Санкт-Петербургский государственный политехнический
университет Петра Великого

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

Лабораторная работа

# Модуляция и выборка (квантование)

Работу выполнил студент
3-го курса, группа 3530901/80201
Сахибгареев Рамис Ринатович

Преподаватель:
Богач Наталья Владимировна

Санкт-Петербург 2021

# Contents

# List of Figures

# Listings

# 1 Part 1: Research and execution of chap11

In this part we need to research and execute existing chap11.ipynb file, that contains information about amplitude modulation, convolution with impulses, sampling and sinc interpolation.

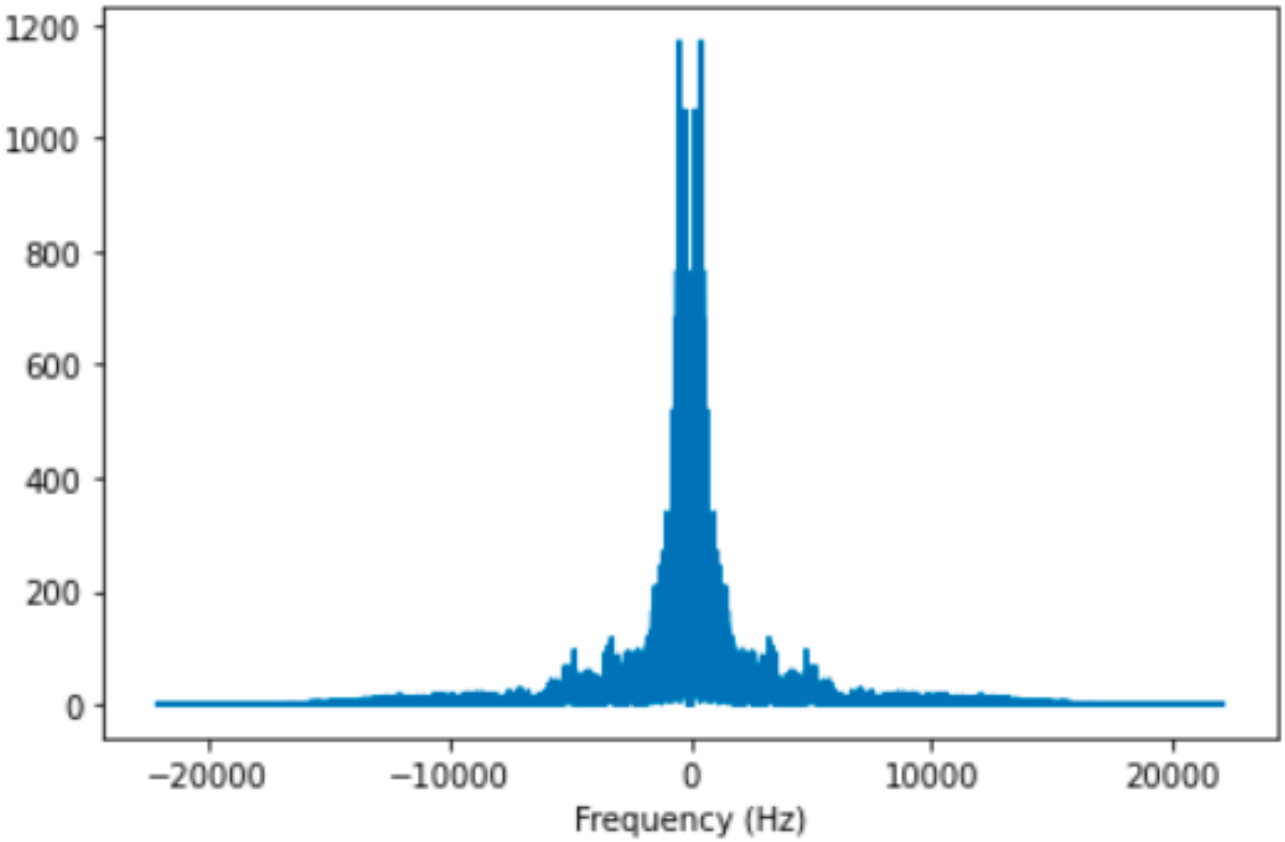


Figure 1: After padding

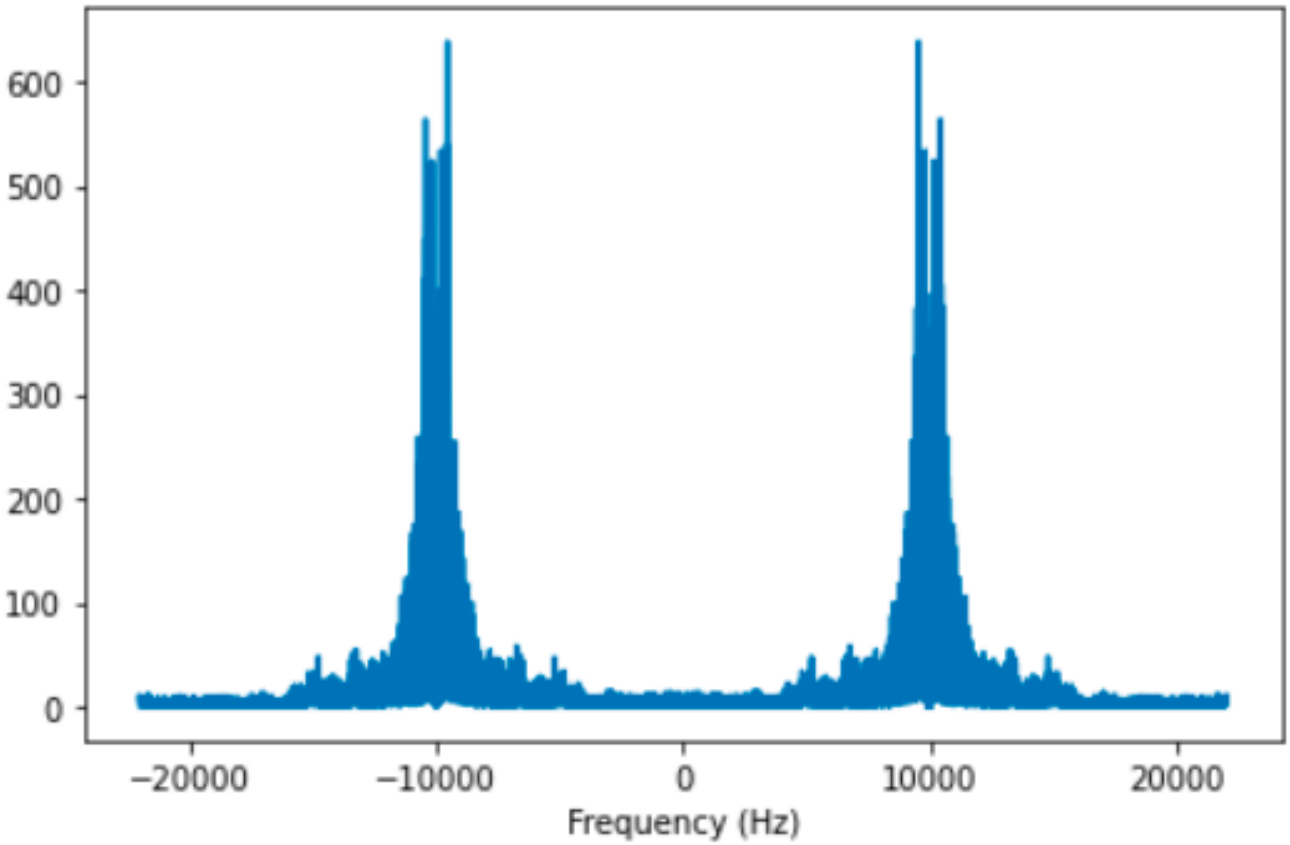


Figure 2: Without padding

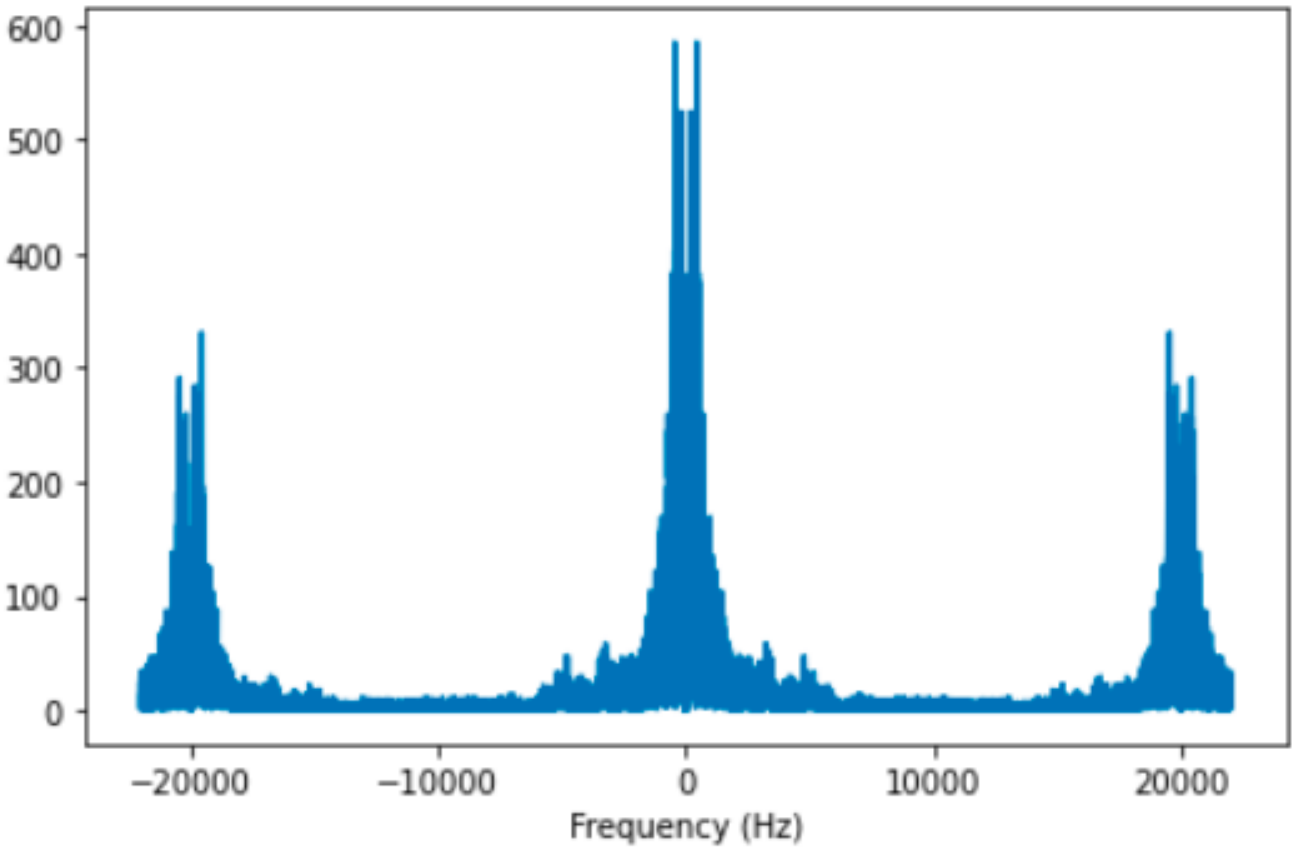


