# Министерство образования и науки Российской Федерации

## Федеральное государственное бюджетное образовательное учреждение высшего образования

## «Новосибирский государственный технический университет»



## Кафедра теоретической и прикладной информатики

### Лабораторная работа № 2 по дисциплине «Методы активной идентификации динамических систем»

**ВЫЧИСЛЕНИЕ ИНФОРМАЦИОННОЙ МАТРИЦЫ ФИШЕРА**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Факультет: | ПМИ |  |
| Группа: | ПМИМ-01 |
| Студенты: | Ершов П.К., Дорош А.Э. |
| Вариант: | 3 |
| Уровень сложности: | 1 |
| Преподаватель: | Чубич В. М. |

Новосибирск

2021

1. **Цель работы**

Научиться вычислять информационную матрицу Фишера (ИМФ).

1. **ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ**

В соответствии с выбранным уровнем сложности разработать программу вычисления ИМФ.

1. Проверить правильность работы программы на специально подготовленном тестовом примере.
2. Вычислить ИМФ согласно варианту задания. Результаты привести в отчете.

1. **Необходимые теоретические данные**

**Алгоритм вычисления значения критерия идентификации:**

1. Определить F, Ψ , Γ, H , Q , R , , и

, где

F, Ψ , Γ, H , Q , R – матрицы модели, которые задаются в условиях задачи, – начальный вектор состояний, – ковариационная матрица ошибок прогнозирования.

Сформировать матрицу по формуле (24) : .

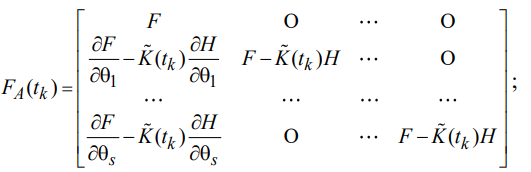
1. Положить ,

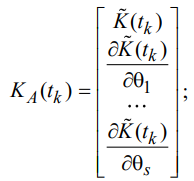
1. Определить .
2. Если k = 0, вычислить и по формулам (21) и (22) соответственно, где

(21)

(22)

тут





*;*

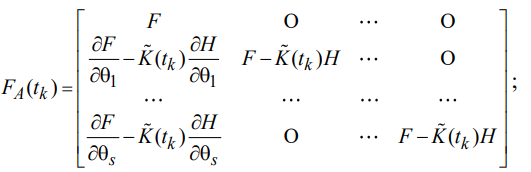
,

после перейти на шаг 8.

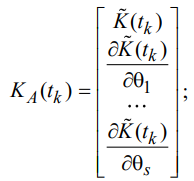
1. Найти по формуле (26) и по формуле

(26)

1. Сформировать матрицы и по формулам (23) и (25) соответственно.



(23)



(25)

1. Вычислить и по формулам (21) и (22) соответственно.
2. Найти по формуле (10) и по формуле

(10)

Вычислить по формуле (12) и по формуле

(12)

Определить по формуле (13) и по формуле

(13)

Найти по формуле (15) и по формуле

(15).

1. Используя выражение (20), получить , отвечающее текущему значению k.

(20)

где матрица

и имеют размерность n x n(s + 1), а вектор вычисляется по формуле (21).

1. Положить .
2. Увеличить k на единицу. Если , перейти на шаг 3, иначе закончить процесс.
3. **Полученные результаты**

Входные данные тестовой модели.

Исходя из указанной (для первого уровня сложности) модели:

*.*

Можно получить следующие входные данные:

|  |  |  |  |
| --- | --- | --- | --- |
| Матрицы для моделей состояния и измерения | Ковариационные матрицы шумов и начальные условия |  |  |
| ,  ,  , | , R= 0.3,  , | *,* | , |

*Результаты работы алгоритма:*

**

*Требуемое значение:*

**

Входные данные дня заданного варианта:

|  |  |  |  |
| --- | --- | --- | --- |
| Матрицы для моделей состояния и измерения | Ковариационные матрицы шумов и начальные условия |  |  |
| ,  ,  , | , R= 0.1,  , | *,*  *,* | , |

