

Санкт-Петербургский Государственный Политехнический Университет  
Институт Компьютерных Наук и Технологий

**Высшая школа интеллектуальных систем и суперкомпьютерных  
технологий**

Отчёт по лабораторной работе №1  
на тему  
**Звуки и Сигналы**

**Работу выполнил**  
Студент группы 3530901/80203  
Тарасенко Н.С.  
**Преподаватель**  
Богач Н.В.

Санкт-Петербург, 2021 год

## 1 Настройка проекта

Устанавливаем jupyter с помощью дистрибутива anaconda. Делаем fork репозитория thinkdsp и открываем необходимые файлы для работы, также создаем собственный файл, где будем работать со звуками.

## 2 Скачивание звука и работа с ним

В этом упражнении мы пользуемся помощью бесплатного ресурса с различными звуками и качаем в папку репозитория звук, подходящий для нашей лабораторной работы. В папке code файл "hard.wav" это и есть наш звук. Дальше мы обращаемся напрямую к файлу звука и обрезаем интервал в полсекунды

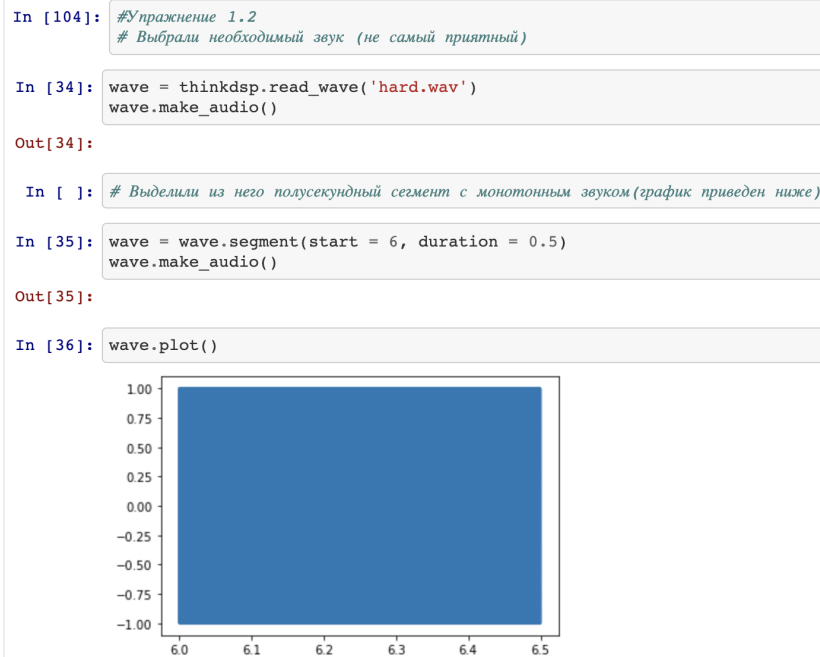
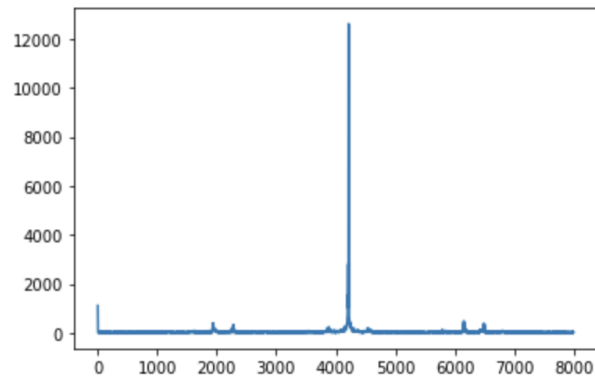


Рис. 1: 2

## 3 Спектр звука

Теперь рассмотрим спектр нашего полусекундного сегмента звука.

```
In [37]: spectr = wave.make_spectrum()  
spectr.plot(8000)
```



```
In [ ]: # С помощью методов low_pass и high_pass отфильтруем гармоники
```

```
In [42]: spectr2 = wave.make_spectrum()  
spectr2.low_pass(7000)  
spectr2.high_pass(4000)  
spectr2.plot(8000)
```

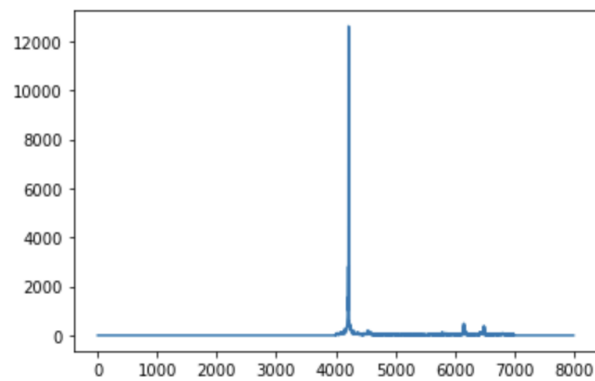


Рис. 2: 3

С помощью функций `lowPass` and `highPass` мы отфильтровали наш спектр и убрали ненужные гармоники.

