

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____
ПРЕПОДАВАТЕЛЬ

Ассистент
должность, уч. степень, звание

подпись, дата

Н.А. Янковский

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №6

КИХ фильтры

Вариант 5

по курсу: Цифровая обработка и передача сигналов

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № _____ 4128

подпись, дата

В. А. Воробьев

инициалы, фамилия

Санкт-Петербург 2023

1 Задание

Исходные данные:

$f = 3N$, $T = 10/F$, где N – номер по списку.

Написать программу, которая позволит:

1. Провести дискретизацию функции $u(t) = \sin(2\pi ft)$ на заданном интервале с частотой дискретизации $15f$.
2. Добавить к полученным дискретным отсчетам u_1, \dots, u_n выборку случайной величины x_1, \dots, x_n , $X \sim N(0, 0.2)$.
3. Реализовать фильтрацию полученного вектора значений двумя способами:
 1. Медианным фильтром 4, 6, 8 порядка
 2. Скользящим средним 4, 6, 8 порядка
4. Рассчитать среднеквадратичное отклонение полученных после фильтрации значений от исходного сигнала без шума.

2 Результат работы

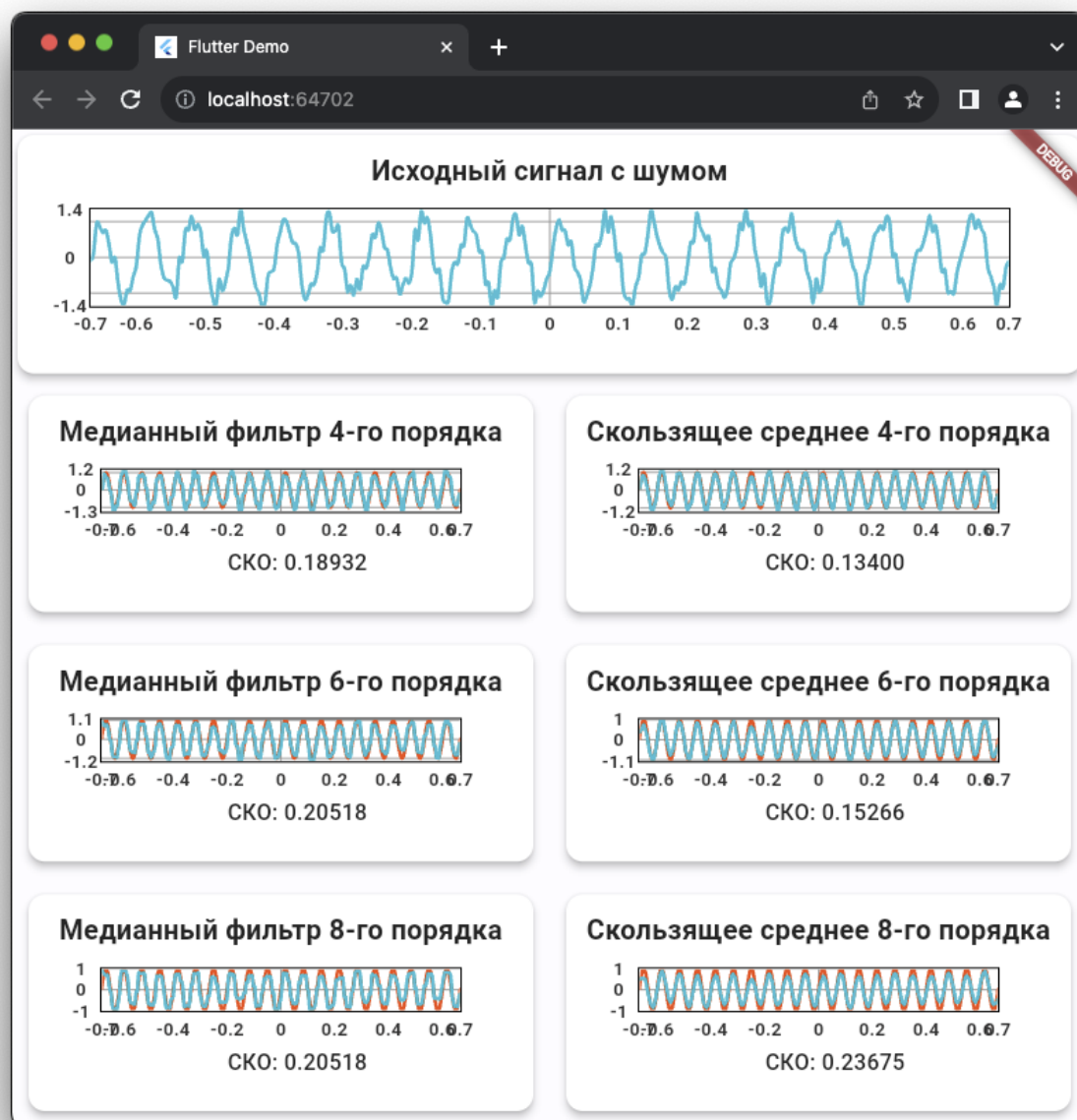


Рисунок 1 – Графики

3 Вывод

В ходе выполнения лабораторной работы была написана программа, которая проводит дискретизацию функции, добавляет шумы, после чего реализует фильтрацию медианным фильтром 4, 6, 8 порядка и скользящим средним 4, 6, 8 порядка. Также программа вычисляет среднеквадратичное отклонение получившихся после фильтрации сигналов от исходного сигнала. Наиболее близким к исходному сигналу, т.е. имеющим наименьшее среднеквадратичное отклонение оказался сигнал, прошедший фильтрацию скользящим средним четвёртого порядка. Для всех сигналов были построены графики.

Таким образом в ходе выполнения работы была изучена и применена на практике фильтрация сигнала медианным фильтром и скользящим средним.

ПРИЛОЖЕНИЕ А

```
preview_app.dart
import 'package:extend_math/extend_math.dart';
import 'package:flutter/material.dart';
import 'package:lab6/logic/calculations.dart';
import 'package:ui_kit/ui_kit.dart';
```

```
