Ivanov_lab11

March 28, 2020

Глава 11 (Иванов Илья, группа 3530901/70203)

[1]:
```
from __future__ import print_function, division

import thinkdsp
import thinkplot

import numpy as np

import warnings
warnings.filterwarnings('ignore')

PI2 = 2 * np.pi

np.set_printoptions(precision=3, suppress=True)
%matplotlib inline
```

1 Упражнение 11.1.

The code in this chapter is in `chap11.ipynb`. Read through it and listen to the examples.

`chap11.ipynb` был просмотрен и разобран.

2 Упражнение 11.2.

Chris "Monty" Montgomery has an excellent video called "D/A and A/D | Digital Show and Tell"; it demonstrates the Sampling Theorem in action, and presents lots of other excellent information about sampling. Watch it at https://www.youtube.com/watchv=cIQ9IXSUzuM.

Видео было просмотрено и разобрано.

3 Упражнение 11.3.

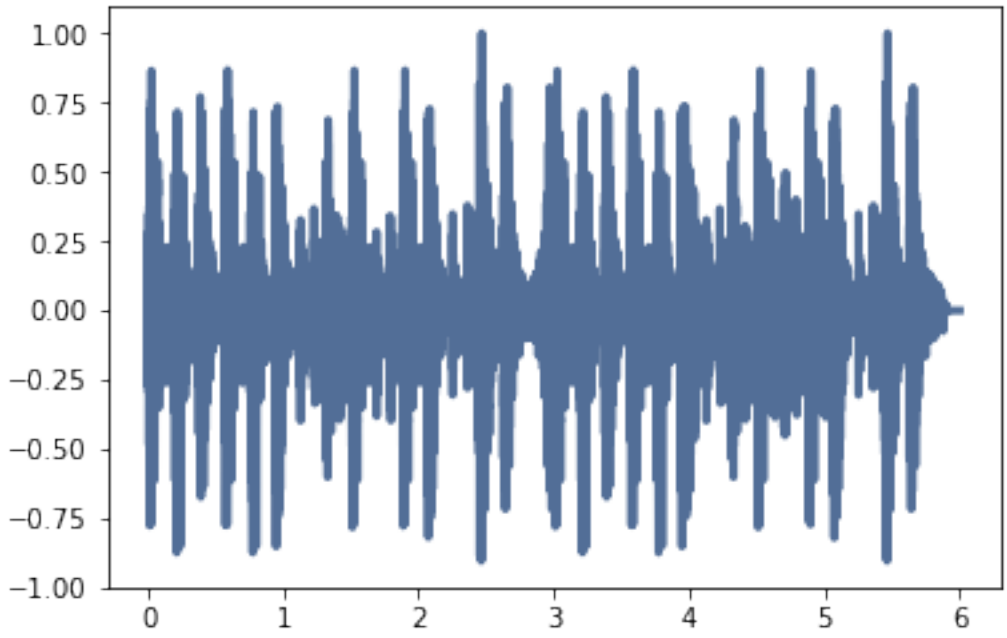
As we have seen, if you sample a signal at a too low framerate, frequencies above the folding frequency get aliased. Once that happens, it is no longer possible to filter out these components, because they are indistinguishable from lower frequencies.

It is a good idea to filter out these frequencies before sampling; a low-pass filter used for this purpose is called an anti-aliasing filter.

Returning to the drum solo example, apply a low-pass filter before sampling, then apply the low-pass filter again to remove the spectral copies introduced by sampling. The result should be identical to the filtered signal.

Используем барабанное соло из `chap11.ipynb`:

[2]:
```
wave = thinkdsp.read_wave('263868__kevcio__amen-break-a-160-bpm.wav')
wave.normalize()
wave.plot()
```



