

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
(Университет ИТМО)

Факультет систем управления и робототехники

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1  
по дисциплине  
*«Теория оптимального управления»*

Студент:

*Группа № R3435*

*Зыкин Л. В.*

Предподаватель:

*ведущий научный сотрудник, доцент*

*Парамонов А. В.*

Санкт-Петербург  
2025

# 1 ЛАБОРАТОРНАЯ РАБОТА №1. СТАТИЧЕСКИЕ МЕТОДЫ ОПТИМИЗАЦИИ

## Постановка задачи

Требуется найти минимум критерия качества для статической задачи оптимизации

$$J(x,u) = 5x^2 + 2u^2 + 3xu + 4x + u - 5,$$

а также выполнить поиск экстремума градиентными методами. Вариант: **13**.  
Для пунктов с ограничениями используется функция ограничения

$$c(x,u) = 4x^2 + u + 1.$$

## 1.1 Поиск глобального минимума на основе необходимых и достаточных условий

### 1.1.1 Случай без ограничений

Найдём стационарную точку, приравняв к нулю градиент функции  $J(x,u)$ :

$$\nabla J(x,u) = \begin{bmatrix} \frac{\partial J}{\partial x} \\ \frac{\partial J}{\partial u} \end{bmatrix} = \begin{bmatrix} 10x + 3u + 4 \\ 3x + 4u + 1 \end{bmatrix} = \mathbf{0}.$$

Получаем систему линейных уравнений

$$\begin{cases} 10x + 3u + 4 = 0, \\ 3x + 4u + 1 = 0. \end{cases}$$

Решая её, из второго уравнения выразим  $x = \frac{-1 - 4u}{3}$  и подставим в первое:

$$\frac{-10 - 40u}{3} + 3u + 4 = 0 \Rightarrow -10 - 40u + 9u + 12 = 0 \Rightarrow 2 - 31u = 0,$$

откуда

$$u^* = \frac{2}{31}, \quad x^* = -\frac{13}{31}.$$

Проверим достаточное условие минимума по положительной определённости матрицы Гессе:

$$H = \begin{bmatrix} 10 & 3 \\ 3 & 4 \end{bmatrix}, \quad \Delta_1 = 10 > 0, \quad \det H = 10 \cdot 4 - 3^2 = 31 > 0.$$

Следовательно,  $H$  положительно определена, а найденная стационарная точка является **единственным глобальным минимумом** квадратичной функции  $J$ .

Значение критерия в точке минимума:

$$\begin{aligned} J(x^*, u^*) &= 5\left(\frac{13}{31}\right)^2 + 2\left(\frac{2}{31}\right)^2 + 3\left(-\frac{13}{31} \cdot \frac{2}{31}\right) \\ &\quad + 4\left(-\frac{13}{31}\right) + \frac{2}{31} - 5 = -\frac{5580}{961} \approx -5.804. \end{aligned}$$

### 1.1.2 С ограничением вида равенства $c(x, u) = 0$

Пусть ограничение имеет вид  $c(x, u) = 4x^2 + u + 1 = 0$ . Используем метод Лагранжа для минимизации  $J(x, u)$  при  $c(x, u) = 0$ . Рассмотрим функцию Лагранжа

$$\mathcal{L}(x, u, \lambda) = J(x, u) + \lambda c(x, u) = 5x^2 + 2u^2 + 3xu + 4x + u - 5 + \lambda(4x^2 + u + 1).$$

Необходимые условия первого порядка:

$$\frac{\partial \mathcal{L}}{\partial x} = 10x + 3u + 4 + 8\lambda x = 0, \quad \frac{\partial \mathcal{L}}{\partial u} = 4u + 3x + 1 + \lambda = 0, \quad c(x, u) = 0.$$

