

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе №11
Дисциплина: Телекоммуникационные технологии
Тема: Модуляция и выборка (квантование)

Работу выполнил:
Ляшенко В.В.
Группа: 3530901/80201
Преподаватель:
Богач Н.В.

Санкт-Петербург
2021

Оглавление

| | | |
|---|-----------------|----|
| 1 | Упражнение 11.1 | 4 |
| 2 | Упражнение 11.2 | 5 |
| 3 | Упражнение 11.3 | 6 |
| 4 | Выводы | 11 |

Список иллюстраций

| | | |
|-----|--|----|
| 3.1 | Звук барабана | 6 |
| 3.2 | Спектр сигнала | 7 |
| 3.3 | Результат применения фильтра сглаживания | 7 |
| 3.4 | Спектр после выборки | 8 |
| 3.5 | Спектр после использования фильтра | 9 |
| 3.6 | Результат масштабирования | 9 |
| 3.7 | Сравнение сигналов | 10 |

Листинги

| | | |
|-----|--|----|
| 3.1 | Получение звука барабана | 6 |
| 3.2 | Применение фильтра сглаживания | 7 |
| 3.3 | Функция выборки | 8 |
| 3.4 | Построение спектра | 8 |
| 3.5 | Применение фильтра сглаживания | 8 |
| 3.6 | Применение масштабирования | 9 |
| 3.7 | Получение сигнала | 10 |
| 3.8 | Сравнение интерполированного и фильтрованного сигналов | 10 |

Глава 1

Упражнение 11.1

В начале мы должны для Jupyter загрузить `chap11.ipynb`, прочитать пояснения и запустить примеры.

Все примеры были успешно запущены.

Глава 2

Упражнение 11.2

В этом упражнении нам требуется посмотреть видео "D/A and A/D | Digital Show and Tell" Криса Монтгомери. В этом видео он демонстрирует теорему о выборках в действии и представляет множество информации о выборках.

Также в этом видео мы можем узнать, почему аналоговый звук в допустимых пределах человеческого слуха (от 20 Гц до 20 кГц) может воспроизводиться с идеальной точностью с использованием 16-битного цифрового сигнала 44,1 кГц.