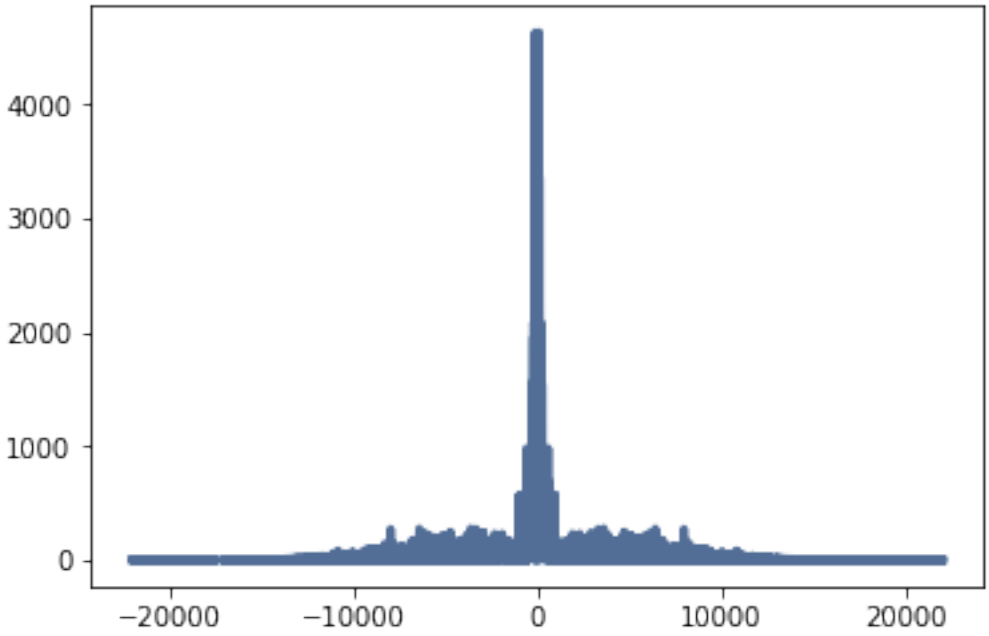
Это сигнал с частотой дискретизации 44100 Гц. Прослушаем его:

[3]:
```
wave.make_audio()
```

[3]:
```
<IPython.lib.display.Audio object>
```

Получим его спектр:

[4]:
```
spectrum = wave.make_spectrum(full=True)
spectrum.plot()
```

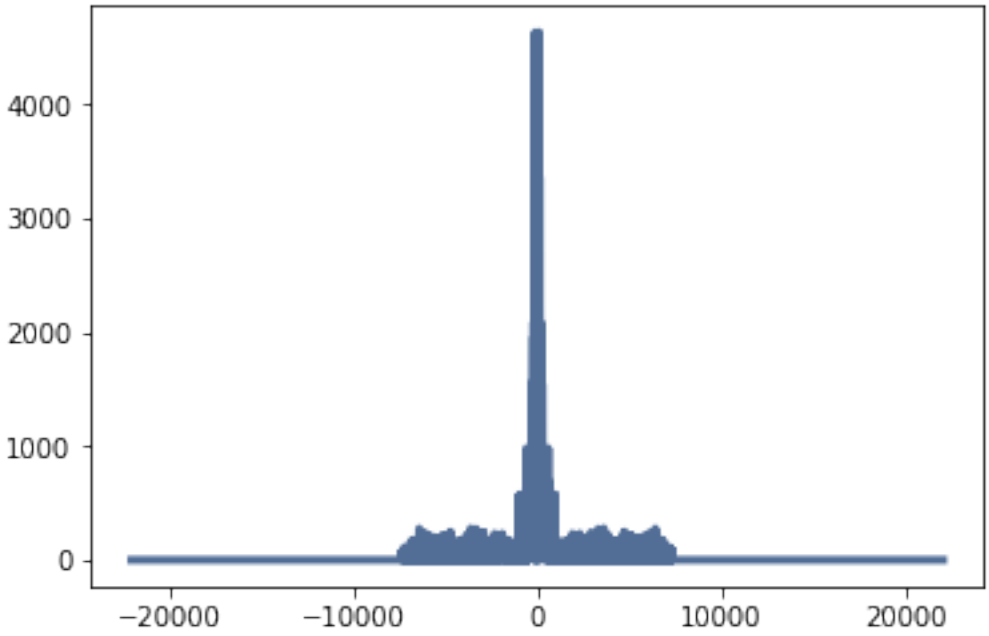


Уменьшим частоту дискретизации в 3 раза:

[5]:
```
factor = 3
framerate = wave.framerate / factor
cutoff = framerate / 2 - 1
```

До выборки применим фильтр сглаживания для удаления частот выше новой частоты заворота, которая равна частоте дискретизации / 2:

[6]:
```
spectrum.low_pass(cutoff)
spectrum.plot()
```



Прослушаем отфильтрованную версию:

[7]:
```
filtered = spectrum.make_wave()
filtered.make_audio()
```

[7]:
```
<IPython.lib.display.Audio object>
```

Звучит вполне неплохо.

