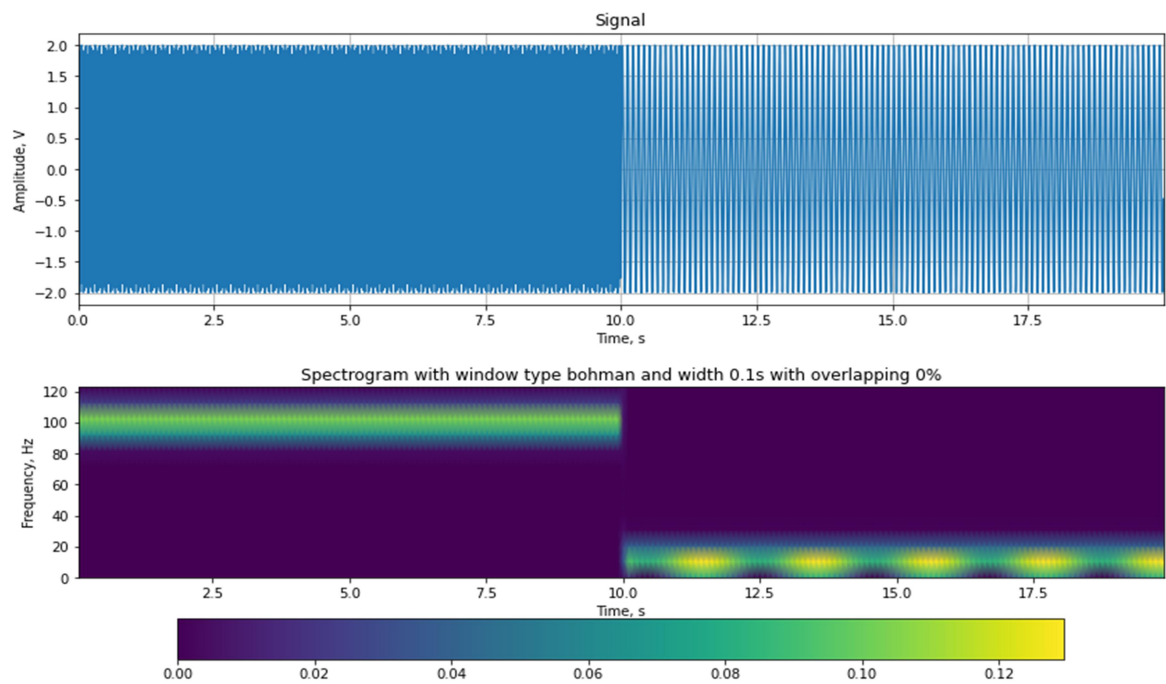


Fig.5.11. Concatenation of two sine waves of 100 and 10 Hz frequency and its spectrogram with a window lasting 0.1 s (without overlapping)

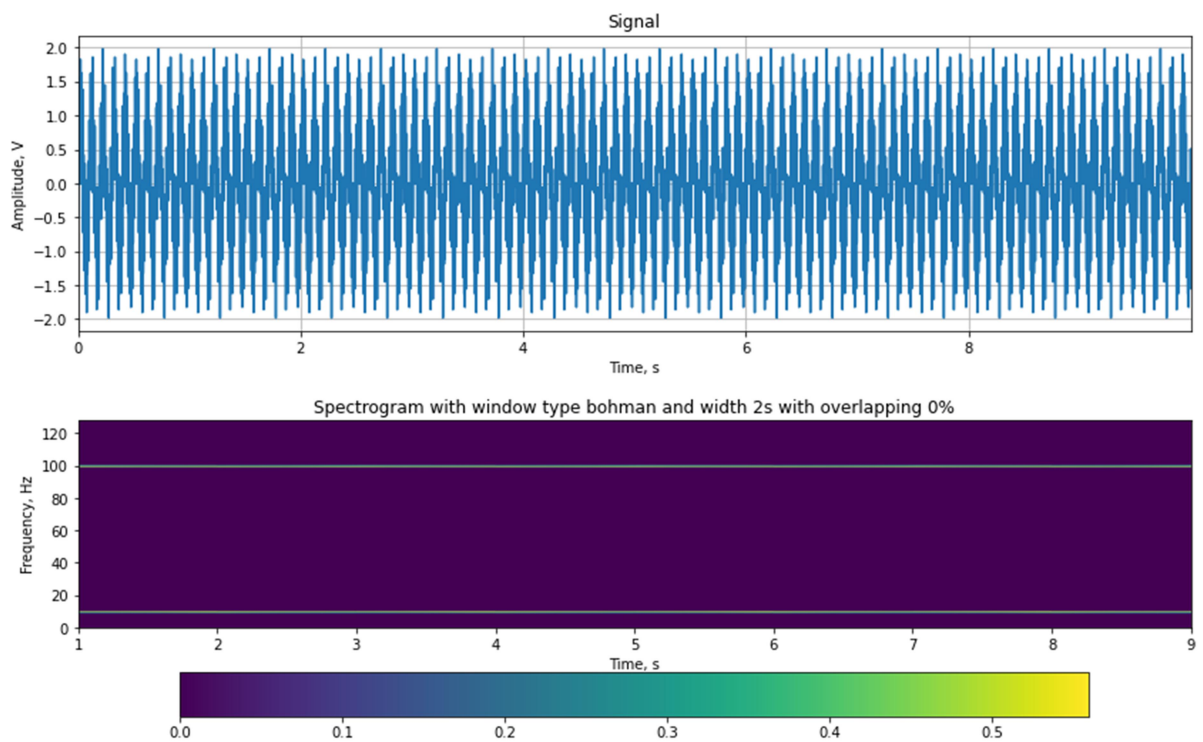


Fig.5.12. Sum of two sine waves of 10 and 100 Hz frequency and its spectrogram with a window lasting 2s (without overlapping)



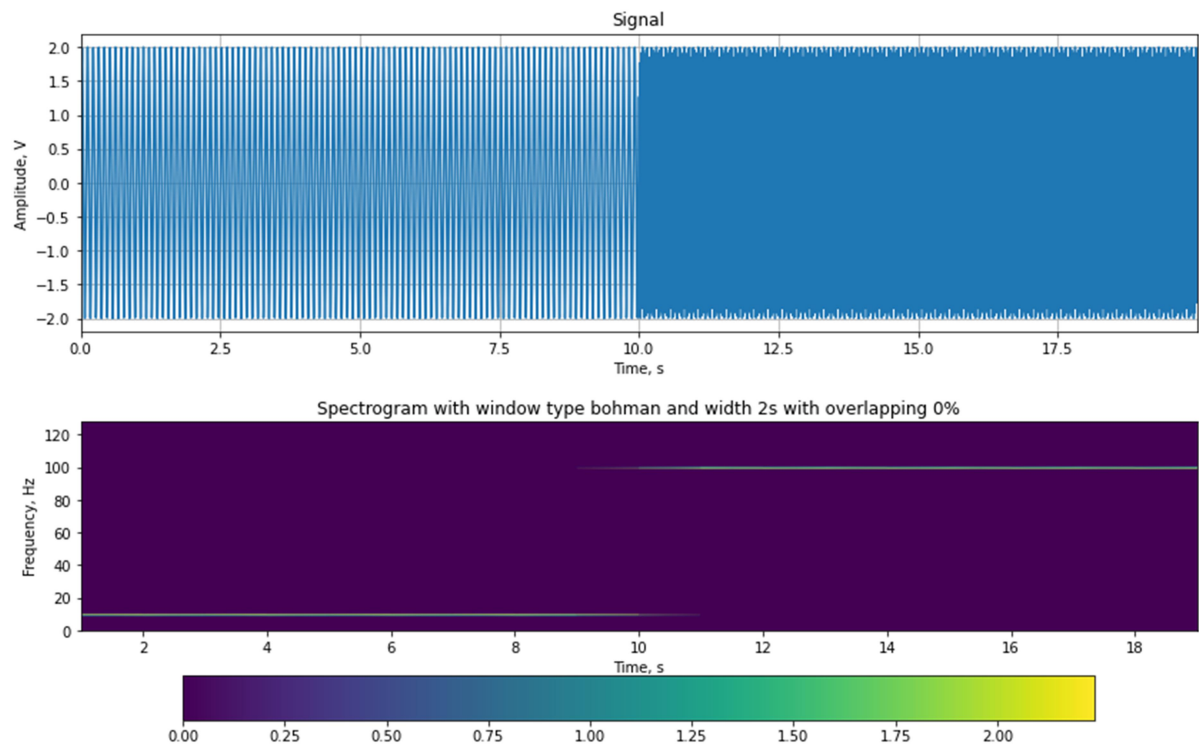


Fig.5.13. Concatenation of two sine waves of 10 and 100 Hz frequency and its spectrogram with a window lasting 2 s (without overlapping)

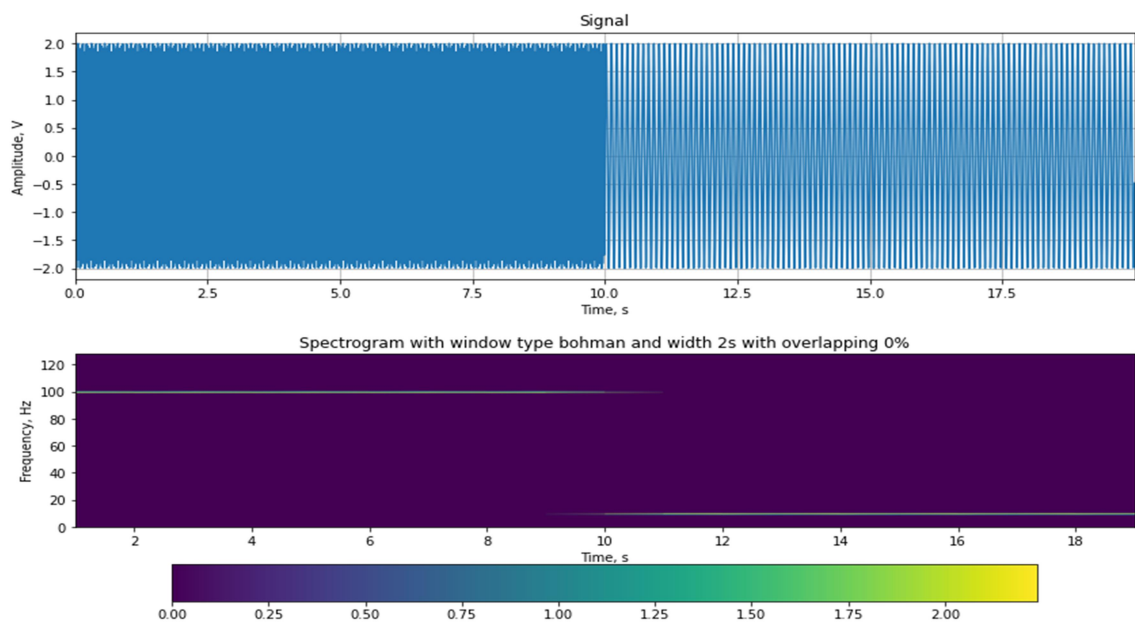


Fig.5.14. Concatenation of two sine waves of 100 and 10 Hz frequency and its spectrogram with a window lasting 2 s (without overlapping)

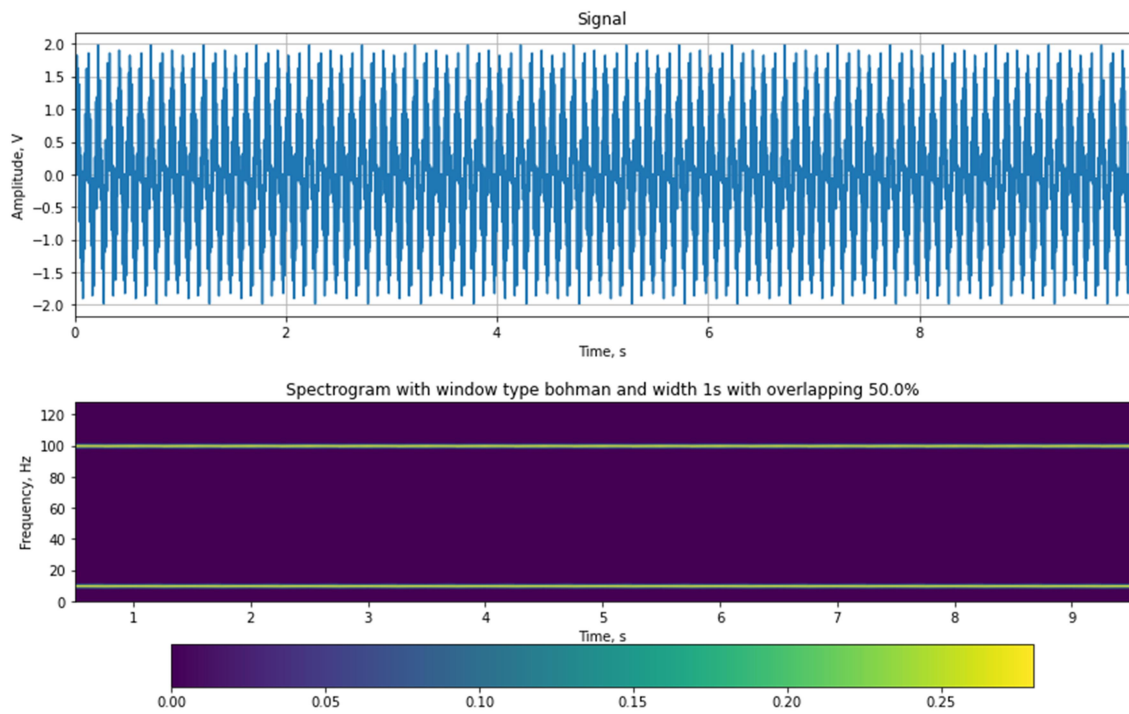
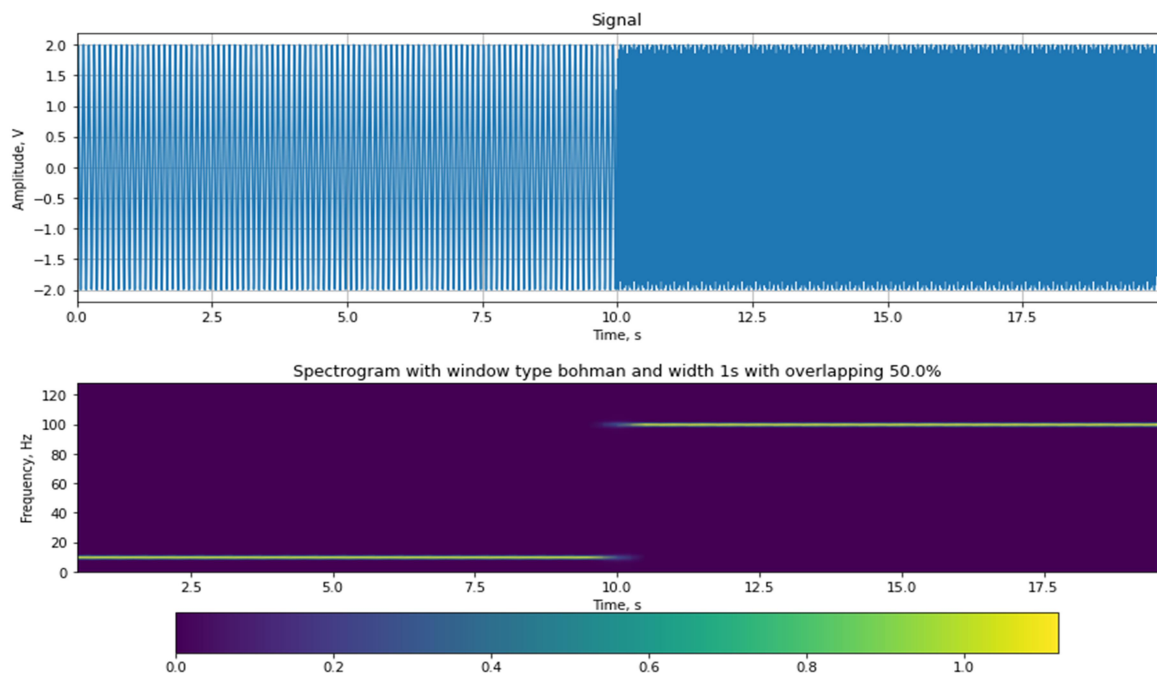


