Санкт-Петербургский государственный политехнический университет

Институт компьютерных наук и технологий

**Кафедра «Компьютерные системы и программные технологии»**

**КУРСОВАЯ РАБОТА**

**Исследование обнаруживающих способностей**

**алгоритмов диагностирования**

по дисциплине «Автоматизация проектирования дискретных устройств»

Выполнил

студент гр. 3540901/02001 <*подпись*> Д. Р. Бараев

Руководитель

доцент, к.т.н. <*подпись*> Е. Н. Бендерская

«07» апреля 2021 г.

Санкт-Петербург

2021

Санкт-Петербургский государственный политехнический университет

**ЗАДАНИЕ**

**НА ВЫПолнение курсового проекта  
(курсовой работы)**

студенту группы 3540901/02001 Бараеву Дамиру Рашидовичу

*(номер группы) (фамилия, имя, отчество)*

***1. Тема проекта (работы):*** Исследование обнаруживающих способностей алгоритмов диагностирования

***2. Срок сдачи законченного проекта (работы)*** 07.04.2021

***3. Исходные данные к проекту (работе)***: Используются построенные модели диагностирования (лабораторная работа №3), для обнаружения изменения параметров объекта диагностирования, используются заданные алгоритмы. Для каждого алгоритма имеются наборы параметров, обеспечивающие заданные уровни вероятности ложного обнаружения (курсовая работа).

***4. Содержание пояснительной записки*** (перечень подлежащих разработке вопросов): введение, основная часть (раскрывается структура основной части), заключение, список использованных источников, приложения.

***Дата получения задания***: «09» декабря 2020 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись) (инициалы, фамилия)*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студента) (инициалы, фамилия)*

\_\_\_\_\_\_\_\_\_\_\_\_

*(дата)*

Содержание

[Введение 4](#_Toc68650548)

[Ход работы 5](#_Toc68650549)

[Заключение 10](#_Toc68650550)

[Список использованных источников 11](#_Toc68650551)

[Приложение. Реализация программы (исходный код) 12](#_Toc68650552)

# Введение

Индивидуальный вариант курсового проекта: *«Алгоритм принятия решений на основе невязок наблюдения,*

*"разладка" - изменение СКО, случай средних дефектов (рассмотреть от 1.5 до 2.5). Дефект вносить в середине периода накопления. Проверить устойчивость алгоритма на случай, когда гипотеза о независимом процессе в режиме НФ не верна, и реальный процесс - авторегрессия второго порядка. Выяснить как влияют коэффициенты корреляции такой авторегрессии на качество*

*принятия решений.»*

Основные задачи проекта соответствуют этапам моделирования с заданным типом дефекта:

1. Составить план экспериментов, подобрать выборку и кол-во экспериментов. Определить значения времени обнаружения и дать им оценку.
2. Определить среднее время обнаружения для дефекта (по варианту) и рассмотреть параметры алгоритма. Найти набор параметров, обеспечивающий наименьшее среднее время обнаружения.
3. Определить среднее время обнаружения для выбранного набора, построить графики зависимостей и проиллюстрировать доверительные интервалы времени обнаружения.

# Ход работы

В ходе работы была разработана программа, способная моделировать авторегрессию 2 порядка с возможностью внедрения дефектов, задания значений выборки, итераций моделирования и пороговых значений.

Для проведения исследований были использованы приведенные ниже пороговые значения:

1. 0.02.
2. 0.04.
3. 0.05.
4. 0.07.
5. 0.09.

