Оглавление

[1. Исходные данные 3](#_Toc27703482)

[2. Программа работы. 4](#_Toc27703483)

[3. Моделирование алгоритма, основанного на принципе невязок 4](#_Toc27703484)

[3.1. Аналитический расчет 4](#_Toc27703485)

[3.2. Имитационное моделирование 7](#_Toc27703486)

[4. Выводы 8](#_Toc27703487)

[5. Листинги 9](#_Toc27703488)

# Исходные данные

Вариант №4.

|  |  |  |  |
| --- | --- | --- | --- |
| Система | Алгоритм 1 | Алгоритм 2 | Уровни вероятности ложного обнаружения |
| А | невязок | Экспоненциального сглаживания | 0.01 |

Исходные параметры:

A) Используется система диагностирования, построенная с помощью структурных инвариантов, представляющих собой разность одноименных параметров различных информационных каналов (лабораторная работа №3);

Для обнаружения изменения параметров объекта диагностирования используются заданные по вариантам алгоритмы:

1. Базовый – алгоритм, основанный на принципе невязок;
2. Дополнительный – алгоритм экспоненциального сглаживания.

Уровень вероятности ложного обнаружения: 0.01 и 0.02.

**Этапы моделирования каждого алгоритма:**

1. Аналитический расчет.

1.1. Для различных наборов параметров алгоритма рассчитать значения вероятности ложного обнаружения и среднего времени обнаружения, используя аналитические зависимости между выбранными показателями качества (вероятность ложного обнаружения, среднее время обнаружения) и параметрами алгоритмов.

1.2. Построить графики зависимостей вероятности ложного обнаружения от каждого параметра алгоритма.

1.3. Построить графики зависимостей среднего времени обнаружения от каждого параметра алгоритма.

1.4. Выбрать наборы параметров, соответствующие заданным уровням вероятности ложного обнаружения.

2. Имитационное моделирование.

2.1. Составить план экспериментов, выбрать объем выборки и количество экспериментов, исходя из требуемого уровня доверительной вероятности для показателей качества обнаружения.

2.2. Для различных наборов параметров алгоритма определить методом моделирования значения вероятности ложного обнаружения. В качестве начальных наборов параметров использовать выбранные в п. 1.4.

2.3. Построить графики зависимостей вероятности ложного обнаружения от каждого параметра алгоритма. Сравнить их с графиками, построенными с использованием аналитических зависимостей.

2.4. Выбрать наборы параметров, соответствующие заданным уровням вероятности ложного обнаружения.

# Программа работы.

* 1. Настроить алгоритм 1 на уровень ложного обнаружения  
     0,01. Результатом настройки является таблица, содержащая  
     несколько наборов параметров, обеспечивающих заданный  
     уровень ложного обнаружения.
  2. Настроить алгоритм 1 на уровень ложного обнаружения  
     0,02. Результатом настройки является таблица, содержащая  
     несколько наборов параметров, обеспечивающих заданный  
     уровень ложного обнаружения.
  3. Настроить алгоритм 2 на уровень ложного обнаружения  
     0,01. Результатом настройки является таблица, содержащая  
     несколько наборов параметров, обеспечивающих заданный  
     уровень ложного обнаружения.
  4. Настроить алгоритм 2 на уровень ложного обнаружения  
     0,02. Результатом настройки является таблица, содержащая  
     несколько наборов параметров, обеспечивающих заданный  
     уровень ложного обнаружения.
  5. Сравнить рассматриваемые алгоритмы с аналогами, найденными самостоятельно в других источниках. Представить не менее двух альтернативных, не рассмотренных в лекциях, методов. Выполнить качественный анализ – отразить достоинства, недостатки, возможные ограничения на свойства системы и требования по априорной информации для формирования значений параметров системы.
  6. Сформулировать выводы о проделанной работе, оформить отчет.

