

Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Телекоммуникационные технологии

Отчет по лабораторной работе №3
FIR-фильтрация

Работу
выполнил:
Чугунов А.А.
Группа: 33501/4
Преподаватель:
Богач Н.В.

Санкт-Петербург
2017

1. Цель работы

Изучить воздействие ФНЧ на тестовый сигнал с шумом.

2. Теоретическая информация

Линейный фильтр — динамическая система, применяющая некий линейный оператор ко входному сигналу для выделения или подавления определённых частот сигнала. Суть фильтрации заключается в пропускании сигнала через линейную цепочку с какими-то заданными параметрами для получения нужного нам конечного сигнала. Пусть имеем два сигнала - начальный и конечный, которые имеют следующий вид:

$$x(t) = A_x e^{j(2\pi ft + \phi_x)} \quad (1)$$

$$y(t) = A_y e^{j(2\pi ft + \phi_y)} \quad (2)$$

Поделив конечный сигнал(2) на начальный(1), получаем частотную характеристику:

$$G(f) = \frac{A_y}{A_x} e^{j(\phi_y - \phi_x)} \quad (3)$$

Спектр выходного сигнала можно посчитать по следующей формуле:

$$Y(f) = X(f)G(f) \quad (4)$$

Таким образом, варьируя частотную характеристику линейной цепи можем получать на выходе различные сигналы. Если же работать во временной области пренебрегая преобразованиями Фурье, можем пропустить через линейную цепь функцию Дирака. Спектр его равен единице. Тогда получаем следующее выражение:

$$Y(f) = G(f) \quad (5)$$

Реакция цепи $g(t)$ на функцию Дирака носит название импульсной характеристики. Импульсная характеристика связана с частотной следующими соотношениями:

$$g(t) = \int_{-\infty}^{\infty} G(f) e^{j(2\pi ft)} df \quad (6)$$

$$G(f) = \int_0^{\infty} g(t) e^{-j(2\pi ft)} dt \quad (7)$$

Таким образом, во временной области получаем следующую формулу:

$$y(t) = x(t) * g(t) \quad (8)$$

3. Ход выполнения работы

Для начала сгенерируем гармонический сигнал. В данной работе будет использоваться синус. (Рисунок. 3.1)

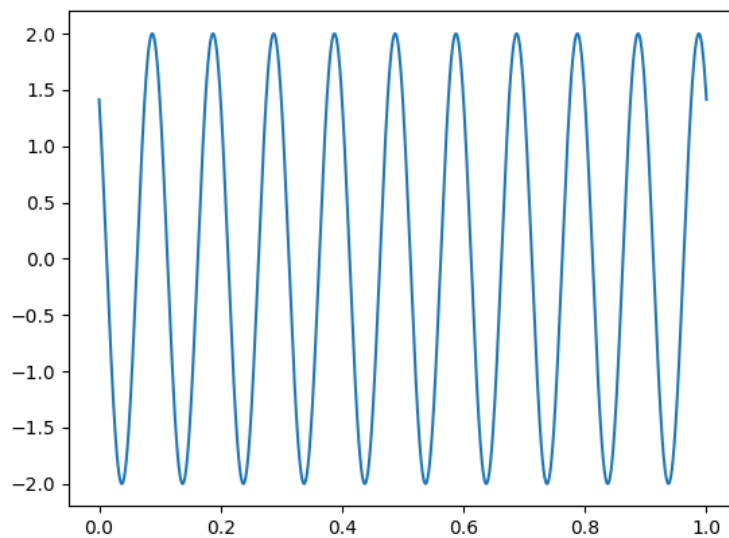


Рисунок 3.1. Гармонический сигнал

Теперь добавим шума в сигнал. Шумом будет служить синус, но более высокой частоты и более низкой амплитуды. (Рисунок. 3.2)