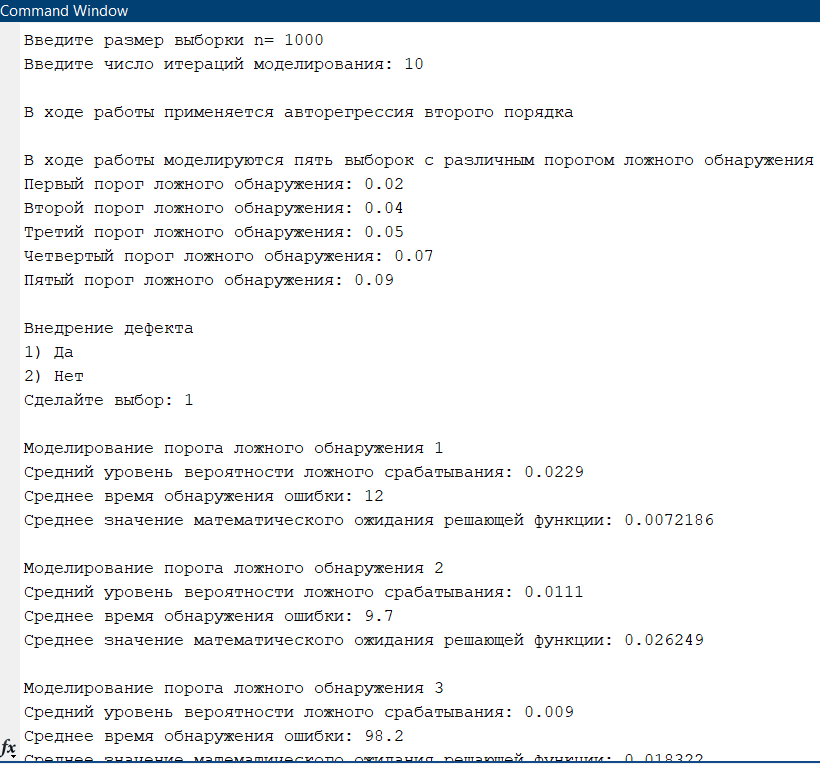


Рисунок 1 - Пример моделирования в MatLab

Исходя из полученных в ходе исследований данных, были получены средние значения параметров обнаружения, представленные в таблице 1.

Таблица 1 - Средние значения параметров обнаружения

| **Количество итераций** | **Порог** | **Дефект** | **Среднее время обнаружения ошибки** | **Среднее значение математического ожидания решающей функции** | **Средний уровень вероятности ложного срабатывания** |
| --- | --- | --- | --- | --- | --- |
| 10 | 0.02 | + | 22.4 | 0.008304 | 0.0229 |
| - | - | 0.010932 | 0.0228 |
| 0.04 | + | 46.7 | 0.02671 | 0.011 |
| - | - | 0.02564 | 0.0111 |
| 0.05 | + | 52.9 | 0.023522 | 0.009 |
| - | - | 0.018558 | 0.009 |
| 0.07 | + | 145.4 | 0.052165 | 0.006 |
| - | - | 0.050992 | 0.006 |
| 0.09 | + | 40.7 | 0.020755 | 0.005 |
| - | - | 0.016533 | 0.005 |
| 50 | 0.02 | + | 15.62 | 0.0096941 | 0.02278 |
| - | - | 0.0095059 | 0.0228 |
| 0.04 | + | 10.38 | 0.026612 | 0.01102 |
| - | - | 0.027361 | 0.01102 |
| 0.05 | + | 70.46 | 0.019709 | 0.009 |
| - | - | 0.018549 | 0.009 |
| 0.07 | + | 16.38 | 0.049866 | 0.006 |
| - | - | 0.050678 | 0.006 |
| 0.09 | + | 58.74 | 0.01967 | 0.005 |
| - | - | 0.020672 | 0.005 |
| 100 | 0.02 | + | 25.19 | 0.0092753 | 0.02284 |
| - | - | 0.0087689 | 0.02281 |
| 0.04 | + | 55.24 | 0.02631 | 0.01103 |
| - | - | 0.025817 | 0.01106 |
| 0.05 | + | 35.54 | 0.018515 | 0.009 |
| - | - | 0.017933 | 0.009 |
| 0.07 | + | 104.86 | 0.051026 | 0.006 |
| - | - | 0.050494 | 0.006 |
| 0.09 | + | 35.5 | 0.019882 | 0.005 |
| - | - | 0.021137 | 0.00499 |

На рисунках 2–7 представлены графики распределения значений уровня вероятности ложного обнаружения, времени обнаружения и математического ожидания решающей функции при различных параметрах.

1. **Результаты проведения 10 итераций моделирования авторегрессии 2-го порядка при наличии дефекта и с его отсутствием.**

