Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №1.1/1.2

з дисципліни
«Інтелектуальні вбудовані системи»

на тему
«Дослідження і розробка моделей випадкових сигналів. Аналіз їх характеристик»

Виконала:

студентка групи ІП-84
Д'яконенко Дар'я
Номер заліковки: 8406

Перевірив:

Регіда П. Г.

Київ 2021

Основні теоретичні відомості

СРЧ обов'язково пов'язані з деякою зовнішнім середовищем. СРЧ забезпечує контроль за зміною параметрів зовнішнього середовища і в ряді випадків забезпечує управління параметрами середовища через деякі впливу на неї. Параметри середовища представляються деякою зміною фізичного середовища. При вимірах фізичного параметра ми отримуємо певний електричний сигнал на вході вимірювального датчика. Для подання такого електричного сигналу можна використовувати різні моделі.

Формула мат. очікування:

$$M_x = \lim_{N\to\infty} \frac{1}{N} \cdot \sum_{i=1}^{N} x_i(t_k) = \lim_{n\to\infty} \frac{1}{N} \sum_{k=0}^{n} x_i(t_k)$$

Формула дисперсії:

$$D_x = \lim_{N\to 0} \cdot \frac{1}{N-1} \sum_{i=1}^{N} x_i(t_k) - M_x)^2 = \lim_{n\to\infty} \frac{1}{n-1} \cdot \sum_{k=0}^{n} (x_i(t_k) - M_x)^2 \geq 0$$

Обчислення значення автокореляційної ф-ії:

$$R_{xx}(t,\tau_s) = \lim_{N\to\infty} \frac{1}{N-1} \sum_{i=1}^{N} (\overbrace{x_i(t_k) - M_x(t_k)}^{\overset{0}{x(t_k)}}) \cdot (\overbrace{x_i(t_k+\tau_s) - M_x(t_k+\tau_s)}^{\overset{0}{x(t_k+\tau_s)}})$$

Обчислення значення взаєинокореляційної ф-ії:

$$R_{xy}(\tau) = \lim_{n\to 0} \cdot \frac{1}{n-1} \cdot \sum_{i=1}^{n} (\underbrace{x_i(t_k) - M_x}_{X(t_k)}) \cdot (\underbrace{y(t_k+\tau) - M_y}_{y(t_k-\tau)})$$

Умови завдання

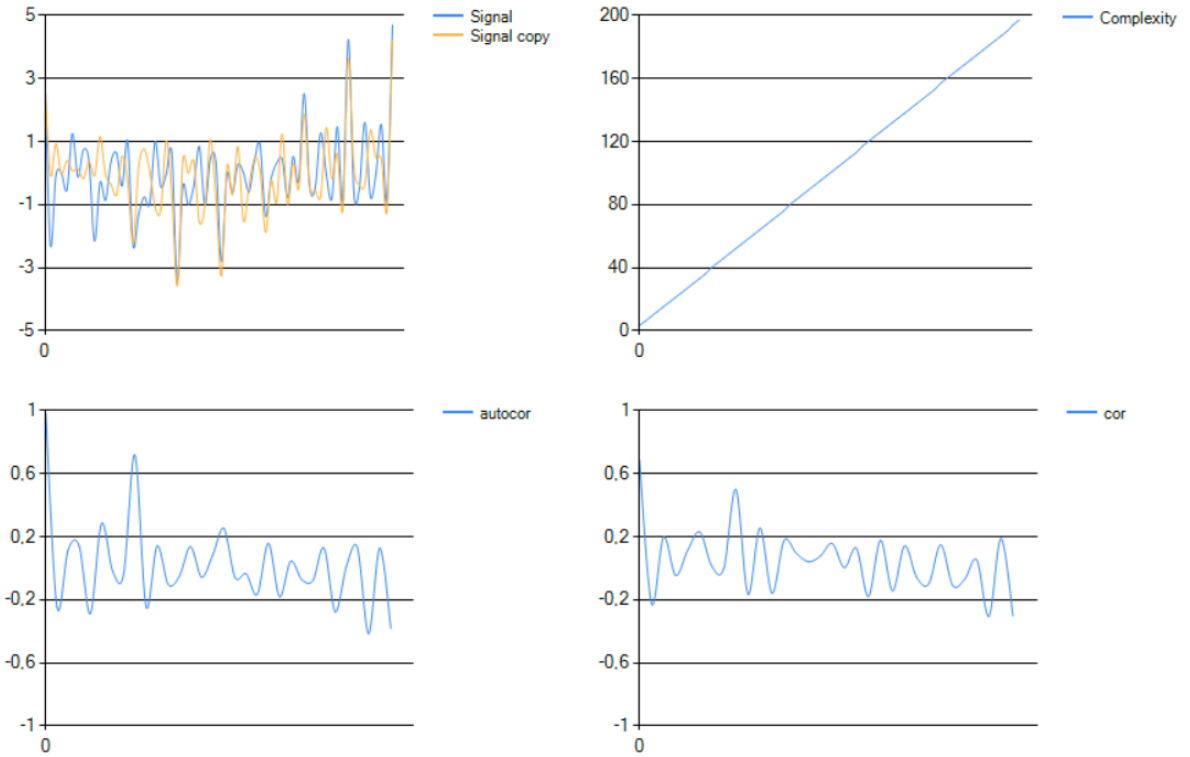
Варіант: 6, число гармонік в сигналі: 12, гранична частота: 1800, кількість дискретних відліків 64.

