**БАЛТИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ «ВОЕНМЕХ» им. Д.Ф. Устинова**

**Факультет информационных и управляющих систем**

**Кафедра И4 Радиоэлектронных систем управления**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7**

по учебной дисциплине

"Основы компьютерного проектирования и моделирования"

Вариант 12

Студент: Мечеткин Д.Д.

Группа: И4М31

**ПРЕПОДАВАТЕЛЬ**

\_\_\_\_\_\_\_\_\_/Петров Ю.В./

"\_\_\_\_"\_\_\_\_\_\_\_\_\_2019 г.

Санкт-Петербург  
2019 г.

**Задание к лабораторной работе №7**

**Моделирование нестационарных случайных процессов**

Произвести моделирование случайных значений радиосигнала (помехи),

нестационарного по корреляционной функции

Закон распределения - нормальный

Корреляционная функция: S^2\*exp(-Alpha\*|tau|)

Параметр корреляционной функции Alpha меняется по закону:

Alpha(t) = Alpha \* [ 1 + 10 \* B(t) ]

B(t) = | 1 , n \* T <= t < n \* T + T/2,

| 0 , n \* T + T/2 <= t < (n+1) \* T ,n = 0,1,2,...

где

Alpha(t)-Параметр корреляционной функции Alpha,

Alpha - его номинальное значение

Математическое ожидание М=-64.1 мкВ

Параметр Alpha=0.14 1/c

Среднеквадратическое отклонение S=120.540мкВ

Период дискретизации dt=0.6 с

Время моделирования Tmod=600 с

Период T=150 с

**Листинг программы** :

nit Lr7data;

interface

Const N = 1000; { Размер реализации }

Var i : integer; { Переменная цикла }

T : Array[1..1000] of real;

{ Массив значений сигналов }

predY, Y : real; { Предыдущее и текущее рассчитанное значение }

p : real; { параметр Р }

Const

dt = 0.6; { Интервал дискретизации по времени, с }

Tmod = 600; { Время моделирования, с }

Period = 150; { Период, с }

Alpha = 0.14; { Параметр, 1/с }

M = -64.1; { Математическое ожидание, мкВ }

S = 120.54; { Среднеквадратическое отклонение, мкВ }

implementation

uses lr7gfx;

// Формирование нормального закона распределения

function FormNormZakon(): real;

begin

FormNormZakon := sqrt(-2 \* ln(Random)) \* sin(2 \* PI \* Random);

end;

// Расчет альфа

function CalcAlpha(t: real): real;

var

B: real;

n: integer;

begin

n := Trunc(t / Period); // Определяем номер периода

// В зависимости от того в какой половине периода B(t) равно 1 или 0

if (t >= (n \* Period)) and (t < (n \* Period + Period / 2)) then

B := 1

else

B := 0;

CalcAlpha := Alpha \* (1 + 10 \* B);

end;

begin

Randomize;

predY := 0;

Y := 0;

for i:=1 to N do

begin

p := exp(-Alpha \* dt);

Y := p \* predY + S \* sqrt(1 - sqr(p)) \* FormNormZakon();

T[i] := M + Y;

predY := Y; // Запоминаем текущее значение как предыдущее для следующей итерации

end;

end.

**Скриншот работы программы.**