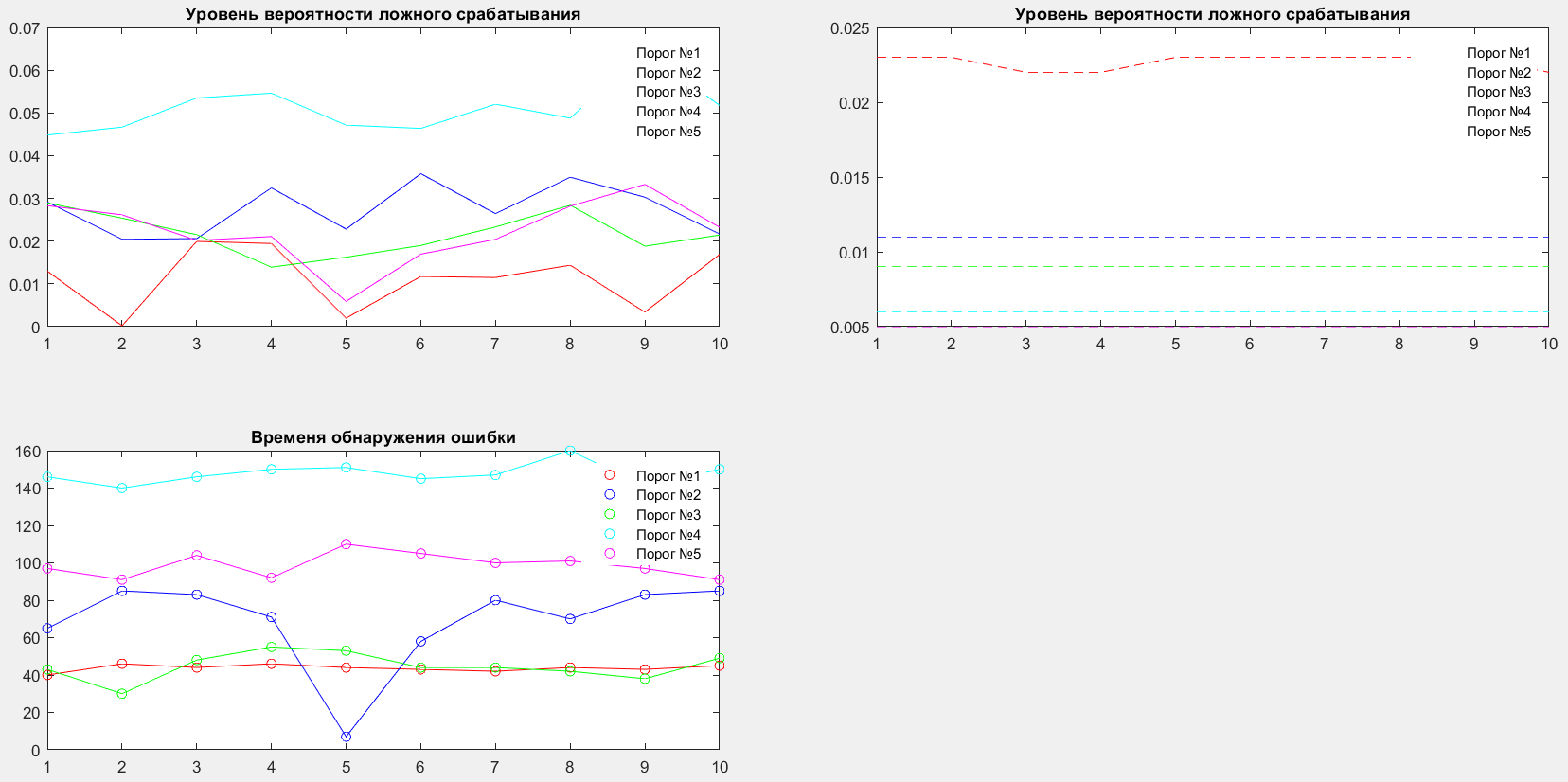


Рисунок 2 – Графики по результатам проведения 10 итераций моделирования авторегрессии 2-го порядка при наличии дефекта

