Санкт-Петербургский государственный политехнический
университет Петра Великого

**Высшая школа интеллектуальных систем и
суперкомпьютерных технологий**

Лабораторная работа

Автокорреляция

Работу выполнил студент
3-го курса, группа 3530901/80201
Сахибгареев Рамис Ринатович

Преподаватель:
Богач Наталья Владимировна

Санкт-Петербург 2021

Contents

1 Part 1: Vocal chirp's pitch estimation **5**

2 Part 2: Signal pitch plot calculation **7**

3 Part 3: GREATEST CRYPTOCURRENCY OF ALL TIME (aka. bitcoin). PART 2 **9**

4 Part 4: Missing fundamental **11**

5 Conclusion **17**

List of Figures

1 Vocal signal lag correlation 6

2 Voice signal spectrogram 7

3 Voice signal spectrogram and estimated frequency 8

4 Bitcoin chart 9

5 Bitcoin's autocorrelation 10

6 Saxophone record spectrogram 11

7 Saxophone record spectrum 12

8 Saxophone correlation 13

9 Saxophone spectrum without 464 Hz harmonic 14

10 Saxophone correlation plot filtered 15

11 Saxophone spectrum plot filtered hard 16

12 Saxophone correlation plot filtered hard 16

Listings

1 sserial-corr and autocorr functions 5

2 Vocal signal lag correlation estimation 5

3 Log spectrum 7

4 Segment frequency estimator 7

5 Estimation check code 8

6 Bitcoin to a wave 9

7 Bitcoin autocorrelation estimation 9

1 Part 1: Vocal chirp's pitch estimation

We cannot define pitch of the signal at the moment, because of low frequency resolution. Also we cannot increase window size because signal is non-periodic. But we can use auto-correlation. Let's write some functions, that allows us to find windows correlation.

```
from thinkdsp import *
import thinkplot

def serial_corr(wave, lag=1):
    N = len(wave)
    y1 = wave.ys[lag:]
    y2 = wave.ys[:N-lag]
    corr = np.corrcoef(y1, y2)[0, 1]
    return corr
def autocorr(wave):
    lags = np.arange(len(wave.ys)//2)
    corrs = [serial_corr(wave, lag) for lag in lags]
    return lags, corrs

```

Listing 1: sserial-corr and autocorr functions

After defining these functions we can apply them to the vocal signal.

```
wave = read_wave('data/voice.wav')
segment = wave.segment(start=0, duration=0.01)
lags, corrs = autocorr(segment)
lag = np.array(corrs[90:110]).argmax() + 90
thinkplot.plot(lags, corrs , color = 'blue')
thinkplot.config(xlabel='Lag', ylabel='Correlation')
print(segment.framerate / lag)

```

Listing 2: Vocal signal lag correlation estimation

```
490.0
```

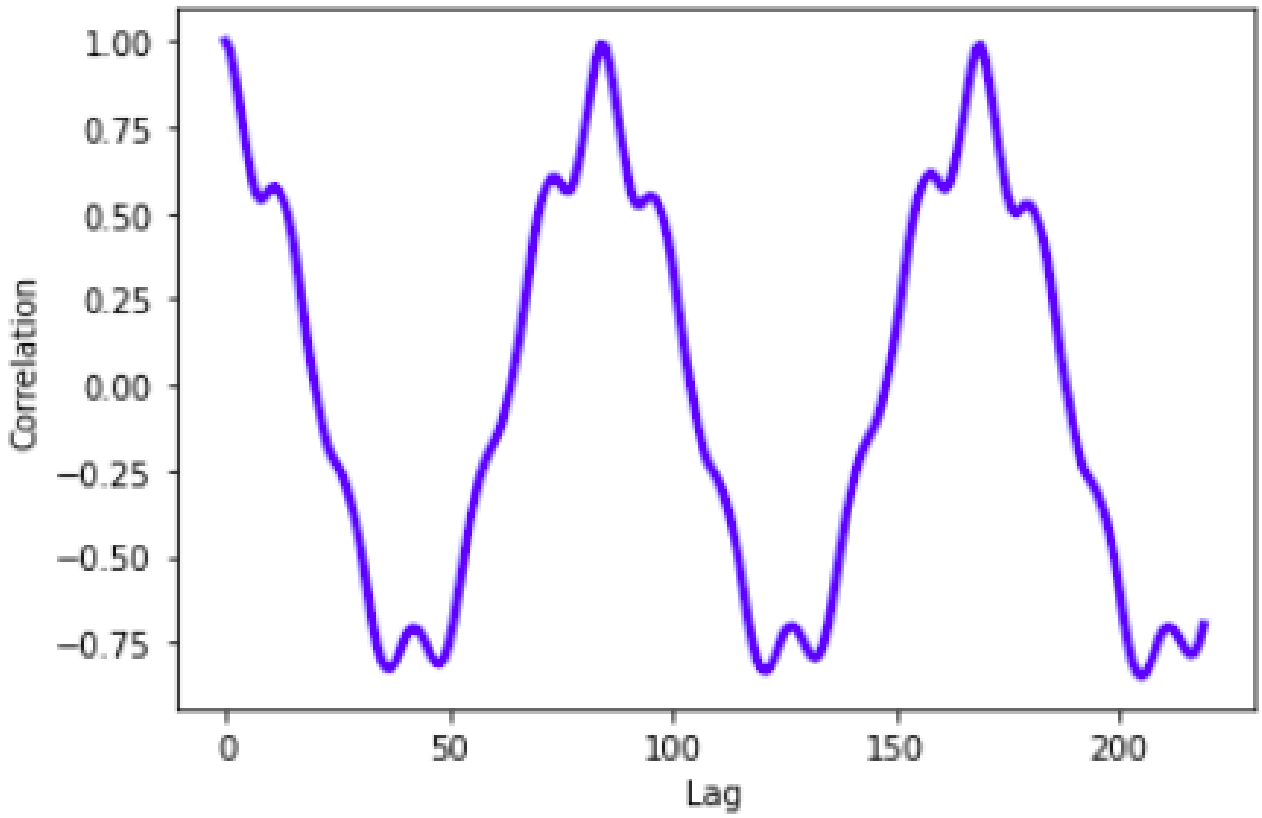

Figure 1: Vocal signal lag correlation

2 Part 2: Signal pitch plot calculation

To automate process of the previous part, let's write a little function, that returns estimated frequency.

```
    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