Fig.5.15. Sum of two sine waves of 10 and 100 Hz frequency and its spectrogram with a window lasting 1s (with overlapping 50%)



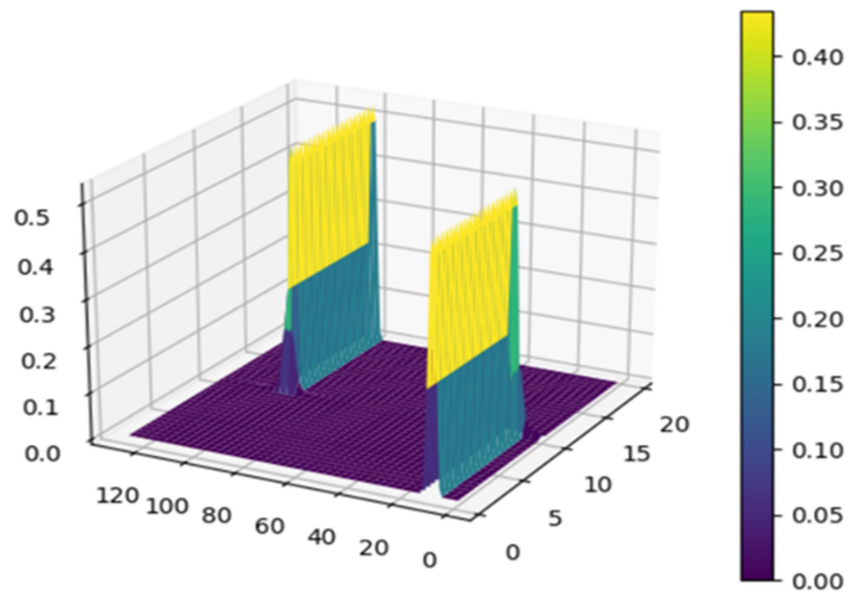
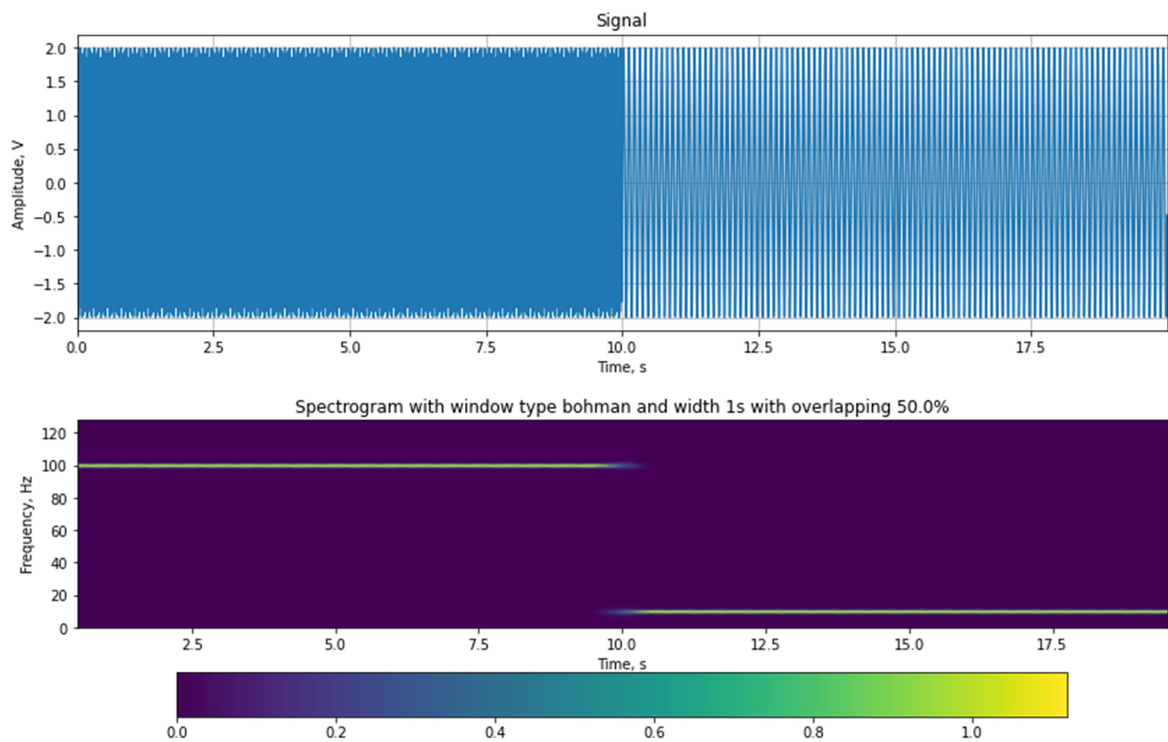


Fig.5.16. Concatenation of two sine waves of 10 and 100 Hz frequency and its spectrogram with a window lasting 1 s (with overlapping 50%)



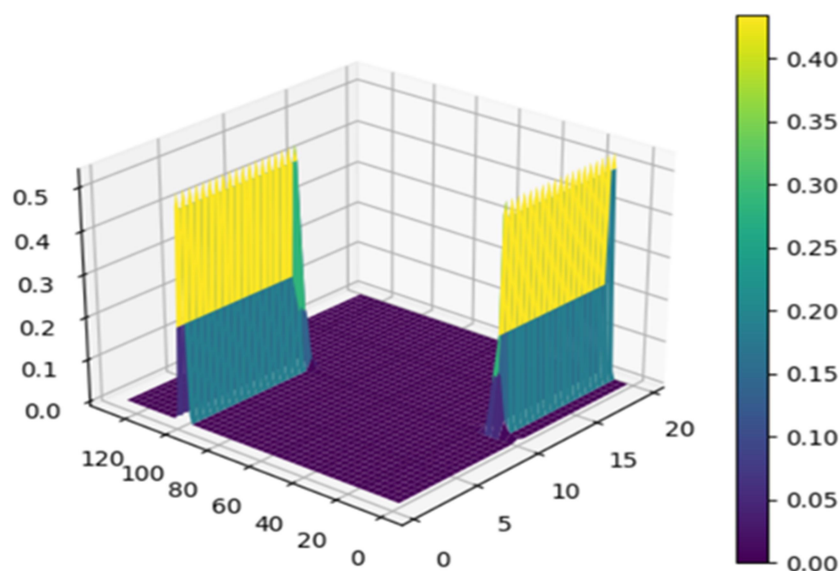


Fig.5.17. Concatenation of two sine waves of 100 and 10 Hz frequency and its spectrogram with a window lasting 1 s (with overlapping 50%)

5. For the signals according to Task 3 of the laboratory work on spectral analysis, construct spectrograms of the signal using a window, the duration and overlap of which are selected optimally to determine the moment of gap in the signal (Fig.5.18, 5.19). Justify your choice and draw conclusions.

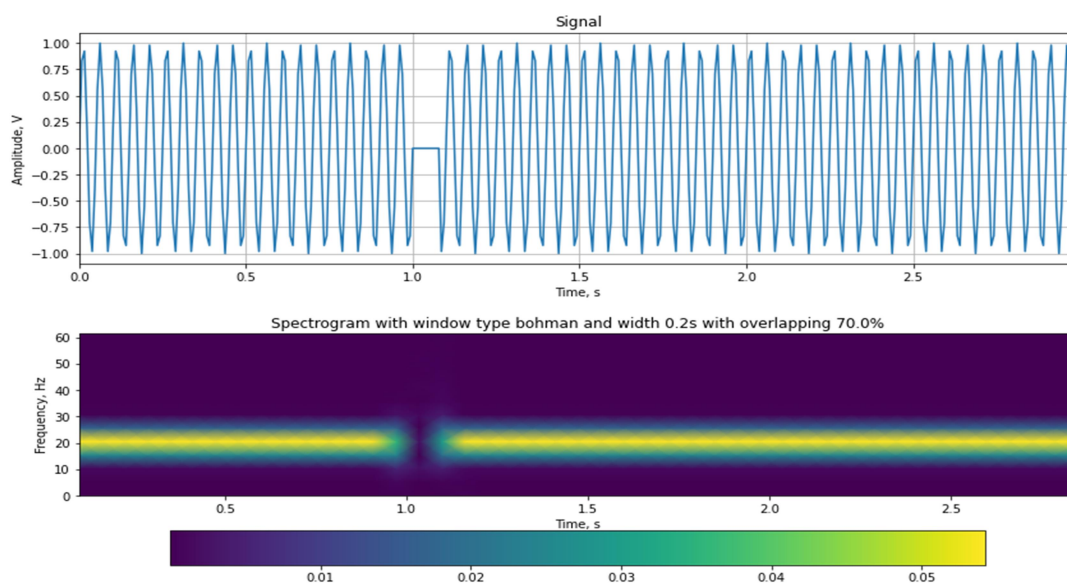


Fig.5.18. Signal of the sine wave with gap in the signal at time moment of 1.05 s and its spectrogram

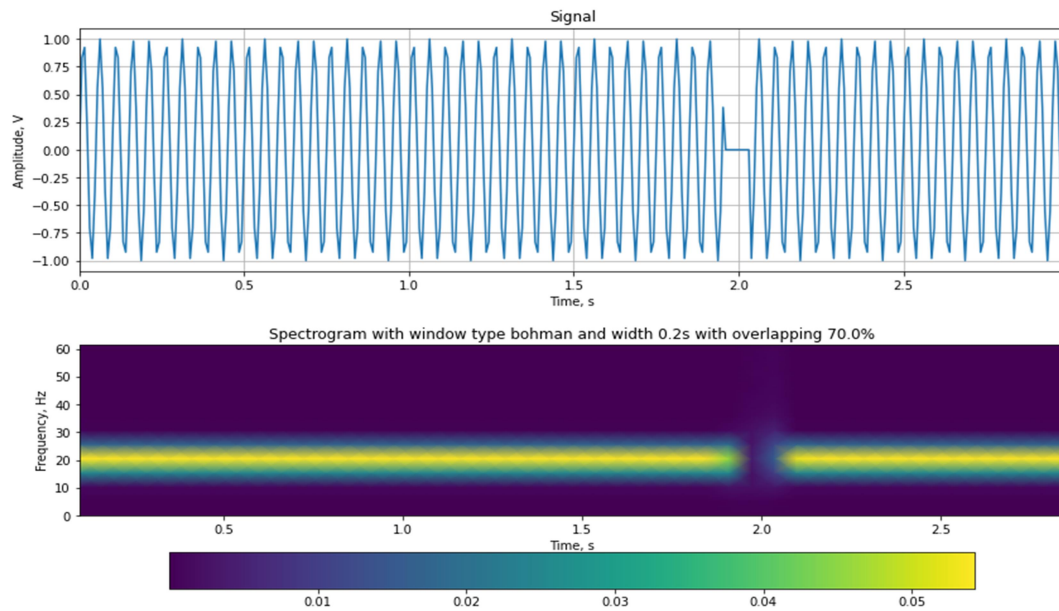
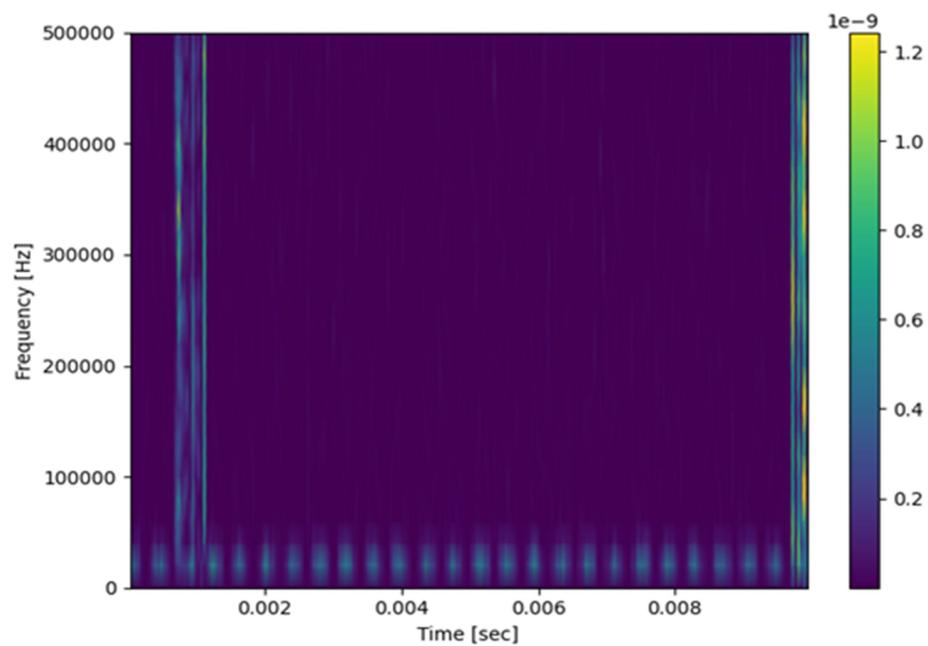


Fig.5.19. Signal of the sine wave with gap in the signal at time moment of 2 s and its spectrogram

6. Construct a function that plots the change in time of the average spectral power density in a given frequency range and time range for a given signal (Fig.5.20). As parameters of the function, pass the name of the signal, time  $t_1$ ,  $t_2$  and frequency  $f_1$ ,  $f_2$ , as well as other necessary parameters.



