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

Телекоммуникационные технологии

Отчёт по лабораторной работе №9

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

Санкт-Петербург
2021

Содержание

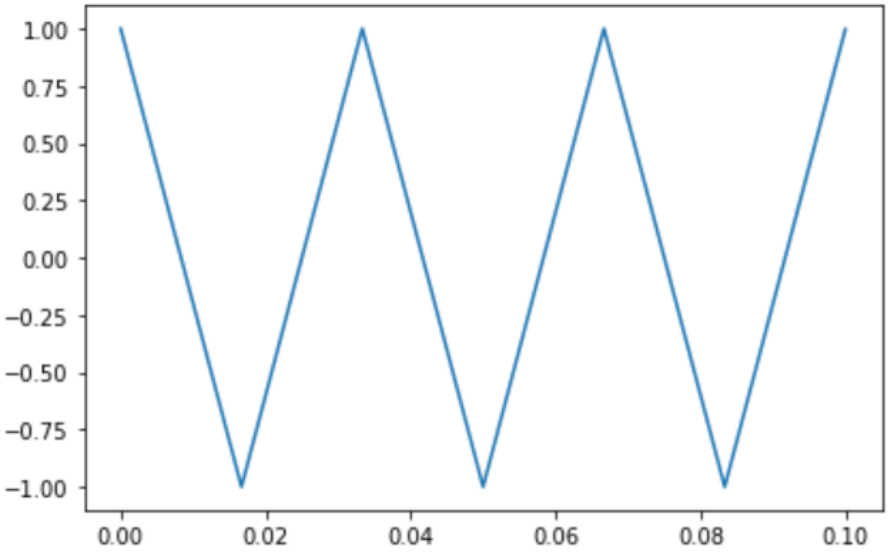
1. Выполнение работы

1.1. Упражнение 1

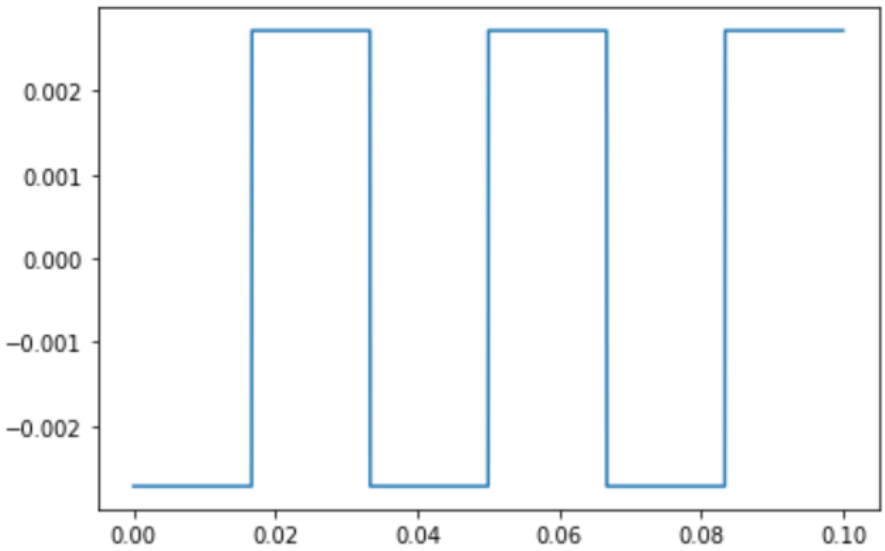
Упражнении просят изучить и запустить примеры, рассмотренные в главе и приведеные в файле chap09, что я собственно и сделал.

1.2. Упражнение 2

Для сравнения работы функций diff и differentiate я создал простой треугольный сигнал и вывел его. 3 периодов вполне даостаточно для наблюдения разницы.