```

Listing 3: Log spectrum

Now let's build the signal's spectrogram. We can clearly see a main spike.

```
    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

```

Listing 4: Segment frequency estimator

```
525.0
```

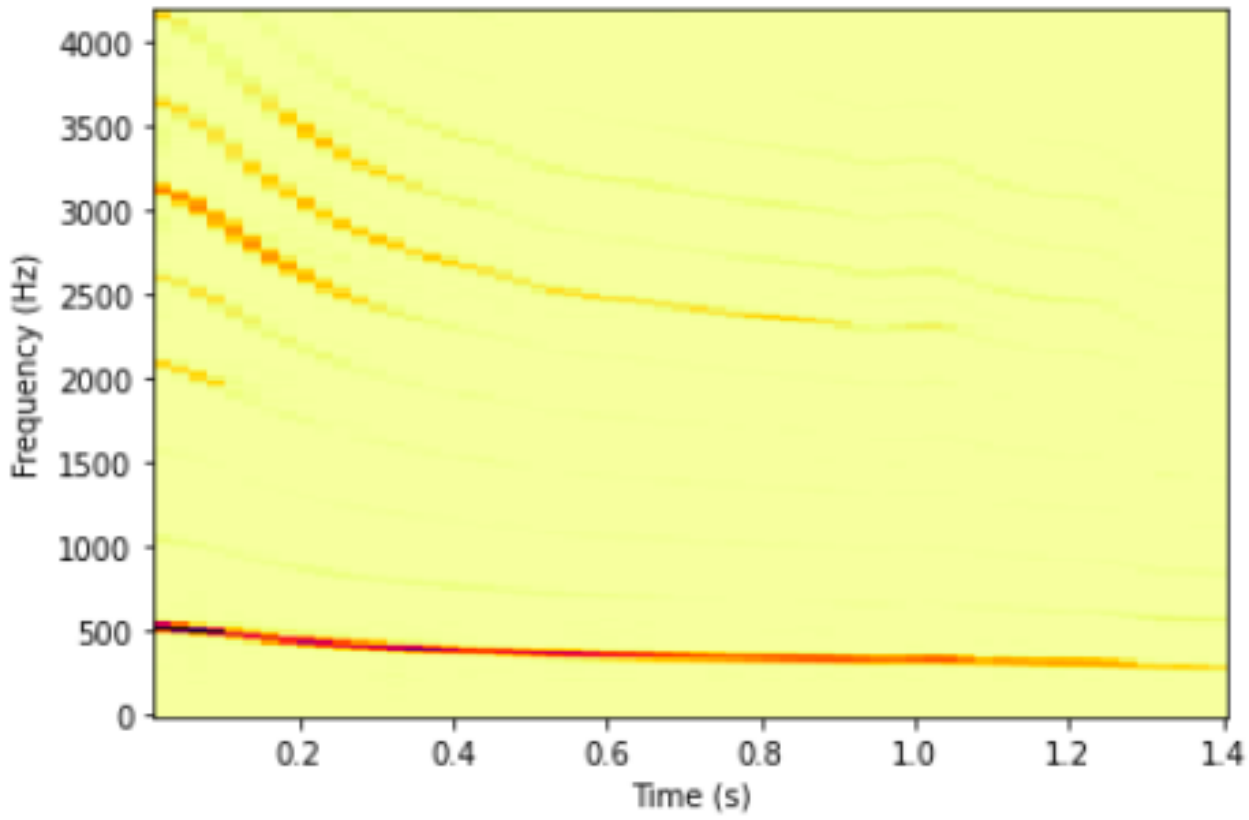

Figure 2: Voice signal spectrogram

We can estimate accuracy of our function by splitting the signal into the small parts and estimating the frequency of each part. After we can apply this values to the plot and check, how estimated frequency close to the real.

```
    starts = np.arange(0.0, 1.4, 0.05)
    ts = []
    freqs = []
    for start in starts:
        ts.append(start + 0.05/2)
        segment = wave.segment(start=start , duration=0.01)
        freq = estimate_fundamental(segment)
        freqs.append(freq)
    wave.make_spectrogram(2048).plot(high=1000)
    plt.plot(ts, freqs , color='aqua', linewidth=4)
    decorate(xlabel='Time (s)', ylabel='Frequency (Hz)')

