Санкт-Петербургский государственный политехнический
университет Петра Великого

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

Лабораторная работа

Линейные стационарные системы

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 chap010

In this part we need to research and execute existing chap10.ipynb file, that contains information about stationary systems, impulse response, and convolution theorem. Also we need to perform padding of an violin record with zeros to remove folding effect.

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

response_base = read_wave('180960__kleeb__gunshot.wav')

start = 0.12
response = response_base.segment(start=start)
response.shift(-start)

response.zero_pad(2**17)

response.normalize()
response.plot()
decorate(xlabel='Time (s)')

response1 = response_base.segment(start=start, duration=1)
response1.shift(-start)

response1.normalize()

violin = read_wave('92002__jcveliz__violin-origional.wav')

start = 0.11
violin2 = violin.segment(start=start, duration=2)
violin2.shift(-start)

violin2.truncate(len(response))
violin2.zero_pad(len(response))

violin2.normalize()
violin2.plot()
decorate(xlabel='Time (s)')

start = 0.11
violin = violin.segment(start=start, duration=2)
violin.shift(-start)
violin.truncate(len(response1))
violin.normalize()

transfer = response.make_spectrum()
transfer1 = response1.make_spectrum()
spec1 = violin.make_spectrum()
spec2 = violin2.make_spectrum()
print(len(spec1), len(spec2), len(transfer), len(transfer1))
out1 = (spec1 * transfer1).make_wave()
out2 = (spec2 * transfer).make_wave()
out1.normalize()
out2.normalize()

```

Listing 1: Padding compare code

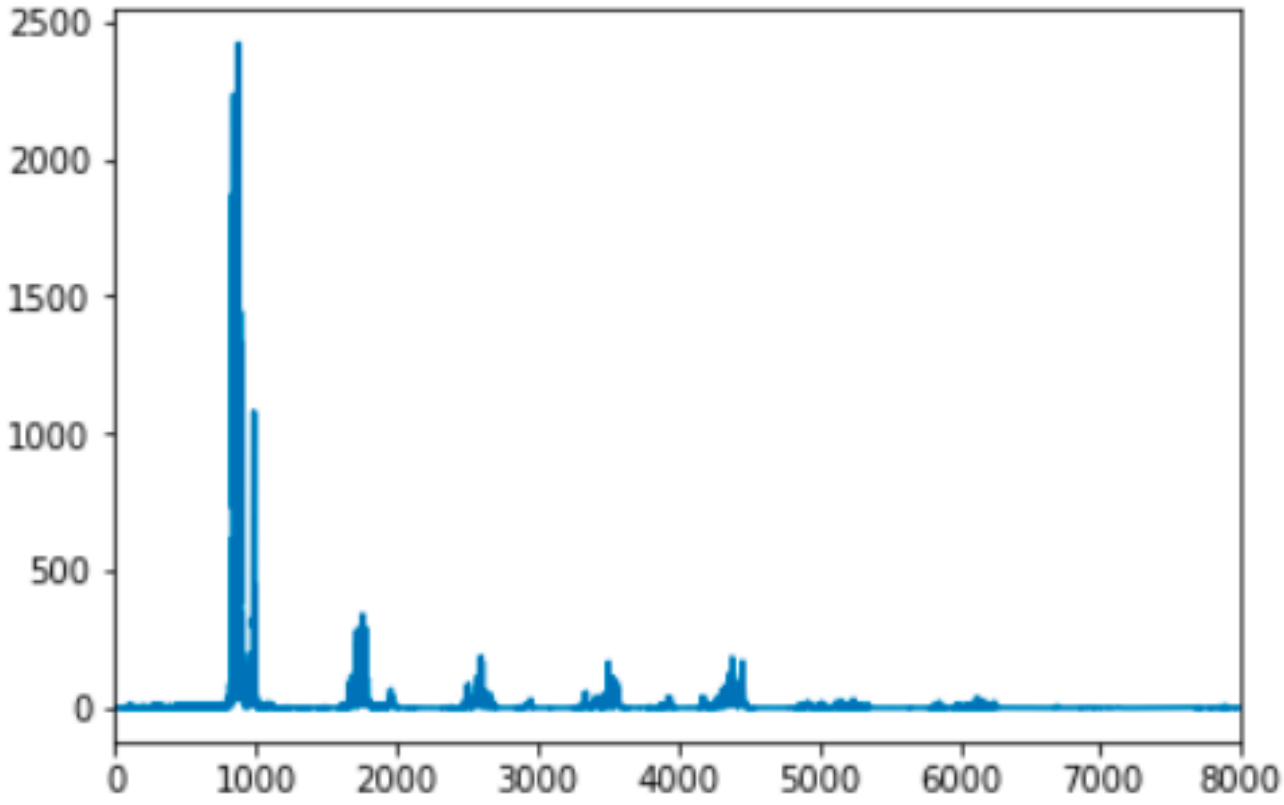

Figure 1: After padding

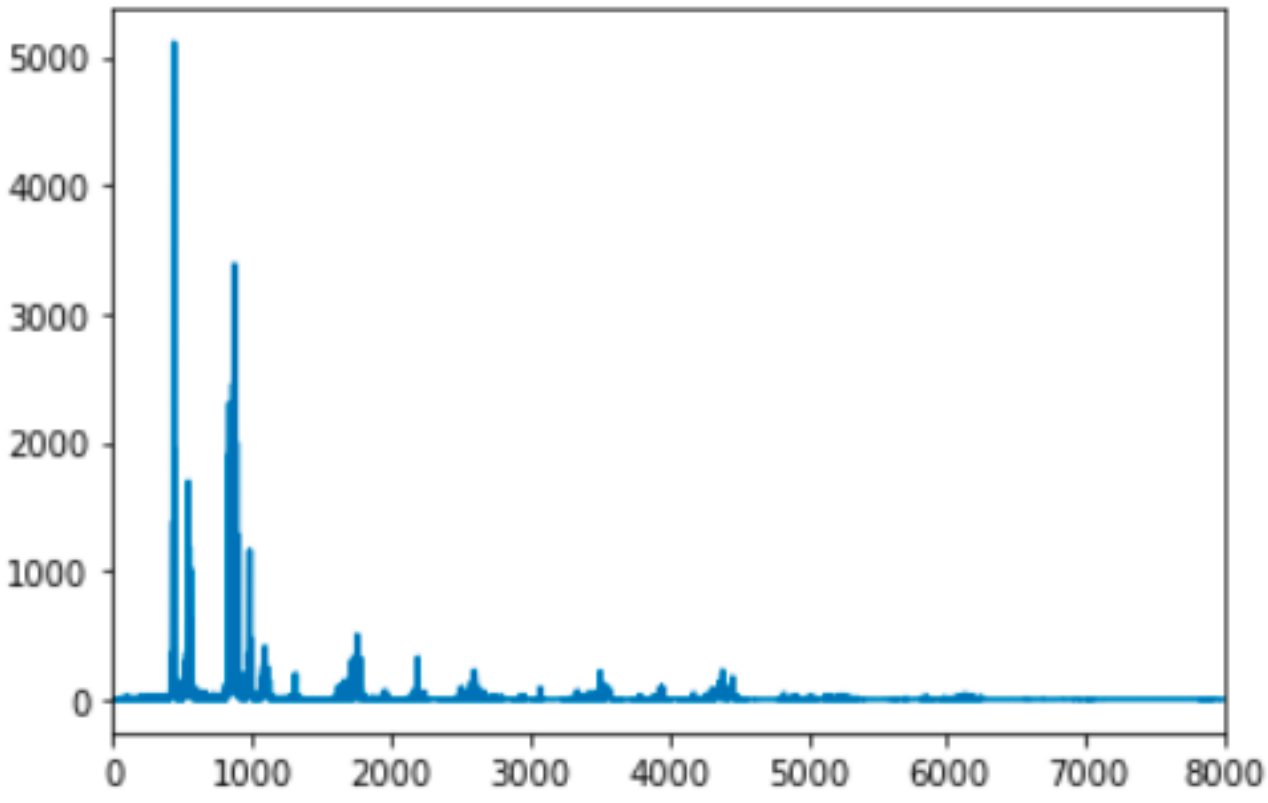

Figure 2: Without padding

We can clearly see, that 500 Hz peak disappeared on the record with padding.

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

2 Part 2: Applying impulse to the signal

In this part we need to try to apply an impulse to the signal by ourselves. For the impulse balloon pop sound was used. Let's firstly read it and display its wave. I've extended it with zeros to make folding effect not to appear.

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

response = read_wave('balloon.wav')

start = 0
duration = 1
response = response.segment(duration=duration)
response.shift(-start)

response.normalize()
response.unbias()
response.zero_pad(2**17)
response.plot()
decorate(xlabel='Time (s)')

transfer = response.make_spectrum()
transfer.plot()
decorate(xlabel='Frequency (Hz)', ylabel='Amplitude')

