

Санкт-Петербургский государственный политехнический  
университет Петра Великого.

**Высшая школа интеллектуальных систем и  
суперкомпьютерных технологий**

ЛАБОРАТОРНАЯ РАБОТА

Модуляция и выборка (квантование)

Работу выполнила студентка:

\_\_\_\_\_ А. И. Луцкевич  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Преподаватель лабораторных  
работ:

\_\_\_\_\_ Н. В. Богач  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

Санкт-Петербург, 2021 г.

## Суть работы 11.1:

Прочитать пояснения и запустить примеры.

Был прочитан весь параграф, были запущены все примеры и просмотрены все результаты.

## Суть работы 11.2:

Необходимо просмотреть ролик Криса "Монти"Монтгомери "D/A and A/D | Digital Show and Tell".

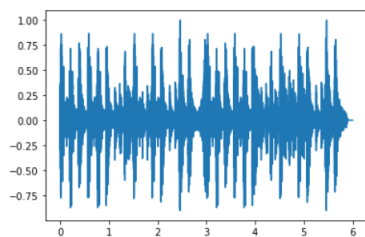
После просмотра данного видеоролика было выяснено, почему аналоговое аудио в приемлемых пределах человеческого слуха может воспроизводиться с идеальной точностью с использованием 16-битного цифрового сигнала 44.1 кГц.

## Суть работы 11.3:

Необходимо применить фильтр НЧ к примеру "Соло на барабане" до выборки, после чего снова с помощью фильтра НЧ удалить спектральные копии, которые вызваны выборкой.

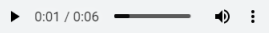
Получим исходный сигнал, выведем его на экран и переведем в аудио:

```
In [3]: wave = read_wave('amen-break-a-160-bpm.wav')  
        wave.normalize()  
        wave.plot()
```



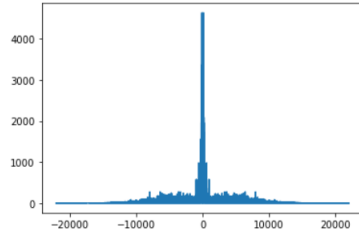
```
In [4]: wave.make_audio()
```

Out[4]:



Далее посмотрим на спектр сигнала.

```
In [5]: spectrum = wave.make_spectrum(full=True)
spectrum.plot()
```

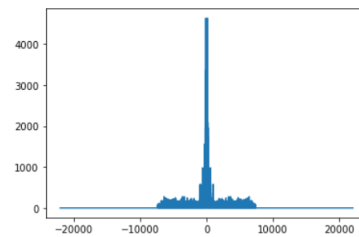


Уменьшим частоту дискретизации в 3 раза.

```
In [6]: factor = 3
framerate = wave.framerate / factor
cutoff = framerate / 2 - 1
```

Применим фильтр сглаживания для удаления частот выше новой частоты свертки.

```
In [7]: spectrum.low_pass(cutoff)
spectrum.plot()
```



```
In [8]: filtered = spectrum.make_wave()
filtered.make_audio()
```

Out[8]:



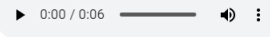
Полученный сигнал похож на исходный сигнал.

Далее напишем специальную функцию `sample`. Она имитирует процесс выборки. Сразу же проверим эту функцию.

```
In [9]: def sample(wave, factor):
ys = np.zeros(len(wave))
ys[::factor] = np.real(wave.ys[::factor])
return Wave(ys, framerate=wave.framerate)
```

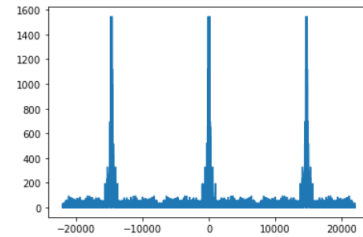
```
In [10]: sampled = sample(filtered, factor)
sampled.make_audio()
```

Out[10]:



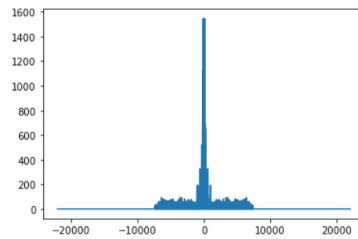
Создадим спектр данного сигнала, чтобы увидеть слабые спектральные копии.

```
In [11]: sampled_spectrum = sampled.make_spectrum(full=True)
sampled_spectrum.plot()
```



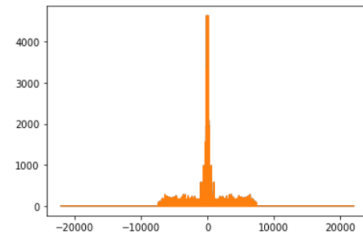
Применим фильтр сглаживания, чтобы убрать спектральные копии.

```
In [12]: sampled_spectrum.low_pass(cutoff)
sampled_spectrum.plot()
```



Масштабируем полученный спектр.

```
In [13]: sampled_spectrum.scale(factor)
spectrum.plot()
sampled_spectrum.plot()
```



Теперь необходимо проверить разницу между спектром до и после фильтрации, дальше преобразуем спектр обратно в волну.

```
In [14]: spectrum.max_diff(sampled_spectrum)
```

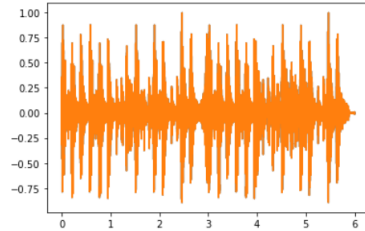
```
Out[14]: 1.8189894035458565e-12
```

```
In [15]: interpolated = sampled_spectrum.make_wave()
interpolated.make_audio()
```

```
Out[15]:
```

Посмотрим на полученный сигнал.

```
In [16]: filtered.plot()  
         interpolated.plot()
```



Делаем вывод о том, что разница между интерполированной и фильтрованной волной очень мала.

## Заключение:

В данной лабораторной работе мы разобрались с сверткой с импульсами, с выборкой и с фильтрацией спектров. Мы изучили блокнот `chap11.ipynb`, 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 на основе "Соло на барабанах" изменили фильтр НЧ до выборки и с помощью него же удалили спектральные копии, полученные в результате выборки.