Глава 3

Упражнение 11.3

Вернемся к примеру "Соло на барабане" и применим фильтр НЧ до выборки, а затем, опять же с помощью фильтра НЧ, удалим спектральные копии, вызванные выборкой. Для начала загрузим звук барабана (Рис.3.1).

```
from thinkdsp import read_wave

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

Листинг 3.1: Получение звука барабана

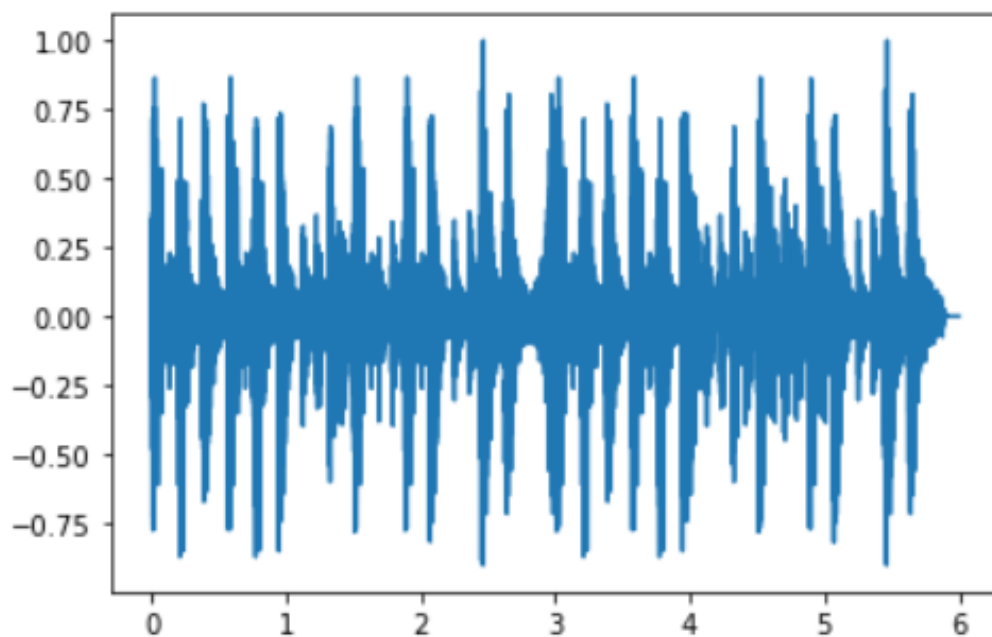


Рис. 3.1: Звук барабана

Данный сигнал имеет частоту дискретизации 44100 Гц. Теперь получим его спектр (Рис.3.2).

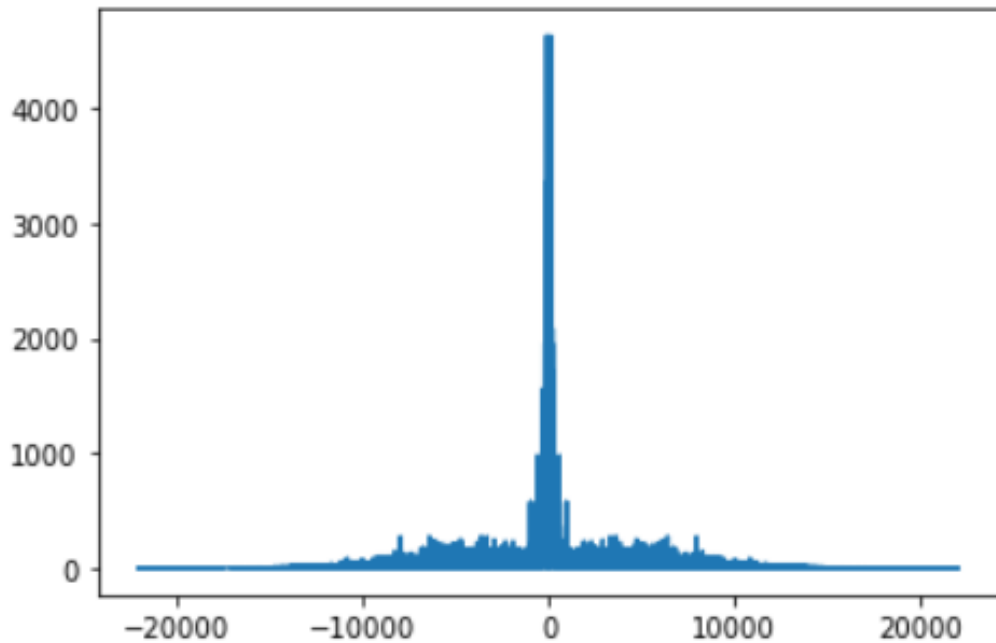


Рис. 3.2: Спектр сигнала

Уменьшим частоту дискретизации в 4 раза. А затем перед дискретизацией мы применим фильтр сглаживания, чтобы удалить частоты выше новой частоты свертки, которая равна частоте кадров разделённой на 2 (Рис.3.3).

