

Санкт-Петербургский государственный политехнический
университет Петра Великого

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

Лабораторная работа

Апериодические сигналы

Выполнил студент гр. 3530901/80201

И.С. Иванов

Преподаватель:

Н.В. Богач

Санкт-Петербург
2021

Содержание

1	Упражнение №1: Изучить примеры из chap03	5
2	Упражнение №2: SawtoothChirp	8
3	Упражнение №3: Пилообразный сигнал (от 2500 Гц до 3000 Гц)	10
4	Упражнение №4: Глиссандо	11
5	Упражнение №5: TromboneGliss	12
6	Упражнение №6: Анализ гласных	13
7	Выводы	14

Список иллюстраций

1	Изучение и проверка примеров из файла (1)	5
2	Изучение и проверка примеров из файла (2)	6
3	Сравнение окон из NumPy	7
4	График конца сигнала	9
5	Спектрограмма сигнала	9
6	Спектр нашего пилообразного сигнала	10
7	Спектрограмма глissандо	11
8	Спектрограмма синтезированного глissандо на тромбоне	13
9	Спектрограмма гласных	14

Листинги

1	Проверка окон и вывод графика	6
2	Класс SawtoothChirp	8
3	Вывод графика и создание wav	8
4	Создание сигнала и создание аудио	10
5	Загрузка звука и выделения фрагмента	11
6	Создание и вывод на экран треугольного сигнала	11
7	Класс TromboneGliss	12
8	Создание сигнала, имитирующего глиссандо	12
9	Вывод спектрограммы сигнала	12
10	Чтение и прослушивание аудио	13
11	Спектр пилообразного сигнала	13

1 Упражнение №1: Изучить примеры из char03

Для выполнения первого пункта необходимо изучить и выполнить примеры из файла char02.ipynb. Также необходимо в примере с утечкой заменить окно Хэмминга одним из окон, представляемых NumPy.

Запустим все примеры из этого файла.

