Digital Signal Processing

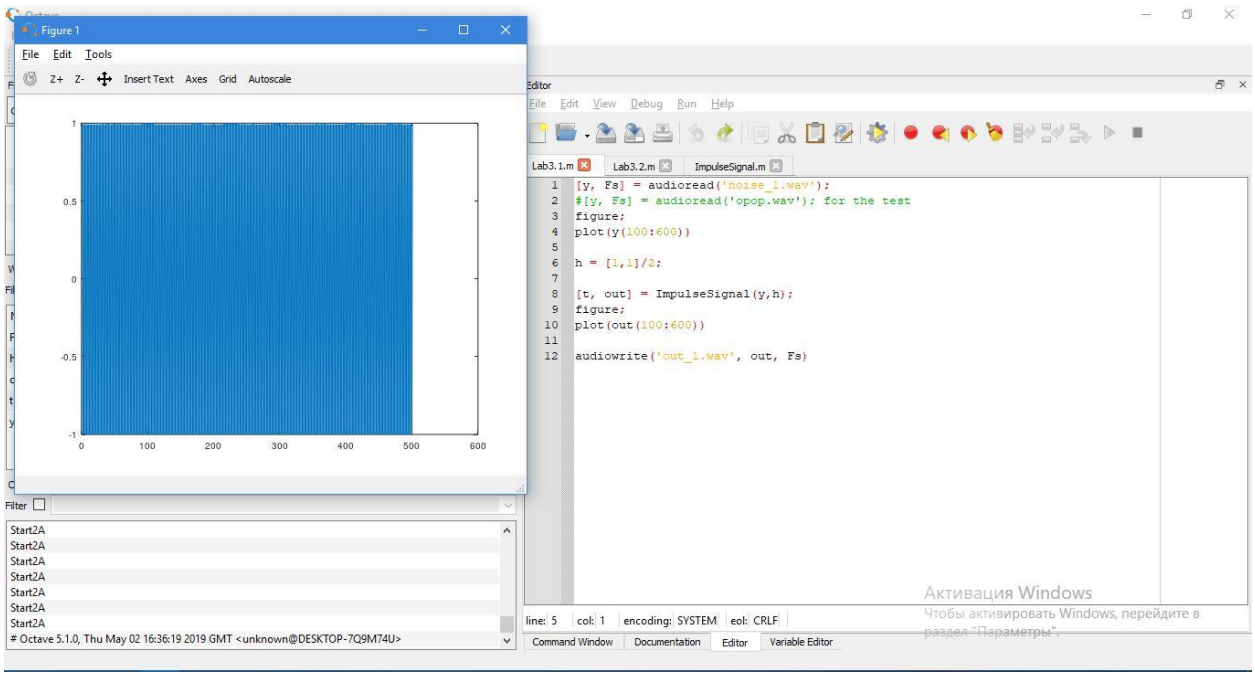
Lab 3: Convolution and Frequency Domain Filtering

Vladislav Paškevitš

Exercise 1

Read the provided noise1.wav file and perform the following:

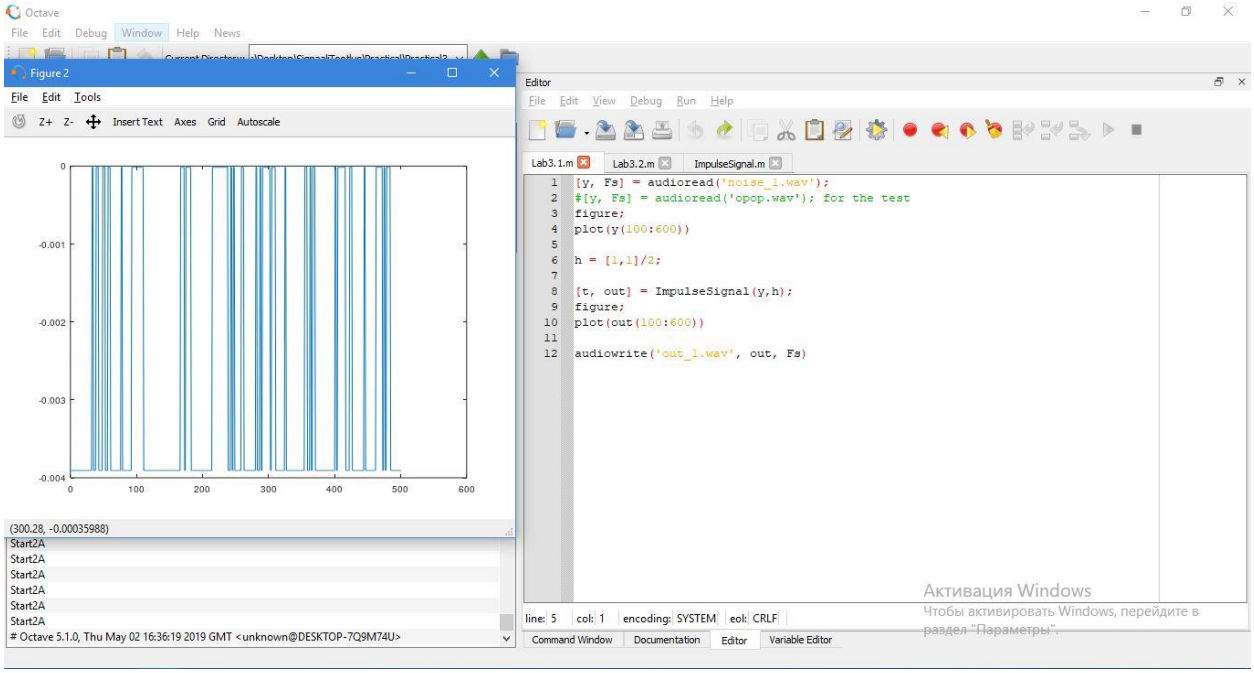
(a) Plot a small section of the audio data