Figure 3: Trying to apply convolution theorem on non periodic signal

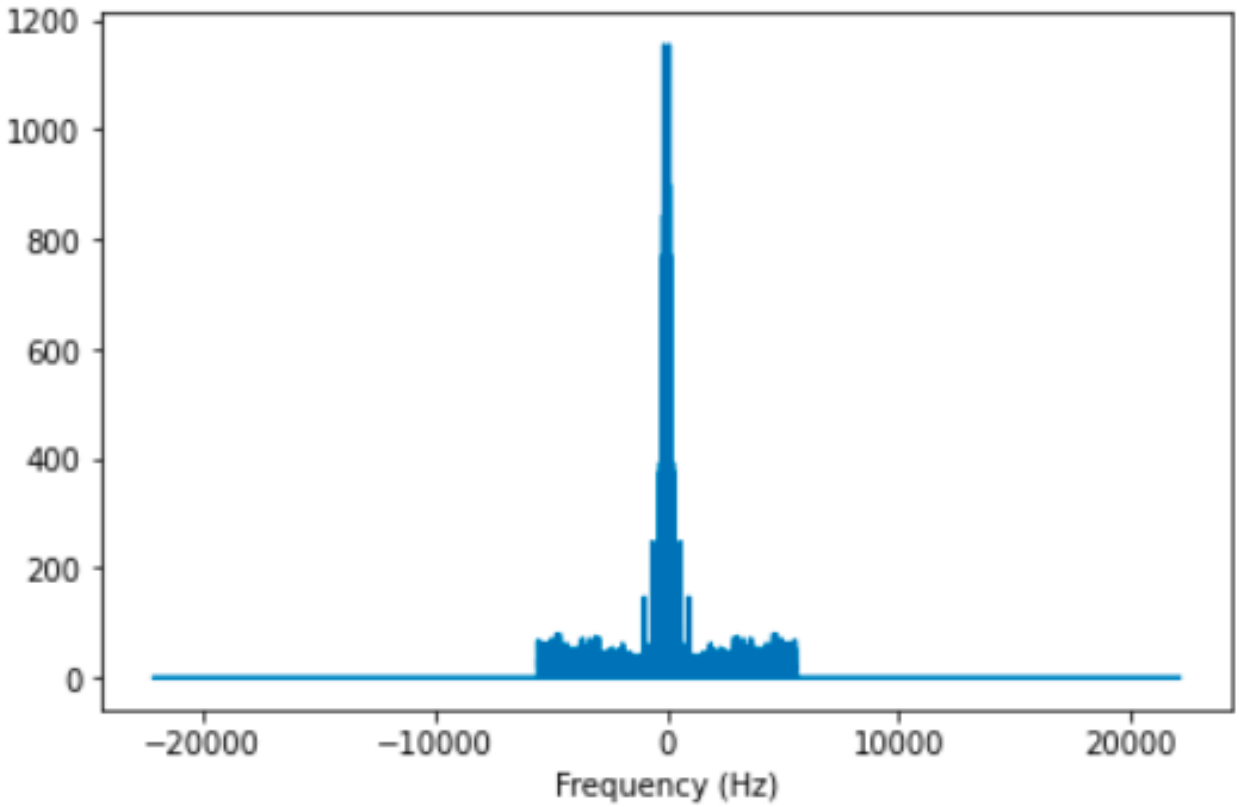


Figure 4: After padding

We can clearly see how aliasing effect appears while we are trying to use DFT with low frequency. It happens because of modulation, that splits the signal into the different parts around frequency of the signal. If this frequency is small, then resulting signals will overlap.