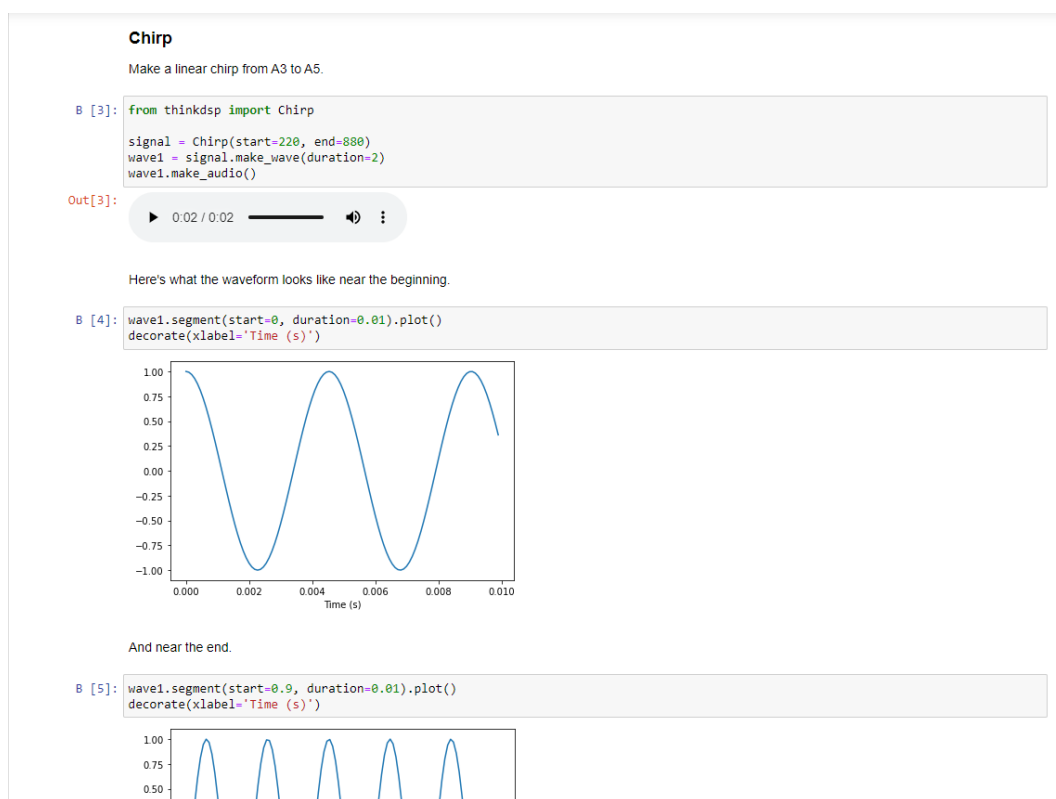


Рис. 1: Изучение и проверка примеров из файла (1)

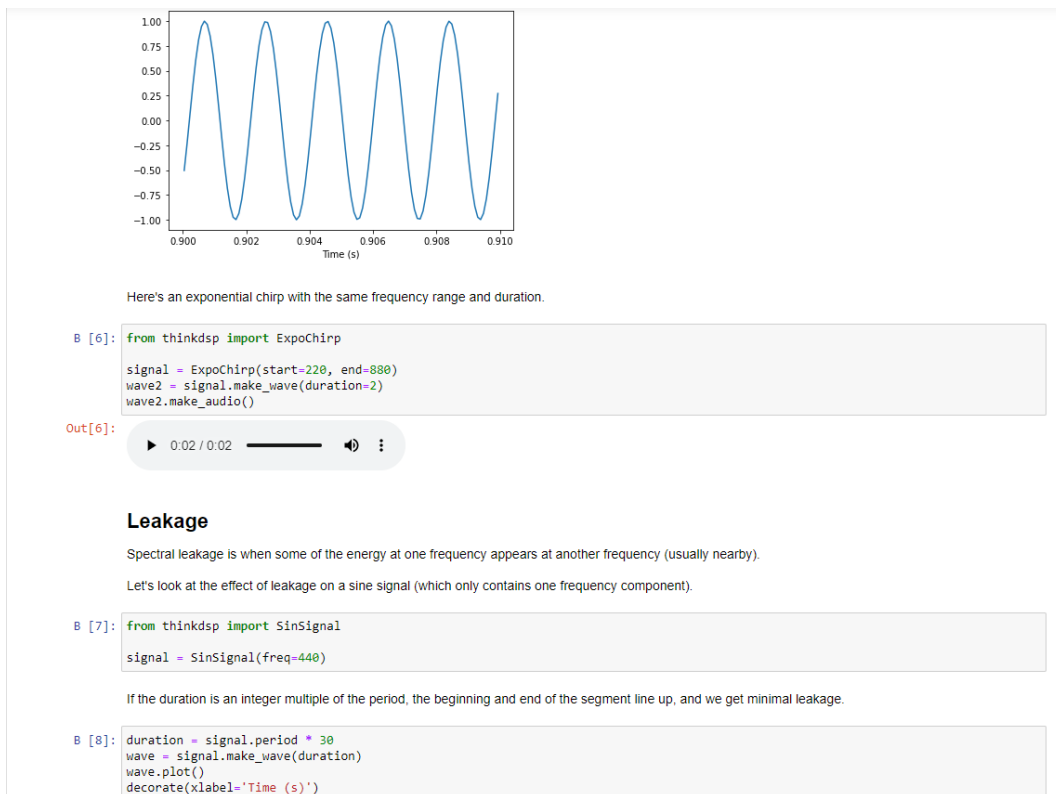


Рис. 2: Изучение и проверка примеров из файла (2)

Далее прогоним проверим окна предоставляемые NumPy и выведем все на график.

```

1     for window_func in [np.bartlett, np.blackman, np.hamming, np.hanning]:
2         wave = signal.make_wave(duration)
3         wave.ys *= window_func(len(wave.ys))
4
5         spectrum = wave.make_spectrum()
6         spectrum.plot(high=880, label=window_func.__name__)
7     decorate()
8