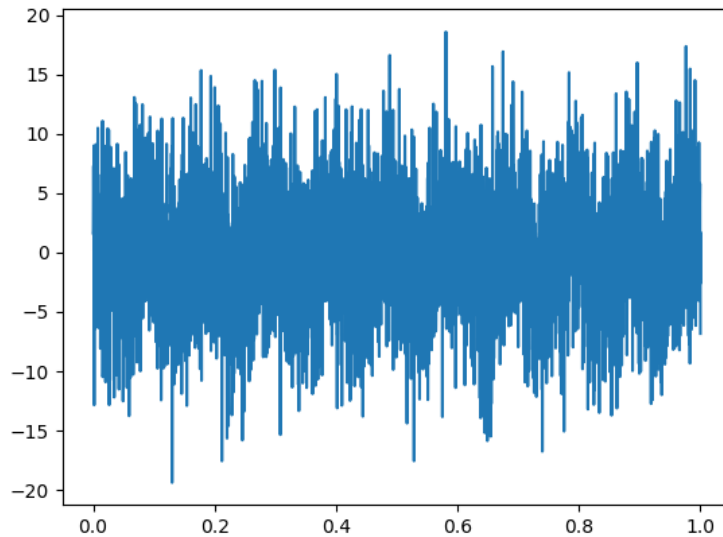


Рисунок 3.2. Зашумленный гармонический сигнал

Рассмотрим спектры обоих сигналов. (Рисунок. 3.3 и Рисунок. 3.4)

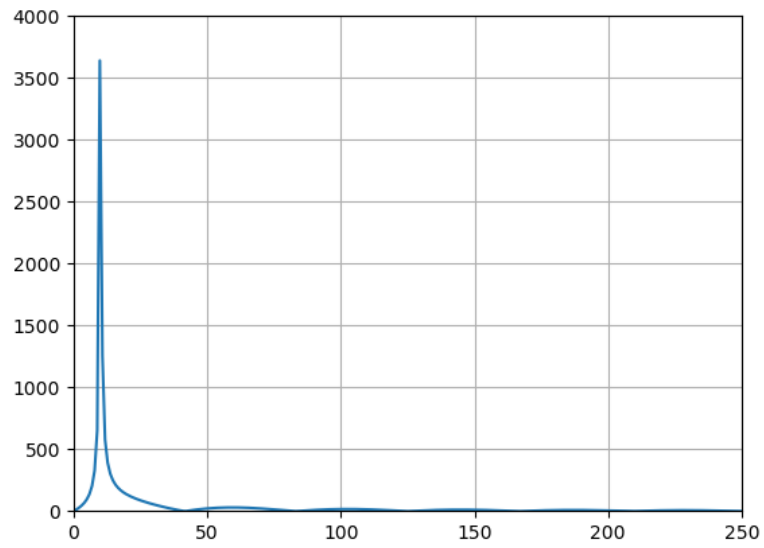


Рисунок 3.3. Спектр гармонического сигнала

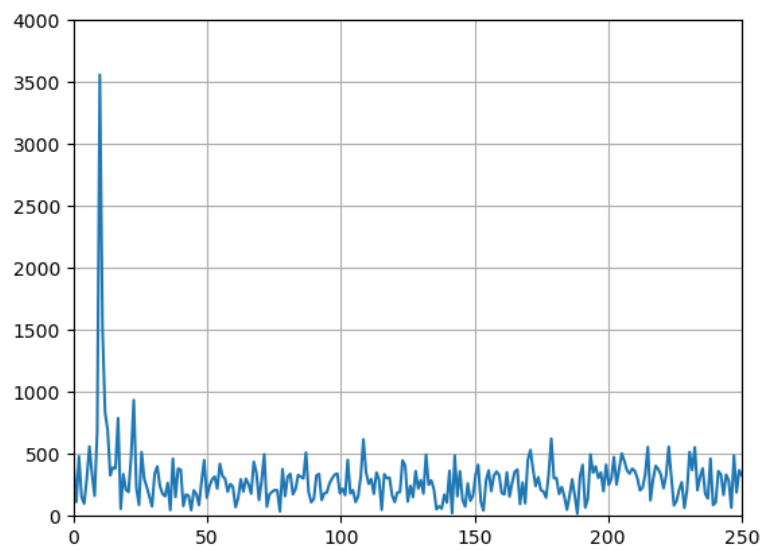


Рисунок 3.4. Спектр зашумленного гармонического сигнала

На спектре зашумленного гармонического сигнала наблюдаем паразитные частоты. Воспользуемся FIR-фильтром, а именно окном Хэмминга.