Преобразуем спектр обратно в сигнал и послушаем результат. Звук стал более приятным и не таким ушираздирающим, стал тише и похожим уже не скример, а на аппарат сердцебиения (когда сердце уже не бьется )

```
In [39]: filtered = spectr2.make_wave()
         filtered.plot()
         filtered.make_audio()
```

Out[39]:

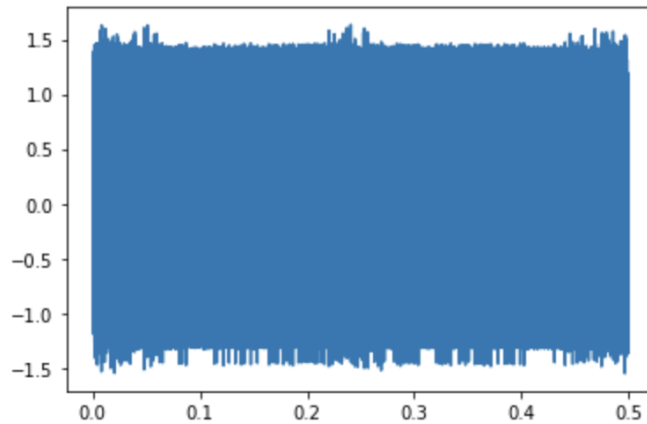


Рис. 3: 4

## 4 Создание сложного сигнала

Создадим сложный сигнал

```
In [134]: #Упражнение 1.3
          #Возьмем несколько сигналов (син/кос с разными частотами):
          #Частоты брал из нот 2-3 октавы
```

```
In [135]: sin_sig_11 = thinkdsp.SinSignal(freq=523.25, amp=0.5, offset=0)
          sin_sig_12 = thinkdsp.SinSignal(freq=587.33, amp=0.5, offset=0)
          sin_sig_17 = thinkdsp.SinSignal(freq=1568.0, amp=0.5, offset=0)
          cos_sig_16 = thinkdsp.CosSignal(freq=880.00, amp=1.0, offset=0)
          cos_sig_17 = thinkdsp.CosSignal(freq=987.77, amp=1.0, offset=0)

          my_mix = sin_sig_11 + sin_sig_12 + sin_sig_17 + cos_sig_16 + cos_sig_17
          my_mix.plot()

          wave = my_mix.make_wave(duration=3, start=0)
          wave.make_audio()
```

Out[135]:

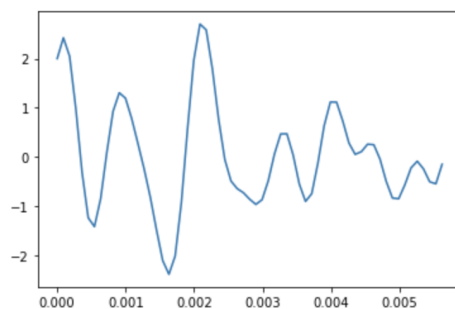
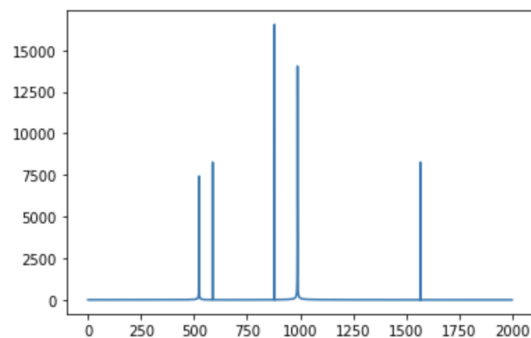


Рис. 4: 3

Наш сигнал получился схожим с сигналом автомобиля.  
Сейчас выведем спектр нашего сигнала. И посмотрим на результаты.

```
In [ ]: # Посмотрим на спектр нашего сигнала
```

```
In [136]: spectrum = wave.make_spectrum()  
spectrum.plot(2000)
```



```
In [138]: sin_sig_13 = thinkdsp.SinSignal(freq=698.46, amp=0.5, offset=0)  
my_mix += sin_sig_3  
my_mix.plot()
```