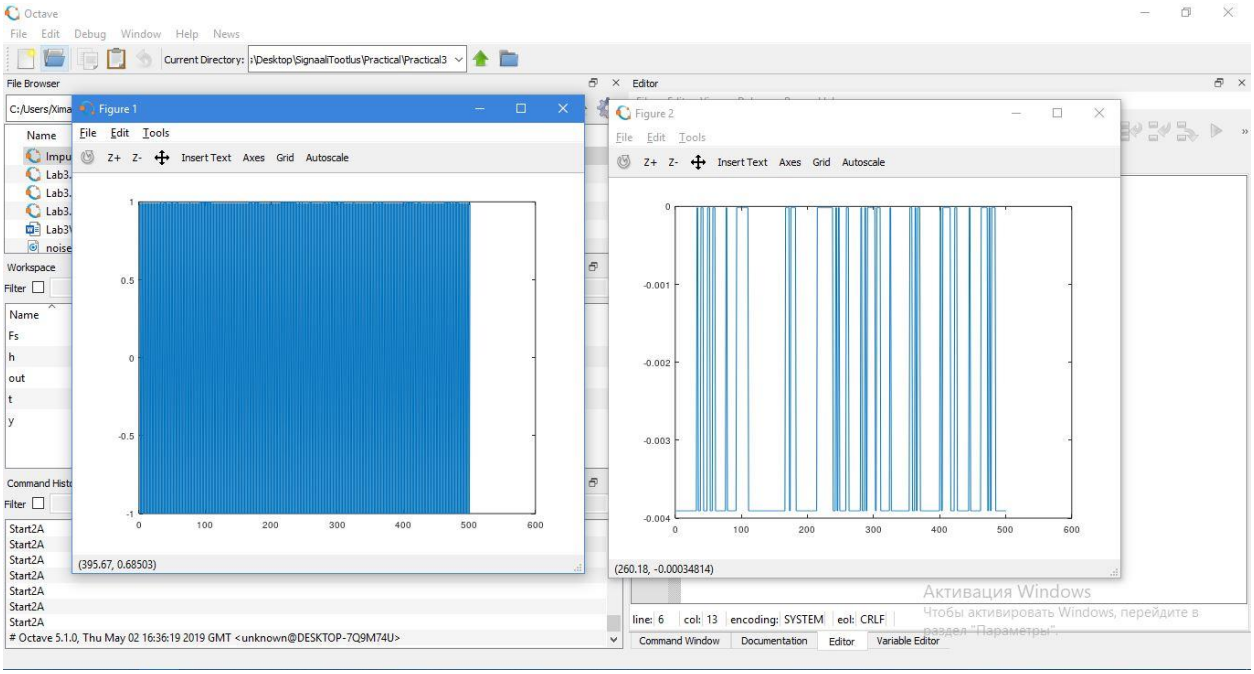
(b) Describe what type of noise is present in the file

The closest thing that would fit is shot noise. I think , that we can not say that this is just an ultrasound, it seems that it is not correct.

(c) Come up with a delta function h[n] that would remove the specific noise present in the audio look at the noise and come up with the simplest delta function that would do the trick.



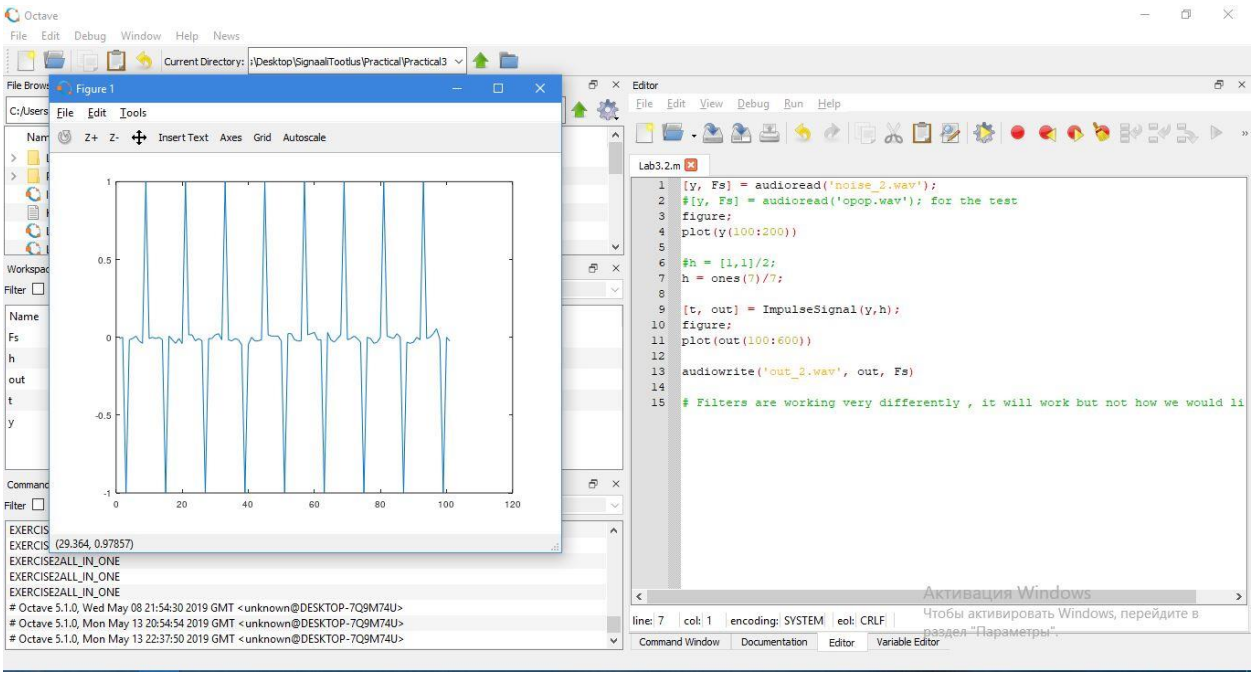
(c) Plot the results on the same graph



Exercise 2

Repeat for noise2.wav, but consider:

