

Министерство образования и науки Российской Федерации

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

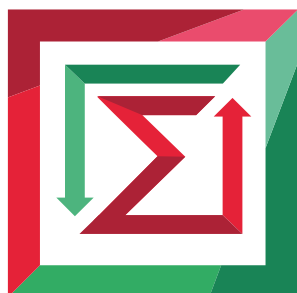


**НГТУ
НЭТИ**

Кафедра теоретической и прикладной информатики

Лабораторная работа № 1
по дисциплине «Методы активной идентификации динамических систем»

ОЦЕНИВАНИЕ НЕИЗВЕСТНЫХ ПАРАМЕТРОВ



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 применять метод максимального правдоподобия при оценивании неизвестных параметров моделей линейных дискретных систем.

2. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

В соответствии с выбранным уровнем сложности:

1. Разработать программы вычисления критерия идентификации и его градиента.
2. Разработать программу нахождения оценок максимального правдоподобия.
3. Следуя своему варианту задания, для указанных истинных значений параметров θ^* компьютерным моделированием получить последовательность из 30 измерений, соответствующую указанному входному сигналу. Используя полученные данные наблюдений, вычислить оценки максимального правдоподобия. Для ослабления зависимости результатов оценивания от выборочных данных, осуществить и обработать пять подобных идентификационных экспериментов, запоминая полученные результаты. Усреднив $\hat{\theta}$, найти $\hat{\theta}_{\text{ср}}$ и заполнить следующую таблицу:

Номер эксперимента	Значения оценок $\hat{\theta}_1$	Значения оценок $\hat{\theta}_2$
1		
2		
3		
4		
5		
$\hat{\theta}_{\text{ср}}$		

Вычислить относительную ошибку оценивания δ_{θ} в пространстве параметров по формуле $\delta_{\theta} = \frac{\|\theta^* - \hat{\theta}_{\text{ср}}\|}{\|\theta^*\|}$, где θ^* - вектор истинных значений параметров.

Отметим, что при решении реальных практических задач истинные значения параметров нам неизвестны. В связи с этим представляется целесообразным вычисление относительной ошибки оценивания δ_y в пространстве откликов. Для этого используется соотношение

$$\delta_Y = \frac{\|Y_{cp} - \hat{Y}_{cp}\|}{\|Y_{cp}\|}.$$

$$Y_{cp} = \{y_{cp}(t_{k+1}), k = 0, 1, \dots, N-1\}, \hat{Y}_{cp} = \{\hat{y}_{cp}(t_{k+1} | t_k), k = 0, 1, \dots, N-1\}$$

это усредненные по всем запускам последовательности измерений и их оценок. При этом $\hat{y}(t_{k+1} | t_k)$ находится при $\theta = \hat{\theta}$ в соответствии с равенством $\hat{y}(t_{k+1} | t_k) = H\hat{x}(t_{k+1} | t_k)$.

3. Необходимые теоретические данные

Алгоритм вычисления значения критерия идентификации:

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

2. Положить $\chi(\theta, \varepsilon) = Nmv \ln 2\pi; k = 0; P(t_k | t_k) = P(t_0)$, где

$$\chi(\theta, \varepsilon) = \frac{1}{2} \left\{ Nmv \ln 2\pi + \sum_{k=0}^{N-1} \left[v \ln \det B(t_{k+1}) + \frac{1}{2} \sum_{i=1}^q \sum_{j=1}^{k_i} (\varepsilon^{ij}(t_{k+1}))^T B^{-1}(t_{k+1}) \varepsilon^{ij}(t_{k+1}) \right] \right\}$$

это критерий идентификации, N – число измерений, m – размерность вектора измерений.

3. По формулам (10), (12), (13), (15) найти $P(t_{k+1} | t_k), B(t_{k+1}), K(t_{k+1}), P(t_{k+1} | t_{k+1})$, где

(10) $P(t_{k+1} | t_k) = FP(t_k | t_k)F^T + \Gamma Q \Gamma^T$ – ковариационная матрица ошибок одношагового прогнозирования,

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

4. Положить $\Delta = 0, i = 1$.