Исключая  $u$  из ограничения, имеем  $u = -1 - 4x^2$ . Подстановка в условия даёт кубическое уравнение для  $x$ :

$$128x^3 - 36x^2 + 34x + 1 = 0.$$

Его действительный корень (единственный с учётом строгой выпуклости  $J$ ) равен

$$x_{\text{--}}^* \approx -0.02847, \quad u_{\text{--}}^* = -1 - 4(x_{\text{--}}^*)^2 \approx -1.00324.$$

Поскольку матрица Гессе  $H \succ 0$ ,  $J$  строго выпукла, то найденная стационарная точка на гладком многообразии  $c = 0$  является единственным глобальным минимумом на множестве  $\{c = 0\}$ . Значение функционала

$$J(x_{\text{--}}^*, u_{\text{--}}^*) \approx -4.014.$$

Для численной проверки и воспроизводимости ниже приведён короткий скрипт, который решает условия ККТ и вычисляет значения:

Листинг 1.1 — Решение условий ККТ для равенства и проверка

```
# lab1/python/task1_kkt.py
import sympy as sp
x,u,lam = sp.symbols('x u lam', real=True)
J = 5*x**2 + 2*u**2 + 3*x*u + 4*x + u - 5
c = 4*x**2 + u + 1
L = J + lam*c
sol = sp.nsolve([
    sp.diff(L,x),
    sp.diff(L,u),
    c
],[x,u,lam], (-0.03, -1.0, 3.0))
print({"x": float(sol[0]), "u": float(sol[1]), "lambda": float(
    sol[2]), "J": float(J.subs({x:sol[0],u:sol[1]}))})
```

### 1.1.3 С ограничением вида неравенства $c(x,u) \leq 0$

Сначала проверим допустимость найденного безусловного минимума  $(x^*, u^*) = \left(-\frac{13}{31}, \frac{2}{31}\right)$ :

$$c(x^*, u^*) = 4(x^*)^2 + u^* + 1 = 4\left(\frac{13}{31}\right)^2 + \frac{2}{31} + 1 > 0,$$

то есть точка  $(x^*, u^*)$  **не** удовлетворяет  $c \leq 0$ . Следовательно, минимум на множестве  $\{c \leq 0\}$  достигается на границе  $c = 0$ . Условия ККТ принимают вид

$$\nabla J(x,u) + \lambda \nabla c(x,u) = 0, \quad c(x,u) \leq 0, \quad \lambda \geq 0, \quad \lambda c(x,u) = 0.$$

Найденная в подп. 1.2 точка на границе даёт множитель

$$\lambda^* = 16(x_{\leq}^*)^2 - 3x_{\leq}^* + 3 \approx 3.10 > 0,$$

что удовлетворяет  $\lambda^* \geq 0$ . Поэтому решение п.1.2 одновременно является решением задачи с неравенством:

$$(x_{\leq}^*, u_{\leq}^*) = (x_{=}^*, u_{=}^*), \quad J(x_{\leq}^*, u_{\leq}^*) \approx -4.014.$$

## 1.2 Градиентные методы поиска минимума $J_1(x,u) = J(x,u)$

### 1.2.1 Метод Ньютона–Рафсона (пошаговый расчёт)

Для квадратичной функции  $J$  матрица Гессе постоянна:  $H = \begin{bmatrix} 10 & 3 \\ 3 & 4 \end{bmatrix}$ .

Метод Ньютона даёт итерацию

$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ u_k \end{bmatrix} - H^{-1} \nabla J(x_k, u_k), \quad H^{-1} = \frac{1}{31} \begin{bmatrix} 4 & -3 \\ -3 & 10 \end{bmatrix}.$$

Начнём с  $(x_0, u_0) = (0, 0)$ . Тогда  $\nabla J(0, 0) = (4, 1)^\top$  и шаг Ньютона

$$\Delta_0 = H^{-1} \nabla J(0, 0) = \frac{1}{31} \begin{bmatrix} 13 \\ -2 \end{bmatrix}, \quad (x_1, u_1) = (0, 0) - \Delta_0 = \left( -\frac{13}{31}, \frac{2}{31} \right).$$

Это в точности аналитический минимум без ограничений, поэтому метод Ньютона сошёлся **за одну итерацию**. Значение функционала совпадает с ранее полученным:  $J(x_1, u_1) = -\frac{5580}{961}$ .