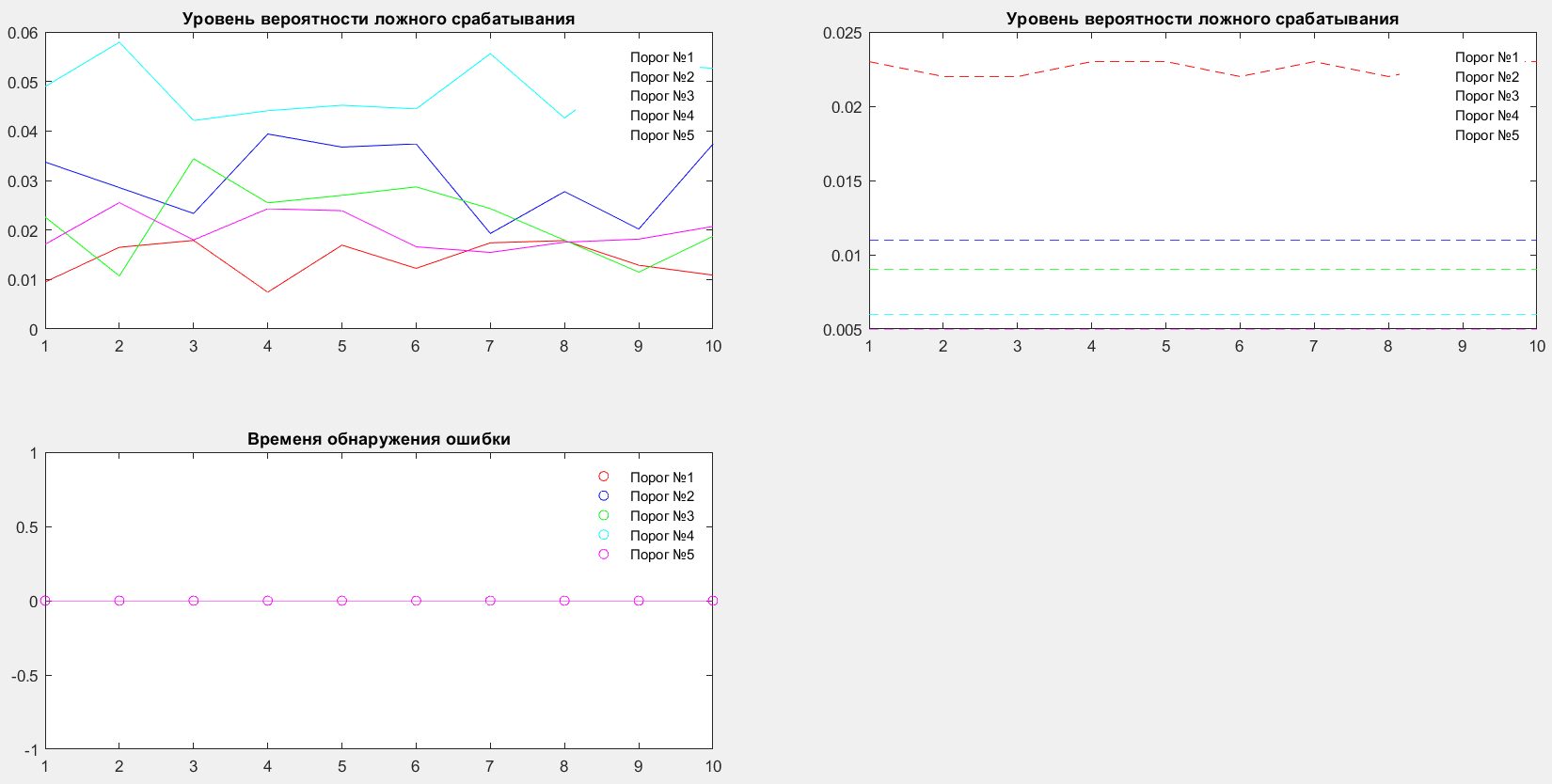


Рисунок 3 - Графики по результатам проведения 10 итераций моделирования авторегрессии 2-го порядка без дефекта

1. **Результаты проведения 50 итераций моделирования авторегрессии 2-го порядка при наличии дефекта и с его отсутствием.**