```

Листинг 1: Проверка окон и вывод графика

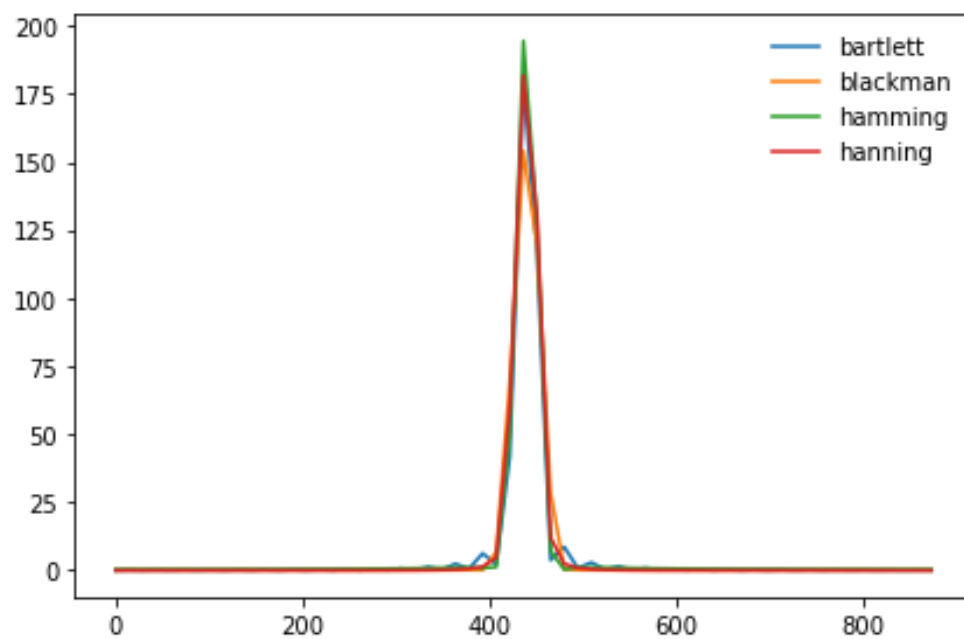


Рис. 3: Сравнение окон из NumPy

2 Упражнение №2: SawtoothChirp

Во втором упражнении необходимо написать класс `SawtoothChirp`, который расширяет `Chirp` и переопределяет `evaluate` для генерации пилообразного сигнала с линейно увеличивающейся частотой.

Подключение зависимостей и написание класса `SawtoothSignal`:

```
1     from thinkdsp import Chirp
2     from thinkdsp import normalize, unbias
3
4     PI2 = 2 * np.pi
5
6     class SawtoothChirp(Chirp):
7         def evaluate(self, ts):
8             freqs = np.linspace(self.start, self.end, len(ts))
9             dts = np.diff(ts, prepend=0)
10            dphis = PI2 * freqs * dts
11            phases = np.cumsum(dphis)
12            cycles = phases / PI2
13            frac, _ = np.modf(cycles)
14            ys = normalize(unbias(frac), self.amp)
15            return ys
16
```

Листинг 2: Класс `SawtoothChirp`

Проверим работу класса. Выведем график сигнала и послушаем получившийся wav:

```
1     signal = SawtoothChirp(start=220, end=880)
2     wave = signal.make_wave(duration=1, framerate=4000)
3     wave.apodize()
4     wave.make_audio()
5
```

Листинг 3: Вывод графика и создание wav

Выведем конец сигнала.