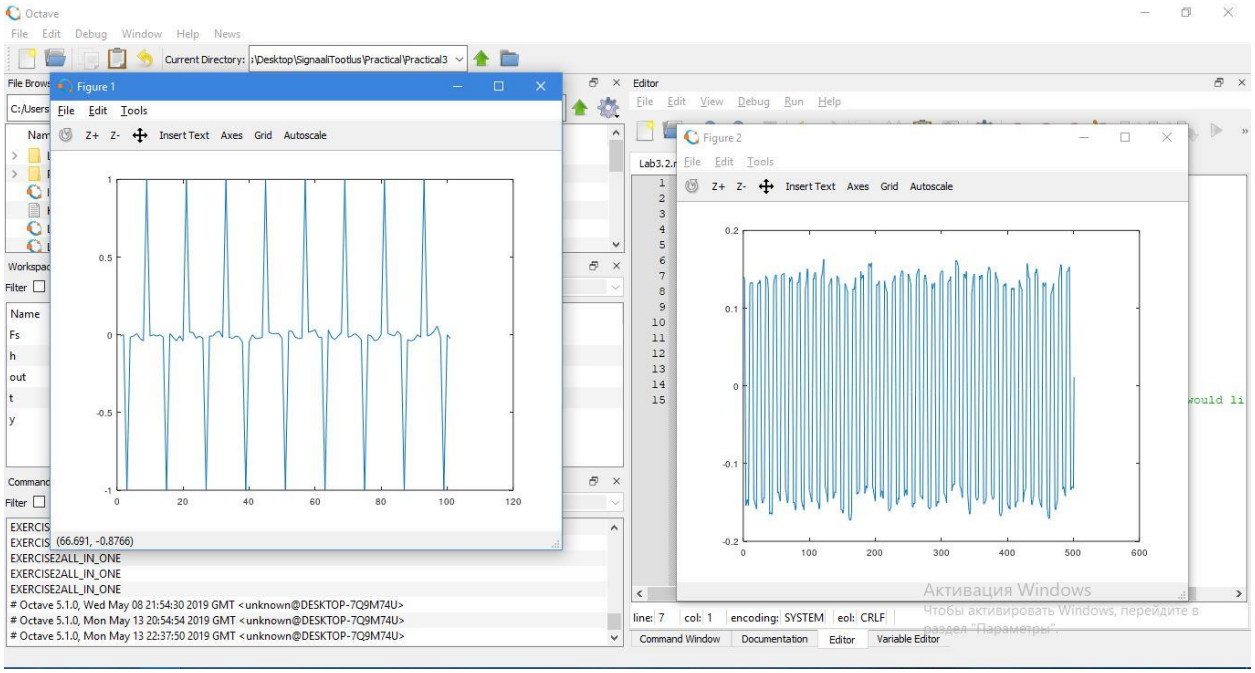
(a) Apply the same delta function you came up with in ex 1.



(b) Did it work, why? why not?

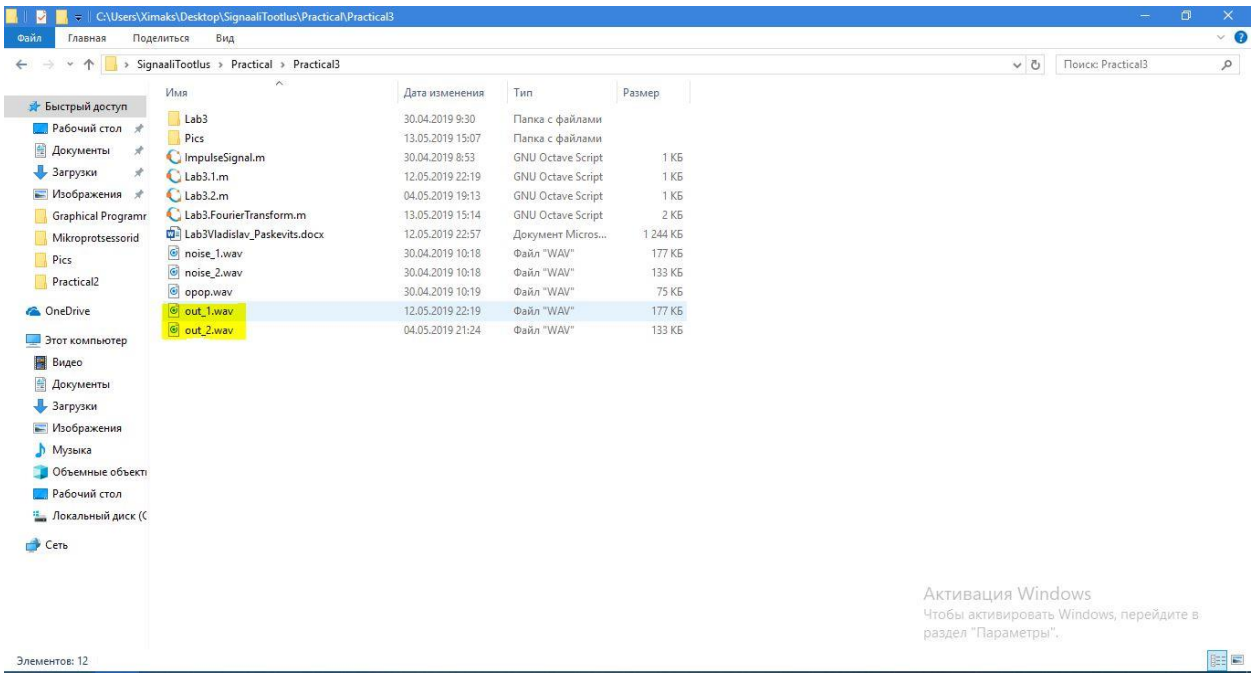
Same delta function, that we came up in ex 1 is not working in this moment.

