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

Отчёт по лабораторной работе №3
Дисциплина: Телекоммуникационные технологии
Тема: Апериодические сигналы

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

Санкт-Петербург
2021

Оглавление

Список иллюстраций

Листинги

Глава 1

Упражнение 3.1

В начале нам требуется загрузить и послушать примеры из блокнота `chap03.ipynb`.

Теперь в примере с утечкой заменим окно Хэмминга одним из окон, предоставляемых NumPy, и посмотрим как они влияют на утечку. Утечка при использовании окна Хэмминга имеет вид, представленный на Рис.1.1.

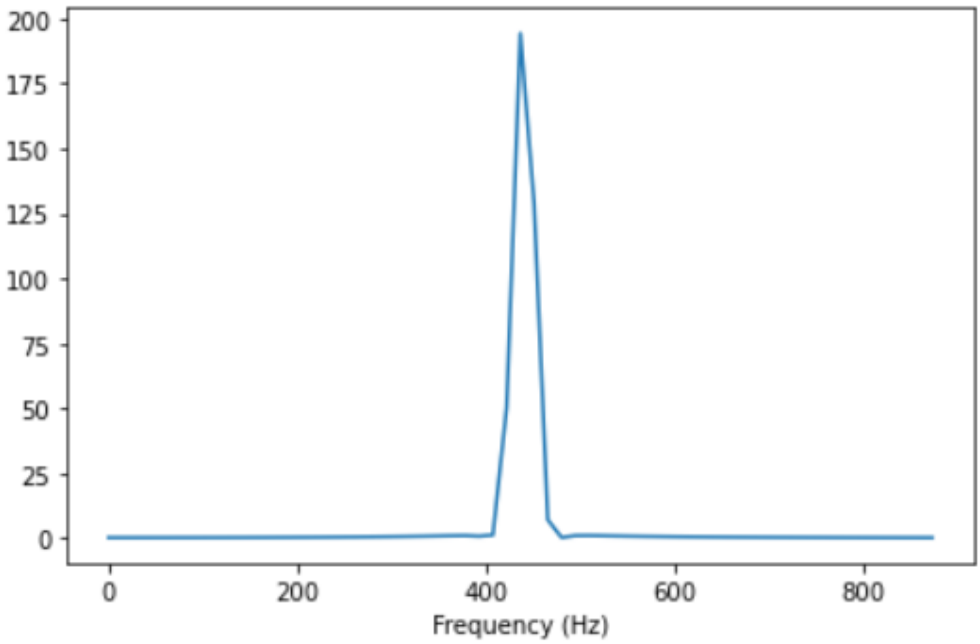

Рис. 1.1: Утечка. Окно Хэмминга

NumPy дает функции для вычисления других оконных функций, такие как `bartlett`, `blackman`, `hanning` и `kaiser`. Используем их (Рис.1.2).

```
import thinkplot
import numpy as np

wave = signal.make_wave(duration)
wave.window(np.bartlett(len(wave)))
spectrum = wave.make_spectrum()
spectrum.plot(high=880, label="bartlett")

wave = signal.make_wave(duration)
wave.window(np.blackman(len(wave)))
```

```
spectrum = wave.make_spectrum()
spectrum.plot(high=880, label="blackman")

wave = signal.make_wave(duration)
wave.window(np.hanning(len(wave)))
spectrum = wave.make_spectrum()
spectrum.plot(high=880, label="hanning")

wave = signal.make_wave(duration)
wave.window(np.kaiser(len(wave),5))
spectrum = wave.make_spectrum()
spectrum.plot(high=880, label="kaiser")

thinkplot.config(xlabel='Frequency (Hz)', legend=True)
```