5. Определить $u^i(t_k)$.

6. Положить $j = 0$.

7. Выбрать $\hat{x}^{ij}(t_{k+1} | t_k) = \bar{x}(t_0)$, если $k = 0$.

8. Выбрать $\hat{x}^{ij}(t_{k+1} | t_k)$, $\varepsilon^{ij}(t_{k+1})$, $\hat{x}^{ij}(t_{k+1} | t_{k+1})$, при помощи соотношений (9), (11), (14), где
 - (9) $\hat{x}^{ij}(t_{k+1} | t_k) = F\hat{x}^{ij}(t_k | t_k) + \Psi u^i(t_k)$ – оценка одношагового прогнозирования состояния $x(t_{k+1})$, соответствующая паре (U_i, Y_{ij}) ,
 - (11) $\varepsilon^{ij}(t_{k+1}) = y^{ij}(t_{k+1}) - H\hat{x}^{ij}(t_{k+1} | t_k)$,
 - (14) $\hat{x}^{ij}(t_{k+1} | t_{k+1}) = \hat{x}^{ij}(t_{k+1} | t_k) + K(t_{k+1})\varepsilon^{ij}(t_{k+1})$ - оценка фильтрации состояния $x(t_{k+1})$, соответствующая паре (U_i, Y_{ij}) .
9. Положить $\Delta = \Delta + \left(\varepsilon^{ij}(t_{k+1})\right)^T B^{-1}(t_{k+1})\varepsilon^{ij}(t_{k+1})$, где
10. Увеличить j на единицу. Если $j \leq k_i$, перейти на шаг 7.
11. Увеличить i на единицу. Если $i \leq q$, перейти на шаг 5.
12. Положить $\chi(\theta, \Xi) = \chi(\theta, \Xi) + v \ln B(t_{k+1}) + \Delta$
13. Увеличить k на единицу. Если $k \leq N - 1$, перейти на шаг 3.
14. Положить $\chi(\theta, \Xi) = \frac{1}{2}\chi(\theta, \Xi)$ и закончить процесс.

Алгоритм вычисления градиентов критерия максимального правдоподобия:

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

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

2. Положить $\left\{ \frac{\partial \chi(\theta, \Xi)}{\partial \theta_\alpha} = 0, \alpha = 1, 2, \dots, s \right\}, k = 0$,

$$P(t_k | t_k) = P(t_0), \left\{ \frac{\partial P(t_k | t_k)}{\partial \theta_\alpha} = \frac{\partial P(t_0)}{\partial \theta_\alpha}, \alpha = 1, 2, \dots, s \right\}$$

где $\frac{\partial \chi(\theta, \Xi)}{\partial \theta_\alpha} = \sum_{k=0}^{N-1} \left\{ \frac{v}{2} Sp \left(B^{-1}(t_{k+1}) \frac{\partial B^{-1}(t_{k+1})}{\partial \theta_\alpha} \right) + \right.$

$$\sum_{i=1}^q \sum_{j=1}^{k_i} \left[\left(\frac{\partial \varepsilon^{ij}(t_{k+1})}{\partial \theta_\alpha} \right)^T B^{-1}(t_{k+1}) \varepsilon^{ij}(t_{k+1}) - \right.$$

$$\left. \left. \frac{1}{2} \left(\varepsilon^{ij}(t_{k+1}) \right)^T B^{-1}(t_{k+1}) \frac{\partial B^{-1}(t_{k+1})}{\partial \theta_\alpha} B^{-1}(t_{k+1}) \varepsilon^{ij}(t_{k+1}) \right] \right\} - \text{градиент}$$

 критерия идентификации.