*Результаты работы алгоритма:*



1. **Текст программы**

Файл initi\_data.m

function [F, Psi, Gamma, H, Q, R, x0, P0, u] = init\_data(theta)

F = [-0.8, 1;

theta(1), 0];

Psi = [1; 1];

Gamma = [1; 1];

H = [1, 0];

Q = theta(2);

R = 0.1;

x0 = [0; 0];

P0 = [0.1, 0; 0, 0.1];

u = 3;

end

Файл init\_grad.m

function [F\_grad, Psi\_grad, Gamma\_grad, H\_grad, Q\_grad, R\_grad, x0\_grad, P0\_grad] = init\_grad()

F\_grad = cell(2,1);

F\_grad{1} = [0, 0; 1, 0];

F\_grad{2} = [0, 0; 0, 0];

Psi\_grad = cell(2,1);

Psi\_grad{1} = [0;0];

Psi\_grad{2} = [0;0];

Gamma\_grad = cell(2,1);

Gamma\_grad{1} = [0;0];

Gamma\_grad{2} = [0;0];

H\_grad = cell(2,1);

H\_grad{1} = [0,0];

H\_grad{2} = [0,0];

Q\_grad = cell(2,1);

Q\_grad{1} = 0;

Q\_grad{2} = 1;

R\_grad = cell(2,1);

R\_grad{1} = 0;

R\_grad{2} = 0;

x0\_grad = cell(2,1);

x0\_grad{1} = [0;0];

x0\_grad{2} = [0;0];

P0\_grad = cell(2,1);

P0\_grad{1} = [0, 0; 0, 0];

P0\_grad{2} = [0, 0; 0, 0];

end

Файл Get\_C.m

function [C] = Get\_C(n, alpha, i)

n2 = (alpha+1) \* n;

C = zeros(n, n2);

C(:, n\*i + 1 : n\*(i+1)) = eye(n);

end

Файл calculate\_Fisher\_Matrix.m

function [M] = calculate\_Fisher\_Matrix(theta, N)

[F, Psi, Gamma, H, Q, R, x0, P0, u] = init\_data(theta);

[F\_grad, Psi\_grad, Gamma\_grad, H\_grad, Q\_grad, R\_grad, x0\_grad, P0\_grad] = init\_grad();

n\_1 = size(x0,1);

n\_2 = size(x0,2);

alpha = length(theta);

% выделение памяти

M = zeros(alpha, alpha);

dM = zeros(alpha, alpha);

x\_a = zeros(n\_1 \* (alpha + 1), n\_2);

sigma\_a = zeros(alpha, alpha);

P\_k1\_k\_grad = cell(alpha,1);

P\_k\_k\_grad = cell(alpha, 1);

B\_grad = cell(alpha,1);

K\_grad = cell(alpha,1);

K\_grad\_ = cell(alpha,1);

P\_k1\_k1\_grad = cell(alpha,1);

F\_a = zeros(n\_1 \* (alpha + 1), n\_1 \* (alpha + 1));

K\_a = zeros(n\_1 \* (alpha + 1), 1);

% сформировать матрицу пси(а) по формуле 24

Psi\_a = zeros(n\_1 \* (alpha + 1), n\_2);

Psi\_a(1:n\_1, :) = Psi;

for i = 1 : alpha

Psi\_a(n\_1\*i+1 : n\_1\*(i+1), :) = Psi\_grad{i};

end

% задание начальных условий 2 шаг

P\_k = P0;

for i = 1 : alpha

P\_k\_k\_grad(i) = {P0\_grad{i}};

end

for k = 0 : N - 1

u = u;

if (k == 0) % то 4 шаг, затем 8

x\_a(1 : n\_1\*1, :) = F\*x0 + Psi \* u; % 21 u0

for i = 1 : alpha

x\_a(n\_1\*i + 1 : n\_1\*(i+1), :) = F\_grad{i} \* x0 + F \* x0\_grad{i} + Psi\_grad{i} \* u; % u0

end

sigma\_a = zeros(n\_1 \* (alpha + 1), n\_1 \* (alpha + 1)); % 22

end

% 5-7 шаги

if (k ~= 0)