Листинг 1.1: Вычисление разных оконных функций

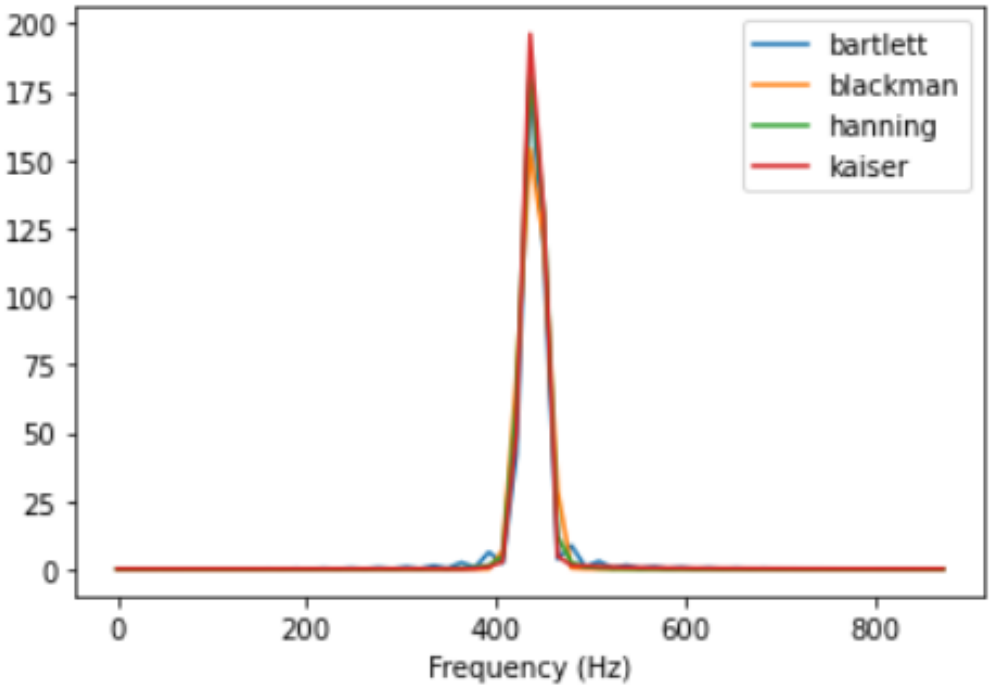

Рис. 1.2: Применение разных окон

Все окна хорошо справляются с уменьшением утечки.

Глава 2

Упражнение 3.2

2.1 Класс SawtoothChirp

Напишем класс `SawtoothChirp`, расширяющий `Chirp` и переопределяющий `evaluate` для генерации пилообразного сигнала с линейно увеличивающейся частотой.

```
from thinkdsp import Chirp, normalize, unbias, PI2

class SawtoothChirp(Chirp):

    def evaluate(self, ts):
        freqs = np.linspace(self.start, self.end, len(ts))
        dts = np.diff(ts, prepend=0)
        dphis = PI2 * freqs * dts
        phases = np.cumsum(dphis)
        cycles = phases / PI2
        frac, _ = np.modf(cycles)
        ys = normalize(unbias(frac), self.amp)
        return ys
```

Листинг 2.1: Класс SawtoothChirp

2.2 Спектрограмма

Нарисуем эскиз спектрограммы этого сигнала, а затем распечатаем.

```
signal = SawtoothChirp(start=220, end=880)
wave = signal.make_wave(duration=1, framerate=4000)
spectrogram = wave.make_spectrogram(128)
spectrogram.plot()
decorate(xlabel='Time (s)', ylabel='Frequency (Hz)')
```

Листинг 2.2: Создание спектрограммы

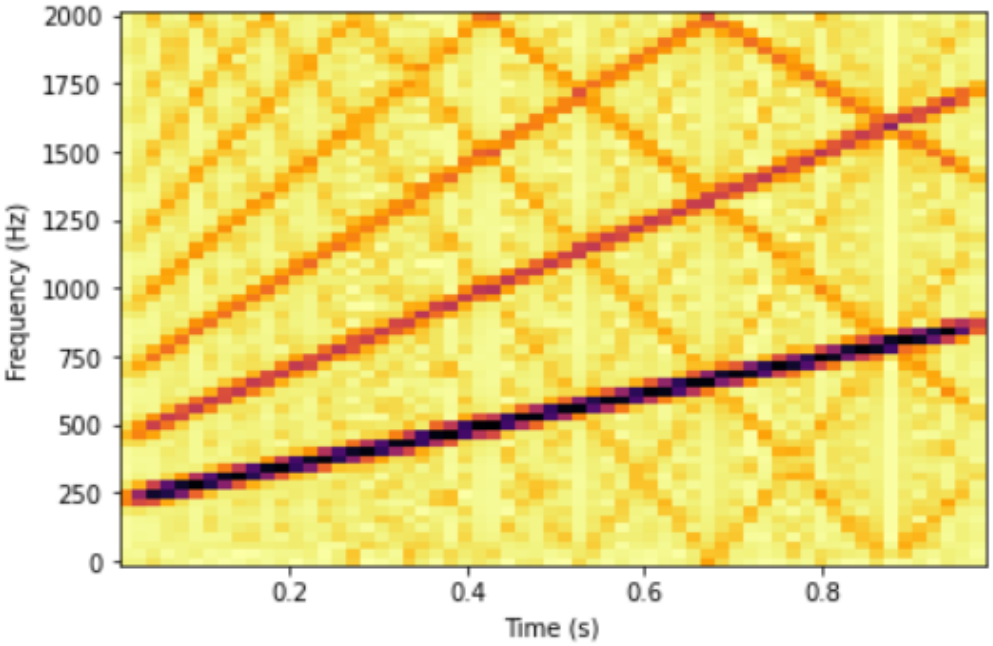

Рис. 2.1: Спектрограмма

Мы можем увидеть на рис.2.1, как гармоники с наложенными частотами отражаются от частоты сворачивания.