3. По формулам (10), (12), (13), (15) найти $(t_{k+1} | t_k)$, $B(t_{k+1})$, $K(t_{k+1})$, $P(t_{k+1} | t_{k+1})$, где

(10) $P(t_{k+1} | t_k) = FP(t_k | t_k)F^T + \Gamma Q \Gamma^T$ - ковариационная матрица ошибок одношагового прогнозирования,

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

4. По формулам, следующим из (10), (12), (13), (15):

$$\begin{aligned} \frac{\partial P(t_{k+1} | t_k)}{\partial \theta_\alpha} &= \frac{\partial F}{\partial \theta_\alpha} P(t_k | t_k) F^T + F \frac{\partial P(t_k | t_k)}{\partial \theta_\alpha} F^T + FP(t_k | t_k) \frac{\partial F^T}{\partial \theta_\alpha} \\ &+ \frac{\partial \Gamma}{\partial \theta_\alpha} Q \Gamma^T + \Gamma \frac{\partial Q}{\partial \theta_\alpha} \Gamma^T + \Gamma Q \frac{\partial \Gamma^T}{\partial \theta_\alpha} \end{aligned}$$

$$\begin{aligned} \frac{\partial B(t_{k+1})}{\partial \theta_\alpha} &= \frac{\partial H}{\partial \theta_\alpha} P(t_{k+1} | t_k) H^T + H \frac{\partial P(t_{k+1} | t_k)}{\partial \theta_\alpha} H^T + \\ &+ HP(t_{k+1} | t_k) \frac{\partial H^T}{\partial \theta_\alpha} + \frac{\partial R}{\partial \theta_\alpha} \end{aligned}$$

$$\begin{aligned} \frac{\partial K(t_{k+1})}{\partial \theta_\alpha} &= \left[\frac{\partial P(t_{k+1} | t_k)}{\partial \theta_\alpha} H^T + P(t_{k+1} | t_k) \frac{\partial H^T}{\partial \theta_\alpha} \right. \\ &\quad \left. - P(t_{k+1} | t_k) H^T B^{-1}(t_{k+1}) \frac{\partial B^{-1}(t_{k+1})}{\partial \theta_\alpha} \right] B^{-1}(t_{k+1}) \end{aligned}$$

$$\begin{aligned} \frac{\partial P(t_{k+1} | t_{k+1})}{\partial \theta_\alpha} &= [I - K(t_{k+1})H] \frac{\partial P(t_{k+1} | t_k)}{\partial \theta_\alpha} - \\ &- \left[\frac{\partial K(t_{k+1})}{\partial \theta_\alpha} H + K(t_{k+1}) \frac{\partial H}{\partial \theta_\alpha} \right] P(t_{k+1} | t_k) \end{aligned}$$

$$\text{ВЫЧИСЛИТЬ } \left\{ \frac{\partial P(t_{k+1} | t_k)}{\partial \theta_\alpha}, \frac{\partial B(t_{k+1})}{\partial \theta_\alpha}, \frac{\partial K(t_{k+1})}{\partial \theta_\alpha}, \frac{\partial P(t_{k+1} | t_{k+1})}{\partial \theta_\alpha}, \alpha = 1, 2, \dots, s \right\}$$

5. Положить $\{\Delta_\alpha = 0, \alpha = 1, 2, \dots, s\}, i = 1$.

6. Определить $u^i(t_k)$.

7. Положить $j = 1$.

8. Выбрать $\hat{x}^{ij}(t_k | t_k) = \bar{x}(t_0), \left\{ \frac{\partial \hat{x}^{ij}(t_k | t_k)}{\partial \theta_\alpha} = \frac{\partial \bar{x}(t_0)}{\partial \theta_\alpha}, \alpha = 1, 2, \dots, s \right\}$, если $k = 0$.

9. Вычислить $\hat{x}^{ij}(t_{k+1} | t_k)$, $\varepsilon^{ij}(t_{k+1})$, $\hat{x}^{ij}(t_{k+1} | t_{k+1})$, при помощи соотношений (9), (11), (14), где