# Моделирование алгоритма, основанного на принципе невязок

## Аналитический расчет

Решающая функция алгоритма, основанного на принципе невязок,  
формируется как невязка (расхождение) между наблюдаемой и принятой моделями случайного процесса z:



Для получения возможности настройки алгоритма вводится модификация алгоритма:

,

где M>0 – глубина памяти,

∈ N (0,1):

1) если

z(n) ∈ N (μ1, β1),

то = ;

2) если z – процесс авторегрессии

z(n) = μ1 + ,

y(n) ∈ N (0,1),

то =

Аналитическая настройка параметров алгоритма.

Вероятность ложного обнаружения для этого алгоритма может быть определена на основе использования марковских процессов. Такой подход позволяет получить следующее выражение для вероятности ложного обнаружения:

,.

где

*-* функция распределения с i степенями свободы,

M – глубина памяти,

h – порог срабатывания алгоритма.

Модификация алгоритма (рекуррентный вид)

Исходная формула преобразуется к виду:

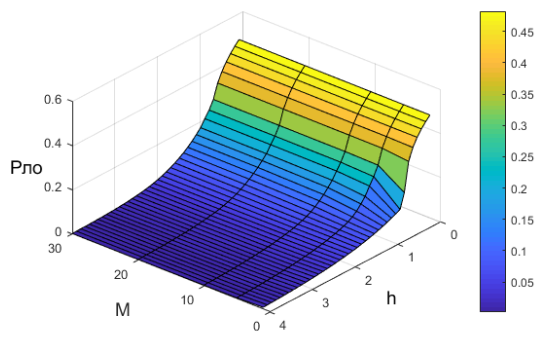
*.*

Начальные условия : *.*

Для построения гиперповерхности и вычисления конкретных  
значений был написан скрипт, представленный в приложении 1*.*

*Таблица 1. Зависимость вероятности ложного обнаружения от параметров алгоритма*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| h | M | | | | |
| 1 | 5 | 10 | 20 | 30 |
| 0,1 | 0,4822 | 0,4821 | 0,4821 | 0,4821 | 0,4821 |
| 0,2 | 0,4625 | 0,4621 | 0,4621 | 0,4621 | 0,4621 |
| 0,3 | 0,4397 | 0,439 | 0,439 | 0,439 | 0,439 |
| 0,4 | 0,4121 | 0,4117 | 0,4117 | 0,4117 | 0,4117 |
| 0,5 | 0,3762 | 0,3783 | 0,3783 | 0,3783 | 0,3783 |
| 0,6 | 0,3229 | 0,3357 | 0,336 | 0,3361 | 0,3361 |
| 0,7 | 0,1924 | 0,2694 | 0,2722 | 0,2723 | 0,2723 |
| 0,8 | 0,1261 | 0,2279 | 0,2337 | 0,2342 | 0,2342 |