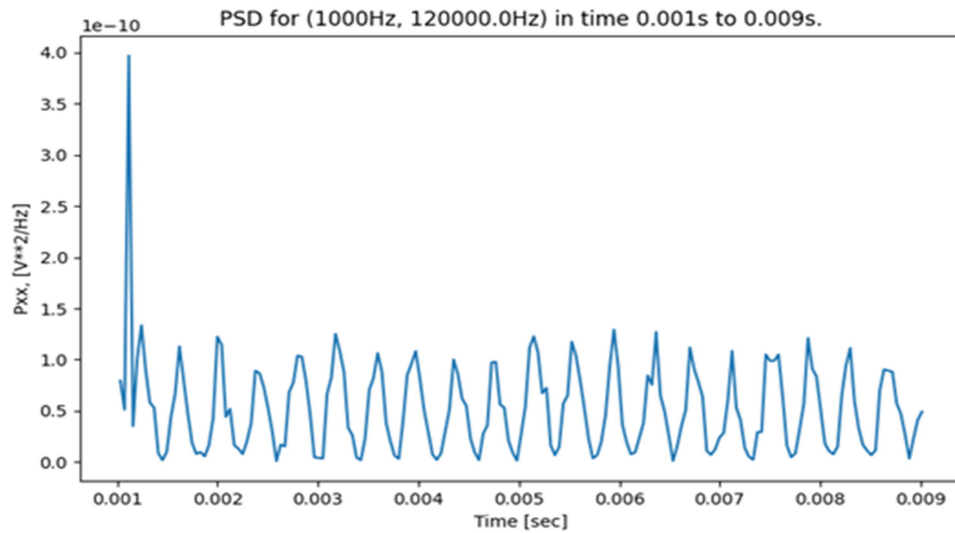


Fig. 5.20. Change in time of the average spectral power density in a given frequency range and time range for a given signal

7 \*. Build spectrograms of digitized biomedical signals for one of the windows (Fig.5.21, 5.22). Use windows of different duration and different overlap, choose one of the combinations, justify your choice. Draw conclusions about the nature of changes in the spectrum of signals over time.

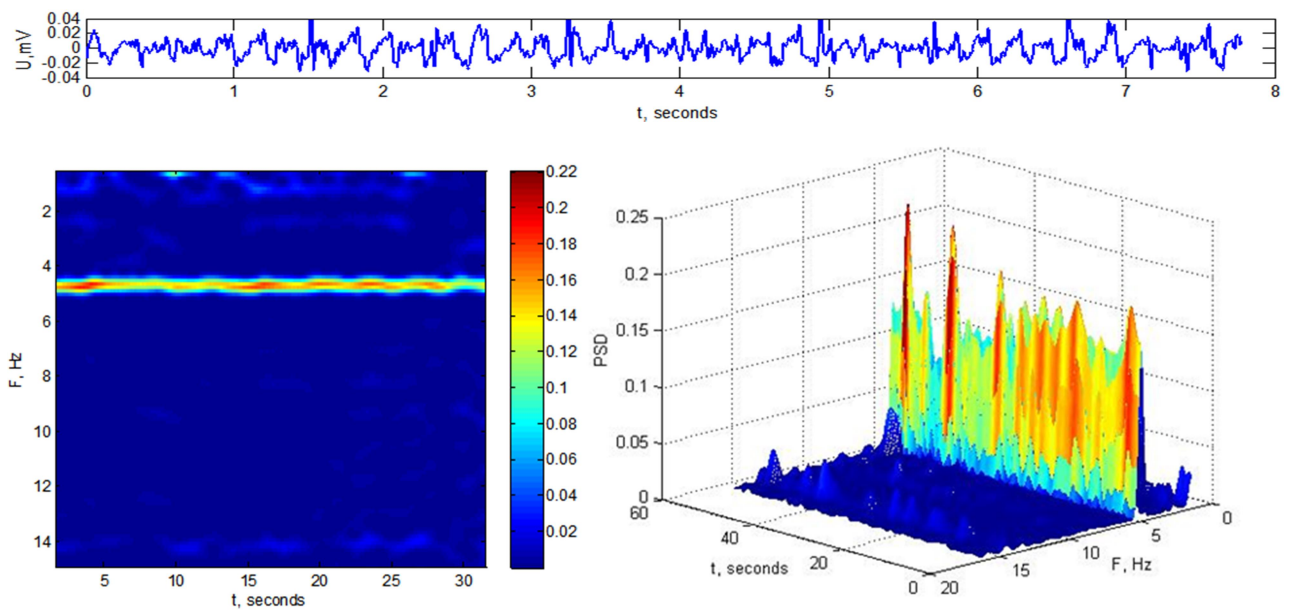


Fig. 5.21. Spectrogram of atrial electrical activity signal



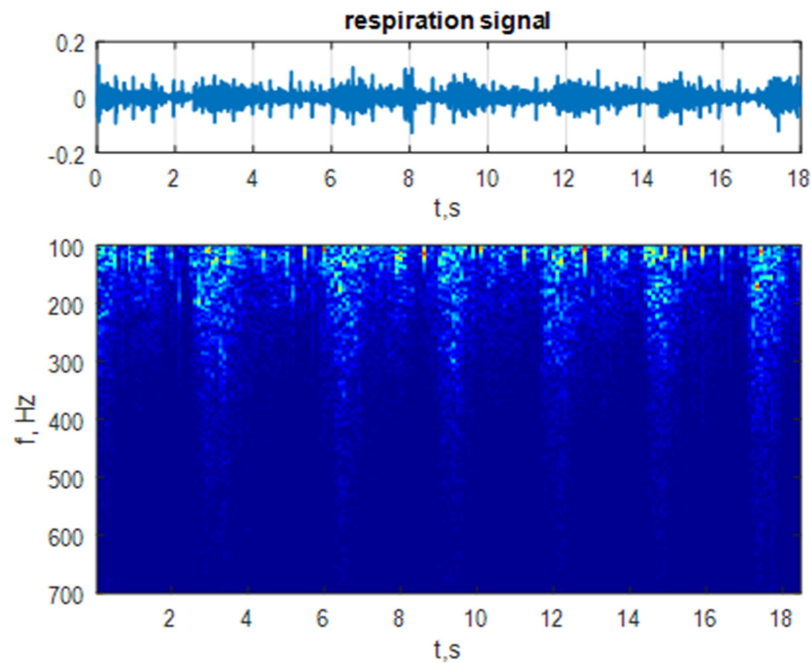


Fig. 5.22. Spectrogram of respiration signal

8 \*. Generate a vector of time samples with a duration of 10 s for a sampling frequency of 128 Hz. Generate a signal of a sequence of rectangular pulses. Generate samples of one window function lasting 0.2 seconds and 2 seconds. Construct signal spectrograms. Construct three-dimensional graphs of the modulus of the spectral function.

9 \*. Generate a vector of time samples with a duration of 10 s for a sampling frequency of 512 Hz. Use the **chirp** function to generate a variable frequency signal: from 0 Hz at time 0 s to 256 Hz at time 10 s with a quadratic frequency change. Generate two sine waves of duration 1 with a frequency of 150 Hz (S1) and 200 Hz (S2) with amplitudes of 1 V. Embed (add) to the signal of variable frequency signal S1 from the 1st to the 2nd second, and signal S2 - from 4- to the 5th second. Construct a signal spectrogram. Construct a three-dimensional graph of the modulus of the spectral function using the **surf** function.

### Control questions

1. What is time-spectral analysis of signals and why is it performed?
2. Write the mathematical expression of the spectrogram.
3. Why use window functions in spectral analysis of signals?

## LABORATORY WORK № 6

### SIGNAL FILTERING

**Purpose:** acquire skills in designing digital filters, setting the specification of filters depending on the properties of signals and noise that need to be filtered; acquire skills of realization of discrete filtering of signals.

#### Tasks

**For all signals plot the following:** graph of the initial signal and its amplitude spectrum, graph of noise and its amplitude spectrum, graph of the noisy signal (input signal of the filter) and its amplitude spectrum, graph of the frequency response of the filter, graph of the filtered signal and its amplitude spectrum.

For all filters, select the maximum allowable cut-off ratio at the cutoff frequency, and select the minimum allowable transfer coefficient at the delay frequency so that the filter order is not less than 5 and not more than 10.

\* In each task, evaluate the signal-to-noise ratio at the input and output of the filter and draw conclusions about the quality of filtering.

1. Generate a vector of time samples lasting 5 s for a sampling frequency of 128 Hz. Generate a rectangular pulse at time 3 s with a duration of 0.1 s of amplitude 1 V. Add to the signal a random noise signal with zero average value of amplitude 0.5 V. Design a Butterworth low-pass filter to remove the signal from noise (functions **buttord**, **butter**, **lfilter**) (Fig.6.1, 6.2).



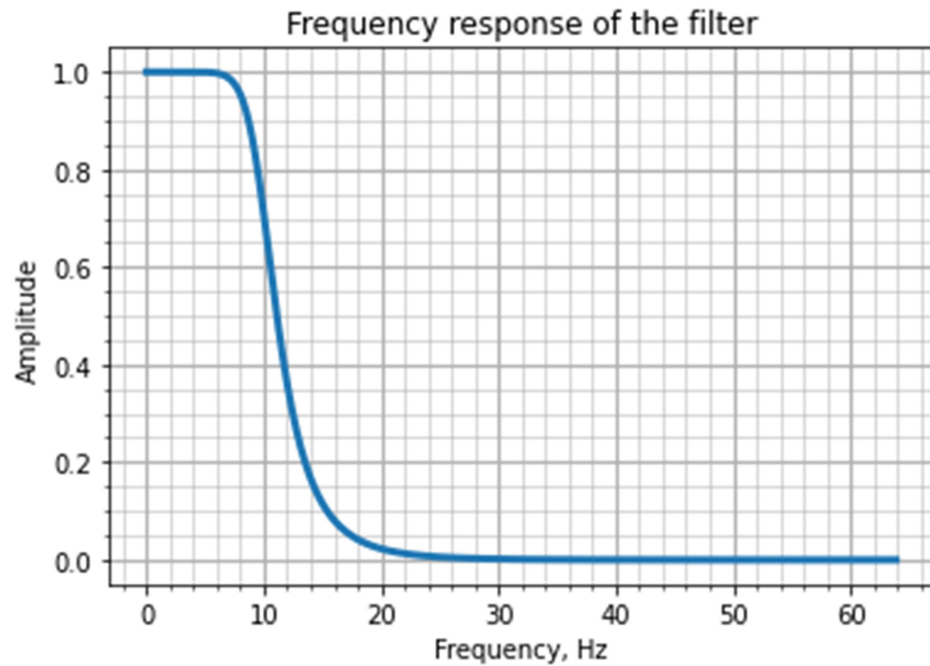


Fig. 6.1. Frequency response of Butterworth low-pass filter

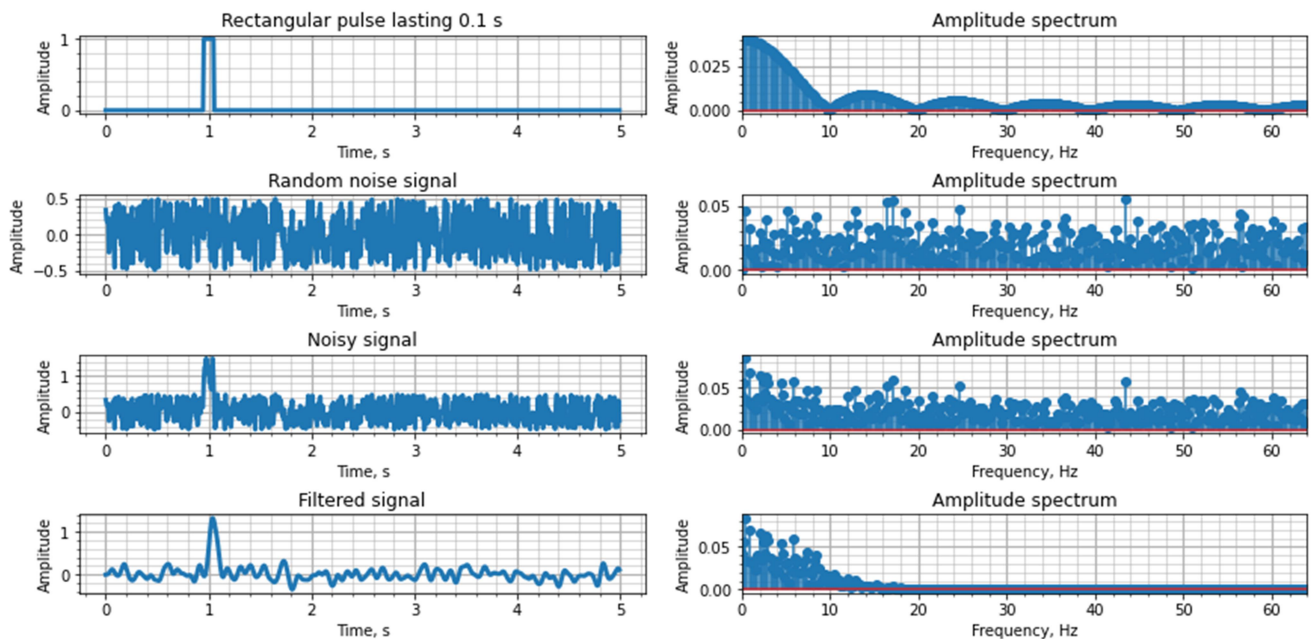


Fig. 6.2. Examples of signals and their amplitude spectra

2. Generate a vector of time samples lasting 1 s for a sampling frequency of 128 Hz. Generate signal of a sine wave with a frequency of 10 Hz and amplitude 1 V. Add a random signal with zero average value and amplitude 2 V. Design low-pass filter, high-pass filter and band-pass filter of Chebyshev type to get rid from noise (**cheb1ord**, **cheby1**) (Fig.6.3, 6.4).