10. По формулам, следующим из (9), (11), (14):

$$\frac{\partial \hat{x}^{ij}(t_{k+1} | t_k)}{\partial \theta_\alpha} = \frac{\partial F}{\partial \theta_\alpha} \hat{x}^{ij}(t_k | t_k) + F \frac{\partial \hat{x}^{ij}(t_k | t_k)}{\partial \theta_\alpha} + \frac{\partial \Psi}{\partial \theta_\alpha} u^i(t_k);$$

$$\frac{\partial \varepsilon^{ij}(t_{k+1})}{\partial \theta_\alpha} = -\frac{\partial H}{\partial \theta_\alpha} \hat{x}^{ij}(t_{k+1} | t_k) - H \frac{\partial \hat{x}^{ij}(t_{k+1} | t_k)}{\partial \theta_\alpha};$$

$$\begin{aligned} \frac{\partial \hat{x}^{ij}(t_{k+1} | t_{k+1})}{\partial \theta_\alpha} &= \frac{\partial \hat{x}^{ij}(t_{k+1} | t_k)}{\partial \theta_\alpha} + \frac{\partial K(t_{k+1})}{\partial \theta_\alpha} \varepsilon^{ij}(t_{k+1}) + \\ &+ K(t_{k+1}) \frac{\partial \varepsilon^{ij}(t_{k+1})}{\partial \theta_\alpha} \end{aligned}$$

и найти $\left\{ \frac{\partial \hat{x}^{ij}(t_{k+1} | t_k)}{\partial \theta_\alpha}, \frac{\partial \varepsilon^{ij}(t_{k+1})}{\partial \theta_\alpha}, \frac{\partial \hat{x}^{ij}(t_{k+1} | t_{k+1})}{\partial \theta_\alpha}, \alpha = 1, 2, \dots, s \right\}$

11. Положить $\Delta_\alpha = \Delta_\alpha + \left(\frac{\partial \varepsilon^{ij}(t_{k+1})}{\partial \theta_\alpha} \right)^T B^{-1}(t_{k+1}) \varepsilon^{ij}(t_{k+1}) -$
 $-\frac{1}{2} \left(\varepsilon^{ij}(t_{k+1}) \right)^T B^{-1}(t_{k+1}) \frac{\partial B^{-1}(t_{k+1})}{\partial \theta_\alpha} B^{-1}(t_{k+1}) \varepsilon^{ij}(t_{k+1}),$
 $\alpha = 1, 2, \dots, s.$

12. Увеличить j на единицу. Если $j \leq k_i$, перейти на шаг 8.

13. Увеличить i на единицу. Если $i \leq q$, перейти на шаг 6.

14. Положить $\frac{\partial \chi(\theta, \varepsilon)}{\partial \theta_\alpha} = \frac{\partial \chi(\theta, \varepsilon)}{\partial \theta_\alpha} + \frac{v}{2} Sp \left(B^{-1}(t_{k+1}) \frac{\partial B^{-1}(t_{k+1})}{\partial \theta_\alpha} \right) + \Delta, \alpha =$
 $1, 2, \dots, s.$

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

4. Входные данные

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

5. Полученные результаты и их анализ

Результаты работы алгоритма вычисления значения критерия идентификации:

Номер эксперимента	Значения оценок $\hat{\theta}_1$	Значения оценок $\hat{\theta}_2$
1	-1.4996	0.8000
2	-1.5011	0.3601
3	-1.5011	0.1913
4	-1.4993	0.8000
5	-1.4992	0.4629
$\hat{\theta}_{cp}$	-1.5001	0.5229

Относительная ошибка оценивания $\delta_\theta = 0.0145$.

Относительная ошибка оценивания $\delta_Y = 5.0292e-04$.