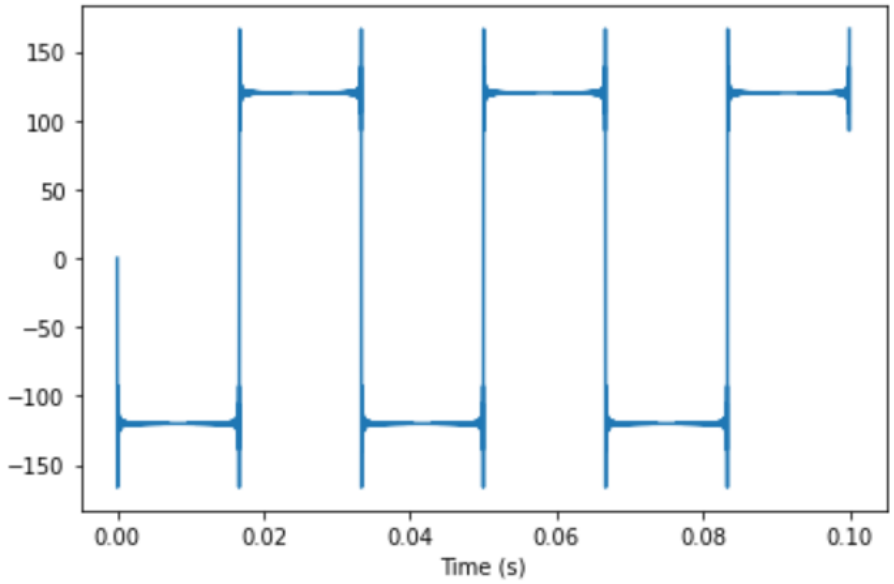


Первой я решил применить diff и вот что я собственно получил. Diff от треугольной функции - прямоугольная функция, что кстати объясняет то, что гармоники прямоугольного и треугольного сигналов спадают по $1/f$ и $1/f^2$ соответственно.



Далее я создал спектр рассматриваемого сигнала, применил к нему вторую функцию - differentiate, и создал на его основе новую волну. Вот что получилось.

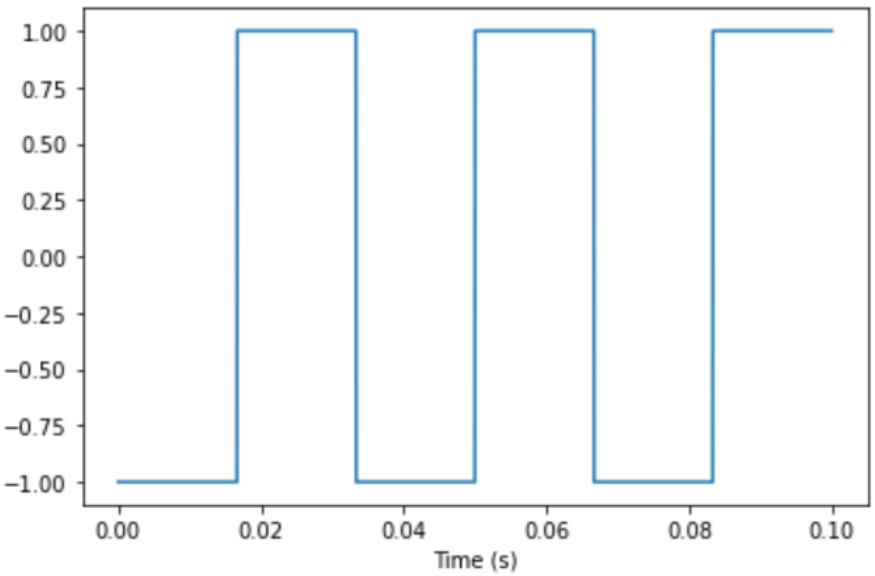
Как можно видеть, применение второй функции дает более размытое изображение в местах разрывов, что, как я понял, связано с неопределенностью функции в местах вершин треугольного сигнала. Полный код для упражнения приведен ниже.




```
in_wave = TriangleSignal(freq=30).make_wave(duration=0.1, framerate=44100)
in_wave.plot()
decorate(xlabel='Time (s)')

out_wave = in_wave.diff()
out_wave.plot()
decorate(xlabel='Time (s)')

out_wave2 = in_wave.make_spectrum().differentiate().make_wave()
out_wave2.plot()
decorate(xlabel='Time (s)')
```