% 5 шаг

K\_ = F \* K;

for i = 1 : alpha

K\_grad\_(i) = {F\_grad{i} \* K + F \* K\_grad{i}};

end

% 6 шаг

F\_a(1:n\_1, 1:n\_1) = F;

for i = 1 : alpha

F\_a(n\_1\*i+1 : n\_1\*(i+1), 1:n\_1) = F\_grad{i} - K\_ \* H\_grad{i};

F\_a(n\_1\*i+1 : n\_1\*(i+1), n\_1\*i+1 : n\_1\*(i+1)) = F - K\_ \* H;

end

K\_a(1:n\_1) = K\_;

for i = 1 : alpha

K\_a(n\_1\*i+1 : n\_1\*(i+1)) = K\_grad\_{i};

end

% 7 шаг

x\_a = F\_a \* x\_a + Psi\_a \* u;

sigma\_a = F\_a \* sigma\_a \* F\_a' + K\_a \* B \* K\_a';

end

% 8 шаг

P\_k1\_k = F \* P\_k \* F' + Gamma \* Q \* Gamma'; % 10 формула

for i = 1 : alpha

P\_k1\_k\_grad(i) = {F\_grad{i} \* P\_k \* F' + F \* P\_k\_k\_grad{i} \* F' ...

+ F \* P\_k \* F\_grad{i}' + Gamma\_grad{i} \* Q \* Gamma' + Gamma \* Q\_grad{i} \* Gamma' ...

+ Gamma \* Q \* Gamma\_grad{i}'};

end

B = H \* P\_k1\_k \* H' + R; % 12

for i = 1 : alpha

B\_grad(i) = {H\_grad{i} \* P\_k1\_k \* H' + H \* P\_k1\_k\_grad{i} \* H' ...

+ H \* P\_k1\_k \* H\_grad{i}' + R\_grad{i}};

end

K = P\_k1\_k \* H' / B; % 13

for i = 1 : alpha

K\_grad(i) = {(P\_k1\_k\_grad{i} \* H' + P\_k1\_k \* H\_grad{i}' ...

- P\_k1\_k \* H' / B \* B\_grad{i}) / B};

end

P\_k1 = (eye(n\_1) - K \* H) \* P\_k1\_k; % 15

for i = 1 : alpha

P\_k1\_k1\_grad(i) = {(eye(n\_1) - K \* H) \* P\_k1\_k\_grad{i} ...

- (K\_grad{i} \* H + K \* H\_grad{i}) \* P\_k1\_k};

end

% шаг 9

for i = 1 : alpha

for j = 1 : alpha

dM(i,j) = trace(H\_grad{i} \* Get\_C(n\_1, alpha, 0) \* (x\_a \* x\_a' + sigma\_a) \* Get\_C(n\_1, alpha,0)' \* H\_grad{j}' / B)...

+ trace(H\_grad{i} \* Get\_C(n\_1, alpha, 0) \* (x\_a \* x\_a' + sigma\_a) \* Get\_C(n\_1, alpha,j)' \* H' / B) ...

+ trace(H \* Get\_C(n\_1, alpha, i) \* (x\_a \* x\_a' + sigma\_a) \* Get\_C(n\_1, alpha,0)' \* H\_grad{j}' / B) ...

+ trace(H \* Get\_C(n\_1, alpha, i) \* (x\_a \* x\_a' + sigma\_a) \* Get\_C(n\_1, alpha,j)' \* H' / B) ...

+ 0.5 \* trace(B\_grad{i} / B \* B\_grad{j} / B);

end

end

M = M + dM;

P\_k = P\_k1;

P\_k\_k\_grad = P\_k1\_k1\_grad;

end

end

Файл main.m

N = 2;

% истинная тета

theta\_ist = [-1.5; 0.5];

M = calculate\_Fisher\_Matrix(theta\_ist, N);

disp(M);