Послушаем получившийся сигнал. Именно отрожающиеся гармоники мы слышим как фоновое шипение.

Глава 3

Упражнение 3.3

Создадим пилообразный чирп, меняющийся от 2500 до 3000 Гц, и на его основе сгенерируем сигнал длительностью 1 с и частотой кадров 20 кГц.

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

Листинг 3.1: Создание пилообразного чирпа

Теперь нам нужно вывести спектр данного сигнала. Но прежде мы должны предположить как он будет выглядеть. Так как основная частота меняется в диапазоне от 2500 до 3000 Гц, то здесь будет большой всплеск. На первой гармоники (диапазон от 5000 до 6000 Гц,) будет всплеск поменьше, а на второй гармоники ([7500;9000]) - ещё ниже. Выведем спектр.

```
spectrogram = wave.make_spectrogram(128)
spectrogram.plot()
decorate(xlabel='Time (s)', ylabel='Frequency (Hz)')
```

Листинг 3.2: Получение спектра чирпа

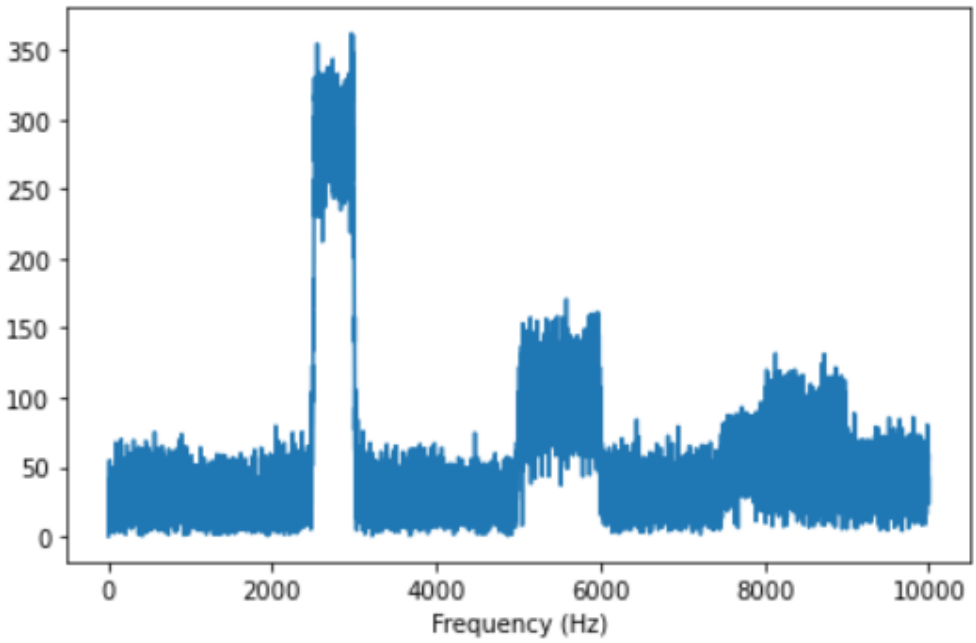

Рис. 3.1: Спектр чирпа

Как мы можем видеть из рис.3.1 полученный спектр совпал с ожидаемым.

Глава 4

Упражнение 3.4

В данном упражнении нам необходимо распечатать спектрограмму первых нескольких секунд звука глиссандо. Воспользуемся подсказкой из пособия и скачаем произведение "Rhapsody in Blue"Джорджа Гершвина, которое содержит глиссандо.

```
from thinkdsp import read_wave
wave = read_wave('sounds/rhapblue11924.wav')
segment = wave.segment(start=2, duration=9)
segment.make_audio()
```

Листинг 4.1: Получение звука глиссандо

Теперь выведем его спектрограмму (Рис.4.1).

```
spectrogram = segment.make_spectrogram(512)
spectrogram.plot()
decorate(xlabel='Time (s)', ylabel='Frequency (Hz)')
```

