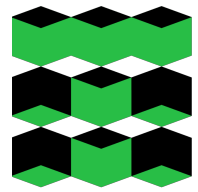
**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное автономное образовательное учреждение

высшего образования



**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа информационных технологий и робототехники

Отделение информационных технологий

Направление подготовки 09.04.04 Программная инженерия

**Отчёт по лабораторной работе №1**

**МЕТОДЫ ФИЛЬТРАЦИИ СИГНАЛОВ**

по дисциплине Основы теории управления автономными системами

Выполнил студент гр. 8ПМ4Л \_\_\_\_\_\_\_ \_\_\_\_\_\_ Сокуров Р.Е.

Подпись Дата Фамилия И.О.

Проверил доцент ОАР \_\_\_\_\_\_\_ \_\_\_\_\_\_ Хожаев И.В.

Подпись Дата Фамилия И.О.

Томск 2024 г.

**Цель**

Изучить основные методы фильтрации измерительных сигналов.

**Задачи**

1) Изучить принцип работы следующих фильтров:

– фильтра экспоненциального сглаживания;

– фильтра скользящего среднего;

– медианного фильтра;

– фильтра с ограничением скорости нарастания сигнала;

2) Сгенерировать полезный сигнал и добавить к нему равномерно распределенный шум и всплески большой амплитуды;

3) Реализовать каждый из ранее перечисленных фильтров любым известным способом, проверить работоспособность фильтров на сгенерированном ранее зашумленном сигнале и изучить влияние настроечного параметра фильтра на качество обработки сигнала;

4) Подобрать комбинацию фильтров, отделяющих полезный сигнал от шума обоих типов; сравнить исходный полезный сигнал и отфильтрованный;

5) Оформить отчет.

**Ход работы**

**1. Генерация полезного сигнала и добавление к нему равномерно распределённого шума и всплесков большой амплитуды.**

В качестве генерируемого сигнала была выбрана функция . Данная функция была рассчитана на массиве , состоящем из 800 элементов. Затем, с помощью библиотеки «NumPy» языка «Python» был добавлен равномерно распределённый шум, а также высокоамплитудные случайные всплески (Рисунок 1). Код отображён в файле «code.py» в приложении А.

Изображение выглядит как линия, График, диаграмма, Шрифт

Автоматически созданное описание

Рисунок 1 – Генерация зашумлённого сигнала

**2. Реализация фильтра экспоненциального сглаживания.**

Фильтр экспоненциального сглаживания – простейший фильтр низких частот, выходной сигнал которого равен взвешенной сумме последнего измеренного значения сигнала и значения отфильтрованного сигнала на предыдущем шаге:

,

где  – измеренные значения (зашумлённый сигнал),  – значения отфильтрованного сигнала,  – настроечный параметр фильтра.

Данный фильтр был реализован в функции «Exponential\_smoothing\_filter» в файле «code.py» приложения А.

Данная функция была использована с тремя разными значениями коэффициента :