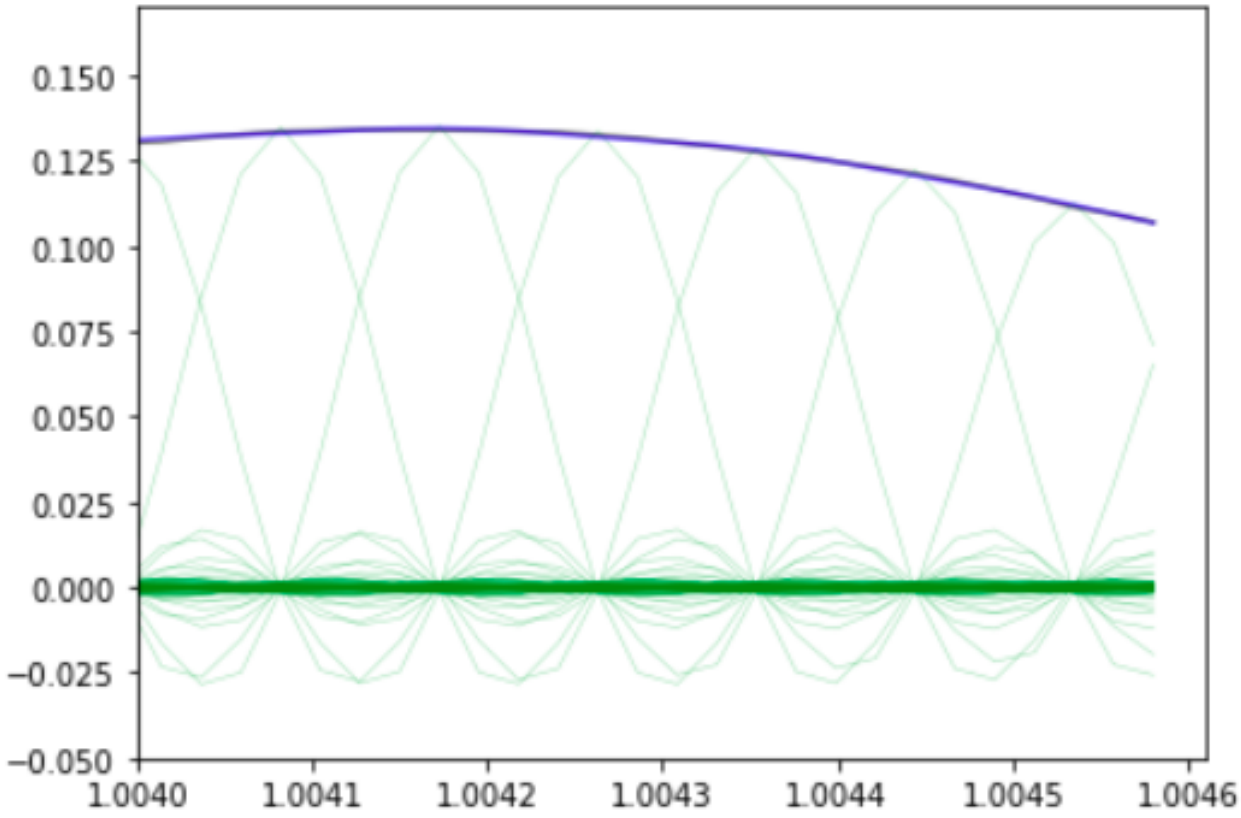


Figure 5: After padding

We can see, how box low pass filter spectrum looks. different sin signals are sums up resulting original wave.

# 2 Part 2: D/A and A/D | Digital Show and Tell

In this part we need to watch video about DC and AC transformations (https://www.youtub

After watching it we've learned, why do we use DFT, how transformation performed, why audio 44.1 KHz and

$$2^16$$

is enough for humans.

# 3 Part 3: Modulating the signal

In this part we need to try remove aliasing by applying low-pass filter before modulating and not after.

```
import numpy as np
import matplotlib as plt
from thinkdsp import *

wave = read_wave('263868__kevcio__amen-break-a-160-bpm.wav')
wave.normalize()
wave.make_spectrum(full=True).plot()
decorate(xlabel='Frequency (Hz)')