(c) Come up with a delta function g[n] that only works for the noise in noise2.wav and ignores the noise in noise1.wav (apply both delta functions for both signals and show the results in a graph)



(d) if 2. b) worked then simplify the delta function h[n] in ex 1.

(e) Save both of the denoised audio files

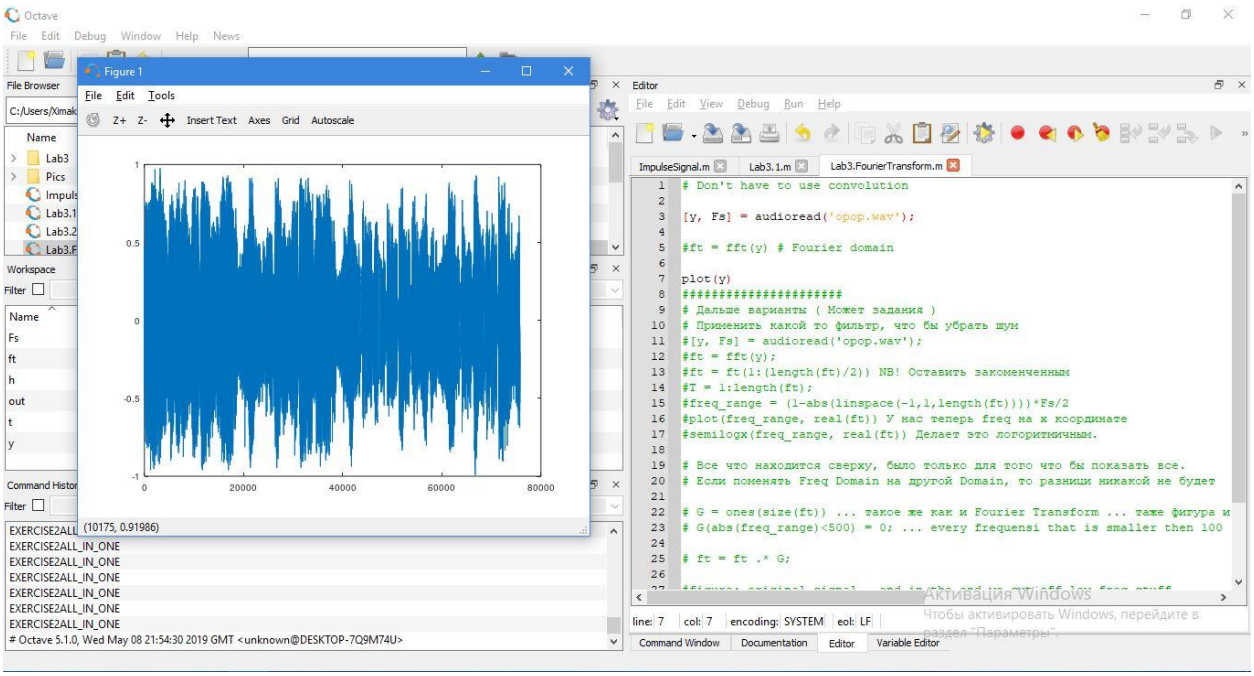


