

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Университет ИТМО

Факультет систем управления и робототехники

ОТЧЁТ  
по дисциплине  
*”Частотные методы”*

по теме:  
ЖЁСТКАЯ ФИЛЬТРАЦИЯ

Студент:  
*Группа R3236*

*Поляков А.А.*

Предподаватель:  
*к.т.н., доцент*

*Перегудин А.А.*

Санкт-Петербург  
2024

## СОДЕРЖАНИЕ

|       |  |    |
|-------|--|----|
| 1     | ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ И БИБЛИОТЕКИ ..... | 3  |
| 2     | ЗАДАНИЕ 1. ЖЁСТКИЕ ФИЛЬТРЫ .....           | 4  |
| 2.1   | Убираем высокие частоты .....              | 4  |
| 2.1.1 | Фурье-образ сигнала и .....                | 5  |
| 2.1.2 | Применяем фильтр .....                     | 6  |
| 2.1.3 | Возвращаемся к очищенному сигналу .....    | 7  |
| 2.1.4 | Выводы.....                                | 7  |
| 2.2   | Убираем специфические частоты .....        | 12 |
| 2.2.1 | Фурье-образ сигнала и .....                | 12 |
| 2.2.2 | Применяем фильтр .....                     | 12 |
| 2.2.3 | Возвращаемся к очищенному сигналу .....    | 12 |
| 2.2.4 | Выводы.....                                | 12 |
| 2.3   | Убираем низкие частоты?.....               | 12 |
| 3     | ЗАДАНИЕ 2. ФИЛЬТРАЦИЯ ЗВУКА .....          | 13 |
| 3.1   | Графики звуковых сигналов .....            | 13 |

## 1 ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ И БИБЛИОТЕКИ

Онлайн версию кода здесь нет, потому что делал основные вычисления в live-script матлабовских, в [репозитории](#) можно найти исходники. Так как в большинстве своём выполнение лабораторной работы сводилось к применению встроенных функций и построению красивых графиков, то не вижу смысла приводить отдельные фрагменты кода, потому что проще увидеть всю картину целиком - заглянуть в исходники, над их оформлением я тоже немного постарался.

## 2 ЗАДАНИЕ 1. ЖЁСТКИЕ ФИЛЬТРЫ

Для начала задаём константы  $a, b, c, d, t_1, t_2$ , что  $t_1 < t_2$ , После составляем функцию:

$$g(t) = \begin{cases} a, t \in [t_1; t_2] \\ 0, t \in [else] \end{cases}$$

Также задаём большой интервал времени  $T$  и маленький шаг дискретизации  $dt$ . На основе всего зашумлённая версия сигнала будет выглядеть так:

$$u = g + b*(\text{rand}(\text{size}(t))-0.5) + c*\sin(d*t);$$

### 2.1 Убираем высокие частоты

Берём  $d = c = 0$ . Тогда в этом пункте мы будем работать со следующей версией шумного сигнала:

$$u = g + b*(\text{rand}(\text{size}(t))-0.5)$$

...из чего сразу следует, что у нас добавляется только "случайный" шум.

