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

Телекоммуникационные технологии

Отчёт по лабораторной работе №5

Работу выполнил:
Смирнов Л. Д.
Группа:
3530901/80202
Преподаватель:
Богач Н. В.

Санкт-Петербург
2021

Содержание

1. Теоретическая часть

В данной главе более подробно (по сравнению с предыдущей) рассматривается понятие корреляции, ее применение и виды. Корреляция — наличие в одной переменной информации о другой. Для оценки коррецляции часто используется коэффициент корреляции Пирсона, вычисляемый по формуле $\rho = \Sigma_i(x_i - \mu_x)(y_i - \mu_y)/N\sigma_x\sigma_y$. Принимаемы коэффициентом значения в промежутке от -1 до +1 показывают степень корреляции между переменными (прямая или обратная зависимость и их сила). Последовательная корреляция — корреляция между двумя идущими друг за другом элементами последовательности. Расчитывается путем вычисленя корреляции для исходного сигнала и его сдвинутой копии. Автокорреляция — корреляция для произвольных интревалов.

2. Выполнение работы

2.1. Упражнение 1

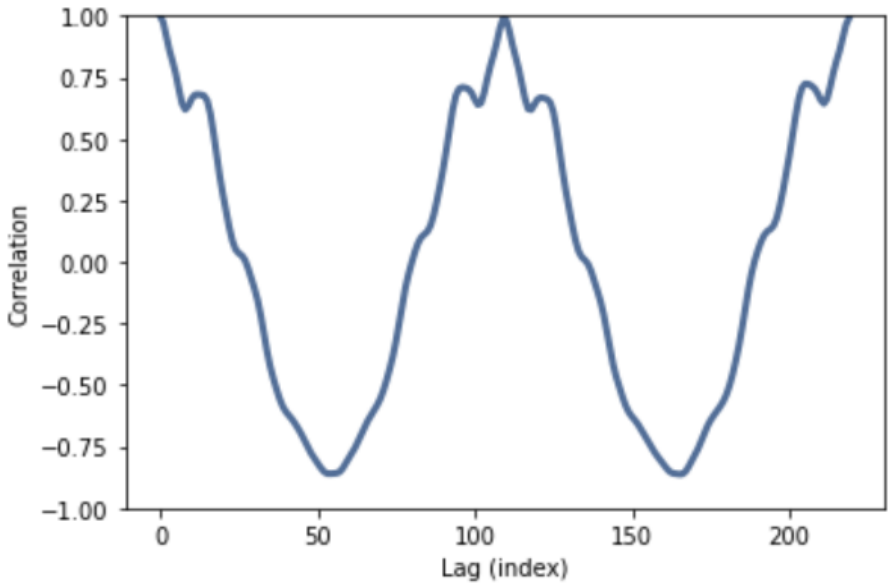
Для начала я взял запись вокального чирпа, который использовался выше.

```
wave = thinkdsp.read_wave('28042__bcjordan__voicedownbew.wav')
wave.normalize()
wave.make_audio()
```

Далее я выделил сегмент, начинающийся с 0.3 секунды длительностью 0.01 секунды и применил к ниму функцию автокорреляции.

```
segment = wave.segment(start=0.3, duration=0.01)
lags, corrs = autocorr(segment)
thinkplot.plot(lags, corrs)
thinkplot.config(xlabel='Lag (index)', ylabel='Correlation', ylim=[-1, 1])
```

Получился следующий график:



После этого я уточнил значение lag для изображенного пика и вычислил на основе этого значения частоту:

```
low, high = 100, 150
lag = np.array(corrs[low:high]).argmax() + low
period = lag / segment.framerate
frequency = 1 / period
frequency
```

Как итог получено значение 404.587. После этого я сделал еще 2 аналогичных измерения для сегментов со стартовыми моментами 0.5 и 0.7 секунды. Полученные значения 364.463 и 347.244 позволяют сказать, что частота со временем (и соответственно высота тона) уменьшается.

2.2. Упражнение 2

Для начала я согласно заданию инкапсулировал код из данной главы в отдельную функцию:

```
def estimate_fundamental(segment, low=70, high=150):
    lags, corrs = autocorr(segment)
    lag = np.array(corrs[low:high]).argmax() + low
    period = lag / segment.framerate
    frequency = 1 / period
    return frequency
```

Применять ее я буду для того же вокального чирпа, который рассматривался в предыдущем пункте:

```
wave.make_spectrogram(2048).plot(high=4200)
thinkplot.config(xlabel='Time (s)',
                     ylabel='Frequency (Hz)',
                     xlim=[0, 1.4],
                     ylim=[0, 4200])
```