### 1.2.2 Метод наискорейшего спуска для двух $\gamma$ (пошагово)

Итерация градиентного спуска:

$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ u_k \end{bmatrix} - \gamma \nabla J(x_k, u_k), \quad \nabla J(x, u) = \begin{bmatrix} 10x + 3u + 4 \\ 3x + 4u + 1 \end{bmatrix}.$$

Собственные значения  $H$ :  $\lambda_{1,2} = 7 \pm 3\sqrt{2} \approx (11.243, 2.757)$ . Для сходимости с постоянным шагом требуется  $0 < \gamma < \frac{2}{\lambda_{\max}} \approx 0.1779$ . Знакопеременные (колебательные) траектории возникают, если  $\frac{1}{\lambda_i} < \gamma < \frac{2}{\lambda_i}$  для некоторых мод (в частности, для  $\lambda_{\max}$ : интервал  $(0.089, 0.178)$ ).

Рассмотрим два случая при старте  $(x_0, u_0) = (0, 0)$ :

1. **Колебательная сходимость:** возьмём  $\gamma_1 = 0.12 \in (1/\lambda_{\max}, 2/\lambda_{\max})$ .

Первые шаги:

$$(x_1, u_1) = (0, 0) - 0.12 (4, 1) = (-0.48, -0.12), \quad J_1 \approx -3.278.$$

$$\nabla J(x_1, u_1) = (10(-0.48) + 3(-0.12) + 4, 3(-0.48) + 4(-0.12) + 1) =$$

$$= (-1.16, -0.04).$$

$$(x_2, u_2) = (x_1, u_1) - 0.12(-1.16, -0.04) = (-0.3408, -0.1152), \quad J_2 \approx -3.716$$

Дальнейшие итерации дают знакопеременные приращения по «жёсткому» направлению, но монотонное убывание  $J$  до  $(x^*, u^*)$ .

2. **Апериодическая (монотонная) сходимость:** возьмём  $\gamma_2 = 0.05$ .

Первые шаги:

$$(x_1, u_1) = (0, 0) - 0.05(4, 1) = (-0.20, -0.05), \quad J_1 \approx -2.125.$$

$$\nabla J(x_1, u_1) = (10(-0.2) + 3(-0.05) + 4, 3(-0.2) + 4(-0.05) + 1) = (1.85, 0.3).$$

$$(x_2, u_2) = (-0.20, -0.05) - 0.05(1.85, 0.3) = (-0.2925, -0.065), \quad J_2 \approx -2.62$$

При таком шаге траектория не меняет знак по собственным направлениям и плавно сходится к минимуму.

Для воспроизводимых вычислений см. листинг с автоматическим построением траекторий для обоих шагов.