Fourier Transform Exercises

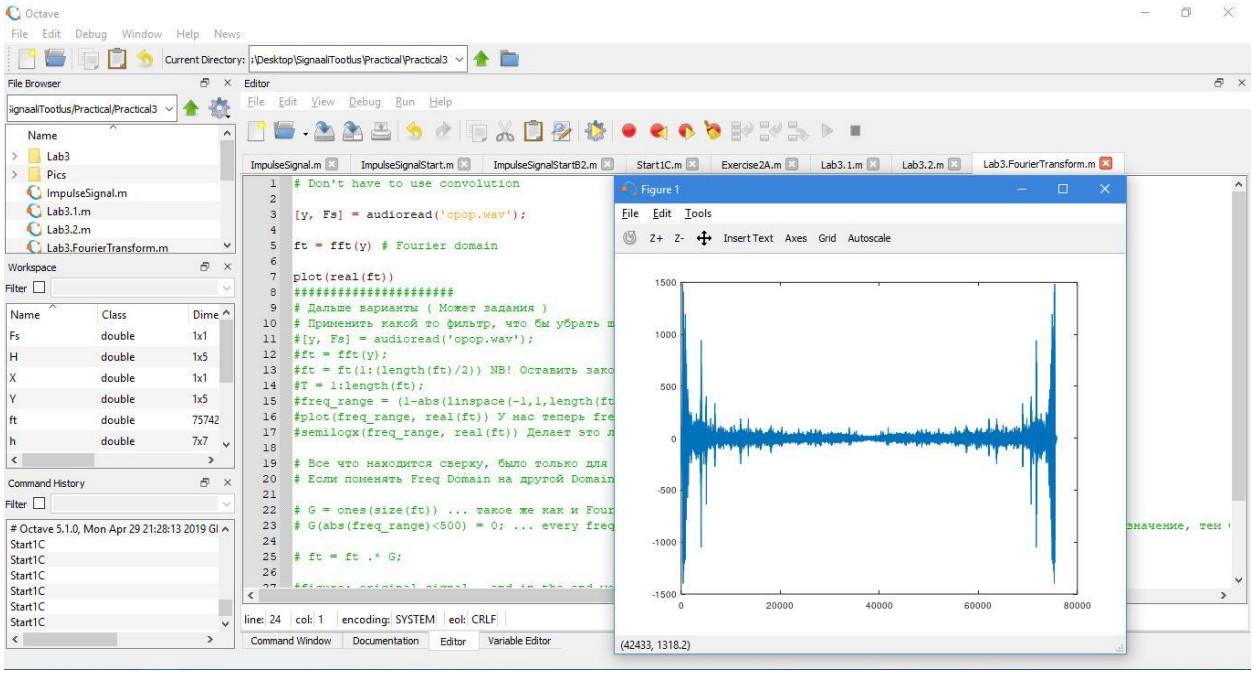
Exercise 1

Read the provided opop.wav file or a .wav file of your choice (preferably mono sound) and perform the following:

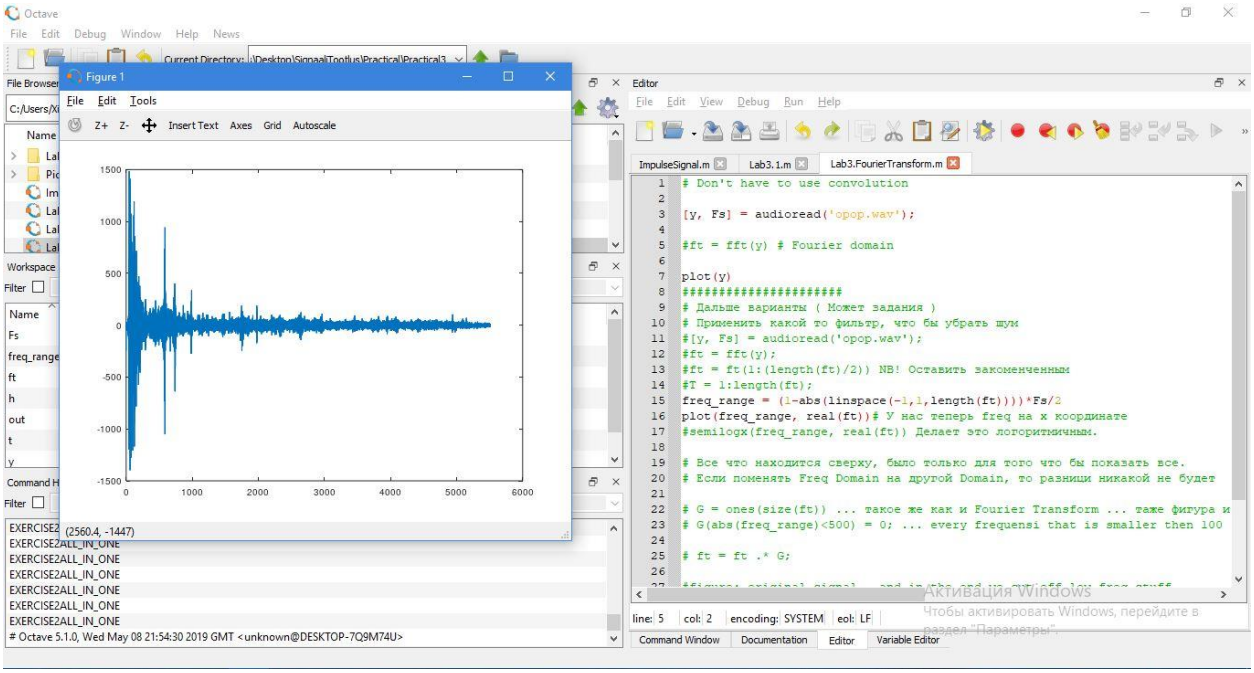
(a) Plot a section of the (time domain) audio data.



(b) Find the Fourier Transform of the audio data.