### 2.1.1 Фурье-образ сигнала и

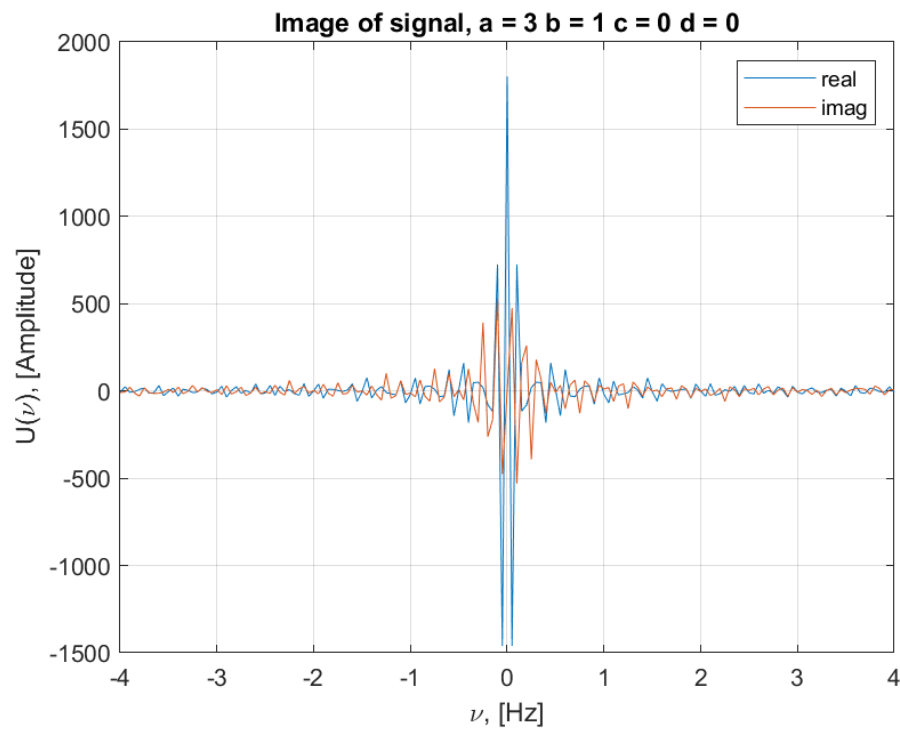


Рисунок 1 — Фурье-образ зашумлённого сигнала

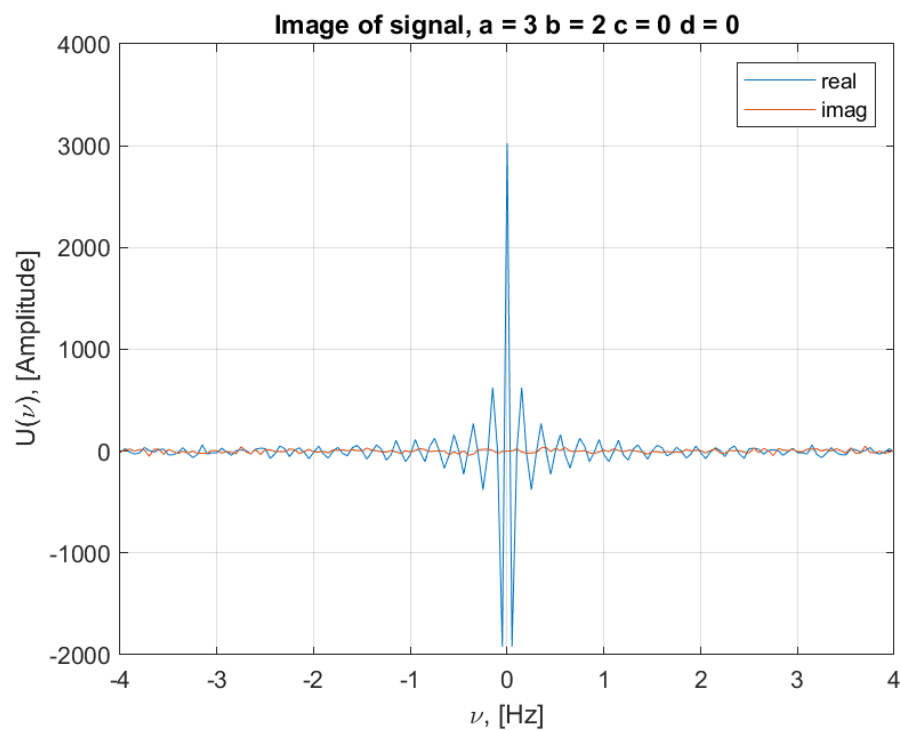


Рисунок 2 — Фурье-образ зашумлённого сигнала

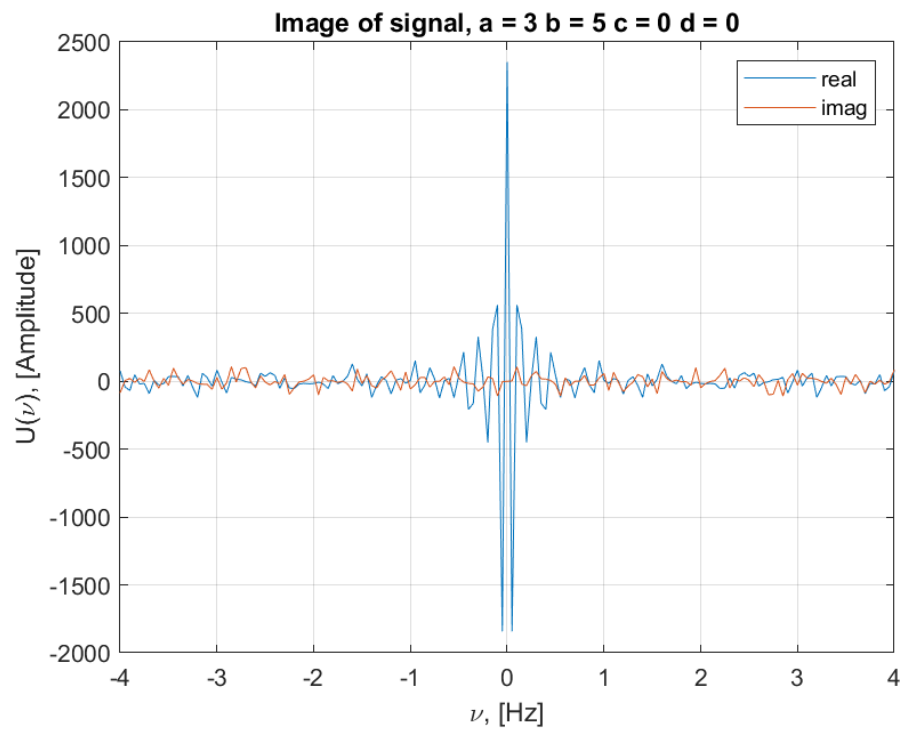


Рисунок 3 — Фурье-образ зашумлённого сигнала

### 2.1.2 Применяем фильтр

В нашем случае ступенька фильтра будет около середины, потому что мы сохраняем нижние частоты, а они сосредоточены около начала графика. Будем применять разные диапазоны фильтра:

