НИУ ИТМО

Факультет систем управления и робототехники

Лабораторная работа № 3
по дисциплине «Частотные методы»

Выполнил: Гридусов Д.Д

Преподаватель: Перегудин А.А

Санкт-Петербург
2024 г.

Содержание

1 Жёсткая фильтрация

1.1 Убираем высокие частоты

Для начала нарисуем график исходной ситуации для случая $c = 0$, $a = 2$, $b = 0.32$ и попытаемся убрать высокие частоты с помощью преборазования Фурье.

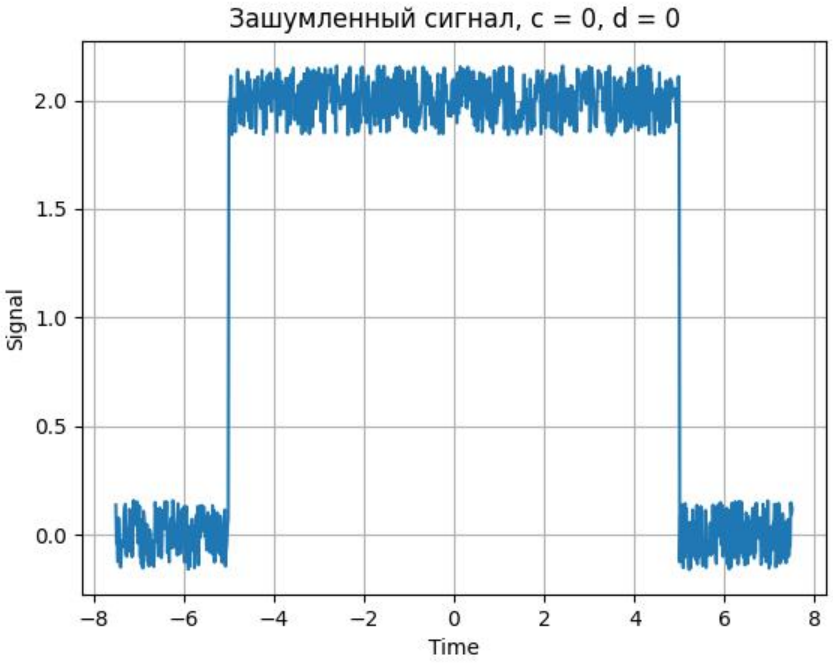

Рис. 1: Зашумленный сигнал c = 0, d = 0

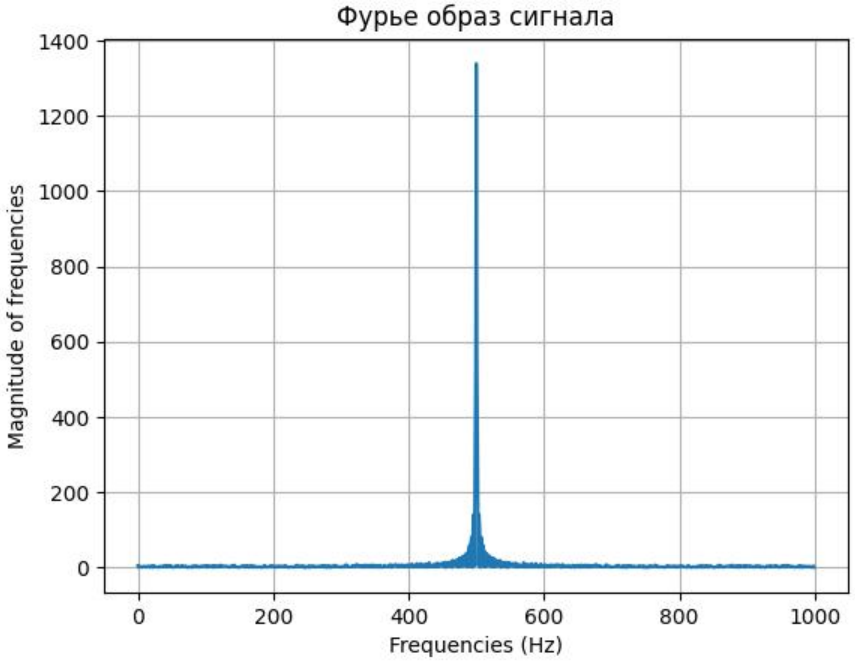

Рис. 2: Фурье-образ

Сразу обратим внимание на то, что параметр b отвечает за степень зашумленности сигнала - покажем это на графике.