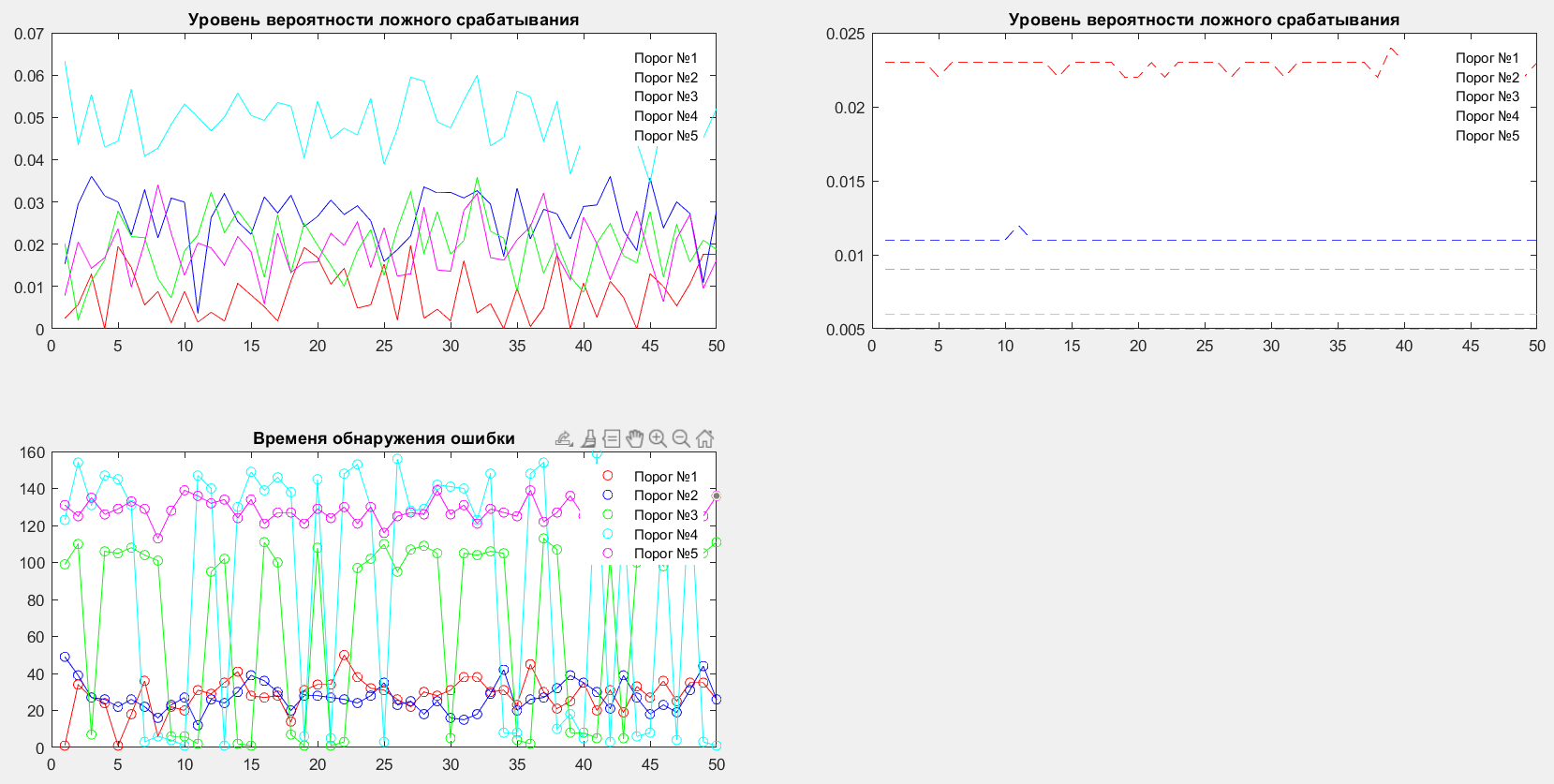


Рисунок 4 - Графики по результатам проведения 50 итераций моделирования авторегрессии 2-го порядка при наличии дефекта

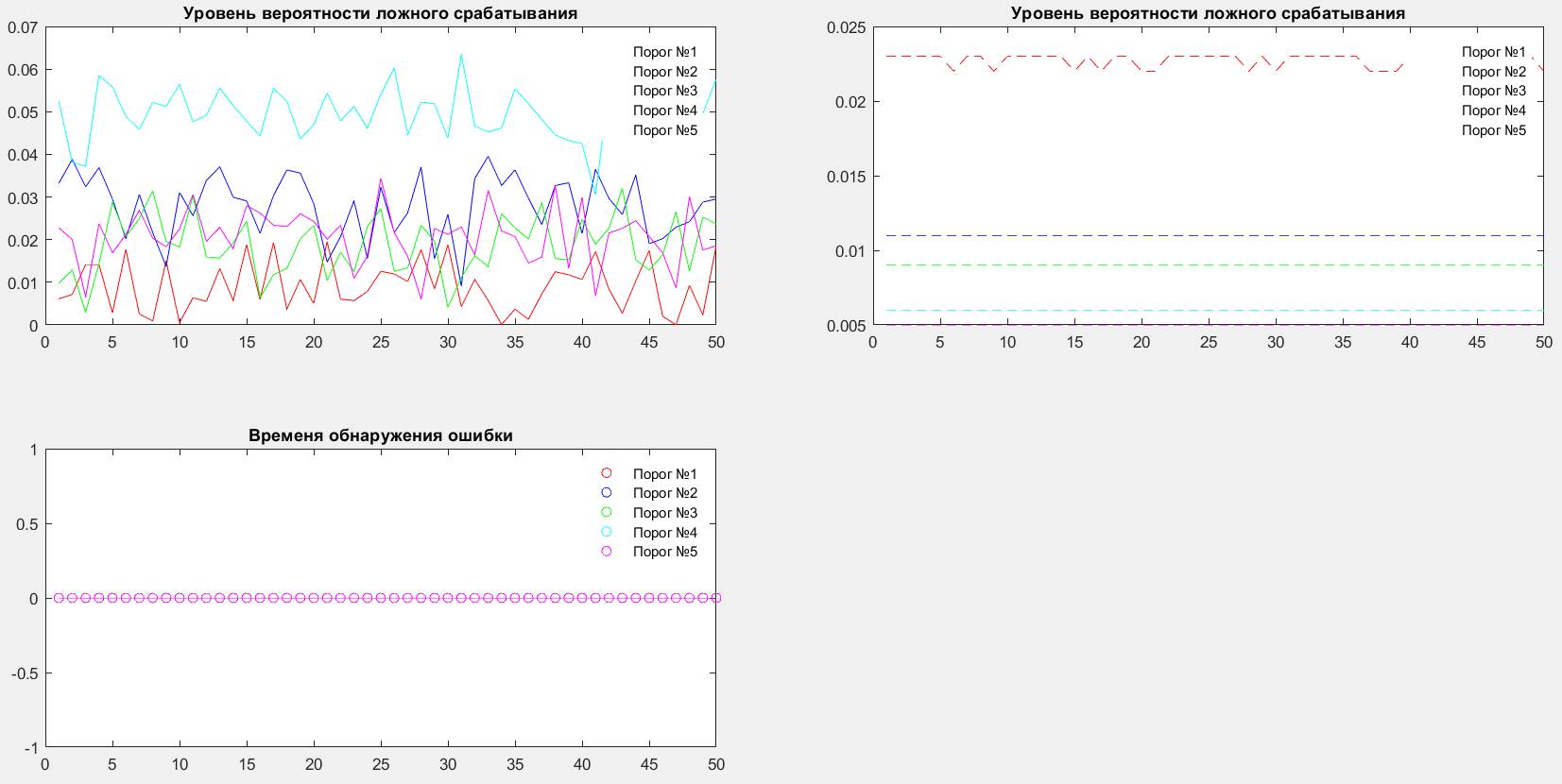


Рисунок 5 - Графики по результатам проведения 50 итераций моделирования авторегрессии 2-го порядка без дефекта