```

Listing 5: Estimation check code

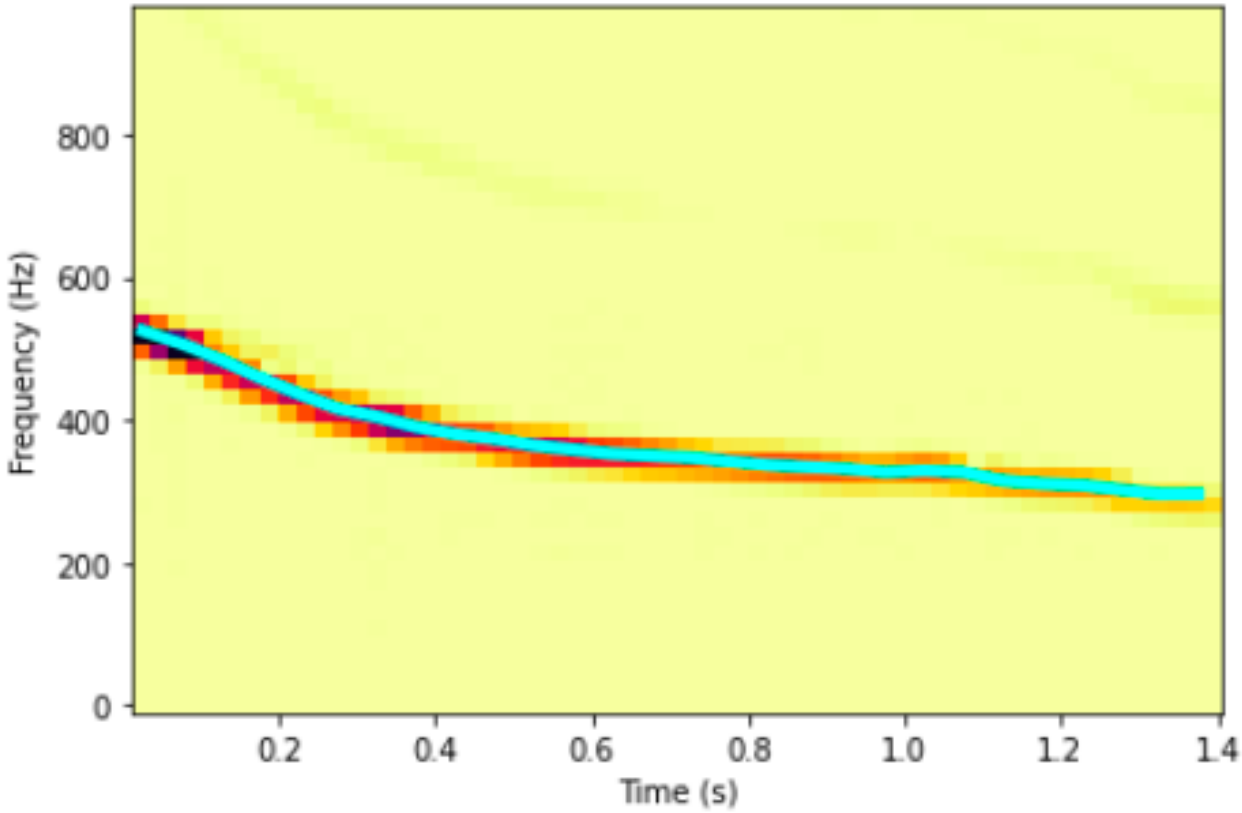

Figure 3: Voice signal spectrogram and estimated frequency

We can clearly see, that estimated frequency follows main spectrogram spike.

3 Part 3: GREATEST CRYPTOCURRENCY OF ALL TIME (aka. bitcoin). PART 2

In this part we need to try to apply autocorrelation method to the bitcoin chart.

Code for parsing the .csv file is the same, as it was in the previous lab.

```
data = pd.read_csv('data/bitcoin.csv')
price = data['Closing Price (USD)']
count = data.index
wave = Wave(price,count,framerate=1)
wave.plot()
decorate(xlabel='Day')

```

Listing 6: Bitcoin to a wave

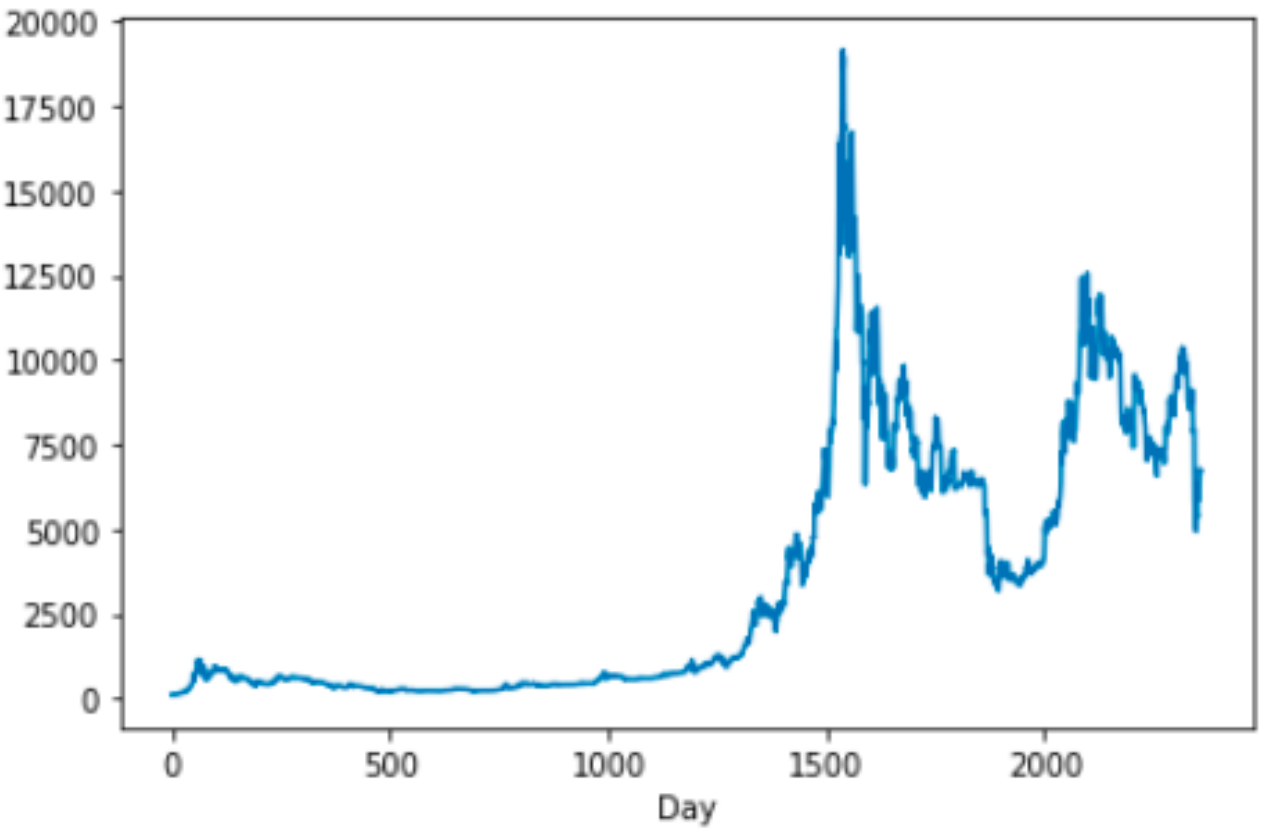

Figure 4: Bitcoin chart

Let's look, how looks a logarithmic graph of a spectrum's power.