```
factor = 4
framerate = wave.framerate / factor
cutoff = framerate / 2 - 1
spectrum.low_pass(cutoff)
spectrum.plot()
```

Листинг 3.2: Применение фильтра сглаживания

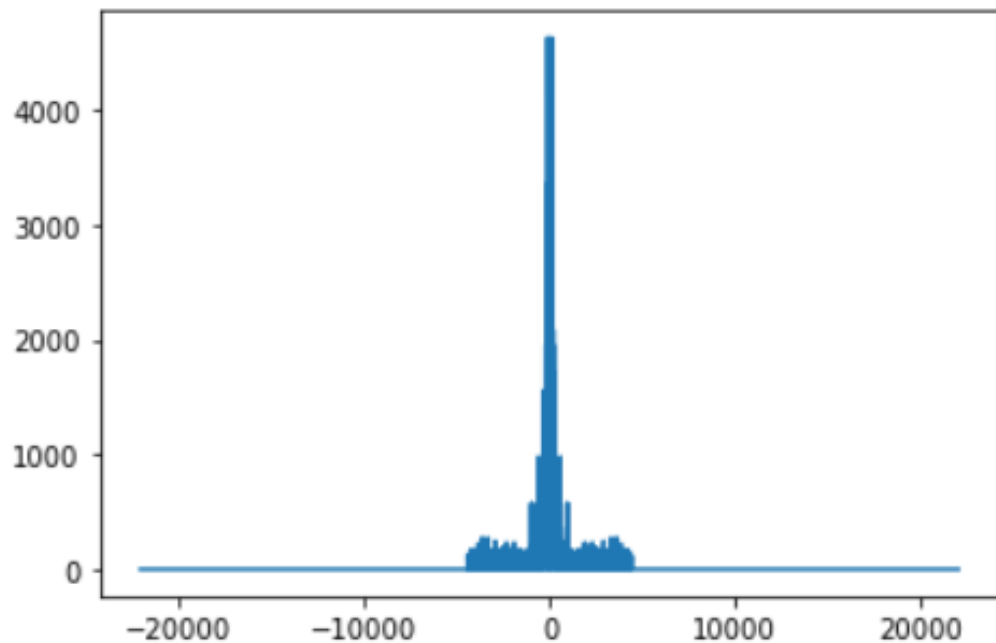


Рис. 3.3: Результат применения фильтра сглаживания

Теперь воспользуемся функцией, имитирующей процесс выборки.

```
from thinkdsp import Wave

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

Листинг 3.3: Функция выборки

Результат применения функции содержит копии спектра, которые слегка заметны при прослушивании звука. Но их можно увидеть при построении спектра (Рис.3.4).

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

Листинг 3.4: Построение спектра

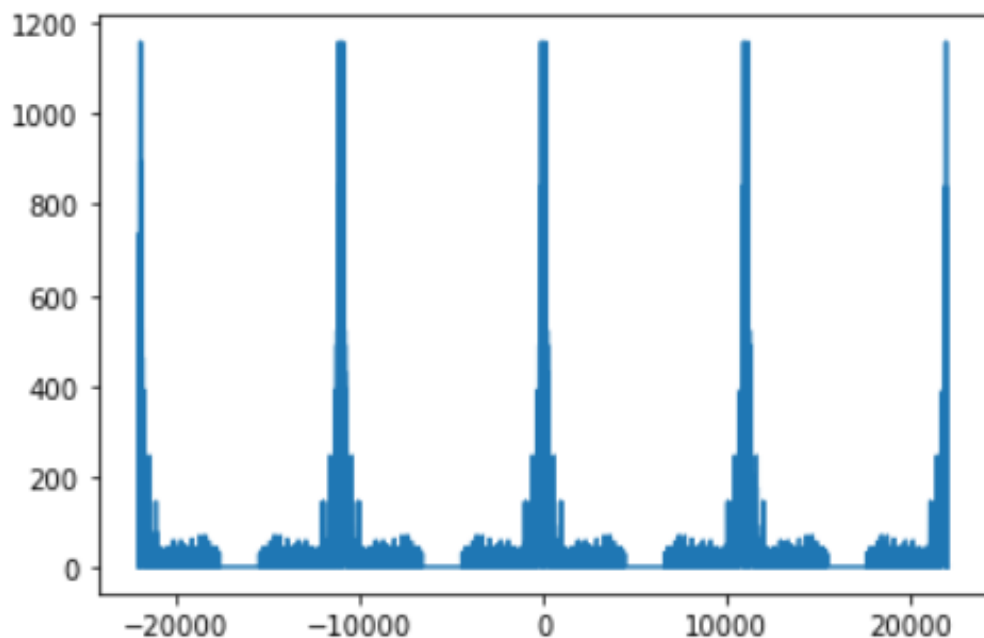


Рис. 3.4: Спектр после выборки

Мы можем избавиться от спектральных копий, снова применив фильтр сглаживания (Рис.3.5).