```

Listing 1: Reading the signal

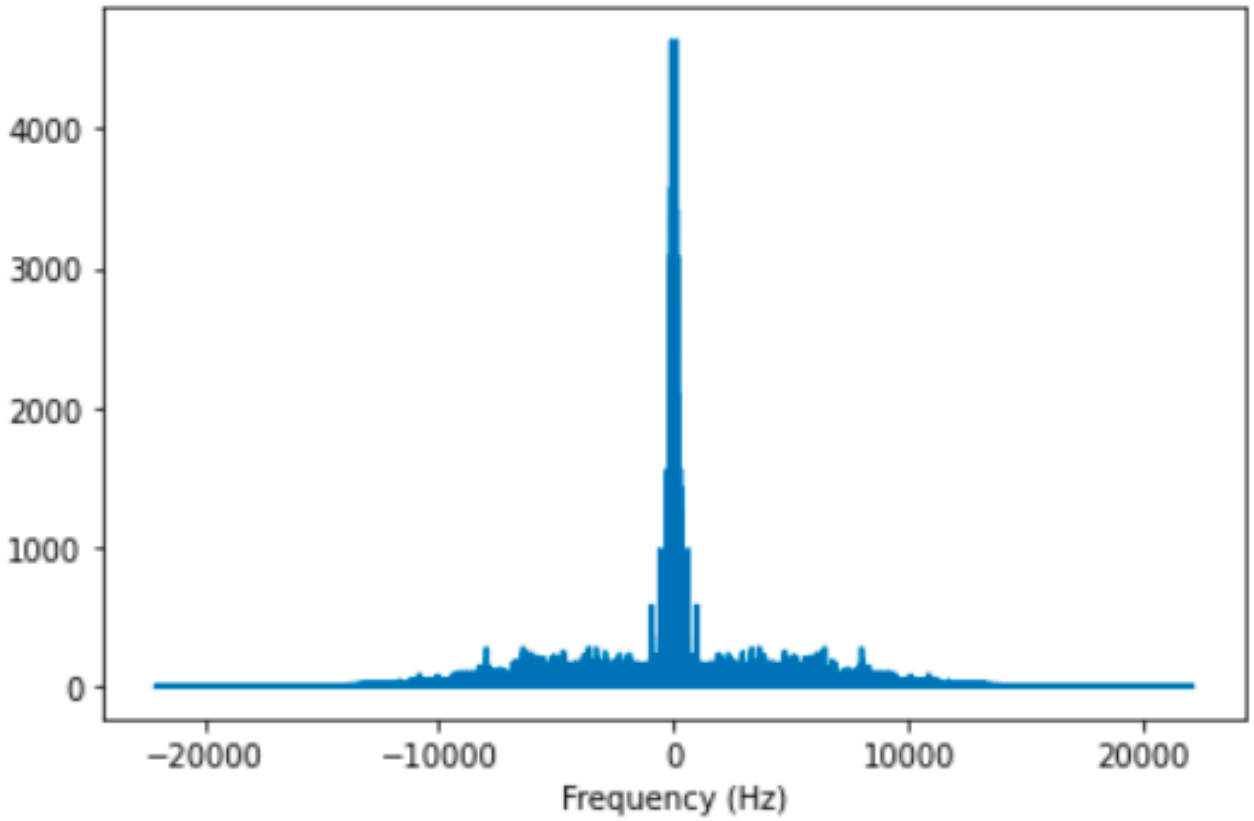


Figure 6: Impulse's wave

We can see, that signal has a lot of high frequency components, but they have low amplitude.

```
spectrum = wave.make_spectrum(full=True)
spectrum.low_pass(5000)
wave = spectrum.make_wave()
spectrum.plot()
decorate(xlabel='Frequency (Hz)')
```

