

Министерство образования и науки Российской Федерации

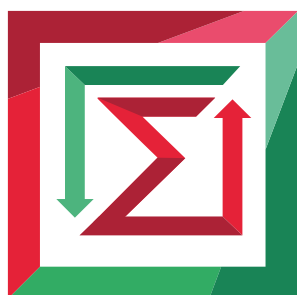
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**НГТУ
НЭТИ**

Кафедра теоретической и прикладной информатики

Лабораторная работа № 3
по дисциплине «Методы активной идентификации динамических систем»
**ВЫЧИСЛЕНИЕ ПРОИЗВОДНЫХ ИНФОРМАЦИОННОЙ МАТРИЦЫ ФИШЕРА
ПО КОМПОНЕНТАМ ВХОДНОГО СИГНАЛА**



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 производные ИМФ по компонентам входного сигнала.

2. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. В соответствии с выбранным уровнем сложности разработать программу вычисления производных ИМФ по компонентам входного сигнала. Проверить правильность работы программы на специально подготовленном тестовом примере.
2. Вычислить производные ИМФ по компонентам входного сигнала согласно варианту задания. Результаты привести в отчете.

3. Необходимые теоретические данные

Алгоритм вычисления значения критерия идентификации:

1. Определить $F, \Psi, \Gamma, H, Q, R, \bar{x}(t_0), P(t_0)$ и

$$\left\{ \frac{\partial F}{\partial \theta_\alpha}, \frac{\partial \Psi}{\partial \theta_\alpha}, \frac{\partial \Gamma}{\partial \theta_\alpha}, \frac{\partial H}{\partial \theta_\alpha}, \frac{\partial Q}{\partial \theta_\alpha}, \frac{\partial R}{\partial \theta_\alpha}, \frac{\partial \bar{x}(t_0)}{\partial \theta_\alpha}, \alpha = 1, 2, \dots, s \right\}, \text{ где}$$

F, Ψ, Γ, H, Q, R – матрицы модели, которые задаются в условиях задачи, $\bar{x}(t_0)$ – начальный вектор состояний, $P(t_0)$ – ковариационная матрица ошибок прогнозирования.

Сформировать матрицу Ψ_A по формуле (24) : $\Psi_A = \begin{bmatrix} \Psi \\ \frac{\partial \Psi}{\partial \theta_1} \\ \dots \\ \frac{\partial \Psi}{\partial \theta_s} \end{bmatrix}$.

2. Положить

$$\left\{ \frac{\partial M(u; \theta)}{\partial u_\alpha(t_\beta)} = 0, \beta = 0, \dots, N-1; \alpha = 1, 2, \dots, r \right\} k = 0; P(t_k | t_k) = P(t_0),$$

$$P(t_k | t_k) = P(t_0)$$

3. Определить $u(t_k)$.

4. Если $k = 0$, вычислить $\bar{x}_A(t_{k+1})$ по формуле (21) где

$$(21) \bar{x}_A(t_{k+1}) = \begin{cases} F_A(t_k)\bar{x}_A(t_k) + \Psi_A u(t_k), & \text{если } k = 1, 2, \dots, N-1; \\ \begin{bmatrix} F\bar{x}(t_0) + \Psi u(t_0) \\ \frac{\partial F}{\partial \theta_1} \bar{x}(t_0) + F \frac{\partial \bar{x}(t_0)}{\partial \theta_1} + \frac{\partial \Psi}{\partial \theta_1} u(t_0) \\ \dots \\ \frac{\partial F}{\partial \theta_s} \bar{x}(t_0) + F \frac{\partial \bar{x}(t_0)}{\partial \theta_s} + \frac{\partial \Psi}{\partial \theta_s} u(t_0) \end{bmatrix}, & \text{если } k = 0; \end{cases}$$

$$F_A(t_k) = \begin{bmatrix} F & O & \dots & O \\ \frac{\partial F}{\partial \theta_1} - \tilde{K}(t_k) \frac{\partial H}{\partial \theta_1} & F - \tilde{K}(t_k)H & \dots & O \\ \dots & \dots & \dots & \dots \\ \frac{\partial F}{\partial \theta_s} - \tilde{K}(t_k) \frac{\partial H}{\partial \theta_s} & O & \dots & F - \tilde{K}(t_k)H \end{bmatrix};$$

$$K_A(t_k) = \begin{bmatrix} \tilde{K}(t_k) \\ \frac{\partial \tilde{K}(t_k)}{\partial \theta_1} \\ \dots \\ \frac{\partial \tilde{K}(t_k)}{\partial \theta_s} \end{bmatrix};$$

$$\tilde{K}(t_k) = FK(t_k);$$

$$B(t_{k+1}) = HP(t_{k+1} | t_k) H^T + R;$$

$$K(t_{k+1}) = P(t_{k+1} | t_k) H^T B^{-1}(t_{k+1}),$$

после перейти на шаг 8.