Листинг 4.2: Получение спектрограммы

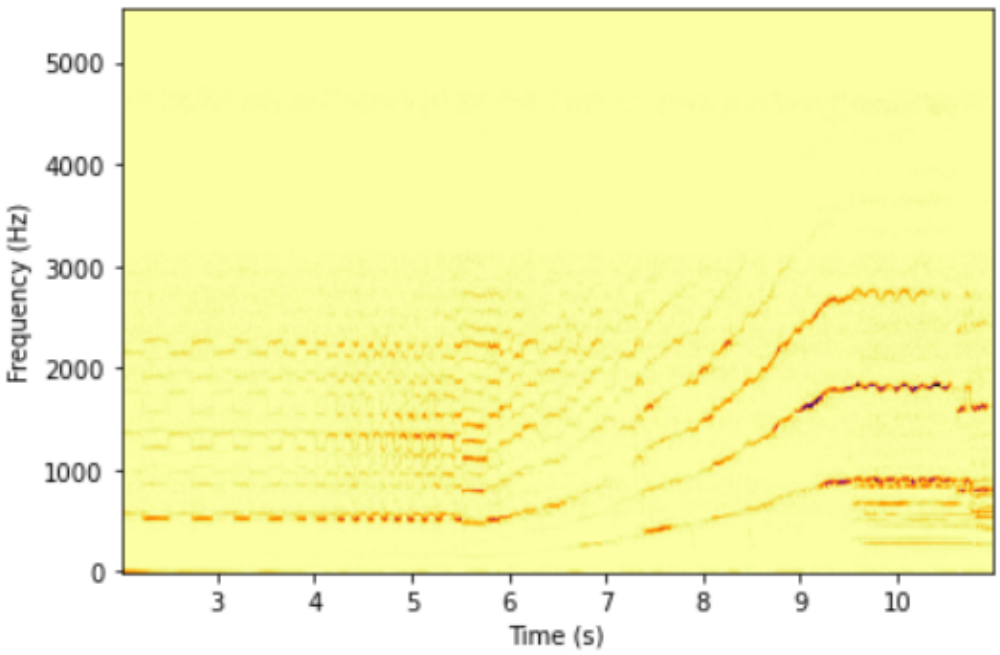

Рис. 4.1: Спектрограмма глиссандо

Глава 5

Упражнение 3.5

5.1 Класс TromboneGliss

Напишем класс `TromboneGliss`, расширяющий `Chirp` и предоставляющий `evaluate`.

```
class TrombonGliss(Chirp):

    def evaluate(self, ts):
        start, end = 1.0 / self.start, 1.0 / self.end
        freqs = 1.0 / np.linspace(start, end, len(ts))

        dts = np.diff(ts, prepend=0)
        dphis = PI2 * freqs * dts
        phases = np.cumsum(dphis)
        ys = self.amp * np.cos(phases)
        return ys
```

Листинг 5.1: Класс TromboneCliss

Создадим сигнал, имитирующий глиссандо на тромбоне от C3 до F3, и обратно до C3. C3 - 262 Гц, F3 - 349 Гц.

```
C3 = 262
F3 = 349
signal = TromboneGliss(C3, F3)
wave_CF = signal.make_wave(duration=1)
wave_CF.apodize()
signal = TromboneGliss(F3, C3)
wave_FC = signal.make_wave(duration=1)
wave_FC.apodize()
wave = wave_CF | wave_FC
wave.make_audio()
```

Листинг 5.2: Создание сигнала

5.2 Спектрограмма

Напечатаем спектрограмму полученного сигнала.

```
spectrogram = wave.make_spectrogram(1024)
spectrogram.plot(high=1000)
```

```
decorate(xlabel='Time (s)', ylabel='Frequency (Hz)')
```