```

Listing 2: Reading the impulse

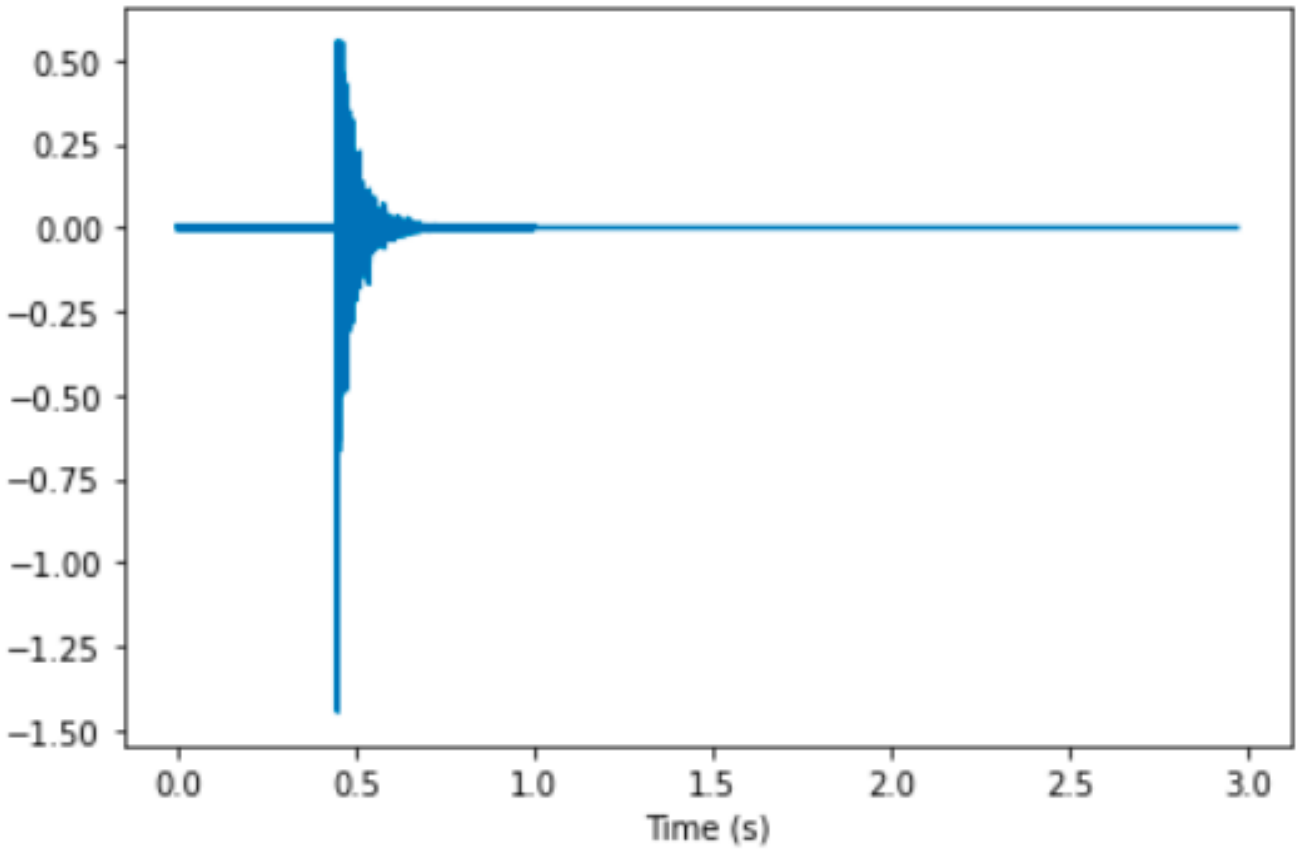

Figure 4: Impulse's wave

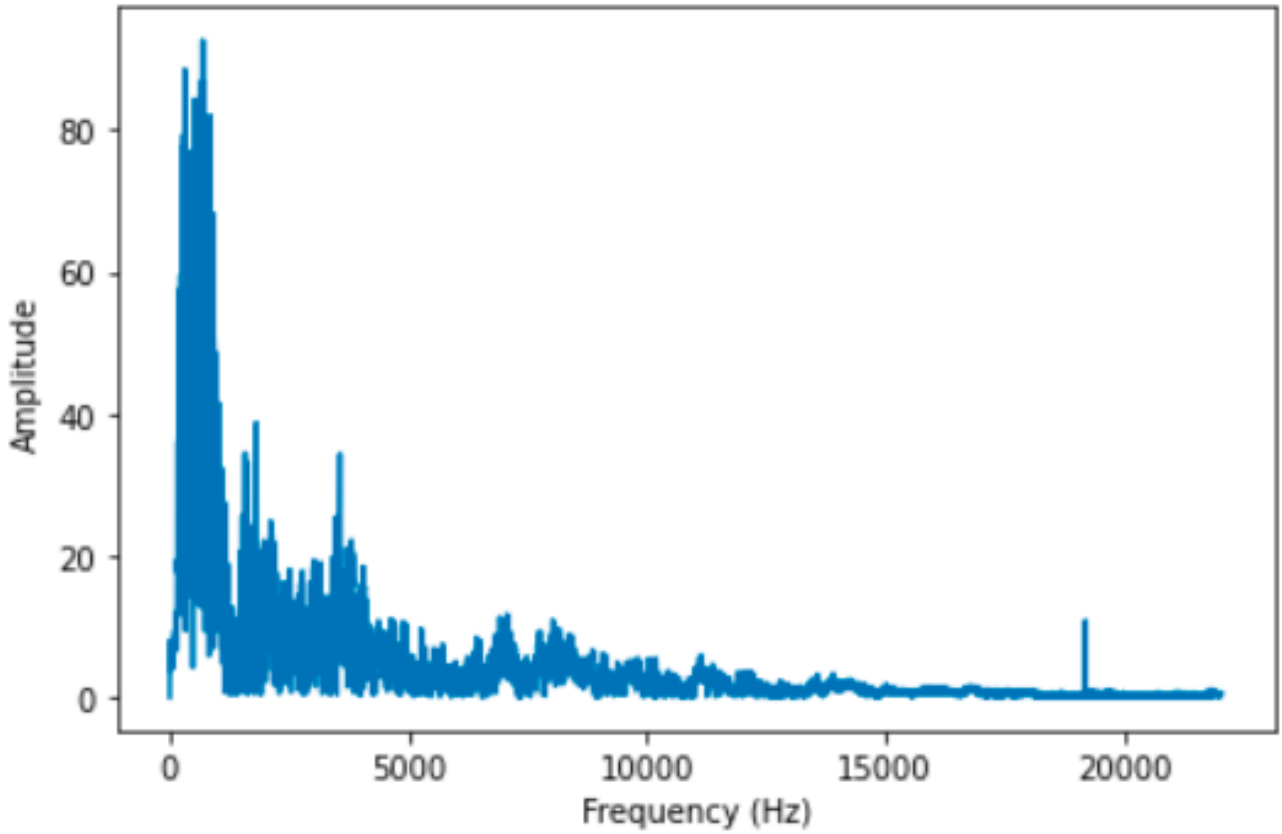

Figure 5: Impulse's spectrum

Next, let's read the record, that will be affected with the impulse. It is also extended with zeros.

```
wave = read_wave('violin.wav')

start = 0.12
wave = wave.segment(start=start, duration=duration * 1.5)
wave.shift(-start)

wave.truncate(len(response))
wave.zero_pad(len(response))
wave.normalize()
wave.plot()
decorate(xlabel='Time (s)')