class PreviewApp extends StatelessWidget {
  const PreviewApp({super.key});
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return Column(
```

```
      children: [
```

```
        Expanded(
```

```
          child: KitTitleContainer(
```

```
            title: "Исходный сигнал с шумом",
```

```
            child: KitLineChart(
```

```
              lines: [
```

```
                KitLineData(
```

```
                  dots: Calculations.noisedDots
```

```
                    .map((e) => KitDot(e.x, e.y))
```

```
                    .toList(),
```

```
                ),
```

```
              ],
```

```

    ),
    ),
    ),
Expanded(
  flex: 3,
  child: Row(
    children: [
      Expanded(
        child: KitColumn(
          childFit: FlexFit.tight,
          children: [
            _PreviewFilter(
              title: "Медианный фильтр 4-го порядка",
              Calculations.medianFiltered4,
              Calculations.median4Diff,
            ),
            _PreviewFilter(
              title: "Медианный фильтр 6-го порядка",
              Calculations.medianFiltered6,
              Calculations.median6Diff,
            ),
            _PreviewFilter(
              title: "Медианный фильтр 8-го порядка",
              Calculations.medianFiltered8,
              Calculations.median6Diff,
            ),

```

```

    ],
  ),
),
Expanded(
  child: KitColumn(
    childFit: FlexFit.tight,
    children: [
      _PreviewFilter(
        title: "Скользящее среднее 4-го порядка",
        Calculations.movingFiltered4,
        Calculations.avg4Diff,
      ),
      _PreviewFilter(
        title: "Скользящее среднее 6-го порядка",
        Calculations.movingFiltered6,
        Calculations.avg6Diff,
      ),
      _PreviewFilter(
        title: "Скользящее среднее 8-го порядка",
        Calculations.movingFiltered8,
        Calculations.avg8Diff,
      ),
    ],
  ),
),
),
],

```

```

        ),
    ),
],
);
}
}

```

```

class _PreviewFilter extends StatelessWidget {

```

```

    final String title;