Листинг 5.3: Получение спектрограммы

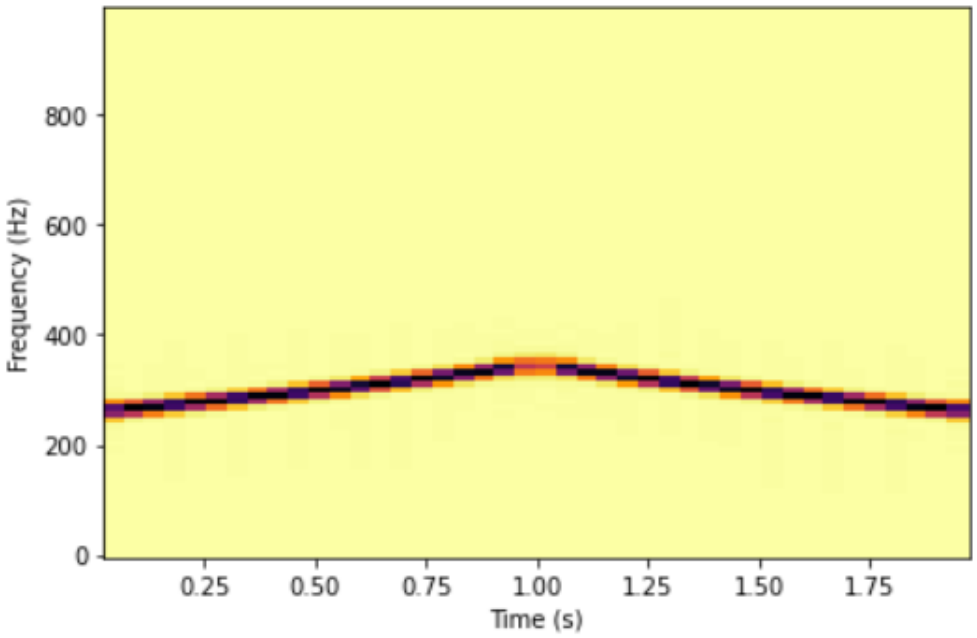

Рис. 5.1: Спектрограмма глиссандо C3-F3

Как мы можем видеть на рис.5.1, глиссандо больше похоже на линейный чирп.

Глава 6

Упражнение 3.6

6.1 Спектрограмма

Скачаем с `https://freesound.org` запись серии гласных звуков и посмотрим на спектрограмму.

```
wave = read_wave('sounds/523079__team-saul-nosthas__vowels-of-various-people.
    wav')
vowels = wave.segment(start=0, duration=6)
vowels.make_spectrogram(512).plot(high=1500)
decorate(xlabel='Time (s)', ylabel='Frequency (Hz)')
vowels.make_audio()
```

Листинг 6.1: Получение спектрограммы гласных звуков

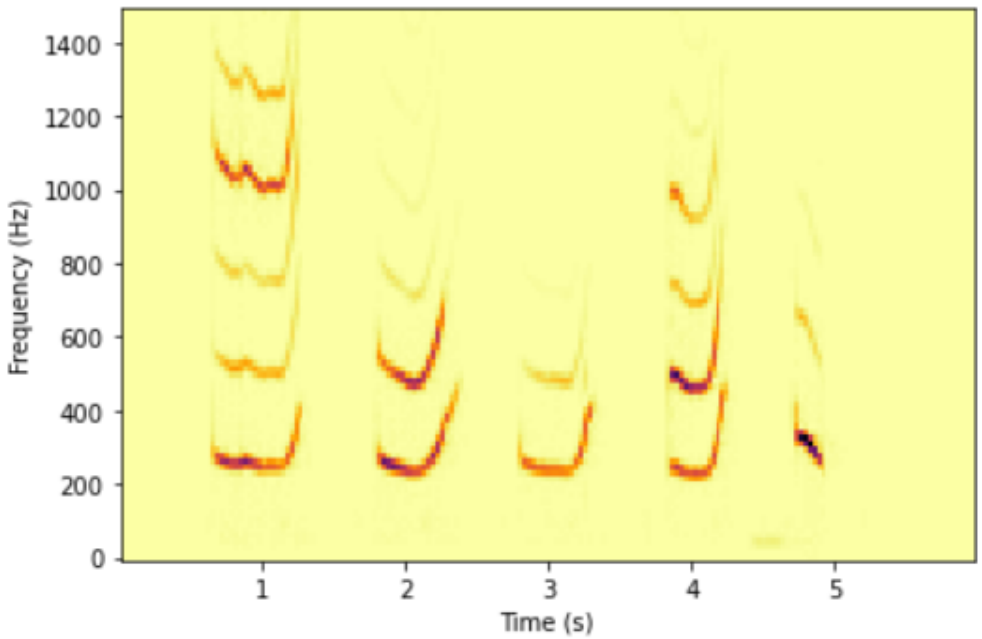

Рис. 6.1: Спектрограмма гласных звуков

Как мы можем видеть на рис.6.1, разные гласные звуки имеют разные частоты, так что различить гласные по спектру возможно, но очень трудно.

6.2 Спектры гласных звуков

Пики на спектрограмме называются формантами. Гласные звуки различаются соотношением амплитуд первых двух формант относительно основного тона.

Посмотрим на спектры каждого звука.