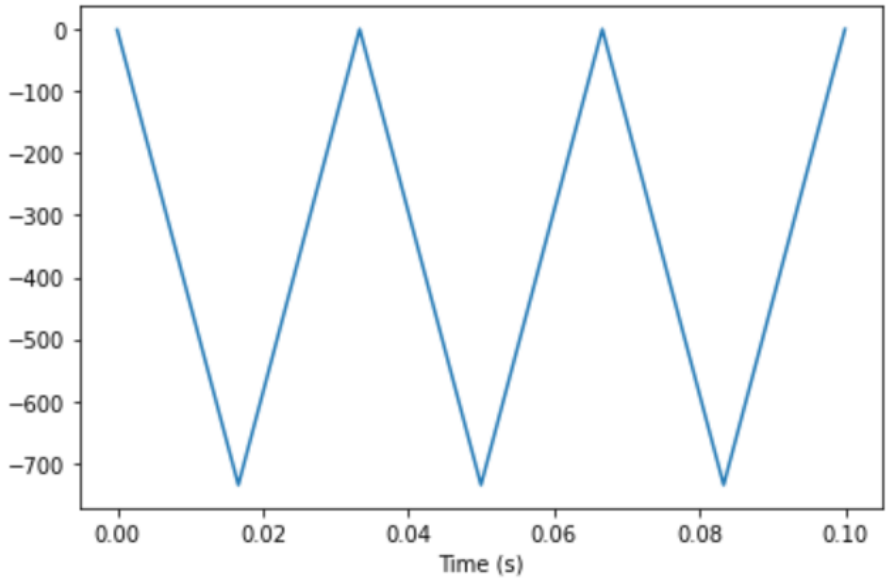
1.3. Упражнение 3

Для сравнения функций cumsum и integrate пойдем в обратную сторону. Для начала я создал прямоугольный сигнал и вывел его.