| 0,9 | 0,1163 | 0,1953 | 0,2044 | 0,2055 | 0,2055 |
| 1 | 0,1073 | 0,1587 | 0,1725 | 0,1749 | 0,175 |
| 1,1 | 0,0990 | 0,1347 | 0,1499 | 0,1538 | 0,154 |
| 1,2 | 0,0913 | 0,1124 | 0,1279 | 0,1334 | 0,1339 |
| 1,3 | 0,0843 | 0,0965 | 0,1097 | 0,1161 | 0,117 |
| 1,4 | 0,0777 | 0,0844 | 0,0940 | 0,1005 | 0,1018 |
| 1,5 | 0,0717 | 0,0755 | 0,0812 | 0,0869 | 0,0885 |
| 1,6 | 0,0662 | 0,0684 | 0,0708 | 0,0750 | 0,0765 |
| 1,7 | 0,0611 | 0,0620 | 0,0622 | 0,0647 | 0,0659 |
| 1,8 | 0,0563 | 0,0562 | 0,0551 | 0,0559 | 0,0567 |
| 1,9 | 0,0520 | 0,0508 | 0,0490 | 0,0484 | 0,0486 |
| 2 | 0,0480 | 0,0460 | 0,0437 | 0,0420 | 0,0417 |
| 2,1 | 0,0443 | 0,0415 | 0,0390 | 0,0366 | 0,0358 |
| 2,2 | 0,0409 | 0,0375 | 0,0348 | 0,0319 | 0,0307 |
| 2,3 | 0,0377 | 0,0339 | 0,0310 | 0,0279 | 0,0264 |
| 2,4 | 0,0348 | 0,0306 | 0,0275 | 0,0244 | 0,0228 |
| 2,5 | 0,0321 | 0,0276 | 0,0245 | 0,0213 | 0,0196 |
| 2,6 | 0,0297 | 0,0249 | 0,0217 | 0,0186 | 0,0169 |
| 2,7 | 0,0274 | 0,0224 | 0,0193 | 0,0162 | 0,0146 |
| 2,8 | 0,0253 | 0,0202 | 0,0171 | 0,0141 | 0,0125 |
| 2,9 | 0,0234 | 0,0182 | 0,0152 | 0,0123 | 0,0108 |
| 3 | 0,0216 | 0,0164 | 0,0135 | 0,0107 | 0,0093 |
| 3,1 | 0,0199 | 0,0148 | 0,0120 | 0,00930 | 0,0080 |
| 3,2 | 0,0184 | 0,0133 | 0,0106 | 0,00810 | 0,0069 |
| 3,3 | 0,0170 | 0,0120 | 0,00940 | 0,00710 | 0,0059 |
| 3,4 | 0,0157 | 0,0108 | 0,00840 | 0,00620 | 0,0051 |
| 3,5 | 0,0145 | 0,00980 | 0,00740 | 0,00540 | 0,0044 |
| 3,6 | 0,0134 | 0,00880 | 0,00660 | 0,00470 | 0,0038 |
| 3,7 | 0,0124 | 0,00790 | 0,00590 | 0,00410 | 0,0033 |
| 3,8 | 0,0114 | 0,00720 | 0,00520 | 0,00360 | 0,0028 |
| 3,9 | 0,0106 | 0,00650 | 0,00460 | 0,00310 | 0,0024 |
| 4 | 0,00980 | 0,00580 | 0,00410 | 0,00270 | 0,0021 |