6.2.1 Звук а

```
segment = vowels.segment(start=0.5, duration=0.75)
segment.make_spectrum().plot(high=1000)
decorate(xlabel='Frequency (Hz)')
```

Листинг 6.2: Получение спектра звука а

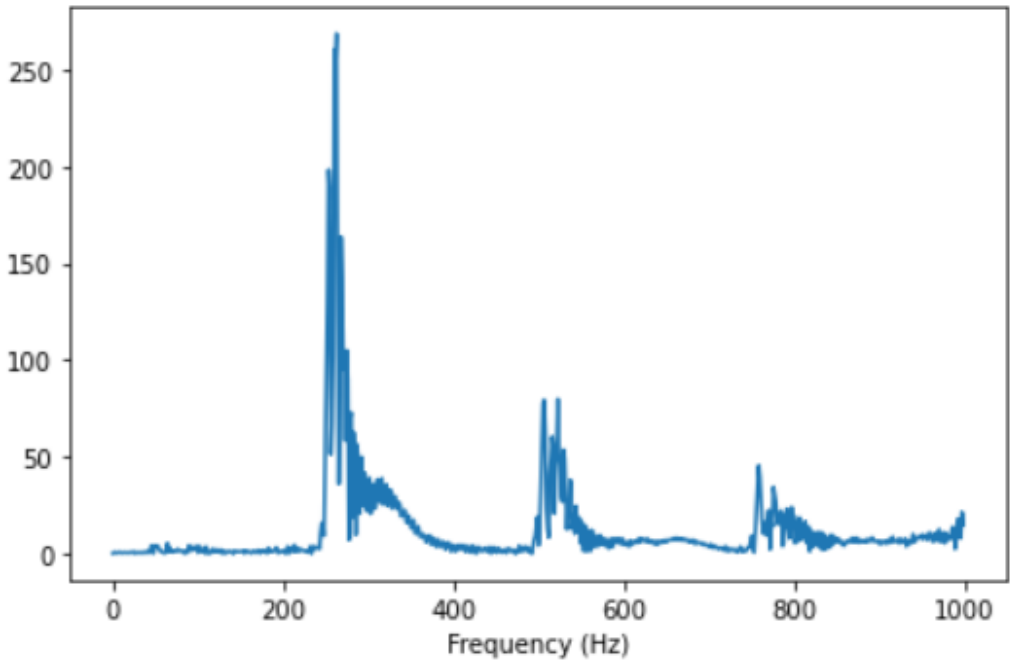

Рис. 6.2: Спектр звука а

На рис.6.2 видно, что основная частота находится между 200 и 300 Гц. Следующие самые высокие пики находятся между 500-600 Гц и 700-800 Гц соответственно.

6.2.2 Звук э

```
segment = vowels.segment(start=1.5, duration=0.9)
segment.make_spectrum().plot(high=1000)
decorate(xlabel='Frequency (Hz)')
```