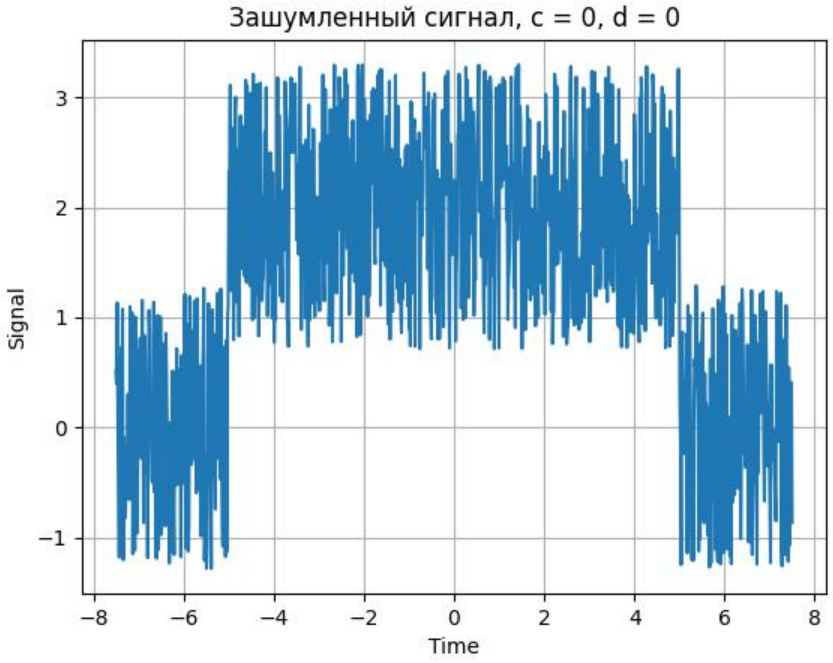

Рис. 3: Зашумленный сигнал с большим параметром b

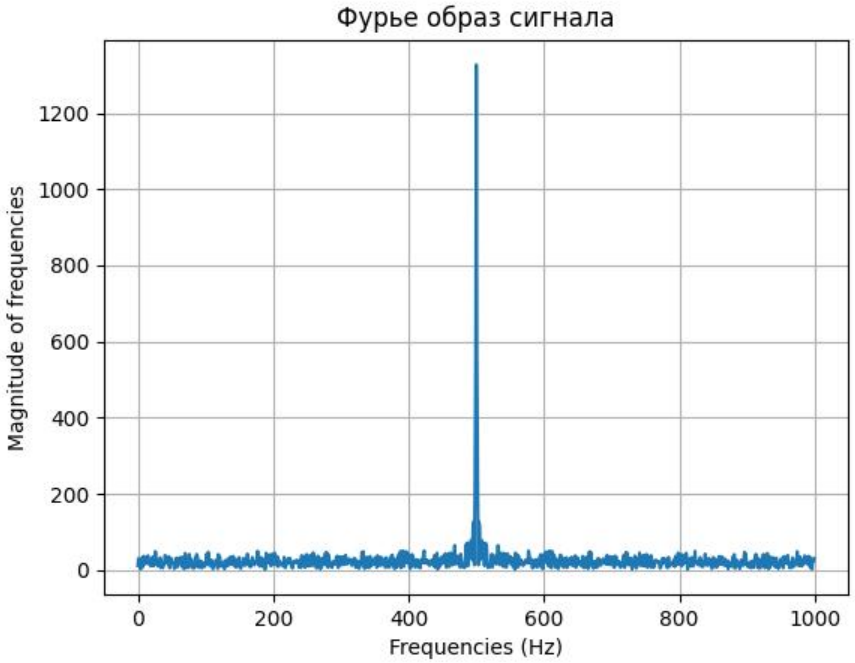

Рис. 4: Фурье-образ более зашумленного сигнала

Напишем функцию для фильтрации верхних частот
Листинг 1.1 Фильтрация верхних частот

```
# Filter that removes only some high frequencies from signal
```

```python
def high_freq_filtration(fourier_image, treshold):
    n = fourier_image.shape[0]
    frequencies_range = np.linspace(-n/2, n/2, 1000)
    counter = 0
    for k in frequencies_range:
        if (abs(k) >= treshold):
            fourier_image[counter] = 0
        counter += 1
    return fourier_image
```

Применим фильтры к обоим сигналам.