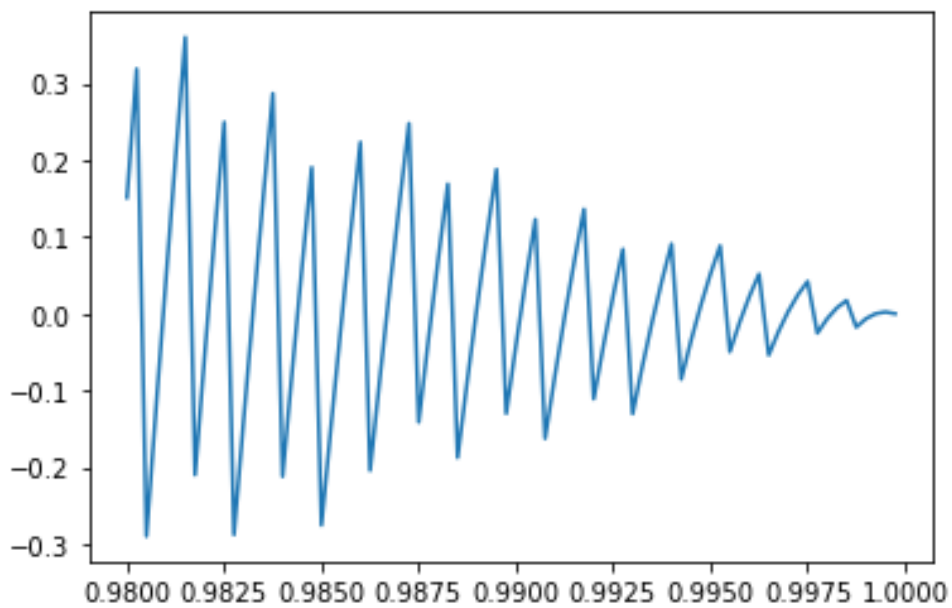


Рис. 4: График конца сигнала

Выведем спектрограмму:

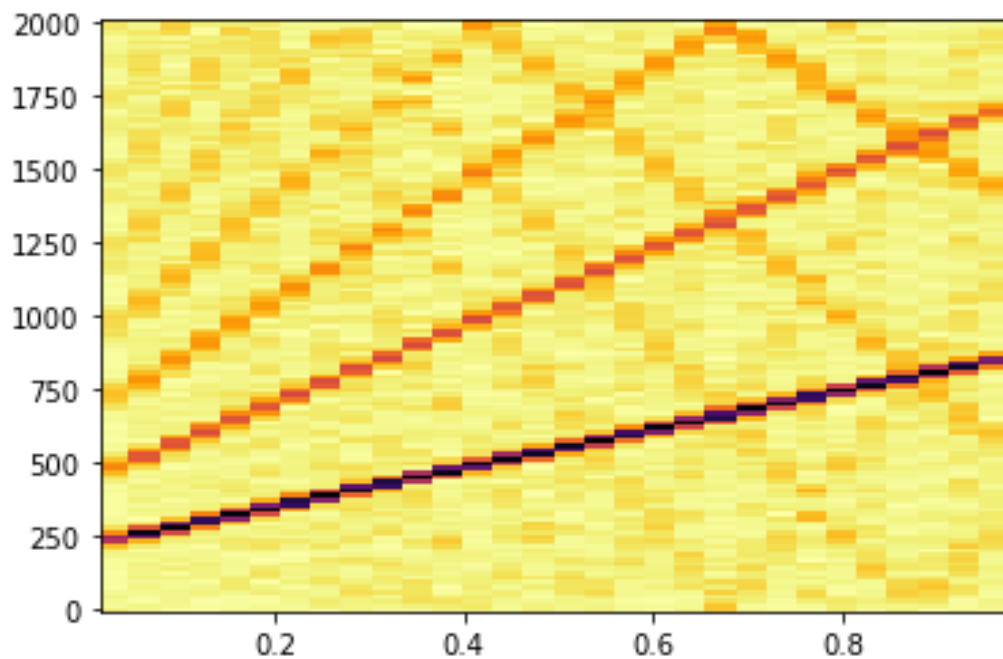


Рис. 5: Спектрограмма сигнала

По результатам выполнения данного упражнения был написан класс `SawtoothChirp`, генерирующий пилообразный сигнал с линейно увеличивающейся частотой. Класс был проверен созданием и прослушиванием аудио и изучением графика и спектрограммы.