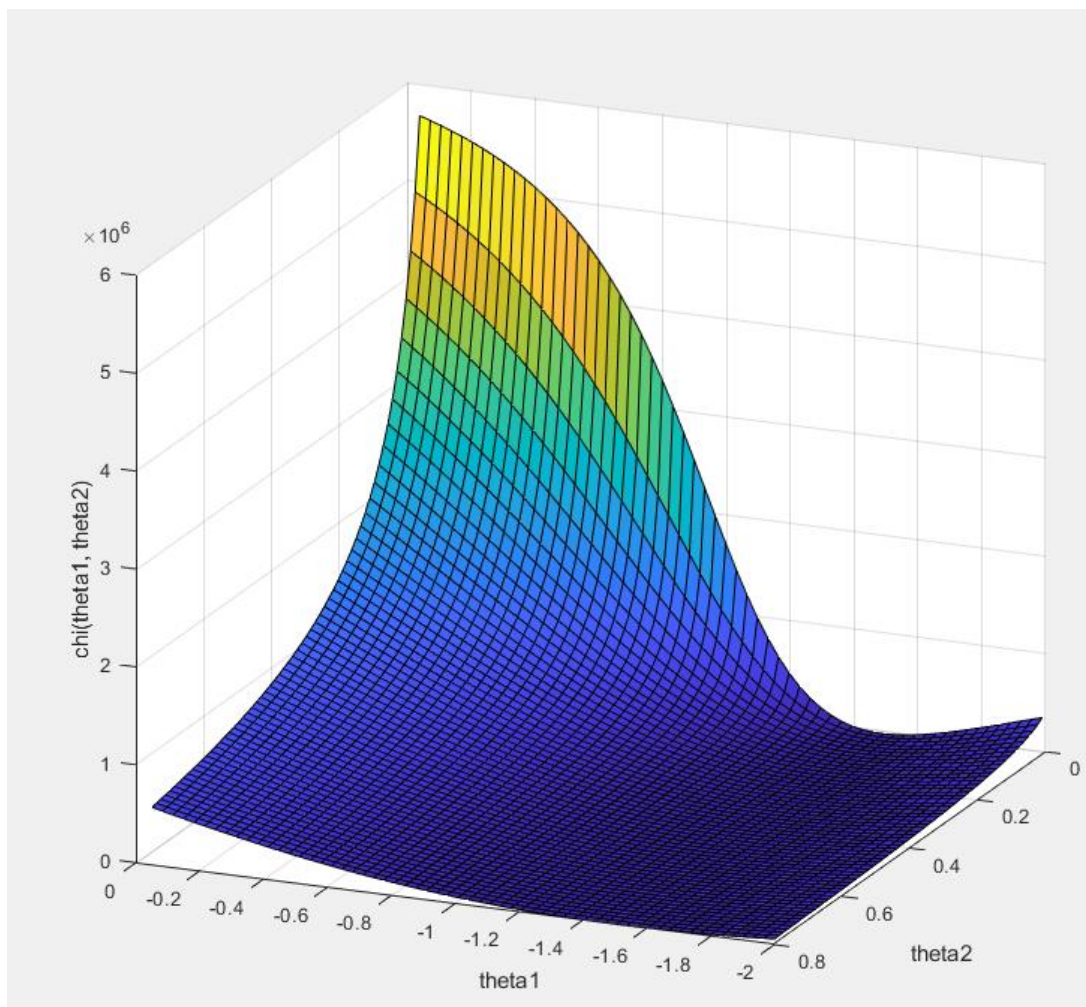
Результаты работы алгоритма вычисления градиентов критерия максимального правдоподобия:

Номер эксперимента	Значения оценок $\hat{\theta}_1$	Значения оценок $\hat{\theta}_2$
1	-1.4996	0.8000
2	-1.5011	0.3601
3	-1.5011	0.1913
4	-1.4993	0.8000
5	-1.4992	0.4629
$\hat{\theta}_{cp}$	-1.5001	0.5229

Относительная ошибка оценивания $\delta_\theta = 0.0145$.

Относительная ошибка оценивания $\delta_Y = 5.0292\text{e-}04$.

6. График оценок по $\hat{\theta}_1$ и $\hat{\theta}_2$



7. Текст программы

Файл initial_data.m

```
function [F, Psi, Gamma, H, Q, R, x_0, P_0, u, x, y] =  
initial_data(theta, N)  
  
    F = [-0.8, 1; theta(1), 0];  
    Psi = [1; 1];  
    Gamma = [1; 1];  
    H = [1, 0];  
  
    Q = theta(2);  
    R = 0.1;  
    x_0 = [0; 0];  
    P_0 = [0.1, 0; 0, 0.1];  
    u = 3;  
  
    w = sqrt(Q) * randn(30, 1);  
  
    v = sqrt(R) * randn(30, 1);  
  
    x = cell(30, 1);  
    y = zeros(30, 1);  
  
    x(1) = {F * x_0 + Psi * u + Gamma * w(1)};  
    y(1) = H * x{1} + v(1);  
  
    for k = 1 : 29  
        x(k+1) = {F * x{k} + Psi * u + Gamma * w(k+1)};  
        y(k+1) = H * x{k+1} + v(k+1);  
    end  
  
end
```