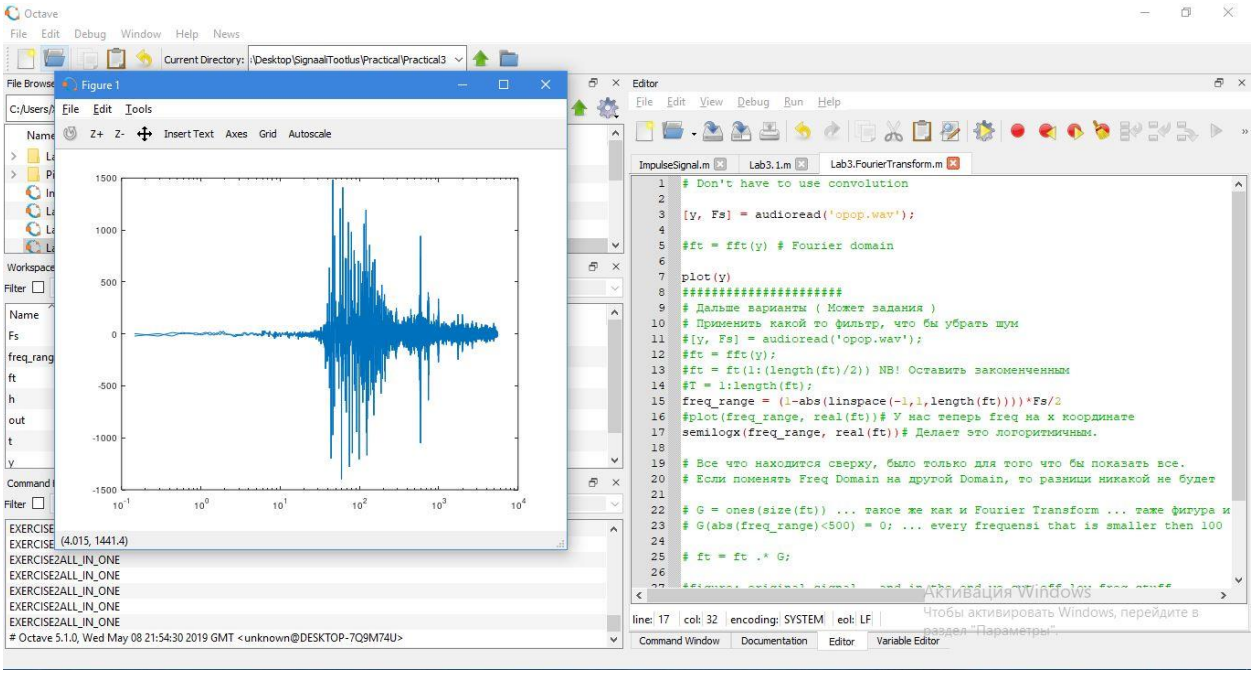


(b) Find the frequency range for the frequency domain signal.



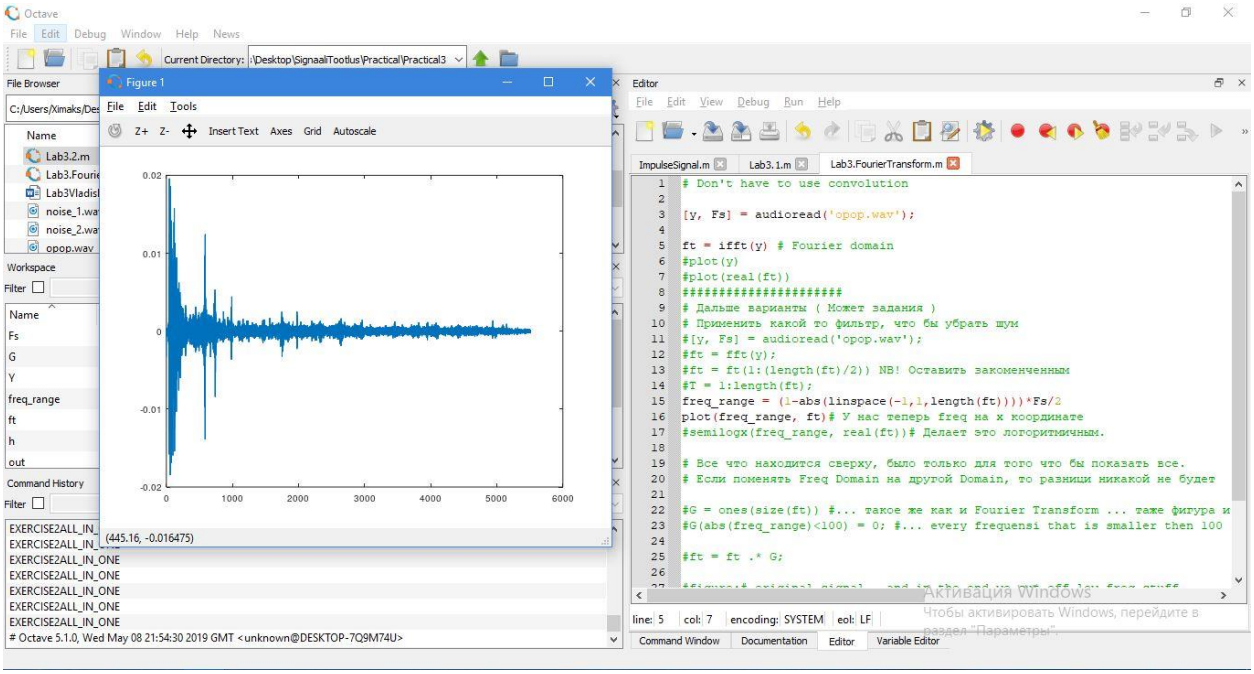
NB, Frequency range for the frequency domain signal is now on the x axis.

(c) Plot the frequency domain and set the frequency axis logarithmic.



Now the X-axis is logarithmic.

(d) Find the Inverse Fourier Transform of the frequency signal.