Листинг 1.2 — Gradient descent: two step sizes and first iterations

```
# lab1/python/task1_gradients.py
import numpy as np

H = np.array([[10.0, 3.0], [3.0, 4.0]])
b = np.array([4.0, 1.0]) # grad J = H @ w + b, w = [x, u]
w_star = np.linalg.solve(H, -b) # analytic minimizer

def grad(w):
    return H @ w + b

def run_gd(gamma, steps=6):
    w = np.array([0.0, 0.0])
    hist = []
    for k in range(steps):
        J = 0.5 * w @ (H @ w) + b @ w - 5 # constant shift does
            not affect trajectory
        hist.append((k, *w, J))
        w = w - gamma * grad(w)
    return np.array(hist)

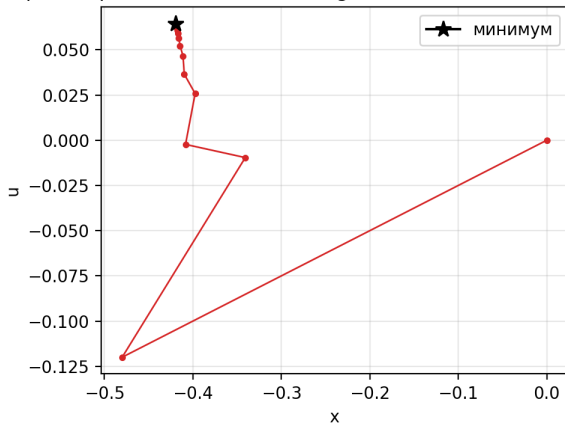
if __name__ == "__main__":
```

```

for gamma in [0.12, 0.05]:
    Hs = run_gd(gamma)
    print(f"gamma={gamma}")
    for row in Hs:
        k,x,u,J=row
        print(int(k), x, u, J)
    print("=> distance to optimum:", np.linalg.norm(Hs
        [-1,1:3]-w_star))

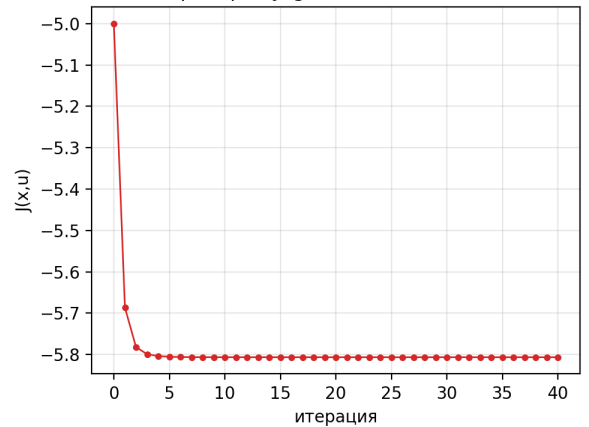
```

Траектория в плоскости (x,u), gamma=0.12 (колебате



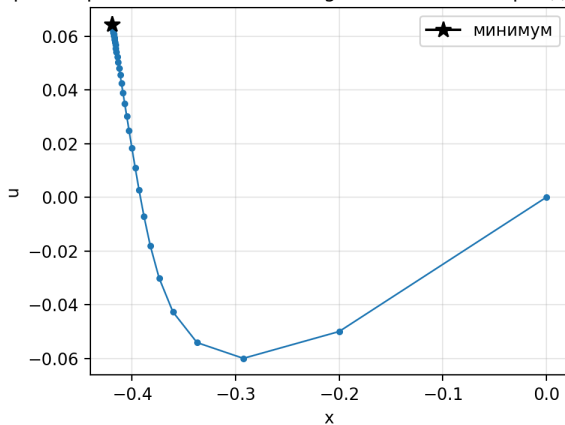
(a) Trajectory (x,u), gamma=0.12

Убывание критерия J, gamma=0.12 (колебательная)



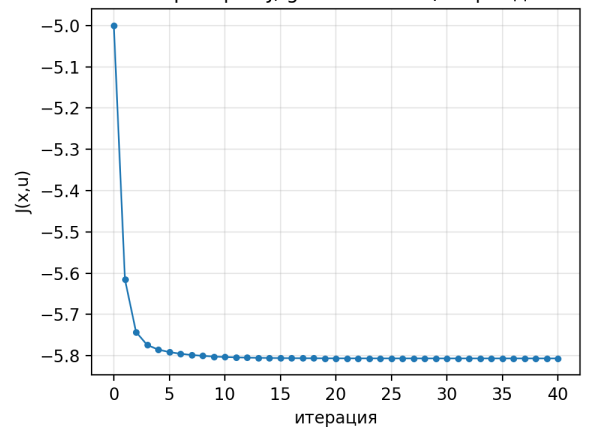
(б) J vs iterations, gamma=0.12

Траектория в плоскости (x,u), gamma=0.05 (апериодич



(в) Trajectory (x,u), gamma=0.05

Убывание критерия J, gamma=0.05 (апериодическа



(г) J vs iterations, gamma=0.05

Рисунок 1 — Gradient descent: trajectories and J decrease for two step sizes