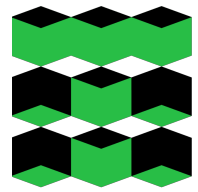
**Министерство науки и высшего образования Российской Федерации**

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 (Листинг 1).

Листинг 1 – Создание зашумлённого сигнала

1. x = np.linspace(-4\*np.pi, 4\*np.pi, 800) # Создание массива иксов

2. y = np.sin(x) # Функция

3. y\_clean = np.sin(x)

4. noise = np.random.uniform(-0.1,0.1,800) # Добавляем равномерно распределенный шум

5. spikes = np.random.choice([0, 1], size=x.shape, p=[0.98, 0.02])  # Случайные всплески (2% вероятность)

6. amplitude = np.random.uniform(-1, 1, size=x.shape)  # Случайная амплитуда всплесков

7. y += noise + spikes \* amplitude # дальше наш сигнал это зашумлённый сигнал

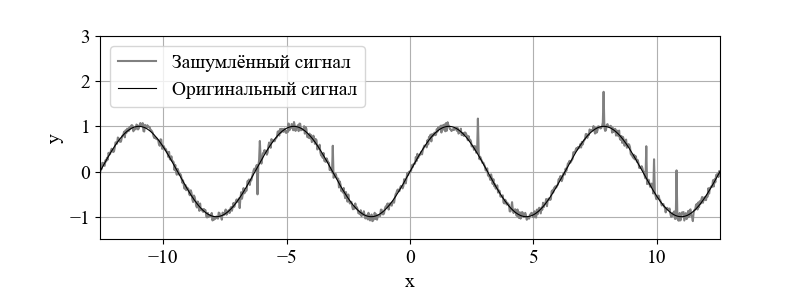


Рисунок 1 – Результат выполнения листинга 1