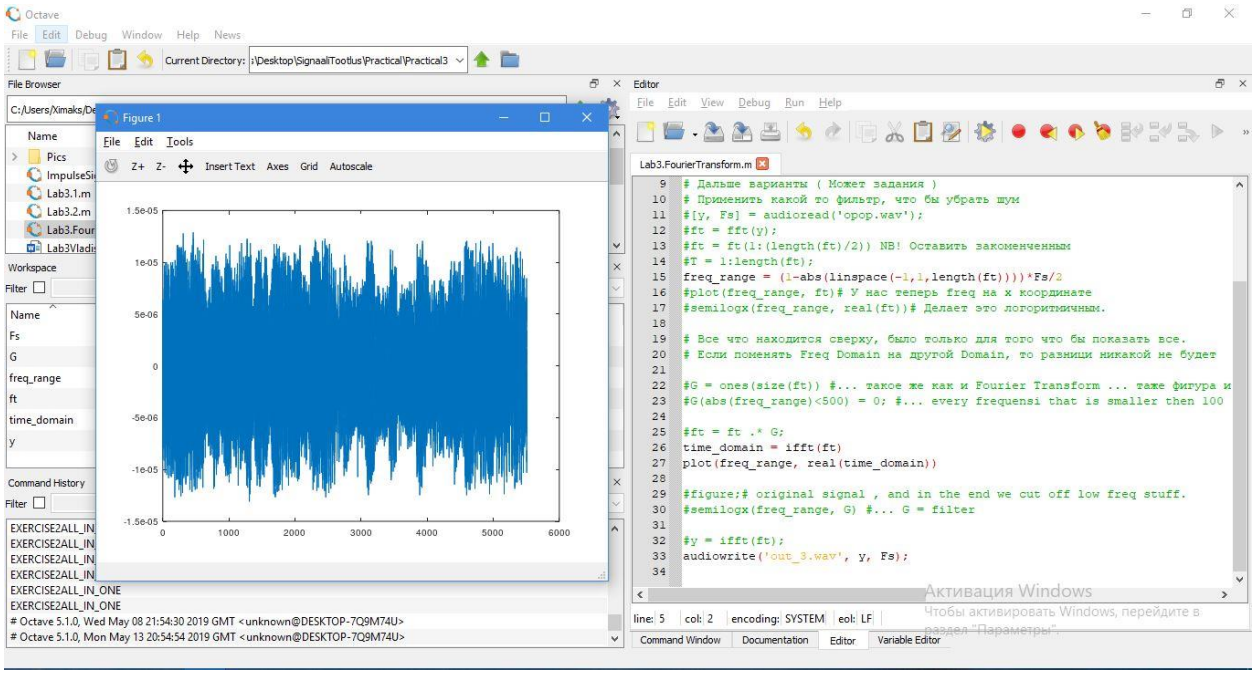
NB! Why I think, this is also “Inverse Fourier Transform” of the frequency signal…

- Ft = ifft(y) already saying, that it’s compute the inverse discrete Fourier Transform.

- Line 15: Changing the signal to frequency signal
  - freq_range = (1-abs(linspace(-1,1,length(ft))))*Fs/2
- Line 16: We are ploting the frequency range, and the ifft of the signal
  - plot(freq_range, ft)

That’s the reasons, left it unchanged.

(e) Plot the new time domain signal.



(g) Save the audio file and make sure it still works. (nothing should have happened)

Exercise 2

Perform the following functions on the signals frequency domain:

(a) Create an ideal (low-pass, high-pass OR band-pass) filter.

- G = ones(size(ft))
- G(abs(freq_range)<500) = 0;

