

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  
Солянкин Илья Андреевич

Преподаватель:  
Богач Наталья Владимировна

Санкт-Петербург 2021

# Содержание

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Часть №1: chap11.ipynb</b>   | <b>5</b>  |
| <b>2</b> | <b>Часть №2: Крис "Монти"Монтгомери - "D/A and A/D   Digital Show and Tell"</b> | <b>6</b>  |
| <b>3</b> | <b>Часть №3: "Соло на барабане"и фильтр НЧ</b>                                  | <b>7</b>  |
| <b>4</b> | <b>Выводы</b>   | <b>13</b> |

## Список иллюстраций

|    |   |    |
|----|---|----|
| 1  | Результат запуска блокнота chap11.ipynb . . . . .     | 5  |
| 2  | Полученный сигнал . . . . .                           | 7  |
| 3  | Перевод полученного сигнала в аудио . . . . .         | 7  |
| 4  | Полученный спектра . . . . .                          | 8  |
| 5  | Применение фильтра . . . . .                          | 8  |
| 6  | Применение фильтра и результат . . . . .              | 9  |
| 7  | Проверка функции sample . . . . .                     | 9  |
| 8  | Полученный спектр . . . . .                           | 10 |
| 9  | Примененный фильтр сглаживания . . . . .              | 10 |
| 10 | Масштабирование спектра . . . . .                     | 11 |
| 11 | Сравнение спектров и преобразование в волну . . . . . | 11 |
| 12 | Результат . . . . .                                   | 12 |

## Листинги

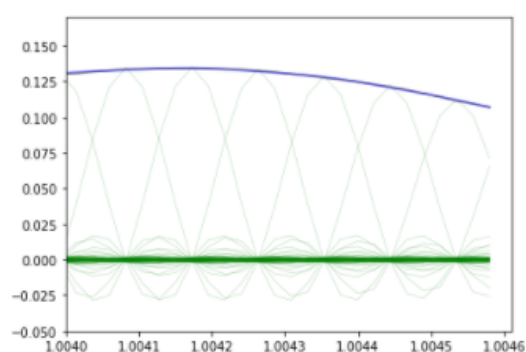
|   |   |    |
|---|---|----|
| 1 | Получение сигнала . . . . .                         | 7  |
| 2 | Получение спектра . . . . .                         | 7  |
| 3 | Уменьшение частоты дискретизации в 3 раза . . . . . | 8  |
| 4 | Применение фильтра . . . . .                        | 8  |
| 5 | Функция <code>sample</code> . . . . .               | 9  |
| 6 | Создание спектра сигнала . . . . .                  | 9  |
| 7 | Применение фильтра сглаживания . . . . .            | 10 |
| 8 | Масштабирование спектра . . . . .                   | 10 |

# 1 Часть №1: chap11.ipynb

В первой части одиннадцатой лабораторной работы нам необходимо изучить и запустить весь код из блокнота chap11.ipynb.

В результате выполнения данного пункта можно сказать, что все листинги из блокнота успешно запускаются:

```
[60] start = short.start + 0.004
      duration = 0.00061
      plot_sinc_demo(short, 4, start, duration)
      decorate(xlim=[start, start+duration],
               ylim=[-0.05, 0.17])
```



The vertical gray lines are the samples.

Again, the thin green lines are the shifted, scaled copies of the sinc function.

In this segment, the interpolation matches the original wave very well.

Notice that each sinc function has a peak at the location of one sample and the value 0 at every other sample location. That way, when we add them up, the sum passes through each of the samples.

Рис. 1: Результат запуска блокнота chap11.ipynb

## **2 Часть №2: Крис "Монти"Монтгомери - "D/A and A/D | Digital Show and Tell"**

Во втором пункте одиннадцатой лабораторной работы нам необходимо посмотреть ролик Криса "Монти"Монтгомери - "D/A and A/D | Digital Show and Tell".

В результате, после просмотра данного видеоролика было выяснено, почему аналоговое аудио в приемлемых пределах человеческого слуха может воспроизводиться с идеальной точностью с использованием 16-битного цифрового сигнала 44.1 кГц.