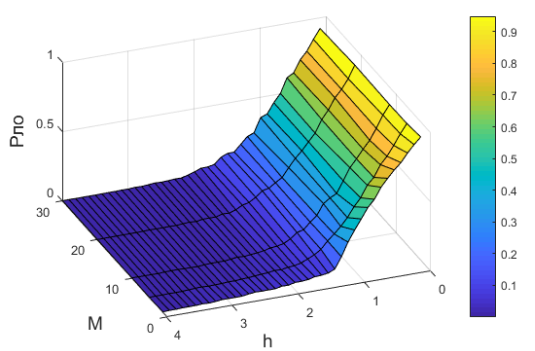
*Рис. 1. Гиперповерхность зависимости ложного обнаружения от порога и глубины памяти.*

## Имитационное моделирование

Код программы находится в приложении 2.

*Таблица 2. Зависимость вероятности ложного обнаружения от параметров алгоритма*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| h | M | | | | |
| 1 | 5 | 10 | 20 | 30 |
| 0,1 | 0,9483 | 0,9334 | 0,9301 | 0,9357 | 0,9203 |
| 0,2 | 0,9016 | 0,8675 | 0,8629 | 0,8457 | 0,8502 |
| 0,3 | 0,849 | 0,7917 | 0,7758 | 0,7775 | 0,7729 |
| 0,4 | 0,7881 | 0,7219 | 0,698 | 0,7246 | 0,7196 |
| 0,5 | 0,7438 | 0,6697 | 0,6422 | 0,6456 | 0,6328 |
| 0,6 | 0,6918 | 0,5862 | 0,5802 | 0,584 | 0,6067 |
| 0,7 | 0,6359 | 0,5382 | 0,5245 | 0,4947 | 0,5016 |
| 0,8 | 0,5569 | 0,4557 | 0,4389 | 0,4408 | 0,4429 |
| 0,9 | 0,5062 | 0,3992 | 0,356 | 0,3843 | 0,3704 |
| 1 | 0,4318 | 0,3298 | 0,3317 | 0,3379 | 0,3523 |
| 1,1 | 0,3601 | 0,2748 | 0,2987 | 0,2792 | 0,2509 |
| 1,2 | 0,2813 | 0,2286 | 0,2222 | 0,2161 | 0,2325 |
| 1,3 | 0,1842 | 0,1862 | 0,1825 | 0,1873 | 0,1845 |
| 1,4 | 0,1037 | 0,1338 | 0,1357 | 0,1555 | 0,1368 |
| 1,5 | 0,0828 | 0,1034 | 0,1242 | 0,1456 | 0,1358 |
| 1,6 | 0,0755 | 0,0811 | 0,0861 | 0,1068 | 0,1154 |
| 1,7 | 0,0733 | 0,0612 | 0,0613 | 0,0778 | 0,0729 |
| 1,8 | 0,0615 | 0,0487 | 0,0507 | 0,0663 | 0,0629 |
| 1,9 | 0,057 | 0,0461 | 0,0461 | 0,0419 | 0,0622 |
| 2 | 0,0567 | 0,0366 | 0,0342 | 0,0256 | 0,0437 |
| 2,1 | 0,0448 | 0,0396 | 0,0268 | 0,0284 | 0,0292 |
| 2,2 | 0,0464 | 0,0315 | 0,0225 | 0,0141 | 0,0159 |
| 2,3 | 0,0362 | 0,0291 | 0,0174 | 0,0152 | 0,017 |
| 2,4 | 0,0359 | 0,0247 | 0,0216 | 0,0148 | 0,0131 |
| 2,5 | 0,0352 | 0,0206 | 0,0142 | 0,0063 | 0,0108 |
| 2,6 | 0,0371 | 0,0185 | 0,0087 | 0,0104 | 0,0131 |
| 2,7 | 0,0313 | 0,0147 | 0,0098 | 0,0067 | 0,0087 |
| 2,8 | 0,0237 | 0,0152 | 0,0094 | 0,0028 | 0,0048 |
| 2,9 | 0,0237 | 0,0115 | 0,0073 | 0,0042 | 0,0064 |
| 3 | 0,0236 | 0,0106 | 0,0027 | 0,0029 | 0,0028 |
| 3,1 | 0,0245 | 0,0086 | 0,0037 | 0,0029 | 0,0034 |
| 3,2 | 0,0187 | 0,0117 | 0,0037 | 0,0045 | 0,0018 |
| 3,3 | 0,0155 | 0,0107 | 0,0062 | 0,0023 | 0,002 |
| 3,4 | 0,0179 | 0,007 | 0,0054 | 0,0016 | 0,001 |
| 3,5 | 0,0145 | 0,0048 | 0,005 | 0,0013 | 0,0006 |
| 3,6 | 0,0145 | 0,0039 | 0,0011 | 0,0013 | 0,0009 |
| 3,7 | 0,0143 | 0,005 | 0,0022 | 0,001 | 0,0003 |
| 3,8 | 0,012 | 0,003 | 0,0028 | 0,0009 | 0,0004 |
| 3,9 | 0,012 | 0,0037 | 0,0022 | 0,0002 | 0,0012 |
| 4 | 0,0098 | 0,0028 | 0,0037 | 0,0002 | 0,0006 |



*Рис. 2. Гиперповерхность зависимости ложного обнаружения от порога и глубины памяти.*

*Таблица 3. Полученные параметры алгоритма*

|  |  |  |
| --- | --- | --- |
| Pло | h | M |
| 0,0087 | 2,6 | 10 |
| 0,0087 | 2,7 | 20 |
| 0,0098 | 4 | 1 |
| 0,0098 | 2,7 | 10 |

# Выводы

В данном случае аналитическая настройка не даёт точных результатов и может лишь иллюстрировать форму зависимости показателей системы диагностики от её параметров.

Увеличение глубины памяти и порога срабатывания решающей функции уменьшает вероятность ложного срабатывания при использовании метода невязок.