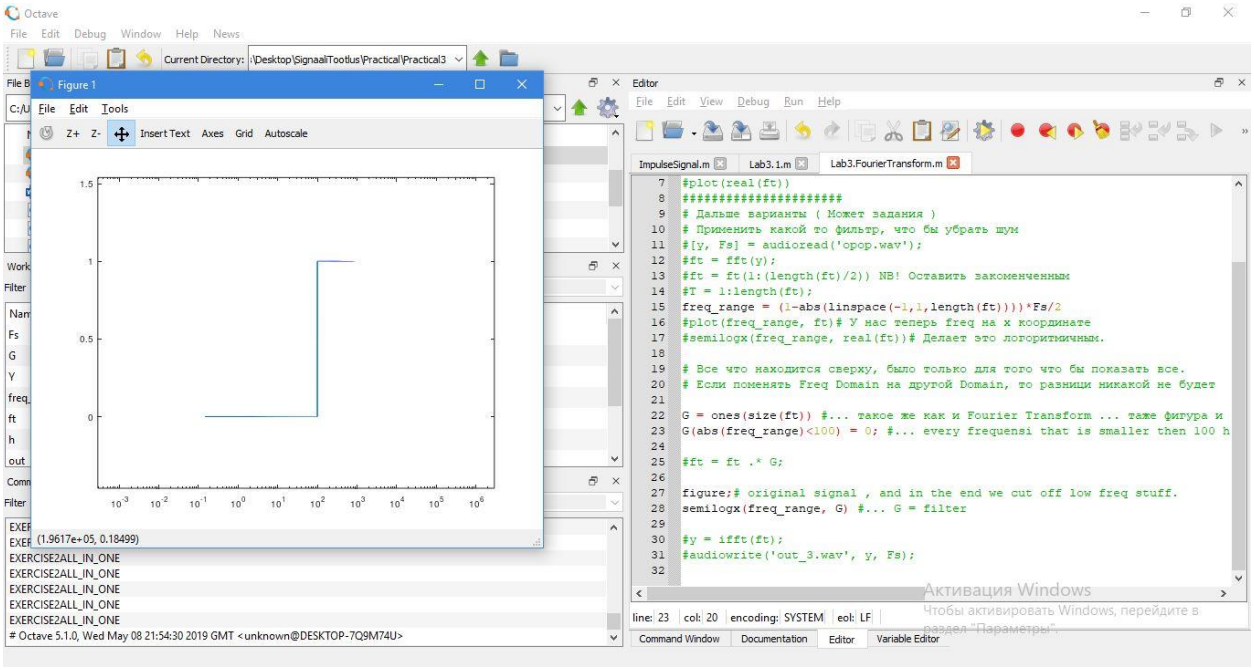
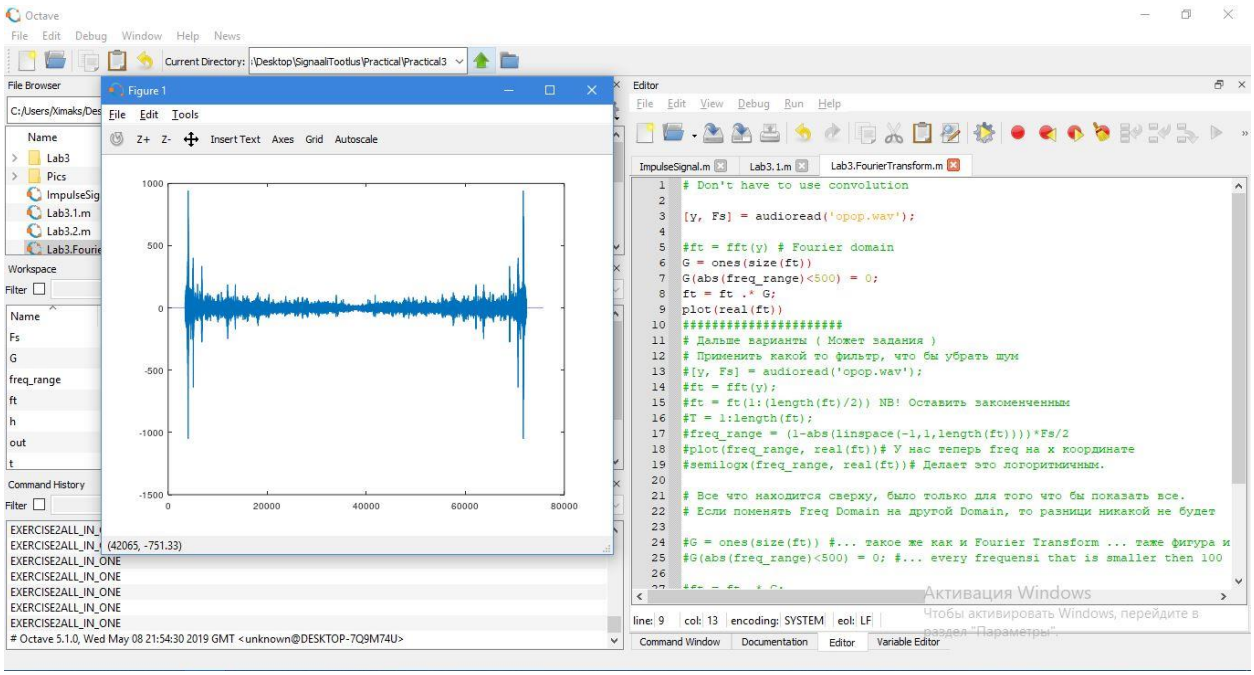
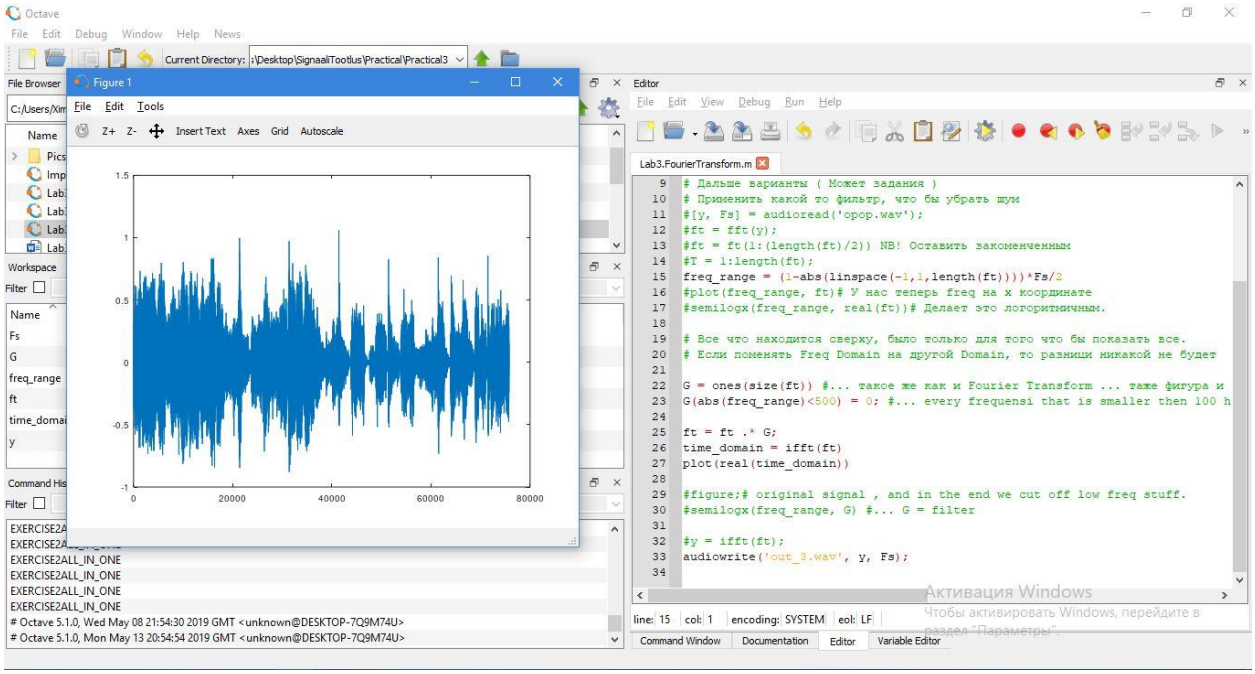


Figure 1 Low_Pass

(b) Apply element wise multiplication between the filter and frequency domain signal



(b) Convert the signal back to time domain, plot it and save it.



(d) Come up with a filter that removes a specific frequency range in the audio (e.g. the ”op op” part of the clip) and explain what it’s doing and how it’s doing it.