### 3 Часть №3: ”Соло на барабанах”и фильтр НЧ

В третьем пункте одиннадцатой лабораторной работы нам необходимо применить фильтр НЧ к примеру ”Соло на барабанах”до выборки, после чего снова с помощью фильтра НЧ удалить спектральные копии, которые вызваны выборкой.

Для начала получим нужный нам сигнал:

```
1 wave = read_wave('263868__kevcio__amen-break-a-160-bpm.wav')
2 wave.normalize()
3 wave.plot()
```

Листинг 1: Получение сигнала

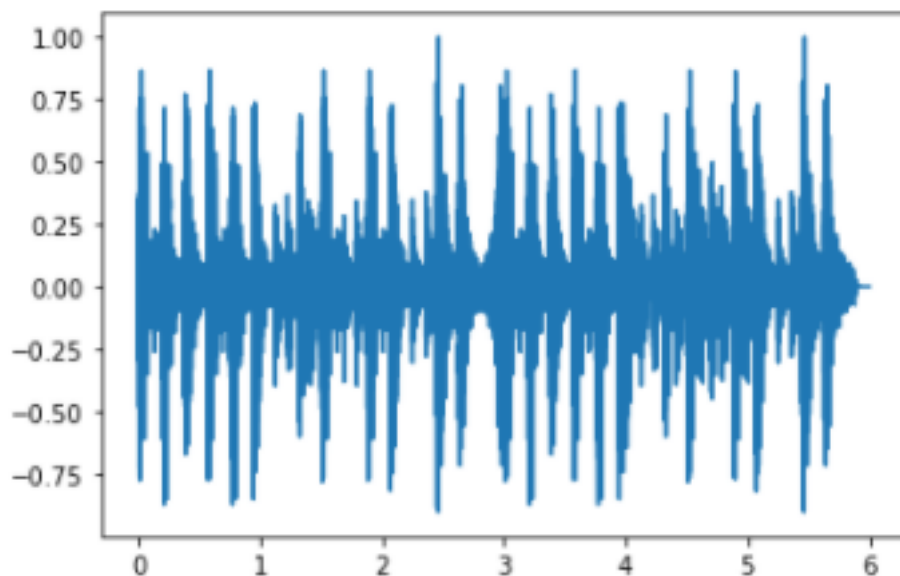


Рис. 2: Полученный сигнал

Сразу после этого переведем его в аудио:

```
In [3]: wave.make_audio()
```

Out[3]:

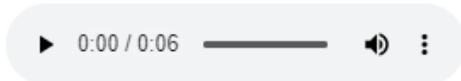


Рис. 3: Перевод полученного сигнала в аудио

Теперь получим спектр данного сигнала:

```
1 spectrum = wave.make_spectrum(full=True)
2 spectrum.plot()
```

Листинг 2: Получение спектра

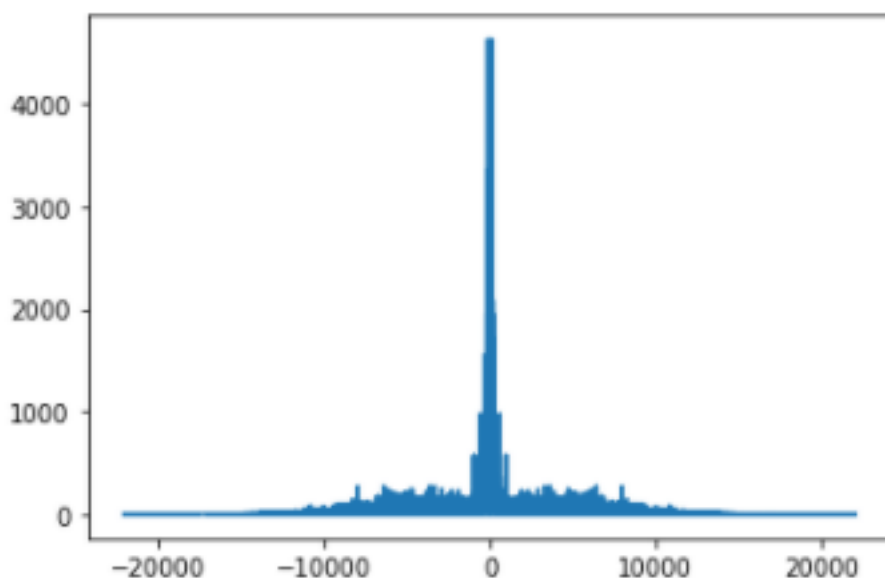


Рис. 4: Полученный спектра

Затем уменьшим частоту дискретизации в 3 раза:

```

1 factor = 3
2 framerate = wave.framerate / factor
3 cutoff = framerate / 2 - 1