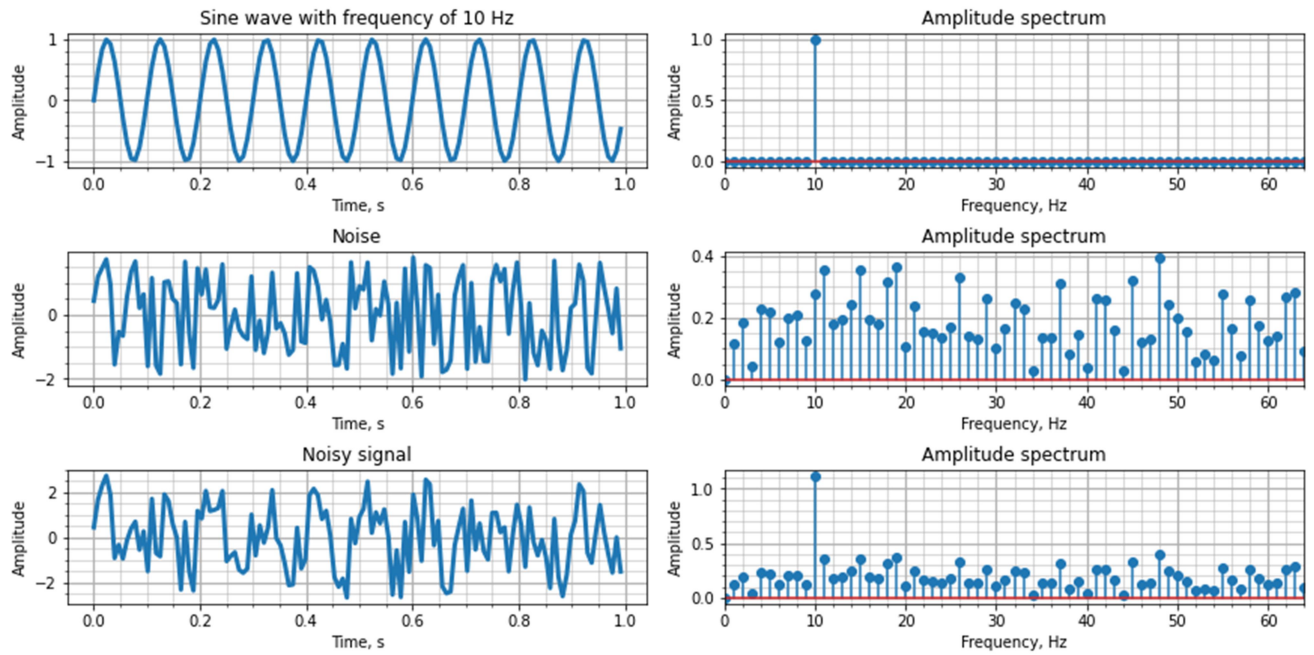


Fig. 6.3. Examples of signals and their amplitude spectra

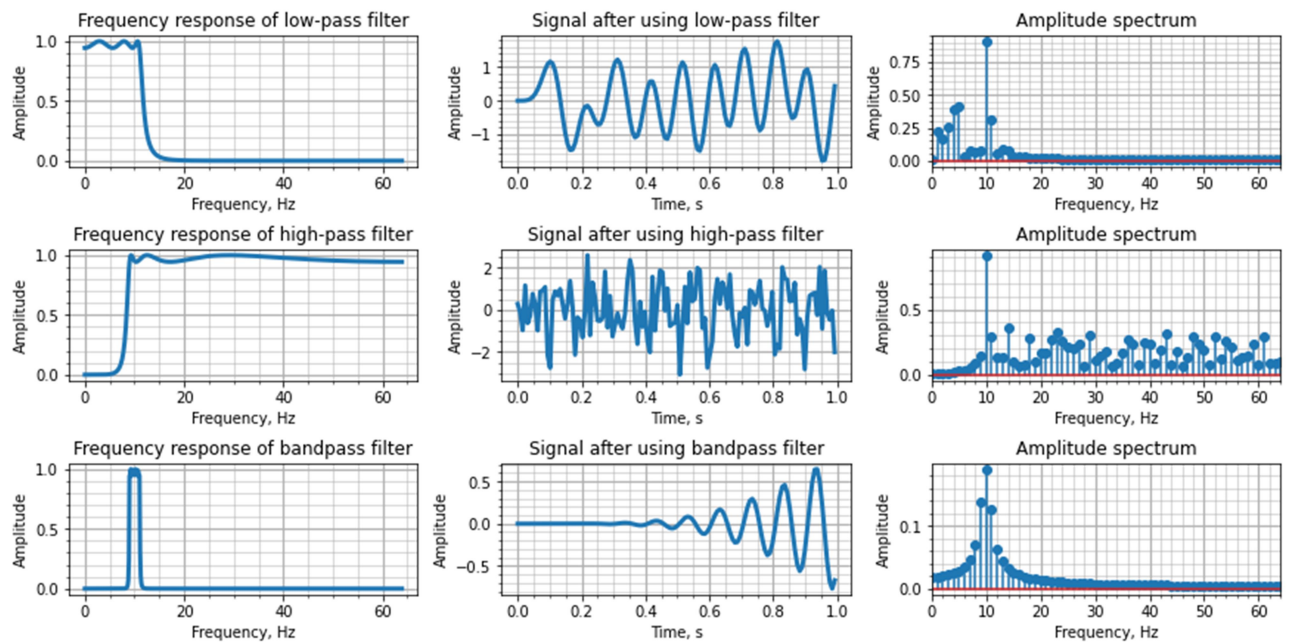


Fig. 6.4. Designing low-pass filter, high-pass filter and band-pass filter of Chebyshev type

3. Generate a vector of time samples lasting 10 s for a sampling frequency of 128 Hz. Generate a random signal of amplitude 10 mV with zero average value, which is contaminated by a network interference of frequency 50 Hz amplitude 1 V. Design Butterworth band-stop filter to remove the signal interference (Fig.6.5, 6.6).

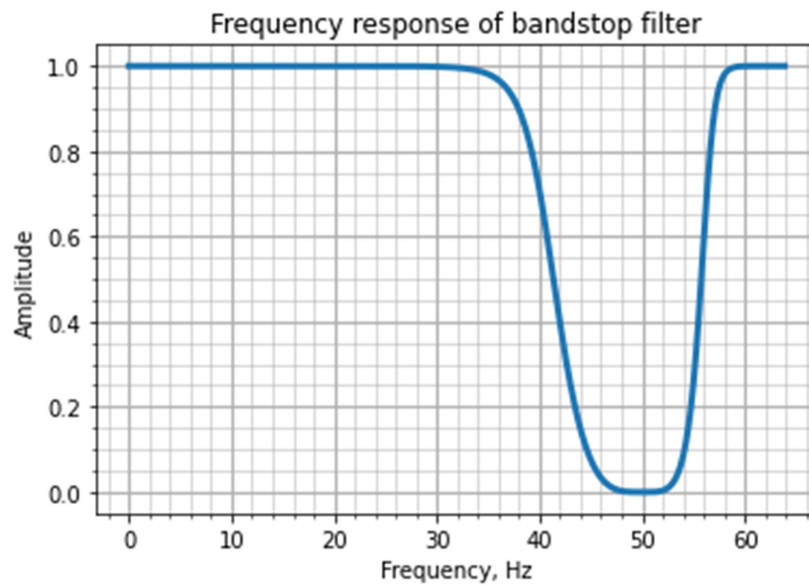


Fig. 6.5. Frequency response of Butterworth band-stop filter

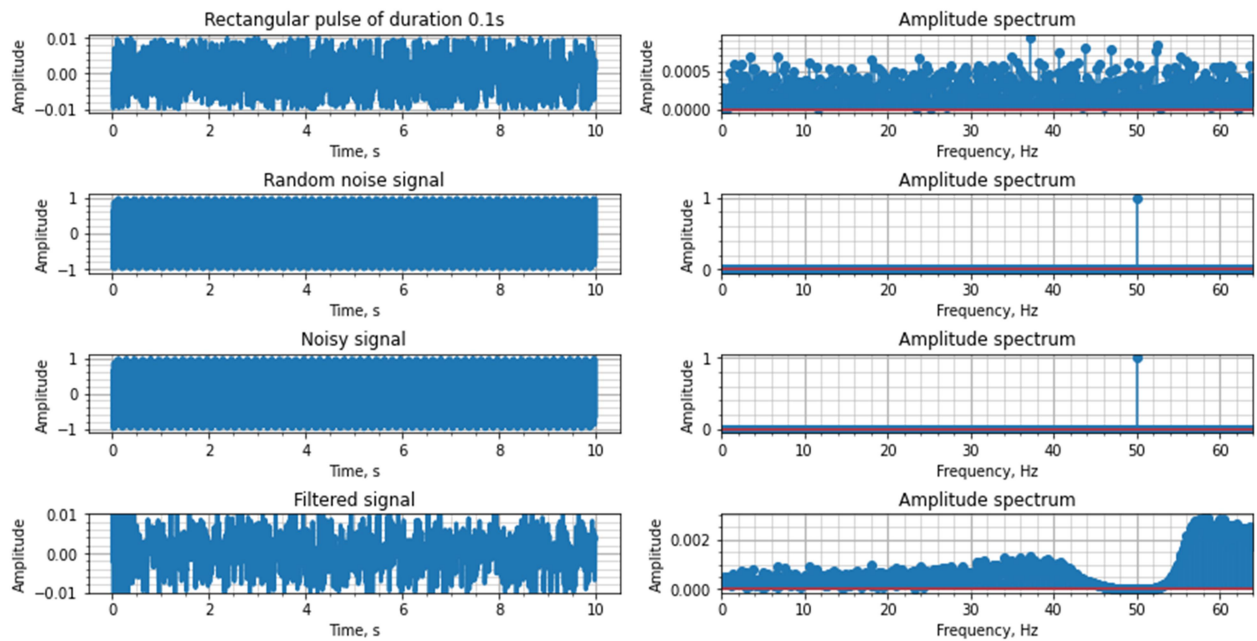


Fig. 6.6. Examples of signals and their amplitude spectra

4. For audio signals that are obtained with different sampling rates, implement the filters to extract three spectral ranges from them: up to 450 Hz; from 450 Hz to 1 kHz; from 1 kHz to 4 kHz. Listen to the received signals, draw conclusions (Fig.6.7-6.9)

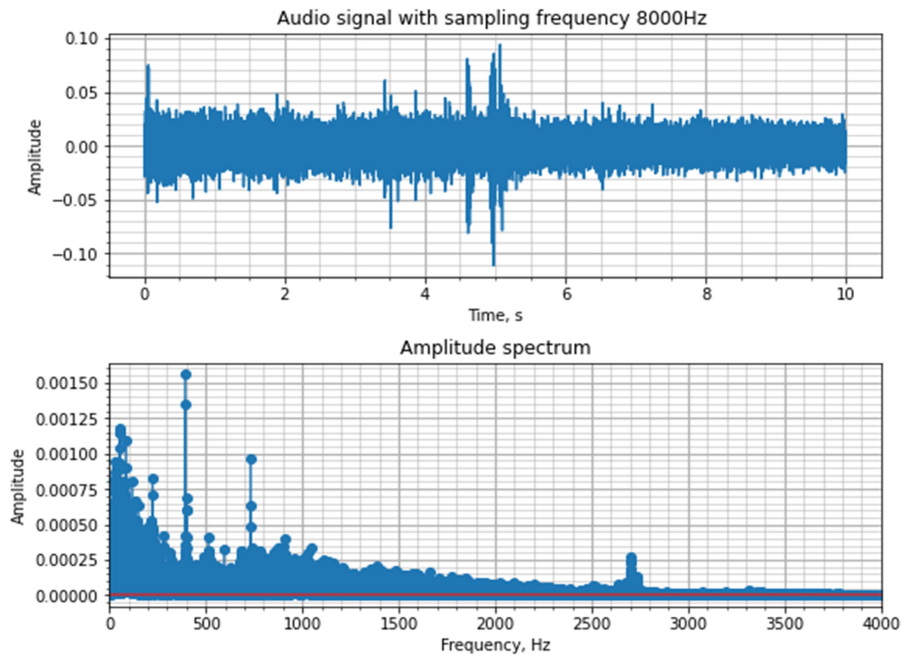


Fig. 6.7. Example of audio signal with sampling frequency 8 kHz and its amplitude spectra