Listing 2: Reading the impulse

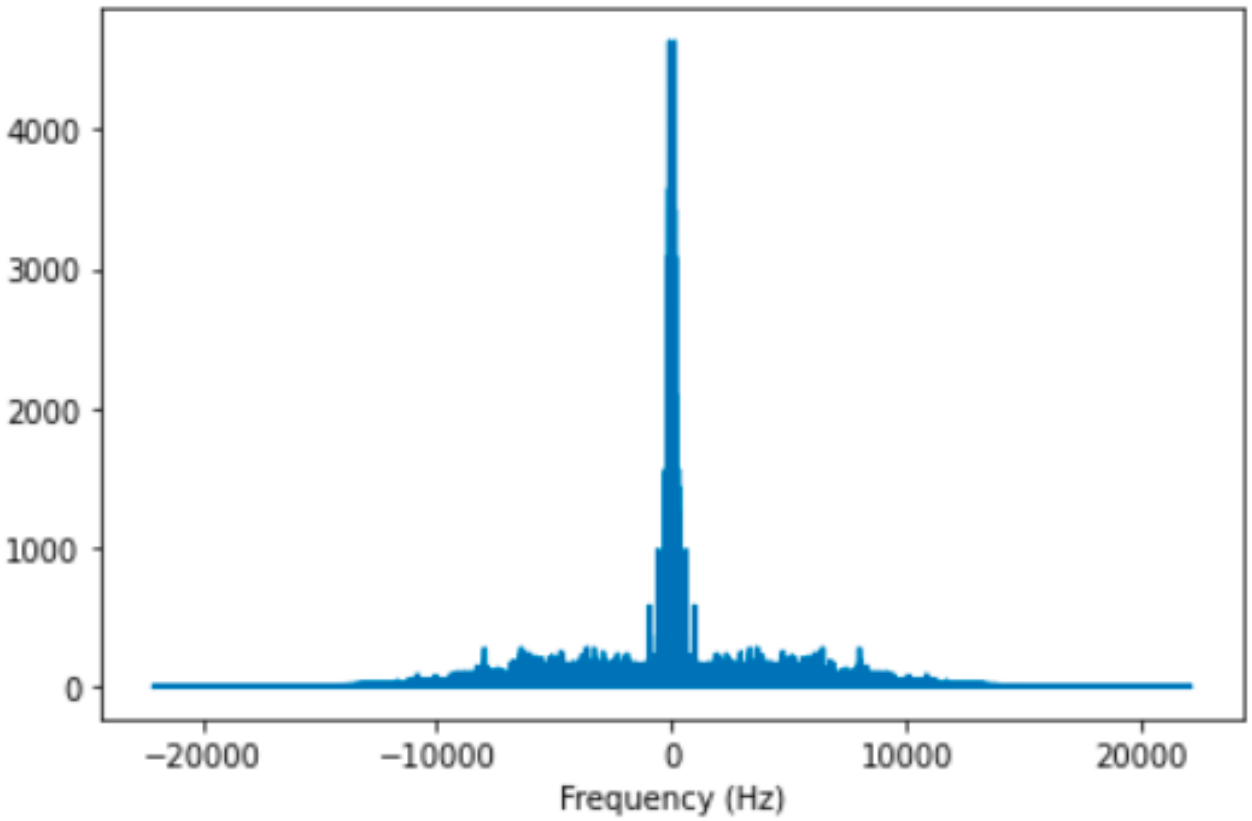


Figure 7: Impulse's wave

```
def sample(wave, factor):
    ys = np.zeros(len(wave))
    ys[::factor] = wave.ys[::factor]
    return Wave(ys, framerate=wave.framerate)
sampled = sample(wave, 4)
sampled.make_spectrum(full=True).plot()