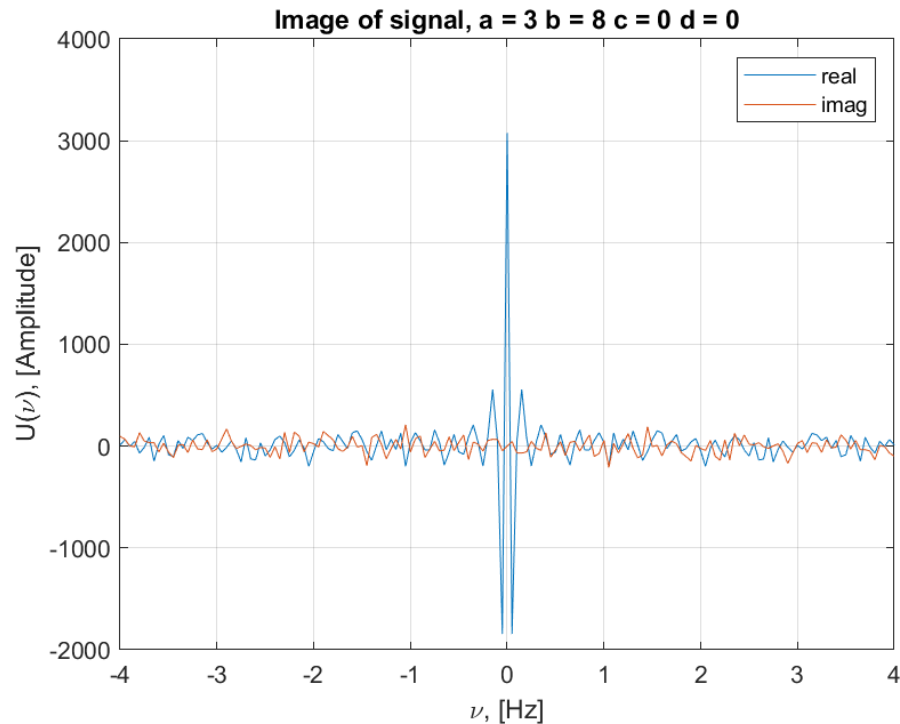


Рисунок 4 — Фурье-образ зашумлённого сигнала

### 2.1.3 Возвращаемся к очищенному сигналу

### 2.1.4 Выводы

Оставьте его неизменным для некоторого диапазона частот  $[-\nu_0, \nu_0]$ , но обнулите его значения на всех остальных частотах, после чего выполните обратное преобразование Фурье. Постройте сравнительные графики исходного и фильтрованного сигналов (для большей наглядности имеет смысл вывести на график лишь некоторую окрестность интервала  $[t_1, t_2]$ ). Постройте сравнительные графики модуля Фурье-образа исходного и фильтрованного сигналов. Исследуйте влияние частоты среза  $\nu_0$  и значения параметра  $b$  на эффективность фильтрации.