Функция, имитирующая процесс выборки:

[8]:
```
def sample(wave, factor):
    ys = np.zeros(len(wave))
    ys[::factor] = wave.ys[::factor]
    return thinkdsp.Wave(ys, framerate=wave.framerate)
```

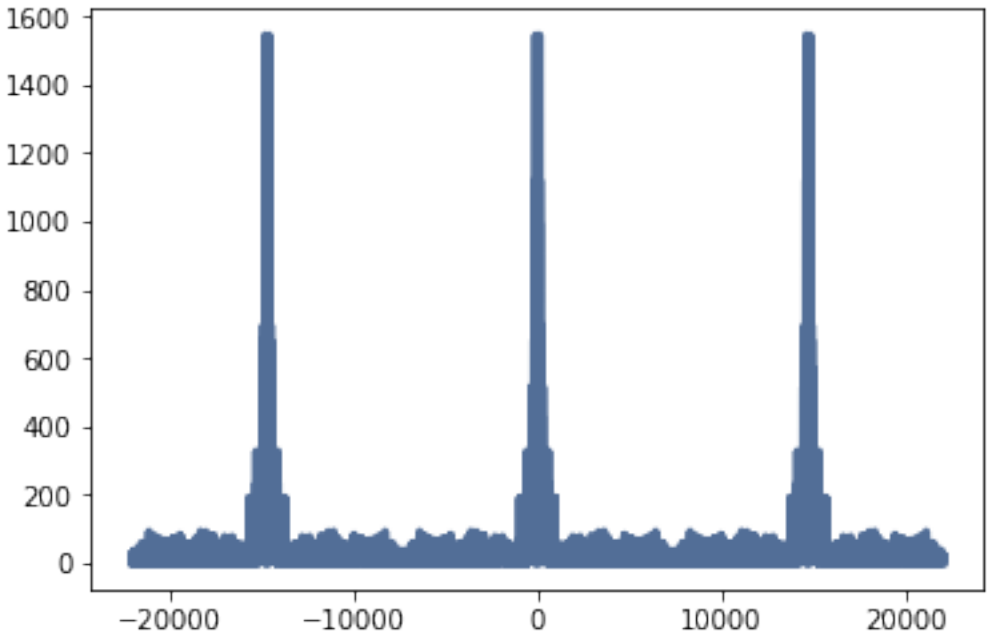
Результат содержит копии оригинального спектра около 20 кГц, но на слух они не сильно заметы:

[9]:
```
sampled = sample(filtered, factor)
sampled.make_audio()
```

[9]:
```
<IPython.lib.display.Audio object>
```

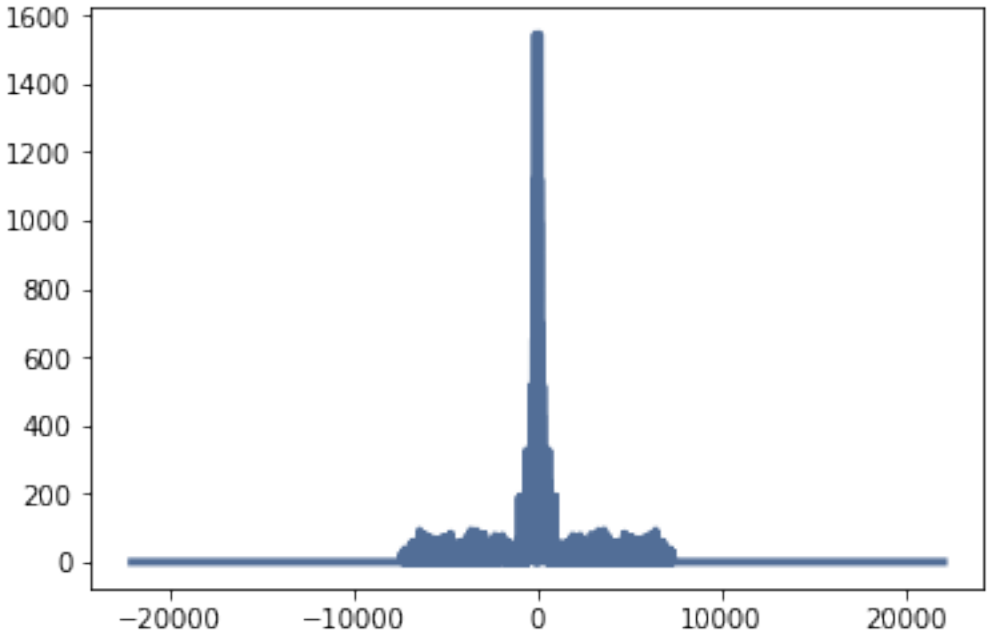
Но они появляются, когда мы строим спектр:

[10]:
```
sampled_spectrum = sampled.make_spectrum(full=True)
sampled_spectrum.plot()
```