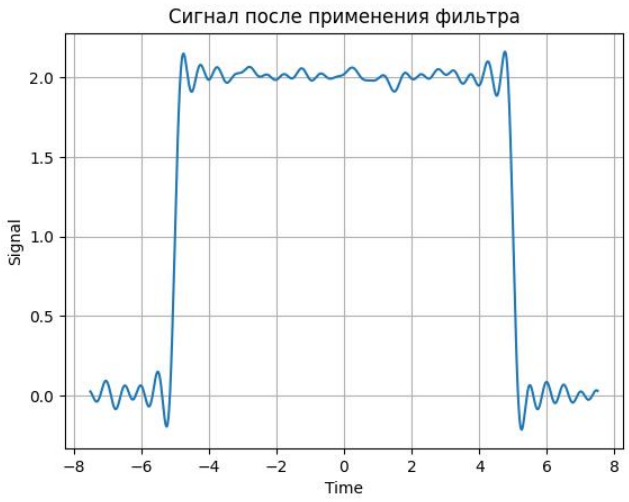

Рис. 5: Фильтрация первого сигнала с $b = 0.32$

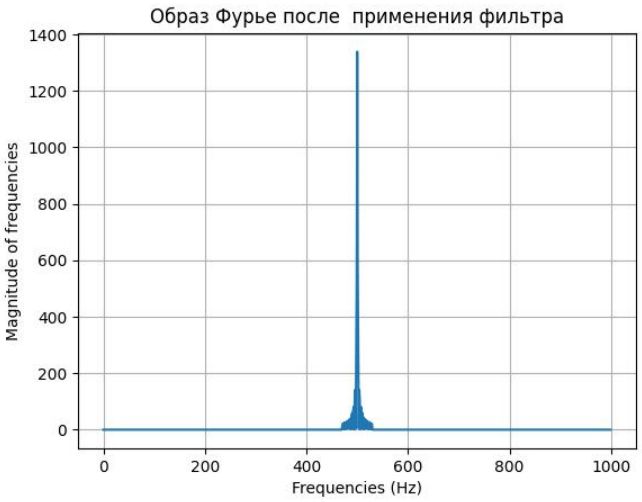

Рис. 6: Фурье-образ первого сигнала

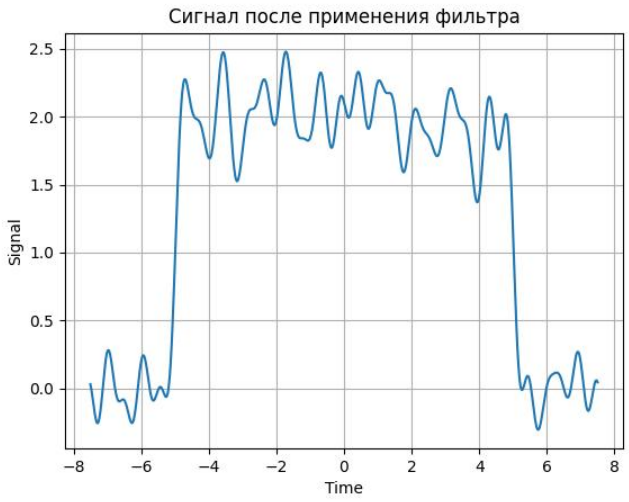

Рис. 7: Фильтрация второго сигнала с $b = 2.6$

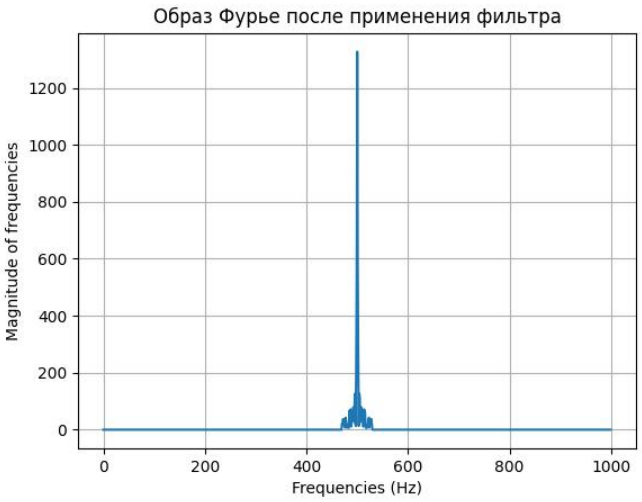

Рис. 8: Фурье-образ второго сигнала

Со вторым сигналам получилось гораздо хуже, что можно было предположить, так как чем больше шума, тем сложнее от него избавиться.

1.2 Убираем специфические частоты

В этом задании у сигнала появиться еще и сдвиг на фазу, но (хоть я и не сразу это понял), получить после фильтрации нужно исходную прямоугольную волну. Империческое наблюдение - чем шире исходная квадртаная волна, тем сильнее она отличается от синусоидальной функции и тем проще фильтру восстановить исходный сигнал.