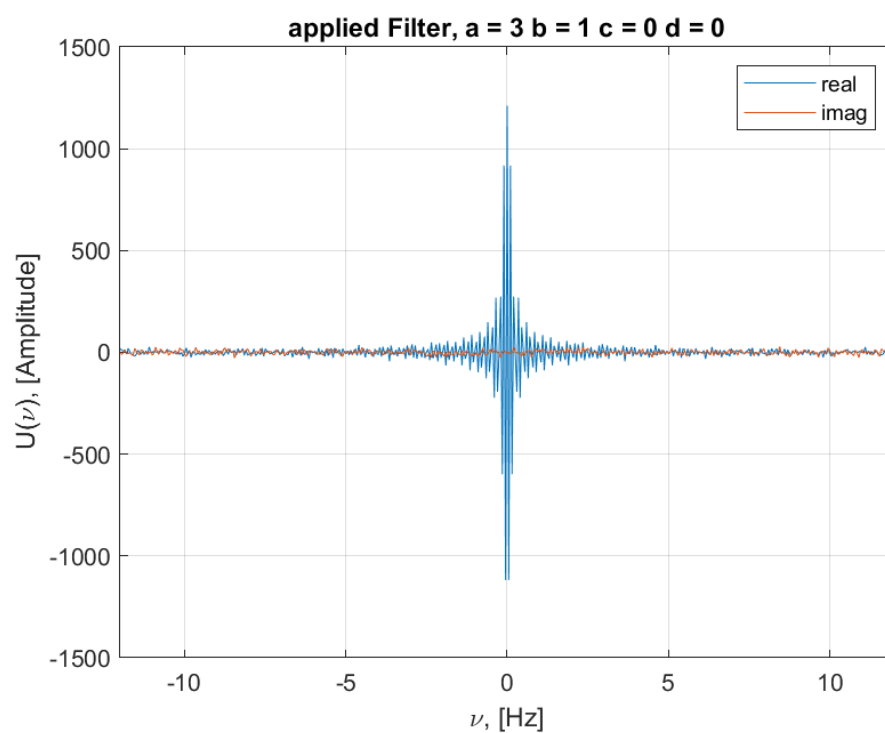


Рисунок 5 — Применили фильтр нижних частот

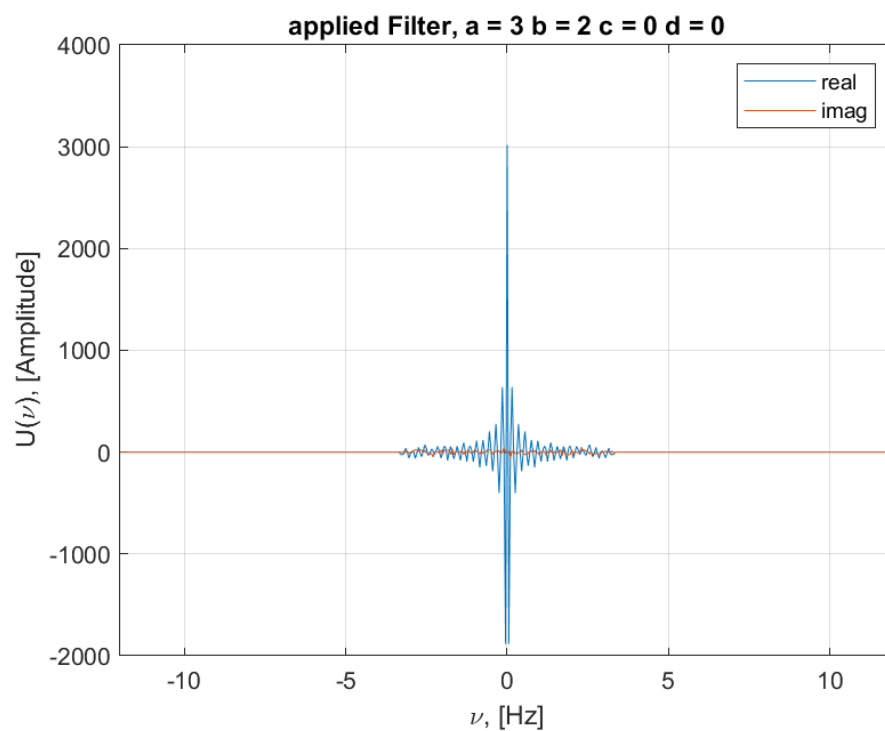


Рисунок 6 — Применили фильтр нижних частот



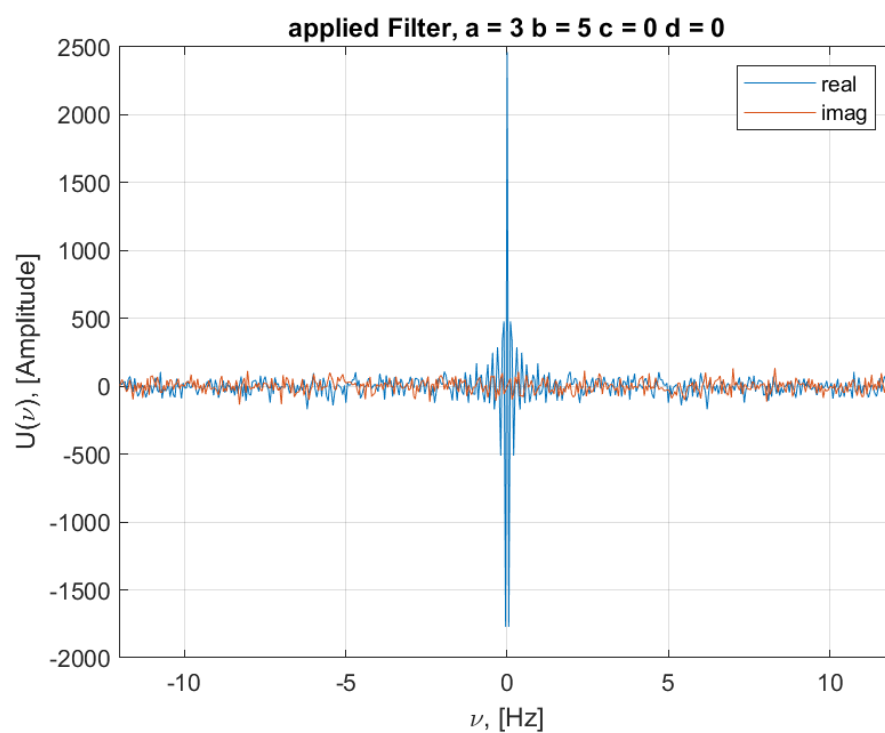


Рисунок 7 — Применили фильтр нижних частот

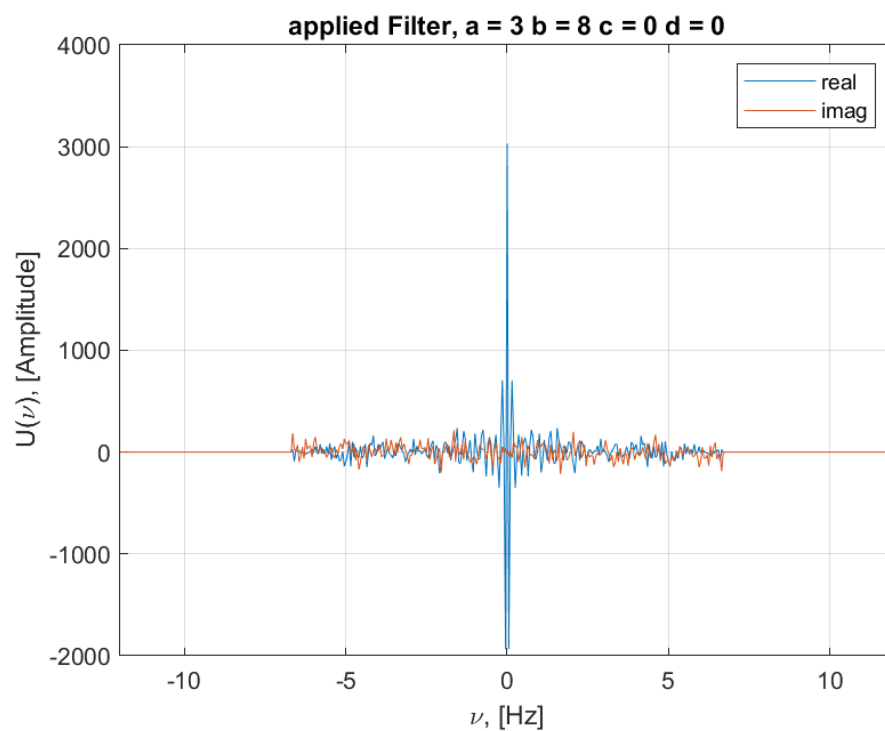


Рисунок 8 — Применили фильтр нижних частот

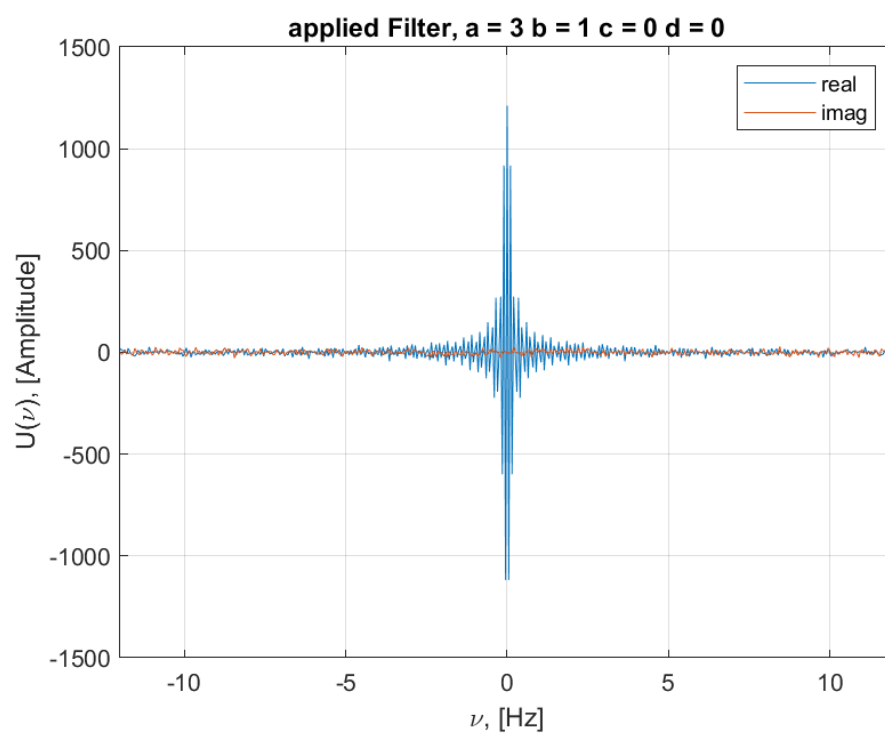


Рисунок 9 — Применили обратное преобразование Фурье

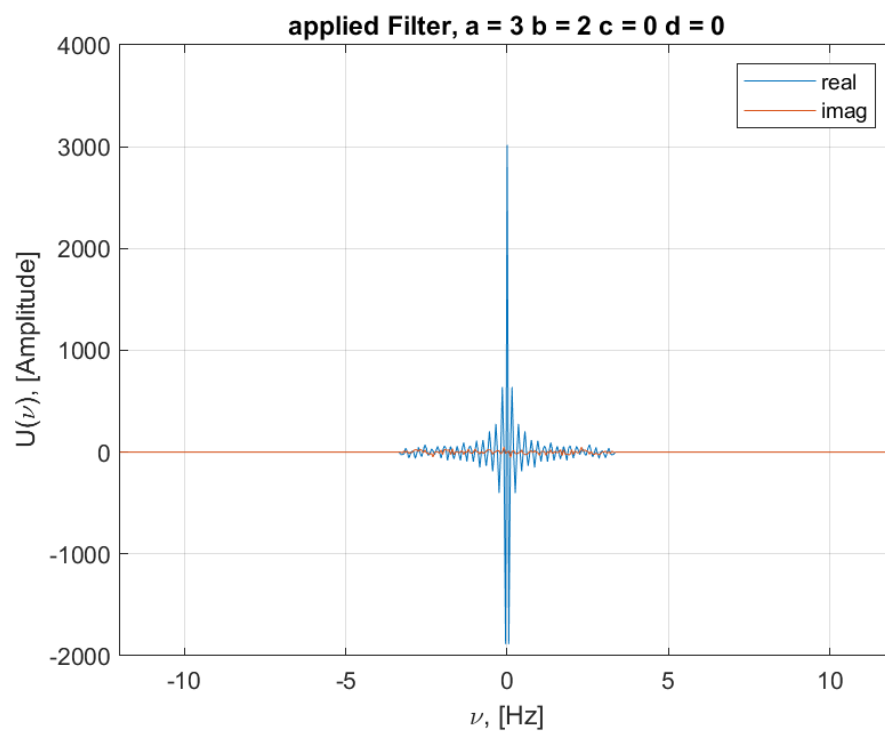