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

Листинг 3.5: Применение фильтра сглаживания

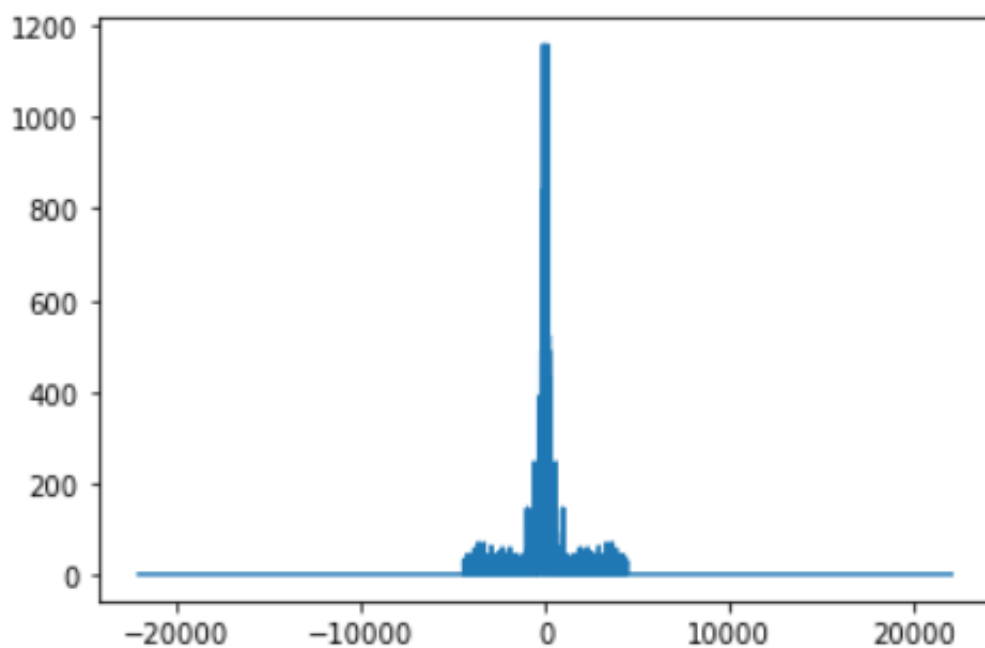


Рис. 3.5: Спектр после использования фильтра

Мы только что потеряли половину энергии в спектре, но мы можем масштабировать результат, чтобы вернуть её.