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

```

Listing 7: Bitcoin autocorrelation estimation

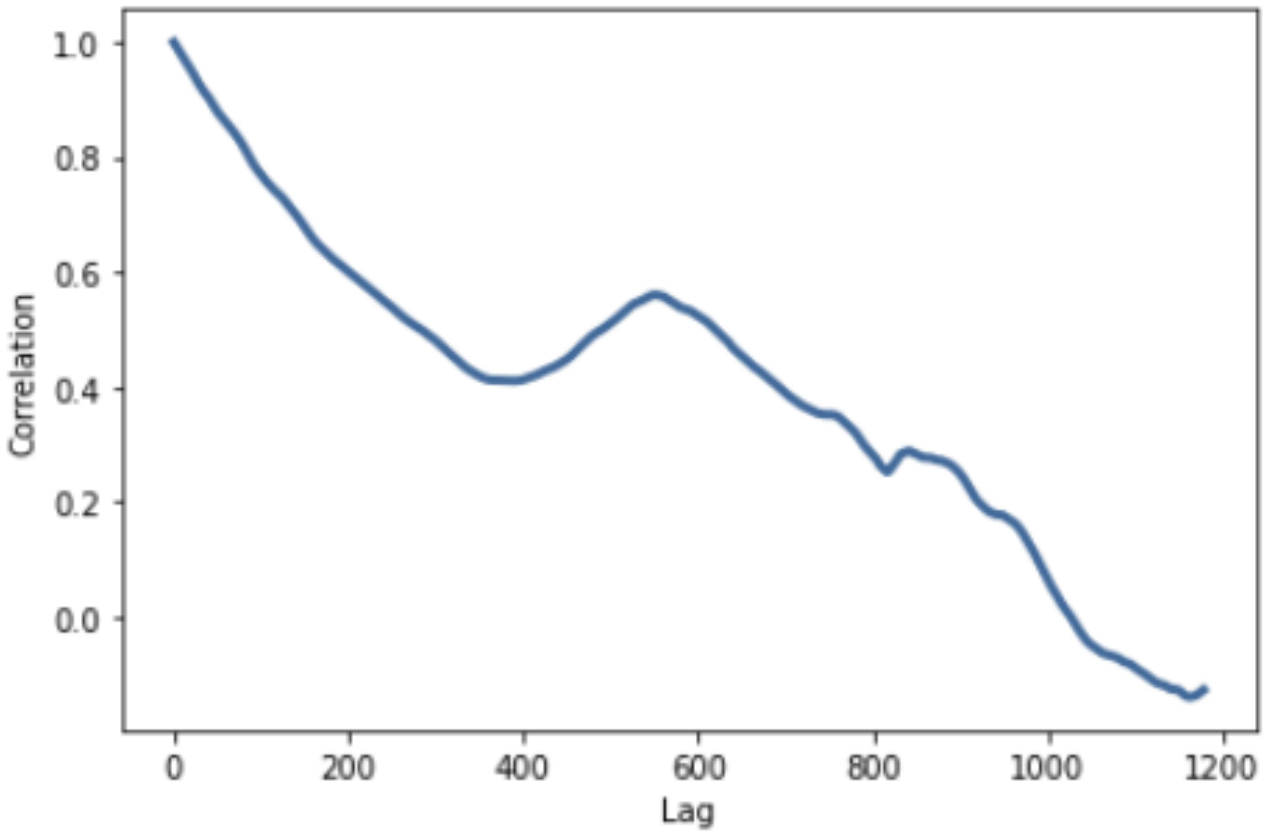

Figure 5: Bitcoin's autocorrelation

As we can see, bitcoin has no signs of a periodic signal, so we cannot predict its values.

4 Part 4: Missing fundamental

In this part we need to explore Missing Fundamental effect.

To do it we can use existing notebook, that uses saxophone record to demonstrate this effect.