Рисунок 10 — Применили обратное преобразование Фурье

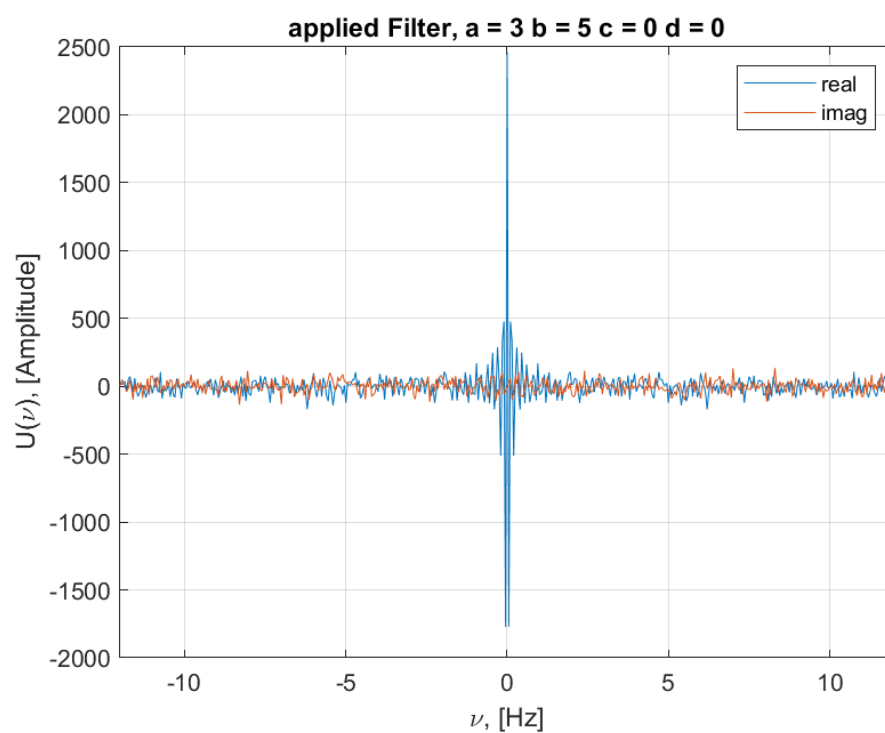


Рисунок 11 — Применили обратное преобразование Фурье

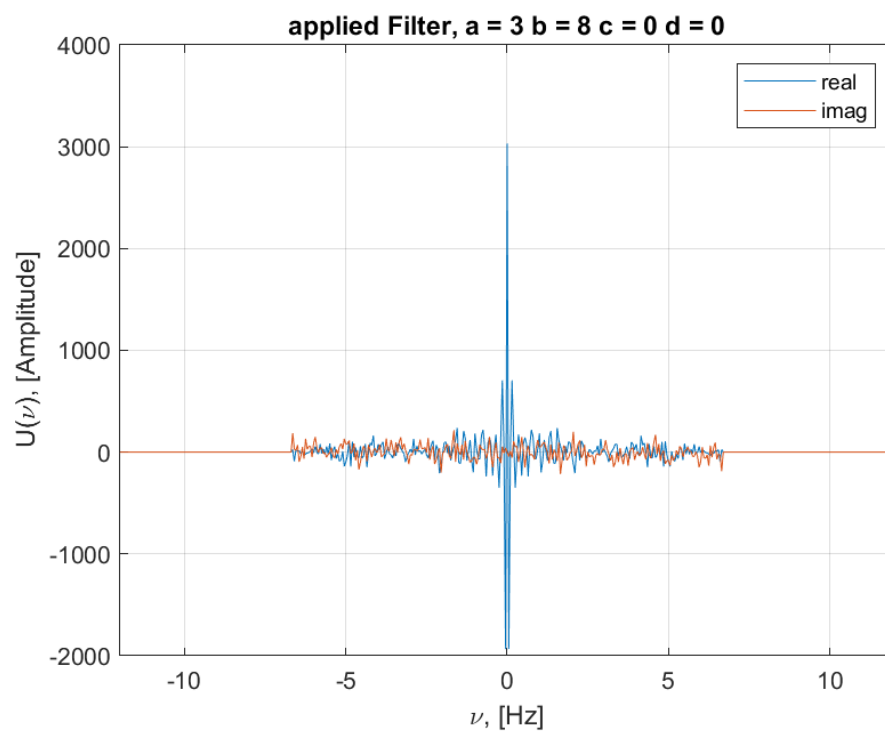


Рисунок 12 — Применили обратное преобразование Фурье

## **2.2 Убираем специфические частоты**

Выберем все параметры  $b, c, d$  ненулевыми. Теперь мы уже имеем дело с двумя компонентами шума - случайным и гармоническим. Соответственно, чтобы убрать обе