Файл algorithm_1.m

```
function [Hi] = algorithm_1(theta, y)  
  
    [F, Psi, Gamma, H, Q, R, x_0, P_0, u, ~, ~] = initial_data (theta,  
30);  
  
    m = 1;  
    v = 1;  
    N = length(y);  
    I = [1, 0;  
        0, 1];  
  
    Hi = N * m * log(2 * pi);  
  
    P_k_k = P_0;
```

```

x_k_k = x_0;
y_t = zeros(1,N); % для подсчета нормы

for k = 0 : N - 1

    P_k1_k = F * P_k_k * F' + Gamma * Q * Gamma';
    B_k1 = H * P_k1_k * H' + R;
    K_k1 = P_k1_k * H' / B_k1;
    P_k1_k1 = (I - K_k1 * H) * P_k1_k;

    delta = 0;
    for i = 1 : 1
        for j = 1 : 1
            x_k1_k = F * x_k_k + Psi * u;
            eps = y(k+1) - H * x_k1_k;
            x_k1_k1 = x_k1_k + K_k1 * eps;
            y_t(k+1) = H * x_k1_k1;
            delta = delta + eps' / B_k1 * eps;
        end
    end

    Hi = Hi + v * log(det(abs(B_k1))) + delta;

    P_k_k = P_k1_k1;
    x_k_k = x_k1_k1;
end

Hi = Hi / 2;
%normaY = norm(y - y_t')/norm(y)
end

```

Файл initial_gradient.m

```

function [F_grad, Psi_grad, Gamma_grad, H_grad, Q_grad, R_grad,
x0_grad, P0_grad] = initial_gradient(theta)

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 algorithm_2.m

```

function [Hi, Hi_grad] = algorithm_2(theta, y)

    N = length(y);
    alpha = length(theta);

    m = 1;
    v = 1;
    I = [1, 0;
         0, 1];

    [F, Psi, Gamma, H, Q, R, x0, P0, u, ~, ~] = initial_data(theta);
    [F_grad, Psi_grad, Gamma_grad, H_grad, Q_grad, R_grad, x0_grad,
    P0_grad] = initial_gradient(theta);

    Hi = N * m * v * log(2 * pi);

    Hi_grad = cell(alpha, 1);
    y_t = zeros(1,N); % для подсчета нормы
    for i = 1 : alpha
        Hi_grad{i} = 0;
    end

    P_k_k = P0;
    P_k_k_grad = P0_grad;

    P_k1_k_grad = cell(alpha, 1);
    B_grad = cell(alpha, 1);
    K_grad = cell(alpha, 1);
    P_k1_k1_grad = cell(alpha, 1);
    x_k1_k_grad = cell(alpha, 1);
    e_grad = cell(alpha, 1);

```

```

x_k1_k1_grad = cell(alpha, 1);

for k = 0 : N - 1
    P_k1_k = F * P_k_k * F' + Gamma * Q * Gamma';
    B_k1 = H * P_k1_k * H' + R;
    K_k1 = P_k1_k * H' / B_k1;
    P_k1_k1 = (I - K_k1 * H) * P_k1_k;

    for a = 1 : alpha
        P_k1_k_grad{a} = F_grad{a} * P_k_k * F' + F *
        P_k_k_grad{a} * F' + F * P_k_k * F_grad{a}' + Gamma_grad{a} * Q *
        Gamma' + Gamma * Q_grad{a} * Gamma' + Gamma * Q * Gamma_grad{a}';
        B_grad{a} = H_grad{a} * P_k1_k * H' + H * P_k1_k_grad{a} *
        H' + H * P_k1_k * H_grad{a}' + R_grad{a};
        K_grad{a} = (P_k1_k_grad{a} * H' + P_k1_k * H_grad{a}' -
        P_k1_k * H' / B_k1 * B_grad{a}) / B_k1;
        P_k1_k1_grad{a} = (I - K_k1 * H) * P_k1_k_grad{a} -
        (K_grad{a} * H + K_k1 * H_grad{a}) * P_k1_k;
    end