5. Найти $\tilde{K}(t_k)$ по формуле (26)

$$(26) \tilde{K}(t_k) = FK(t_k)$$

6. Сформировать матрицу $F_A(t_k)$ по формуле (23)

$$F_A(t_k) = \begin{bmatrix} F & O & \dots & O \\ \frac{\partial F}{\partial \theta_1} - \tilde{K}(t_k) \frac{\partial H}{\partial \theta_1} & F - \tilde{K}(t_k)H & \dots & O \\ \dots & \dots & \dots & \dots \\ \frac{\partial F}{\partial \theta_s} - \tilde{K}(t_k) \frac{\partial H}{\partial \theta_s} & O & \dots & F - \tilde{K}(t_k)H \end{bmatrix}; \quad (23)$$

7. Вычислить $\bar{x}_A(t_{k+1})$ по формуле (21).

8. Найти $P(t_{k+1} | t_k)$, $B(t_{k+1})$, $K(t_{k+1})$, $P(t_{k+1} | t_{k+1})$ по формулам (10), (12), (13) и (15) соответственно.

$$(10) P(t_{k+1} | t_k) = FP(t_k | t_k)F^T + \Gamma Q \Gamma^T$$

$$(12) B(t_{k+1}) = HP(t_{k+1} | t_k) H^T + R$$

$$(13) K(t_{k+1}) = P(t_{k+1} | t_k) H^T B^{-1}(t_{k+1})$$

$$(15) P(t_{k+1} | t_{k+1}) = [I - K(t_{k+1})H] P(t_{k+1} | t_k).$$

9. Положить $\beta = 0$.

10. Если $k = 0$, вычислить $\left\{ \Psi_A \frac{\partial u(t_k)}{\partial u_\alpha(t_\beta)}, \frac{\partial \bar{x}_A(t_{k+1})}{\partial u_\alpha(t_\beta)}, \alpha = 1, 2, \dots, r \right\}$ при помощи равенств (32) и (33) и перейти на шаг 13.

$$(32) \Psi_A \frac{\partial u(t_k)}{\partial u_\alpha(t_\beta)} = \begin{cases} \Psi_A \begin{bmatrix} 0 \\ 0 \\ 1 \\ \dots \\ 0 \end{bmatrix} < -\alpha = \begin{bmatrix} (\Psi_A)_{1\alpha} \\ (\Psi_A)_{2\alpha} \\ (\Psi_A)_{3\alpha} \\ \dots \\ (\Psi_A)_{n(s+1)\alpha} \end{bmatrix}, & \text{если } \beta = k \\ 0, & \text{если } \beta \neq k \end{cases}$$

$$(33) \frac{\partial \bar{x}_A(t_{k+1})}{\partial u_\alpha(t_\beta)} = \begin{cases} F_A(t_k) \frac{\partial \bar{x}_A(t_k)}{\partial u_\alpha(t_\beta)} + \Psi_A \frac{\partial u(t_k)}{\partial u_\alpha(t_\beta)}, & \text{если } k = 1, 2, \dots, N-1; \\ \Psi_A \frac{\partial u(t_0)}{\partial u_\alpha(t_\beta)}, & \text{если } k = 0 \end{cases}$$

11. Сформировать $\left\{ \Psi_A \frac{\partial u(t_k)}{\partial u_\alpha(t_\beta)}, \alpha = 1, 2, \dots, r \right\}$ по формуле (33).

12. Вычислить $\left\{ \frac{\partial \bar{x}_A(t_{k+1})}{\partial u_\alpha(t_\beta)}, \alpha = 1, 2, \dots, r \right\}$ по формуле (32).

13. Используя выражение (31), получить $\left\{ \Delta \frac{\partial M(U, \theta)}{\partial u_\alpha(t_\beta)}, \alpha = 1, 2, \dots, r \right\}$, отвечающее текущему значению β и k .