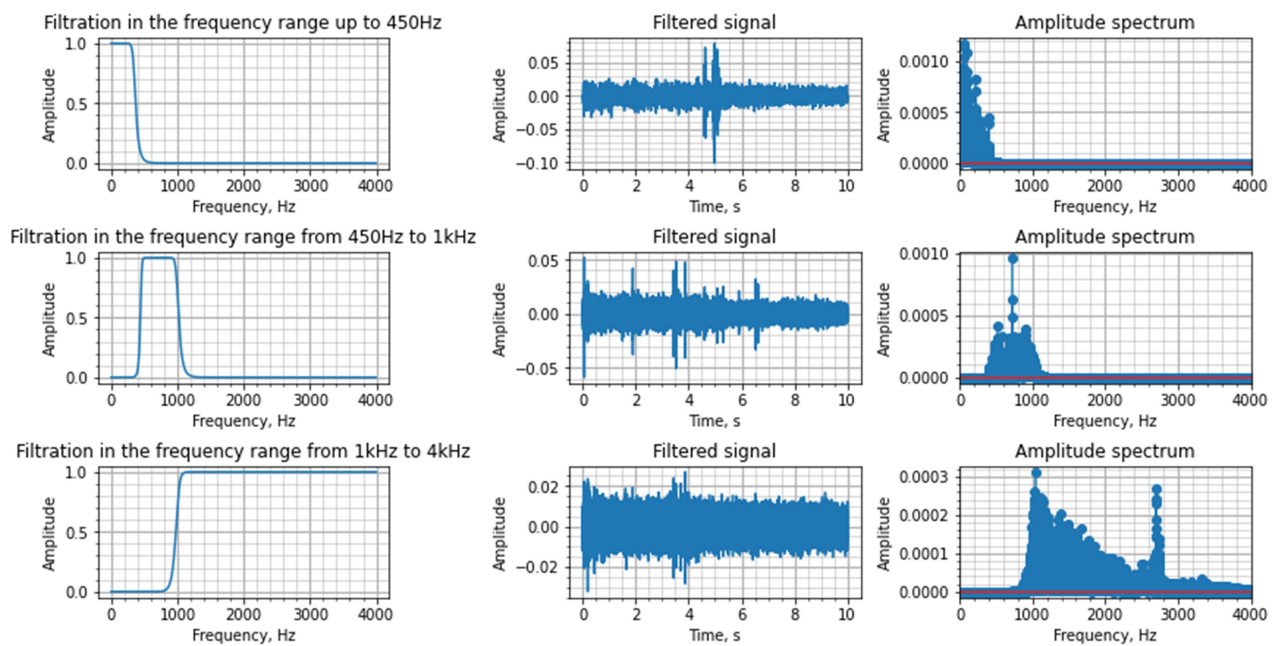


Fig. 6.8. Designing low-pass, band-pass and high-pass filters for audio signal

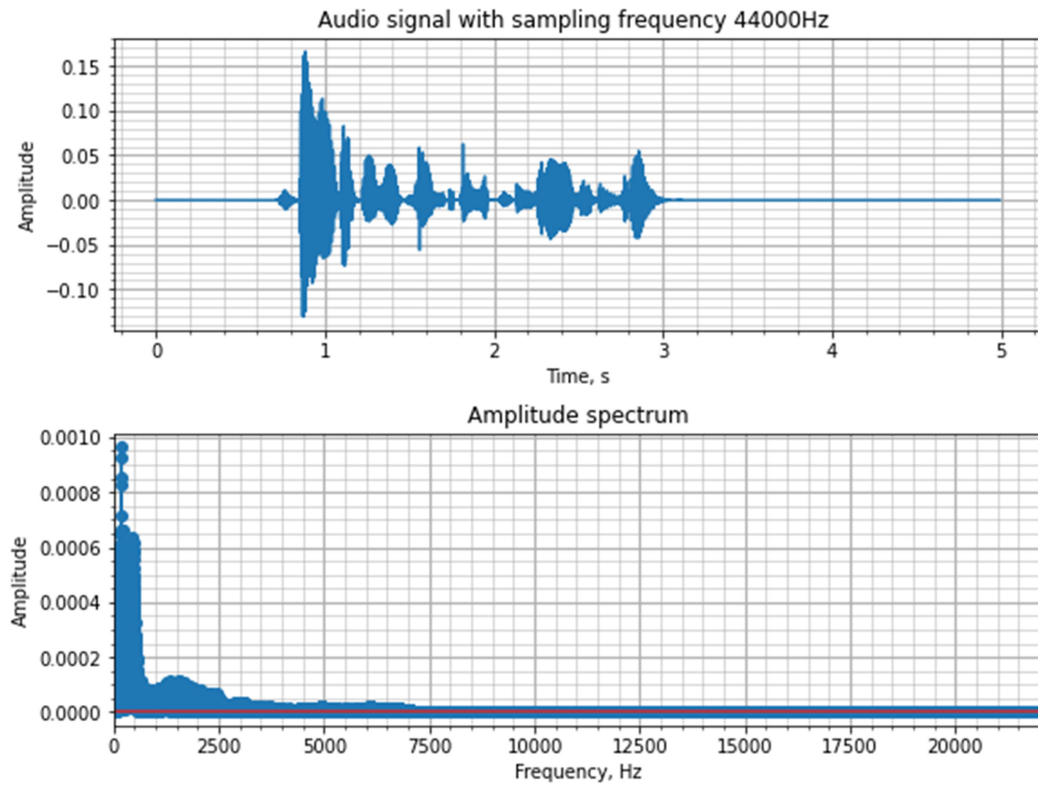


Fig. 6.9. Example of audio signal with sampling frequency 44 kHz and its amplitude spectra

5 \*. For digitized biomedical signals, design filters to get rid of the 50 Hz network interference.

6 \*. Construct a function that for the EEG signal from the file, for a given channel and time interval by filtering calculates the average spectral power density in the following spectral ranges: delta- rhythm (1-3.5 Hz), theta-rhythm (4-7 Hz), alfa-rhythm 8-13 Hz), beta-rhythm (14-30 Hz). Get graphs of each rhythm (Fig. 6.10-6.12).



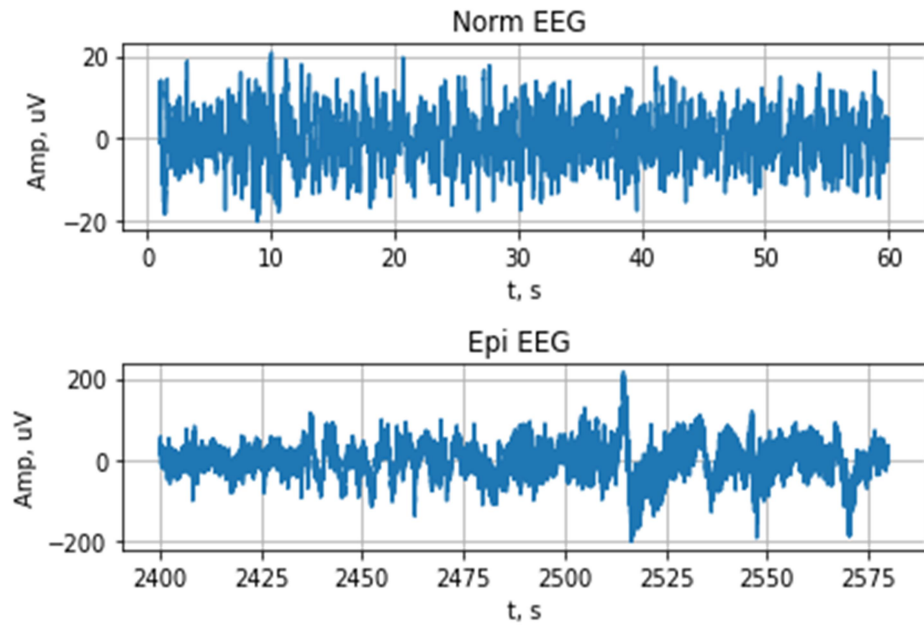


Fig. 6.10. Examples of EEG signals in norm and epilepsy

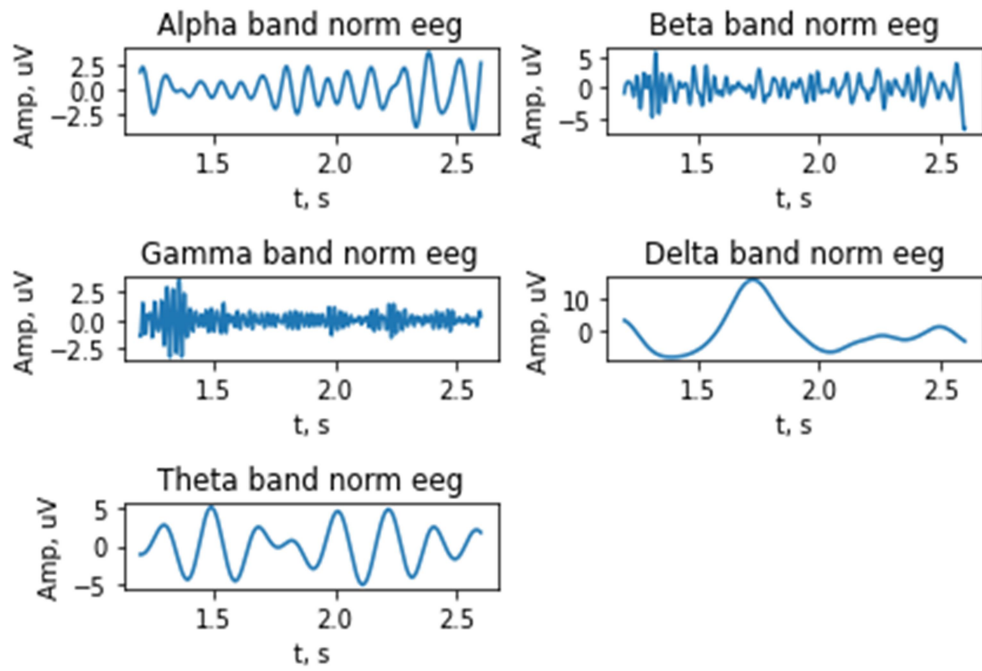


Fig. 6.11. Example of filtered EEG rhythms for norm case: delta- rhythm (1-3.5 Hz), theta-rhythm (4-7 Hz), alfa-rhythm 8-13 Hz), beta-rhythm (14-30 Hz)

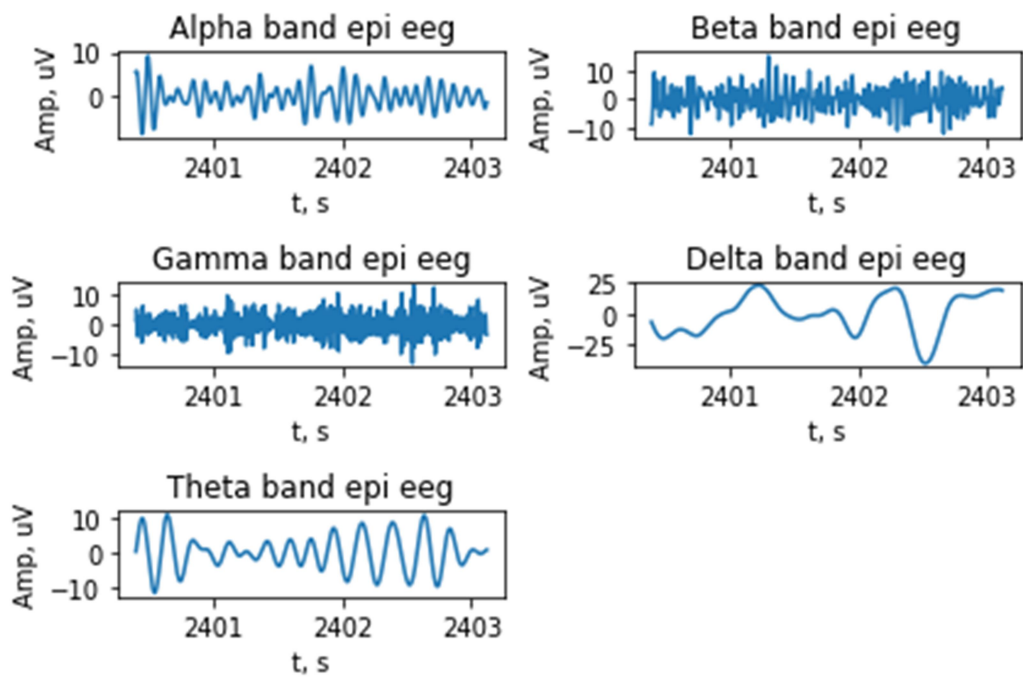


Fig. 6.12. Example of filtered EEG rhythms for epilepsy case: delta- rhythm (1-3.5 Hz), theta-rhythm (4-7 Hz), alfa-rhythm 8-13 Hz), beta-rhythm (14-30 Hz)

7 \*. Apply to signals the Chebyshev filter of the II type, the elliptical filter. Compare the results of filtering with different filters, draw conclusions.

### Control questions

1. What is signal filtering and for what purpose is it performed?
2. How is filtering implemented in Python, and what specifications are required?
3. Is there a difference in the results of applying different filters to filter the same signals? Why?

## \* LABORATORY WORK № 7

### CORRELATION ANALYSIS OF DETERMINISTIC SIGNALS

**Purpose:** investigate the properties of signals using correlation analysis; acquire skills of correlation analysis of signals.

#### Tasks

1. Generate a vector of time samples lasting 1 s for a sampling frequency of 256 Hz. Generate a random white Gaussian noise signal (**randn** function). Calculate and plot the autocorrelation function. Draw conclusions about the nature of ACF for noise (Fig.7.1).