1. **Результаты проведения 100 итераций моделирования авторегрессии 2-го порядка при наличии дефекта и с его отсутствием.**

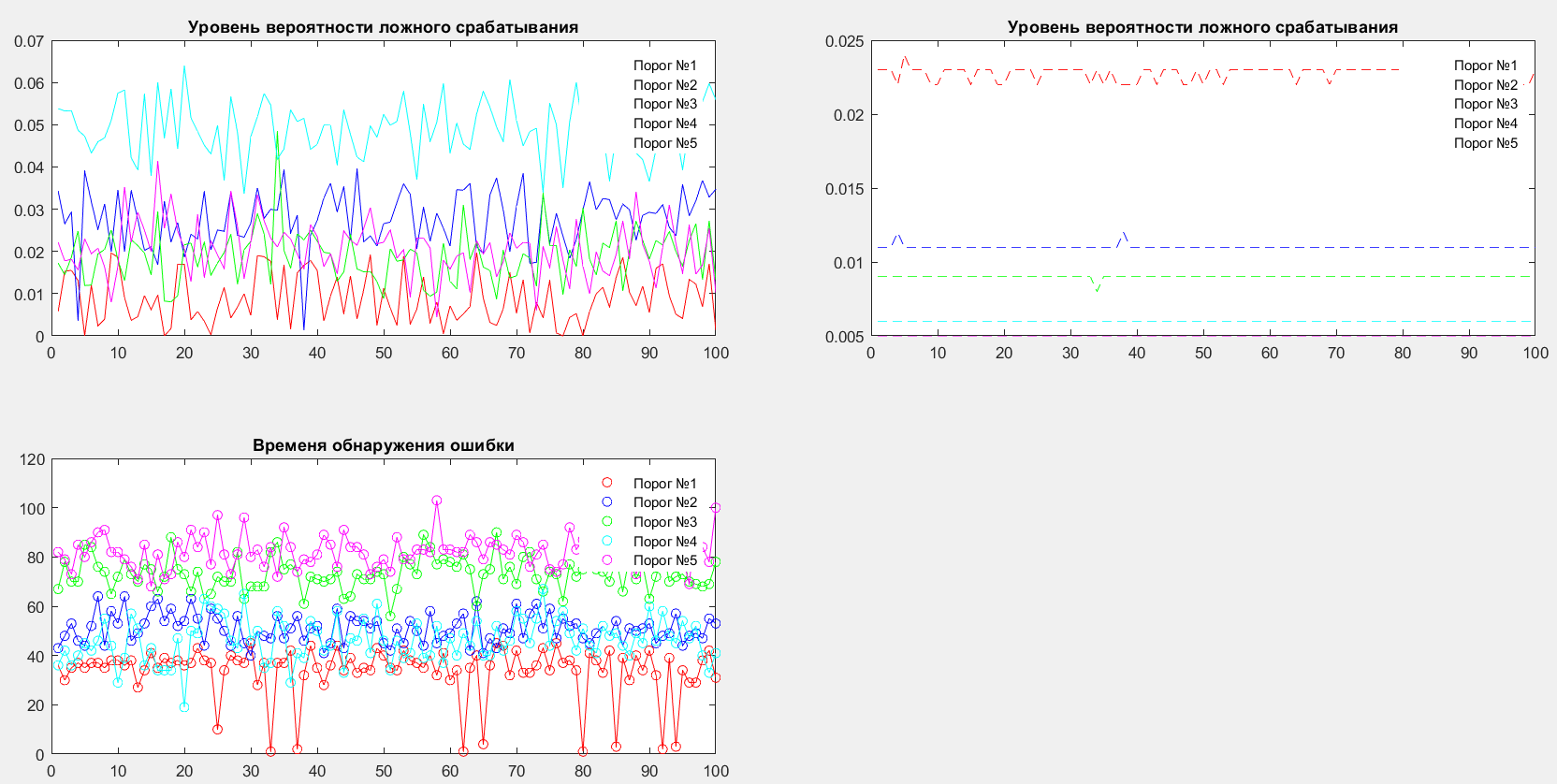


Рисунок 6 - Графики по результатам проведения 100 итераций моделирования авторегрессии 2-го порядка при наличии дефекта