Firstly, let's take a look on the record spectrogram.

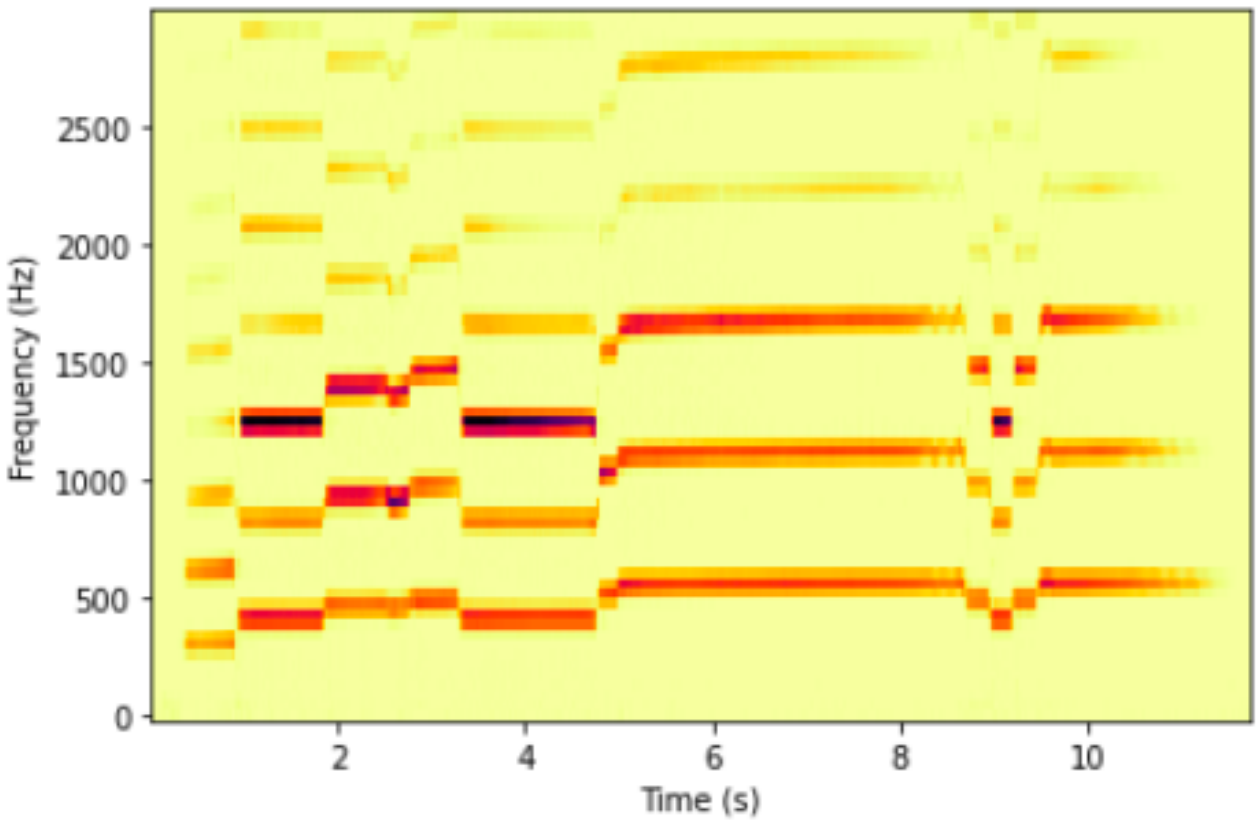

Figure 6: Saxophone record spectrogram

We can clearly see spikes, spectrum of the stable segment also confirms it.

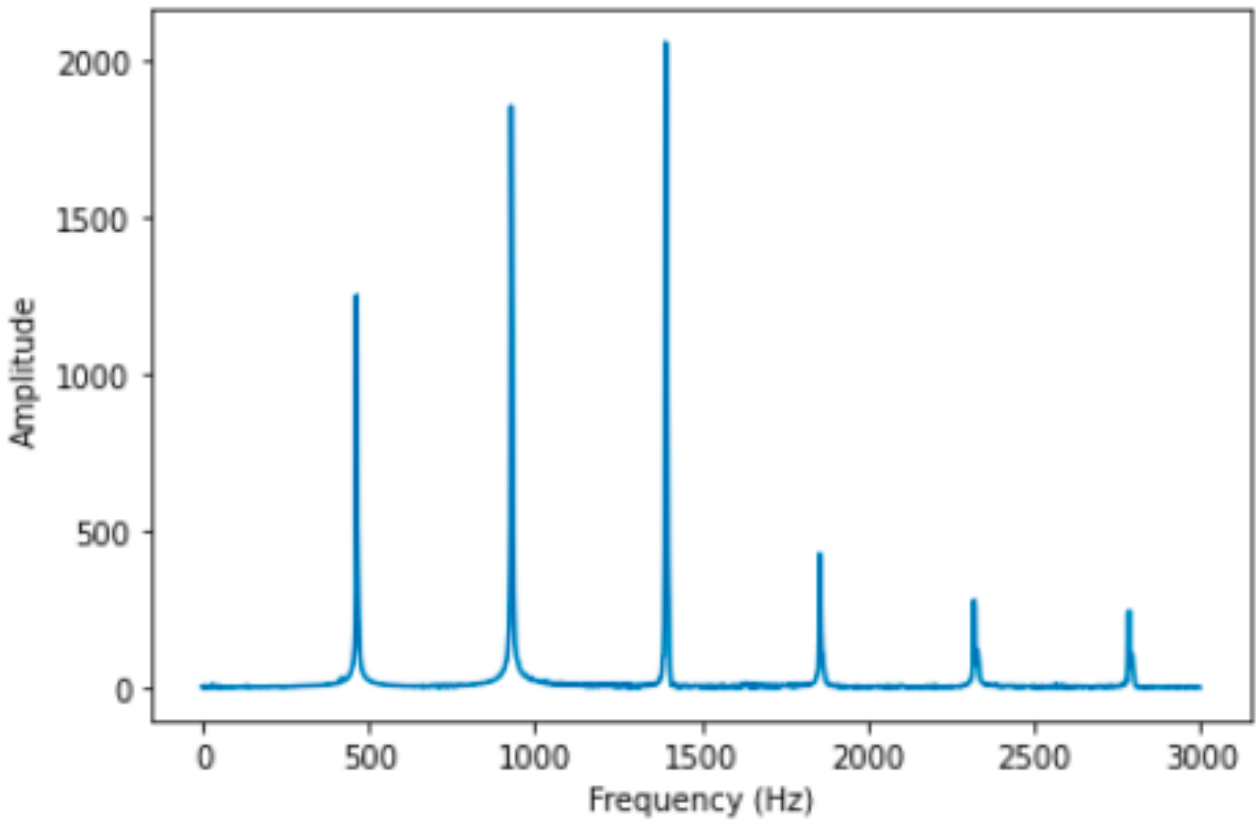

Figure 7: Saxophone record spectrum

Triangle signal with same fundamental frequency hears differently - with more lower pitch. Let's take a look on the correlation plot.