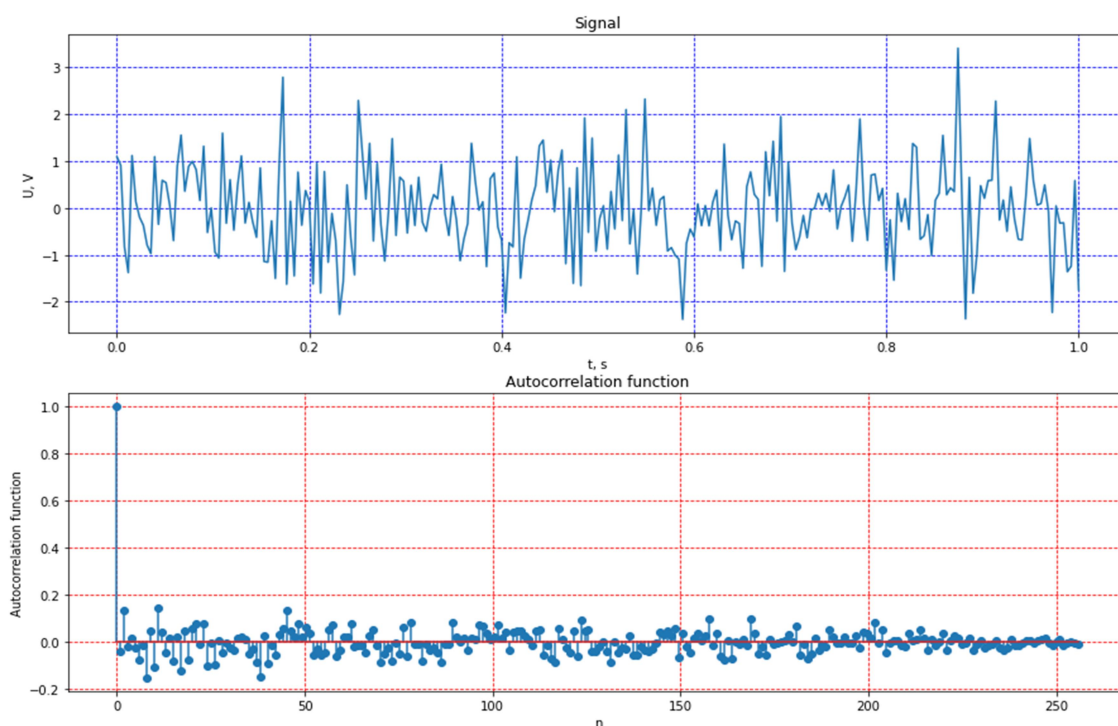


Fig.7.1. Random white Gaussian noise signal and autocorrelation function

2. Generate a vector of time samples lasting 1 s for a sampling frequency of 256 Hz. Generate a discrete analog of the signal  $x(t) = 5 \cos(2\pi 50t) + 2 \cos(2\pi 100t)$ . Plot a graph of the autocorrelation function (Fig.7.2). Draw conclusions about the nature of the ACF for the periodic signal.



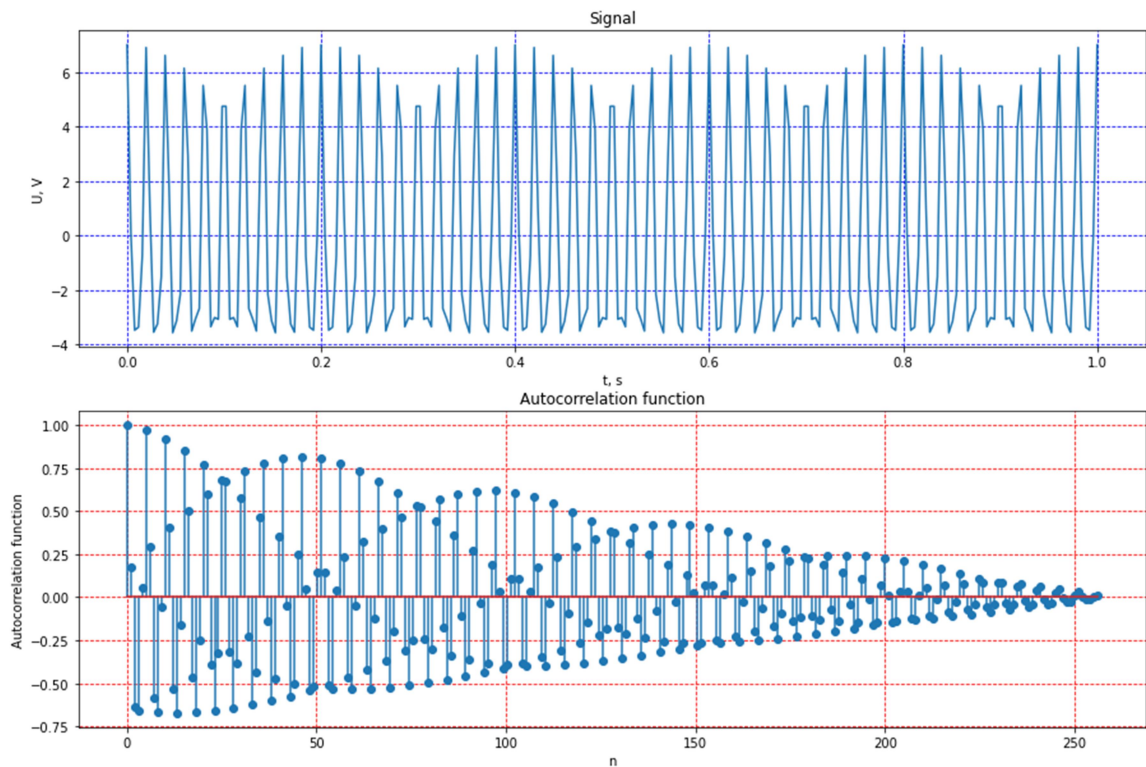


Fig.7.2. Signal and autocorrelation function

3. Generate a vector of time samples lasting 10 s for a sampling frequency of 256 Hz. Generate signals of rectangular pulse of amplitude 1 V for 1 sec. at a time of 3 sec. (signal 1), and a rectangular pulse of amplitude 1 V for 1 sec. at time 0 sec. (signal 2). Plot a graph of the cross-correlation function, draw conclusions about the possibility of determining the localization in time of a rectangular pulse using correlation analysis (Fig.7.3).

4. Repeat step 3, pre-adding to the signal 1 noise with zero average value. Draw conclusions, compare with Task 3 (Fig.7.4).

5 \*. Calculate and plot autocorrelation functions for digitized biosignals, draw conclusions.

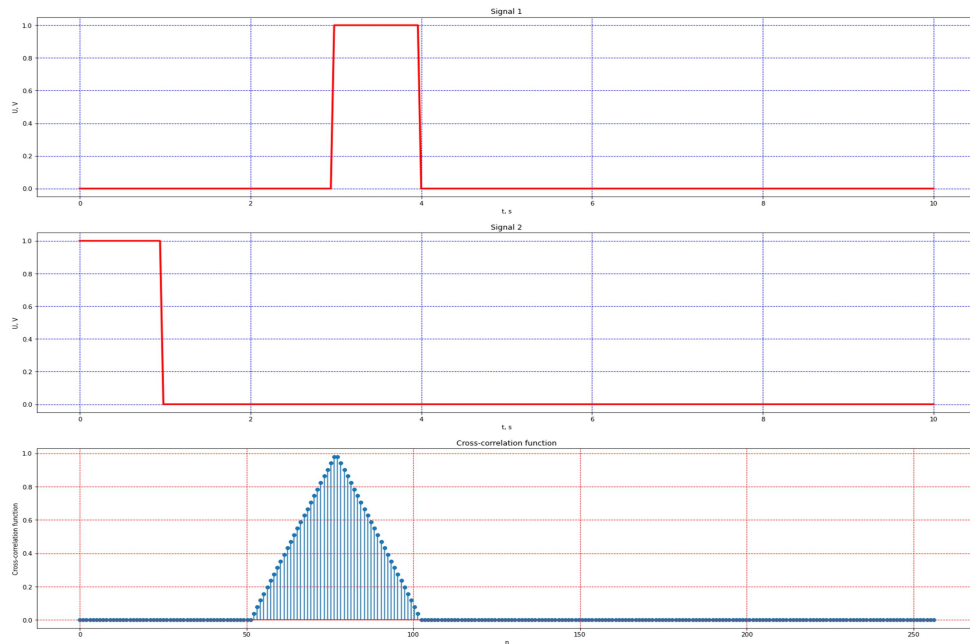


Fig.7.3. Rectangular signals and cross-correlation function

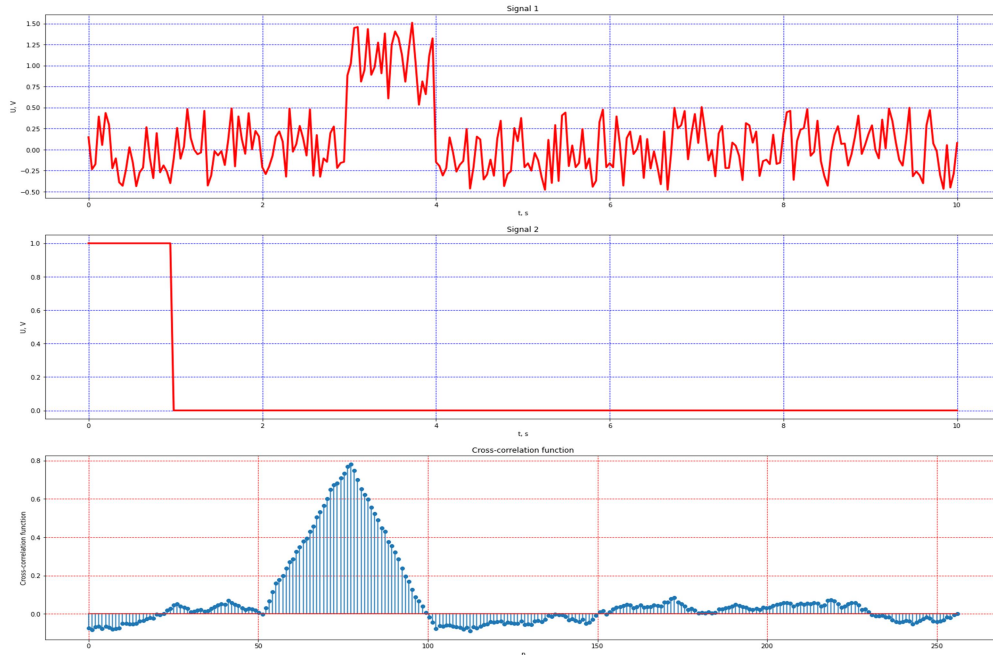


Fig.7.4. Rectangular signals with noise component in first signal and cross-correlation function

### Control questions

1. What is the correlation analysis of signals and for what purpose is it performed?
2. How are the results of correlation analysis presented and what information can be obtained from them?
3. What is the difference between autocorrelation and cross-correlation function?\*

## \* LABORATORY WORK № 8

### WAVELET SIGNAL ANALYSIS

**Purpose:** to investigate the reflection of signal properties in the wavelet scalogram; acquire skills in implementing wavelet signal conversion.

#### Tasks

1. Plot graphs of 256 samples of Ricker wavelet and Morle wavelet (Fig.8.1).

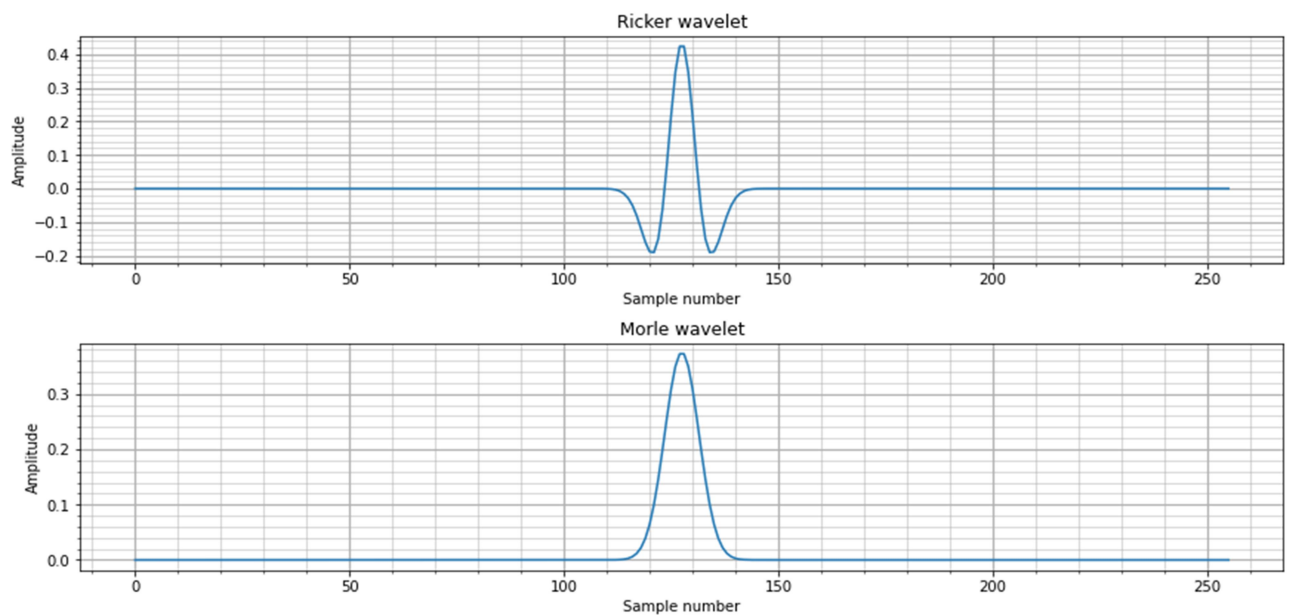


Fig.8.1. Ricker wavelet and Morle wavelet

2. Form a vector of scale factors  $a = [.01: .02: .11 .2: .2: 1 2: 2: 30]$ .

Build scalograms of signals from laboratory work on spectral-temporal analysis. Based on the analysis of scalograms, draw conclusions about the conformity of the temporal properties of signals with their reflection on the scalogram (Fig.8.2, 8.3). Compare the reflection of the characteristics of the signals on the scalogram and spectrogram, draw conclusions.