```

Listing 3: Reading a violin sound

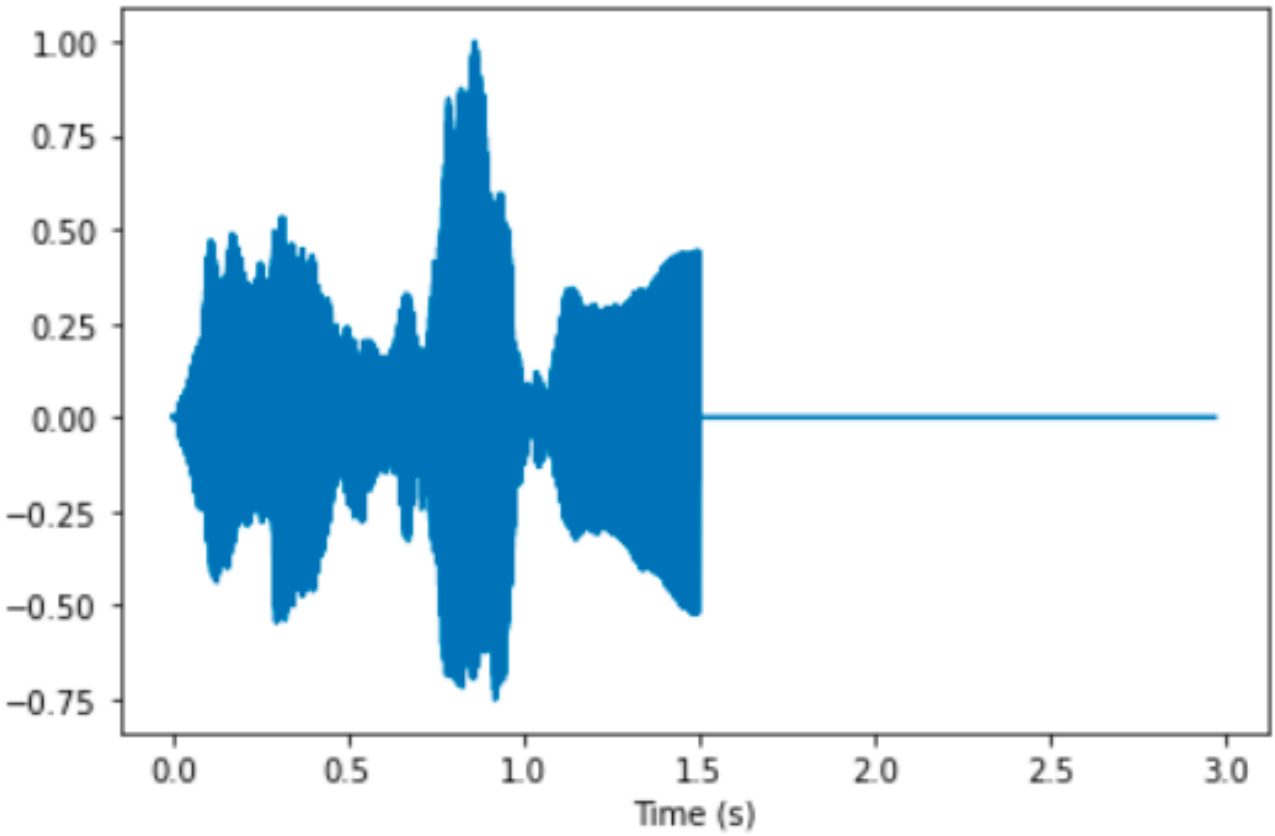

Figure 6: Violin's wave

Now we can apply the impulse to the violin sound using DFT multiplication.

```
output = (wave.make_spectrum() * transfer).make_wave()
output.normalize()
output.plot()