Лістинг програми

```
public class Program
    {

        public const int HARMONICS_NUMBER = 12;
        public const int BORDER_FREQUENCY = 1800;
        public const int CALLS_NUMBER = 64;

        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }

        public static double calculateM(double[] arr)
        {
            double sum = 0;

            for (int i = 0; i < arr.Length; i++)
            {
                sum += arr[i];
            }

            return sum / arr.Length;
        }

        public static double calculateD(double M, double[] arr)
        {
            double sum = 0;

            for (int i = 0; i < arr.Length; i++)
            {
                sum += ((arr[i] - M) * (arr[i] - M));
            }

            return sum / (arr.Length - 1);
        }

        public static double[] generateSignal(int CALLS_NUMBER, int HARMONICS_NUMBER, int
BORDER_FREQUENCY)
        {
            double[] signals = new double[CALLS_NUMBER];
            var rand = new Random();

            for (int i = 1; i <= HARMONICS_NUMBER; i++) {
                double frequency = (i*BORDER_FREQUENCY) / HARMONICS_NUMBER;
                double amplitude = rand.NextDouble();
                Thread.Sleep(25); // to refresh random seed
                double phase = rand.NextDouble();
                for(int time = 0; time < CALLS_NUMBER; time++)
                {
                    double signal = amplitude * Math.Sin((frequency * time) + phase);
                    signals[time] += signal;
                }
            }

            return signals;
        }


        public static double[] corFunc(double[] x, double[] y)
```

```
{
    double[] corValues = new double[CALLS_NUMBER/2];
    double avgX = x.Average();
    double avgY = y.Average();
    double stdevX = Math.Sqrt(calculateD(avgX, x));

    int half = CALLS_NUMBER / 2;
    double v, m, std;
    for(int i = 0; i<half; i++)
    {
        v = 0;
        for(int j = 0;j< x.Length-i; j++)
        {
            v += x[j] * y[j + i];
            m = y.Average();
            std = Math.Sqrt(calculateD(avgY,y));
            corValues[i] = (v / (x.Length - i) - avgX * m) / (stdevX * std);
        }
    }


    return corValues;
}

public static double[] autoCorFunc(double [] signals)
{
    return corFunc(signals, signals);
}

public static double[] complexity()
{
    double[] time = new double[CALLS_NUMBER];
    Stopwatch sw = new Stopwatch();

    for (int i = 0; i < CALLS_NUMBER; i++)
    {
        sw.Start();
        generateSignal(i, HARMONICS_NUMBER, BORDER_FREQUENCY);
        sw.Stop();
        time[i] = sw.ElapsedMilliseconds/100;
    }
    return time;
}
   }
```

Результат виконання програми



M = 0,00221918907017556

D = 1,93499597288333

Висновки

Під час виконання лабораторної роботи був згенерований випадковий сигнал з випадковими амплітудою та частотою та досліджені його дисперсія та маточікування, побудований графік складності алгоритму. Також були досліджені та побудовані авто- та взаємнокореляційні функції.