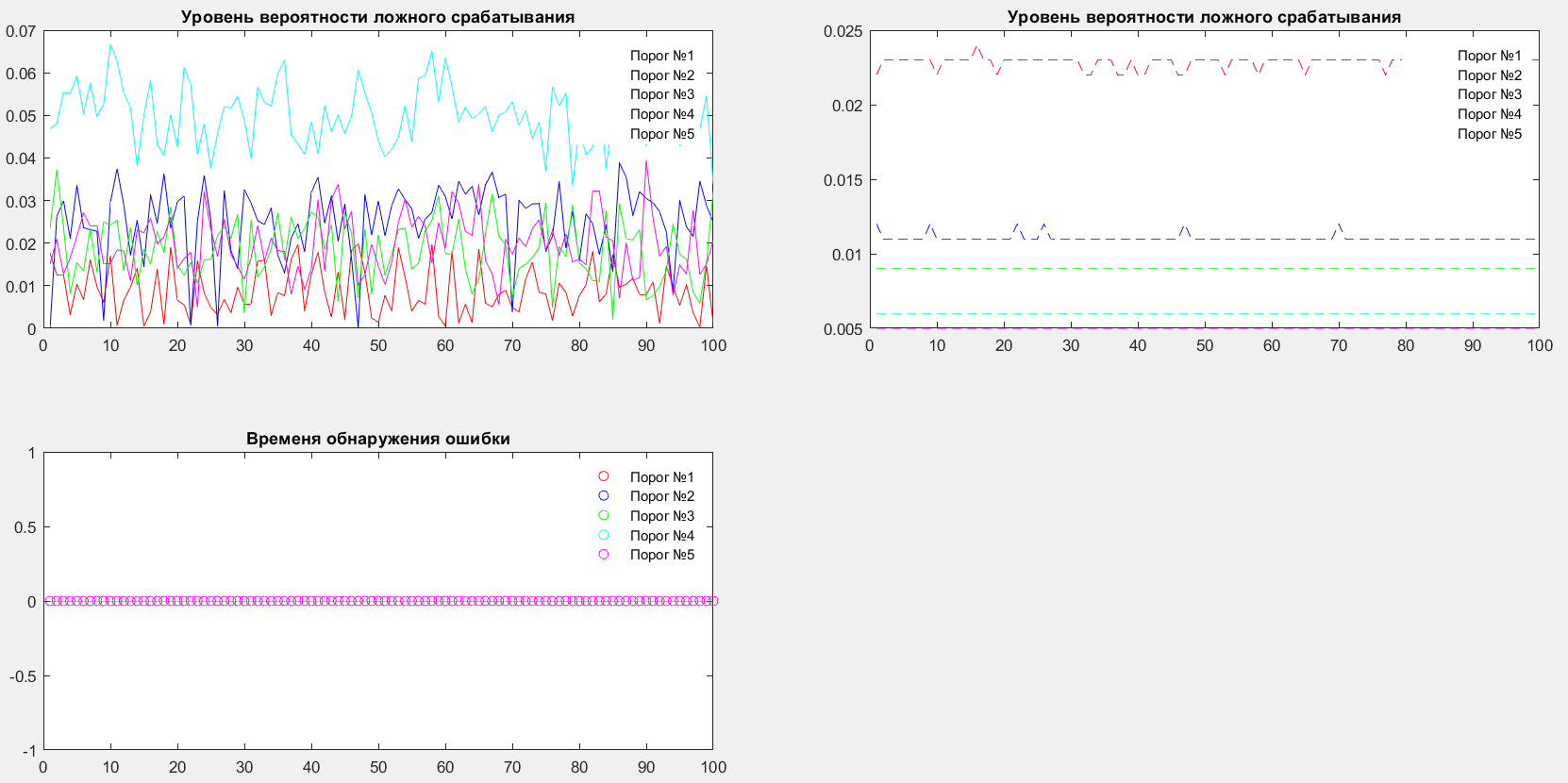


Рисунок 7 - Графики по результатам проведения 100 итераций моделирования авторегрессии 2-го порядка без дефекта

# Заключение

В ходе выполнения данного проекта был составлен алгоритм на основе невязок наблюдения, который использовался для оценивания результатов ложного обнаружения дефекта, а также среднего времени обнаружения дефекта введенного в виде СКО в середине периода накопления.

По полученным результатам можем сделать вывод, что при увеличении периода накопления алгоритма, увеличивается и время запаздывания обнаружения. Также при исследовании модели без дефекта выяснено, что увеличение параметра h снижает вероятность ложного обнаружения. \Система обнаруживает возникающие ошибки, однако, по моему мнению, чувствительна к изменению параметров алгоритма.

# Список использованных источников

1. Е. Н. Бендерская “Функциональная диагностика систем управления”, 2000 г.