```

Listing 4: Applying the impulse

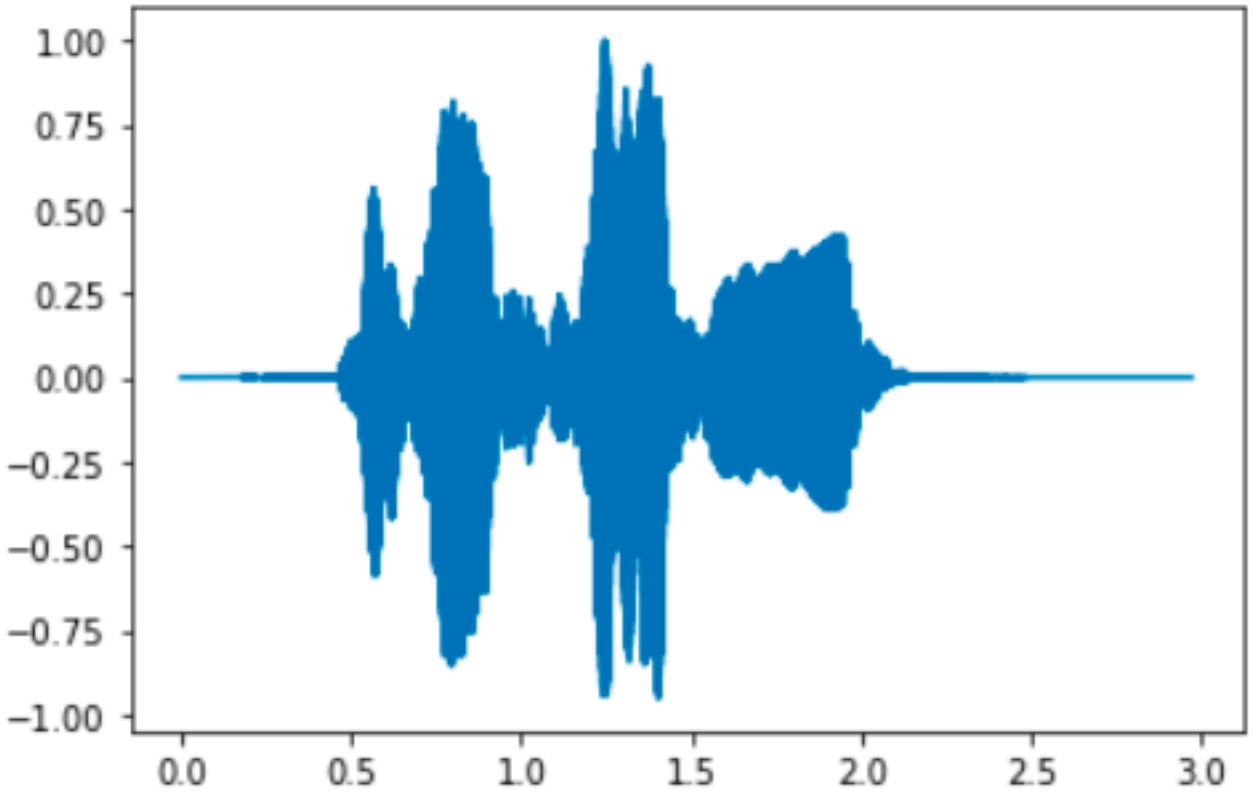

Figure 7: Impulse applied

Now we can listen the waves and compare them. Also let's compare DFT multiplication to the convolve function.

```
In [40]: output.make_audio()
```

Out[40]:


```
In [41]: conv = wave.convolve(response)
         conv.normalize()
         conv.make_audio()
```

Out[41]:


```
In [42]: wave.make_audio()
```

Out[42]:

Figure 8: Comparing results

After listening, no differences between DFT multiplication and convolve method have been found. Also modified sound is more muted, than the original one.

3 Conclusion

We've learned, how we can convert any signal to the impulse and use it as a filter using impulse response. Also we can apply signal using DFT multiplication and convolution. However, we can face the folding effect, but we can easily remove it by extending the signal with zeros. It happens, because real signals are non-periodic, while we are trying to make them periodic.