    delta = 0;
    delta_grad = cell(alpha, 1);
    for a = 1 : alpha
        delta_grad(a) = {0};
    end

    for i = 1 : 1 % шаг 5
        for j = 1 : 1 % шаг 8
            if (k == 0)
                x_k_k = x0;
                x_k_k_grad = x0_grad;
            end

            x_k1_k = F * x_k_k + Psi * u; % 9
            eps = y(k+1) - H * x_k1_k; % 11
            x_k1_k1 = x_k1_k + K_k1 * eps; % 14
            y_t(k+1) = H * x_k1_k1;
            delta = delta + eps' / B_k1 * eps; % для хи

            for a = 1 : alpha % 10 шаг
                x_k1_k_grad{a} = F_grad{a} * x_k_k + F *
                x_k_k_grad{a} + Psi_grad{a} * u;
                e_grad{a} = - H_grad{a} * x_k1_k - H *
                x_k1_k_grad{a};
                x_k1_k1_grad{a} = x_k1_k_grad{a} + K_grad{a} * eps
                + K_k1 * e_grad{a};
            end

            for a = 1 : alpha
                delta_grad{a} = delta_grad{a} + e_grad{a}' / B_k1
                * eps - 0.5 * eps' / B_k1 * B_grad{a} / B_k1 * eps;
            end
        end
    end

    for a = 1 : alpha

```

```

        Hi_grad{a} = Hi_grad{a} + v / 2 * trace(B_k1 \ B_grad{a})
+ delta_grad{a};
    end

    P_k_k = P_k1_k1;
    P_k_k_grad = P_k1_k1_grad;

    x_k_k = x_k1_k1;
    x_k_k_grad = x_k1_k1_grad;

    Hi = Hi + v * log(det(abs(B_k1))) + delta; % вне алгоритма
градиента, но сама функция нужна
    end

    Hi = Hi / 2;
    Hi_grad = [Hi_grad{1} ; Hi_grad{2}];
    norma_grad_Y = norm(y - y_t')/norm(y);
end

```

Файл main.m

```

N = 30;
t_n = 5;

theta_ist = [-1.5; 0.5];

theta_1 = zeros(2,t_n);
theta_2 = zeros(2,t_n);
rng(0, 'twister');

for i = 1 : t_n
    [ ~, ~, ~, ~, ~, ~, ~, ~, ~, ~, X, Y] = initial_data(theta_ist, N);

    opt = optimoptions('fmincon', 'Algorithm', 'interior-point');
    fun = @(x) algorithm_1(x, Y);
    theta_1(:,i) = fmincon(fun, [-1.25; 0.2], [], [], [], [], [-2;
0.01], [-0.05; 0.8], [], opt);

    opt = optimoptions('fmincon', 'Algorithm', 'interior-point',
'SpecifyObjectiveGradient',true, 'GradObj','on');
    fun_grad = @(x) algorithm_2(x, Y);
    theta_2(:,i) = fmincon(fun_grad, [-1.25; 0.2], [], [], [], [], [-
2; 0.01], [-0.05; 0.8], [], opt);
end

disp(theta_1)
disp(theta_2)
m_1 = mean(theta_1, 2)
m_2 = mean(theta_2, 2)
theta_disp1 = norm(theta_ist - m_1) / norm(theta_ist)
theta_disp2 = norm(theta_ist - m_2) / norm(theta_ist)

f = @(x,y) algorithm_2([x; y], Y);

```

```
xx = linspace(-2, -0.05, 60);
yy = linspace(0.01, 0.8, 60);
a = zeros(length(xx), length(yy));

for i = 1 : length(xx)
    for j = 1 : length(yy)
        a(j, i) = f(xx(i), yy(j));
    end
end

surf(xx, yy, a);
hold on
xlabel('theta1'), ylabel('theta2'), zlabel('chi(theta1, theta2)');
```