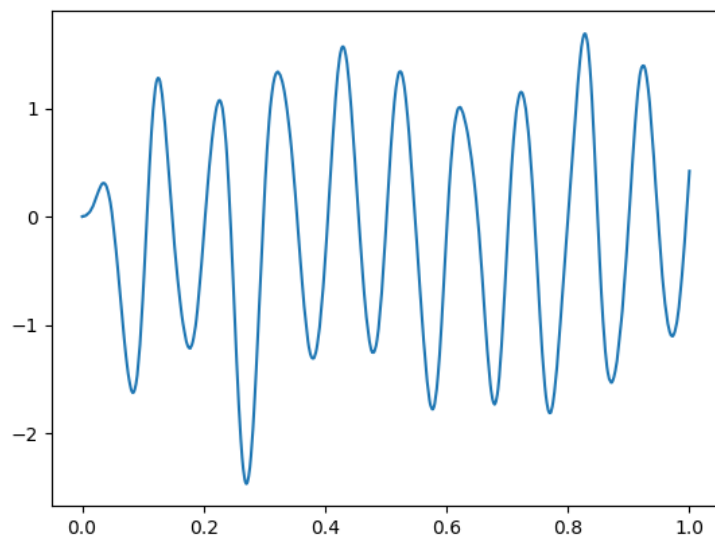


Рисунок 3.5. Сигнал после фильтрации

Получили неплохие результаты, однако, стоит отметить, что они неидеальны. Рассмотрим спектр сигнала после фильтрации.(Рисунок. 3.6)

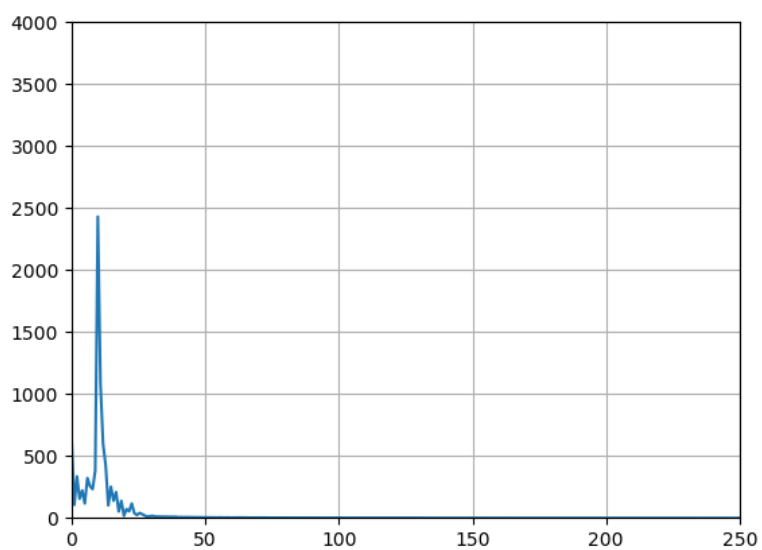


Рисунок 3.6. Спектр после фильтрации

В частотной области паразитные частоты выше частоты гармонического сигнала исчезли.

Код на языке Python представлен ниже:

Листинг 1: Filtration.py

```
1 from scipy.fftpack import fft
2 import numpy as np
```

```

3 import time as time
4 import random
5 import matplotlib.pyplot as plt
6 from scipy import signal
7
8
9 def spectrum(noiseSig, Fdiscrete):
10     Nfft = int(2 ** np.ceil(np.log2(len(noiseSig))))
11     sp = fft(noiseSig, Nfft)
12     f = np.arange(0, Nfft - 1) / Nfft * Fdiscrete
13     plt.figure()
14     plt.grid()
15     plt.plot(f[:int(Nfft / 2)], np.abs(sp[:int(Nfft / 2)]))
16     plt.axis([0, 250, 0, 4000])
17
18
19 Fdiscrete = int(4e3)
20 t = np.linspace(0, 1, int(Fdiscrete))
21 A = 2
22 f0 = 10
23 phi = np.pi / 4.0
24 # noinspection PyTypeChecker
25 sig = A * np.cos(phi + 2 * np.pi * f0 * t)
26 mean = 0
27 std = 1
28 samples = 5 * np.random.normal(mean, std, size=Fdiscrete)
29
30 noiseSig = sig + samples # type: np.ndarray
31 plt.figure()
32 plt.plot(t, sig)
33 plt.figure()
34 plt.plot(t, noiseSig)
35 spectrum(noiseSig, Fdiscrete)
36
37 Length = 300
38 a = signal.firwin(Length, cutoff=f0, window="hamming", nyq=Fdiscrete / 2)
39 newSig = signal.lfilter(a, 1, noiseSig)
40
41 plt.figure()
42 plt.plot(t, newSig)
43 spectrum(newSig, Fdiscrete)
44 plt.show()

```

4. Выводы

В дополнение к работе по фильтрации, мы выполнили фильтрацию FIR-фильтром и у нас получились неплохие результаты. Более того, оконный фильтр явно показал нам, как можно "отрезать" ненужные нам частоты.