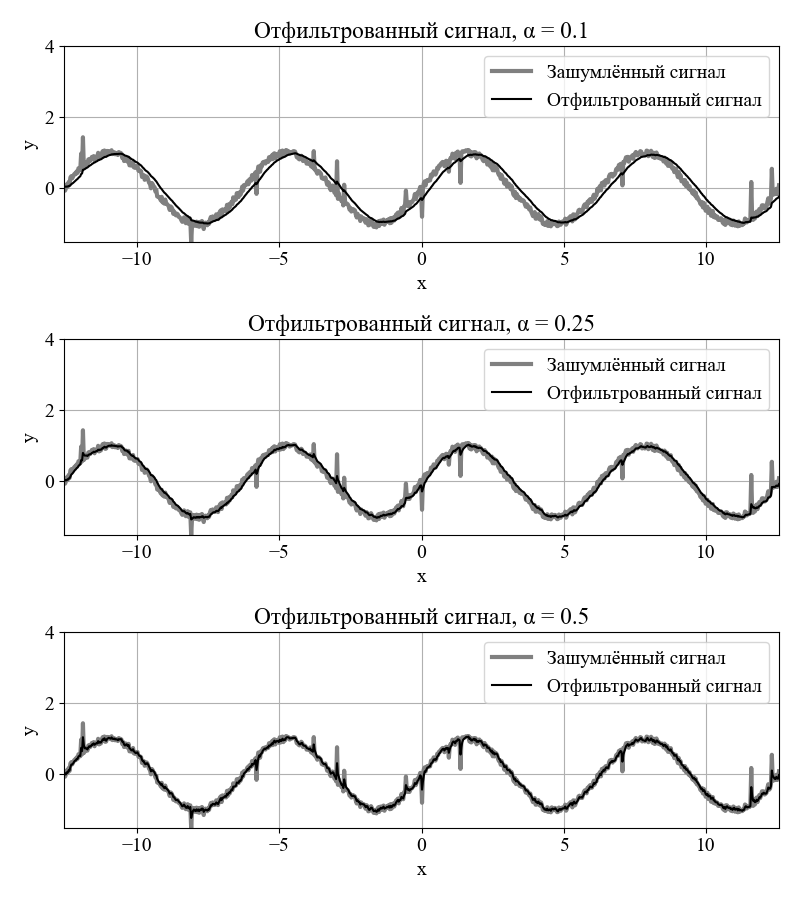


Рисунок 2 – Демонстрация работы фильтра экспоненциального сглаживания

По итогам анализа полученных графиков, оптимальным было принято значение , поскольку данное значение значительно сглаживает высокочастотные помехи в сигнале, но ещё не создаёт столь значимое запаздывание, как, например, значение .

**3. Реализация фильтра скользящего среднего**

Фильтр скользящего среднего является фильтром низких частот. Он хранит последние  измерений зашумленного сигнала и выдает на выход их среднее арифметическое.

Данный фильтр был реализован в функции «moving\_average\_filter» в файле «code.py» приложения А, значение :

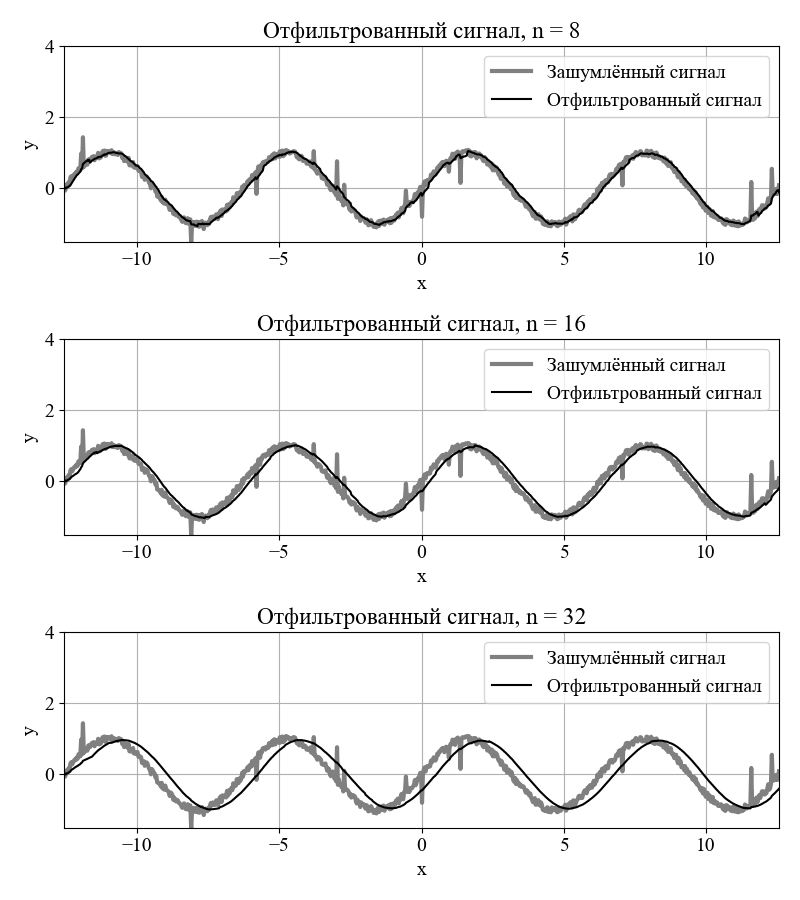


Рисунок 3 – Демонстрация работы фильтра скользящего среднего

Здесь наилучшим значением показал себя размер окна, равный 16, поскольку отставание отфильтрованного сигнала по вазе становится уже слишком большим при , а при  видны участки с резкими переходами.