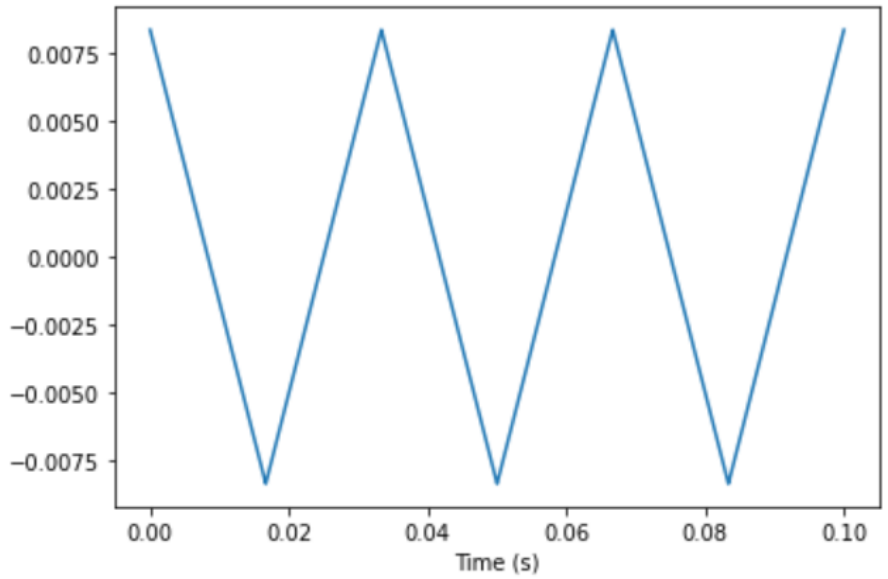


Как я уже упоминул, процесс у нас обратный, и примнение функции cumsum дает

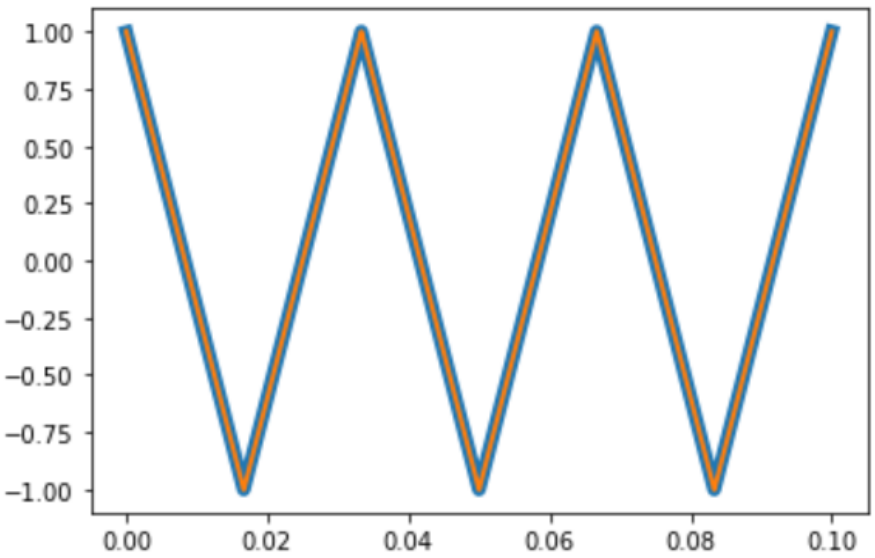
обратный результат, то есть треугольный сигнал из прямоугольного.



Далее, так же, как и в предыдущем упражнении, создаем спектр сигнала, применяем к нему вторую функцию (integrate) и генерируем новый сигнал на его основе.



Только в этот раз разница проявляется не в расплывчатости некоторых значений сигнала, а в разном масштабе их значений. Для удобства сравнения я провел нормализацию и вывел оба графика:

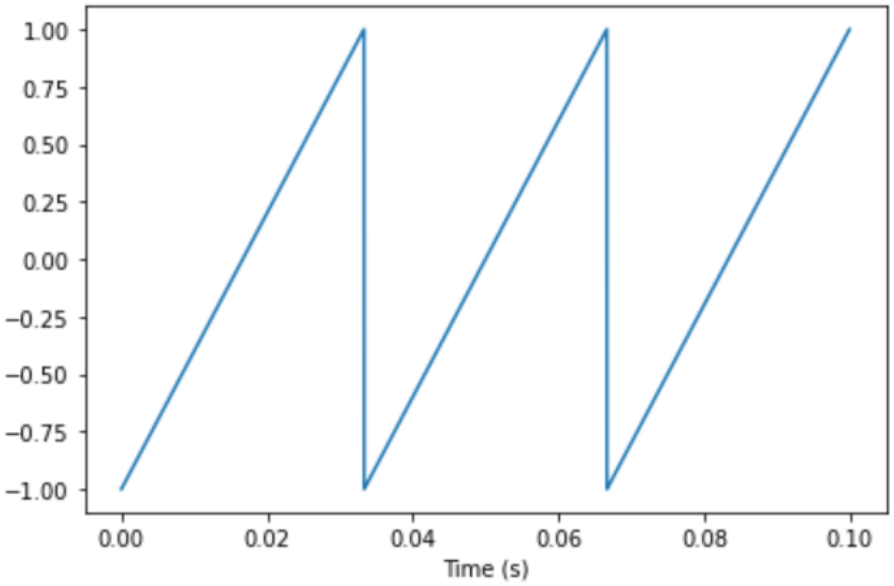


Как можем видеть, графики выглядят похоже, и разница между ними очень мала. Полный код для этого упражнения приведен ниже:

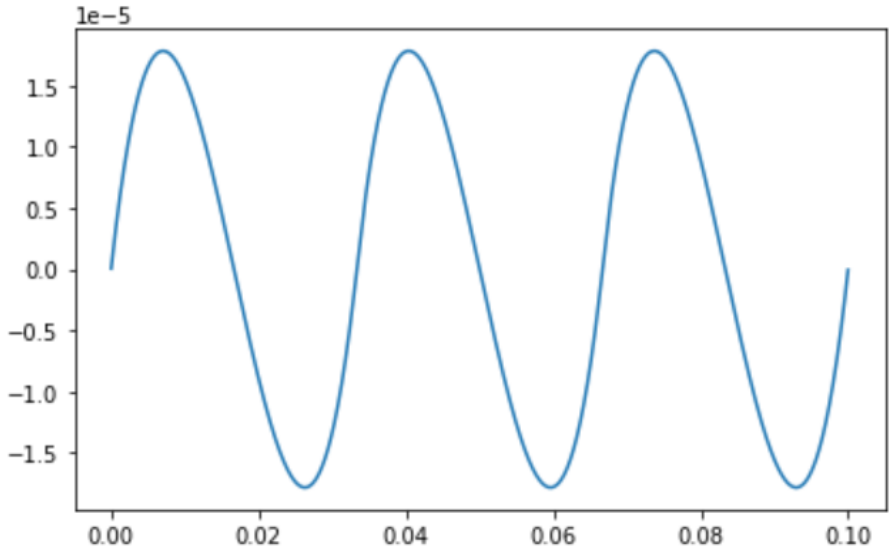
```
in_wave = SquareSignal(freq=30).make_wave(duration=0.1, framerate=44100)
in_wave.plot()
decorate(xlabel='Time (s)')

out_wave = in_wave.cumsum()
out_wave.plot()
decorate(xlabel='Time (s)')

spectrum = in_wave.make_spectrum().integrate()
spectrum.hs[0] = 0
out_wave2 = spectrum.make_wave()
out_wave2.plot()
decorate(xlabel='Time (s)')

out_wave.unbias()
out_wave.normalize()
out_wave2.normalize()
out_wave.plot(linewidth=6.0)
out_wave2.plot(linewidth=2.0)
```

1.4. Упражнение 4

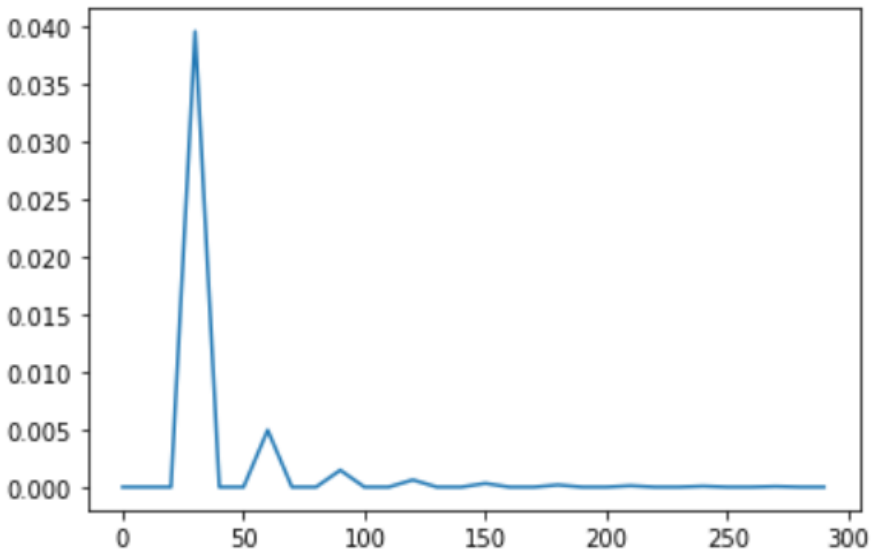
В данном упражнении предлагается исследовать влияние двойного интегрирования на пилообразный сигнал. Для начала я его создал и вывел:



А затем, как и сказанно в упражнении вычислил его спектр и применил к нему дважды функцию integrate.