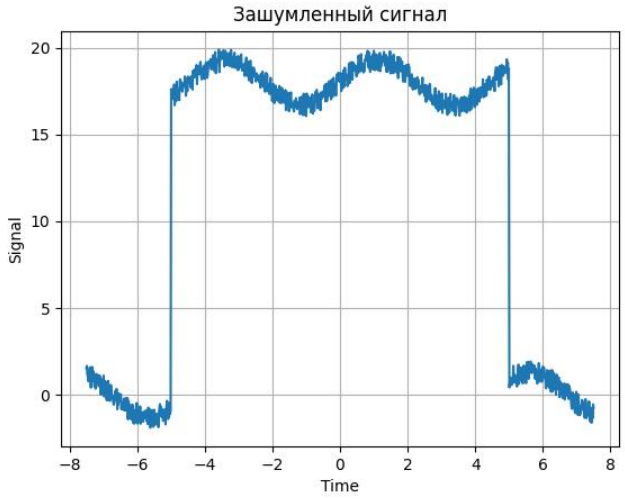

Рис. 9: Зашумленный сигнал + сдвиг на фазу

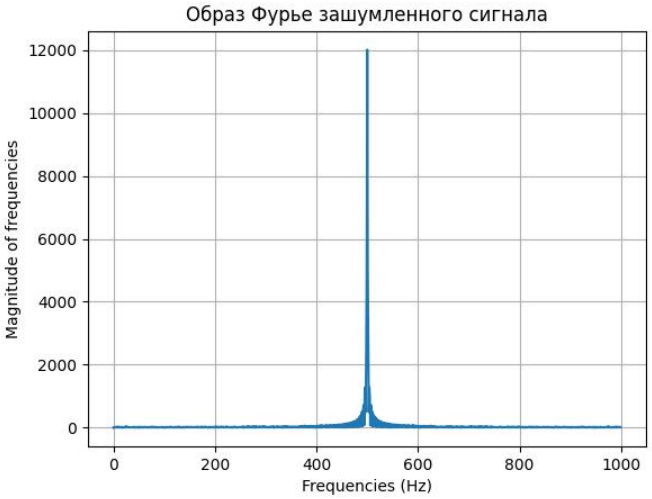

Рис. 10: Фурье-образ

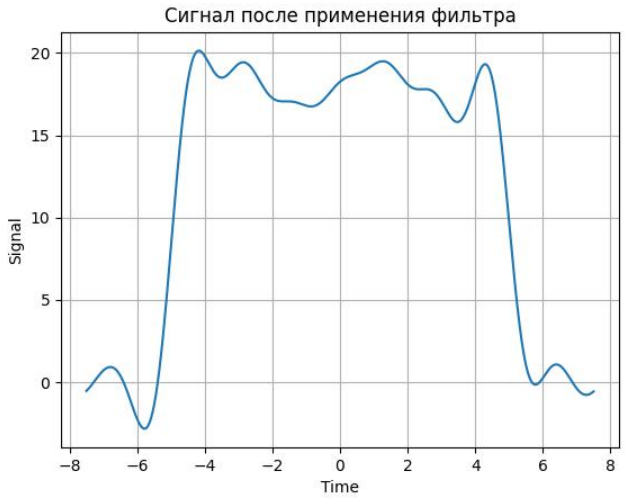

Рис. 11: Фильтрация шума и сдвига

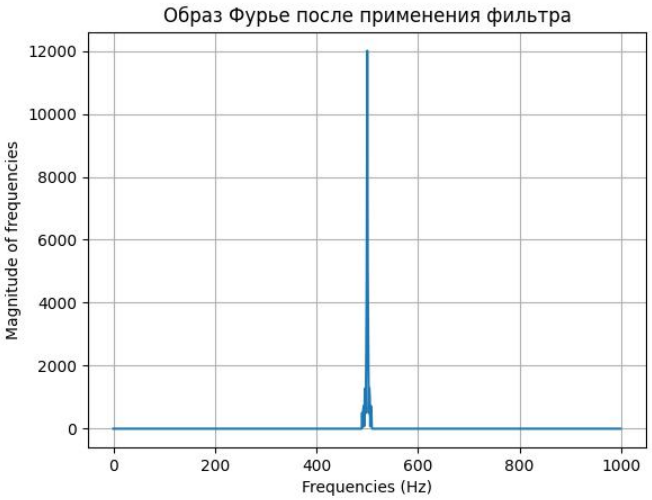

Рис. 12: Фурье-образ отфильтрованного сигнала

Заметно, что фильтр справляется достаточно неплохо, но что будет если убрать шум и оставить только сдвиг на фазу? Положим $b = 0$:

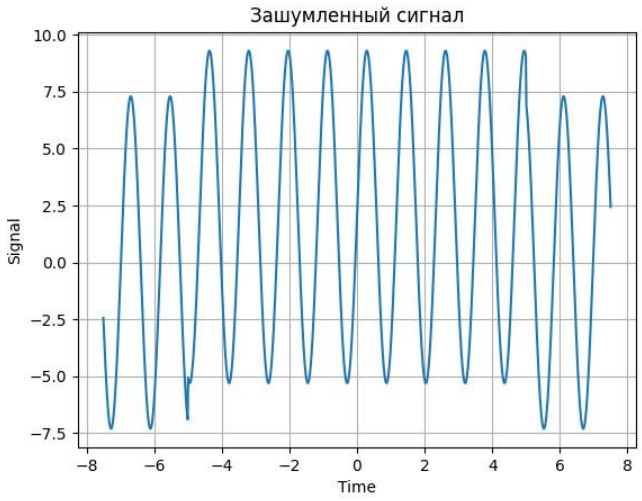

Рис. 13: $b = 0$ и сдвиг на фазу

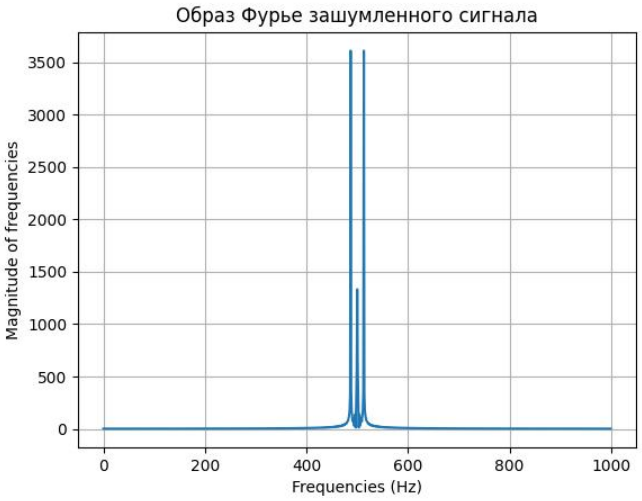

Рис. 14: Фурье-образ

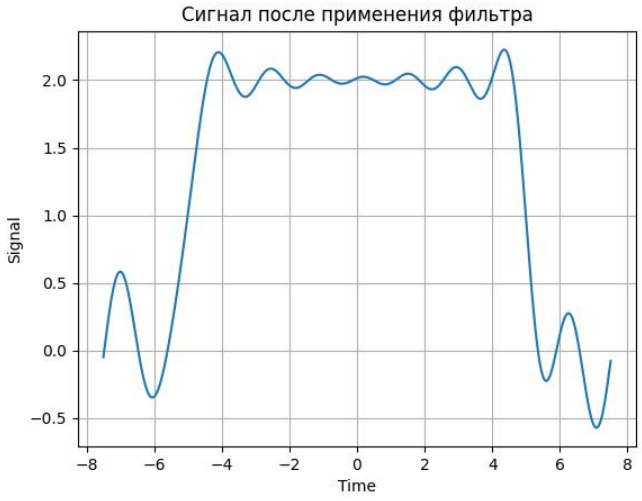

Рис. 15: Результат фильтрации

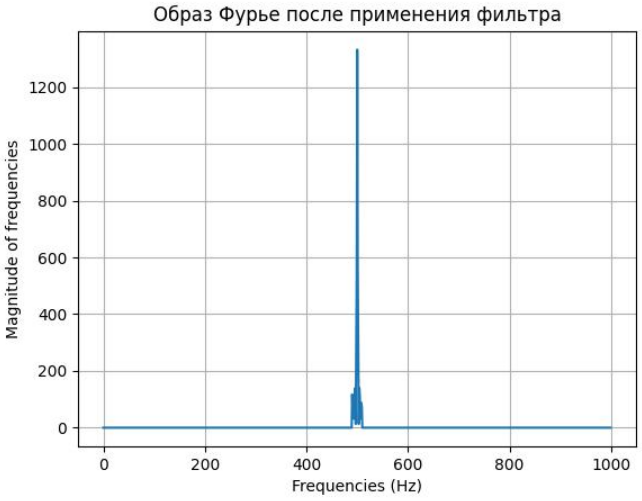

Рис. 16: Фурье-образ отфильтрованного сигнала

Восстановленный сигнал очень похож на квадратную волну, что не может не радовать.

Листинг 1.2 Фильтрация специфических частот

```
# Filter that will remove both high and low freqs
def spec_freq_filtration(fourier_image, treshold1, treshold2):
  n = fourier_image.shape[0]
  frequencies_range = np.linspace(-n/2, n/2, 1000)
  counter = 0
  for k in frequencies_range:
    if (abs(k) <= treshold1) or (abs(k) >= treshold2):
      fourier_image[counter] = 0
      counter += 1
  return fourier_image
```

1.3 Фильтрация нижних частот

Листинг 1.3 Фильтрация нижних частот

```
# Filter that will remove only some low freqs
def low_freq_filtration(fourier_image, treshold1):
  n = fourier_image.shape[0]
  frequencies_range = np.linspace(-n/2, n/2, 1000)
  counter = 0
  for k in frequencies_range:
    if (abs(k) <= treshold1):
      fourier_image[counter] = 0
      counter += 1
  return fourier_image
```