# Приложение. Реализация программы (исходный код)

1. *Kurs.m*

clc,clear

n = input('Введите размер выборки n= ');

iter = input('Введите число итераций моделирования: ');

disp(' ')

disp('В ходе работы применяется авторегрессия второго порядка')

disp(' ')

h = zeros(1,5);

gsum = zeros(5,iter);

Pdelta = zeros(5,iter);

N = zeros(5,iter);

x = zeros(1,iter);

disp('В ходе работы моделируются пять выборок с различным порогом ложного обнаружения')

h(1,1) = input('Первый порог ложного обнаружения: ');

h(1,2) = input('Второй порог ложного обнаружения: ');

h(1,3) = input('Третий порог ложного обнаружения: ');

h(1,4) = input('Четвертый порог ложного обнаружения: ');

h(1,5) = input('Пятый порог ложного обнаружения: ');

%h(1,1) = 0.02;

%h(1,2) = 0.04;

%h(1,3) = 0.05;

%h(1,4) = 0.07;

%h(1,5) = 0.09;

disp(' ')

disp('Внедрение дефекта')

disp('1) Да')

disp('2) Нет')

choice = input('Сделайте выбор: ');

disp(' ')

for i=1:1:iter

x(1,i) = i;

end

for i=1:1:5

disp(['Моделирование порога ложного обнаружения ', num2str(i)])

[gsum(i,:),Pdelta(i,:),N(i,:)] = kursModel(n, choice, h(1,i), iter);

end

tiledlayout('flow')

%Значения математического ожидания решающей функции

nexttile

plot(x,gsum(1,:),'r-');

hold on

plot(x,gsum(2,:),'b-');

hold on

plot(x,gsum(3,:),'g-');

hold on

plot(x,gsum(4,:),'c-');

hold on

plot(x,gsum(5,:),'m-');

hold on

legend('Порог №1','Порог №2','Порог №3','Порог №4','Порог №5')

title('Уровень вероятности ложного срабатывания')

hold off

%Уровень вероятности ложного срабатывания

nexttile

plot(x,Pdelta(1,:),'r--');

hold on

plot(x,Pdelta(2,:),'b--');

hold on

plot(x,Pdelta(3,:),'g--');

hold on

plot(x,Pdelta(4,:),'c--');

hold on

plot(x,Pdelta(5,:),'m--');

hold on

legend('Порог №1','Порог №2','Порог №3','Порог №4','Порог №5')

title('Уровень вероятности ложного срабатывания')

hold off

%Время обнаружения ошибки

nexttile

plot(x,N(1,:),'r-o');

hold on

plot(x,N(2,:),'b-o');

hold on

plot(x,N(3,:),'g-o');

hold on

plot(x,N(4,:),'c-o');

hold on

plot(x,N(5,:),'m-o');

hold on

legend('Порог №1','Порог №2','Порог №3','Порог №4','Порог №5')

title('Временя обнаружения ошибки')

hold off

1. *kursModel.m*

function [answer, second, third] = kursModel(n, choice, h, iter)

u=0; %Значение мат.ожидания

sigm = 1.5; %Изначальное значение sigma

a1 = 0.6; %Значение первого условия стационарности

a2 = 0.2; %Значение второго условия стационарности

g=0; %Итерационное значение решающей функции

x1 = 0; %Дополнительное значение x

gsum = zeros(1,iter); %Суммарное значение решающей функции

Pdelta = zeros(1,iter); %Значение вероятности ложного срабатывания

N = zeros(1,iter); %Период обнаружения ошибки

if(choice == 1)

error = randi([1 n/2], 1, 1); %Генерация момента ошибки

Ntrigger = 0;

end

xm = zeros(1,n);

B = sigm\*sqrt((sigm^2)\*(((1+a2)\*(((1-a2)^2)-a1^2))/(1-a2))); %Значение beta

%Моделирование

for j=1:1:iter

for i=1:1:n

if (choice == 1 && i == error)

sigm = 2.5;

B = sigm\*sqrt((sigm^2)\*(((1+a2)\*(((1-a2)^2)-a1^2))/(1-a2)));

Ntrigger = 1;

end

z = rand;

bz = B\*z;

x = a1\*x1 + bz;

g = abs((((x-u-a1\*(x1-u))^2)/(B^2))-1);

gsum(1,j) = gsum(1,j) +g;

if (choice == 1 && Ntrigger == 1)

N(1,j) = N(1,j) +1;

if(gsum(1,j)/(sqrt(2)\*n)>=h)

Ntrigger = 0;

end

end

if(gsum(1,j)/(sqrt(2)\*n)>=h)

Pdelta(1,j) = Pdelta(1,j) + 1;

gsum(1,j)=0;

end

xm(1,i) = x;

x1=x;

end

gsum(1,j) = gsum(1,j)/(sqrt(2)\*n);

Pdelta(1,j) = Pdelta(1,j)/n;

end

%Анализ результатов

disp(['Средний уровень вероятности ложного срабатывания: ', num2str(mean(Pdelta))])

if(choice == 1)

disp(['Среднее время обнаружения ошибки: ',num2str(mean(N))])

end

disp(['Среднее значение математического ожидания решающей функции: ',num2str(mean(gsum))])

answer = gsum;

second = Pdelta;

third = N;

disp(' ')

end