```

Листинг 3: Уменьшение частоты дискретизации в 3 раза

После этого нужно применить фильтр сглаживания для удаления частот выше новой частоты свертки:

```

1 spectrum.low_pass(cutoff)
2 spectrum.plot()

```

Листинг 4: Применение фильтра

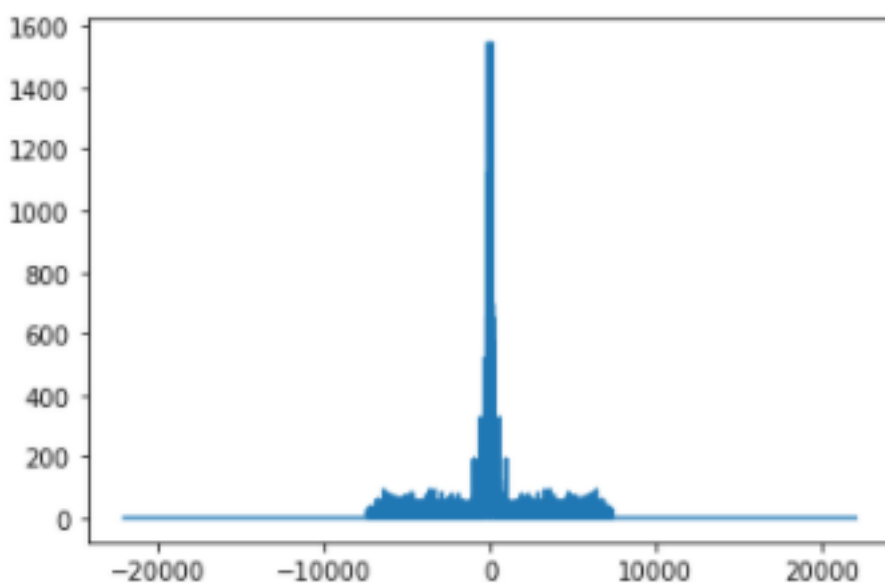


Рис. 5: Применение фильтра



```
B [7]: filtered = spectrum.make_wave()
      filtered.make_audio()

Out[7]:
```




Рис. 6: Применение фильтра и результат

Полученный сигнал все еще похож на исходный сигнал.

Теперь напишем функцию `sample`, которая будет имитировать процесс выборки:

```
1 def sample(wave, factor):
2     ys = np.zeros(len(wave))
3     ys[::factor] = np.real(wave.ys[::factor])
4     return Wave(ys, framerate=wave.framerate)
```

Листинг 5: Функция `sample`

Сразу же проверим нашу функцию:

```
B [9]: sampled = sample(filtered, factor)
      sampled.make_audio()

Out[9]:
```




Рис. 7: Проверка функции `sample`

Полученный сигнал после вызова функции должен содержать слабозаметные спектральные копии, которые можно увидеть, создав спектр данного сигнала:

```
1 sampled_spectrum = sampled.make_spectrum(full=True)
2 sampled_spectrum.plot()
```