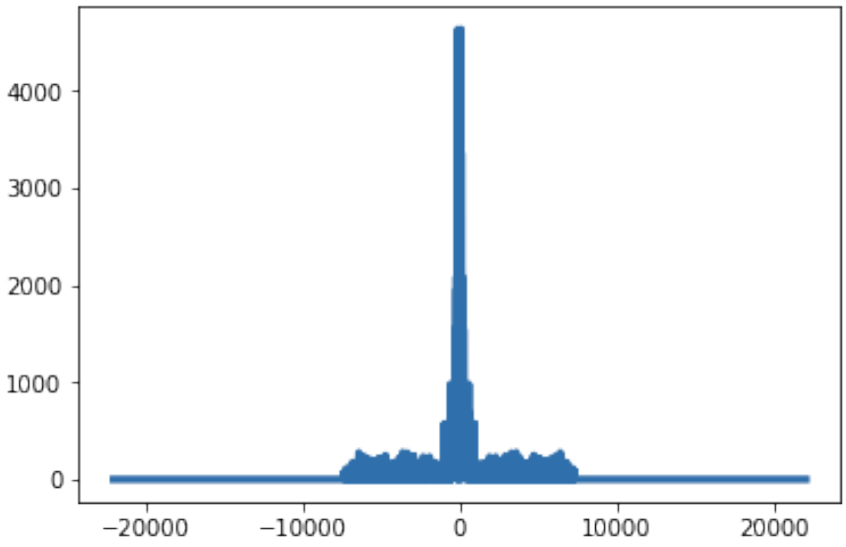
Избавимся от спектральных копий, снова применив фильтр сглаживания:

[11]:
```
sampled_spectrum.low_pass(cutoff)
sampled_spectrum.plot()
```



Мы только что потеряли половину энергии в спектре, но мы можем масштабировать результат, чтобы вернуть её:

[12]:
```
sampled_spectrum.scale(factor)
spectrum.plot()
sampled_spectrum.plot()
```



Теперь разница между спектром до и после выборки должна быть небольшой.

[13]:
```
spectrum.max_diff(sampled_spectrum)
```

[13]:
```
1.8749713606747085e-12
```

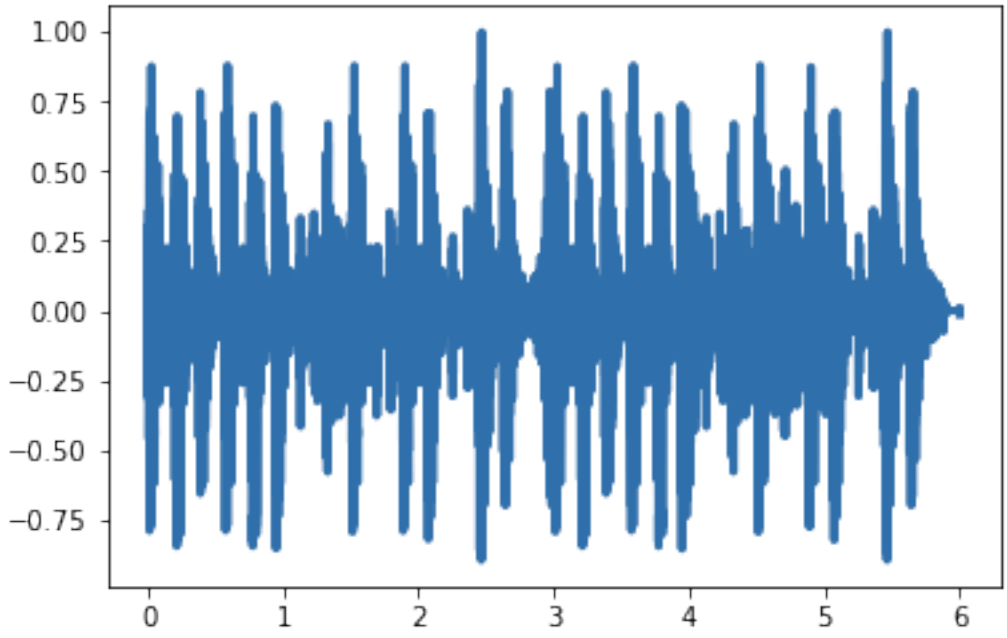
После фильтрации и масштабирования мы можем преобразовать спектр обратно в волну:

[14]:
```
interpolated = sampled_spectrum.make_wave()
interpolated.make_audio()
```

[14]:
```
<IPython.lib.display.Audio object>
```

Разница между интерполированной волной и фильтрованной волной также должна быть небольшой.

[15]:
```
filtered.plot()
interpolated.plot()
```



[16]:
```
filtered.max_diff(interpolated)
```

[16]:
```
6.705919263258977e-16
```

4 Вывод

В ходе выполнения данной лабораторной работы была изучена амплитудная модуляция, играющая важную роль как в радиосвязи, так и в понимании теоремы о выборках и её практическое применение.