Листинг 6.3: Получение спектра звука э

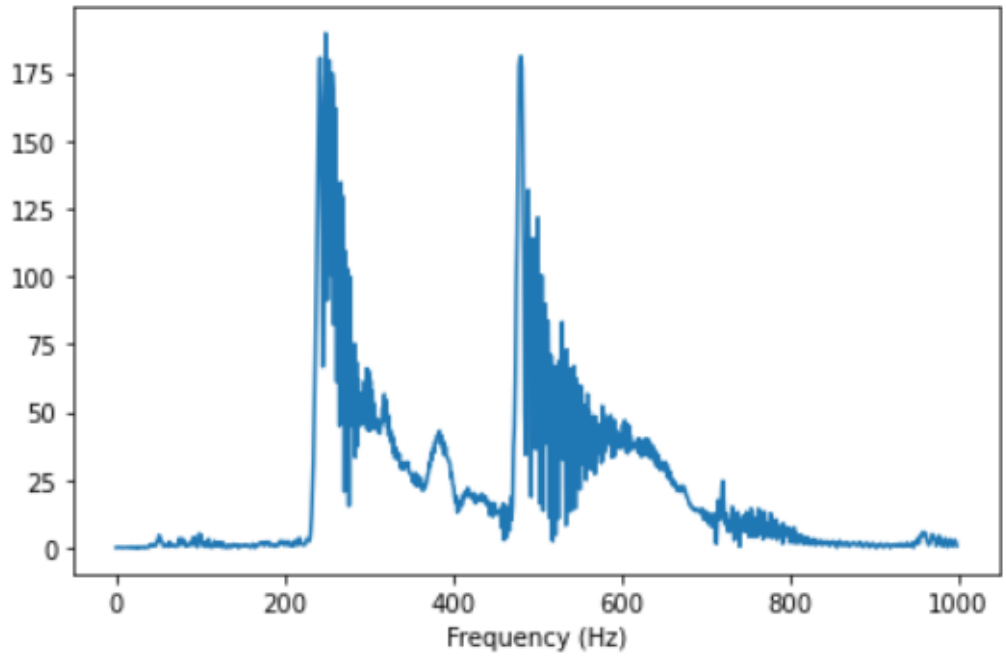

Рис. 6.3: Спектр звука э

На рис.6.3 мы видим два похожих пика на 300 и 500 Гц соответственно.

6.2.3 Звук и

```
segment = vowels.segment(start=2.7, duration=0.7)
segment.make_spectrum().plot(high=1000)
decorate(xlabel='Frequency (Hz)')
```

Листинг 6.4: Получение спектра звука и

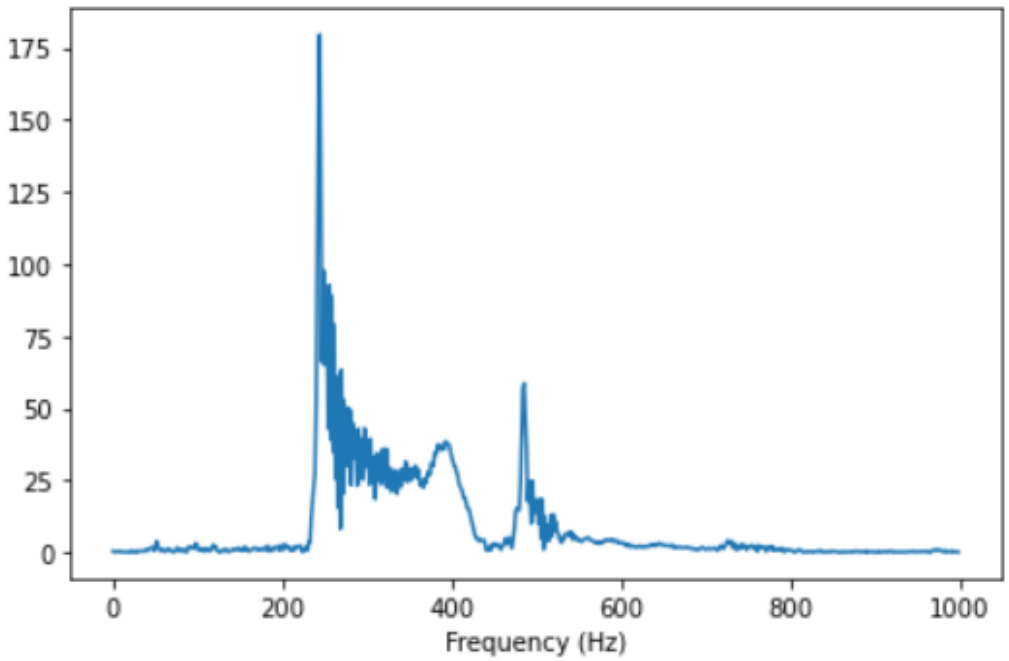

Рис. 6.4: Спектр звука и

На рис.6.4 мы видим основную частоту на 200, а следующий пик на 500 Гц.

6.2.4 Звук о

```
segment = vowels.segment(start=3.6, duration=0.8)
segment.make_spectrum().plot(high=1000)
decorate(xlabel='Frequency (Hz)')
```