3 Упражнение №3: Пилообразный сигнал (от 2500 Гц до 3000 Гц)

В третьем упражнении нам необходимо создать пилообразный chirp, меняющийся от 2500 до 3000 Гц, и на его основе сгенерировать сигнал длительностью 1с и framerate = 20 кГц.

Создадим пилообразного сигнала и прослушаем его:

```
1 signal = SawtoothChirp(start=2500, end=3000)
2 wave = signal.make_wave(duration=1, framerate=20000)
3 wave.make_audio()
4
```

Листинг 4: Создание сигнала и создание аудио

Посмотри на его спектр.

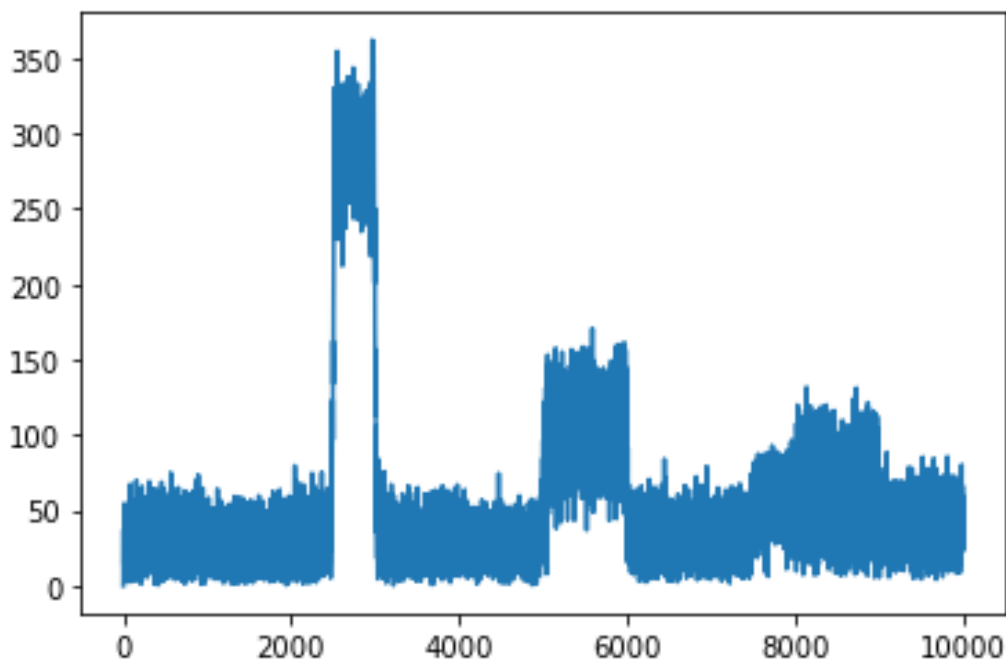


Рис. 6: Спектр нашего пилообразного сигнала

По результатам выполнения данного упражнения, сравнивая два сигнала, можно сделать вывод, что сигнал богат частотами. Аудио сигнала режет слух.

4 Упражнение №4: Глиссандо

В четвертом упражнении необходимо скачать звук глиссандо и вывести спектрограмму. Звук был взят отсюда <https://archive.org/details/rhapblue11924>.

Загрузка звука и выделение фрагмента:

```
1 wave = read_wave('Sounds/rhapblue11924.wav').segment(start=7.75, duration  
=1.5)  
2 wave.make_audio()  
3
```