# Листинги

*Приложение 1*

clear all;

M = [1 2 4 10 15 20 25 30 35 40 60 100];

h = [1 2 3 4 5 6 7 8 9 ];

%%M = 1:10:100;

%%h = 1:1:5;

for ii=1:size(M,2)

for jj=1:size(h,2)

%% Вычисления числителя

PiMUL = 1;

for Mi=1:M(ii)

PiMUL = PiMUL\*Pi(Mi,h(jj));

end

%% Вычисление знаменателя

PiSUM\_at\_MUL = 0;

for j=1:(M(ii)-1)

PiMUL2=1;

for i=1:j

PiMUL2=PiMUL2\*Pi(i,h(jj));

end

PiSUM\_at\_MUL=PiSUM\_at\_MUL+PiMUL2;

end

Plo(ii,jj) = round( ((1 - PiMUL)/(2+PiSUM\_at\_MUL-PiMUL)) \*10000)/10000;

end

end

surf(h,M,Plo)

%% Вычисление pi

function rez = Pi(Mi,h)

if(h<=sqrt(Mi/h))

rez=chi2cdf(h\*sqrt(2\*Mi)+Mi,Mi) - chi2cdf(-h\*sqrt(2\*Mi)+Mi,Mi);

else

rez=chi2cdf(h\*sqrt(2\*Mi)+Mi,Mi);

end

end

Приложение 2

clear all;

%% Исходные данные

% Количество снятых точек

N = 10000;

% Момент возникновения неисправности

N\_def = N/2;

a1 = 0.1;

a2 = 0.5;%0.35;

A = [a1 a2; 1 0];

F = [1;0];

C = [1 0];

Mi = 0;

Di = 1;

Mo = 0;

Do = 4;

Di\_mult = 1;

Do\_mult = 1;

diAuto = (1+a2)\*((1-a2)^2-a1^2)/(1-a2);

% Глубина памяти

memDepth = [1 5 10 20 30];

% Порог срабатывания

h = 0.1:.1:4;

M\_size = length(memDepth);

H\_size = length(h);

probFailDetect = zeros(M\_size,H\_size);

for k = 1:H\_size

for i = 1:M\_size

x = [0; 0];

P = Do\*eye(2);

G = zeros(1,N);

X1\_est = [0; 0];

zNew = zeros(1, N);

zNormal = zNew;

countFail = 0;

%% Цикл работы фильтра

for j = 1:N

if (j > N\_def)

x = A\*x+F\*diAuto\*normrnd(Mi,Di^0.5\*Di\_mult,1,1);

Y = C\*x+normrnd(Mo, Do^0.5\*Do\_mult,1,1);

else

x = A\*x+F\*diAuto\*normrnd(Mi,Di^0.5,1,1);

Y = C\*x+normrnd(Mo,Do^0.5,1,1);

end

Q = A\*P\*A'+F\*Di\*F';

K = Q\*C'\*(C\*Q\*C'+Do)^-1;

P = Q-K\*C\*Q;

X1\_est = A\*X1\_est+F\*Mi+K\*(Y-C\*(A\*X1\_est+F\*Mi));

zNew(j) = Y-C\*(C\*X1\_est+F\*Mi);

S = Do+C\*P\*C'-Do\*K'\*C'-C\*K\*Do;

zNormal(1,j)= zNew(j)\*(S^-0.5);

%% Алгоритм обнаружения дефекта методом невязок

if(j <= memDepth(i))

if(j ~= 1)

G(j) = 1/sqrt(2\*j)\*sum(((zNormal(1,1:j)-Mia1\*(zNormal(1,j-1)-Mi))./Di).^2-1);

else

G(j) = 1/sqrt(2\*j)\*sum(((zNormal(1,1:j)-Mi)./Di).^2-1);

end

else

G(j) = 1/sqrt(2\*memDepth(i))\*sum(((zNormal(1,jmemDepth(i):j)-Mi- a1\*(zNormal(1,j-1)-Mi))./Di).^2-1);

end

if(abs(G(j)) >= h(k))

countFail = countFail+1;

end

end

probFailDetect(k,i) = countFail/N;

end

end

surf(memDepth,h,probFailDetect)