decorate(xlabel='Frequency (Hz)')
```

Listing 3: Modulating the signal

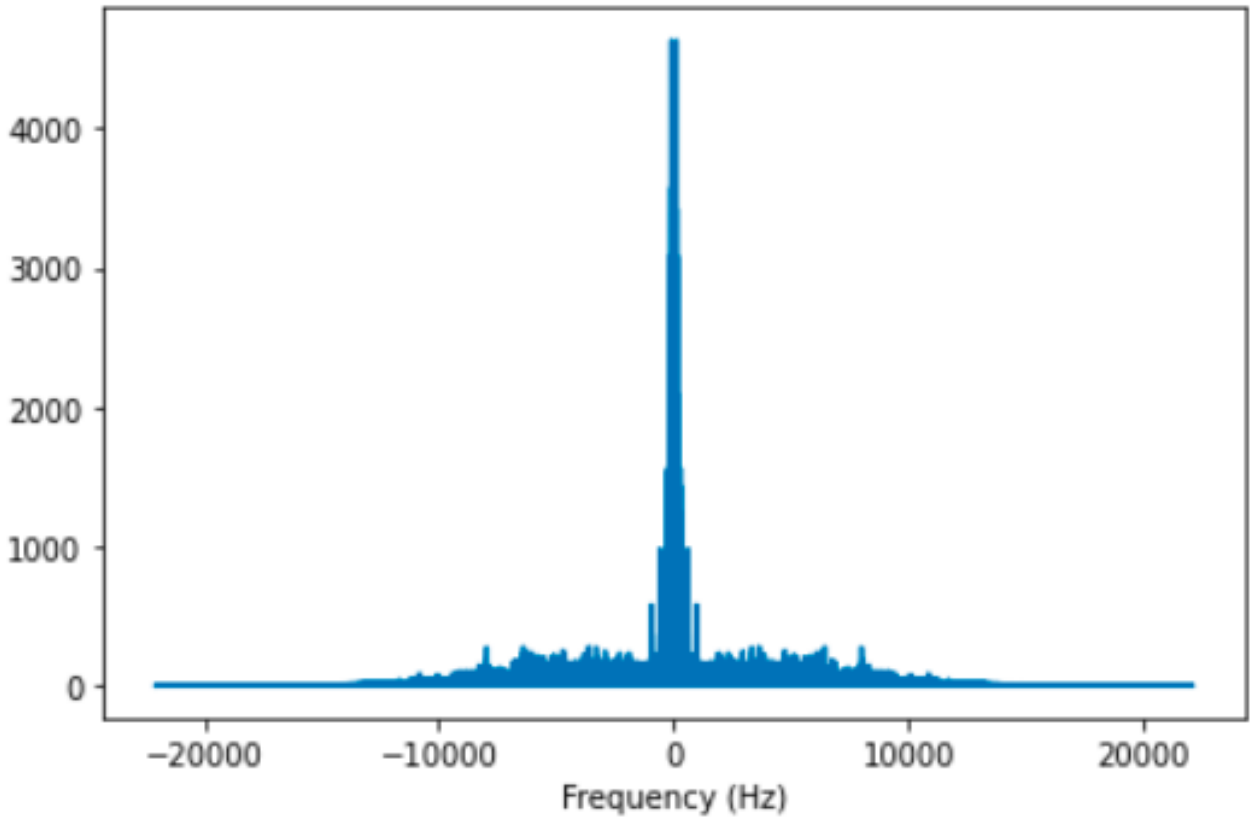


Figure 8: Impulse's wave

We can see, that no overlapping happened.

```
spectrum = sampled.make_spectrum(full=True)
spectrum.low_pass(5512.5)
sampled = spectrum.make_wave()
spectrum.plot()
decorate(xlabel='Frequency (Hz)')