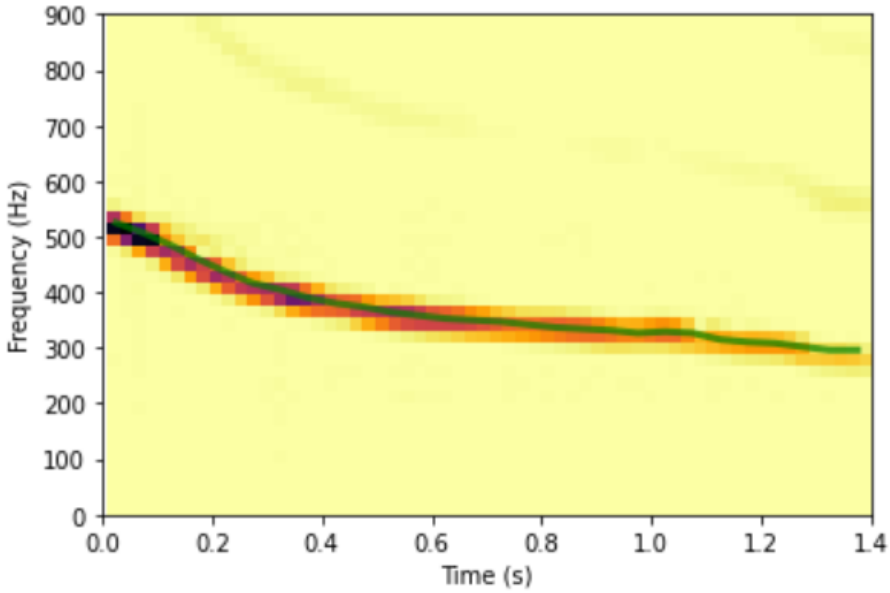
Далее я применил эту функцию, чтобы пройтись по сегменту звука с заданным шагом и заполнить массивы временных точек (середины отрезков, заданных шагами) и соответствующих значений частот, вычисленных функцией estimate_fundamental.

```
step = 0.05
starts = np.arange(0.0, 1.4, step)

ts = []
freqs = []

for start in starts:
    ts.append(start + step/2)
    segment = wave.segment(start=start, duration=duration)
    freq = estimate_fundamental(segment)
    freqs.append(freq)
```

Как итог выведем кривую для отслеживания высоты тона:

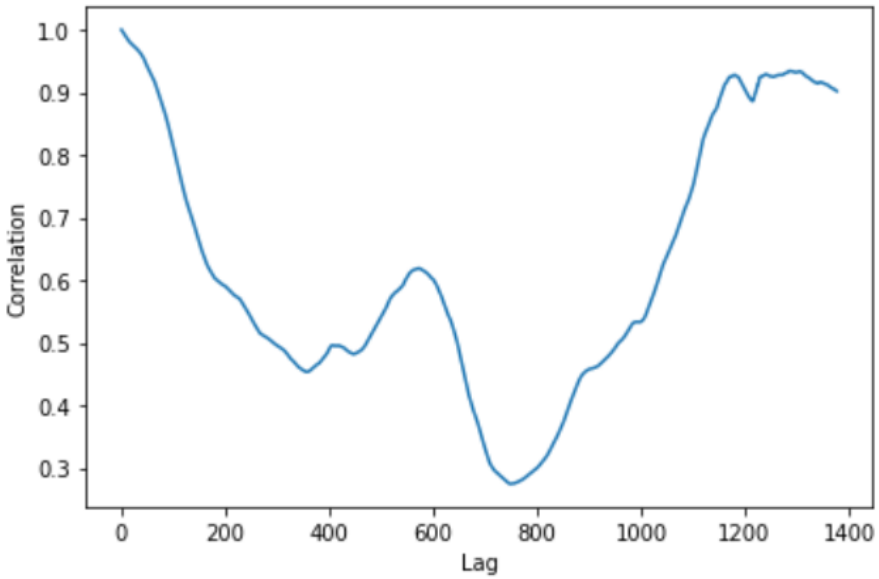
```
wave.make_spectrogram(2048).plot(high=900)
thinkplot.plot(ts, freqs, color='green')
thinkplot.config(xlabel='Time (s)',
                     ylabel='Frequency (Hz)',
                     xlim=[0, 1.4],
                     ylim=[0, 900])
```

Получено довольно четкое изображение:



2.3. Упражнение 3

Для этого упражнения я взял отчет по изменению цены биткоина за те же даты, что и в предыдущей лабораторной. Собственно я прочитал .csv файл, сделал на его основе волну, как и в предыдущей лабораторной, и применил к ней автокорреляционную функцию, использованную уже неоднократно в этой главе. Получился вот такой график корреляции:



На его основе сложно ответить на вопрос о том, как быстро убывают значения корреляции, так как они не только убывают, но и возрастают, причем с разной скоростью. Полный код для этого упражнения приведен ниже.

```
df = pd.read_csv('BTC_USD_2013-10-01_2021-04-19-CoinDesk.csv',
                 parse_dates=[0])

ys = df['Closing Price (USD)']
ts = df.index
wave = Wave(ys, ts, framerate=1)
wave.plot()
decorate(xlabel='Time (days)',
         ylabel='Price')

lags, corrs = autocorr(wave)
plt.plot(lags, corrs)
decorate(xlabel='Lag',
         ylabel='Correlation')
```

2.4. Упражнение 4

В качестве выполнения данного упражнения, как и указывалось в книге, я посмотрел прикрепленный ролик на YouTube и изучил примеры, приведенные в файле saxophone.ipynb для разных фрагментов записи.

3. Выводы

В ходе выполнения данной лабораторной работы я лучше изучил понятие корреляции, а так же применение ее видов (последовательной и атокорреляции) для исследования сигналов и тонов звука.