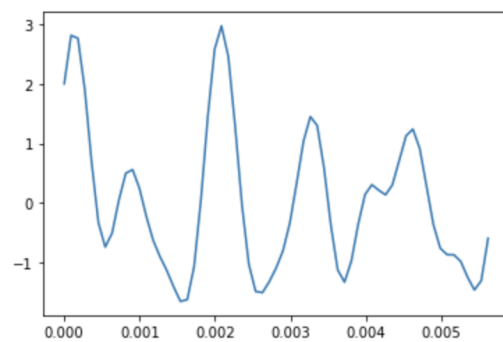


Рис. 5: Исходный звук

```
In [ ]: #Получился не очень приятный звук  
#Посмотрим на его спектр
```

```
In [140]: wave = my_mix.make_wave(duration=3, start=0, framerate=31025)  
wave.make_audio()
```

Out[140]:

```
In [141]: spectrum = wave.make_spectrum()  
spectrum.plot(2000)
```

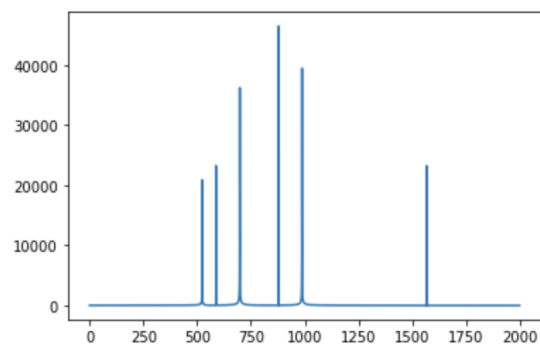


Рис. 6: 6

## 5 Функция изменения длины сигнала

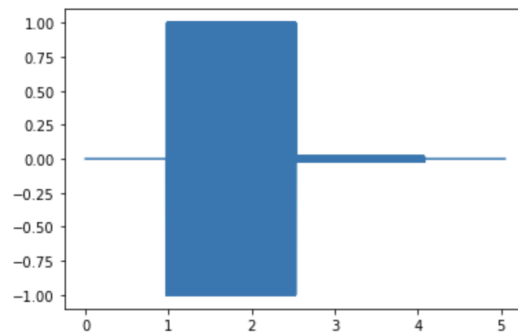
Мы написали метод а дальше создали 2 сигнала и применили разные коэффициенты. Первый маленький коэф делает сигнал короче, а больший коэффициент увеличивает длину сигнала.

```
In [ ]: # Упражнение 1.4
        # Реализовали метод stretch
        # Попробовали ускорить наш сигнал и замедлить и получили следующие звуки.
```

```
In [157]: def stretch(wave, factor):
           wave.ts *= factor
           wave.framerate /= factor

           w_fast = thinkdsp.read_wave('hard.wav')
           w_slow = thinkdsp.read_wave('hard.wav')
           stretch(w_fast, 0.3)
           stretch(w_slow, 50.0)
```

```
In [158]: w_fast.plot()
```



```
In [159]: w_fast.make_audio()
```

Out[159]:

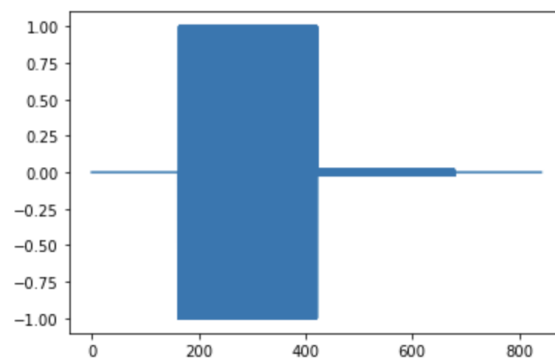
```
In [160]: w_slow.make_audio()
```

Out[160]:

```
In [161]: w_slow.plot()
```

Рис. 7: 7

```
In [161]: w_slow.plot()
```



```
In [ ]:
```

Рис. 8: 8

Во время выполнения лабораторной работы получены навыки работы со звуками, волнами

и спектрами. Также я научился находить более высокие и фундаментальные пики, определять частоту, а также ускорять и замедлять звуки и строить графики. SASHA PRIVET