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

Листинг 3.6: Применение масштабирования

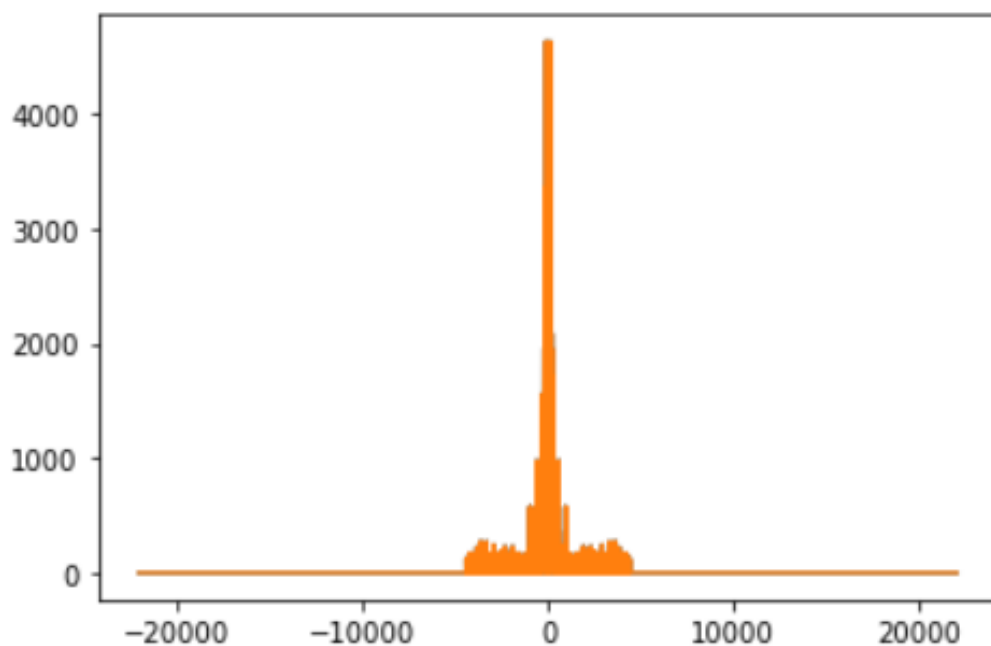


Рис. 3.6: Результат масштабирования

Как мы видим, между спектрами до дискретизации и после нет большой разницы.

Применение функции `max_diff` это подтверждает. Полученная разница: $1.8749713606747085e-12$.

После фильтрации и масштабирования мы можем снова получить сигнал.

```
interpolated = sampled_spectrum.make_wave()  
interpolated.make_audio()
```

Листинг 3.7: Получение сигнала

Разница между интерполированным и фильтрованным сигналом также должна быть небольшой.

```
filtered.plot()  
interpolated.plot()  
  
filtered.max_diff(interpolated)
```

Листинг 3.8: Сравнение интерполированного и фильтрованного сигналов

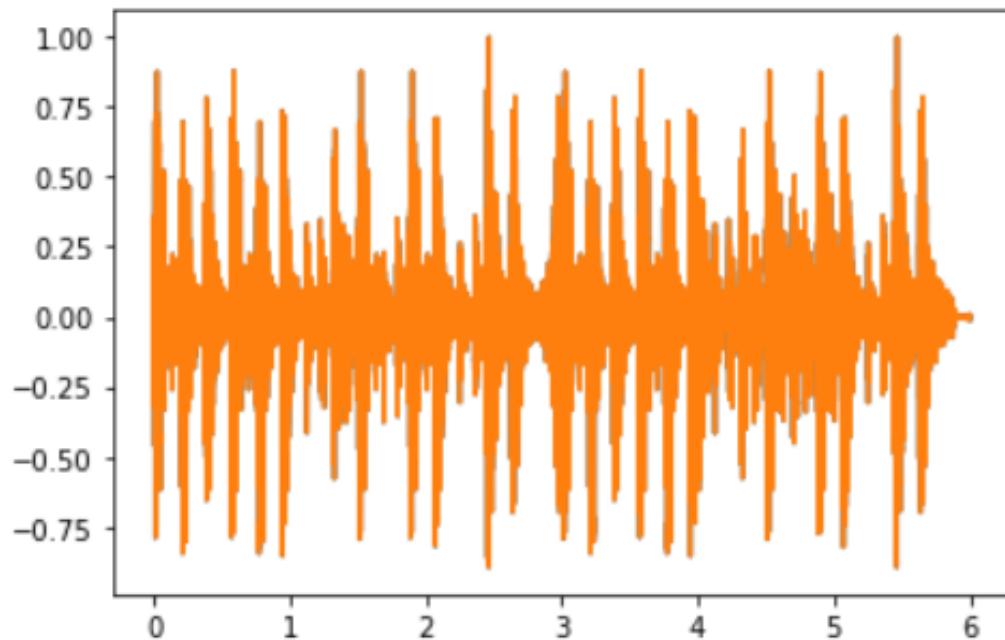


Рис. 3.7: Сравнение сигналов

Полученная разница: $5.56290642113787e-16$.

Глава 4

Выводы

В результате выполнения данной работы мы изучили амплитудную модуляцию, которая играет важную роль в радиосвязи, и ещё теорему о выборках, которая является важнейшей в цифровой обработке сигналов. Также мы получили навыки их применения.