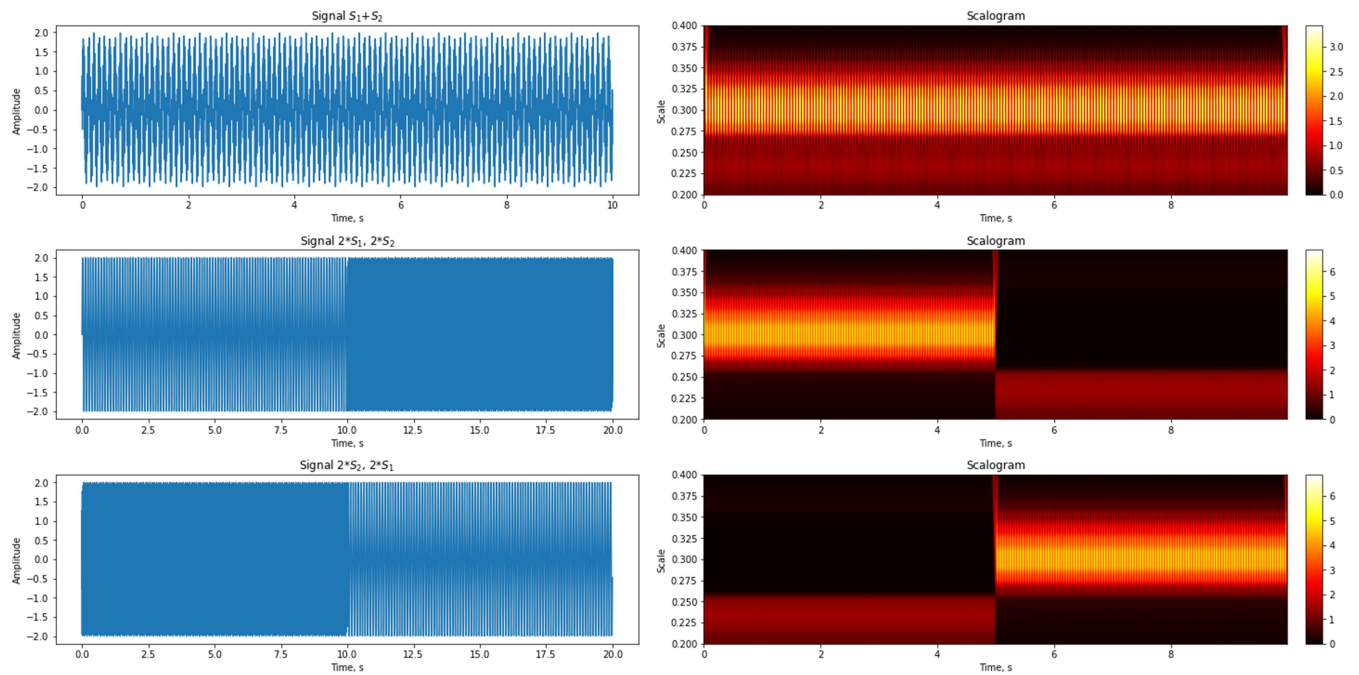


Fig.8.2. Sum of two sine waves of 10 and 100 Hz frequency and its scalogram;  
concatenation of two sine waves of 10 and 100 Hz frequency and its scalogram;  
concatenation of two sine waves of 100 and 10 Hz frequency and its scalogram

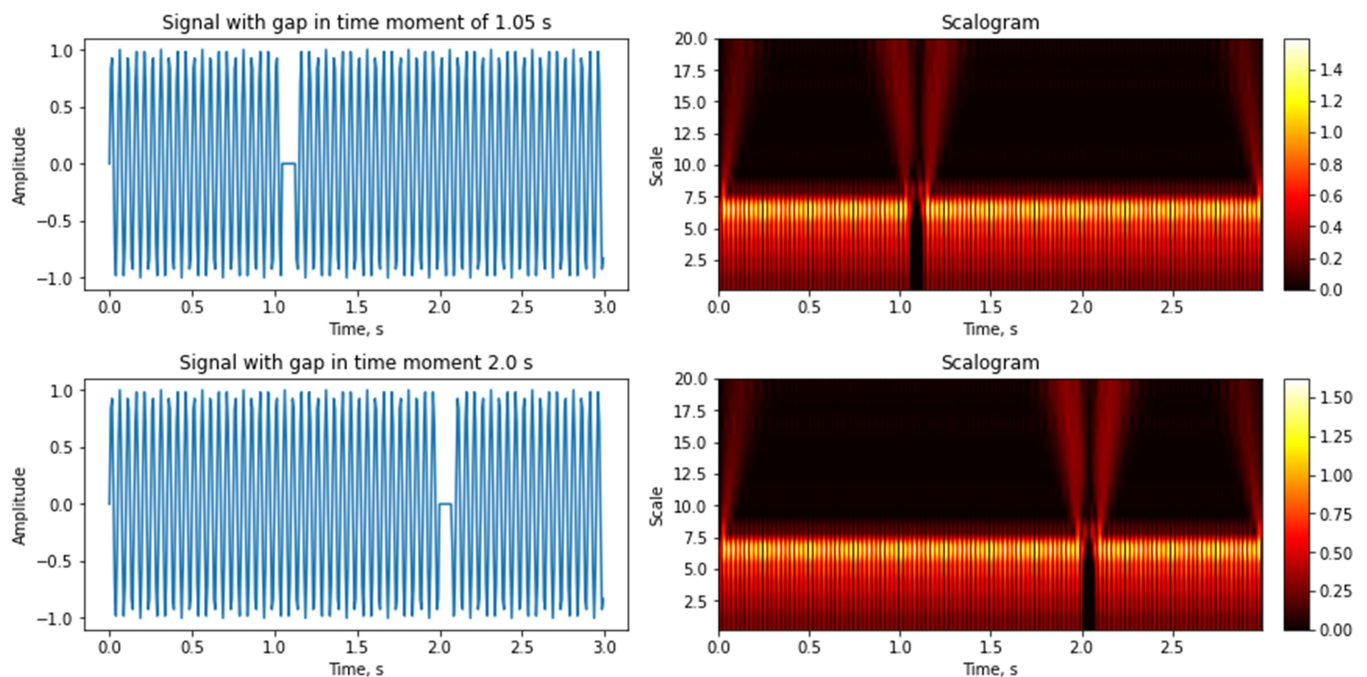


Fig.8.3. Signals of the sine wave with gaps in the signal at time moments of 1.05 s  
and 2 s and their scalograms

3. Construct a function that plots the change in time of the sum of the modules of the wavelet coefficients in a given range of scales for a given signal.

4 \*. Using symbolic calculations, plot the first 10 Gaussian wavelets, which are derived from the Gaussian function. Perform wavelet normalization.

5 \*. Perform continuous wavelet analysis of one-dimensional biomedical signals; pick up the scales to obtain informative scalograms. Compare with the results of spectral and spectral-temporal analysis, draw conclusions

### **Control questions**

1. What is wavelet analysis of signals and for what purpose it is performed.
2. What do you need to perform a wavelet analysis?
3. How clearly are the results of wavelet analysis of signals? What information can be obtained from them?

## \* LABORATORY WORK № 9

### PROCESSING OF MEDICAL IMAGES

**Purpose:** get acquainted with the principles of obtaining and processing biomedical images, the main functions of their digital processing using the Python language, as well as gaining skills in the study of biomedical images.

#### TASKS

1. Introduction to the basic functions of working with images.

1.1. Get acquainted with the work and parameters of image opening functions. Download a biomedical image. Determine the size of the image matrix, get information about the type of image.

1.2. Display a matrix corresponding to the studied image for the following cases:

- 1) use of default brightness parameters;
- 2) the use of three levels of brightness;
- 3) the use of brightness levels in the form of an empty array;
- 4) use of brightness levels from 20 to 30;
- 5) use of brightness levels from 0.4 to 0.6.

Display a color scale in each window (Fig.9.1).

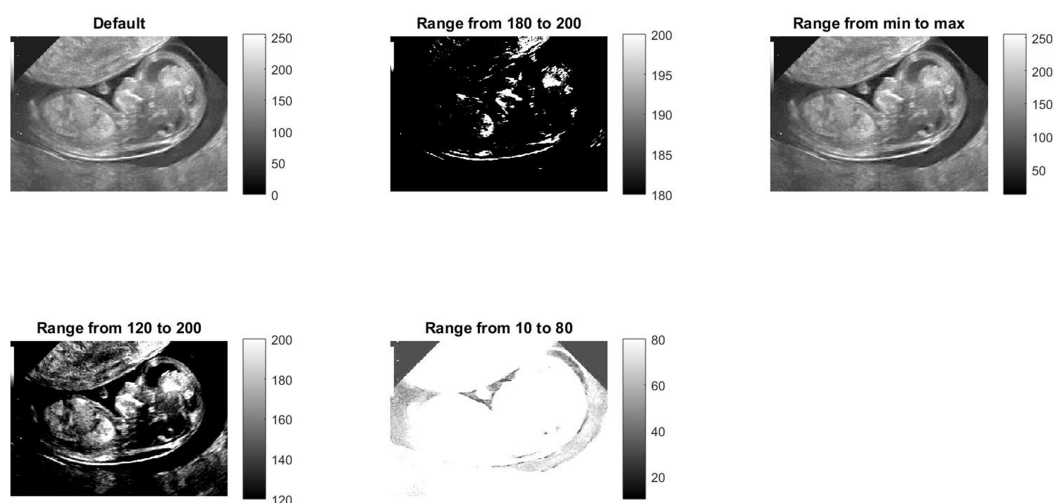


Fig.9.1. Brightness parameters

1.3. Get acquainted with the function of recording image files of different types. Write the same image in a jpeg file with quality parameters  $q = 0, 5, 10, 20, 50, 75$ ,



100 (Fig.9.2). Construct the dependence of the degree of compression of the image on the quality parameter.

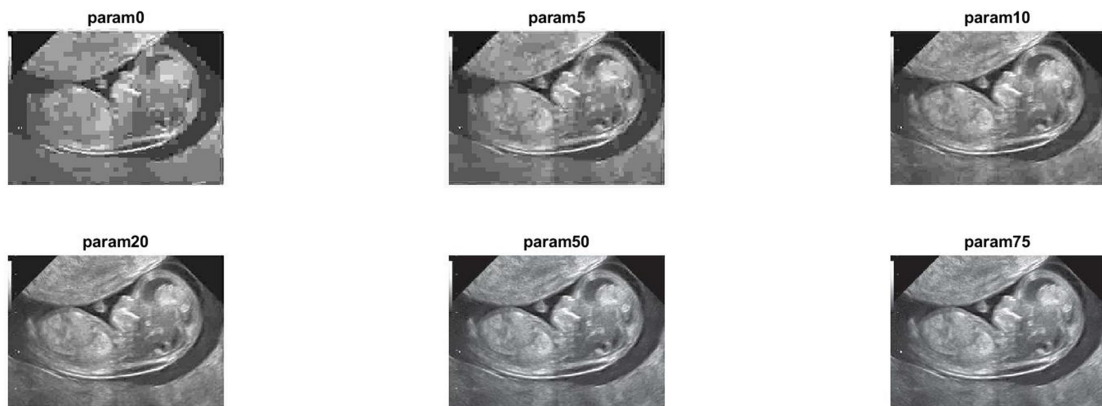


Fig.9.2. Quality parameters

1.4. Get the conversion of a full color image to a grayscale image, display it on the screen (Fig.9.3).

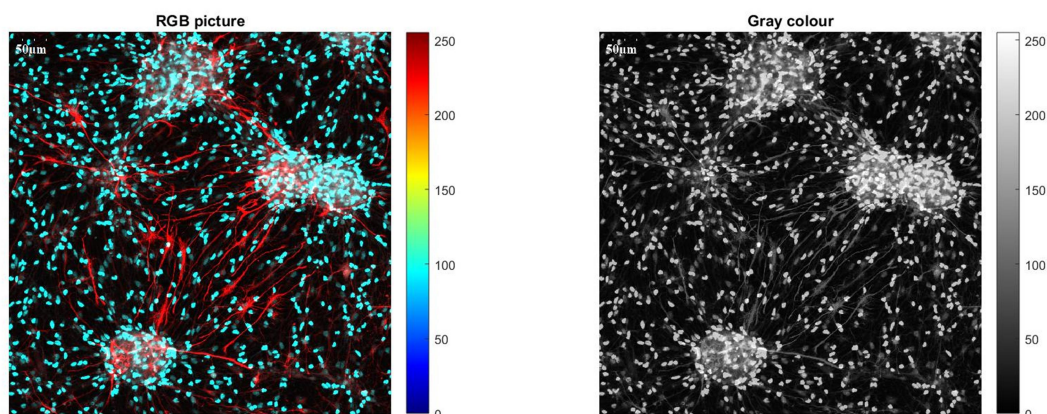


Fig.9.3. Conversion of a full color image to a grayscale image

2. Spatial filtering of speckle noise of the image using a median filter

2.1. Add simulated speckle noise to the studied biomedical image (Fig.9.4).

2.2. Perform the median filtering of the image for the sizes of 3x3 and 10x10 pixels. Determine the signal-to-noise ratio before and after filtering for the studied image. Get the difference between noisy and filtered images (Fig.9.5).