$$(31) \Delta \frac{\partial M_{ij}(U, \theta)}{\partial u_\alpha(t_\beta)} = \sum_{k=0}^{N-1} \left\{ sp \left[\frac{\partial H}{\partial \theta_i} C_0 \left(\frac{\partial \bar{x}_A(t_{k+1})}{\partial u_\alpha(t_\beta)} \bar{x}_A^T(t_k) + \bar{x}_A(t_k) \frac{\partial \bar{x}_A^T(t_{k+1})}{\partial u_\alpha(t_\beta)} \right) C_0^T \frac{\partial H^T}{\partial \theta_j} B^{-1}(t_{k+1}) \right] + \right. \\ \left. + sp \left[\frac{\partial H}{\partial \theta_i} C_0 \left(\frac{\partial \bar{x}_A(t_{k+1})}{\partial u_\alpha(t_\beta)} \bar{x}_A^T(t_k) + \bar{x}_A(t_k) \frac{\partial \bar{x}_A^T(t_{k+1})}{\partial u_\alpha(t_\beta)} \right) C_j^T H^T B^{-1}(t_{k+1}) \right] \right. \\ \left. + sp \left[H C_j \left(\frac{\partial \bar{x}_A(t_{k+1})}{\partial u_\alpha(t_\beta)} \bar{x}_A^T(t_k) + \bar{x}_A(t_k) \frac{\partial \bar{x}_A^T(t_{k+1})}{\partial u_\alpha(t_\beta)} \right) C_0^T \frac{\partial H^T}{\partial \theta_j} B^{-1}(t_{k+1}) \right] \right. \\ \left. + sp \left[H C_j \left(\frac{\partial \bar{x}_A(t_{k+1})}{\partial u_\alpha(t_\beta)} \bar{x}_A^T(t_k) + \bar{x}_A(t_k) \frac{\partial \bar{x}_A^T(t_{k+1})}{\partial u_\alpha(t_\beta)} \right) C_j^T H^T B^{-1}(t_{k+1}) \right], i, j = 1, \dots, s \right\}$$

где матрица $C_i = [O, \dots, O, I, O, \dots, O]$,

где блок O, \dots, O имеет длину $i, i = 0, \dots, s$ и имеют размерность $n \times n(s+1)$, а вектор $\bar{x}_A(t_k)$ вычисляется по формуле (21).

14. Положить $\frac{\partial M(U; \theta)}{\partial u_\alpha(t_\beta)} = \frac{\partial M(U; \theta)}{\partial u_\alpha(t_\beta)} + \Delta \frac{\partial M(U, \theta)}{\partial u_\alpha(t_\beta)}, \alpha = 1, 2, \dots, r$.

15. Увеличить β на единицу. Если $\beta \leq N - 1$, перейти на шаг 10.
16. Увеличить k на единицу. Если $k \leq N - 1$, перейти на шаг 3, иначе закончить процесс.

4. Полученные результаты

Входные данные тестовой модели.

Исходя из указанной (для первого уровня сложности) модели:

$$\begin{aligned}x(t_{k+1}) &= \theta_1 x(t_k) + \theta_2 u(t_k) + w(t_k); \\y(t_{k+1}) &= x(t_{k+1}) + v(t_{k+1}), k = 0, 1.\end{aligned}$$

$$\begin{aligned}E[w(t_i)w(t_k)] &= 0.1\delta_{ik} = Q\delta_{ik} \\E[v(t_{i+1})v(t_{k+1})] &= 0.3\delta_{ik} = R\delta_{ik} \\E[x(t_0)] &= 0 = \bar{x}(t_0) \\E[x^2(t_0)] &= 0.1 = P(t_0). \\U^T[u(t_0)u(t_1)] &= [2, 2] \\\theta &= [\theta_1, \theta_2] = [1, 1].\end{aligned}$$

Можно получить следующие входные данные:

Матрицы для моделей состояния и измерения	Ковариационные матрицы шумов и начальные условия	θ^*, Ω_θ	U, Ω_U
$F = \theta_1,$ $\Psi = \theta_2,$ $\Gamma = 1,$ $H = 1$	$Q = 0.1, R = 0.3,$ $\bar{x}_0 = 0,$ $P_0 = 0.1$	$\theta^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$	$U^T = [2, 2],$

Результаты работы алгоритма:

```
>> main
  7.6923    8.4615
  8.4615   15.3846

      0    3.8462
  3.8462   12.3077
```

Требуемое значение:

$$\frac{\partial M(U; \theta)}{\partial u(t_0)} = \begin{pmatrix} 7.70 & 8.47 \\ 8.47 & 15.38 \end{pmatrix}; \quad \frac{\partial M(U; \theta)}{\partial u(t_1)} = \begin{pmatrix} 0 & 3.85 \\ 3.85 & 12.31 \end{pmatrix}.$$

Входные данные для заданного варианта:

Матрицы для моделей состояния и измерения	Ковариационные матрицы шумов и начальные условия	θ^*, Ω_θ	U, Ω_U
$F = \begin{bmatrix} -0.8 & 1 \\ \theta_1 & 0 \end{bmatrix},$ $\Psi = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$ $\Gamma = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$ $H = \begin{bmatrix} 1 & 0 \end{bmatrix}$	$Q = \theta_2, R = 0.1,$ $\bar{x}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$ $P_0 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$	$\theta^* = \begin{bmatrix} -1.5 \\ 0.5 \end{bmatrix},$ $\theta_1 \in [-2; -0.05],$ $\theta_2 \in [0.01; 0.8]$	$U^T = [3, \dots, 3],$ $u(t_k) \in [0; 5]$