Листинг 6.5: Получение спектра звука о

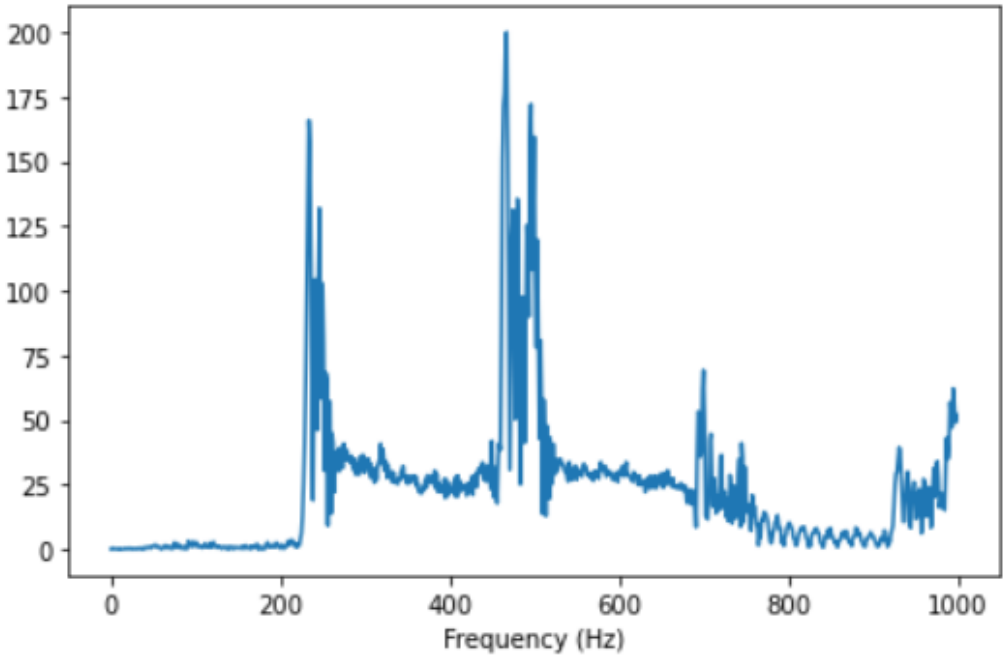

Рис. 6.5: Спектр звука о

Звук о имеет высокоамплитудную форманту около 500 Гц, даже выше основной частоты.

6.2.5 Звук у

```
segment = vowels.segment(start=4.5, duration=0.6)
segment.make_spectrum().plot(high=1000)
decorate(xlabel='Frequency (Hz)')
```

Листинг 6.6: Получение спектра звука у

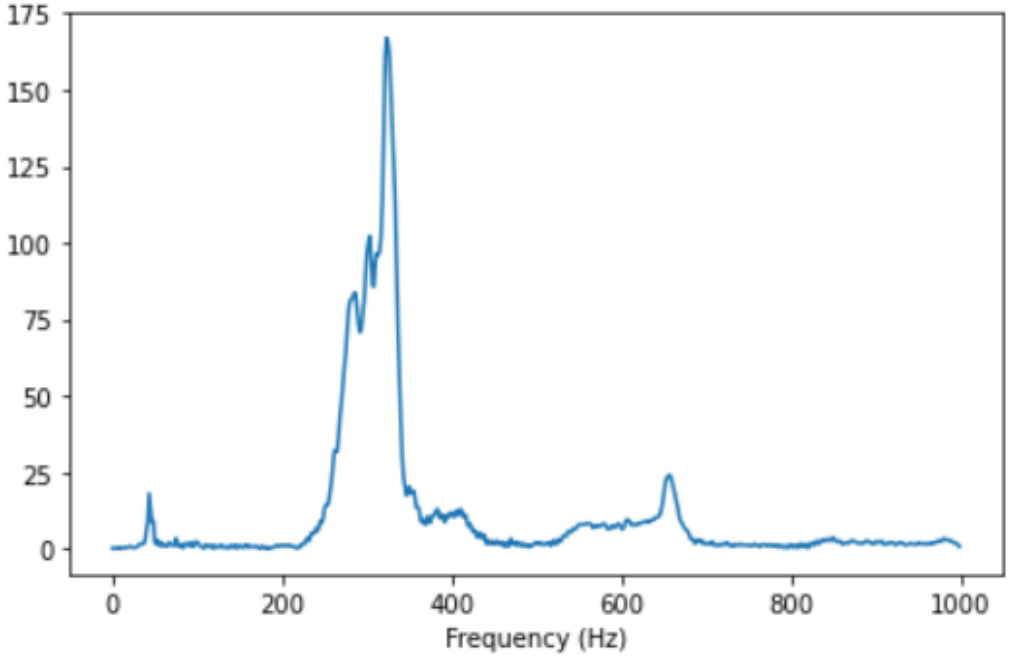

Рис. 6.6: Спектр звука у

Звук у имеет высокоамплитудную форманту около 300 Гц и не имеет высокочастотных составляющих.

Глава 7

Выводы

В результате выполнения данной работы мы познакомились с понятиями апериодических сигналов, частотные компоненты которых изменяются во времени. К ним относятся практически все сигналы. А также с чирпами - сигналами с переменной частотой.

Кроме того, мы научились работать со спектрограммами и окнами.

Спектрограмма - это способ визуализации кратковременных ПФ. У неё на оси x время, а на оси y частоты.

Окно - это функция, преобразующая апериодический сегмент во нечто похожее на периодическое.