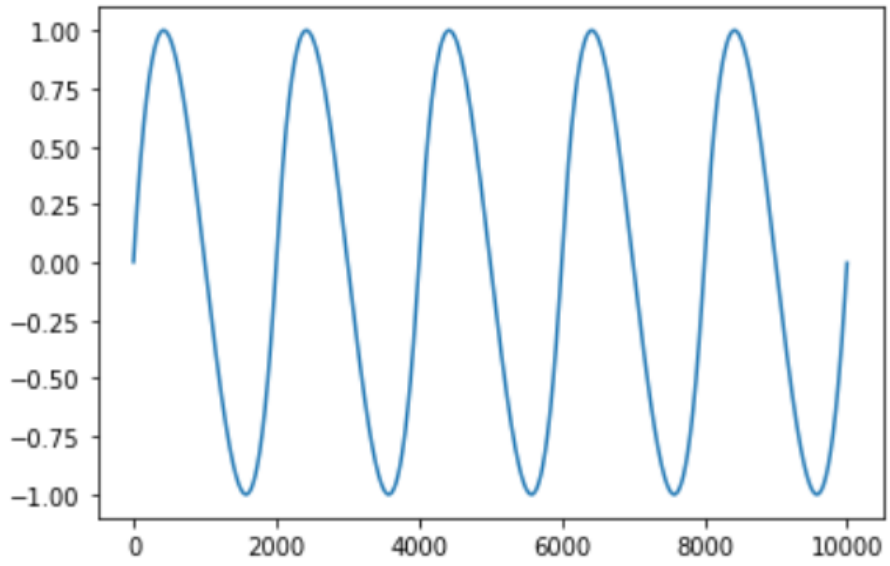


Форма получившегося сигнала кубическая, а его схожесть с синусоидой объясняется тем, что применение функции integrate привело к фильтрации почти всех высоких частот, что можно увидеть на приведенном ниже рисунке.