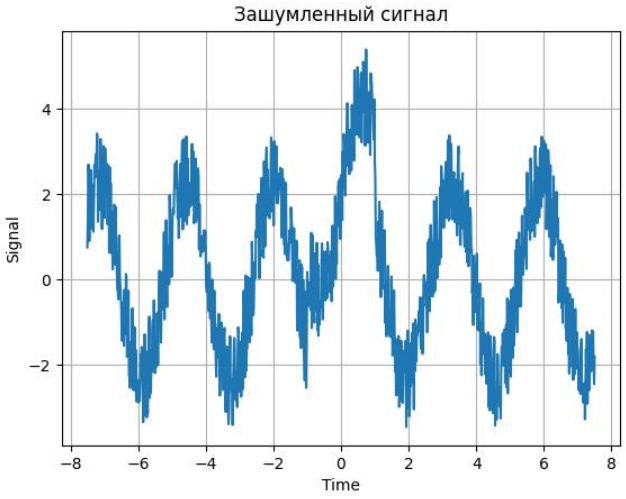

Рис. 17: Зашумленный сигнал + сдвиг, все параметры больше 0

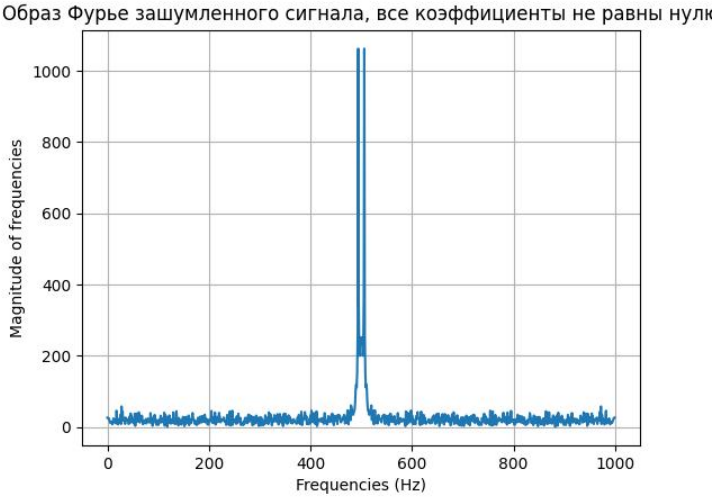

Рис. 18: Фурье-образ

Исходя из логики предыдщего пункта, чтобы убрать шум и сдвиг, нам нужно обнулить нехарактерные исходному сигналу два "горба"у образа Фурье, оставив при этом частоты, находящиеся между этими горбами. Но, как будет видно и по графику, это не получается сделать при любом диапазоне запретных частот.

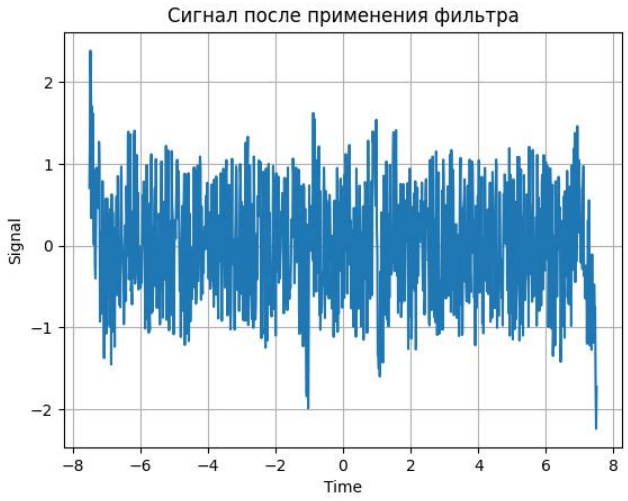

Рис. 19: Результат фильтрации

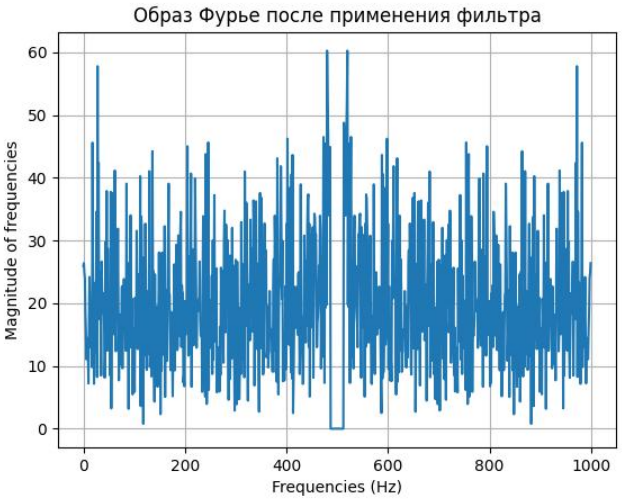

Рис. 20: Фурье-образ отфильтрованного сигнала

Видно, что фильтрации не получилось.