**4. Реализация медианного фильтра**

Медианный фильтр является фильтром высоких частот. Его выходной сигнал равен медиане последних  измерений фильтруемого сигнала. Степень сглаживания сигнала также определяется шириной окна .

Данный фильтр был реализован в функции «median\_filter» в файле «code.py» приложения А, значение :

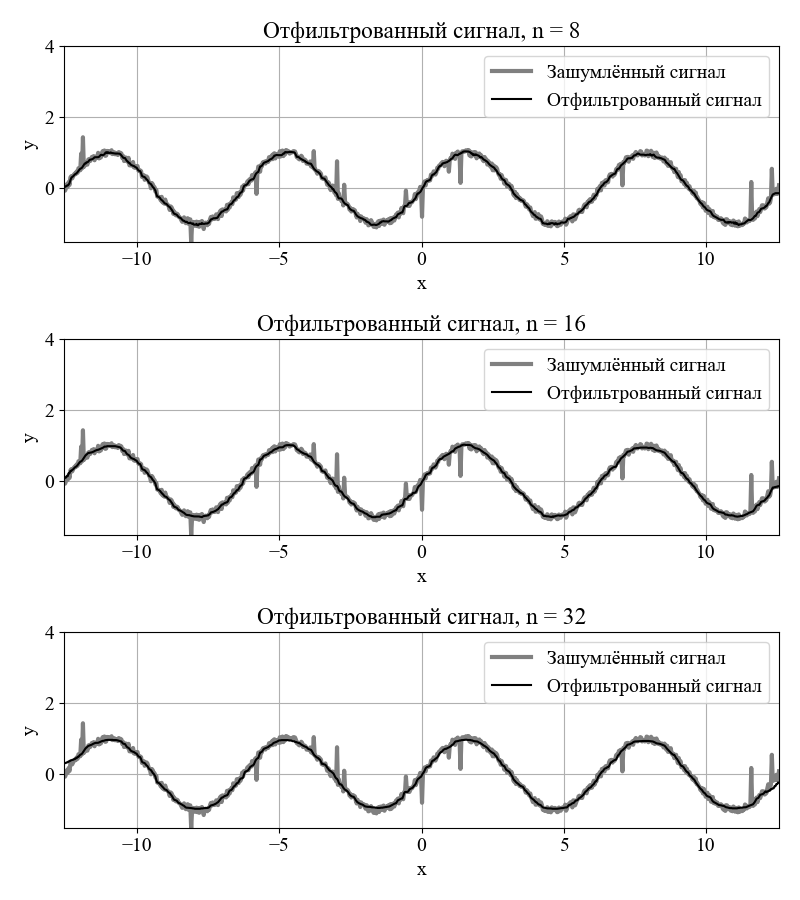


Рисунок 4 – Демонстрация работы медианного фильтра

Здесь, уже при размере окна  фильтр уже справился с низкими частотами (высокоамплитудных всплесков в итоговом сигнале нет), однако лучшим всё же показался размер окна , хотя стоит учесть, что фильтр может выдавать некорректные значения в начале (до того, как заполнится размер окна).

**5. Реализация фильтра с ограничением скорости нарастания сигнала**

Фильтр с ограничением скорости нарастания сигнала выводит на  
выход значение входного сигнала без изменений, если приращение между  
текущим и предыдущим его значением меньше заданного порогового  
значения. В противном случае значение выходного сигнала фильтра не  
изменяется.