```

```

    final List<Point2> points;

```

```

    final double diff;

```

```

    const _PreviewFilter(

```

```

        this.points,

```

```

        this.diff, {

```

```

        required this.title,

```

```

    });

```

```

    @override

```

```

    Widget build(BuildContext context) {

```

```

        return KitTitleContainer(

```

```

            title: title,

```

```

            child: KitLineChart(

```

```

                xAxisName: "CKO: ${diff.toStringAsFixed(5)}",

```

```

                lines: [

```

```

                    KitLineData(

```



```

        color: Colors.deepOrange,
        dots: Calculations.dots.map((e) => KitDot(e.x, e.y)).toList(),
    ),
    KitLineData(
        dots: points.map((e) => KitDot(e.x, e.y)).toList(),
    ),
],
),
);
}
}

```

variant.dart

```

import 'package:extend_math/extend_math.dart';
import 'dart:math';

```

```

abstract final class Variant {
    static const _n = 5;
    static const fParam = 3 * _n;
    static const T = 10 / fParam;
    static const step = 1 / 15.0 / fParam;
    static const interval = MathInterval(-T, T);

    static double fn(double x) => sin(2 * pi * fParam * x);
}

```

```

calculations.dart
import 'dart:math';

import 'package:extend_math/extend_math.dart';
import 'package:lab6/logic/variant.dart';

abstract final class Calculations {
  static final random = Random();
  static final dots = Variant.interval.applyFx(Variant.fn, step: Variant.step);
  static final noisedDots =
    dots.map((e) => Point2(e.x, e.y + random.nextDouble() * 0.8 - 0.4)).toList();
  static final xDots = noisedDots.xDots;

  // Filtered
  static final medianFiltered4 =
    FiltersList.medianFilter(noisedDots.yDots, 3).joinX(xDots);
  static final medianFiltered6 =
    FiltersList.medianFilter(noisedDots.yDots, 5).joinX(xDots);
  static final medianFiltered8 =
    FiltersList.medianFilter(noisedDots.yDots, 7).joinX(xDots);
  static final movingFiltered4 =
    FiltersList.movingAverageFilter(noisedDots.yDots, 3).joinX(xDots);
  static final movingFiltered6 =
    FiltersList.movingAverageFilter(noisedDots.yDots, 5).joinX(xDots);
  static final movingFiltered8 =
    FiltersList.movingAverageFilter(noisedDots.yDots, 7).joinX(xDots);

```

```

// Diff
static final origDiff = MathList.calculateRMSE(dots.yDots, noisedDots.yDots);
static final median4Diff =
    MathList.calculateRMSE(dots.yDots, medianFiltered4.yDots);
static final median6Diff =
    MathList.calculateRMSE(dots.yDots, medianFiltered6.yDots);
static final median8Diff =
    MathList.calculateRMSE(dots.yDots, medianFiltered8.yDots);
static final avg4Diff =
    MathList.calculateRMSE(dots.yDots, movingFiltered4.yDots);
static final avg6Diff =
    MathList.calculateRMSE(dots.yDots, movingFiltered6.yDots);
static final avg8Diff =
    MathList.calculateRMSE(dots.yDots, movingFiltered8.yDots);
}

```

main.dart

```

import 'package:flutter/material.dart';
import 'package:lab6/ui/preview_app.dart';

```

```

void main() {
    runApp(const MyApp());
}

```

```

class MyApp extends StatelessWidget {

```

```

const MyApp({super.key});

// This widget is the root of your application.
@override
Widget build(BuildContext context) {
  return MaterialApp(
    title: 'Flutter Demo',
    theme: ThemeData(
      colorScheme: ColorScheme.fromSeed(seedColor: Colors.blue),
      useMaterial3: true,
    ),
    home: const Scaffold(
      body: PreviewApp(),
    ),
  );
}
}

```

```

web_plugin_registrant.dart
// Flutter web plugin registrant file.
//
// Generated file. Do not edit.
//

// ignore_for_file: type=lint

```

```
void registerPlugins() {}
```