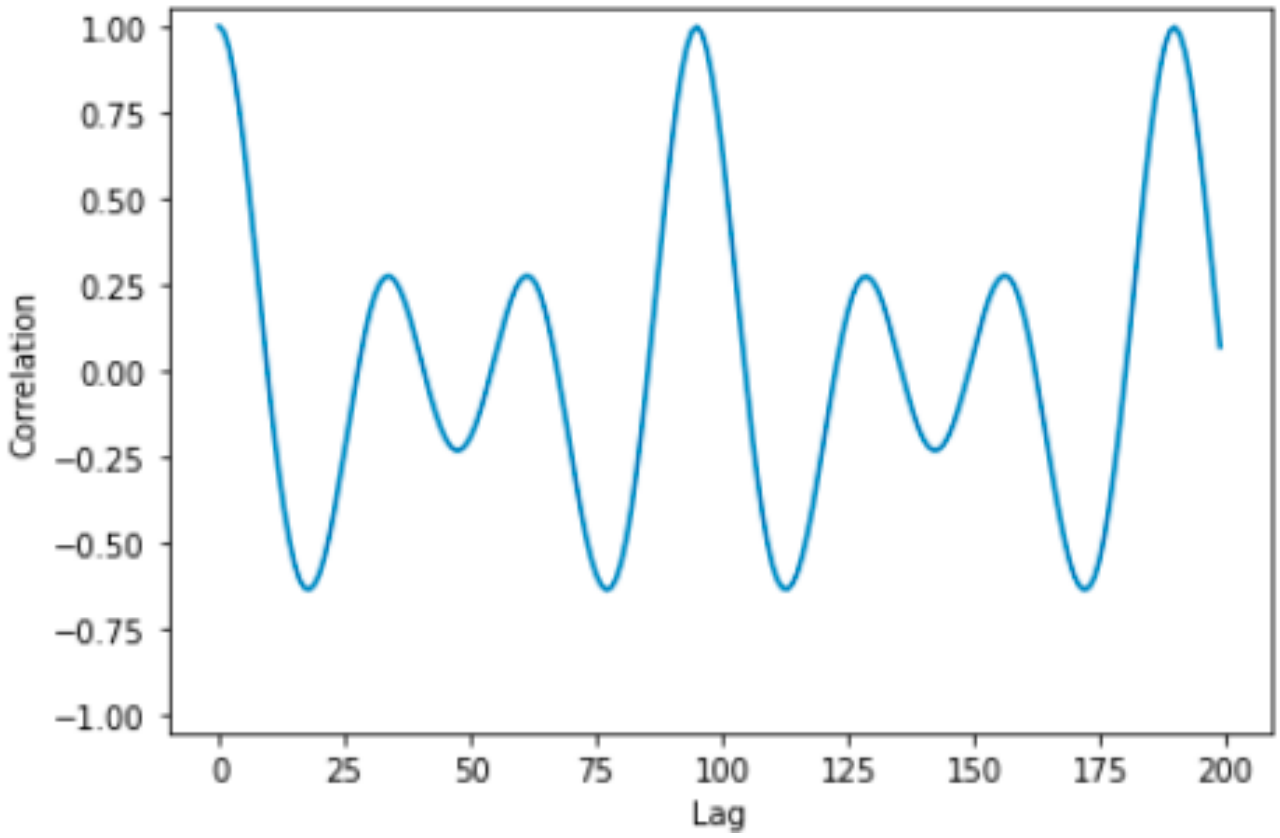

Figure 8: Saxophone correlation

We can clearly see spike on the frequency of 464 Hz. But what would happen, if we cutoff this frequency from the signal? Let's check out.

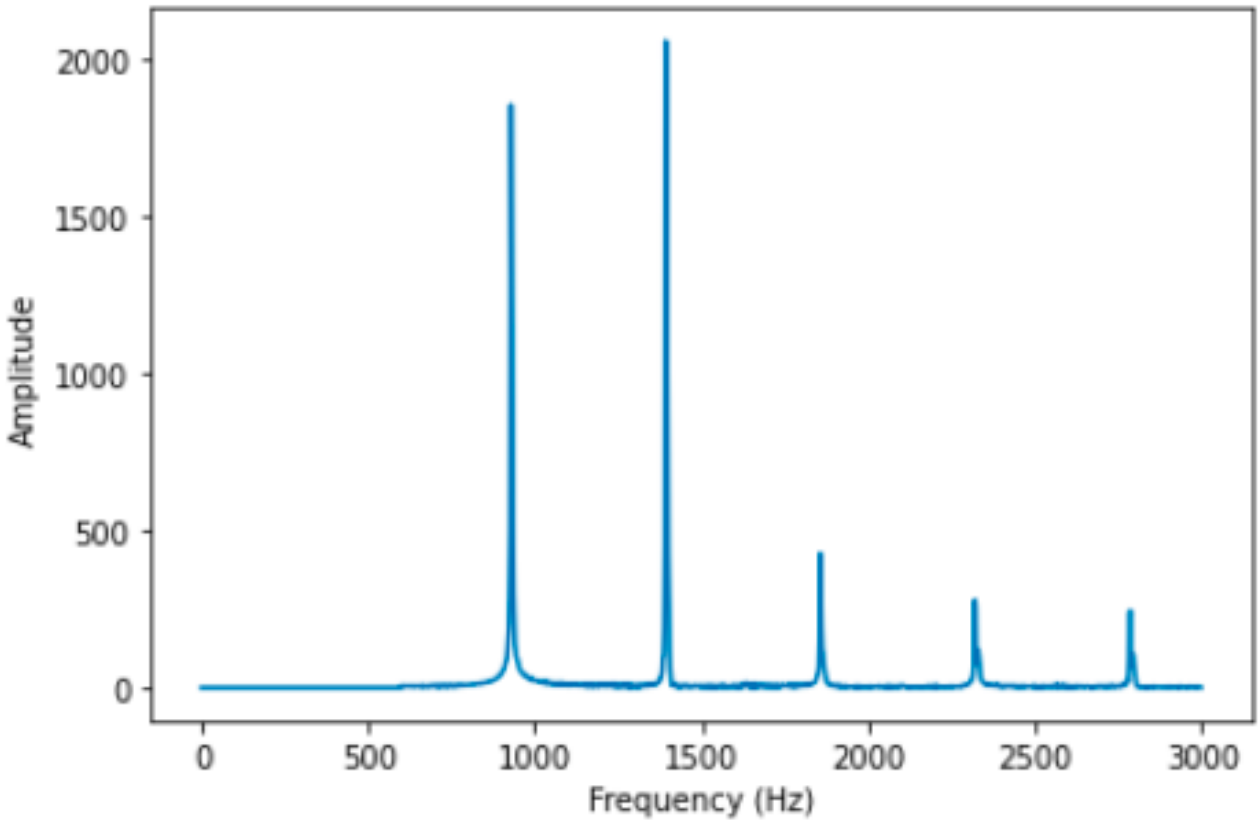

Figure 9: Saxophone spectrum without 464 Hz harmonic

However, If we listen new record, we won't find pitch difference. Also if we take a look on the correlation plot, we won't find any difference too.