```

Listing 4: Getting modulated

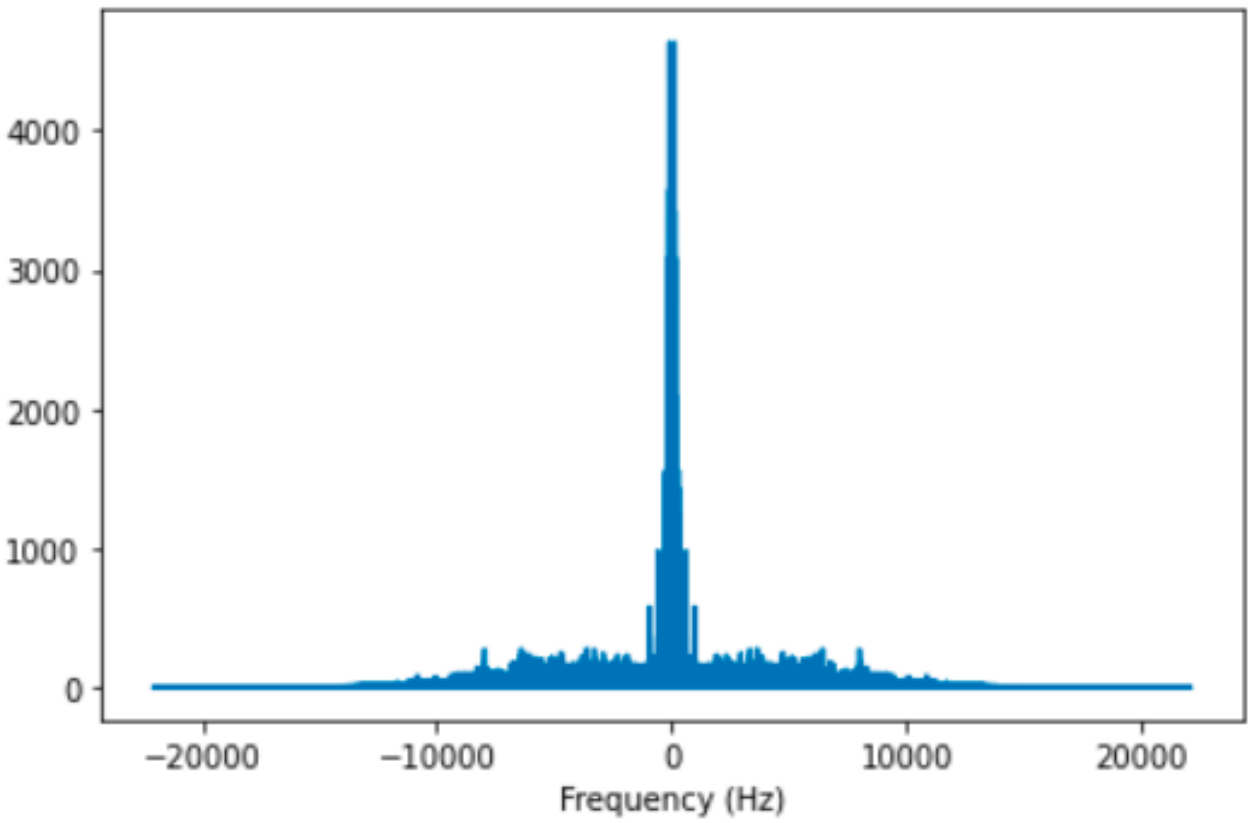


Figure 9: Impulse's wave

```
diff = wave.ys - sampled.ys
np.mean(abs(diff))
```

```
0.07478781153466153
```

Figure 10: Difference

We can see, that the original and modulated signal has little difference.

# 4 Conclusion

We’ve learned, how amplitude modulation, convolution with impulses, samplingand sinc interpolation done and how we can use it for data transport. Also we’ve watched an video, that tells how and why DC and AC transformations performed. Finally we’ve tested a way how we can remove aliasing effect by applying a low-pass before modulation, not after.