Листинг 6: Создание спектра сигнала

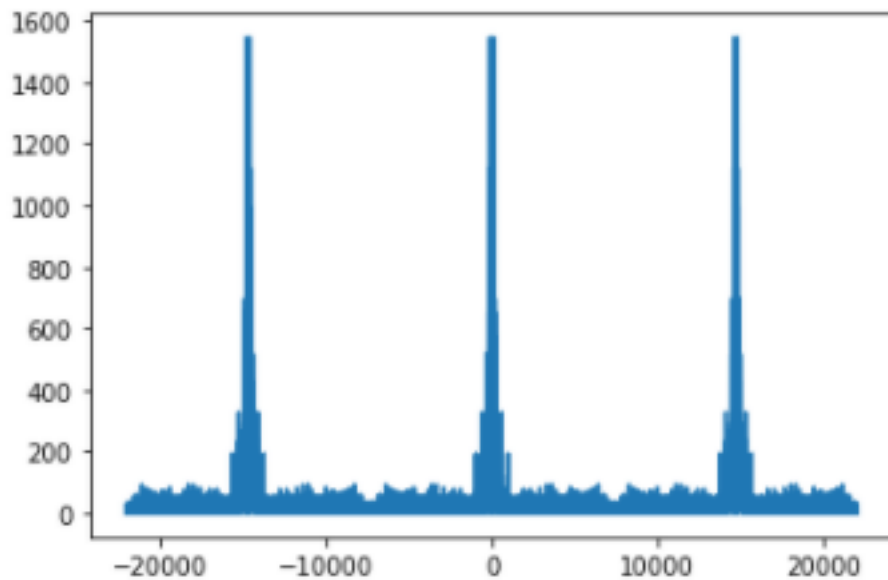


Рис. 8: Полученный спектр

На спектре можно увидеть спектральные копии. Чтобы их убрать нам необходимо применить фильтр сглаживания:

```
1 sampled_spectrum.low_pass(cutoff)
2 sampled_spectrum.plot()
```

Листинг 7: Применение фильтра сглаживания

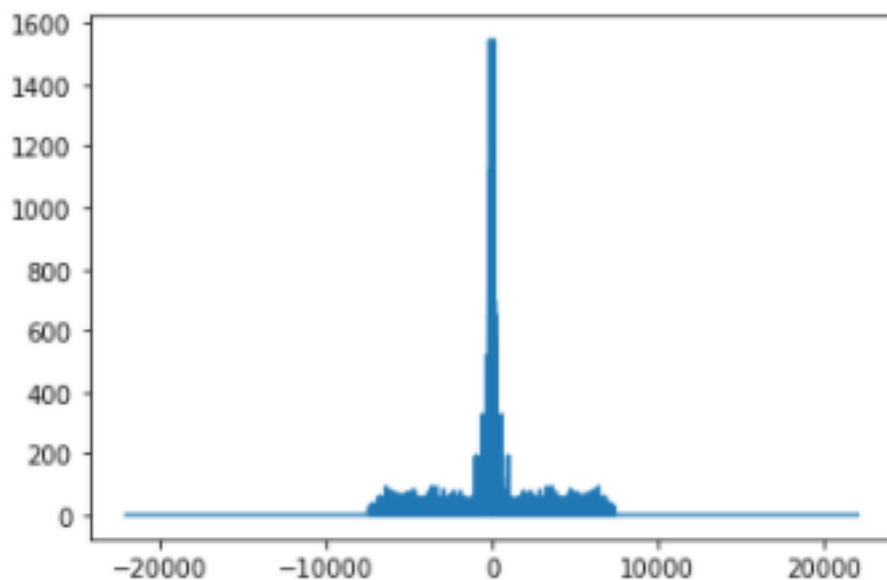


Рис. 9: Примененный фильтр сглаживания

Теперь масштабируем полученный спектр:

```
1 sampled_spectrum.scale(factor)
2 spectrum.plot()
```

```
3 sampled_spectrum.plot()
```