2 Фильтрация звука

Сначала прочитаем звуковую волну из .wav файла и построим ее образ Фурье: Судя по распре-

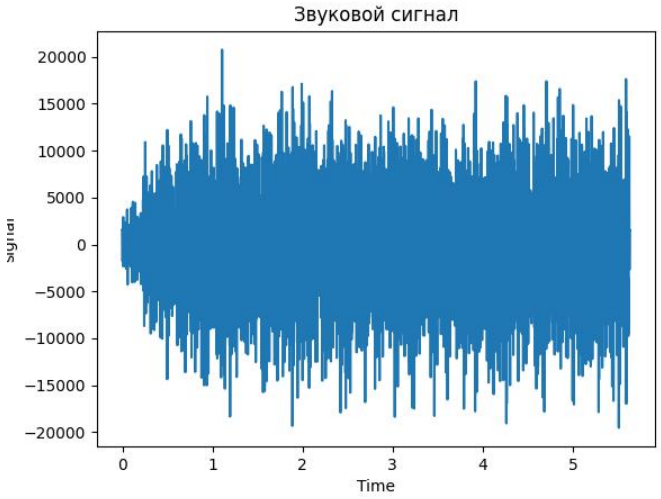

Рис. 21: Зашумленная звуковая волна

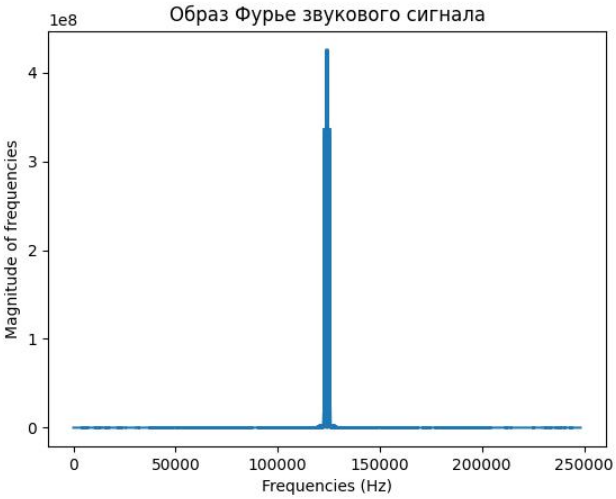

Рис. 22: Фурье-образ исходной звуковой волны

делению частот лучше всего будет убирать высокие частоты, так мы и сделаем.

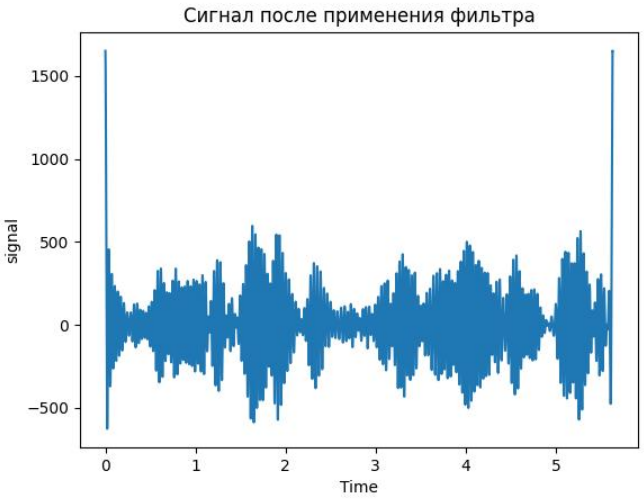

Рис. 23: Звуковая волна после фильтра.

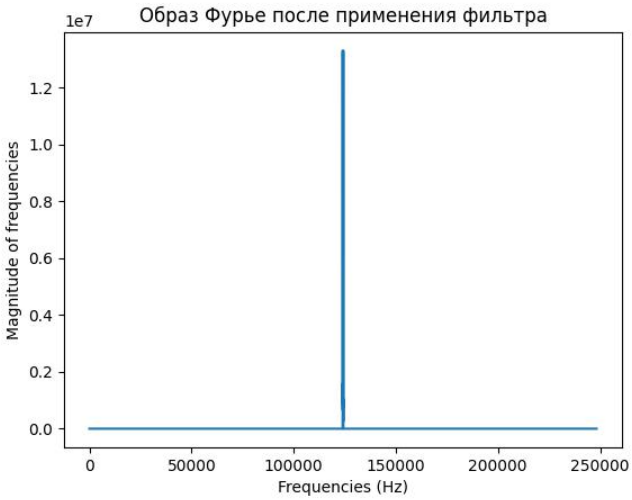

Рис. 24: Фурье-образ отфильтрованного сигнала

Вывод: заметно, что фильтр помог и остался только содержательный звук

3 Исходный код на github

Весь код лабораторной работы №3: исходники