Листинг 5: Загрузка звука и выделения фрагмента

Вывод спектрограммы.

```
1 wave.make_spectrogram(512).plot(high=5000)  
2
```

Листинг 6: Создание и вывод на экран треугольного сигнала

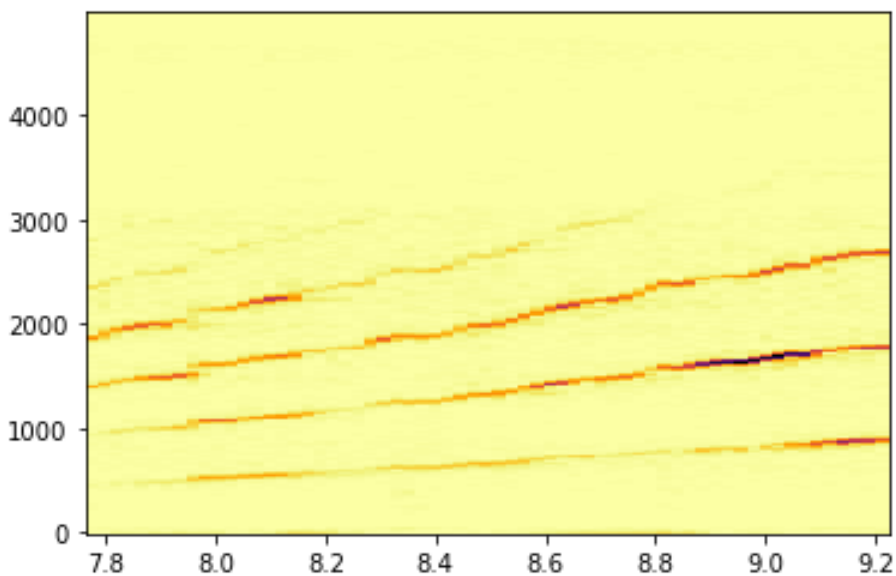


Рис. 7: Спектрограмма глиссандо

5 Упражнение №5: TromboneGliss

В пятом упражнении необходимо написать класс `TromboneGliss`, расширяющий `Chirp` и преоставляющий `evaluate`. Создать сигнал, имитирующий глissандо на тромбоне от C3 до F3 и обратно до C3.

Создадим класс `TromboneGliss`:

```
1 class TromboneGliss(Chirp):
2     def evaluate(self, ts):
3         l1, l2 = 1.0 / self.start, 1.0 / self.end
4         lengths = np.linspace(l1, l2, len(ts))
5         freqs = 1 / lengths
6
7         dts = np.diff(ts, prepend=0)
8         dphis = PI2 * freqs * dts
9         phases = np.cumsum(dphis)
10        ys = self.amp * np.cos(phases)
11        return ys
12
```

Листинг 7: Класс `TromboneGliss`

Зададим требуемый сигнал

```
1 low = 262
2 high = 349
3 signal = TromboneGliss(high, low)
4 wave1 = signal.make_wave(duration=1)
5 wave1.apodize()
6 wave1.make_audio()
7
8 signal = TromboneGliss(low, high)
9 wave2 = signal.make_wave(duration=1)
10 wave2.apodize()
11 wave2.make_audio()
12
13 wave = wave1 | wave2
14 wave.make_audio()
15
```

Листинг 8: Создание сигнала, имитирующего глissандо

Выведем спектрограмму.

```
1 sp = wave.make_spectrogram(1024)
2 sp.plot(high=1000)
3
```