### Листинг 8: Масштабирование спектра

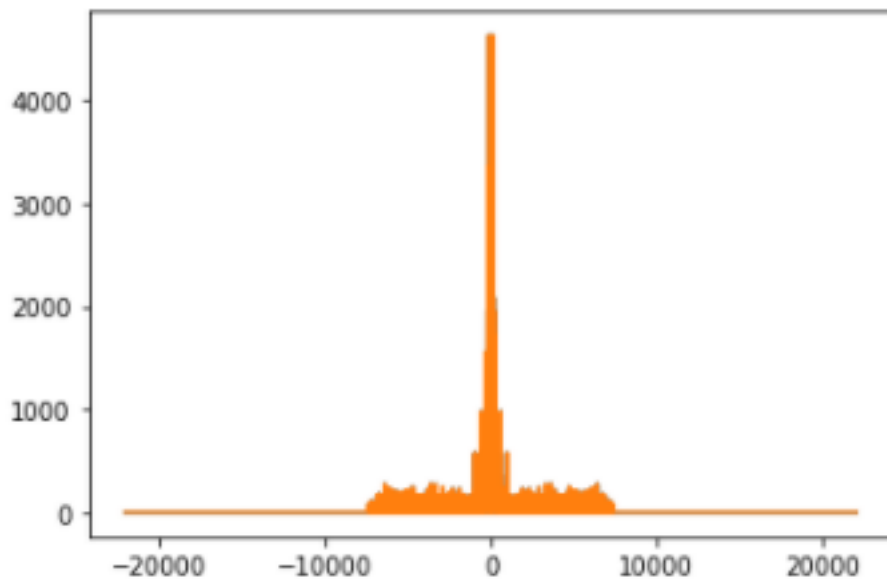


Рис. 10: Масштабирование спектра

После этого проверим разницу между спектром до и после фильтрации, после чего преобразуем спектр обратно в волну:

```
B [13]: spectrum.max_diff(sampled_spectrum)
```

```
Out[13]: 1.8189894035458565e-12
```

```
B [14]: interpolated = sampled_spectrum.make_wave()  
interpolated.make_audio()
```

```
Out[14]:
```

▶ 0:00 / 0:06 ————— 🔊 ⋮

Рис. 11: Сравнение спектров и преобразование в волну

И, наконец, посмотрим на полученный сигнал:

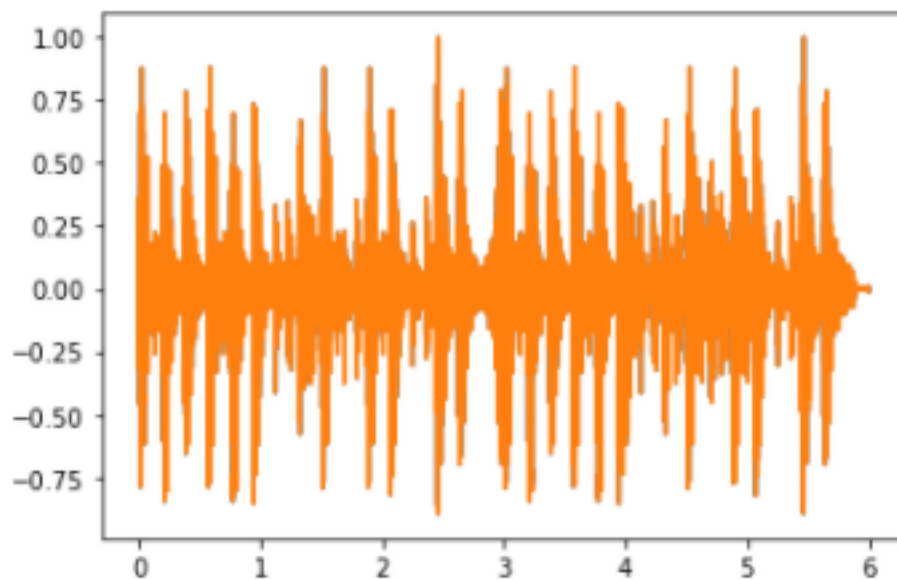


Рис. 12: Результат

По итогу можно сделать вывод о том, что разница между интерполированной и фильтрованной волной очень мала.

## 4 Выводы

В результате выполнения данной лабораторной работы мы разобрались со сверткой с импульсами, с выборкой и, наконец, с фильтрацией спектров. Мы изучили весь блокнот `chap11.ipynb`, также посмотрели видеоролик Криса "Монти"Монтгомери - "D/A and A/D | Digital Show and Tell из которого мы узнали почему аналоговое аудио в приемлемых пределах человеческого слуха может воспроизводиться с идеальной точностью с использованием 16-битного цифрового сигнала 44.1 кГц. Также в последнем пункте мы на основе "Соло на барабане"изменили фильтр НЧ до выборки и с помощью него же удалили спектральные копии, полученные в результате выборки.