Результаты работы алгоритма:

```
>> main
    2.3642      0
      0      0

    0      0
    0      0
```

5. Текст программы

Файл `init_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, 1];

    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_grad] = 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] =
    init_grad();

    n_1 = size(x0,1);
    n_2 = size(x0,2);
    alpha = length(theta);
    r = length(u);
    % выделение памяти
    M_grad = cell(r, N); %zeros(alpha, alpha);
    for i = 1: r
        for j = 1: N
            M_grad{i, j} = zeros(alpha, alpha);
        end
    end

    dM = cell(r, N); %zeros(alpha, alpha);
    for i = 1: r
        for j = 1: N
            dM{i, j} = zeros(alpha, alpha);
        end
    end
    dM_buf = zeros(alpha, alpha);

    x_a = zeros(n_1 * (alpha + 1), r);
    F_a = zeros(n_1 * (alpha + 1), n_1 * (alpha + 1));

    % сформировать матрицу пси(a) 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

    Psi_a_grad = cell(r, 1); %zeros(n_1 * (alpha + 1), r);
    for i = 1: r
        Psi_a_grad{i} = zeros(n_1 * (alpha + 1), r);
    end

    x_a_all = cell(N, N);
    x_a_grad = cell(r, 1); %zeros(n_1 * (alpha + 1), r);
    for i = 1: r
        x_a_grad{i} = zeros(n_1 * (alpha + 1), r);
    end
    for i = 1: N
        for j = 1: N
            x_a_all{i, j} = x_a_grad;
        end
    end
    % задание начальных условий 2 шаг
    P_k = P0;
```



```

for k = 0 : N - 1
    if (k == 0) % 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
    end

    % 5-7 шаги
    if (k ~= 0)
        % 5 шаг
        K_ = F * K;

        % 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

        % 7 шаг
        x_a = F_a * x_a + Psi_a * u(k);
    end
    % 8 шаг

    P_k1_k = F * P_k * F' + Gamma * Q * Gamma'; % 10 формула
    B = H * P_k1_k * H' + R; % 12
    K = P_k1_k * H' / B; % 13
    P_k1 = (eye(n_1) - K * H) * P_k1_k; % 15

for b = 0 : N - 1
    if (k == 0)
        if (b == k)
            for i = 1 : r
                u_g = zeros(r, 1);
                u_g(i) = 1;
                Psi_a_grad{i} = Psi_a * u_g;
            end
        end
        if (b ~= k)
            for i = 1 : r
                Psi_a_grad{i} = zeros(n_1 * (alpha + 1), r);
            end
        end

        for i = 1 : r
            x_a_grad{i} = Psi_a_grad{i};
        end
    end

    if (k ~= 0)
        if (b == k)
            for i = 1 : r
                u_g = zeros(r, 1);
                u_g(i) = 1;
            end
        end
    end
end

```

```

        Psi_a_grad{i} = Psi_a * u_g;
    end
end
if (b ~= k)
    for i = 1 : r
        Psi_a_grad{i} = zeros(n_1 * (alpha + 1), r);
    end
end
x_a_grad = x_a_all{k, b + 1};
for i = 1 : r
    x_a_grad{i} = F_a * x_a_grad{i} + Psi_a_grad{i};
end
end

for t = 1: r
    % шаг 9
    dM_buf = dM{t, b + 1};
    for i = 1 : alpha
        for j = 1 : alpha

            dM_buf(i,j) = trace(H_grad{i} * Get_C(n_1, alpha, 0) *
* (x_a_grad{t} * x_a' + x_a * x_a_grad{t}') * Get_C(n_1, alpha, 0)' *
H_grad{j})' / B) ...
            + trace(H_grad{i} * Get_C(n_1, alpha, 0) *
(x_a_grad{t} * x_a' + x_a * x_a_grad{t}') * Get_C(n_1, alpha, j)' * H' / B)
...
            + trace(H * Get_C(n_1, alpha, i) * (x_a_grad{t} *
x_a' + x_a * x_a_grad{t}') * Get_C(n_1, alpha, 0)' * H_grad{j})' / B) ...
            + trace(H * Get_C(n_1, alpha, i) * (x_a_grad{t} *
x_a' + x_a * x_a_grad{t}') * Get_C(n_1, alpha, j)' * H' / B);
        end
    end
    dM{t, b + 1} = dM_buf;
    M_grad{t, b + 1} = M_grad{t, b + 1} + dM{t, b + 1};
end
x_a_all{k + 1, b + 1} = x_a_grad;
end
P_k = P_k1;
end
end

```

Файл main.m

```

N = 2;
% истинная тета
theta_ist = [-1.5; 0.5];

M = calculate_Fisher_Matrix(theta_ist, N);
disp(M{1});
disp(M{2});

```