Данный фильтр был реализован в функции «rate\_limit\_filter» в файле «code.py» приложения А, значение :

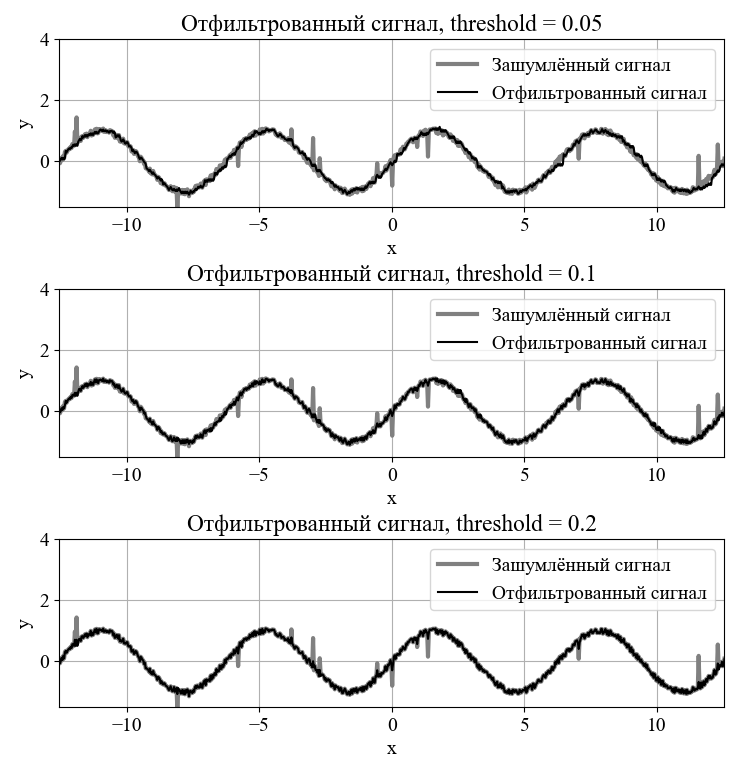


Рисунок 5 – Демонстрация работы фильтра с ограничением скорости нарастания

В данном фильтре, в отличие от всех предыдущих, именно увеличение значения  ведёт к большей схожести с оригинальным сигналом. Наилучшим значением было принято , поскольку при  уже появляются искажение сигнала.

**6. Реализация ансамбля (комбинации) фильтров**

Для реализации комбинации фильтров, которые лучше всех воспроизводят исходный сигнал из зашумленного, были выбраны медианный фильтр с  и фильтр экспоненциального сглаживания с :

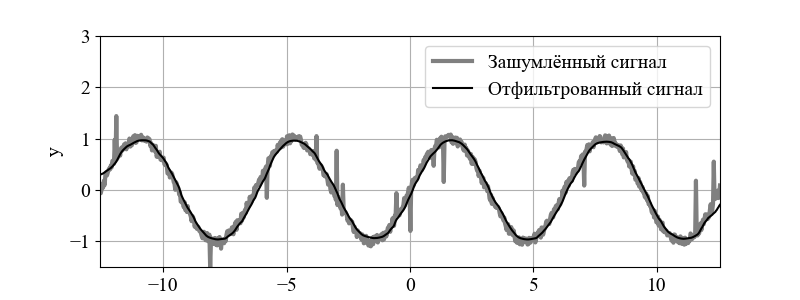


Рисунок 6 – Отфильтрованный двумя фильтрами сигнал

Несмотря на отклонение значений в начале и в конце (из-за особенностей работы медианного фильтра), данный ансамбль отлично справляется со своей задачей фильтрации, при условии, что интересующая нас информация находится не на краях графика.

# Заключение

В ходе данной лабораторной работы был составлен полезный сигнал , из которого, путём искусственного зашумления, получили материал для дальнейшей обработки фильтрами. Были исследованы 4 вида фильтров: фильтр экспоненциального сглаживания, фильтр скользящего среднего, медианный фильтр, фильтр с ограничением скорости нарастания сигнала. Каждый из них рассматривался при различных значениях настраиваемых параметров.

По итогам данных исследований было выбрано два фильтра, комбинация которых использовалась в п.6 работы для максимального приближения зашумлённого сигнала к исходному.

Приложение А

Реализация фильтров