ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ _____
ПРЕПОДАВАТЕЛЬ

Ассистент		Н.А. Янковский
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8

Восстановление аналогового сигнала

Вариант 5

по курсу: Цифровая обработка и передача сигналов

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4128		В. А. Воробьев
	подпись, дата	инициалы, фамилия

Санкт-Петербург 2023

1 Задание

Исследовать процедуру дискретизации синусоидального сигнала $u(t)$ с частотой F Гц. Длительность наблюдения сигнала $10/F$ секунд.

Написать программу, которая позволит:

1. Сформировать выборки отсчетов $u^{(i)}[n]$ (результаты дискретизации) исследуемого сигнала с частотами дискретизации $f^{(i)}_d$ равными:
   a. $1.5F$ Гц;
   b. $1.75F$ Гц;
   c. $2F$ Гц;
   d. $3F$ Гц;
   e. $1000F$ Гц.
2. Применить ряд Котельникова для восстановления исходного сигнала по его дискретным отсчетам.
3. Вывести графики исходного сигнала и восстановленных сигналов $u^{(i)}(t)$, где i — индекс частоты дискретизации.

   Сделать выводы о точности восстановления исследуемого сигнала $u(t)$ при использовании различных частот дискретизации. Объяснить, чем вызваны искажения на краях построенных графиков.

2 Результат работы

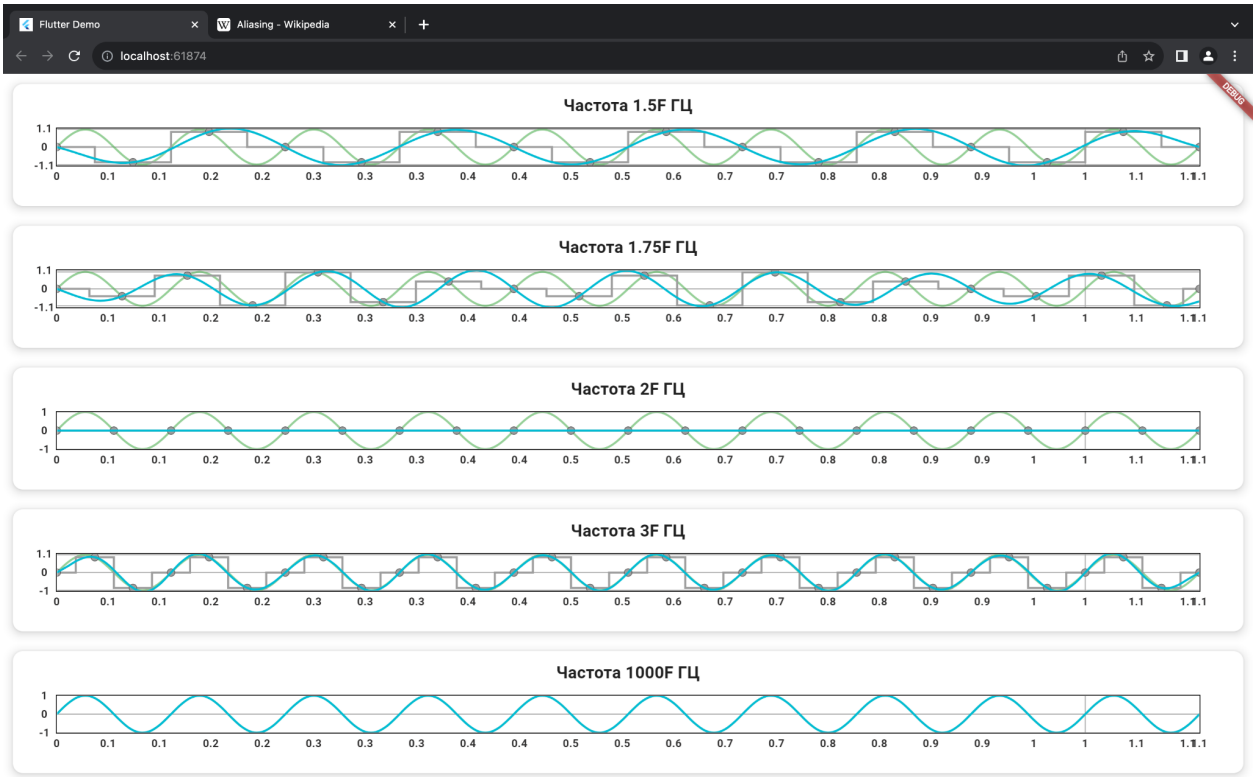

Рисунок 1 – Графики

3 Вывод

В результате лабораторной работы мы исследовали процедуру дискретизации синусоидального сигнала с использованием различных частот дискретизации. Мы разработали программу для формирования выборок отсчетов и восстановления исходного сигнала с применением ряда Котельникова. Исходный код был загружен на GitHub(URL: https://github.com/vladcto/suai-labs/tree/6686f91be498b3187506cf6730b6dc13df848bd1/5_semester/%D0%A6%D0%9E%D0%9F%D0%A1/lab8)

Были созданы графики исходного сигнала и восстановленных сигналов для каждой частоты дискретизации. В ходе анализа графиков мы обратили внимание на искажения на краях восстановленных сигналов.

Из проведенного исследования можно сделать выводы:

1) Точность восстановления исследуемого сигнала зависит от частоты дискретизации. В соответствии с теоремой Котельникова, зная частоту самой высокой гармоники спектра, сигнал можно восстановить со 100 процентной точностью.
2) Искажения на краях графиков восстановленных сигналов объясняются явлением алиасинга, которое проявляется при недостаточно высокой частоте дискретизации для корректного восстановления высокочастотных компонент сигнала.