компоненты, надо применить два фильтра, один из них мы уже нашли в пункте до.

### **2.2.1 Фурье-образ сигнала и**

### **2.2.2 Применяем фильтр**

Постараемся обнулить Фурье-образ на некоторых диапазонах частот, чтобы максимально убрать влияние обеих компонент помехи...

### **2.2.3 Возвращаемся к очищенному сигналу**

Постройте сравнительные графики исходного и фильтрованного сигналов, а также графики модулей их Фурье образов. Исследуйте влияние частот среза, а также значений параметров  $b, c, d$  на вид помехи и эффективность фильтрации (отдельно рассмотрите случай  $b=0$ ).

### **2.2.4 Выводы**

## **2.3 Убираем низкие частоты?**

Рассмотрите фильтр, который обнуляет Фурье-образ на всех частотах в некоторой окрестности точки  $\nu = 0$ . Пропустите сигнал через такой фильтр и оцените результат. Сделайте выводы.

### 3 ЗАДАНИЕ 2. ФИЛЬТРАЦИЯ ЗВУКА

Скачаем файл с [гугл-диска](#), а после немного [погуглим](#) и выясним, что ”Голос человека лежит в диапазоне 85-3000Гц поэтому стараться чистить будем всё, кроме этого промежутка. Очевидно, что нам понадобится два последовательно применённых фильтра, так и сделаем...

#### 3.1 Графики звуковых сигналов

Графики исходного и фильтрованного звукового сигнала, а также графики модулей их Фурье-образов

Не забывайте, что с помощью функции матлаба `sound(y,f)` - можно нативно проиграть звук после преобразований, чтобы сравнить ”до”и ”после”.