Листинг 9: Вывод спектрограммы сигнала

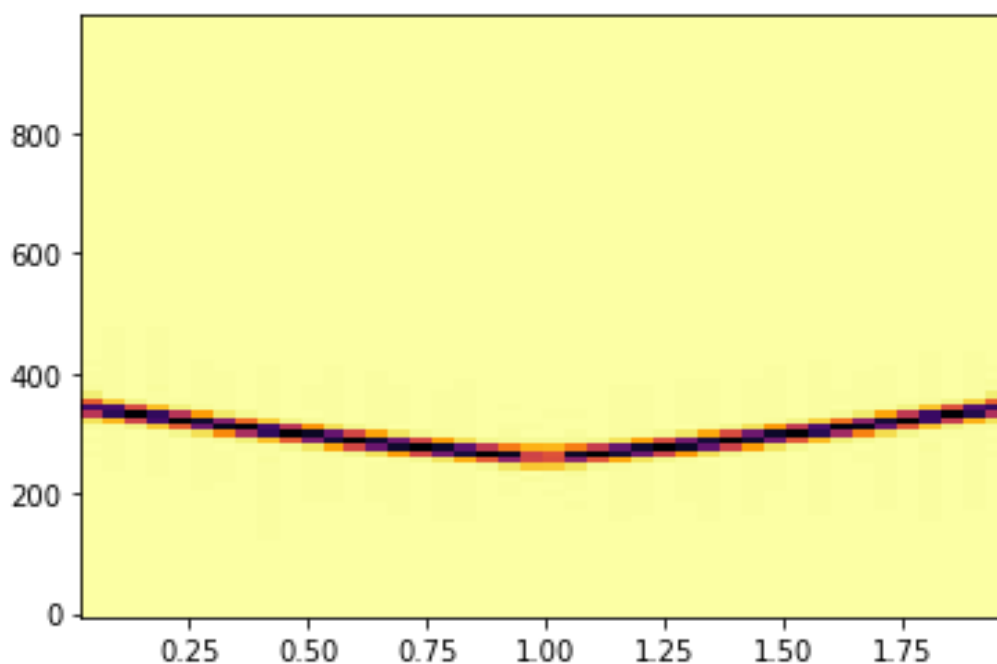


Рис. 8: Спектрограмма синтезированного глissандо на тромбоне

6 Упражнение №6: Анализ гласных

В шестом упражнении необходимо найти запись серии гласных звуков букв алфавита и построить их спектрограммы.

Прочитаем файл со звуком и прослушаем его:

```
1 wave = read_wave('Sounds/87778__marcgascon7__vocals.wav')
2 wave.make_audio()
3
```

Листинг 10: Чтение и прослушивание аудио

Посмотрим на спектрограмму:

```
1 wave.make_spectrogram(1024).plot(high=1000)
2
```

Листинг 11: Спектр пилообразного сигнала

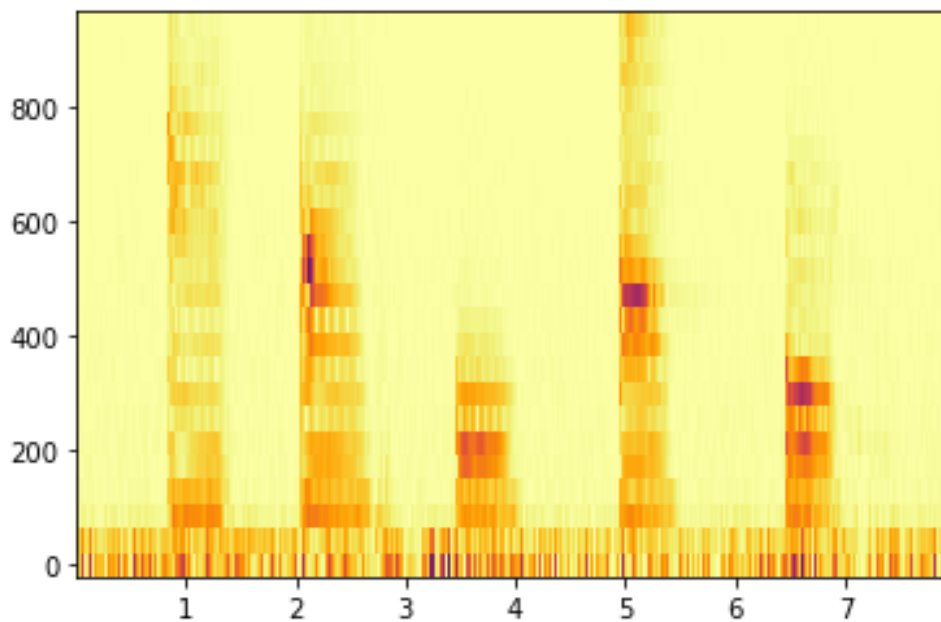


Рис. 9: Спектрограмма гласных

7 Выводы

В результате выполнения данной лабораторной работы мы изучили как надо работать с аperiodическими сигналами, и что такое chirp. Также научились строить спектрограмму chirp и работать с leakage. Были реализованы классы для реализации пилообразного chirp и для имитации глиссато на тромбоне. К этому всему были изучены спектрограммы гласных звуков.