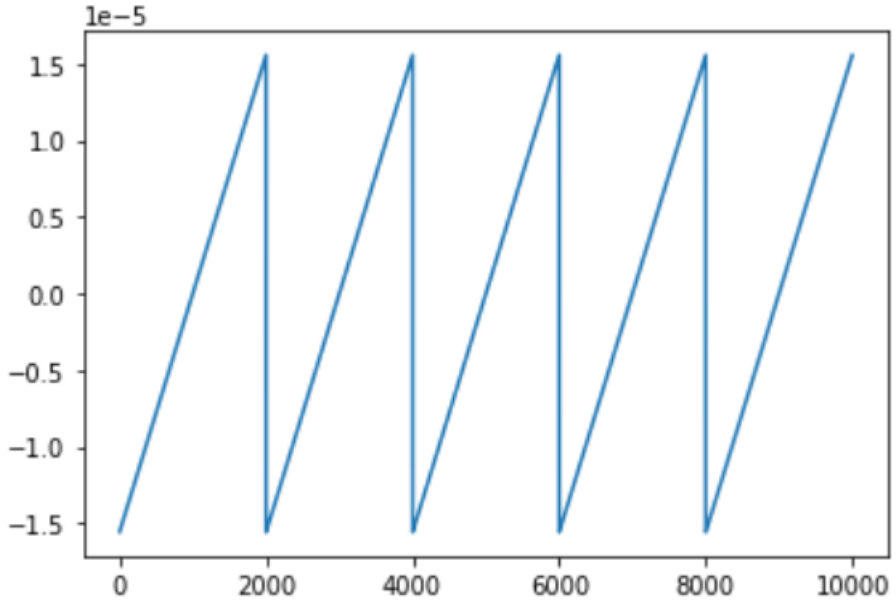


1.5. Упражнение 5

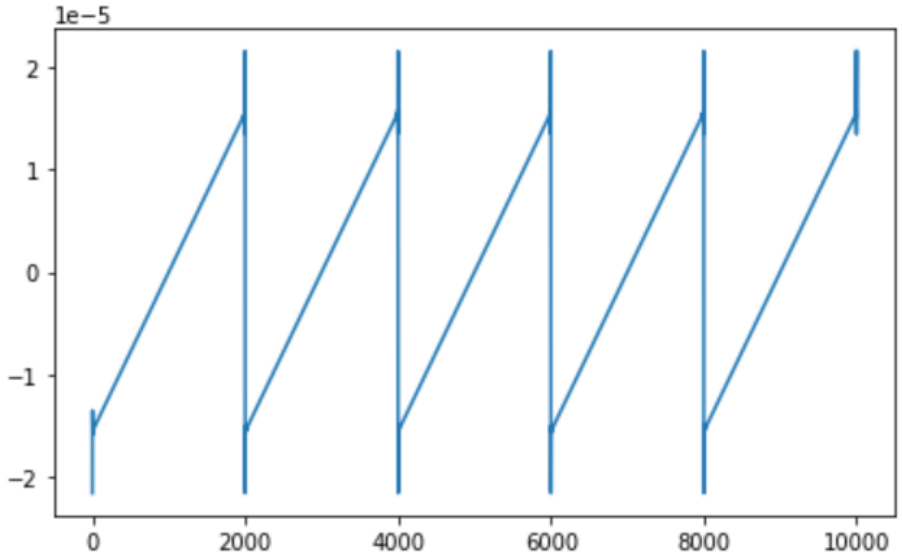
Для начала я создал кубический сигнал:



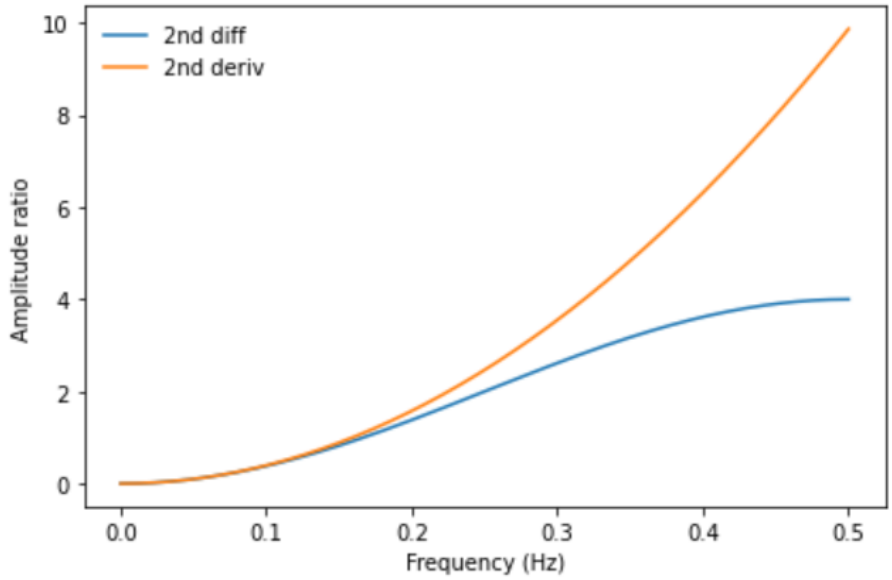
Потом применил к нему дважды функцию diff и ожидаемо получил пилообразный сигнал (обратный переход, по сравнению с предыдущим упражнением):



Затем вычислил спектр и применил к нему дважды differentiate. Как и в случае с трегуольным сигналом, видим шум в точках разрывов, связанный с неопределенностью параболической функции (первая производная кубического сигнала) на вершинах.



Как итог я вывел фильтры для второй разницы и второй производной фильтры и сравнил их:



Как легко заметить по графику, оба фильтра усиливают высокие частоты, причем вторая производная делает это намного более явно с увеличением частоты, чем вторая разность.

2. Выводы

В ходе выполнения данной лабораторной работы я изучил на практике применение таких функций как diff, cumsum, differentiate и integrate к различным видам сигналов, а также познакомился со сравнением окон во временной области и фильтров - в частотной.