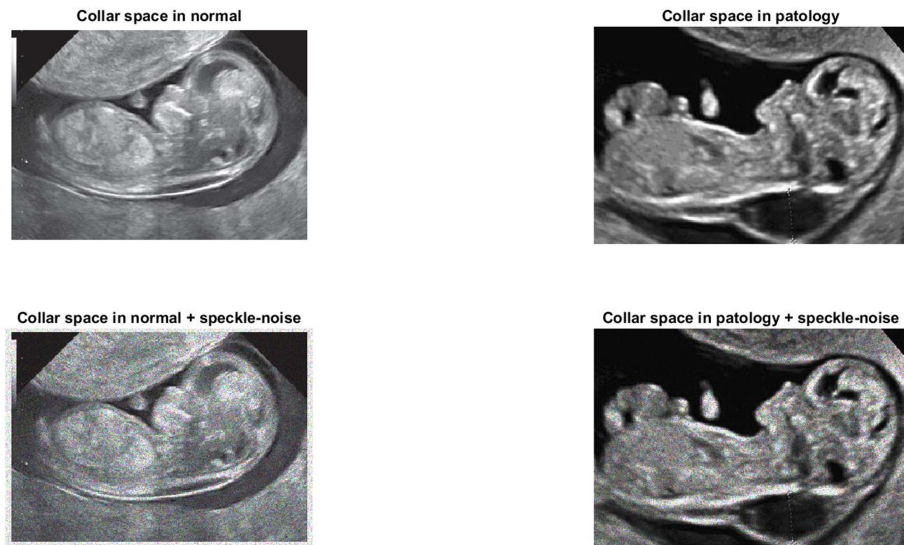


Fig.9.4. Speckle noise in biomedical image

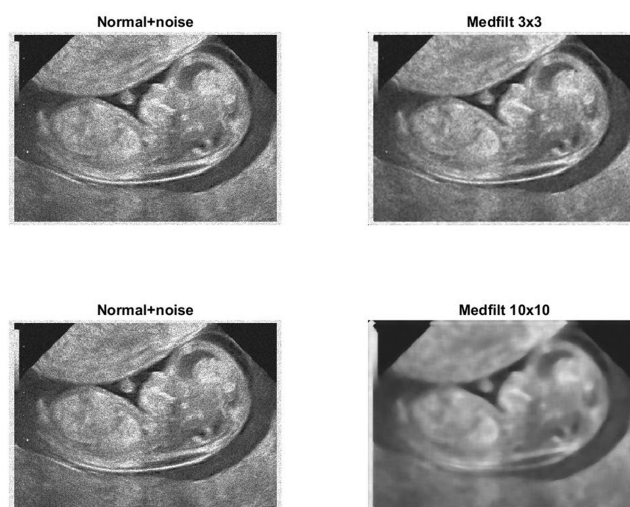
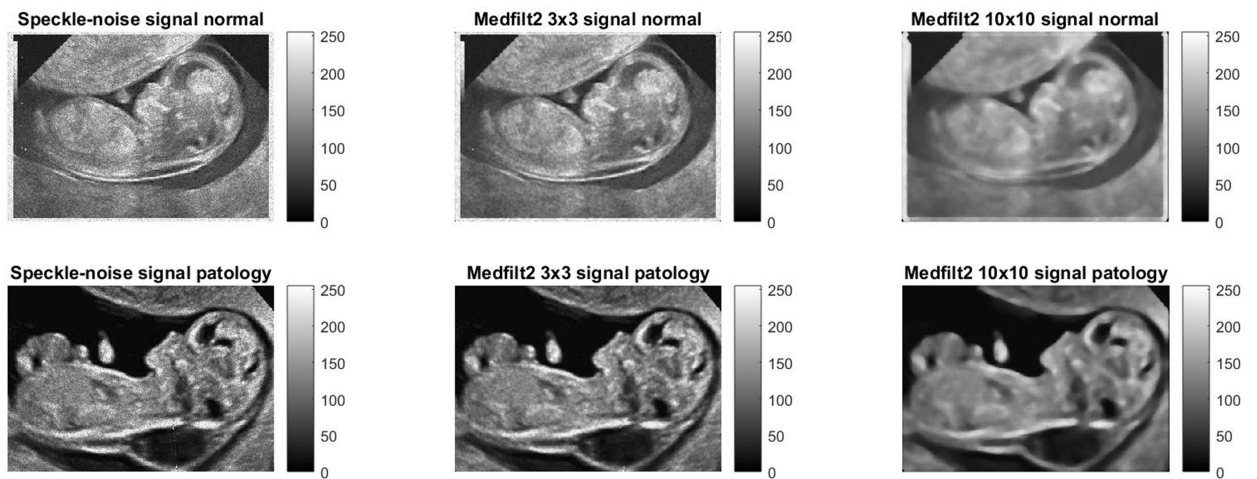


Fig. 9.5. Median filtering of image



2.3. Perform spatial image filtering using a number of different masks (Fig.9.6). Determine the signal-to-noise ratio before and after filtering for each case. Draw conclusions.

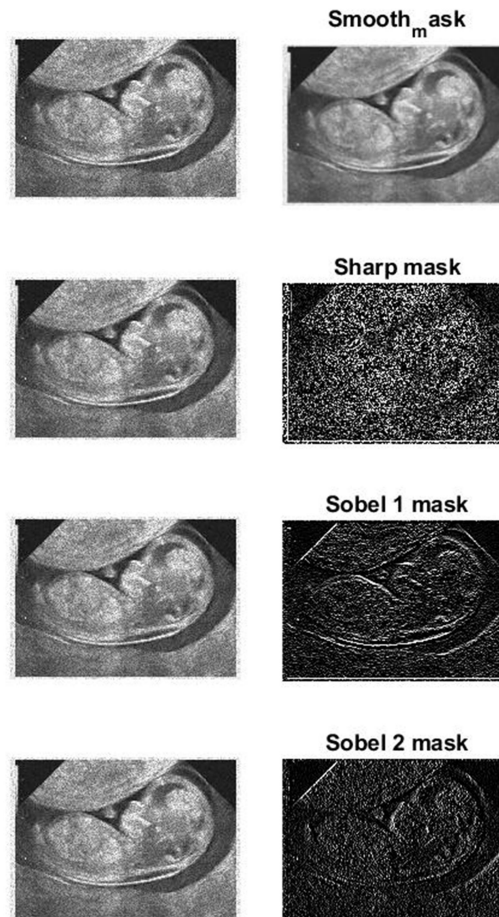


Fig.9.6. Spatial image filtering using different masks

3.1. To study the theoretical foundations of the two-dimensional Fourier transform. Construct the centered amplitude spectra of the original image and the noisy image (Fig.9.7).

Obtain the reconstructed image from the available two-dimensional Fourier spectrum (Fig.9.8). Draw conclusions about the display of image properties in the frequency domain. Required functions are `fft2`, `fftshift`, `ifft2`, `ifftshift`.

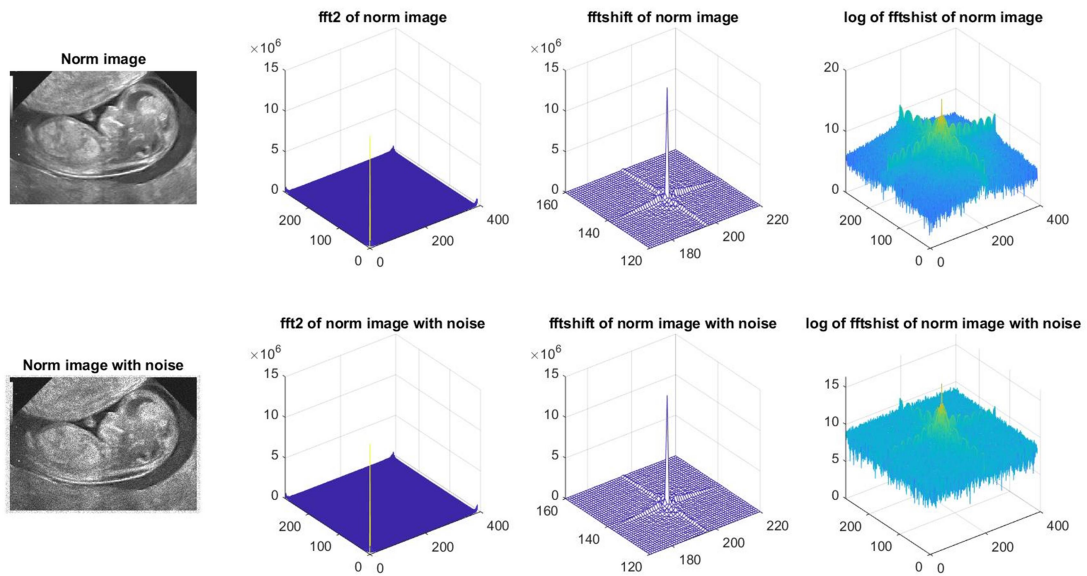


Fig.9.7. Centered amplitude spectra of the original image and the noisy image

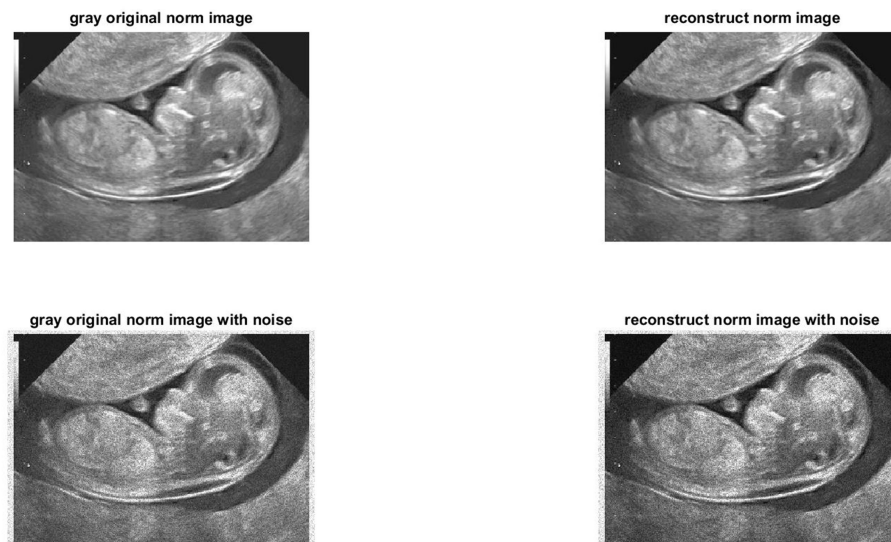


Fig.9.8. Reconstructed image from the available two-dimensional Fourier spectrum

#### 4. Processing of the histogram of image brightness

4.1. Build histograms of grayscale images read from the file. Display the image and color scale.

4.2. Equalize histograms, display the resulting images and histograms obtained after processing (Fig.9.9).

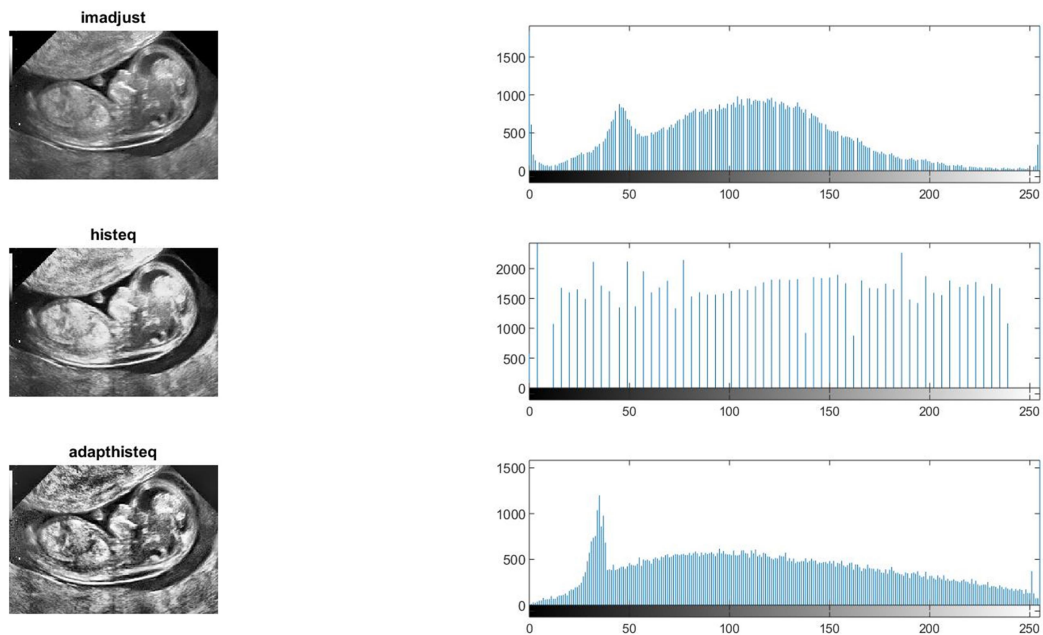


Fig.9.9. Equalizing of histograms

## 6. Methods of finding the boundaries of image elements

6.1. Study the theoretical foundations of the basic methods of finding boundaries in images. Find the boundaries of image elements by methods of Sobel; Prewitt; Roberts; Canny; Gauss.

7. Select the boundaries of the elements in the images obtained using a confocal microscope.

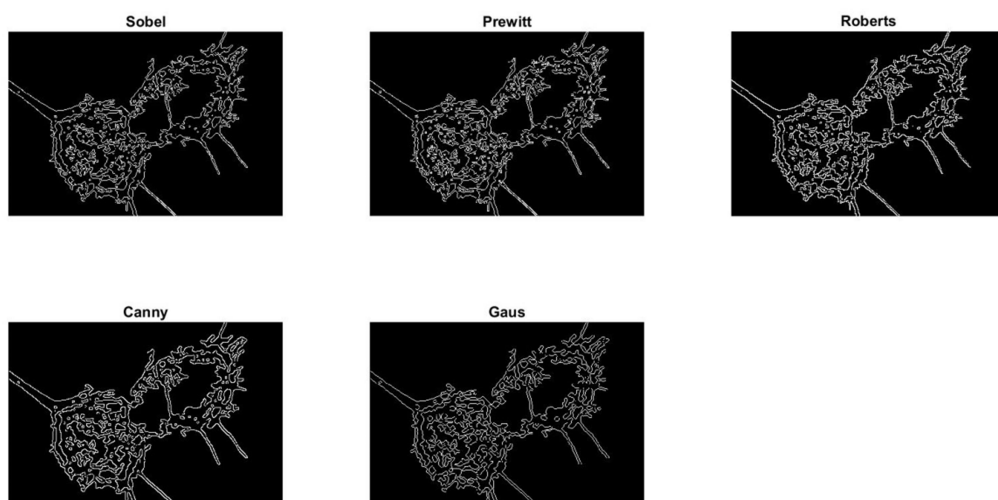


Fig.9.10. Selecting the boundaries of the elements in the images

**Control questions**

1. Define the concept of a digital image.
2. What is median image filtering?
3. What is image spatial filtering?
4. What is image histogram equalization?
5. What detectors do you know for detecting the edges of objects in an image?

## REFERENCES

1. Najarian, K., & Splinter, R. (2012). Biomedical Signal and Image Processing (2nd ed.). CRC Press. <https://doi.org/10.1201/b11978>
2. Rangayyan, Rangaraj M. Biomedical Signal Analysis Wiley (2015). IEEE Press series in biomedical engineering. [DOI:10.1002/9781119068129](https://doi.org/10.1002/9781119068129)
3. Bronzino J. D. The biomedical engineering handbook. Boca Raton: CRC Press LLC, 2000.
4. Marques de Sá J. P. Pattern recognition: concepts, methods, and applications / Marques de Sá J. P. — Springer, 2001. — 318 p.
5. Blinowska, K. J., & Zygierecz, J. (2011). Practical Biomedical Signal Analysis Using MATLAB® (Series in Medical Physics and Biomedical Engineering). CRC Press.
6. Gopi, E. S. (2012). *Digital Signal Processing for Medical Imaging Using Matlab*. Springer.
7. Kaniusas, E. (2012). *Biomedical Signals and Sensors I: Linking Physiological Phenomena and Biosignals*. Springer.
8. Devasahayam, S. R. (2012). *Signals and systems in biomedical engineering: signal processing and physiological systems modeling*. Springer.
9. Advanced Methods of Biomedical Signal Processing/ Sergio Cerutti (Editor), Carlo Marchesi (Editor), April 2011, Wiley-IEEE Press, 608 pages.