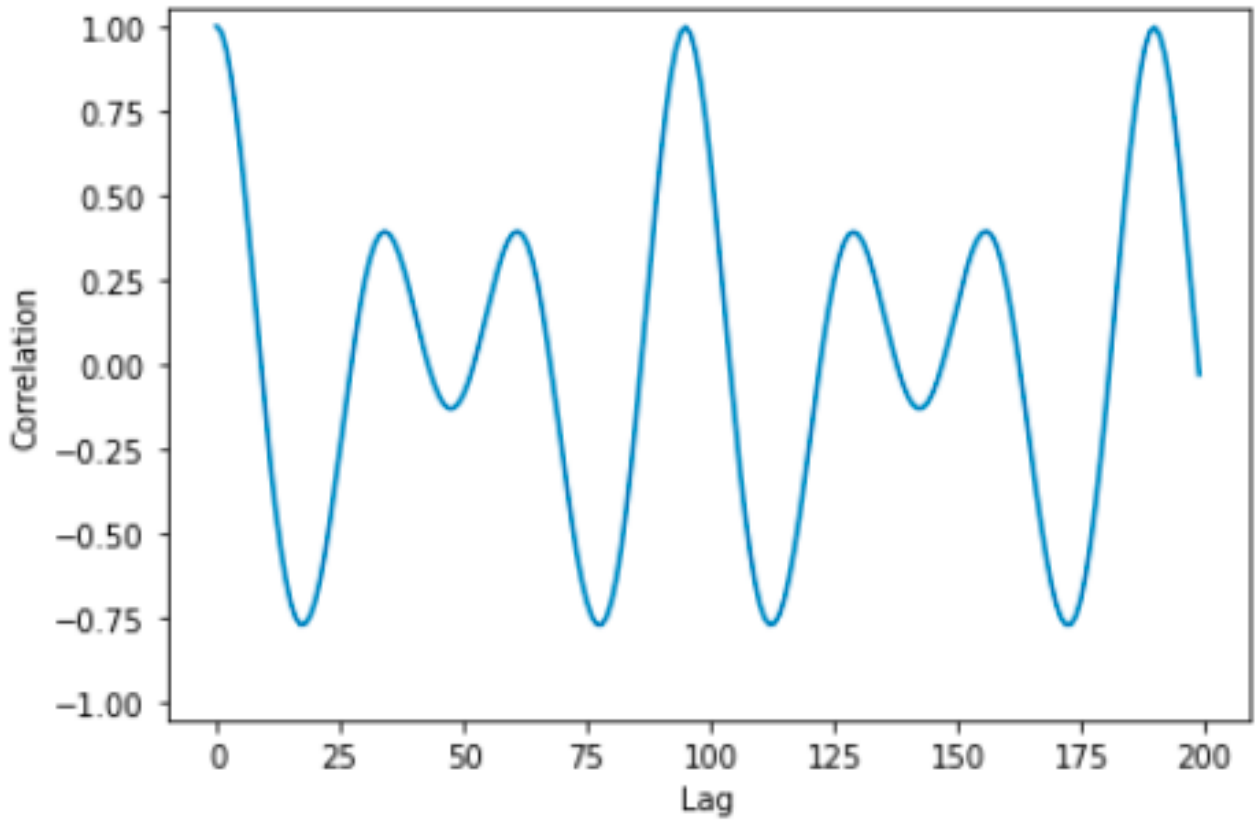

Figure 10: Saxophone correlation plot filtered

This effect called missing fundamental - even when we removed fundamental harmonic it remains fundamental frequency of the signal. But if we remove its high frequency factors from the signal too, then no traces of the original signal would left.

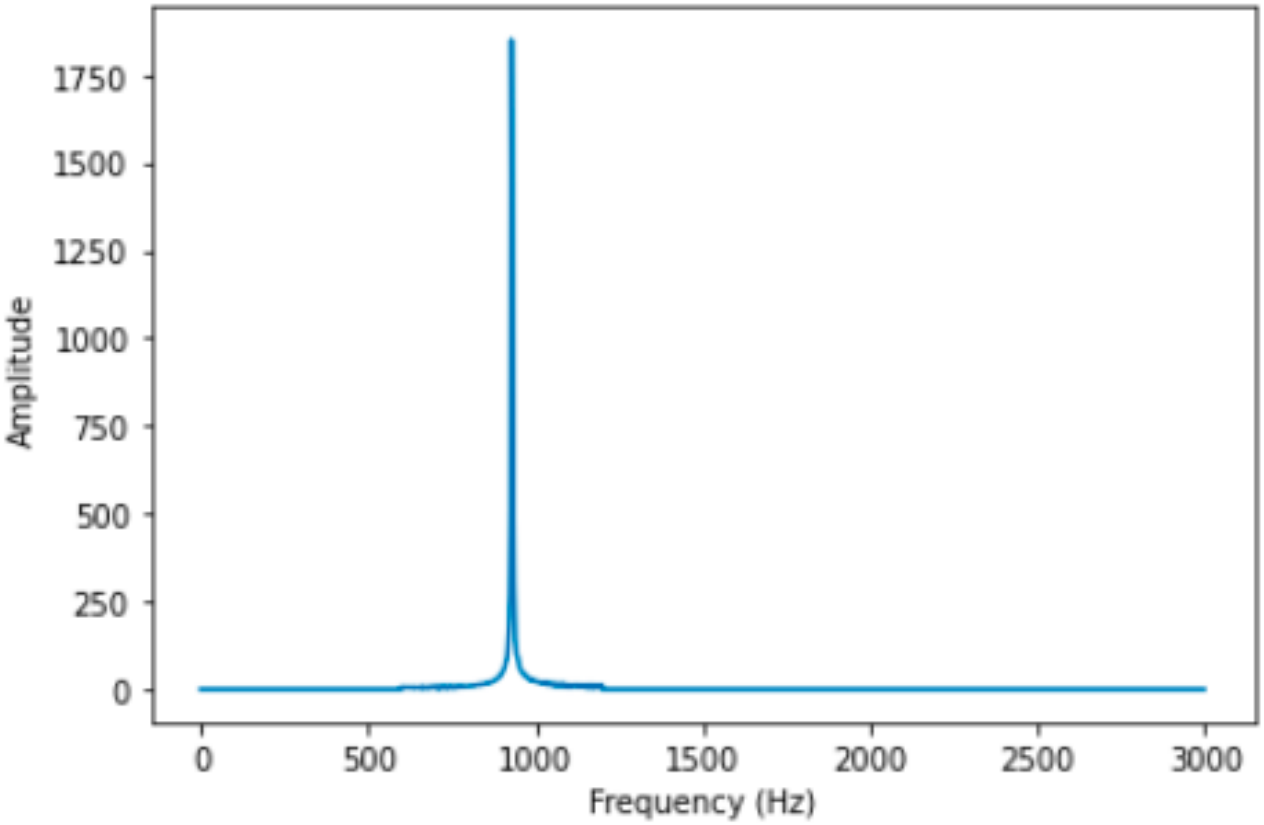

Figure 11: Saxophone spectrum plot filtered hard

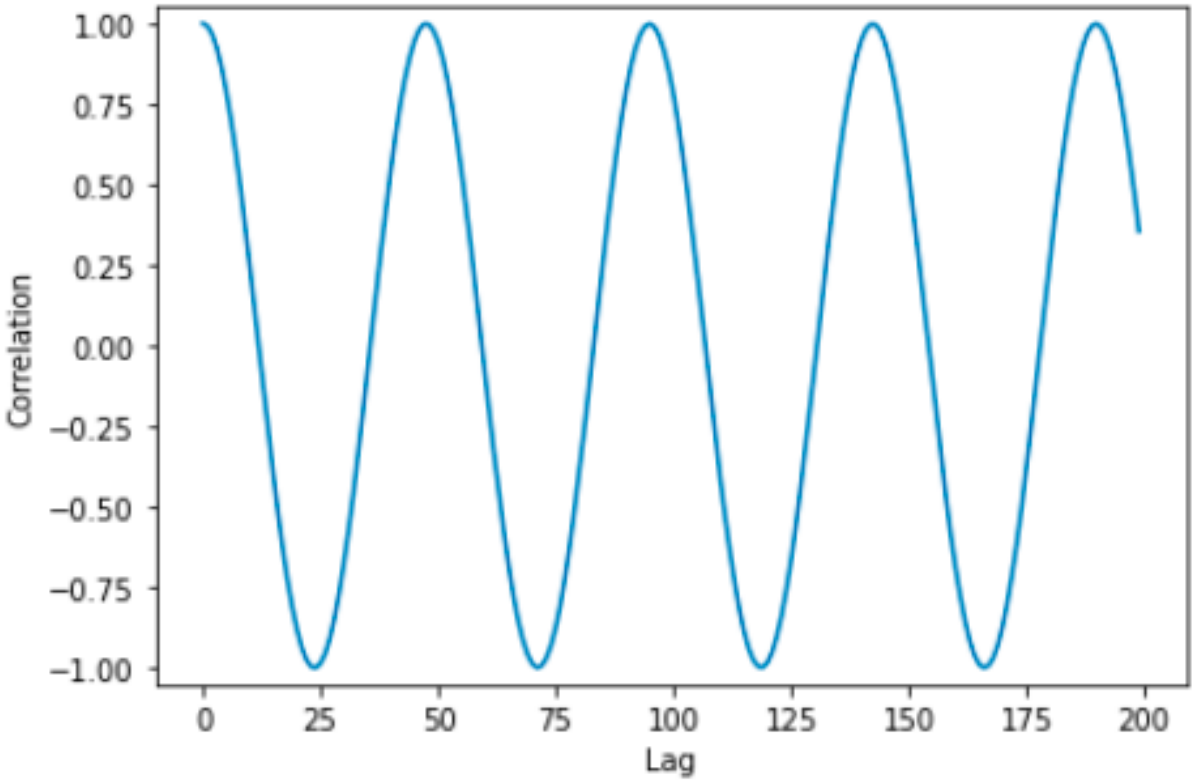

Figure 12: Saxophone correlation plot filtered hard

5 Conclusion

We've learned, what autocorrelation is, how we can compute it and how it affects the signals. Autocorrelation can be used to estimate fundamental frequency of the signal or say, that signal is chaotic and there is no period. Autocorrelation is a reason of missing fundamental effect, when out ears can hear phantom base frequency even if signal doesn't have a harmonic with this frequency.