ПРИЛОЖЕНИЕ

preview_app.dart

```
import 'package:flutter/material.dart';
import 'package:lab8/logic/calculations.dart';
import 'package:ui_kit/ui_kit.dart';

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

  @override
  Widget build(BuildContext context) {
    return KitColumn(
      childFit: FlexFit.tight,
      children: [
        KitTitleContainer(
          title: "Частота 1.5F ГЦ",
          child: KitLineChart(
            lines: [
              KitLineData(
                dots: Calculations.discrete5.toKitDot,
                color: Colors.green.withAlpha(150),
              ),
              _DiscreteLineData(dots: Calculations.discrete1.toKitDot),
              KitLineData(dots: Calculations.restored1.toKitDot)
            ],
          ),
        ),
        KitTitleContainer(
          title: "Частота 1.75F ГЦ",
```

```
  child: KitLineChart(
    lines: [
      KitLineData(
        dots: Calculations.discrete5.toKitDot,
        color: Colors.green.withAlpha(150),
      ),
      _DiscreteLineData(dots: Calculations.discrete2.toKitDot),
      KitLineData(
        dots: Calculations.restored2.toKitDot,
      )
    ],
  ),
),
KitTitleContainer(
  title: "Частота 2F ГЦ",
  child: KitLineChart(
    lines: [
      KitLineData(
        dots: Calculations.discrete5.toKitDot,
        color: Colors.green.withAlpha(150),
      ),
      _DiscreteLineData(dots: Calculations.discrete3.toKitDot),
      KitLineData(dots: Calculations.restored3.toKitDot)
    ],
  ),
),
KitTitleContainer(
  title: "Частота 3F ГЦ",
  child: KitLineChart(
    lines: [
```

```
            KitLineData(
              dots: Calculations.discrete5.toKitDot,
              color: Colors.green.withAlpha(150),
            ),
            _DiscreteLineData(dots: Calculations.discrete4.toKitDot),
            KitLineData(dots: Calculations.restored4.toKitDot)
          ],
        ),
      ),
      KitTitleContainer(
        title: "Частота 1000F ГЦ",
        child: KitLineChart(
          lines: [
            KitLineData(
              dots: Calculations.discrete5.toKitDot,
              color: Colors.green.withAlpha(150),
            ),
            KitLineData(dots: Calculations.restored5.toKitDot)
          ],
        ),
      ),
    ],
  );
 }
}

class _DiscreteLineData extends KitLineData {
 _DiscreteLineData({required super.dots})
     : super(
        color: Colors.grey,
```

```
        isStepped: true,
        showDots: true,
      );
}
```

variant.dart

```
import 'dart:math';

import 'package:extend_math/extend_math.dart';

abstract final class Variant{
  static const variant = 9;
  static const frequency1 = 1.5 * variant;
  static const frequency2 = 1.75 * variant;
  static const frequency3 = 2.1 * variant;
  static const frequency4 = 3.0 * variant;
  static const frequency5 = 1000.0 * variant;
  static const interval = MathInterval(0, 10 / variant);
  static double fx(double x) => sin(2 * pi * variant * x);
}
```

calculations.dart

```
import 'package:extend_math/extend_math.dart';
import 'package:lab8/logic/variant.dart';
import 'package:ui_kit/ui_kit.dart';

abstract final class Calculations {
  static final restoreDots = Variant.interval.applyFx((x) => x, step: 0.01);

  // Discrete
  static final discrete1 =
```

```
    Variant.interval.applyFx(Variant.fx, step: 1 / Variant.frequency1);
  static final discrete2 =
    Variant.interval.applyFx(Variant.fx, step: 1 / Variant.frequency2);
  static final discrete3 =
    Variant.interval.applyFx(Variant.fx, step: 1 / Variant.frequency3);
  static final discrete4 =
    Variant.interval.applyFx(Variant.fx, step: 1 / Variant.frequency4);
  static final discrete5 =
    Variant.interval.applyFx(Variant.fx, step: 1 / Variant.frequency5);

  // Reconstruct
  static final restored1 = ReconstructSignal.reconstructSignal(
    discrete1, Variant.frequency1, restoreDots);
  static final restored2 = ReconstructSignal.reconstructSignal(
    discrete2, Variant.frequency2, restoreDots);
  static final restored3 = ReconstructSignal.reconstructSignal(
    discrete3, Variant.frequency3, restoreDots);
  static final restored4 = ReconstructSignal.reconstructSignal(
    discrete4, Variant.frequency4, restoreDots);
  static final restored5 = ReconstructSignal.reconstructSignal(
    discrete5, Variant.frequency5, restoreDots);
}

extension Point2ToKitDot on List<Point2> {
  List<KitDot> get toKitDot => map((e) => KitDot(e.x, e.y)).toList();
}
```

main.dart

```
import 'package:flutter/material.dart';
import 'package:lab8/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(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a blue toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can be
        // tested with just a hot reload.
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
```

```
        useMaterial3: true,
      ),
      home: const PreviewApp(),
    );
  }
}
```

web_plugin_registrant.dart

```
// Flutter web plugin registrant file.
//
// Generated file. Do not edit.
//

// ignore_for_file: type=lint

void registerPlugins() {}
```